

Formation : Data scientist openclassrooms

Projet 9: Réalisez un traitement dans un environnement Big Data sur le Cloud

Préparé par : Ben Douma Hosni

Sommaire

- Problématique
- Données
- Architecture d'un environnement Big Data
- Spark
- Les services AWS
- Déploiement de la solution en local
- Déploiement de la solution sur cloud AWS
- Quelques interfaces du déploiement
- Conclusion

Problématique

Le start-up de l'AgriTech, "**Fruits!**" cherche à:

- mettre en place une première version du moteur de classification des images de fruits.
- construire une première version de l'architecture **Big Data** vu que le volume de données va **augmenter** très rapidement

Données



Fruits-360 dataset: A dataset of images containing fruits and vegetables

Version: 2020.05.18.0

Dataset propriétés

Le nombre de images: 90483.

Taille de l'ensemble d'entraînement: 67692 images (un fruit/légume par image).

Taille de l'ensemble de test: 22688 images (un fruit/légume par image).

Nombre de classes: 131 (fruits et légumes).

Taille d'image: 100x100 pixels.

Big Data

- un ensemble de données massives, complexes et souvent hétérogènes qui sont difficiles à gérer et à traiter avec des outils traditionnels de gestion de données. La **quantité de données générées chaque jour est en constante augmentation**.
- Pour traiter ces données on doit **distribuer leur stockage et paralléliser leur traitement sur plusieurs ordinateurs** . Cette approche est techniquement possible aujourd'hui grâce à plusieurs technologies

Caractéristiques des données massives

Les 3 V de Big Data :

- **Le Volume** se réfère à la quantité de données générées et stockées.
- **La Variété** fait référence à la diversité des sources et des types de données.
- **La Vélocité** fait référence à la vitesse à laquelle les données sont générées, stockées et analysées.

Les outils pour le Big Data

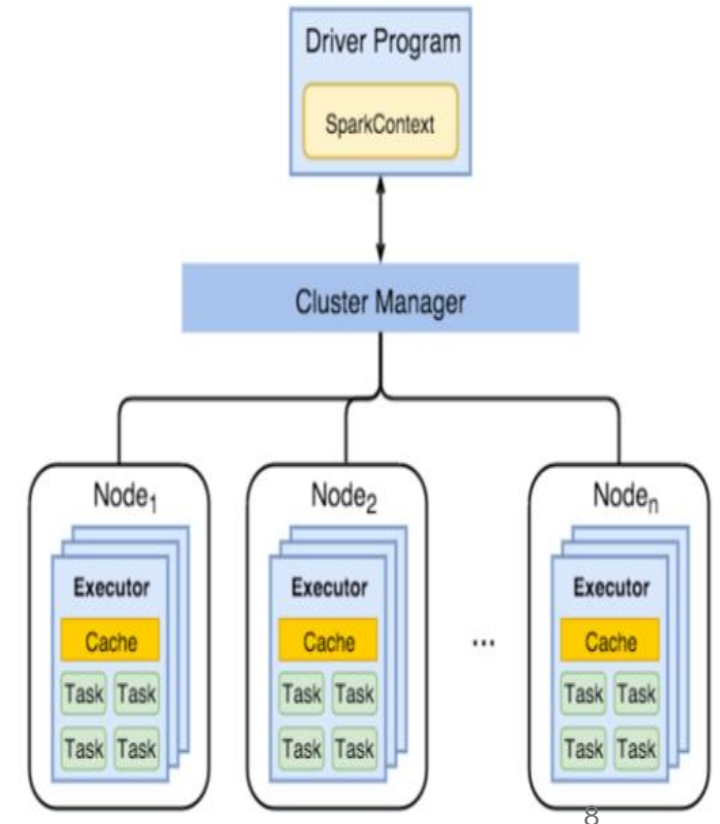
- **Hadoop** : le système de stockage et de traitement de données en cluster le plus populaire
- **Spark** : le moteur de traitement de données rapide pour des traitements distribués et des analyses de données interactives
- **NoSQL** : les systèmes de gestion de bases de données pour les données non structurées

Spark

Apache Spark est une plateforme de traitement sur cluster générique. C'est un moteur de traitement libre, assurant un traitement parallèle et distribué sur des données massives.

