Fruits!

Déployez un modèle dans le cloud

Présenté par: Lynda HADJEMI
Parcours : Data scientiste

Sommaire:

Fruits!

1-Contexte et problématique

Fruits!

La start-up AgriTech, nommée "Fruits!", cherche à proposer des solutions innovantes pour la récolte des fruits, tout en préservant la biodiversité de ces derniers.

L'objectif est de permettre des traitements spécifiques pour chaque espèce de fruits en développant des robots cueilleurs intelligents.

- Se faire connaître en mettant à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.
- Cette application permettre de:
  - sensibiliser le grand public à la biodiversité des fruits et de mettre en place une première version du moteur de classification des images de fruits.
  - construire une première version de l'architecture Big Data nécessaire.

2- Jeu de données

Fruits!

Jeu de données **Fruits-360** : un ensemble de données d'images de format jpg contenant des fruits en plusieurs variété et plusieurs espèces.

Nombre d'images : 90 483

- Training set : 67 692
- Test set : 22 688

Données labellisés(131 fruits) → Image size : 100px*100px
Couleur : R,G,B

Multi-fruits : 103 images.

2-Jeu de données:

Fruits!

- Photographiés à plusieurs angles.
- Fond d’images éliminé en blanc.
- Plusieurs variétés de fruits .

Avocado

Avocado ripe

Banana

Banana
Lady Finger

Banana red


BIG DATA
ARCHITECTURE
ANALYTIXLABS

3- BIG DATA:

Le terme "Big Data" fait référence à des ensembles de données volumineux, complexes et variés qui dépassent les capacités des méthodes traditionnelles de gestion et d'analyse de données.

Le concept de Big Data est souvent caractérisé par les trois V suivants :

- Volume: La taille gigantesque de données.
- Variété: La diversité de types de données.
- Vélocité: La rapidité à laquelle de nouvelles données sont générées.

4-les différentes briques d'architecture choisies sur le cloud:

4.1- Prestataire cloud choisi:

AWS (Amazon Web Services) : une plateforme de services cloud proposée par Amazon. Il offre une large gamme de services, y compris le calcul, le stockage, la base de données, l'analyse, l'apprentissage automatique, la sécurité, la gestion des identités, le réseau, le déploiement d'applications

4.2-IAAS ou PAAS ?

Solution IAAS:(Infrastructure as a Service)

- Les serveurs sont vierges.
- Installation de tous les outils pour notre script (Spark, Java, Python, Jupyter...) par nous-même.
- Inconvénient: Chronophage.

Solution PAAS: (Platform as a Service)

- Les Framework importants sont déjà installés.
- Les patchs de sécurité sont automatiquement mis à jour
- Faciliter de clonage ses clusters.

➢ Pour ce projet, la solution PaaS est choisie.

4-les différentes briques d'architecture choisies sur le cloud:

EC2: Elastic Compute Cloud

- Des serveurs virtuels dans le cloud.

S3 (Simple Storage Service).

- permet de stocker des objets, tels que des fichiers et des données

L'instance m5.large est un type d'instance de aws.

