

# DÉPLOIEMENT D'UN MODÈLE DANS LE CLOUD

UTILISATION DU TRANSFERT LEARNING DANS UN CONTEXTE DE BIG DATA



## Fruits!

# CONTEXTE GÉNÉRAL

## ■ Contexte :

- La start-up **Fruits!** tente de développer des robots cueilleurs capables de reconnaître les fruits. En première approche, l'entreprise souhaite mettre en ligne une application de reconnaissance de fruit.
- Le volume de données peut devenir très important, requérant une architecture spécifique du Big Data.
- Un alternant a réalisé un premier script (notebook) posant les bases d'une classification dans un contexte Big Data.



# Fruits!

## ■ Mission :

1. Vérifier le code de l'alternant en local
2. Ajouter une étape de standardisation et réduction de dimension (PCA) dans un contexte de calculs distribués (SPARK)
3. Création d'un environnement Cloud (AWS) pour déployer le code.



# I) DÉMARCHE DE MODÉLISATION

- 1) PRÉSENTATION DES DONNÉES
  - 2) PREPROCESSING
  - 3) EXTRACTION DE FEATURES ET RÉDUCTION DE DIMENSION
- 

# I) PRÉSENTATION DES DONNÉES

## Présentation des données

- Jeu de données Fruits360 provenant de Kaggle
  - Données d'entraînement : 67692 images
  - Données supplémentaires (test) : 22688 images
  - 131 variétés de fruits et légumes

**Objectif :** Utiliser la méthode de Transfert Learning afin d'extraire des « features » de ces images (prémices d'une classification)

## 2) LES DIFFÉRENTES ÉTAPES DE PREPROCESSING

Présentation des données

- Jeu de données Fruits360 provenant de Kaggle
  - Données d'entraînement : 67692 images
  - Données supplémentaires (test) : 22688 images
  - 131 variétés de fruits et légumes

**Objectif :** Utiliser la méthode de Transfert Learning afin d'extraire des « features » de ces images (prémices d'une classification)

Pré-traitement

Importation

Reformatage

Image vers Vecteur

MobileNetV2 Preprocessing



100x100x3

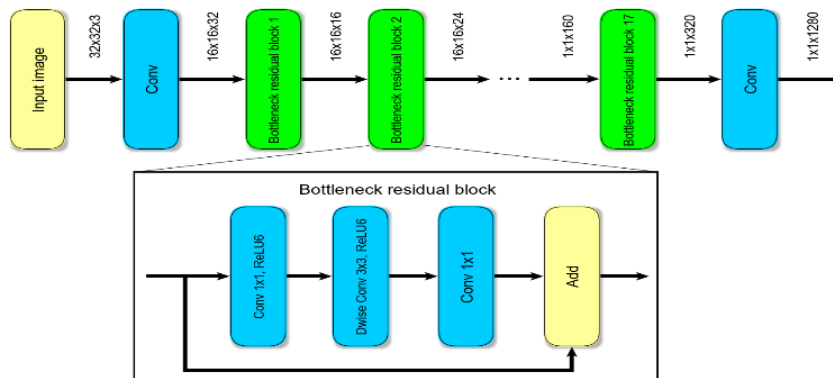
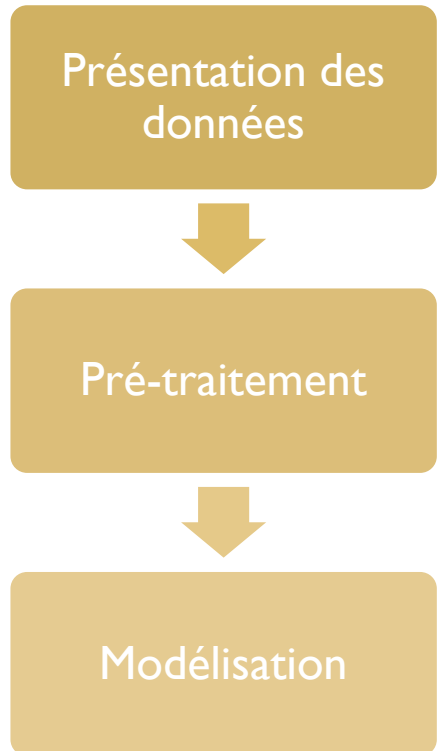


