
APACHE
nifi
Introducción
THE
APACHE
SOFTWARE FOUNDATION

Introducción

Apache NiFi es un sistema de flujo de datos basado en conceptos de "programación basada en flujo".

Es compatible con el enrutamiento de datos, la transformación y la lógica de mediación del sistema potentes y escalables basados en gráficos.

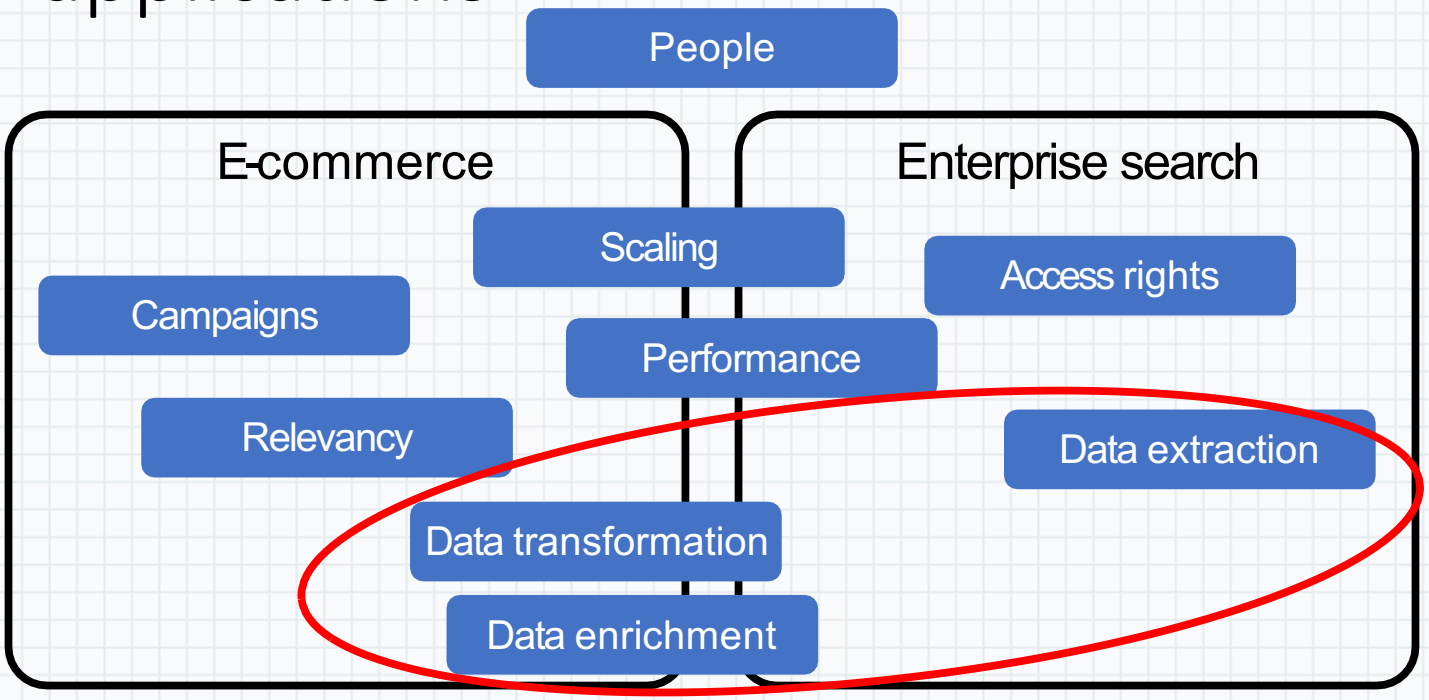
El NiFi tiene una interfaz de usuario basada en la web para diseñar, controlar, retroalimentar y monitorear flujos de datos.

Introducción

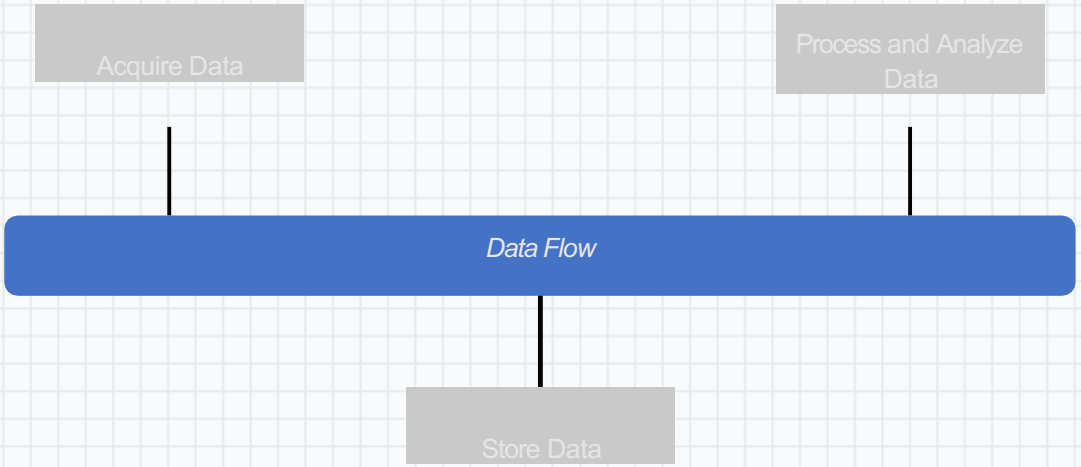
Es altamente configurable en múltiples dimensiones de calidad de servicio, como tolerancia a fallas frente a entrega garantizada, baja latencia frente a alto rendimiento y colas basadas en prioridades.

NiFi proporciona datos detallados para todos los datos recibidos, bifurcados, clonados, modificados, enviados y finalmente descartados al alcanzar su estado configurado final.

Challenges for search applications

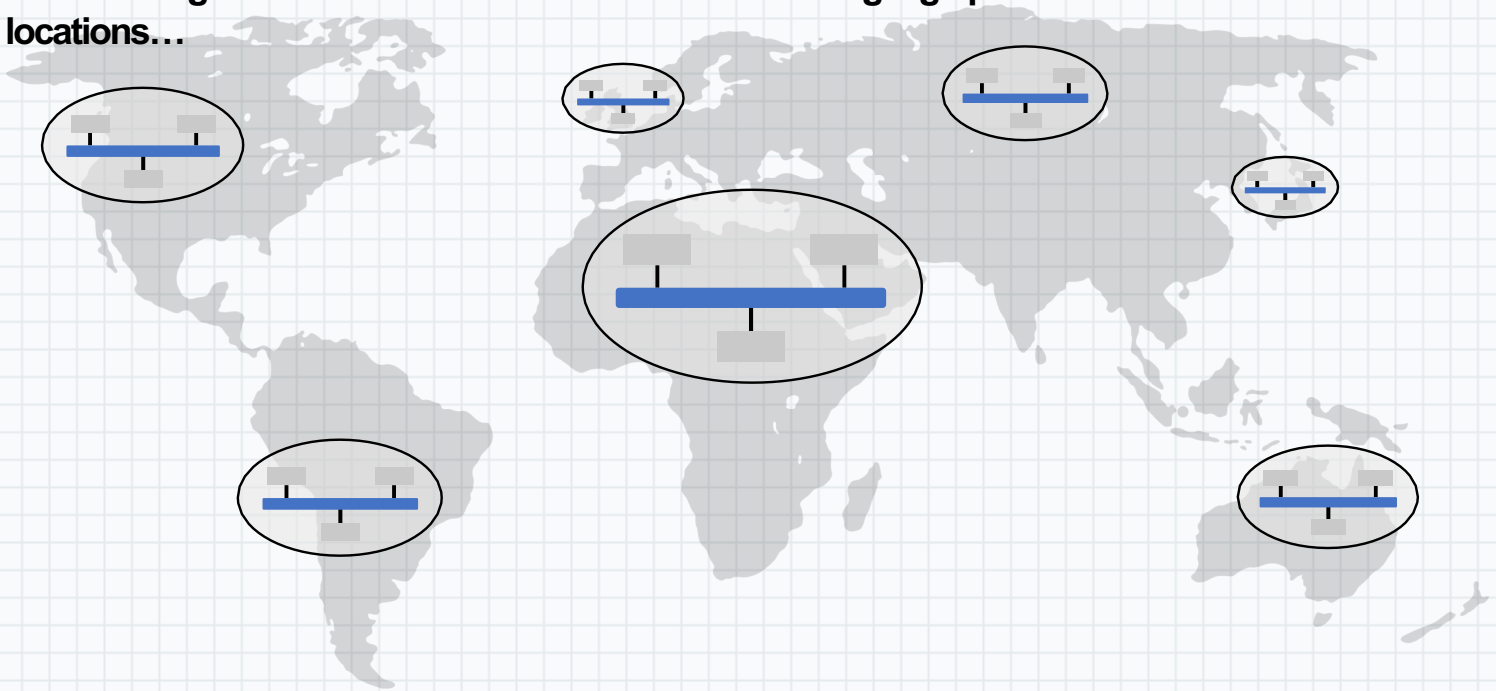


Simplistic View of Enterprise Data Flow



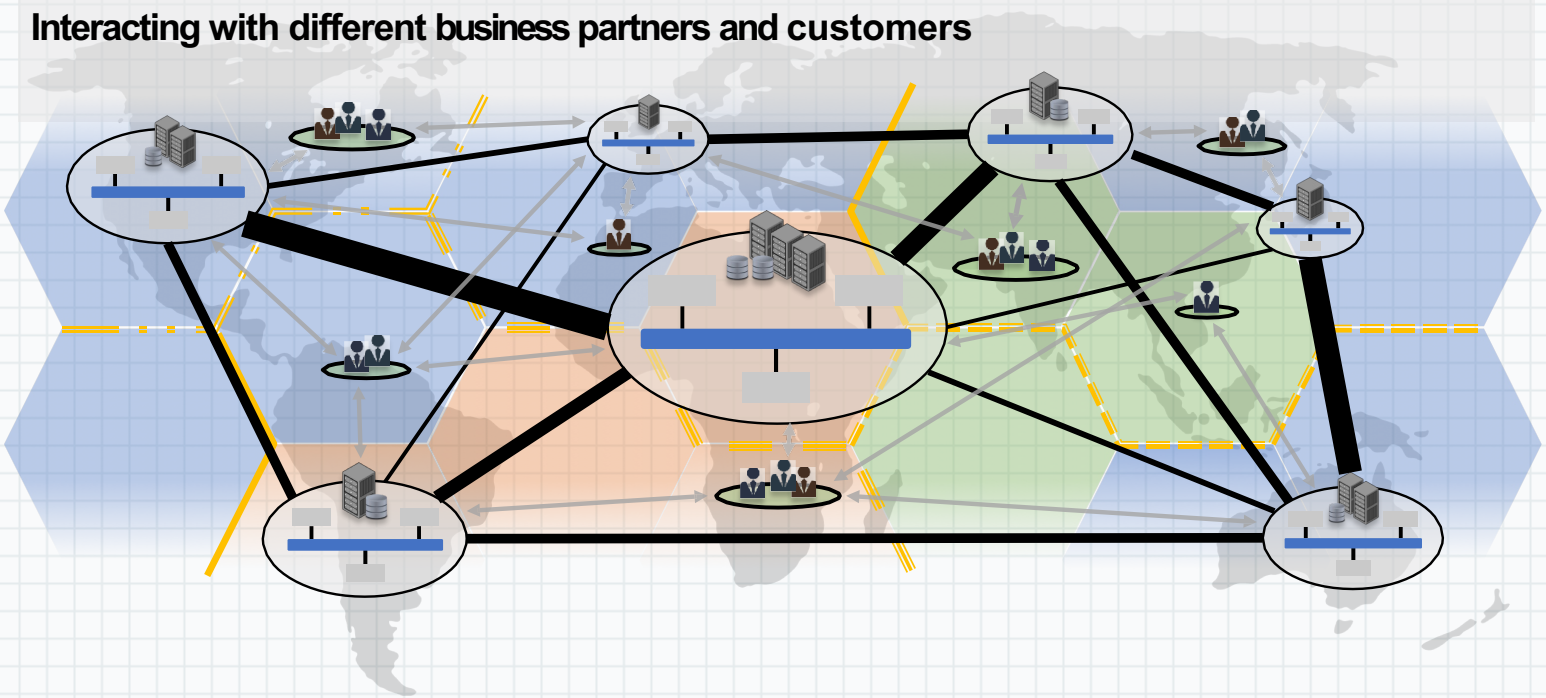
Realistic View of Enterprise Data Flow

Different organizations/business units across different geographic locations…



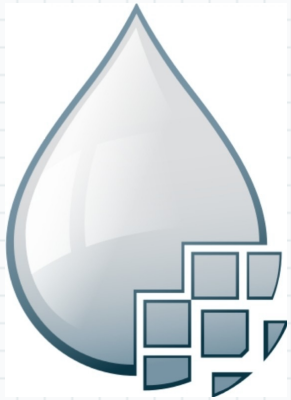
Realistic View of Enterprise Data Flow

Interacting with different business partners and customers



Apache NiFi

- Created to address the challenges of global enterprise dataflow
- Key features:
  - **Visual Command and Control**
  - **Data Lineage (Provenance)**
  - **Data Prioritization**
  - **Data Buffering/Back-Pressure**
  - **Control Latency vs. Throughput**
  - **Secure Control Plane / Data Plane**
  - **Scale Out Clustering**
  - **Extensibility**



flt«tftt
HORTONWORKs

Apache NiFi

What is Apache NiFi used for?

- Reliable and secure transfer of data between systems
- Delivery of data from sources to analytic platforms
- Enrichment and preparation of data:
  - Conversion between formats
  - Extraction/Parsing
  - Routing decisions

What is Apache NiFi NOT used for?

- Distributed Computation
- Complex Event Processing
- Joins / Complex Rolling Window Operations

Terminología

Para hablar de NiFi, hay algunos términos clave que los usuarios deben conocer. Explicaremos esos términos específicos de NiFi aquí, en un alto nivel.

Administrador de flujo de datos: un administrador de flujo de datos (DFM) es un usuario de NiFi que tiene permisos para agregar, eliminar y modificar componentes de un flujo de datos de NiFi.

Terminology

FlowFile

- Unit of data moving through the system
- Content + Attributes (key/value pairs)

Processor

- Performs the work, can access FlowFiles

Connection

- Links between processors
- Queues that can be dynamically prioritized

Process Group

- Set of processors and their connections
- Receive data via input ports, send data via output ports

Features

- **Interfaz de usuario basada en web**
  - **Experiencia perfecta entre diseño, control, retroalimentación y monitoreo**
- **altamente configurable**
  - **Entrega tolerante a fallas versus entrega garantizada**
  - **Baja latencia versus alto rendimiento**
  - **priorización dinámica**
- **El flujo se puede modificar en tiempo de ejecución**
- **contrapresión**

Features

- **Fecha Procedencia**
  - **Siga el flujo de datos de principio a fin**
- **Diseñado para la extensión**
  - **Cree sus propios procesadores y más**
  - **Permite un desarrollo rápido y pruebas efectivas**
- **Seguro**
  - **SSL, SSH, HTTPS, contenido encriptado, etc...**
  - **Autorización multiusuario y autorización interna/gestión de políticas**

