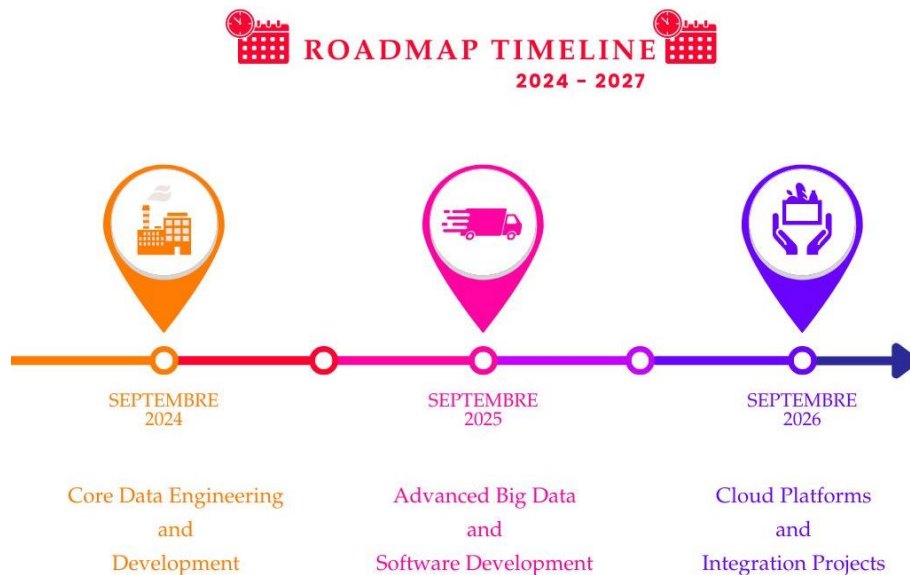


## Table des matières

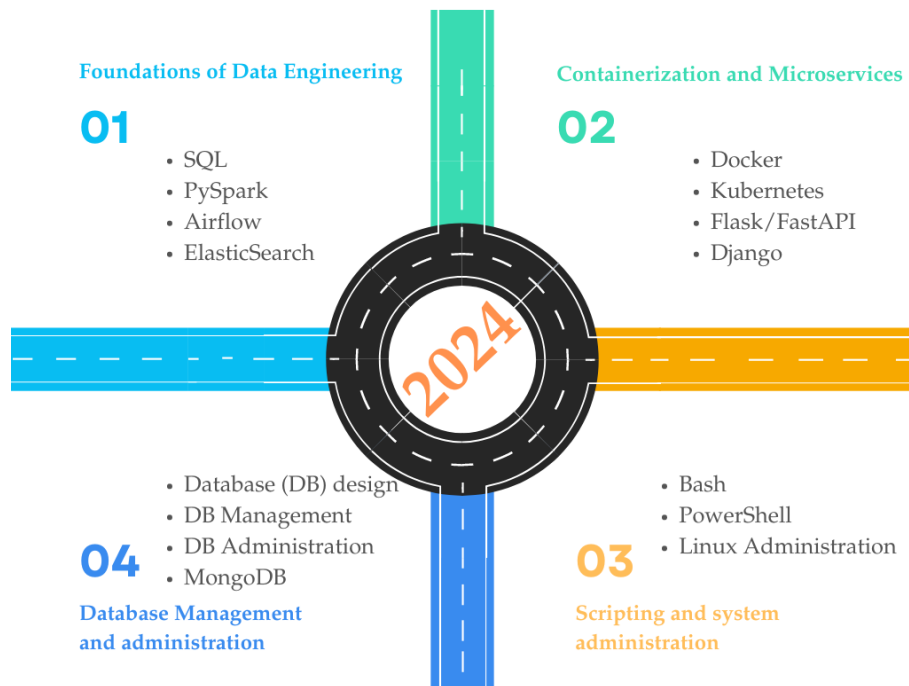
I. Core Data Engineering and Development.....	3
1. Foundations of Data Engineering .....	3
2. Containerization and Microservices .....	4
3. Scripting and system administration .....	5
4. Database Management and administration .....	5
II. Advanced Big Data and Software Development .....	6
1. Advanced Big Data technologies.....	7
2. Advanced Software Development.....	7
3. Software development and Design Patterns .....	8
4. System level programming and DevOps .....	9
III. Cloud Platforms and Integration Projects.....	10
1. Cloud Platforms ans Services.....	10
2. Integration Projects and Real-World Applications.....	10
3. Advanced Topics and Refinement .....	11
4. Capstone Projects and Certification Preparation.....	11

