Déployez un modèle dans le cloud

Fruits!

Sommaire

- Problématique
- Présentation des données
- Big Data?
- Traitement des images
- Conclusions

Introduction

Fruits !

Souhaite proposer des solutions innovantes pour la récolte des fruits.
Développer des robots cueilleurs intelligent à l'aide d'une application
qui permettra aux utilisateurs de prendre en photo un fruit et d'obtenir
des informations sur ce fruit.

Mission

Développer un environnement Big Data
Réaliser une première chaîne de traitement des données avec le pré-processing et une étape de réduction de dimension.

Présentation des données

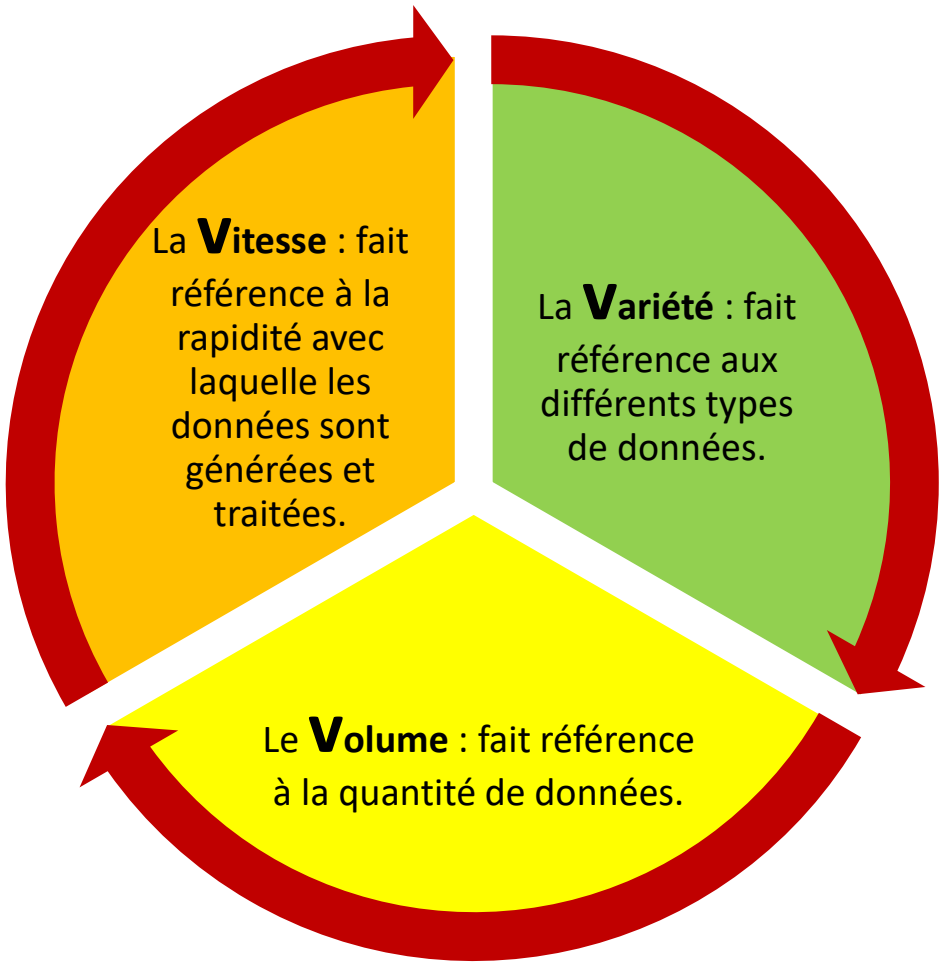
- Ensemble de données contenant des images de haut qualité de fruit avec les labels associés
- 22700 images au format JPG 100x100 pixels
- 131 variétés différentes
- Chaque fruit est photographié sous différents angles.

Big Data

1- Qu’est-ce que le Big Data ?

Le **Big Data** est un terme utilisé pour décrire des ensembles de données volumineux, complexes et hétérogènes qui sont collectés à partir de diverses sources et qui ne peuvent pas être traités par les méthodes traditionnelles de gestion de données.

