

Data Ingestion with Apache NiFi¹ ²

Hong-Tri Nguyen
Department of Computer Science
hong-tri.nguyen@aalto.fi

CS-E4640 Big Data Platforms, Spring 2026
January 30, 2026

¹<https://nifi.apache.org/components/>

²<https://github.com/rdsea/bigdataplatforms/tree/master/tutorials/nifi>

Outline

NiFi

Basic cases

Architecture

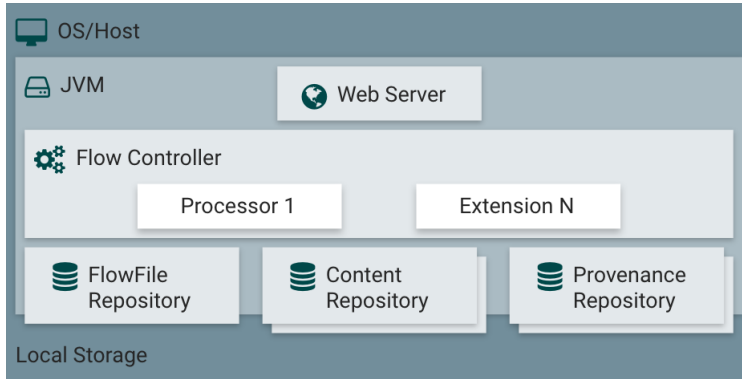


Figure: NiFi architecture

How NiFi works?

A NiFi dataflow consists of atomic elements which can be in a group or individual processor connecting each others

- ▶ A Processor has its own task like ingest data, transform data, or load data
- ▶ Data in NiFi:
 - ▶ The FlowFile Repository contains metadata for all the current FlowFiles in the flow.
 - ▶ The Content Repository holds the content for current and past FlowFiles.
 - ▶ The Provenance Repository holds the history of FlowFiles.
- ▶ A FlowFile is an abstract layer of actual any data, including attribute and content
- ▶ A Connection is a Queue for FlowFiles to link processors to formalize a dataflow
- ▶ A Controller Service is a shared service used by a processor, like a DB connection or credential service

Details of components

Life cycle of FlowFile

- ▶ FlowFiles are persisted at the current system and communication is passed-by-reference
- ▶ A new FlowFile is created since there are updates or changes from the content after ingesting data from src

Types of processors

- ▶ Data ingestion processors
- ▶ Data transformation processors
- ▶ Data egress/sending data processors
- ▶ Database access processors
- ▶ Routing and mediation processors
- ▶ Attribute extraction processors

Ready for Dirty Hand?

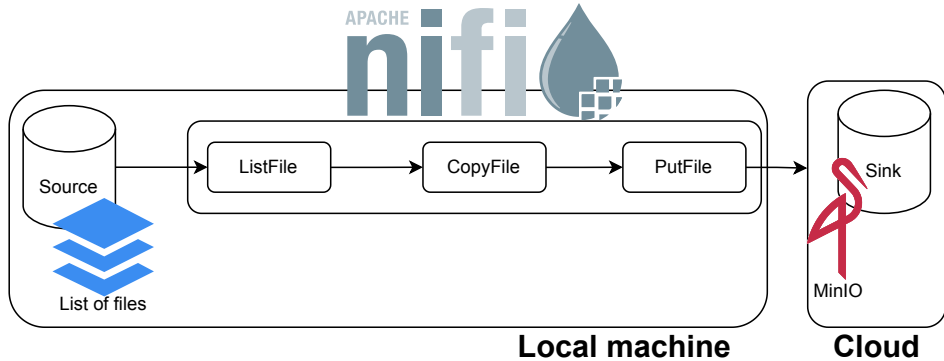
Basic cases

1. Define a flow for ingesting data into storage bucket
2. Define a flow for ingesting data via AMQP
3. Capture changes in legacy databases, filter the records, and do ingestion to a message queue

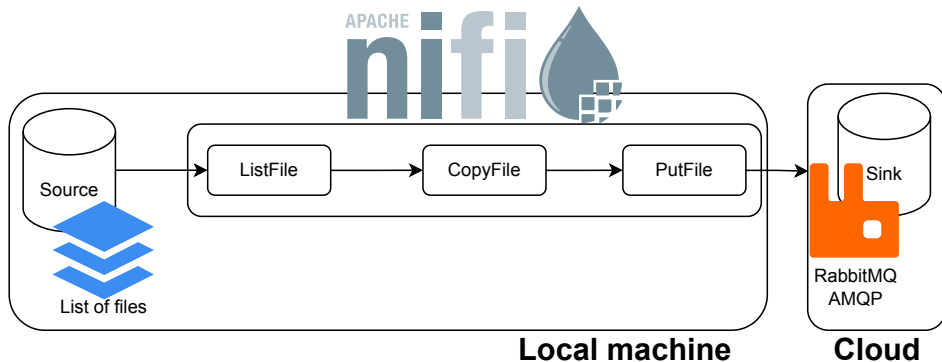
Pre-Requisites

- ▶ Pre-Requisites
 - ▶ A storage bucket
 - ▶ A storage message queue
 - ▶ A relational database
 - ▶ Apache NiFi installed and running
- ▶ NiFi canvas:
 - ▶ ListFile lists files in a specified directory
 - ▶ FetchFile fetches those files
 - ▶ PutS3Object/PutGCSObject stores the files into a storage bucket
 - ▶ PublishAMQP stores the data in a queue
 - ▶ CaptureChangeMySQL captures the changes from mySQL

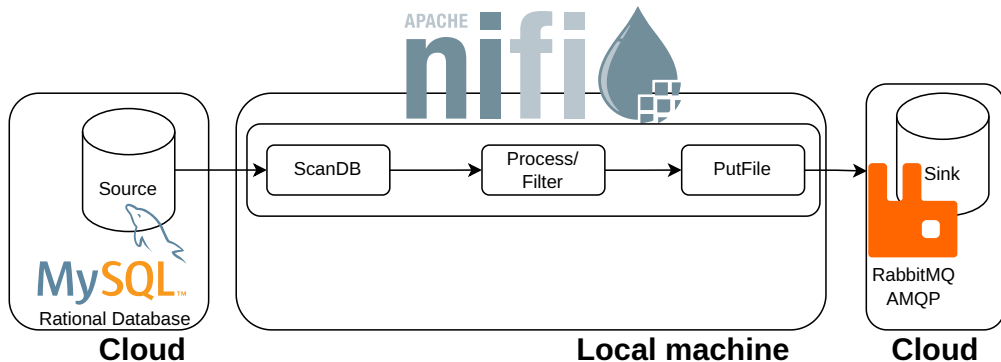
1. Ingesting data into bucket



2. Ingesting data to AMQP



3. Capturing changes from legacy database



Time now is for you