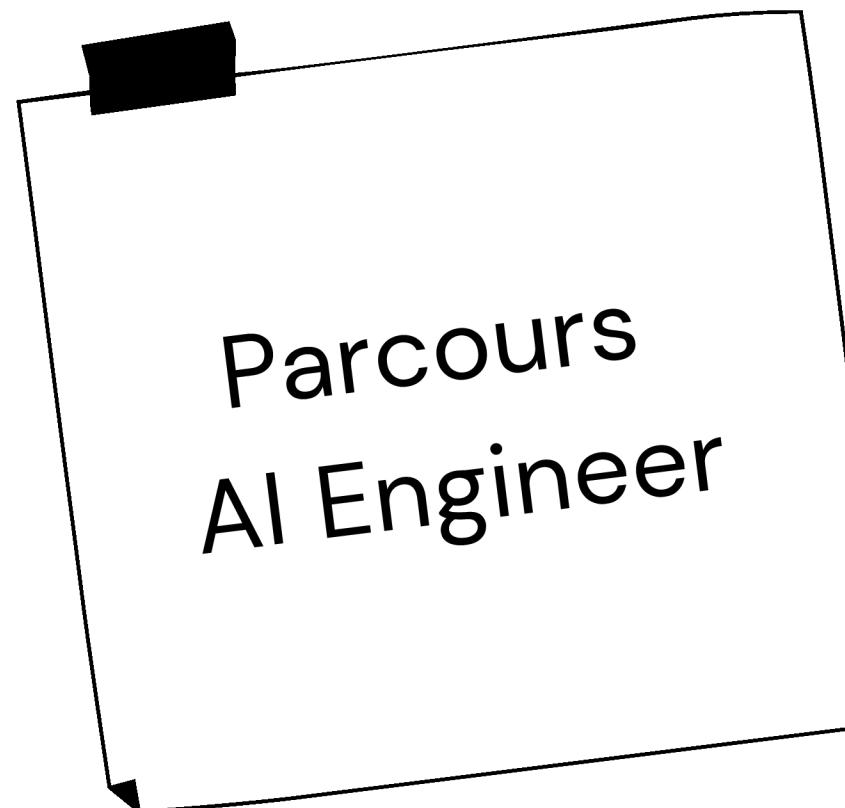
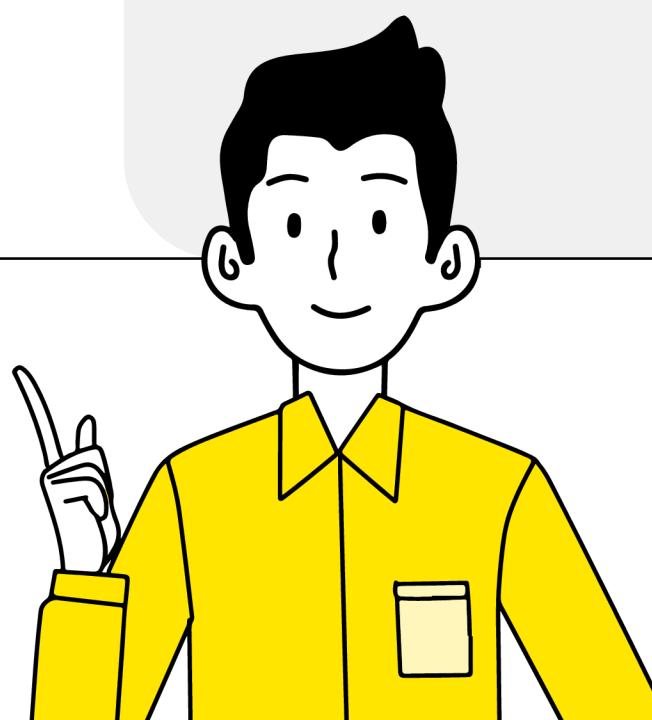


Réalisez un traitement dans un environnement Big Data sur le Cloud.



Notre mission:

**Prendre en main l'environnement AWS
pour préparer l'évolution de Fruits!**



Les étapes d'analyse

Partie 1

Prise en main des données

Partie 2

Mise en place de l'environnement

Partie 3

Ma situation

Partie 4

Conclusion



Partie 1: **Prise en main des** **données**

Partie 1: Prise en main

Les Fichiers

- **Note Book d'un alternant :**

Beaucoup de détails et présentation de la démarche à faire.

- **Dossier d'images :**

Contenant plusieurs dossiers de différents fruits avec des photographies sous tous les angles du fruit.

