

NoSQL, systèmes distribués et passage en production de projets Data

Thierry GAMEIRO MARTINS

Séances

1. Introduction et prise en main d'Onyxia

- 2. Le stockage des données en NoSQL
- 3. Les systèmes de stockage distribués
- 4. Le passage en productions
- 5. Orchestration par Airflow et pratique DevOps
- 6. Déploiement conteneurisé sous Kubernetes
- 7. Architecture Data et MLOps

Modalité d'évaluation

Objectifs

Présenter par groupe (de 4 ou 5 personnes) un POC (*Proof of Concept*) d'une chaîne de traitement de la données comme solution pour un client

- **Présentation des travaux** : exposé de 15 minutes
- **Questions/réponses** : 10 minutes de question individuelles
- **Livrable** : slides détaillant votre solution, à envoyer avant le jour de la présentation

Barème de la présentation

- Présentation du sujet et de la problématique
- Explication des différentes étapes de traitements de la donnée (pré-processing, collecte, valorisation ou d'exposition de la donnée)
- Les briques technologiques utilisées et les raisons de leurs choix
- Comment la solution répond à la problématique
- Forme de la présentation

Sujets

- 6 sujets proposés
- Possibilité de proposer son propre sujet (à valider avant)
- Date limite pour le choix du sujet avant troisième séance

Liste des sujets

- Analyse de tweets <https://tinyurl.com/y5v4j8f6>
- Parsing de données IOT (Airparif) <https://tinyurl.com/y6xdod7p>
- Analyse des données de disponibilité des vélib à Paris <https://tinyurl.com/yykzr6hv>
- Analyse des données de subventions aux associations parisiennes
<https://tinyurl.com/y5be9ynp>
- Analyse et comparaison des trajets uber / Taxi à New York
<https://tinyurl.com/y29k2jco>
- Système de recommandation de films <https://tinyurl.com/v2oynmf>

Prise en main d'Onyxia

Onyxia est une application web développée par l'INSEE qui permet aux data scientists et data ingénieurs de :

- Lancer des services (éditeur de code, base de données, outils d'orchestration, etc.)
- D'explorer des données et d'entraîner des modèles
- Déployer des applications
- Se former en Data Science

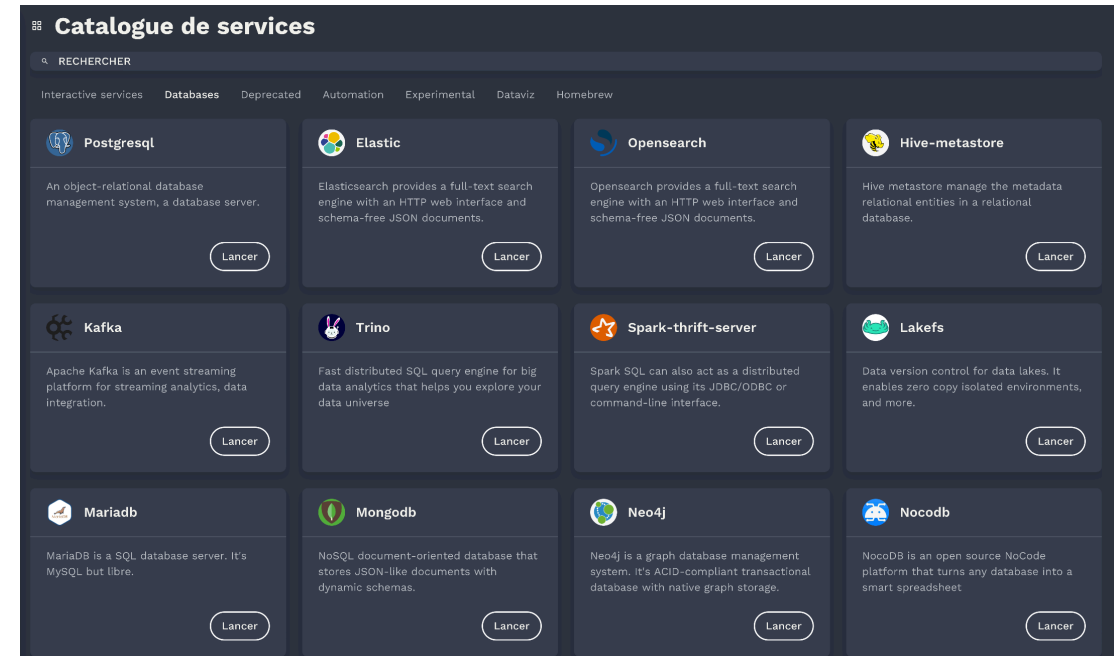
Lien pour se connecter disponible sur :
<https://datalab.sspcloud.fr>



