

Apache NIFI

Agenda para las próximas sesiones

➤ Sesión 1

- Introducción a Apache Nifi
- Conceptos relacionados
- Prácticas

➤ Sesión 2

- Metadatos. Nifi Expression Language
- Requerimientos y Dimensionamiento
- Prácticas

➤ Sesión 3

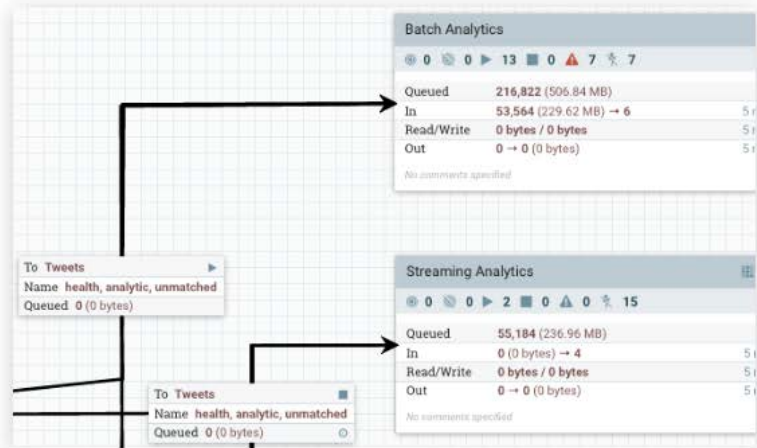
- Integración con Buscadores
- Nifi Registry
- Otras consideraciones

Apache Nifi

➤ Descripción en ASF:



An easy to use, powerful, and reliable system to process and distribute data.



1.16.0

- Released March 27, 2022
- Sources:
 - nifi-1.16.0-source-release.zip (asc, sha256, sha512)
- Binaries
 - nifi-1.16.0-bin.tar.gz (asc, sha256, sha512)
 - nifi-1.16.0-bin.zip (asc, sha256, sha512)
 - nifi-toolkit-1.16.0-bin.tar.gz (asc, sha256, sha512)
 - nifi-toolkit-1.16.0-bin.zip (asc, sha256, sha512)
 - nifi-stateless-1.16.0-bin.tar.gz (asc, sha256, sha512)
- Release Notes
- Migration Guidance

Version 1.16.0

Version 1.16.0 of Apache NiFi is a substantial feature, improvement, bug fix and security focused release.

Release Date: TBD (In RC prep/voting as of Mar 9 2022)

Highlights of the 1.16.0 release include:

- Framework Level Retry now supported. For many years users build flows in various ways to make retries happen for a configured number of attempts. Now this is easily and cleanly configured in the UI/API and simplifies the user experience and flow design considerably! To those waiting for years for this thank you for your patience.
- Disconnected nodes can now rejoin the cluster, updating their flow to match whatever the cluster gives it (after writing a backup of the disconnected node's flow locally, of course!) - even when the flow is significantly different
- The cluster now allows changes to the dataflow even while there are disconnected nodes
- New components to read CEF messages, easily perform geohashing on records, and to listen for SNMP traps.
- New components to make data enrichment far easier - especially when enrichment comes from web services.
- UI now shows when component properties have leading or trailing whitespace - very helpful when copying a value from elsewhere and pasting it into NiFi
- UI now shows when all data in a queue is penalized, and the processor will no longer run millions of times per minute in order to accomplish nothing
- MiNiFi Java can now hot-load nars just like NiFi.
- Tons of bug fixes and improvements for processors that interact with cloud services on AWS, Azure, and GCP as well as SFTP, FTP, Kafka and more.
- Stateless NiFi getting better and better with improved classloader isolation and ability to run off local nars only.
- Security: Aggressively update libraries such as Jetty, Netty, Logging related libs (log4j, slf4j, etc.), and moved to latest H2 Database to eliminate any worries about a vulnerability in the older line. Included a super slick mechanism to upgrade from the old database to the new one without the user having to worry about it.

<https://cwiki.apache.org/confluence/display/NIFI/Release+Notes#ReleaseNotes-Version1.16.0>

En pocas palabras

- **Objetivo: Automatizar el flujo de datos entre sistemas:**
 - Llevando ficheros (CSV ó JSON), o mensajes, a:
 - Bases de datos
 - Persistencia/Hadoop
 - Buscadores
 - ...o viceversa
- Interfaz gráfica: drag & drop
- Principal concepto y tarea: seleccionar y configurar procesadores
- Escalabilidad vertical y horizontal (clúster)
- Aporta garantía de envío entre procesadores. Se almacena toda la información en disco
- Presenta opciones adicionales: seguridad, trazabilidad de código y datos, etc.

¿Dónde debe estar el dato a mover?

- El dato tiene que estar accesible para el sistema donde se encuentre Apache Nifi.
- Tiene adaptadores para conectarse a sistemas remotos de almacenamiento (o donde se encuentre el dato) y cogerlos datos
- Si necesitamos instalar un agente en el sistema remoto para recoger los datos locales y enviarlos a otro sitio podríamos utilizar Apache MiNifi



A subproject of Apache NiFi to collect data where it originates.

About

Minifi—a subproject of Apache NiFi—is a complementary data collection approach that supplements the core tenets of NiFi in dataflow management, focusing on the collection of data at the source of its creation.

Specific goals for the initial thrust of the Minifi effort comprise:

- Small size and low resource consumption
- Central management of agents
- Generation of data provenance (full chain of custody of information)
- Integration with NiFi for follow-on dataflow management

Perspectives of the role of Minifi should be from the perspective of the agent acting immediately at, or directly adjacent to, source sensors, systems, or servers.

De donde viene....

[Productos](#)[Soluciones](#)[Clientes](#)[Servicios y soporte](#)[Sobre nosotros](#)[EMPEZAR](#)

Hortonworks to Acquire Onyara to Turn Internet of Anything Data Into Actionable Insights

New Hortonworks DataFlow powered by Apache NiFi will securely and easily collect, conduct and curate any data, from anything, anywhere

SANTA CLARA, Calif., ~~Aug. 25, 2015~~ Hortonworks, Inc. (NASDAQ: HDP), the leader in Open Enterprise Hadoop, today announced it has signed a definitive agreement to acquire Onyara, Inc., the creator of and key contributor to Apache NiFi, a top-level open source project. The acquisition will make it easy for customers to automate and secure data flows and to collect, conduct and curate real-time business insights and actions derived from *data in motion*. As a result of the acquisition, Hortonworks® is introducing Hortonworks DataFlow powered by Apache NiFi which is complementary to the industry's leading Open Enterprise Hadoop platform, Hortonworks Data Platform (HDP™). The transaction is expected to close in the third quarter of 2015.