Spark est caractérisée par :

- **In-memory Processing** : Les données sont stockées en mémoire, et aussi le traitement.
- **Codé en Scala**, ce langage orienté objet et de programmation fonctionnelle (par nature supporte le parallélisme).
- **Lazy Evolution** : Le calcul est exécuté au dernier



Les services cloud AWS

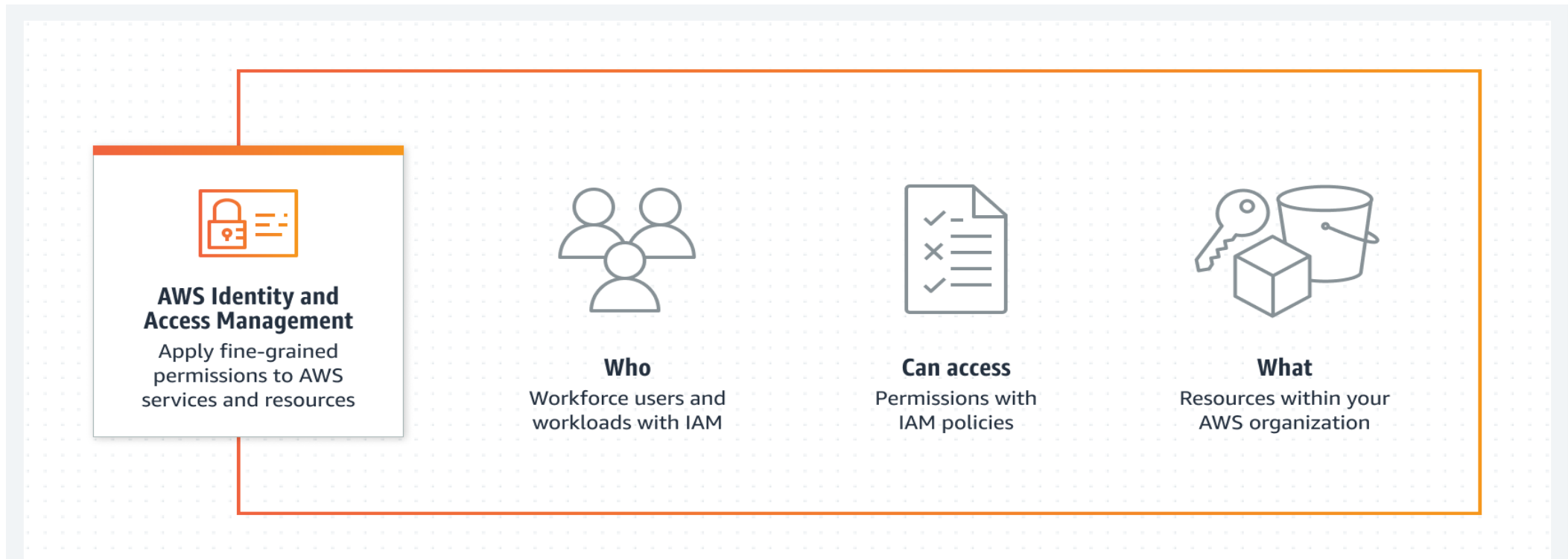
AWS (Amazon Web Services) est le leader mondial des **services cloud** qui motorise le site Amazon.com. C'est la **plateforme de cloud computing** la plus complète du marché avec plus de 200 services proposés.

Les services AWS a utilisé dans ce projet:

- **IAM**
- **S3**

IAM (AWS Identity and Access Management)

- C'est le **service de gestion des identités et des accès d'AWS**.
- **Permet d'identifier, puis autoriser ou interdire l'action selon les droits qui vous ont été accordés** par l'administrateur du compte.