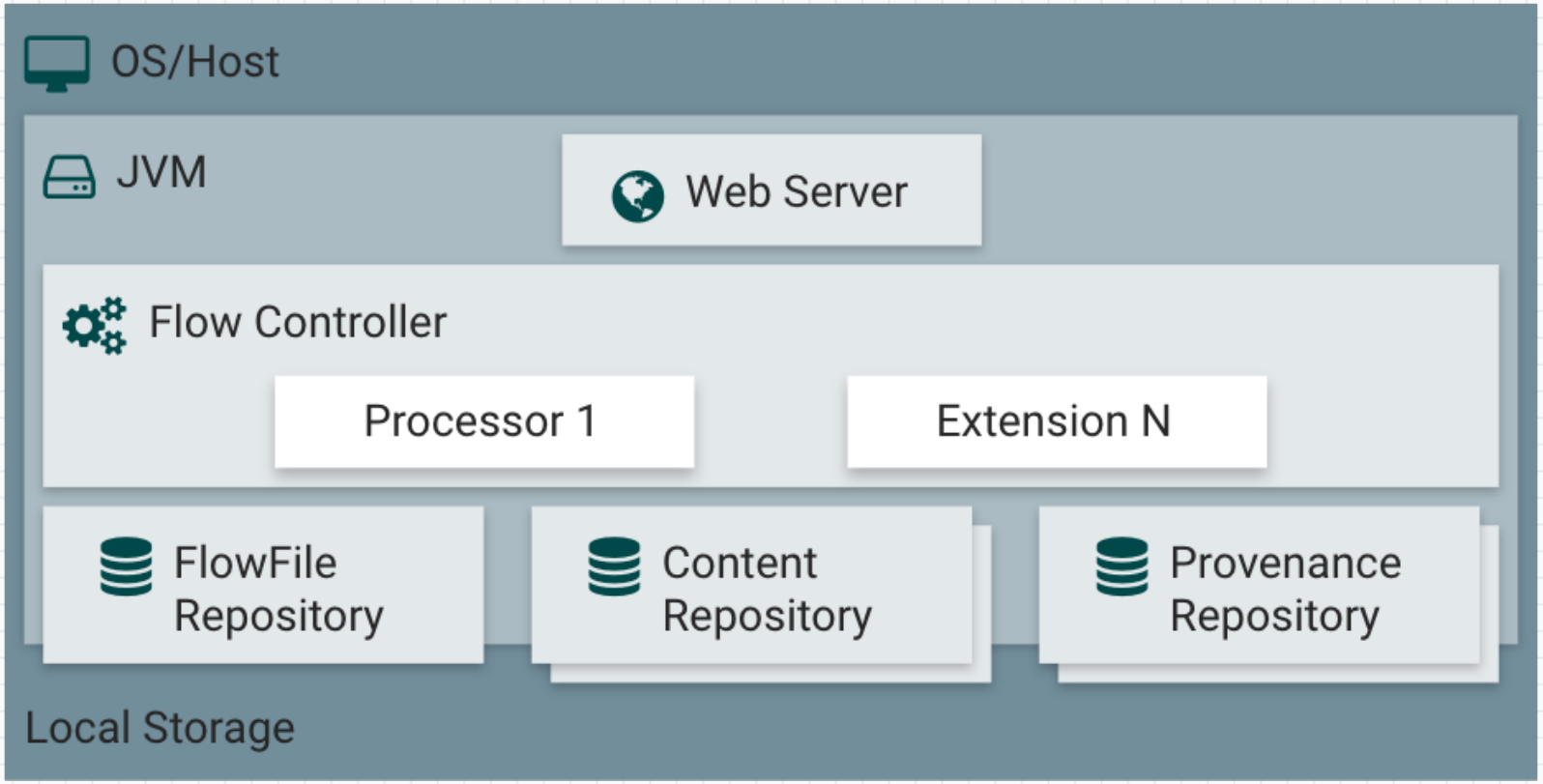
Carácteristicas

- Permite la creación visual y la gestión de gráficos controlados por procesador
- Es intrínsecamente asíncrono, lo que permite un rendimiento muy alto y una amortiguación natural, incluso cuando las tasas de procesamiento y rendimiento fluctúan.
- Proporciona un modelo altamente concurrente sin que un desarrollador tenga que preocuparse por las complejidades típicas de concurrencia
- Promueve el desarrollo de componentes sueltos y cohesivos que luego se pueden reutilizar en otros contextos y promueve la capacidad de prueba de la unidad.

Carácteristicas

- Las conexiones con recursos limitados hacen que las funciones críticas como la contrapresión y la liberación de presión sean muy naturales e intuitivas.
- El manejo de errores se vuelve tan natural como el camino feliz, en lugar de una huella de grano grueso: todo
- Los puntos en los que los datos entran y salen del sistema, así como la forma en que fluyen, se entienden bien y se rastrean fácilmente.

Arquitectura



Arquitectura

- El NiFi se ejecuta dentro de una JVM en un sistema operativo host. Los componentes principales de NiFi en la JVM son los siguientes:
- **Web Server**
  - El propósito del servidor web es alojar la API de control y comando basada en HTTP de NiFi.
- **Flow Controller**
  - El controlador de flujo es el cerebro de la operación. Proporciona rutas para que las extensiones funcionen y administra el cronograma para cuando las extensiones reciben recursos para ejecutar.

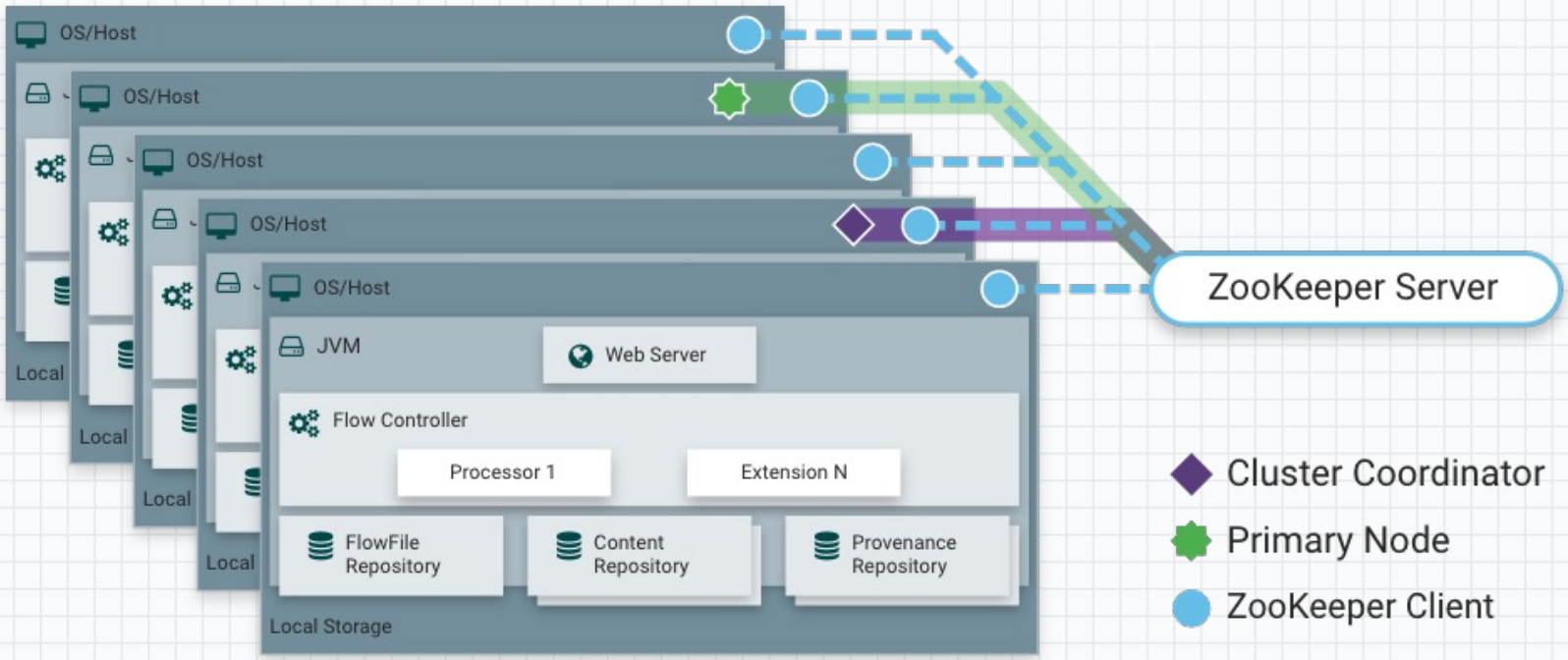
Arquitectura

- **Extensions**
  - Hay varios tipos de extensiones NiFi que se describen en otros documentos. El punto clave aquí es que las extensiones operan y se ejecutan dentro de la JVM.
- **FlowFile Repository**
  - El Repositorio de FlowFile es donde NiFi realiza un seguimiento del estado de lo que sabe sobre un FlowFile en particular que está actualmente activo en la transmisión. La implementación del repositorio es conectable. El enfoque predeterminado es un registro de escritura anticipada persistente ubicado en una partición de disco específica.

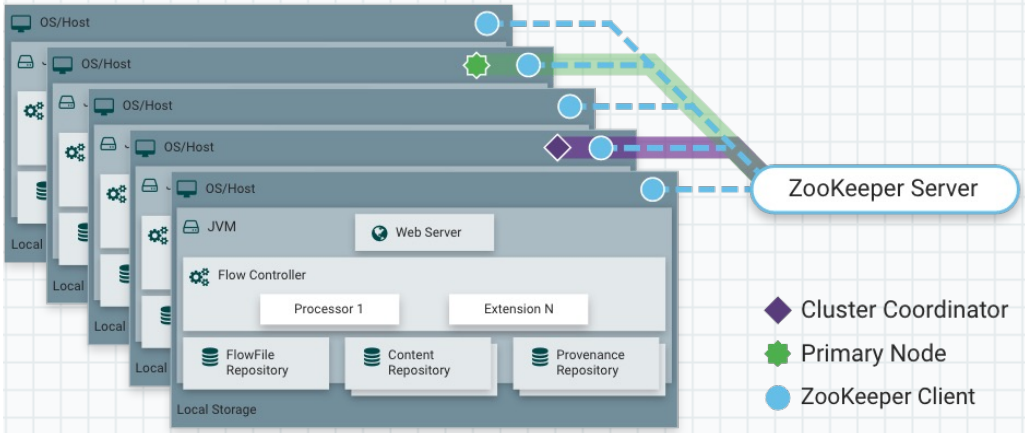
Arquitectura

- **Content Repository**
  - El Repositorio de contenido es donde viven los bytes de contenido reales de un FlowFile determinado. La implementación del repositorio es conectable. El enfoque estándar es un mecanismo muy simple que almacena bloques de datos en el sistema de archivos. Se puede especificar más de una ubicación de almacenamiento del sistema de archivos para activar diferentes particiones físicas para reducir la contención en un solo volumen.
- **Provenance Repository**
  - El Repositorio de procedencia es donde se almacenan todos los datos de eventos de procedencia. La construcción del repositorio es conectable, y la implementación predeterminada utiliza uno o más volúmenes de disco físico. Dentro de cada ubicación, los datos de eventos se indexan y se pueden buscar.

Arquitectura Cluster



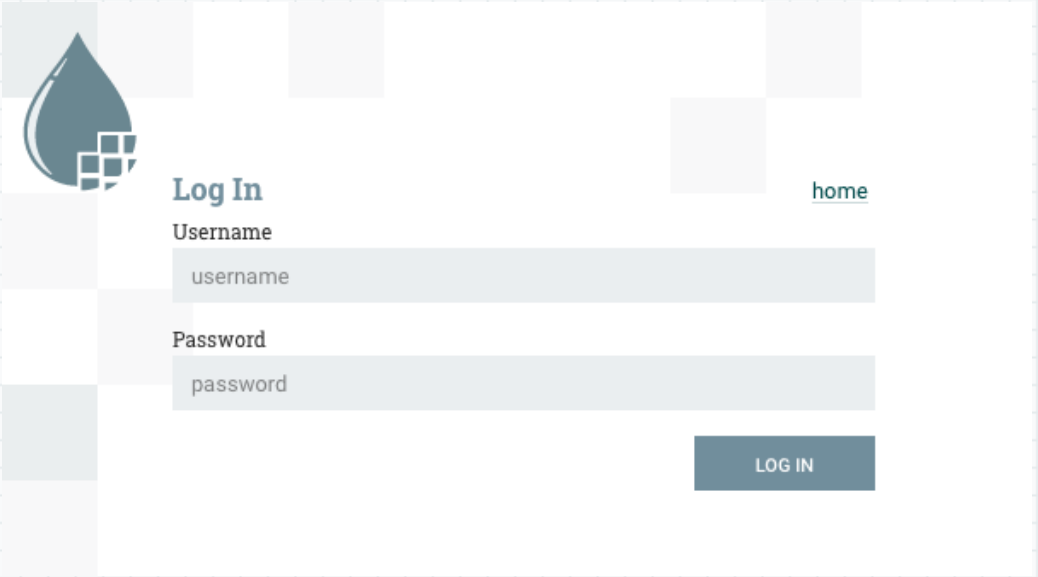
Arquitectura Cluster



Cada nodo en un clúster NiFi realiza las mismas tareas en los datos, pero cada uno opera en un conjunto diferente de datos. Apache ZooKeeper elige un único nodo como coordinador de clúster y ZooKeeper gestiona automáticamente la conmutación por error. Todos los nodos del clúster informan sobre la frecuencia cardíaca y el estado al coordinador del clúster.

El Coordinador del Clúster es responsable de desconectar y conectar los nodos. Además, cada clúster tiene un nodo principal, también elegido por ZooKeeper. Como administrador de DataFlow, puede interactuar con el clúster de NiFi a través de la interfaz de usuario (UI) de cualquier nodo. Todos los cambios realizados se replican en todos los nodos del clúster, lo que permite múltiples puntos de entrada.

User Authentication



- NiFi admite la autenticación de usuarios mediante certificados de cliente (LDAP o Kerberos) o mediante nombre de usuario/contraseña. La autenticación de nombre de usuario/contraseña la realiza un proveedor de identidad de inicio de sesión.

El proveedor de identidad de inicio de sesión es un mecanismo conectable para autenticar a los usuarios a través de su nombre de usuario/contraseña. El proveedor de identidad de inicio de sesión que se utilizará se configura en dos propiedades en el archivo nifi.properties

Instalando NIFI

- Cree una instancia azure (t2.medium) en el entorno de Labs.

- Descarga NiFi desde el enlace:
- http://nifi.apache.org/download.html

- Descomprima y descomprima en el directorio de instalación deseado.
- Reemplace los archivos de arranque y propiedades que están en la carpeta conf.

Starting NIFI

- From the <installdir>/bin directory, execute the following commands by typing ./nifi.sh <command>:

- start: starts NiFi in the background
- stop: stops NiFi that is running in the background
- status: provides the current status of NiFi
- run: runs NiFi in the foreground and waits for a Ctrl-C to initiate shutdown of NiFi

Access to NIFI

- Now that NiFi has been started, we can bring up the User Interface (UI) in order to create and monitor our dataflow.
- To get started, open a web browser and navigate to http://localhost:8080/nifi.
- The port can be changed by editing the nifi.properties file in the NiFi conf directory, but the default port is 8080.

Install NIFI as a service

- From the <installdir>/bin directory, execute the following commands by typing ./nifi.sh <command>:

- Install: installs NiFi as a service that can then be controlled via

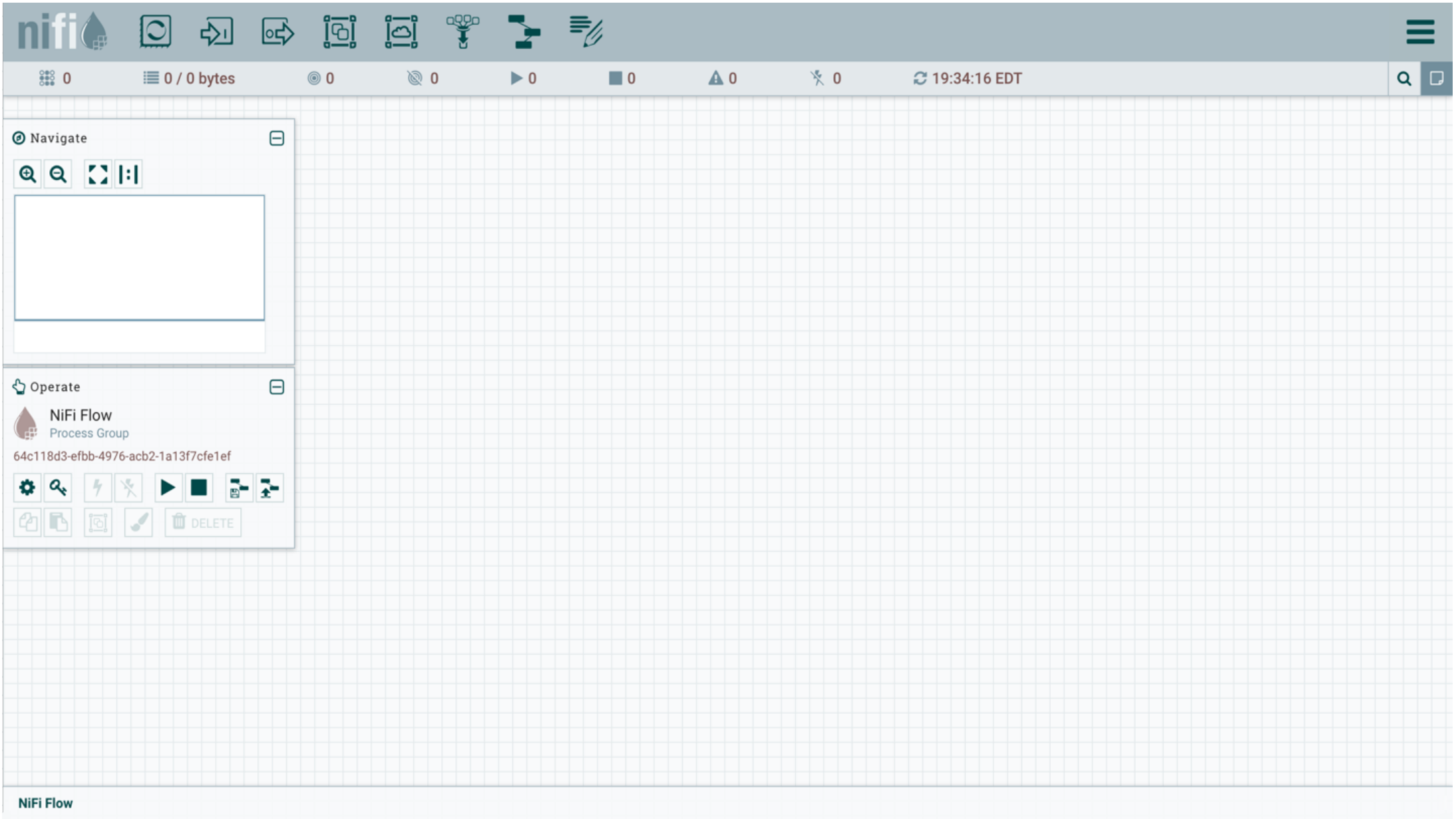
- service nifi start
- service nifi stop
- service nifi status

Using Nifi

- Now that NiFi has been started, we can bring up the User Interface (UI) in order to create and monitor our dataflow.
- To get started, open a web browser and navigate to http://localhost:8080/nifi.
- The port can be changed by editing the nifi.properties file in the NiFi conf directory, but the default port is 8080.

