Apache NiFi Assignment: Real-Time Data Ingestion & Processing

Student:

Nader Saeed

1. المقدمة

يهدف هذا المشروع العملي الى بناء خط تدفق بيانات (Data Pipeline) متكامل، يقوم بسحب البيانات من مصادر مختلفة (قاعدة بياناتPostgreSQL وملفات Json)، ثم معالجتها وتوحيدها، وتخزنها في النهاية داخل نظام ملفات Hadoop الموزع(HDFS). تم تنفيذ هذا المشروع بالاعتماد على بيئة حاويات (Docker) لضمان استقرار وسهولة ربط الخدمات المختلفة.

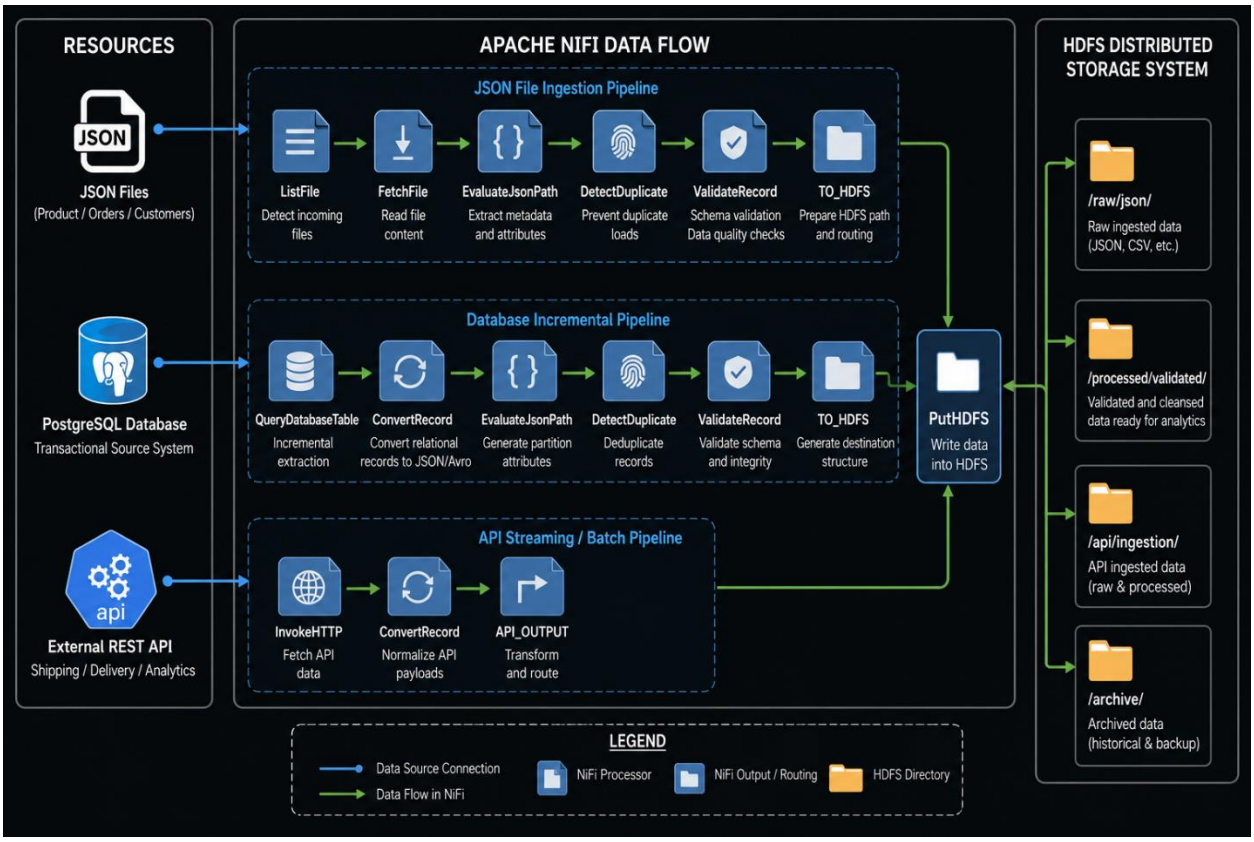
2. المكونات التقنية وبيئة العمل.

