**ID EXPERT NIFI**



# EXPERTISES TECHNIQUES & FONCTIONELLES

 **Gestion des flux de données avec NIFI : Conception, déploiement et optimisation des flux de données (dataflows) en utilisant Apache NIFI** pour collecter, transformer, sécuriser et intégrer des données en temps réel.

 **Création de processeurs personnalisés : Développement de processeurs NIFI (NAR)** pour répondre à des besoins spécifiques, tels que l'encryptage de données, la gestion de la qualité des données, et l’automatisation des processus de collecte et de traitement.

 **Optimisation des performances de NIFI : Tuning des processors NIFI pour maximiser les performances et minimiser la latence des flux de données**. Expertise dans la gestion des ressources et l’optimisation des partitions de données, en particulier avec Kafka.

 **Gestion des erreurs et surveillance : Mise en place de systèmes de monitoring et de gestion des erreurs dans NIFI,** avec des outils comme Grafana, Prometheus et Jolokia pour assurer la fiabilité et la disponibilité des flux de données.

 **Sécurisation des flux de données : Utilisation de NIFI pour sécuriser les données en transit** (ex. : cryptage via des processeurs dédiés), ainsi que l’intégration avec des outils de sécurité comme Keytool pour garantir la confidentialité et l’intégrité des données.

## Compétences complémentaires :

 **Kafka : Intégration de NIFI avec Kafka** pour la gestion des flux de données en temps réel, la surveillance des consommateurs Kafka (Kafka Lag), et l’optimisation des partitions.

 **Big Data & Cloud : Traitement de données massives via Hadoop/HDFS**, Spark/Scala, et intégration avec des environnements Cloud comme Amazon S3 et Azure.

 **Bases de données & Visualisation : Intégration de NIFI avec des bases de données NoSQL (MongoDB, Neo4j**), et utilisation d'outils de visualisation comme Grafana et Kibana pour suivre les performances et les alertes des flux de données.

## CI/CD & DevOps : Mise en place de pipelines CI/CD pour le déploiement de NIFI

avec Jenkins, Maven et SonarQube.

## Outils et technologies maîtrisés :

 **Big Data & Data Engineering :** NIFI, Kafka, HDFS, Spark, Scala, Hive, MongoDB, Neo4j

 **Sécurisation & Monitoring :** Keytool, Jolokia, Elasticsearch, Kibana, Grafana, Prometheus

 **CI/CD & DevOps :** Jenkins, Maven, SonarQube, Git, Nexus

# EXPÉRIENCES PROFESSIONNELLES

|  |  |
| --- | --- |
| **EDF** | 2023 – ce jour |
| EXPERT NIFI |  |

**Projet :** Projet TEGG. L'objectif principal de ce projet est de récupérer, traiter et analyser les données issues des capteurs piézomètres. En particulier, il s'agit d'intégrer un ensemble de règles métiers pour garantir la qualité des données et produire des rapports détaillés d'analyse, le tout en utilisant une expertise approfondie de l'outil **NiFi** pour l'orchestration et l'automatisation des flux de données.

## Missions :

 **Récupérer les données depuis Amazon S3 :** Mise en place de processus d'intégration et de **récupération des données stockées sur S3, en utilisant NiFi** pour orchestrer le transfert de manière fluide et efficace.

 **Contrôler la qualité des données :** Application de règles de validation et de **nettoyage des données via NiFi** pour s'assurer de la conformité et de la fiabilité des informations avant tout traitement ultérieur.

 **Appliquer des règles métiers :** Intégration des règles métiers spécifiques aux données piézométriques pour filtrer et **enrichir les données dans NiFi** avant leur traitement dans les systèmes en aval.

 **Déposer les données dans Neo4j** : Une fois validées et enrichies, les données seront déposées dans la base de données **Neo4j, en utilisant des processus automatisés dans NiFi pour la gestion du flux et l'exportation des données structurées**.

 **Reporting et analyse :** Générer des rapports analytiques basés sur les données traitées, en exploitant les capacités d'analyse distribuée avec Spark/Scala.

**Environnement technique : Nifi**, Java, Neo4j, Spark/Scala

|  |  |
| --- | --- |
| **SOCIETE GENERALE** | 2021 – 2023 |
| EXPERT NIFI |  |

**Projet 1 :** L'objectif principal de ce projet est de surveiller en temps réel les retards des consommateurs **NiFi** dans **Apache Kafka**. Cette surveillance sera réalisée à travers des tableaux de bord et des systèmes d'alerte, afin de prévenir toute perte de données et garantir une gestion fluide des flux.

## Missions :



****

****

**Installer Jolokia et Kafka Lag Exporter** : Mise en place des outils de surveillance nécessaires pour récupérer les métriques de retard des consommateurs **Kafka** et exposer ces informations.

