

Réalisez un traitement dans un environnement Big Data sur le Cloud



Contenu de la présentation



01

Problématique
e

02

Création de
l'environnement
t

03

Traitement

04

Démonstration
n

05

Conclusion



01 Problématique



Mission

Pour la start-up "Fruits", il nous faut configurer un environnement big data, et y exécuter une réduction de dimensionnalité



La Donnees

Fruits-360 dataset :

- Nombres total d'images : 94110
- Nombre total de classes : 141
- exemples :



02

Création de l'environnemen t



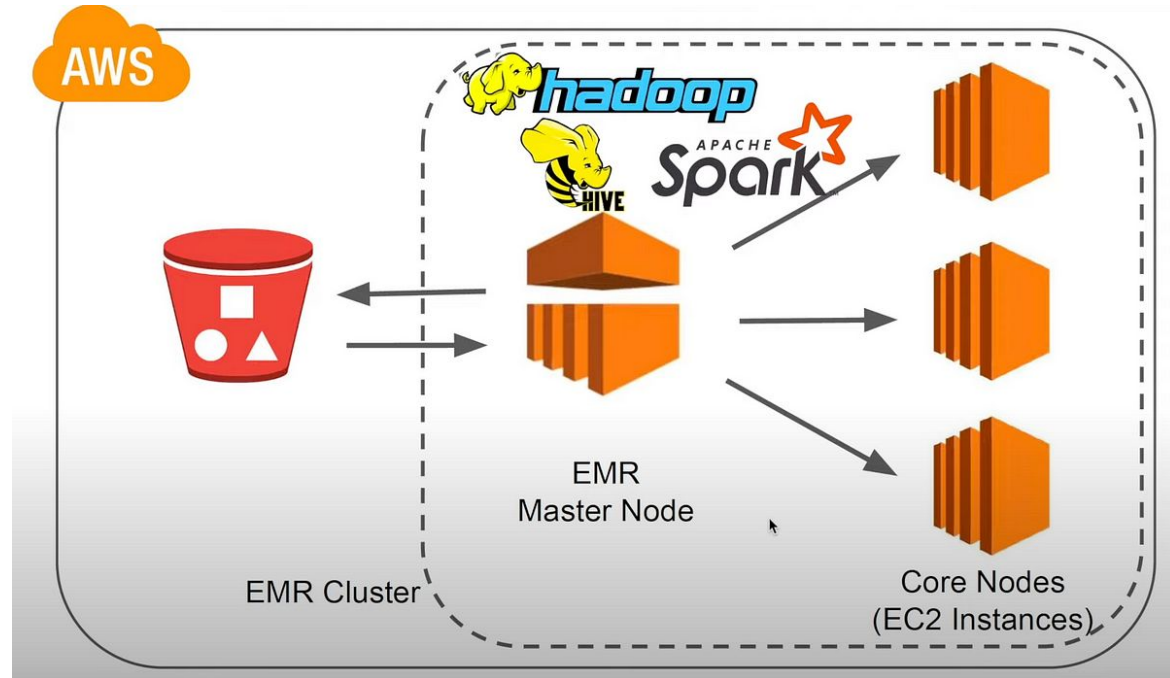
Stockage



AWS Simple Storage Service (S3)

- Espace de stockage persistant
- Peut être comparé à un drive ou un serveur FTP
- On peut y créer un "bucket" et l'appeler depuis nos applications autorisées

Cluster Machines



Avec EMR Cluster, on peut répartir les calculs nécessaires au projet sur différents nœuds afin de répartir la charge

Ajout prochaine slide, explication spark hadoop, spark lazy

Spark “on top of Hadoop”

Spark est un framework de traitement de données distribué

Hadoop fournit le stockage (HDFS) et la gestion des ressources (YARN).

Spark exécute des traitements rapides en mémoire en exploitant ces ressources.

Sur AWS EMR, on utilise souvent S3 à la place de HDFS, ce qui permet une persistance de la donnée .

Avantages :

- Compatibilité avec Hadoop : Spark peut utiliser l'infrastructure existante.
- Traitement 100x plus rapide que Hadoop MapReduce (grâce à l'in-memory).
- Utilisation de HDFS / S3 pour stocker et traiter les Big Data efficacement.
- Supporte le batch, le streaming et le machine learning (contrairement à Hadoop MapReduce).

