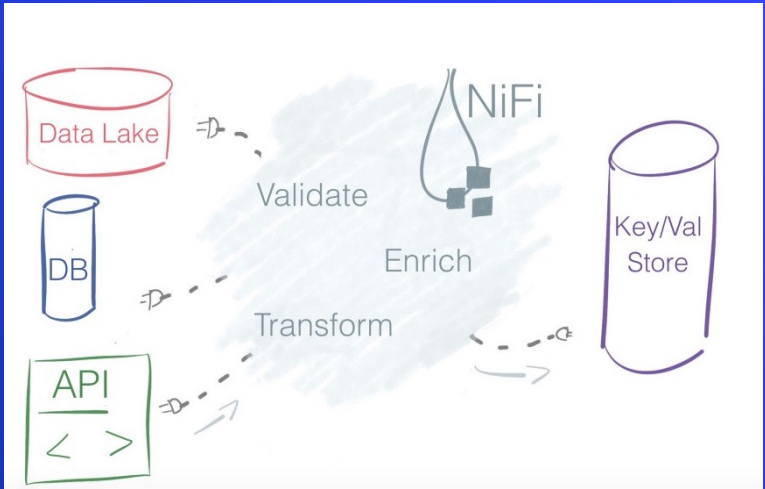
Apache NIFI Introduction

Introduction

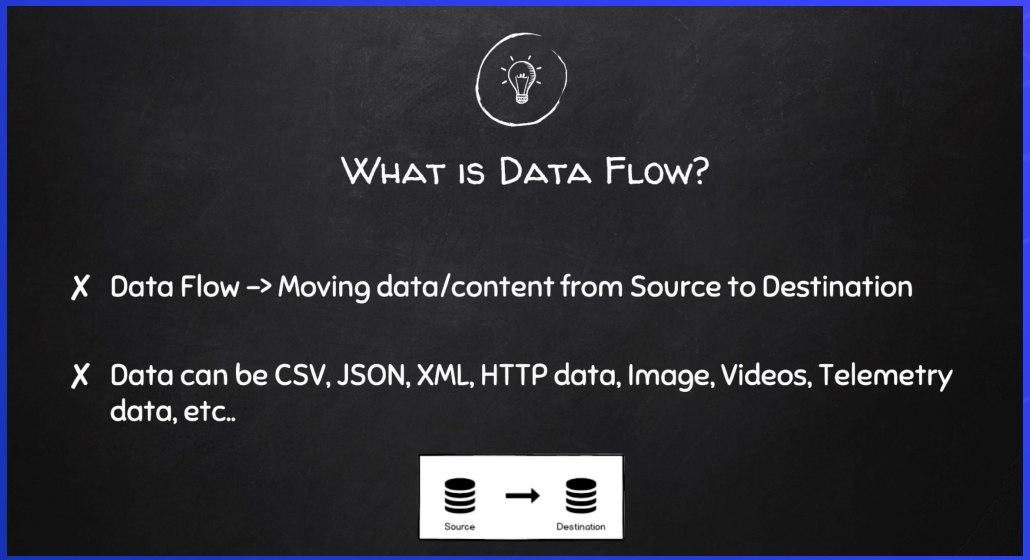
Avec l'augmentation du nombre et de la complexité des systèmes producteurs et consommateurs de données dans les organisations, il devient nécessaire de disposer d'un outil centralisé pour assurer des échanges de qualité entre ces différents systèmes.

NiFi est une technologie qui répond à ce besoin en permettant de gérer et d'automatiser ces flux de données, et participe à l'amélioration de la gouvernance des données.



