Soutenance Projet 8

Fruits!

Déploiez un modèle dans le cloud

Michel Blazevic
09/2022

OPENCLASSROOMS - Formation Data Scientist

Ordre du jour

1. Introduction
    1. Problématique
    2. Présentation base de données
    3. Passer à l'échelle Big Data
2. Architecture Cloud choisie
    1. Stockage des données AWS: S3
    2. Service PaaS AWS: EMR
    3. Contrôle d'accès AWS: IAM
3. Travail effectué
    1. Calcul distribué: PySpark
    2. Pipeline transformation
    3. Résultat
4. Conclusion & Aller plus loin

1. Introduction

1. Problématique
2. Présentation base de données
3. Passer à l’échelle Big Data

Problématique

Problématique:

- **Fruits!**: Start-up de l'AgriTech voulant proposer solution innovantes pour la récoltes des fruits/légumes
- 1er avancée: créer une application mobile permettant au grand public d'avoir un moteur de classification d'images de fruits/légumes

  ⇒ Construire une première version de l'architecture Big Data

  Mission:

⇒ Développer première chaîne de traitement dans un environnement Big Data

⇒ Sélection services pour utilisation du cloud

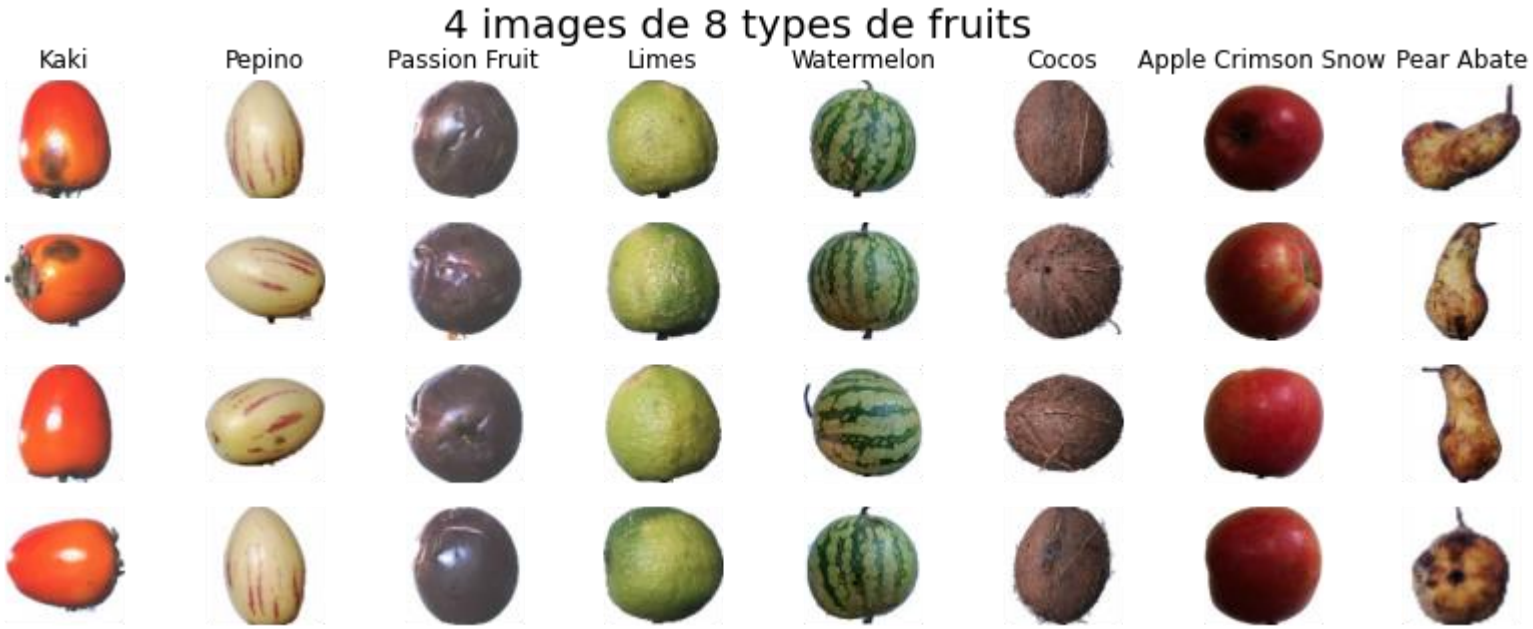
⇒ Utilisation PySpark pour calcul distribué et effectuer réduction de dimension sur un échantillon d'images