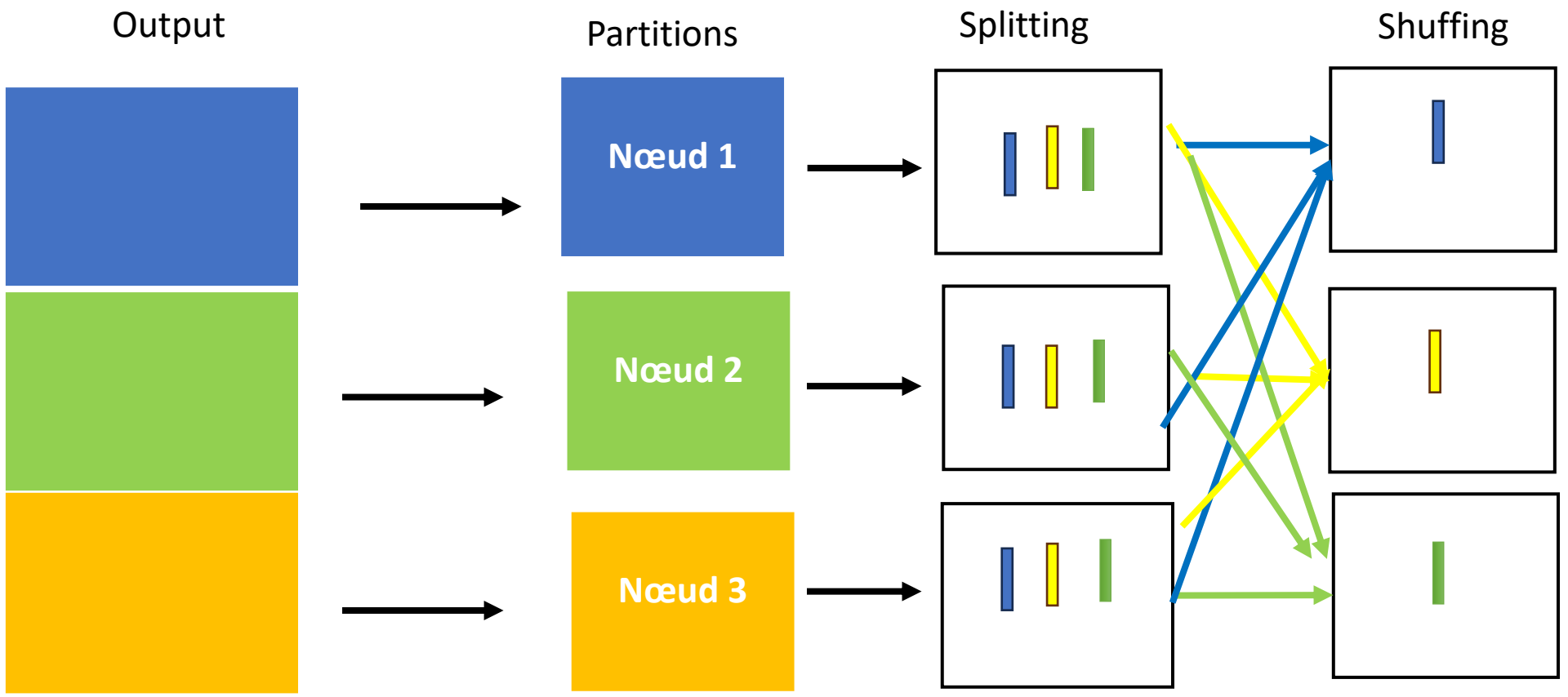
Les données du Big Data sont caractérisées par les 3V:
le volume, la variété et la vitesse.



2- Big Data Frameworks

	Hadoop	Spark
Architecture	Stocke et traite les données sur un stockage externe	Stocke et traite les données dans la mémoire interne
Traitement de données	Par lots	En temps réel
Performances	Moins rapides	Plus rapides
Machine learning	Bibliothèques externes	Bibliothèques intégrées
Sécurité	Kerberos	authentification avec un mot de passe secret
Base de données	HDFS	HDFS,…

3- Comment Spark traite les données



3- Amazon Web Services

Le Cloud AWS est une plateforme de services cloud développée par le géant américain Amazon. AWS regroupe plus de 200 services répartis en diverses catégories telles que le stockage cloud, la puissance de calcul, l'analyse de données, l'intelligence artificielle ou même le développement de jeux vidéo.