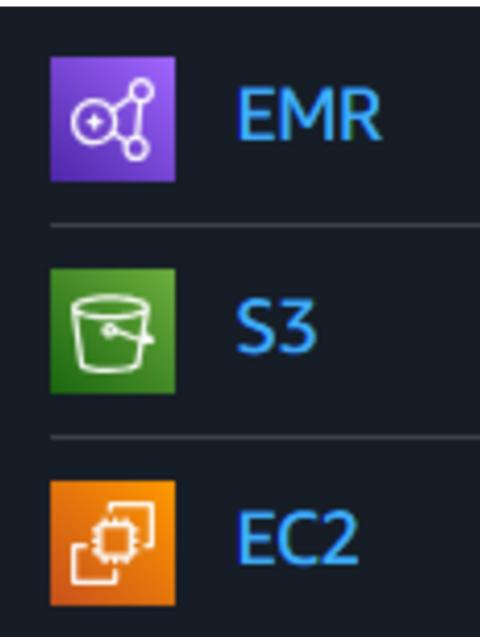
PySpark



Partie 1: Prise en main

Outils

- **Spark :** Apache Spark est un **moteur de traitement** distribué open source conçu pour **traiter rapidement de grandes quantités de données** en parallèle sur un cluster, tout en fournissant des **interfaces simples pour le développement de tâches d'analyse de données** en mémoire.
- **Amazon Web Services :**



Plateforme de cloud computing fournie par Amazon, offrant une vaste gamme de **services** comme **le stockage, la puissance de calcul, l'analyse, et l'intelligence artificielle**, permettant aux entreprises et aux développeurs de créer et de **gérer des applications à l'échelle mondiale sans se soucier de l'infrastructure matérielle**.



Partie 1: Prise en main

Démarche

CHAINE DE TRAITEMENT PySPARK

- 1. Création de la Spark Session et SparkContext :** spark = SparkSession.getOrCreate() sc = spark.sparkContext
- 2. Chargement des données dans un Dataframe Spark :** images = spark.read()
- 3. Préparation du modèle pour l'extraction de features et diffusion des poids aux workers :**
model = MobileNetV2() new_model = Model(inputs=model.input, outputs=model.layers[-2].output)
sc.broadcast(new_model.get_weights())
- 4. Crédit à une fonction de preprocessing et d'extraction de features en utilisant une Pandas UDF :**
@pandas_udf() décorateur qui rend la fonction compatible avec Spark et optimisée pour le traitement parallèle via Pandas
- 5. Réduction de dimensions pour garder 95% de la variance :** features_pca_df = pca.transform(features_df)
- 6. Enregistrement des features au format Parquet :** features_pca_df.write.parquet()





Partie 2: Mise en place de l'environnement.

Partie 2: Mise en place de l'environnement.

Etape 1

Choix de la solution technique.

Etape 2

Choix de la solution de stockage.

Etape 3

Mise en place.

Etape 1

Solutions techniques.

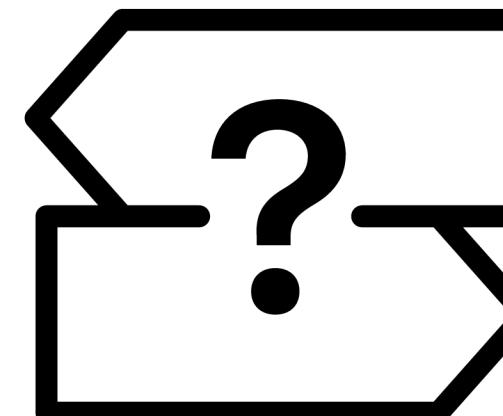
Solution IAAS

Avantage :

- Liberté totale de mise en œuvre de la solution
- Facilité de mise en œuvre à partir d'un modèle qui s'exécute en local sur une machine Linux

Inconvénients :

- Cronophage
- Nécessité d'installer et de configurer toute la solution
- Possible problèmes techniques
- Solution non pérenne dans le temps.



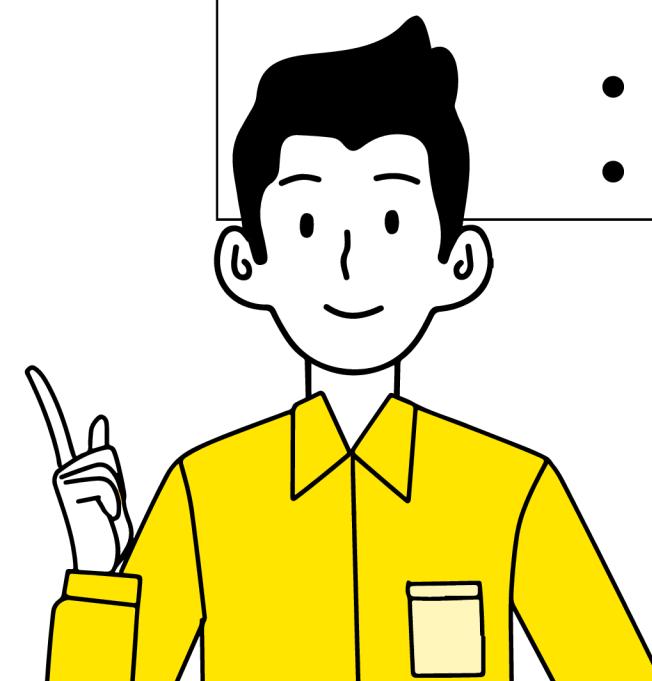
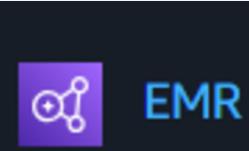
Solution PAAS

Avantage :

- Facilité et rapidité de mise en œuvre
- Il suffit de peu de configuration pour obtenir un environnement fonctionnel
- Il est très facile de recréer des clusters à l'identique.
- etc ...

