

Déployez un modèle dans le cloud

Projet 8: Moustapha ABDELLAHI

OPENCLASSROOMS

PLAN

- **Problématique et jeu des données**
- **Le Big Data**
- **Architecture retenue et chaîne de traitement**
- **Conclusions**

PROBLEMATIQUE



Fruits!

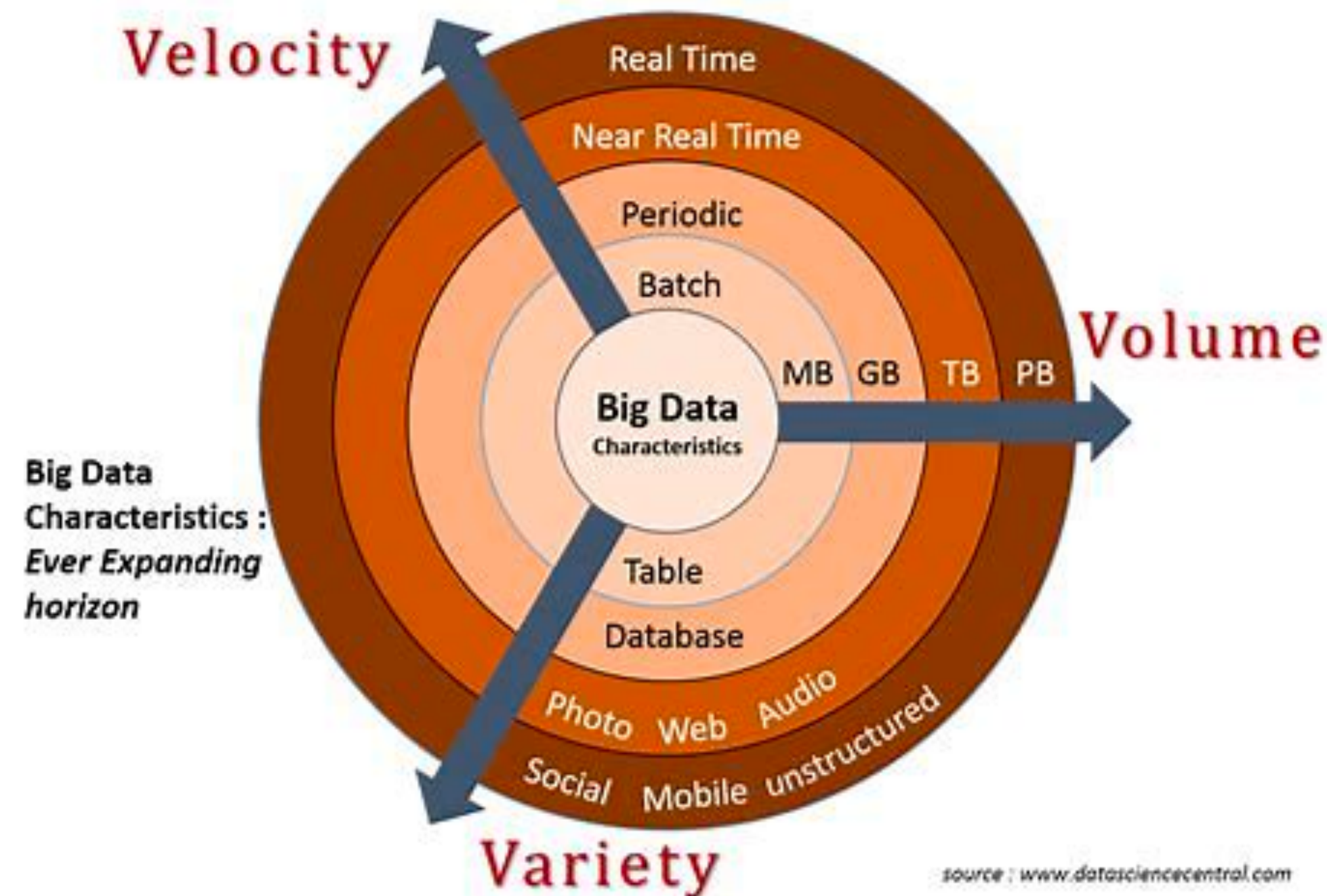
- **CONTEXTE:**
 - Une startup de l'Agri-Tech, nommée 'Fruits ' cherche à proposer des solutions innovantes pour la récoltes des fruits (robot cueilleur intelligents)
 - Mettre à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.
- **MISSION:**
 - Développer dans un environnement Big Data une chaîne de traitement des données comprenant le preprocessing et une étape de réduction de dimension
- **OBJECTIF:**
 - Anticiper le passage à l'échelle en termes de volume de données

- **ORIGINE :**
- Images de 131 variétés de fruits et labels associés (Fruits 360)
- Plusieurs variétés du même fruit et labels associés (exemple : pomme « red » et « golden »)
- **CARACTERISTIQUES DES DATASETS:**
- Images 100x100 JPEG RGB
- Photos sur fond blanc centrées sur le fruit
- Photos sous tous les angles (rotation tri-axiales)
- Nombre total : 90483 images
- Jeu d'entraînement : 67 692 images



QU'EST-CE QUE LE BIG DATA?