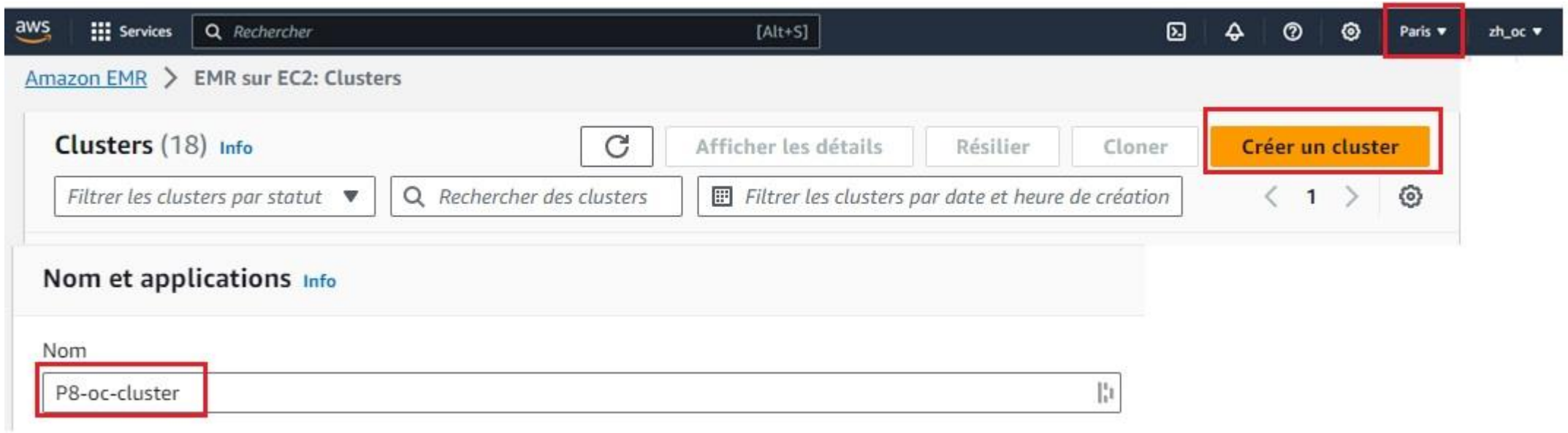
4- Configuration de la console AWS

Etape1:

Création d'un bucket sur S3 dans lequel je télécharge le contenu du dossier Test, le fichier d'amorçage (zh_bootstrap-emr.sh) et la création d'un dossier(OC_Result) pour télécharger les parquets

Etape2:

Création du cluster EMR dans Instances EC2 situées en France (eu-west-3b)



Big Data - suite

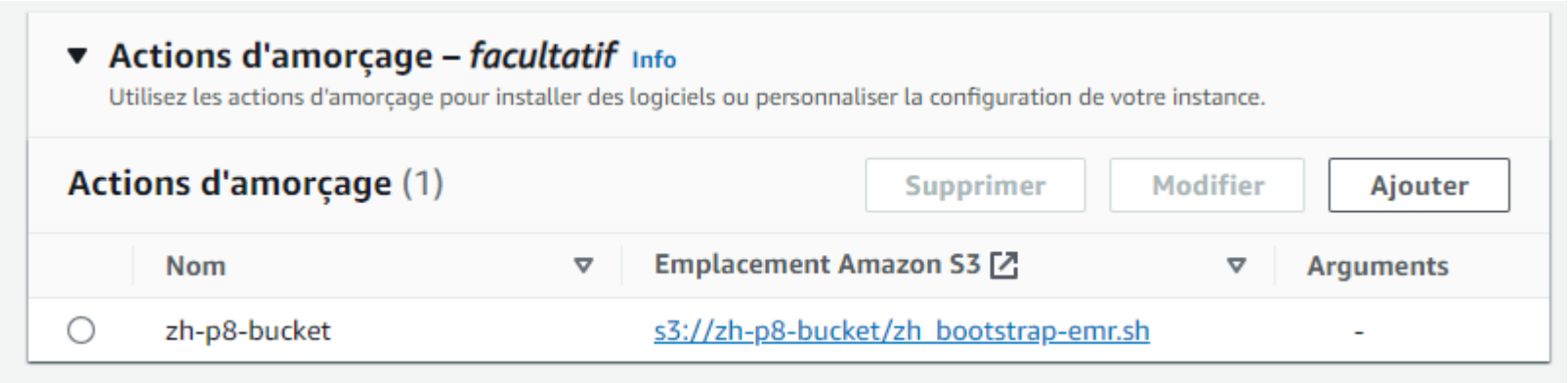
J'ai choisi spark, tensorflow et jupyterhub comme application