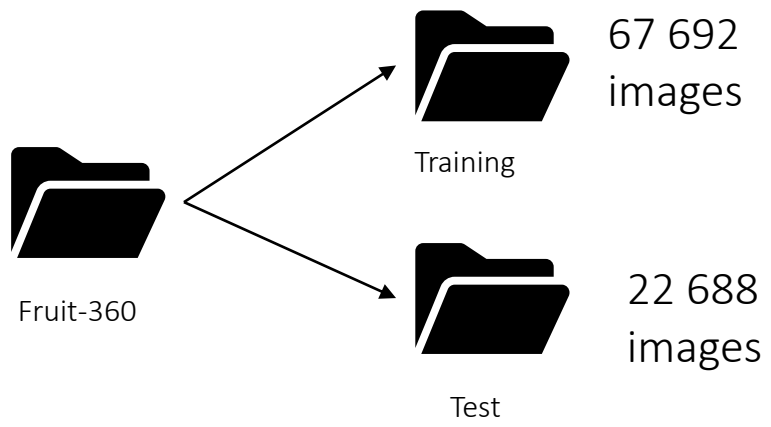
Présentation bases de données

https://www.kaggle.com/datasets/moltean/fruits

Fruits 360

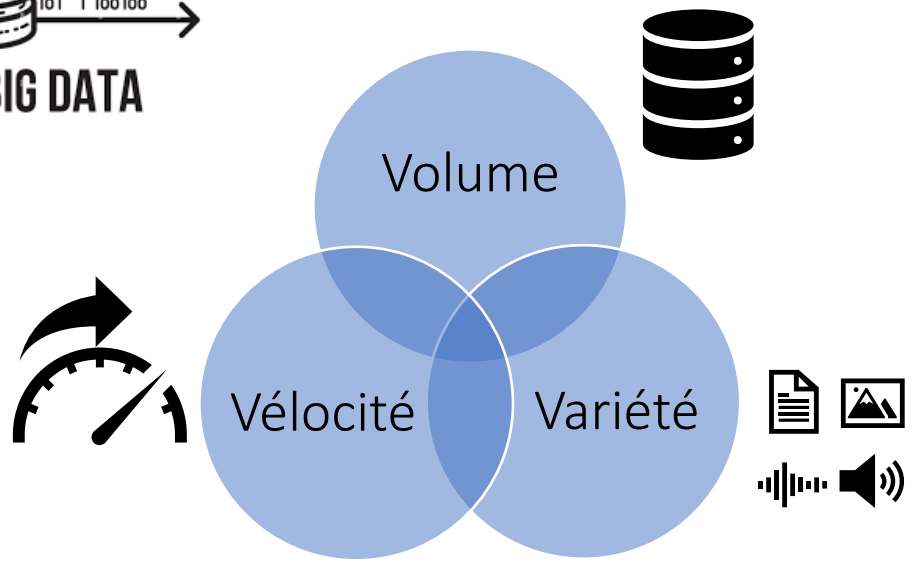
- 90483 images
- Chaque image 100x100 pixels
- 131 labels différents (fruits et légumes)
- Photos prises avec rotation 3 axes sur 360°
- Image des fruits extrait de l'arrière plan (luminosité..)





Passer à l'échelle Big Data

Les 3V du big data

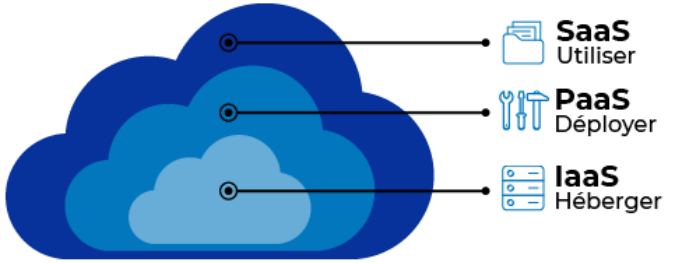


- Volume: le stockage est amené à être repenser lorsqu'il y a une augmentation des quantités
- Vélocité: L'augmentation de la quantité de données demande une rapidité de calcul accrue
- Variété: les données peuvent être sous différents formats et structuré ou non structurées (images, textes, csv, json…)

Le cloud

⇒ Environnement virtuel composé d'un ensemble de matériel et service accessible partout pouvant répondre à une problématique Big Data

⇒ Plusieurs types de cloud:



Software as a Service

Plateform as a Service

Infrastructure as a Service

⇒ Différents fournisseur d'accès au cloud:

Azure

Choix AWS

aws

Google Cloud

2. Architecture Cloud choisie

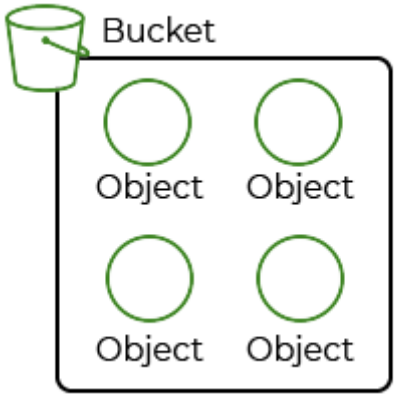
1. Stockage des données AWS: S3
2. Service PaaS AWS: EMR
3. Contrôle d’accès AWS: IAM

Stockage des données AWS: S3

→ S3: Simple Storage Service -> Stockage/hébergeur des fichiers dans le cloud

→ Avantages:

- Configuration des droits d'accès
- Versioning et configuration d'une date d'expiration
- Chiffrer et réplications des fichiers
- Pas de limites de place et moins cher que stockage sur serveur
- Différentes types d'archivage à des prix différents
- Accès avec API (boto3)



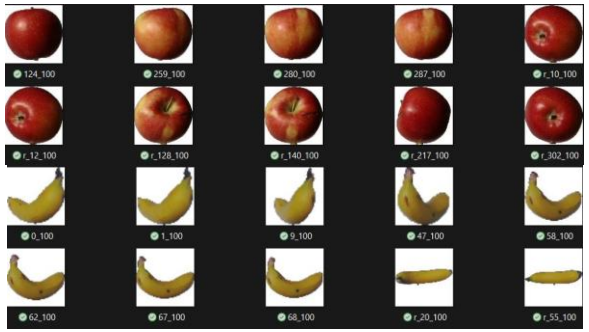
Amazon S3

P8-bucket-mb

fruits360/

import-lib/

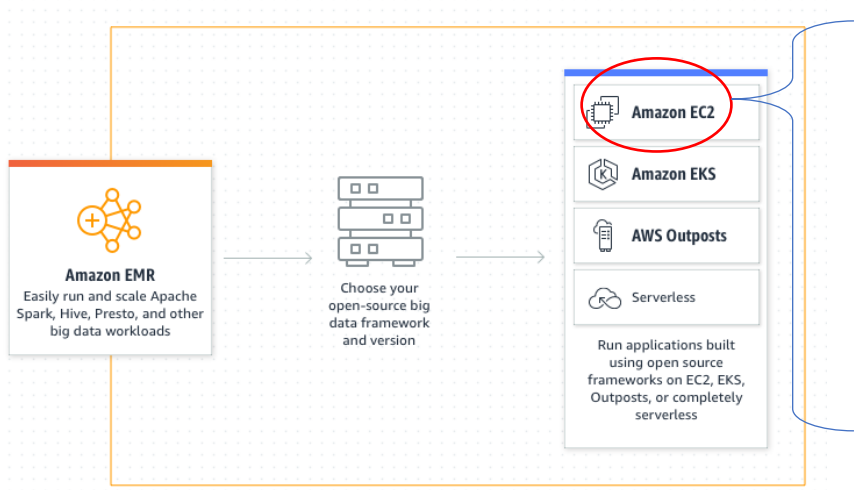
Result_pca_p8/



Service PaaS AWS: EMR

⇒ Amazon Elastic MapReduce

- Plateform de cluster gérée et simplifiant les framework de Big Data
- Permet de traiter et analyser grand volume de données



Ec2: Amazon Elastic Compute Cloud

- Service virtuel scalable d'accès à des serveurs
- Configuration processeur, mémoire et stockage

Amazon EC2

Choix:

MASTER Master - 1	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB
CORE Core - 2	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB

Détail de configuration

- EMR-6.7.0
- Hadoop Amazon 3.2.1
- JupyterEnterpriseGateway 2.1.0,
- TensorFlow 2.4.1
- Livy 0.7.1
- Spark 3.2.1

Avec Bootstrapping:

