

Réalisez un traitement dans un environnement Big Data sur le Cloud

Date de la soutenance : 19/06/2024

Antoine Arragon

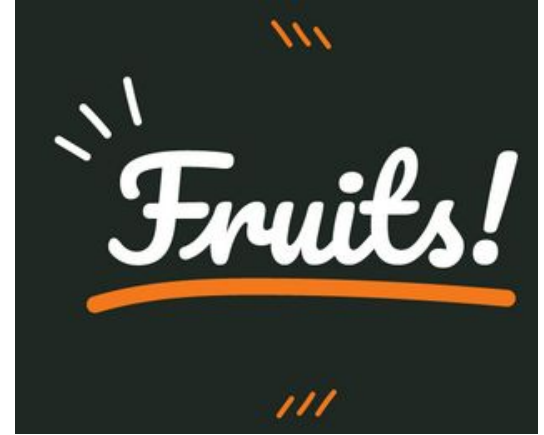


Contexte :

- La société 'Fruits!' souhaite développer des robots intelligents cueilleurs de fruits.
- Dans un 1er temps il s'agit de développer une application mobile de reconnaissance et d'obtentions d'informations à partir de photos de fruits réalisées par des usagers.

Objectifs :

- Mettre en place un environnement cloud permettant de faire face à un accroissement rapide du volume de données et de leur traitement dans un cadre sécurisé : AWS - IAM - EMR - S3 - Spark ;
- Réaliser un script PySpark permettant le preprocessing et l'extraction de features à partir des images ainsi qu'une réduction de dimension de ces caractéristiques.



Le jeu de données : “Fruits-360 datasets” - Version 2020.05.18.0

- Contient 90483 images de 131 types de fruits différents
- Réparties en training et test set

Utilisation dans ce travail :

- Pour des raisons de temps et de coûts techniques nous n’avons utilisé qu’un échantillon très restreint de ce jeu de données en sélectionnant aléatoirement 5 types de fruits et 10 images pour chaque type.
- Ex d’images :



Plan :



1. Choix et présentation de la solution cloud
 - 1.1. Problématiques Big Data
 - 1.2. Amazon Web Services : IAM - EC2 - EMR - S3

2. Réalisation de la chaîne de traitement dans le cloud
 - 2.1. Preprocessing - Extraction de features - ACP
 - 2.2. Enregistrement des résultats dans S3
 - 2.3. Spark jobs

Conclusion



1. Choix et présentation de la solution cloud

1.1. Problématiques Big Data

Le stockage de gros volumes de données :

- Un besoin croissant et flexible de capacités supplémentaires ;
- Avec un délai d'accès raisonnable ;
- Questions de sécurité, quels accès ?

Le traitement de ces données :

- Limites à l'acquisition de nouvelles capacités ;
- Choix d'accroissement de la puissance : verticale ou horizontale ?



1. Choix et présentation de la solution cloud

1.2. Amazon Web Services : IAM - EC2 - S3 - EMR

Plateforme de services cloud fournie par Amazon et lancée en 2006 (principaux concurrents : Microsoft Azure et Google Cloud Platform (GCP)) ;

Multitudes de services à disposition - notre utilisation :

- EC2 : Elastic Compute Cloud ➤ instances de calcul à la demande ;
- IAM : Identity and Access Management ➤ Politiques de sécurité et accès ;
- S3 : Simple Storage Service ➤ stockage quasiment infini ;
- EMR : Elastic MapReduce ➤ service de traitement de données avec applications pré-installées et configurées.

1. Choix et présentation de la solution cloud

1.2. Amazon Web Services : IAM - EC2 - S3 - EMR

1e étape : Création d'un utilisateur Admin1 avec accès administrateur.

2e étape : Création d'un bucket S3, sur le serveur Ouest Européen (Paris), pour des raisons RGPD :






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

oc-p9-bigdata-data [Info](#)

Nous y avons transféré les données utilisées pour nos tests :

Nous y avons également uploadé le notebook réalisé en local, que

nous avons adapté à l'environnement cloud.

