WIKI LAB

Plateforme Intelligente De Gestion Des Risques Et De Conformité

Solution IA : Plus rapide, plus précise, plus fiable

✔ Détection des fraudes en temps réel

✔ Évaluation des risques nouvelle génération

✔ Conformité réglementaire automatisée

Version 07 / Juillet 2025

Onglet 1 : Présentation générale **4**
GENERALITES 5
I. Architecture Globale du LAB 5
II. Contributeurs 6
III. Datasets 7
IV. Scénario 7
CRÉATION DES CAS D'USAGE 9
I. Accès au module Administration 9
1. Gestion des Use Cases 9
2. Gestion des Utilisateurs : 10
3. Paramètres 10
4. Journaux (Logs) 11
II. Configuration d'un Nouveau Use Case 11
GESTION DES DATASETS 14
I. Ingestion des données 14
II. Script d'ingestion (Job) 17
CRÉATION DE SCÉNARIO 20
I. Étapes de création de scénario: 20
II. Utilisation d'un Scénario IA 23
1. Interface de Monitoring de l'Instance 23
2. Création de Jobs Automatisés 26
3. Importation des Datasets 27
Onglet 2 : Ingestion des données **31**
DATA INGESTION 32
Introduction 32
I. Ingestion manuelle via l'interface Kepler 32
A. Ingestion depuis un fichier 33
B. Ingestion depuis une base de données 34
II. Ingestion automatique via Script (Job planifié) 35
III. Cycle de Vie du Script 37
IV. Avantages d'une ingestion centralisée dans Kepler 38
V. Extraction des Données 39
1. Mode append (incrémental) 39
2. Mode overwrite (écrasement complet) 39
Conclusion 40
DATA PREPARATION 42
Introduction 42
I. Étapes de préparation dans Jupyter 42
1. Accès à l'environnement 42
2. Développement des notebooks 43
3. Visualisation et vérification 44
II. Automatisation avec Cron Job 44

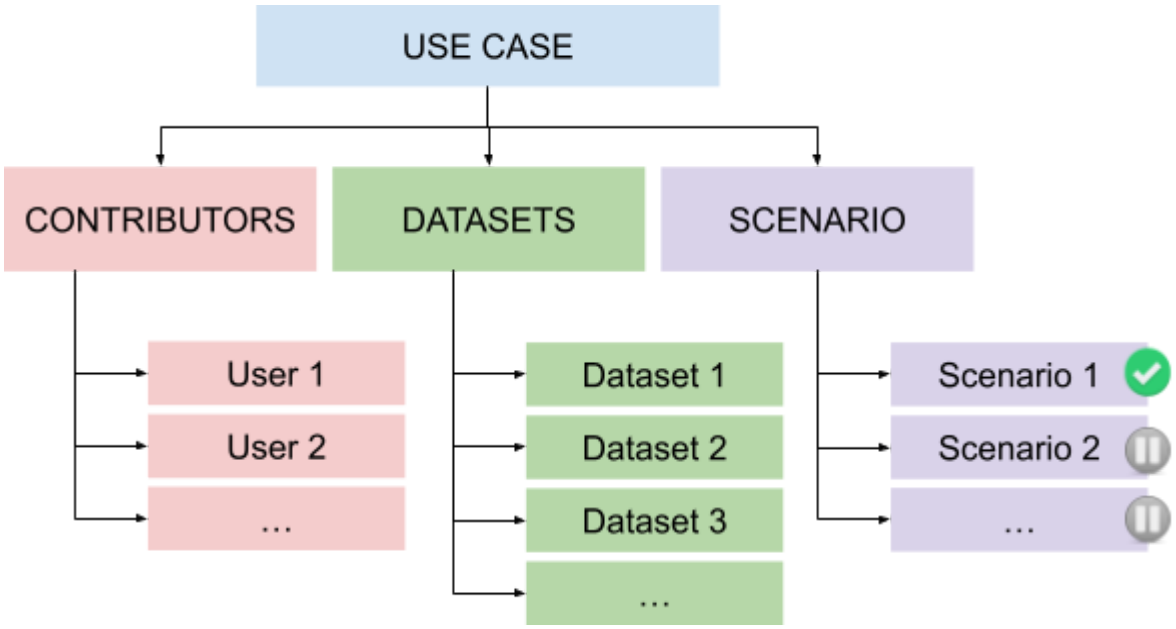
1. Transformation du notebook en script exécutable 44
2. Création du Cron Job 44
3. Paramètres à définir 45
4. Enregistrement 45
Conclusion 46
Onglet 4: Entraînement des données 47
DATA TRAINING 48
Introduction 48
I. Phase manuelle dans Jupyter 48
1. Accès à l'environnement Jupyter via KEPLER : 48
2. Travail dans le répertoire training 49
II. Création d'un Job d'entraînement automatisé 50
1. Création du Job dans KEPLER 50
III. Objectifs de l'étape de Training 51
IV. Visualisation et ajustements 52
Conclusion 52
Onglet 5: Scorage des données 53
DATA SCORING 54
Introduction 54
I. Phase manuelle dans Jupyter 54
II. Automatisation avec un Cron Job 55
III. Objectifs de l'étape de Scoring 56
IV. Enregistrement et automatisation 59
Conclusion 59

Par tanitlab AI

Onglet 1 : Présentation générale

GENERALITES

I. Architecture Globale du LAB



Chaque Use Case dans **KeplerAI** agit comme un segment autonome qui porte des notions de droits d'accès, des données et des algorithmes. Le Use Case contient trois éléments clés :

1. **Contributeurs**
   - Liste des utilisateurs associés au Use Case (ex: *User1: Data Scientist, User2: AML Analyst*).
   - Gestion des permissions via le module Administration.
2. **Datasets**
   - Collections de données spécifiques au Use Case (ex: *Dataset1: Transactions_2023, Dataset2: Client_Profiles*).
   - Formats supportés : CSV, JSON, Parquet (taille max. 5 Go par fichier).
3. **Scénarios**
   - Workflows d'analyse ou de modélisation (ex: *Scenario1: Détection d'Anomalies, Scenario2: Scoring Clients*).
   - Chaque scénario dispose :
     - D'un ou plusieurs notebooks Jupyter dédiés
     - De ressources Spark paramétrables (CPU/RAM)

- **1 unique scénario** peut être activé par Use Case

II. Contributeurs

Le module de gestion des contributeurs permet d'administrer simplement les accès et permissions des utilisateurs sur la plateforme Kepler AI.

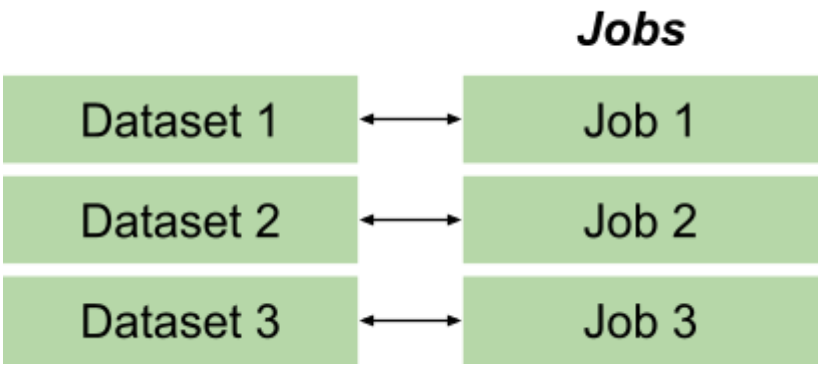
	Admin	Devops	Manager	Standard
User 1	X	X		
User 2			X	
…				X

Fonctionnalités principales :

- Attribution visuelle et intuitive des rôles
- Quatre niveaux d'accès disponibles :
  - **Admin** - Accès complet à tous les paramètres
  - **DevOps** - Déploiement et supervision technique
  - **Manager** - Validation et suivi des projets
  - **Standard** - Utilisation basique des fonctionnalités

III. Datasets

Les datasets dans **KeplerAI** sont des collections de données structurées spécifiques à chaque Use Case. Chaque dataset est traité par des jobs dédiés pour assurer une transformation



optimale.

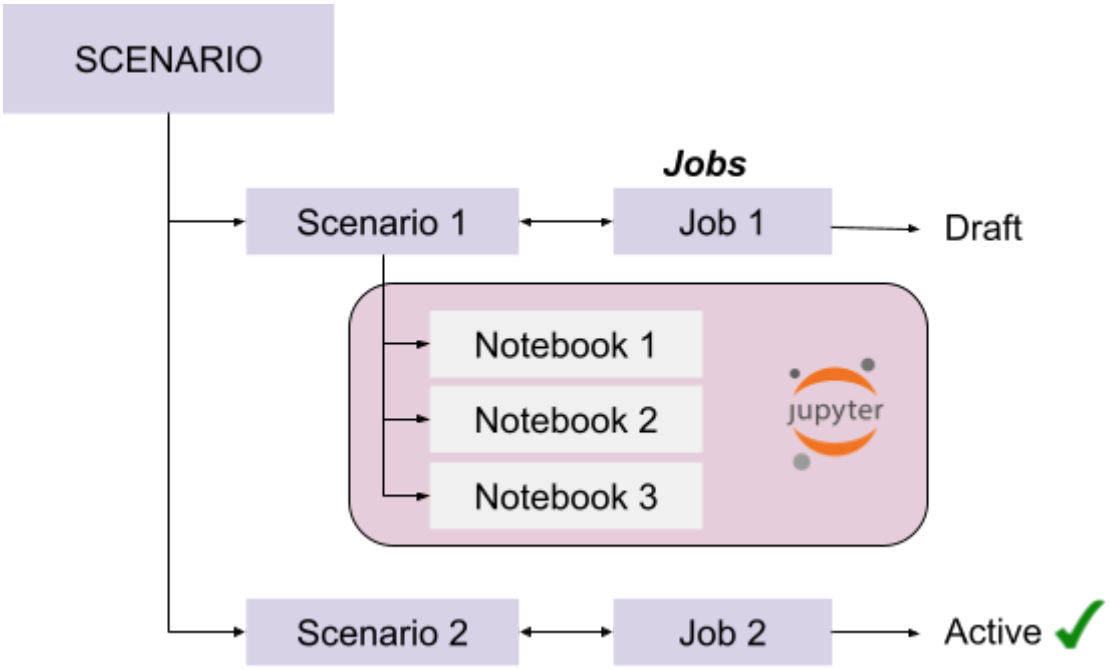
Des jobs pourront être utilisés en particulier afin de mettre à jour régulièrement les données sources.

IV. Scénario

Il s'agit de scénario de test et de développement. La plateforme permet de tester de nouveaux modèles sans affecter celui qui est activé et en production.

1 unique scénario peut être actif par use case.

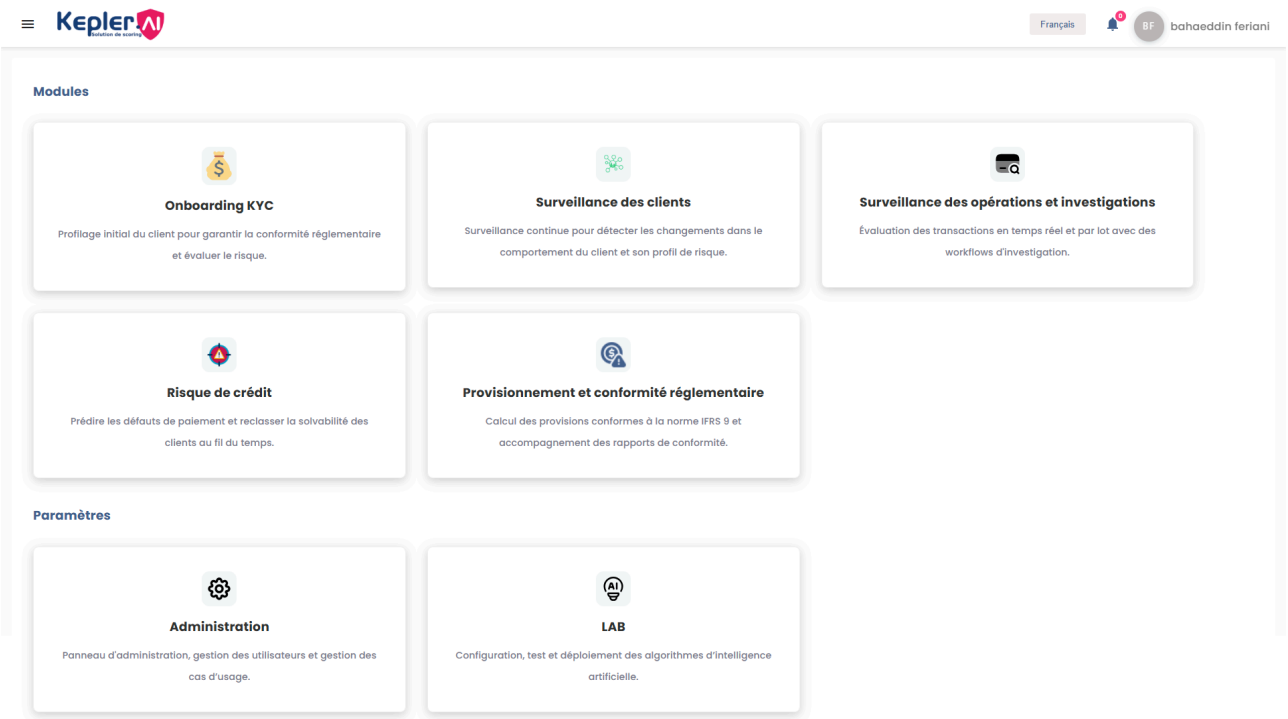
Un scénario combine plusieurs notebooks Jupyter en un seul job, produisant une sortie Draft (brouillon testable) ou Active (déployé en production).


SCENARIO
Jobs
Scenario 1
Job 1
Draft
Notebook 1
Notebook 2
Notebook 3
jupyter
Scenario 2
Job 2
Active

Par tanitlab AI

CRÉATION DES CAS D'USAGE

Avant toute modélisation ou exploration de données, la première étape clé dans **KeplerAI** consiste à **créer un Use Case** depuis le module **Administration**. Cette étape permet de structurer le travail à venir et d'organiser les droits d'accès.



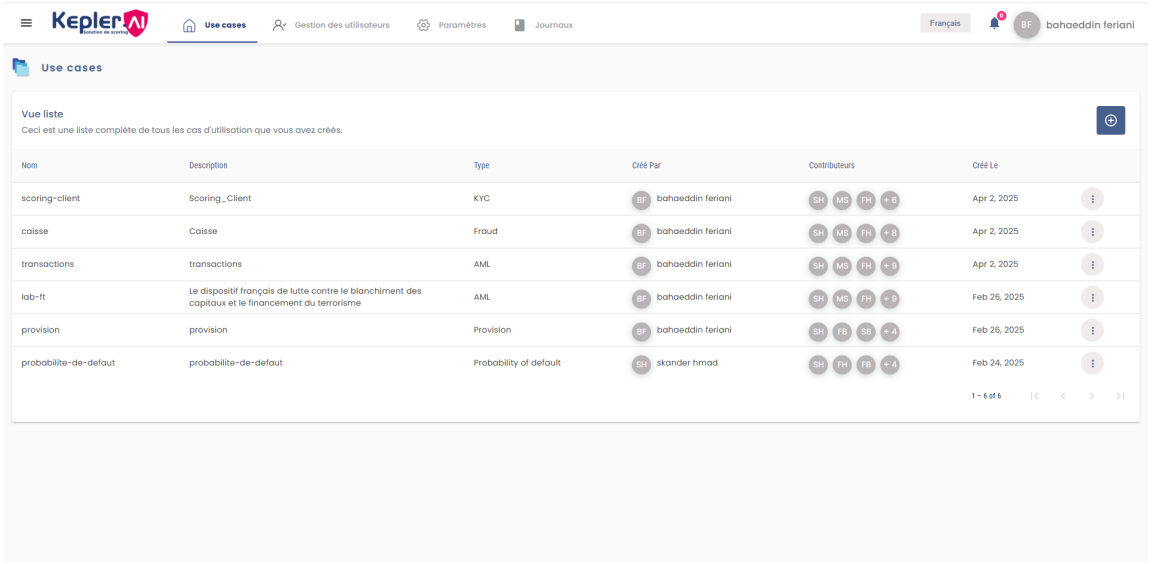
I. Accès au module Administration

L'administrateur accède au module **Administration** depuis le menu principal. Ce module centralise les fonctionnalités suivantes :

1. Gestion des Use Cases

- **Liste complète :** Affiche tous les cas d'utilisation créés.
- **Bouton de création :** Permet d'ajouter un nouveau Use Case.

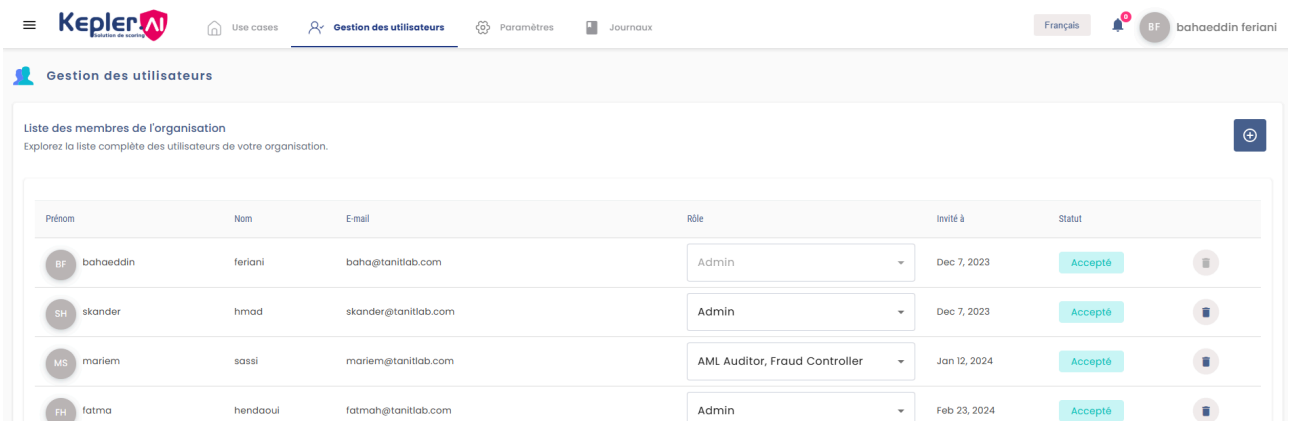
Administration>Use cases



2. Gestion des Utilisateurs :

- Ajout, modification ou suppression d'utilisateurs.
- Attribution ou modification des rôles (ex : Admin , AML Auditor,…).

