

# Data engineer

---

# Algorithmique et programmation structurée orientée Python

# Objectifs pédagogiques :

**Lien avec le cursus** : Base incontournable pour tous les modules manipulant du code Python (POO, Spark, Machine Learning, CI/CD)

- Comprendre la logique d'un programme structuré
- Maîtriser les instructions de base : variables, conditions, boucles, fonctions
- Développer de premiers scripts Python réutilisables et testables

# Importance pour le cursus Data Engineer :

Le **Data Engineer** doit concevoir et maintenir des traitements automatisés. La maîtrise de **Python** est **indispensable** :

- Pour **écrire** des scripts robustes d'intégration, de transformation et d'analyse
- Pour **automatiser** les flux dans Airflow ou les scripts de test dans CI/CD
- Pour **interfacer** des APIs, manipuler des données JSON, CSV, ou interagir avec des bases

# Python et Python Orienté Objet

# Objectifs pédagogiques :

Passage nécessaire de la logique procédurale à la conception modulaire, orientée projet

- Comprendre les concepts fondamentaux de la **POO** (classes, objets, héritage)
- **Structurer un projet** en modules, packages, et classes
- Utiliser Python dans un environnement de développement (VSCode) avec gestion de versions

## Importance pour le cursus Data Engineer :

La POO permet de **structurer** les composants d'un **pipeline** ou d'un projet **data** de façon **claire, modulaire et réutilisable** :

- Gestion de **connecteurs, transformations** et **loaders**
- Création de composants **réutilisables** pour ingestion, transformation, exposition
- Intégration dans des frameworks Python orientés objets comme **Spark, Airflow**, etc.

# Linux – Les fondamentaux

# Objectifs pédagogiques :

**Lien avec le cursus :** Ce module est essentiel pour maîtriser l'environnement d'exécution des traitements big data, des conteneurs, des scripts et des déploiements cloud.

- **Comprendre** la structure d'un système GNU/Linux
- **Naviguer, manipuler, configurer** le système à travers le terminal
- **Gérer** les **utilisateurs**, les **droits**, les **processus** et les fichiers
- Poser les bases nécessaires à **l'automatisation**, la **containerisation**, le **cloud**

# Linux – Programmation Shell Bash

# Objectifs pédagogiques :

Lien avec le cursus : Ce module permet de construire les scripts d'automatisation indispensables aux flux de données, aux déploiements, aux jobs de CI/CD ou de traitement périodique.

- Maîtriser la syntaxe et la logique de Bash
- Créer des scripts robustes avec conditions, boucles, fonctions
- Automatiser des tâches d'administration, de traitement ou de supervision
- Savoir structurer un script, le documenter et le maintenir

## Importance pour le cursus Cloud Data Engineer :

Le **Data Engineer** doit automatiser tous les traitements répétitifs :

- **Enchaînement de commandes** pour ingestion de données, nettoyage, validation
- **Déploiement automatique** de jobs Spark ou Hadoop
- **Intégration dans des pipelines** CI/CD, ou déclenchement programmé avec cron
- **Surveillance de répertoires**, sauvegardes, transferts FTP/SFTP

Ce module est également essentiel pour comprendre le fonctionnement des scripts utilisés dans **Dockerfile**, **GitLab CI**, **Airflow BashOperator**, ou dans les **machines distantes cloud**.

# Langage SQL et NoSQL – Fondamentaux à l'avancée

# Objectifs pédagogiques :

**Lien avec le cursus :** Ce module constitue le socle des compétences nécessaires à tout travail d'intégration, d'analyse ou d'exposition de données dans un contexte Cloud ou Big Data.

- Savoir **manipuler** efficacement des **bases de données relationnelles** avec SQL (SELECT, JOIN, GROUP BY...)
- **Comprendre** les modèles **NoSQL** (clé/valeur, document, graphe, colonne) et leurs **usages**
- **Concevoir** un schéma de données adapté aux besoins métiers
- **Effectuer** des opérations **complexes** d'agrégation, de **transformation** et **d'analyse**

# Importance pour le cursus Data Engineer :

Le Data Engineer est amené à :

- **Intégrer** des données issues de systèmes hétérogènes (relationnel, fichiers, API)
- **Stocker ces données** dans des systèmes adaptés à leur volumétrie et à leurs requêtes
- **Optimiser** l'accès à ces données via **SQL** standard ou dialectes cloud (BigQuery, Presto, etc.)
- **Utiliser** NoSQL pour les **traitements massifs**, les données **non structurées**, les événements ou les **pipelines** analytiques temps réel

# Hadoop – Développement

# Objectifs pédagogiques :

**Lien avec le cursus** : Même si les technologies comme Spark ou BigQuery ont supplanté Hadoop dans bien des contextes, ce module permet de comprendre la genèse des traitements distribués.