- ❖ Apache NiFi أداة إدارة تدفق البيانات
- ❖ مصادر البيانات قاعدة بيانات متصلة (PostgreSQL) ومصادر ملفات Jsonمن python بيانات من API
- ❖ Apache Hadoop بيئة التخزين النهائية (HDFS) صمن حاوية itvdelab-1
- ❖ Docker container بيئة التشغيل

3. Architecture Diagram.

تم تصميم هذا النظام على ان يكون Best Practices تم فصل العمليات لكل المصادر في قروبات منفصلة وايضاً تم عمل عمليات المعالجة لكل مصدر منفصلة وذألك بترتب علية عدة أسباب منطقية

- ❖ لمعرفة تتبع تدفق بيانات كل نظام بحيث إذا جدث خلل من أحد المصادر نعرف الخلل من أي مصدر ونقوم بإصلاحه بأقرب وقت.
- ❖ ايضاً إذا تم جمع جميع الموارد في عمليات معالجه مشتركه (هذا قد يوفر الموارد اكيد ولم يتم هدارها) ولاكن نحن لم نقم بجمعها وذألك من منطق انه إذا تعطل مصدر واحد قد يودي الى إيقاف خط الانابيب بشكل كامل ولانت بطريقة التقسم التي لدي سيتوقف معالجات المصدر المتعطل فقط بينما تستمر المصادر الأخرى بتدفق.
- ❖ ايضاً نحن لا نضمن او نعلم ان البيانات التي تأتي من جميع المصادر هي نفس البيانات حتى تكون في نفس المعالجات ولا نضمن ايضاً ان تستمر البيانات نفسها ولا تتغير ر في أجد المصادر.



4. مرحلة التحليل والتنظيف (Data Transformation & Validation)

لضمان جودة البيانات قبل تخزينها، تم تطبيق سلسلة من عمليات الفلترة:

- ❖ مرحلة التحقق والفلترة (Data Validation & Cleaning):

تعتبر هذه المرحلة الأهم لضمان عدم دخول "بيانات تالفة" إلى مستودع البيانات، وتضمنت العمليات التالية:

استخراج السِمات (EvaluateJsonPath): تم استخراج حقل المعرف id من داخل ملفات الـ JSON وتحويله إلى سمة خاصة بكل ملف (Attribute): عبر المسار $.id.

عملية الـ Validation: تم تمرير البيانات عبر معالجات التحقق للتأكد من وجود المعرفات الأساسية وصحة هيكلية الملف.

إزالة التكرار (DetectDuplicate): تم إعداد "مخزن مؤقت" لمقارنة المعرفات المستخرجة، بحيث يتم استبعاد أي سجل مكرر وصل من الـ API أو القاعدة، مما يضمن فرادة السجلات في HDFS.

- مرحلة التحميل إلى HDFS (Load to Hadoop)

تم تنفيذ عملية الـ Load باستخدام معالج PutHDFS وفق الإعدادات التقنية التالية:

Config Files: الربط المباشر مع ملفات النظام core-site.xml و hdfs-site.xml المستخرجة من مسار opt/hadoop-3.3.0/etc/hadoop/./

Target Directory: المسار /user/nifi/data داخل بيئة هادوب.

Permissions: التأكد من منح صلاحيات الكتابة (777) للمجلد لضمان نجاح عملية الرفع دون تعارضات.

- نتائج الاختبار والتحقق ((Results & Verification

تم التأكد من سلامة النظام عبر:

فحص الطوابير: التأكد من تصفير كافة الـ Queues بعد التشغيل، مما يدل على انتقال البيانات بنجاح.

الأوامر التنفيذية: استخدام الأمر hdfs dfs -ls /user/nifi/data للتأكد من وجود الملف الموحد وحجمه.

جودة البيانات: فتح الملفات المخزنة في HDFS والتأكد من أنها تحتوي على سجلات نظيفة وغير مكررة وخالية من الأخطاء الهيكلية.