



# Data Lake partie 2 : ingestion avancée, monitoring et sécurité

*Azure Data lake Storage Gen2, Azure Monitor, Azure Databricks, Microsoft Entra ID, Azure KeyVault*

## Contexte Fictif

En tant que **Data Ingénieur** au sein de l'entreprise **DataMoniSec**, vous êtes chargé de mettre en place une infrastructure de données robuste et sécurisée sur **Microsoft Azure**. Votre mission s'articule de la **sécurité** et du **monitoring** du data lake.

Votre mission est de :

- **Configurer** un Data Lake pour centraliser les données de l'entreprise.
- **Ingérer** des données provenant de différentes sources.
- Mettre en place des mesures de **sécurité** avancées pour protéger les données sensibles.
- Configurer **Azure Databricks** pour permettre à l'équipe Data Science d'analyser les données.

- Implémenter un système de **monitoring** et d'**alertes** pour surveiller l'infrastructure.
- [Bonus] : **Spark, terraform, pricing, ingestion avancée**



## Partie 1 : Veille sur les Systèmes de Sécurité

### Objectif

Acquérir une **vision panoramique** des méthodes de sécurité disponibles sur Azure pour protéger les données dans un Data Lake. Je ne vous demande pas d'être un expert sur chacun de ces éléments mais d'avoir une intuition de comment ça marche et de pouvoir décrire en une phrase ce que fait chaque service.

Vous pouvez également vous lancer dans le projet et vous formez sur ces concepts au fur et à mesure.

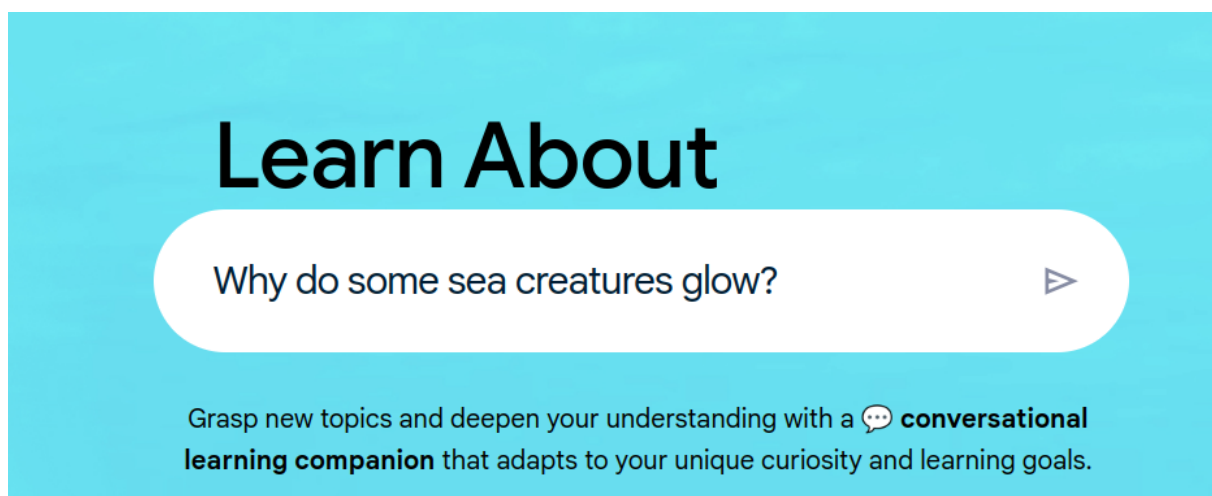
### Veille

**Important pour le projet :**

- **Storage Access Keys**
- **Shared Access Signatures (SAS) (Delegation Key)**

- **Microsoft Entra ID** (anciennement Azure Active Directory) (service principal)
- **Azure Key Vault**
- **IAM et Role-Based Access Control (RBAC)**

**Ressources** : Utilisez des outils tels que Google, la documentation officielle d'Azure, des modèles de langage (LLMs), et autres sources fiables. Je vous recommande d'utiliser le nouvel outil **Learning about** de Google (un LLM spécial pour apprendre sur un sujet, il nécessite un VPN car il est disponible qu'aux états-unis)



