



Openclassrooms projet 8 : Déployez un modèle dans le cloud

Dabidin Keshika

26/08/2023





Sommaire

- 1, Introduction
- 2, Présentation du jeu de données
- 3, Présentation de la Big Data
- 4, Processus de création de l'environnement Big Data
- 5, Traitement des fichiers images
- 6, Validation des résultats
- 7, Conclusion



Introduction

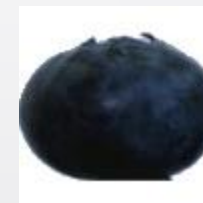
- Start-up « Fruits » veut mettre à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.
- Le développement de cette application permettra de construire une première version de l'architecture Big Data nécessaire.
- Objectifs :
 1. Déployer le traitement des données dans un environnement Big Data.
 2. Développer les scripts en pyspark pour effectuer du calcul distribué.





Présentation du jeu de données

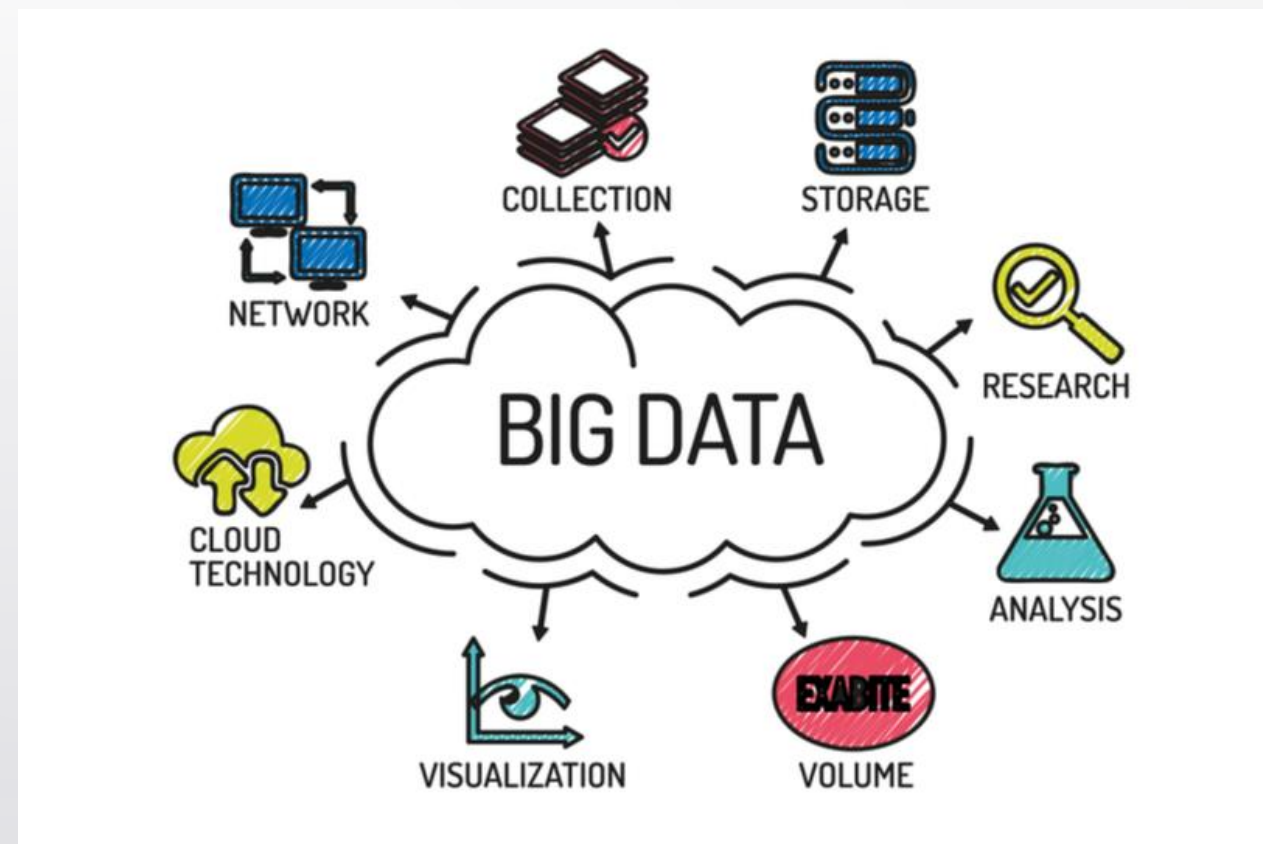
- Le dossier original contient plus de 90 000 images d'échantillons de fruits de diverses catégories.
- Il est demandé d'utiliser le fichier nommé Test contenant 22688 échantillons de fruits.
- Notebook alternant proposé : première version pour le Big Data environment.