- Le Big Data : les mégadonnées ou les données massives, désigne les ressources d'informations qui peuvent être représentées par les « 3V » suivants:
- **Volume** : énormes quantités de données pour être traitées sur une seule machine (dépassant la capacité de la RAM et celle du stockage)
- **Variété** : différents types de données
- **Vélocité** : vitesse de circulation des données (latence à minimiser)



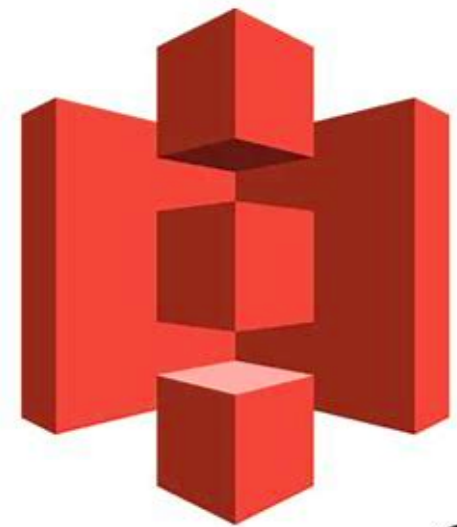
QUELLES REPONSES AUX ENJEUX DU BIG DATA?

- **STOCKAGE DISTRIBUE:** Système de fichiers distribué (ex. HDFS)
- **Volume:** passage à l'échelle possible
- **Variété:** capacité d'évoluer
- **Vélocité:** partitionnement