<input type="checkbox"/>	 Apple Granny Smith/	Dossier
<input type="checkbox"/>	 Apricot/	Dossier
<input type="checkbox"/>	 Banana/	Dossier
<input type="checkbox"/>	 Blueberry/	Dossier
<input type="checkbox"/>	 Cocos/	Dossier

1. Choix et présentation de la solution cloud

1.2. Amazon Web Services : IAM - EC2 - S3 - EMR

3e étape : choix d'utilisation du service EMR

Permet de configurer un cluster correspondant à nos besoins :

- Choix du nombre et de la puissance des machines ➤ 'scalabilité' horizontale et verticale ;
- Possibilité d'ajuster manuellement ou automatiquement les capacités de l'instance en fonction du volume de données à traiter :

Configuration de cluster

Groupes d'instances

Capacité

1 primaire(s)

2 unité(s) principale(s)

0 tâche(s)

<input checked="" type="checkbox"/>	Primaire
	m5.xlarge
	4 vCore, 16 Gio de mémoire, Stockage E
<input checked="" type="checkbox"/>	Principal (Unité principale)
	m5.xlarge
	4 vCore, 16 Gio de mémoire, Stockage E
	m5.xlarge
	4 vCore, 16 Gio de mémoire, Stockage E

▼ Dimensionnement et mise en service du cluster - *requis* [Info](#)

Choisissez la manière dont Amazon EMR doit dimensionner votre cluster.

Choisir une option

☒ Définir manuellement la taille du cluster

Utilisez cette option si vous connaissez vos modèles de charge de travail à l'avance.

☐ Utiliser la mise à l'échelle gérée par EMR

Surveillez les principales métriques de charges de travail afin qu'EMR puisse optimiser la taille du cluster et l'utilisation des ressources.

☐ Utiliser un autoscaling personnalisée

Pour dimensionner de manière programmatique les unités principales et les nœuds de tâches, créez des politiques d'autoscaling personnalisées.

1. Choix et présentation de la solution cloud

1.2. Amazon Web Services : IAM - EC2 - S3 - EMR

- Choix d'un environnement préconfiguré adapté à notre travail : applications Hadoop, Spark, TensorFlow et JupyterHub installées :
- Possibilité d'ajouter des packages supplémentaires, s'installant sur l'ensemble des machines à l'initialisation du cluster et lien avec le bucket S3 :

Classification	▼	Propriété	▼	Valeur
jupyter-s3-conf		s3.persistence.bucket		oc-p9-bigdata-data
jupyter-s3-conf		s3.persistence.enabled		true

Applications

Version d'Amazon EMR
emr-6.11.0

Applications installées
Hadoop 3.3.3, JupyterHub
1.4.1, Spark 3.3.2, TensorFlow
2.11.0

```
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install keras==2.11.0
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas
sudo python3 -m pip install matplotlib
sudo python3 -m pip install seaborn
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
```

1. Choix et présentation de la solution cloud

1.2. Amazon Web Services : IAM - EC2 - S3 - EMR

- Création d'une paire de clé SSH pour établir la connexion à nos instances EC2 ;
- Etablissement d'un tunnel SSH pour avoir accès au réseau local du driver :
 - Ajout de règles entrantes dans les groupes de sécurité EC2 ;
 - Suivi recommandations AWS - Config PuTTY.

Étape 1: Ouvrez un tunnel SSH vers le nœud primaire Amazon EMR.

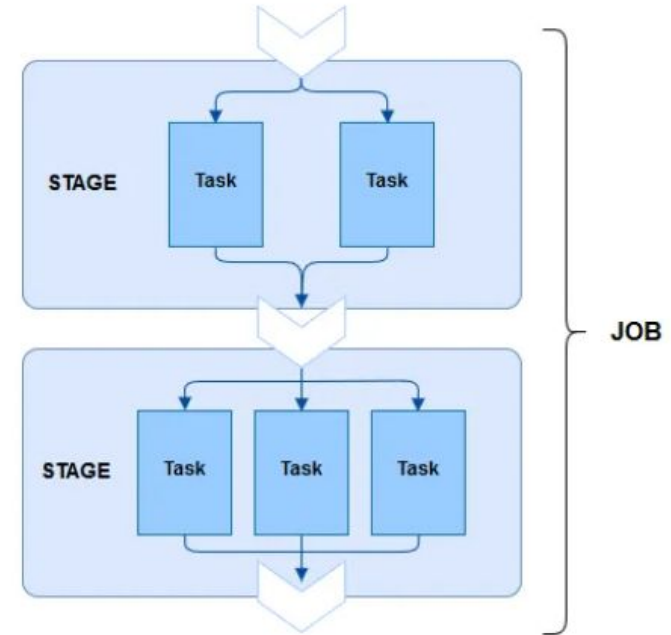
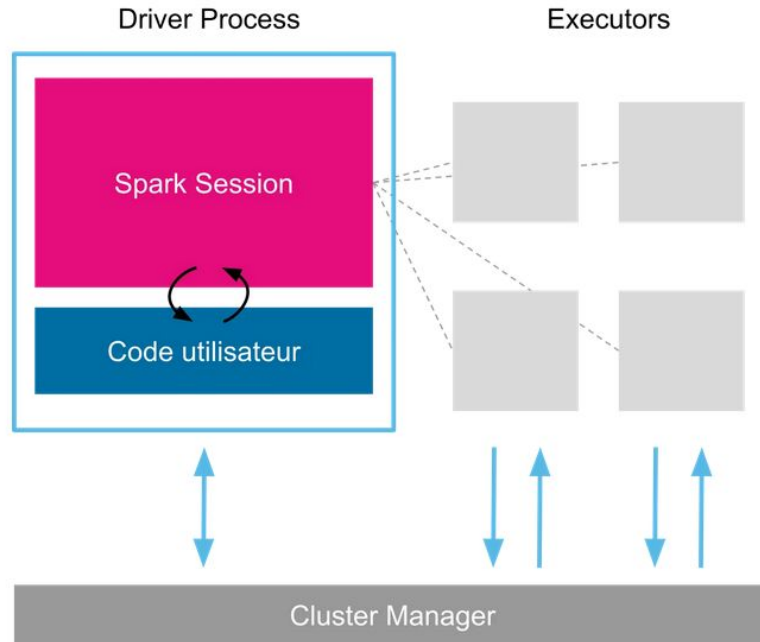
Windows | Mac/Linux

