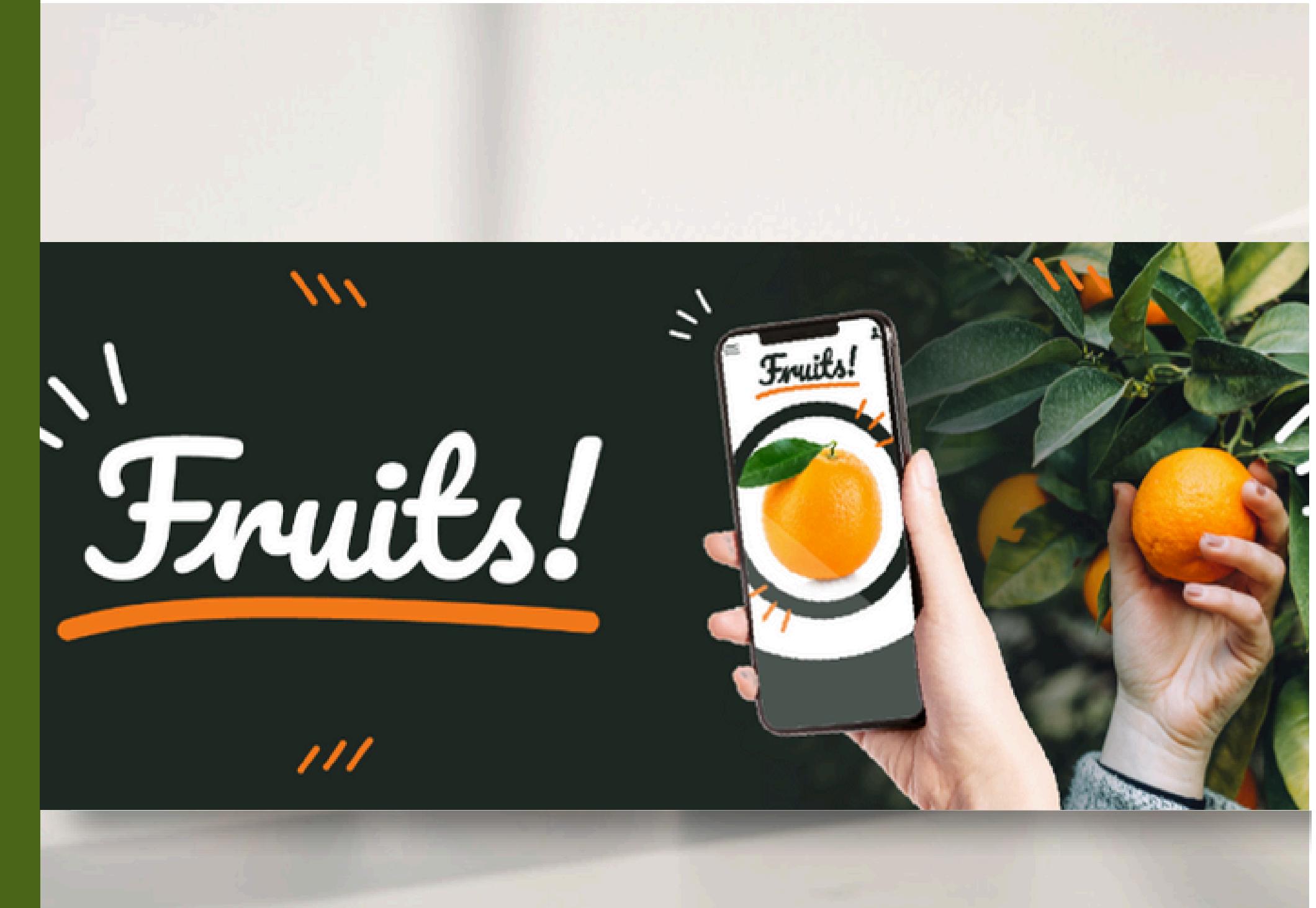


# Réalisez un traitement dans un environnement Big Data

---



Présenté par Thierry KAPPE

# PLAN



- 01 *Contexte et problématique*
- 02 *Présentation du jeu de données*
- 03 *Processus et création de l'environnement Big Data*
- 04 *Chaîne de traitement des images*
- 05 *Démonstration exécution du script PySpark sur le cloud*

## CONTEXTE & PROBLEMATIQUE



Fruits! est une start-up de l'AgriTech qui cherche à proposer des **solutions innovantes pour la récolte des fruits**. Sa volonté étant de **préserver la biodiversité** des fruits en permettant des traitements spécifiques pour chaque espèce de fruits et en **développant des robots cueilleurs intelligents**.

Fruits! souhaite se faire connaître grâce à **une application au grand public** qui permettrait de prendre en photo un fruit et d'obtenir les informations sur ce fruit.