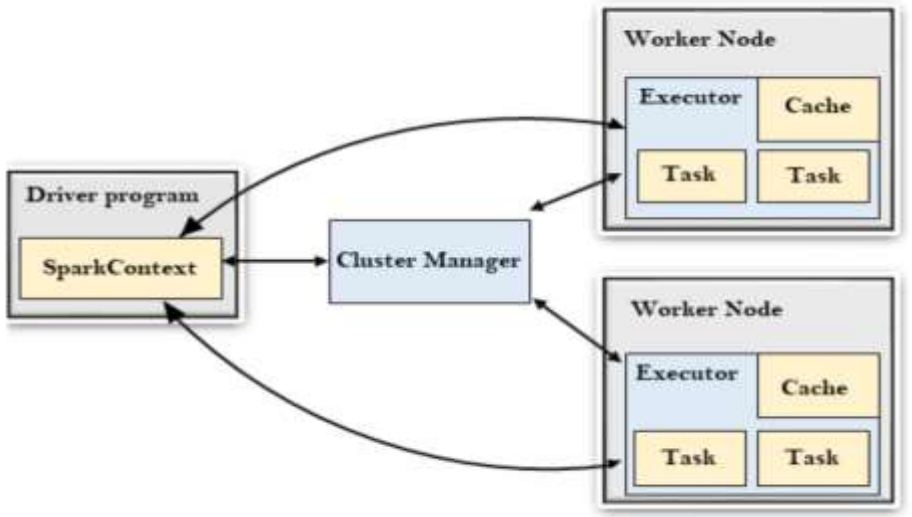
- **Taille de l'Instance :** x.Large.
- **vCPU :** 4 unités de calcul virtuelles.
- **Mémoire :** Environ 16Go de RAM.

IAM (Identity and Access Management) est le service de gestion des identités et des accès d'Amazon Web Services (AWS).

Spark:

Spark : est un framework open-source. conçu pour fournir une plateforme rapide et unifiée pour l'analyse de données à grande échelle.

- Traitement en Mémoire
- APIs Polyglottes
- Traitement de Données Distribué
- Flexibilité dans le Stockage des Données



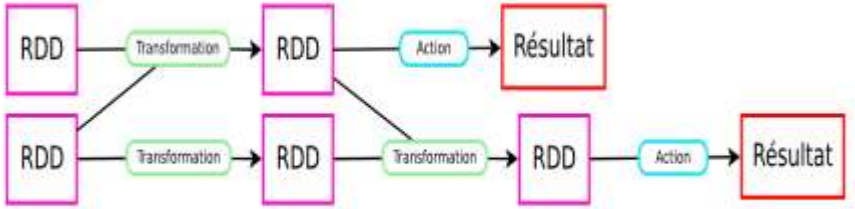
Spark architecture:

- Les instructions sont envoyées au driver lorsqu'on interroge Spark grâce à PySpark.
- La communication avec le driver s'effectue grâce au SparkContext.
- Le gestionnaire de cluster instancie les différents workers.
- Les workers instancient un exécuteur.
- Les exécuteurs sont chargés d'exécuter les différentes tâches de calcul.

Spark RDD (Resilient Distributed Datasets):

Tolérance aux pannes:

- transformations et les actions réalisées sur les RDD permettent de construire un **graphe acyclique orienté (DAG).**
- Les connexions entre les nœuds sont soit des transformations, soit des actions.
- Le graphe est dit *acyclique* car aucun RDD ne permet de se transformer en lui-même via une série d'actions.
- Un RDD peut être régénéré à partir de ses RDD parents.



graphe acyclique orienté (DAG : "directed acyclic graph«)

5-Création de l’Environnement Big data (IAM,S3, EC2):

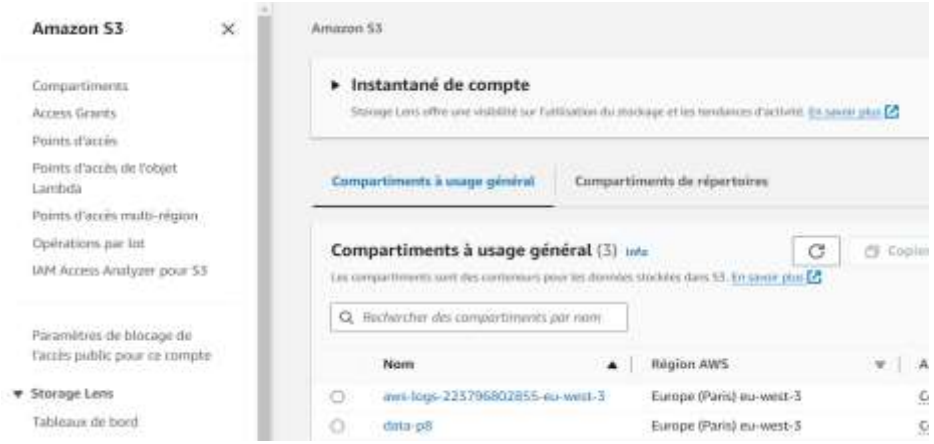
1-) Amazon IAM (Identity and Access Management) – Autorisations Utilisateur