Locations

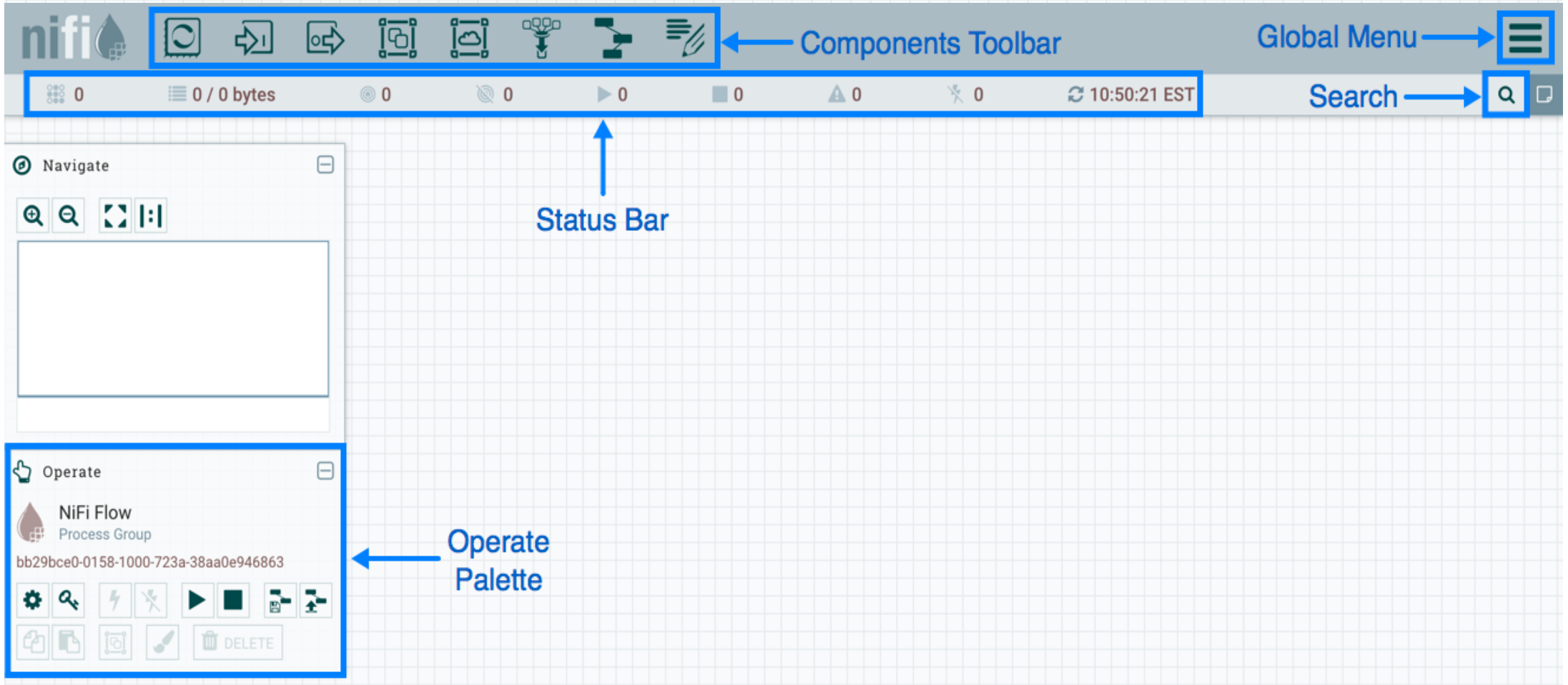
- When NiFi first starts up, the following files and directories are created:
  - content_repository
  - database_repository
  - flowfile_repository
  - provenance_repository
  - work directory
  - logs directory
- Within the conf directory, the *flow.xml.gz* file and the templates directory are created


nifi
0
0 / 0 bytes
0
0
0
0
0
0
19:34:16 EDT
Navigate
Operate
NiFi Flow
Process Group
64c118d3-efbb-4976-acb2-1a13f7cfe1ef
DELETE
NiFi Flow

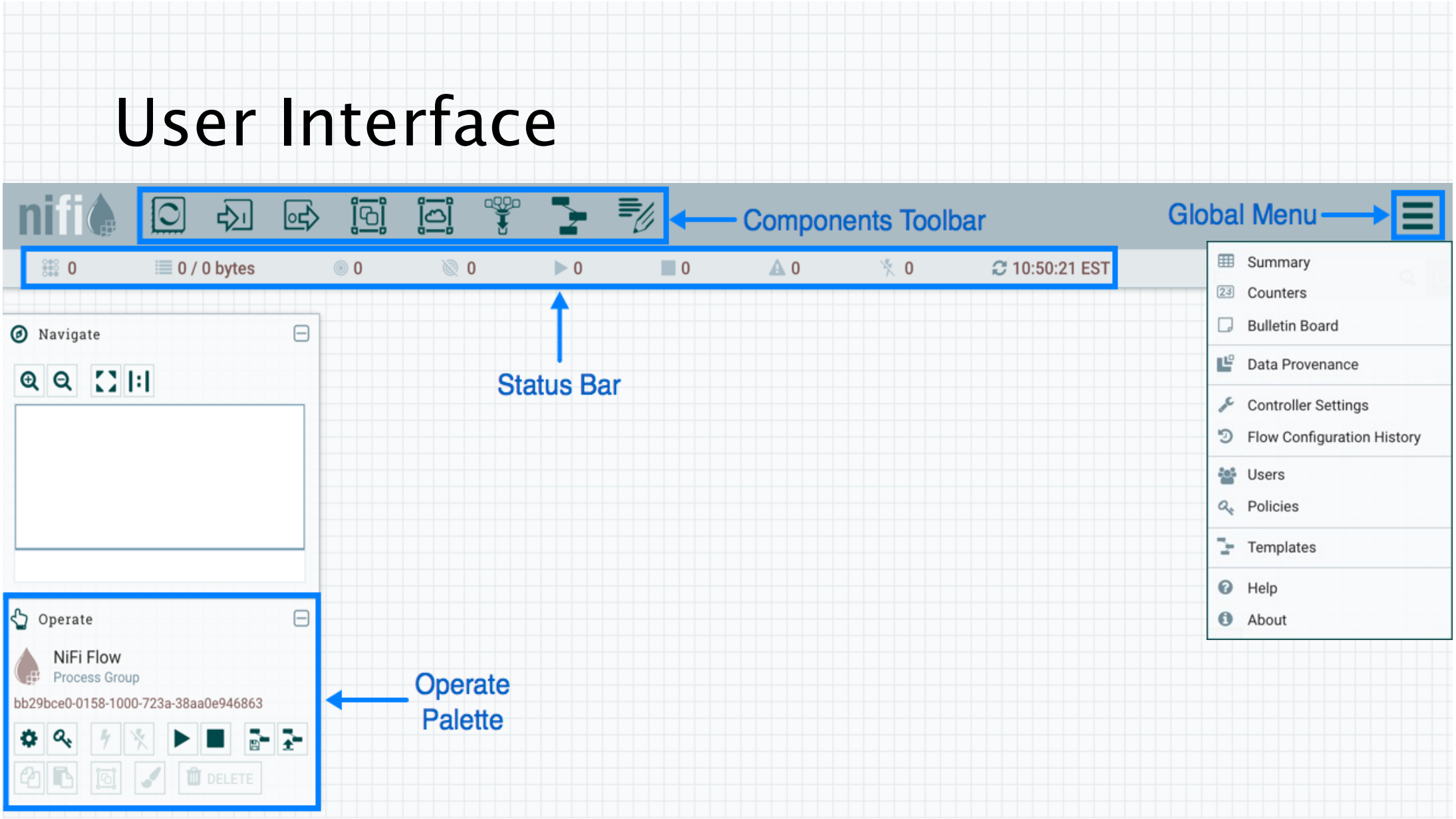
User Interface

- La interfaz de usuario de NiFi proporciona mecanismos para crear flujos de datos automatizados, así como para ver, editar, monitorear y administrar esos flujos de datos. La interfaz de usuario se puede dividir en varios segmentos, cada uno responsable de diferentes funcionalidades de la aplicación. Esta sección proporciona capturas de pantalla de la aplicación y destaca diferentes segmentos de la interfaz de usuario. Cada segmento se analiza con más detalle más adelante en el documento.
- Cuando se inicia la aplicación, el usuario puede navegar a la interfaz de usuario yendo a la dirección predeterminada de http://<nombre de host>:8080/nifi en un navegador web. No hay permisos establecidos de forma predeterminada, por lo que cualquiera puede ver y modificar el fluio de datos.

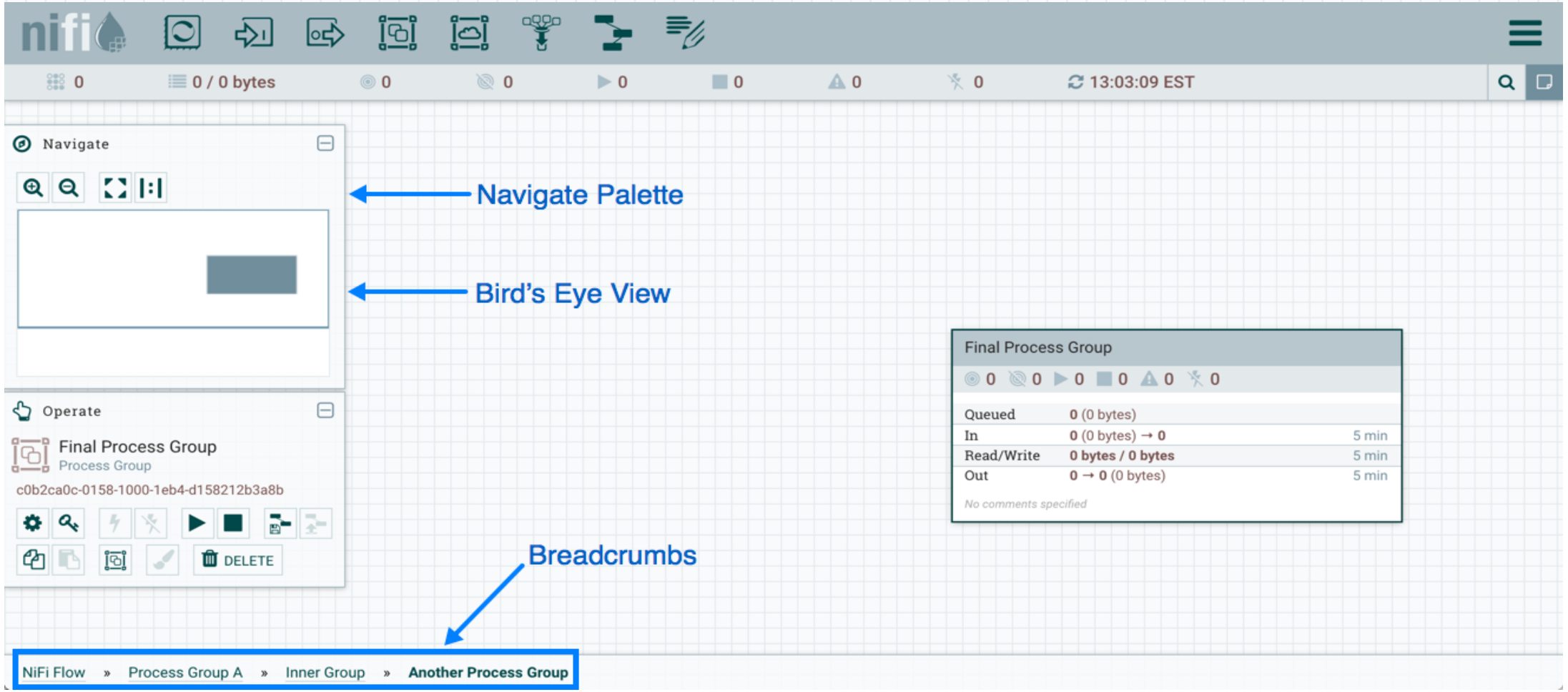
User Interface



User Interface



User Interface



Adding a processor

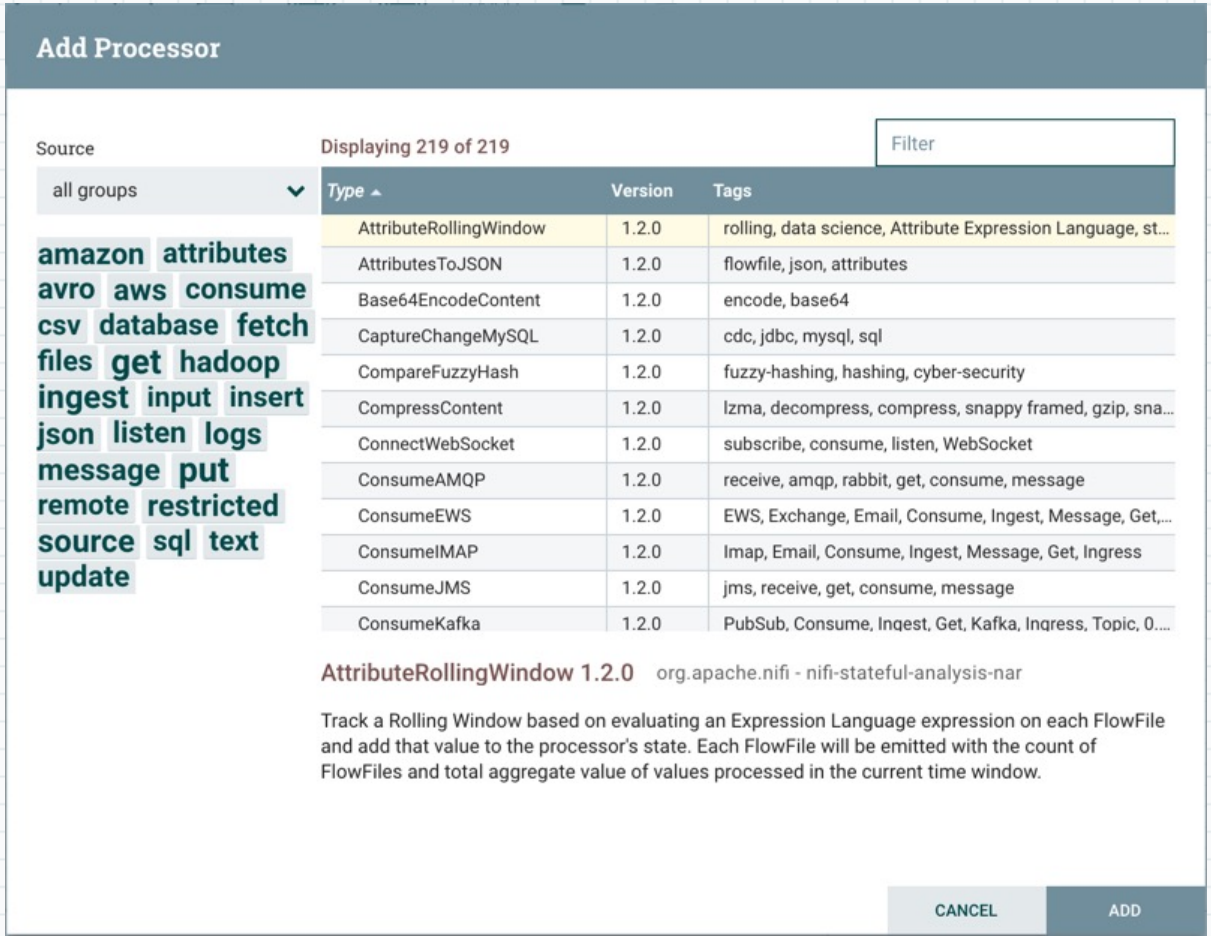
- Simply drag components from the toolbar to the canvas, configure the components to meet specific needs, and connect the components together.