Inconvénients :

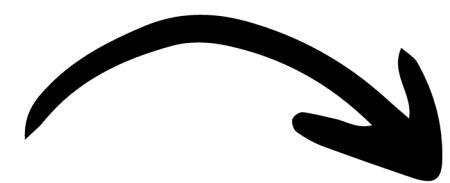
- manque de liberté sur la version des packages disponibles.



Etape 2

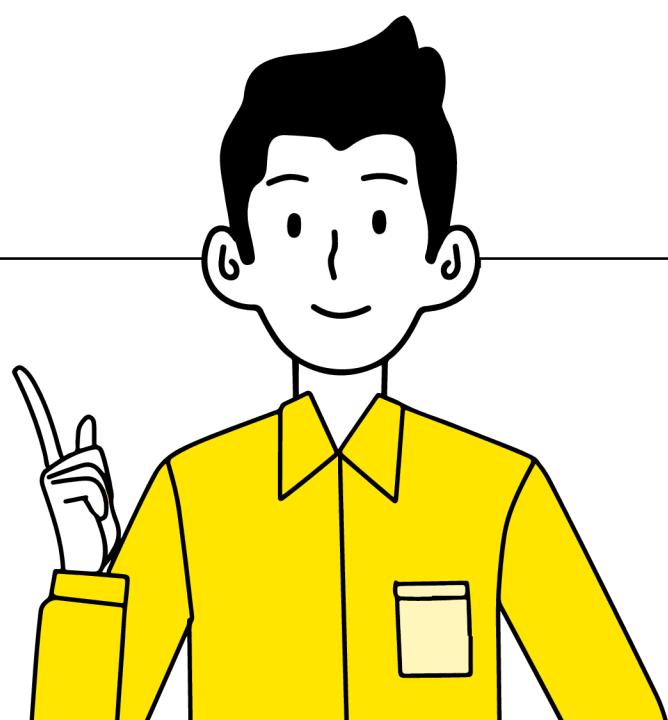
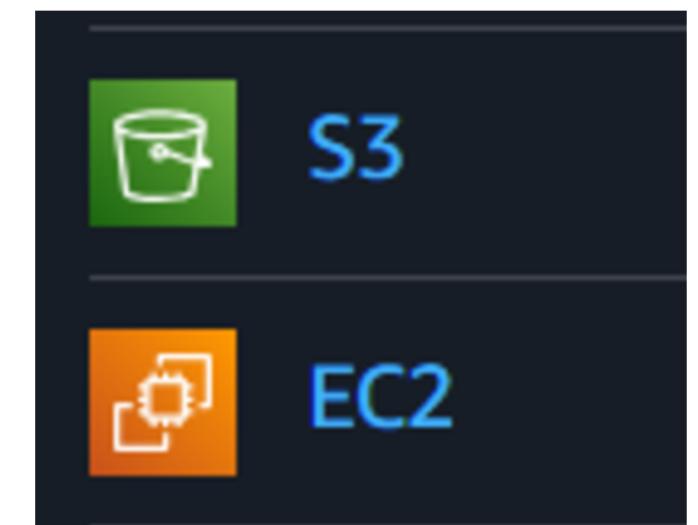
Solutions de stockages

Amazon S3 avec EC2



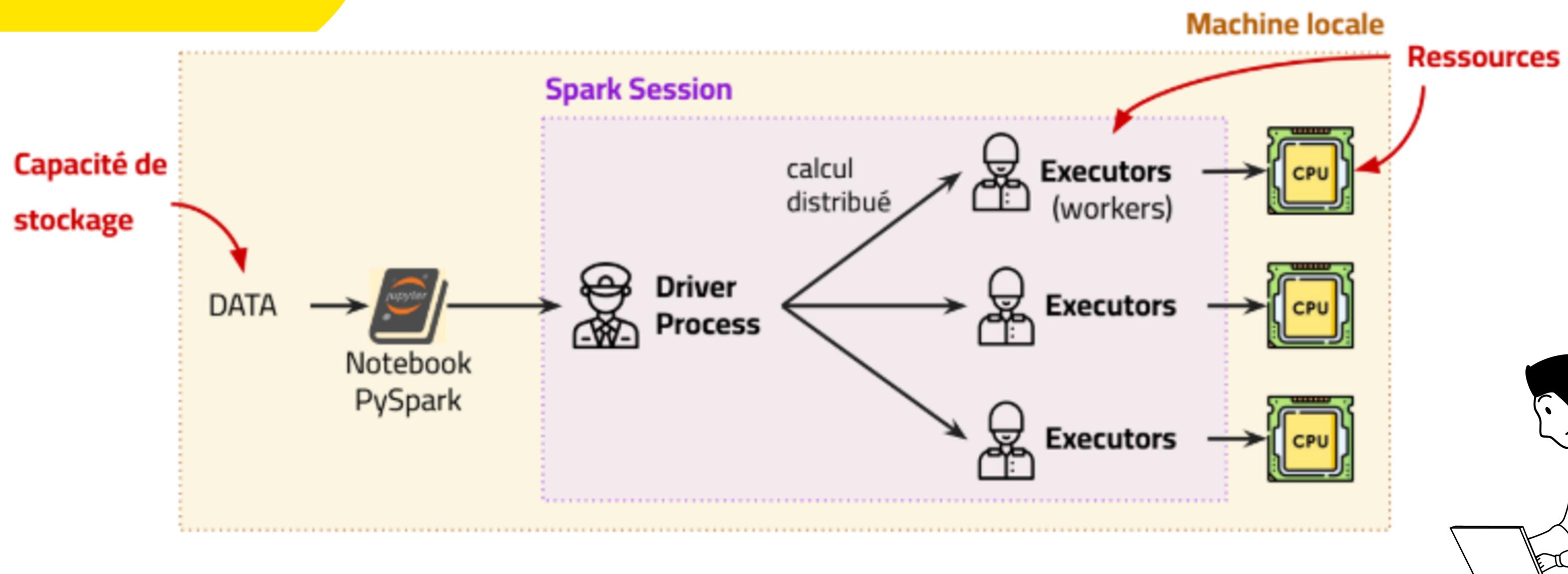
Sécurité et connexion aux applications.

