Projet 8

Déployez un modèle dans le cloud

22 Avril 2022

MBIADOU SALEU G. Raïssa

PLAN

01

Contexte, mission et jeu de données

Contexte

- « Fruits ! » : start-up de l'AgriTech
- Proposer des solutions innovantes pour la récolte des fruits (robot cueilleur intelligent)
- Mettre à disposition du grand public une application mobile permettant d'afficher les informations sur un fruit en le prenant en photo

MISSION

🎯Développer une **chaîne de traitement d'images** incluant **preprocessing** et **réduction de dimension** dans un environnement **Big Data**

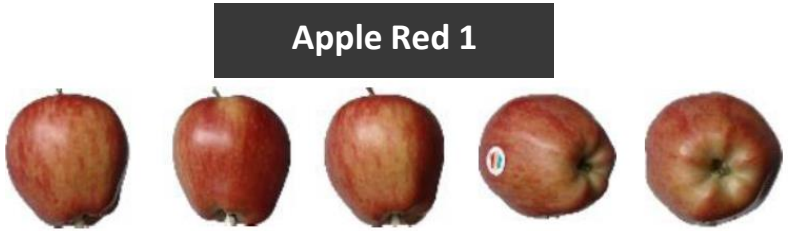
OBJECTIF: Anticiper le passage à l'échelle dans un contexte d'adoption massive

LES DONNÉES

Origine: Kaggle
- Images de 131 variétés de fruits et légumes labélisés (Fruits 360)
- Plusieurs variétés du même fruit (exemple : pomme « red » et « golden »)

Caractéristiques :
- Images 100x100 JPEG RGB
- Photos sur fond blanc centrée sur le fruit
- Photos sous tous les angles (rotation tri-axiales)

- Total : **90 483** images
- Jeu d'entraînement : **67 692** images
- Jeu de Test : **22 688** images
- Jeu multi fruits non labellisé : **103** images



02

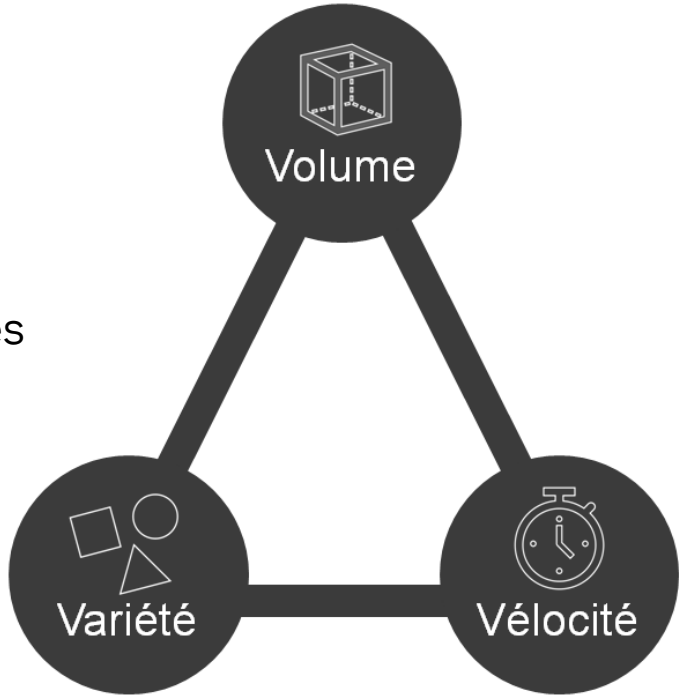
Le Big Data

Le Big Data c'est quoi ?