L'environnement Big Data

- Gestion et analyse de grands ensembles de données complexes.
- Nécessite des technologies et des compétences spécifiques.





L'environnement Big Data

- Le Volume des données générées nécessite de repenser la manière dont elles sont stockées.
- La Vélocité à laquelle nous parvenons ces données implique de mettre en place des solutions de traitement en temps réel qui ne paralysent pas le reste de l'application.
- Les données se présentent sous une grande Variété de formats : ces données peuvent être structurées (documents JSON), semi-structurées (fichiers de log) ou non structurées (textes, images). L'ingestion, l'analyse et la rétention de ces données prendront des formes différentes selon leur nature, ce qui implique de mettre en place des outils appropriés.



L'environnement Big Data

- Stratégie employée dans le cadre du projet :



**Choix de la
technologie**
Apache Spark



**Choix du
prestataire
cloud**
Amazon Web
Services (AWS)



**Choix de la
solution
technique**
le service EMR



**Choix de la
solution de
stockage**
Amazon S3

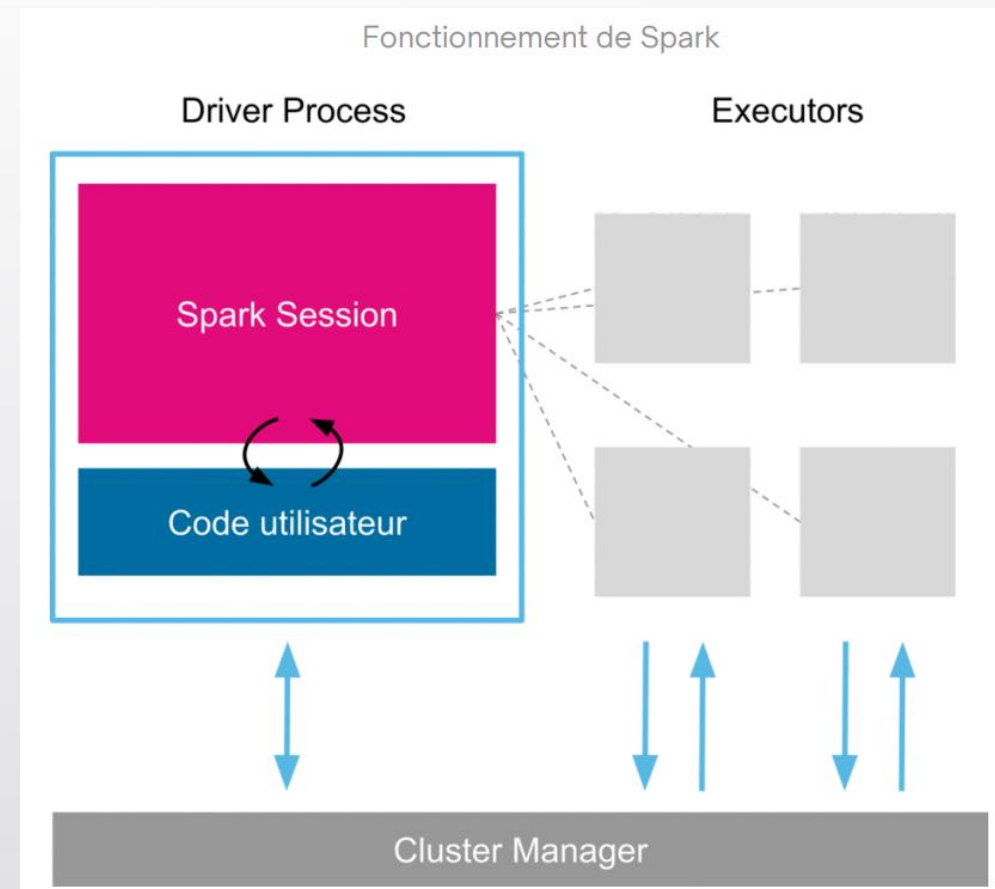


**Configuration de
l'environnement
de travail**
Marketing Executive



Apache Spark

- Un framework open source de calcul distribué pour le traitement et l'analyse de données massives.
- Le framework étant écrit en Scala, PySpark est l'implémentation de Spark pour Python contenant les différents composants de Spark.





S3 : Stockage des données

- Avantages :

1. Solution peu coûteuse dans le temps.