**Optimiser les processeurs NiFi et les partitions Kafka** : Réalisation du tuning des processeurs **NiFi** et des partitions **Kafka** afin d'améliorer la gestion des flux et minimiser les latences.

**Récupérer les métriques de NiFi et Kafka** : Intégration et collecte des métriques via les outils adaptés pour permettre un suivi optimal du système.

**Environnement technique :** Nifi, Kafka, Jolokia, Kafka Lag Exporter, Grafana, Druid, Telegarfe, Prometheus

**Projet 2 :** L'objectif de ce projet est de développer un **custom processor NiFi** permettant d'encrypter les données provenant de **MongoDB**, afin de garantir la sécurité des informations sensibles lors de leur transfert ou stockage.

## Missions :

 **Développer un NAR NiFi** : Création d'un **NiFi Archive (NAR)** personnalisé pour implémenter le traitement d'encryption des données MongoDB.

 **Mettre en place une chaîne CI/CD** : Mise en œuvre de pipeline d'intégration et de déploiement continu (CI/CD) pour automatiser la construction, les tests et le déploiement du processeur.

**Environnement technique : Nifi, Java, Maven**, SonarQub, Jenkis, Git, MongoDB

**Projet 3 :** Projet DHR-Usage-Layer. L’objectif de ce projet est de rendre exploitable la BV

(Business View) technique de la DRH

## Missions :

 Automatiser la création ou la suppression des tables hive

 Mettre à jour les partitions des tables hive selon une source de données

 Collecter les données de Hive et les inscrire dans une base de données PostgreSQL  Indexer les logs applicatifs dans Elascticsearch

**Environnement technique :** Spark/Scala, HUE/hive, **kafka, Elastcisearch/kibana**

**Projet 4 :** L'objectif principal de ce projet est d'étudier les impacts de l'**upgrade HDF** (Hortonworks Data Platform) sur les différents composants du système, avec un focus particulier sur les flux gérés par **NiFi**. Il s'agit de comprendre les effets de l'upgrade sur l'orchestration des flux de données **NiFi**, d'évaluer le comportement de **Kafka** et d'optimiser l'ensemble des processus après la mise à jour.

## Missions :

 **Étudier l'impact de l'upgrade sur les flux NiFi** : Une analyse détaillée sera effectuée pour identifier l'impact de l'upgrade HDF sur les **flux NiFi**, en particulier sur la performance, la compatibilité des processeurs, et l'intégrité des données traitées. Il sera nécessaire de tester les flux existants et d'adapter les configurations de **NiFi** en conséquence pour maintenir la stabilité.

 **Étudier le comportement de Kafka après l'upgrade** : Bien que l'accent soit mis sur **NiFi**, une étude sera menée sur les performances de **Kafka** après l'upgrade, afin de garantir une bonne interaction entre **NiFi** et **Kafka**, et d'éviter toute dégradation des flux de données.

 **Optimiser les flux de données après l'upgrade** : Après avoir analysé les impacts sur **NiFi** et **Kafka**, des ajustements seront réalisés pour optimiser les flux de données. Cela inclut la mise à jour des **processeurs NiFi**, l'amélioration de la gestion des ressources, et la réduction de la latence dans les pipelines de données.

**Environnement technique : Python, NIFI, Kafka**, Ranger, ElasticSearch/Kibana, HDFS, Fluentd, Telagraf

|  |  |
| --- | --- |
| **SNCF** | 2018 – 2020 |
| EXPERT NIFI |  |

**Projet 1 :** Dans le cadre de la migration vers Azure de la plateforme Big Data SNCF, ce projet vise à mettre en place un système d’ingestion de données performant et sécurisé en utilisant HDF (Hortonworks Data Platform), **en particulier à travers NiFi** pour l’orchestration des flux de données.

## Missions :

 **Mettre en place un système d’ingestion avec NiFi** : Création et configuration de pipelines d’ingestion de données via **NiFi**, permettant de collecter, transformer et transférer efficacement les données vers les systèmes cibles dans la plateforme Big Data.

 **Sécuriser la plateforme d’ingestion avec Keytool** : Mise en place de la sécurité sur les flux de données via **Keytool** pour générer et gérer les certificats

nécessaires, assurant ainsi une communication sécurisée entre les composants

du système d’ingestion.

 **Mettre en place un système d’alerte et de monitoring avec Grafana et InfluxDB** : Configuration de **Grafana** pour la visualisation des métriques en temps réel, et intégration avec **InfluxDB** pour stocker et interroger les données de performance, ainsi que la mise en place d’un système d’alerte afin de détecter rapidement toute anomalie dans les flux de données.