Notre Setup

p8

Mise à jour il y a moins d'une minute

Résilier

Cloner dans AWS CLI

Cloner


▼ Récapitulatif

Informations sur le cluster

ID de cluster

j-2TD7GU37W1KOT

ARN du cluster

 arn:aws:elasticmapreduce:eu-west-3:724772064051:cluster/j-2TD7GU37W1KOT

Configuration de cluster

Groupes d'instances

Capacité

1 primaire(s) | 1 unité(s) principale(s) | 2 tâche(s)

Applications

Version d'Amazon EMR

emr-6.3.0

Applications installées


JupyterHub 1.2.2, Spark 3.1.1, TensorFlow 2.4.1

Gestion des clusters


Destination des journaux dans Amazon S3

[aws-logs-724772064051-eu-west-3/elasticmapreduce](#)

DNS public du nœud primaire


 ec2-15-188-60-131.eu-west-3.compute.amazonaws.com

Connexion au nœud primaire à l'aide de SSH

Connexion au nœud primaire à l'aide de SSM 

Statut et heure

Statut

 [Action d'amorçage](#)

Heure de création

4 mars 2025 18:49 (UTC+01:00)

Temps écoulé

6 minutes, 29 secondes

Propriétés

Actions d'amorçage

Instances (Matériel)

Étapes

Applications

Configurations

Surveillance

Évènements

Identifications (1)


Journaux de cluster

Info

Archiver les fichiers journaux dans Amazon S3

Activé

Emplacement Amazon S3

[s3://aws-logs-724772064051-eu-west-3/elasticmapreduce/](#) 

Chiffrement pour les journaux

Désactivé

Résiliation du cluster et remplacement des nœuds

Info

Option de résiliation

Résilier manuellement le cluster

Protection contre la résiliation

Désactivé

Temps d'inactivité

-

Remplacement des nœuds défectueux

Activé

Modifier

03

Traitement



Lazy evaluation

Spark utilise une évaluation paresseuse (lazy evaluation), ce qui signifie que les transformations ne sont pas exécutées immédiatement.

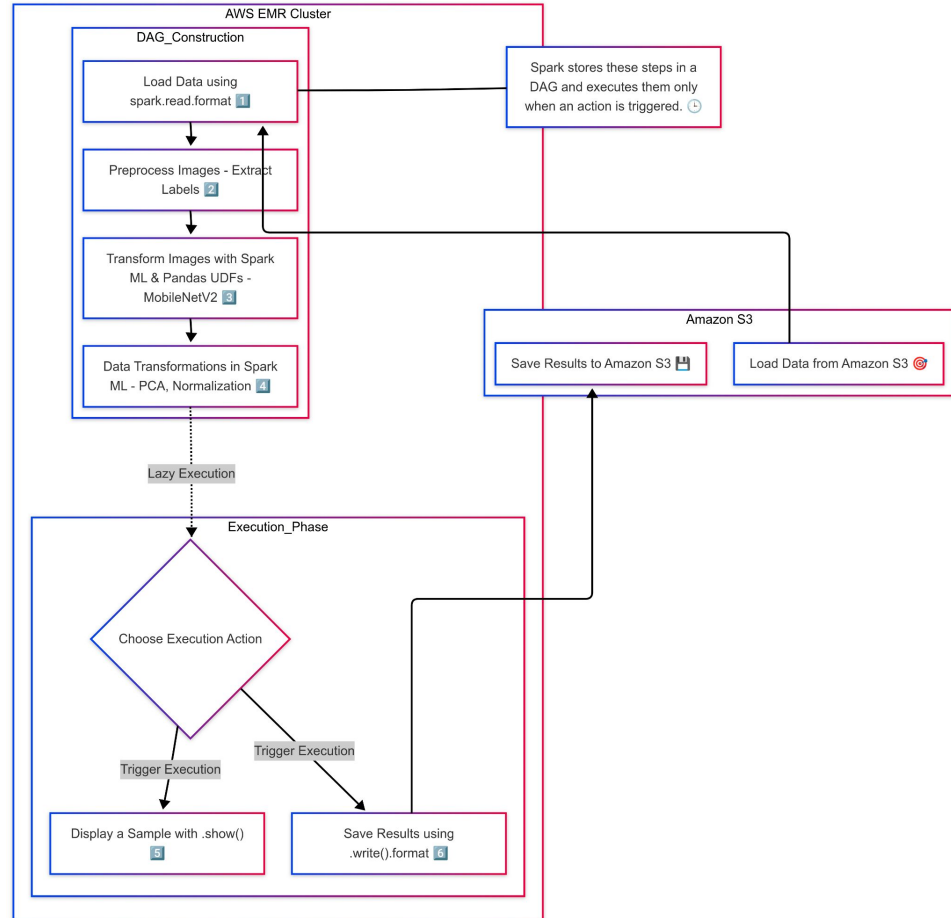
À la place, Spark construit un DAG (Directed Acyclic Graph), qui représente le workflow des opérations sur les données.