- Pandas
- Numpy
- Pillow
- matplotlib

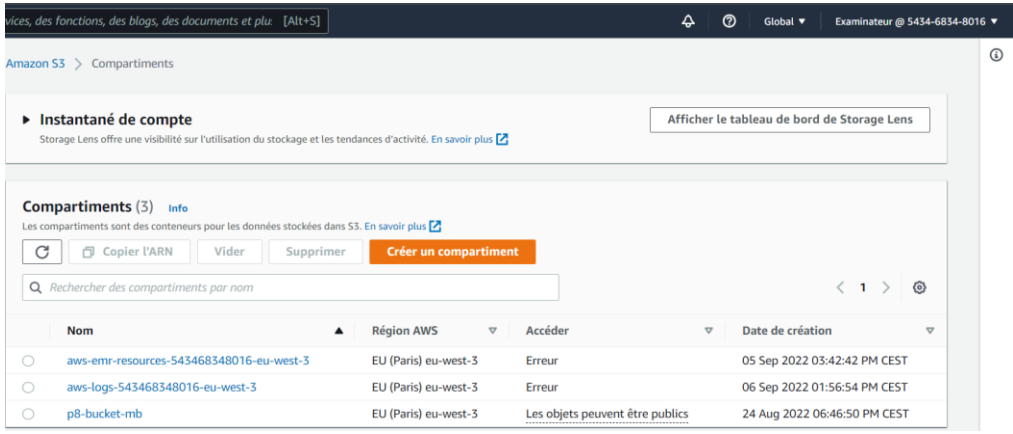
Contrôle d’accès AWS: IAM

→ Clefs pour répartir les droits d’accès selon les utilisateurs, les rôles et la politique des services

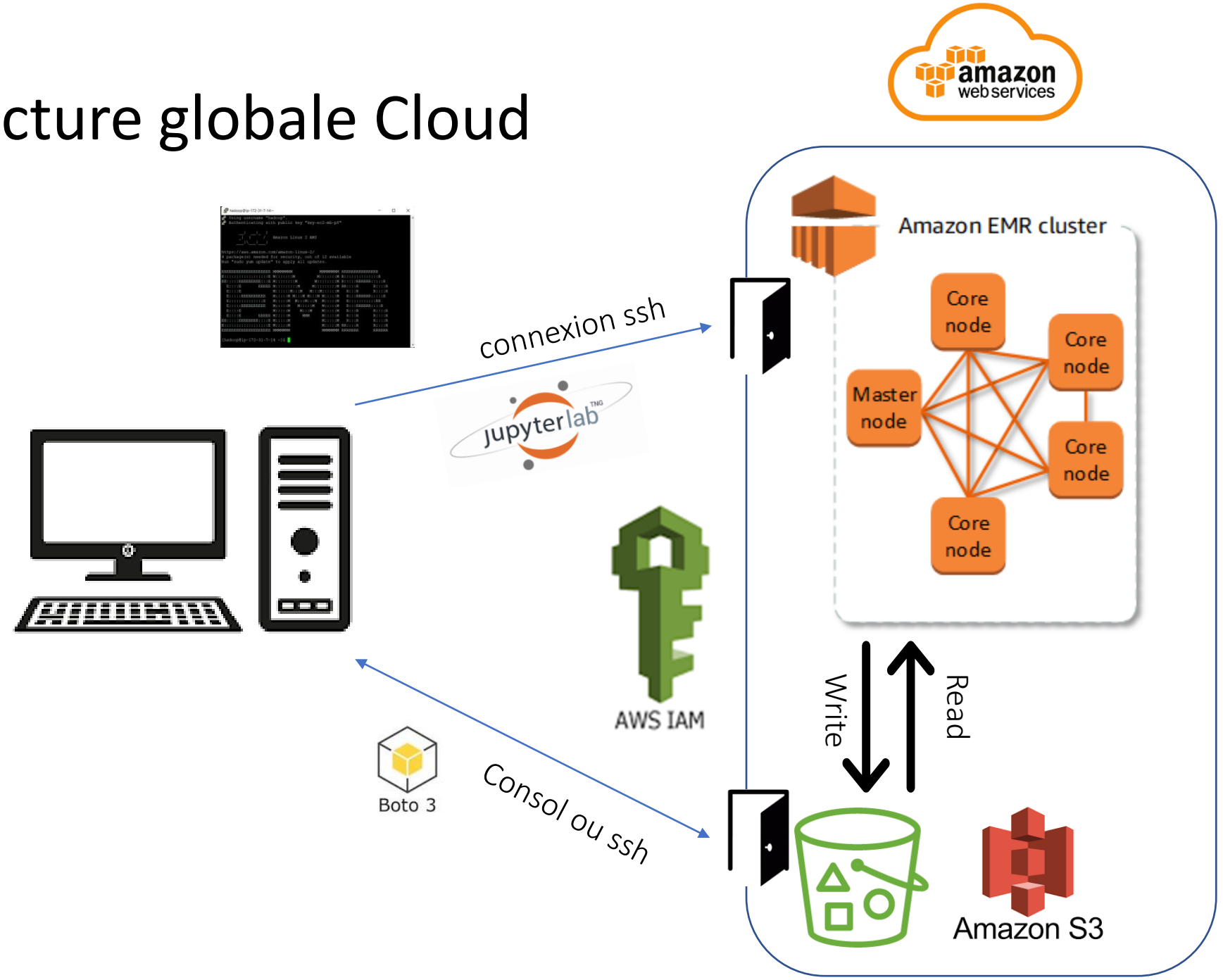
AWS IAM

Accès console:

url = https://michblaz-aws-94.signin.aws.amazon.com/console
Login= Examinateur
Mdp= Examinateur_iam_092022