Stockage des données



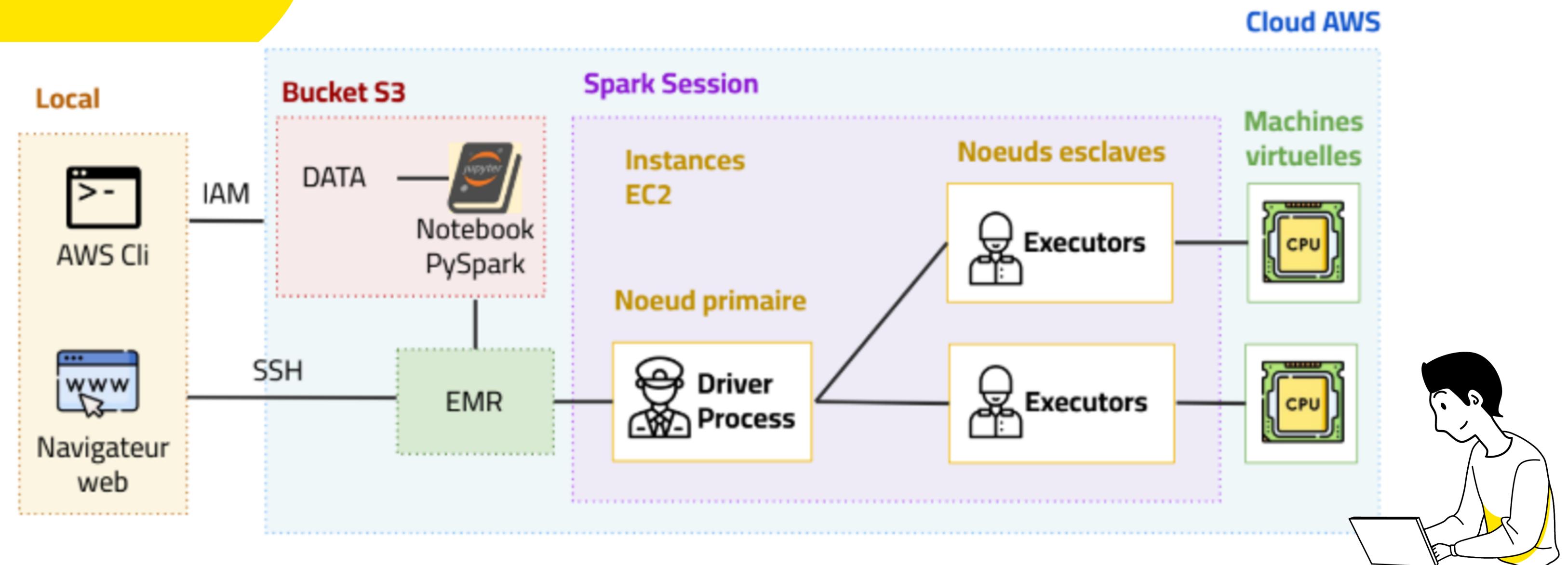
Partie 2: Mise en place de l'environnement.

Pourquoi faire ça
en plus de Spark?



Partie 2: Mise en place de l'environnement.

Pourquoi faire ça
en plus de Spark?

