PROJET 9 :

Réaliser un traitement dans un environment Big Data sur le Cloud

Octobre 2024

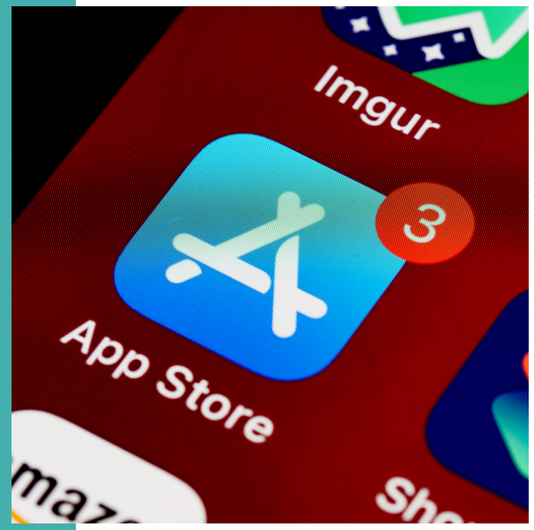
Lokman AALIOUI

Contexte

Notre start-up "Fruits!" est une entreprise dans le domaine de l'AgriTech. Nous développons des solutions qui préservent la biodiversité des fruits, en permettant des traitements spécifiques adaptés à chaque espèce, grâce à des robots cueilleurs intelligents.

Nous avons décidé de lancer une application mobile qui permettra aux utilisateurs d’identifier des fruits simplement en les photographiant.

Mission



- Modifier un notebook créé précédemment
- Construire une chaîne de traitement de données solide et évolutive dans un environnement Big Data AWS
- Développer des scripts en PySpark et mettre en place une instance EMR opérationnelle
- Respecter les contraintes du RGPD

Les données

Images :

- 94 110 images
- 141 classes
- Un répertoire par classe, avec plusieurs photos du même fruit sous différents angles
- Format 100x100p
- Sur fond blanc

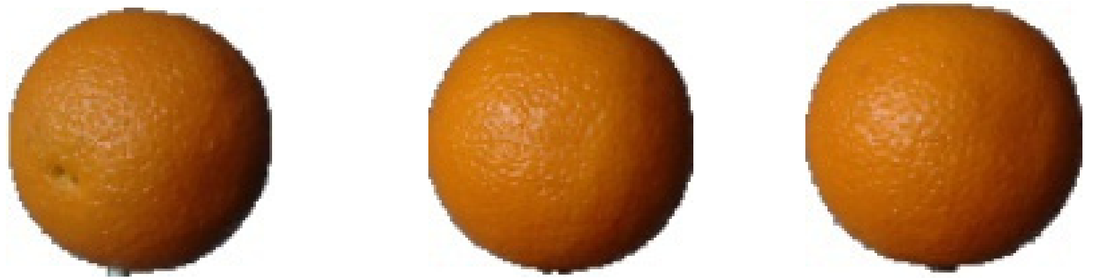
2 jeux de données :

- 71 291 données d’entraînement
- 22 819 données de test

Exemples :

- Watermelon

- Orange



Feuille de route

1 → 2 → 3 → 4

Environnement Big Data

Processus de création de l'environnement : S3 et EMR

Traitement

Réalisation de la chaine de traitement des images

Démo PySpark

Exécution du script PySpark dans le cloud

Conclusion

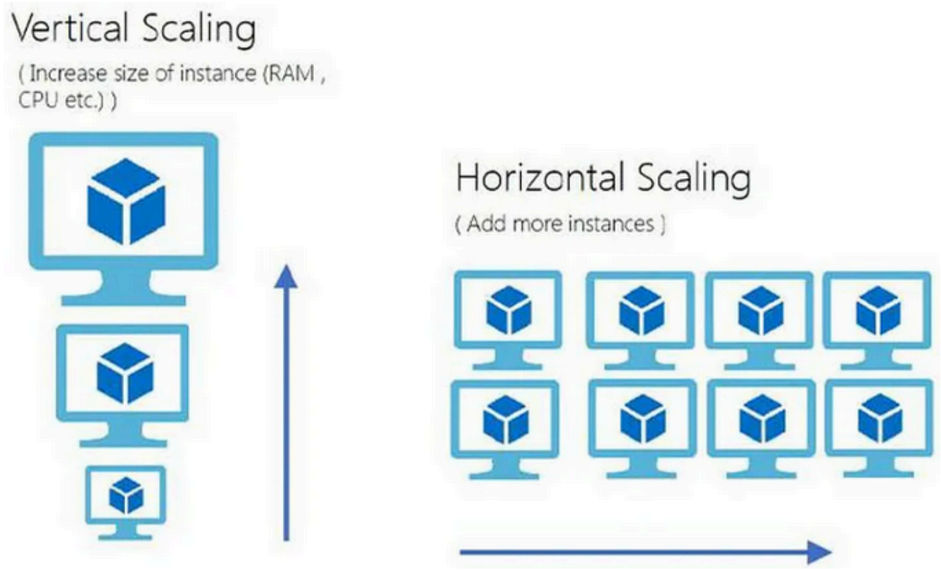
Processus de création de l'environnement : S3 et EMR

Environnement Big Data

Les 3 V :

- Volume : Notre dataset initial est conséquent, et va tendre à augmenter avec le temps
- Vitesse : Les données doivent être traitées en temps réél, quand un utilisateur utilise l'application
- Variété : Les images reçues seront de qualité, taille, éclairage différents, seront parfois illisibles ou hors sujet

Scalabilité :



Processus de Calcul distribué

MapReduce (Google, 2006) :