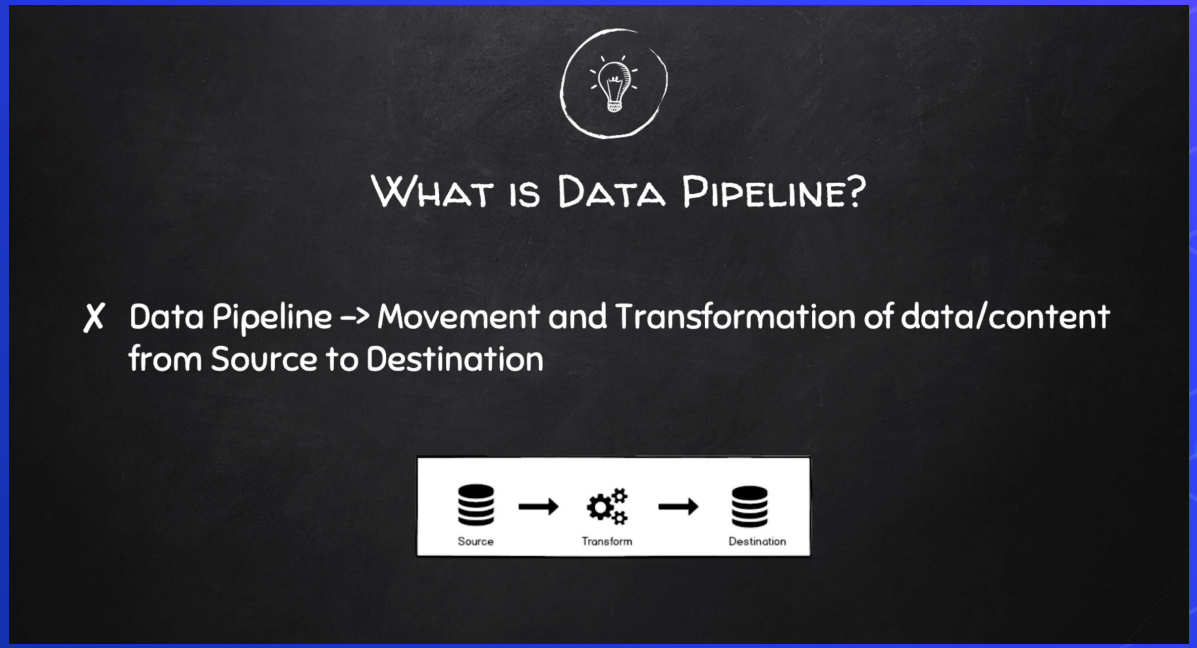
Data Flow

Le flux de données (data flow) fait référence à la manière dont les données se déplacent dans un système ou une application. Il décrit la séquence dans laquelle les données se déplacent d'une source vers une destination.



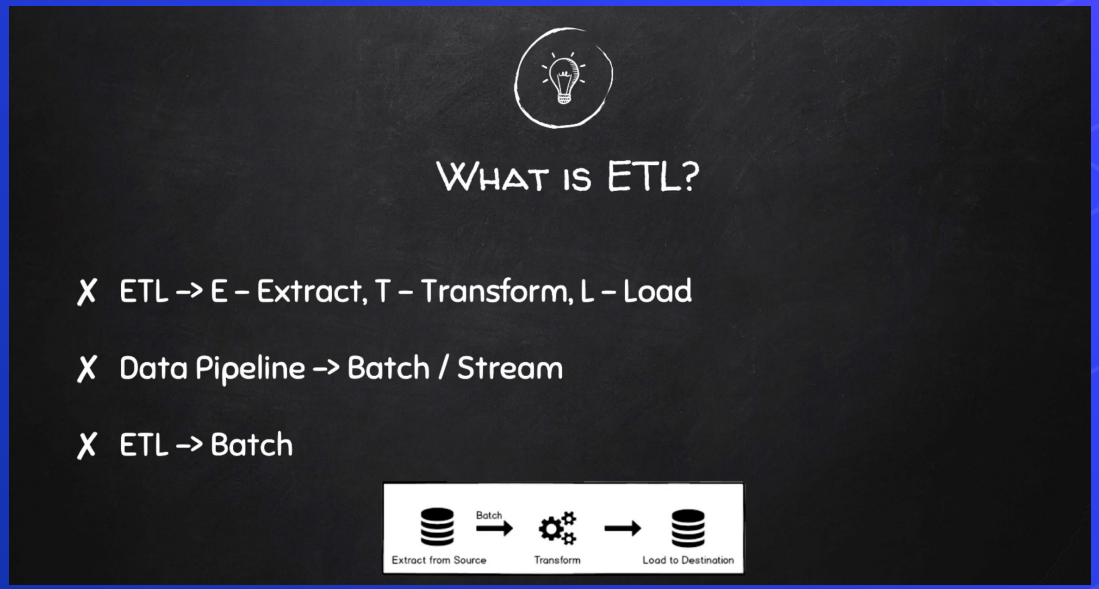
Data Pipeline

Un Data Pipeline (ou pipeline de données) est un concept utilisé dans le domaine de l'informatique et du traitement des données pour décrire un flux de travail automatisé qui permet de collecter, transformer et acheminer les données d'un système à un autre.