Catalogue de services

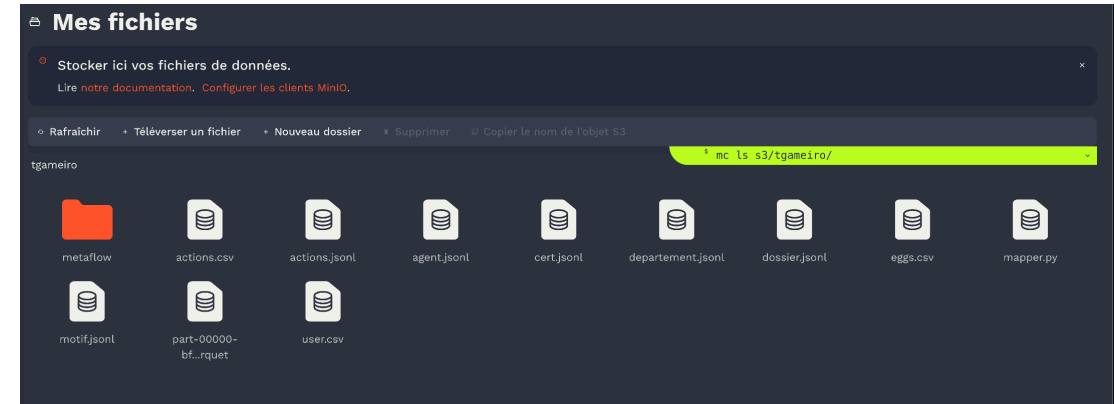
- **Base de données** (MongoDB, Elastic, PostgreSQL, etc.)
- **Outils d'orchestration** (ArgoCD, Argo Workflow, MLFlow etc.)
- **Environnement de développement** (Jupyter, VSCode, RStudio)
- **Visualisation** (Superset, Redash)

Configurable (initscript, ressources, stockage, git, secret, etc.)



Stockage des secrets et fichiers

- Un explorateur S3 est disponible pour envoyer des fichiers ou récupérer ses fichiers depuis le Datalab
- Un outil d'exploration de données pour visualiser directement des fichiers au format `.csv` ou `.parquet`
- Un gestionnaire de secret pour les injecter dans les services
- Gestion des configurations (git, customisations de services, etc.)



<https://static.data.gouv.fr/resources/recensement-de-la-population-fichiers-detail-individus-localises-au-canton-ou-ville-2020-1/20231023-122841/fd-indcvi-2020.parquet>