Avec le paysage technologique en constante évolution, il est facile de se sentir submergé ou distrait par la multitude d'informations disponibles. Ainsi, développer ses compétences en tant qu'ingénieur des données ou développeur logiciel nécessite une feuille de route claire et bien structurée. En effet, une roadmap agit comme une boussole garantissant la cohérence, la continuité du processus d'apprentissage en identifiant les priorités et en évaluant leur niveau de difficulté.



Conçue sur trois ans, cette feuille de route décompose des objectifs d'apprentissage complexes en étapes trimestrielles gérables. Son objectif est de concilier l'acquisition de nouvelles connaissances avec une expérience pratique. Suivre ce plan assure une progression régulière et ciblée vers la maîtrise des compétences essentielles dans ces deux domaines.

## I. Core Data Engineering and Development



### 1. Foundations of Data Engineering

#### ○ Module : SQL

Difficulté : 3/5

SQL (Structured Query Language) est fondamental pour la gestion et l'interrogation des données relationnelles. Bien que les requêtes de base soient relativement simples, maîtriser les jointures complexes, les sous-requêtes et l'optimisation des performances peut être difficile.

#### ○ Module : PySpark

Difficulté : 4/5

PySpark est une API Python pour Apache Spark, utilisée pour le traitement des données à grande échelle. Son modèle de calcul distribué et son intégration avec l'écosystème de Spark ajoutent de la complexité, notamment pour l'optimisation des performances et le débogage.

#### ○ Module : Airflow

Difficulté : 4/5

Apache Airflow est une plateforme pour orchestrer des flux et des pipelines de données complexes. Elle implique la configuration et la gestion de Graphes Acyclics Dirigés (DAGs) et l'intégration de diverses tâches, ce qui peut être complexe en raison des besoins en configuration des dépendances et en gestion des erreurs.

○ Module : Elasticsearch

Difficulté : 4/5

Elasticsearch est un moteur de recherche et d'analyse distribué. Il excelle dans la recherche en texte intégral et l'analyse de données en temps réel. Cependant, maîtriser son DSL (Domain Specific Language) et configurer sa nature distribuée peut être complexe.

## 2. Containerization and Microservices

○ Module : Docker

Difficulté : 3/5

Docker est un outil pour créer, déployer et gérer des conteneurs. Bien que les bases de la conteneurisation des applications et de la gestion des images Docker soient relativement simples, des sujets avancés comme le réseau, la sécurité et l'évolutivité peuvent être plus complexes.

○ Module : Kubernetes

Difficulté : 4/5

Kubernetes est une plateforme d'orchestration pour gérer des applications conteneurisées. Cela implique d'apprendre à gérer les clusters, les pods, les déploiements et les services. Sa configuration et sa gestion peuvent être complexes, notamment dans les environnements de production.

○ Module : Flask/FastAPI

Difficulté : 3/5

Flask et FastAPI sont des frameworks pour la création d'applications web en Python. Flask est connu pour sa simplicité et sa flexibilité, tandis que FastAPI offre des performances élevées avec une documentation API automatique. Les deux sont relativement faciles à apprendre, mais leur intégration dans des systèmes plus grands peut être complexe.

○ Module : Django

Difficulté : 3/5

Django est un framework web Python de haut niveau qui encourage un développement rapide et un design propre et pragmatique. Il est relativement facile de commencer avec Django, mais maîtriser ses fonctionnalités et l'intégrer avec d'autres systèmes peut être difficile.

### 3. Scripting and system administration

- Module : PowerShell

Difficulté : 3/5

Docker est un outil pour créer, déployer et gérer des conteneurs. Bien que les bases de la conteneurisation des applications et de la gestion des images Docker soient relativement simples, des sujets avancés comme le réseau, la sécurité et l'évolutivité peuvent être plus complexes.