224x224x3

Vecteur 3D  
[0,255]

Vecteur 3D  
[-1,1]

### 3) EXTRACTION DE FEATURES ET RÉDUCTION DE DIMENSION



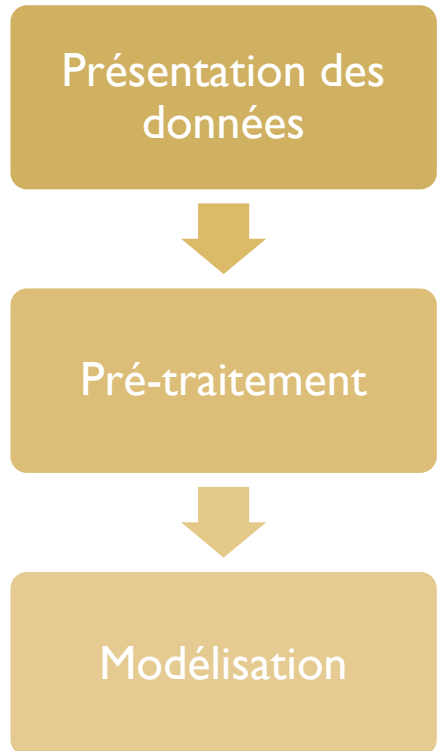
Spécificités : Bottleneck Residual Blocks

- Réduction du nombre de paramètres
- Temps de calculs
- Poids du modèle

**Modèle :** Sandler et al., MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Computer Vision and Pattern Recognition*, 2018  
**Src image :** Seidaliyeva et al., Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensor*, 2020

Modèle	Paramètres	Output
MobileNetV2	2,257,984	1280

### 3) EXTRACTION DE FEATURES ET RÉDUCTION DE DIMENSION



Extraction de features

Dimension : 1280



Standardisation

Dimension : 1280



Reduction de dimension (ACP)

Dimension : 200

■ 80% de la variance totale




Label	Features	Scaled_features	PCA_features
Apple Braeburn	[0.7831, 0.05896 ...]	[1.4111, 0.0823 ...]	[7.253, -6.478 ...]