2. Transferts avec les serveurs intéressants.

Les étapes sont :

1. Création clef secrète (iAM)
2. Awscli configuration
3. Upload des fichiers sur le « bucket »

| <input type="checkbox"/> | Name | Type |
|--------------------------|----------------------|--------|
| <input type="checkbox"/> | Apple Braeburn/ | Folder |
| <input type="checkbox"/> | Apple Crimson Snow/ | Folder |
| <input type="checkbox"/> | Apple Golden 1/ | Folder |
| <input type="checkbox"/> | Apple Golden 2/ | Folder |
| <input type="checkbox"/> | Apple Golden 3/ | Folder |
| <input type="checkbox"/> | Apple Granny Smith/ | Folder |
| <input type="checkbox"/> | Apple Pink Lady/ | Folder |
| <input type="checkbox"/> | Apple Red 1/ | Folder |
| <input type="checkbox"/> | Apple Red 2/ | Folder |
| <input type="checkbox"/> | Apple Red 3/ | Folder |
| <input type="checkbox"/> | Apple Red Delicious/ | Folder |



EC2 : Création clef publique/privée et lien SSH

Successfully created key pair

Key pairs (1) [Info](#)

[Refresh](#) [Actions](#) [Create key pair](#)

< 1 > [Settings](#)

| <input type="checkbox"/> | Name | Type | Created | Fingerprint | ID |
|--------------------------|---------|------|------------------------|--|-----------------------|
| <input type="checkbox"/> | p8-data | rsa | 2023/08/25 16:33 GMT+2 | c0:58:a3:71:11:3a:e1:cb:94:8a:46:aa:80:... | key-009c034cfea1748f9 |

g-023dc3adbe734e78c - ElasticMapReduce-master

[Details](#) [Inbound rules](#) [Outbound rules](#) [Tags](#)

Inbound rules (9)



[Manage tags](#)

[Edit inbound rules](#)

< 1 > [Settings](#)