Colonne

Densité

Télécharger le fichier

| CANTVILLE | NUMMI | ACHLR | AEMMR | AGED | AGER20 | AGEREV | AGEREVQ | ANAI | ANEMR | APAF |
|-----------|-------|-------|-------|------|--------|--------|---------|------|-------|------|
| 0101 | 1 | 3 | 8 | 74 | 79 | 73 | 70 | 1942 | 04 | 0 |
| 0101 | 2 | 4 | 9 | 39 | 39 | 38 | 35 | 1980 | 00 | 2 |
| 0101 | 2 | 4 | 9 | 9 | 10 | 8 | 5 | 2010 | 00 | 3 |
| 0101 | 3 | 2 | 8 | 55 | 54 | 54 | 50 | 1964 | 04 | 0 |
| 0101 | 4 | 6 | 9 | 1 | 2 | 0 | 0 | 2017 | 00 | 3 |
| 0101 | 4 | 6 | 9 | 43 | 54 | 42 | 40 | 1975 | 00 | 1 |
| 0101 | 4 | 6 | 9 | 38 | 39 | 37 | 35 | 1980 | 00 | 2 |
| 0101 | 5 | 6 | 9 | 46 | 54 | 45 | 45 | 1973 | 02 | 2 |
| 0101 | 5 | 6 | 9 | 16 | 17 | 15 | 15 | 2003 | 02 | 3 |
| 0101 | 6 | 4 | 8 | 59 | 64 | 58 | 55 | 1963 | 05 | 0 |
| 0101 | 7 | 6 | 9 | 19 | 19 | 18 | 15 | 1997 | 00 | 3 |
| 0101 | 7 | 6 | 9 | 45 | 54 | 44 | 40 | 1971 | 00 | 2 |
| 0101 | 8 | 4 | 9 | 16 | 17 | 15 | 15 | 2006 | 03 | 3 |
| 0101 | 8 | 4 | 9 | 41 | 54 | 40 | 40 | 1981 | 03 | 2 |
| 0101 | 8 | 4 | 9 | 4 | 5 | 3 | 0 | 2018 | 03 | 3 |

Exercice

1. Se connecter à Onyxia et s'y inscrire avec son mail de l'Université (Paris-Saclay ou Evry)
2. Télécharger les fichiers `.csv` suivant [usagers-2022.csv](#) et [vehicules-2022.csv](#) sur votre poste. Les importer dans l'explorateur de fichier S3 et le consulter via l'explorateur de données
3. Lancer un service `jupyter-python` et changer la configuration `role` en `admin` dans `Kubernetes` et activer `Enable custom service port` dans `Networking`
4. Créer un notebook et récupérer vos données depuis le S3 avec les commandes suivants :
 - i. Lister les fichiers `!mc ls s3/<nom utilisateur>`
 - ii. Télécharger le fichier `!mc cp s3/<nom utilisateur>/<nom du fichier> ./`