**Problème :** Comment traiter simultanément des milliers (millions) d'images rapidement ?

1. Environnement de calculs distribués
2. Architecture Cloud pour le passage à l'échelle



## II) ENVIRONNEMENTS DE CALCULS DISTRIBUÉS

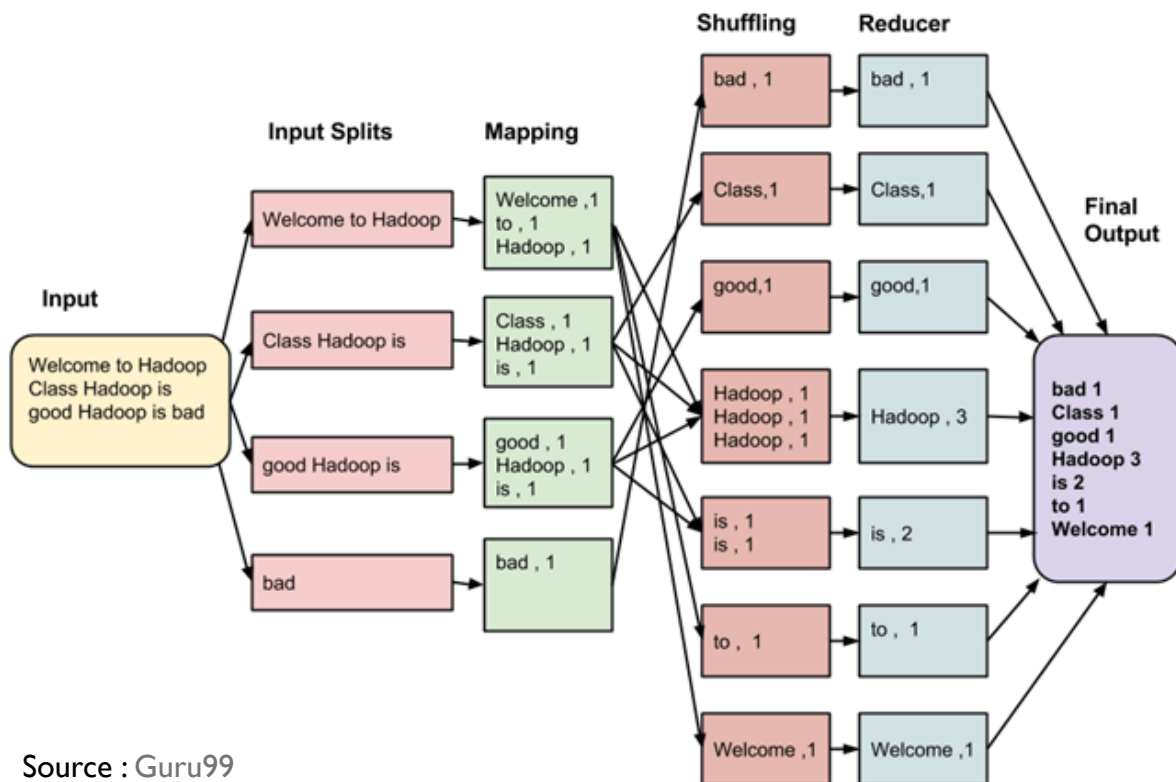
- 1) LA MÉTHODE MAPREDUCE
  - 2) PRÉSENTATION DE L'ENVIRONNEMENT HADOOP
  - 3) PRÉSENTATION DE L'ENVIRONNEMENT SPARK
- 



# I) LA MÉTHODE MAPREDUCE

**Paradigme** : Diviser pour mieux régner

➤ Méthode MapReduce

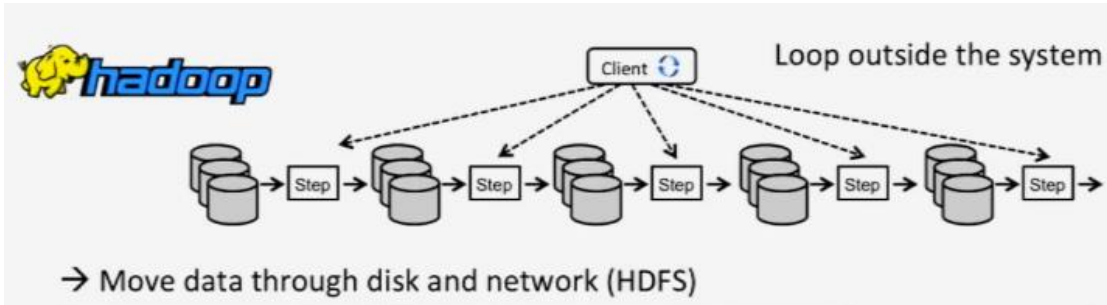


1. *Splits* : séparation des données
2. **Mapping** : Application d'une fonction sur la plus petite unité (clé-valeur)
3. *Shuffling* : Regroupement des différentes unités
4. **Reducing** : Agrégation des résultats