## Livrable

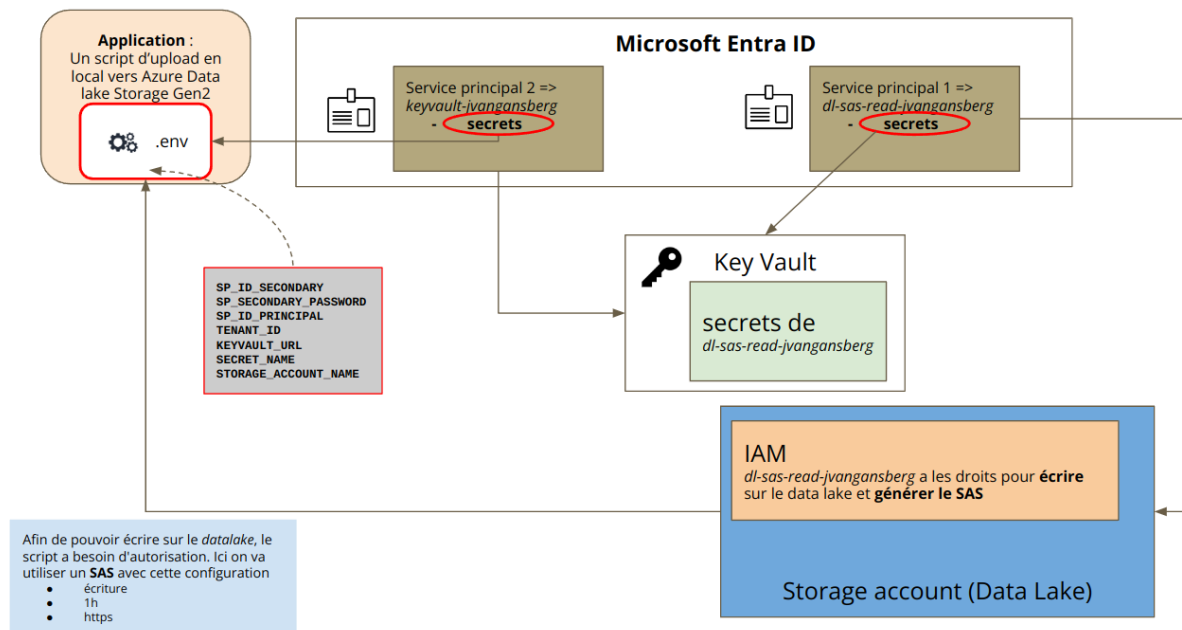
- Un rapport qui synthétise vos recherches en décrivant chaque méthode de sécurité, ses avantages, ses limitations, et des cas d'utilisation.
- [Bonus] Sur la base du volontariat, un apprenant pourra expliquer un ou plusieurs des composants aux autres (5 minutes maximum)

## Partie 2 : Création et ingestion sécurisée sur le data lake

### Objectif

L'objectif de cette partie est de créer des scripts Python pour stocker des données dans un **Azure Data Lake Storage Gen 2** de manière sécurisée. Vous allez vous concentrer sur la gestion des identités et des secrets en utilisant des

outils comme **Azure Key Vault**, tout en pratiquant les concepts de sécurité avec Azure.



## Concrètement :

1. **Télécharger et stocker des fichiers CSV ou Parquet** sur le Data Lake.
2. **Créer une configuration sécurisée** pour générer et utiliser des **SAS Tokens** afin de upload des fichiers.
3. **Explorer une approche avec deux Service Principals (SP)** pour renforcer la sécurité et pratiquer l'utilisation de Key Vault.



## Pourquoi deux SP ?

C'est un peu comme des poupées russes 🧸 : chaque couche ajoute une barrière supplémentaire pour protéger vos données. Cette configuration peut sembler un peu complexe au premier abord (et je comprends pourquoi 😊), mais elle a un objectif clair : **réduire l'exposition des secrets sensibles** tout en vous faisant pratiquer des concepts importants comme Key Vault, les rôles Azure, et la gestion des SAS Tokens.