- Adding Components to the Canvas
- The User Interface section above outlined the different segments of the UI and pointed out a Components Toolbar. This section looks at each of the Components in that toolbar:

Adding a processor

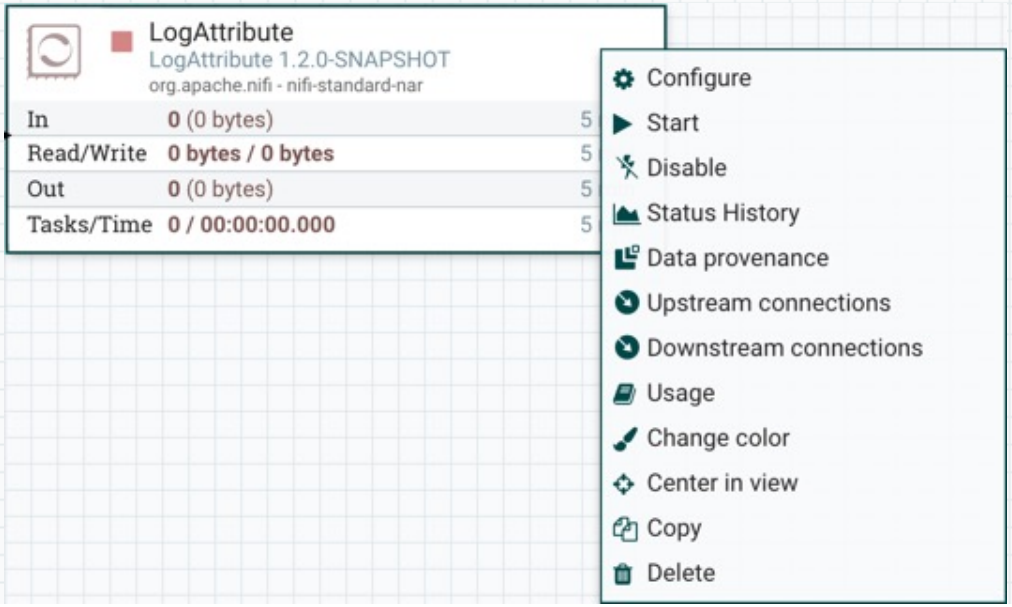
- **Processor**: The Processor is the most commonly used component, as it is responsible for data ingress, egress, routing, and manipulating. There are many different types of Processors. In fact, this is a very common Extension Point in NiFi, meaning that many vendors may implement their own Processors to perform whatever functions are necessary for their use case. When a Processor is dragged onto the canvas, the user is presented with a dialog to choose which type of Processor to use:

Adding a processor



Adding a processor

- Once you have dragged a Processor onto the canvas, you can interact with it by right-clicking on the Processor and selecting an option from the context menu. The options available to you from the context menu vary, depending on the privileges assigned to you.



Adding a processor

While the options available from the context menu vary, the following options are typically available when you have full privileges to work with a Processor:

- **Configure**: This option allows the user to establish or change the configuration of the Processor.
- **Start** or **Stop**: This option allows the user to start or stop a Processor; the option will be either Start or Stop, depending on the current state of the Processor.
- **Enable** or **Disable**: This option allows the user to enable or enable a Processor; the option will be either Enable or Disable, depending on the current state of the Processor.

Adding a processor

- **Status History**: This option opens a graphical representation of the Processor's statistical information over time.
- **Data provenance**: This option displays the NiFi Data Provenance table, with information about data provenance events for the FlowFiles routed through that Processor.
- **Upstream connections**: This option allows the user to see and "jump to" upstream connections that are coming into the Processor.
- **Downstream connections**: This option allows the user to see and "jump to" downstream connections that are going out of the Processor. This is particularly useful when processors connect into and out of other Process Groups.

Adding a processor

- **Usage**: This option takes the user to the Processor's usage documentation.
- **Change color**: This option allows the user to change the color of the Processor, which can make the visual management of large flows easier.
- **Center in view**: This option centers the view of the canvas on the given Processor.

Adding a processor

- **Copy**: This option places a copy of the selected Processor on the clipboard, so that it may be pasted elsewhere on the canvas by right-clicking on the canvas and selecting Paste. The Copy/Paste actions also may be done using the keystrokes Ctrl-C (Command-C) and Ctrl-V (Command-V).
- **Delete**: This option allows the DFM to delete a Processor from the canvas.

Adding a processor

- **Input Port**: Input Ports provide a mechanism for transferring data into a Process Group. When an Input Port is dragged onto the canvas, the DFM is prompted to name the Port. All Ports within a Process Group must have unique names.

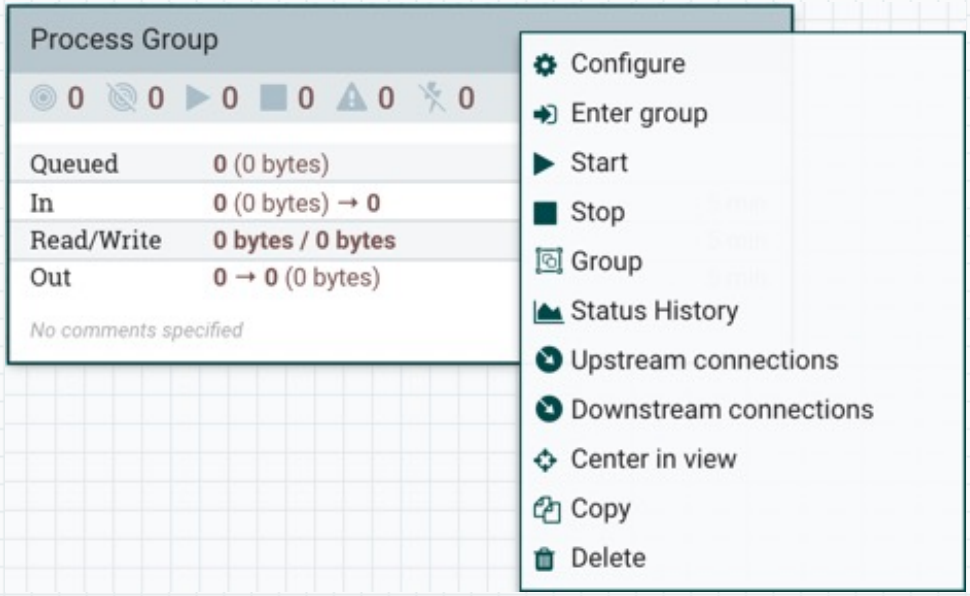
- **Output Port**: Output Ports provide a mechanism for transferring data from a Process Group to destinations outside of the Process Group. When an Output Port is dragged onto the canvas, the DFM is prompted to name the Port. All Ports within a Process Group must have unique names.

Adding a processor

- **Process Group**: Process Groups can be used to logically group a set of components so that the dataflow is easier to understand and maintain. When a Process Group is dragged onto the canvas, the DFM is prompted to name the Process Group. All Process Groups within the same parent group must have unique names. The Process Group will then be nested within that parent group.

Adding a processor

- Once you have dragged a Process Group onto the canvas, you can interact with it by right-clicking on the Process Group and selecting an option from context menu.The options available to you from the context menu vary, depending on the privileges assigned to you.

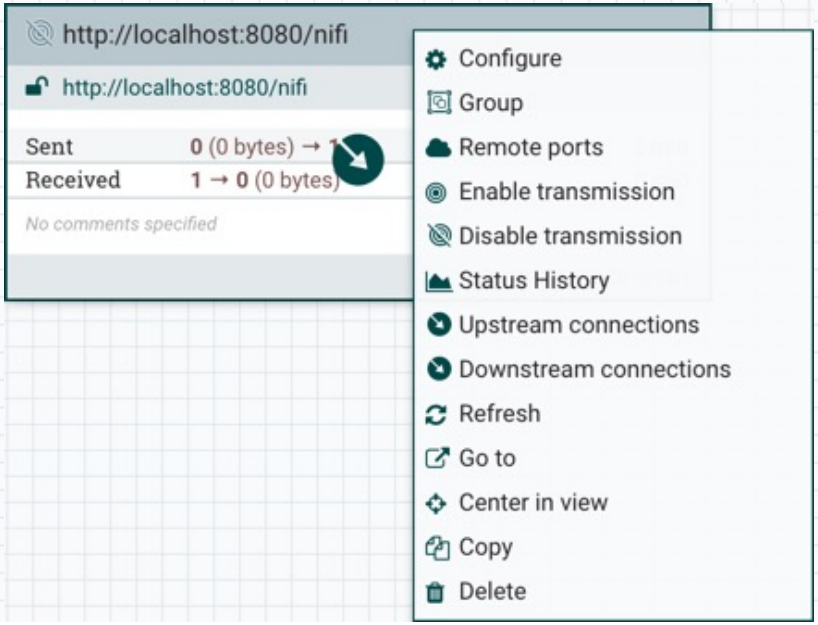


Adding a processor

- **Remote Process Group**: Remote Process Groups appear and behave similar to Process Groups. However, the Remote Process Group (RPG) references a remote instance of NiFi. When an RPG is dragged onto the canvas, rather than being prompted for a name, the DFM is prompted for the URL of the remote NiFi instance. If the remote NiFi is a clustered instance, the URL that should be used is the URL of any NiFi instance in that cluster. When data is transferred to a clustered instance of NiFi via an RPG, the RPG will first connect to the remote instance whose URL is configured to determine which nodes are in the cluster and how busy each node is.

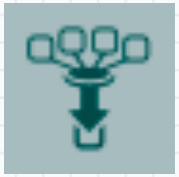
Adding a processor

- Once you have dragged a Remote Process Group onto the canvas, you can may interact with it by right-clicking on the Remote Process Group and selecting an option from context menu. The options available to you from the context menu vary, depending on the privileges assigned to you.



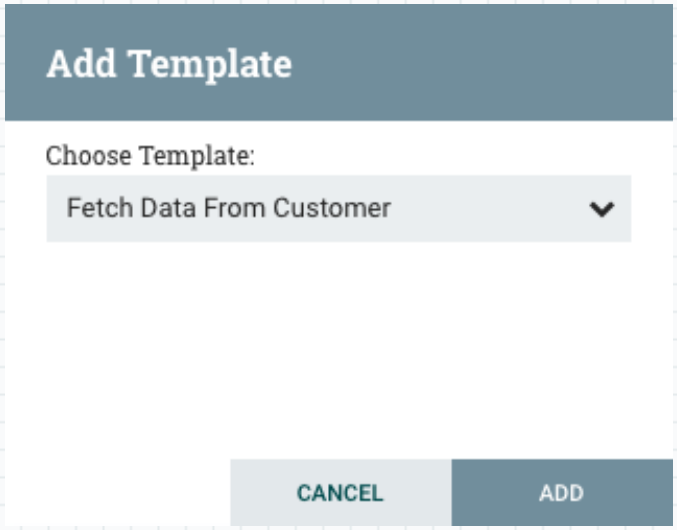
Adding a Funnel

- Funnels are used to combine the data from many Connections into a single Connection. This has two advantages. First, if many Connections are created with the same destination, the canvas can become cluttered if those Connections have to span a large space. By funneling these Connections into a single Connection, that single Connection can then be drawn to span that large space instead. Secondly, Connections can be configured with FlowFile Prioritizers. Data from several Connections can be funneled into a single Connection, providing the ability to Prioritize all of the data on that one Connection, rather than prioritizing the data on each Connection independently.



Adding a template

- Templates can be created by DFMs from sections of the flow, or they can be imported from other dataflows. These Templates provide larger building blocks for creating a complex flow quickly. When the Template is dragged onto the canvas, the DFM is provided a dialog to choose which Template to add to the canvas:



Adding a label

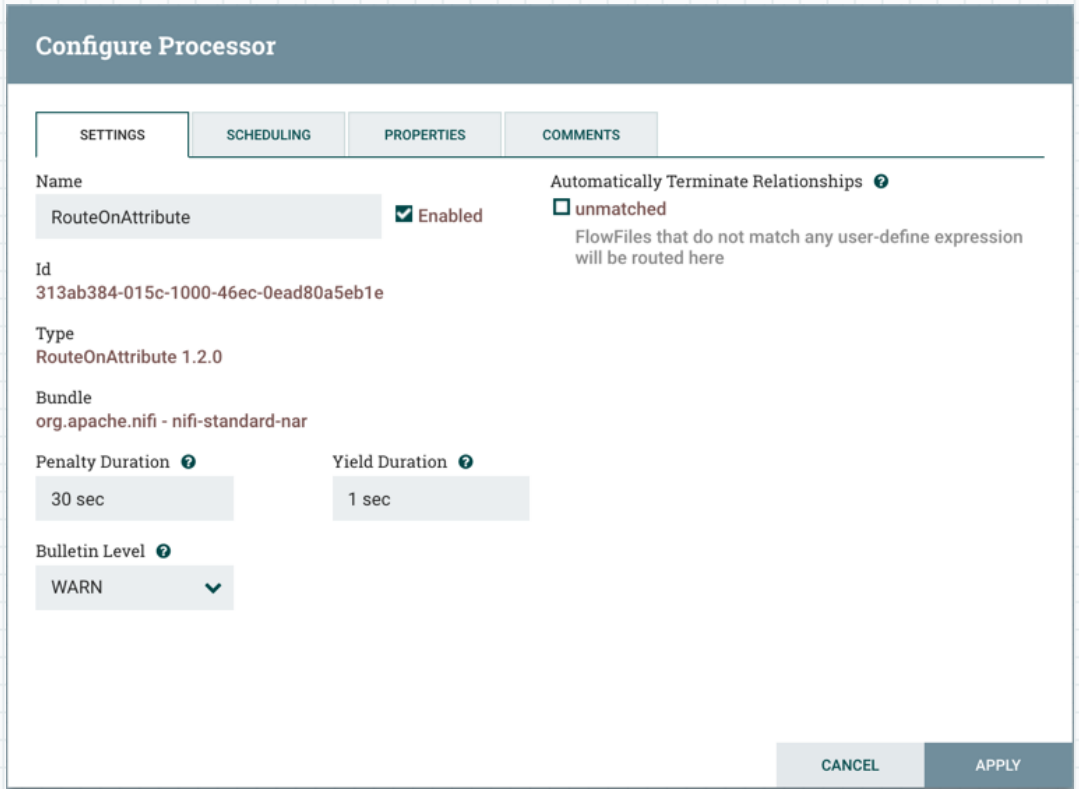
- Labels are used to provide documentation to parts of a dataflow. When a Label is dropped onto the canvas, it is created with a default size. The Label can then be resized by dragging the handle in the bottom-right corner. The Label has no text when initially created. The text of the Label can be added by right-clicking on the Label and choosing Configure.

Configurando un procesador

- To configure a processor, right-click on the Processor and select the Configure option from the context menu. The configuration dialog is opened with four different tabs, each of which is discussed below. Once you have finished configuring the Processor, you can apply the changes by clicking the Apply button or cancel all changes by clicking the Cancel button.