- permet de gérer l'accès aux ressources AWS de manière sécurisée
- Autorisations tout type de manipulation dans les buckets S3

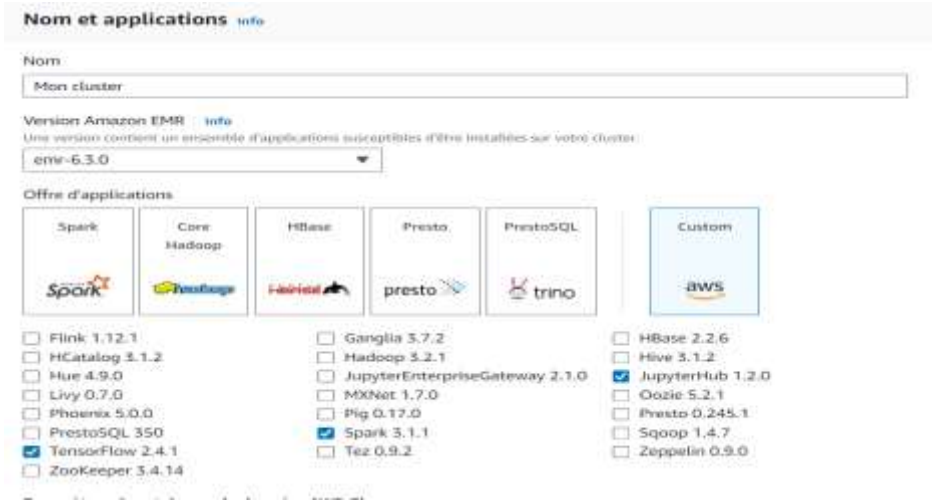
2-Amazon IAM – Configuration AWS:

Paire de clés: Permet de sécuriser l’accès à la connexion Utilisateur
eu-west-3: Serveurs à Paris -> Respecte le RGPD
RGPD: Règlement Général sur la Protection des Données
Nous oblige à utiliser des serveurs en Europe uniquement

3-Amazon S3:



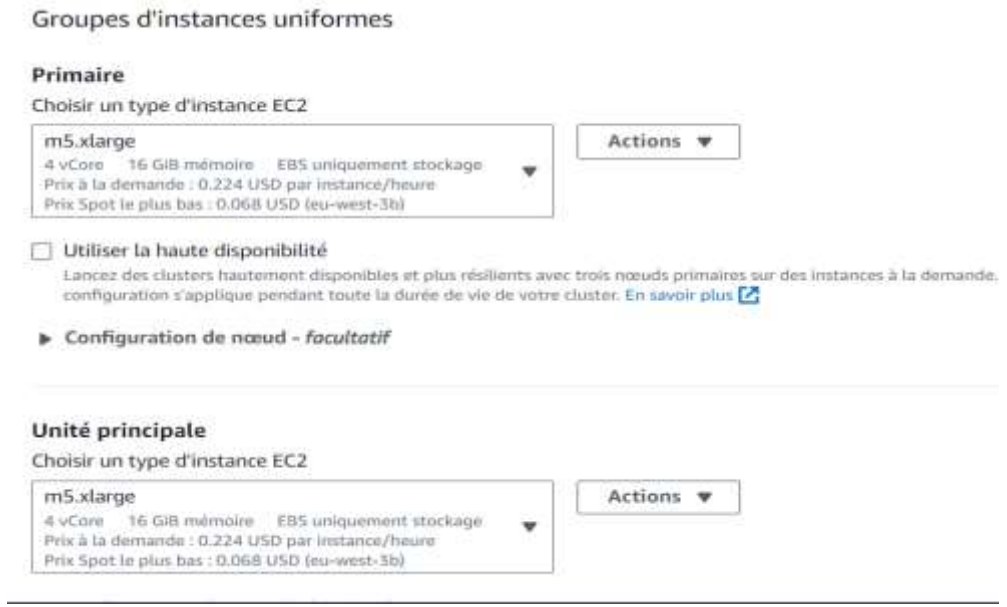
4-Amazon EMR:



5-Amazon EMR Boostrapping:

- wheel : Accélère l'installation des packages Python.
- pillow : Bibliothèque de traitement d'image en Python.
- pyarrow : Lit les fichiers parquets et les convertit en Dataframe.
- boto3 + s3fs + fsspec : Permettent d'interagir avec Amazon S3.
- Pandas : Pour les differentes manipulations des dataframes.

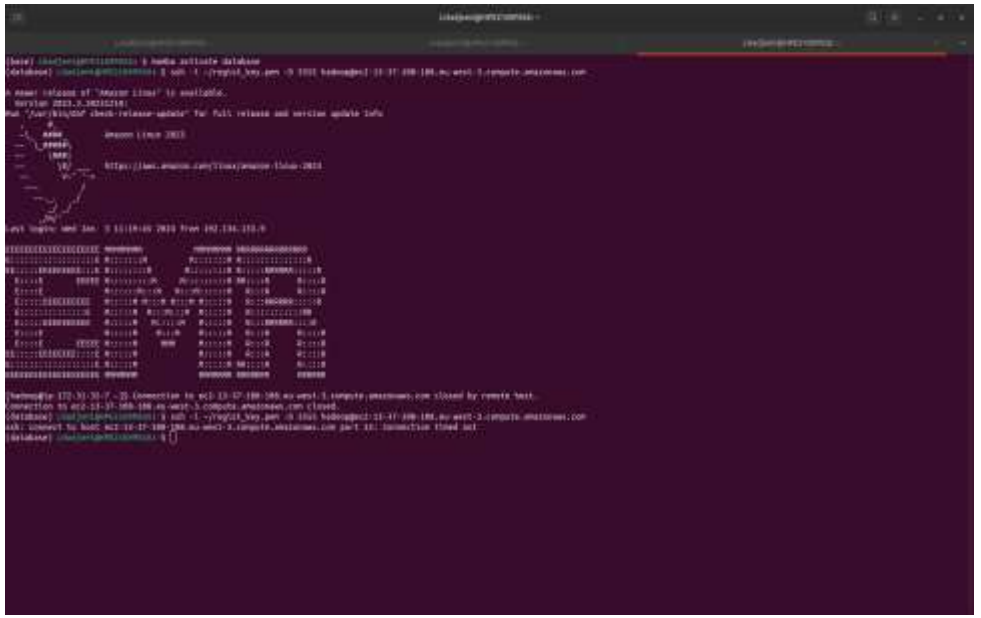
6-Amazon EMR choix d'instances:



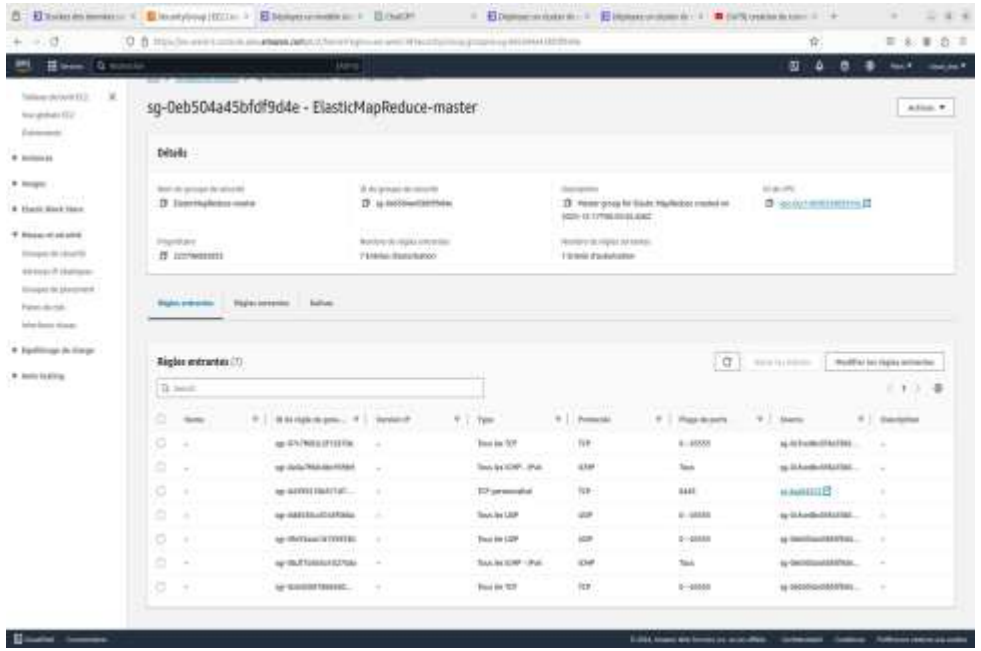
7- Création des tunnel SSH:

Pour pouvoir s’éxecuter et ouvrir le JupyterHub, la création des clés ssh est indispensable.

- On établit la connexion au bash une fois le clusters créer.



- Autorisation les tunnel ssh port 5555 sur aws.



- Configuration de l’extension Foxyproxy pour le navigateur.

6-Chaine de traitement Spark en local:

Création de SparkSession

Une instance SparkSession est la façon dont Spark exécute Les fonction définies par l'utilisateur

Load Data

Spark Dataframe:

```
spark.read.format("binaryFile") \
  .option("pathGlobFilter", "*.jpg") \
  .load(PATH_Data)
```

```
root
 |-- path: string (nullable = true)
 |-- modificationTime: timestamp (nullable = true)
 |-- length: long (nullable = true)
 |-- content: binary (nullable = true)
 |-- label: string (nullable = true)
```