INFRASTRUCTURE DISTRIBUEE

STOCKAGE**CALCUL**

• SOLUTIONS DE STOCKAGE



Amazon S3



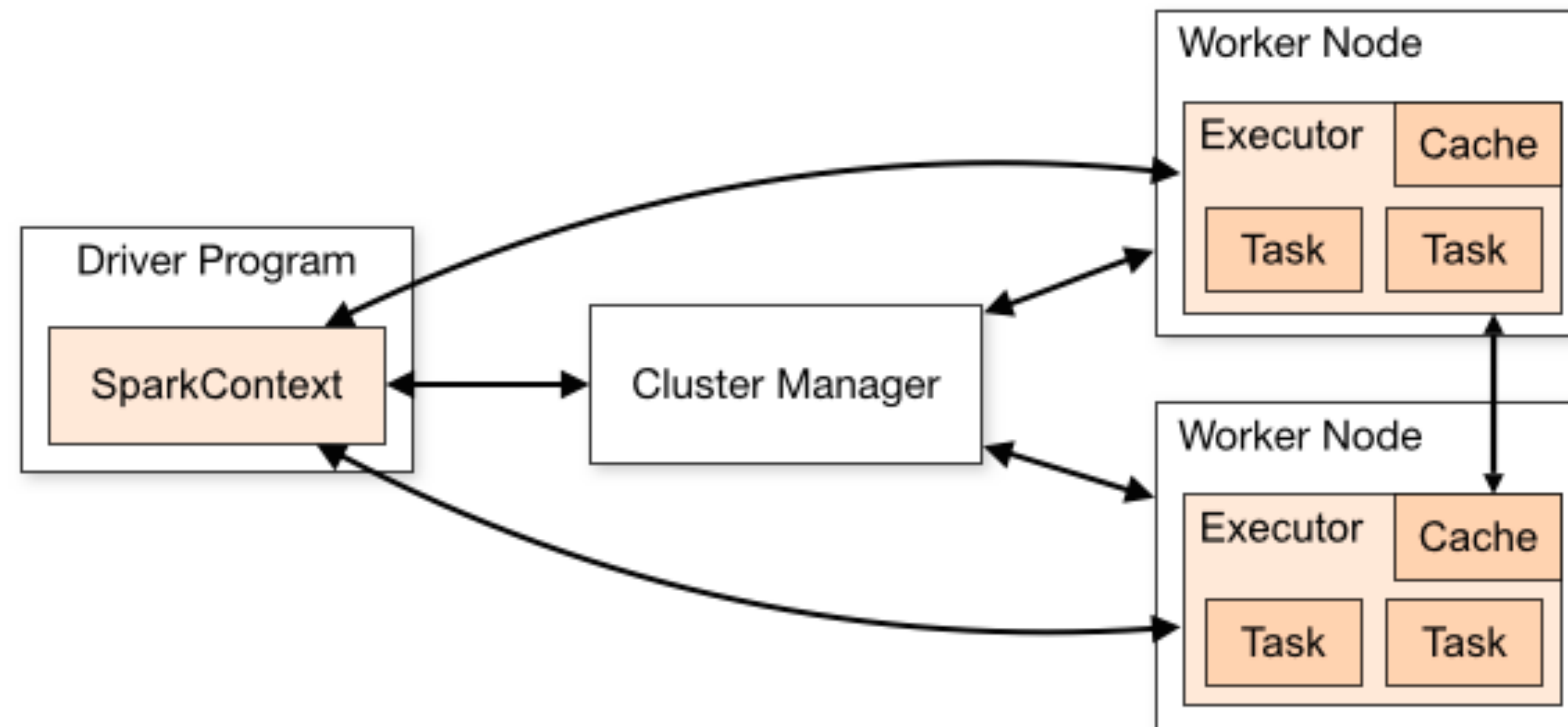
Azure Blob
Storage



Google
Cloud
Storage

- **CLUSTER DE CALCUL DISTRIBUE:**

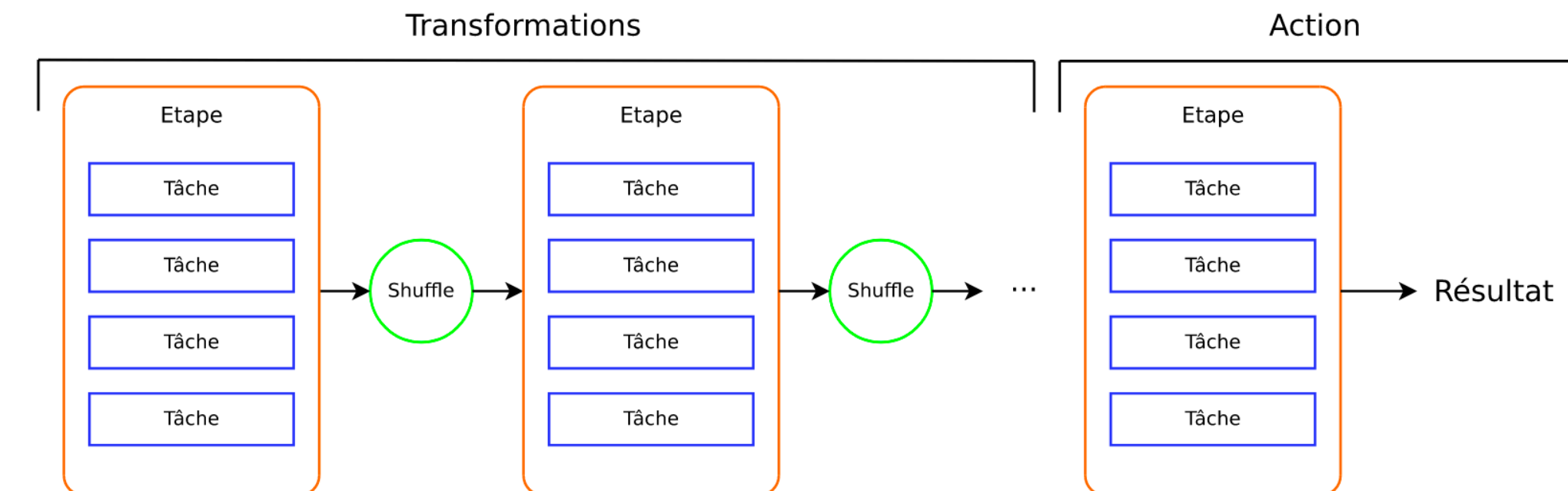