Configuration de mise en service

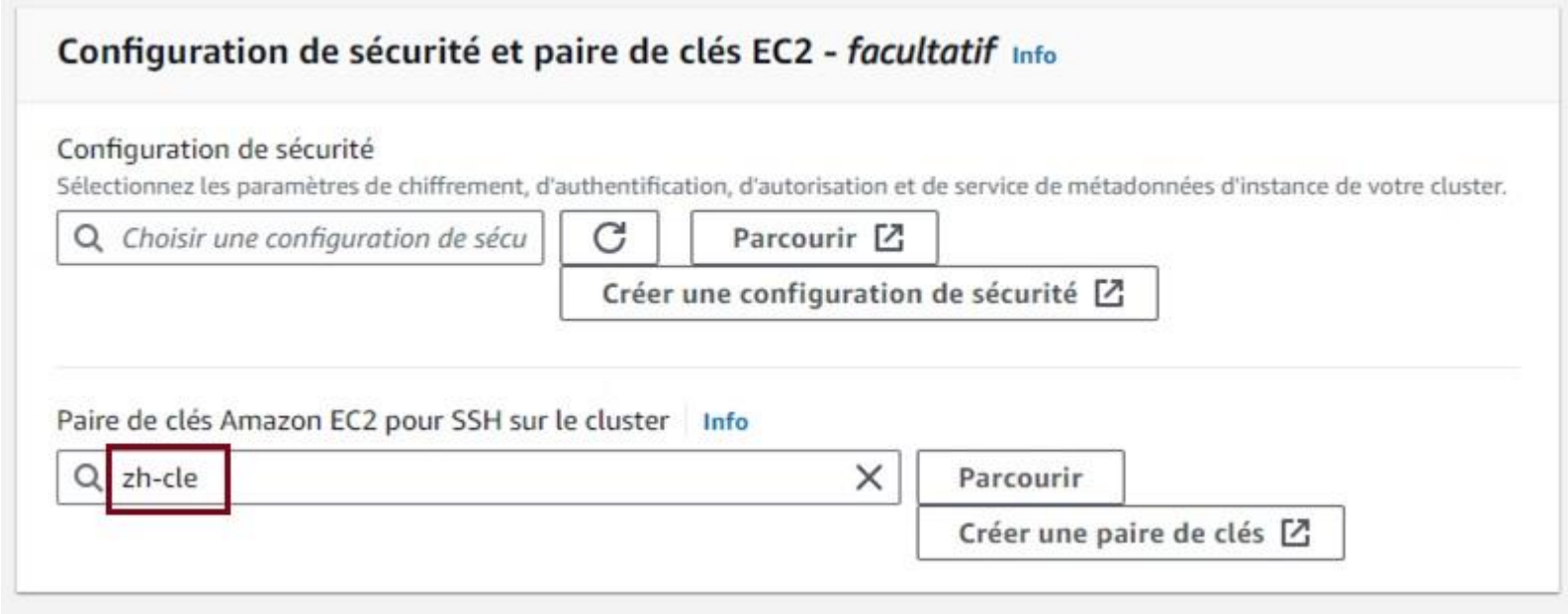
Définissez la taille de votre noyau et tâchegroupes d'instance. Amazon EMR tente de fournir cette capacité lorsque vous lancez votre cluster.

