Cahier des charges

Phase 2 : Traitement distribué

Mastère 2 Data Engineer

Promotion 2025-2026

Lyon Ynov Campus

Objectif global

Étendre le pipeline existant (PostgreSQL alimenté par Kafka/KRaft en Phase 1) pour :

- extraire les données depuis PostgreSQL,
- traiter à grande échelle via Hadoop MapReduce (Java) ou Apache Spark (Java),
- stocker les résultats dans HBase ou Hive,
- automatiser l'infrastructure Big Data (Hadoop/HDFS, et si besoin YARN) avec Ansible.

Périmètre fonctionnel

Domaine	Outil / Composant	Description
Extraction	PostgreSQL	Lecture des données issues de la Phase 1 via JDBC
Traitement distribué	Hadoop (MapReduce Java) ou Spark (Java)	Application de transformation ou d'agrégation parallèle
Stockage distribué	HBase ou Hive	Persistance des résultats du traitement
Automatisation	Ansible	Déploiement complet de Hadoop et des services associés
Documentation	README + diagramme d'architecture	Description du pipeline, du traitement et des choix techniques

Périmètre technique

Architecture cible

[PostgreSQL (Phase 1)] -> [Job Java – Hadoop / Spark] -> [HBase] ou [Hive]

Spécifications minimales

Langage : Java 11 ou supérieur.

Build : Maven ou Gradle (tests unitaires inclus).

PostgreSQL : réutiliser vos tables.

Traitement : MapReduce ou Spark (lecture JDBC → HDFS → résultats).

Stockage final : HBase (clé + familles de colonnes) ou Hive (table Parquet, partitionnée si besoin).

Hadoop : cluster pseudo-distribué ou mini-cluster via Ansible.

YARN : recommandé pour MapReduce ; optionnel pour Spark local.

Ansible : déploiement idempotent des services (HDFS, YARN, HBase/Hive).

Livrables attendus

Code & Infrastructure

Répertoire processing-java/ contenant :

- Code Java (MapReduce ou Spark) + tests.
- pom.xml / build.gradle.
- Fichier de configuration (application.conf ou .properties).

Répertoire ansible/ avec :

- Inventaire (hosts.ini), variables (group_vars/), rôles et playbooks.
- Templates : core-site.xml.j2, hdfs-site.xml.j2, yarn-site.xml.j2, hive-site.xml.j2, hbase-site.xml.j2.

Scripts d'exécution

Documentation

- README.md clair avec :
  - choix du moteur (Hadoop ou Spark) ;
  - description du traitement ;
  - commandes d'exécution et de vérification ;
  - schéma d'architecture (draw.io, Mermaid…).
- Runbook : procédures de déploiement, d'arrêt, et de relance du cluster.

Résultat observable

Le job Java lit PostgreSQL, traite les données et les écrit dans HBase / Hive.

Résultats vérifiables :

- Hive : SELECT * FROM <table> LIMIT 10;
- HBase : scan '<table>' LIMIT 10