*Cet application permettrait :*

- **Sensibiliser le grand public sur la biodiversité des fruits;**
- **Mettre en place une première version d'un moteur de classification des images des fruits et de l'architecture Big Data nécessaire.**

Notre mission est de :

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

# PRESENTATION

## JEU DE DONNEES



Données issues d' un kernel Kaggle

- 90423 images et 131 classes
- 2 jeux de données training(67692) et test set (22688)

131 dossiers :

- images représentant un fruit ou un légume
- images avec fond blanc et sous 3 axes
- taille 100 x 100 pixels en jpg
- plusieurs variétés pour certains fruits

Red Apple Category 1					
Red Apple Category 2					
Red Apple Category 3					
Red Apple Category 4					
Red Apple Category 5					
Banana					
Orange					
Pomegranate					

# PROCESSUS CREATION ENVIRONNEMENT BIG DATA



## Pourquoi un environnement Big Data?

**Big Data (données massives)** : données telles que les solutions classiques de stockage, de gestion et de traitement ne suffisent plus.

### Les 3V du Big Data

Le **Volume** des données générées nécessite de repenser la manière dont elles sont stockées.

- Avec une croissance rapide de l'application, le volume de données collectées augmentera considérablement.
- Le Big Data permet de stocker, gérer et analyser ces grandes quantités de données de manière évolutive.

La **Vélocité** à laquelle nous parvenant ces données implique de mettre en place des solutions de traitement en temps réel qui ne paralysent pas le reste de l'application.

- Le nombre d'utilisateurs et de photos prises augmentera rapidement, générant une quantité importante de données en temps réel.
- Le Big Data offre les outils nécessaires pour traiter et analyser ces données à grande vitesse.

Les données se présentent sous une grande **Variété** de formats : structurées ( doc JSON), semi structurés ( fichiers de log), non structurées (textes, images)...

- L'application mobile générera différents types de données, tels que des images de fruits, des informations associées et des métadonnées.
- Le Big Data permet de traiter et d'analyser efficacement ces multiples sources de données hétérogènes.

# PROCESSUS

CREATION  
ENVIRONNEMENT  
BIG DATA



## Les outils du Big Data?

- **Calculs distribués** : distribution du stockage et des traitement des données sur plusieurs unités de calcul réparties en clusters, au profit d'un seul projet afin de diviser le temps d'exécution d'une requête.
  - **Apache Spark** : framework open-source permettant de traiter des bases de données massives en utilisant le calcul distribué (in-memory). Outil qui permet de gérer et de coordonner l'exécution de tâches sur des données à travers un groupe d'ordinateurs.
  - **Algorithme MapReduce** :
    - a. Largement utilisé pour le traitement parallèle et distribué de grandes quantités de données.
    - b. Permet de diviser les données en ensembles plus petits, de les traiter indépendamment (MAP) et de les agréger pour obtenir le résultat final (REDUCE).
  - Développement des scripts en **pySpark**, la librairie python (proche de pandas) permettant de communiquer avec Spark.
- ⇒ **Avantages** : évolutivité (ajout de ressources supplémentaires), performances (accélération du temps de calculs), tolérance aux pannes (plus résilients aux pannes ou erreurs).