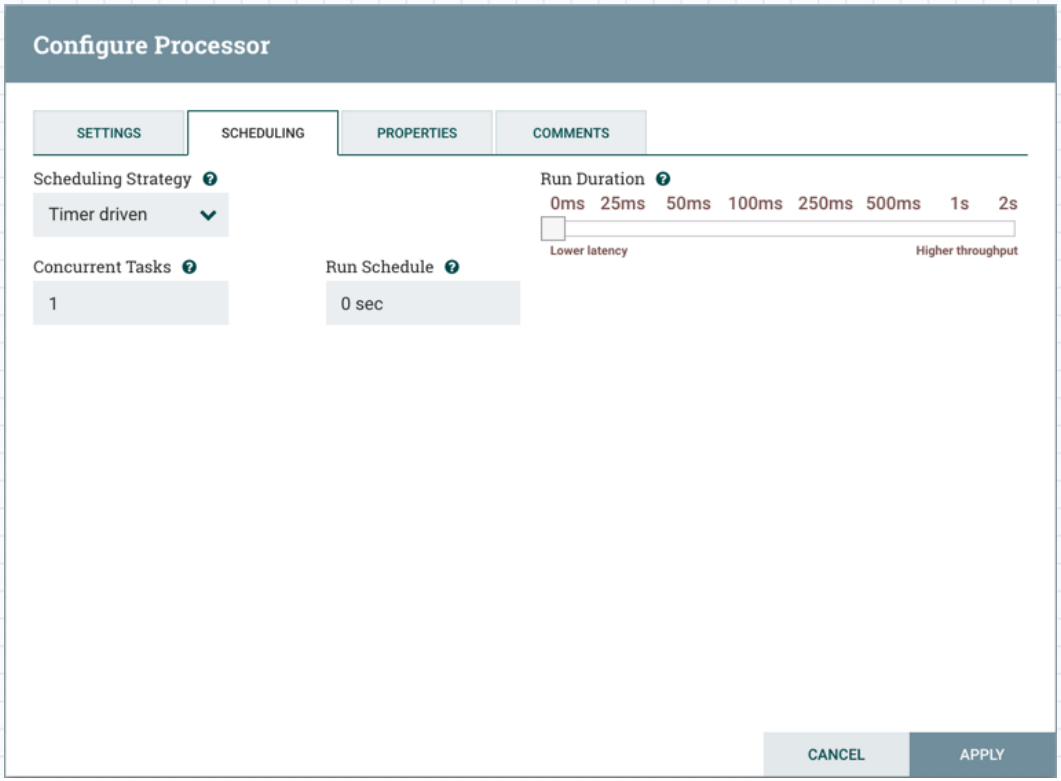
Configurando un procesador

- Settings Tab
  - The first tab in the Processor Configuration dialog is the Settings tab:



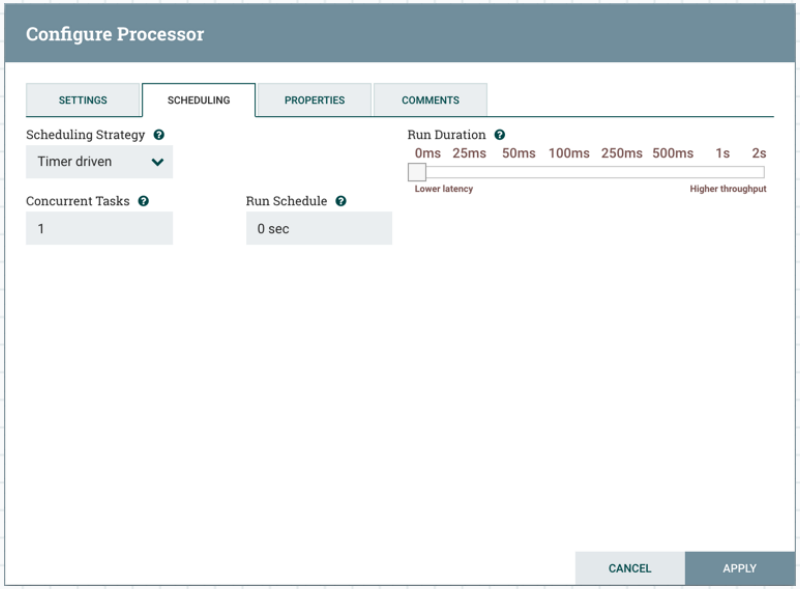
Configurando un procesador

- Scheduling Tab
  - The second tab in the Processor Configuration dialog is the Scheduling Tab:



Configurando un procesador

- Scheduling Tab

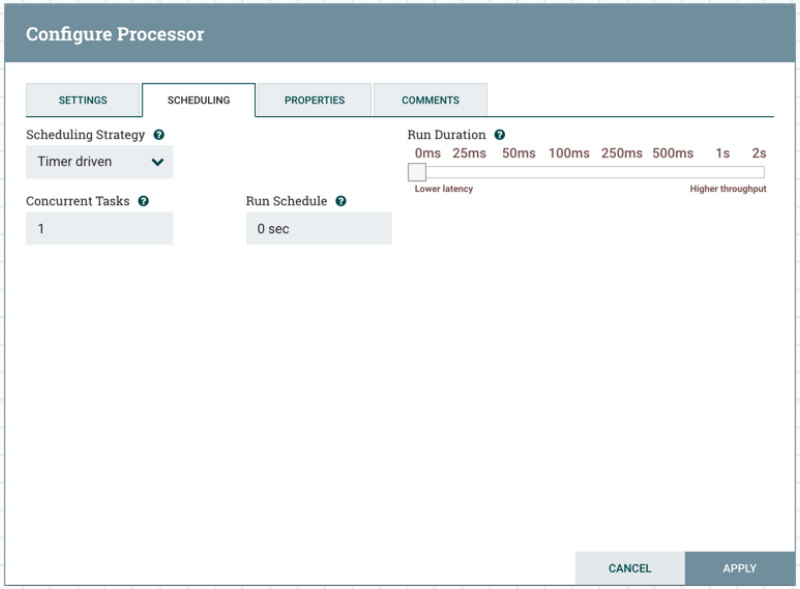


The first configuration option is the Scheduling Strategy. There are three possible options for scheduling components:

Timer driven: This is the default mode. The Processor will be scheduled to run on a regular interval. The interval at which the Processor is run is defined by the ‘Run schedule’ option (see below).

Configurando un procesador

- Scheduling Tab

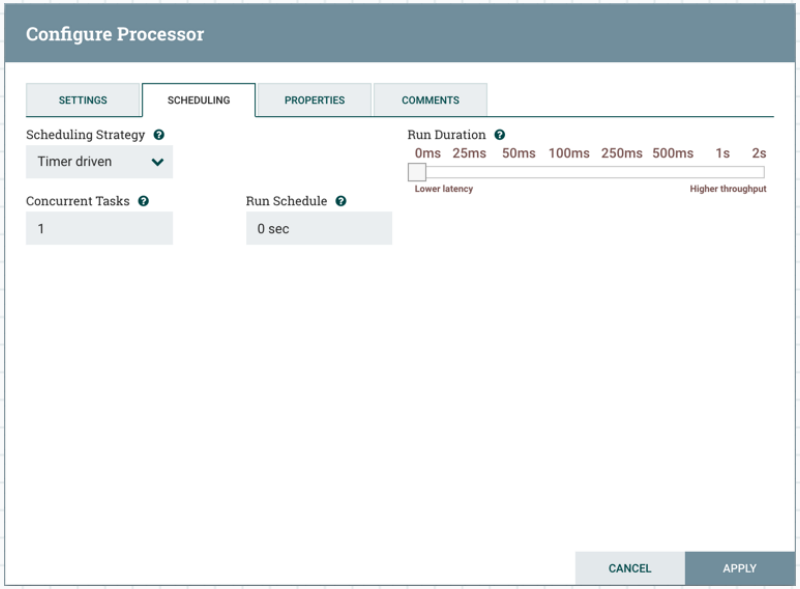


Event driven: When this mode is selected, the Processor will be triggered to run by an event, and that event occurs when FlowFiles enter Connections feeding this Processor. This mode is currently considered experimental and is not supported by all Processors. When this mode is selected, the 'Run schedule' option is not configurable, as the Processor is not triggered to run periodically but as the result of an event.

Additionally, this is the only mode for which the 'Concurrent tasks' option can be set to 0. In this case, the number of threads is limited only by the size of the Event-Driven Thread Pool that the administrator has configured.

Configurando un procesador

- Scheduling Tab



CRON driven: When using the CRON driven scheduling mode, the Processor is scheduled to run periodically, similar to the Timer driven scheduling mode. However, the CRON driven mode provides significantly more flexibility at the expense of increasing the complexity of the configuration.

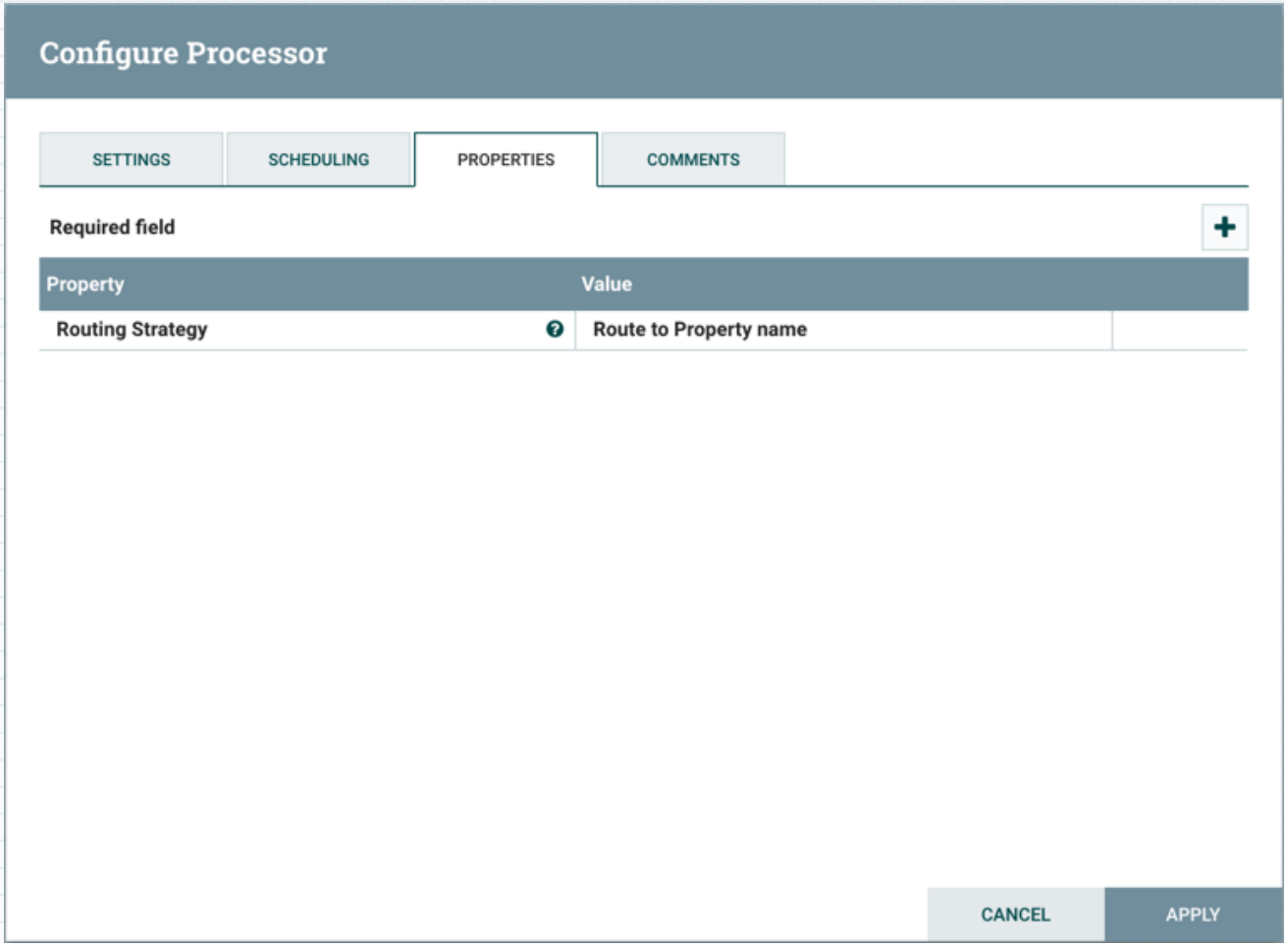
Configurando un procesador

- The CRON driven scheduling value is a string of six required fields and one optional field, each separated by a space. These fields are:

Field	**Valid values**
Seconds	0-59
Minutes	0-59
Hours	0-23
Day of Month	1-31
Month	1-12 or JAN-DEC
Day of Week	1-7 or SUN-SAT
Year (optional)	empty, 1970-2099

Configurando un procesador

- Properties Tab
  - The Properties Tab provides a mechanism to configure Processor-specific behavior. There are no default properties. Each type of Processor must define which Properties make sense for its use case. Below, we see the Properties Tab for a RouteOnAttribute Processor:

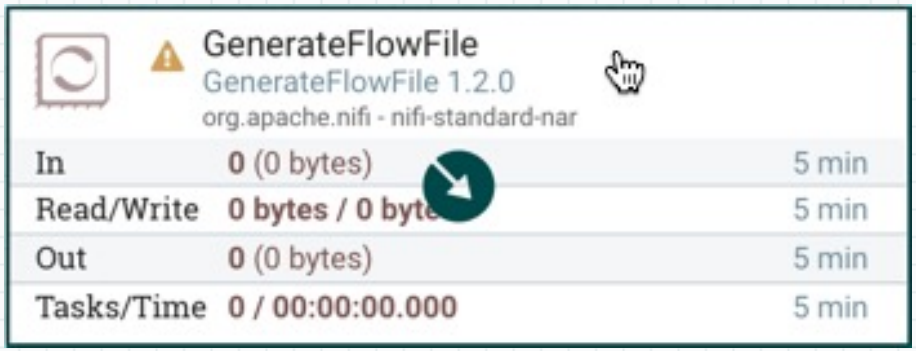


Configurando un procesador

- Additional Help
  - You can access additional documentation about each Processor's usage by right-clicking on the Processor and selecting 'Usage' from the context menu. Alternatively, select Help from the Global Menu in the top-right corner of the UI to display a Help page with all of the documentation, including usage documentation for all the Processors that are available. Click on the desired Processor to view usage documentation.

Conectando Componentes

- Once processors and other components have been added to the canvas and configured, the next step is to connect them to one another so that NiFi knows what to do with each FlowFile after it has been processed. This is accomplished by creating a Connection between each component. When the user hovers the mouse over the center of a component, a new Connection icon ( ) appears:

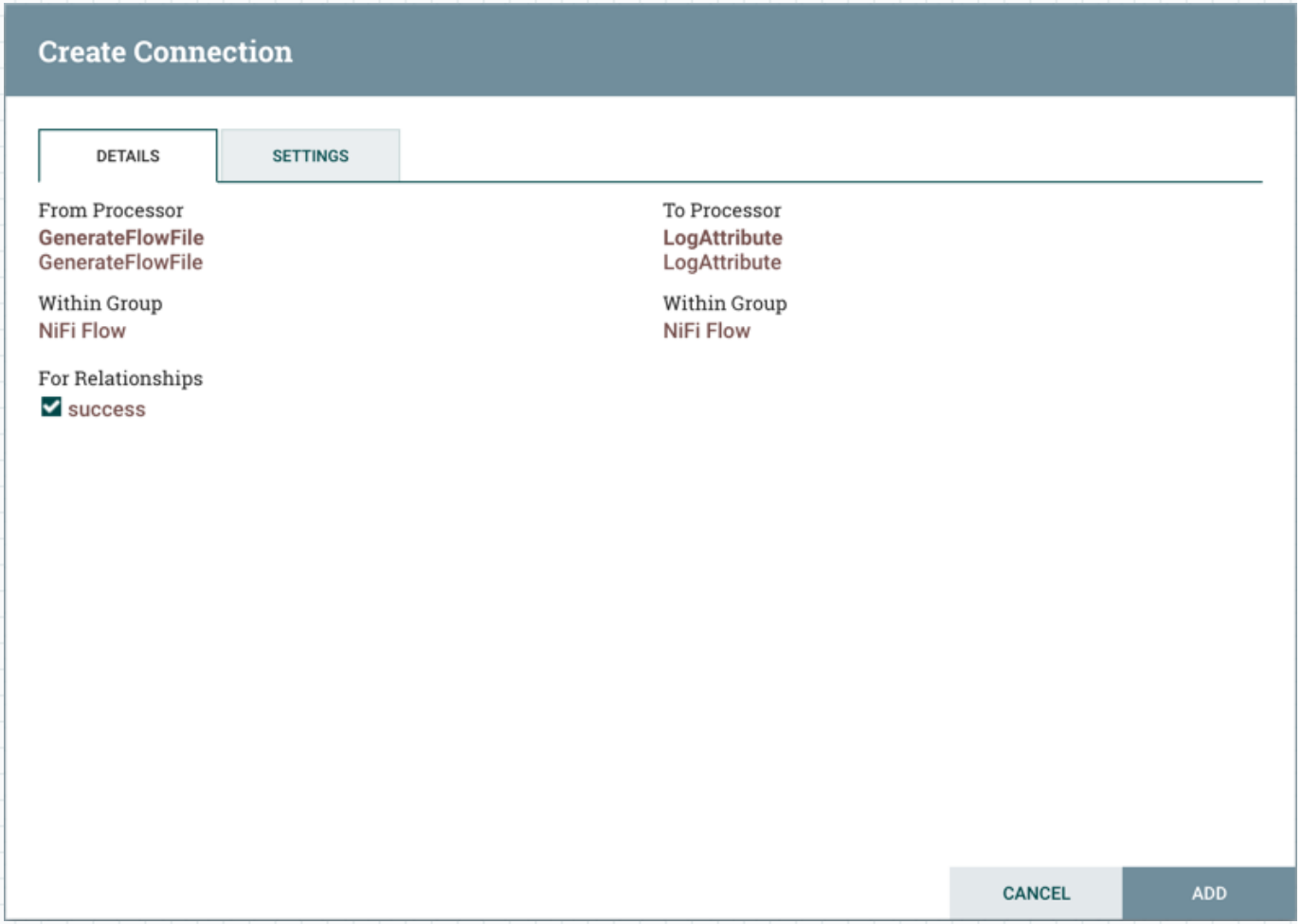


Conectando Componentes

- The user drags the Connection bubble from one component to another until the second component is highlighted. When the user releases the mouse, a *Create Connection* dialog appears. This dialog consists of two tabs: ‘Details’ and ‘Settings’. They are discussed in detail below. Note that it is possible to draw a connection so that it loops back on the same processor. This can be useful if the DFM wants the processor to try to re-process FlowFiles if they go down a failure Relationship. To create this type of looping connection, simply drag the connection bubble away and then back to the same processor until it is highlighted. Then release the mouse and the same *Create Connection* dialog appears.