## Logique de cette configuration

### 1. SP Secondaire (Accès à Key Vault) :

- Ce SP a un rôle ultra-limité : accéder uniquement au secret stocké dans Key Vault. Il ne peut pas manipuler directement les données du Data Lake.
- Pourquoi ? Si un secret doit être exposé dans l'application, autant que ce soit celui d'un SP qui n'a **aucun accès direct** au Data Lake. Cela limite l'impact d'une compromission.
- De plus, grâce à **Key Vault**, vous bénéficiez de fonctionnalités comme :
  - **Journalisation des accès** : Qui a accédé à quel secret, quand et depuis où.
  - **Rotation des clés** : En cas de fuite, vous pouvez régénérer les clés sans toucher au reste de l'infrastructure.

### 2. SP Principal (Accès au Data Lake) :

- Ce SP est plus puissant : il génère les **SAS Tokens** nécessaires pour manipuler les données. Cependant, son mot de passe (ou secret) est **stocké dans Key Vault**, donc inaccessible directement depuis l'application.
- Pourquoi ? Cette séparation permet de ne jamais exposer directement les informations sensibles d'un SP puissant. Vous isolez les responsabilités et réduisez les surfaces d'attaque.

## Avantages de cette approche

### 1. Limitation des privilèges exposés :

- L'application n'a accès qu'à un SP (le Secondaire) avec une permission limitée : `get` sur les secrets. C'est comme donner une clé qui ouvre seulement une boîte spécifique.
- Le SP Principal est protégé par Key Vault : son mot de passe n'est jamais directement dans le code.

### 2. Audit et contrôle :

- Key Vault permet d'auditer toutes les actions : chaque tentative de lecture ou d'accès est journalisée.
- Si une fuite est détectée, vous savez **exactement qui a accédé aux secrets** et pouvez réagir rapidement.

### 3. Rotation simplifiée des clés :

- En cas de fuite, vous pouvez régénérer toutes les clés dans Key Vault sans toucher à l'application.
- Seule la clé d'accès au SP Secondaire devra être mise à jour dans l'application.

---

## Pourquoi cette approche est cohérente

Certes, on déplace le problème (il faut toujours un secret dans l'application, celui du SP Secondaire). Mais :

1. Ce secret est **moins critique** : il n'offre aucun accès direct au Data Lake.
2. Le point d'entrée est **Key Vault**, qui est conçu pour sécuriser et journaliser les accès.
3. La séparation des SP garantit que, même si le SP Secondaire est compromis, l'accès au Data Lake reste protégé.

---

## Conclusion

Cette configuration peut paraître un peu compliquée, mais elle est une **bonne pratique de sécurité** :

- Vous minimisez les impacts d'une compromission.

- Vous utilisez les outils Azure (Key Vault, journaux, rotation) pour gérer les secrets efficacement.

Et soyons honnêtes : le vrai but ici, c'est aussi de vous faire manipuler ces concepts et outils ! 😊

## Activités

### Étape 1 : Configuration Initiale

1. **Créez un Data Lake Storage Gen 2** dans votre groupe de ressources (par exemple, `RG_JVANGANSBERG` ).
2. Configurez le **conteneur de stockage** pour accueillir vos fichiers. (exemple : `data` )
3. Créer un Key Vault `keyvault [ initial_prénom nomdefamille ]`
4. Effectuer la configuration sécurisée (voir **Guide de Configuration Sécurisée** plus bas)

### Étape 2 : Travail avec les données

Vous allez manipuler deux types de données :

- **Données CSV** : [Inside Airbnb - Get the Data](#).
- **Données Parquet** : [Hugging Face Datasets](#).

## Niveaux de Difficulté

### A. Données CSV depuis Inside Airbnb

1. **Niveau 1** : Faire un script qui va stocker un csv sur le data lake. Vous pouvez aller sur le lien plus haut afin de télécharger un csv relatif à la ville de **Barcelone** (vous pouvez prendre celui que vous voulez mais disons [reviews.csv](#) si vous ne savez pas lequel prendre). Pour sauvegarder le csv sur le data lake vous allez faire un script d'ingestion qui va générer un SAS mais vous ne devez pas exposer la SAK. Vous créerez au préalable deux services principales pour générer ce SAS de façon sécurisée.