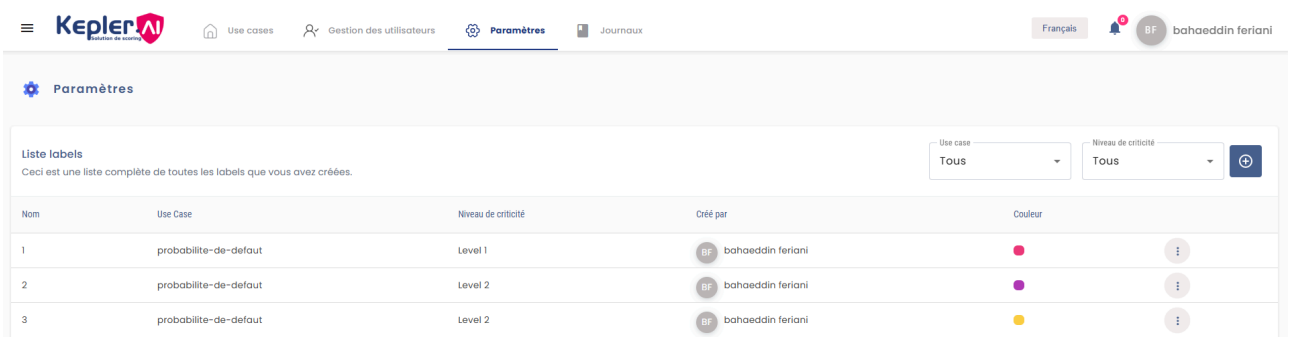
Administration>Gestion des utilisateurs



3. Paramètres

- **Labels personnalisés :** Création de libellés spécifiques selon les Use Cases et les niveaux de criticité .

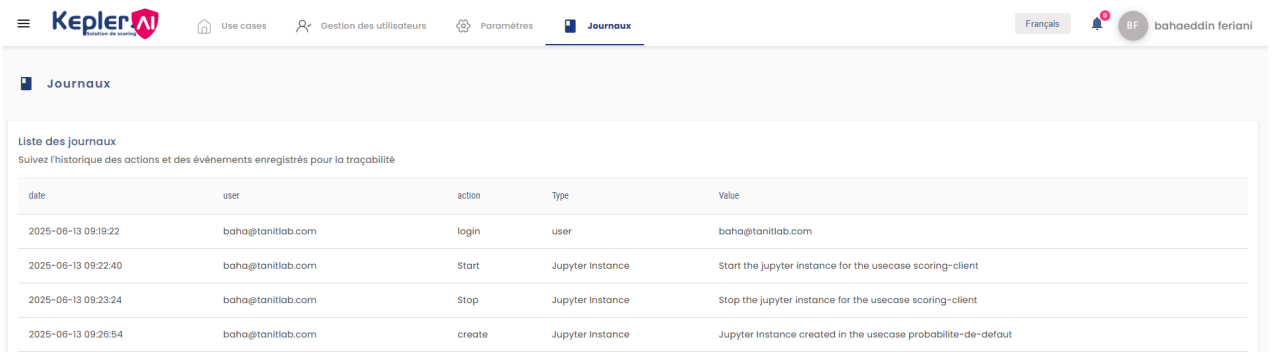
Administration>Paramètres



4. Journaux (Logs)

- Historique des actions et événements pour la traçabilité (ex : connexions, modifications,...).

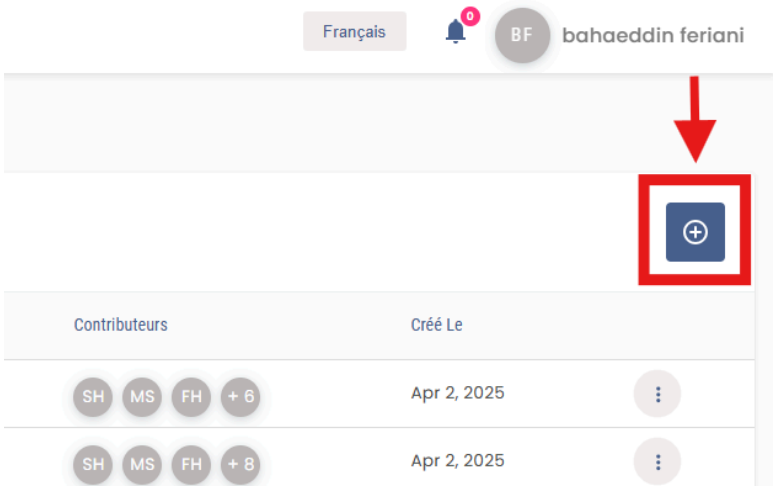
Administration>Journaux



II. Configuration d'un Nouveau Use Case

Depuis le menu **Administration**, l'administrateur clique sur le bouton **"➕"** (situé en haut à gauche de l'écran) pour initier la création d'un nouveau Use Case.

Administration>Use cases



Une fenêtre de configuration s'ouvre, permettant de renseigner les éléments suivants :

- **Nom du Use Case** : un titre explicite et unique pour identifier le cas d'usage
- **Description** : une description du Use Case (ex. : détection de Transactions Atypiques)
- **Type de Use Case** : l'administrateur sélectionne la catégorie pertinente parmi :
    - Onboarding KYC (Know Your Customer)
    - Surveillance des clients
    - Surveillance des opérations et investigations
    - Risque de crédit
    - Provisionnement et conformité réglementaire

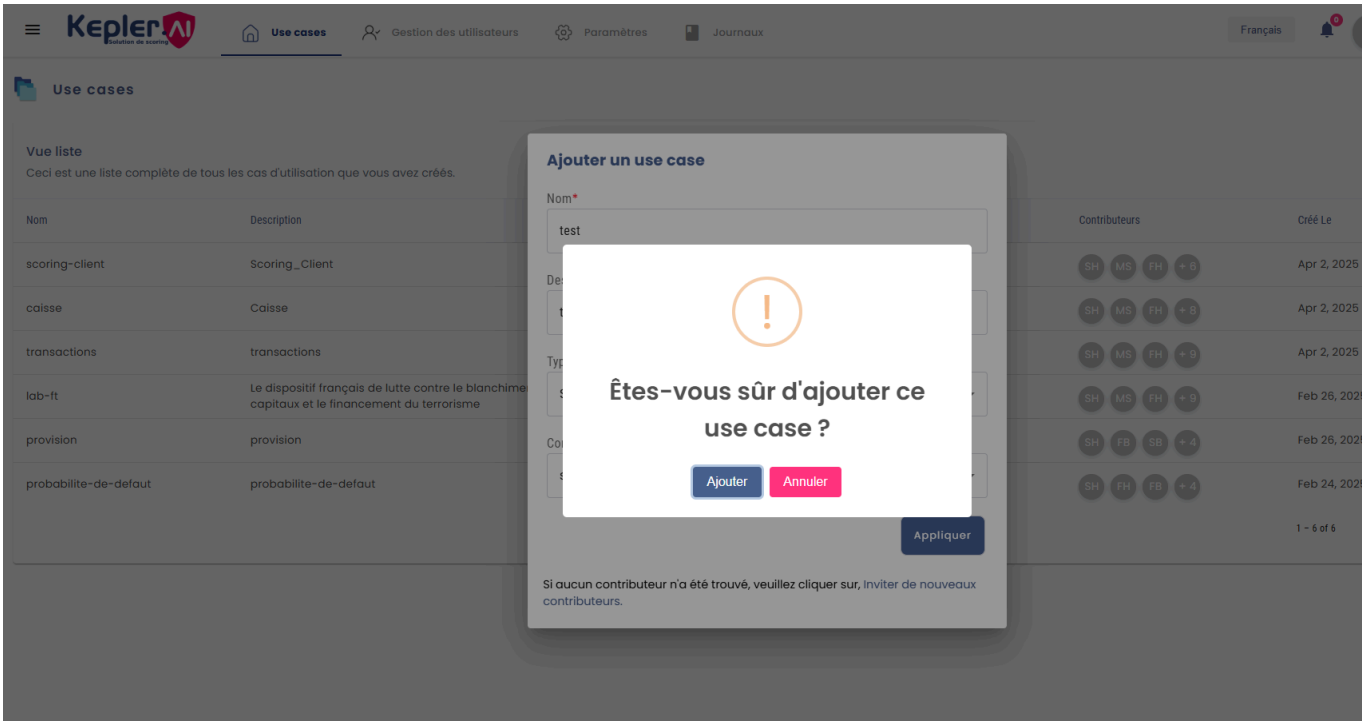
- **Contributeurs** : il s'agit des utilisateurs (Data Scientists, Analysts…) qui auront accès à ce Use Case. Ils pourront visualiser les données, créer des jobs de traitement, lancer des modèles, et suivre les alertes dans la plateforme de monitoring.

Administration>Use Cases>Ajouter Un Use Case

🔔 **Remarque :** Seuls les contributeurs ajoutés pourront voir ce Use Case dans leur environnement de travail.

Une fois tous les champs remplis, l'administrateur valide la création. Le Use Case devient alors actif et visible par les contributeurs désignés.

Administration>Use Cases>Ajouter Un Use Case>Appliquer



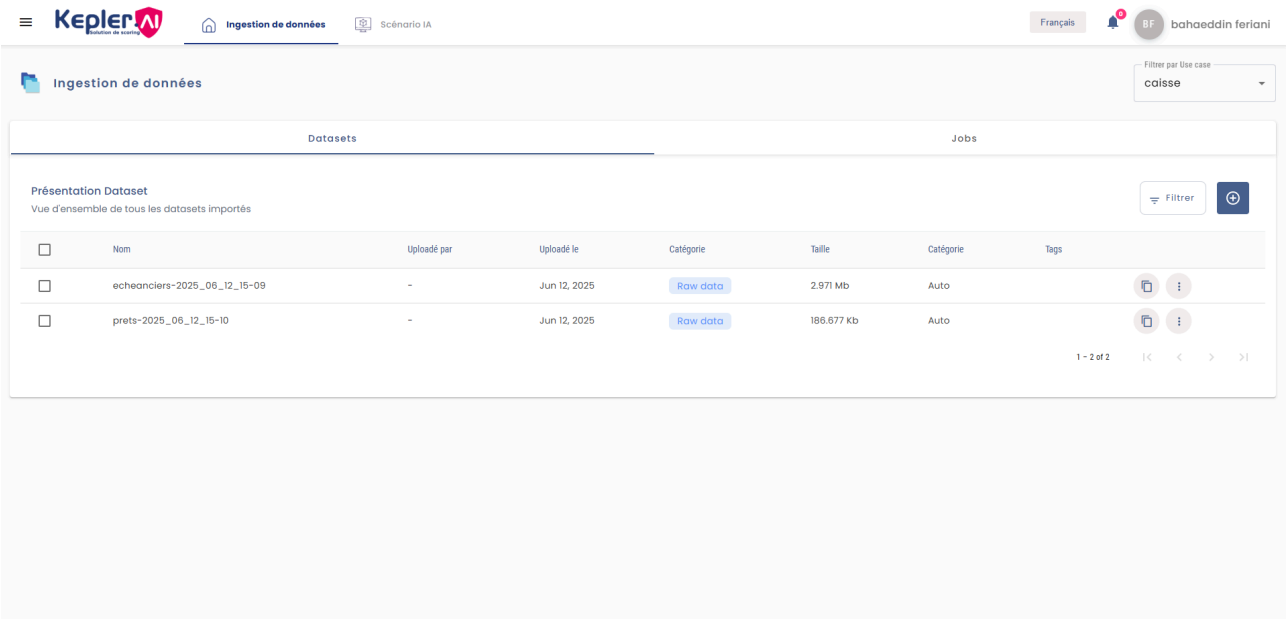
Par tanitlab AI

GESTION DES DATASETS

I. Ingestion des données

L'ingestion des données se fait depuis le **module LAB** de l'application **KeplerAI**. C'est la première étape essentielle permettant d'introduire des données issues de différentes sources pour leur traitement, analyse et exploitation.

LAB>Ingestion de données>Datasets



Les données peuvent provenir de **fichiers structurés** (CSV, XLSX, JSON, Parquet) ou de **bases de données relationnelles** telles que **MySQL**, **PostgreSQL** ou **SQL Server**.

📍 **Étapes** :

1. **Sélection du Use Case concerné**

   Depuis le module LAB, l'utilisateur doit d'abord sélectionner le *Use Case* pour lequel il souhaite ingérer des données. Cela permet de contextualiser les jeux de données et de les associer à un scénario spécifique.

2. Ajout d'un Dataset

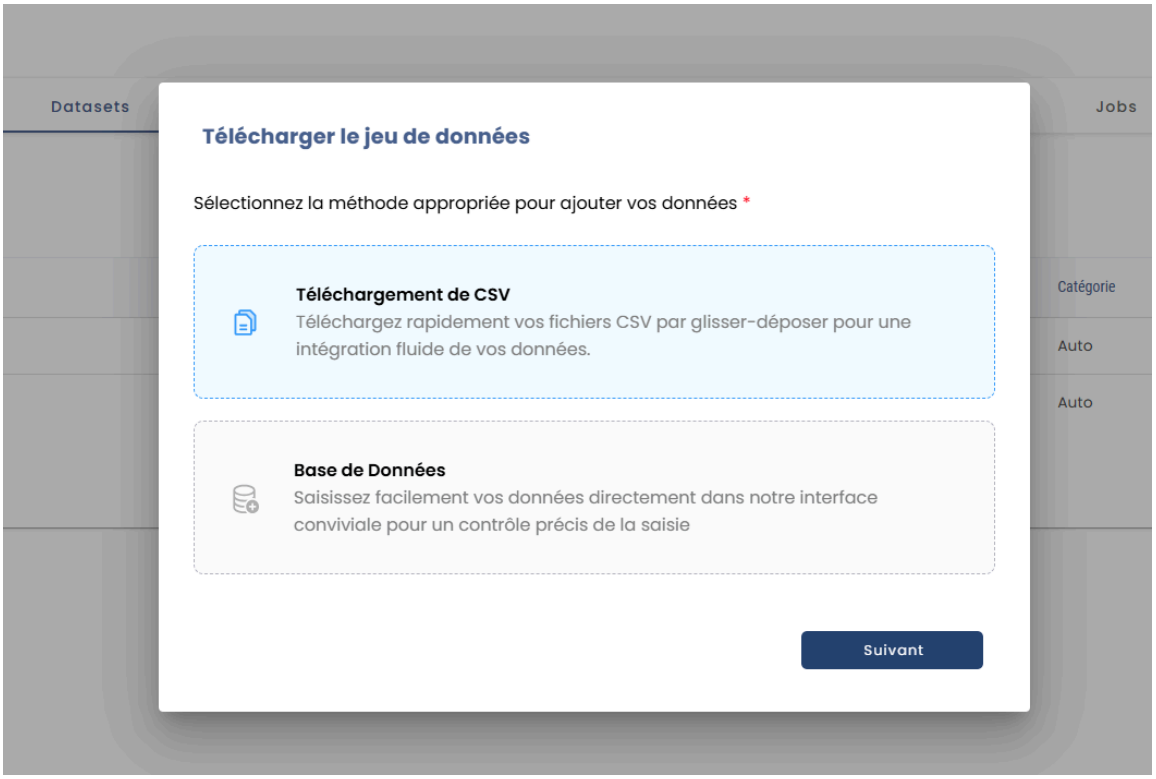
Cliquer sur le bouton ➕ « Ajouter un dataset ».

LAB>Ingestion de données>Datasets

💡**Les datasets ajoutés ici seront réutilisables lors de la création de scénarios, notamment dans la phase d'ingestion de données d'un Use Case.**

3. Choix du mode d'ingestion selon la source

LAB>Ingestion de données>Datasets>+

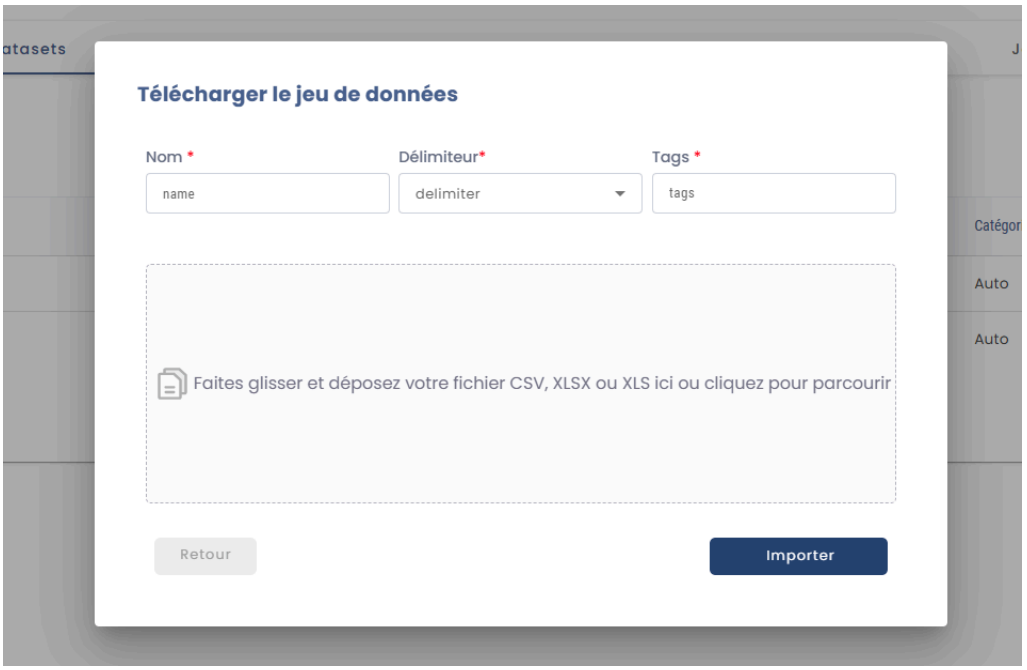


Deux cas sont possibles selon le type d'insertion des données :

➔ 📁 **Ingestion manuel depuis un fichier** :

- Télécharger le jeu de données
- Renseigner les informations suivantes : **Nom** du dataset (name), **Délimiteur** utilisé (delimiter) et **Tags** associés (tags).
- Glisser et déposer le fichier **CSV**, **XLSX** ou **JSON**, ou cliquer pour le sélectionner.