Architecture globale Cloud



3. Travail effectué

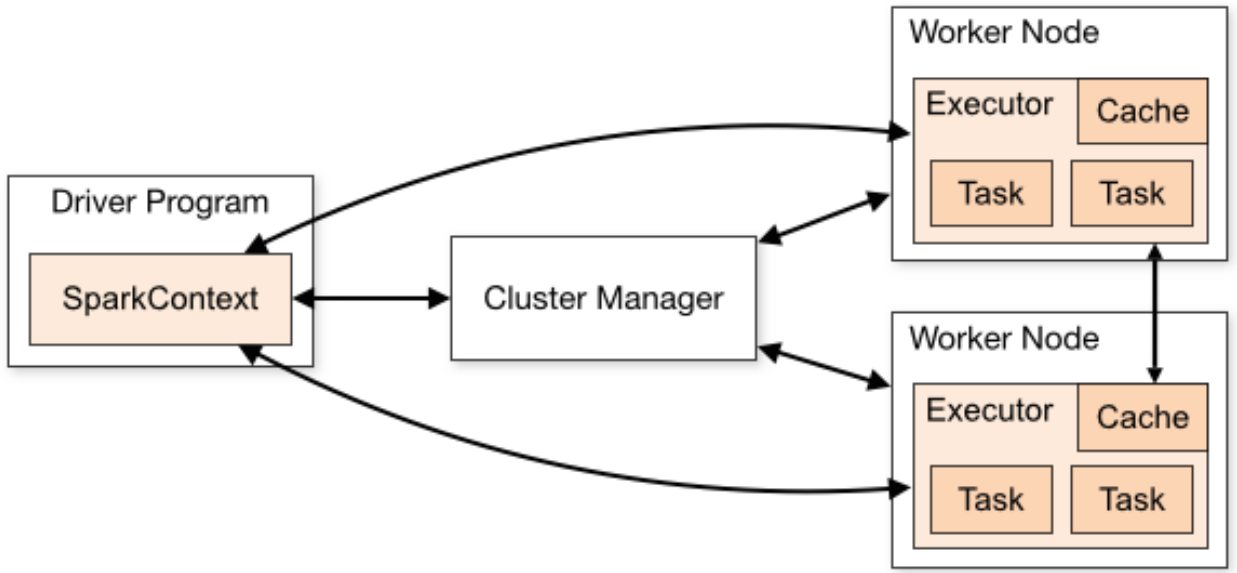
1. Calcul distribué: PySpark
2. Pipeline transformation
3. Résultat

Calcul distribué: PySpark

⇒ Partitionner les données sur plusieurs serveurs pour effectuer des calculs distribués

⇒ Cluster d'une application Spark:

- Driver: gère et organise les étapes du cluster
- Manager: alloue/distribue les ressources
- Worker: les esclaves qui stock les données et exécute les tâches/calculs

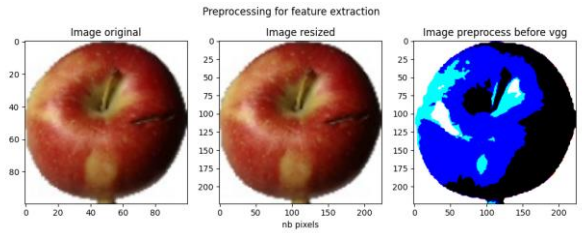


Pipeline transformation

Rappel objectif: créer première chaîne de traitement d'image

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1662488060805_0001	pyspark	idle	Link	Link	None	✓



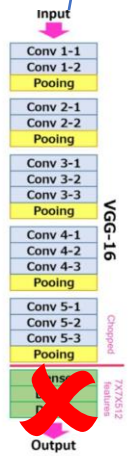
fruits360/	Folder
import-lib/	Folder
Result_pca_p8/	Folder

Mise en place environnement

- Création env Spark
  - SparkContext
  - SparkSession
- Import données S3
  - Binarytype
  - RDD

