

# Déployer un modèle dans le Cloud

Corentin Jay - OC Data Scientist - Projet 8





# Sommaire

- ❖ Problématique et jeu de données
- ❖ Environnement Big Data
- ❖ Chaîne de traitement
- ❖ Exécution du script PySpark
- ❖ Conclusion



**Fruits!**



# **Problématique et jeu de données**

# Problématique

La start-up **Fruits!** souhaite proposer des solutions innovantes pour la récolte des fruits en développant des robots cueilleurs intelligents afin de préserver la biodiversité.

Application mobile permettant aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.



# Mission

- ❖ Mettre en place un environnement Big Data
- ❖ Réaliser une première chaîne de traitement des données avec le preprocessing et une étape de réduction de dimension

# Contraintes

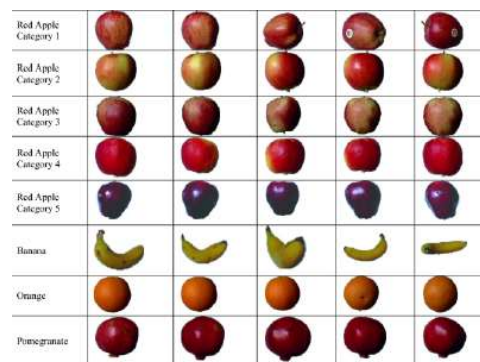
- ❖ Tenir compte de l'augmentation importante du volume de données
- ❖ Respecter les contraintes RGPD
- ❖ Gestion des coûts liés à l'architecture Big data





## Jeu de données

- + Le jeu de données est issu d'un kernel Kaggle (Fruits 360)
- + Se compose de : 131 dossiers représentant chacun un fruit, composés au total de 90'423 images de fruits
- + Images de tailles 100x100 (jpg) avec fond blanc, sur 3 axes de rotation





**Fruits!**

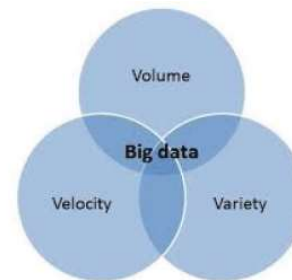


# **Environnement Big Data**



# Pourquoi le Big Data ?

- ❖ Volume de données à venir trop important pour une seule machine
- ❖ Plateforme multi-langages
- ❖ Fortes capacités de stockage
- ❖ Système collaboratif avec rôles
- ❖ Traitements par calculs distribués (Apache Spark)
- ❖ Principe des trois V :


























# Choix du prestataire

- ❖ Le plus connu, leader sur le marché
- ❖ Offre de Cloud Computing la plus large
- ❖ S3 : service historique d'AWS



<b>SaaS</b> Software as a service				
<b>FaaS</b> Function as a service				
<b>DBaaS</b> Database as a service				
<b>PaaS</b> Platform as a service				
<b>STaaS</b> Storage as a service				
<b>IaaS</b> Infrastructure as a service				

# Services AWS retenus



Instance EC2 : location de serveur sur mesure, lit et envoie les données sur S3.

- Format t2.micro
- Région : eu-west-3c



Service de stockage S3 : importante capacité de stockage de fichiers (application, résultats), tous formats.



Gestion des rôles IAM : gestion des différents rôles au sein d'une organisation, but collaboratif et d'audit.



Cluster EMR (Elastic Map Reduce) : serveur Cloud avec capacités de calculs distribués.

- Emr-6.9.0 (environnement Hadoop 3.3.3 : JupyterHub 1.4.1, Spark 3.3.0)
- m5.xlarge : 1 nœud maître et 2 nœuds principaux



# Instance EC2


Paire de clés attribuée au lancement

 **instancep8key**

Instance : i-088e39de7cce13431 (Instancep8corentinjay)

[Détails](#) | [Sécurité](#) | [Mise en réseau](#) | [Stockage](#) | [Vérifications de statut](#) | [Surveillance](#) | [Balises](#)

## ▼ Résumé de l'instance [Informations](#)

ID d'instance  
 i-088e39de7cce13431 (Instancep8corentinjay)

Adresse IPv6  
-

Type de nom d'hôte  
Nom de l'adresse IP: ip-172-31-38-167.eu-west-3.compute.internal

Réponse à un nom DNS de ressource privée IPv4 (A)  
-

Adresse IP attribuée automatiquement  
-

Rôle IAM  
-

IMDSv2  
Required

Adresse IPv4 publique  
-

État de l'instance  
 Arrêté(e)

Nom DNS de l'IP privé (IPv4 uniquement)  
 ip-172-31-38-167.eu-west-3.compute.internal

Type d'instance  
t2.micro

ID de VPC  
 vpc-0e6bb415af09227a4 [🔗](#)

ID de sous-réseau  
 subnet-0f140d724f238d53b [🔗](#)