- **Comprendre** l'architecture Hadoop et le fonctionnement de HDFS
- Développer des traitements **MapReduce**
- Interagir avec le **FileSystem Hadoop** (lecture, écriture, permissions)
- **Exécuter** et **monitorer** des jobs sur le cluster
- **Préparer** à l'usage de frameworks modernes s'appuyant sur **HDFS**

# Importance pour le cursus Data Engineer :

Le **Data Engineer** doit :

- **Comprendre** les fondements du traitement distribué pour tirer parti d'outils comme **Spark** ou **Dataflow**
- Savoir **manipuler** un **cluster** Hadoop en contexte d'entreprise ou d'architecture **hybride on-premise + cloud**
- **Gérer** efficacement des volumes **massifs** de données **non structurées**
- **Préparer** et structurer les données dans **HDFS** avant de les traiter dans des **pipelines cloud**

# PySpark – Traitement des données

# Objectifs pédagogiques :

**Lien avec le cursus** : Cœur des traitements massivement parallèles dans une architecture Big Data. PySpark est utilisé dans les pipelines Cloud modernes (GCP, Azure) ou sur clusters Hadoop.

- **Comprendre** l'architecture **Spark** et l'exécution distribuée
- **Manipuler** des **DataFrames** en Python avec l'API PySpark
- Appliquer des **transformations** complexes sur de grands volumes de données
- **Optimiser** les traitements : partitionnement, cache, sérialisation
- Travailler avec des formats avancés (**Parquet, JSON, CSV**)

# Importance pour le cursus Data Engineer :

Le Data Engineer doit :

- **Développer** et **orchestrer** des traitements distribués **scalables**, sans se soucier de l'infrastructure
- **Exploiter** les capacités d'élasticité du cloud avec des moteurs comme Dataproc (GCP) ou Synapse (Azure)
- **Nettoyer, transformer, agréger** des données brutes provenant de multiples sources
- **Optimiser** les pipelines de données dans un cadre CI/CD

**PySpark** représente la colonne vertébrale des jobs de **transformation** de données, notamment dans les architectures Data

# Data Visualisation – Rendre visible l'invisible

# Objectifs pédagogiques :

**Lien avec le cursus** : Sensibilisation essentielle à la dimension visuelle de la donnée. Ce module précède ou complète l'utilisation d'outils comme Power BI, Looker ou Qlik Sense.

- **Comprendre** le rôle stratégique de la **visualisation** de données
- **Identifier** les bonnes pratiques de **narration** visuelle
- **Choisir** le bon graphique selon le message à transmettre
- **Concevoir** des visualisations **lisibles, exploitables, sans biais**

# Importance pour le cursus Data Engineer :

Le Data Engineer :

- **Prépare et transforme** les données en amont des outils BI
- **Travaille** avec les équipes Data Analyst sur les datasets publiables
- **Structure** les données pour maximiser leur **intelligibilité** et **impact**
- Soutient les **équipes métiers** en exposant des **KPIs compréhensibles**

Ce module est un complément indispensable aux modules Power BI, Looker, Qlik Sense, et aux pipelines de traitement (Spark, SQL, ETL).

# Power BI – Les fondamentaux à l'expertise

# Objectifs pédagogiques :

**Lien avec le cursus** : Outil clé de restitution de la donnée pour les métiers. Ce module fait le pont entre l'ingénierie de la donnée (Data Engineer) et son exploitation (Data Analyst / décideurs).