Preprocessing

Featurize Pandas-udf

Modèle transfert Learning MobileNetV2

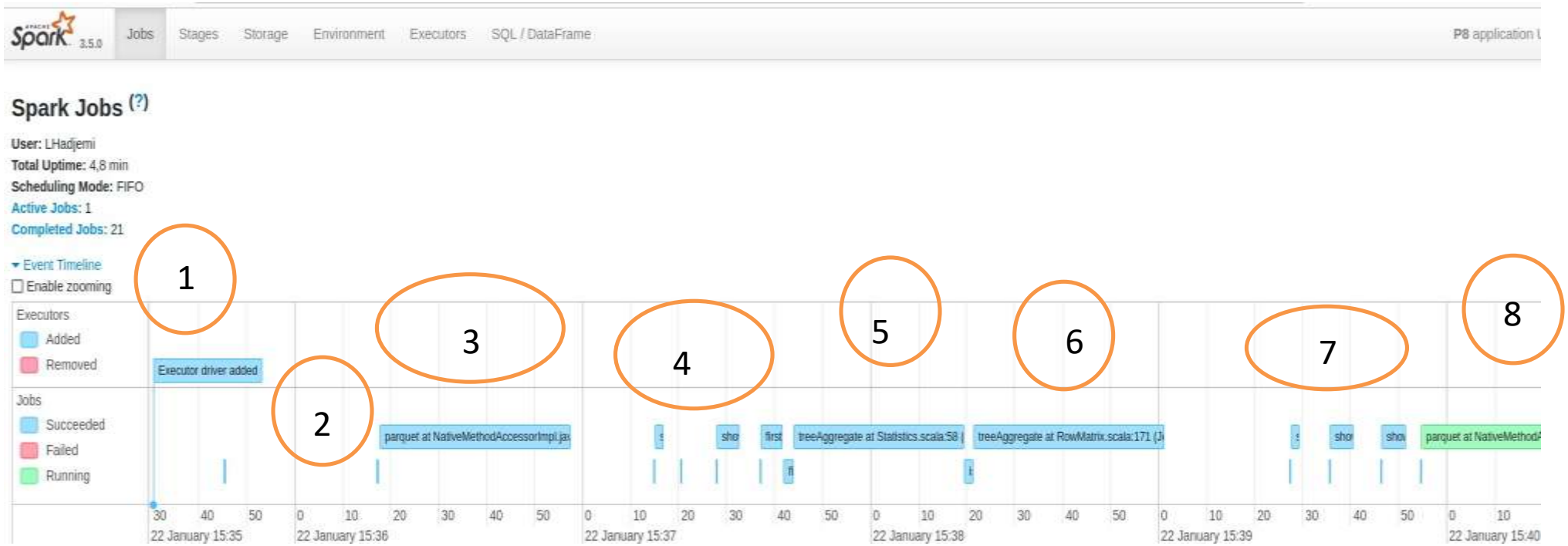
	path	label	features
0	file:/media/LHadjemi/Data/deployer-un-modele-d...	Avocado	[1.2062845, 0.00094540615, 0.0, 0.0, 0.0828499...
1	file:/media/LHadjemi/Data/deployer-un-modele-d...	Avocado	[0.28114322, 0.0, 0.015231324, 0.0, 0.0, 0.0, ...
2	file:/media/LHadjemi/Data/deployer-un-modele-d...	Avocado	[0.36894366, 0.0020400512, 0.13478272, 0.03039...
3	file:/media/LHadjemi/Data/deployer-un-modele-d...	Avocado	[1.6595063, 0.0, 0.12816602, 0.003942761, 0.01...
4	file:/media/LHadjemi/Data/deployer-un-modele-d...	Avocado	[0.64518535, 0.0, 0.0017532817, 0.0, 0.0465762...

PCA (spark.ml)