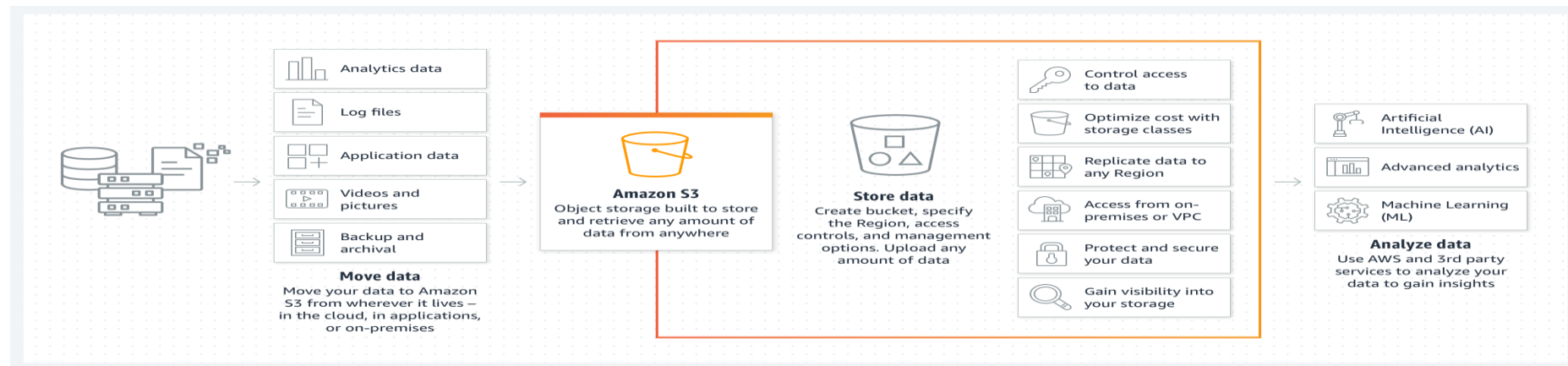


Amazon S3

AWS S3 est **un service Cloud de stockage d'objets**. Accessible par le biais d'une interface web, ce service permet de stocker, de protéger et de restaurer des données et des fichiers à partir de » Buckets » (sceaux).

Amazon S3 présente de nombreux avantages:

- C'est une **solution de stockage à la fois simple et solide**.
- Il n'y a pas de limite fixe en termes de capacité de stockage et de transfert de données
- Les objets S3 sont automatiquement dupliqués sur de multiples serveurs, ce qui réduit le risque de panne au maximum



Déploiement de la solution en local

- 1- Création de la SparkSession
- 2- Préparation des données
- 3- Choix du modèle
- 4- Diffusion des poids
- 5- Extraction de features

SparkSession - in-memory

SparkContext

[Spark UI](#)

Version

v3.5.1

Master

local

AppName

P8

	path	label	features
0	file:/content/data/Test1/43_100.jpg	Test1	[0.9958219, 0.09498338, 0.0, 0.0, 0.0, 0.59119...
1	file:/content/data/Test1/5_100.jpg	Test1	[0.8150559, 0.007707878, 0.0, 0.0, 0.0, 0.7270...
2	file:/content/data/Test1/40_100.jpg	Test1	[0.7846502, 0.055052727, 0.0, 0.0, 0.0, 0.7817...
3	file:/content/data/Test1/41_100.jpg	Test1	[0.80328274, 0.07817287, 0.0, 0.0, 0.0, 0.7249...

Réduction de dimension avec PCA

path	label	features	features_vectors	features_scaled	vectors_pca
file:/content/dat...	Test1	[0.9958219, 0.094...	[0.99582189321517...	[1.80463993488163...	[-14.172223681857...
file:/content/dat...	Test1	[0.8150559, 0.007...	[0.81505590677261...	[0.11238220223625...	[24.3312349446141...
file:/content/dat...	Test1	[0.7846502, 0.055...	[0.78465020656585...	[-0.1722636114148...	[-8.6621640118564...
file:/content/dat...	Test1	[0.80328274, 0.07...	[0.80328273773193...	[0.00216657628296...	[-13.643489946741...
file:/content/dat...	Test1	[0.9480304, 0.031...	[0.94803041219711...	[1.35723550388768...	[-11.344624525250...
file:/content/dat...	Test1	[0.9011911, 0.173...	[0.90119111310755...	[0.91871787151109...	[-12.000703511616...

