

Déployez un modèle dans le cloud

Projet 8

Sommaire

- La problématique et le jeu de données
- Le processus de création de l'environnement Big Data
- La chaîne de traitement des images
- La synthèse et la conclusion

Rappel de la problématique

La start-up 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.



Le Big Data

le Big Data fait référence à la gestion et à l'analyse de grands ensembles de données complexes. Il offre des opportunités pour découvrir des insights précieux, mais nécessite des technologies et des compétences spécifiques pour en tirer pleinement parti



Les données

Le jeu de données "Fruits 360"

Il contient des images de différents types de fruits capturées sous différents angles et conditions d'éclairage. Il contient plus de 90 000 images d'échantillons de fruits.

Le notebook réalisé par l'alternant

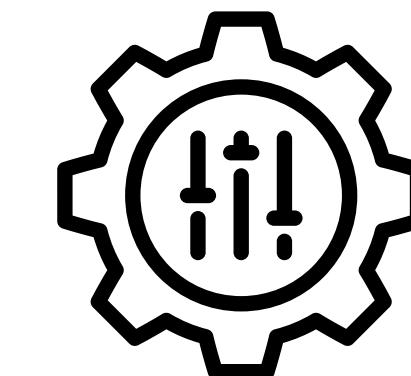
une première approche dans un environnement Big Data AWS EMR

Objectifs

Déployer le traitement des données dans un environnement Big Data
Développer les scripts en pyspark pour effectuer du calcul distribué