Apache NiFi was made available through the NSA Technology Transfer Program in the fall of 2014. Over the past eight years, Onyara's engineers were the key contributors to the U.S. government software project that evolved into Apache NiFi. In July 2015, NiFi became a Top-Level Project, signifying that its community and technology have been successfully governed under the Apache Software Foundation.

"Nearly a decade ago when IoAT began to emerge, we saw an opportunity to harness the massive new data types from people, places and things, and deliver it to businesses in a uniquely secure and simple way," said Joe Witt, chief technology officer at Onyara. "We look forward to joining the Hortonworks team and continuing to work with the Apache community to advance NiFi."

Conceptos. Data Flow

- Es todo lo relacionado con el dato:
 - El dato de por si
 - Y sus metadatos: cómo son los datos que están involucrados más información adicional como fecha de creación, modificación, propietario, etc. Se almacenan como parejas de clave-valor
- En cuanto se crea se almacena en disco (y si la configuración es en clúster se notifica a los miembros su creación y definición)

Flow based programming: https://en.wikipedia.org/wiki/Flow-based_programming#Concepts

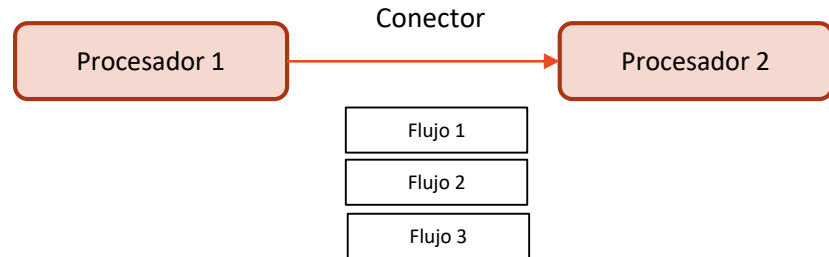
Conceptos: Procesador

- Es un conjunto de transformaciones y reglas que se aplican a los flujos de datos y que permiten crear nuevos juegos de datos.
- Hay un juego muy variado de procesadores que pueden ser aplicados a múltiples flujos de datos.
- En un proceso o flujo completo los procesadores se pasan entre si las referencias a los flujos de datos.
- Todos ellos están ejecutándose en paralelo en diferentes threads

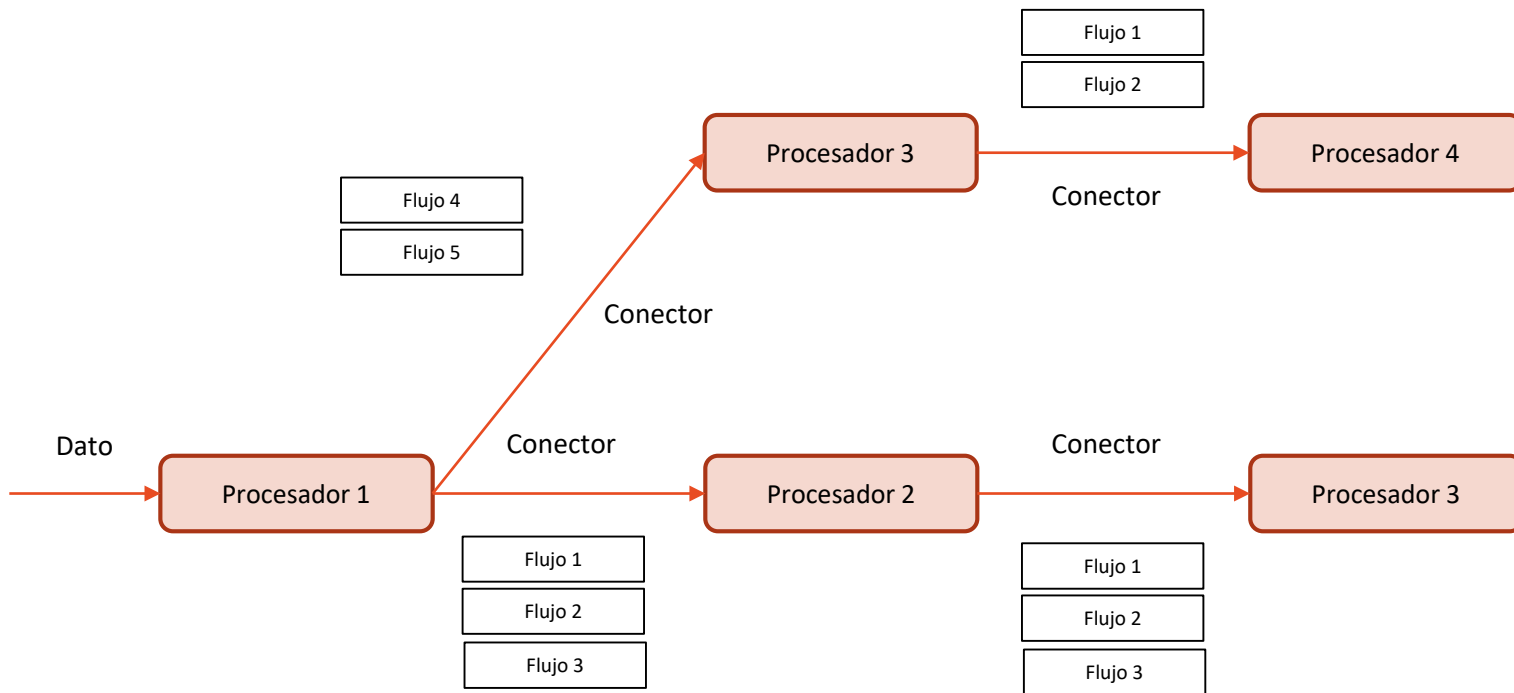


Conceptos: Conector

- Es, básicamente, una cola en la que se reflejan las condiciones por las que se envía la información de un procesador a otro: los flujos de datos
- Por ejemplo, define las reglas por las que se prioriza el paso de información: FIFO, LIFO, qué registros se pasan, qué registros no, etc.
- Tiene opciones para poder reaccionar ante la acumulación de datos para evitar convertirse en un cuello de botella (limitar, por ejemplo el número de mensajes que puedo recibir y si se acumulan más el anterior nodo se para momentáneamente...