Nom	Type d'instance	Taille de l'instance(s)
Unité principale	m5.xlarge	2
Tâche - 1	m5.xlarge	1

Je séléctionne 2 instances Principales et une instances maitres => 3 instances EC2

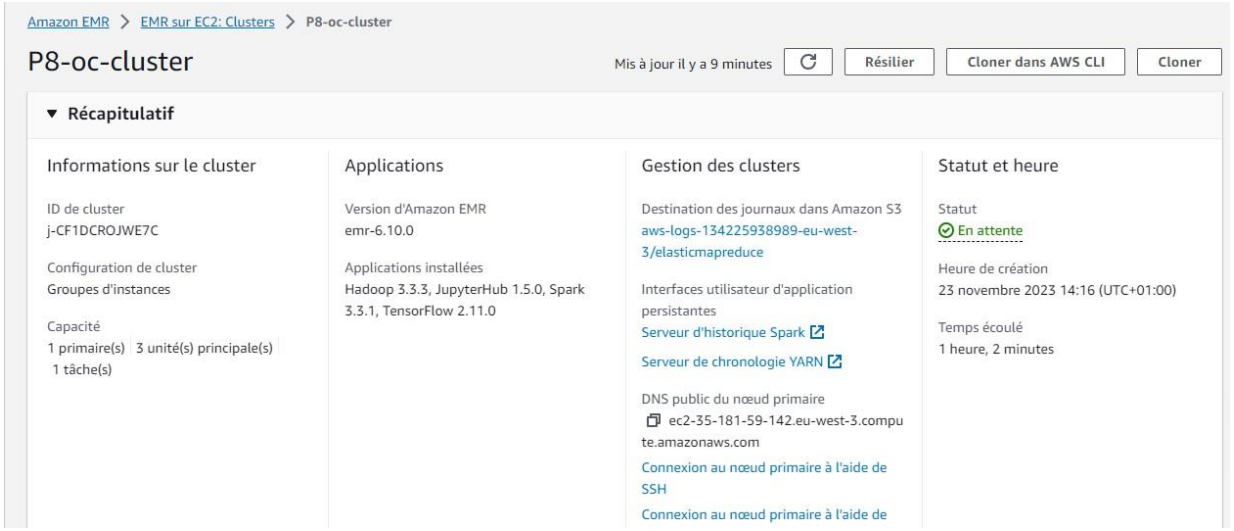


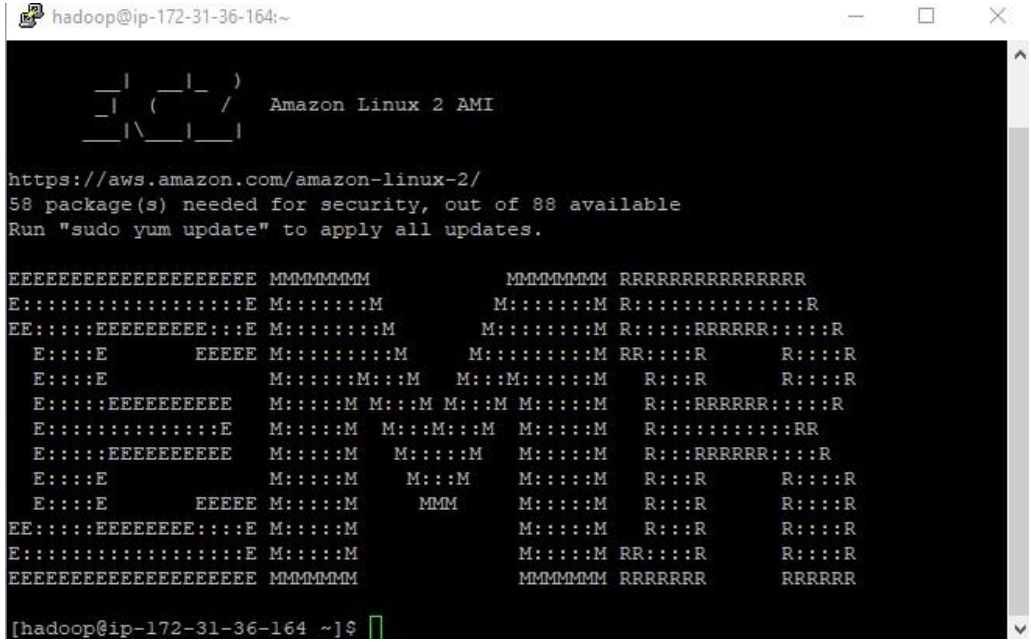
zh-bootstrap-emr.sh est fichier pour installer les bibliothèques manquantes