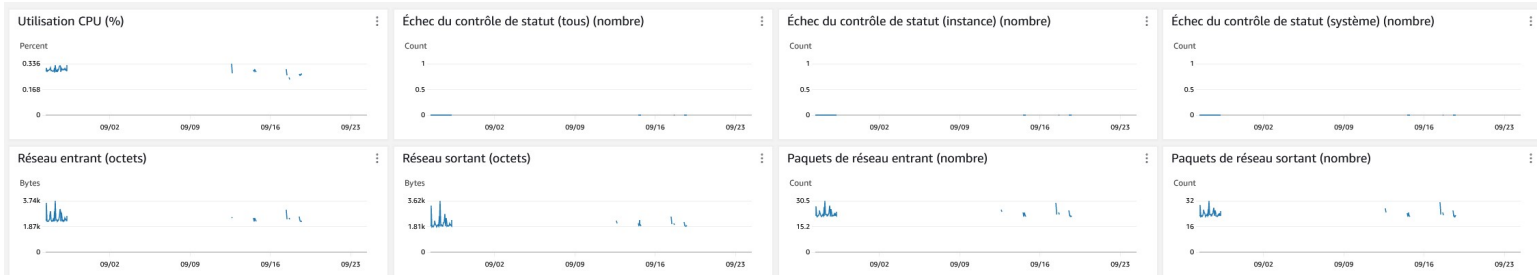
Adresses IPv4 privées  
 172.31.38.167

DNS IPv4 public  
-

Adresses IP élastiques  
-

Recherche d'AWS Compute Optimizer  
[Inscrivez-vous à AWS Compute Optimizer pour obtenir des recommandations. | En savoir plus](#) [🔗](#)

Nom du groupe Auto Scaling  
-





# Stockage sur S3

Amazon S3 > Compartiments > s3p8corentinjay

s3p8corentinjay [Infos](#)

Objets Propriétés Autorisations Métriques Gestion Points d'accès

Objets (19)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[Inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

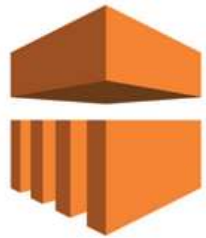
Rechercher des objets en fonction du préfixe

Nom	Type	Dernière modification	Taille	Classe de stockage
bootstrap-emr-v3.sh	sh	19 Sep 2023 01:27:42 PM CEST	366.0 o	Standard
config.json	json	15 Sep 2023 01:19:50 PM CEST	158.0 o	Standard
Final_results	-	19 Sep 2023 03:04:24 PM CEST	9.7 Ko	Standard
j-0364013TNPDV203RM3D/	Dossier	-	-	-
j-03773231HA54QHZVP022/	Dossier	-	-	-
j-03792152SLA2MJQGLBE/	Dossier	-	-	-
j-03820231J93AG64VN8XN/	Dossier	-	-	-
j-03856376NKCX084QDU/	Dossier	-	-	-
j-03899151K5770KDH025/	Dossier	-	-	-
j-039012416AKHGV87891/	Dossier	-	-	-
j-03909803D2KMH2E5J58H/	Dossier	-	-	-
j-03945051FB5EOPCDH6Y/	Dossier	-	-	-
j-039457153P9Z10V95PN/	Dossier	-	-	-
j-03947803J8P7V6F4LIMQ/	Dossier	-	-	-
j-0394987387TGNZLBDGQV/	Dossier	-	-	-
j-04205772X3ZGC03DUGSA/	Dossier	-	-	-
jupyter/	Dossier	-	-	-
Results/	Dossier	-	-	-
Test/	Dossier	-	-	-