L'exécution ne commence que lorsqu'une action est appelée.

Un DAG en Spark est constitué de :

- Nœuds (Nodes) : Chaque nœud représente une transformation ou une action.
- Arcs (Edges) : Les arcs dirigés définissent l'ordre des transformations.
- Stages (Étapes de calcul) : Spark divise le DAG en stages pour paralléliser l'exécution.

L'exécution du notebook



Gestion de la donnée intermédiaire (entre l'input et l'output)

Spark est un framework de traitement de données distribué

Hadoop fournit le stockage (HDFS) et la gestion des ressources (YARN).

Spark exécute des traitements rapides en mémoire en exploitant ces ressources.

Sur AWS EMR, on utilise souvent S3 à la place de HDFS, ce qui permet une persistance de la donnée .

Avantages :

- Compatibilité avec Hadoop : Spark peut utiliser l'infrastructure existante.
- Traitement plus rapide que Hadoop MapReduce (grâce à l'in-memory).
- Utilisation de HDFS / S3 pour stocker et traiter les Big Data efficacement.
- Supporte le batch, le streaming et le machine learning (contrairement à Hadoop MapReduce).

Spark UI

Spark Jobs ^(?)

User: livy

Total Uptime: 18 min

Scheduling Mode: FIFO

Active Jobs: 1

Completed Jobs: 3

▶ Event Timeline

▼ Active Jobs (1)

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
3 (15)	Job group for statement 15 first at RowMatrix.scala:62 (kill)	2025/03/06 15:06:57	3.0 min	0/2	0/710 (1 running)

Page: 1

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

▼ Completed Jobs (3)

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
2 (15)	Job group for statement 15 first at PCA.scala:44	2025/03/06 15:00:22	6.6 min	2/2	710/710
1 (12)	Job group for statement 12 showString at NativeMethodAccessorImpl.java:0	2025/03/06 14:58:26	7 s	1/1	1/1
0 (6)	Listing leaf files and directories for 131 paths: s3://p8-data-nm/Test/Apple Braeburn. ... load at NativeMethodAccessorImpl.java:0	2025/03/06 14:57:08	9 s	1/1	131/131

Spark UI

Stages for All Jobs

Active Stages: 1

Pending Stages: 1

Completed Stages: 4

Active Stages (1)

Page: 1

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

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
5	Job group for statement 15 first at RowMatrix.scala:62	2025/03/06 15:06:57	3.6 min	0/1 (1 running)			43.5 MiB	

Page: 1

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

Pending Stages (1)

Page: 1

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

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	rdd at PCA.scala:89	Unknown	Unknown	0/709				

Page: 1

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

Completed Stages (4)

Page: 1

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

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
3	Job group for statement 15 first at PCA.scala:44	2025/03/06 15:04:03	2.9 min	1/1			43.5 MiB	
2	Job group for statement 15 rdd at PCA.scala:89	2025/03/06 15:00:22	3.6 min	709/709	98.4 MiB			87.0 MiB
1	Job group for statement 12 showString at NativeMethodAccessorImpl.java:0	2025/03/06 14:58:26	3 s	1/1				
0	Listing leaf files and directories for 131 paths: s3://p8-data-nm/Test/Apple Braeburn. ... load at NativeMethodAccessorImpl.java:0	2025/03/06 14:57:08	5 s	131/131				