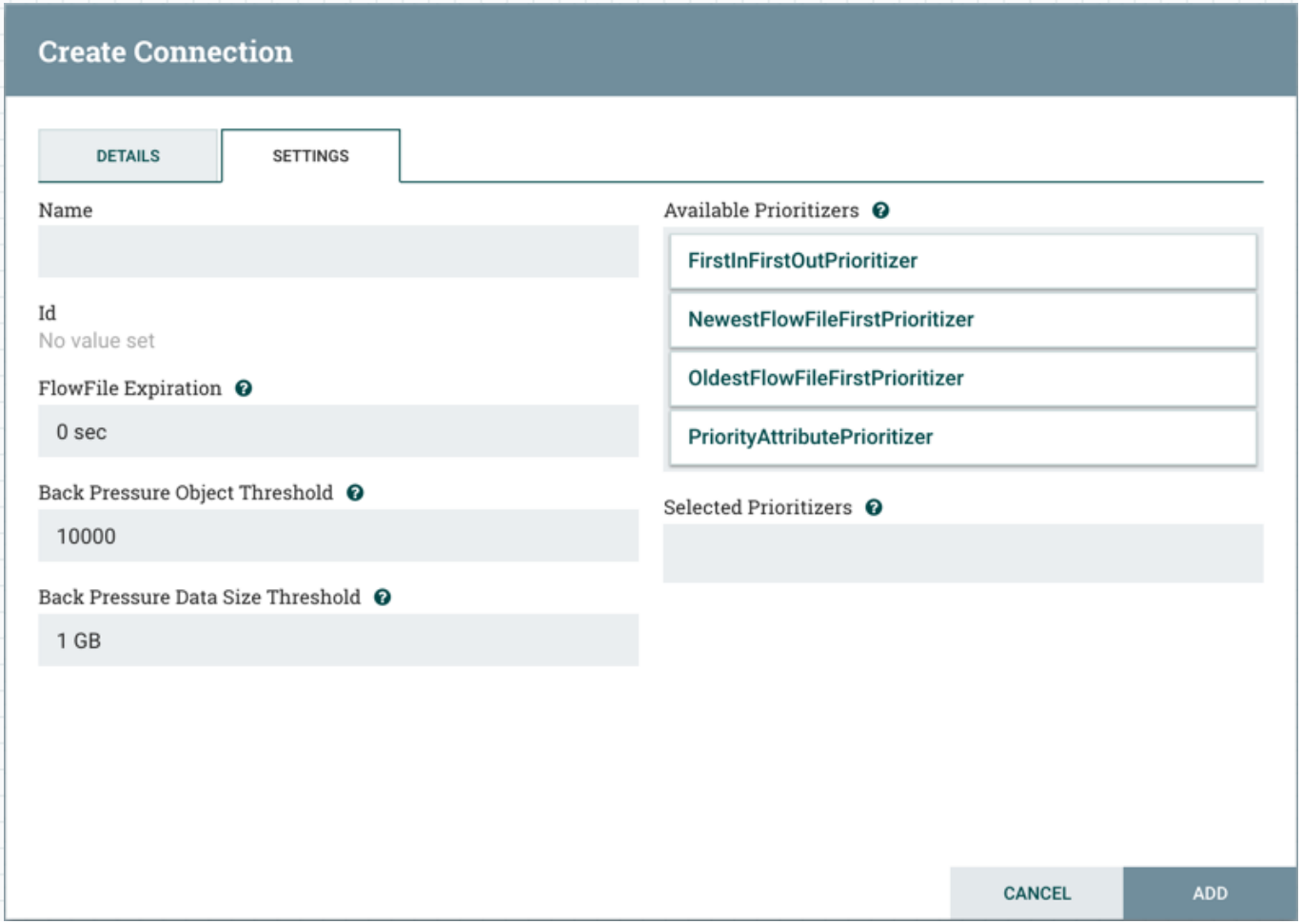
Conectando Componentes

- Details Tab
  - The Details Tab of the *Create Connection* dialog provides information about the source and destination components, including the component name, the component type, and the Process Group in which the component lives:

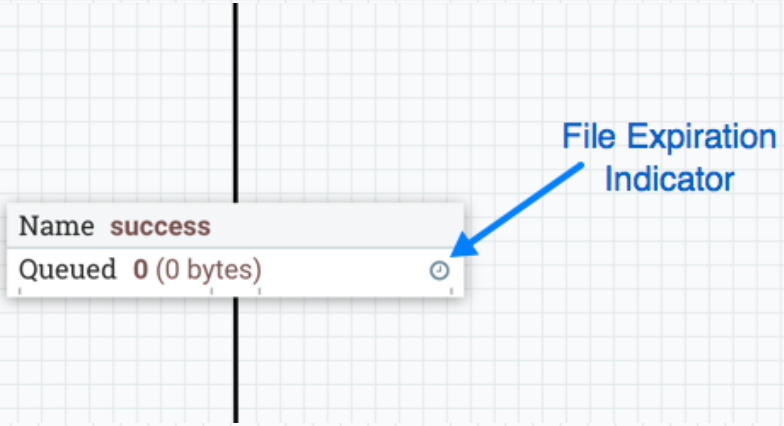


Conectando Componentes

- Settings
  - The Settings Tab provides the ability to configure the Connection's name, FlowFile expiration, Back Pressure thresholds, and Prioritization:



Conectando Componentes

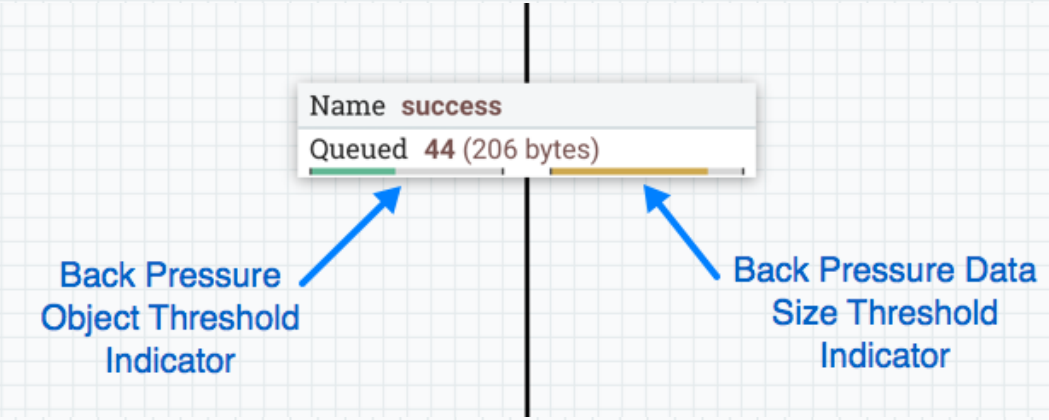


- FlowFile Expiration

- FlowFile expiration is a concept by which data that cannot be processed in a timely fashion can be automatically removed from the flow. The expiration period is based on the time that the data entered the NiFi instance. The default value of 0 sec indicates that the data will never expire. When a file expiration other than 0 sec is set, a small clock icon appears on the connection label, so the DFM can see it at-a-glance when looking at a flow on the canvas.

Conectando Componentes

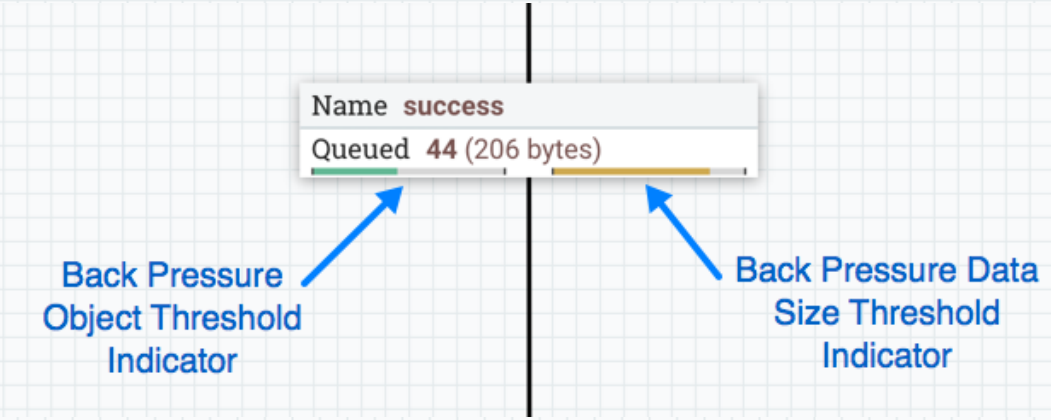
- Back Pressure



- NiFi provides two configuration elements for Back Pressure. These thresholds indicate how much data should be allowed to exist in the queue before the component that is the source of the Connection is no longer scheduled to run. This allows the system to avoid being overrun with data.

Conectando Componentes

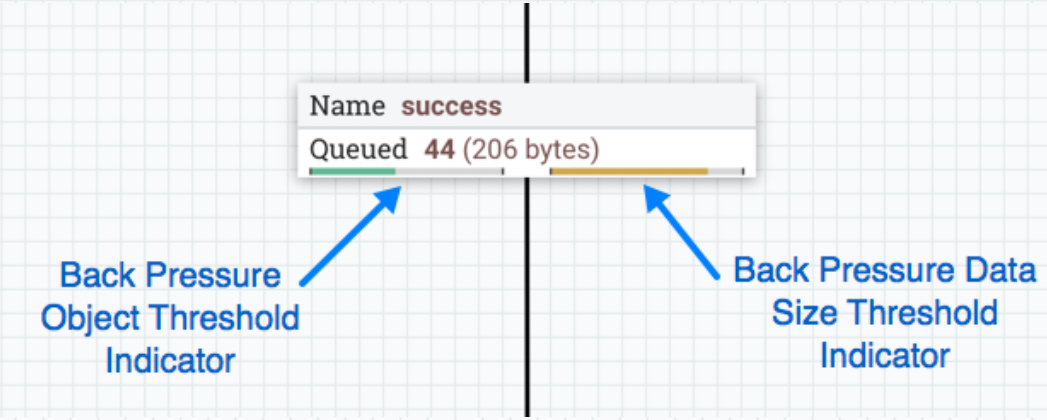
- Back Pressure



- The first option provided is the “Back pressure object threshold.” This is the number of FlowFiles that can be in the queue before back pressure is applied.

Conectando Componentes

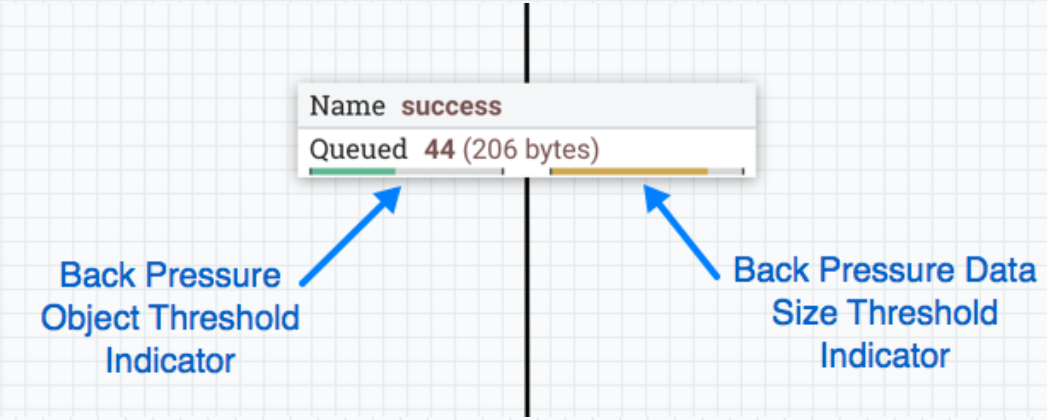
- Back Pressure



- The second configuration option is the “Back pressure data size threshold.” This specifies the maximum amount of data (in size) that should be queued up before applying back pressure. This value is configured by entering a number followed by a data size (B for bytes, KB for kilobytes, MB for megabytes, GB for gigabytes, or TB for terabytes).

Conectando Componentes

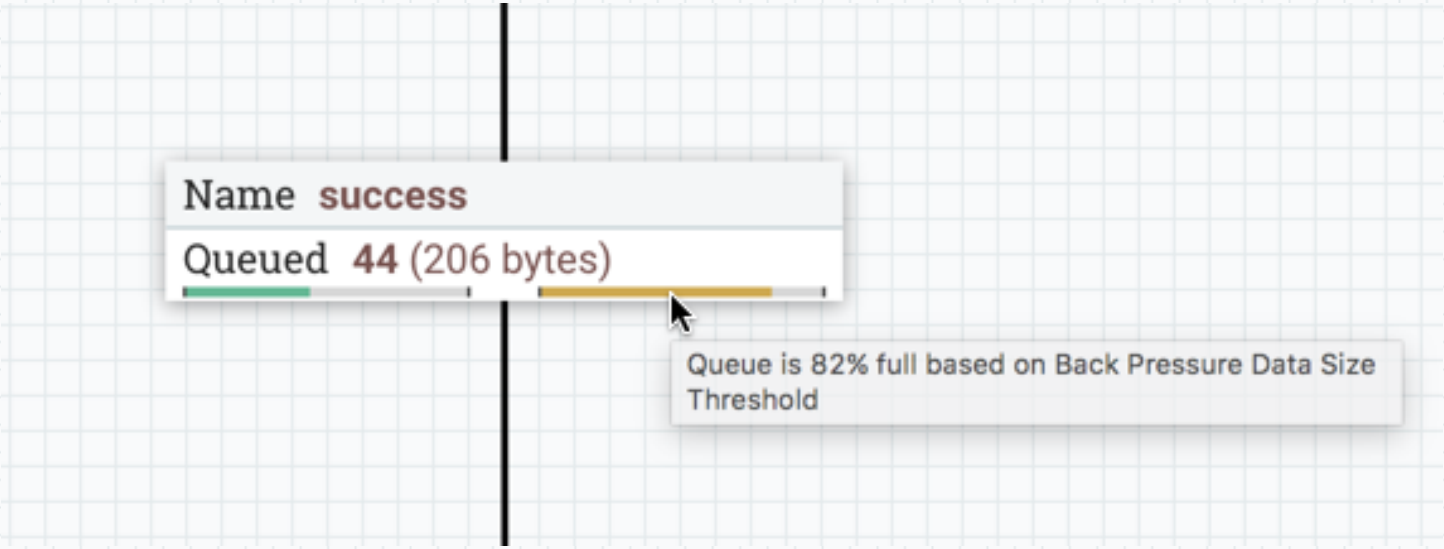
- Back Pressure



- When back pressure is enabled, small progress bars appear on the connection label, so the DFM can see it at-a-glance when looking at a flow on the canvas. The progress bars change color based on the queue percentage: Green (0-60%), Yellow (61-85%) and Red (86-100%).

Conectando Componentes

- Hovering your mouse over a bar displays the exact percentage.



Conectando Componentes

- When the queue is completely full, the Connection is highlighted in red.

Conectando Componentes

- Prioritization
- The right-hand side of the tab provides the ability to prioritize the data in the queue so that higher priority data is processed first. Prioritizers can be dragged from the top (‘Available prioritizers’) to the bottom (‘Selected prioritizers’). Multiple prioritizers can be selected. The prioritizer that is at the top of the ‘Selected prioritizers’ list is the highest priority. If two FlowFiles have the same value according to this prioritizer, the second prioritizer will determine which FlowFile to process first, and so on. If a prioritizer is no longer desired, it can then be dragged from the ‘Selected prioritizers’ list to the ‘Available prioritizers’ list.