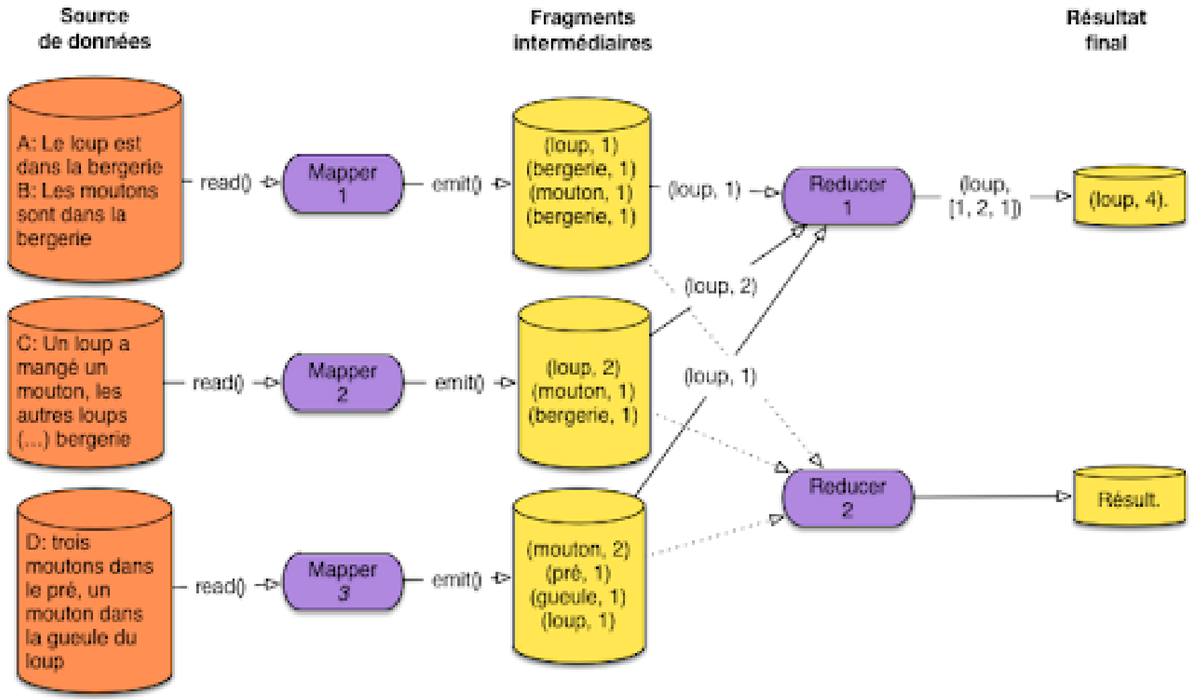
- Technologie pour le traitement de données massives
- Map : Divise les données en sous-tâches
- Reduce : Combine les résultats pour produire une sortie finale

Spark (Apache, 2009) :

- Amélioration de MapReduce
- Traitement en mémoire, donc plus rapide
- API simple et flexible pour le traitement des données

Principe :

- Exécution de tâches sur plusieurs machines.
- Permet de traiter de grandes quantités de données rapidement.



Technologies mises en places

Stockage :

- Images
- Résultats
- Notebook

Traitement :

- Cluster
- Calcul distribué

Sécurité :

- Contrôle d’accès

Rappel des règles RGPD

En théorie :

- Collecte des données : Une photo d'un fruit peut être considérée comme une donnée personnelle
- Consentement : Le consentement de l'utilisateur doit être explicitement obtenu
- Droit à l'oubli : Les utilisateurs doivent pouvoir demander la suppression de leurs données
- Protection des données : Les entreprises doivent garantir la sécurité des données

En pratique :

- Hébergement en Europe : En hébergeant nos données sur un serveur Europe, les informations des utilisateurs ne quittent pas l'espace économique européen (EEE)
- Responsabilité légale : Le transfert de données vers des pays non conformes au RGPD est restreint et nécessite des accords spécifiques
- Sanctions financières : Les amendes peuvent atteindre 20 millions d'euros ou 4 % du chiffre d'affaires annuel mondial

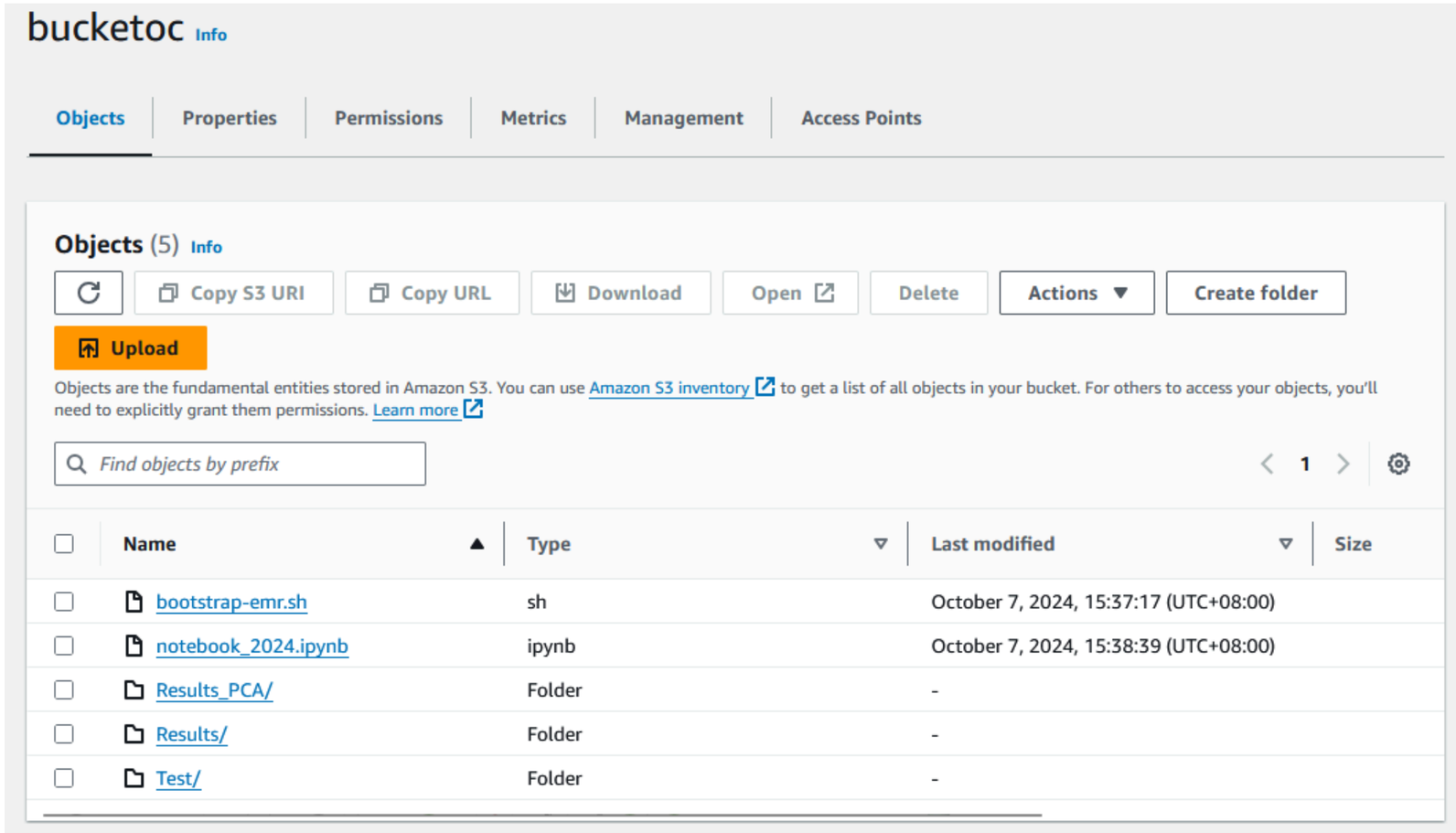
- Centres de données en Europe : AWS propose des régions en Europe, telles que Francfort, Paris ou Dublin
- Certifications de conformité : AWS possède plusieurs certifications, dont ISO 27001, garantissant des normes de sécurité élevées pour la gestion des données.