- **Maîtriser** l'interface de Power BI Desktop et **Service**
- **Créer** des modèles de données **efficaces** à partir de sources diverses
- **Construire** des rapports **interactifs, dynamiques** et **lisibles**
- **Maîtriser** les fonctions DAX pour des indicateurs personnalisés
- **Automatiser** les mises à jour, partager et **gouverner les rapports**

# Importance pour le cursus Data Engineer :

Le Data Engineer :

- **Fournit** des **datasets** propres, **normalisés** et prêts à **consommer**
- Doit **tester** et **valider** la pertinence métier des jeux de données
- **Travaille** avec les **Data Analysts** pour exposer les données de façon cohérente
- Peut mettre en place des **modèles** et **indicateurs** techniques (monitoring, volumétrie)

Ce module forme à la **mise en œuvre finale** des efforts d'ingénierie de données dans une logique orientée utilisateur.

# Fondamentaux DevOps

# Objectifs pédagogiques :

**Lien avec le cursus :** Prépare les bases culturelles et pratiques nécessaires à l'automatisation des traitements, à la gestion de l'intégration continue (CI) et du déploiement continu (CD), à la collaboration entre développeurs, data engineers et ops.

- Comprendre les principes fondateurs de la culture DevOps
- Identifier les outils et pratiques pour améliorer les workflows data
- Intégrer une démarche CI/CD dans un environnement Cloud ou hybride
- Renforcer la qualité, la répétabilité et la robustesse des livrables techniques

# Importance pour le cursus Data Engineer :

Le DevOps est **l'épine dorsale des pratiques d'industrialisation data**

- **Automatisation** des traitements (ETL, modèles ML, etc.)
- Mise en place de **tests** et validations dans des **pipelines reproductibles**
- **Déploiement** de code, de **conteneurs**, de **flux**, et de **modèles**
- **Collaboration** continue avec les équipes **ops / sécurité / produit**

# Git et GitLab CI/CD

# Objectifs pédagogiques :

**Lien avec le cursus :** Indispensable à l'**industrialisation** des pipelines data. Permet aux Data Engineers d'intégrer leur travail dans des **chaînes de traitement collaboratives**, traçables, testées, versionnées et **déployées automatiquement**.

- Maîtriser **Git** pour la **gestion** du code source et des **versions**
- **Comprendre** les principes et mécanismes de **CI/CD**
- **Créer** et **configurer** des pipelines dans **GitLab**
- **Automatiser** les **tests**, le **build** et le **déploiement** d'une application data

# Importance pour le cursus Data Engineer

- **Git** est la **base** de toute collaboration **technique** (code, infrastructure, data pipeline)
- **GitLab CI/CD** est utilisé pour **orchestrer** les étapes d'un projet Cloud Data
- Ce module forme à **livrer** du code **stable, testé, traçable** et prêt à **déployer** sur GCP/Kubernetes

# TDD avec PyTest

# Objectifs pédagogiques :

**Lien avec le cursus** : Forme aux méthodes de développement **fiables** et **maintenables**, nécessaires pour sécuriser les transformations de données et les applications cloud déployées à grande échelle.

- **Comprendre** la démarche de Test Driven Development (TDD)
- **Utiliser** PyTest pour **écrire** et **organiser** les tests
- **Assurer** la **qualité** logicielle des scripts Python et Spark
- **Intégrer** les tests dans une chaîne **CI/CD**

## Importance pour le cursus Data Engineer :

- Le **TDD** renforce la **fiabilité** et la **maintenabilité** du code data
- **PyTest** est la solution de référence pour les tests Python dans les **pipelines** DevOps
- **Indispensable** pour tester les **transformations**, les **modèles**, les API, et les **traitements** distribués

# Docker pour Linux – Déploiement de conteneurs virtuels

# Objectifs pédagogiques :

**Lien avec le cursus :** Docker permet de packager les composants data (scripts, APIs, notebooks, modèles, jobs Spark) dans des conteneurs reproductibles, portables et isolés, prêts à être déployés dans le Cloud ou sur une plateforme Kubernetes.

- **Comprendre** le principe et les cas d'usage de la **conteneurisation**
- **Maîtriser** Docker CLI et la **création** de **Dockerfile**
- **Créer, configurer, tester** et **publier** des images Docker
- **Déployer** et lier plusieurs **conteneurs** dans des **environnements** isolés

# Importance pour le cursus Data Engineer :

- **Conteneurisation = reproductibilité + portabilité**
- Les **pipelines** de données **modernes** reposent sur **Docker**
- Docker est la base de l'**orchestration** avec **Kubernetes**, et s'intègre dans toutes les chaînes CI/CD et déploiement Cloud

# Kubernetes – Orchestrer ses conteneurs

# Objectifs pédagogiques :

**Lien avec le cursus :** Kubernetes est la plateforme de référence pour le déploiement scalable et résilient d'applications data en production. Elle est utilisée sur GCP (GKE), AWS (EKS), Azure (AKS), et dans toutes les architectures modernes.