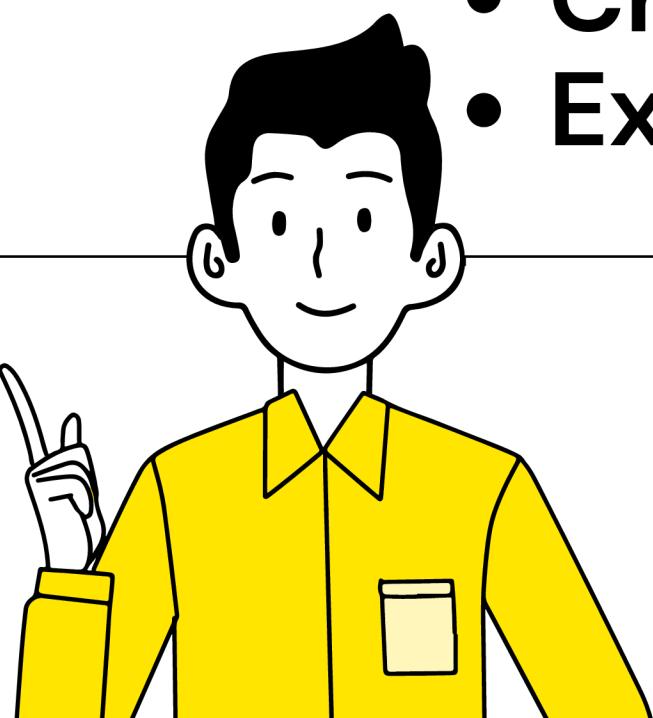


Etape 3

Mise en place

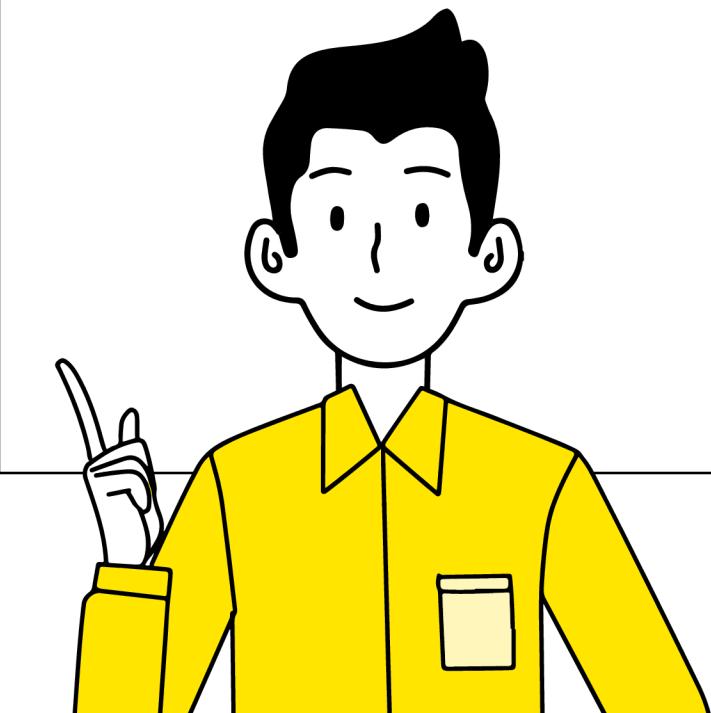
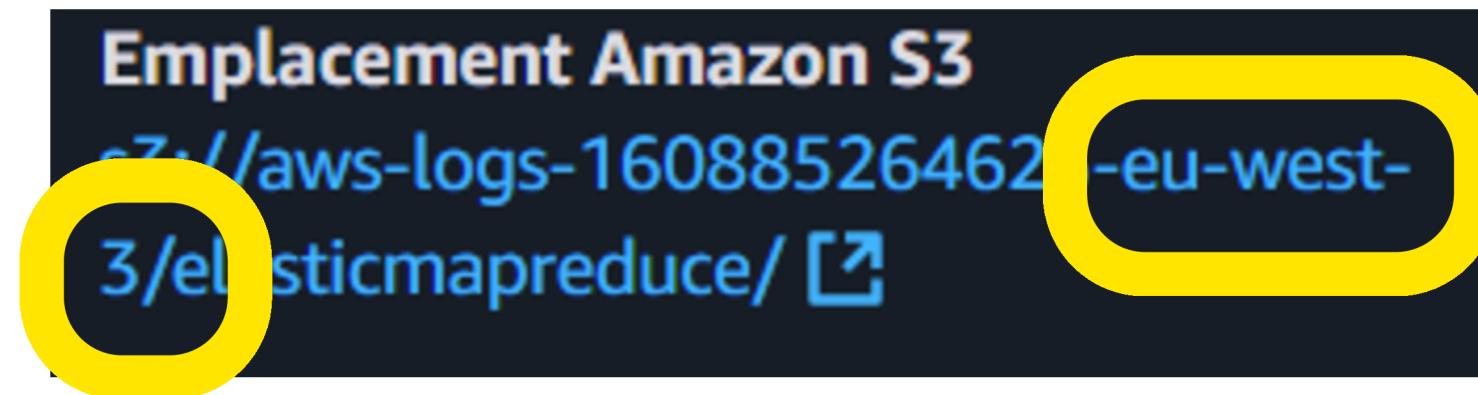
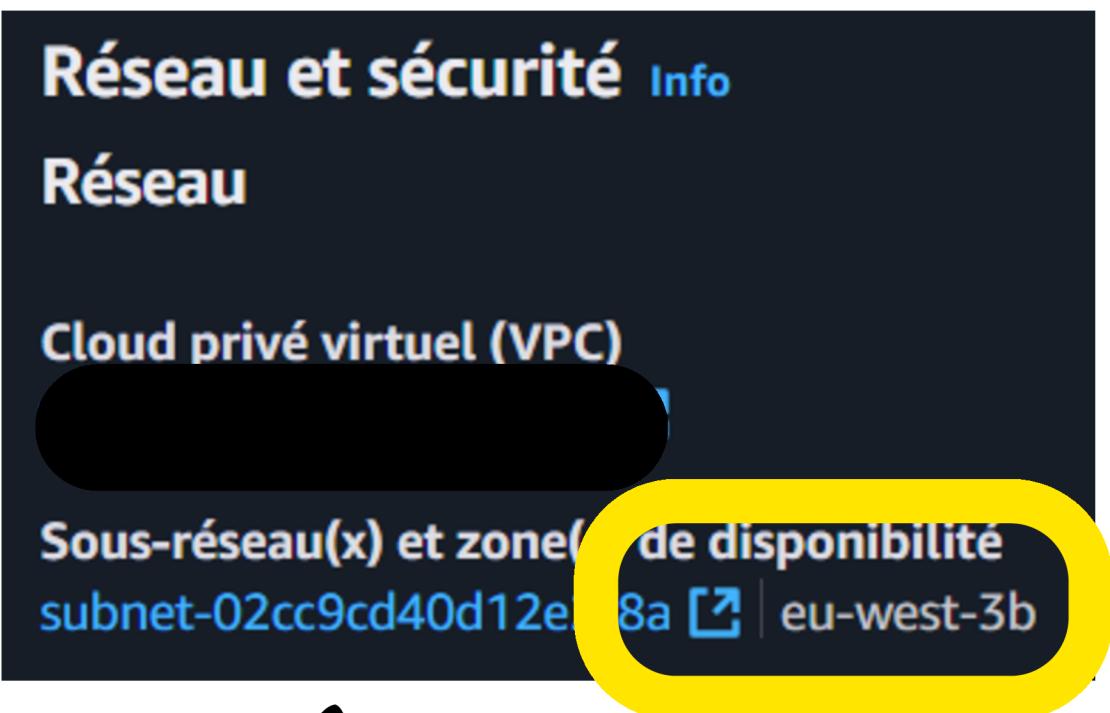
Points clés :