LAB>Ingestion de données>Datasets>+>Téléchargement de CSV

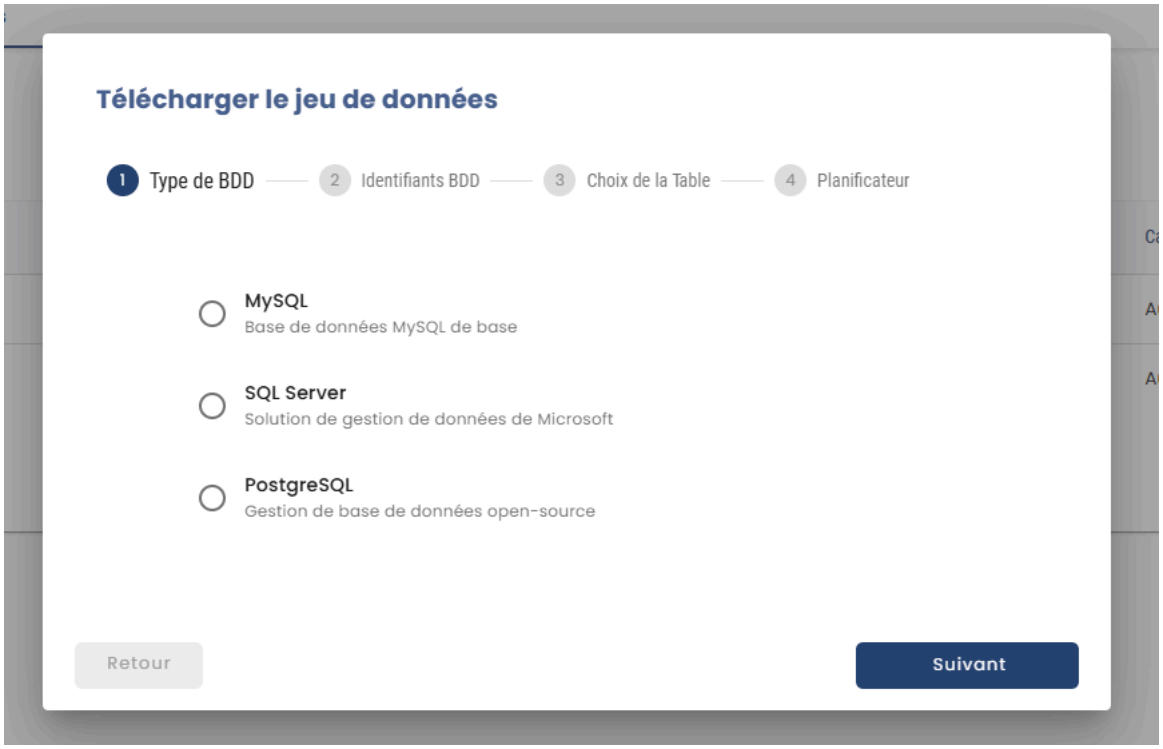


➔ 🗄️ **Ingestion depuis une base de données** :

- **Étape 1 :** Choisir le **type de base de données** (MySQL, SQL Server, PostgreSQL)
- **Étape 2 :** Fournir les **identifiants de connexion** (adresse, port, nom d'utilisateur, mot de passe)
- **Étape 3 :** Sélectionner la **table** à importer

- **Étape 4 :** Configurer un **planificateur** pour automatiser l'ingestion (optionnel)

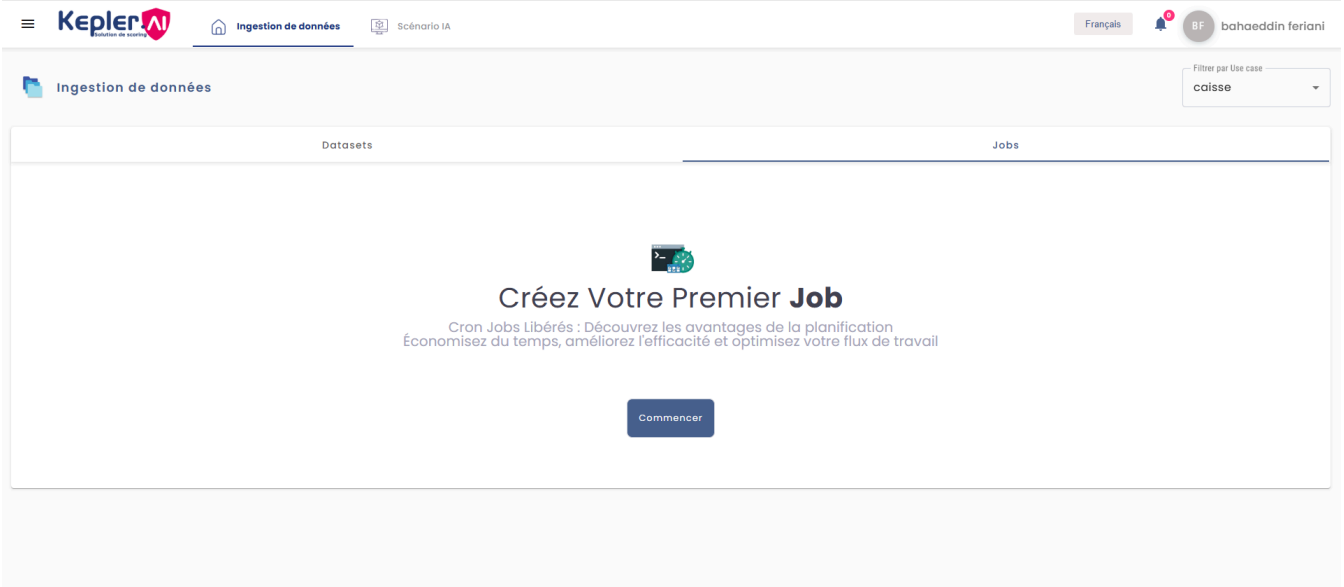
LAB>Ingestion de données>Datasets>+>Base de Données



II. Script d'ingestion (Job)

L'ingestion automatisée des données dans **KeplerAI** se fait via la création de **Jobs planifiés**, appelés **Cron Jobs**, disponibles dans le module **LAB** .

LAB>Ingestion de données>Jobs



Ces Jobs permettent de déclencher des scripts à des intervalles réguliers (quotidien, hebdomadaire, mensuel…) pour traiter ou mettre à jour les données de manière autonome, sans intervention manuelle.

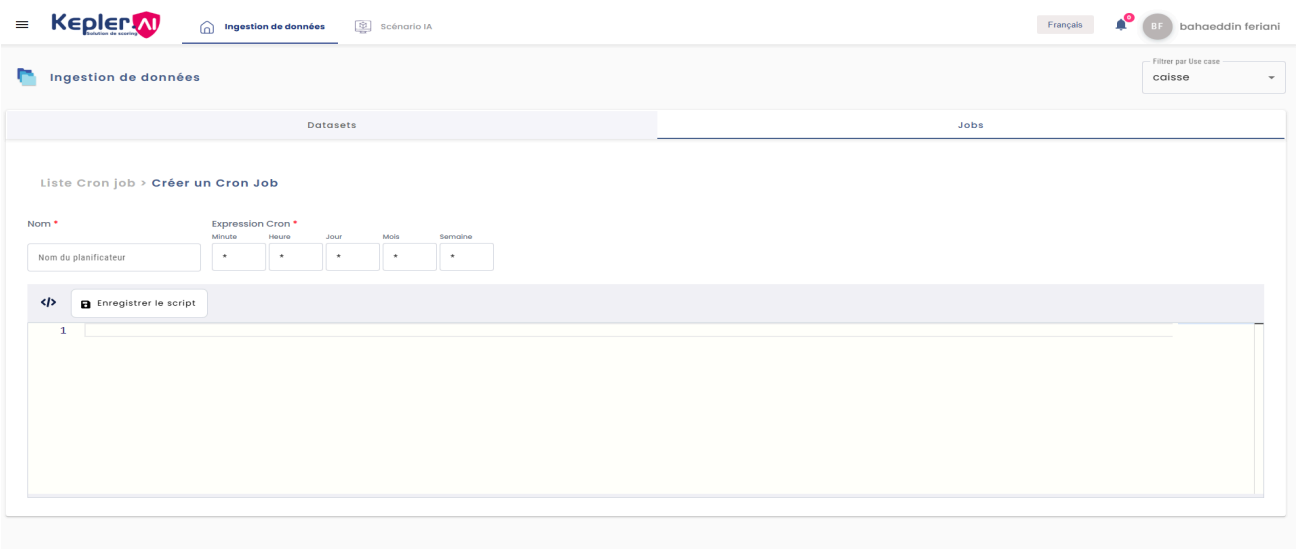
📍 **Étapes** :

1. Sélectionner le **Use Case** concerné
2. Cliquer sur le bouton **"+"** ou sur **"Commencer"**
3. Une interface de création de **Cron Job** s'affiche , structurée autour des éléments suivants :

Champ	Description
Nom du Job	Saisir un nom explicite pour identifier la tâche d'ingestion automatisée.
Expression CRON	Définir la fréquence d'exécution à l'aide d'une expression CRON (Minute, Heure, Jour, Mois,semaine).
Zone de saisie du script	Un éditeur de texte permet à l'utilisateur d'écrire directement ou de coller un script d'ingestion (Python ou Spark) à exécuter périodiquement.

LAB>Ingestion de données>Jobs>Commencer

Par tanitlab AI



Par tanitlab AI

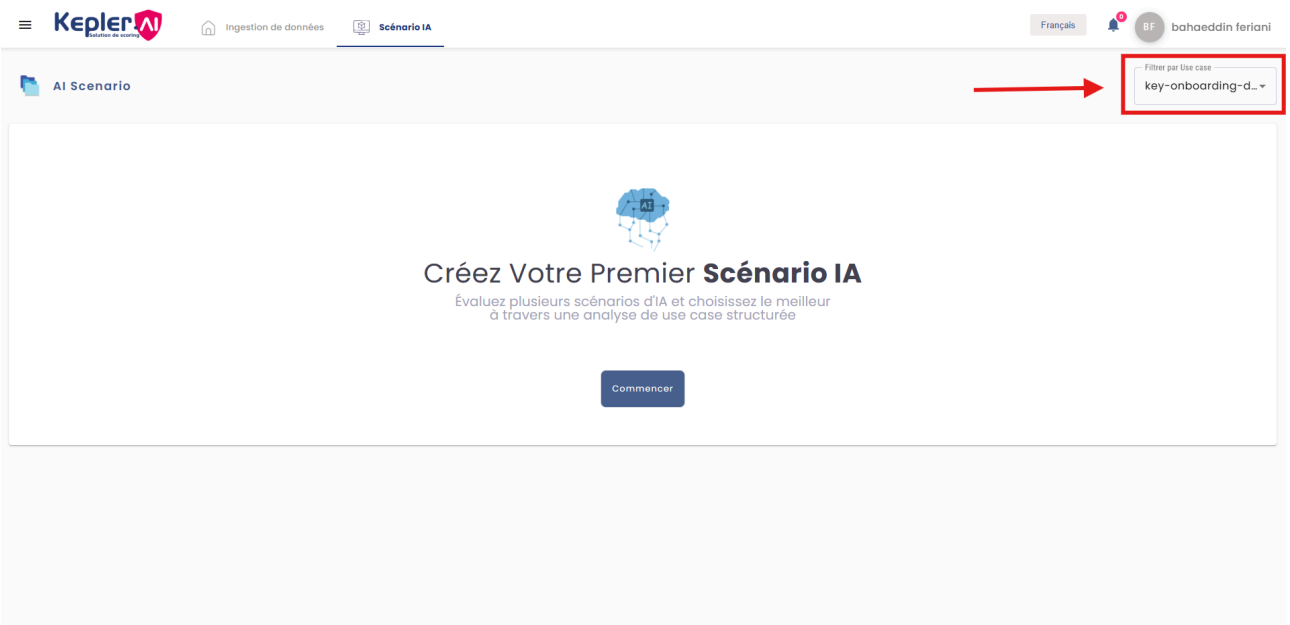
CRÉATION DE SCÉNARIO

Une fois le Use Case créé et accessible, le **Data Scientist** peut entamer la création de son **Scénario IA** via le module **LAB**, qui regroupe toutes les fonctionnalités d'ingénierie et d'entraînement de modèles IA.

I. Étapes de création de scénario:

1. Depuis le module **LAB , le module scénario IA** Sélectionner le Use Case précédemment créé (via le menu déroulant)

LAB>Scénario IA

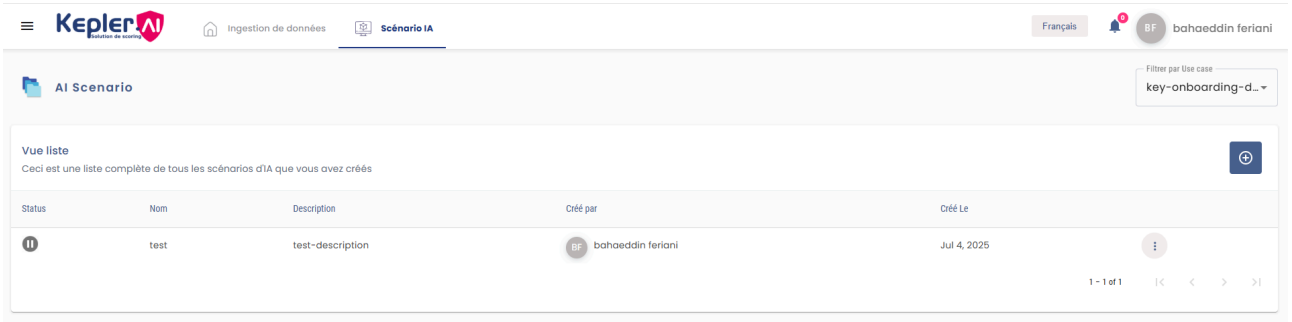


2. Cliquer sur le bouton **"+"** ou sur **"Commencer"** , puis renseigner :
    a. **Nom du scénario**
    b. **Description**

LAB>Scénario ***IA>Commencer***

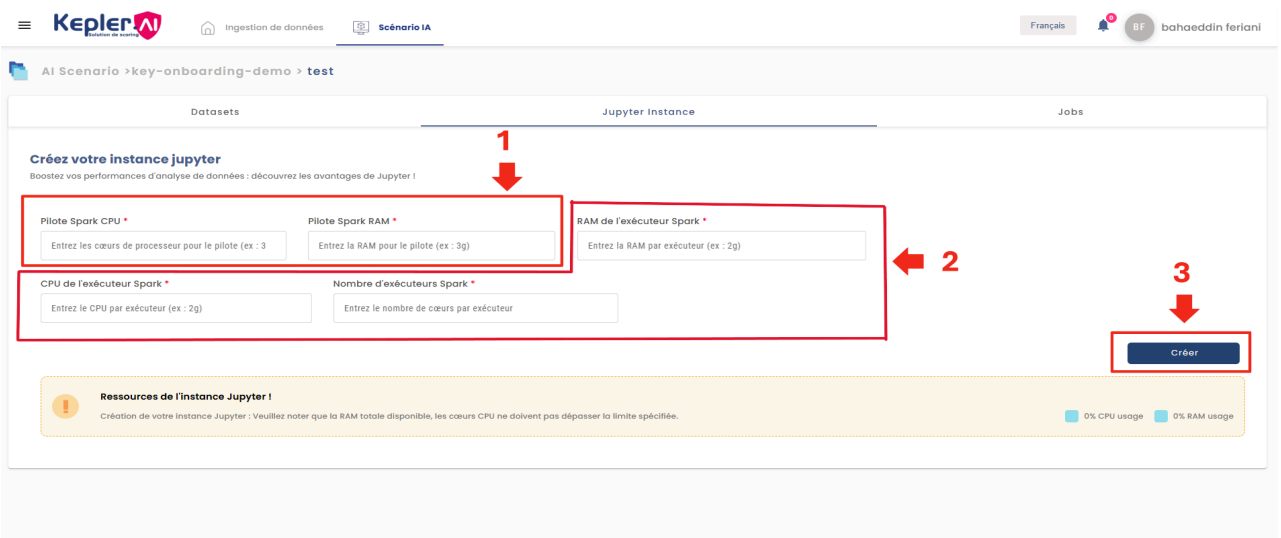
3. Une fois le scénario confirmé, il apparaît dans la liste des scénarios du Use Case.

LAB>Scénario IA



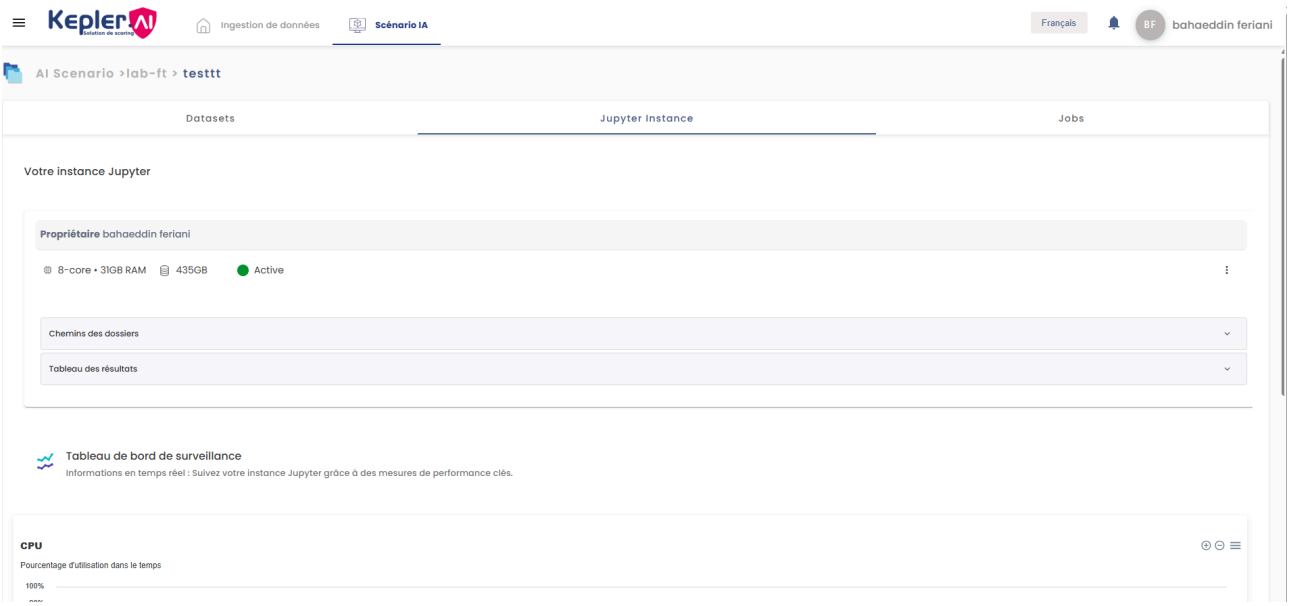
4. Cliquer sur ce scénario pour passer à l'étape de **création d'une instance Jupyter** :
    1. Configurer le Pilote Spark
        a. CPU : Spécifiez le nombre de cœurs (exemple : 3)
        b. RAM : Entrez la mémoire en Go (exemple : 30)