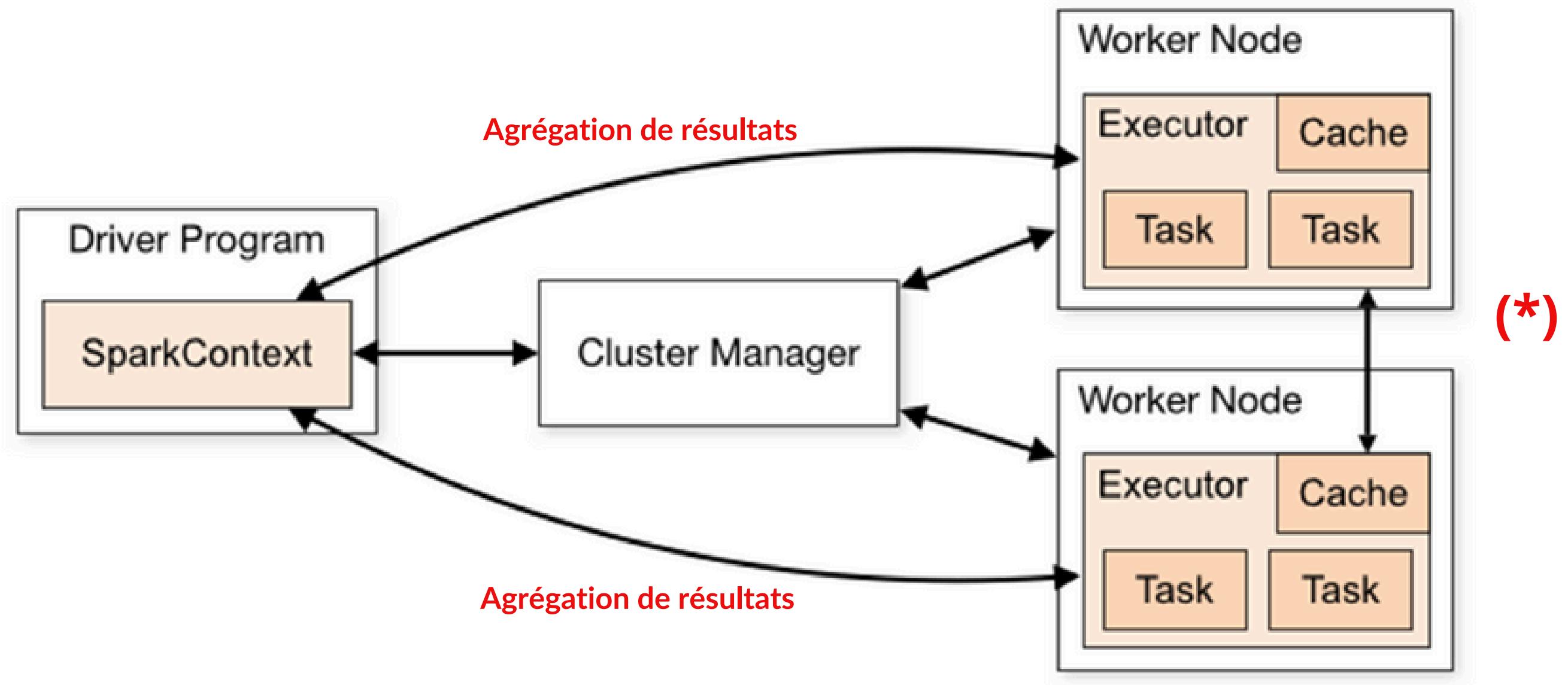
# PROCESSUS

CREATION  
ENVIRONNEMENT  
BIG DATA



## Le processus de calcul distribué?

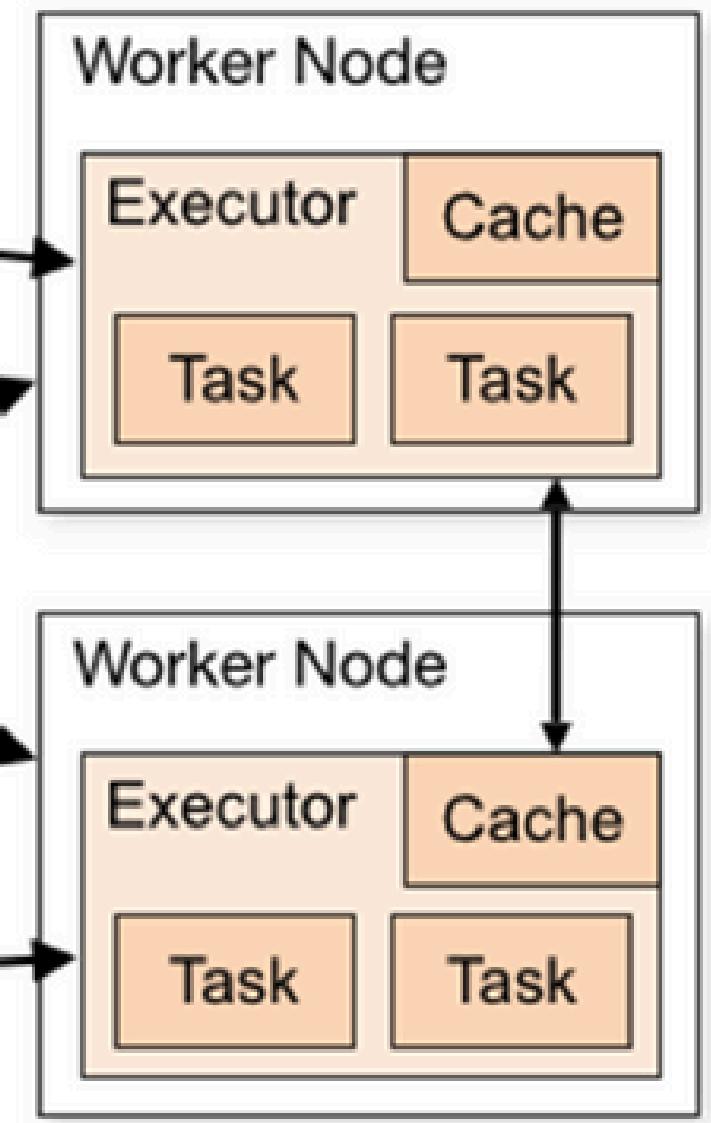
1 partition + opération = résultats



### Application Spark:

Le **driver** distribue et planifie les tâches entre les différents **exécuteurs** qui les exécutent et permettent un traitement réparti. Il est le responsable de l'exécution du code sur les différentes machines.