## 2) PRÉSENTATION DE L'ENVIRONNEMENT HADOOP

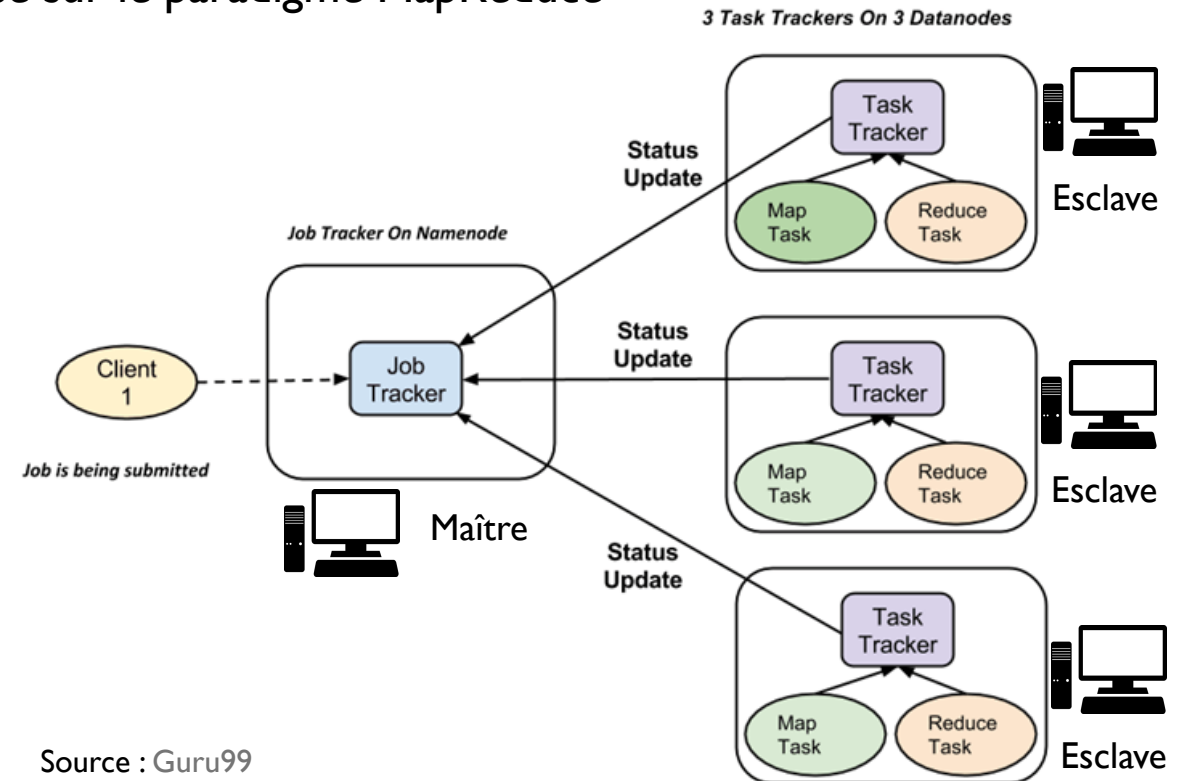


➤ Environnement de calculs distribués basé sur le paradigme MapReduce



HDFS : *Hadoop Distributed File System*

- Chaque fichier est découpée en bloc
- Chaque bloc est répartis sur plusieurs machines
- Chaque bloc est répliqué sur une machine différentes