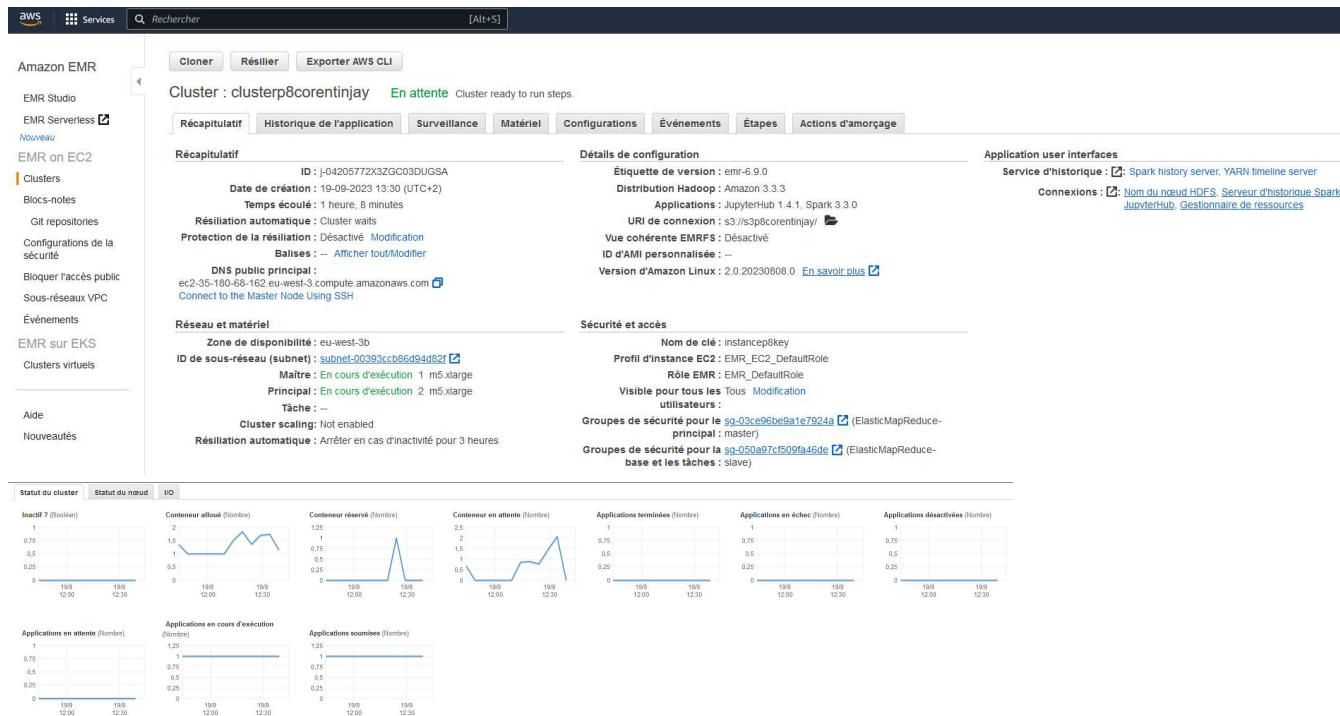
```

C:\ Invite de commandes
Microsoft Windows [version 10.0.19045.3324]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\JayCo>aws s3 ls
2023-08-31 10:28:18 s3p8corentinjay
  
```



# Cluster emr-6.9.0







**Fruits!**



# **Chaîne de traitement**

# Etape 1 : test du process en local

- ❖ Exécution du notebook de l'alternant sur Jupyter :
- ❖ Définition des chemins en local
- ❖ Création d'une session Spark 
- ❖ Transfert learning (MobilenetV2, weights = Imagenet)
- ❖ Création des features des images
- ❖ Enregistrement du résultat au format .parquet 

Schéma d'architecture du modèle :

Input	Operator	$t$	$c$	$n$	$s$
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

## Etape 2 : création de l'architecture Cloud

- ❖ Lancement d'une instance EC2
  - ❖ Type **t2.micro** / Région **eu-west-3c** (Paris)
  - ❖ Génération d'une paire de clé (format .pem)
- ❖ Création du stockage sur S3 ('s3p8corentinjay')
  - ❖ Connexion à S3 :
  - ❖ Upload des fichiers par AWS CLI

```

Invite de commandes
Microsoft Windows [version 10.0.19045.3324]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\JayCo>aws s3 ls
2023-08-31 10:28:18 s3p8corentinjay
  
```

Amazon S3 > Compartiments > s3p8corentinjay > Test/

Test/ Copier l'URL S3

Objets Propriétés

Objets (30)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez afficher l'[Documentation Amazon S3](#) pour obtenir une liste de tous les objets du votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

☐ Copier l'URL S3 ☐ Copier l'URL ☐ Télécharger ☐ Ouvrir ☐ Supprimer

Rechercher des objets en fonction du préfixe

Objet	Type	Dernière modification	Taille	Classe de stockage
140_100.jpg	jpg	31 Aug 2023 10:29:17 AM CEST	4.1 Ko	Standard
150_100.jpg	jpg	31 Aug 2023 10:29:17 AM CEST	5.9 Ko	Standard
158_100.jpg	jpg	31 Aug 2023 10:29:17 AM CEST	3.0 Ko	Standard

## Etape 3 : lancement du cluster emr

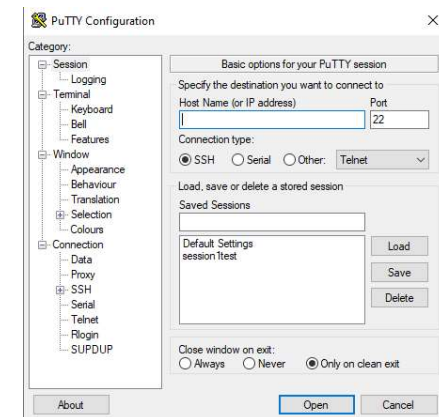
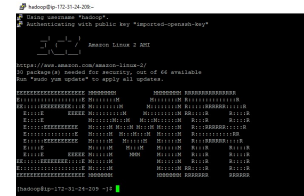
- ❖ Paramétrage du cluster emr :
  - ❖ **Emr-6.9.0**
  - ❖ Logiciels **Hadoop+JupyterHub+Spark**
  - ❖ Clé de sécurité liée à l'instance EC2 ('instancep8key')
  - ❖ Journalisation sur S3 ('s3p8corentinjay')
  - ❖ Nœuds m5.xlarge : 1 nœud maître et 2 nœuds principaux
  - ❖ Fichiers **config.json** et **bootstrap-emr.sh**
- ❖ Instanciation du cluster (10-15min)

```
bootstrap-emr-v3 - Bloc-notes
Fichier  Edition  Format  Affichage  Aide
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 pandas
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
sudo python3 -m pip install tensorflow
```