*Note : Pour ce point vous avez le choix entre d'abord télécharger le fichier en local ou pas.*



2. **Niveau 2** : Faire la même chose mais en faisant un script qui va télécharger toutes les données relatif à l' `espagne` .
3. **[Facultatif]** : Faire la même chose avec `Azure Data Factory` . Comme c'est un service Azure vous n'aurez pas besoin de la configuration du niveau 1, faites le uniquement si vous voulez revoir `Azure Data Factory` mais le focus de cette semaine c'est la sécurité et le monitoring.

## B. Données Parquet depuis Hugging Face

1. **Niveau 1** : Pareil qu'au point A avec un fichier parquet
2. **Niveau 2** : Tous (ou un batch de plusieurs 5 à 20) les fichiers parquet
3. **Niveau 3** : Avec Data Factory

La configuration du B sera quasiment identique, c'est juste pour manipuler un fichier parquet plus tard dans le brief.

---

## Guide de Configuration Sécurisée

### 1. Créez deux Service Principals (SP)

- **SP Secondaire** :
  - Nom: `sp-keyvault-[Initial prénom+nomdefamille]`
  - Stratégie d'accès (*accés policies*): Accès à `Azure Key Vault` (uniquement `get` ).
  - Permet de récupérer les secrets (mot de passe du SP principal).
- **SP Principal** :
  - Nom: `sp-datalake-[Initial prénom+nomdefamille]`
  - Rôle: `Storage Blob Data Contributor`

### 2. Configurez Key Vault

- Stockez dans Key Vault le **secret du SP Principal**.
- Attribuez une **Access Policy** ou un rôle **Key Vault Secrets User** au SP Secondaire pour qu'il puisse accéder au secret.

### 3. Génération des SAS Tokens

- Utilisez le SP Principal pour générer une User Delegation Key.



- Avec cette clé, créez des **SAS Tokens** qui permettent d'écrire ou de lire des fichiers dans le Data Lake.

Dans votre application (le script de upload) vous ne devrez stocker les informations suivantes :

```
SP_ID_SECONDARY
SP_SECONDARY_PASSWORD
SP_ID_PRINCIPAL
TENANT_ID
KEYVAULT_URL
SECRET_NAME
STORAGE_ACCOUNT_NAME
```

Rendez votre code modulaire, le code qui vous permettra de générer un SAS sera commun aux fichiers parquet et aux fichiers csv. Ça représente une opportunité pour modulariser votre code.

## Journalisation


L'un des avantages d'utiliser **Azure Key Vault** comme point d'entrée est la possibilité de **journaliser les interactions** avec celui-ci. Vous pouvez activer cette fonctionnalité via les **paramètres de diagnostic**. Il existe différentes manières de sauvegarder ces journaux, et l'une des plus simples est de les **stocker en tant qu'archives dans un compte de stockage**.

Remarque : Bien que vous puissiez stocker les journaux sur le data lake, il est déconseillé de les conserver sur le même data lake que vous cherchez à sécuriser. Cela pourrait présenter un risque de sécurité en exposant les détails des accès et des opérations sur le data lake lui-même.

## Diagnostic setting ...

 Save  Discard  Delete  Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

 A more flexible, faster, and robust way to collect metrics is in preview! Click [here](#) to configure platform metrics collection from microsoft.keyvault/vaults to storage account, event hubs, and Log Analytics workspace. [Learn more.](#)

Diagnostic setting name journalisation-test-keyvault-jv

### Logs

Category groups ⓘ

☒ audit ☒ allLogs

Categories


☒ Audit Logs  
☒ Azure Policy Evaluation Details


### Metrics

### Destination details

☐ Send to Log Analytics workspace

☒ Archive to a storage account

 You'll be charged normal data rates for storage and transactions when you send diagnostics to a storage account.

