Openclassrooms projet 8 :

Déployez un modèle dans le cloud

Dabidin Keshika

26/08/2023

Sommaire

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 parviennent ces données implique de mettre en place des solutions de traitement en temps réel qui ne paralysent pas le reste de l'application.

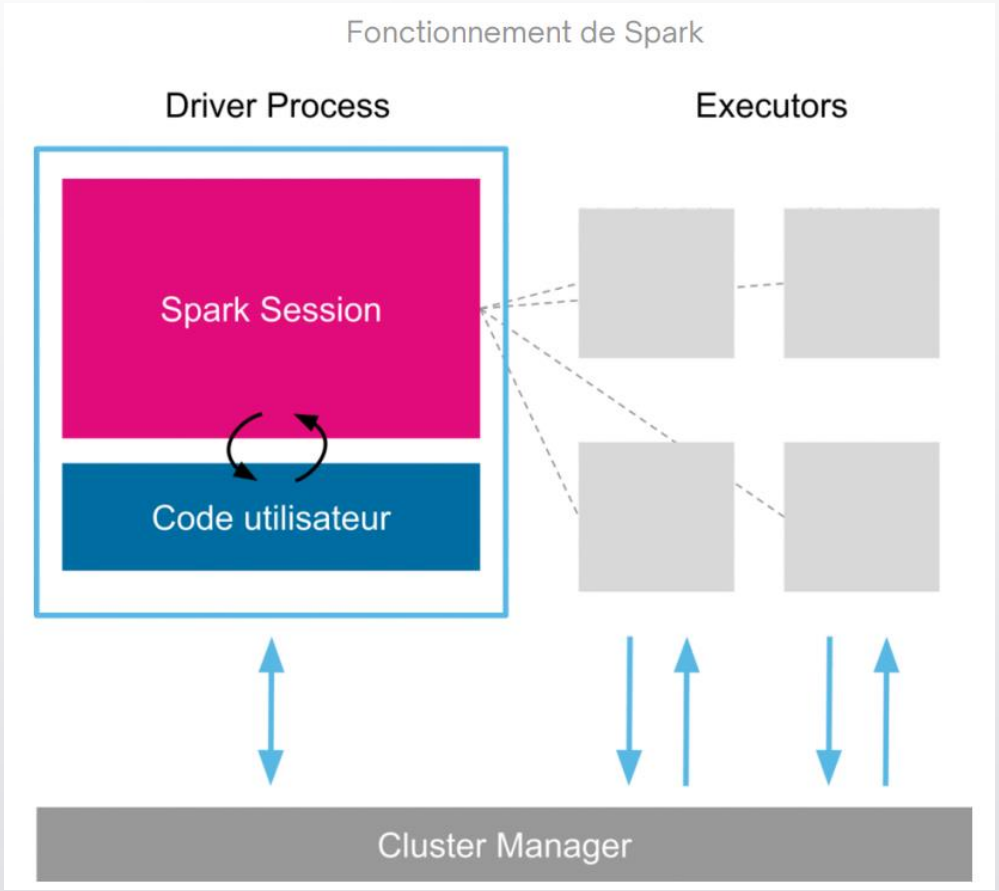
- Les données se présentent sous une grande Variété de formats : ces données peuvent être structurées (documents JSON), semi-structurées (fichiers de log) ou non structurées (textes, images). L'ingestion, l'analyse et la rétention de ces données prendront des formes différentes selon leur nature, ce qui implique de mettre en place des outils appropriés.

L'environnement Big Data

- Stratégie employée dans le cadre du projet :