Ecriture et puis lecture vers pd.DataFrame au format paquet, engine pyarrow

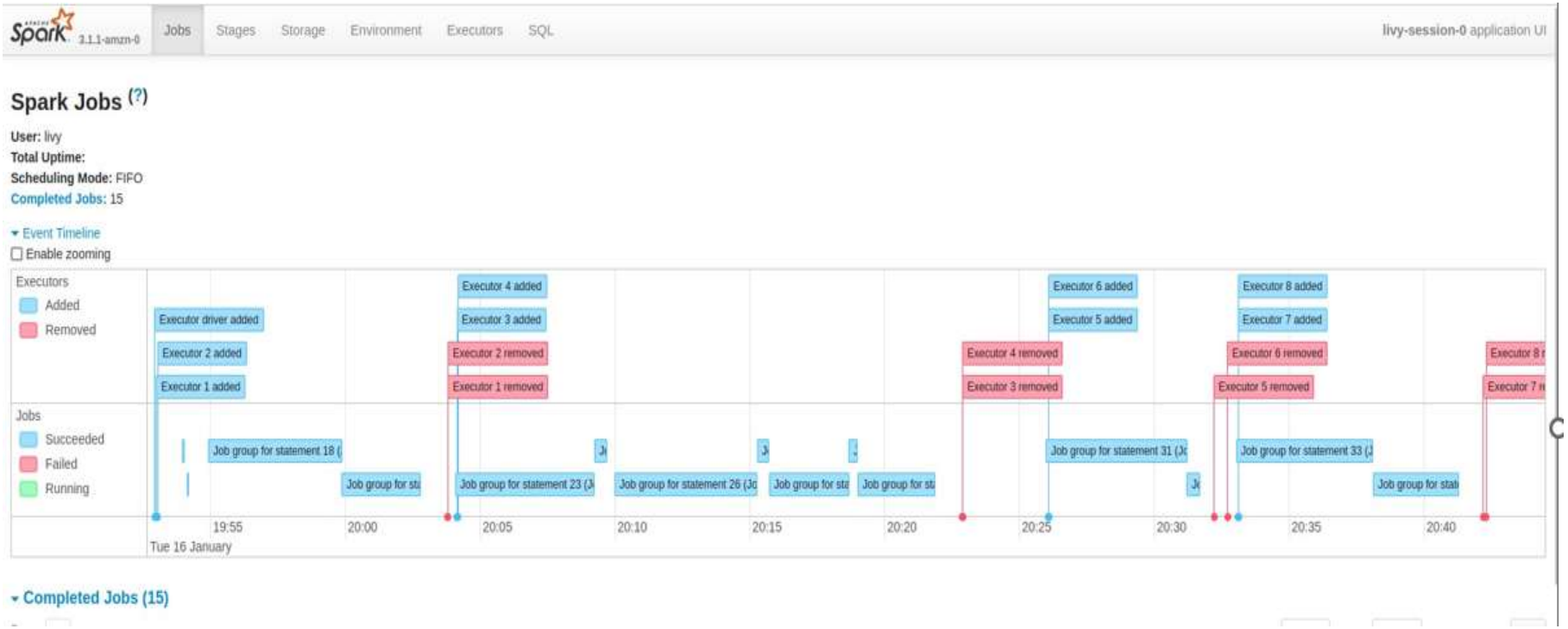
pca_vectors	features_array
{'type': 1, 'size': None, 'indices': None, 'va...	[-16.95106, -4.3774457, -11.170761, -10.679921...
{'type': 1, 'size': None, 'indices': None, 'va...	[-17.011036, -13.817756, -10.5200405, 2.280178...
{'type': 1, 'size': None, 'indices': None, 'va...	[-15.204871, 0.99675465, -8.867973, -0.4300575...
{'type': 1, 'size': None, 'indices': None, 'va...	[-15.122531, 2.5644705, -9.907209, -1.5638206,...
{'type': 1, 'size': None, 'indices': None, 'va...	[-1.7547172, -2.3506665, -12.355202, -4.484647...

6-Chaine de traitement SparkUI en local:



1-Activation des machine de clusters (création de driver)
2-Chargement des données images.
3-La conversion des images au format binaires.
4-Application du model mobilenetv2
5-Données au format paquet.
6-lecture de tableau pandas
6-chargement de PCA
7- Application de PCA sur les features.
8-chargement des données au format paquet

7-Chaine de traitement SparkUI sur le cloud :



7-1-Comparaison du temps d'exécution:

Nombres d'instances	Temps d'exécution
1 primaire (m5.xlarge) 2 unité principale (m5.xlarge)	3039.4540 Seconde
1 primaire (m5.xlarge) 4 unité principale (m5.xlarge)	1508,6541 Seconde

7-Chaine de traitement SparkUI sur le cloud :