LAB>Scénario IA>Exemple Scénario>Jupyter Instance



2. Configurer les Exécuteurs Spark
   a. CPU par exécuteur : Définissez la puissance (exemple : 4)
   b. RAM par exécuteur : Spécifiez la mémoire (exemple : 20)
   c. Nombre d'exécuteurs : Déterminez la quantité nécessaire (exemple : 5)
3. Créer Cliquez sur le bouton "Créer" pour lancer l'instance avec les paramètres définis.
4. Interface de l'instance Jupyter après création :

LAB>Scénario IA>Exemple Scénario>Jupyter Instance



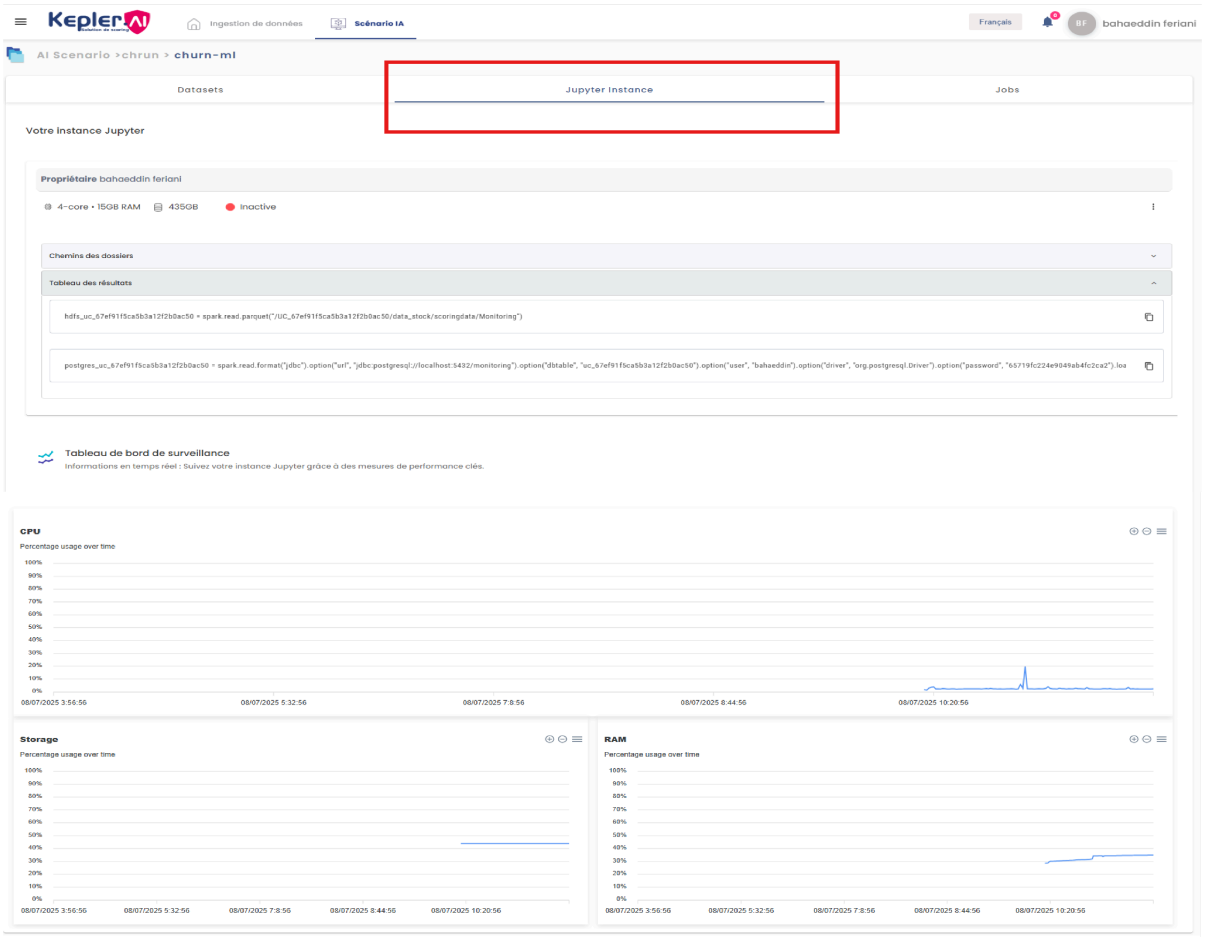
Une fois ces paramètres validés, l'instance Jupyter est automatiquement déployée, permettant au Data Scientist d'écrire et exécuter son code d'analyse, de transformation et de modélisation directement dans l'environnement **KeplerAI**.

II. Utilisation d'un Scénario IA

KeplerAI intègre une plateforme unifiée permettant d'orchestrer l'ensemble du cycle de vie des projets d'intelligence artificielle. Grâce à son interface centralisée, les équipes data peuvent piloter efficacement leurs expérimentations tout en garantissant traçabilité et reproductibilité.

1. Interface de Monitoring de l'Instance

LAB>Scénario IA>Exemple Scénario>Jupyter Instance



Une fois l'instance Jupyter lancée dans **KeplerAI**, le Data Scientist bénéficie d'une interface de monitoring centralisée, lui permettant de :

- **Suivre en temps réel** l'état de son instance grâce à des **indicateurs de performance** (CPU, RAM, statut des exécuteurs…).
- **Mettre en pause**, **supprimer** ou **relancer** son instance à tout moment.
- Supprimer ou stopper l'instance permet de libérer les ressources infrastructures pour 'autres initiatives et use case
- Cliquer sur **"Ouvrir"** pour accéder directement à l'environnement **Jupyter Notebook** intégré à Kepler.

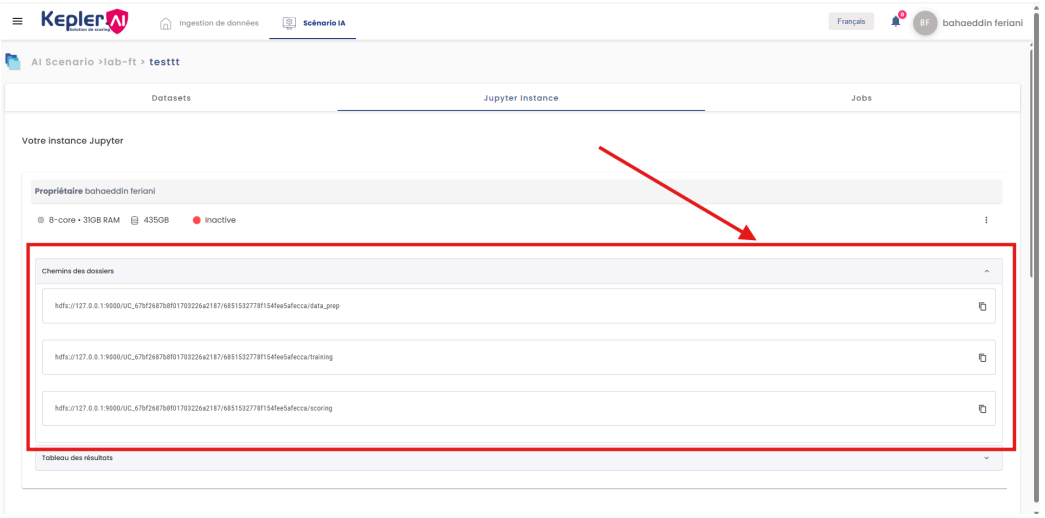
LAB>Scénario IA>Exemple Scénario>Jupyter Instance

La plateforme **KeplerAI** permet également de faciliter la navigation vers les résultats produits par les traitements IA. Depuis l'interface de monitoring ou directement dans l'environnement Jupyter :

1. Cliquer sur ce scénario AI pour passer à la liste complète de tous les scénarios d'IA
2. KYC (Know Your Customer) sur un de ces scénarios ou créer un nouveau scénario
3. Créer une instance Jupyter en remplissant les inputs puis cliquer sur le bouton Créer pour lancer la création
4. Un nouvel interface s'ouvre comportant tous les composants de votre instance Jupyter (user, charts...)
5. Lorsque vous cliquez sur la section "Chemins des dossiers", une liste s'affiche comportant tous les chemins HDFS que vous pouvez copier pour les intégrer dans d'autres scripts ou workflows.
    - **Data Prep:** pour la data préparation et transformation afin d'aboutir à un dataset exploitable
    - **Training:** pour les algorithms de training (apprentissage) et testing (test)

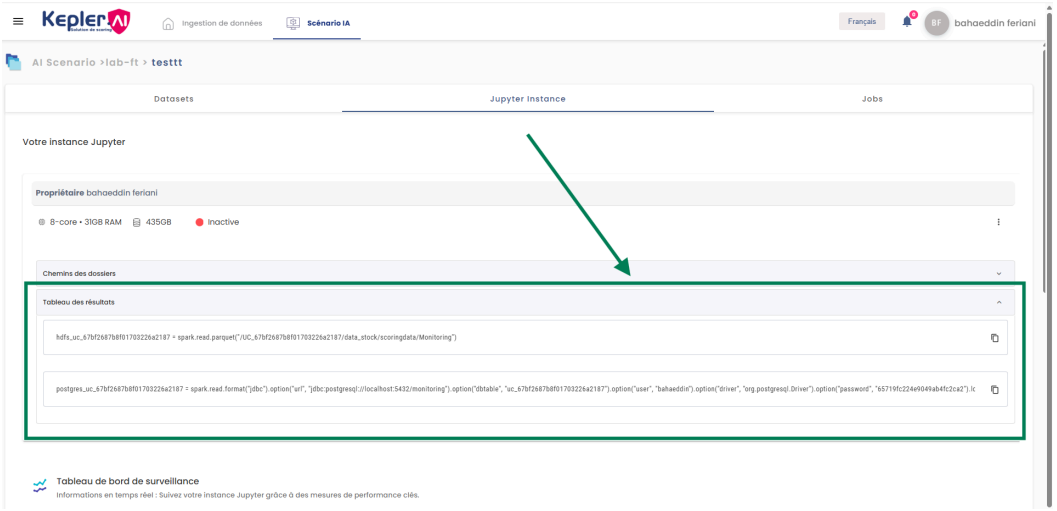
- **Scoring:** pour les algorithms de scoring des nouvelles instances et qui vont être utilisés en production

LAB> AI Scenario > Example Scenario > Jupyter Instance>Chemins des dossiers



- Des liens vers les résultats stockés en HDFS ou dans des bases PostgreSQL sont disponibles et peuvent être copiés en un clic pour consultation ou partage.Pour y accéder, suivez les étapes suivantes :
    1. **Cliquer sur ce scénario IA** pour afficher la liste complète de tous les scénarios d'IA disponibles.
    2. **Sélectionner un scénario existant** ou bien **créer un nouveau scénario** selon vos besoins.
    3. **Créer une instance Jupyter** en remplissant les champs requis, puis cliquer sur le bouton **"Créer"** pour lancer la génération de l'environnement.
    4. Une **nouvelle interface s'ouvre**, affichant tous les composants de votre instance Jupyter (utilisateur, graphiques, etc.).
    5. En cliquant sur la section **"Tableau des résultats"**, une **liste de liens vers les résultats** (HDFS ou PostgreSQL) s'affiche. Ces liens peuvent être **copiés en un clic** pour faciliter la consultation ou le partage.

LAB> AI Scenario > Example Scenario > Jupyter Instance>Tableau des résultats



💡 **Ces fonctionnalités permettent un gain de temps considérable dans l'analyse des résultats, le traçage des données, et l'industrialisation des traitements IA.**

2. Création de Jobs Automatisés

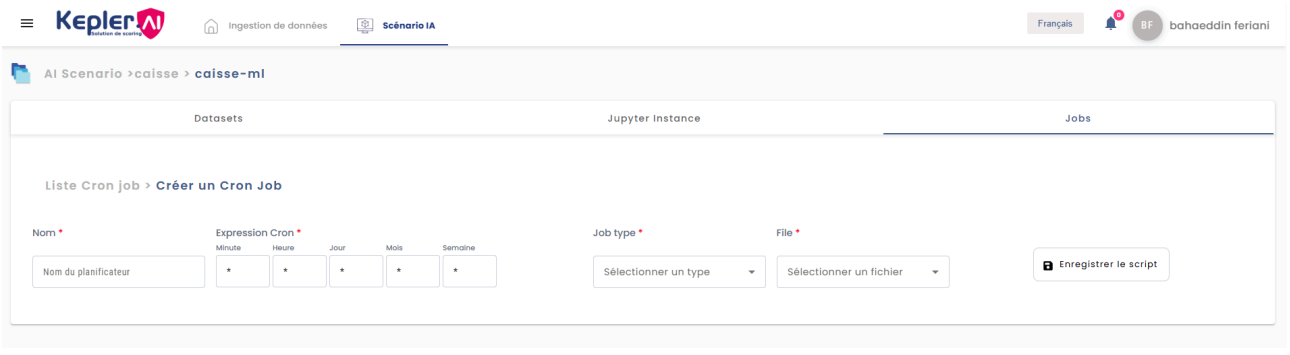
Kepler propose un système de planification de traitements via l'onglet **Jobs**.

LAB>Scénario IA>Exemple Scénario>Jobs

📍 **Étapes :**

1. Depuis la fenêtre **Jobs**, Cliquer sur le bouton **"+"** ou sur **"Commencer"**
2. Dans l'interface de création de Job, le Data Scientist renseigne :

LAB>Scénario IA>Exemple Scénario>Jobs



- **Nom du Job**
- **Expression CRON*** : heure ou fréquence d'exécution du script
- **Type de Job*** :
  a. Scenario Training folder
  b. Scenario Scoring folder
  c. Scenario Data Prep folder
- **fichier utiliser**
- Le **Job est automatiquement associé au Use Case actif**, identifié par son ID unique au format UC_<ID>.

💡 **Le Job ainsi créé permet d'automatiser des tâches critiques telles que l'entraînement de modèles, le scoring périodique, ou la préparation des données, tout en assurant une traçabilité et une reproductibilité complètes.**

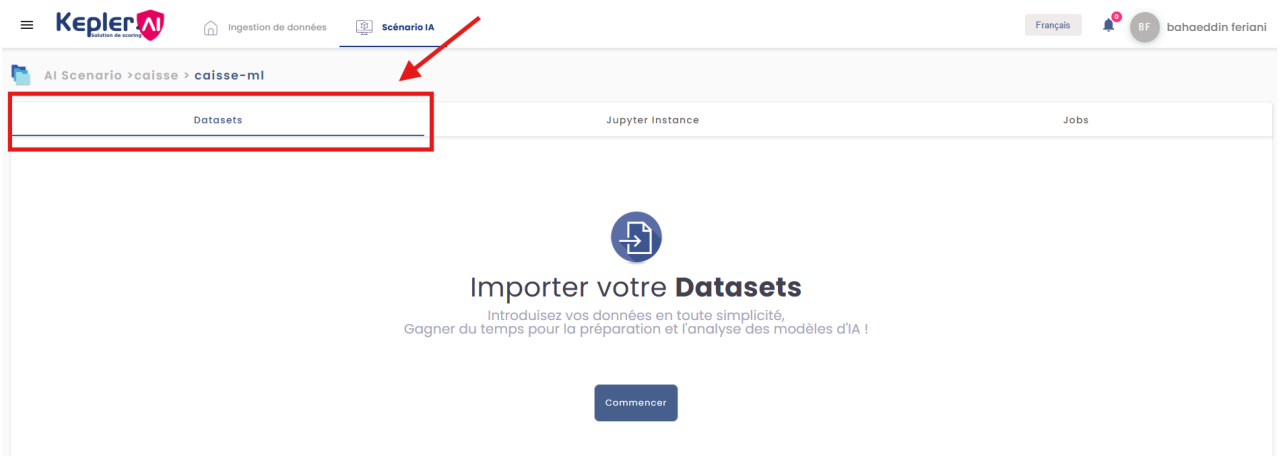
La section suivante détaille les étapes clés du cycle de vie d'un scénario IA, à savoir : **la préparation des données (Data Preparation), l'entraînement des modèles (Data Training)**, ainsi que **le scoring et l'exploitation des résultats (Data Scoring).**
Ces étapes sont essentielles pour garantir la qualité, la performance et la robustesse des solutions déployées.

3. Importation des Datasets

Les datasets utilisés dans les scénarios d'IA de KeplerAI ont déjà été importés via l'interface d'ingestion de données, en fonction du use case sélectionné. Dans cette étape, il s'agit simplement d'associer ces datasets existants au scénario IA pour lancer les phases de préparation, d'entraînement et de scoring des modèles.

📍 **Étapes :**

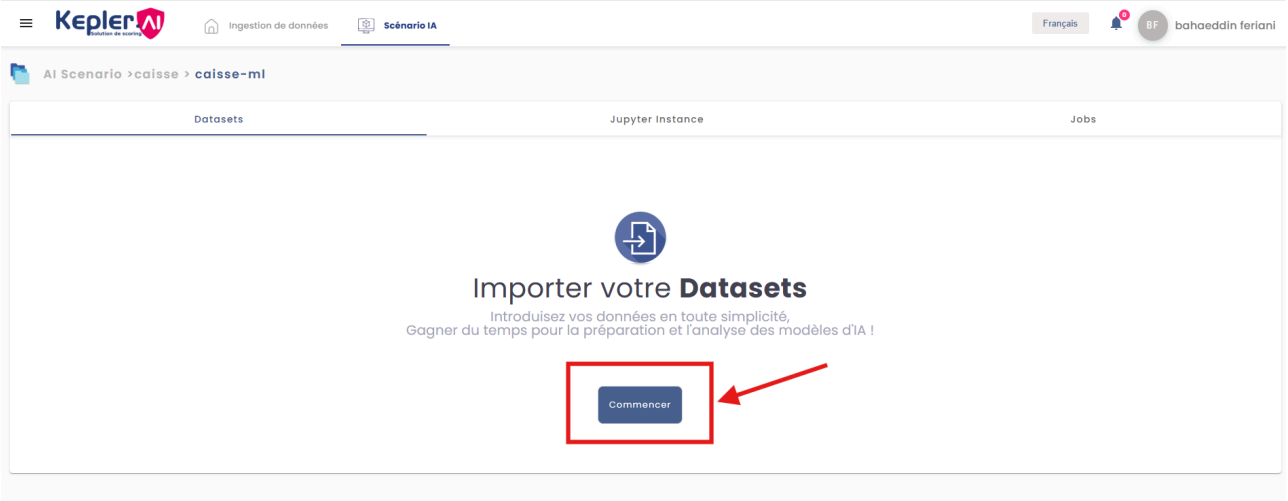
1. **Accéder au module d'importation :**
   LAB>Scénario IA>Exemple Scénario>Datasets



2. **Lancer l'importation :**

   Cliquez sur le bouton « + » ou sur « **Commencer** » pour démarrer le processus d'importation.