L'environnement Big Data

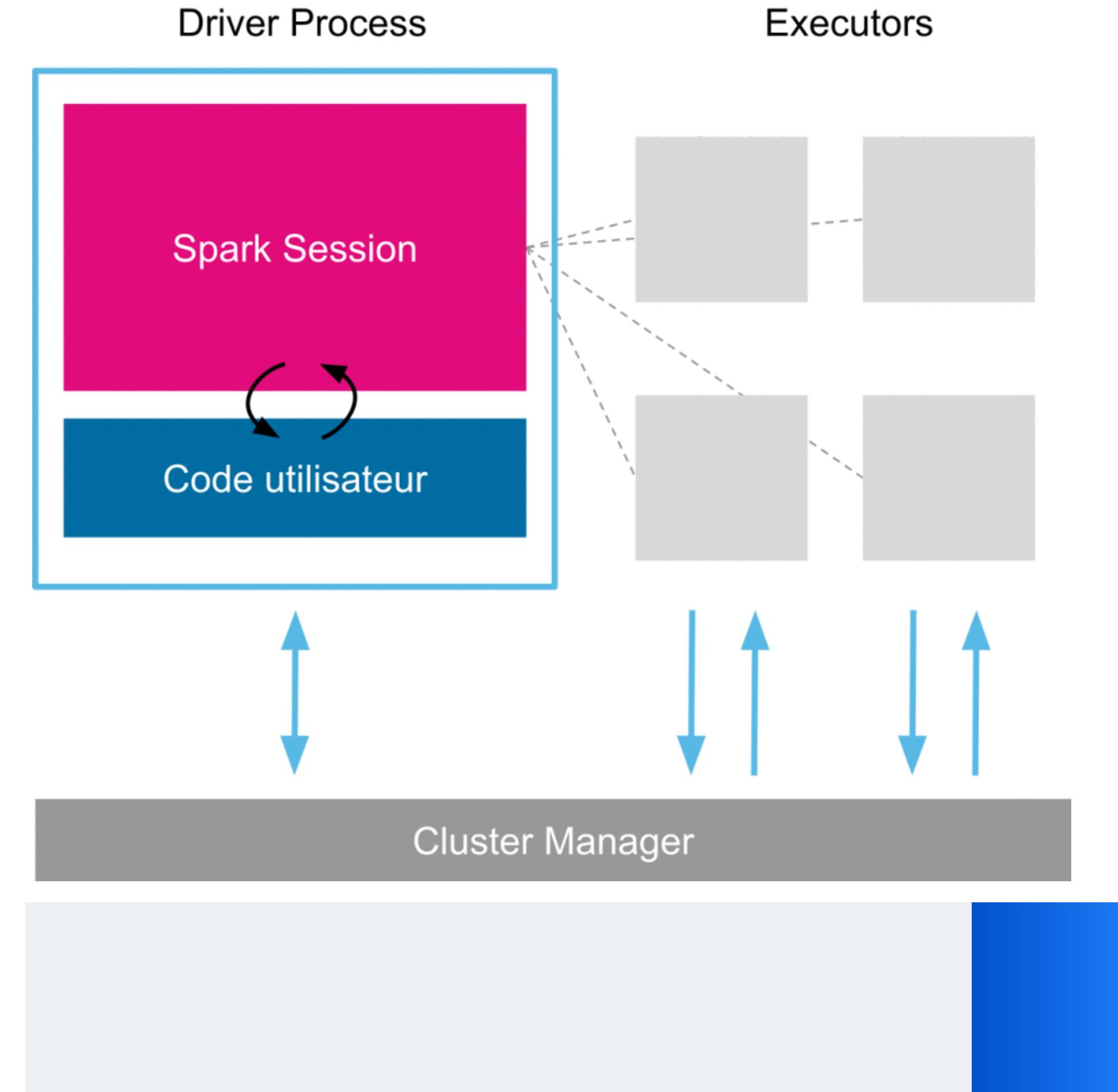


Apache Spark

L'ENVIRONNEMENT BIG DATA

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.

Fonctionnement de Spark



L'ENVIRONNEMENT BIG DATA

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) est un service de calcul évolutif proposé par Amazon Web Services (AWS). Il permet aux utilisateurs de créer et de gérer des instances de serveur virtuel dans le cloud.

Création d'une paire de clés

Créer une paire de clés [Informations](#)

Paire de clés

Une paire de clés, composée d'une clé privée et d'une clé publique, est un ensemble d'informations d'identification de sécurité que vous utilisez pour prouver votre identité lors de la connexion à une instance.

Nom

Saisir le nom de la paire de clés

Le nom peut avoir un maximum de 255 caractères ASCII. Il ne peut pas inclure d'espaces avant ou après.

Type de paire de clés [Informations](#)

RSA ED25519

Format de fichier de clé privée

.pem
À utiliser avec OpenSSH

.ppk
À utiliser avec PuTTY

Balises - facultatif

Aucune balise n'est associée à cette ressource.

Ajouter une balise

Vous pouvez ajouter jusqu'à 50 identifications supplémentaires.

Annuler Créer une paire de clés

L'ENVIRONNEMENT BIG DATA

Le serveur EMR

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.

Création de cluster

The screenshot shows the AWS EMR console with the following details:

Informations sur le cluster	Applications
ID de cluster j-33M63FE98SKOM	Version d'Amazon EMR emr-6.10.0
Configuration de cluster Groupes d'instances	Applications installées JupyterHub 1.5.0, Spark 3.3.1, TensorFlow 2.11.0
Capacité 1 primaire(s) 2 unité(s) principale(s) 0 tâche(s)	
Gestion des clusters	Statut et heure
Destination des journaux dans Amazon S3 p8-ocr/journal	Statut En attente
Interfaces utilisateur d'application persistantes	Heure de création 12 juin 2023 12:20 (UTC+02:00)

L'ENVIRONNEMENT BIG DATA

Le stockage des données: S3

Amazon Simple Storage Service (S3) offre une solution de stockage sécurisée, évolutive et fiable pour vos besoins de stockage d'objets.

