

## Projet 9: Réaliser un traitement dans un environnement Big Data sur le Cloud

Maodo FALL

OpenClassrooms

Soutenance du projet  
25 avril 2025



# Sommaire

- 1 **Problématique et jeu de données**
  - Problématique
  - Jeu de données
- 2 **Processus de création de l'environnement Big Data**
- 3 **Chaîne de traitement d'image**
- 4 **Démonstration sur le cloud**
- 5 **Conclusion**

# Sommaire

- 1 **Problématique et jeu de données**
  - Problématique
  - Jeu de données
- 2 Processus de création de l'environnement Big Data
- 3 Chaîne de traitement d'image
- 4 Démonstration sur le cloud
- 5 Conclusion

# Start-up Fruits



# Objectifs

Je suis Data Scientist au sein de la très jeune start-up de l'AgriTech, nommée "Fruits!", qui cherche à proposer des solutions innovantes pour la récolte des fruits.

## Objectif court terme :

- Créer une application mobile grand public de reconnaissance de fruits par photographie pour sensibiliser à la biodiversité des fruits.

## Objectif long terme :

- Moteur de classification des images de fruits
- Développer des robots cueilleurs intelligents

# Missions

Le développement de l'application mobile permettra de construire une première version de l'architecture Big Data nécessaire.

Je reprends le travail à l'aide d'un document, formalisé par un alternant qui vient de quitter l'entreprise.

## Missions :

- préparer la chaîne de traitement en local ( mise à jour du script PySpark avec une étape de réduction de dimension PCA)
- migrer la chaîne de traitement dans le cloud

# Présentation du jeu de données

Le jeu de données provient de [Kaggle](#) et est constitué de plus de 90000 images de fruits et légumes dont 22688 sont dans le sous-dossier Test composé de 131 sous-dossiers(classes) d'images.

- Les 131 classes/variétés de fruits sont, entre autres, Apple Braeburn, Banana, Kiwi, Watermelon etc...
- Les images sont de dimension  $100 * 100$  pixels.
- Le volume de données va rapidement augmenter après la livraison du projet



Figure – exemples de fruits : Pomme et mandarine

# Sommaire

- 1 Problématique et jeu de données
  - Problématique
  - Jeu de données
- 2 **Processus de création de l'environnement Big Data**
- 3 Chaîne de traitement d'image
- 4 Démonstration sur le cloud
- 5 Conclusion



# Architecture Big Data

- Le volume de données va rapidement augmenter après la livraison du projet :
  - Nécessité donc d'un passage à l'échelle pour garder de bonnes performances
- Par nature, le Big Data implique de gérer des volumes massifs de données. Donc pour les traiter efficacement :
  - On doit distribuer le stockage et le calcul (ex : Hadoop, Spark).
  - Il faut des architectures scalables

# Définitions

## Calculs distribués

Désigne un modèle d'exécution dans lequel une tâche informatique est répartie entre plusieurs ordinateurs (nœuds) qui collaborent pour résoudre un problème commun.

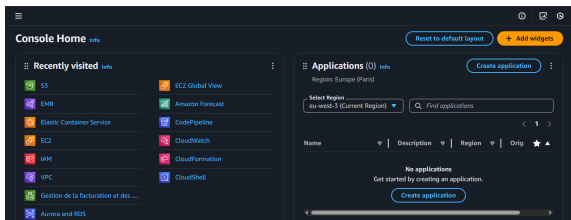
## Apach Spark

Moteur de traitement distribué open-source pour le traitement rapide de grands volumes de données. Il repose sur le paradigme de calcul en mémoire (in-memory computing), ce qui lui permet d'être plus rapide que les systèmes traditionnels comme Hadoop MapReduce, notamment pour les tâches itératives comme en machine learning.