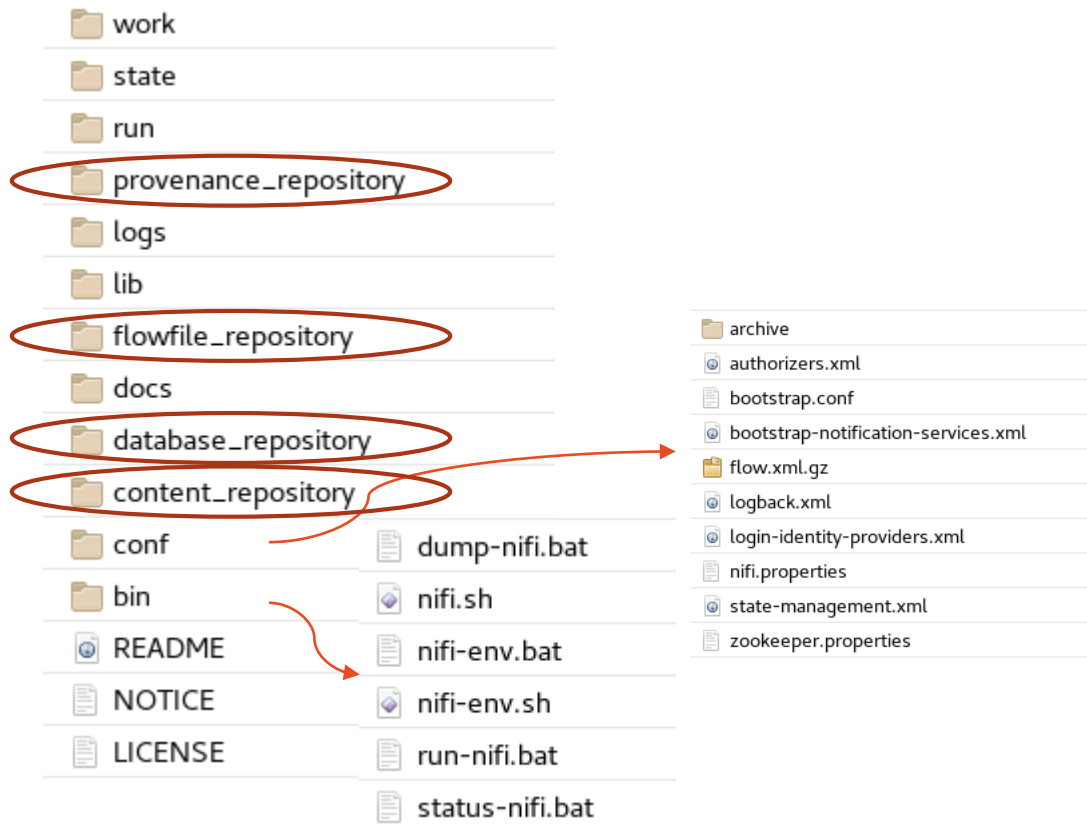


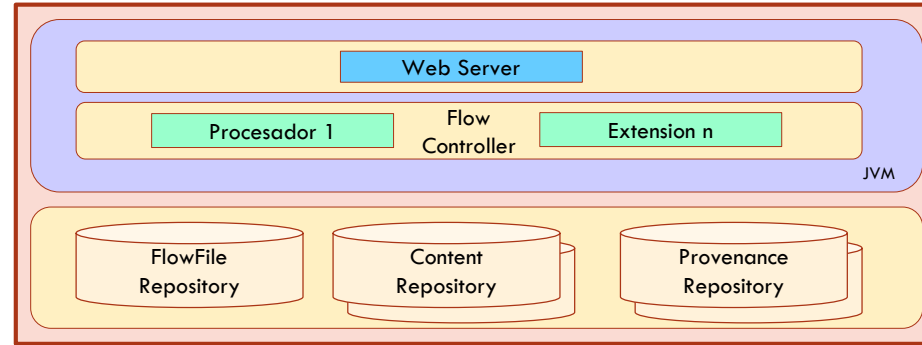
Y con todo esto soy capaz de componer...



Instalación

- Dependencias:
 - Windows, Linux, Mac
 - Java JRE 1.8
- Variables de entorno:
JAVA_HOME,
NIFI_HOME
- Descargar,
desempaquetar y
arrancar (bat, sh)





Web Server

Flow Controller, encargado de la gestión de threads para la ejecución de los procesos y las extensions (que se ejecutan dentro de la misma JVM)

FlowFile Repository, donde se almacena el estado del FlowFile activo. Es modificable, el de por defecto es un Write-Ahead Log para garantizar la consistencia en caso de caídas

Content Repository, donde se almacena el contenido de un Flowfile vivo. Almacena por defecto la información en distintas particiones para ofrecer la posibilidad de reducir la contención si usáramos un único volumen