En français : données massives

Les 3V du Big Data:

- ❑ Volume : énormes quantités de données
- ❑ Variété : différents types de données
- ❑ Vélocité : vitesse de circulation des données (latence à minimiser)



Solutions de stockage big data

Google
Cloud Storage

Microsoft Azure
Blob Storage

Amazon Web Services
S3

Apache
Hadoop

Solution : une Infrastructure distribuée

LE STOCKAGE DISTRIBUÉ

Volume : passage à l'échelle possible

Variété : capacité d'évolution

Vélocité : partitionnement

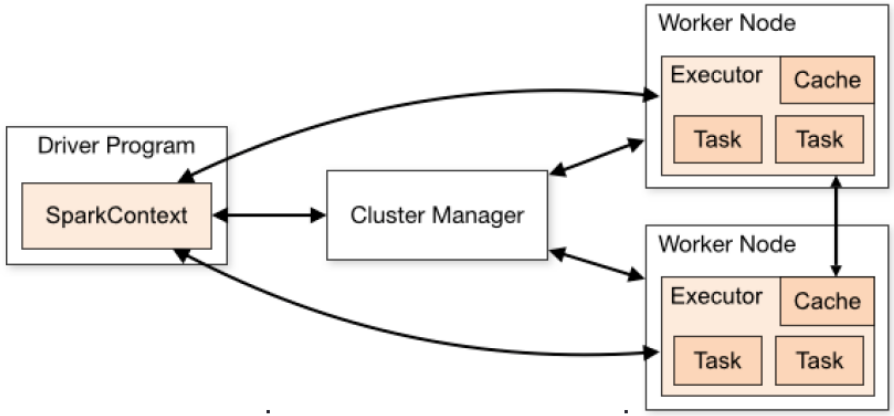
Résilience :

- redondance
- tolérance aux pannes

SOLUTION : UNE INFRASTRUCTURE DISTRIBUÉE

CALCULS DISTRIBUÉS

- ❑ Diviser les opérations en micro opérations distribuables entre différentes machines, réalisables en parallèle
- ❑ Agréger les résultats sur une même machine



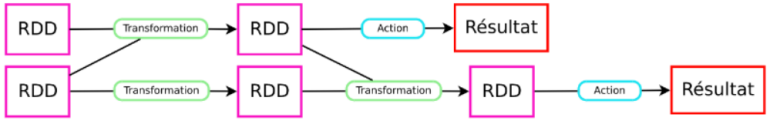
Application maître :
Configuration / Initialisation
Aggrégation des calculs

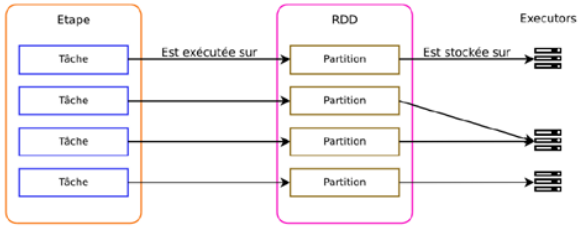
Cluster Manager :
Gestion des ressources
Distribution des calculs entre les workers

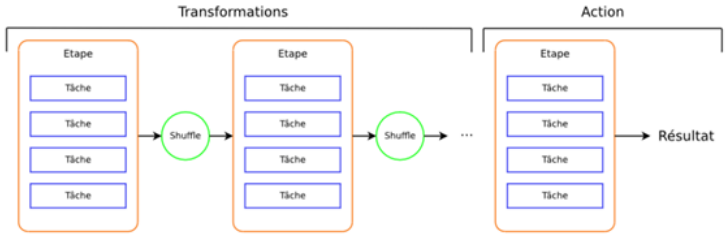
Workers :
Exécution des tâches en parallèle

CLUSTER DE CALCUL (FONCTIONNEMENT)

- RDD (Resilient Distributed Datasets) : principale innovation de Spark.
- Permettent d'effectuer des calculs parallèles en mémoire sur un cluster de façon complètement tolérante aux pannes
- Job Spark = ensemble d'étapes et étape = ensemble de tâches
- Chaque tâche s'exécute sur une partition différente des données et ces partitions sont crées par les RDD