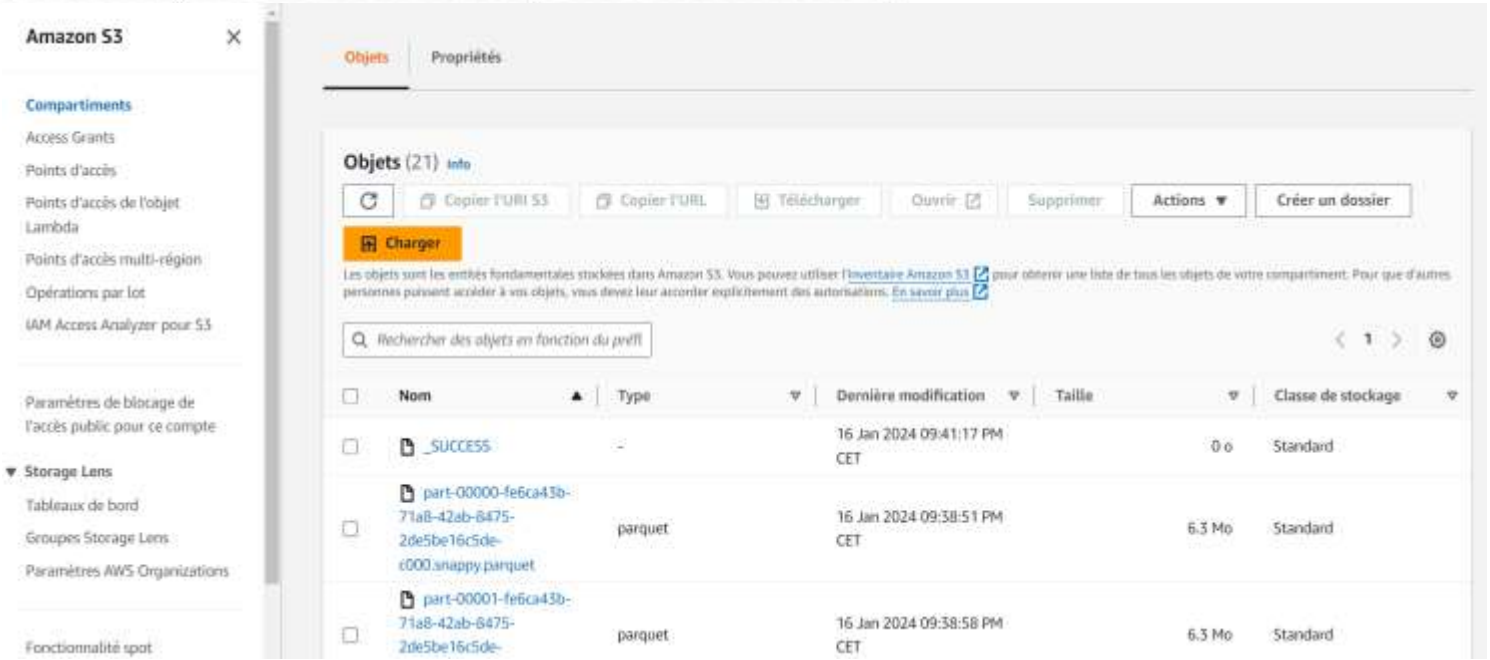
7-2- Résultats finaux:

```
32]: data_final.show(5, True)
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…

+--------------------+-------------+--------------------+--------------------+--------------------+
|                path|        label|         cnn_vectors|         pca_vectors|      features_array|
+--------------------+-------------+--------------------+--------------------+--------------------+
|s3://data-p8/Test...|Pineapple Mini|[0.0,3.6077606678...|[-6.0440096337977...|[-6.0440097, 2.89...|
|s3://data-p8/Test...|    Watermelon|[0.16857005655765...|[0.32359665142388...|[0.32359666, 4.18...|
|s3://data-p8/Test...|Pineapple Mini|[0.0,4.9579777717...|[-5.7261558602160...|[-5.7261558, 3.45...|
|s3://data-p8/Test...|Pineapple Mini|[0.0,4.8031039237...|[-4.9360270642147...|[-4.936027, 2.071...|
|s3://data-p8/Test...|    Watermelon|[0.02986907027661...|[-3.0326533321854...|[-3.0326533, 3.54...|
+--------------------+-------------+--------------------+--------------------+--------------------+
only showing top 5 rows
```

7-3- Données sur S3:



7-5-Démonstration sur le cloud aws

Démonstration sur le cloud aws

Conclusion:

- Prise en main de Pyspark et découverte de plusieurs approche comme format paquet.

- Nombreuses possibilités techniques , choix d’instances, et choix de version adapté a EMR.

- AWS (IAM + S3 + EMR) nous propose un service big data complet.

- SparkUI nous permet de visualiser les détails des calculs distribués