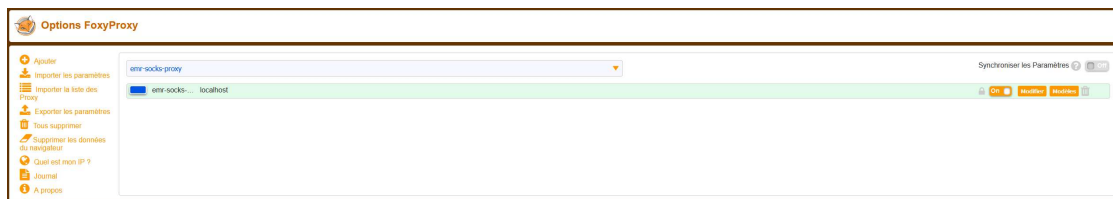
```
config - Bloc-notes
Fichier  Edition  Format  Affichage  Aide
| [
  {
    "classification": "jupyter-s3-conf",
    "properties": {
      "s3.persistence.bucket": "s3p8corentinjay",
      "s3.persistence.enabled": "true"
    }
  }
]
```

## Etape 4 : connexion au cluster

- ❖ Utilisation du logiciel Putty
- ❖ Transposition de la clé .pem en clé .ppk
- ❖ Connexion au cluster



- ❖ Connexion via proxy (FoxyProxy)



```

foxyproxy.settings - Bloc-notes
Fichier Edition Format Affichage Aide
1
{
  "name": "emr-socks-proxy",
  "active": true,
  "address": "localhost",
  "port": 8157,
  "username": "",
  "password": "",
  "type": 3,
  "group": 1,
  "title": "emr-socks-proxy",
  "color": "#808080",
  "index": 9807199254789991,
  "whitePatterns": [
    {
      "title": "ec2*.amazonaws.com",
      "active": true,
      "pattern": "ec2*.amazonaws.com",
      "importedPattern": "ec2*.amazonaws.com",
      "type": 1,
      "protocols": 1
    },
    {
      "title": "ec2*.compute*",
      "active": true,
      "pattern": "ec2*.compute*",
      "importedPattern": "ec2*.compute*",
      "type": 1,
      "protocols": 1
    },
    {
      "title": "s3*",
      "active": true,
      "pattern": "s3*",
      "importedPattern": "http://s3*",
      "type": 1,
      "protocols": 2
    }
  ]
}

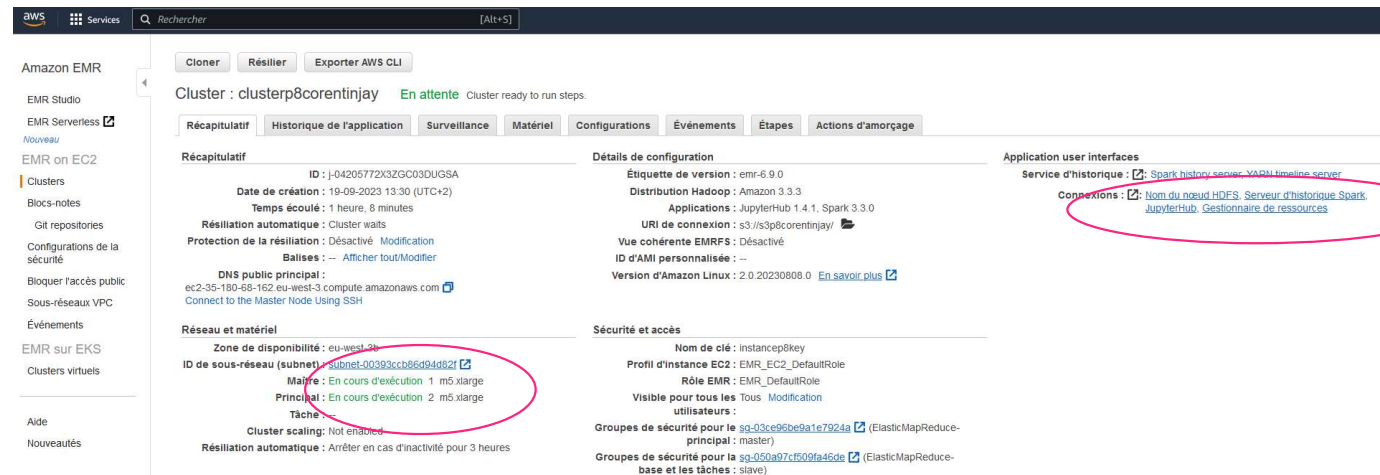
```