Shuffle = redistribution des données entre les noeuds

03

Architecture retenue et chaîne de traitement

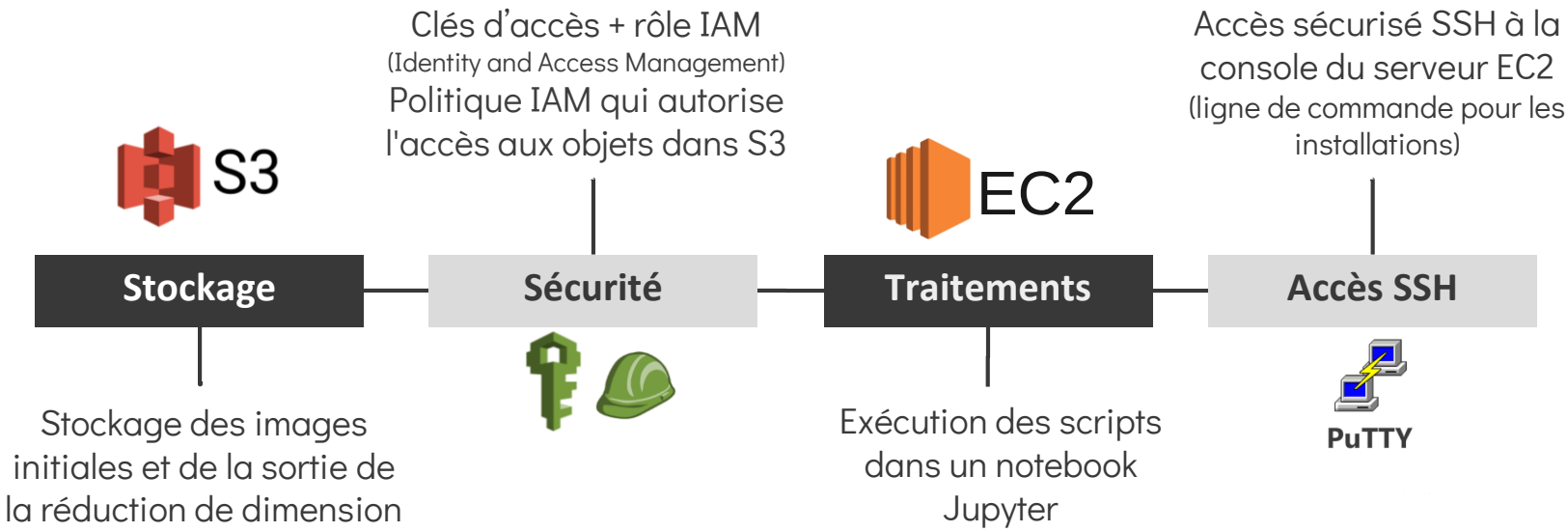
Rappel mission

Objectif : mettre en place les premières briques de traitement des images qui serviront lorsqu'il faudra passer à l'échelle en termes de volume de données

Preprocessing

Réduction de dimension

ARCHITECTURE BIG DATA



Technologies utilisées

SPARK (PYSPARK)

Framework open source de calcul distribué (pour la parallélisation des calculs – Pyspark = API python)