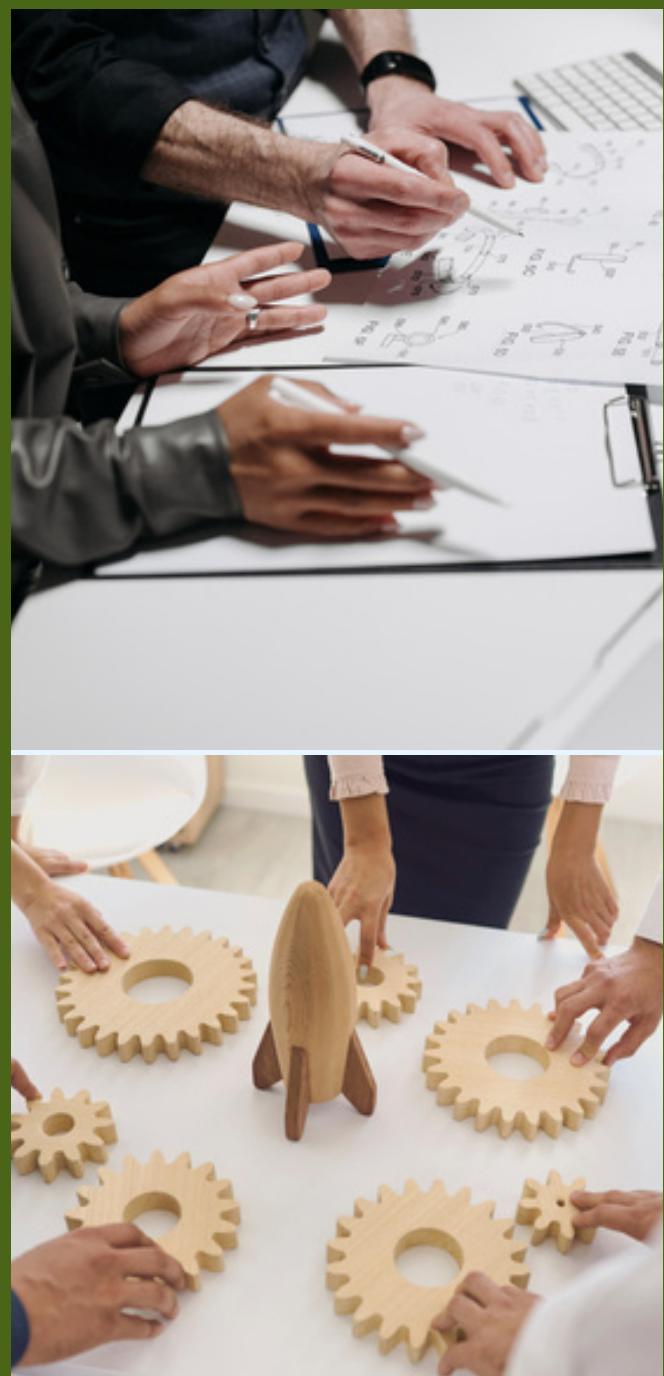
**Cluster Manager** : assure le suivi des ressources disponibles.