Features extraction

- Preprocessing
  - Resize
  - Preprocess_input
- VGG16



Réduction dimension

- StandardScaler
- PCA

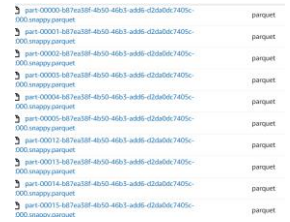
Write S3

Format parquet

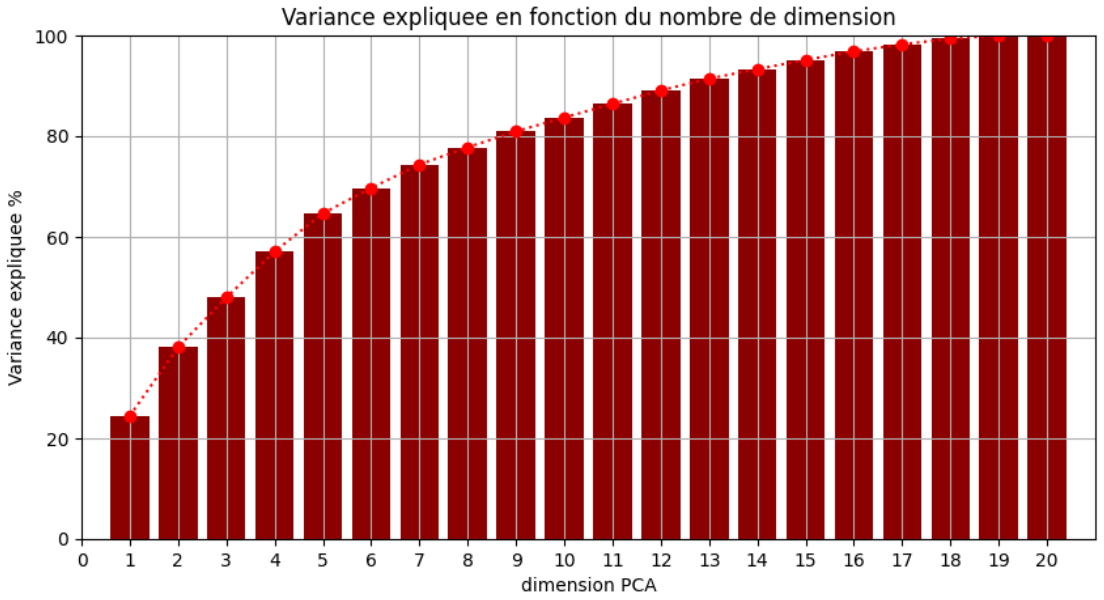
```
+--------------------+--------------------+-----+
|                path|             content|label|
+--------------------+--------------------+-----+
|s3a://p8-bucket-m...|[FF D8 FF E0 00 1...|pomme|
|s3a://p8-bucket-m...|[FF D8 FF E0 00 1...|pomme|
|s3a://p8-bucket-m...|[FF D8 FF E0 00 1...|pomme|
+--------------------+--------------------+-----+
```

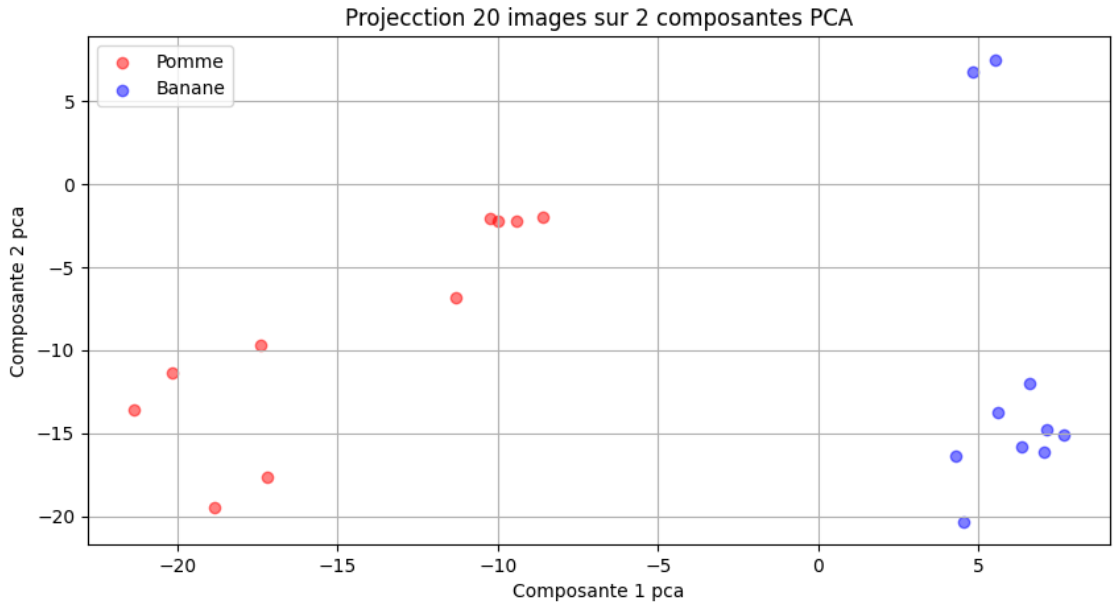
```
+--------------------+------+--------------------+--------------------+
|                path| label|  features_sparktypes|      scaledFeatures|
+--------------------+------+--------------------+--------------------+
|s3a://p8-bucket-m...| pomme|[60.6305160522460...|[2.78719159822297...|
|s3a://p8-bucket-m...|banane|[4.01261711120605...|[0.18446045700158...|
|s3a://p8-bucket-m...| pomme|[8.80388259887695...|[0.40471546688117...|
+--------------------+------+--------------------+--------------------+
```

```
+--------------------+------+--------------------+
|                path| label|        pca_features|
+--------------------+------+--------------------+
|s3a://p8-bucket-m...| pomme|[-18.827460106368...|
|s3a://p8-bucket-m...|banane|[5.60933227696263...|
|s3a://p8-bucket-m...| pomme|[-10.233328862411...|
+--------------------+------+--------------------+
```