LAB>Scénario IA>Exemple Scénario>Dataset



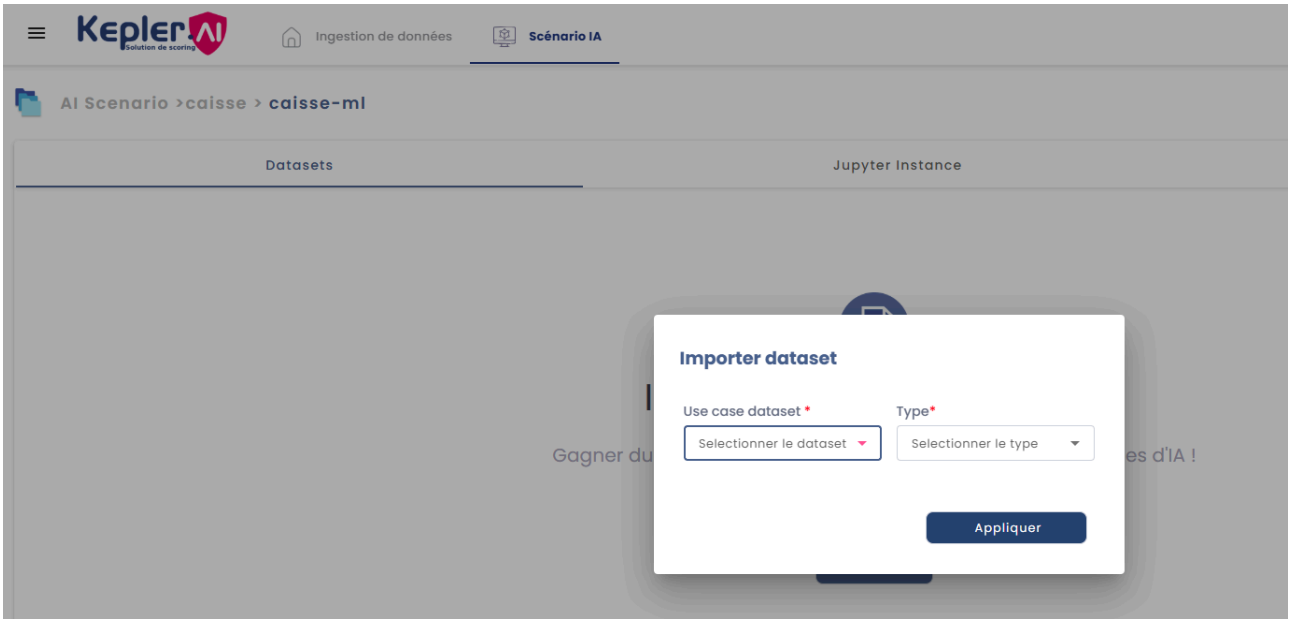
3. **Sélection du Use Case :**

   Dans le champ « **Use Case Dataset** », sélectionnez le Use Case auquel le dataset sera associé.

4. **Choix du type de données :**

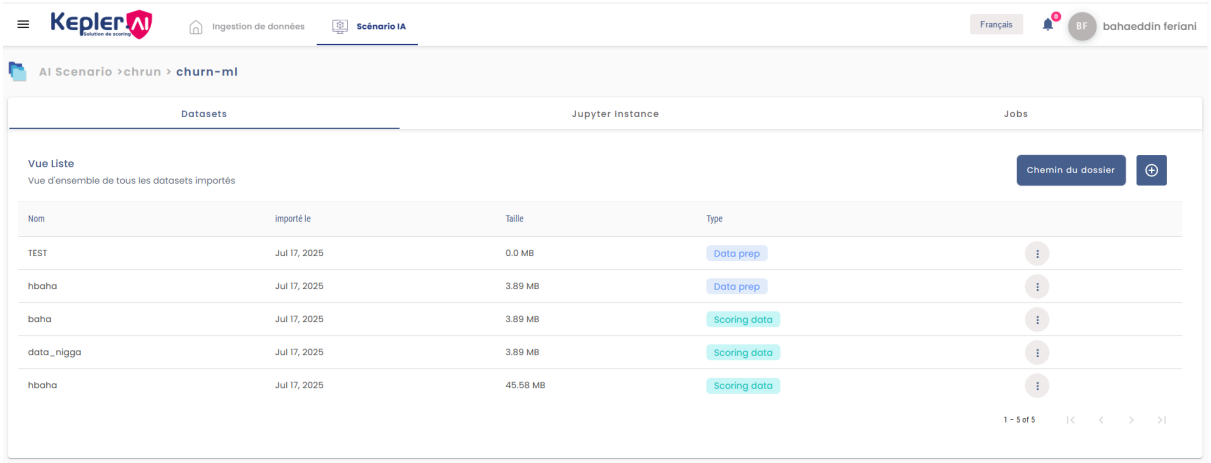
   Dans le champ « **Type** », sélectionnez le format de données ou la méthode d'ingestion (ex. : Fichier CSV, Base de données…).

LAB>Scénario IA>Exemple Scénario>Dataset>Get Started



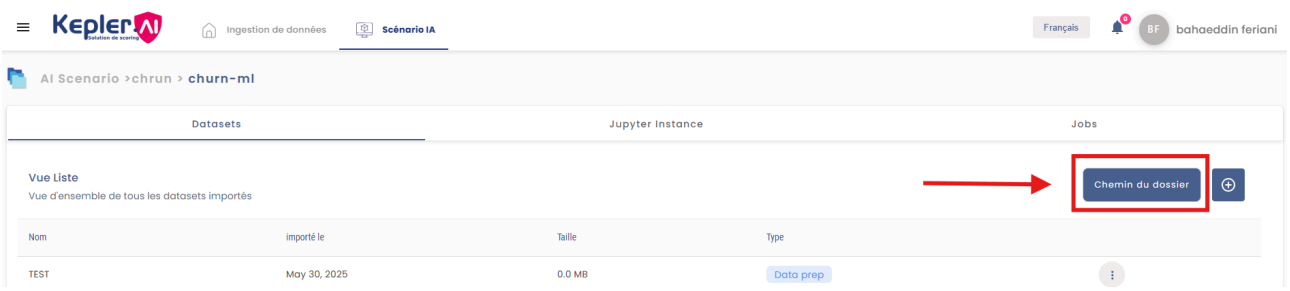
Une fois le dataset importé avec succès, il apparaîtra automatiquement dans la **vue liste**, qui présente l'ensemble des datasets associés au Use Case sélectionné.

LAB>Scénario IA>Exemple Scénario>Dataset



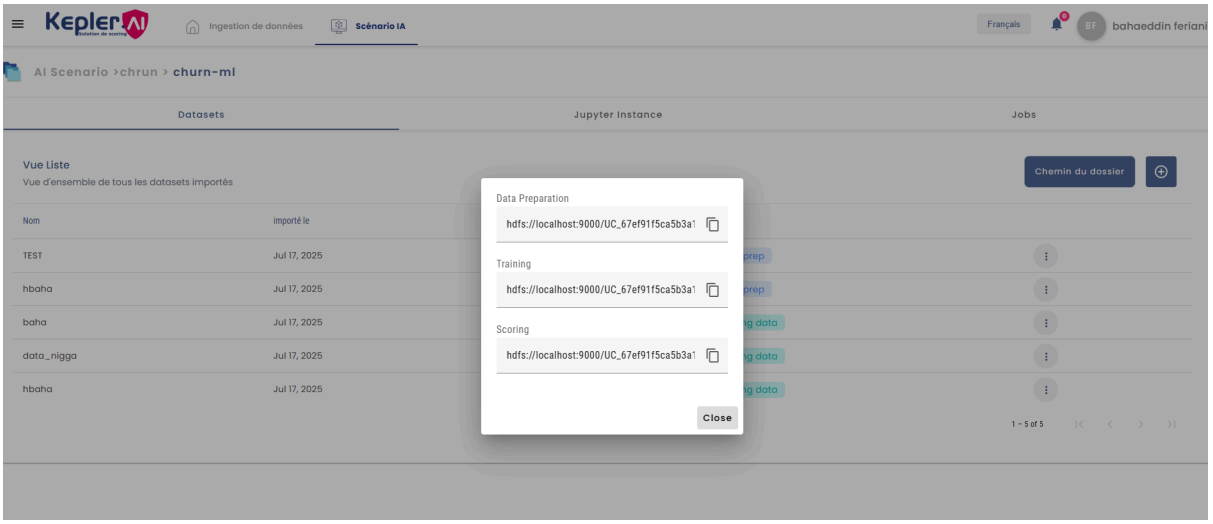
Depuis cette vue, vous pouvez :

LAB>Scénario IA>Exemple Scénario>Dataset



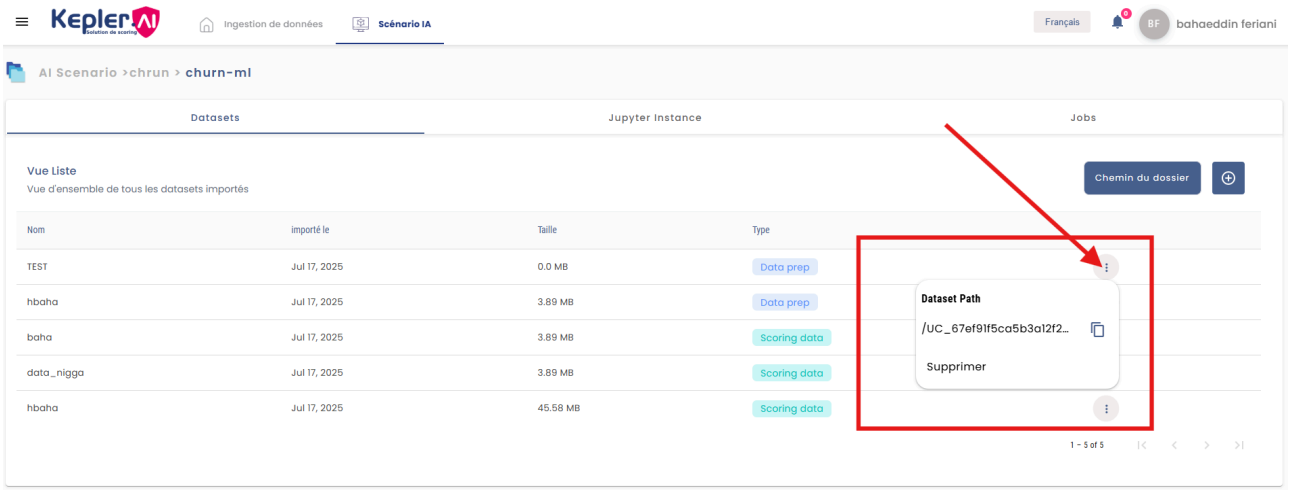
- **Copier le chemin d'accès** du dataset pour l'utiliser dans les différents modules :
    - Data Preparation
    - Training
    - Scoring

LAB>Scénario IA>Exemple Scénario>Dataset>Chemin du dossier



- **Supprimer** un dataset si nécessaire.

LAB>Scénario IA>Exemple Scénario>Dataset>⋮



💡 **Utilisez les chemins de datasets copiés pour intégrer rapidement vos données dans les notebooks Jupyter ou dans les scripts d'ingestion automatisés.**

Par tanitlab AI

Onglet 2 :
Ingestion des données

Par tanitlab AI

DATA INGESTION

Introduction

La phase de **Data Ingestion** est essentielle pour initier tout projet de traitement et d'analyse des données. Elle consiste à importer les données brutes depuis différentes sources (fichiers ou bases de données) vers le système distribué **HDFS**, où elles seront centralisées, organisées et prêtes à être exploitées.

L'ingestion dans la plateforme **KeplerAI** peut se faire selon deux modes :

- **Manuellement via l'interface graphique**
- **Automatiquement via un script planifié (Job Cron)**

Ces deux méthodes s'intègrent dans une démarche DevOps orientée **scalabilité**, **répétabilité** et **automatisation** des pipelines de données.

I. Ingestion manuelle via l'interface Kepler

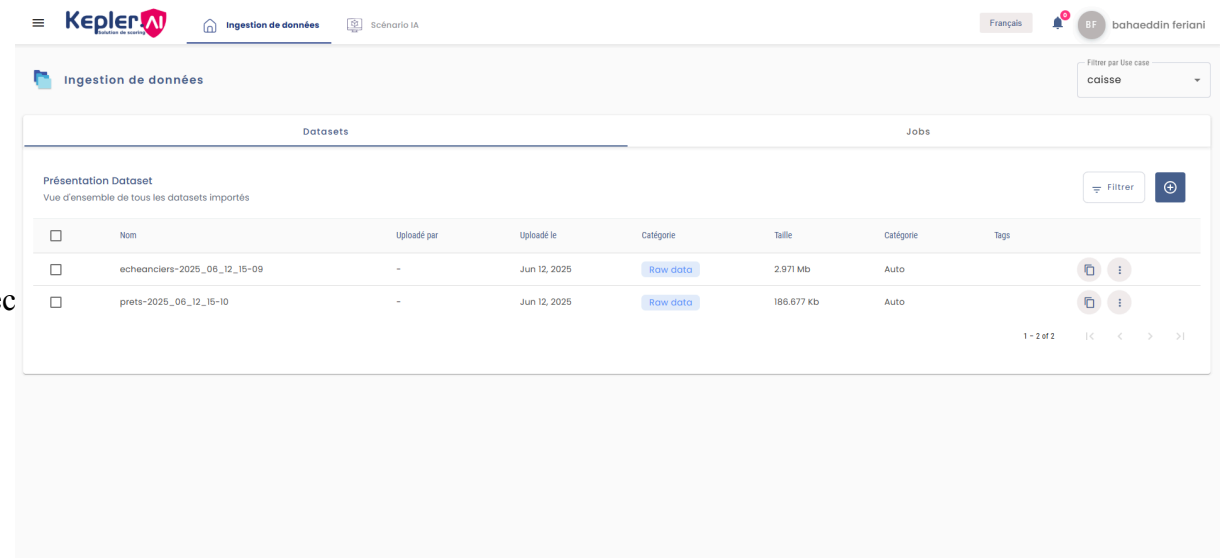
Cette méthode est particulièrement utile pour charger rapidement un dataset ponctuel ou tester un jeu de données avant d'automatiser son traitement.

📍 **Étapes à suivre :**

1. Accès à la plateforme

- Ouvrir **KeplerAI**
- Naviguer vers :
  LAB>Ingestion de données>Datasets

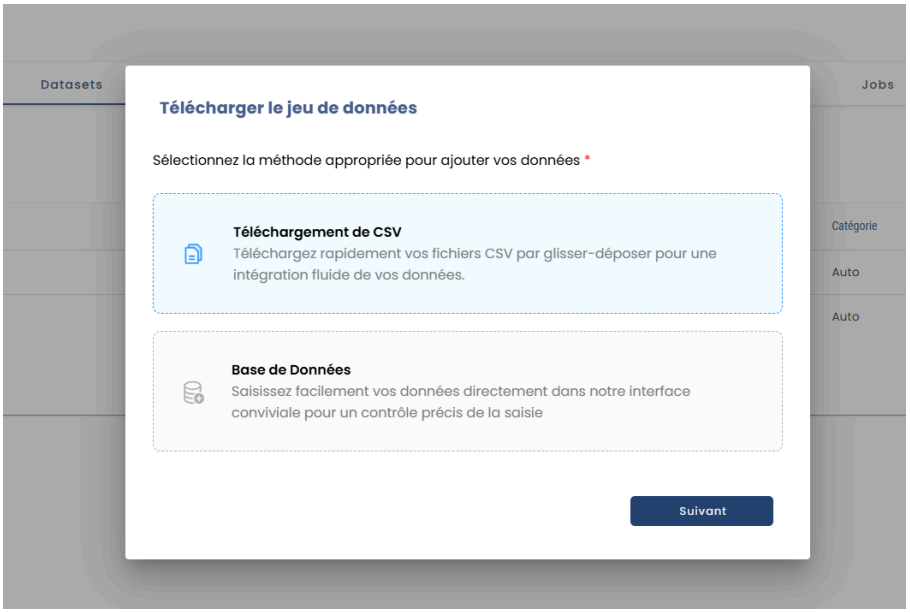
2. Sélec



- Sélectionner le **Use Case** dans lequel le dataset sera utilisé
- Cette étape est importante pour **contextualiser** les données et les **lier aux scénarios métiers**

3. Ajout d'un nouveau Dataset

- Cliquer sur ➕ **Ajouter un dataset**
  LAB>Ingestion de données>Datasets>+



- Choisir le mode d'ingestion :
  - 📁 **Fichier local (CSV, XLSX, JSON)**
  - 🗄️ **Base de données relationnelle (MySQL, PostgreSQL, SQL Server, etc.)**

A. Ingestion depuis un fichier

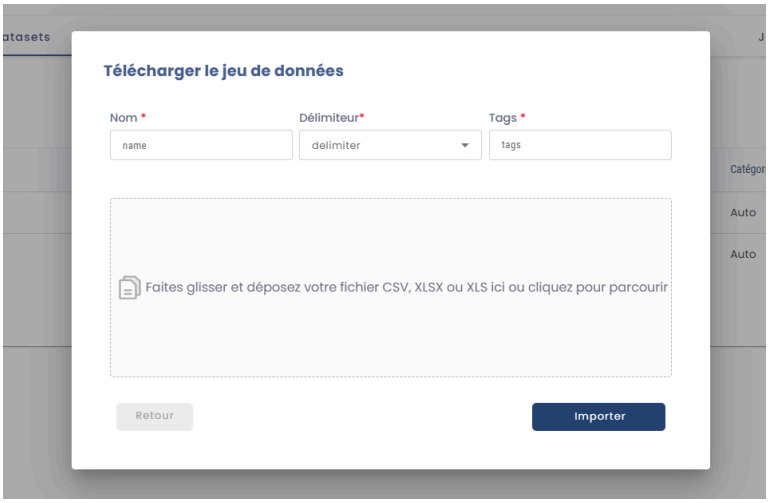
📍 **Étapes :**

1. Renseigner les métadonnées :
   - **Nom du dataset**
   - **Délimiteur** utilisé (, ; ; ; tabulation, etc.)
   - **Tags** pour organiser ou filtrer les datasets

2. **Glisser-déposer** le fichier dans l'interface ou le **sélectionner manuellement**

3. Valider pour importer le dataset

LAB>Ingestion de données>Datasets>+>Téléchargement de CSV



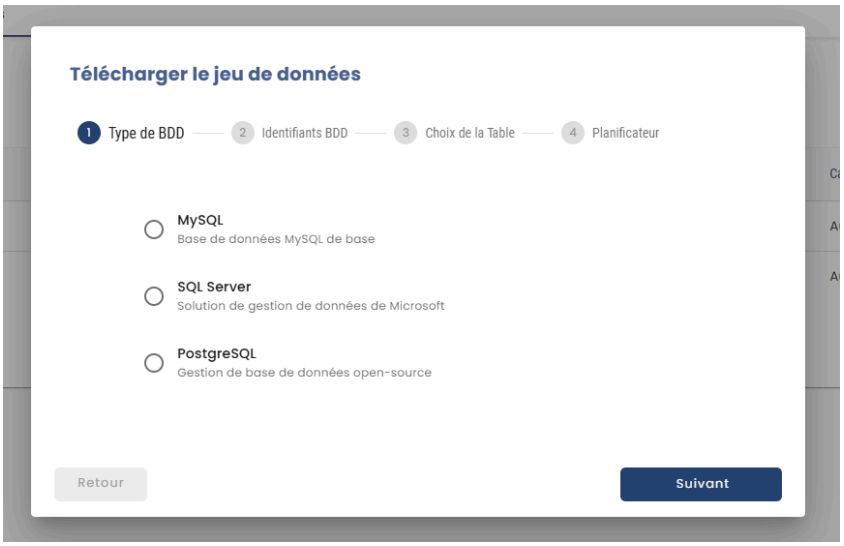
💡Les fichiers importés sont enregistrés dans le **répertoire RAW** du Use Case, pour un usage ultérieur (préparation, visualisation, modélisation…).