| <input type="checkbox"/> | Name | Security group rule... | IP version | Type | Protocol | Port range | Source | Description |
|--------------------------|------|------------------------|------------|-----------------|----------|------------|-----------------------------|-------------|
| <input type="checkbox"/> | - | sgr-0b703bd7ad6a29... | IPv4 | SSH | TCP | 22 | 0.0.0.0/0 | - |
| <input type="checkbox"/> | - | sgr-0c246a500da9860... | - | All TCP | TCP | 0 - 65535 | sg-023dc3adbe734e7... | - |
| <input type="checkbox"/> | - | sgr-06b281f05fd1e0f38 | - | All TCP | TCP | 0 - 65535 | sg-08e24eb4025845f... | - |
| <input type="checkbox"/> | - | sgr-015278625baa2af3f | - | Custom TCP | TCP | 8443 | pl-a5a742cc | - |
| <input type="checkbox"/> | - | sgr-04532ce271c6e1db2 | - | All UDP | UDP | 0 - 65535 | sg-023dc3adbe734e7... | - |
| <input type="checkbox"/> | - | sgr-0c8418ae5834dd8... | IPv6 | SSH | TCP | 22 | ::/0 | - |
| <input type="checkbox"/> | - | sgr-02fa9bdd2010803... | - | All UDP | UDP | 0 - 65535 | sg-08e24eb4025845f... | - |
| <input type="checkbox"/> | - | sgr-0d68d6c58de2a55... | - | All ICMP - IPv4 | ICMP | All | sg-023dc3adbe734e7... | - |
| <input type="checkbox"/> | - | sgr-091f37a09fce25e8f | - | All ICMP - IPv4 | ICMP | All | sg-08e24eb4025845f... | - |



EMR (Serveur de calculs distribués)

AWS EMR (Elastic MapReduce) est un service cloud qui facilite la configuration, la gestion et l'exécution de clusters Big Data. Il offre une solution évolutive et flexible en utilisant des frameworks populaires tels que Hadoop et Spark.

Cluster: P8_fruits **Starting**

Summary Application user interfaces Monitoring Hardware

Summary

ID: j-2B5V0EYL9Q4ZU
Creation date: 2023-08-26 00:57 (UTC+2)
Elapsed time: 0 seconds
After last step completes: Cluster waits
Termination protection: Off [Change](#)
Tags: -- [View All / Edit](#)
Master public DNS: --

Network and hardware

Availability zone: eu-west-1b
Subnet ID: [subnet-02aa41e659fc9ea94](#) [🔗](#)
Master: **Provisioning** 1 m5.xlarge
Core: **Provisioning** 2 m5.xlarge
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 hour



EMR sécurité SSH (Serveur de calculs distribués)

- Nous souhaitons maintenant pouvoir accéder à nos applications :

1, JupyterHub pour l'exécution de notre notebook.

2, Serveur d'historique Spark pour le suivi de l'exécution des tâches de notre script lorsqu'il sera lancé.

- Création du tunnel ssh vers le Driver
- Configuration de FoxyProxy

Enable an SSH Connection

EMR applications publish user interfaces as web sites hosted on the master node. For security reasons, these web sites are only available on the master node's local web server.

To reach the web interfaces, you must establish an SSH tunnel with the master node using either dynamic or local port forwarding. If you use dynamic port forwarding, you must also configure a proxy server to view the web interfaces.

Step 1: Open an SSH Tunnel to the Amazon EMR Master Node - [Learn more](#)

Windows

Mac / Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish an SSH tunnel with the master node using dynamic port forwarding, type the following command. Replace ~/p8-data.pem with the location and filename of the private key file (.pem) used to launch the cluster.

```
ssh -i ~/p8-data.pem -ND 8157 hadoop@ec2-34-245-139-97.eu-west-1.compute.amazonaws.com
```

Note: Port 8157 used in the command is a randomly selected, unused local port.

- 3. Type yes to dismiss the security warning.

Step 2: Configure a proxy management tool - [Learn more](#)

Chrome

Firefox

1. Go to <https://addons.mozilla.org/>, search for **FoxyProxy Standard**, and follow the instructions to add FoxyProxy to Firefox.
2. Using a text editor, create a JSON file named foxyproxy-settings.json from the following example configuration. If you specified a different port number when you set up your SSH tunnel, replace 8157 with your port number.

```
{
  "k20d21508277536715": {
    "active": true,
    "address": "localhost",
    "port": 8157,
    "username": "",
    "password": "",
    "type": 3,
    "proxyDNS": true,
    "title": "emr-socks-proxy",
    "color": "#0055E5",
    "index": 9007199254740991,
    "whitePatterns": [
      {
        "title": "*ec2*.amazonaws.com*",
        "active": true,
        "pattern": "*ec2*.amazonaws.com*",
        "importedPattern": "*ec2*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      }
    ]
  }
}
```

Close



Connexion au Jupyter Hub

- On se connecte avec les informations par défaut :

login: jovyan

password: jupyter

- Import d'un notebook déjà rédigé en local directement sur S3 et ouverture depuis l'interface JupyterHub.