 Showing all storage accounts including classic storage accounts

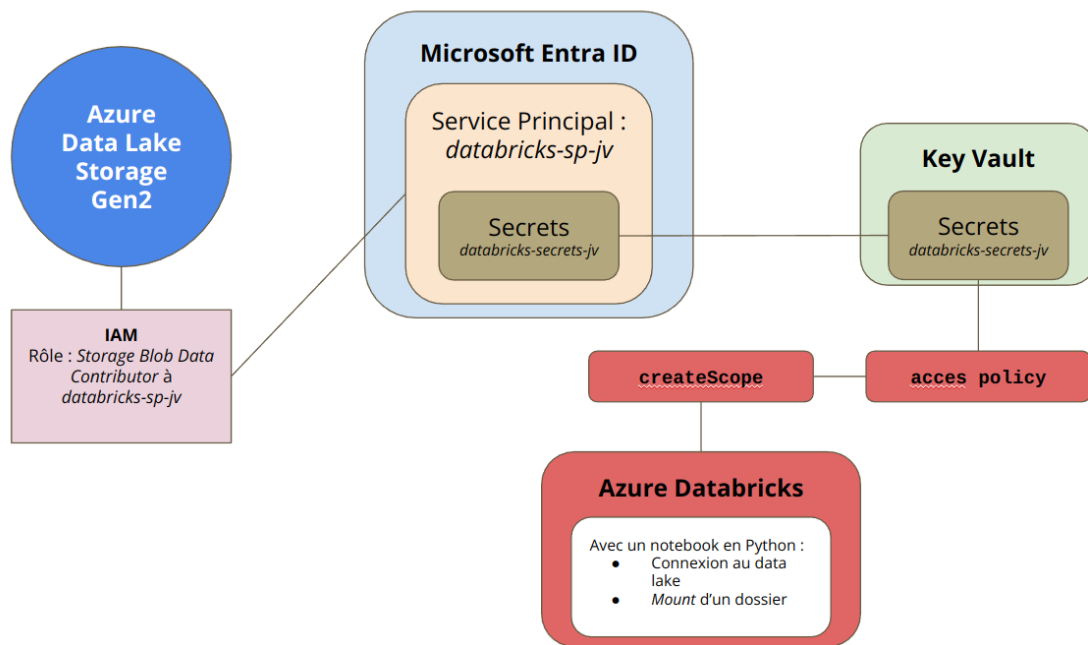
## Livrables

1. **Repertoire du projet sur github**
2. **Documentation :**
  - Expliquez les rôles des SP et la configuration.
  - Justifiez les permissions attribuées.
3. **Journal Key Vault :** Fournissez les logs qui montrent les actions sur les secrets.

## Partie 3 : Configuration d'Azure Databricks

### Objectif

Configurer Azure Databricks pour permettre à l'équipe Data Science d'analyser les données, tout en mettant en œuvre les mesures de sécurité appropriées.



Cette partie s'appuie sur ces tutos microsofts que vous devrez suivre en détail :

- <https://learn.microsoft.com/en-us/azure/databricks/connect/storage/tutorial-azure-storage>
- <https://learn.microsoft.com/en-us/azure/databricks/dbfs/mounts?source=recommendations>

Ces tutoriels font appels à plusieurs outils Azure que vous aurez explorer plus haut durant votre veille.

## Activités

- Créer la ressource : `databricks- [ initial_prenom nom_de_famille ]` Sélectionner bien la version "Standard" et surtout pas "Premium".
- **Configurer le cluster Azure Databricks** avec la configuration suivante :
  - **Version : 15.4 LTS**
  - **Type de Cluster : Standard-DS3\_v2** (prix max : **0.75 DBU/h**)
  - **Arrêt Automatique** : Après **1 heure** d'inactivité
  - **Mode d'Accès : Single User**
- **Intégrer les Composants de Sécurité :**

