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.sspcloud.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 ?

```
s3cmd mb s3://test/ --host minio.lab.sspcloud.fr --host-bucket
minio.lab.sspcloud.fr
```

Il s'agit d'une erreur de permission. Sur Onyxia il est seulemment possibled d'avoir un bucket à son nom

Exercice 2 : Partager ses données

1. Déplacer le fichier `vehicules-2022.csv` dans un dossier `public` avec s3cmd

```
s3cmd mv s3://user/vehicules-2022.csv s3://user/public/vehicules-2022.csv
--host minio.lab.sspcloud.fr --host-bucket minio.lab.sspcloud.fr
```

2. Appliquer une policy à votre bucket qui autorise tout le monde à lire les données dans le dossier `public`.

Dans un fichier `policy.json` écrire le contenu suivant :

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
```

```
            "Principal":{
               "AWS":[
                  "*"
               ]
            },
            "Action":[
               "s3:GetObject"
            ],
            "Resource":[
               "arn:aws:s3:::tgameiro/public/*"
            ]
         }
      ]
}
```

Puis appliquer la policy avec la commande suivante :

```
s3cmd setpolicy policy.json s3://user --host minio.lab.sspcloud.fr --host-
bucket minio.lab.sspcloud.fr
```

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 `donnees-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")
```

```
list_files = s3.list_objects_v2(Bucket="donnees-insee")
for f in list_files["Contents"]:
    print(f["Key"])
```

2. Télécharger le fichier parquet `bpe` dans votre service Onyxia en utilisant boto3

```
s3.download_file(Bucket="donnees-insee",
Key="diffusion/BPE/2023/bpe.parquet", Filename="bpe.parquet")
```

3. Ajouter le fichier téléchargé en utilisant boto3 dans votre bucket s3 personnel dans un dossier `data`

```
s3.upload_file(Bucket="user", Key="diffusion/BPE/2023/bpe.parquet",
Filename="bpe.parquet")
```

Exercice 4 : Avantage du format parquet

1. Lire le fichier `bpe.parquet` avec `pandas` et enregistre-le sous format CSV.

```
import pandas as pd

df = pd.read_parquet("bpe.parquet")
df.to_csv("bpe.csv")
```

2. Comparer les tailles des deux fichiers en python. Quel est le format le plus léger ?

```
import os

size_csv = os.path.getsize("bpe.csv")
size_parquet = os.path.getsize("bpe.parquet")

def convert_to_mb(size: int):
    return size // 1024 // 1024

convert_to_mb(size_csv), convert_to_mb(size_parquet)
# (859, 203)
```

3. Comparer le temps de chargement du datasets et le temps de lecture de la colonne `NUMVOIE`

On peut utiliser les *magic commands* sur jupyter :

```
%%time
numvoie = pd.read_parquet("bpe.parquet")["NUMVOIE"]

#CPU times: user 15.6 s, sys: 4.18 s, total: 19.8 s
#Wall time: 6.34 s
```

```
%%time
numvoie = pd.read_csv("bpe.csv")["NUMVOIE"]

#CPU times: user 16.7 s, sys: 5.09 s, total: 21.8 s
#Wall time: 21.8 s
```