B. Ingestion depuis une base de données

📍Étapes :

1. Choisir le type de base de données : MySQL, PostgreSQL, SQL Server, etc.

LAB>Ingestion de données>Datasets>+>Base de Données



2. Fournir les informations de connexion :
    - Adresse IP ou nom d'hôte
    - Port
    - Nom de la base
    - Identifiants d'accès (username / password)
3. Sélectionner la **table source** à importer
4. (Optionnel) Configurer un **planificateur** pour définir une fréquence d'importation automatique

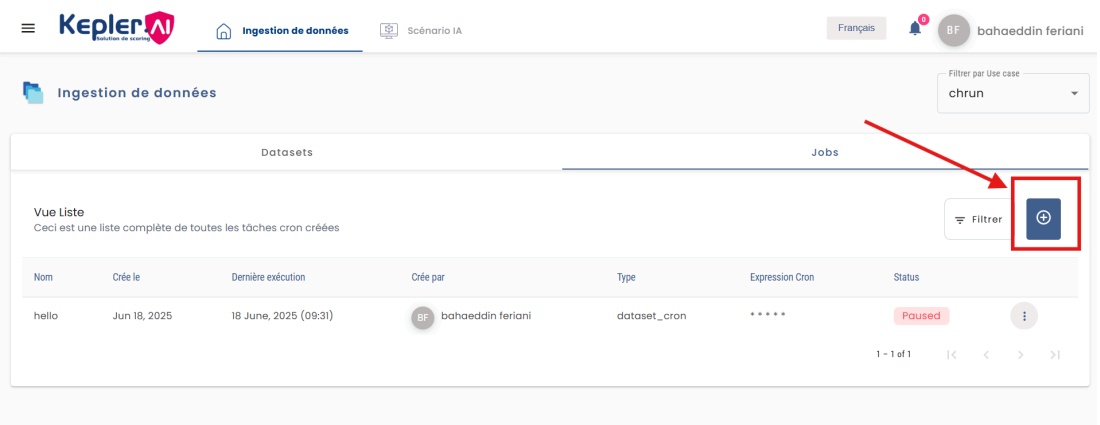
II. Ingestion automatique via Script (Job planifié)

Pour automatiser l'ingestion des données, KeplerAI permet de créer des **Jobs planifiés (Cron Jobs)**. Ces Jobs exécutent périodiquement un **script PySpark** ou **Python**, permettant une mise à jour régulière des données sans intervention manuelle.

📍 **Étapes de création d'un Job :**

1. Accès à la section Jobs

- Naviguer vers : ***LAB > Ingestion de données > Jobs***



- Cliquer sur **"Commencer"** ou sur le bouton ➕ **Ajouter un Job**

2. Sélection du Use Case

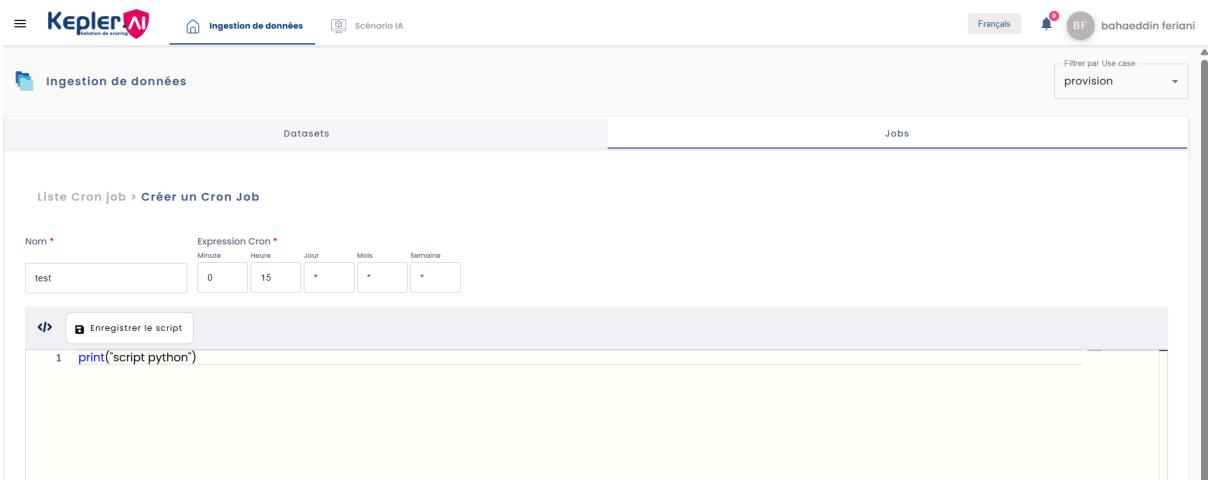
- Associer le Job au Use Case concerné (comme pour les Datasets)

3. Remplir les champs du Job

Champ	Description
Nom du Job	Saisir un nom explicite pour identifier la tâche d'ingestion automatisée.
Expression CRON	Définir la fréquence d'exécution à l'aide d'une expression CRON (Minute, Heure, Jour, Mois, Jour de semaine).
Zone de saisie du script	Un éditeur de texte permet à l'utilisateur d'écrire directement ou de coller un script d'ingestion (Python ou Spark) à exécuter périodiquement.

💡 Le script doit être **validé et testé** dans **Jupyter** avant d'être déployé dans un Job.

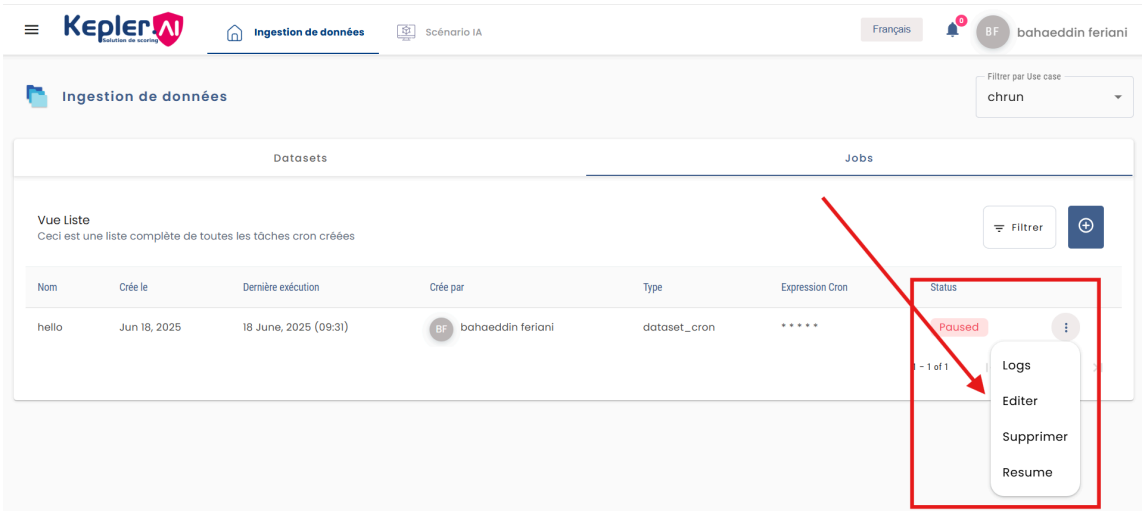
LAB>Ingestion de données>Jobs>Commencer



III. Cycle de Vie du Script

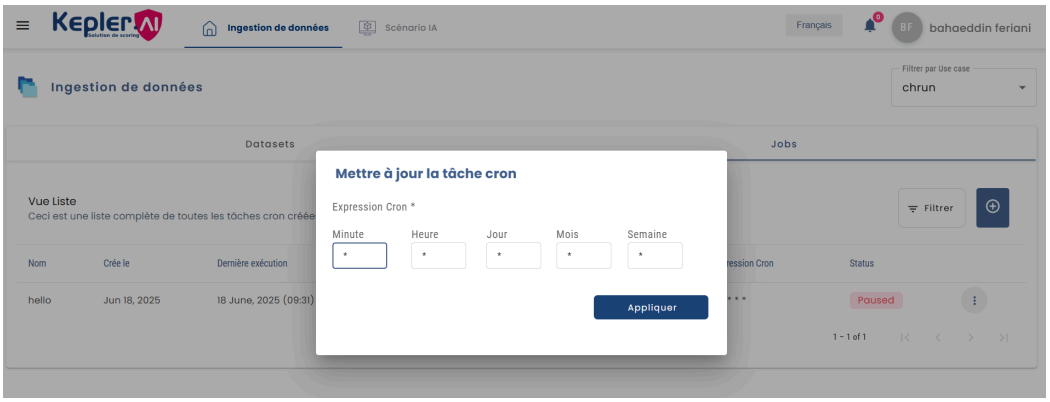
Les scripts d'ingestion automatisés suivent un cycle structuré pour une meilleure gestion :

LAB>Ingestion de données>Jobs



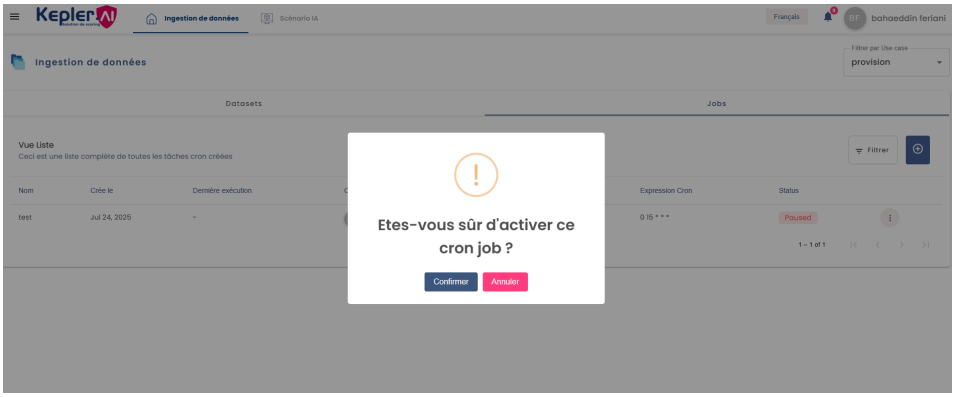
- **Éditer**: Mettre à jour la tâche cron

  LAB>Ingestion de données>Jobs>Editer

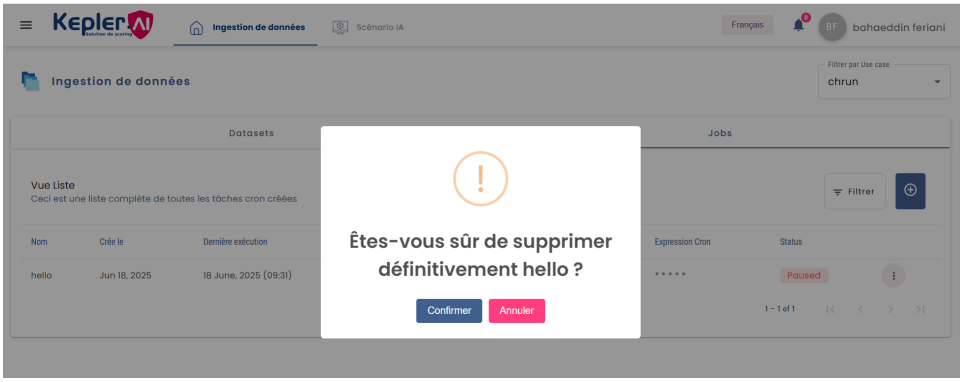


- **Active** : Script validé et actif en production

  LAB>Ingestion de données>Jobs>Résume



- **Paused** : Script temporairement désactivé
- **Deleted** : Script supprimé ou archivé



Cette organisation permet une **traçabilité complète** et une **gestion fluide** des pipelines de données.

IV. Avantages d'une ingestion centralisée dans Kepler

- **Gain de temps** grâce à l'automatisation
- **Réutilisabilité** des datasets dans plusieurs scénarios
- **Sécurisation** des accès via les identifiants paramétrés
- **Intégration native** dans les processus analytiques et IA

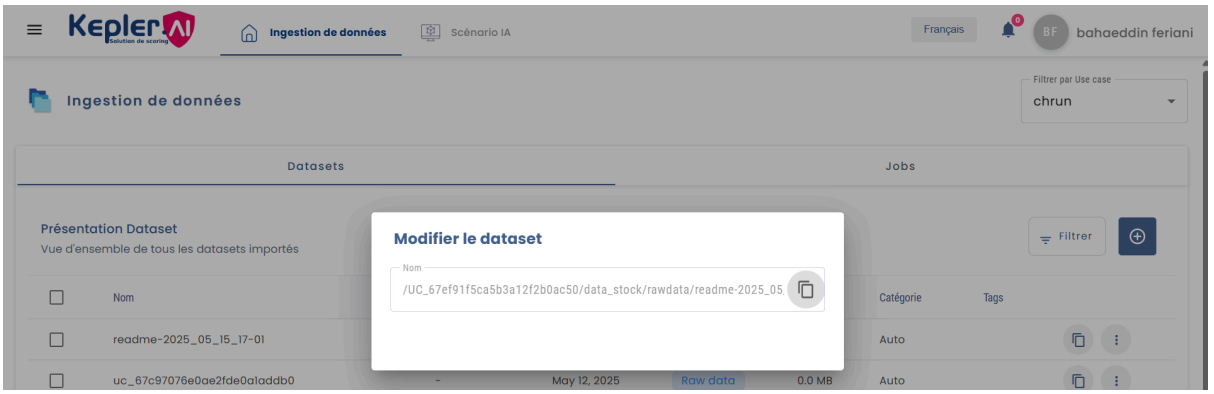
🗂️ **Structuration** des données dans des répertoires HDFS clairs :

bash

CopierModifier

/UC_<ID>/data_stock/raw_data/table_<NomTable>

- ***LAB>Ingestion de données>Datasets>icon Couper***



Cela permet de **centraliser les données brutes** de chaque Use Case tout en conservant une organisation claire et accessible pour les étapes suivantes (préparation, entraînement, scoring...).

V. Extraction des Données

Une fois les données ingérées dans HDFS, la phase d'extraction permet de les **charger sous forme de DataFrame Spark**, favorisant la performance et la résilience à grande échelle.

Deux **modes d'extraction** sont disponibles, en fonction des besoins métiers :

1. Mode append (incrémental)

- Filtre automatiquement les enregistrements de la journée précédente (**J-1**) via une colonne **timestamp** ou **date**
- Ajoute uniquement les **nouvelles données**, évitant la duplication
- Idéal pour les flux de données quotidiens

2. Mode overwrite (écrasement complet)

- Réécrit **toutes les données** extraites
- Supprime les fichiers existants dans le répertoire HDFS cible
- Recommandé pour les cas nécessitant une **reconstruction totale** de la base pour garantir sa **cohérence**

Conclusion

La **Data Ingestion** dans KeplerAI offre une solution complète, modulable et efficace pour importer des données depuis différentes sources, manuellement ou automatiquement. Elle s'inscrit dans une approche DevOps moderne, garantissant une disponibilité continue des données, prêtes à être exploitées dans les processus d'analyse, de modélisation ou de scoring.

Par tanitlab AI

Onglet 3 : Préparation des données

Par tanitlab AI

DATA PREPARATION

Introduction

Une fois le Use Case et le Scénario IA crées dans KEPLER, le Data Scientist peut initier l'étape de **préparation des données**, indispensable pour assurer la qualité et la pertinence des informations exploitées par les algorithmes d'intelligence artificielle.

Cette étape est divisée en deux grandes phases :

- La phase **de développement de l'algorithme** dans l'environnement Jupyter.
- La phase **d'automatisation** via les Cron Jobs dans l'interface LAB.

I. Étapes de préparation dans Jupyter

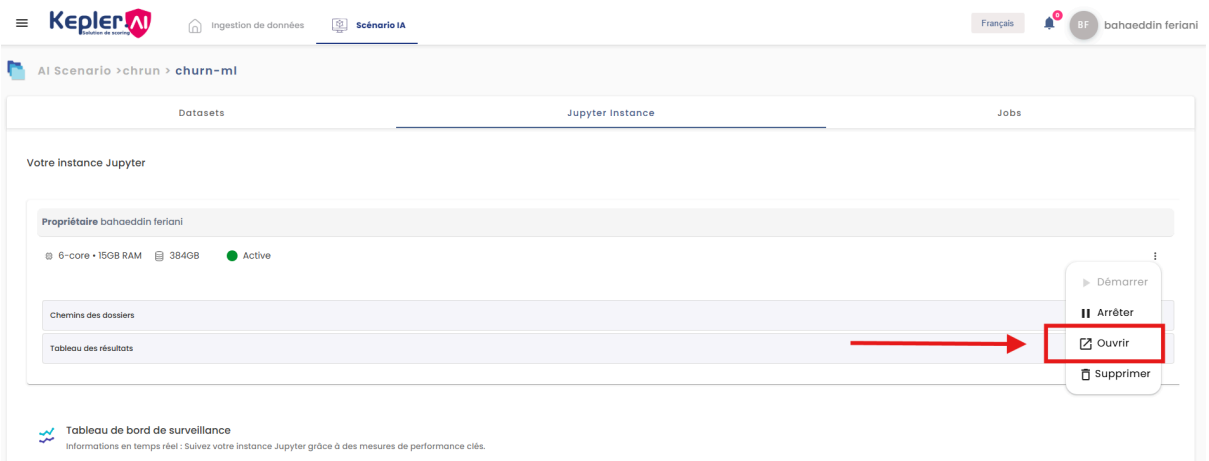
1. Accès à l'environnement

Le Data Scientist accède à l'environnement Jupyter via l'interface KEPLER en suivant les étapes suivantes :

1. Depuis la page d'accueil (**Home**), accéder au module **LAB** ;
2. Cliquer sur **Scénario IA** dans la barre de navigation latérale ;
3. Sélectionner un scénario existant dans la **liste des scénarios IA** ;
4. Dans la section **Jupyter Instance** :

    - Si l'instance est affichée comme **inactive**, cliquer sur **Activer** ;
    - Une fois activée, cliquer sur **Ouvrir** pour lancer l'environnement Jupyter.