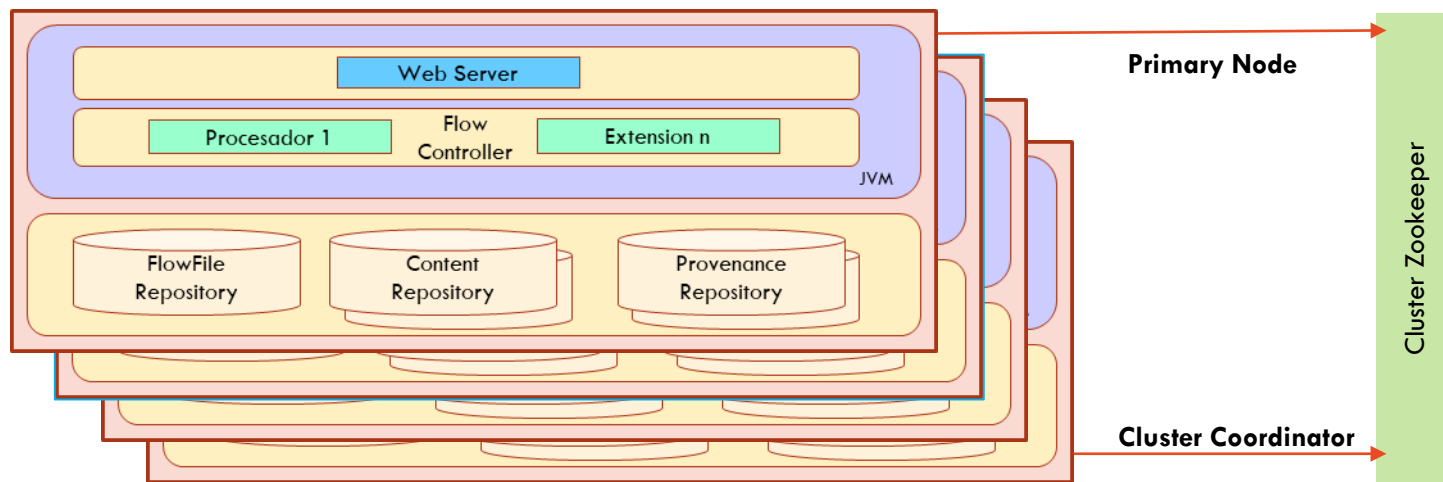
Provenance Repository, que actúa como un repositorio de las operaciones históricas. No se borra contenido, se crea uno nuevo adicional. La información está indexada para poder realizar búsquedas en su interior

Cluster

Utiliza Apache **Zookeeper** para determinar qué componentes del cluster están disponibles
Nifi lleva **embebido** Zookeeper pero puede hacer uso de un cluster externo

Zookeeper selecciona y asigna dos roles:

- **Cluster Coordinator**, recibe los heartbeats y estado de los miembros del cluster.
- **Primary Node** ejecuta aquellos procesadores que no son escalables o que no pueden ser ejecutados en paralelo en el cluster. (En un principio todos los procesadores se ejecutan en todos los miembros del cluster).



La **edición** de un DataFlow se puede hacer desde cualquiera de los nodos del cluster y es replicada a todos los miembros

Apariencia

The screenshot displays the Apache NiFi web interface. At the top, there is a toolbar with various icons for flow management and a status bar showing metrics like '0 / 0 bytes' and '11:51:23 C'. On the left, a 'Navigate' sidebar shows a search bar and a list of flows. Below it, an 'Operate' sidebar shows the 'Copia_Ficheros' process group with its ID '75e6eb3c-0165-1000-bce2-f2eaa8a8eca0' and a 'DELETE' button. The main canvas shows two flow components: 'Copia_Ficheros' and 'Manejando NIFI EL'. Both components have a table of metrics including 'Queued', 'In', 'Read/Write', and 'Out' with their respective values and durations. A dropdown menu is open on the right, showing options like 'Summary', 'Counters', 'Bulletin Board', 'Data Provenance', 'Controller Settings', 'Flow Configuration History', 'Templates', 'Help', and 'About'.

Copia_Ficheros

Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 → 0 (0 bytes)	5 min

Manejando NIFI EL

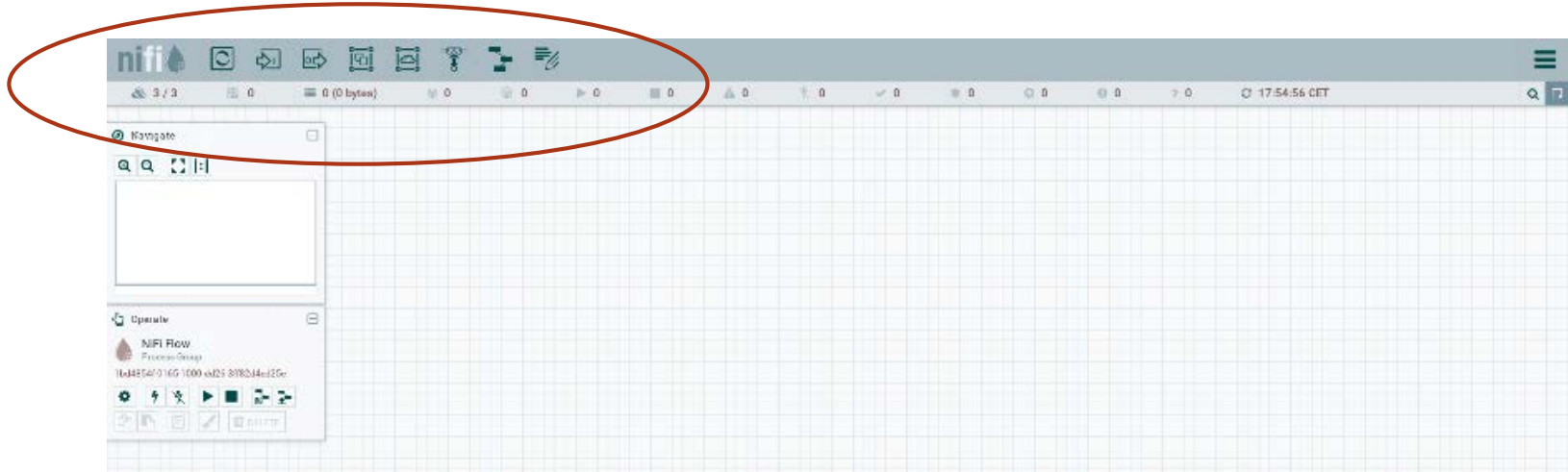
Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 → 0 (0 bytes)	5 min