The screenshot shows the JupyterHub login page. At the top, there is a header with the JupyterHub logo. Below the header, there is a 'Sign in' form. The form has an orange header bar with the text 'Sign in'. Inside the form, there are two input fields: 'Username:' with the value 'jovyan' and 'Password:' with masked characters (dots). Below the password field is an orange 'Sign in' button.



Le traitement des images

- 1, L'importation des images : associer leur label, les redimensionner...
- 2, Le modèle: MobileNetV2 : creer un nouveau modèle dépourvu de la dernière couche.
- 3, L'extraction de features : Pandas UDF: featuriser avec pd.Series, prétraiter une image.
- 4, Réduction de dimension : Conversion en vecteur dense, Standardisation, PCA.
- 5, Sauvegarde du résultat : Test de fonctionnement.



L'importation des images

```

1 # Chargement des données
2 images = spark.read.format("binaryFile") \
3     .option("pathGlobFilter", "*.jpg") \
4     .option("recursiveFileLookup", "true") \
5     .load(PATH_Data)

```

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

```
1 images.show(5)
```

```

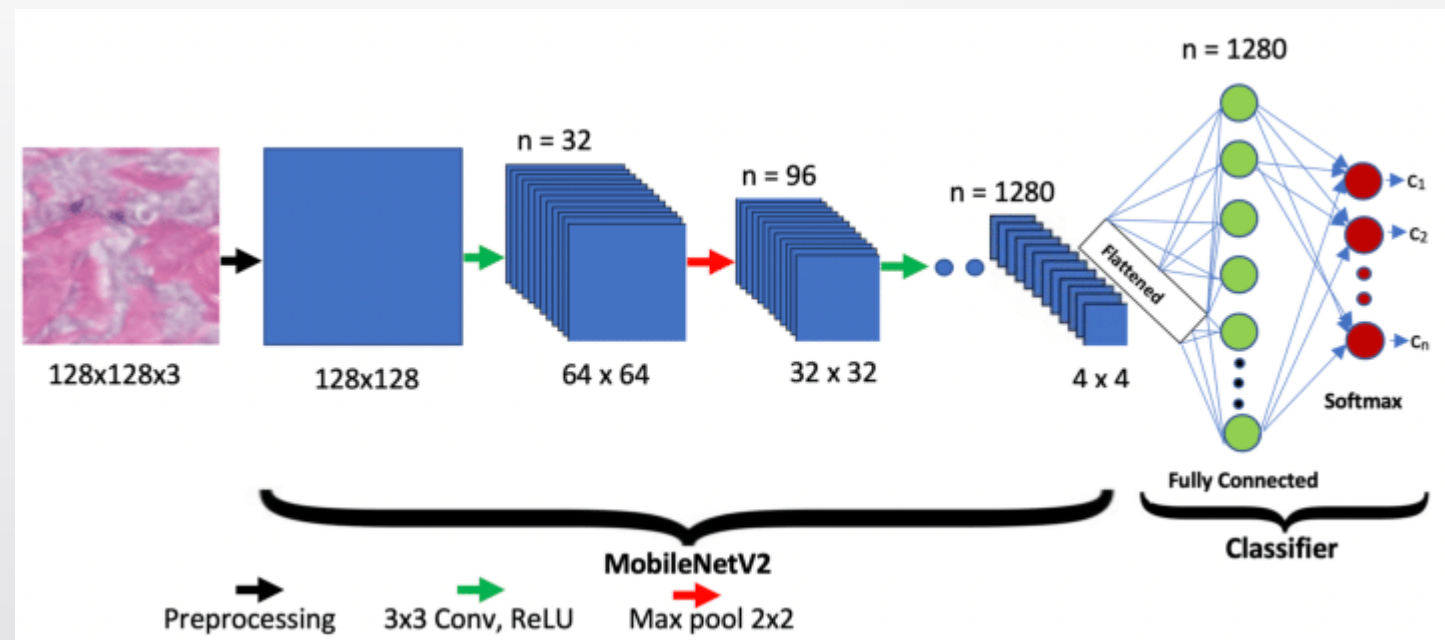
+-----+-----+-----+-----+
|           path|modificationTime|length|           content|
+-----+-----+-----+-----+
|file:/home/keshik...|2023-08-24 15:31:...| 7353|[FF D8 FF E0 00 1...|
|file:/home/keshik...|2023-08-24 15:31:...| 7350|[FF D8 FF E0 00 1...|
|file:/home/keshik...|2023-08-24 15:31:...| 7349|[FF D8 FF E0 00 1...|
|file:/home/keshik...|2023-08-24 15:31:...| 7348|[FF D8 FF E0 00 1...|
|file:/home/keshik...|2023-08-24 15:32:...| 7328|[FF D8 FF E0 00 1...|
+-----+-----+-----+-----+
only showing top 5 rows

