TP : Amazon S3

L'objectif de ce TP est de manipuler des données dans un bucket Amazon S3 en utilisant l'outil de ligne de commande s3cmd et le SDK Python boto3.

Exercice 1 : utiliser S3cmd

1. Installer le client S3cmd et lancer la commande suivante pour lister les fichiers

> Pour installer s3cmd, lancer la commande `sudo apt update && sudo apt install s3cmd`

```
s3cmd ls s3://user/ --host minio.lab.sspcloud.fr --host-bucket
minio.lab.sspc
loud.fr
```

> Les arguments `--host` et `--host-bucket` doivent être ajoutés à chaque fois, sinon par défaut s3cmd utilise `amazonaws.com`

> Remplacer `user` par votre username onyxia

2. Lancer une commande pour créer un bucket. Quelle est l'erreur renvoyée et pourquoi ?

Exercice 2 : Partager ses données

1. Déplacer le fichier `vehicules-2022.csv` dans un dossier `public` avec s3cmd
2. Appliquer une policy à votre bucket qui autorise tout le monde à lire les données dans le dossier `public`.
3. Vérifier que cela fonctionne en se connectant avec votre navigateur web à l'URL https://minio.lab.sspcloud.fr/user/public/vehicules-2022.csv

> `user` est votre identifiant de bucket

Exercice 3 : Utiliser boto3 pour manipuler ses données

> La documentation du sdk est disponible ici

1. Utiliser la librairie boto3 pour lister seulement **le nom** des fichiers du bucket `donnee-insee`. Vous pouvez utiliser la commande suivante pour créer un client s3 :

```
import boto3
s3 = boto3.client("s3", endpoint_url="https://minio.lab.sspcloud.fr")
```

1. Télécharger le fichier parquet `bpe` dans votre service Onyxia en utilisant boto3
2. Ajouter le fichier téléchargé en utilisant boto3 dans votre bucket s3 personnel dans un dossier `data`

Exercice 4 : Avantage du format parquet

1. Lire le fichier `bpe.parquet` avec `pandas` et enregistre-le sous format CSV.
2. Comparer les tailles des deux fichiers en python. Quel est le format le plus léger ?
3. Comparer le temps de chargement du datasets et le temps de lecture de la colonne `NUMVOIE`