# Synthèse

**clusterp8corentinjay** désormais actif et en attente



The screenshot displays the AWS Management Console interface for an Amazon EMR cluster. The cluster name is **clusterp8corentinjay**, and its status is **En attente** (Pending), indicating it is ready to run. The console shows various tabs for cluster management, including Récapitulatif, Historique de l'application, Surveillance, Matériel, Configurations, Événements, Étapes, and Actions d'amorçage. The **Récapitulatif** (Summary) tab is active, providing a comprehensive overview of the cluster's configuration and state.

**Récapitulatif**

- ID : j-04205772X3ZGC03DUGSA
- Date de création : 19-09-2023 13:30 (UTC+2)
- Temps écoulé : 1 heure, 8 minutes
- Résiliation automatique : Cluster waits
- Protection de la résiliation : Désactivée
- Configuration de la résiliation : Afficher tout/Modifier
- DNS public principal : ec2-35-190-68-162.eu-west-3.compute.amazonaws.com
- Connect to the Master Node Using SSH

**Détails de configuration**

- Étiquette de version : emr-6.9.0
- Distribution Hadoop : Amazon 3.3.3
- Applications : JupyterHub 1.4.1, Spark 3.3.0
- URI de connexion : s3://s3p8corentinjay/
- URI cohérente EMRFS : Désactivée
- ID d'AMI personnalisée : --
- Version d'Amazon Linux : 2.0.20230808.0

**Application user interfaces**

- Service d'historique : [Spark history server](#), [VARN timeline server](#)
- Connexions : [Nom du nœud HDFS](#), [Serveur d'historique Spark](#), [JupyterHub](#), [Gestionnaire de ressources](#)

**Réseau et matériel**

- Zone de disponibilité : eu-west-3
- ID de sous-réseau (subnet) : [subnet-00393ccb8694d82f](#)
- Maître : En cours d'exécution 1 m5.xlarge
- Principal : En cours d'exécution 2 m5.xlarge
- Tâche : En cours d'exécution
- Cluster scaling : Not enabled
- Résiliation automatique : Arrêter en cas d'inactivité pour 3 heures

**Sécurité et accès**

- Nom de clé : instancep8key
- Profil d'instance EC2 : EMR\_EC2\_DefaultRole
- Rôle EMR : EMR\_DefaultRole
- Visible pour tous les utilisateurs : Tous
- Groupe de sécurité pour le principal : [sg-03ce9b6e9a1e7924a](#) (ElasticMapReduce- principal : master)
- Groupe de sécurité pour la base et les tâches : [sg-050a07cf500fa46de](#) (ElasticMapReduce- base et les tâches : slave)

Spark et JupyterHub sont désormais accessibles et utilisables



**Fruits!**



**Script  
PySpark**

# Import des bibliothèques, définition des chemins sur S3 et lecture des images

```

jupyterhub P8_01_notebook Last Checkpoint: il y a quelques secondes (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Tr

Installation librairies

In [1]: import pandas as pd
import numpy as np
import io
import os
import tensorflow as tf
from PIL import Image
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import Model
from pyspark.sql.functions import col, pandas_udf, PandasUDFType, element_at, split

Starting Spark application

ID YARN Application ID Kind State Spark UI Driverlog User Current session?
0 application_1695123410191_0001 pyspark idle Link Link None ✓

SparkSession available as 'spark'.

In [2]: print(tf.__version__)
2.11.0

In [3]: %info
Current session configs: {'driverMemory': '1000M', 'executorCores': 2, 'proxyUser': 'jovyan', 'kind': 'pyspark'}
ID YARN Application ID Kind State Spark UI Driverlog User Current session?
0 application_1695123410191_0001 pyspark idle Link Link None ✓

In [5]: sc = spark.sparkContext

Chemin Images et chargement

In [6]: PATH = 's3://s3p8corentinjay'
PATH_Data = PATH + '/Test'
PATH_Results = PATH + '/Results'
print('PATH: ' + PATH)
PATH_Data = PATH + '/Test'
PATH_Results = PATH + '/Results'

In [7]: images = spark.read.format("binaryfile") \
.option("pathGlobFilter", "*.jpg") \
.option("recursiveFileLookup", "true") \
.load(PATH_Data)