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

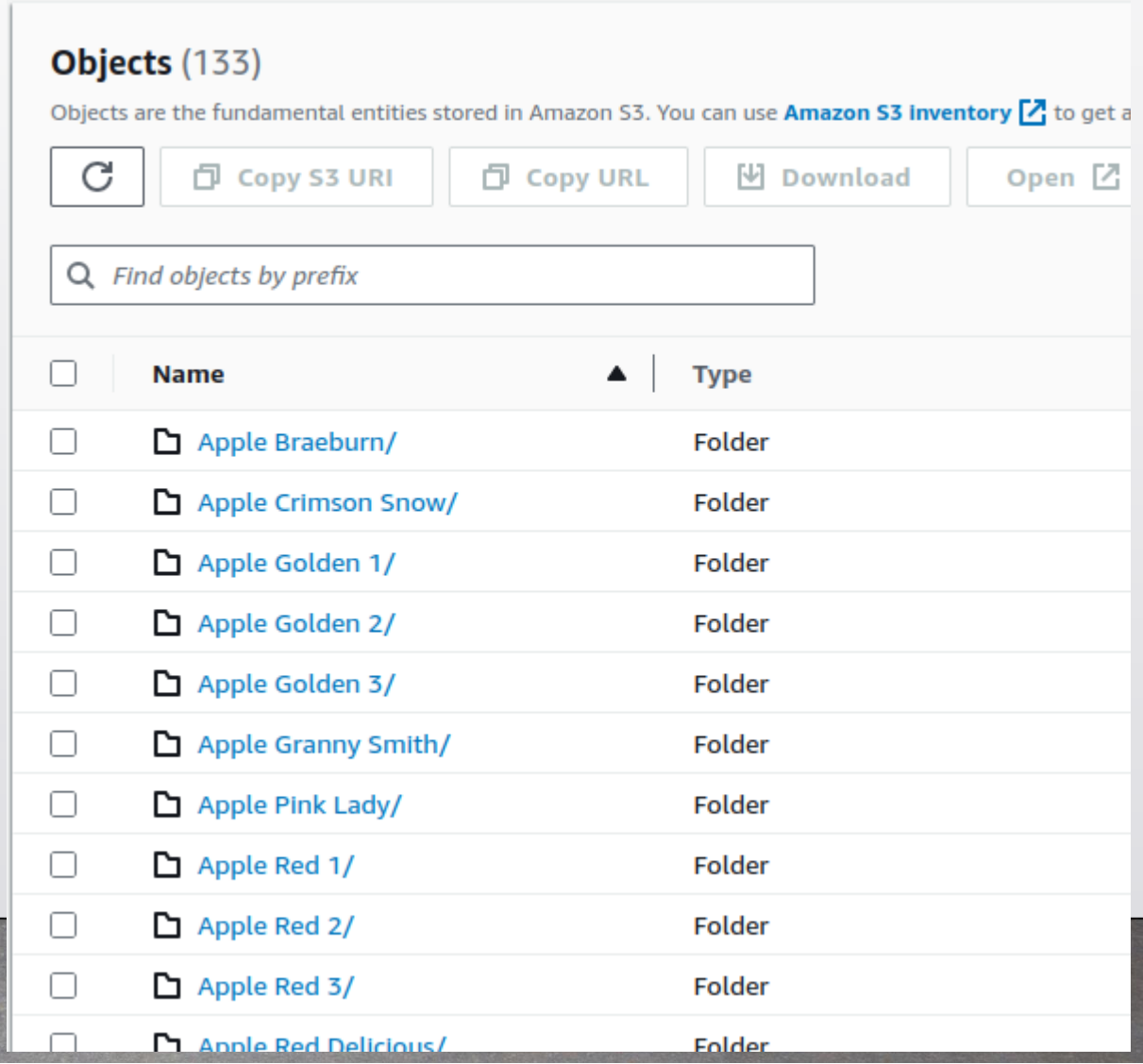
- Avantages :

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 »



EC2 : Création clef publique/privée et lien SSH

Successfully created key pair

Key pairs (1) Info

Search

Actions | Create key pair

Name	Type	Created	Fingerprint	ID
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)