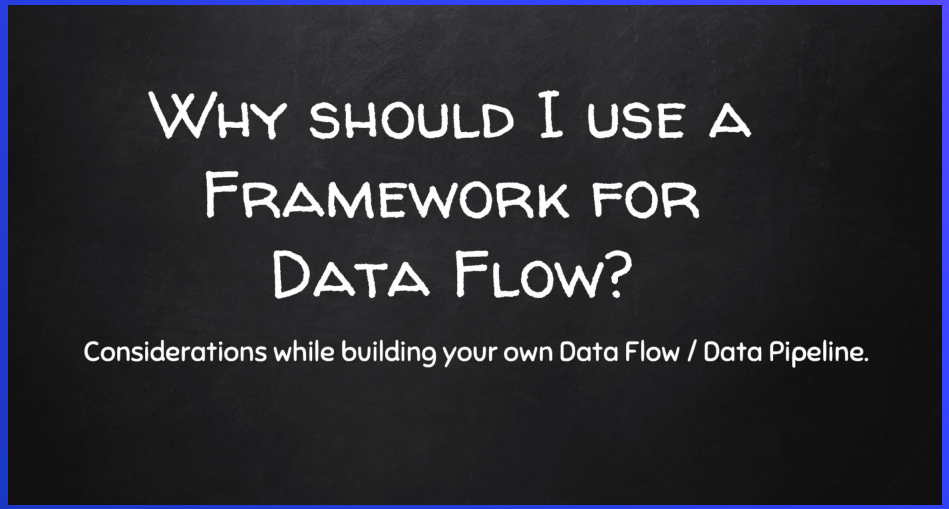


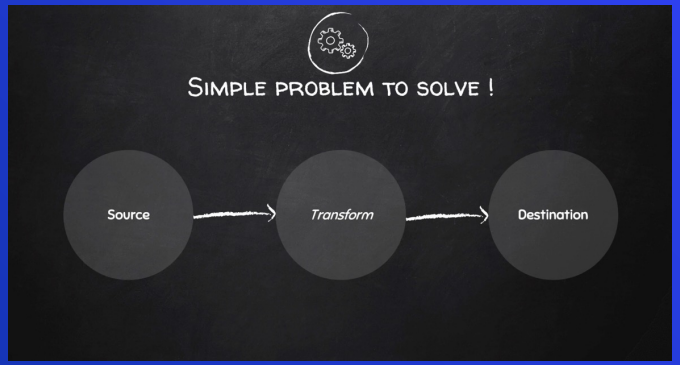
ETL

ETL est l'acronyme de Extract, Transform, Load (Extraction, Transformation, Chargement en français). Il s'agit d'un processus couramment utilisé dans le domaine du traitement des données pour extraire des données à partir de différentes sources, les transformer conformément aux besoins et les charger dans une destination cible, généralement une base de données ou un entrepôt de données.



Pourquoi un Framwork pour les Data Flow





Considération

Four V's

- ✗ Volume –> refers to the vast amounts of data generated every second.
- ✗ Velocity –> refers to the speed at which new data is generated and the speed at which data moves around.
- ✗ Variety –> refers to the different types of data we can now use.
- ✗ Veracity – refers to the messiness or trustworthiness of the data.

Considerations

- ✗ Support for multiple data format – CSV, JSON, Plaintext, Images, Videos, etc.
- ✗ Support for various types of sources and destinations – FTP, HTTP, SQL Databases, NoSQL Databases, Search Engines, Cache Server, etc.
- ✗ Scalable and Reliable for large volume and high-velocity data.
- ✗ You should also consider Data Cleansing and Data Validation logics.

Apache Nifi

NiFi est un logiciel libre de gestion de flux de données. Il permet de gérer et d'automatiser des flux de données entre plusieurs systèmes informatiques, à partir d'une interface web et dans un environnement distribué.

Par exemple, NiFi peut être très utile dans un cas d'usage comme **l'alimentation d'un DataLake Hadoop** à partir de plusieurs sources de données.

Basé sur le paradigme de programmation flow-based programming, NiFi fournit une interface web qui permet de **construire un flux de données en Drag and Drop**. Ainsi, il est possible de **définir**, de **contrôler en temps réel**, et d'une certaine manière, de **sécuriser** l'acheminement de données.

Apache Nifi

Apache NiFi **assure l'intégralité du flux de données**, il est tolérant aux pannes, est **scalable** et a été conçu pour gérer de gros volumes de données en temps réel.

Apache NiFi est compatible avec Kerberos qui **assure l'authentification**, avec Apache Ranger qui permet la **sécurité des autorisations d'accès** et avec Apache Knox qui gère la **sécurité au niveau authentification** et celle des appels REST and HTTP.

Flow-based programming

En programmation informatique, la programmation basée sur les flux (FBP) est un paradigme de programmation qui définit les applications comme des réseaux de processus "boîte noire", qui échangent des données à travers des connexions prédéfinies par passage de messages, où les connexions sont spécifiées en dehors des processus. Ces processus de boîte noire peuvent être reconnectés à l'infini pour former différentes applications sans avoir à être modifiés en interne. FBP est donc naturellement orienté vers les composants.