```



Transfert learning : MobileNetV2

MobileNetV2 est un modèle de réseau de neurones convolutifs (CNN) qui a été développé par Google. Il est spécialement conçu pour être utilisé sur des appareils mobiles et des applications à ressources limitées en termes de puissance de calcul et de mémoire





Extraction de features et réduction de dimensions

```
1 features_df.show()
```

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

| path | label | features |
|----------------------|----------------|----------------------|
| s3://p8-ocr/Test/... | Watermelon | [0.6506585, 0.230... |
| s3://p8-ocr/Test/... | Watermelon | [0.08841578, 0.83... |
| s3://p8-ocr/Test/... | Watermelon | [0.13241422, 0.22... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.002079701, 4.6... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.0, 4.49807, 0.... |
| s3://p8-ocr/Test/... | Watermelon | [0.0, 0.91131, 0.... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.0, 4.583824, 0... |
| s3://p8-ocr/Test/... | Watermelon | [0.13633335, 0.20... |
| s3://p8-ocr/Test/... | Watermelon | [0.0, 0.22407952,... |
| s3://p8-ocr/Test/... | Watermelon | [0.23570964, 0.15... |
| s3://p8-ocr/Test/... | Raspberry | [0.14059144, 0.45... |
| s3://p8-ocr/Test/... | Raspberry | [0.40123066, 0.05... |
| s3://p8-ocr/Test/... | Cauliflower | [0.0, 0.32475963,... |
| s3://p8-ocr/Test/... | Raspberry | [0.028396703, 0.2... |
| s3://p8-ocr/Test/... | Cauliflower | [0.0, 1.6184936, ... |
| s3://p8-ocr/Test/... | Cauliflower | [0.0, 0.9022645, ... |
| s3://p8-ocr/Test/... | Raspberry | [0.062141612, 0.1... |
| s3://p8-ocr/Test/... | Cauliflower | [0.0, 0.70285535,... |
| s3://p8-ocr/Test/... | Pineapple | [0.0, 2.3203015, ... |
| s3://p8-ocr/Test/... | Raspberry | [0.14664671, 0.24... |

only showing top 20 rows





Extraction de features et réduction de dimensions

```

1 # Réduction de dimension PCA
2 # Entraînement de l'algorithme
3 pca = PCA(k=nombre_cp, inputCol='features_scaled', outputCol='vectors_pca')
4 action_pca = pca.fit(df_preprocess)
5