# Choix du prestataire Cloud

## Amazon Web Service (AWS) :

- Prestataire le plus connu, offre la plus large en cloud computing
- Location de la puissance de calcul à la demande
- Adaptable à notre volume de données croissant (scalable)
- Diminution des coûts par rapport à une location de serveur complet sur une durée fixe




# Configuration de l'environnement

- **Choix du service de calcul EMR**
  - Solution de type Plateforme As A Service (PAAS)
  - Location d'instances EC2 avec applications pré-installées
  - Intégration native avec S3
  - Facilité de gestion
- **Choix du service de stockage S3**
  - Stockage illimité et élastique
  - Séparation du stockage et du calcul
  - Accès rapide et simple aux données
  - Coût optimisé
- **Service IAM (Identity and Access Management) :**


Gère de manière sécurisée les identités, les rôles et les permissions pour contrôler qui peut accéder à quelles ressources AWS et avec quels droits.

# Mon bucket S3 "p9-data-maodo"

 [Amazon S3](#) > [Compartiments](#) > p9-data-maodo






## p9-data-maodo Info

[Objets](#) | [Propriétés](#) | [Autorisations](#) | [Métriques](#) | [Gestion](#) | [Points d'accès](#)

 [Copier l'URI S3](#) [Copier l'URL](#) [Télécharger](#)

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 dans votre bucket. Vous pouvez également leur accorder explicitement des autorisations. [En savoir plus](#)

☐ Afficher les versions

<input type="checkbox"/>	Nom	Type	Dernière modification
<input type="checkbox"/>	 <a href="#">bootstrap-emr-latest.sh</a>	sh	19 Apr 2025 02:10:06 PM CET
<input type="checkbox"/>	 <a href="#">jupyter/</a>	Dossier	-
<input type="checkbox"/>	 <a href="#">Result_ACP/</a>	Dossier	-
<input type="checkbox"/>	 <a href="#">Results/</a>	Dossier	-
<input type="checkbox"/>	 <a href="#">Test/</a>	Dossier	-

# Mise en place du service EMR

Nom et applications - [info](#)

Donnez un nom à votre cluster et choisissez les applications que vous voulez y installer.

Nom

clone\_vt\_fruits\_p9

Version Amazon EMR [info](#)

Cette version contient un ensemble d'applications susceptibles d'être installées sur votre cluster.

emr-7.8.0

Offre d'applications

Spark  
Interactive

Core  
Hadoop

Flink

HBase

Presto

Trino

Custom

☐ AmazonCloudWatchAgent

☐ Flink 1.20.0

☐ HBase 2.6.1

☐ 1.300032.2

☐ Hive 5.1.3

☐ Hue 4.11.0

☐ Livy 0.8.0

☐ Pig 0.17.0

☐ TensorFlow 2.16.1

☐ Zappellin 0.11.1

☐ Hadoop 3.4.1

☐ JupyterEnterpriseGateway 2.6.0

☐ Oozie 5.2.1

☐ Presto 0.287

☐ Tez 0.10.2

☐ ZooKeeper 3.9.3

☒ Hive 5.1.3

☒ JupyterHub 1.5.0

☐ Phoenix 5.2.1

☒ Spark 3.5.4

☐ Trino 467

**Configuration de cluster – requies** info

Choisissez une méthode de configuration pour les groupes de nœuds primaires, principaux et de secours.

**Groupe d'instances uniformes**

Choisissez le même type d'instance EC2 et la même option d'achat (à la demande ou Spot) pour tous les nœuds de votre groupe de nœuds. [En savoir plus](#)

**Flottes d'instances flexibles**

Choisissez parmi la plus grande variété d'options de provisionnement pour les instances EC2. Diversifiez les types d'instances et les options d'achat pour utiliser une stratégie d'allocation. [En savoir plus](#)

**Groupes d'instances uniformes**

**Primaire**

Choisir un type d'instance EC2

m5.xlarge  
4 vCore    16 GiB mémoire  
EBS uniquement stockage  
Prix à la demande : 0,224 USD par insta...  
Prix Spot le plus bas : 0,076 USD (eu-west-...)