Résultat


Variance expliquee en fonction du nombre de dimension
Variance expliquee %
dimension PCA
0
20
40
60
80
100
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Projecction 20 images sur 2 composantes PCA
Pomme
Banane
Composante 2 pca
Composante 1 pca
5
0
−5
−10
−15
−20
−20 −15 −10 −5 0 5

Conclusion et aller plus loin

Conclusion:

- Découverte Big Data et Environnement Cloud
- Utilisation services AWS pour stockage (s3) et traitement de données (EMR)
- Faire des calculs distribués avec PySpark Apache
- Première chaîne de traitement pour **`Fruits!`**
  (Features extraction, Scaling, PCA)

Aller plus loin/mise à l'échelle:

- Suivre les applications Spark depuis Spark UI
- Voir les choix de configurations du cluster
  (types d'instances, nb workers nodes…)
- Améliorer extraction de features (preprocessing images, autres CNN …)

Merci!
Des questions ?

Annexe 1

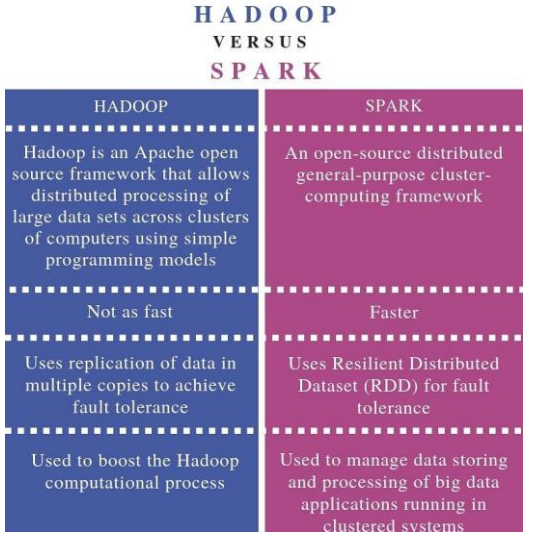
- Différence Big Data / Cloud:

  -> Le big Data est un espace virtuel composé de données massives/volumineuses ne pouvant être traitées par une machine ou par un être humain.

  -> Le Cloud est l'environnement virtuel avec un ensemble d'éléments matériels accessibles partout

  Différence: ⟶ Le cloud (outil) donne accès et stock les données et le big data est la possession et l'exploitation des données massives