A cette étape nous sélectionnons la paire de clés EC2 créé précédemment
Elle nous permettra de nous connecter en ssh à nos instances EC2 sans avoir à entrer nos login/mot de passe.

Il ne nous reste plus qu'à attendre que le serveur soit prêt.
Cette étape peut prendre entre **15 et 20 minutes**.

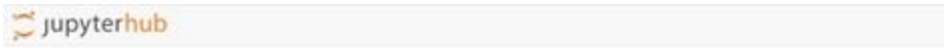


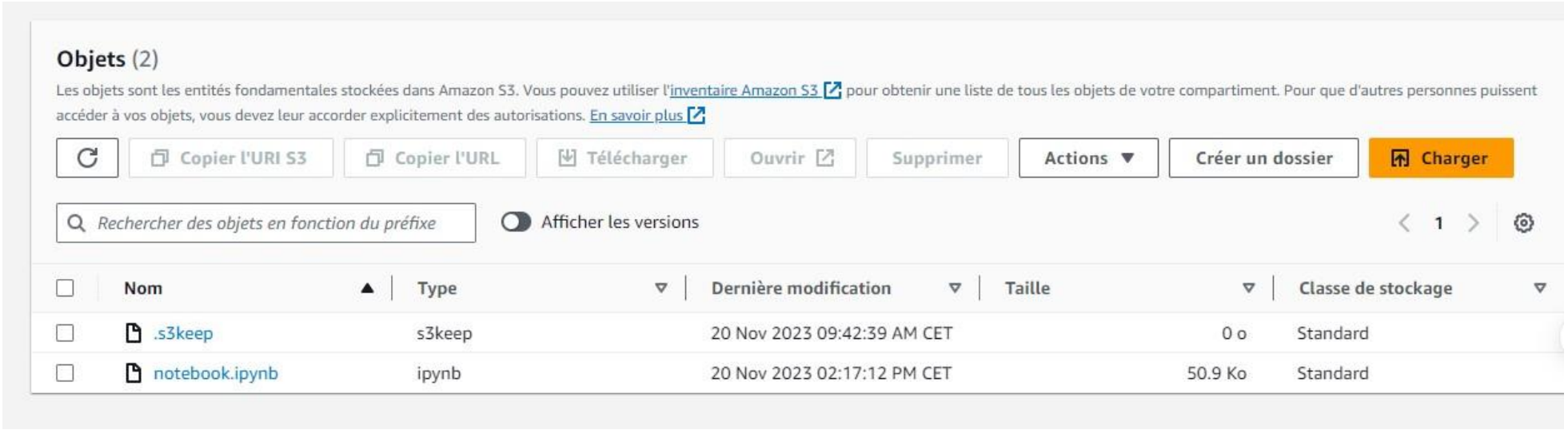


Nous avons correctement établi le tunnel **ssh** avec le driver sur le port "5555"

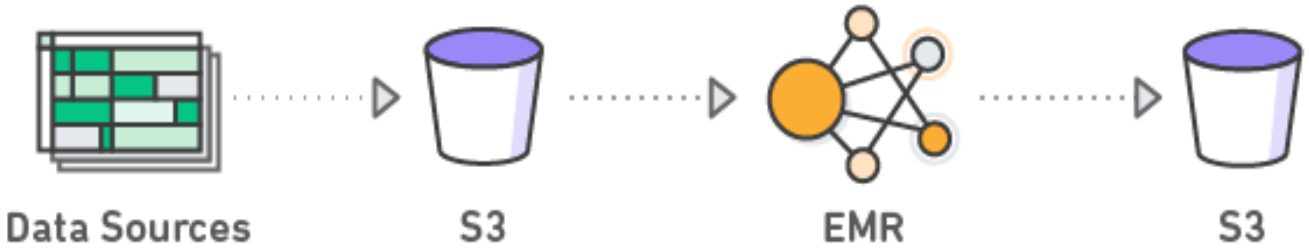
Big Data - suite

Etape3:





Traitement des images



https://www.kaggle.com/datasets/moltean/fruits

Télécharger Les données Dans S3

Utilisez Amazon EMR pour Traitement des images + Extraction Features + Réduction de Dimension (ACP)

Charger des données au format parquet dans S3

Traitement d'image - suite

- 1280 descripteur par image
- Conversion au format vecteur dense
- Standardisation

```
+--------------------+-------------+--------------------+--------------------+--------------------+
|                path|        label|            features| features_vectorized|      scaledFeatures|
+--------------------+-------------+--------------------+--------------------+--------------------+
|s3://zh-p8-bucket...|Pineapple Mini|[0.0, 5.068508, 0...|[0.0,5.0685081481...|[-0.7568072935344...|
|s3://zh-p8-bucket...|Pineapple Mini|[0.015393238, 4.6...|[0.01539323758333...|[-0.7282748230803...|
|s3://zh-p8-bucket...|    Watermelon|[0.11004015, 0.05...|[0.11004015058279...|[-0.5528399781295...|
|s3://zh-p8-bucket...|     Raspberry|[0.6104076, 0.644...|[0.61040759086608...|[0.37462697797761...|
|s3://zh-p8-bucket...|     Raspberry|[0.079946205, 0.5...|[0.07994620501995...|[-0.6086212657221...|
|s3://zh-p8-bucket...|   Cauliflower|[0.0, 0.63743114,...|[0.0,0.6374311447...|[-0.7568072935344...|
+--------------------+-------------+--------------------+--------------------+--------------------+
```

PCA finale

```
+-------------+--------------------+
|        label|        features_pca|
+-------------+--------------------+
|Pineapple Mini|[-9.3577244748706...|
|Pineapple Mini|[-6.8753150180104...|
|Pineapple Mini|[-13.269800796270...|
|     Raspberry|[-2.4566974559442...|
|     Raspberry|[-6.0856911357879...|
|   Cauliflower|[-4.4442835489278...|
+-------------+--------------------+
only showing top 6 rows
```

```
cumsum = 0
for i in pca.explainedVariance.cumsum():
    cumsum += 1
    if(i > 0.8):
        print(
            '{} composantes expliquent 80% de la variance'.format(cumsum))
        break
```

```
161 composantes expliquent 80% de la variance
```

```
pca = PCA(
    k=cumsum,
    inputCol='scaledFeatures',
    outputCol='features_pca')
model_pca = pca.fit(df_scaled)
df_final = model_pca.transform(df_scaled)
```