```

```

In [8]: images.show(5)

+-----+-----+-----+-----+
| path | modificationTime | length | content |
+-----+-----+-----+-----+
| s3://s3p8corentinjay/Test/Apple Golden 1/96_100.jpg | 2023-09-15 14:39:37 | 6304 | [FF 08 FF E0 00 1... |
| s3://s3p8corentinjay/Test/Apple Golden 1/95_100.jpg | 2023-09-15 14:39:37 | 6277 | [FF 08 FF E0 00 1... |
| s3://s3p8corentinjay/Test/Apple Golden 1/94_100.jpg | 2023-09-15 14:39:36 | 6231 | [FF 08 FF E0 00 1... |
| s3://s3p8corentinjay/Test/Apple Golden 1/104_100.jpg | 2023-09-15 14:39:32 | 6220 | [FF 08 FF E0 00 1... |
| s3://s3p8corentinjay/Test/Apple Golden 1/93_100.jpg | 2023-09-15 14:39:36 | 6217 | [FF 08 FF E0 00 1... |
+-----+-----+-----+-----+
only showing top 5 rows

Sélection colonnes et ajout labels

In [9]: images = images.withColumn('label', element_at(split(images['path'], '/'), -1))
print(images.printSchema())
print(images.select('path', 'label').show(25, False))

root
 |-- path: string (nullable = true)
 |-- modificationTime: timestamp (nullable = true)
 |-- length: long (nullable = true)
 |-- content: binary (nullable = true)
 |-- label: string (nullable = true)

None
+-----+-----+-----+-----+
| path | modificationTime | length | content | label |
+-----+-----+-----+-----+
| s3://s3p8corentinjay/Test/Apple Golden 1/96_100.jpg | 2023-09-15 14:39:37 | 6304 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/95_100.jpg | 2023-09-15 14:39:37 | 6277 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/94_100.jpg | 2023-09-15 14:39:36 | 6231 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/104_100.jpg | 2023-09-15 14:39:32 | 6220 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/93_100.jpg | 2023-09-15 14:39:36 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/119_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/84_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/86_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/85_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/82_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/81_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/83_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/80_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/79_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/78_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/72_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/77_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/76_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/74_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/75_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/73_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Apple Golden 1/63_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Apple Golden 1 |
| s3://s3p8corentinjay/Test/Kiwi/7_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Kiwi |
| s3://s3p8corentinjay/Test/Kiwi/36_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Kiwi |
| s3://s3p8corentinjay/Test/Kiwi/9_100.jpg | 2023-09-15 14:39:32 | 6217 | [FF 08 FF E0 00 1... | Kiwi |
+-----+-----+-----+-----+
only showing top 25 rows

```

## Transfert learning (MobilenetV2), extraction des features des images et enregistrement en format .parquet

### Préparation du modèle

```
In [10]: model = MobileNetV2(weights='imagenet',
    include_top=True,
    input_shape=(224, 224, 3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224.h5
14536120/14536120 [=====] - 1s 0us/step

In [11]: new_model = Model(inputs=model.input,
    outputs=model.layers[-2].output)

In [12]: broadcast_weights = sc.broadcast(new_model.get_weights())

In [13]: new_model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	input_1[0][0]
Conv1 (Conv2D)	(None, 112, 112, 32)	864	input_1[0][0]
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 112, 112, 32)	0	bn_Conv1[0][0]
expanded_conv_depthwise (DepthwiseConv2D)	(None, 112, 112, 32)	288	Conv1_relu[0][0]

### Extraction des features des images

```
In [16]: features_df = images.repartition(24).select(col("path"),
    col("label"),
    featurize_udf("content").alias("features"))

In [17]: print(PATH_Results)

s3://s3p8corentinjay/Results

In [18]: features_df.write.mode("overwrite").parquet(PATH_Result)
```



# Lecture des features, vectorisation, réduction de dimension (PCA), enregistrement en .csv et représentation graphique

```

Chargement des données enregistrées

In [17]: # Ouverture du fichier au format parquet
df = pd.read_parquet(PATH_Result, engine='pyarrow')

In [20]: # Aperçu
df.head()

   path  ...  features
0  s3://s3p8corentinjay/Test/Apple Golden 1/72_10...  ...  [0.0, 0.05176872, 0.010632622, 0.0, 0.0, 0.0, ...
1  s3://s3p8corentinjay/Test/Kiwi/44_100.jpg  ...  [0.7344478, 0.0, 0.0, 0.0, 0.0, 0.0, 0.53...
2  s3://s3p8corentinjay/Test/Blueberry/30_100.jpg  ...  [0.6798481, 0.1693784, 0.0, 0.1060800, 0.0585...
3  s3://s3p8corentinjay/Test/Apple Golden 1/85_10...  ...  [0.0, 0.022609487, 0.3742004, 0.0, 0.0260995...
4  s3://s3p8corentinjay/Test/Kiwi/9_100.jpg  ...  [0.9519107, 0.0, 0.0, 0.0029155284, 0.0, 0.0, ...