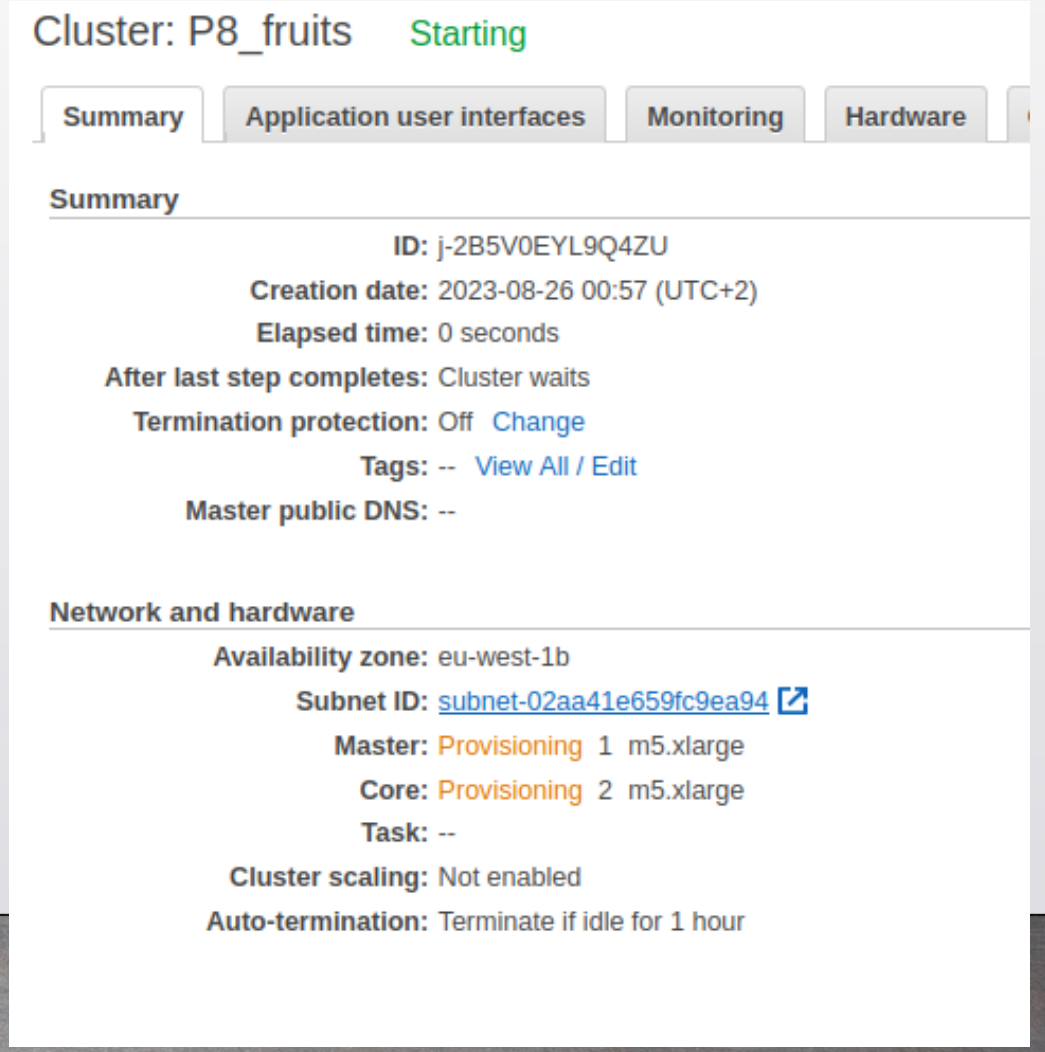
Filter security group rules

Manage tags | Edit inbound rules