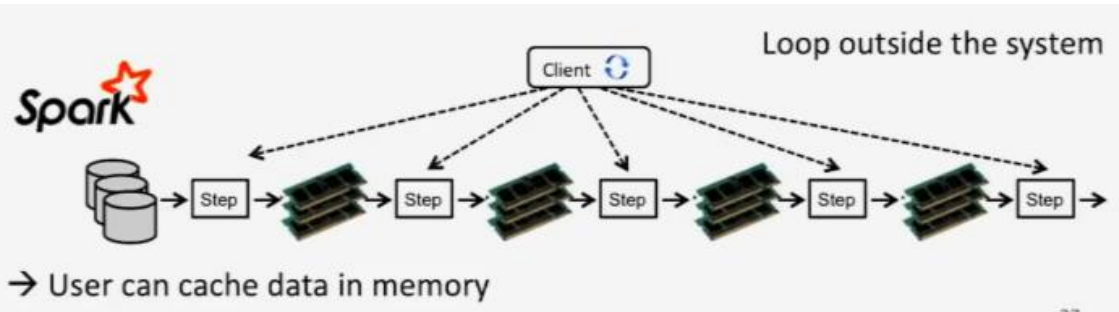


Source : Guru99

### 3) PRÉSENTATION DE L'ENVIRONNEMENT SPARK

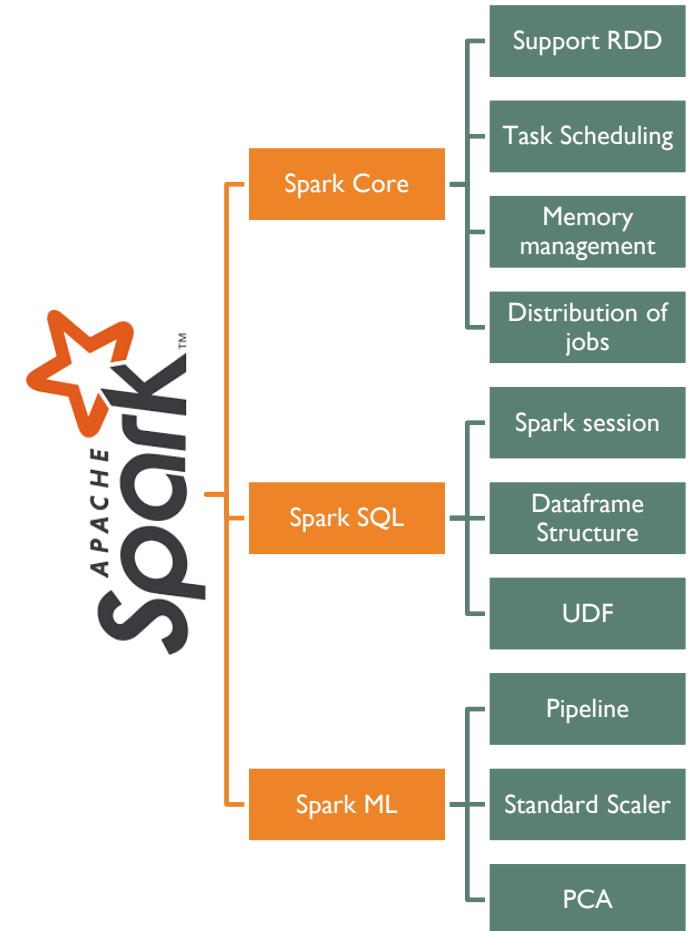
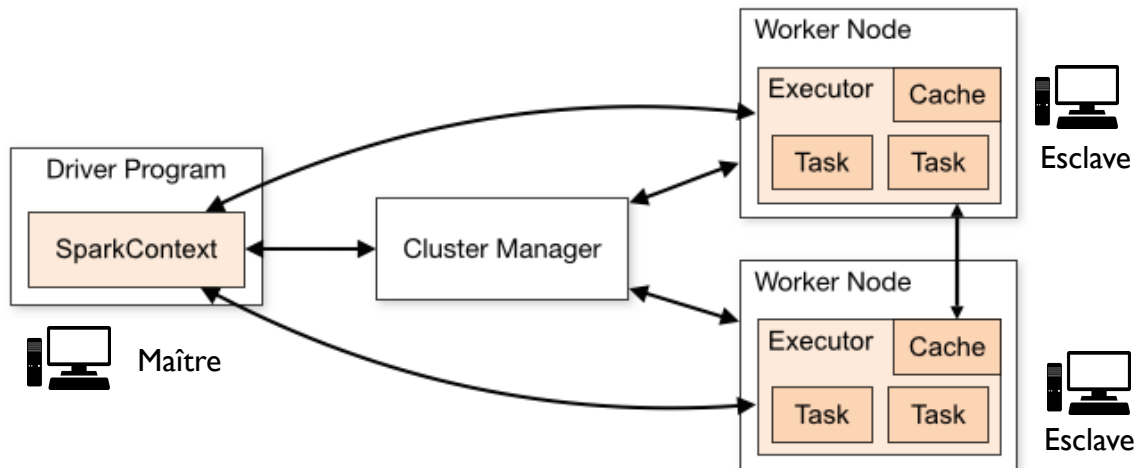


➤ Environnement de calculs distribués basé sur le socle Hadoop



RDD : *Resilient distributed dataset*

- Collection d'étapes de transformations des données réparties en plusieurs partitions
- Stockée dans la mémoire vive
- Exécutés seulement lors d'une étape d'action