Upload de nos données sur S3

The screenshot shows the AWS Management Console with the S3 service selected. The main heading is 'Instantané de compte'. Below it, there's a section about 'Storage Lens' with a 'Afficher le tableau de bord de Storage Lens' button. The central part of the screen is titled 'Compartiments (3) Info' and contains a table of three buckets:

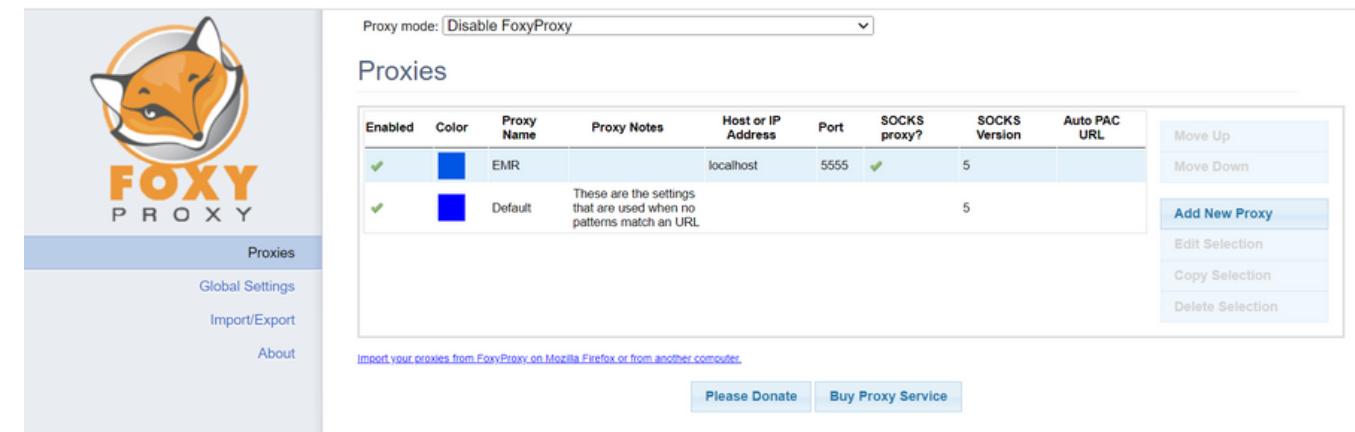
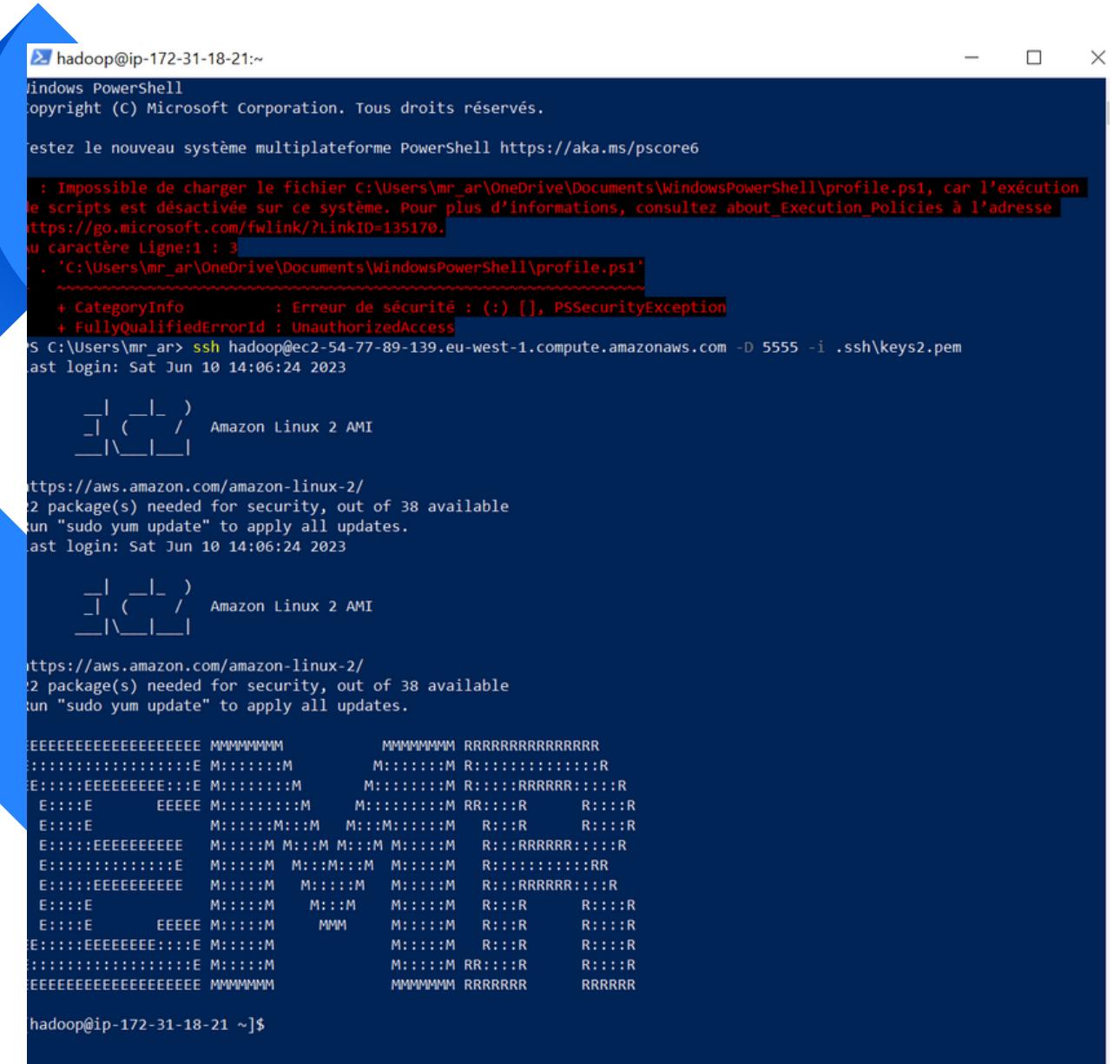
Nom	Région AWS
aws-logs-138256186588-eu-west-1	UE (Irlande) eu-west-1
config-bucket-138256186588	UE (Irlande) eu-west-1
p8-ocr	UE (Irlande) eu-west-1