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



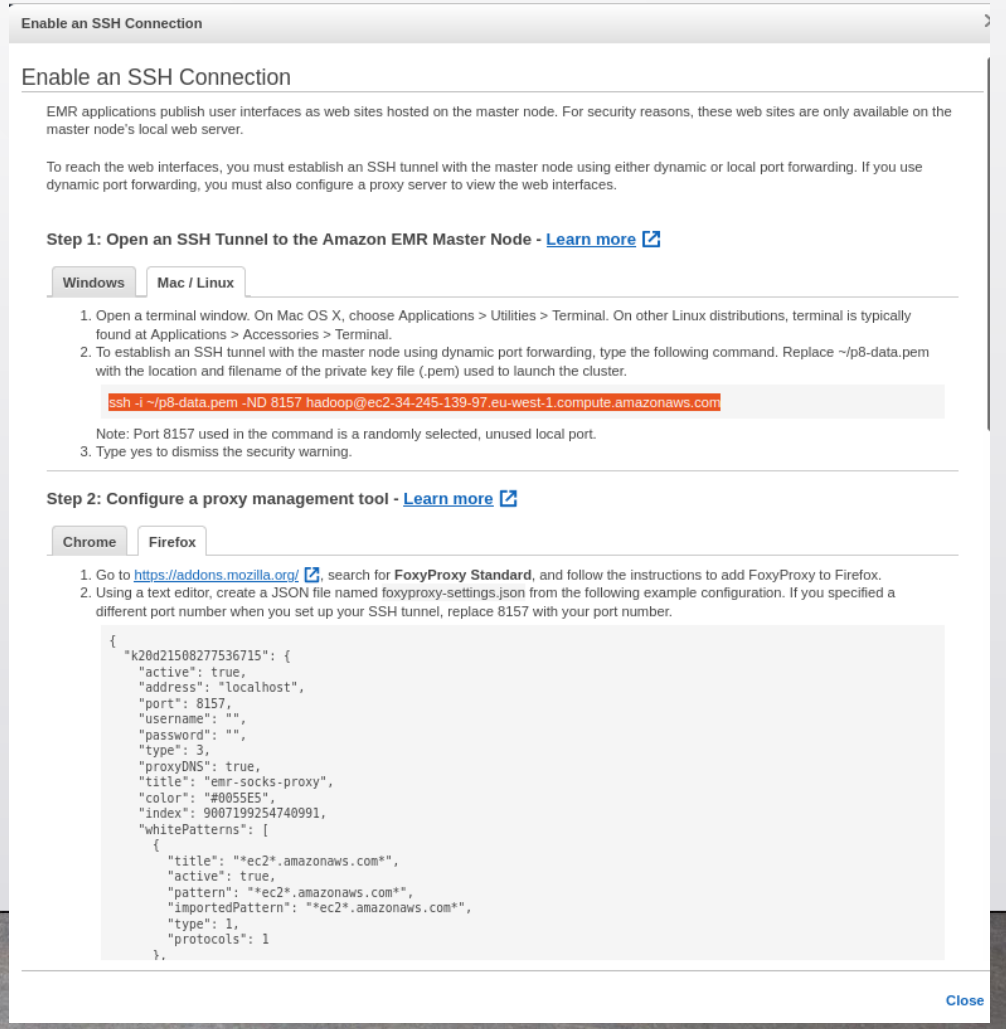
EMR sécurité SSH (Serveur de calculs distribués)