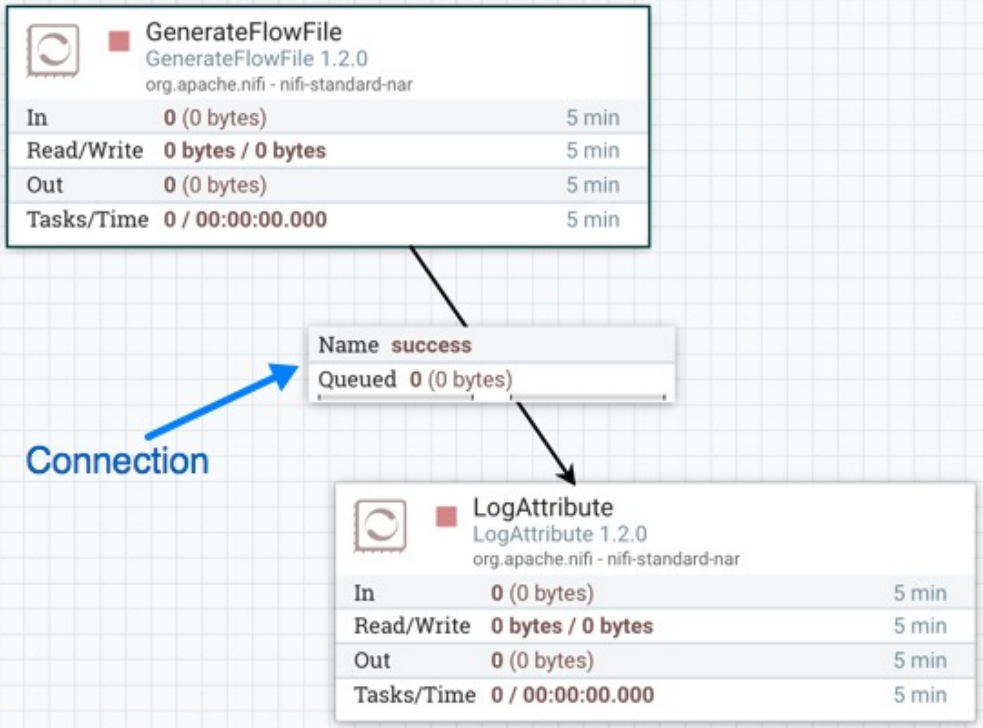
Conectando Componentes

- The following prioritizers are available:

- **FirstInFirstOutPrioritizer**: Given two FlowFiles, the one that reached the connection first will be processed first.
- **NewestFlowFileFirstPrioritizer**: Given two FlowFiles, the one that is newest in the dataflow will be processed first.
- **OldestFlowFileFirstPrioritizer**: Given two FlowFiles, the one that is oldest in the dataflow will be processed first. *This is the default scheme that is used if no prioritizers are selected.*
- **PriorityAttributePrioritizer**: Given two FlowFiles that both have a "priority" attribute, the one that has the highest priority value will be processed first.

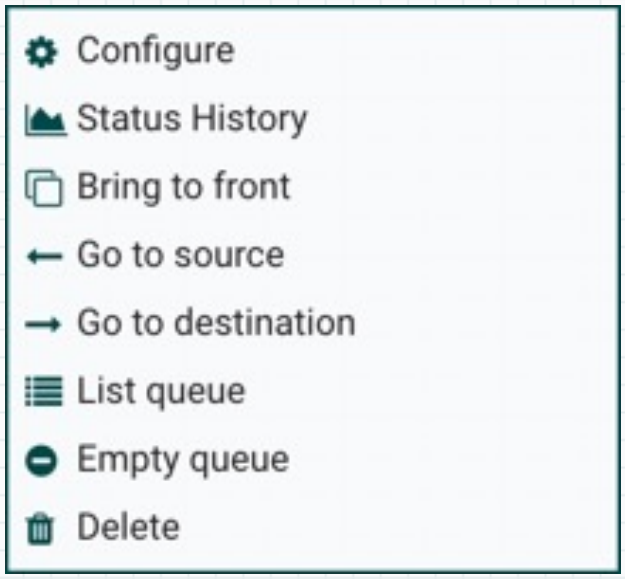
Conectando Componentes

- Changing Configuration and Context Menu Options
- After a connection has been drawn between two components, the connection's configuration may be changed, and the connection may be moved to a new destination; however, the processors on either side of the connection must be stopped before a configuration or destination change may be made.



Conectando Componentes

- To change a connection’s configuration or interact with the connection in other ways, right-click on the connection to open the connection context menu.



Conectando Componentes

- The following options are available:

- **Configure**: This option allows the user to change the configuration of the connection.
- **Status History**: This option opens a graphical representation of the connection’s statistical information over time.
- **Bring to front**: This option brings the connection to the front of the canvas if something else (such as another connection) is overlapping it.

Conectando Componentes

- The following options are available:
- **Go to source**: This option can be useful if there is a long distance between the connection's source and destination components on the canvas. By clicking this option, the view of the canvas will jump to the source of the connection.
- **Go to destination**: Similar to the "Go to source" option, this option changes the view to the destination component on the canvas and can be useful if there is a long distance between two connected components.
- **List queue**: This option lists the queue of FlowFiles that may be waiting to be processed.

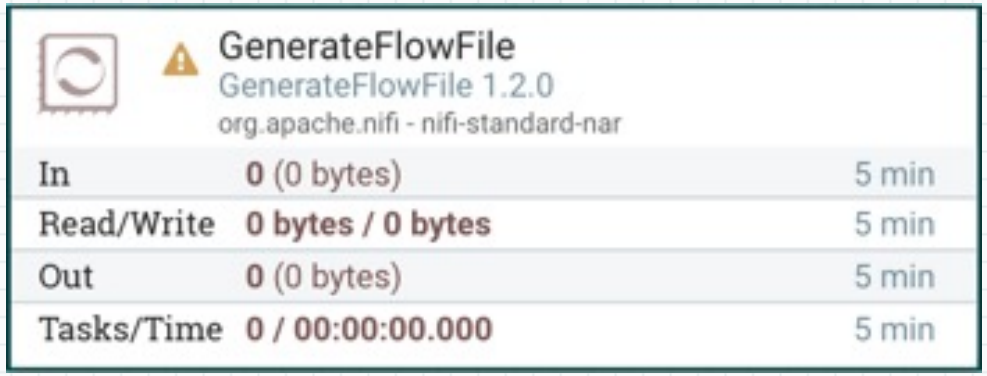
Connectando componentes

- Las siguientes opciones están disponibles:

- Cola vacía: esta opción permite que el DFM borre la cola de FlowFiles que pueden estar esperando para ser procesados. Esta opción puede ser especialmente útil durante las pruebas, cuando el DFM no se preocupa por eliminar datos de la cola. Cuando se selecciona esta opción, los usuarios deben confirmar que desean eliminar los datos en la cola.
- Eliminar: esta opción permite que el DFM elimine una conexión entre dos componentes. Tenga en cuenta que los componentes de ambos lados de la conexión deben detenerse y la conexión debe estar vacía antes de poder eliminarla.

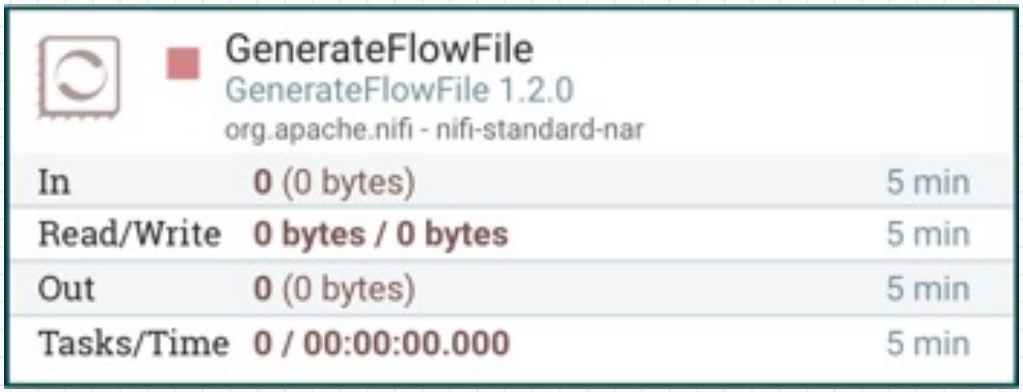
Validando un procesador

- Antes de intentar iniciar un Procesador, es importante asegurarse de que la configuración del Procesador sea válida. Se muestra un indicador de estado en la parte superior izquierda del procesador. Si el procesador no es válido, el indicador mostrará un indicador de advertencia amarillo con un signo de exclamación que indica que hay un problema:



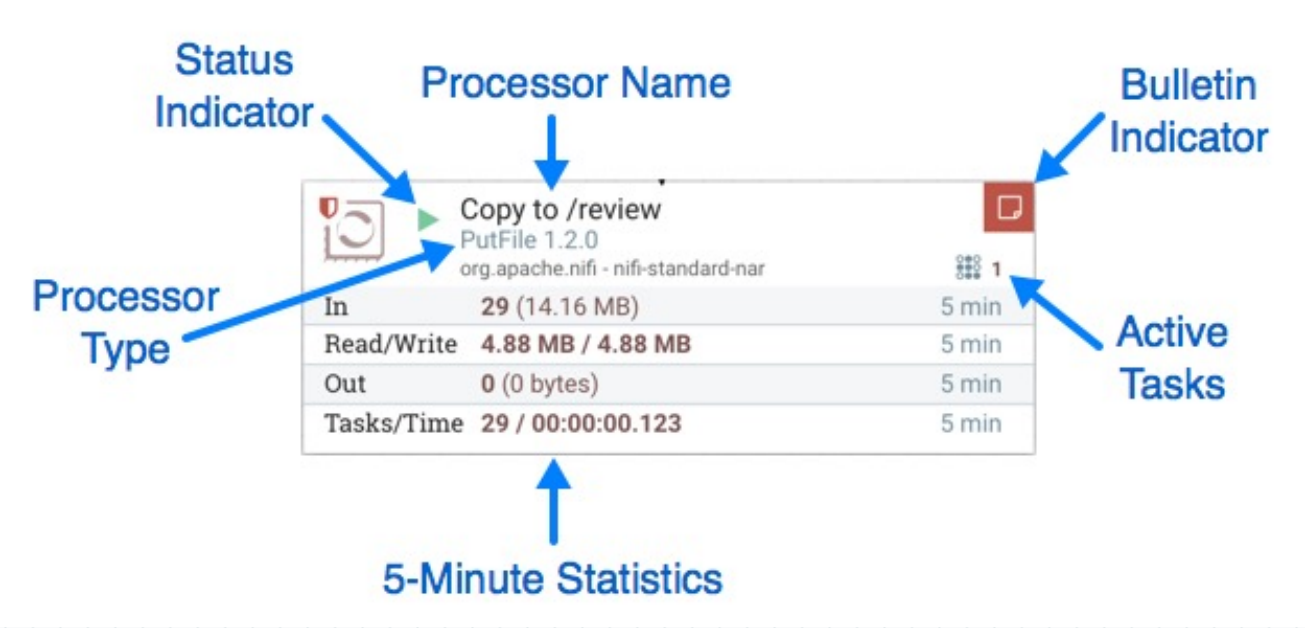
Validando un procesador

- En este caso, al pasar el cursor sobre el ícono del indicador, aparecerá una información sobre herramientas que muestra todos los errores de validación del Procesador. Una vez que se manejen todos los errores de validación, el indicador de estado cambiará a un ícono Detener, lo que indica que el Procesador es válido y está listo para iniciarse, pero no se está ejecutando actualmente:



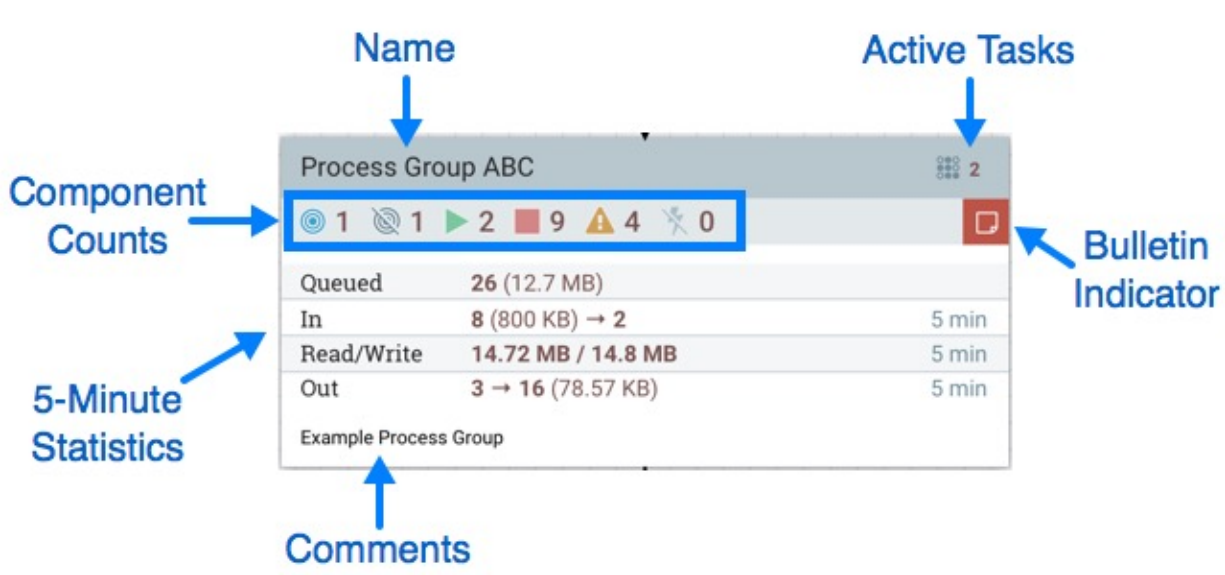
Analizando um procesador

- El NiFi proporciona una cantidad significativa de información sobre cada Procesador en la pantalla. El siguiente diagrama muestra la anatomía de un procesador:

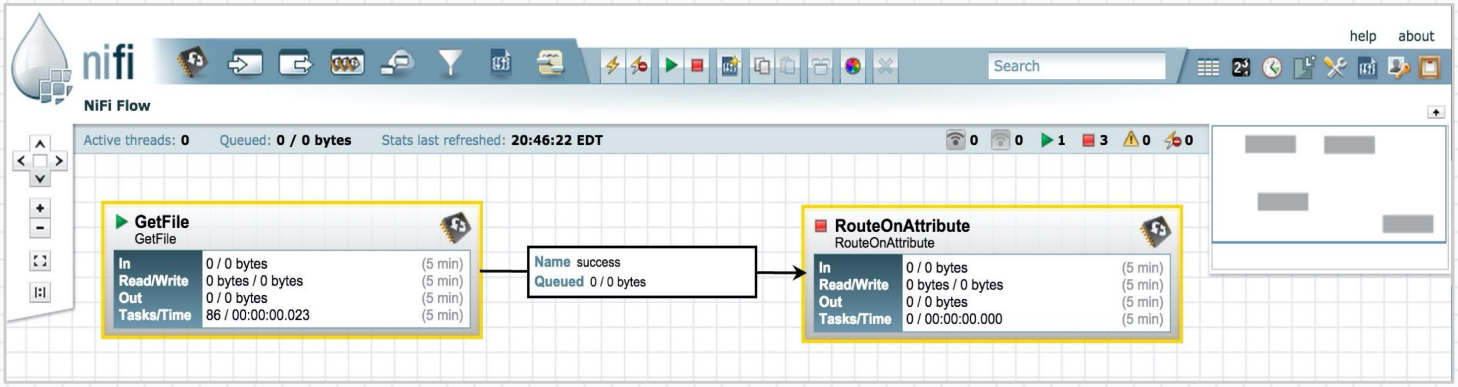


Análisis de un procesador (de grupo)

- El grupo de procesos proporciona un mecanismo para agrupar los componentes en una construcción lógica para organizar el flujo de datos de una manera que lo haga más comprensible desde un nivel superior. La siguiente imagen destaca los diferentes elementos que componen la anatomía de un Grupo de Procesos:

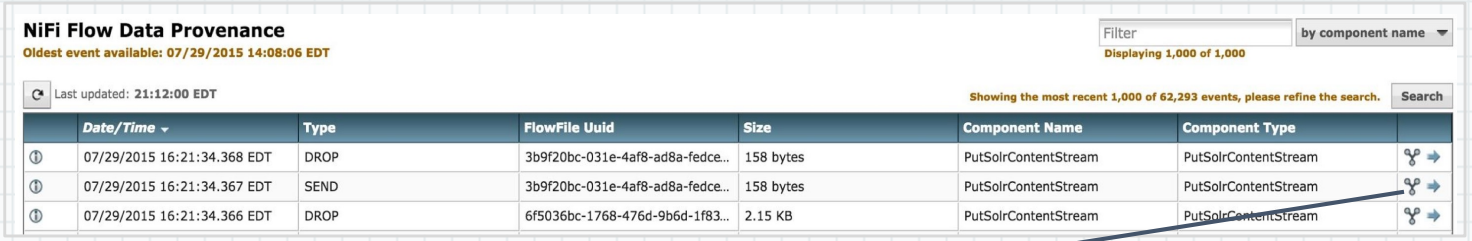


Visual Command & Control

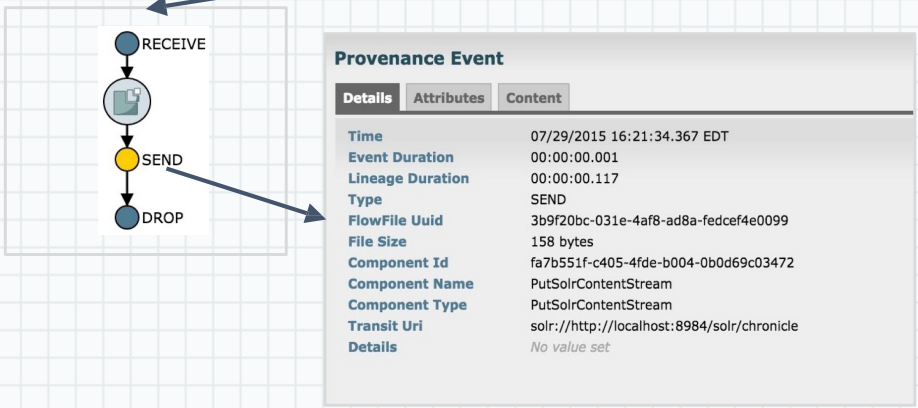


- Drag and drop processors to build a flow
- Start, stop, and configure components in real time
- View errors and corresponding error messages
- View statistics and health of data flow
- Create templates of common processor & connections