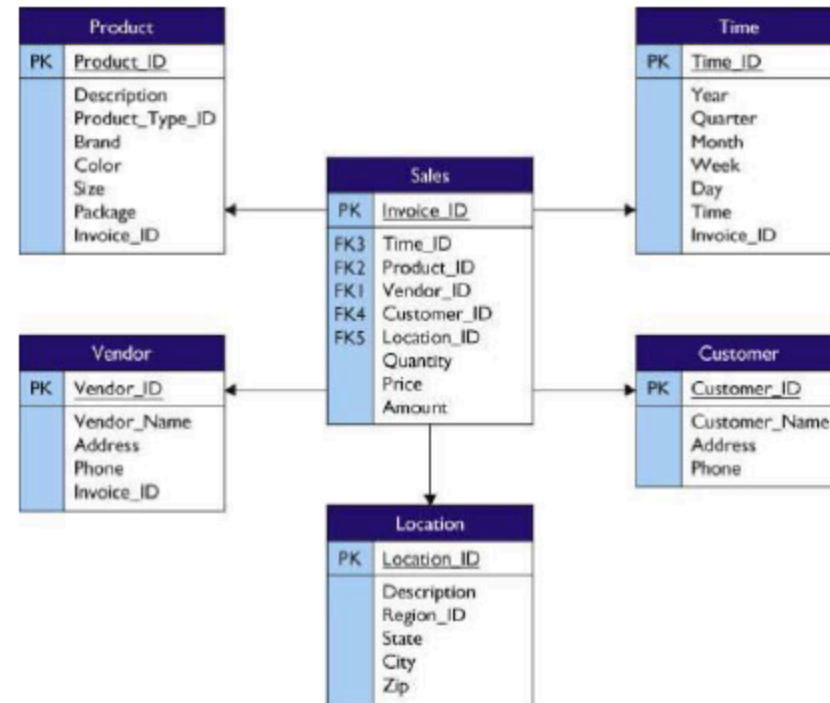
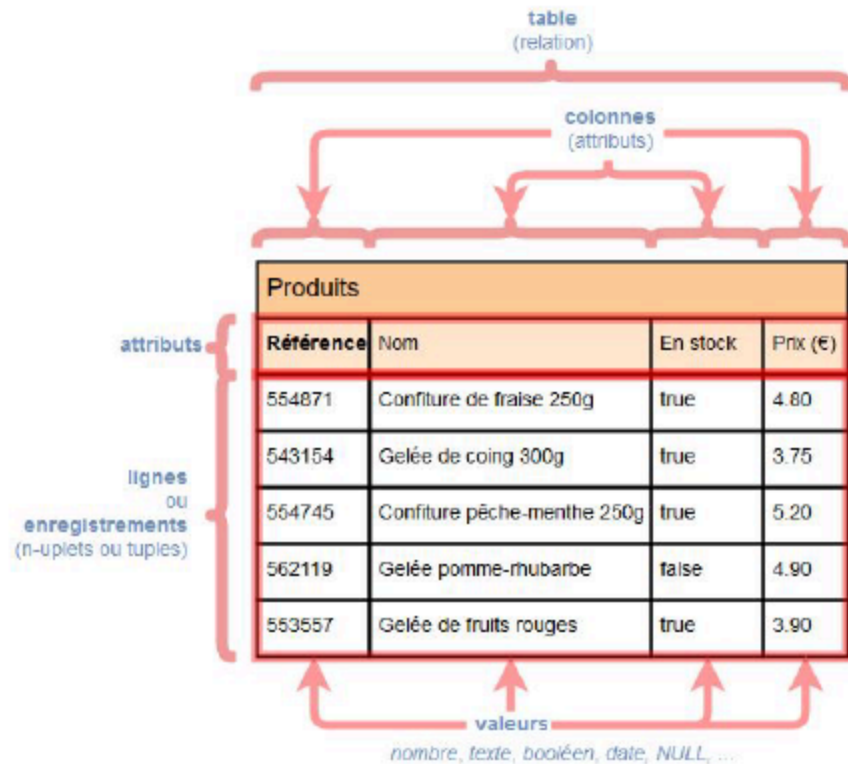
Rappel : les bases de données relationnelles

Les base de données relationnels (SGBDR)

- Logiciel de stockage et de gestion de données régit par des **transactions**
- Organise les données sous la forme d'un **schémas relationnel** visant à éviter la redondance
- Interrogeable par du **SQL** : *Structured Query Langage*



Schéma relationnel



Les contraintes ACID des SGBDR

- **Atomicité** : Une transaction se fait au complet ou pas du tout, sinon remettre les données dans l'état où elles étaient (rollback)
- **Cohérence** : Tout changement doit être valide selon toutes les règles définies en base (contraintes d'intégrité)
- **Isolation** : Toute transaction doit s'exécuter comme si elle était la seule sur le système. Aucune dépendance possible entre les transactions
- **Durabilité** : Lorsqu'une transaction a été confirmée, elle demeure enregistrée

Langage SQL

- **DDL** (Data Definition Language) : CREATE , ALTER , DROP
- **DML** (Data Manipulation Language) : SELECT , INSERT , UPDATE , DELETE , JOIN
- **DCL** (Data Control Language) : GRANT , REVOKE
- **TCL** (Transaction Control Language) : COMMIT , ROLLBACK , SAVEPOINT

Exercice

1. Lancer un service PostgreSQL
2. Se connecter au service jupyter-python et importer les deux csv dans la base de données postgresql
 - i. Dans un notebook, lire les csv avec pandas
 - ii. Installer la librairie python psycopg2 et créer une connexion sqlalchemy.engine.create_engine avec les informations de connexions au format suivant postgresql+psycopg2://user:password@hostname/defaultdb
 - iii. Importer les tables avec la méthode to_csv du dataframe et la connexion sqlalchemy
 - iv. Lancer dans un terminal la commande psql -U postgres -h <hostname> -d defaultdb pour se connecter à la base. Vérifier avec SELECT count(*) FROM usagers que les données sont bien présentes