Spark UI

Summary

	▲ RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(2)	0	17.5 MiB / 5.1 GiB	8.6 MiB	2	1	0	238	239	10 min (1 s)	33 MiB	43.5 MiB	29.2 MiB	0
Dead(9)	0	126 KiB / 42.9 GiB	0.0 B	18	0	0	604	604	15 min (3 s)	65.4 MiB	0.0 B	57.8 MiB	0
Total(11)	0	17.6 MiB / 48 GiB	8.6 MiB	20	1	0	842	843	25 min (5 s)	98.4 MiB	43.5 MiB	87 MiB	0

Executors

Show 20 entries

Search:

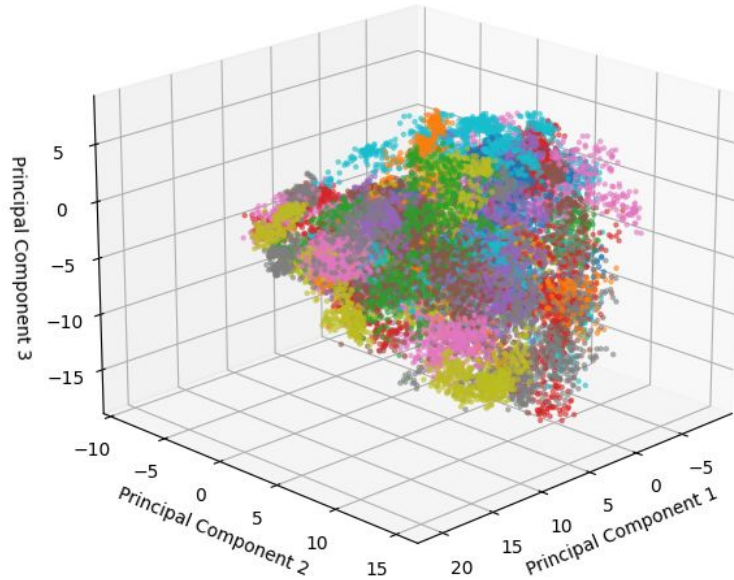
Executor ID	▲ Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
driver	ip-172-31-32-79.eu-west-3.compute.internal:45743	Active	0	8.8 MiB / 353.4 MiB	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B		Thread Dump
1	ip-172-31-40-106.eu-west-3.compute.internal:35935	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
2	ip-172-31-47-119.eu-west-3.compute.internal:33455	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
3	ip-172-31-33-28.eu-west-3.compute.internal:42649	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
4	ip-172-31-47-119.eu-west-3.compute.internal:43985	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	58	58	10 s (0.3 s)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
5	ip-172-31-40-106.eu-west-3.compute.internal:38279	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	36	36	7 s (0.3 s)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
6	ip-172-31-33-28.eu-west-3.compute.internal:42957	Dead	0	0.0 B / 4.8 GiB	0.0 B	2	0	0	37	37	8 s (0.4 s)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
7	ip-172-31-33-28.eu-west-3.compute.internal:40887	Dead	0	42.1 KiB / 4.8 GiB	0.0 B	2	0	0	1	1	3 s (0.2 s)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
8	ip-172-31-33-28.eu-west-3.compute.internal:34049	Active	0	8.7 MiB / 4.8 GiB	8.6 MiB	2	1	0	238	239	10 min (1 s)	33 MiB	43.5 MiB	29.2 MiB	stdout stderr	Thread Dump
9	ip-172-31-47-119.eu-west-3.compute.internal:44755	Dead	0	42 KiB / 4.8 GiB	0.0 B	2	0	0	235	235	7.2 min (1 s)	32.7 MiB	0.0 B	28.9 MiB	stdout stderr	Thread Dump
10	ip-172-31-40-106.eu-west-3.compute.internal:45969	Dead	0	42 KiB / 4.8 GiB	0.0 B	2	0	0	237	237	7.2 min (1 s)	32.7 MiB	0.0 B	28.9 MiB	stdout stderr	Thread Dump

