

DÉPLOYEZ UN MODÈLE DANS LE CLOUD



Victoire MOHEBI
Janvier 2023



AGENDA

- I** Problématique & Présentation des données
- II** Présentation de l'architecture Big Data retenue
- III** Présentation de l'environnement Big Data dans le cloud
- IV** Prétraitement des images
- V** Conclusion & Perspective



I. PROBLÉMATIQUE ET PRÉSENTATION DES DONNÉES



PROBLÉMATIQUE

- ❑ Startup « *Fruit!* » souhaite créer une application mobile grand public de reconnaissance de fruit et affichage d'informations
- ❑ Développer des robots cueilleurs intelligents
- ❑ Mettre en place un moteur de classification des images de fruits



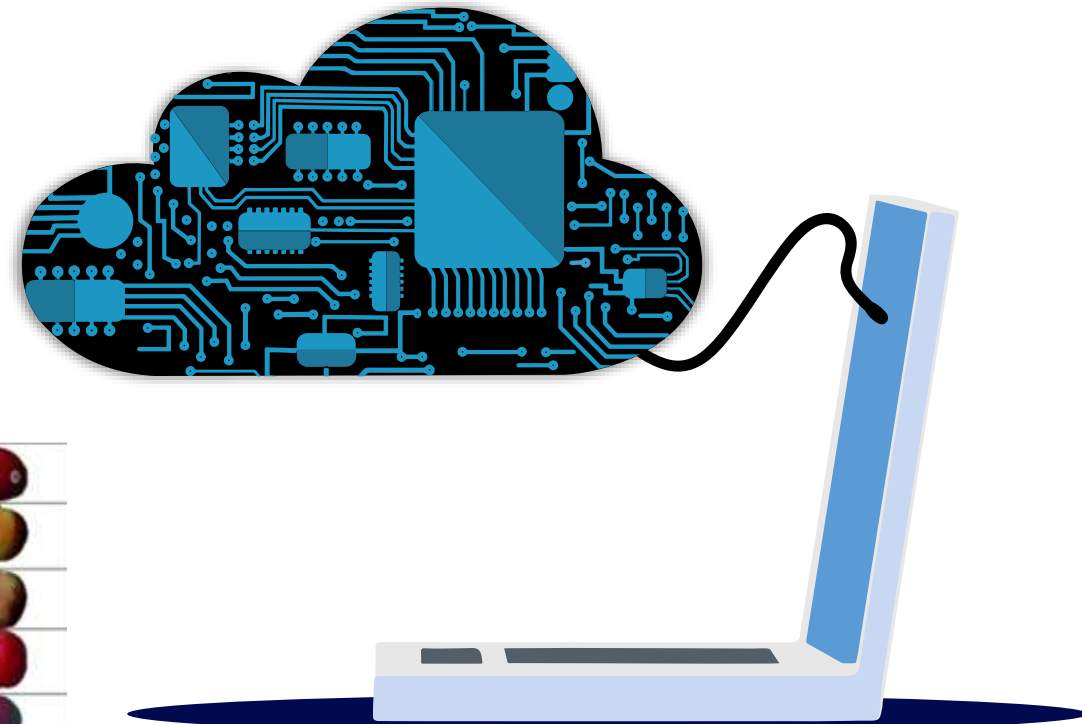
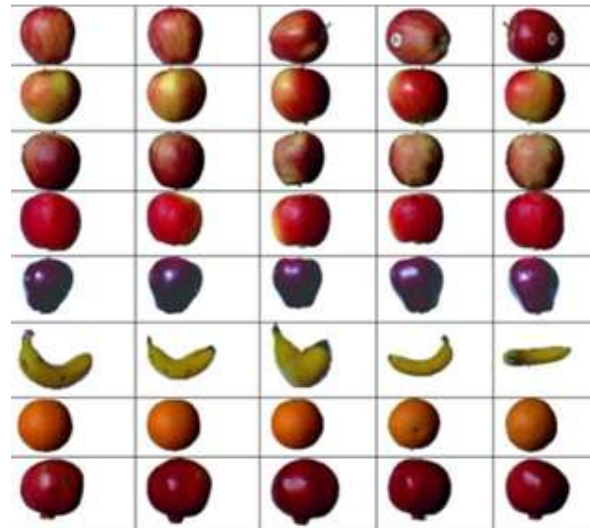
Fruits!


MISSION

- ❑ Mettre en place l'architecture Big Data
 - Augmentation du volume des données
 - Preprocessing et réduction de dimension
- ❑ Moyens : Scripts pyspark + solution évolutive

PRÉSENTATION DES DONNÉES

- Le jeu de donnée « fruit 360 » est disponible sur Kaggle
- Au total 90380 image de 131 fruits et legumes
- Jeu de donnée d'entraînement : 67 692 images (un fruit ou un légume par image).
- Jeu de donnée de test : 22 688 images (un fruit ou un légume par image)
- Jeu de données de multi fruits non labellisé : 103 images
- Taille de l'image : 100 x 100 pixels