Actions ▼

☐ Utiliser la haute disponibilité

Lancez des clusters hautement disponibles et plus résilients avec trois nœuds primaires sur des instances à la demande. La configuration s'applique pendant toute la durée de vie de votre cluster. [En savoir plus](#)

**Configuration de nœud – facultatif**

**Unité principale**

Choisir un type d'instance EC2

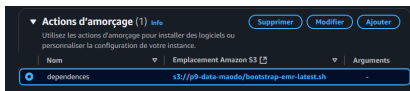
m5.2xlarge  
8 vCore    32 GiB mémoire  
EBS uniquement stockage  
Prix à la demande : 0,448 USD par insta...  
Prix Spot le plus bas : 0,132 USD (eu-west-...)

Actions ▼

**Configuration de nœud – facultatif**

### Figure – Applications et groupes d'instances

# Mise en place du service EMR



```

1  #!/bin/bash
2
3  sudo python3 -m pip install -U setuptools
4  sudo python3 -m pip install -U pip
5  sudo python3 -m pip install wheel
6  sudo python3 -m pip install pillow
7  sudo python3 -m pip install pandas
8  sudo python3 -m pip install numpy
9  sudo python3 -m pip install matplotlib
10 sudo python3 -m pip install pyarrow
11 sudo python3 -m pip install boto3
12 sudo python3 -m pip install s3fs
13 sudo python3 -m pip install fsspec
14
15 # Ajout de tensorflow et du module tree
16 sudo python3 -m pip install tensorflow
17 sudo python3 -m pip install dm-tree

```

Figure – Actions d'amorçage - Bootstrap

# Mise en place du service EMR

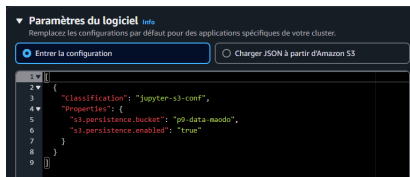


Figure – Paramètres logiciel

**Rôle Identity and Access Management (IAM) - requis**

**Fonction du service**  
 EMR\_DefaultRole [↗](#)

**Profil d'instance**  
 EMR\_EC2\_DefaultRole [↗](#)

Figure – rôles IAM

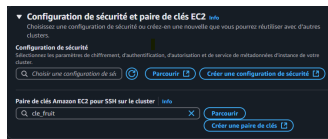
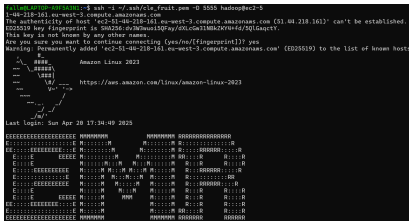


Figure – clés EC2 SSH



# Mise en place du service EMR



### Figure – Connexion à l'EMR

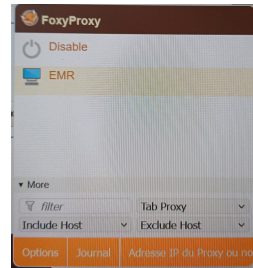


Figure –  
Paramétrage de  
FoxyProxy

# Sommaire

- 1 Problématique et jeu de données
  - Problématique
  - Jeu de données
- 2 Processus de création de l'environnement Big Data
- 3 Chaîne de traitement d'image
- 4 Démonstration sur le cloud
- 5 Conclusion



# Chargement des images

```
images = spark.read.format("binaryFile") \
    .option("pathGlobFilter", "*.jpg") \
    .option("recursiveFileLookup", "true") \
    .load(PATH_Data)
```

```
images.show(5)
```

path	modificationTime	length	content
s3://p9-data-maad...	2025-04-14 12:44:25	7353	[FF D8 FF E0 00 1...
s3://p9-data-maad...	2025-04-14 12:44:25	7350	[FF D8 FF E0 00 1...
s3://p9-data-maad...	2025-04-14 12:44:25	7349	[FF D8 FF E0 00 1...
s3://p9-data-maad...	2025-04-14 12:44:25	7348	[FF D8 FF E0 00 1...
s3://p9-data-maad...	2025-04-14 12:44:25	7328	[FF D8 FF E0 00 1...