- **Comprendre** l'architecture de **Kubernetes** et ses composants
- **Déployer, monitorer** et **maintenir** des applications conteneurisées
- Utiliser les **fichiers** manifestes **YAML** pour décrire l'état souhaité
- Gérer la **scalabilité**, la **résilience**, les mises à jour **continues**

## Importance pour le cursus Data Engineer :

- Kubernetes est **la brique centrale** de l'infrastructure cloud-native
- Permet **d'automatiser** le **déploiement** des **pipelines** data, des jobs **Spark**, des **APIs** de prédiction
- Assure haute **disponibilité**, montée en **charge**, supervision, **rollback**

# Google Cloud Platform – Core Infrastructure

# Objectifs pédagogiques :

**Lien avec le cursus** : Ce module installe les fondations techniques nécessaires à l'utilisation professionnelle de Google Cloud Platform (GCP). Il permet d'appréhender l'environnement GCP, de comprendre les briques de base (réseau, VM, IAM, stockage), et de se préparer à l'industrialisation des projets data.

- Naviguer dans la console **GCP**, comprendre les **projets, organisations, facturation**
- Comprendre les **services** fondamentaux de compute, de **réseau**, de **sécurité** et de **stockage**

# Objectifs pédagogiques :

- Être capable de **créer**, **configurer** et **exploiter** les ressources manuellement et en ligne de commande
- Poser les bases d'un **déploiement** de **pipeline** data **sécurisé** et **scalable**

## Importance pour le cursus Data Engineer :

- **GCP** est le socle des **projets** data cloud du cursus
- Tous les **outils** du parcours (**BigQuery, Dataflow, AI Platform, Composer**) sont basés sur cette infrastructure
- Ce module est un **prérequis** aux modules spécialisés **GKE, BigQuery, Data Engineering, IA GCP**
- Il permet une bonne **compréhension** des coûts, **IAM, réseaux** et **provisioning**, indispensables à tout déploiement Cloud

# Google Kubernetes Engine – Architecting

# Objectifs pédagogiques :

**Lien avec le cursus** : Ce module est le prolongement logique de Kubernetes et GCP, en ciblant spécifiquement l'architecture et le déploiement de conteneurs sur Google Kubernetes Engine (GKE), la solution Kubernetes managée de GCP.

- **Comprendre** l'architecture et les **fonctionnalités** propres à GKE
- **Créer** et **configurer** un cluster GKE (autopilot et standard)
- **Déployer** des workloads conteneurisés sur **GKE** en suivant les bonnes pratiques **Cloud**
- Intégrer **GKE** à **IAM**, aux ressources **GCP**, à la facturation et à la **supervision**

## Importance pour le cursus Data Engineer :

- **GKE** est la plateforme de référence pour le **déploiement** en **production** de **pipelines** de données, modèles **ML**, **jobs Spark**, **APIs**, **dashboards**, etc.
- Maîtriser **GKE** permet de rendre les projets **scalables**, **supervisés**, **sécurisés** et **redéployables**
- Ce module constitue l'aboutissement infrastructurel de toute la **chaîne data cloud-native**

# Python TensorFlow – Keras

# Objectifs pédagogiques :

**Lien avec le cursus** : Ce module introduit la programmation d'algorithmes **d'intelligence artificielle** en Python avec TensorFlow. Il constitue une première approche concrète du Deep Learning et prépare aux modèles plus complexes utilisés sur le cloud (GCP, Vertex AI).

- **Comprendre** les concepts de base du **Machine Learning** et du **Deep Learning**
- Utiliser **TensorFlow** et **Keras** pour entraîner des modèles **supervisés**

# Objectifs pédagogiques :

- **Créer, entraîner, évaluer et sauvegarder** un modèle de réseau de neurones simple
- **Manipuler** des jeux de données avec **numpy**, **pandas** et **tf.data**

## Importance pour le cursus Data Engineer :

- Les **modèles** de **ML/IA** font partie intégrante des **pipelines** data modernes
- **TensorFlow** est une base incontournable avant de passer à l'industrialisation sur Vertex AI ou DataFlow
- Ce module permet de **comprendre** la logique des **flux de données**, de **l'apprentissage** et du **traitement** GPU/TPU

# **Big Data and Machine Learning Fundamentals on Google Cloud Platform**

# Objectifs pédagogiques :

**Lien avec le cursus** : Ce module pose les bases de l'écosystème **GCP** appliqué à la **Data Science** et au **Machine Learning**. Il prépare l'apprenant à utiliser les services managés du **Cloud** pour **construire, entraîner** et **déployer** des modèles.