Below the table are buttons for 'Créer un compartiment' (Create a compartment), 'Copier l'ARN' (Copy ARN), 'Vider' (Empty), and 'Supprimer' (Delete). There's also a search bar and navigation controls.

L'ENVIRONNEMENT BIG DATA

Configuration SSH

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



L'ENVIRONNEMENT BIG
DATA

Connexion au notebook JupyterHub

The diagram illustrates the workflow for connecting a JupyterHub notebook to an Amazon S3 bucket. A large blue arrow points from the JupyterHub interface at the top right down to the Amazon S3 interface at the bottom right. The JupyterHub interface shows a file list with a 'New' button highlighted. The Amazon S3 interface shows a file list for the 'jupyter/' folder, with a 'Copier l'URI S3' (Copy S3 URI) button highlighted.

JupyterHub File List:

Name	Last Modified	File size
e-5OTY4VKPDT21945FF6DN15E35	il y a 52 ans	
notebook.ipynb	il y a 2 jours	
_metadata	il y a 4 jours	

Amazon S3 Object List:

Objets (1)
joxyan/

Actions available for objects:

- Copier l'URI S3
- Copier l'URL
- Télécharger
- Ouvrir
- Supprimer
- Actions
- Créer un dossier
- Charger

Search bar: Rechercher des objets en

Filter: Afficher les versions

Table Headers:

Nom	Type	Dernière modification	Taille	Classe de stockage
-----	------	-----------------------	--------	--------------------

Table Data:

joxyan/	Dossier	-	-	-
---------	---------	---	---	---

La chaîne de traitement des images

L'importation des images

associer leur label, les redimensionner

Le modèle: MobileNetV2

créer un nouveau modèle dépourvu de la dernière couche

L'extraction de features

Pandas UDF: featuriser avec pd.Series, prétraiter une image

Réduction de dimension

Conversion en vecteur dense, Standardisation, PCA

Sauvegarde du résultat

Test du fonctionnement



L'importation des images

In [4]: # Chargement des données

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

In [6]: #Je ne conserve que le path de l'image
#j'ajoute une colonne contenant les labels de chaque image
images = images.withColumn('label', element_at(split(images['path'], '/'),
print(images.printSchema())
print(images.select('path', 'label').show(5, False))