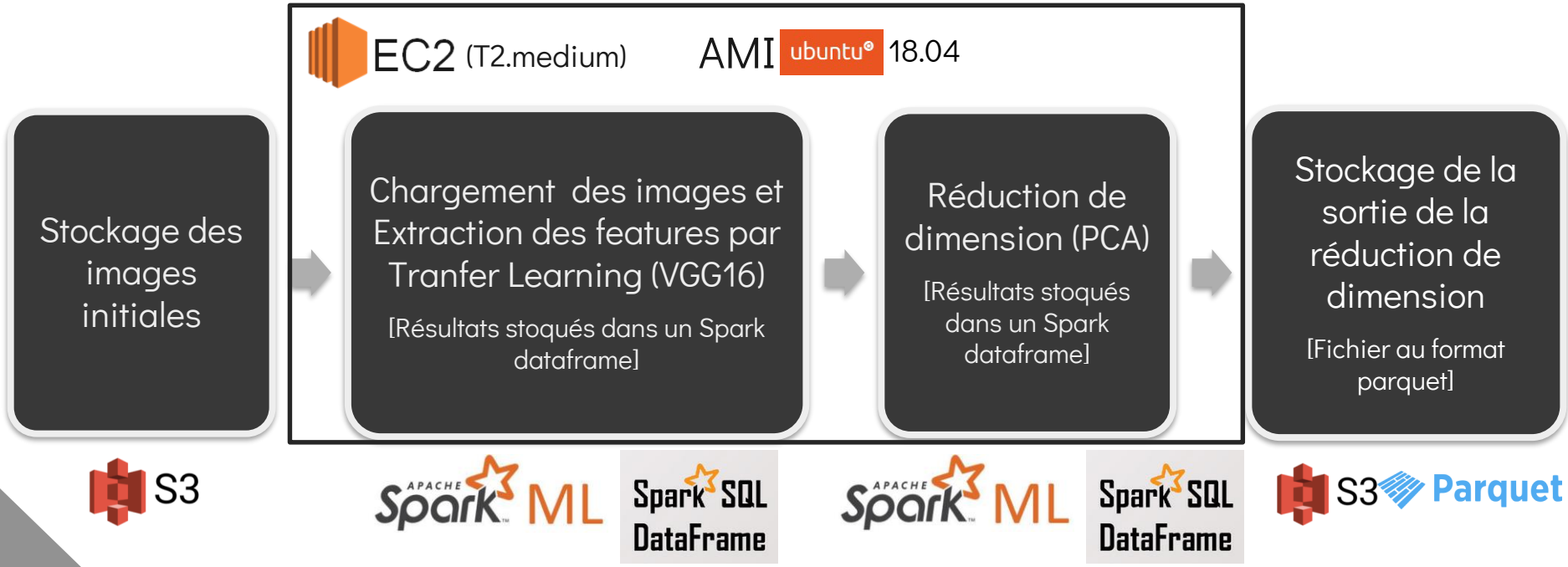
BOTO3

SDK pour accéder au bucket S3 afin d'éffectuer des opérations de lecture et écriture de fichiers

PARQUET

Format de fichier pour une exploitation optimisée en mode distribué conçue pour les données massives

CHAÎNE DE TRAITEMENT



VGG16, PCA (RAPPELS)

Transfer Learning (VGG16)

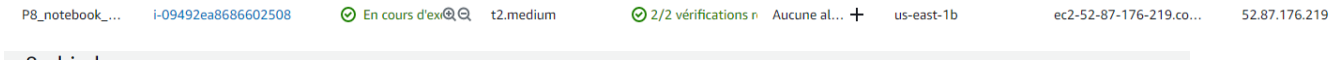
Modèle de Réseaux Neuronal Convolutif pré-entraîné sur plus d'un million d'images de 1000 catégories différentes provenant de la base de données ImageNet.
Il comprend 16 couches profondes.

PCA

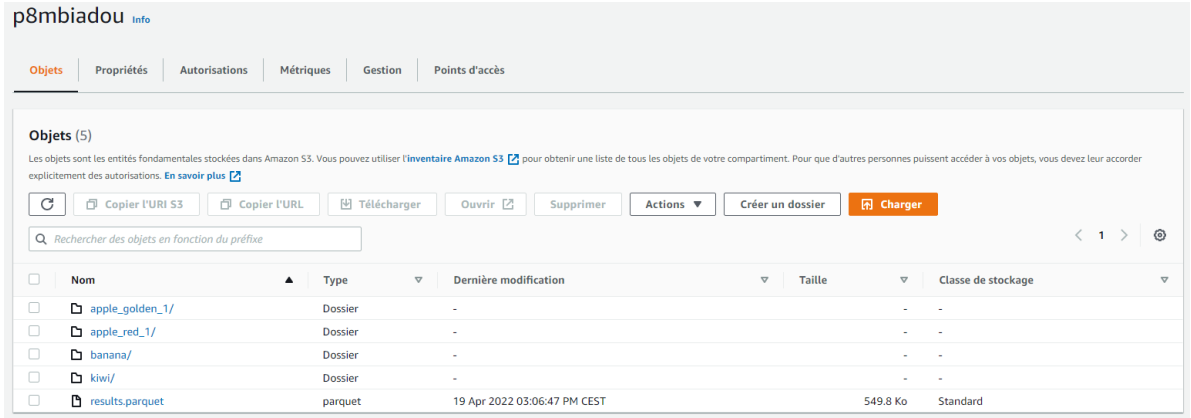
Méthode largement utilisée en réduction de dimension qui cherche à représenter les données dans un sous-espace de plus petite dimension de sorte à **conserver au maximum la variance** du nuage de données.