1. Téléchargez PuTTY.exe sur votre ordinateur à partir de : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
2. Démarrez PuTTY.
3. Dans la liste Category (Catégorie), cliquez sur Session.
4. Dans le champ Host Name (Nom d'hôte), entrez `hadoop@ec2-15-237-178-198.eu-west-3.compute.amazonaws.com`.

```
hadoop@ip-172-31-3-230:~  
A newer version of Amazon Linux is available!  
Amazon Linux 2023, GA and supported until 2028-03-15.  
https://aws.amazon.com/linux/amazon-linux-2023/  
10 package(s) needed for security, out of 14 available  
Run "sudo yum update" to apply all updates.  
EEEEEEEEEEEEEEEEEEEE MMMMMM MMMMMM RRRRRRRRRRRRRR  
E:EEEEEEEEEEEEEEEE M:EEEE M:EEEE R:EEEEEEEEEEEE  
EE:EEEEEEEEEEEEEEEE M:EEEE M:EEEE R:EEEEEEEEEEEE  
E:EE EEEEE M:EEEE M:EEEE RR:R R:R  
E:EE EEEEE M:EEEE M:EEEE M:EEEE R:R R:R  
E:EEEEEEEEEEEE M:EE M:EE M:EE R:RRRRRRRRR  
E:EEEEEEEEEEEE M:EE M:EE M:EE R:EEEEEEEE  
E:EE EEEEE M:EE M:EE M:EE R:RRRRRRRRR  
E:EE EEEEE M:EE M:EE M:EE R:R R:R  
EE:EEEEEEEEEEEE M:EE M:EE M:EE R:R R:R  
E:EEEEEEEEEEEE M:EE M:EE RR:R R:R  
EEEEEEEEEEEEEEEE MMMMMM MMMMMM RRRRRR RRRRRR  
[hadoop@ip-172-31-3-230 ~]$
```

2. Réalisation de la chaîne de traitement dans le cloud

Point sur Spark





2. Réalisation de la chaîne de traitement dans le cloud

2.1. Preprocessing - Extraction de Features - ACP

- 1) Chargement des images au format binaire, label et path dans un dataframe Spark
- 2) Utilisation d'une fonction qui :
 - a) fait un preprocessing des images pour les mettre au bon format d'input du modèle (100x100 ➤ 224x224 notamment et normalisation des pixels)
 - b) charge le modèle MobileNetV2 duquel on retire la dernière couche pour réaliser l'extraction de features - diffusion des poids à l'ensemble des "workers" :

```
broadcast_weights = sc.broadcast(new_model.get_weights())
```
 - c) réalise l'extraction de features de manière distribuée et renvoie les résultats dans un dataframe Spark avec un vecteur de dimension de 1280.