- **Upload des données sur le S3.**
- **Configurer l'EMR :**
 - Création d'un cluster
 - Configurer les logiciels
 - Matériel
 - Connexion au données (S3)
 - Action d'amorçage (genre de requirements.txt)
 - Création de clés Sécurités
- **Création du tunnel SSH.**
- **Exécution du code.**



Etape 3

RGPD : emplacement en Europe

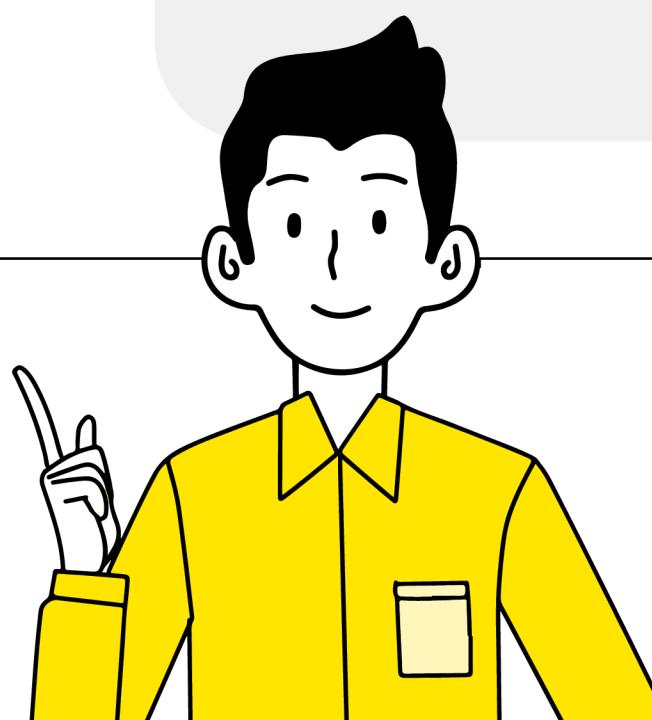




Partie 3: Ma situation.

Malgré le matériel adapté, j'ai malheureusement rencontré des difficultés de mise en place.

Voici donc un compte-rendu des alternatives que j'ai tenté de mettre en place.