Répertoire en ligne

Bucket s3 :

- Contenu
- bootstrap-emr.sh
  permet d'initier l'emr

```
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pyspark
sudo python3 -m pip install pandas==1.2.5
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install tensorflow
sudo python3 -m pip install typing-extensions==4.3.0
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
```

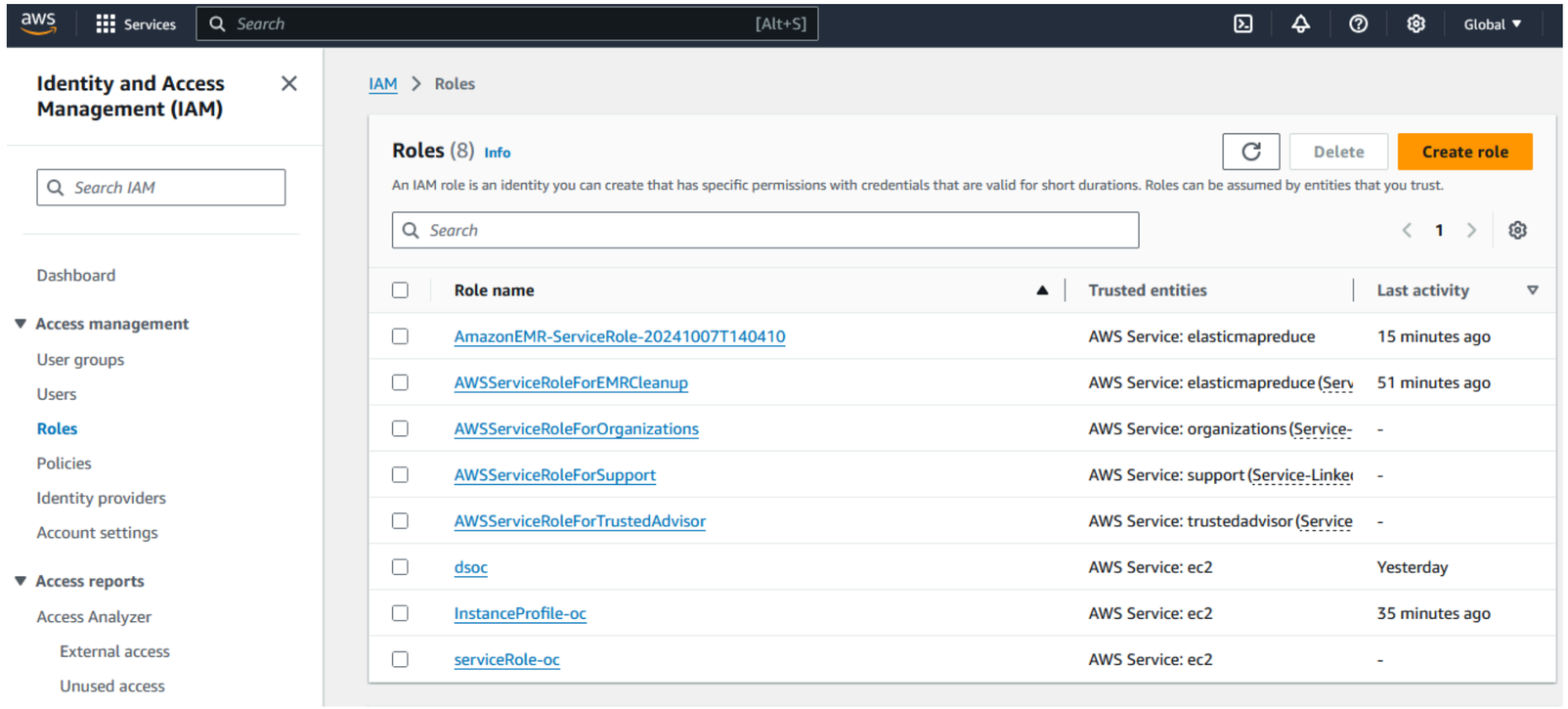


Sécurité

IAM :

- Création d’un user
- Gestion de ses droits
- Création d’une paire de clé de sécurité, dont une sera enregistrée en local pour être utilisée lors d el’initialisation du traitement