[5 rows x 3 columns]

In [21]: # Format de la colonne features (array)
df.loc[0, 'features'].shape
(1280,)

Réduction de dimension

In [ ]: from pyspark.ml.linalg import Vectors, VectorUDT
from pyspark.ml.feature import PCA
from pyspark.sql import functions as F

In [37]: # Conversion de la colonne features (array) en vecteurs
to_vector = F.udf(lambda x: Vectors.dense(x), VectorUDT())
sparkDf = features_df.select('path', 'label', 'features', to_vector('features')).alias('features_vec')

In [39]: # Aperçu du fichier après transformation
sparkDf.show(5)

+-----+-----+-----+-----+
| path | label | features | features_vec |
+-----+-----+-----+-----+
| s3://s3p8corentinjay/Test/Apple Golden 1/72_10... | Apple Golden 1 | [0.0, 0.05176872, ... | [0.0, 0.0517687201... |
| s3://s3p8corentinjay/Test/Kiwi/44_100.jpg | Kiwi | [0.7344478, 0.0, ... | [0.73444777272127... |
| s3://s3p8corentinjay/Test/Blueberry/30_100.jpg | Blueberry | [0.6798481, 0.169... | [0.67984808789862... |
| s3://s3p8corentinjay/Test/Apple Golden 1/85_10... | Apple Golden 1 | [0.0, 0.022609487... | [0.0, 0.0226094871... |
| s3://s3p8corentinjay/Test/Kiwi/9_100.jpg | Kiwi | [0.9519107, 0.0, ... | [0.95191067457199... |
+-----+-----+-----+-----+
only showing top 5 rows

In [41]: # Utilisation du PCA (k=2)
pcaSparkEstimator = PCA(inputCol="features_vec", outputCol="pca_features", k=2)
pca = pcaSparkEstimator.fit(sparkDf)
pca_matrix = pca.transform(sparkDf)

In [42]: # Aperçu du fichier après réduction de dimension
pca_matrix.show(5)

+-----+-----+-----+-----+-----+
| path | label | features | features_vec | pca_features |
+-----+-----+-----+-----+-----+
| s3://s3p8corentinjay/Test/Apple Golden 1/72_10... | Apple Golden 1 | [0.0, 0.05176872, ... | [0.0, 0.0517687201... | [3.90602536096789... |
| s3://s3p8corentinjay/Test/Kiwi/44_100.jpg | Kiwi | [0.7344478, 0.0, ... | [0.73444777272127... | [1.61327350587742... |
| s3://s3p8corentinjay/Test/Blueberry/30_100.jpg | Blueberry | [0.6798481, 0.169... | [0.67984808789862... | [-18.679180859050... |
| s3://s3p8corentinjay/Test/Apple Golden 1/85_10... | Apple Golden 1 | [0.0, 0.022609487... | [0.0, 0.0226094871... | [4.05523933523759... |
| s3://s3p8corentinjay/Test/Kiwi/9_100.jpg | Kiwi | [0.9519107, 0.0, ... | [0.95191067457199... | [1.91753014252427... |
+-----+-----+-----+-----+-----+
only showing top 5 rows

```

```

In [46]: # Sélection des colonnes retenues
pca_matrix_final = pca_matrix.select('path', 'label', 'pca_features')

# Aperçu des 5 premières lignes
pca_matrix_final.show(5)

+-----+-----+-----+
| path | label | pca_features |
+-----+-----+-----+
| s3://s3p8corentinjay/Test/Apple Golden 1/72_10... | Apple Golden 1 | [3.90602536096789... |
| s3://s3p8corentinjay/Test/Kiwi/44_100.jpg | Kiwi | [1.61327350587742... |
| s3://s3p8corentinjay/Test/Blueberry/30_100.jpg | Blueberry | [-18.679180859050... |
| s3://s3p8corentinjay/Test/Apple Golden 1/85_10... | Apple Golden 1 | [4.05523933523759... |
| s3://s3p8corentinjay/Test/Kiwi/9_100.jpg | Kiwi | [1.91753014252427... |
+-----+-----+-----+
only showing top 5 rows

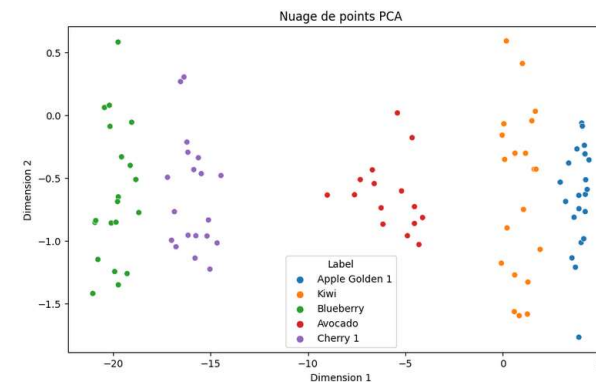
```

## Sauvegarde du résultat au format csv

