Jobintech

صندوق الإيداع والتدبير
CAISSE DE DÉPÔT ET DE GESTION

Statement of Work

Industrialization and Automation of NiFi Pipelines (CI/CD)

Date : November 5, 2025
Version : 1.0

Implementation Team

Ihssane BOUTAHAR
Imane ELFETOCHI
Saad KHALMADANI
Bakr KAMAL
Rayan BIAD

Client

[DXC Technology Maroc]

November 5, 2025

1. General Information

Project	Project 2: Industrialization and Automation of NiFi Pipelines
Validation Date	[11/24/25]
Document Version	1.0
Author	[Team Project 2]

Contents

1 Context and Objectives

1.1 Current Problem

Data pipelines (NiFi flows) are currently deployed manually across development, staging, and production environments. This methodology is time-consuming, a major source of human error (omissions, wrong configurations, environment drift), and hinders the velocity of data teams.

1.2 Project Objectives

The main goal is to design, implement, and document a **Continuous Integration and Continuous Delivery (CI/CD) pipeline** to automate the entire lifecycle management of NiFi flows.

The expected benefits are:

- **Reliability:** Drastically reduce human errors during deployments.
- **Efficiency:** Achieve significant time savings for development and operations teams (DevOps/DataOps).
- **Standardization:** Ensure environments (Dev, Staging, Prod) are consistent and deployments are reproducible.
- **Traceability:** Establish clear auditing and centralized versioning for all deployments.

2 Project Scope

2.1 In-Scope Features

The project includes the following elements:

- Integration of **GitHub** as the sole Source Control Management (SCM) for all NiFi flows.
- Integration of **NiFi Registry** as the central "Artifact Repository" for flow versioning.
- Creation of the CI/CD pipeline (e.g., via GitHub Actions, Jenkins, GitLab CI, etc.) to orchestrate the process.
- Automation of deployment and configuration (variable management) across the 3 environments (Dev, Staging, Prod).
- Integration of Quality Gates (automated checks) within the pipeline.

2.2 Out-of-Scope Items

The following elements are **not** part of this project:

- Creation or modification of the business logic of existing NiFi flows.
- Installation, initial configuration, or ongoing maintenance of the NiFi and NiFi Registry servers (these are considered prerequisites).

- Migration or correction of existing NiFi flows to ensure CI/CD compatibility (unless a "pilot" flow is defined).

3 Functional Requirements (FR)

The CI/CD system must enable the following actions:

3.1 Source Management and Versioning

FR-01: The source code of *all* NiFi flows must be stored and versioned in a GitHub repository.

FR-02: The pipeline must integrate with **NiFi Registry** to store and version "flow bundles" ready for deployment.

FR-03: A branching strategy (e.g., GitFlow, Trunk-Based) must be defined and adhered to for managing developments, fixes, and releases.

3.2 Continuous Integration (CI) Process

FR-04: A change pushed to a development branch (e.g., `develop` or `main`) must automatically trigger the CI pipeline.

FR-05 (Quality Control): The CI pipeline must integrate automated quality checks (e.g., flow syntax validation, unit tests if applicable, verification of best practices).

FR-06: If quality checks pass, the pipeline must "build" the flow and push it as a new version to NiFi Registry (in the appropriate bucket, e.g., "Development").

3.3 Continuous Deployment (CD) Process

FR-07 (Dev Deployment): The pipeline must automatically deploy the new flow version (from NiFi Registry) to the **Development** NiFi environment.

FR-08 (Staging Promotion): Promotion to the **Staging** environment must be a controlled step (either manually triggered or gated by a Pull Request validation).

FR-09 (Production Promotion): Deployment to **Production** must strictly require manual validation (e.g., "Click-to-deploy" after approval) for security reasons.

FR-10 (Configuration Management): The pipeline must securely manage environment-specific configurations (e.g., passwords, endpoints) without storing them in clear text on GitHub.

4 Technical Constraints and Requirements (TR)

TR-01 (Mandatory Stack): The solution *must* utilize the existing technology stack consisting of **NiFi**, **NiFi Registry**, and **GitHub**.

TR-02 (CI/CD Tool): GitHub Actions

TR-03 (Secrets Security): Sensitive information (API keys, passwords) must be stored securely (e.g., GitHub Secrets, Vault) and injected at the time of deployment.

TR-04 (Reproducibility): The chain must guarantee a 100% reproducible deployment.

TR-05 (Traceability): The pipeline must provide clear logs for every deployment (success or failure), enabling easy auditing and troubleshooting.

5 Expected Deliverables

D-01: Operational CI/CD Pipeline: All scripts, configurations (e.g., YAML files), and tools configured to execute the full lifecycle.

D-02: Deployment Documentation (User): Clear documentation (e.g., `README.md`) for NiFi flow developers, explaining the development process and deployment triggers.

D-03: Technical Documentation (Admin/Maintenance): Documentation describing the CI/CD chain architecture, configuration, and maintenance procedures.

D-04 (Optional): A demonstration/training session for the concerned teams.

6 Project Framework

6.1 Schedule and Deadlines

- Desired Start Date: [11/05/25]
- Mandatory Delivery Date (Go-Live): [11/28/25]

6.2 Validation Criteria (Acceptance)

The project will be considered successful when:

1. A "pilot" NiFi flow (simple) can be deployed from Dev to Prod using the CI/CD pipeline *exclusively*.
2. A change to the pilot flow is deployed using the same process.
3. A deliberately erroneous flow is *blocked* by the quality control check (FR-05).
4. Deliverables (D-01, D-02, D-03) are submitted and approved.