## III) EXÉCUTION DU CODE : LOCAL ET CLOUD

- 1) RÉSUMÉ DE L'EXÉCUTION LOCALE ET DANS LE CLOUD
  - 2) PRÉSENTATION DE L'ARCHITECTURE CLOUD AWS
  - 3) DÉTAILS DE CRÉATION DE L'ESPACE DE STOCKAGE CLOUD
  - 4) DÉTAILS DE CRÉATION DU CLUSTER DE CALCUL
- 

# I) PRÉPARATION DU SCRIPT D'EXTRACTION DE FEATURES EN DEUX ÉTAPES

Local :   Ubuntu  

Extraction de  
features



Standardisation



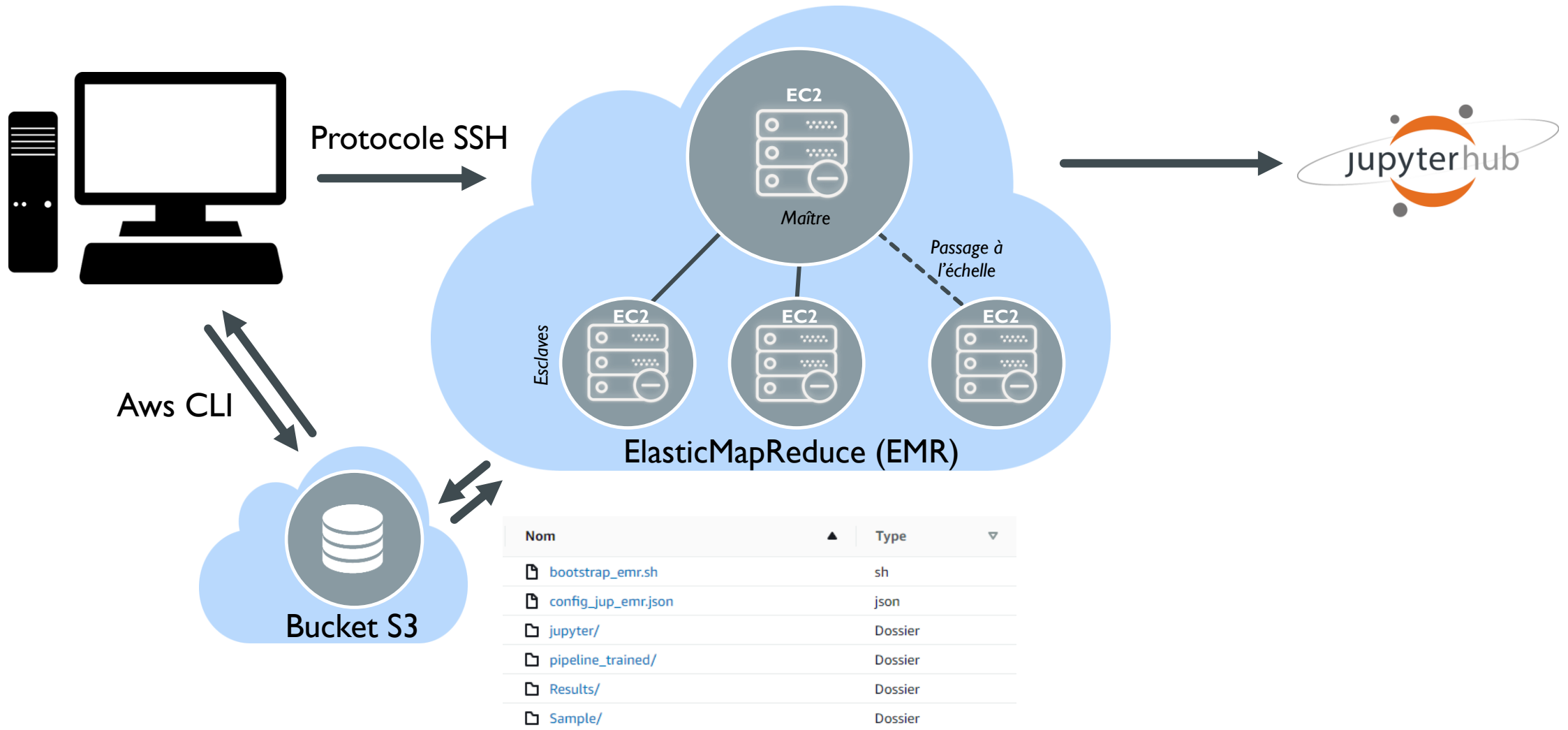
Reduction de  
dimension  
(ACP)