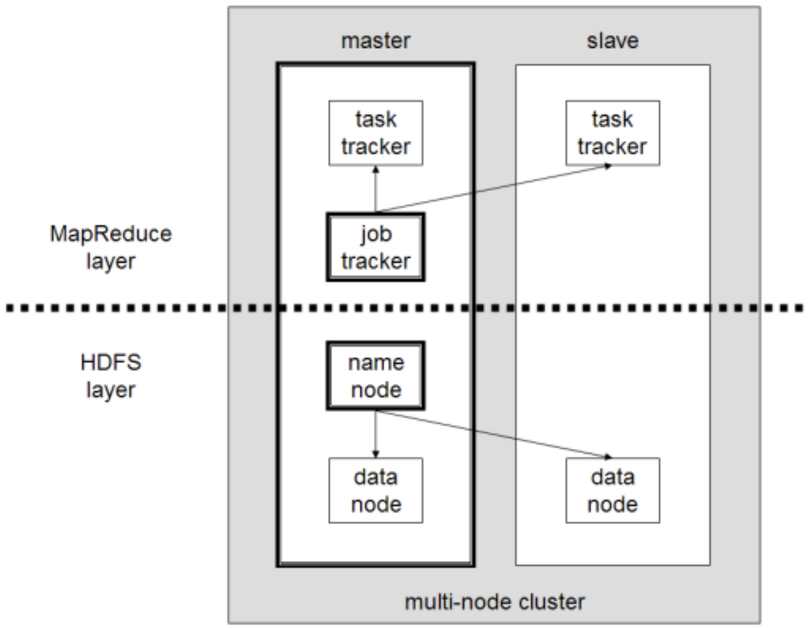
cleppkoc.ppk



Configuration de l’EMR

Principe :

- HDFS (en bas) : C'est la couche de stockage distribué qui gère les fichiers.
- MapReduce (en haut) : C'est la couche de traitement des données



Choix des logiciels :

Name

cluster-oc-cle-pem

Amazon EMR release Info

A release contains a set of applications which can be installed on your cluster.

emr-6.13.0

Application bundle

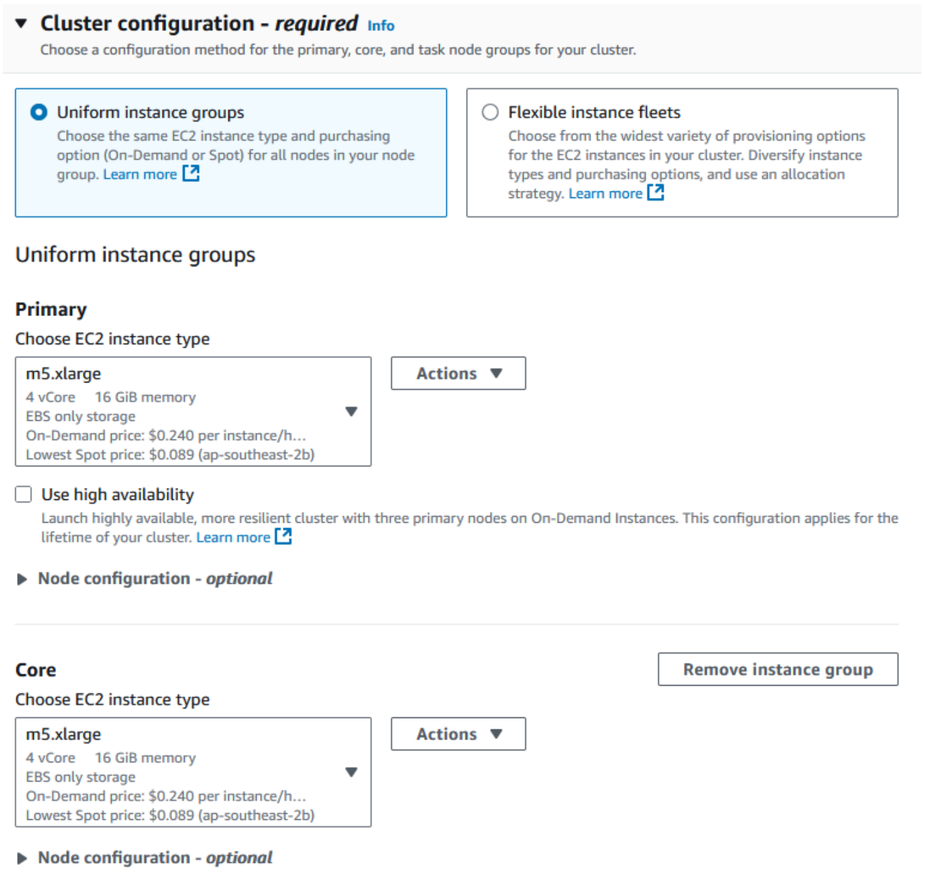
Spark | Core Hadoop | Flink | HBase | Presto | Trino | Custom

- [ ] Flink 1.17.0
- [ ] HCatalog 3.1.3
- [ ] Hue 4.11.0
- [ ] Livy 0.7.1
- [ ] Phoenix 5.1.3
- [x] Spark 3.4.1
- [ ] Tez 0.10.2
- [ ] ZooKeeper 3.5.10
- [ ] Ganglia 3.7.2
- [ ] Hadoop 3.3.3
- [ ] JupyterEnterpriseGateway 2.6.0
- [ ] MXNet 1.9.1
- [ ] Pig 0.17.0
- [ ] Sqoop 1.4.7
- [ ] Trino 414
- [ ] HBase 2.4.17
- [ ] Hive 3.1.3
- [x] JupyterHub 1.5.0
- [ ] Oozie 5.2.1
- [ ] Presto 0.281
- [x] TensorFlow 2.11.0
- [ ] Zeppelin 0.10.1

Configuration de l'EMR

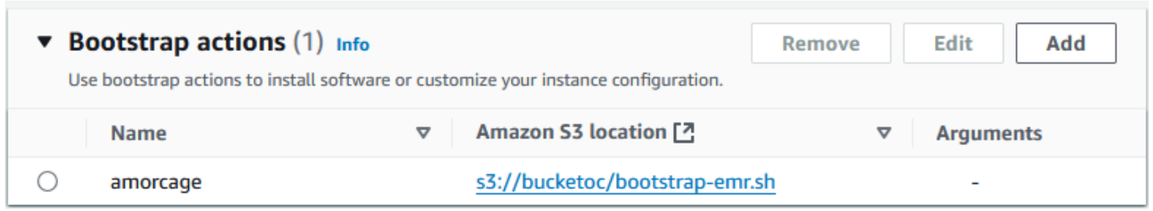
Choix des instances :