- Module : Bash Scripting

Difficulté : 4/5

Kubernetes est une plateforme d'orchestration pour gérer des applications conteneurisées. Cela implique d'apprendre à gérer les clusters, les pods, les déploiements et les services. Sa configuration et sa gestion peuvent être complexes, notamment dans les environnements de production.

- Module : Linux Administration

Difficulté : 3/5

Flask et FastAPI sont des frameworks pour la création d'applications web en Python. Flask est connu pour sa simplicité et sa flexibilité, tandis que FastAPI offre des performances élevées avec une documentation API automatique. Les deux sont relativement faciles à apprendre, mais leur intégration dans des systèmes plus grands peut être complexe.

### 4. Database Management and administration

- Module : DB design

Difficulté : 4/5

La conception de bases de données implique la création de schémas, la définition des relations et l'optimisation des performances. Concevoir des bases de données efficaces et évolutives nécessite une compréhension de la normalisation, de l'indexation et de l'optimisation des requêtes, ce qui peut être complexe.

### ○ Module : DB Management

Difficulté : 4/5

La gestion de bases de données couvre l'administration continue des bases de données, y compris les sauvegardes, la récupération et la sécurité. Gérer efficacement des bases de données requiert une connaissance approfondie des systèmes de bases de données et des meilleures pratiques.

### ○ Module : DB Administration

Difficulté : 4/5

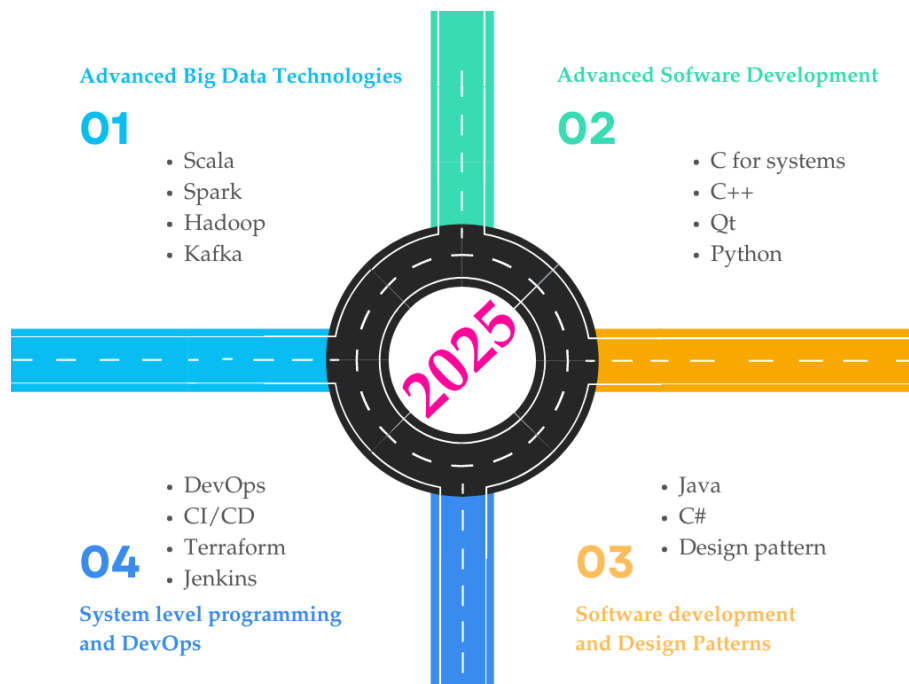
L'administration de bases de données implique la configuration et la maintenance des bases de données, y compris l'optimisation des performances et le dépannage. Ce rôle exige une expertise à la fois dans le logiciel de base de données et dans l'infrastructure matérielle et réseau sous-jacente.

### ○ Module : MongoDB

Difficulté : 3/5

MongoDB est une base de données NOSQL conçue pour la flexibilité et l'évolutivité. Son modèle orienté document diffère des bases de données relationnelles, ce qui peut être difficile à comprendre au début, mais il est relativement simple comparé à des systèmes NOSQL plus complexes.