2. Réalisation de la chaîne de traitement dans le cloud

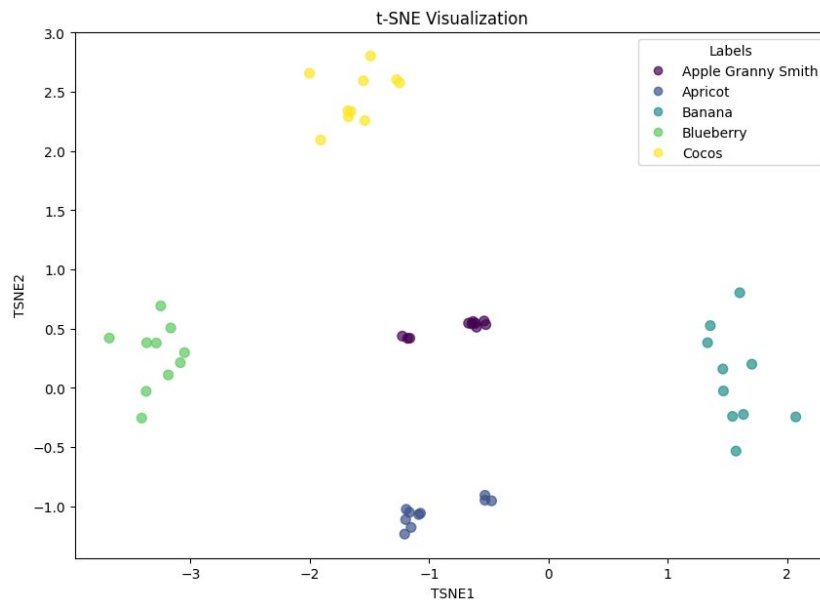
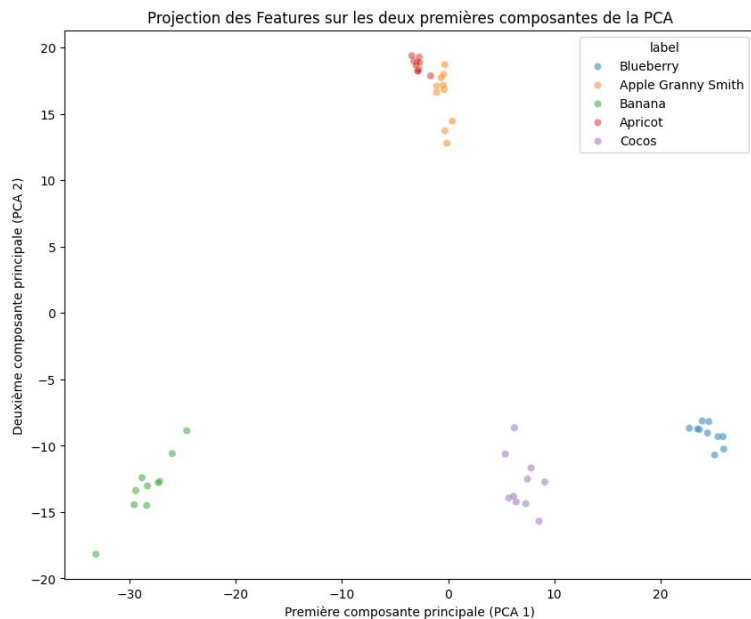
2.1. Preprocessing - Extraction de Features - ACP

- 3) la suite du travail consistait en une réduction de dimension, via une analyse en composantes principales :
- a) Transformation des résultats obtenus précédemment au format “VectorUDT” ;
 - b) Utilisation d’un pipeline permettant une standardisation des données puis une ACP ;
 - c) Choix : nombre composantes = 95% de la variance expliquée ➤ 28 composantes.



2. Réalisation de la chaîne de traitement dans le cloud

2.1. Preprocessing - Extraction de Features - ACP



2. Réalisation de la chaîne de traitement dans le cloud

2.2. Enregistrement des résultats sur le bucket S3

```
PATH_data = PATH + Images/Images  
PATH_Result = PATH + '/Images/Results'
```

```
# Enregistrement des données traitées au format "parquet" :
```

```
features_df.drop('scaled_features').write.mode("overwrite").parquet(PATH_Result)
```

Amazon S3 > Compartiments > oc-p9-bigdata-data > Images/ > Results/

Results/

Copier l'URI S3

Objets

Propriétés

Objets (21) Info



Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Actions

Créer un dossier

Charger

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 de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

Rechercher des objets en fonction du préfixe

< 1 >

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	_SUCCESS	-	13 Jun 2024 10:16:26 AM CEST	0 o	Standard
<input type="checkbox"/>	part-00000-da213f1c-4f37-4c28-a5cb-9b86b047c452-c000.snappy.parquet	parquet	13 Jun 2024 10:16:03 AM CEST	17.2 Ko	Standard
<input type="checkbox"/>	part-00001-da213f1c-4f37-4c28-a5cb-9b86b047c452-c000.snappy.parquet	parquet	13 Jun 2024 10:16:03 AM CEST	17.3 Ko	Standard

2. Réalisation de la chaîne de traitement dans le cloud

2.3. Spark jobs

Spark Jobs ^(?)