- Vérification du script de l'alternant :
  - Extraction de features sur l'ensemble des images d'entraînements
- Ajout d'une pipeline de réduction de dimension
  - Standardisation + PCA
- Output :
  - Pipeline entraîné sur les données

Cloud : Amazon Web Services 

- Création d'un espace de stockage
- Création d'une instance EMR
- Lancement du notebook Feature Extraction sur l'instance EMR :
  - Input : Echantillon d'images
    - Echantillon d'images tests
    - Pipeline pré-entraîné
  - Output :
    - Fichier csv

## 2) PRÉSENTATION DE L'ARCHITECTURE CLOUD AWS



### 3) DÉTAILS DE CRÉATION DE L'ESPACE DE STOCKAGE CLOUD

Stockage

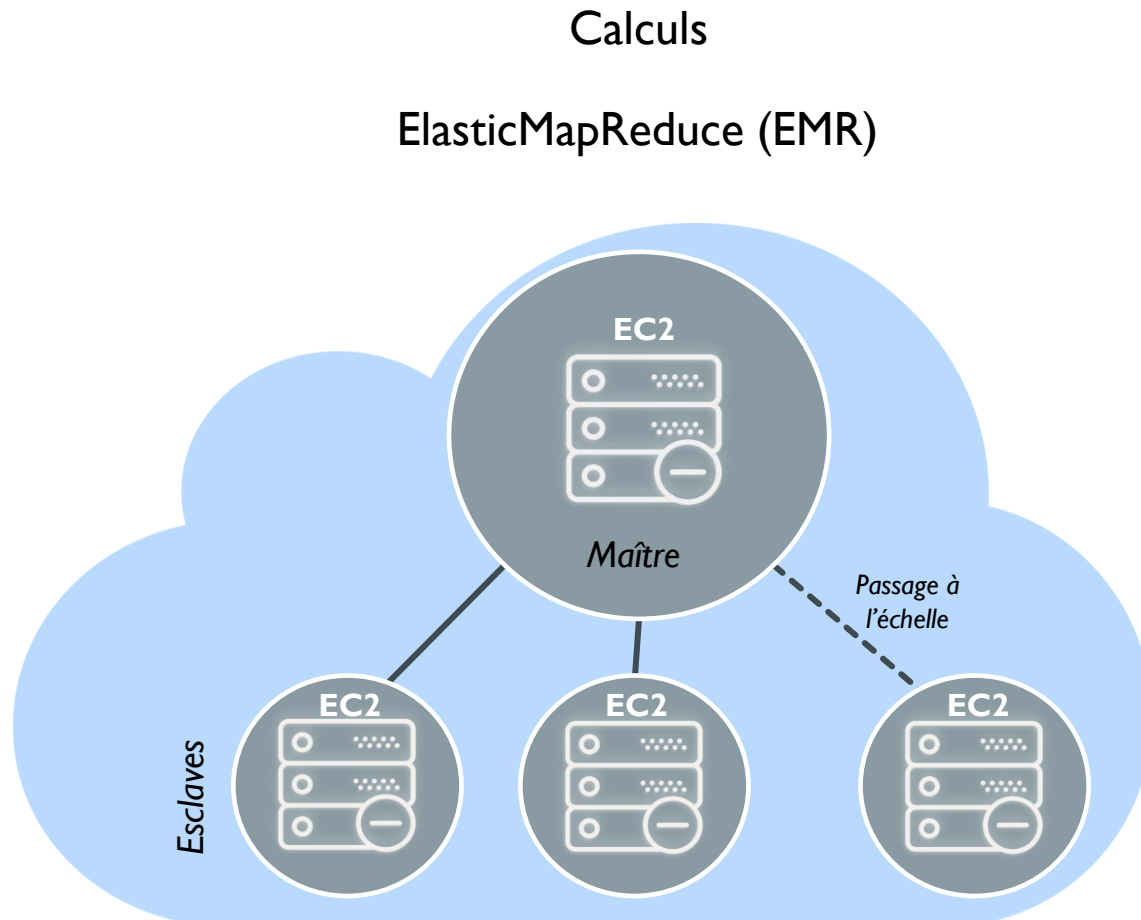
Bucket S3



- Création du Bucket S3 via la console AWS (interface web)
- Création d'un utilisateur IAM (administrateur)
- Configuration de l'Interface en Lignes de Commande (CLI) d'AWS
- Chargement des fichiers par CLI sur le bucket S3

Nom	Type
<a href="#">bootstrap_emr.sh</a>	sh
<a href="#">config_jup_emr.json</a>	json
<a href="#">jupyter/</a>	Dossier
<a href="#">pipeline_trained/</a>	Dossier