Provenance/Lineage

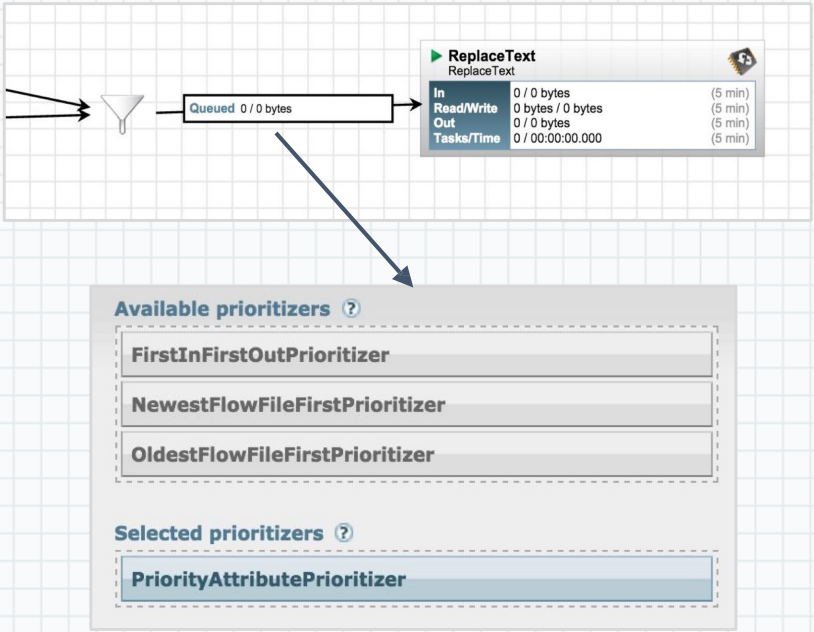


- Tracks data at each point as it flows through the system
- Records, indexes, and makes events available for display
- Handles fan-in/fan-out, i.e. merging and splitting data
- View attributes and content at given points in time



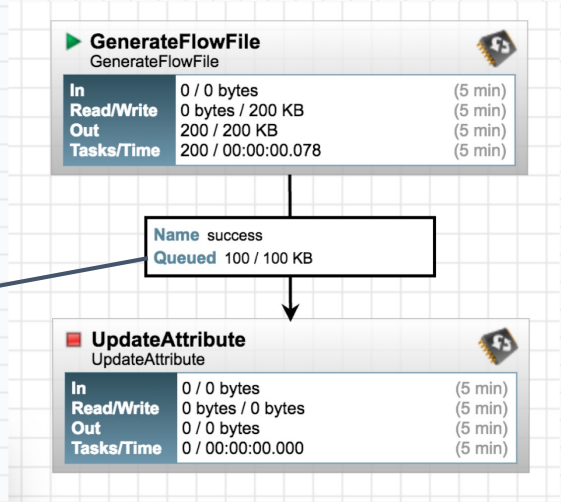
Prioritization

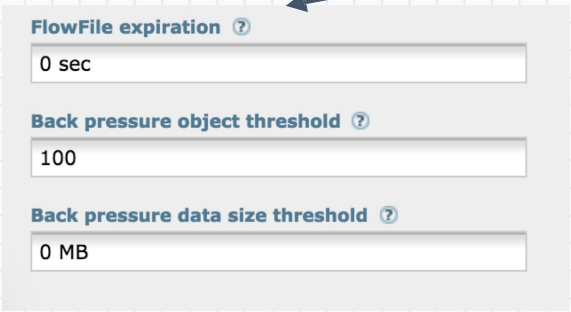
- Configure a prioritizer per connection
- Determine what is important for your data – time based, arrival order, importance of a data set
- Funnel many connections down to a single connection to prioritize across data sets
- Develop your own prioritizer if needed



Back-Pressure

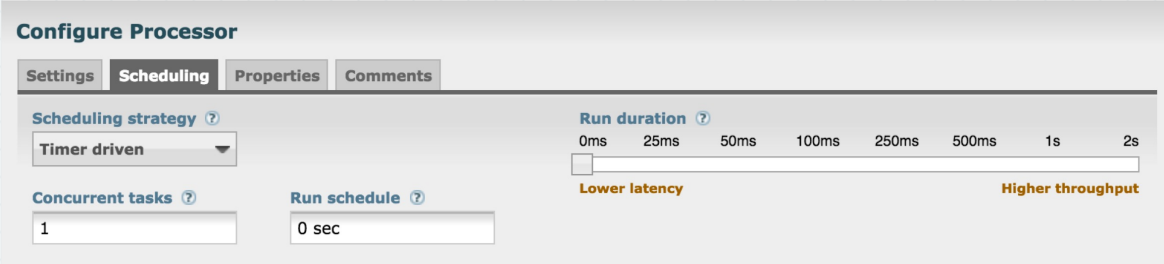
- Configure back-pressure per connection
- Based on number of FlowFiles or total size of FlowFiles
- Upstream processor no longer scheduled to run until below threshold





Latency vs. Throughput

- Choose between lower latency, or higher throughput on each processor
- Higher throughput allows framework to batch together all operations for the selected amount of time for improved performance
- Processor developer determines whether to support this by using @SupportsBatching annotation



Security

Control Plane

- Pluggable authentication
  - 2-Way SSL, LDAP, Kerberos
- Pluggable authorization
  - File-based authority provider out of the box
  - Multiple roles to defines access controls
- Audit trail of all user actions

Data Plane

- Optional 2-Way SSL between cluster nodes
- Optional 2-Way SSL on Site-To-Site connections (NiFi-to-NiFi)
- Encryption/Decryption of data through processors
- Provenance for audit trail of data

Extensibility

Built from the ground up with extensions in mind

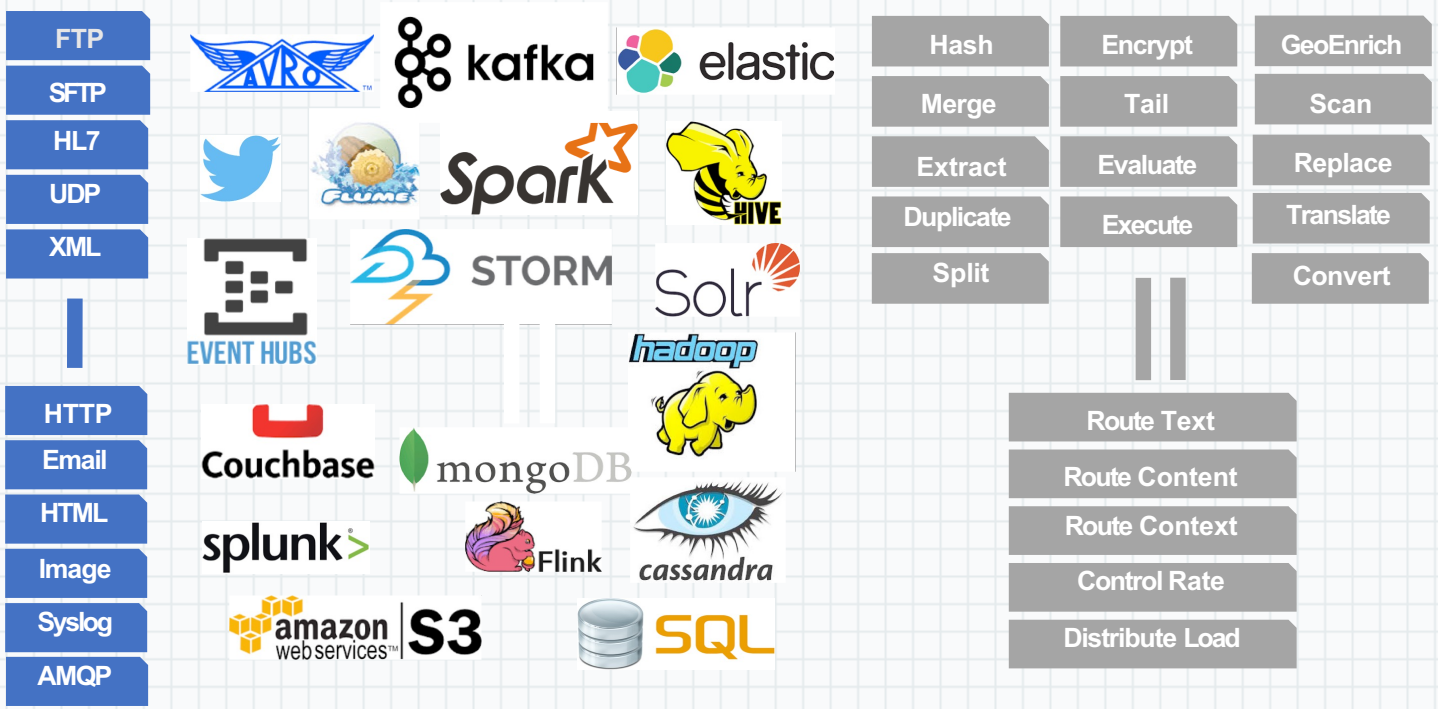
Service-loader pattern for…

- Processors
- Controller Services
- Reporting Tasks
- Prioritizers

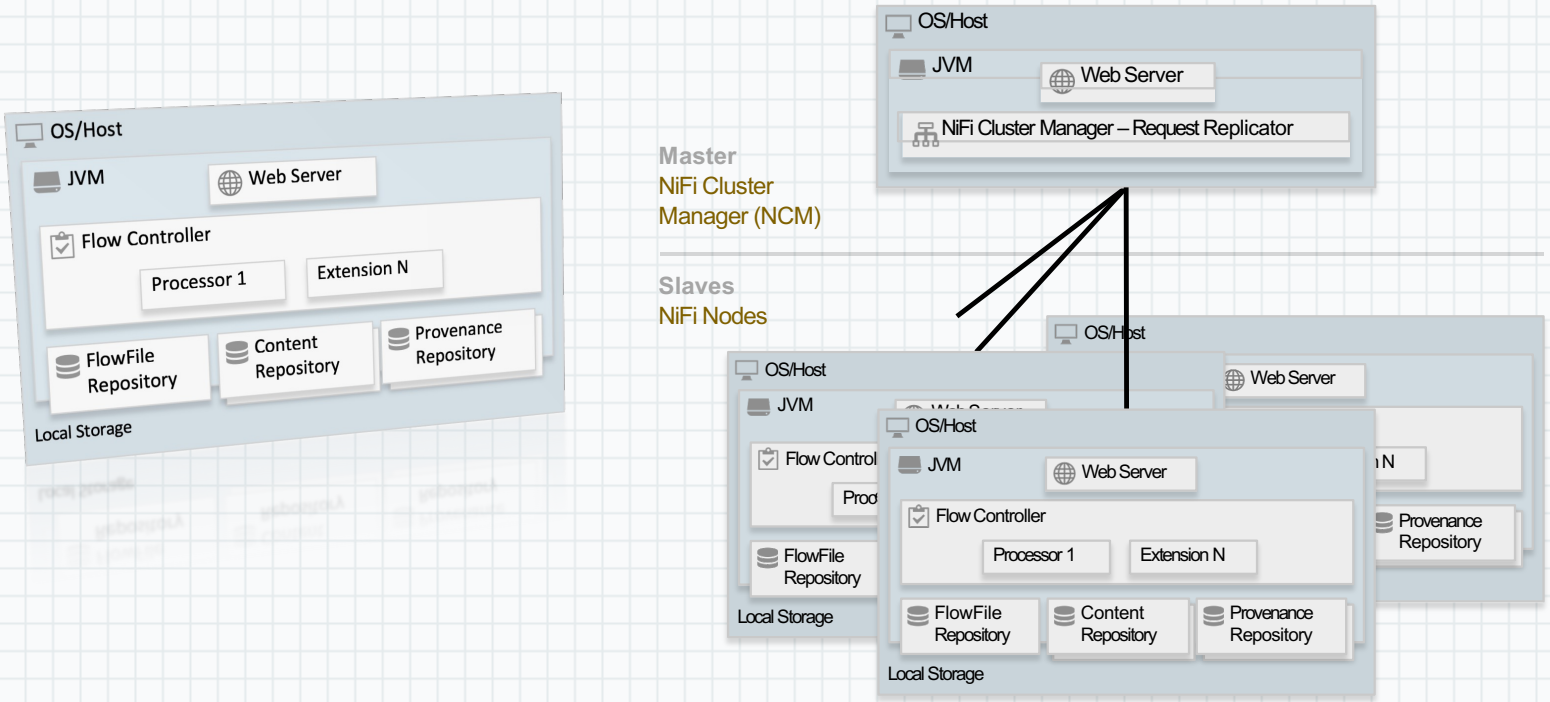
Extensions packaged as NiFi Archives (NARs)

- Deploy NiFi lib directory and restart
- Provides ClassLoader isolation
- Same model as standard components

Rapid Ecosystem Adoption: 130+ Processors



Architecture



Current Stream Processing Integrations

Spark Streaming - NiFi Spark Receiver

- https://github.com/apache/nifi/tree/master/nifi-external/nifi-spark-receiver

Storm – NiFi Spout & Bolt

- https://github.com/apache/nifi/tree/master/nifi-external/nifi-storm-spout

Flink – NiFi Source & Sink

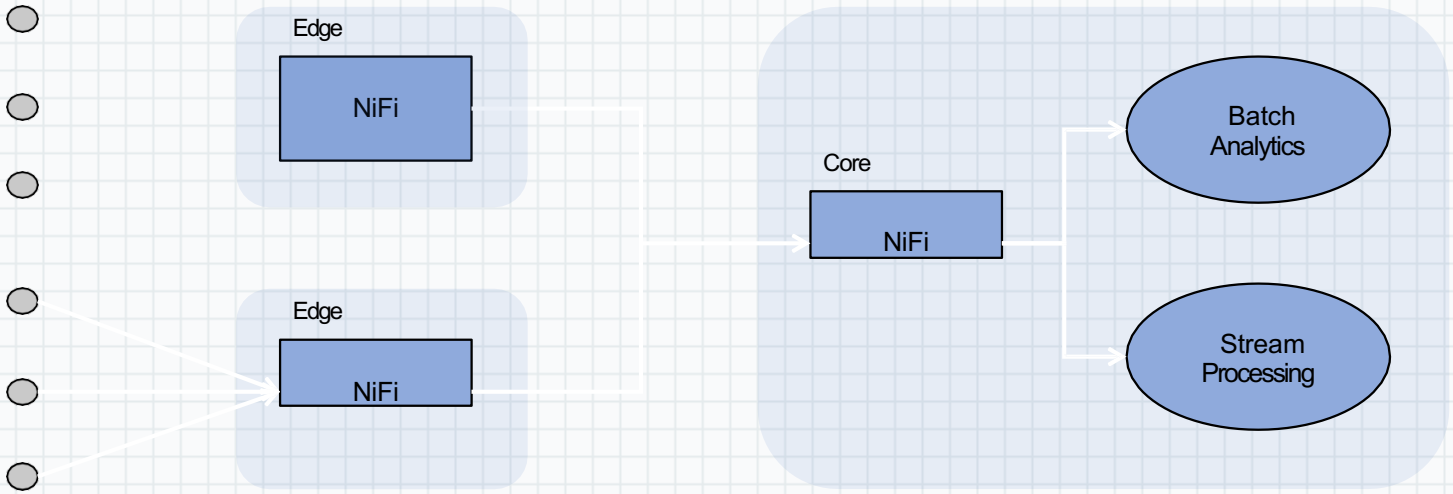
- https://github.com/apache/flink/tree/master/flink-streaming-connectors/flink-connector-nifi

Apex - NiFi Input Operators & Output Operators

- https://github.com/apache/incubator-apex-malhar/tree/master/contrib/src/main/java/com/datatorrent/contrib/nifi

Drive Data to Core for Analysis

- Drive data from sources to central data center for analysis
- Tiered collection approach at various locations, think regional data centers



Thanks!

- Resources
  - Apache NiFi Mailing Lists
    - https://nifi.apache.org/mailing_lists.html
  - Apache NiFi Documentation
    - https://nifi.apache.org/docs.html
  - Getting started developing extensions
    - https://cwiki.apache.org/confluence/display/NIFI/Maven+Projects+for+Extensions
    - https://nifi.apache.org/developer-guide.html
- Contact Info:
- docker run --name nifi \ -p 8080:8080 \ -d \ apache/nifi:latest

Enlaces

- https://nifi.apache.org/docs.html
- https://br.hortonworks.com/apache/nifi/
- https://www.batchiq.com/nifi-on-aws.html
- http://nifi.rocks


APACHE
nifi


THE
APACHE®
SOFTWARE FOUNDATION