Le paradigme de NiFi repose sur celui du flow-based programming. Ce paradigme envisage une application comme étant une « usine de données », dans laquelle les données transitent à travers des connexions entre divers postes de traitement, à l'image d'un tapis roulant. Il est alors possible de modifier ces connexions sans modifier le fonctionnement interne de ces modules de traitement

Architecture d’Apache NiFi

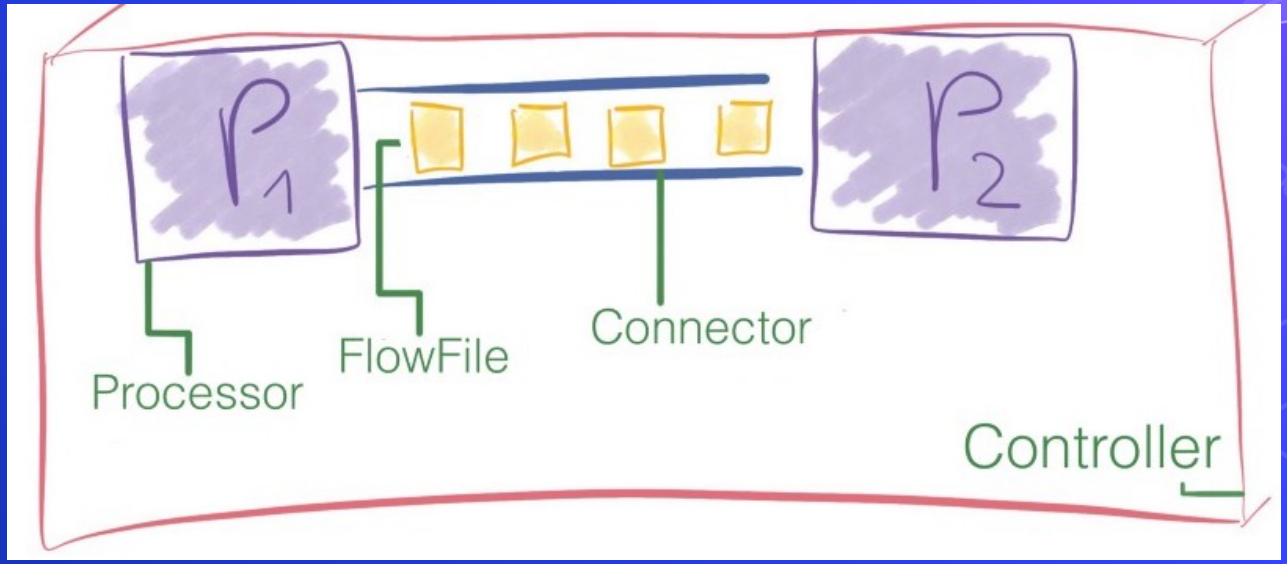
Un flux de données NiFi est constitué de trois éléments de base :

1) Les FlowFiles : ils représentent les données qui transitent dans NiFi. Ils sont composés d’une part de l’information utile à proprement parler (le contenu), et d’autre part d’un ensemble d’attributs clé/valeur qui font office de métadonnées ;

2) Les Processors : ce sont les modules qui opèrent sur les données en exploitant les métadonnées. Ils ont en général une entrée et une sortie, et orientent les Flowfiles vers d’autres Processors en fonction du résultat de leur traitement ;

3) Les connexions : elles relient deux Processors adjacents. Elles sont empruntées par les Flowfiles en fonction des traitements opérés par les Processors en amont. Chaque Flowfile constitue une queue dans laquelle résident les Flowfiles dans l’attente d’être traitées par les Processors en aval

Architecture d’Apache NiFi



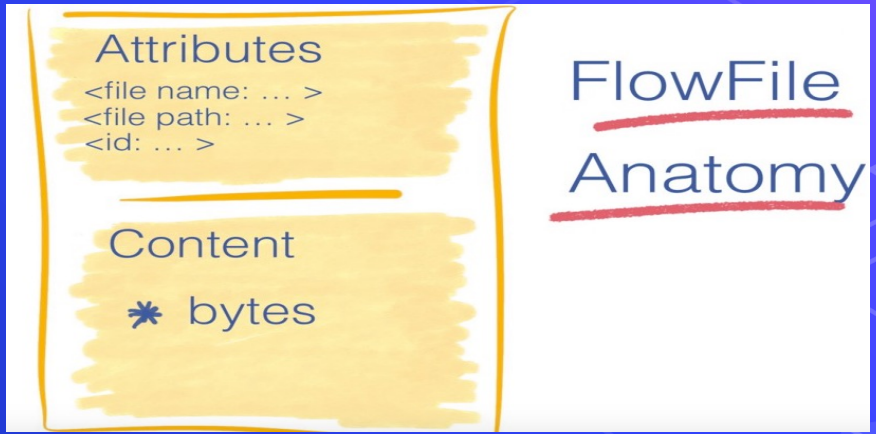
FlowFile Anatomy

Un FlowFile se compose de deux parties :

Les attributs, qui sont des paires clé/valeur. Par exemple, le nom du fichier, le chemin d'accès au fichier et un identifiant unique sont des attributs standard.