<a href="#">Results/</a>	Dossier
<a href="#">Sample/</a>	Dossier

## 4) DÉTAILS DE CRÉATION DU CLUSTER DE CALCUL



- Configuration logiciels :
  - Emr-6.9.0 + Hadoop 3.3.3 + Spark 3.3.0 + JupyterHub 1.4.1
  - Configuration JupyterHub (config\_jup\_emr.json)
- Matériel :
  - Maître : m5x.large (4Vcore 10Go RAM) x1
  - Principal (workers) : m5x.large x1
- Paramètres des clusters :
  - Journalisation
  - Actions d'amorçages > bootstrap\_emr.sh (installations de bibliothèques python)
- Sécurité :
  - Paire de clés EC2 (connexion SSH)





# PRÉSENTATION DU NOTEBOOK JUPYTER SUR L'INSTANCE EMR

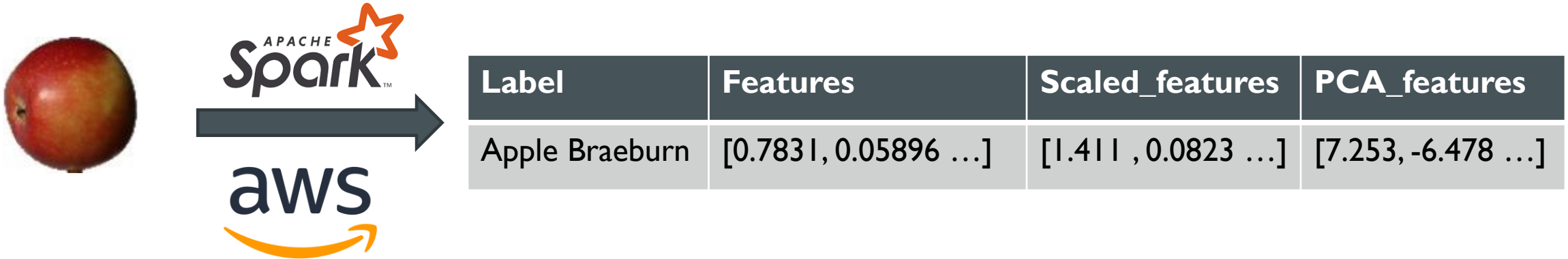
Lien du notebook



# SYNTHÈSE ET CONCLUSION



# SYNTHÈSE ET CONCLUSION



- Utilisation de Transfert Learning pour extraire des features d'images + Réduction de dimension
- Contexte de calculs distribués
- Déploiement sur une instance EMR d'AWS facilitant le passage à l'échelle.
- Prochaines étapes :
  - Entraînement d'un classifieur (+1 Couche Fully-Connected ou apprentissage supervisé)
  - Déploiement du modèle entraîné