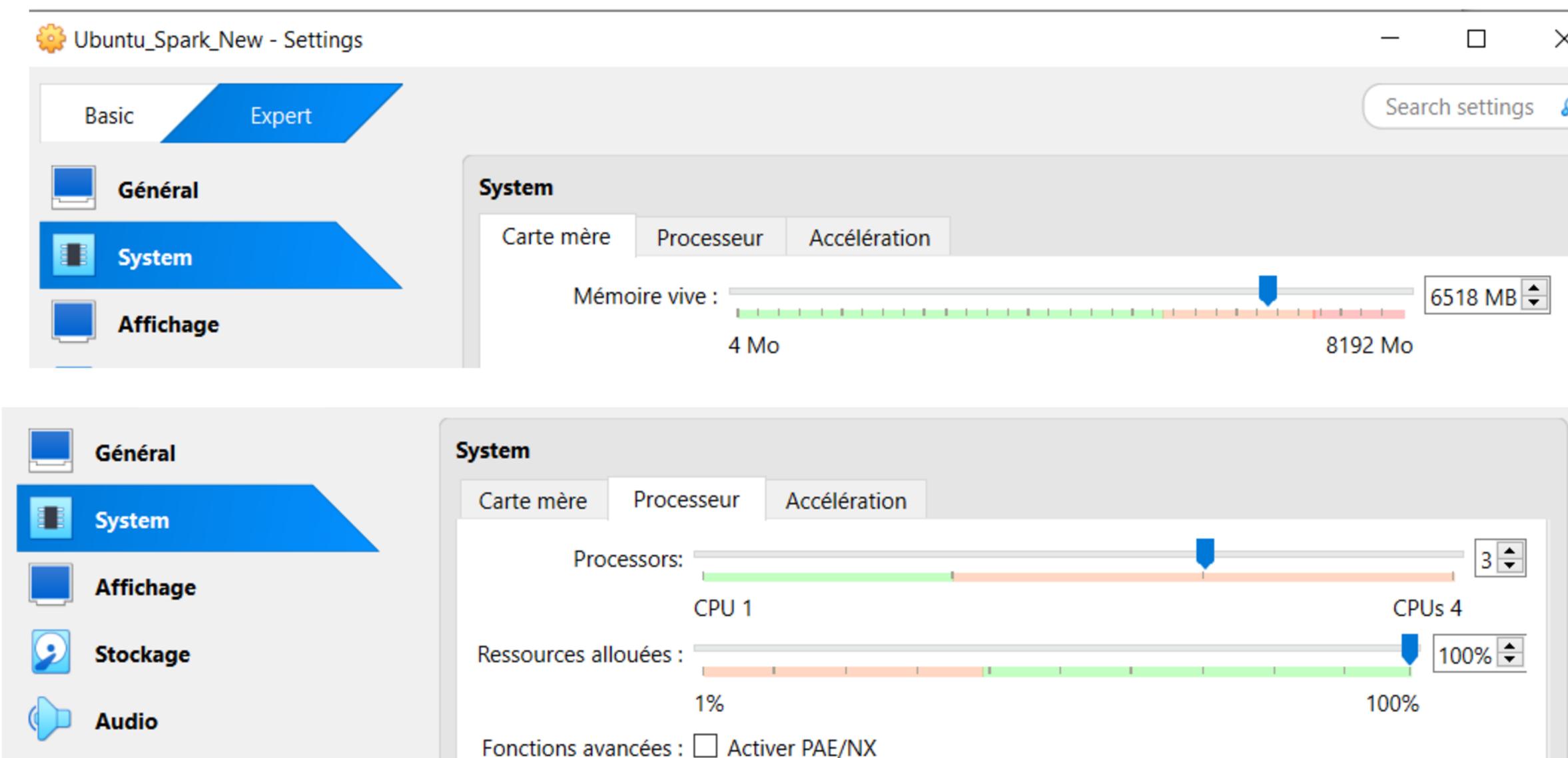


Partie 3: Ma situation

Environnement

Étape Notebook : 3 et suite

Création de la machine virtuelle sous Linux



Ma configuration



Partie 3: Ma situation

Environnement

Étape Notebook : 3 et suite

20 nov. 14:56

P8_Notebook_Linux_EMR_PySpark_V2.ipynb - P8_Mode_operatoire - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

P8_MODE_OPERATOIRE

- > data
- > fruits-360_dataset
- > img
- > myenv
- > myenv39
- > spark-3.4.4-bin-hadoop3
- > spark-3.5.3-bin-hadoop3
- ClefProjet11.pem
- P8_Notebook_Linux_EMR_PySpark...
- P8_Notebook_Linux_EMR_PySpark...
- spark-3.4.4-bin-hadoop3.tgz
- spark-3.5.3-bin-hadoop3.tgz

Code + Markdown | Run All | Restart | Clear All Outputs | Go To | Variables | Outline ...

Uninstalling numpy-1.19.5:
Successfully uninstalled numpy-1.19.5
Successfully installed numpy-2.0.2
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

3.4 Import des librairies

```
import pandas as pd
from PIL import Image
import numpy as np
import io
import os
```

[1] ✓ 15.5s Python

```
import tensorflow as tf
```

[2] ✘ Python

The Kernel crashed while executing code in the current cell or a previous cell.
Please review the code in the cell(s) to identify a possible cause of the failure.
Click [here](#) for more info.
View Jupyter log for further details.

```
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras import Model
```

[] Python

```
from pyspark.sql.functions import col, pandas_udf, PandasUDFType, element_at, split
from pyspark.sql import SparkSession
```

Spaces: 4 Cell 12 of 91

Paramètres

ctrl droite

Partie 3: Ma situation

Environnement

Étape Notebook : 3 et suite

Situation:

- Impossible d'importer TensorFlow sur la machine virtuelle.
- Spark ne fonctionne pas sous Windows.

Solution mis en place :

- Utilisation de Google Colab et Google Drive
- Mise en place du PCA sur la solution local
- Import des données et sauvegarde du traitement sur Google Drive



Partie 3: Ma situation

AWS

Étape Notebook : 4.10

Les étapes précédentes sont suivies pour être comprise.

Retour sous Linux :

- 4.8.2 : Crédit du tunnel SSH : EMR ne s'affiche pas.
- Crédit du FoxyProxy
- Connexion au Jupyterhub

Ici, je fais une erreur



Partie 3: Ma situation

AWS

Ce qui cause mon erreur.