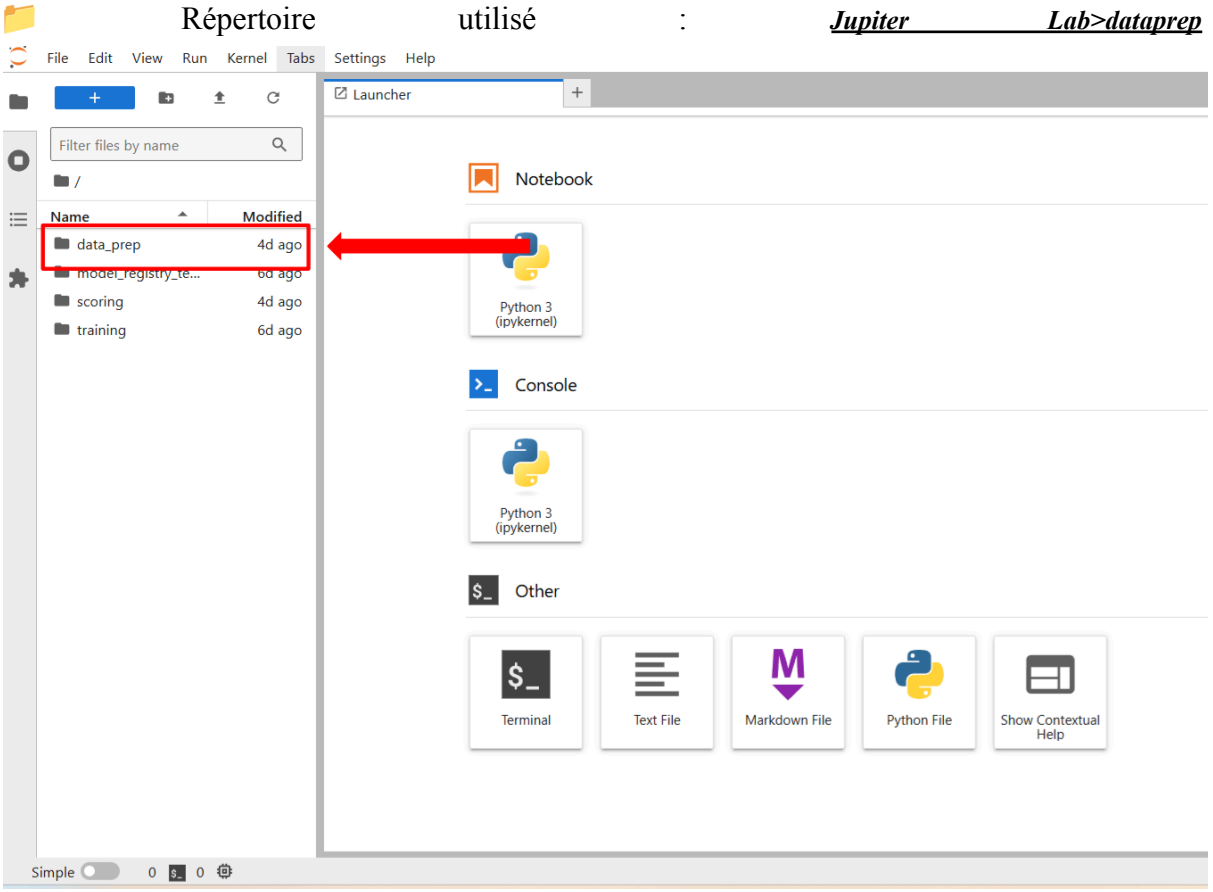
LAB>Scénario IA>Exemple Scénario>Jupyter Instance



Une fois dans l'instance Jupyter, l'utilisateur retrouve une arborescence de dossiers contenant trois répertoires principaux :

- dataprep
- training
- scoring

Répertoire utilisé : ***Jupiter Lab>dataprep***



Ces dossiers sont **vides par défaut** et servent à **organiser les fichiers** nécessaires au pipeline IA.

2. Développement des notebooks

Le Data Scientist commence son travail de préparation de données dans des fichiers **.ipynb (Jupyter Notebooks)**. Ces notebooks sont considérés comme des **brouillons** pour concevoir, tester et ajuster les traitements.

📍Étapes typiques réalisées dans les notebooks :

- Chargement des données brutes
- Nettoyage initial : suppression des doublons, filtrage des anomalies
- Jointure de sources multiples
- Application de mappings ou enrichissements depuis des référentiels
- Sélection des colonnes pertinentes
- Génération d'une **première table d'historique**, qui sera visible dans la **vue Monitoring**

3. Visualisation et vérification

Les résultats obtenus sont **vérifiés manuellement** dans l'interface Jupyter.
L'interface KEPLER propose automatiquement les **chemins HDFS** des datasets, ce qui facilite l'accès, la lecture et l'extraction des données.

✅ Cette phase garantit la **fiabilité** des traitements avant leur automatisation.

II. Automatisation avec Cron Job

1. Transformation du notebook en script exécutable

Une fois les traitements validés, le Data Scientist :

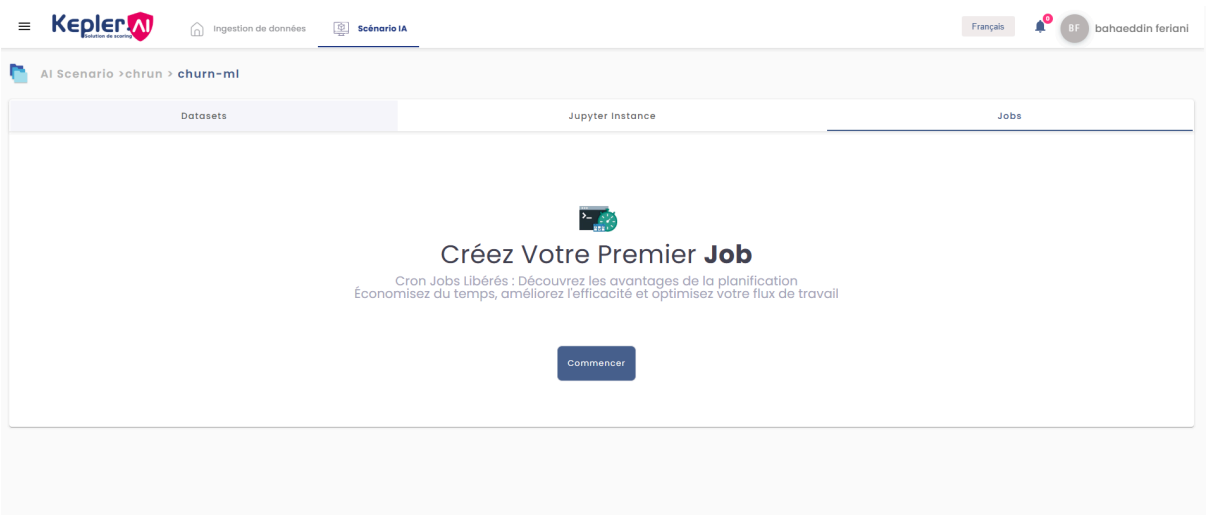
- Regroupe les fonctions essentielles dans un **fichier .py**
- Place ce fichier dans le répertoire approprié (dataprep, scoring, etc.)
- Prépare ce fichier à être exécuté automatiquement via un Cron Job

⚠️ Les Cron Jobs **n'acceptent que les fichiers Python** (.py) – les fichiers .ipynb (Jupyter) ne sont pas compatibles.

2. Création du Cron Job

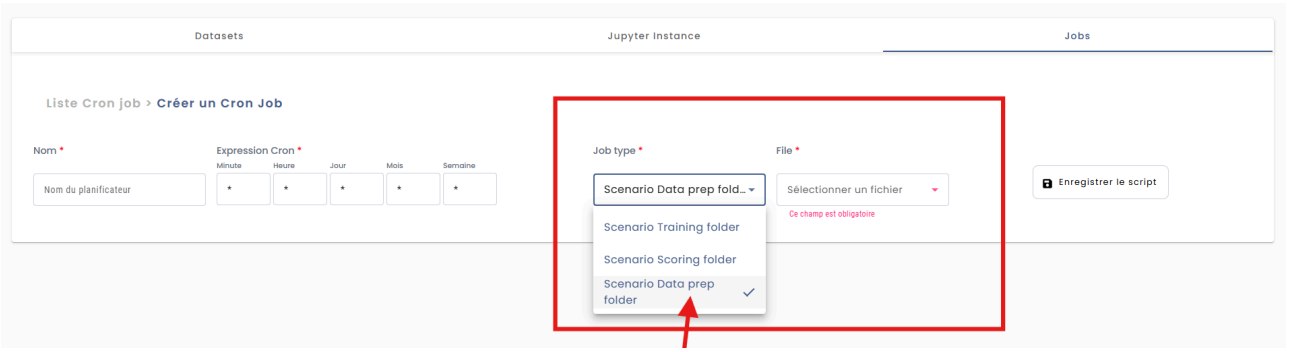
Kepler propose un système de planification de traitements :

1. Depuis la fenêtre **Jobs**, Cliquer sur le bouton **"+"** ou sur **"Commencer"**
2. Dans l'interface de création de Job, le Data Scientist renseigne :



3. Paramètres à définir

LAB>Scénario IA>Exemple Scénario>Jobs



- **Nom du Job**
- **Expression CRON*** : heure ou fréquence d'exécution du script
- **Type de Job*** :
    a. Scenario Training folder
    b. Scenario Scoring folder
    c. Scenario Data Prep folder
- **fichier utiliser .py** : script final à exécuter automatiquement
- Le **Job est automatiquement associé au Use Case actif**, identifié par son ID unique au format UC_<ID>.

4. Enregistrement

Une fois le Job enregistré, il s'exécutera **automatiquement** selon la planification définie.

Les données nettoyées sont :

- Enregistrées dans un **nouveau chemin HDFS**, défini dans le script
- Prêtes à être exploitées dans les étapes suivantes : `training`, `scoring` ou visualisation

✅ Cette automatisation permet une **mise à jour continue des jeux de données** sans intervention manuelle.

Conclusion

L'étape de préparation des données dans KEPLER repose sur une approche rigoureuse et structurée, combinant :

- Une **phase exploratoire manuelle** dans Jupyter pour développer et tester les traitements

- Une **phase automatisée** via Cron Job pour assurer la régularité et la pérennité du pipeline IA

Cette démarche garantit des données **fiables, traçables et continuellement mises à jour**, essentielles au bon fonctionnement des algorithmes d'apprentissage automatique.

Par tanitlab

Onglet 4: Entraînement des données

DATA TRAINING

Introduction

Une fois les données **préparées et nettoyées** via le Job de **Data Preparation**, le **Data Scientist** peut passer à l'étape suivante : **l'entraînement du modèle d'intelligence artificielle**.

Cette étape consiste à utiliser les données transformées pour construire un modèle prédictif ou de classification à l'aide d'algorithmes de machine learning.

I. Phase manuelle dans Jupyter

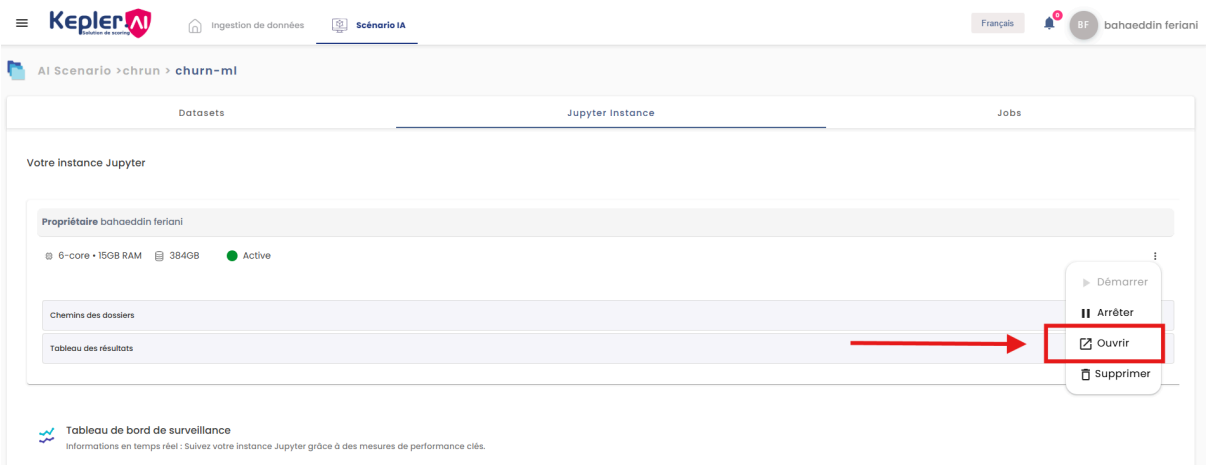
Avant toute automatisation, le Data Scientist effectue ses tests et développements dans l'environnement **Jupyter Lab**, comme pour la phase de préparation des données. Cette étape s'effectue manuellement, sans intervention de Cron Jobs, et permet de développer, tester et valider les notebooks liés à l'entraînement du modèle.

1. Accès à l'environnement Jupyter via KEPLER :

Le Data Scientist accède à l'environnement Jupyter en suivant les étapes suivantes depuis l'interface KEPLER :

1. Depuis la page d'accueil (**Home**), accéder au module **LAB** ;
2. Dans la barre de navigation latérale, cliquer sur **Scénario IA** ;
3. Sélectionner un **scénario existant** dans la liste ;
4. Dans la section **Jupyter Instance** :
    - Si l'instance est **inactive**, cliquer sur **Activer** ;
    - Une fois l'instance activée, cliquer sur **Ouvrir** pour lancer Jupyter Lab.

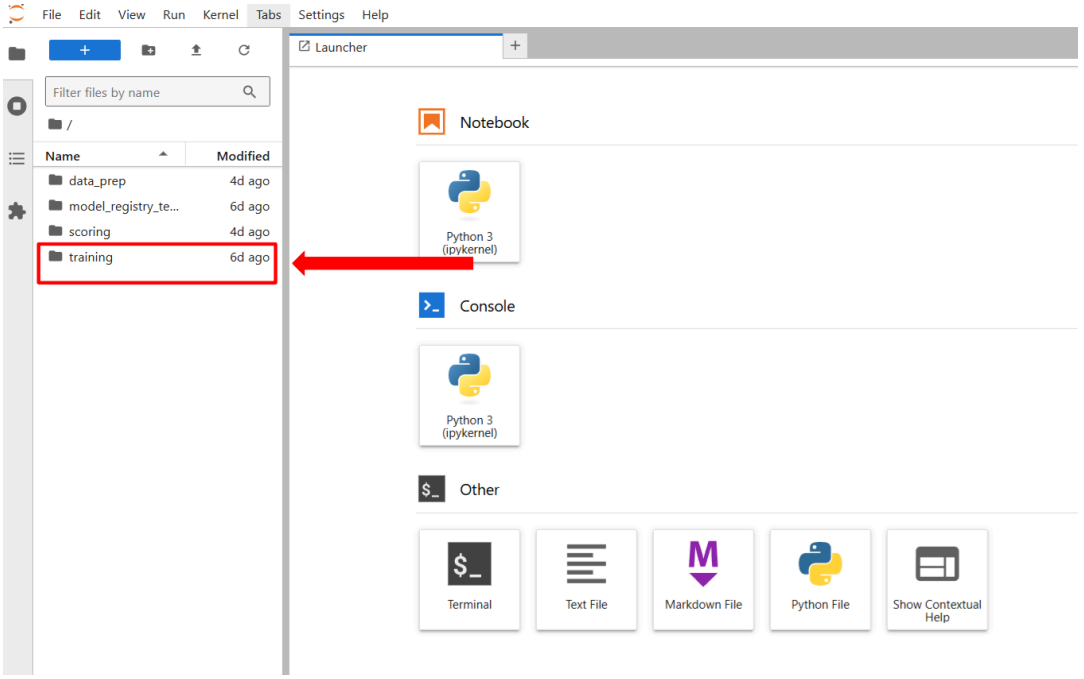
LAB>Scénario IA>Exemple Scénario>Jupyter Instance



2. Travail dans le répertoire training

Une fois dans Jupyter, le Data Scientist se rend dans le dossier training du scénario, où il crée un ou plusieurs fichiers .ipynb pour expérimenter et entraîner son modèle.

Répertoire utilisé : ***Jupiter Lab>training***



📍Étapes typiques dans le notebook de training :

- Chargement des données nettoyées depuis le **chemin HDFS** généré dans la phase précédente (dataprep)

- **Feature engineering** : sélection, transformation, encodage ou normalisation des variables
- **Séparation du dataset** en ensembles **train/test** ou utilisation de **cross-validation**
- Entraînement du modèle (ex. : XGBoost, Random Forest, SVM, etc.)
- **Évaluation du modèle** via des métriques : précision, recall, AUC, RMSE…
- Sauvegarde du modèle entraîné dans un répertoire HDFS (/model ou /scenario_training_output)

> 📌 À ce stade, cette phase est **totalement manuelle et indépendante des Cron Jobs**. Elle ne s'exécute **qu'une seule fois**, sauf si le modèle doit être réentraîné volontairement.

II. Création d'un Job d'entraînement automatisé

Dans certains cas, notamment lorsque les données évoluent fréquemment, il peut être pertinent d'**automatiser** l'entraînement via un **Cron Job**.