User: livy

Total Uptime:

Scheduling Mode: FIFO

Completed Jobs: 22

▶ Event Timeline

▼ Completed Jobs (22)

Page: 1

1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group) ▼	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
21 (29)	Job group for statement 29 parquet at NativeMethodAccessorImpl.java:0	2024/06/13 08:15:49	36 s	1/1 (1 skipped)	20/20 (2 skipped)
20 (29)	Job group for statement 29 parquet at NativeMethodAccessorImpl.java:0	2024/06/13 08:15:39	10 s	1/1	2/2
19 (27)	Job group for statement 27 treeAggregate at RowMatrix.scala:171	2024/06/13 08:13:23	29 s	2/2 (1 skipped)	24/24 (2 skipped)
18 (27)	Job group for statement 27 isEmpty at RowMatrix.scala:441	2024/06/13 08:13:21	2 s	1/1 (1 skipped)	1/1 (2 skipped)
17 (27)	Job group for statement 27 treeAggregate at Statistics.scala:58	2024/06/13 08:12:57	24 s	2/2 (1 skipped)	24/24 (2 skipped)
16 (27)	Job group for statement 27 first at RowMatrix.scala:62	2024/06/13 08:12:55	2 s	1/1 (1 skipped)	1/1 (2 skipped)
15 (27)	Job group for statement 27 first at PCA.scala:44	2024/06/13 08:12:51	4 s	2/2	3/3
14 (27)	Job group for statement 27 first at StandardScaler.scala:113	2024/06/13 08:12:50	0.3 s	1/1 (2 skipped)	1/1 (22 skipped)
13 (27)	Job group for statement 27 first at StandardScaler.scala:113	2024/06/13 08:12:14	36 s	1/1 (1 skipped)	20/20 (2 skipped)
12 (27)	Job group for statement 27 first at StandardScaler.scala:113	2024/06/13 08:12:05	9 s	1/1	2/2
11 (19)	Job group for statement 19 treeAggregate at RowMatrix.scala:171	2024/06/13 08:10:36	28 s	2/2 (1 skipped)	24/24 (2 skipped)
10 (19)	Job group for statement 19	2024/06/13 08:10:34	2 s	1/1 (1 skipped)	1/1 (2 skipped)



2. Réalisation de la chaîne de traitement dans le cloud

2.3. Spark jobs

Spark Jobs ^(?)

User: livy

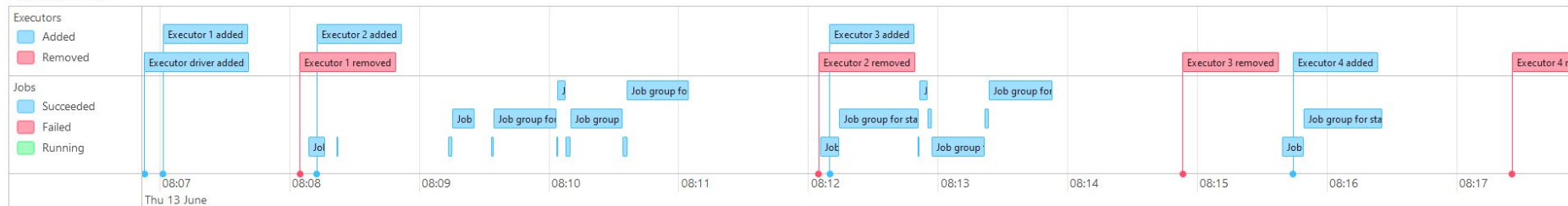
Total Uptime:

Scheduling Mode: FIFO

Completed Jobs: 22

▼ Event Timeline

☐ Enable zooming





Conclusion

- Mise en place d'une instance EMR opérationnelle : les traitements effectués en local ont pu être réalisés sur le cloud.
- Le choix d'AWS et du service EMR-IAM-S3 offre plusieurs avantages :
 - Changement d'échelle en fonction du volume de données assez simple ;
 - Tarification à la demande ;
 - Gestion des accès et sécurisation des données ;
 - Stockage illimité.
- Ce choix s'avérera d'autant plus pertinent avec la croissance du volume de données utilisées.
- Une plus grande expertise permettrait sans doute de mieux configurer et adapter les instances à notre travail.

Merci de votre attention.