Contenu, une référence au flux d'octets compose le contenu du FlowFile.

Remarque: Le FlowFile ne contient pas les données elles-mêmes. Cela limiterait considérablement le débit du pipeline.

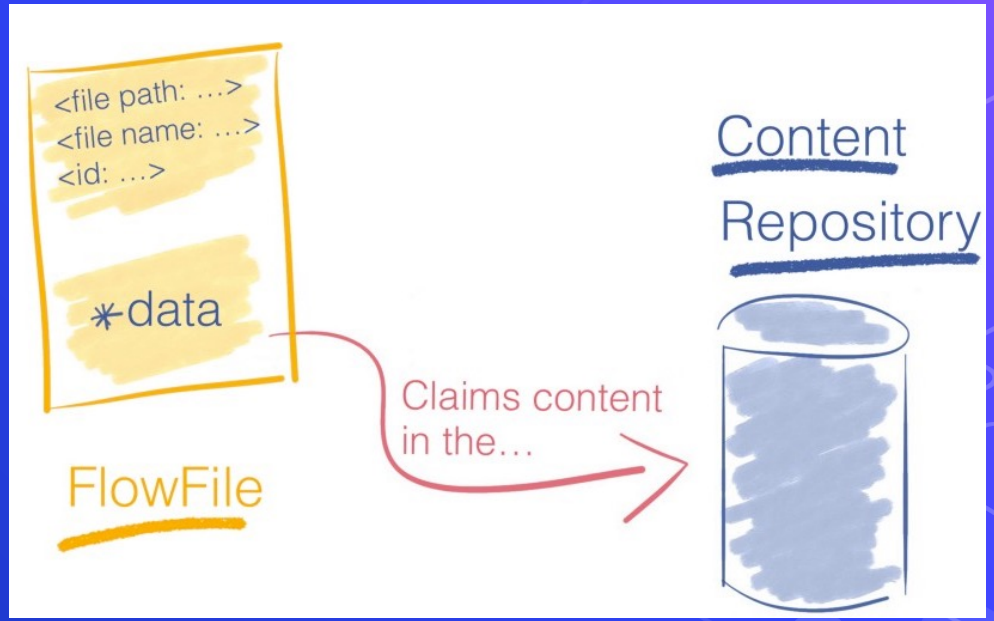


Anatomie d'un FlowFile - Il contient les attributs des données ainsi qu'une référence aux données associées

Content Repository

un FlowFile contient un pointeur qui renvoie à des données stockées à un certain endroit dans le stockage local. Cet endroit s'appelle le dépôt de contenu (Content Repository).

Pour accéder au contenu, le FlowFile réclame la ressource au Content Repository. Ce dernier conserve les traces du décalage exact du disque par rapport à l'endroit où se trouve le contenu et le renvoie au FlowFile.



Le Content Repository stocke le contenu du FlowFile

Content Repository

Tous les processeurs n'ont pas besoin d'accéder au contenu du FlowFile pour effectuer leurs opérations - par exemple, l'agrégation du contenu de deux FlowFiles ne nécessite pas de charger leur contenu en mémoire.Lorsqu'un processeur modifie le contenu d'un FlowFile, les données précédentes sont conservées.

Le NiFi copie en écriture, il modifie le contenu tout en le copiant vers un nouvel emplacement. Les informations originales sont laissées intactes dans le dépôt de contenu.

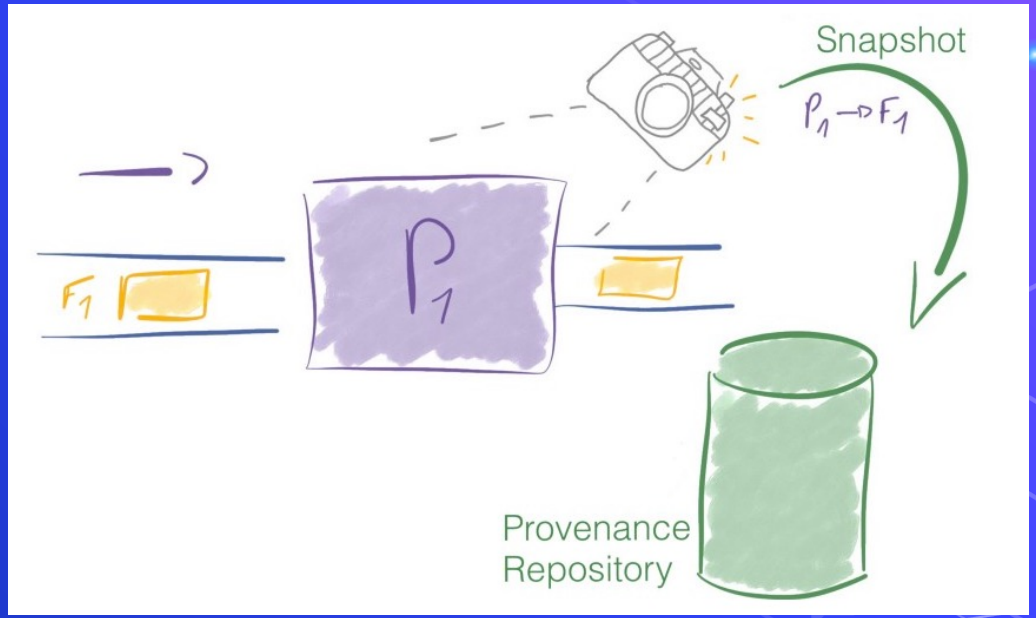
Exemple: Imaginez un processeur qui comprime le contenu d'un FlowFile. Le contenu original reste dans le référentiel de contenu, et une nouvelle entrée est créée pour le contenu compressé.Le Content Repository renvoie finalement la référence au contenu compressé. Le FlowFile est mis à jour pour pointer vers les données compressées.Le schéma ci-dessous résume l'exemple avec un processeur qui compresse le contenu des FlowFiles.