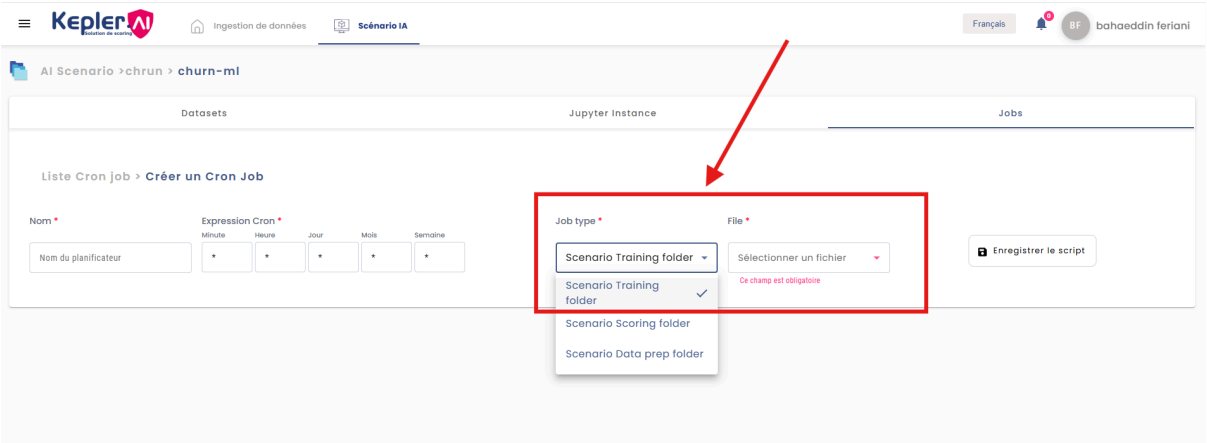
> ⚠️ Cependant, **l'entraînement du modèle n'est pas prévu pour être exécuté régulièrement** comme le Data Prep ou le Scoring. Il est généralement **lancé une seule fois**, ou manuellement à la demande.

1. Création du Job dans KEPLER

Une fois le script de training converti depuis le notebook .ipynb vers un **fichier .py**, le Job peut être créé dans l'interface :

Accès :

LAB > Scénario IA > Exemple Scénario > Jobs



🛠️ Paramètres du Job de Training :

- **Nom du Job** : ex. Training_Model_XGB_V1
- **Expression Cron** *(optionnelle)* : uniquement si l'entraînement doit être relancé régulièrement
- **Job Type** : Scenario Training Folder
- **Fichier .py** : le script final d'entraînement du modèle

📌 Le fichier doit être développé au préalable dans Jupyter, puis **sélectionné manuellement** lors de la création du Job.

III. Objectifs de l'étape de Training

L'objectif de cette étape est de produire un modèle IA performant à partir des données traitées.

Les principales actions sont :

- **Utiliser les données de qualité** issues du Data Prep
- Construire un modèle fiable et robuste
- **Évaluer sa performance** sur des jeux de données de test
- **Enregistrer le modèle** dans HDFS pour l'étape suivante (scoring)

- **Analyser les résultats et logs** depuis l'interface monitoring

IV. Visualisation et ajustements

Une fois le Job exécuté, le Data Scientist peut :

- Consulter les **logs d'exécution** dans KEPLER
- Suivre les **performances du modèle** (courbes, scores)
- Revenir dans Jupyter pour **ajuster son code** si nécessaire
- Reprogrammer un réentraînement si les résultats sont insatisfaisants

Conclusion

L'entraînement du modèle est une **étape clé du pipeline IA**, visant à transformer des données structurées en **modèles prédictifs exploitables**.

Contrairement à la Data Preparation ou au Scoring, le **Job de Training est rarement automatisé**, car il repose sur une phase de développement manuel et d'ajustement important dans Jupyter.

📌Le modèle entraîné sera ensuite utilisé dans la phase **Scoring**, qui peut, elle, être automatisée pour des évaluations quotidiennes ou périodiques.

Par tanitlab AI

Onglet 5:
Scorage des données

DATA SCORING

Introduction

Après avoir entraîné et validé le modèle dans la phase précédente, le Data Scientist passe à l'étape finale du pipeline IA : le **scoring**.

Cette étape consiste à appliquer le modèle sauvegardé sur de nouvelles données afin de générer des prédictions exploitables (scores, classes, probabilités...).

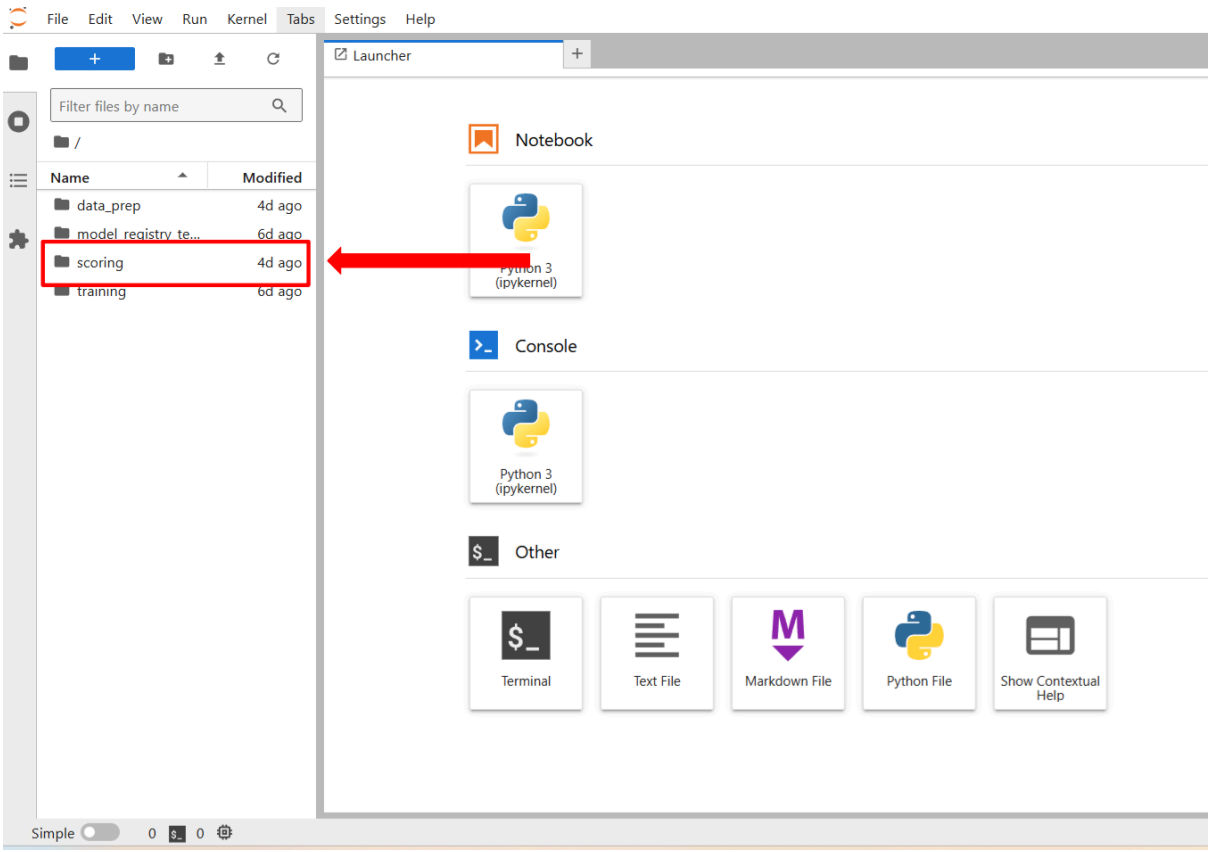
Elle s'effectue en deux temps :

- D'abord, en phase de développement manuel dans **Jupyter**, pour tester et valider le processus.
- Ensuite, via un **Cron Job** automatisé dans l'interface **LAB** pour une exécution régulière en production.

I. Phase manuelle dans Jupyter

Comme pour la préparation des données et le training, le Data Scientist commence dans **Jupyter Lab** pour développer et tester son code de scoring.

Répertoire utilisé :***Jupyter Lab > scoring***



📍 Étapes typiques dans le notebook de scoring :

- Chargement du modèle entraîné (depuis /model ou un dossier scenario_training_output)
- Chargement des nouvelles données à scorer (via un chemin HDFS ou table métier)
- Application du modèle sur ces données
- Post-traitements éventuels : normalisation des scores, ajout de tags métier, enrichissement
- Sauvegarde des résultats dans un dossier HDFS (souvent /scenario_scoring_output)

📌 **Cette phase reste manuelle** et permet de vérifier que le scoring se déroule correctement avant toute automatisation.

II. Automatisation avec un Cron Job

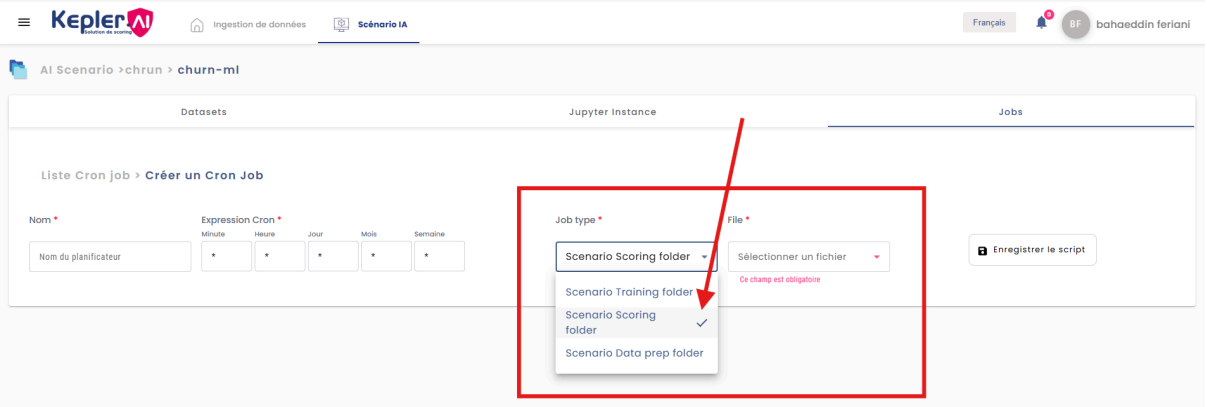
Une fois le notebook de scoring validé, il est converti en fichier **.py**, en respectant les contraintes suivantes :

- Le script doit être en **Python** (⚠️ PySpark non supporté dans les Cron Jobs)
- Le fichier doit être enregistré dans le dossier scoring de l'arborescence

Le Data Scientist peut alors automatiser l'exécution via l'interface **Jobs** dans KEPLER.

Accès :

LAB > Scénario IA > Exemple Scénario > Jobs



🛠️**Paramètres du Job de Scoring :**

- **Nom du Job** : ex. Scoring_V1_Risk_Client
- **Job Type** : Scenario Scoring Folder
- **Type de script** : Python (obligatoire)
- **Expression Cron** : définit la fréquence d'exécution (ex : 0 18 * * * → tous les jours à 18h)
- **Fichier .py** : script de scoring préalablement développé dans Jupyter

III. Objectifs de l'étape de Scoring

L'étape de scoring constitue la **phase finale du pipeline IA**. Elle a pour vocation d'**exploiter le modèle entraîné** afin de produire, de manière régulière et automatisée, des **prédictions exploitables** à partir des données actualisées.

Au-delà de l'application du modèle, cette étape vise à **structurer et déployer les résultats** dans une architecture technique adaptée, garantissant leur **consommabilité** par les outils de supervision, de décision et d'analyse.

⚙️ Résultats attendus et actions associées :

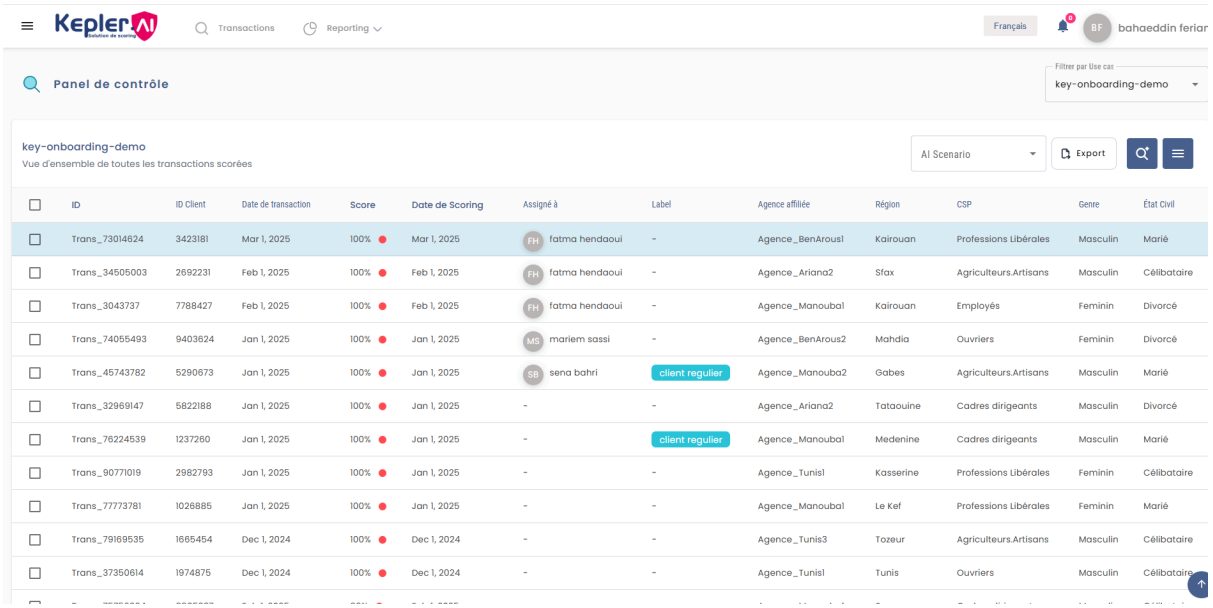
- **Rechargement automatique du modèle** validé (préalablement sauvegardé dans le dossier scenario_training_output).
- **Application du modèle** sur les nouvelles données issues d'une source HDFS ou métier.
- **Calcul des prédictions** : scores, probabilités ou classes, selon le cas d'usage.
- **Post-traitements métier** : normalisation, enrichissement sémantique, filtrage, tagging ou catégorisation.
- **Enregistrement structuré des résultats** dans deux espaces complémentaires :
    - Une **table HDFS**, contenant l'intégralité des colonnes calculées et des métadonnées analytiques.
    - Une **table PostgreSQL**, configurée avec des **colonnes personnalisées** pour un affichage optimisé dans les dashboards du **module Monitoring de KeplerAI**.

🔗 Intégration dans le Monitoring de KeplerAI

Dans l'interface **Monitoring** de **KeplerAI** :

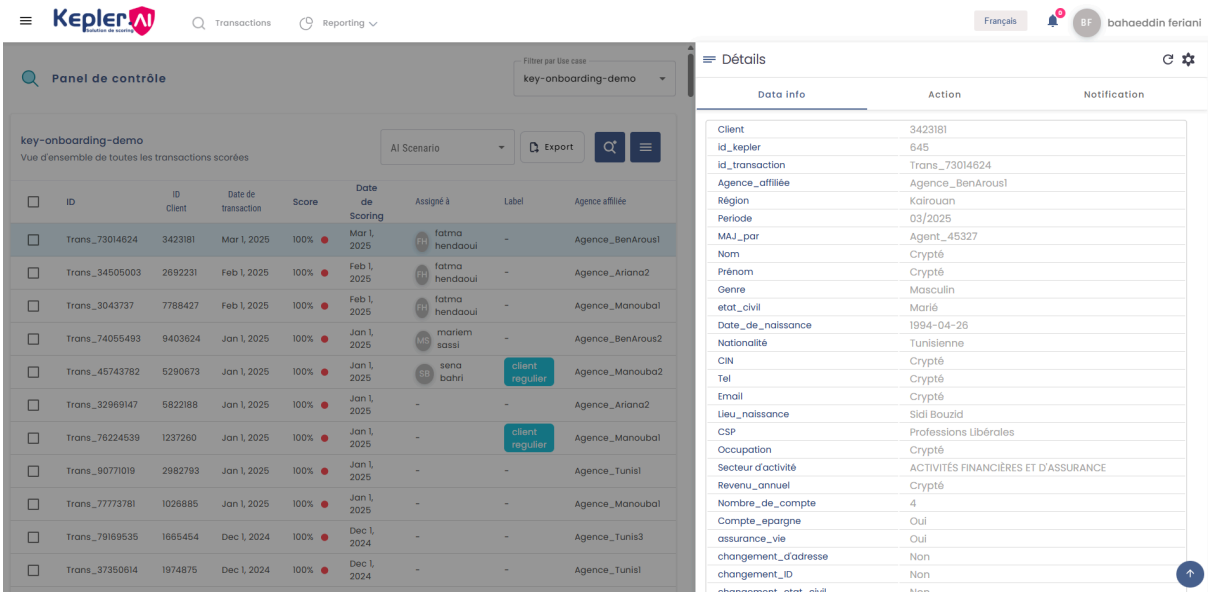
- Le tableau principal est **alimenté par la table PostgreSQL**, sélectionnée pour sa légèreté et sa pertinence visuelle.

Modules exemple "onboarding KYC">Panel de contrôle



- Chaque ligne du tableau correspond à une transaction ou prédiction.
- Lors d'un clic sur une ligne, un **popup contextuel** s'ouvre automatiquement. Celui-ci effectue une **jointure dynamique via l'identifiant id_kepler** pour **interroger la table HDFS** et afficher tous les détails complémentaires (features, scores, Id...).

Modules exemple "onboarding KYC">Panel de contrôle>Détails



🚀 Déploiement final

Le déploiement du scoring ne se limite pas à l'exécution du script Python. Il implique une **orchestration complète des livrables** :

- Génération de la **table HDFS complète** à partir des résultats du scoring.
- Création de la **table PostgreSQL allégée**, dédiée à l'affichage dans les interfaces utilisateur.
- Mise en place de la **liaison automatique** entre les deux tables via id_kepler.
- Vérification du bon fonctionnement de l'interface de **Monitoring** (tableau + pop-up).
- Intégration dans les workflows d'analyse et de prise de décision.

✅Cette approche garantit une **traçabilité complète**, une **consultation fluide** des prédictions, et une **intégration robuste** dans les outils métiers existants.

IV. Enregistrement et automatisation

Une fois configuré, le Job s'exécute automatiquement à la fréquence définie. Il devient ainsi un **maillon clé de la chaîne IA en production**, assurant un **suivi en temps réel** ou régulier des phénomènes à surveiller (risques, anomalies, comportements clients...).

📌 **Tous les Jobs automatisés** sont visibles et traçables depuis l'interface KEPLER. Les logs d'exécution et les erreurs éventuelles peuvent y être consultés pour faciliter les ajustements.

Conclusion

Le scoring est l'étape finale et concrète du pipeline IA. Il permet d'appliquer les modèles entraînés sur de nouvelles données pour générer de la **valeur métier**.
Contrairement au **Training**, cette étape est **automatisée par défaut**, car elle repose sur un processus stable, reproductible et destiné à tourner régulièrement.

📌 **Rappel** : seuls les scripts en Python sont pris en charge par les Cron Jobs. Le processus doit donc être testé, converti en .py, puis sélectionné manuellement lors de la création du Job.