(\*) Une machine peut traiter plusieurs partitions et résultats mis en cache

# PROCESSUS

## CREATION ENVIRONNEMENT BIG DATA

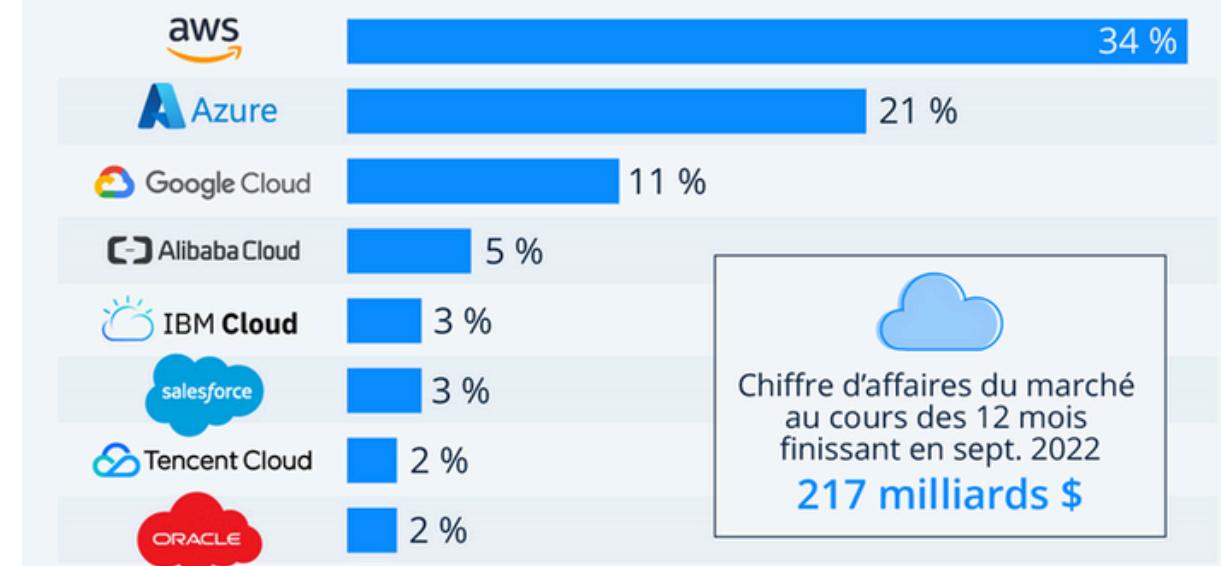


# Déploiement de la solution dans le cloud

- Louer de la puissance de calcul à la demande : pouvoir, quel que soit la charge de travail, obtenir suffisamment de puissance de calcul pour pouvoir traiter les images, même si le volume de données venait à fortement augmenter.
- Diminuer les coûts si l'on compare les coûts d'une location de serveur complet sur une durée fixe (1 mois, 1 année...).
- Le prestataire le plus connu et qui offre à ce jour l'offre la plus large dans le cloud est **Amazon Web Services (AWS)**.

## Cloud : les géants de la tech se partagent le marché mondial

Part de marché mondiale des principaux fournisseurs de services cloud au troisième trimestre 2022 \*



\* inclut les services PaaS (platform as a service), IaaS (infrastructure as a service), ainsi que les services de cloud privé hébergé.

Source : Synergy Research Group



# PROCESSUS

CREATION  
ENVIRONNEMENT  
BIG DATA



# Briques d'architecture Big Data avec AWS



IAM Contrôles d'accès



S3

Stockage

- Images
- Résultats
- Notebook



EMR

Cluster de calculs distribués

- Traitement des images

# PROCESSUS

CREATION  
ENVIRONNEMENT  
BIG DATA



# Configuration de l'environnement de travail

Tableau de bord IAM

Recommandations de sécurité

- L'utilisateur racine dispose de la MFA.
- Désactiver ou supprimer les clés d'accès pour l'utilisateur racine.

Gérer les clés d'accès

Ressources IAM

Groupes d'utilisateurs	Utilisateurs	Rôles	Politiques	Fournisseurs d'identité
0	0	7	1	0

Compte AWS

Informations d'identification de sécurité

Créer

Quick Links

- Service IAM (Identity and Access Management)
  - a.Création d'un utilisateur
  - b.Gestion des droits (contrôle S3) (Politiques)
  - c.Création d'une paire de clés qui nous permettra de nous connecter devoir saisir systématiquement login/mot de passe (**Informations d'identification de sécurité / Crée une clé d'accès**)



- Installation et configuration de AWS Cli ( interface en ligne de commande d'AWS, permet d'interagir avec les différents services d'AWS)

# PROCESSUS

## CREATION ENVIRONNEMENT BIG DATA



# Stockage des données sur S3 (Simple Storage Service)

## S3 : Solution pour la gestion du stockage des données

- Stockage d'une grande variété d'objets (fichiers, image, vidéos...)
- Évolutivité avec espace disponible illimité.
- Indépendant des serveurs EC2.
- Accès aux données très rapide.
- Possibilité de définir des politiques d'accès IAM pour contrôler les autorisations. d'accès aux buckets et aux objets.
- Chiffrement côté serveur pour sécuriser les données stockées dans S3.
- Classes de stockage (options) adaptées à l'utilisation.



## Mise en oeuvre :

- Création d'un compartiment (“bucket”) : **tkap9**
- Choisir la même région pour les serveurs EC2 et S3.
- Chargement des données sur le bucket S3 :
  - a. Fichier de configuration avec amorçage
  - b. Répertoire des images **Test**
  - c. Notebook avec Script (JupyterHub)
- Écriture des résultats dans le répertoire **Results**.

The screenshot shows the AWS S3 console interface for the 'tkap9' bucket. The top navigation bar includes 'Amazon S3', 'Compartiments', and 'tkap9'. The main area displays 'Objets (6)' with a table listing files and folders. The table has columns for 'Nom', 'Type', 'Dernière modification', 'Taille', and 'Classe de stockage'. The objects listed are:

Nom	Type	Dernière modification	Taille	Classe de stockage
bootstrap.sh	sh	08 Nov 2024 03:29:58 PM CET	293.0 o	Standard
bootstrap1.sh	sh	19 Nov 2024 11:30:58 AM CET	253.0 o	Standard
bootstrap2.txt	txt	19 Nov 2024 12:26:06 PM CET	176.0 o	Standard
jupyter/	Dossier	-	-	-
Results/	Dossier	-	-	-
Test/	Dossier	-	-	-

## PROCESSUS

CREATION  
ENVIRONNEMENT  
BIG DATA



# Création d'un cluster de calculs distribués avec EMR (Elastic MapReduce)

- Elastic MapReduce (EMR) : plateforme permettant *d'exécuter des traitements de données distribuées à grande échelle*, en utilisant des frameworks tels que Hadoop et Spark.
- Il utilise des *instances EC2* (Elastic compute cloud, serveur) avec des *applications préinstallées* et configurées pour créer et gérer le cluster de calculs distribués.
- Le service est *entièrement géré par AWS*.