Enrutamiento en base a contenido

Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 → 0 (0 bytes)	5 min

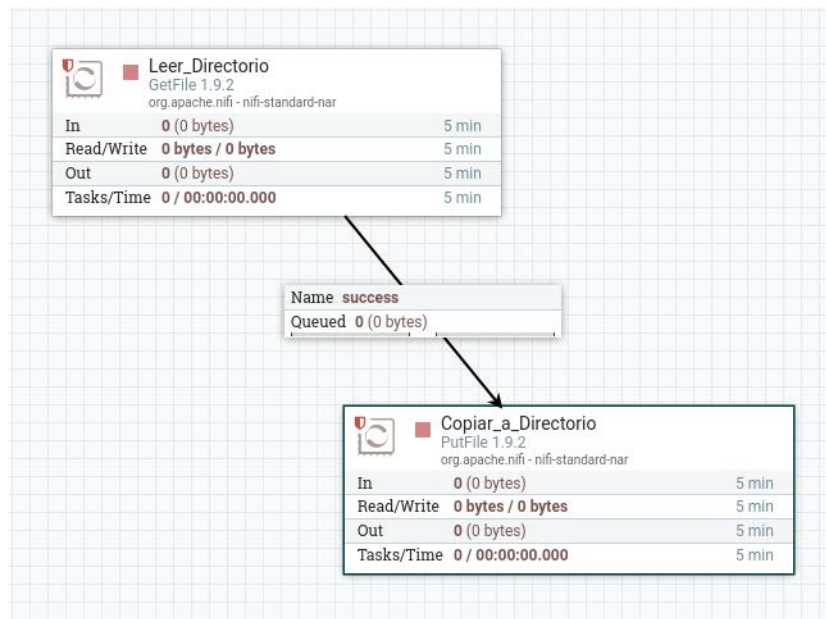
Apariencia de un cluster para el usuarios

- Exactamente igual que trabajar con un cluster de un único nodo.
- Trabajamos directamente con cualquiera de los nodos. El flujo se almacena localmente y se distribuye a través de la aplicación.



Objetivo para hoy: Ejemplo 1

- A medida que aparecen ficheros en un directorio analizarlos y moverlos a un directorio de salida



Instalación en local

- Descargamos el código de:
<https://nifi.apache.org/download.html>
- Instalamos en la plataforma que queramos
 - Directamente en nuestra plataforma del master (aconsejable por tener en un único sitio distintos componentes de los que recuperar datos o a los que enviar datos)
 - O directamente en el portátil
- Desempaquetamos y listos para ejecutar
- También existe la opción de una imagen Docker
<https://hub.docker.com/r/apache/nifi/>

Versiones disponibles

- En la plataforma del Master está desplegada la versión 1.13.2
- Si se instala la versión 1.16.0 hay que tener en cuenta dos cambios:
 - Incluye un certificado autofirmado para aceptar la conexión con el servidor. La conexión pasa a ser <https://127.0.0.1:8443/nifi>
 - Es necesario dar de alta un usuario en nifi (no es usuario de sistema operativo) a través del comando:

```
nifi.sh set-single-user-credentials <username> <password>
```

que se usará en la conexión a Nifi

Instalación

- Instalación prerequisites: Java JRE 1.8 (Oracle o OpenJDK)
- Desempaquetar y listo para ejecutar
- Asegurarse de que el JRE que se ejecuta es el válido si se cuenta con varias versiones de Java
- Si se ejecuta en imagen virtual asegurar que el firewall no impide el tráfico

```
[umaster@ibmuamdocke bin]$ pwd
/home/umaster/Master_Platform/nifi-1.16.0/bin
[umaster@ibmuamdocke bin]$ ./nifi.sh start
nifi.sh: JAVA_HOME not set; results may vary

Java home:
NiFi home: /home/umaster/Master_Platform/nifi-1.16.0

Bootstrap Config File: /home/umaster/Master_Platform/nifi-1.16.0/conf/bootstrap.conf

[umaster@ibmuamdocke bin]$ █
```

Arranque

```
[umaster@ibmuamdocker bin]$ ./nifi.sh start  
nifi.sh: JAVA_HOME not set; results may vary
```

```
Java home:  
NiFi home: /home/umaster/Master_Platform/nifi-1.16.0
```

```
Bootstrap Config File: /home/umaster/Master_Platform/nifi-1.16.0/conf/bootstrap.conf
```

```
[umaster@ibmuamdocker bin]$ tail -f ../logs/nifi-app.log  
2022-04-07 23:11:22,726 INFO [main] o.e.jetty.server.handler.ContextHandler Started o.e.j.w.WebAppContext@1a28b346{nifi-error/,file:///home/umaster/Master_Platform/nifi-1.16.0/work/jetty/nifi-web-error-1.16.0.war/webapp/,AVAILABLE}{./work/nar/extensions/nifi-server-nar-1.16.0.nar-unpacked/NAR-INF/bundled-dependencies/nifi-web-error-1.16.0.war}  
2022-04-07 23:11:23,043 INFO [main] o.eclipse.jetty.server.AbstractConnector Started ServerConnector@2add4d24{SSL, (ssl, http/1.1)}{127.0.0.1:8443}  
2022-04-07 23:11:23,043 INFO [main] org.eclipse.jetty.server.Server Started @38473ms  
2022-04-07 23:11:23,114 INFO [main] org.apache.nifi.nar.NarAutoLoader Starting NAR Auto-Loader for directory ./extensions ...  
2022-04-07 23:11:23,115 INFO [main] org.apache.nifi.web.server.JettyServer NiFi has started. The UI is available at the following URLs:  
2022-04-07 23:11:23,115 INFO [main] org.apache.nifi.web.server.JettyServer https://127.0.0.1:8443/nifi  
2022-04-07 23:11:23,117 INFO [main] org.apache.nifi.BootstrapListener Successfully initiated communication with Bootstrap  
2022-04-07 23:11:23,117 INFO [main] org.apache.nifi.NiFi Started Application Controller in 23.669 seconds (23669253669 ns)  
2022-04-07 23:11:36,481 INFO [pool-7-thread-1] o.a.n.c.r.WriteAheadFlowFileRepository Initiating checkpoint of FlowFile Repository  
2022-04-07 23:11:36,483 INFO [pool-7-thread-1] o.a.n.c.r.WriteAheadFlowFileRepository Successfully checkpointed FlowFile Repository with 0 records in 1 milliseconds  
[umaster@ibmuamdocker bin]$
```

- Definimos usuario nifi/nifipassword
- Nos conectamos en nuestro caso con `https://127.0.0.1:8443/nifi`

Instalación como docker

- Disponible en:

<https://hub.docker.com/r/apache/nifi/>

- Si se desea incluir el docker en la misma red en la que está desplegado el entorno hadoop deberemos indicarlo así:

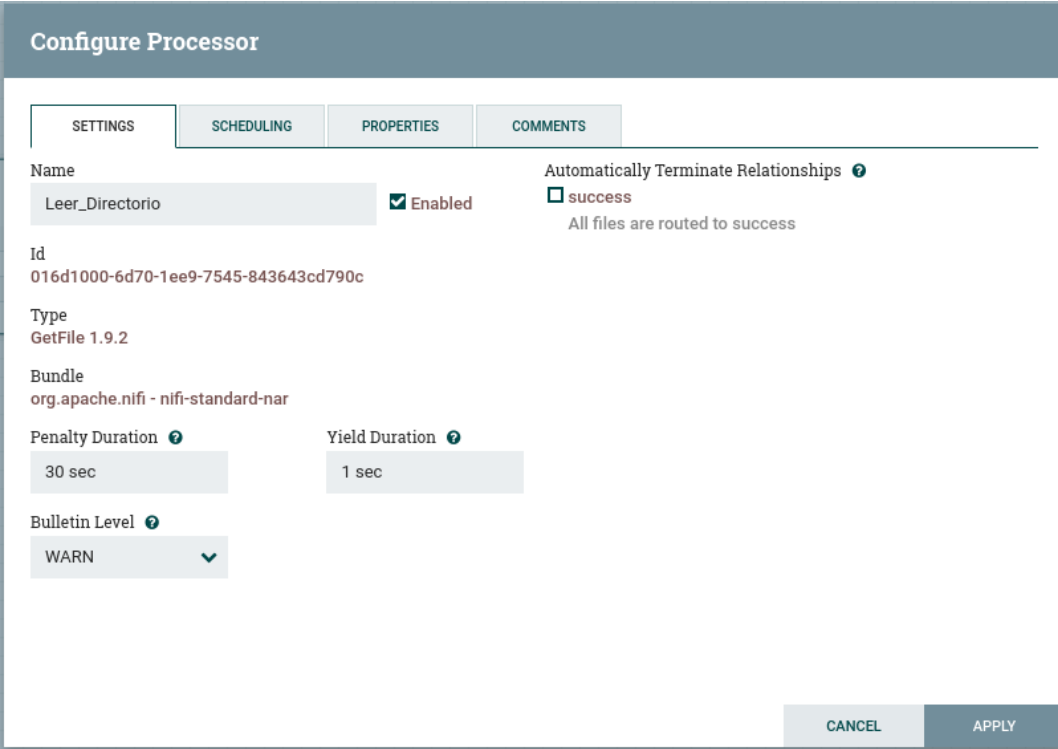
```
docker run --name nifi -p 8080 --network=master_platform_default -d  
apache/nifi:latest
```

- Si queremos crear un cluster utilizando docker compose podemos acudir a:

<https://www.nifi.rocks/apache-nifi-docker-compose-cluster/>

Seleccionando y configurando un procesador (I)

- Cogemos de la paleta el procesador GetFile



The screenshot shows the 'Configure Processor' dialog box for the 'GetFile' processor. The 'SETTINGS' tab is active, displaying various configuration options. The 'Name' field is set to 'Leer_Directorio', and the 'Enabled' checkbox is checked. The 'Id' field shows a UUID. The 'Type' is 'GetFile 1.9.2' and the 'Bundle' is 'org.apache.nifi - nifi-standard-nar'. The 'Penalty Duration' is set to '30 sec' and the 'Yield Duration' is set to '1 sec'. The 'Bulletin Level' is set to 'WARN'. The 'Automatically Terminate Relationships' checkbox is unchecked, and the 'success' radio button is selected, with a note that 'All files are routed to success'. 'CANCEL' and 'APPLY' buttons are at the bottom right.

Configure Processor			
SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Name	Leer_Directorio		
Id	016d1000-6d70-1ee9-7545-843643cd790c		
Type	GetFile 1.9.2		
Bundle	org.apache.nifi - nifi-standard-nar		
Penalty Duration	30 sec		
Yield Duration	1 sec		
Bulletin Level	WARN		
Automatically Terminate Relationships	<input type="checkbox"/> success All files are routed to success		
CANCEL		APPLY	

Configurando GetFile (II)

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Scheduling Strategy ?

Timer driven

Concurrent Tasks ?1

Run Schedule ?0 sec

Execution ?

All nodes

CANCEL

APPLY

Configurando GetFile (III)

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property		Value
Input Directory	?	Datos/In
File Filter	?	[\.\]*
Path Filter	?	No value set
Batch Size	?	10
Keep Source File	?	false
Recurse Subdirectories	?	true
Polling Interval	?	0 sec
Ignore Hidden Files	?	true
Minimum File Age	?	0 sec
Maximum File Age	?	No value set
Minimum File Size	?	0 B
Maximum File Size	?	No value set

CANCEL

APPLY

Configurando GetFile (IV)

The image shows a 'Configure Processor' dialog box with four tabs: 'SETTINGS', 'SCHEDULING', 'PROPERTIES', and 'COMMENTS'. The 'COMMENTS' tab is currently selected. The main area of the dialog is a large, empty light gray rectangle, likely intended for text input. At the bottom right, there are two buttons: 'CANCEL' and 'APPLY'.

Configurando PutFile (I)

- Cogemos de la paleta el procesador PutFile

The screenshot shows the 'Configure Processor' dialog for the PutFile processor. The 'SETTINGS' tab is active, displaying various configuration options. The 'Name' field is 'Copiar_a_Directorio', 'Id' is '016d1000-102d-1eee-3ba2-c720d32bdbb7', and 'Type' is 'PutFile 1.9.2'. The 'Bundle' is 'org.apache.nifi - nifi-standard-nar'. The 'Penalty Duration' is '30 sec' and 'Yield Duration' is '1 sec'. The 'Bulletin Level' is set to 'WARN'. On the right, the 'Automatically Terminate Relationships' section has checkboxes for 'failure' and 'success', both of which are checked. The 'failure' checkbox is accompanied by the text 'Files that could not be written to the output directory for some reason are transferred to this relationship', and the 'success' checkbox is accompanied by 'Files that have been successfully written to the output directory are transferred to this relationship'. At the bottom right, there are 'CANCEL' and 'APPLY' buttons.