only showing top 5 rows

Figure – Chargement des images

```
images = images.withColumn('label', element_at(split(images['path'], '/'), -2))
print(images.printSchema())
print(images.select('path', 'label').show(5, False))
```

```
root
```

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

```
None
```

path	label
s3://p9-data-maad.../Test/Watermelon/r_106_100.jpg	Watermelon
s3://p9-data-maad.../Test/Watermelon/r_109_100.jpg	Watermelon
s3://p9-data-maad.../Test/Watermelon/r_108_100.jpg	Watermelon
s3://p9-data-maad.../Test/Watermelon/r_107_100.jpg	Watermelon
s3://p9-data-maad.../Test/Watermelon/r_95_100.jpg	Watermelon

Figure – extraction des labels

# Extraction de features et PCA

- Redimensionnement et prétraitement des images
- MobileNetV2 : CNN pré-entraîné sur la base ImageNet pour la détection de features et la classification d'images
- Transfer Learning : MobileNetV2 pré-entraîné (rapide, faible dimensionnalité sortie)
- Chargement du modèle sur le driver et diffusion des poids sur les workers(broadcast)
- Réduction de dimension : PCA
- Stockage des résultats au format parquet dans le bucket s3

# Résultats

Amazon S3 > Comparaisons > gl-data-mongo > Results

### Results/

Objets (25)

Les objets sont les weblogs fondamentaux stockés dans Amazon S3, vous pouvez utiliser l'interface Amazon S3 pour obtenir une liste de tous les objets de votre compte. Pour en savoir plus, consultez les tutoriels des utilisateurs. [En savoir plus](#)

Rechercher des objets en fonction du profil

Afficher les versions

Nom	Type	Dernière modification	Taille
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:17:14 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:18:00 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:18:56 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:17:25 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:17:17 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:17:44 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:11 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	20 Apr 2015 08:12:08 PM CEST	

Figure – Extraction features

Amazon S3 > Comparaisons > gl-data-mongo > Results\_ACP

### Result\_ACP/

Objets (25)

Les objets sont les weblogs fondamentaux stockés dans Amazon S3, vous pouvez utiliser l'interface Amazon S3 pour obtenir une liste de tous les objets de votre compte. Pour en savoir plus, consultez les tutoriels des utilisateurs. [En savoir plus](#)

Rechercher des objets en fonction du profil

Afficher les versions

Nom	Type	Dernière modification	Taille
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:42 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:16 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:15 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:29 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	
gl-data-mongo-2015-04-21-14-14-14	parquet	21 Apr 2015 08:42:18 PM CEST	

Figure – Features ACP

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺



# Sommaire

- 1 Problématique et jeu de données
  - Problématique
  - Jeu de données
- 2 Processus de création de l'environnement Big Data
- 3 Chaîne de traitement d'image
- 4 **Démonstration sur le cloud**
- 5 Conclusion



# Sommaire

- 1 Problématique et jeu de données
  - Problématique
  - Jeu de données
- 2 Processus de création de l'environnement Big Data
- 3 Chaîne de traitement d'image
- 4 Démonstration sur le cloud
- 5 Conclusion

# Conclusion

- Mise en place d'une chaîne complète de traitement d'images dans un environnement Big Data sur AWS
- Utilisation de :
  - Amazon S3 pour le stockage illimité et accessible
  - Amazon EMR pour le calcul distribué avec Apache Spark
  - JupyterHub pour le développement et le pilotage des notebooks
- Intégration d'un modèle pré-entraîné (MobileNetV2) pour l'extraction des features
- Réduction de dimension par PCA et stockage des résultats au format Parquet dans S3
- Architecture conçue pour être scalable et évolutive
- Base technique solide pour un futur moteur de reconnaissance de fruits

MERCI BEAUCOUP !