⇒ **Avantages** : évolutivité, flexibilité, gestion simplifiée.

- **Création du serveur EMR en 4 étapes :**
  - a. Configuration logiciel
  - b. Configuration matériel
  - c. Actions d'amorçage
  - d. Options de sécurité



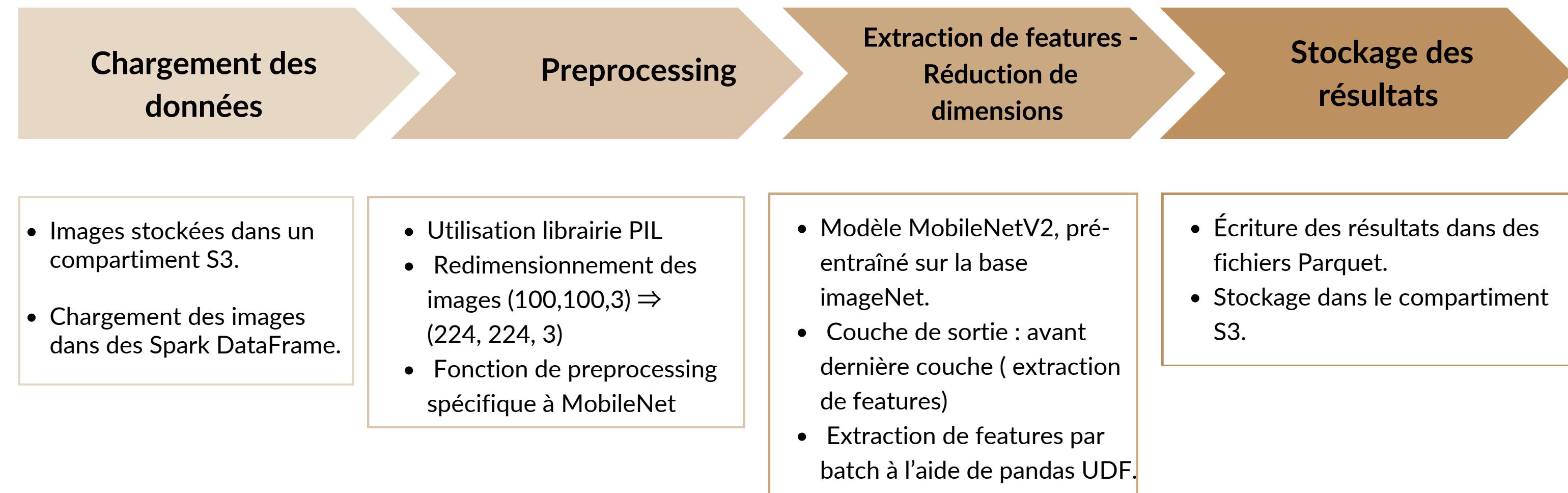


## CHAINE TRAITEMENT DES IMAGES



# Chaîne de traitement d'images

- Exécution du Notebook depuis *JupyterHub*, hébergé sur notre serveur EMR.
- Utilisation d'un *kernel pySpark*.
- Démarrage d'une *session Spark* à l'exécution de la première cellule.



## CHAINE

### TRAITEMENT DES IMAGES



# Chargement des données

Chargement des données avec `spark.read()` :

- Traitement des fichiers en tant que **données binaires**.
- À l'emplacement spécifié (compartiment S3), recherche récursive dans les sous-répertoires des fichiers avec l'extension `".jpg"`.
- Chargement des images dans un **DataFrame Spark**.

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

Schéma du Spark DataFrame

- Ajout de la colonne **label** issu du chemin d'accès des fichier : **label** représente la catégorie de l'image (nom du fruit), avant dernier élément (-2) du "path".

```
+-----+-----+-----+
|      path|modificationTime|length|      content|
+-----+-----+-----+
|s3://tkap9/Test/W...|2024-11-07 15:48:40| 7353|[FF D8 FF E0 00 1...
|s3://tkap9/Test/W...|2024-11-07 15:48:40| 7350|[FF D8 FF E0 00 1...
|s3://tkap9/Test/W...|2024-11-07 15:48:40| 7349|[FF D8 FF E0 00 1...
|s3://tkap9/Test/W...|2024-11-07 15:48:40| 7348|[FF D8 FF E0 00 1...
|s3://tkap9/Test/W...|2024-11-07 15:48:41| 7328|[FF D8 FF E0 00 1...
+-----+-----+-----+
only showing top 5 rows
```

```
+-----+-----+
|path|      label|
+-----+
|s3://tkap9/Test/Watermelon/r_106_100.jpg|Watermelon|
|s3://tkap9/Test/Watermelon/r_109_100.jpg|Watermelon|
|s3://tkap9/Test/Watermelon/r_108_100.jpg|Watermelon|
|s3://tkap9/Test/Watermelon/r_107_100.jpg|Watermelon|
|s3://tkap9/Test/Watermelon/r_95_100.jpg|Watermelon|
+-----+
only showing top 5 rows
```