Configure Processor			
SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Name Copiar_a_Directorio		<input checked="" type="checkbox"/> Enabled	Automatically Terminate Relationships <input checked="" type="checkbox"/> failure Files that could not be written to the output directory for some reason are transferred to this relationship <input checked="" type="checkbox"/> success Files that have been successfully written to the output directory are transferred to this relationship
Id 016d1000-102d-1eee-3ba2-c720d32bdbb7			
Type PutFile 1.9.2			
Bundle org.apache.nifi - nifi-standard-nar			
Penalty Duration 30 sec	Yield Duration 1 sec		
Bulletin Level WARN			

Configurando PutFile (II)

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Scheduling Strategy ?

Timer driven ▼

Concurrent Tasks ?

1

Execution ?

All nodes ▼

Run Duration ?

0ms 25ms 50ms 100ms 250ms 500ms 1s 2s

Lower latencyHigher throughput

Run Schedule ?

0 sec

CANCEL

APPLY

Configurando PutFile (III)

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property		Value	
Directory	?	Datos/Out	
Conflict Resolution Strategy	?	fail	
Create Missing Directories	?	true	
Maximum File Count	?	No value set	
Last Modified Time	?	No value set	
Permissions	?	No value set	
Owner	?	No value set	
Group	?	No value set	

CANCEL

APPLY

Configurando PutFile (IV)

The image shows a 'Configure Processor' dialog box with a dark blue header. Below the header is a tabbed interface with four tabs: 'SETTINGS', 'SCHEDULING', 'PROPERTIES', and 'COMMENTS'. The 'COMMENTS' tab is currently selected, showing a large, empty light gray text area. At the bottom right of the dialog are two buttons: 'CANCEL' and 'APPLY'.

Conectando (I)

- Conectamos ambos procesadores bajo la condición de success (éxito)

The screenshot shows a 'Configure Connection' dialog box with two tabs: 'DETAILS' and 'SETTINGS'. The 'SETTINGS' tab is active. It contains two columns of configuration options. The left column is for the 'From Processor' and the right column is for the 'To Processor'. Both columns have identical settings: 'Leer_Directorio' and 'Copiar_a_Directorio' for the processor type, 'GetFile' and 'PutFile' for the operation, 'Within Group' for the scope, 'Coger_y_Dejar_Fichero' for the file handling, and 'For Relationships' with a checked 'success' condition. At the bottom right, there are 'CANCEL' and 'APPLY' buttons.

From Processor	To Processor
Leer_Directorio	Copiar_a_Directorio
GetFile	PutFile
Within Group	Within Group
Coger_y_Dejar_Fichero	Coger_y_Dejar_Fichero
For Relationships	
<input checked="" type="checkbox"/> success	

Conectando (II)

Configure Connection

DETAILS

SETTINGS

Name

Id

016d1000-c6d0-1eef-cfa5-8b630adf3dae

FlowFile Expiration ?

0 sec

Back Pressure

Object Threshold ?

10000

Size Threshold ?

1 GB

Load Balance Strategy ?

Do not load balance

Available Prioritizers ?

FirstInFirstOutPrioritizer

NewestFlowFileFirstPrioritizer

OldestFlowFileFirstPrioritizer

PriorityAttributePrioritizer

Selected Prioritizers ?

CANCEL

APPLY

Ejecutando el flujo

- Activa cada uno de los componentes y comprueba qué pasa cuando están parados
- Comprueba la actualización de estadísticas de ejecución
- ¿Cómo podrías completar el flujo?

Procesadores (I)

No es una lista detallada

Data Transformation	Routing and Mediation	Database Access	Attribute Extraction	System Interaction
CompressContent	ControlRate	ConvertJSONToSQL	EvaluateJsonPath	ExecuteProcess
ConvertCharacterSet	DetectDuplicate	ExecuteSQL	EvaluateXPath	ExecuteStreamCommand
EncryptContent	DistributeLoad	PutSQL	EvaluateXQuery	
ReplaceText	MonitorActivity	Select / Put HiveQL	ExtractText	
TransformXml	RouteOnAttribute	Hbase	HashAttribute	
JoltTransformJSON	ScanAttribute	Cassandra	HashContent	
Convert to/from Avro, Json	RouteOnContent	DynamoDB	IdentifyMimeType	
	ScanContent	PutXXXXRecord Mgmt	UpdateAttribute	
	ValidateXml			

Procesadores (II)

No es una lista detallada

Data Ingestion	Data Ingestion	Data Egress	Splitting and Aggregation	HTTP
GetFile	ListHDFS / FetchHDFS	PutEmail	SplitText	GetHTTP
GetFTP	FetchS3Object	PutFile	SplitJson	ListenHTTP
GetSFTP	GetKafka	PutFTP / PutSFTP	SplitXml	InvokeHTTP
GetJMSQueue	GetMongo	PutJMS	UnpackContent	PostHTTP
GetJMSTopic	GetTwitter	PutSQL	MergeContent	HandleHttpRequest / HandleHttpResponse
GetHTTP		PutKafka	SegmentContent	
ListenHTTP		PutMongo	SplitContent	
ListenUDP		MQTT	MergeRecord	
GetHDFS			SplitRecord	

Procesadores (III)

No es una lista detallada

Amazon Web Services	Azure Cloud Services	Google Cloud Services	Searchers	Others
FetchS3Object	Blob Storage Mgmt	Google Cloud Bucket Mgmt	SolR	Kudu
PutS3Object	Event Hub Mgmt		ElasticSearch	Druid
PutSNS	Queue Storage Mgmt			Kinesis
GetSQS	Amazon Lambda			
PutSQS				
DeleteSQS				
AWS Gateway Mgmt				
Amazon Lambda				

Mejoras para la productividad del desarrollador

- El entorno gráfico nos permite copiar & pegar procesadores (rapidez)
- Podemos agrupar nodos procesadores en un único “Processor group” para poder hacer más manejable la consola gráfica
- Podemos exportar e importar grupos de procesadores en un fichero especial denominado “Template”

Manejando Templates

The screenshot shows the Apache NiFi web interface. The top navigation bar includes links to various tools and a breadcrumb trail. The main workspace displays a flow with two processors connected by a relationship. The left sidebar has a 'Navigate' panel and an 'Operate' panel. The 'Operate' panel shows 'Multiple components selected' and a 'Create Template' button. The bottom status bar indicates 'NiFi Flow' and 'Copia_Ficheros'.