- M5 : instance équilibrée
- xlarge : la moins chère



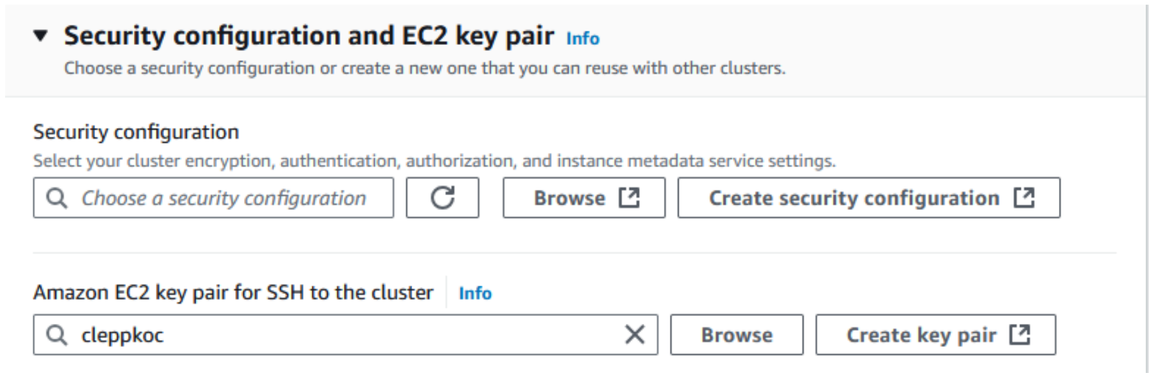
Amorçage :

- Avec le fichier bootstrap-emr.sh



Sécurité :

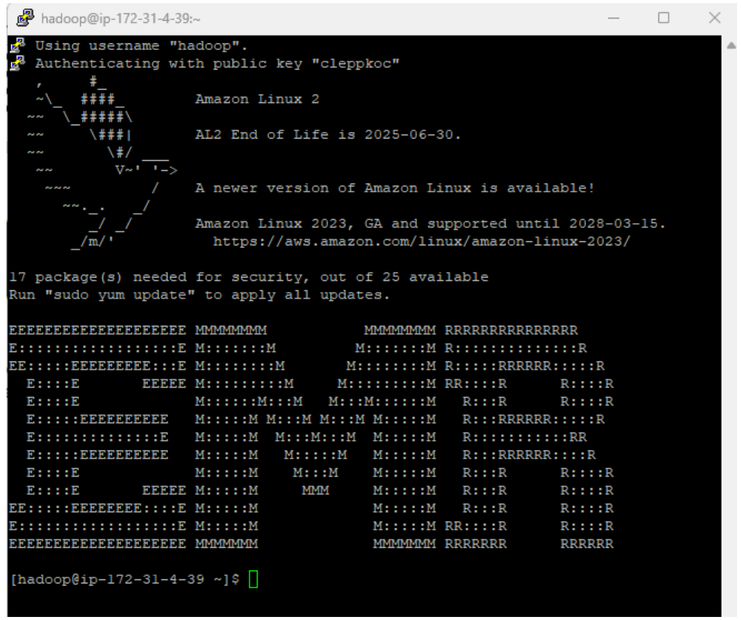
- Chargement de la clé stockée en local



Création du tunnel SSH

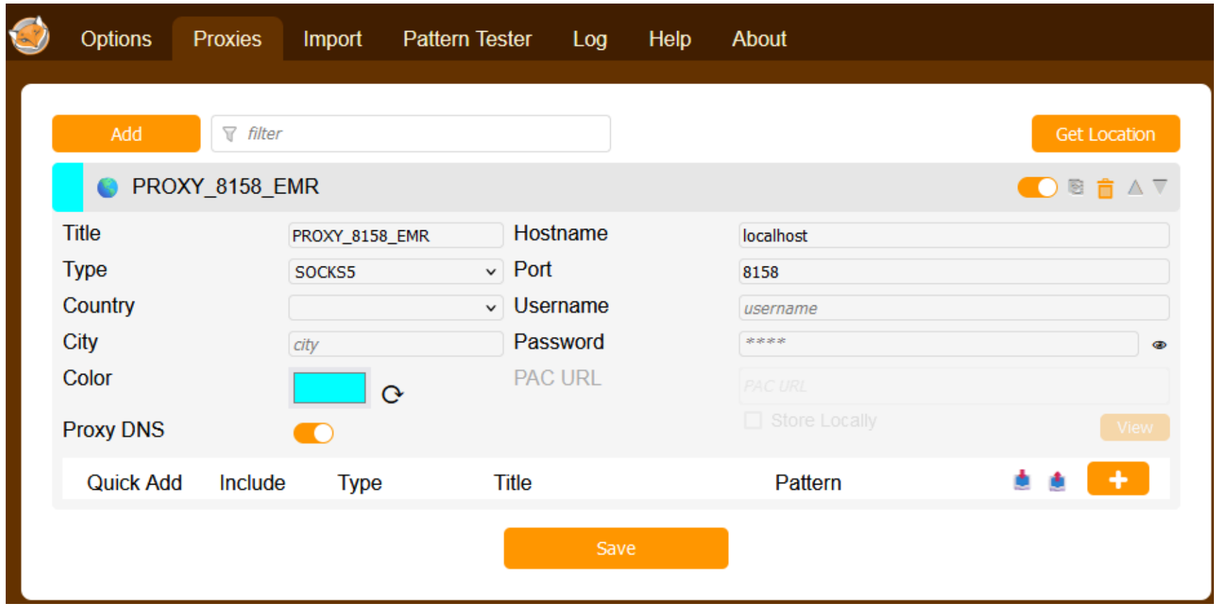
Putty :

- Logiciel client open-source qui permet d'établir des connexions sécurisées via SSH
- Utilisé pour accéder à distance à des serveursou ordinateurs ou gérer des serveurs à distance

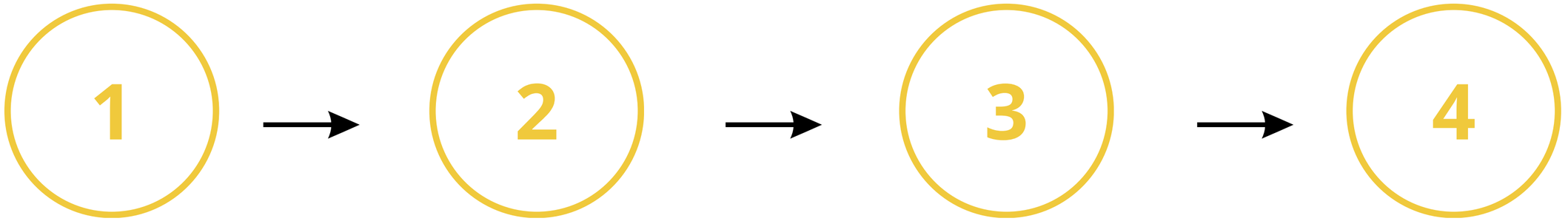


FoxyProxy :