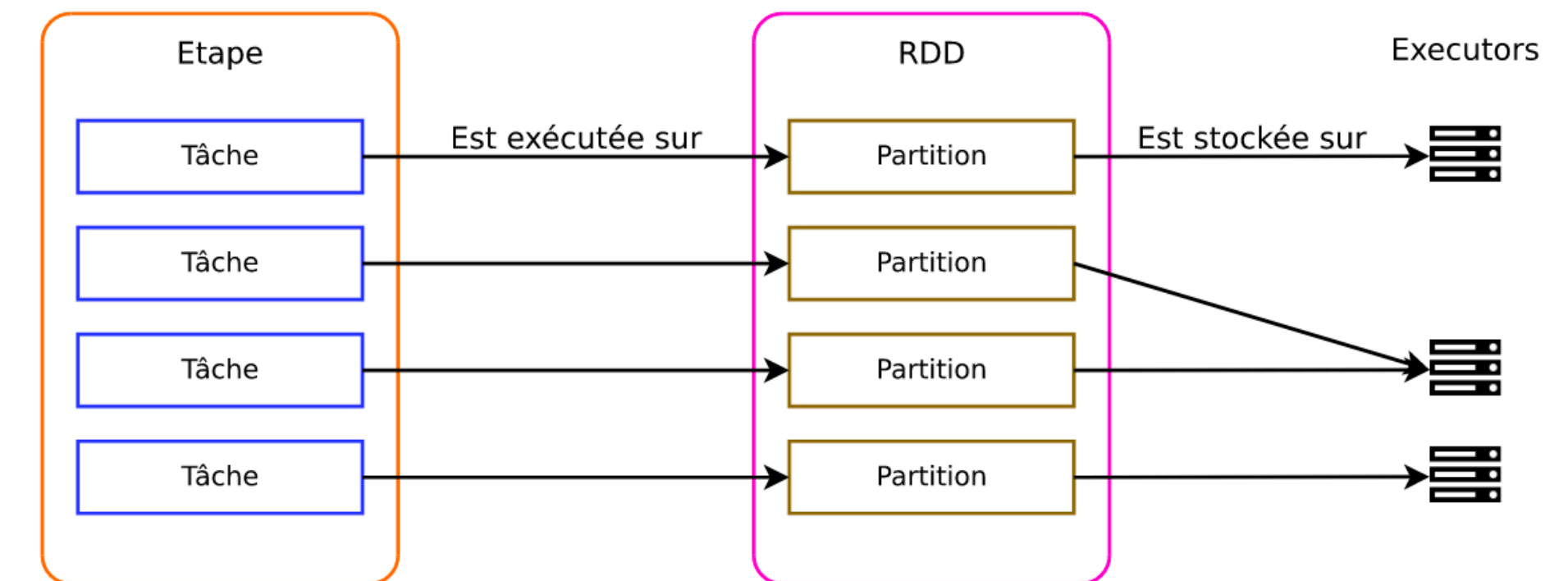
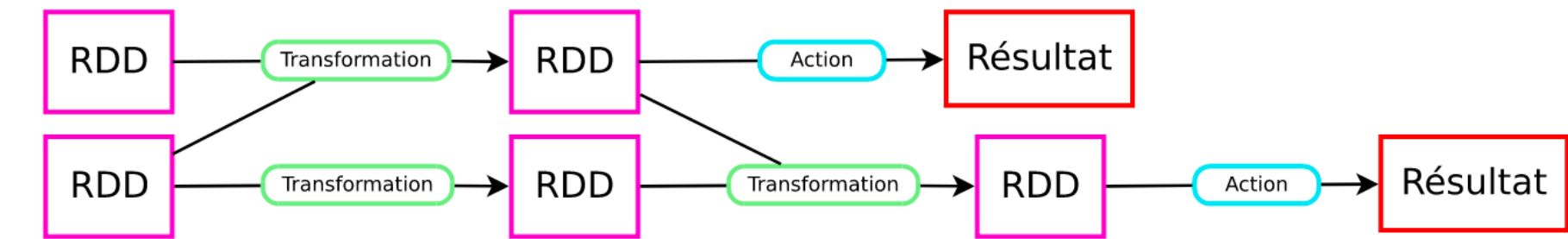
- Diviser les opérations en micro-opérations distribuables entre plusieurs machines, exécutables en parallèle
- Agréger les résultats sur une même machine