- 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



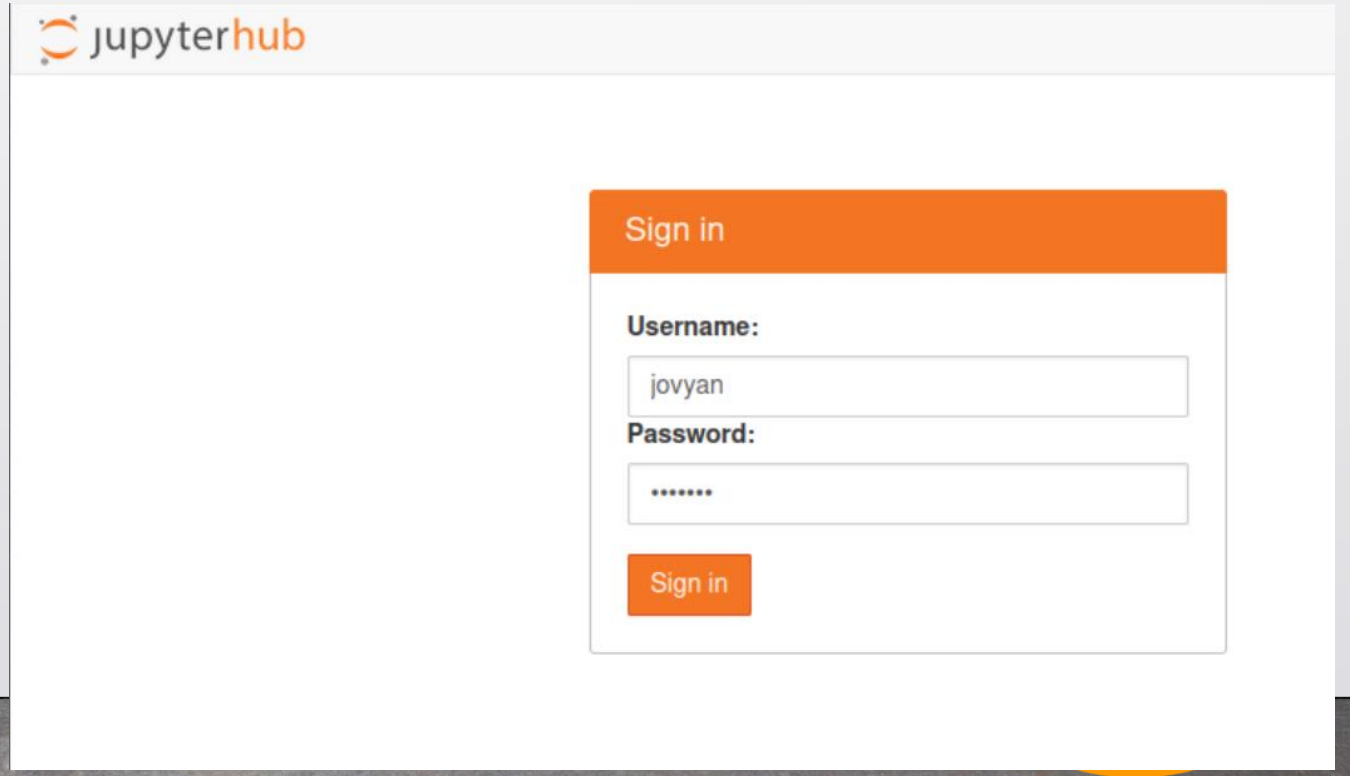
Connexion au Jupyter Hub

- 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.



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 dimmension : Conversion en vecteur dense, Standardisation, PCA.

5, Sauvegarde du résultat : Test de fonctionnement.

L'importation des images

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

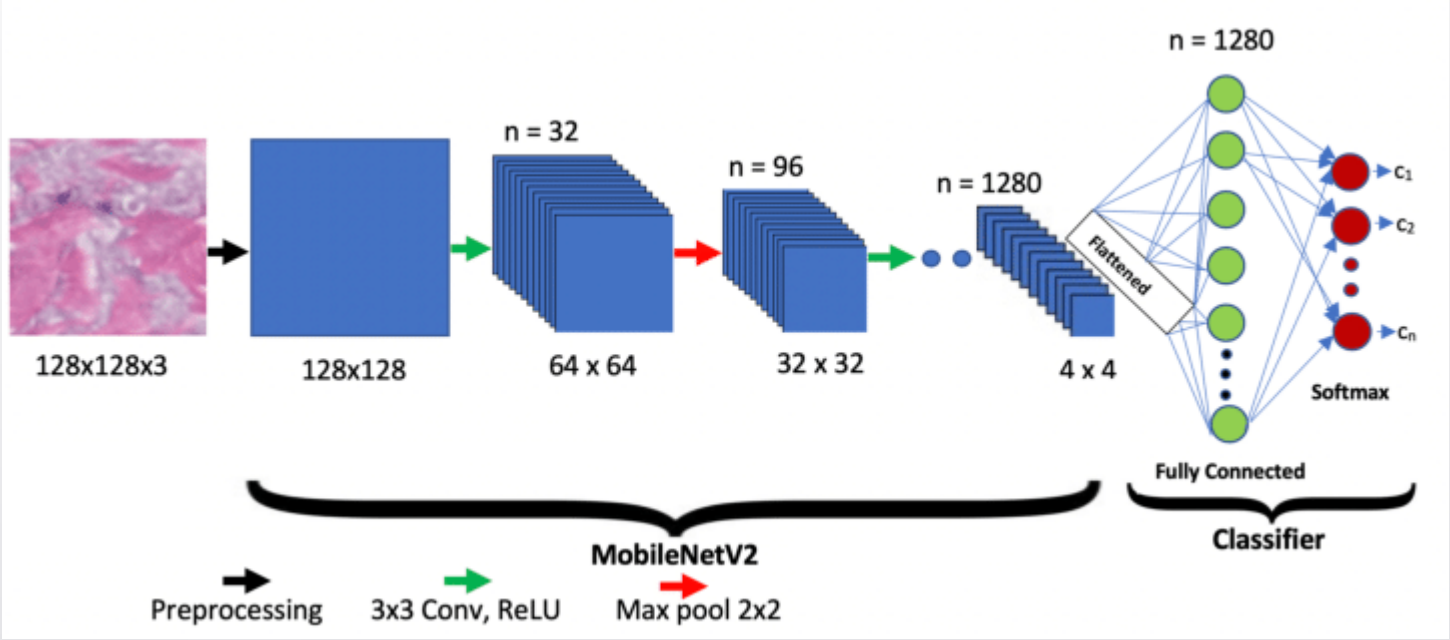
```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…
```

```
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