- **Azure Key Vault** : Pour stocker et gérer les secrets et les clés de chiffrement.
- **Microsoft Entra ID** : Pour l'authentification et l'autorisation des utilisateurs.
- **Rôles IAM** : Pour définir les permissions d'accès aux ressources.
- **Monter le Data Lake dans Databricks** :
  - Utiliser les bonnes pratiques pour connecter **Azure Databricks** à **Azure Data Lake Storage Gen2** en utilisant les méthodes sécurisées (par exemple, en utilisant **Azure Key Vault** pour stocker les credentials).
- **Charger un Fichier CSV ou Parquet** :
  - Utiliser le code Spark pour charger le fichier depuis le Data Lake et afficher les 20 premières lignes :

```
df = spark.read.format("csv") \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .load(file_path)

df.show()
```

- Le code spark obligatoire pour ce projet s'arrête là, donc pas besoin d'avoir des connaissances au préalable. Nous consacrerons un temps pour apprendre Spark plus tard dans la formation. Si vous le souhaitez vous pourrez, aller plus loin dès maintenant (voir bonus)

## Livrables

- **Documentation** : Décrivez les étapes de configuration, les challenges rencontrés, et comment vous les avez résolus.
- **Notebook Databricks** : Contenant le code utilisé pour monter le Data Lake et charger les données.

## Partie 4 : Monitoring et Alertes

### Objectif

Mettre en place un système de monitoring et d'alertes pour surveiller le Data Lake et être en mesure de réagir rapidement en cas d'incidents.

L'objectif de cette partie est de découvrir `Logs Analytics`, `Activity logs`, `Metrics`, `insights` et `Alerts`. Ce sont des services que vous pouvez atteindre de différentes façons. Par exemple, dans votre `storage account`, il y a une section `monitoring` avec ces services pré-configurés pour surveiller votre instance de `storage account`. Vous pouvez également utiliser ces services de manière indépendante, en faisant par exemple une recherche `Metrics` dans la barre de recherche. Ou enfin vous pouvez les utiliser via le service unifié `Monitor`.

Vous pouvez effectuer une veille avant d'utiliser ces outils ou alors au fur-et-à-mesure que vous les utilisez.

- **Activity Logs :**

- **Objectif :** Analysez les actions effectuées sur le Data Lake durant la partie précédente
- **Appliquer des Filtres :** Utilisez les filtres pour vous concentrer sur les événements pertinents : les événements qui ont été créés par votre compte, ou sur vos ressources (adls, databricks)
- **Exporter les Logs :** Configurez l'exportation des Activity Logs vers un **Log Analytics Workspace**.

- **Metrics:**

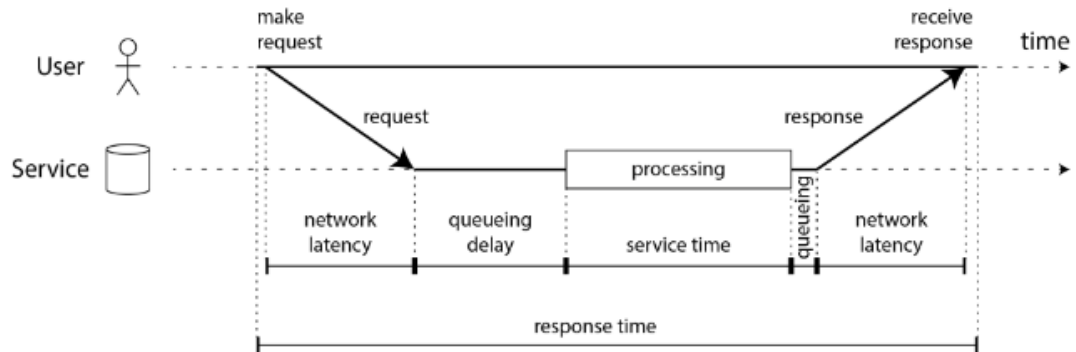
- **Identifier les Métriques Clés :** Par exemple, le débit entrant/sortant, la latence, le nombre de requêtes réussies/échouées.
- **Créer un Dashboard :** Utilisez `Azure Monitor` pour créer un dashboard personnalisé pour le Data Lake.