- **Driver Program** : Configuration / initialisation / Agrégation des calcul
- **Cluster Manager** : Gestion des ressources / Distribution des calculs entre les workers
- **Workers** : Exécution des tâches en parallèle

• CALCUL DISTRIBUE:

- **RDD (RESILIENT DISTRIBUTED DATASETS)** : servent à réaliser des calculs parallèles en mémoire sur un cluster en tolérant les pannes
- **Job Spark** : ensemble d'étapes séparées par des shuffles
- **Etape** : ensemble de tâches
- **Shuffle** : redistribution des données entre les noeuds
- **Tâche** : s'exécute sur une **partition** différente des données
- **Partitions** : sont créées par les RDD et réparties sur les différents **exécuteurs**.



OBJECTIF DE LA MISSION:

Mettre en place les premières briques de traitement d'images qui serviront lorsqu'il faudra passer à l'échelle en termes de volume de données

Preprocessing

Réduction de dimension

ARCHITECTURE BIG DATA :



Stockage

**Stockage des
images initiales
et du résultat de
PCA**



Sécurité

**Clé d'accès et
rôle IAM
Politique IAM qui
permet l'accès à
S3**



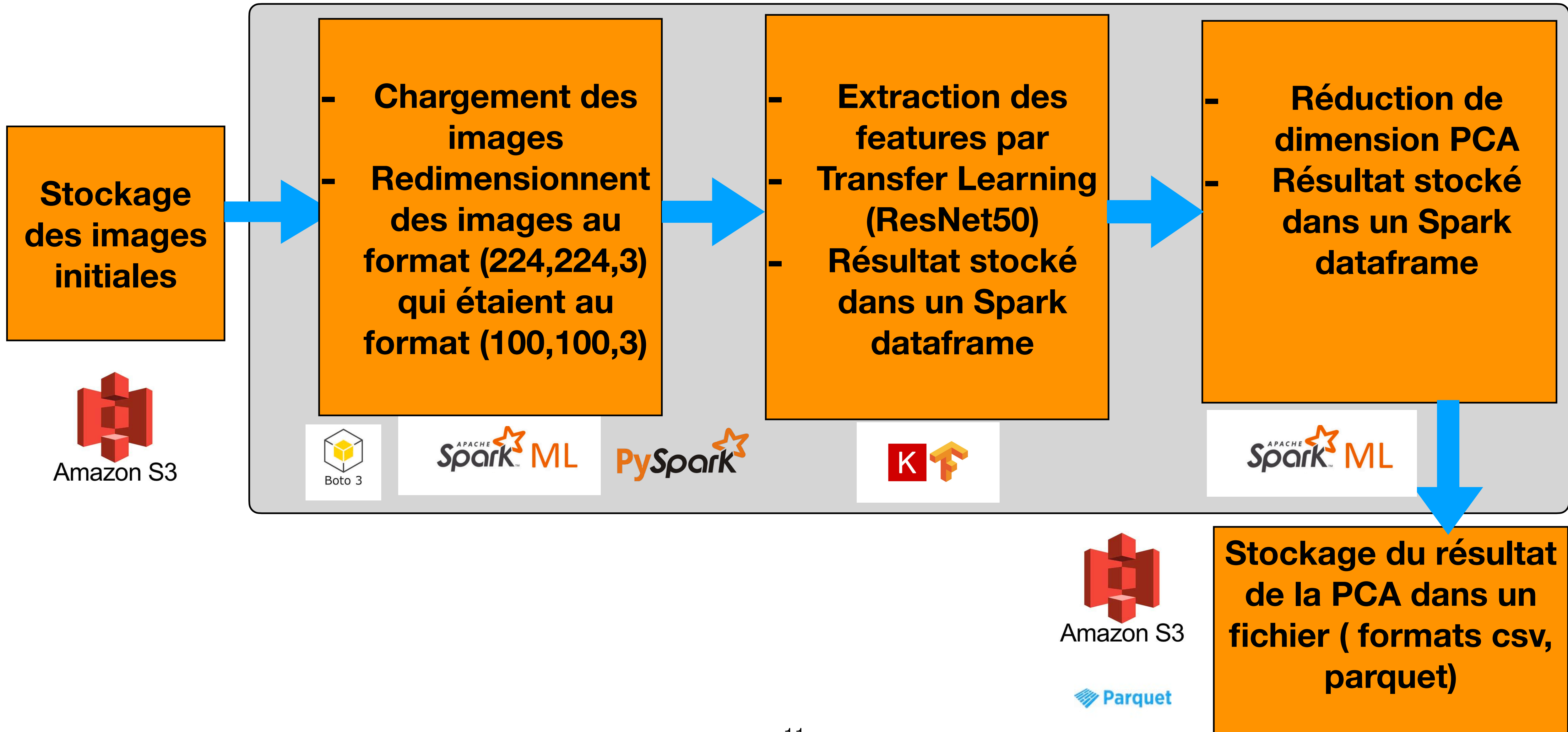
Traitement

**Exécution des
scripts dans
Jupyter notebook**

Accès SSH

**Accès sécurisé SSH à la
console du serveur EC2
(Ligne de commande
pour les installations)**

CHAÎNE DE TRAITEMENT :



RESNET50:

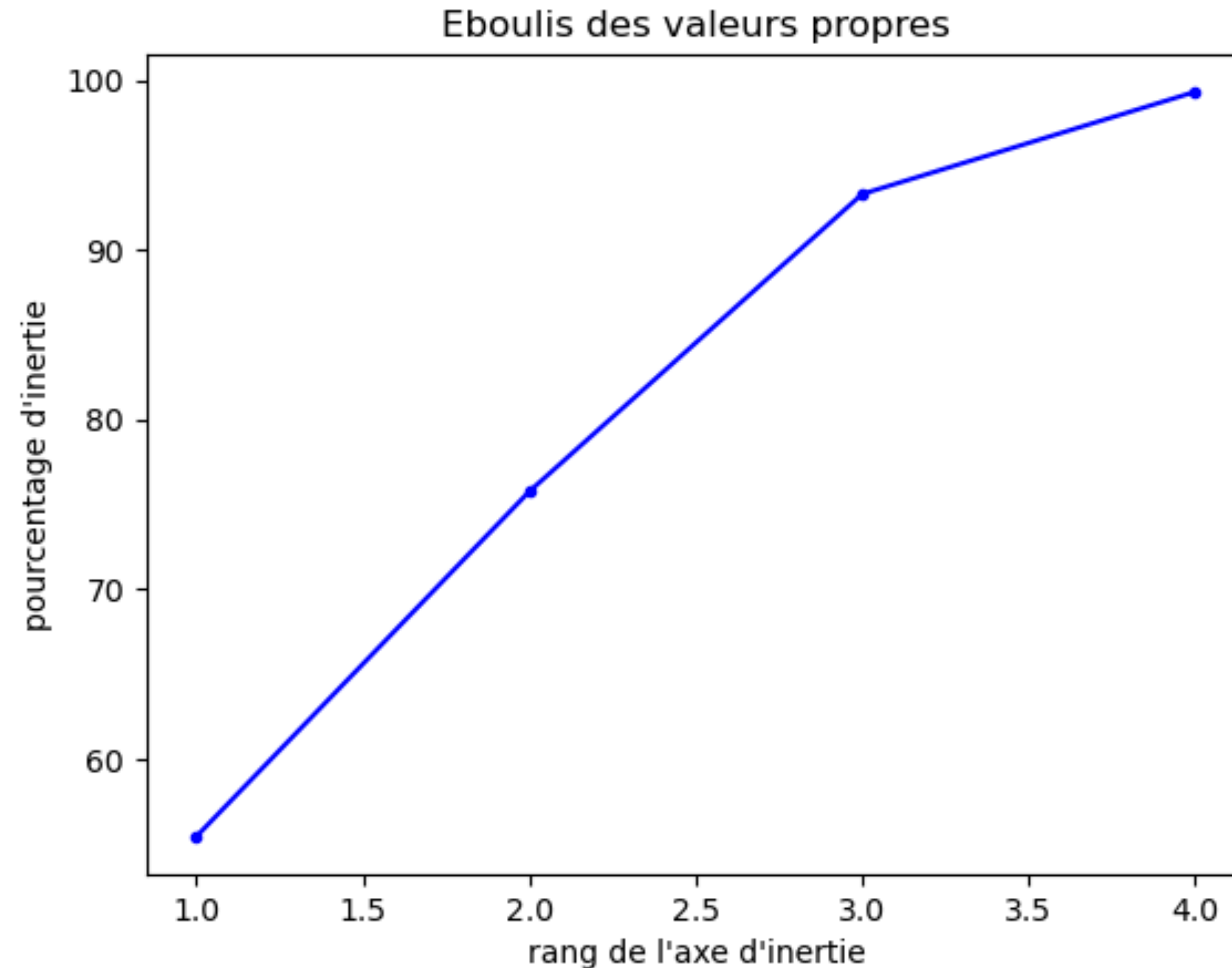
- ResNet50 est un réseau neuronal convolutif de 50 couches de profondeur.
- Il a été pré-entraîné sur plus d'un million d'images à partir de la base de données ImageNet.
- Le réseau pré-entraîné peut classer les images en 1000 catégories d'objets, telles que le clavier, la souris, et de nombreux animaux.
- Le réseau a une taille d'entrée d'image de 224 par 224.

PCA :

- Méthode largement utilisée en réduction de dimension
- Cherche à représenter les données dans un sous-espace de plus petite dimension
- Conserve au maximum la variance du nuage de données.