## II. Advanced Big Data and Software Development



## 1. Advanced Big Data technologies

### ○ Module : Scala

Difficulté : 4/5

Scala est un langage de programmation fonctionnel et orienté objet qui fonctionne sur la Machine Virtuelle Java (JVM). Il est connu pour sa syntaxe concise et ses fonctionnalités puissantes, mais présente une courbe d'apprentissage élevée en raison de ses concepts avancés.

### ○ Module : Spark

Difficulté : 5/5

Apache Spark est un moteur d'analyse unifié pour le traitement des big data. Il fournit un traitement haute performance pour l'ensemble de données à grande échelle et des analyses de données complexes. Son modèle de calcul distribué et son écosystème peuvent être complexes à maîtriser.

### ○ Module : Hadoop

Difficulté : 5/5

Apache Hadoop est un cadre pour le stockage et le traitement distribués de grands ensembles de données. Il implique la compréhension du système de fichiers distribué (HDFS) de Hadoop et du modèle de programmation MapReduce, ce qui peut être assez difficile.

### ○ Module : Kafka

Difficulté : 4/5

Apache Kafka est une plateforme de diffusion d'événements distribuée. Elle prend en charge la messagerie à haut débit et tolérante aux pannes pour le streaming de données en temps réel. Sa nature distribuée et son intégration avec d'autres outils big data ajoutent à sa complexité.

## 2. Advanced Software Development

### ○ Module : C for system

Difficulté : 5/5

La programmation système en C implique la programmation de bas niveau, y compris la manipulation directe du matériel et l'optimisation des performances. Elle nécessite une connaissance approfondie de l'architecture informatique et de la gestion de la mémoire.

### ○ Module : C++

Difficulté : 4/5

C++ est un langage de programmation puissant avec un support pour la programmation orientée objet et générique. Il est connu pour ses performances et sa complexité, y compris la gestion manuelle de la mémoire et les fonctionnalités avancées telles que les modèles.

### ○ Module : Qt

Difficulté : 4/5

Qt est un framework de développement d'applications multiplateformes. Il est utilisé pour créer des applications GUI et nécessite la compréhension de ses API étendues et du mécanisme signal-slot, ce qui peut être complexe.

### ○ Module : Python

Difficulté : 3/5

Python est un langage de programmation polyvalent connu pour sa lisibilité et sa facilité d'utilisation. Bien que l'apprentissage des bases soit simple, maîtriser les fonctionnalités avancées, les bibliothèques et les frameworks peut être plus complexe.

## 3. Software development and Design Patterns

### ○ Module : Java

Difficulté : 4/5

Java est un langage de programmation orienté objet largement utilisé. Il est connu pour sa portabilité et son écosystème étendu. La maîtrise implique de comprendre son modèle de concurrence, la gestion de la mémoire et les fonctionnalités d'entreprise.

### ○ Module : C#

Difficulté : 4/5

C# est un langage de programmation orienté objet moderne développé par Microsoft. Il est utilisé pour créer une variété d'applications et implique de comprendre ses fonctionnalités, telles que LINQ et la programmation asynchrone.

### ○ Module : Design Pattern

Difficulté : 4/5

Les modèles de conception sont des solutions réutilisables générales aux problèmes de conception logicielle courants. Maîtriser les modèles de conception nécessite de comprendre leur application dans divers scénarios et comment ils améliorent la maintenabilité et l'évolutivité du code.



## 4. System level programming and DevOps

### ○ Module : DevOps

Difficulté : 4/5

DevOps est un ensemble de pratiques qui combine le développement logiciel et les opérations informatiques. Il se concentre sur l'automatisation du processus de livraison de logiciels et l'amélioration de la collaboration entre les équipes de développement et d'exploitation.

### ○ Module : CI/CD

Difficulté : 4/5

L'intégration continue et le déploiement continu (CI/CD) impliquent l'automatisation de l'intégration et le déploiement des modifications du code. La mise en place de pipelines CI/CD nécessite la compréhension de divers outils et des meilleures pratiques pour l'automatisation.