Showing 1 to 11 of 11 entries

Previous **1** Next

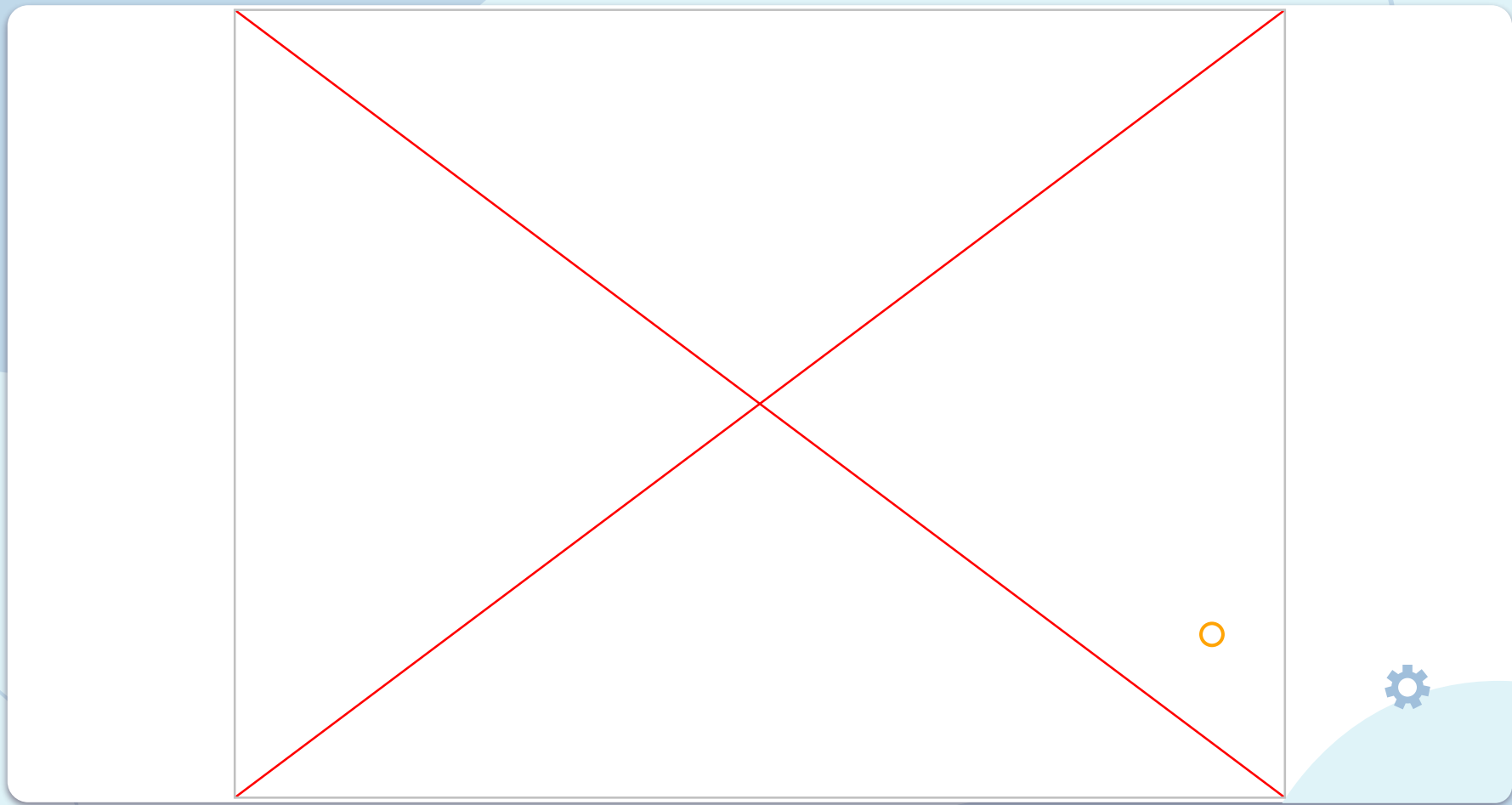
Résultat du PCA

PCA Visualization (3D)



04 Démonstration





05

Conclusion



Conclusion

Grâce à AWS EMR et Apache Spark, nous avons pu tirer parti du calcul distribué pour traiter efficacement le dataset Fruits-360.

Cette approche a permis d'améliorer la scalabilité et l'efficacité du traitement des données, démontrant ainsi la puissance du Big Data sur le Cloud pour des cas d'usage réels en machine learning.

Amélioration possible:

- L'utilisation d'outils de CI/CD
- Des monitoring avancés (ex: Prometheus, Grafana)
- Des tests automatisés assureraient un passage à l'échelle plus fluide et une meilleure fiabilité des traitements.