RESULTAT DE LA PCA

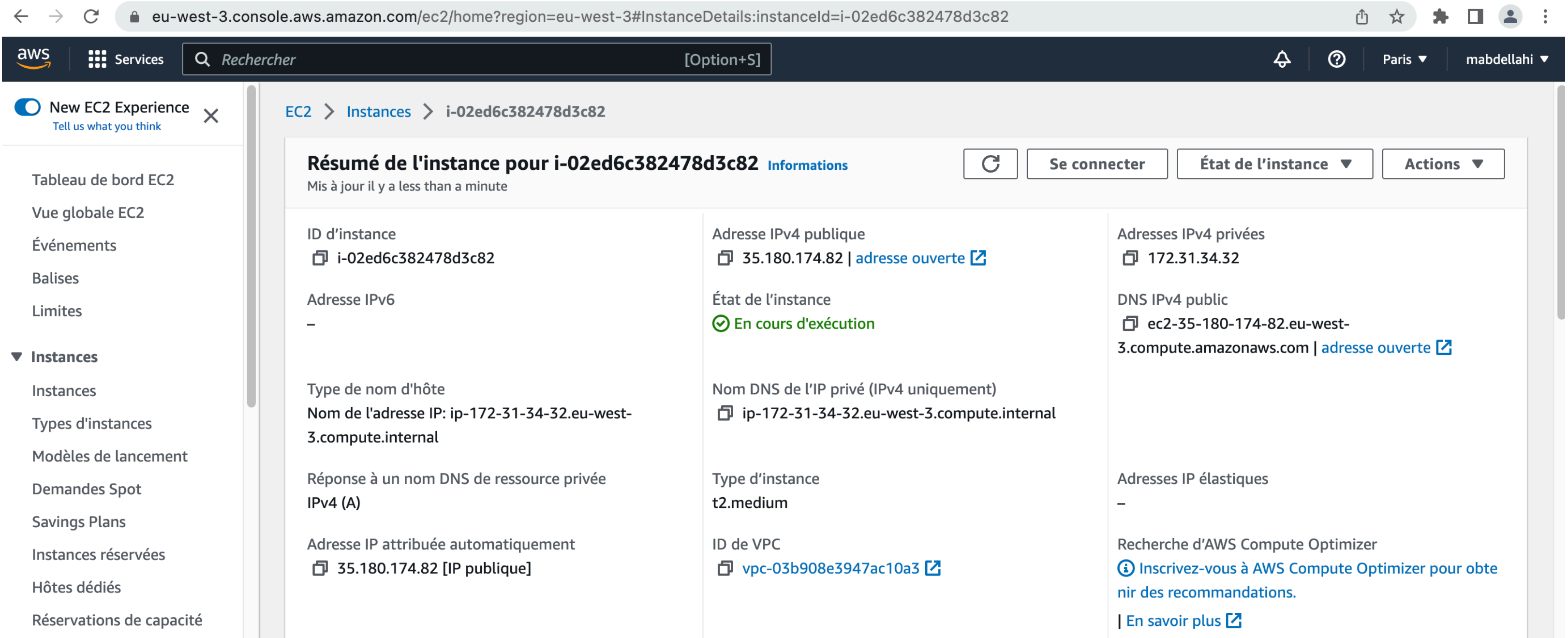
- **Avant PCA**
 - Dataset des features des image : **(10, 100352)**
- **Après PCA**
 - Dataset des features des image : **(10, 4)**