- Redirige le trafic du navigateur à travers le tunnel SSH en configurant un proxy SOCKS
- Permet d'utiliser une connexion sécurisée pour accéder à des ressources distantes



Processus de traitement



Chargement des données

- Stockage dans s3
- Upload des images dans des Spark Dataframe

Pré-processing

- Association image/label
- Redimensionnement des images

Préparation du modèle

- Importation du modèle MobileNetV2
- Featurisation des images avec Pandas UDF

Stockage des résultats

- Ecriture des résultats au format parquet
- Stockage dans s3

Chargement des données

Chargement au format binaire :

- Avec spark.read
- Recherche dans le dossier les fichiers avec extensions .jpg
- Chargement dans un Dataframe Spark

Création des labels:

- Association label/image
- label étant le nom du fruit
- Ajout d'une colonne label issu du chemin d'accès des fichiers

```
+--------------------+--------------------+------+--------------------+
|                path|    modificationTime|length|             content|
+--------------------+--------------------+------+--------------------+
|s3://bucketoc/Tes...|2024-10-07 22:02:40|  7353|[FF D8 FF E0 00 1...|
|s3://bucketoc/Tes...|2024-10-07 22:02:42|  7350|[FF D8 FF E0 00 1...|
|s3://bucketoc/Tes...|2024-10-07 22:02:41|  7349|[FF D8 FF E0 00 1...|
|s3://bucketoc/Tes...|2024-10-07 22:02:41|  7348|[FF D8 FF E0 00 1...|
|s3://bucketoc/Tes...|2024-10-07 22:01:43|  7328|[FF D8 FF E0 00 1...|
+--------------------+--------------------+------+--------------------+
```

```
+-----------------------------------------+----------+
|path                                     |label     |
+-----------------------------------------+----------+
|s3://bucketoc/Test/Watermelon/r_106_100.jpg|Watermelon|
|s3://bucketoc/Test/Watermelon/r_109_100.jpg|Watermelon|
|s3://bucketoc/Test/Watermelon/r_108_100.jpg|Watermelon|
|s3://bucketoc/Test/Watermelon/r_107_100.jpg|Watermelon|
|s3://bucketoc/Test/Watermelon/r_95_100.jpg |Watermelon|
+-----------------------------------------+----------+
```

Réduction des dimensions

Analyse en Composantes Principales :

- La PCA est utilisée pour réduire la dimensionnalité des données tout en préservant autant que possible l'information importante

Colonne PCAfeatures :

- Correspond aux caractéristiques réduites

path	label	features	pcaFeatures
s3://bucketoc/Tes...	Tamarillo	[0.17159399390220...	[14.4558652790233...
s3://bucketoc/Tes...	Kumquats	[0.32141199707984...	[-0.1457465248047...
s3://bucketoc/Tes...	Potato Sweet	[0.66653639078140...	[-4.1908676983914...
s3://bucketoc/Tes...	Peach 2	[0.71462690830230...	[3.52180296196950...
s3://bucketoc/Tes...	Potato Sweet	[0.11092062294483...	[-0.6717270222704...
s3://bucketoc/Tes...	Papaya	[1.55914247035980...	[-1.4745985046023...
s3://bucketoc/Tes...	Tomato 2	[0.68183600902557...	[10.4618128708815...
s3://bucketoc/Tes...	Rambutan	[0.0,2.2619524002...	[-2.7053442754509...
s3://bucketoc/Tes...	Tomato 2	[0.97659122943878...	[11.3783808122340...
s3://bucketoc/Tes...	Strawberry Wedge	[0.10791978985071...	[-1.5491609724678...
s3://bucketoc/Tes...	Raspberry	[0.43169024586677...	[-0.6283486724777...
s3://bucketoc/Tes...	Peach Flat	[1.30646216869354...	[-0.1865244137097...
s3://bucketoc/Tes...	Grape Blue	[0.0,0.0,0.0,0.03...	[8.34169040067717...
s3://bucketoc/Tes...	Nectarine Flat	[0.44839087128639...	[5.26922131398869...
s3://bucketoc/Tes...	Mandarine	[0.13073460757732...	[-1.0906487876975...
s3://bucketoc/Tes...	Pineapple Mini	[0.0,4.2920012474...	[-5.3132939953158...
s3://bucketoc/Tes...	Tomato Cherry Red	[0.00452825007960...	[17.7368940024298...
s3://bucketoc/Tes...	Pomegranate	[1.43168532848358...	[7.73088204782318...
s3://bucketoc/Tes...	Pepper Red	[0.01432266086339...	[8.92947495747453...
s3://bucketoc/Tes...	Tangelo	[0.17021134495735...	[3.50354036739801...

Interface Spark

Timeline :