Annexe 2



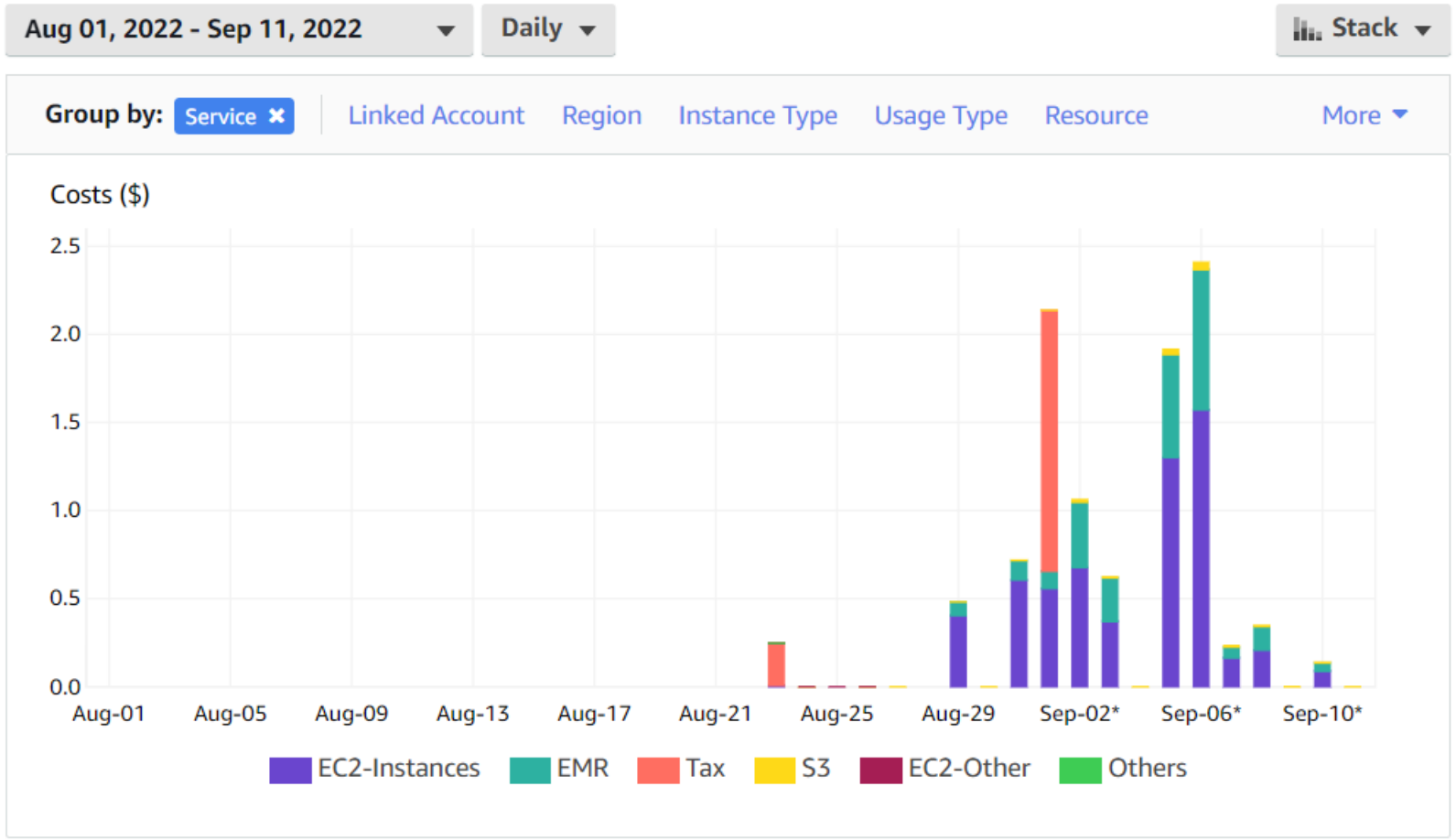
- Différence hadoop/spark
  - Deux frameworks big data
  - Hadoop infrastructure de données distribuées (plusieur nœuds d'un cluster)
  - Spark sait travailler sur les données distribuées mais ne gère pas le stockage distribué. Il s'appuie sur un système de stockage distribué
  - Hadoop utilise la composante de stockage HDFS et le traitement avec MapReduce
  - Spark peut fonctionner sans hadoop mais nécessite un autre système de gestion de fichier.
  - Spark à la différence de mapreduce va effectuer toutes les opérations d'analyses nécessaie
  - Il y a la redondance d'écriture des données sur le cluster avant de lancer létape suivante
  - Gestion des pannes: Hadoop et Spark gèrent la résilience au panne mais spark donne accès à des RDD (resilient distributed data) réparti sur le cluster de données. Qui peuvent êter récupérér complêtement après une panne ou défaillance

Annexe 3

- Type de service cloud

  - Iaas: accès à des serveurs définis: problème technique : pas notre problème mais si volonté de changer la puissance on doit demander
  - Paas: Accès a des serveurs mais on gère pour nous le nombre de machines et des fonctionnalités
  - Saas: accès à un loficiel comme un service donc seulement à une utilité définie

Annexe 4


Aug 01, 2022 - Sep 11, 2022
Daily
Stack
Group by:
Service
Linked Account
Region
Instance Type
Usage Type
Resource
More
Costs ($)
2.5
2.0
1.5
1.0
0.5
0.0
Aug-01
Aug-05
Aug-09
Aug-13
Aug-17
Aug-21
Aug-25
Aug-29
Sep-02*
Sep-06*
Sep-10*
EC2-Instances
EMR
Tax
S3
EC2-Other
Others

Annexe 5

- Récapitulons les points à vérifier pour optimiser une application :
    - Stockez en cache les données quand c'est nécessaire
    - Utilisez les bonnes structures de données
    - Ajustez le nombre de partitions