```

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

| path | label | features | features_vectors | features_scaled | vectors_pca |
|----------------------|----------------|----------------------|----------------------|----------------------|----------------------|
| s3://p8-ocr/Test/... | Watermelon | [0.6506585, 0.230... | [0.65065848827362... | [0.44808956363776... | [-17.281090895514... |
| s3://p8-ocr/Test/... | Watermelon | [0.08841578, 0.83... | [0.08841577917337... | [-0.5936218928654... | [-14.425315457069... |
| s3://p8-ocr/Test/... | Watermelon | [0.13241422, 0.22... | [0.13241422176361... | [-0.5121025045787... | [-11.327575276781... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.002079701, 4.6... | [0.00207970105111... | [-0.7535835610806... | [-13.450613627224... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.0, 4.49807, 0... | [0.0,4.4980697631... | [-0.7574367874115... | [-8.4788524004423... |
| s3://p8-ocr/Test/... | Watermelon | [0.0, 0.91131, 0... | [0.0,0.9113100171... | [-0.7574367874115... | [-10.586632943196... |
| s3://p8-ocr/Test/... | Pineapple Mini | [0.0, 4.583824, 0... | [0.0,4.5838241577... | [-0.7574367874115... | [-12.052415231517... |
| s3://p8-ocr/Test/... | Watermelon | [0.13633335, 0.20... | [0.13633334636688... | [-0.5048412330093... | [-7.3239574315667... |
| s3://p8-ocr/Test/... | Watermelon | [0.0, 0.22407952,... | [0.0,0.2240795195... | [-0.7574367874115... | [-7.8230243681277... |
| s3://p8-ocr/Test/... | Watermelon | [0.23570964, 0.15... | [0.23570963740348... | [-0.3207189327961... | [-7.7782315697715... |
| s3://p8-ocr/Test/... | Banana | [0.14050144, 0.45... | [0.14050144258400... | [0.4060510210671... | [6.5022866742520... |



Validation du Résultat

```
1 #Chargement des données enregistrées et validation du résultat
2 df = pd.read_parquet(PATH_Result, engine='pyarrow')
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=)
```

```
1 df.head()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=)
```

| | path | ... |
|---|---|----------------------------|
| 0 | s3://p8-ocr/Test/Watermelon/r_72_100.jpg | ... {'type': 1, 'size': No |
| 1 | s3://p8-ocr/Test/Watermelon/r_109_100.jpg | ... {'type': 1, 'size': No |
| 2 | s3://p8-ocr/Test/Watermelon/r_105_100.jpg | ... {'type': 1, 'size': No |
| 3 | s3://p8-ocr/Test/Pineapple Mini/140_100.jpg | ... {'type': 1, 'size': No |
| 4 | s3://p8-ocr/Test/Pineapple Mini/130_100.jpg | ... {'type': 1, 'size': No |

Objets (25)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[inventaire Amazon S3](#) pour obtenir une liste de vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)



Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Rechercher des objets en fonction du préfixe

Afficher les versions

| <input type="checkbox"/> | Nom | Type |
|--------------------------|---|---------|
| <input type="checkbox"/> | _SUCCESS | - |
| <input type="checkbox"/> | part-00000-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00001-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00002-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00003-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00004-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00005-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00006-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00007-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |
| <input type="checkbox"/> | part-00008-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet | parquet |



Conclusion

- Découverte de l'environnement de Big Data
- Création d'un réel cluster de calculs pour répondre à l'objectif qui était de pouvoir anticiper une future augmentation de la charge de travail.
- Le meilleur choix retenu a été l'utilisation de AWS (Amazon Web Services). Nous avons utilisé les principaux services comme: EC2 pour l'hébergement de machines virtuelles, S3 pour le stockage d'objets.
- Maintenance à prévoir par la suite.



Merci pour votre
attention.