```
# Réduction de dimension PCA
# Entrainement de l'algorithme
pca = PCA(k=nombre_cp, inputCol='features_scaled', outputCol='vectors_pca')
action_pca = pca.fit(df_preprocess)

```

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/...	Raspberry	[0.14050144, 0.45...	[0.14050144258409...	[-0.4969510319671...	[-6.5923866743530...

Validation du Résultat

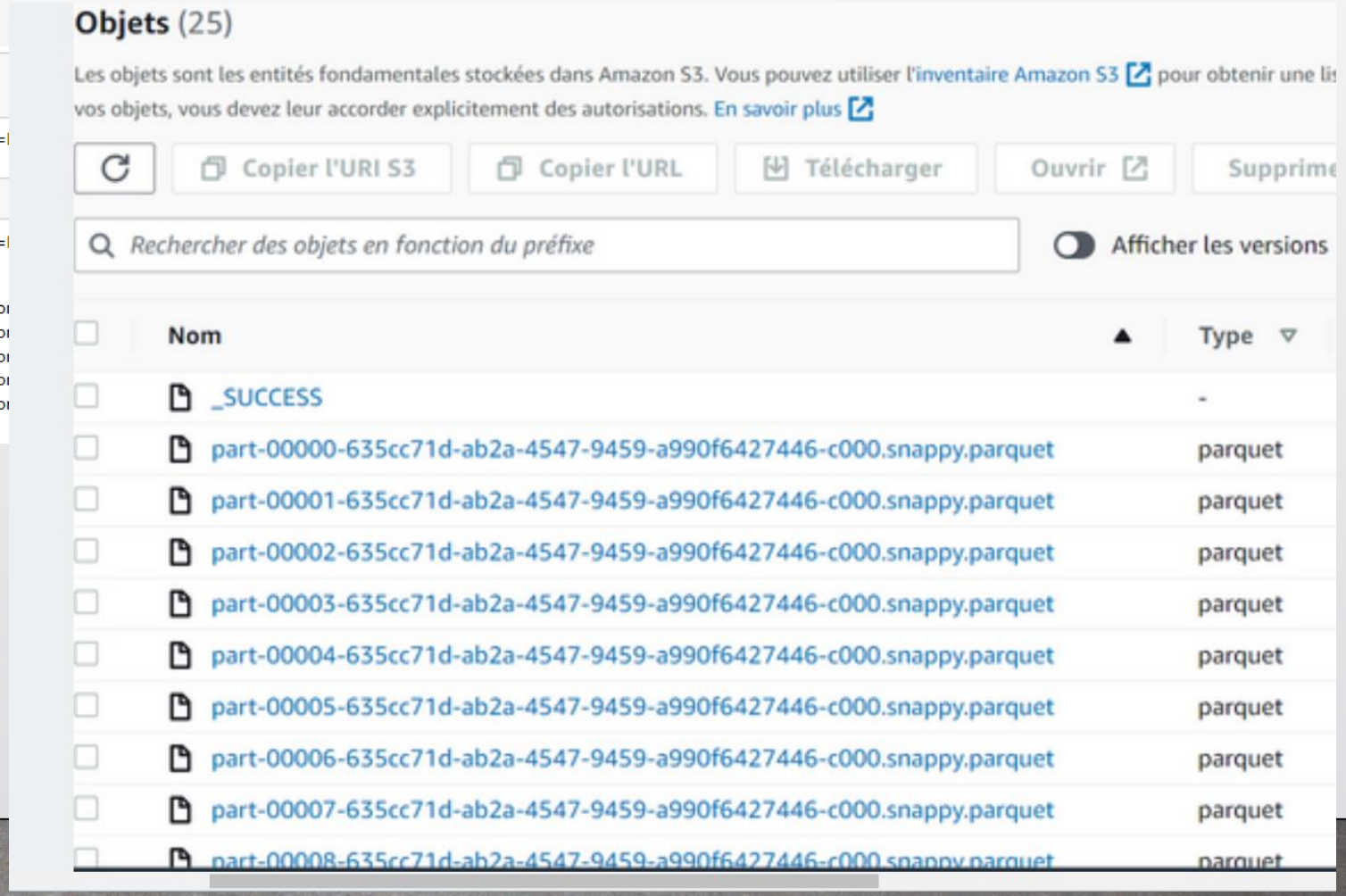
```
1  #Chargement des données enregistrées et validation du résultat
2  df = pd.read_parquet(PATH_Result, engine='pyarrow')
```

```
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



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.