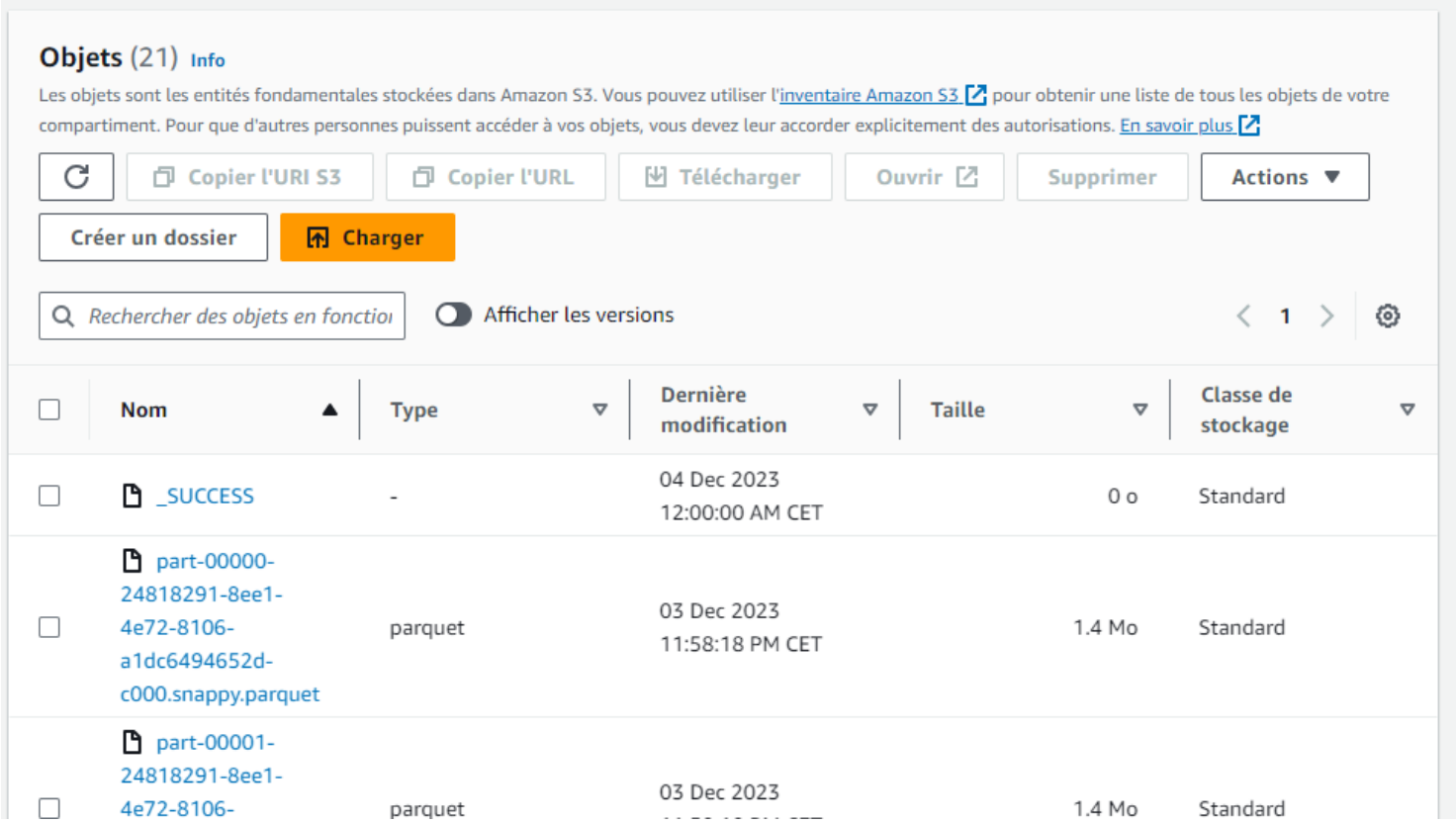
Traitement d’image - suite

Historique de session Spark

Job Id (Job Group)	Description	Submitted	Duration ▾	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
23 (20)	Job group for statement 20 first at PCA.scala:44	2023/12/03 22:06:05	4.2 min	2/2	711/711
28 (25)	Job group for statement 25 first at PCA.scala:44	2023/12/03 22:18:08	4.1 min	2/2	711/711
35 (30)	Job group for statement 30 first at PCA.scala:44	2023/12/03 22:39:34	4.0 min	2/2	711/711
4 (7)	Job group for statement 7 showString at NativeMethodAccessorImpl.java:0	2023/12/03 21:24:04	4.0 min	1/1	710/710
9 (9)	Job group for statement 9 showString at NativeMethodAccessorImpl.java:0	2023/12/03 21:29:35	3.8 min	1/1	710/710
33 (29)	Job group for statement 29 showString at NativeMethodAccessorImpl.java:0	2023/12/03 22:31:03	3.7 min	1/1	710/710
11 (10)	Job group for statement 10 first at StandardScaler.scala:113	2023/12/03 21:33:58	3.6 min	1/1	710/710

Traitement d’image - suite

Sauvegarde sur S3



Conclusions

❖ Le projet a permis de déployer un développement ML sur le cloud en environnement Big Data, en utilisant :

– Apache Spark et Pyspark pour les traitements distribués
– AWS IAM pour la gestion des utilisateurs et des autorisations
– AWS S3 pour le stockage des données

❖ Limites :

– outil payant