- Interface de connexion aux applications différentes.
- Ordinateur en difficulté.
- Chargement des données qui ne se fait pas.
- Utilisation de la même session Spark que le local.



Je ne prends pas de screen pensant que cela vient de ma démarche différente du projet.



Partie 3: Ma situation

AWS

- Retour sous Windows et changement de Clusters pour changer la clé EC2.
- Tentative de connexions avec Google Colab et AWS. Impossible avec l'état de JAVA

Recherche de solution

```
!java -version
openjdk version "11.0.25" 2024-10-15
OpenJDK Runtime Environment (build 11.0.25+9-post-Ubuntu-1ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.25+9-post-Ubuntu-1ubuntu122.04, mixed mode, sharing)

from pyspark import SparkConf
from pyspark.sql import SparkSession

# Update master to 'yarn' for EMR cluster
conf = SparkConf().setAppName('Colab to EMR') \
    .setMaster('yarn') \
    .set('spark.hadoop.fs.s3a.access.key', aws_access_key) \
    .set('spark.hadoop.fs.s3a.secret.key', aws_secret_key) \
    .set('spark.yarn.stagingDir', 's3://projet11/Test/') \
    .set("spark.executor.memory", "2g")\
    .set("spark.driver.memory", "2g")

spark = SparkSession.builder.config(conf=conf).getOrCreate()

PySparkRuntimeError                                     Traceback (most recent call last)
<ipython-input-16-9784d3300087> in <cell line: 14>()
     12
     13
--> 14 spark = SparkSession.builder.config(conf=conf).getOrCreate()

----- 4 frames -----
/usr/local/lib/python3.10/dist-packages/pyspark/java_gateway.py in launch_gateway(conf, popen_kwargs)
    105
    106          if not os.path.isfile(conn_info_file):
--> 107              raise PySparkRuntimeError(
    108                  error_class="JAVA_GATEWAY_EXITED",
    109                  message_parameters={},
PySparkRuntimeError: [JAVA_GATEWAY_EXITED] Java gateway process exited before sending its port number.
```



Partie 3: Ma situation

AWS

- Je sais que AWS est pensé pour justement régler ce problème de matériel et d'environnement.
- En suivant les instruction AWS pour Windows le tunnel EMR est établie.
Avec l'interface d'application AWS.
- Sans arriver à me connecter à Jupyterhub.

Recherche de solution

```
hadoop@ip-172-31-19-183:~  
Using username "hadoop".  
Authenticating with public key "Projet11Win"  
  
A newer release of "Amazon Linux" is available.  
Version 2023.6.20241111:  
Run "/usr/bin/dnf check-release-update" for full release and version update info  
#  
~\ _###_ Amazon Linux 2023  
~~ \####\ https://aws.amazon.com/linux/amazon-linux-2023  
~~ \##|  
~~ \#/ V~'-->  
~~ /  
~~ .-/ /  
/m/  
Last login: Wed Nov 20 15:22:00 2024  
  
EEEEEEEEEEEEEEEEEE MMMMMMM M::::::M R:::::R R:::::R  
E:::::::E M:::::::M M:::::::M R:::::R R:::::R  
EE:::::E EEEEEEE M:::::::M M:::::::M R:::::RRRRRR:::::R  
E:::::E EEEEEEE M:::::::M M:::::::M R:::::R R:::::R  
E:::::E EEEEEEE M:::::::M M:::::::M R:::::R R:::::R  
E:::::E EEEEEEE M:::::::M M:::::::M R:::::RRRRRR:::::R  
E:::::E EEEEEEE M:::::::M M:::::::M R:::::R R:::::R  
EE:::::E EEEEEEE M:::::::M M:::::::M R:::::R R:::::R  
E:::::E EEEEEEE M:::::::M M:::::::M R:::::R R:::::R  
EEEEEEEEEEEEEEEEEE MMMMMMM RRRRRRRRRRRRRR  
M:::::::M R:::::R R:::::R  
M:::::::M R:::::RRRRRR:::::R  
M:::::::M RR:::::R R:::::R  
M:::::::M R:::::R R:::::R  
M:::::::M R:::::RRRRRR:::::R  
M:::::::M R:::::R R:::::R  
M:::::::M R:::::R R:::::R  
M:::::::M RRRRRRR  
RRRRRRR
```



Partie 3: Ma situation

AWS

Recherche de solution

- Je sais qu'AWS est pensé pour justement régler ce problème de matériel et d'environnement. **Alors, je retourne sous LINUX pour recréer et déboguer une session Spark complète.**
- Clone le cluster, change les clés EC2, fait des fausses manipulations, rencontre quelques difficultés avec FoxyProxy et tombe par hasard sur mon estimation de facturation.



Je ne peux pas continuer.



Partie 4: Conclusion

Partie 4 Conclusion

Local :

- La démarche en local a dû être faite via Google Colab.
- J'ai mon dossier Results au format "parquet" en local.
- La réduction PCA a été intégrée sur l'étape locale.

AWS :

- J'ai fait des erreurs dans une situation de base qui n'était pas propice à la bonne réalisation du projet.
- J'ai compris l'intérêt d'AWS et de Spark.
- Ma situation personnelle ne permet pas de continuer à tester et trouver des solutions.



Merci!