Provenance Repository

Chaque fois qu'un FlowFile est modifié, NiFi prend un instantané (snapshoot) du FlowFile et de son contexte à ce moment-là. Le nom de cet instantané dans NiFi est "Provenance Event". Le référentiel de provenance enregistre les événements de provenance.

Provenance nous permet de retracer la lignée des données et de construire la chaîne complète de conservation pour chaque élément d'information traité dans NiFi.

le dépôt de preuves* propose également de rejouer les données à partir de n'importe quel point dans le temps.

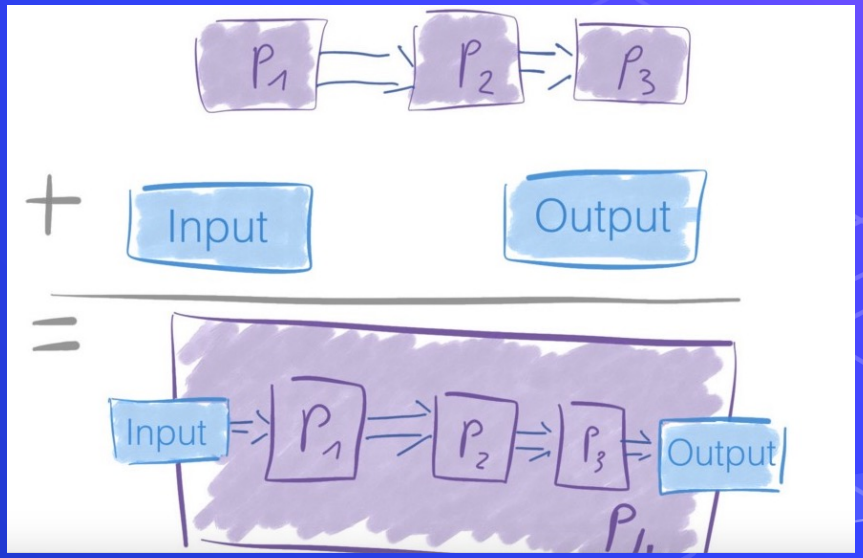


Le Provenance Repository stocke les métadonnées et les informations contextuelles de chaque FlowFile

Group processor

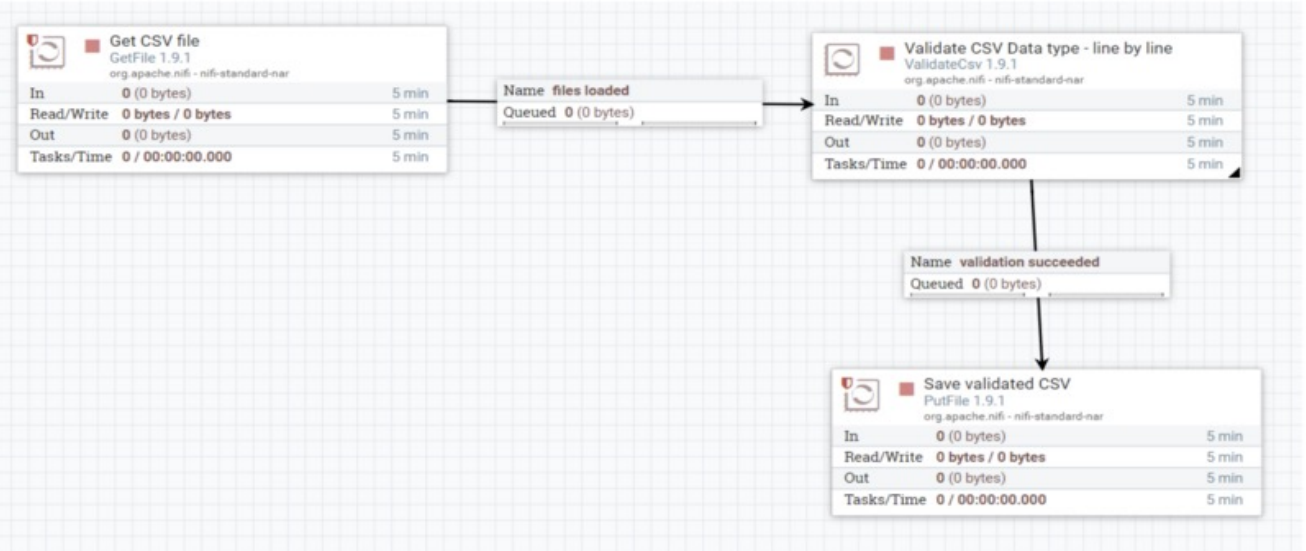
Un groupe de transformateurs (processors), avec leurs relations, peuvent former un nouveau processus. Vous ajoutez un port d'entrée et un port de sortie afin qu'il puisse recevoir et envoyer des données.