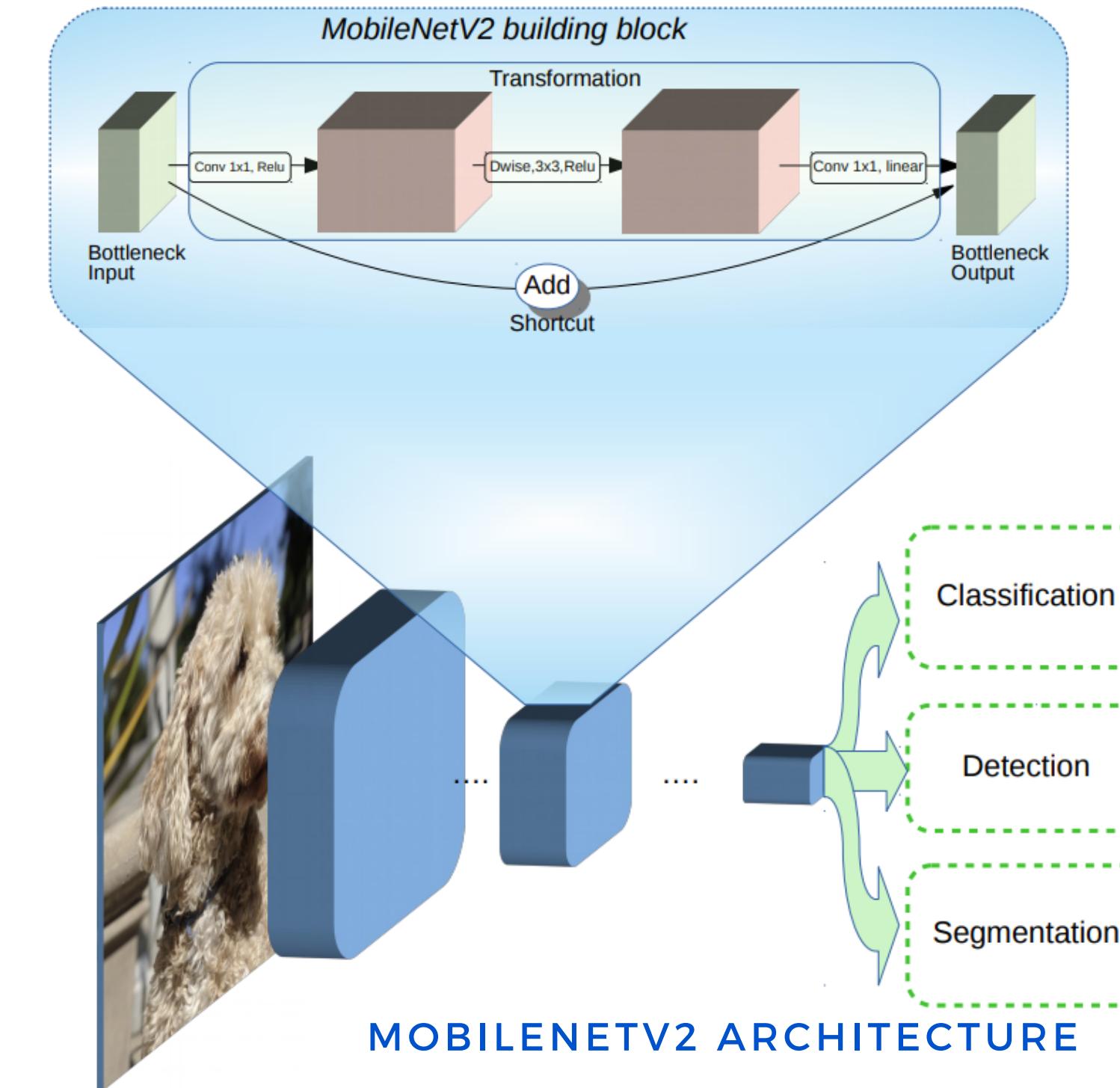
```
root
|-- path: string (nullable = true)
|-- modificationTime: timestamp (nullable = true)
|-- length: long (nullable = true)
|-- content: binary (nullable = true)
|-- label: string (nullable = true)

None
+-----+-----+
|path |label |
+-----+-----+
|s3://p8-ocr/Test/Watermelon/r_106_100.jpg|Watermelon|
|s3://p8-ocr/Test/Watermelon/r_109_100.jpg|Watermelon|
|s3://p8-ocr/Test/Watermelon/r_108_100.jpg|Watermelon|
|s3://p8-ocr/Test/Watermelon/r_107_100.jpg|Watermelon|
|s3://p8-ocr/Test/Watermelon/r_95_100.jpg|Watermelon|
+-----+-----+
only showing top 5 rows
```