```

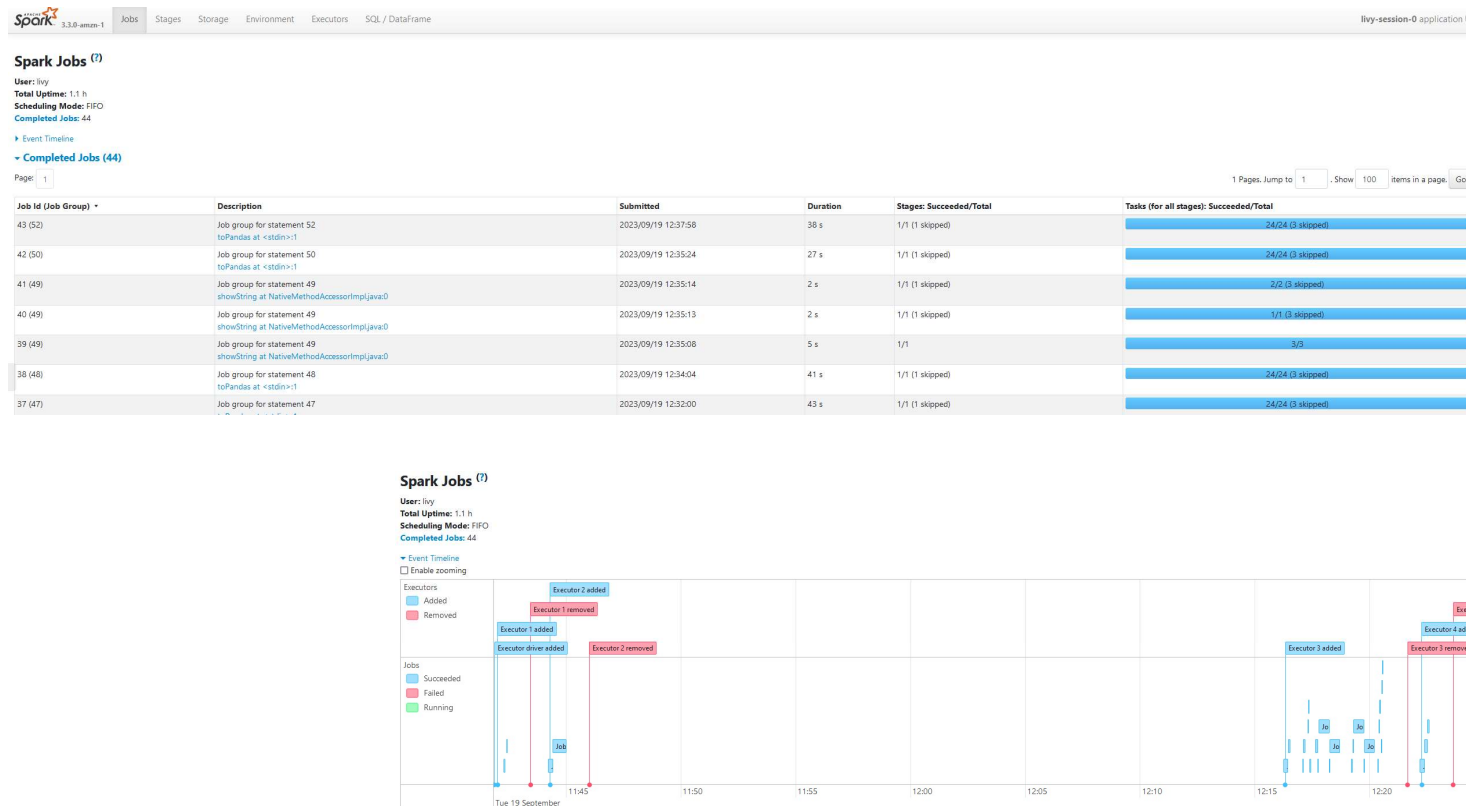
In [53]: pca_matrix_final.toPandas().to_csv('Final_results.csv')

```





## Suivi du processus sur Spark UI

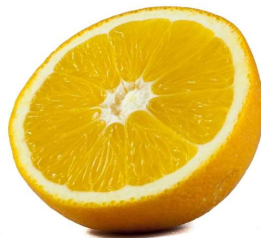




# Conclusion

## Conclusions

- ❖ Avantages :
  - ❖ Stockage quasi illimité sur S3
  - ❖ Script prêt à être utilisé sur de plus gros volumes de données
  - ❖ Spark UI (suivi des tâches/jobs)
  - ❖ Instances/clusters adaptables selon les besoins
- ❖ Précautions :
  - ❖ Gestion des coûts liés à l'architecture Cloud AWS
  - ❖ Gestion des dépendances de versions



## Axes d'évolution

- ❖ Prétraitements (recadrage, plusieurs fruits, arrière plan, etc.)
- ❖ Entraîner le modèle
- ❖ Identification de la maturité des fruits, et des fruits abimés



**Merci**

Corentin Jay

OpenClassrooms - Data Scientist

Projet 8