# Modèle MobileNetV2 avec Transfer Learning

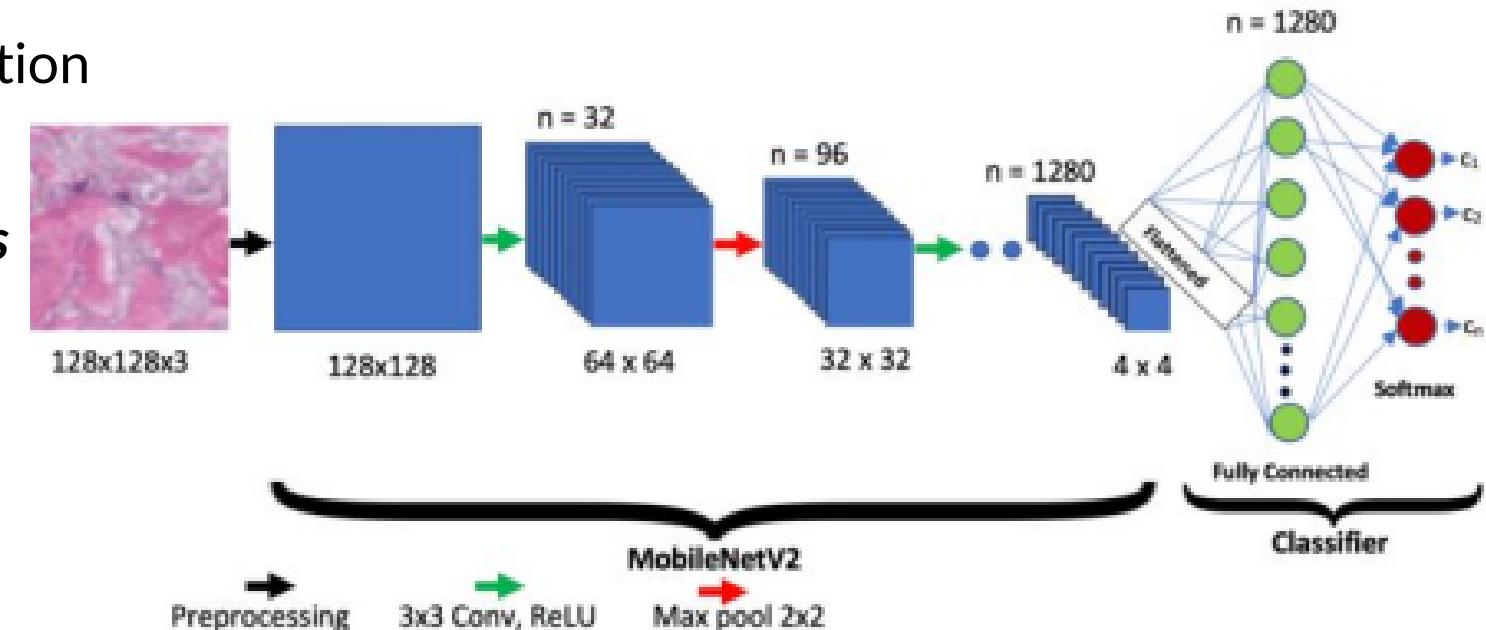
## CHAINE

### TRAITEMENT DES IMAGES



Choix du modèle **MobileNetV2** :

- Modèle de réseau de neurones convolutifs (CNN) pré-entraîné sur la base ImageNet pour la détection de features et la classification d'images.
- Spécialement conçu sur pour **appareils mobiles avec ressources limitées** :
  - a. Rapidité d'exécution (adapté pour le traitement d'un gros volume de données).
  - b. Faible dimensionnalité du vecteur de sortie (1,1,1280).



Transfer Learning :

- Consiste à utiliser la connaissance déjà acquise par un modèle entraîné (ici MobileNetV2) en l'adaptant à notre problématique.
- Crée une instance du modèle MobileNetV2 pré-entraîné avec les poids du jeu de données ImageNet, incluant la couche de classification finale.

Préparation du modèle :

- Crée un nouveau modèle avec pour couche de sortie l'avant-dernière couche (extraction des features images) du modèle MobileNetV2 .
- Dimension vecteur de sortie (1, 1, 1280).
- **Diffusion des poids** avec `sparkContext.broadcast()` de PySpark :
  - a. Chargement du modèle sur le driver puis diffusion des poids aux workers.
  - b. Permet de distribuer une variable à travers le cluster afin qu'elle soit disponible pour tous les nœuds de calcul.