- **Insights :**

- **Utiliser Azure Storage Insights :** Pour obtenir une vue détaillée des performances et de l'utilisation du Data Lake.
- Quel est la différence entre `Metrics` et `Insights` ?
- **Comprendre les Latences :** C'est quoi la différence entre E2E latency et server latency ?
  - **E2E Latency :** Temps total pour qu'une requête soit traitée du début à la fin.

- **Server Latency** : Temps que le serveur met à traiter la requête, excluant les délais de réseau.

Schéma pour vous aider :



- **Alerts :**
  - **Configurer des Alertes sur les Métriques :**
    - **Exemple** : Définir une alerte lorsque le trafic entrant (Ingress) dépasse **1 Go**. Testez son déclenchement en simulant l'événement.
  - **Configurer des Alertes sur les Activity Logs :**
    - **Exemple** : Avertir l'administrateur lorsque les **Storage Access Keys** sont régénérées.
  - **Définir des Règles par Sévérité :**
    - **Sévérité 1-3** : Définir que le dépassement du seuil ingress est de sévérité plus faible et donc envoie uniquement un **email** à l'administrateur.
    - **Sévérité 4-5** : Définir que la régénération des clés est plus grave et donc il va envoyer un **email** et d'un **SMS**.
  - Dans cette partie, on considère que vous êtes l'administrateur, configurez donc votre email et votre numéro de téléphone. Pour effectuer ces tests.

## Livrables

- **Captures d'Écran** : Des dashboards, des alertes configurées, et des notifications reçues.
- **Rapport** : Décivant le système de monitoring mis en place, les raisons des choix effectués, et comment le système répond aux besoins de l'entreprise.

---

## Partie 5 : Bonus

### Option 1 : Ingestion Avancée (niveau 2)

- **Objectif** : Automatiser l'ingestion des données avec des scripts avancés ou des pipelines.
- **Activités** :
  - **Pour Inside Airbnb** :
    - Écrire un **script Python** qui parse la page HTML pour extraire les URLs des fichiers liés à l'Espagne (par exemple, en utilisant des expressions régulières ou des bibliothèques comme BeautifulSoup).
    - Utiliser **Azure Data Factory** pour créer un pipeline avec un **ForEach** qui télécharge chaque fichier et l'ingère dans le Data Lake.
  - **Pour Hugging Face** :
    - Automatiser le téléchargement de plusieurs fichiers Parquet en utilisant des scripts ou Azure Data Factory.
    - Gérer les exceptions et les cas où certains fichiers ne sont pas disponibles.

### Option 2 : Exploration des Données avec Spark

- **Objectif** : Approfondir l'analyse des données en utilisant **PySpark**.
- **Activités** :
  - Charger les données dans un DataFrame Spark.
  - Effectuer des opérations d'analyse : filtrage, agrégation, jointures.
  - Visualiser les résultats directement dans le notebook.
  - **Ressource** : Suivre un tutoriel vidéo recommandé (lien fourni par le formateur).

### Option 3 : Révocation des Accès en Cas d'Incident

- **Objectif** : Automatiser la réponse aux incidents critiques.
- **Activités** :



- Utiliser **Azure Logic Apps** ou **Azure Functions** pour créer un workflow qui révoque automatiquement les accès au Data Lake en cas d'alerte critique.
- Intégrer ce workflow avec le système d'alertes configuré précédemment.

## Option 4 : Inspection du Pricing

- **Objectif** : Comprendre et optimiser les coûts associés à vos configurations Azure.
- **Activités** :
  - **Analyser le Coût** de la configuration Azure Databricks mise en place :
    - Utiliser la **Calculatrice de Prix Azure** pour estimer le coût mensuel.
    - Identifier les principaux facteurs de coût.
  - **Optimiser les Coûts** :
    - Explorer des options telles que la mise à l'échelle automatique, l'utilisation d'instances spot, ou la réduction de la taille du cluster.
    - Proposer des recommandations pour équilibrer performance et coût.

## Option 5 : Utilisation de terraform

- **Objectif** : Utiliser terraform pour gérer vos ressources Azure.