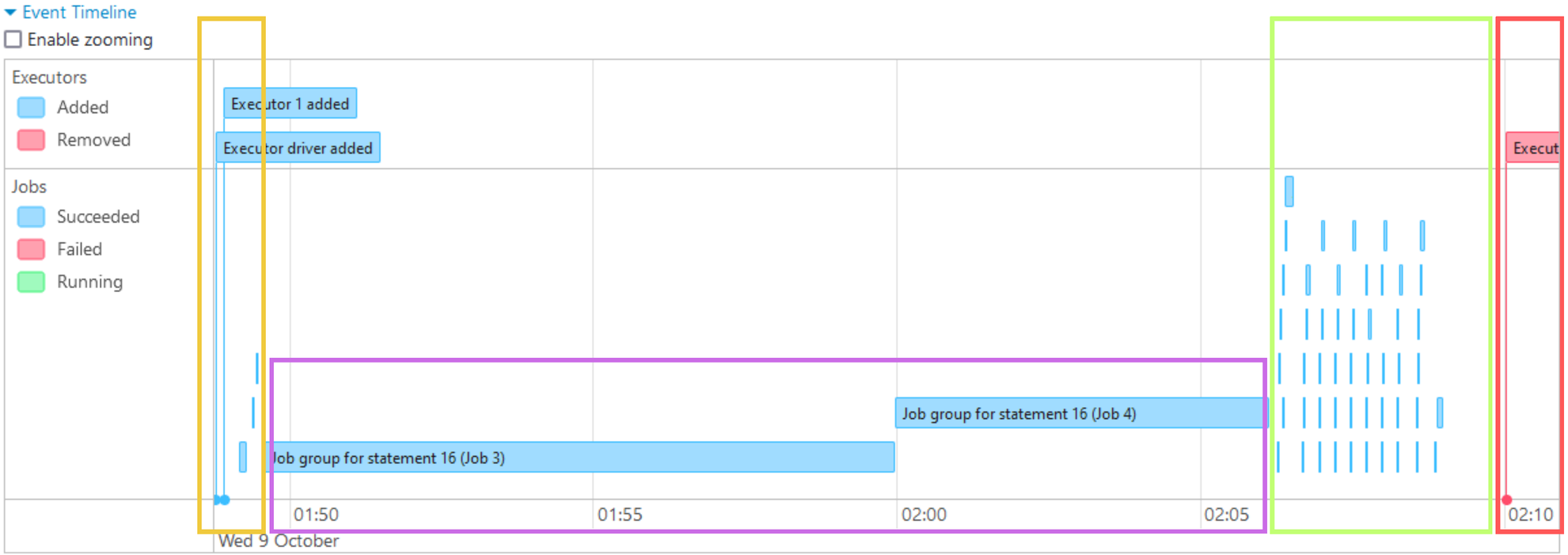
- permet de suivre l'exécution des tâches et des jobs au sein de votre cluster

Activation des machines du cluster et conversion des images en binaire

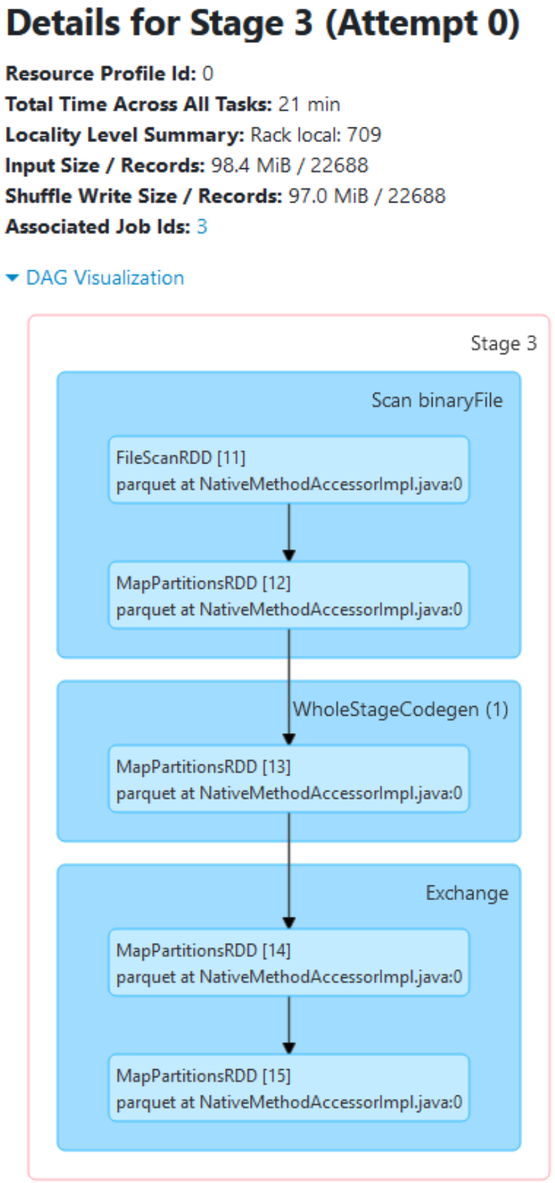
Extraction des features avec MobileNetV2 et export au format parquet

Application PCA et export des resulats au format parquet

Désactivation des machines du cluster



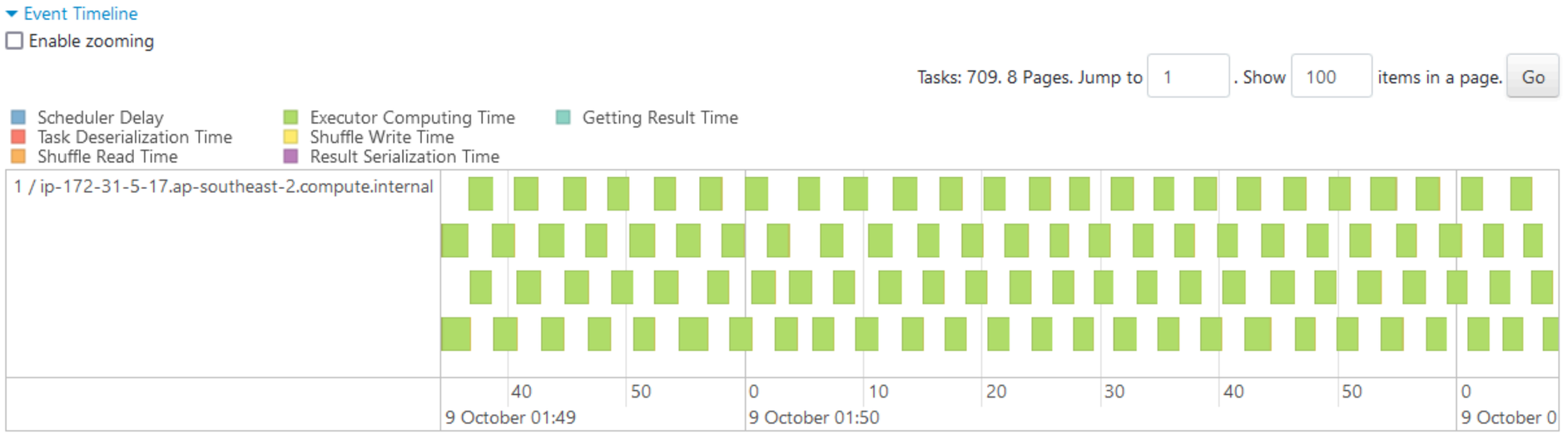
Interface Spark



Traitement des images :