Transfert learning: MobileNetV2

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-



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.



EXTRACTIONS DE FEATURES + PCA

```
In [13]: #Exécutions des actions d'extractions de features
features_df = images.repartition(24).select(col("path"),
                                         col("label"),
                                         featurize_udf("content").alias("features"))
```

```
features_df.show()
```

```
+-----+-----+-----+
|       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...
```

```
[20]: # 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)
```

```
[21]: # Transformation des images sur les k premières composantes
df_final = action_pca.transform(df_preprocess)
```

```
[23]: df_final.show()
```

```
+-----+-----+-----+
|       path|    label|      features|      features_
vectors|      features_scaled|      vectors_pca|
+-----+-----+-----+
|s3://p8-ocr/Test/...| Watermelon|[0.6506585, 0.230...|[0.6506584882
7362...|[0.44808956363776...|[-17.281090895514...|
|s3://p8-ocr/Test/...| Watermelon|[0.08841578, 0.83...|[0.0884157791
7337...|[-0.5936218928654...|[-14.425315457069...|
|s3://p8-ocr/Test/...| Watermelon|[0.13241422, 0.22...|[0.1324142217
6361...|[-0.5121025045787...|[-11.327575276781...|
|s3://p8-ocr/Test/...| Pineapple Mini|[0.002079701, 4.6...|[0.0020797010
5111...|[-0.7535835610806...|[-13.450613627224...|
|s3://p8-ocr/Test/...| Pineapple Mini|[0.0, 4.49807, 0....|[0.0, 4.498069
7631...|[-0.7574367874115...|[-8.4788524004423...|
```

Validation du résultat

The diagram illustrates the validation process. On the left, a smartphone displays a Jupyter Notebook interface. The notebook shows the following code and its execution results:

```
df = pd.read_parquet('s3://p8-data/0')  
df.head()  
features  
0    s3://p8-data/0.0,  
0.44703647, 0.0,  
1    s3://p8-data/0.05587  
4.080593, 0.05587  
2    s3://p8-data/0.0  
7    s3://p8-data/0.0
```

A large watermark reading "RESULT" is overlaid on the screen. On the right, a screenshot of the Amazon S3 console shows a list of objects in a folder named "results/". The objects are listed as follows:

Nom	Type	Dernière modification	Taille	Classe de stockage
_SUCCESS	-	12 Jun 2023 08:46:42 PM CEST	0 o	Standard
part-00000-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:41:06 PM CEST	18.4 Mo	Standard
part-00001-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:40:50 PM CEST	18.3 Mo	Standard
part-00002-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:41:17 PM CEST	18.4 Mo	Standard
part-00003-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:41:33 PM CEST	18.4 Mo	Standard
part-00004-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:41:46 PM CEST	18.6 Mo	Standard
part-00005-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:42:23 PM CEST	18.3 Mo	Standard
part-00006-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:42:11 PM CEST	18.1 Mo	Standard
part-00007-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:42:37 PM CEST	18.0 Mo	Standard
part-00008-635cc71d-ab2a-4547-9459-a990f6427446-c000.snappy.parquet	parquet	12 Jun 2023 08:42:52 PM CEST	18.0 Mo	Standard

Conclusion

- 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..





Merci !