Les groupes de processeurs sont un moyen facile de créer de nouveaux processeurs à partir de ceux qui existent déjà.



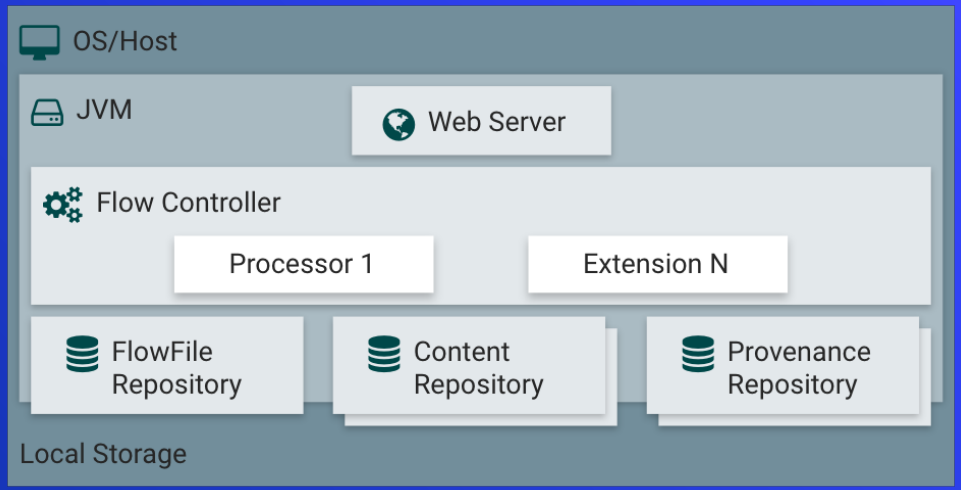
Construction d'un nouveau processeur à partir de trois processeurs existants

Exemple d’un Flux Nifi



Exemple d’un flux Nifi pour la validation d’un fichier CSV

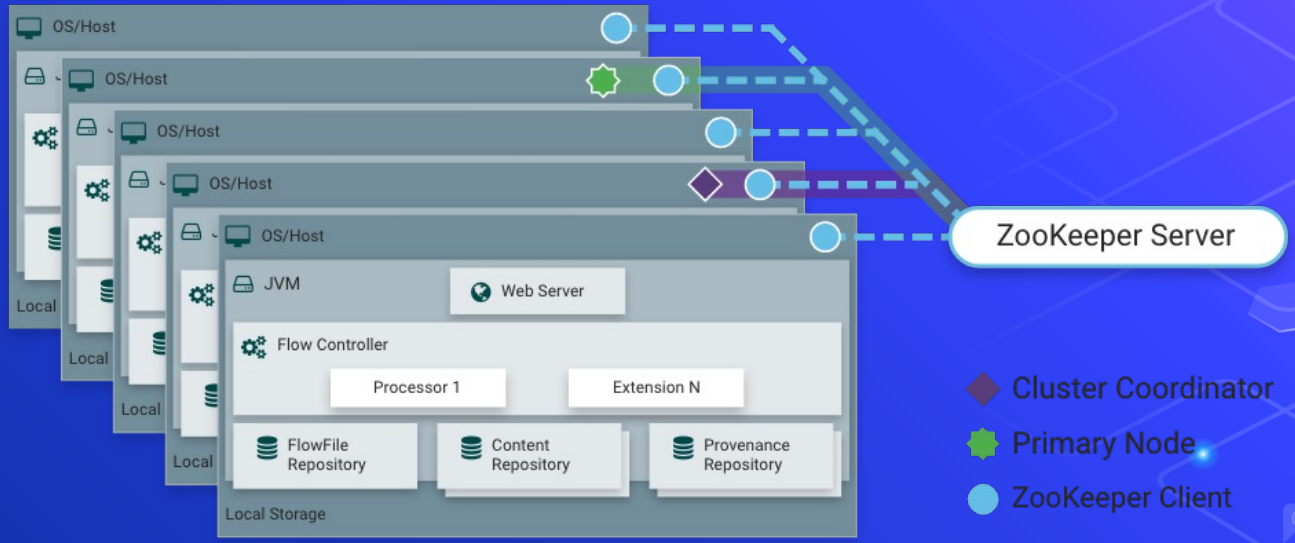
Architecture d’Apache NiFi (mode standalone)