Déploiement avec Databricks



Databricks est une plateforme de gestion de données et d'analyse basée sur le cloud, conçue pour simplifier et accélérer le traitement des données à grande échelle.

Parmi ces avantages :

- **Simplicité d'utilisation:** Fournit une interface utilisateur intuitive et des outils d'automatisation pour simplifier les tâches complexes de gestion et d'analyse des données.
- **Intégration avec les technologies Big Data:** Intègre nativement des technologies populaires telles que Apache Spark et Apache Hadoop pour le traitement efficace des données à grande échelle.

Déploiement avec Databricks

1- Création d'un cluster et importation du notebook

Pour que l'exploitation des données soit conforme au RGPD. Le cluster est créé dans la région de **paris**

Calculer >

hosni ben douma's Cluster

[Configuration](#) [Notebooks \(1\)](#) [Bibliothèques](#) [Journal des événements](#) [Interface utilisateur Spark](#) [Journaux de driver](#) [Métriques](#) [Applications](#) [Interface utilisateur Spark compute - Maître](#) [Terminer](#) [Modifier](#)

Policy ⓘ

Non restreint

☒ Noeud multiple ☐ Noeud unique

Mode d'accès ⓘ [Accès mono-utilisateur ⓘ](#)

[Utilisateur unique](#) [hosni ben douma](#)

Performances

Version de Databricks Runtime

15.0 ML (includes Apache Spark 3.5.0, Scala 2.12)

☐ Utiliser l'accélération de Photon ⓘ

Type de worker ⓘ

	Nombre minimal de workers	Nombre maximal de workers	Actuel
i3.xlarge 30,5 Go de mémoire, 4 cœurs	2	2	2

Utilisez les types d'instances Fleet pour améliorer la disponibilité et le placement des instances Spot [En savoir plus](#)

Type de driver

i3.xlarge 30,5 Go de mémoire, 4 cœurs

Résumé

2-2 workers	61-61 Go de mémoire 8-8 cœurs
1 driver	30,5 Go de mémoire, 4 cœurs
Environnement d'exécution	15.0.x-cpu-ml-scala2.12
Unity Catalog i3.xlarge 3 DBU/heure	

2- Stockage des images sur S3 et accès aux données

Pour respecter le RGPD l'instance S3 est créée dans la région **d'Irlande**

- Création d'un bucket "projet9"
- Chargement de données sur "projet9"
- Création d'un IAM et enregistrement des clés d'accès
- Accès aux données

```
images_df = spark.read.format("image").load("/mnt/s3/Test1")
```

▼  images_df: pyspark.sql.dataframe.DataFrame

▼ image: struct

origin: string
height: integer
width: integer
nChannels: integer
mode: integer
data: binary

3- Extractions et enregistrement des features

Le dataset **df** :

```
+-----+-----+
|path                                     |label|
+-----+-----+
|dbfs:/mnt/s3/Test1/Apple Crimson Snow/r_25_100.jpg|Apple Crimson Snow|
|dbfs:/mnt/s3/Test1/Apple Crimson Snow/r_24_100.jpg|Apple Crimson Snow|
|dbfs:/mnt/s3/Test1/Apple Crimson Snow/r_30_100.jpg|Apple Crimson Snow|
|dbfs:/mnt/s3/Test1/Apple Crimson Snow/r_32_100.jpg|Apple Crimson Snow|
|dbfs:/mnt/s3/Test1/Apple Crimson Snow/r_26_100.jpg|Apple Crimson Snow|
+-----+-----+
```