### ○ Module : Terraform

Difficulté : 4/5

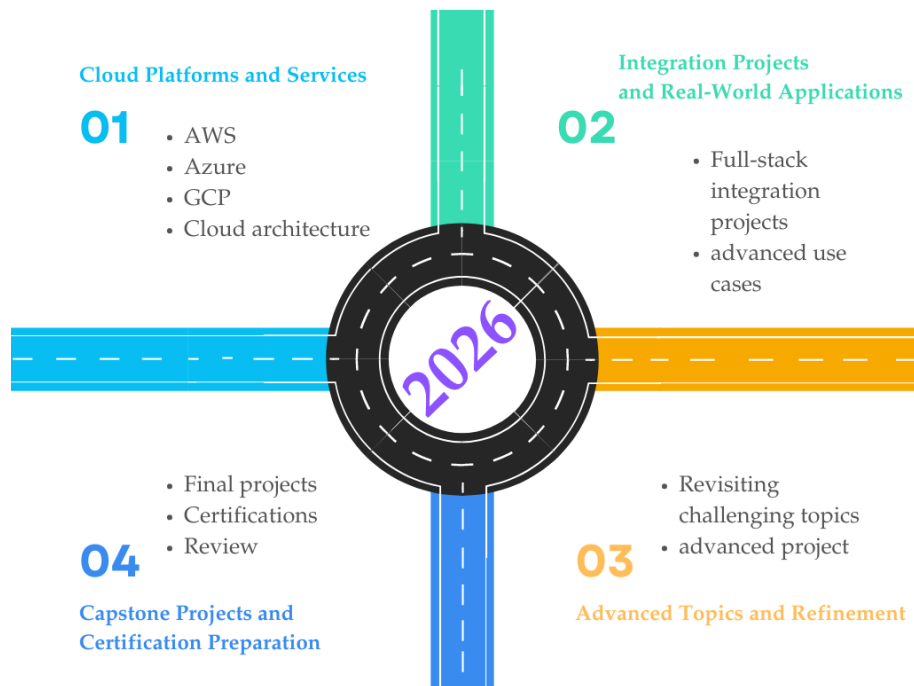
Terraform est un outil d'infrastructure as code (IaC) qui permet de gérer et de provisionner l'infrastructure à l'aide de fichiers de configuration. Apprendre Terraform implique de comprendre son langage de configuration et de l'intégrer avec des fournisseurs de cloud.

### ○ Module : Jenkins

Difficulté : 4/5

Jenkins est un serveur d'automatisation open-source utilisé pour la construction, le déploiement et l'automatisation des tâches. Maîtriser Jenkins implique de comprendre ses plugins et de créer des pipelines CI/CD efficaces.

### III. Cloud Platforms and Integration Projects



#### 1. Cloud Platforms and Services

Les plates-formes cloud (par exemple, AWS, Azure, GCP) offrent une gamme de services pour l'informatique, le stockage et les réseaux. La maîtrise implique de comprendre divers services, les stratégies de déploiement et les meilleures pratiques en matière de sécurité.

#### 2. Integration Projects and Real-World Applications

Les sujets avancés dans le domaine de l'ingénierie des données, du développement logiciel, et de l'informatique en général, représentent le stade où l'on passe de la maîtrise des compétences fondamentales à la spécialisation et à l'optimisation des connaissances acquises. Ce processus de perfectionnement se concentre sur l'approfondissement des concepts, l'amélioration de l'efficacité, et l'application de meilleures pratiques pour résoudre des problèmes complexes dans des environnements réels.

### 3. Advanced Topics and Refinement

Les sujets avancés dans le domaine de l'ingénierie des données, du développement logiciel, et de l'informatique en général, représentent le stade où l'on passe de la maîtrise des compétences fondamentales à la spécialisation et à l'optimisation des connaissances acquises.

### 4. Capstone Projects and Certification Preparation

La phase finale de ce parcours de formation est centrée sur la mise en pratique des compétences acquises à travers des projets de fin d'études (Capstone Projects) et la préparation aux certifications professionnelles.