- **Comprendre** les concepts de Big Data et d'IA dans **GCP**
- Découvrir les outils **GCP** pour **l'analyse** de données et le **Machine Learning**
- **Entraîner** un modèle simple avec **AutoML** ou Vertex AI
- **Manipuler** les données avec **BigQuery** et préparer les jeux d'apprentissage

## Importance pour le cursus Data Engineer :

- Il marque la **transition** entre la **manipulation** locale de la donnée (Python, Spark) et son traitement **scalable** dans le **cloud**
- Le Data Engineer cloud doit comprendre les **enjeux d'intégration**, de coût, de performances et de gouvernance autour des **modèles IA**
- Ce module est **prérequis** aux modules **Vertex AI**, **Dataflow**, **AutoML** et **pipelines** ML automatisés

# BigQuery – Analyse SQL sur données massives

# Objectifs pédagogiques :

**Lien avec le cursus** : BigQuery est un composant essentiel du stack **GCP**. Ce module permet de **manipuler** des datasets massifs **sans serveur**, directement via SQL, et de **comprendre** la tarification et les bonnes pratiques **d'optimisation**.

- Savoir **créer, interroger, transformer** des données avec BigQuery
- **Comprendre** les bonnes pratiques de **structuration, partitionnement, clustering**
- **Optimiser** les requêtes et réduire les **coûts**
- Utiliser les fonctions **analytiques** avancées et les jointures sur grands **volumes**

## Importance pour le cursus Data Engineer :

- **BigQuery** est au cœur des architectures Data sur **GCP** : il combine **scalabilité, vitesse** et **simplicité SQL**
- Il permet **d'intégrer** l'analyse dans des **pipelines** CI/CD, **Looker, Dataflow** ou **Vertex AI**
- Ce module donne une maîtrise **concrète** de la gestion des données dans un entrepôt **Cloud Serverless**

# Data Engineering on Google Cloud Platform

# Objectifs pédagogiques :

**Lien avec le cursus** : Ce module est le cœur du rôle de Data Engineer Cloud. Il permet de **construire** des pipelines de données **industrialisés, résilients, scalables**, à l'aide des services natifs **GCP** (Dataflow, Pub/Sub, Cloud Composer).

- Mettre en place des **flux de traitement** ETL en mode **batch** et **streaming**
- Utiliser **Pub/Sub, Dataflow, Cloud Storage, BigQuery**
- Orchestrer des **workflows** avec Cloud Composer (Airflow)
- Superviser et **surveiller** la **qualité** et la **performance** des pipelines

## Importance pour le cursus Data Engineer :

- C'est **l'ossature** technique du métier de Data Engineer dans **GCP**
- Sans maîtrise de ces **outils**, il est **impossible** de faire passer à l'échelle un **projet Data**
- Ce module articule les compétences : **Cloud, Python, Spark, CI/CD, IA**

# Analyse et visualisation des données dans Looker

# Objectifs pédagogiques :

**Lien avec le cursus : Looker** est la solution de **BI** native **GCP**. Ce module permet de créer des **dashboards** professionnels, de modéliser les **données**, de rendre les **insights** accessibles aux **utilisateurs** métiers.

- **Comprendre** le fonctionnement de **Looker** (modèle LookML)
- **Connecter** des sources **BigQuery**, **filtrer**, **explorer les données**
- Créer des **dashboards** personnalisés et **interactifs**
- Partager des **rapports**, **automatiser** les mises à jour

## Importance pour le cursus Data Engineer :

- Le Data Engineer fournit la donnée **brute**, mais doit aussi collaborer avec les **métiers**
- Looker permet de **valoriser** les jeux de données grâce à des vues **métier lisibles**
- C'est un outil de **restitution** qui boucle la chaîne Data : **ingestion** → **traitement** → **visualisation**

**Merci pour votre attention**

**Des questions ?**