- Scan binaryFile : Lecture du fichier binaire
- WholeStageCodegen : Optimisation Spark SQL. Combinaison de plusieurs opérations (filter, map, ...) en une seule fonction pour améliorer les performances.
- Exchange : Réorganisation des données à travers les partitions pour se préparer à l'algorithme Reduce

Interface Spark



Summary Metrics for 709 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	1 s	2 s	2 s	2 s	2 s
GC Time	0.0 ms	0.0 ms	0.0 ms	0.0 ms	0.2 s
Input Size / Records	74.2 KiB / 32	125.9 KiB / 32	140.7 KiB / 32	157.4 KiB / 32	226.8 KiB / 32
Shuffle Write Size / Records	68.8 KiB / 32	123.6 KiB / 32	138.9 KiB / 32	155.5 KiB / 32	225.5 KiB / 32

Timeline du stage 3 :

- Blocs verts : le temps que les nœuds de calcul prennent pour exécuter les tâches
- Le travail est réparti en 709 tâches par Spark
- Duration = 2s : le travail est réparti de manière équilibrée
- Garbage collection time = 0 : processus bien optimisé pour la mémoire

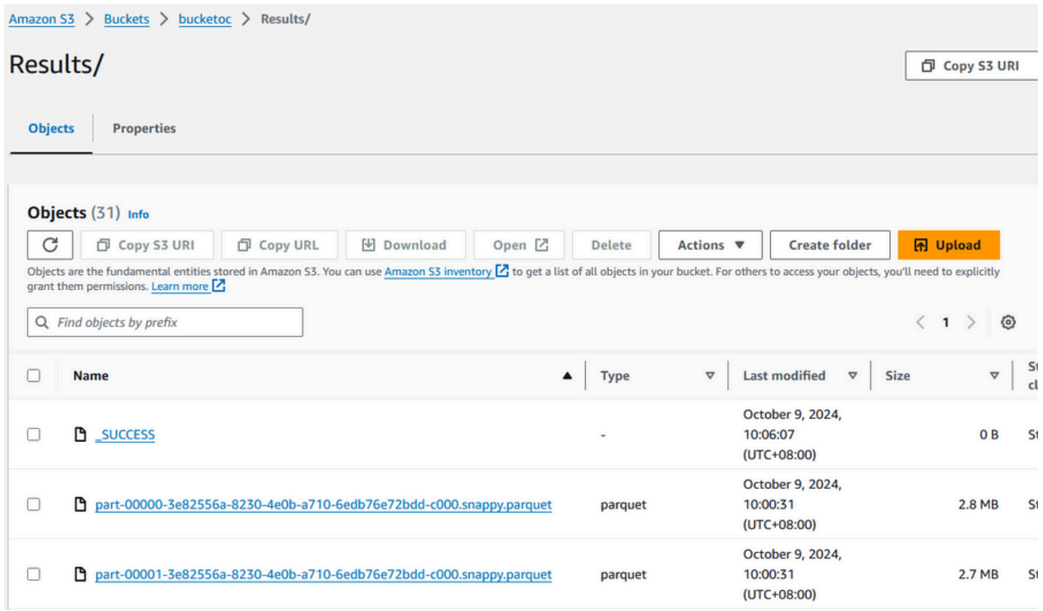
Résultats

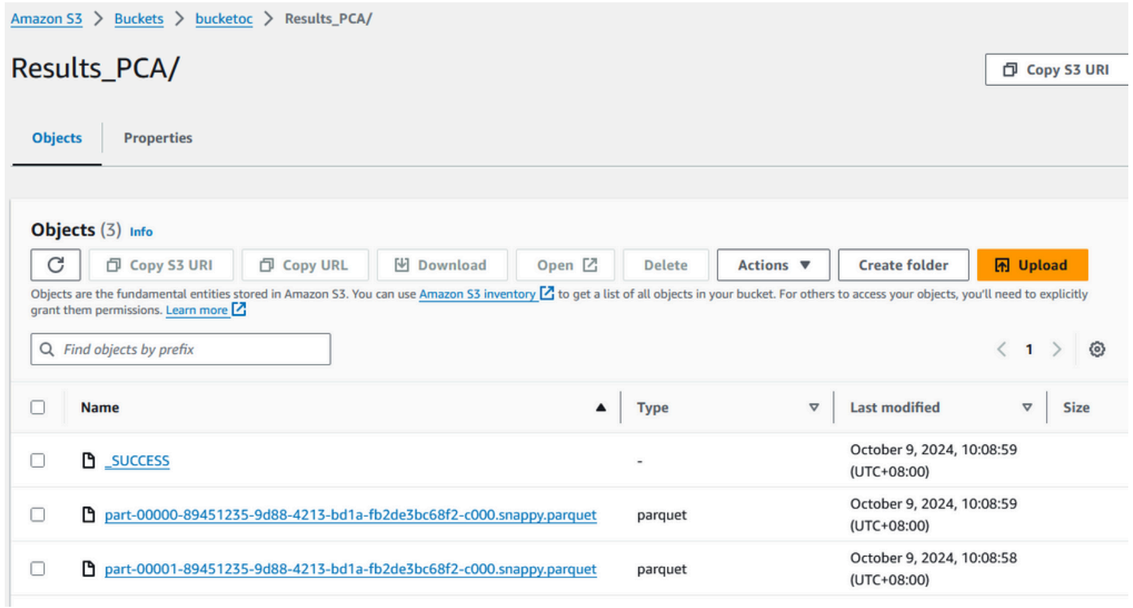
Format Parquet :

- Le format Parquet est un format de fichier en colonnes optimisé pour le stockage et la gestion de grandes quantités de données, idéal pour Hadoop et Spark

Résultats :

- Les résultats sont enregistrés dans 2 dossier différents du répertoire s3





Conclusion

Grâce à l’intégration des différentes briques de traitement, nous avons non seulement établi une base solide pour notre application mobile, mais aussi préparé le terrain pour l’évolutivité future de notre architecture Big Data.

La combinaison de PySpark avec AWS EMR nous permet de traiter efficacement les données, tout en assurant la conformité avec les réglementations en vigueur.