## CHAINE TRAITEMENT DES IMAGES



# Pre-processing

- Dimensions des images d'origine : (100,100,3) / (100\*100 pixels et 3 canaux de couleur RVB).
- Dimensions des images attendues en entrée de MobileNetV2 : (224, 224, 3)

⇒ Nous devons les redimensionner avant de les confier en entrée du modèle.

- Avec **librairie PIL** (Python Imaging Library) :
  - a. Ouverture des données binaires de l'image en tant qu'image.
  - b. **Redimensionnement de l'image** à une taille (224, 224, 3).
- Application de la fonction preprocess\_input de TensorFlow, **fonction de prétraitement spécifique** pour prétraiter les images avant de les passer en entrée du modèle MobileNet.



## CHAINE

### TRAITEMENT DES IMAGES



## Traitement des données et stockage

### *Extraction de features*

- A partir des images pré-traitées, répartition des données et application itérative du modèle aux batches de données d'images pour en extraire les features, en utilisant un pandas UDF.
- Résultat : DataFrame avec colonnes d'origine + features images.
- Les données sont traitées en parallèle sur différents nœuds, ce qui permet d'utiliser la puissance de calcul distribué du cluster.
- De plus, le modèle est chargé une seule fois et réutilisé pour tous les batches de données, ce qui évite les coûts de chargement répétitifs et réduit la consommation de mémoire.

### *Réduction de dimensions PCA :*

*Analyse en composantes principales pour réduire la dimensionnalité tout en préservant un maximum d'informations.*

### *Stockage des résultats :*

- Données du DataFrame écrites dans un fichier Parquet (*format de stockage optimisé pour le Big Data*).
- Mode "overwrite" : si le fichier de destination existe déjà, il sera écrasé.
- *Dans le répertoire "Results" du compartiment S3.*

# DEMONSTRATION

## EXECUTION SCRIPT

### PYSPARK SUR LE

### CLOUD



A screenshot of the AWS EMR console showing the details of a cluster named "tka\_p9\_cluster1". The cluster is in the "En attente" (Pending) state. The "Récapitulatif" section provides an overview of the cluster's configuration, including its ID (j-3DRENGHNSBFHO), installed applications (Ganglia 3.7.2, Hadoop 3.2.1, Hive 3.1.3, Hue 4.10.0, JupyterEnterpriseGateway 2.1.0, JupyterHub 1.4.1, Pig 0.17.0, Spark 3.2.1, Zeppelin 0.10.0), and its capacity (1 primary node, 1 master unit, 2 tasks). The "Gestion des clusters" section lists the log destination (aws-logs-686255964105-eu-west-1/elasticmapreduce), application interfaces (Spark History Server, YARN Timeline Service, Tez UI), and the primary node's DNS (ec2-3-249-234-90.eu-west-1.compute.amazonaws.com).

A screenshot of the AWS EMR console showing the "Interfaces utilisateur d'application sur le noeud primaire" (Application User Interfaces on the Primary Node) for the "tka\_p9\_cluster1" cluster. The "Hue" interface is highlighted with a red oval. The "URL de l'interface utilisateur" column lists various endpoints for different services, such as Ganglia, HDFS, and Spark History Server.

A screenshot of a JupyterHub session titled "P9\_tkappe\_notebook\_cloud" (auto-sauvegardé). The toolbar at the top includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widow", "Help", "Logout", "Control Panel", "Non fiable", and "PySpark". The "PySpark" button is also highlighted with a red oval.

Déployez un modèle dans le cloud

Contexte :

## CONCLUSION



### **Mise en place d'une architecture Big Data :**

**EMR (Elastic MapReduce) avec Apache Spark pour le traitement distribué des données volumineuses, qui nous permet d'instancier un cluster avec les programmes et librairies nécessaires : Spark, Hadoop, JupyterHub, TensorFlow...**

**S3 (Simple Storage Service) pour le stockage des données : images d'origine et résultats.**

**IAM (Identity & Access Management) pour la gestion des contrôles d'accès.**

Appropriation de la **chaîne de traitement d'images** : chargement des données, preprocessing, préparation du modèle MobileNetV2 avec transfert learning et diffusion des poids, extraction de features, réduction de dimensions.

**L'utilisation d'un environnement Big Data offre des avantages pour “Fruits!” en termes de traitement des données, de performance, d'évolutivité et de préparation pour l'avenir :**

**Il sera facile de faire face à une montée de la charge de travail et passer à l'échelle en redimensionnant le cluster de machines.**

**Les coûts augmenteront en conséquence mais resteront inférieurs aux coûts engendrés par l'achat de matériels ou par la location de serveurs dédiés.**

**L'architecture Big Data pose les bases pour des fonctionnalités avancées, comme l'entraînement de modèles de classification des fruits.**



Merci.