QUELQUES CAPTURES

Instance EC2



Bucket S3



Jupyter notebook sur EC2

RÉCAPITULATIF

Stockage fichiers sur S3 :

- Upload via AWS CLI ou Interface Web
- Lecture des fichiers depuis Spark
- Enregistrement de fichiers depuis Spark vers S3

Instance EC2: T2.medium (8GO RAM, 30GO SSD) /
OS Ubuntu Server 18.04

Configuration : Python 3.9.7 / Java 8 / Spark 3.2.1 / Pillow

Configuration sur machine distante : accès via SSH

- Chargement clés IAM / AWS
- Installation des logiciels et packages
- Mise en place d'un Notebook Jupyter accessible à distance contenant les scripts en Pyspark exécutables

DÉMO

Accéder à AWS

Notebook **Jupyter** (ps: lancer PuTTY)

04

Conclusion

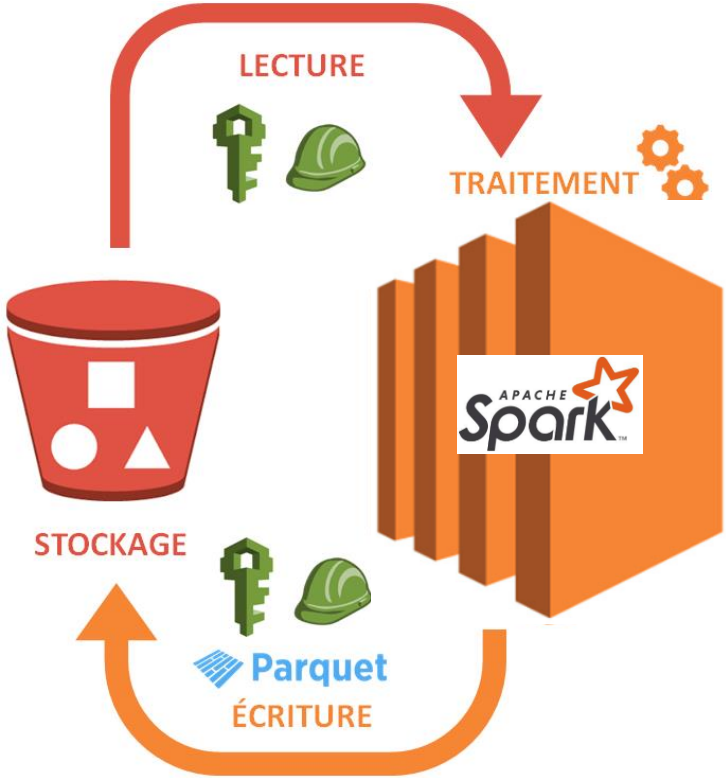
Conclusion

Notions apprises

- Prise en main Pyspark
- Découverte de l'écosystème AWS
- Administration d'un serveur Linux par SSH

Difficultés rencontrées

- Nombreuses possibilités techniques : choix complexes
- Débug complexe dû à des erreurs peu explicites (SSL: WRONG_VERSION_NUMBER lors de la mise en place du Notebook Jupyter)



MERCI

Des questions ?