Apache NiFi fonctionne au sein d’une JVM, installée sur un OS. Il est constitué des principaux composants : ***WebServer, Flow Controller***, ***Processor, Content Repository*** , ***Provenance Repository***

Architecture d’Apache NiFi (mode cluster)

Apache NiFi peut également être implémenté en mode cluster : les flux de données peuvent être distribués et traités à travers différents nœuds d’un cluster NiFi. Le schéma ci-dessous donne un aperçu d’une architecture NiFi distribuée :



Architecture d'Apache NiFi

Apache NiFi fonctionne au sein d'une JVM, installée sur un OS. Il est constitué des principaux composants suivants :

WebServer : qui permet d'héberger l'interface HTTP de NiFi.

Flow Controller**:** qui sert de broker facilitant les interactions entre processeurs à travers l'échange de *FlowFiles*.

Processor ***:*** qui permet d'exécuter des traitements.

FlowFile Repository : qui permet de stocker l'état d'un ou plusieurs *FlowFiles* actifs. L'implémentation par défaut d'un *FlowFile Repository* est basée sur une approche WAL (*Write-Ahead Log*).

Content Repository : qui permet de stocker le contenu (en octets) d'un ou plusieurs *FlowFiles*. L'implémentation par défaut d'un *Content Repository* est basée sur une approche standard : stockage de chaque bloc de données au sein du système de fichiers.

Provenance Repository : qui permet de stocker les événements associés à la provenance des données. Lesdits événements stockés sont indexés.

Fonctionnalités clés

QoS (*Quality of Service*) : contrôle et qualification des données avec possibilité de gérer la tolérance aux pertes (*loss tolerance*), la fiabilité des messages (*guaranteed delivery*),…
Security : support de 2 types de sécurité :

D'utilisateur à système : authenfication SSL bidirectionnelle → si un utilisateur rentre des informations sensibles – comme son mot-de-passe – au sein d'un flux, celles-ci seront encryptées côté serveur de sorte qu'elles ne soient jamais exposées côté client.

De système à système : sécurisation des flux de données via cryptage SSL, avec possibilité, pour un flux, de décrypter/encrypter un contenu via des clés partagées ou tout autre mécanisme de cryptographie.

Data Recovery, ***Data Recording*** : le *Content Repository* permet aux données d'être automatiquement enregistrées, mais aussi supprimées suivant leur ancienneté et fréquence d'utilisation, surtout en cas d'espace nécessaire. NiFi permet également de récupérer des données en rejouant des flux via le mécanisme WAL de *FlowFile Repository*.

Fonctionnalités clés

Data Buffering : mise en cache des données en mouvement ou non. NiFi offre également la possibilité de gérer dynamiquement les pressions de mémoire (principe de *back pressure*).
Data Provenance :

Collecte de données à partir de chaque point des flux.

Traitement des données (indexation, *merging*, *splitting*,…).

Visualisation du contenu et des attributs.

Queue Prioritization : configuration d'un niveau de priorité par connexion (en fonction du temps, de l'ordre d'arrivée des données,…).
Clustering : voir sous-section précédente.

Conclusion

L’avantage le plus probant de NiFi par rapport à d’autres outils de gestion de données est sans aucun doute son interface utilisateur, qui permet de créer des flux de données de manière intuitive par glisser-déposer sans avoir besoin d’écrire de code. Cela permet en outre d’amener les experts métiers au plus proche du traitement de la donnée. En ce sens, il possède des similitudes avec certains ETL.

NiFi se caractérise aussi par sa polyvalence : il possède de d’innombrables connecteurs et est indépendant des sources et des systèmes de traitement auxquels il est relié. Il est aussi très extensible et il est facile d’y incorporer de nouveaux connecteurs après les avoir développés.

Installation

1) **Téléchargez** le binaire de nifi à partir de https://nifi.apache.org/download.html

2) Lancez ensuite le fichier run-nifi.bat qui se trouve dans le répertoire nifi-1.4.0-binnifi-1.4.0bin si vous êtes sur Windows, ou bien binnifi.sh run pour les utilisateurs linux/mac.

3) sur un votre navigateur Web, saisissez l'adresse : **localhost:8080**