**Environnement technique : HDF (Nifi, Kafka**, zookeeper, grafana, ambari), influxDB, keytoo

**Projet 2 :** L’objectif de ce projet est d’estimer les coûts de péage des sillons ferroviaires en fonction des différentes redevances et de suivre le processus de facturation des sillons. Pour cela, les données doivent être collectées, stockées et traitées efficacement, en utilisant des outils adaptés pour gérer les volumes et la complexité des informations liées à l’infrastructure des trains.

## Missions :

 **Récolter les données avec NiFi :** Mise en place de **flux de données NiFi** pour collecter les informations nécessaires à l’estimation des coûts de péage des sillons à partir des différentes sources (base de données, API, fichiers, etc.).

 **Stocker les données dans HDFS :** Une fois collectées, les données seront stockées dans HDFS pour assurer une gestion fiable et scalable des données brutes avant le traitement.

 **Développer une application Spark/Scala pour nettoyer et agréger les données :** Développement d’une application Spark/Scala pour nettoyer, agréger et traiter les données collectées, en appliquant les règles de calcul des coûts de péage et en générant les informations nécessaires à la facturation des sillons.

**Environnement technique : Nifi, HDFS/ HADOOP**, Spark, Scala, Maven, Github Jenkis, Nexus

|  |  |
| --- | --- |
| **RENAULT** | 2016 - 2017 |
| EXPERT NIFI |  |

**Projet 1 :** L’objectif de ce projet est d'optimiser le temps passé par les garagistes autour des véhicules lors des contrôles en automatisant et en améliorant les processus de collecte, de stockage et de visualisation des données liées aux inspections.

## Missions :

 **Récolte des données avec NiFi :** Mise en place de **flux de données avec NiFi** pour collecter les informations nécessaires lors des inspections des véhicules, telles que les diagnostics, les rapports de maintenance, etc.

 **Stockage des données chaudes dans Elasticsearch** : Stockage des données en temps réel (données chaudes) dans Elasticsearch pour un accès rapide et une recherche performante.

 **Stockage des données froides dans HDFS/Hadoop :** Archivage des données moins réactives ou historiques (données froides) dans HDFS pour une gestion efficace à long terme.

 **Utilisation de Kibana pour la visualisation :** Mise en place de tableaux de bord dans Kibana pour permettre aux garagistes d'accéder facilement aux données analysées et visualisées, afin d'accélérer les contrôles et diagnostics.

## Environnement technique : NIFI, ElasticSearch, HDFS/ HDOOP

**Projet 2 :** Ce projet vise à quantifier et prédire les pannes des robots dans une chaîne de production. En analysant les données relatives aux arrêts des robots, l'objectif est de réduire les interruptions et optimiser la production en identifiant les séquences de pannes et en prédisant leur occurrence à l'aide de modèles statistiques et d'outils d'analyse avancés.

## Missions :

 **Récolte des données avec NiFi** : Mise en place de flux de données via **NiFi** pour collecter toutes les informations pertinentes sur les robots (temps de fonctionnement, pannes, alertes, etc.) en temps réel depuis les différents capteurs et systèmes de monitoring.

 **Stockage des données dans HDFS/Hadoop** : Stockage des données collectées dans **HDFS** pour une gestion fiable et scalable des données brutes issues des robots, permettant leur analyse en profondeur.

 **Établir un modèle statistique pour prédire les pannes avec Machine Learning** : Utilisation de **Zeppelin**, **Spark MLlib** et **Scala** pour construire un modèle statistique et de machine learning qui analyse les séquences de pannes des robots et prédit les pannes futures en se basant sur des patterns historiques.

 **Quantifier les arrêts sur panne par tranche de durée avec Hive** : Analyse des pannes en fonction de leur durée et de leur fréquence avec **Hive** via **HUE**, permettant de segmenter et quantifier les arrêts sur panne afin d’identifier les moments les plus critiques de la production.

 **Environnement technique : HDF (nifi, kafka**, zookepper, Storm, ambari), HDP (HDFS, Spark, Zeppelin, Oozie, Sqoop, Ranger, Hive, HBase), ELK (ElasticSearch, logstash &Kibana), HUE : **Hadoop User Experience**



# FORMATIONS & CERTIFICATIONS

 **2016 :** Master 2 Data Scientist (Université Paris Saclay)

 **2015 :** Master 1 Business Intelligence (Université Paris Saclay)

 **2014 :** Licence 3 Informatique (Université Paris Sud, Orsay)

 **2021** : Databricks Certified Associate Developer for Apache Spark Scala

 **Microsoft Certified Azure Data Engineer ** **Google Cloud Professional Data Engineer**