Leer ficheros de sourcedir
GetFile 1.7.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

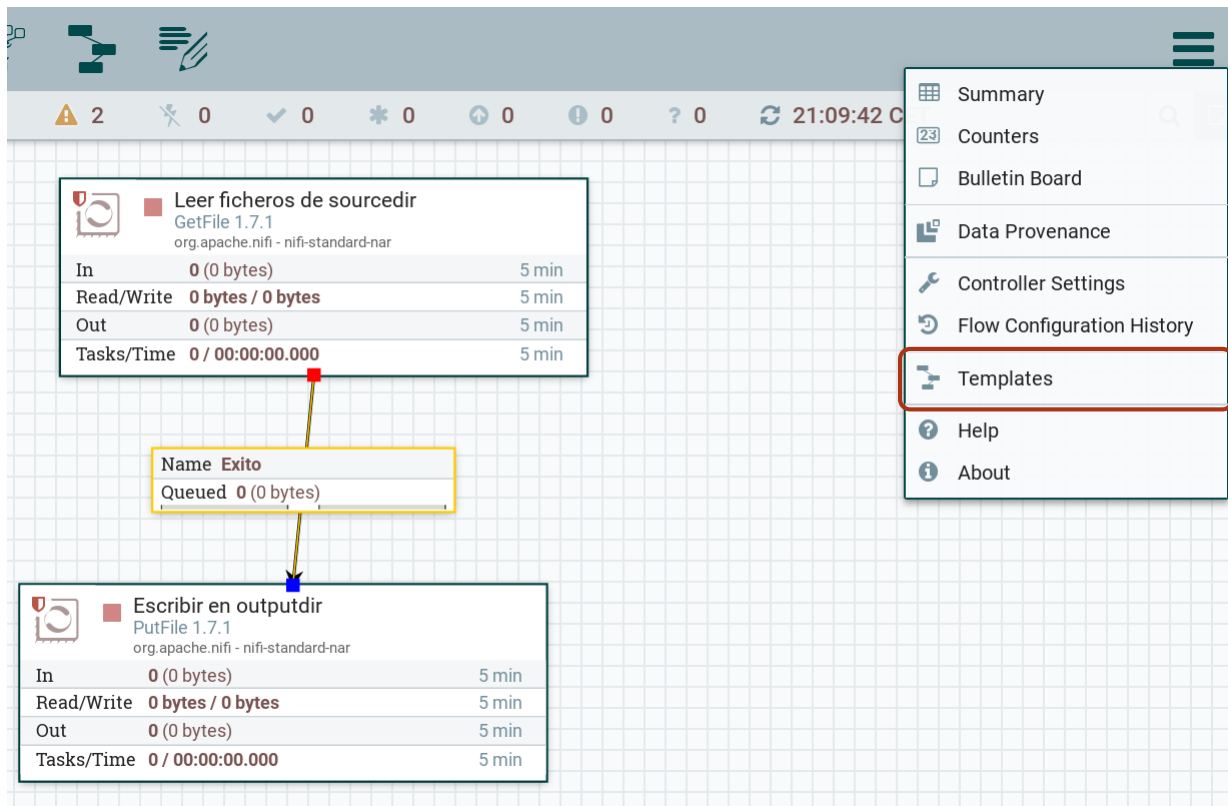
Name Exit
Queued 0 (0 bytes)

Escribir en outputdir
PutFile 1.7.1
org.apache.nifi - nifi-standard-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

- El Template nos va a permitir agrupar todo el trabajo en un único componente que podremos intercambiar con otros desarrolladores, entornos, etc.
- Marcamos todos los elementos a incluir y pulsamos la opción e Crear Template. Le asignamos un nombre y nos vamos a la opción de gestionar templates

Invocando Templates



Gestión de Templates

- Con esta gestión podremos exportar a fichero en cualquier momento el estado de mi flujo.
- A través de las opciones de la consola puedo reutilizar ese template cuando el desarrollador lo desee
- Hay opciones adicionales para poder ver el estado de un flujo (componentes) en cualquier momento en el tiempo



The screenshot shows the 'NiFi Templates' interface. At the top right are icons for a new template and a close button. Below the title, it says 'Displaying 1 of 1'. There is a 'Filter' input field and a dropdown menu set to 'by name'. A table lists the templates with columns for Date/Time, Name, Description, and Process Group Id. The table contains one entry: a template named 'Template_de_ejemplo' created on 11/01/2018 at 21:13:32 CET, with the description 'Ejemplo de prueba de creación de template' and process group ID '75e6b3c-0165-1000-bce2-f2eaa8a8eca0'. Action icons for edit and delete are visible at the end of the row.

Date/Time	Name	Description	Process Group Id
11/01/2018 21:13:32 CET	Template_de_ejemplo	Ejemplo de prueba de creación de template	75e6b3c-0165-1000-bce2-f2eaa8a8eca0

Gestión y Monitorización

- Consola de trabajo
- Nifi Rest API. Permite obtener información de la instalación y su monitorización:

GET /nifi-api/system-diagnostics

GET /nifi-api/flow/status

GET /nifi-api/flow/about

GET /nifi-api/flow/bulletin-board

GET /nifi-api/flow/cluster/summary

GET /nifi-api/flow/templates

Conclusiones

- Fácil de instalar y ejecutar en un único nodo
- Sencillo para crear flujos de datos en los que:
 - Los datos no son modificados, son *movidos*
 - Se pueden pasar parámetros de un procesador a otro (son metadatos de los datos, no específicamente el contenido)
 - Podemos ejecutar comandos de sistema operativo pasándole el fichero o simplemente ejecutándolo sin aceptar información de conexión (*execute stream command* o *execute process*)
- Múltiples fuentes y destinos:
 - Local y cloud
 - Combinando diferentes tecnologías

Referencias interesantes

- <http://nifi.apache.org>
- <https://es.hortonworks.com/apache/nifi/>