Extraction des features avec **MobileNetV2**

```
▼ features_df: pyspark.sql.dataframe.DataFrame
  path: string
  label: string
  ▼ features: array
    element: float
```

Enregistrement des features sous formes des parquets

4-Stockage du résultats dans s3

Amazon S3 > Compartiments > projet9 > Results/

Results/

Copier l'URI S3

ObjetsPropriétés

Objets (27) 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"/>	 _committed_8183172264757496160	-	15 Apr 2024 04:40:49 PM CEST	2.4 Ko	Standard
<input type="checkbox"/>	 _started_8183172264757496160	-	15 Apr 2024 04:40:12 PM CEST	0 o	Standard
<input type="checkbox"/>	 _SUCCESS	-	15 Apr 2024 04:40:50 PM CEST	0 o	Standard
<input type="checkbox"/>	 part-00000-tid-8183172264757496160-cc46f476-b707-48b8-bf9d-e927874591cd-433-1-c000.snappy.parquet	parquet	15 Apr 2024 04:40:40 PM CEST	59.8 Ko	Standard
<input type="checkbox"/>	 part-00001-tid-8183172264757496160-cc46f476-b707-48b8-bf9d-e927874591cd-425-1-c000.snappy.parquet	parquet	15 Apr 2024 04:40:37 PM CEST	54.5 Ko	Standard
<input type="checkbox"/>	 part-00002-tid-8183172264757496160-cc46f476-b707-48b8-bf9d-e927874591cd-426-1-c000.snappy.parquet	parquet	15 Apr 2024 04:40:35 PM CEST	62.4 Ko	Standard

Quelques interfaces du déploiement

hosni ben douma's Cluster  

Plus ...

Terminer

Modifier

Configuration

Notebooks (1)

Bibliothèques

Journal des événements

Interface utilisateur Spark

Journaux de driver

Métriques

Applications

Interface utilisateur Spark compute - Maître ...

[Ouvrir dans un nouvel onglet](#)

Jobs

Stages

Storage

Environment

Executors

SQL / DataFrame

JDBC/ODBC Server

Structured Streaming

Connect

Spark Jobs (?)

User: root

Started At: 2024/04/15 13:34:57

Total Uptime: 2.6 h


Scheduling Mode: FAIR


Completed Jobs: 17

▼ Event Timeline

☐ Enable zooming

Executors

 Added


 Removed

Executor 1 added

Executor 0 added


Executor driver added

Jobs

 Succeeded

 Failed

 Running

[Ouvrir dans un nouvel onglet](#) 



Spark Master at spark://10.206.211.2:7077

URL: spark://10.206.211.2:7077
Workers: 2 Alive, 0 Dead, 0 Decommissioned, 0 Unknown
Cores in use: 8 Total, 8 Used
Memory in use: 49.8 GiB Total, 39.8 GiB Used
Resources in use:
Applications: 1 [Running](#), 0 [Completed](#)
Drivers: 0 Running (0 Waiting), 0 Completed (0 Killed, 0 Failed, 0 Error, 0 Relaunching)
Status: [ALIVE](#)

▼Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20240415133429-10.206.209.33-46559	10.206.209.33:46559	ALIVE	4 (4 Used)	24.9 GiB (19.9 GiB Used)
worker-20240415133430-10.206.201.150-33575	10.206.201.150:33575	ALIVE	4 (4 Used)	24.9 GiB (19.9 GiB Used)

▼Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20240415133507-0000 (kill)	Databricks Shell	8	20396		2024/04/15 13:35:07	root	RUNNING	2.7 h

Conclusion

Grâce à Databricks, la configuration des services cloud et des bibliothèques est simplifiée.

Spark, avec son architecture distribuée, est un puissant outil pour le traitement des données massives.

Cependant, les coûts peuvent augmenter et la complexité de la configuration initiale peut poser des défis. Malgré cela, l'association de Spark avec les services cloud AWS offre une solution robuste pour relever les défis de gestion et d'analyse des données massives