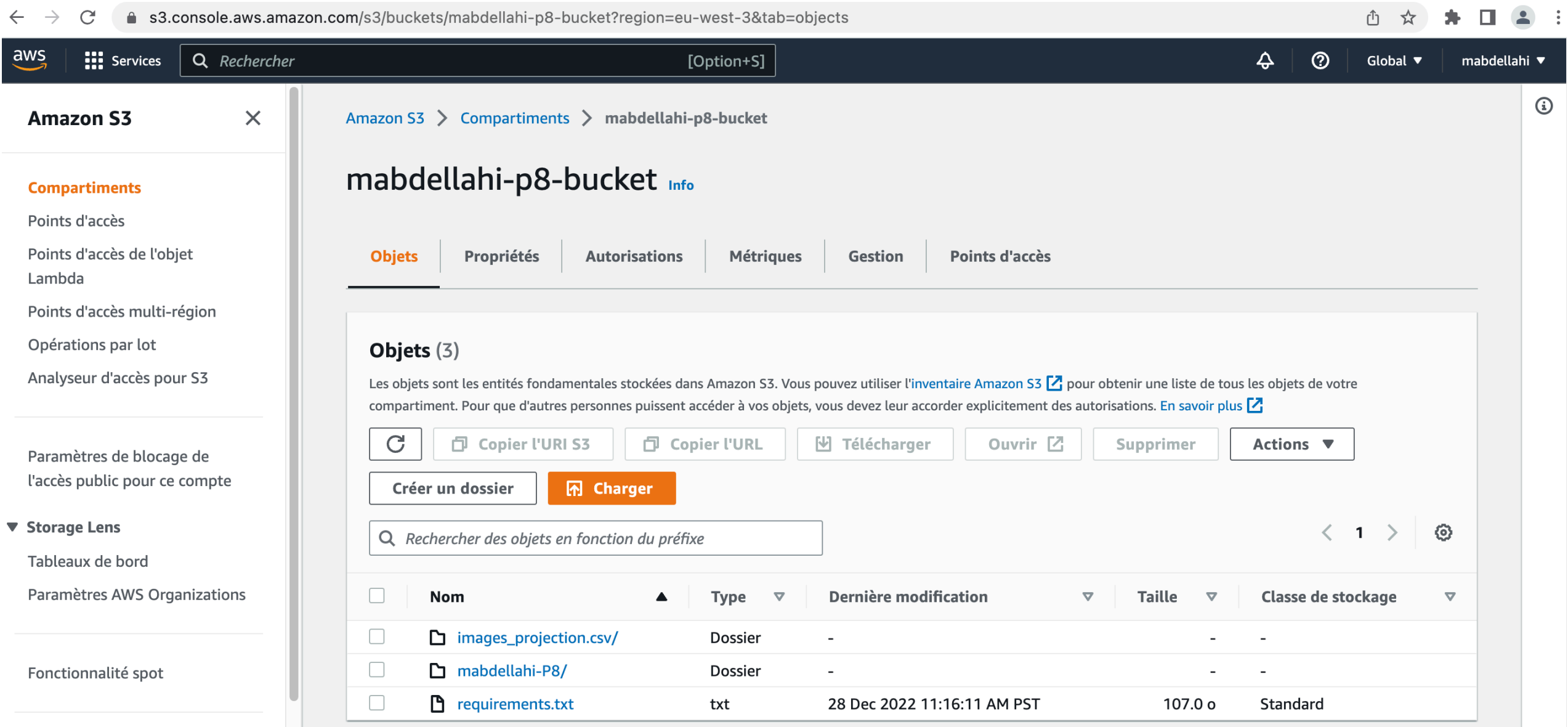
CHAINE DE TRAITEMENT

DES CAPTURES D'ECRAN :

EC2 Instance



S3 Bucket



DES CAPTURES D'ECRAN :

Jupyter notebook sur EC2

Non sécurisé | https://35.180.174.82:8888/notebooks/ABDELLAHI_Moustapha_1_notebook_012023.ipynb

jupyter ABDELLAHI_Moustapha_1_notebook_012023 Dernière Sauvegarde : mardi dernier à 15:06 (auto-sauvegardé) Logout

File Edit View Insert Cell Kernel Widgets Help

Python 3 (ipykernel)

Entrée [23]:

```
conf = SparkConf()
conf.set('spark.jars.packages', 'org.apache.hadoop:hadoop-aws:3.3.2')
#conf.set('spark.hadoop.fs.s3a.aws.credentials.provider', 'org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider')
conf.set('spark.executor.extraJavaOptions', '-Dcom.amazonaws.services.s3.enableV4=true')
conf.set('spark.driver.extraJavaOptions', '-Dcom.amazonaws.services.s3.enableV4=true')
conf.set('spark.hadoop.fs.s3a.access.key', access_key)
conf.set('spark.hadoop.fs.s3a.secret.key', secret_key)
conf.set('spark.hadoop.fs.s3a.impl', "org.apache.hadoop.fs.s3a.S3AFileSystem")
conf.set('fs.s3a.endpoint', 's3-eu-west-3.amazonaws.com')
#conf.set('spark.hadoop.fs.s3a.session.token', token)
```

Out[23]: <pyspark.conf.SparkConf at 0x7f02fb861a90>

Initialization of Spark instance

Entrée [24]:

```
# Initiate a Spark session
spark = SparkSession.builder.config(conf=conf).getOrCreate()
spark._sc.setSystemProperty("com.amazonaws.services.s3.enableV4", "true")
```

RECAPITULATIF DE L'INFRASTRUCTURE AWS UTILISEE

- **Stockage fichiers sur S3 :**
 - Upload à l'aide AWS CLI ou Interface Web
 - Lecture des fichiers depuis Spark
 - Enregistrement des fichiers depuis Spark dans S3
- **Instance EC2 (t2.medium, 4 Go RAM, 30 Go disque serveur, OS = ubuntu-bionic-18.04)**
- **Configuration : Python3 /Java 11.0.17/ SPARK / Hadoop 3.3.2/Boto3**
- **Configuration sur une machine distante : accès via SSH**
 - **Chargement clé IAM / AWS**
 - **Installation des logiciels et packages nécessaires**
 - **Création d'un Jupyter Notebook accessible à distance contenant les scripts en Pyspark exécutables**

COMMENT PASSER A L'ECHELLE?

- **Aucune de modification du code Spark/Python à faire**
- **Stockage de fichiers peut rester sur S3**
- **Choisir une instance EC2 de plus grande capacité RAM / processeur**
- **Choisir EMR instance, SageMaker ou Databricks**

Enseignements:

- **Prise en main de Pyspark**
- **Découverte de l'écosystème AWS**
- **Administration d'un serveur Linux par SSH**

Difficultés rencontrées:

- **Nombreuses possibilités techniques : choix complexes**
- **Problèmes de compatibilité entre les différents packages et logiciels**
- **Débug complexe à cause des erreurs de superpositions des versions de (Spark/Java/S3)**