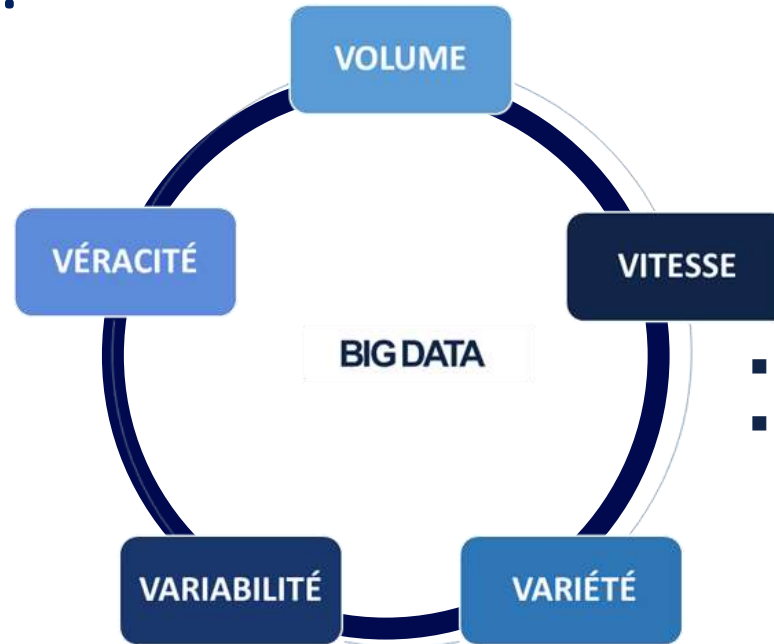
The background is a dark, blue-toned photograph of a server room with rows of server racks. Overlaid on this is a network diagram consisting of a central cloud icon in a circle, connected by lines to several other circular icons: a smartphone, a laptop, a desktop monitor, a database cylinder, and a camera lens. The text is in a bold, white, sans-serif font.

II. PRÉSENTATION DE L'ARCHITECTURE BIG DATA RETENUE

QU'EST-CE QUE LE BIG DATA

HOW BIG IS THE BIG DATA ?

- Traiter des volumes de données massifs
- **RÈGLE DE 5V :**
Volume, Vitesse, Variété, Variabilité, Véracité



ARCHITECTURE BIG DATA: SCALABILITÉ VERTICALE

La distribution, un élément clé dans l'architecture Big Data

- Ajout des RAM du CPU...
- Très cher et que l'on utilisait des hardwares qui étaient voués à être jetés

QU'EST-CE QUE LE CALCUL DISTRIBUÉ?

LES ENJEUX DE PASSAGE À L'ÉCHELLE ?

- Capacité réseau limitée
- Stockage insuffisant, que ce soit en RAM ou sur disque dur
- Puissance de calcul insuffisante

SCALABILITÉ HORIZONTALE

- Passage à l'échelle est plus facile
- Calculs parallélisables dans une architecture distribuée

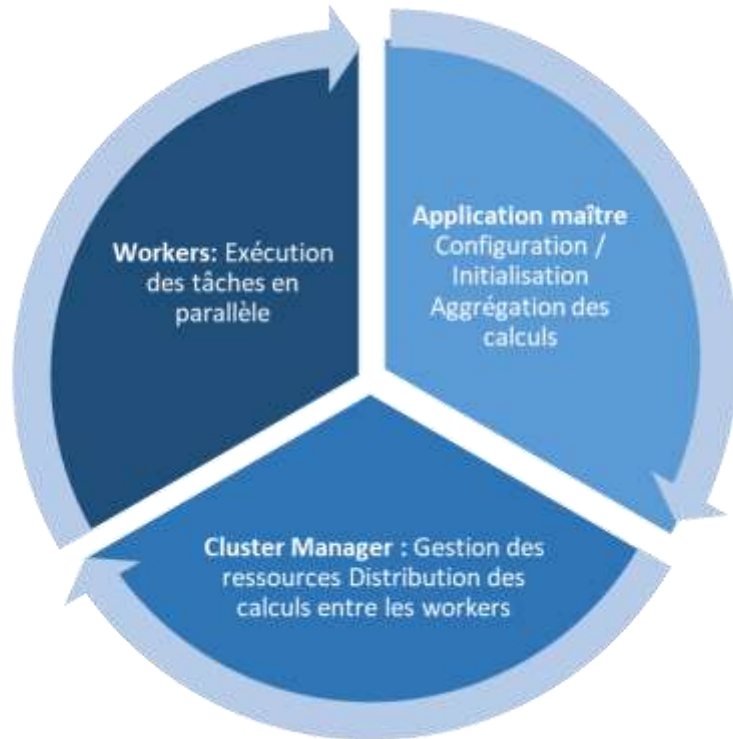


SPARK ; PLATEFORME DE CALCUL DISTRIBUÉ

- Spark (ou Apache Spark) , Framework open source de calcul distribué in-memory pour le traitement et l'analyse de données massives
- Stocker et traiter le big data sur différents ordinateurs communiquant via un réseau
- Consiste en la réalisation d'opérations sur des données qui ne sont pas stockées en un seul endroit, mais éparpillées au sein d'un réseau de différentes machines (un « cluster »).

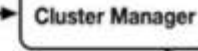


ARCHITECTURE SPARK



Spark peut lancer un traitement sur une machine locale ou sur une machine en ligne

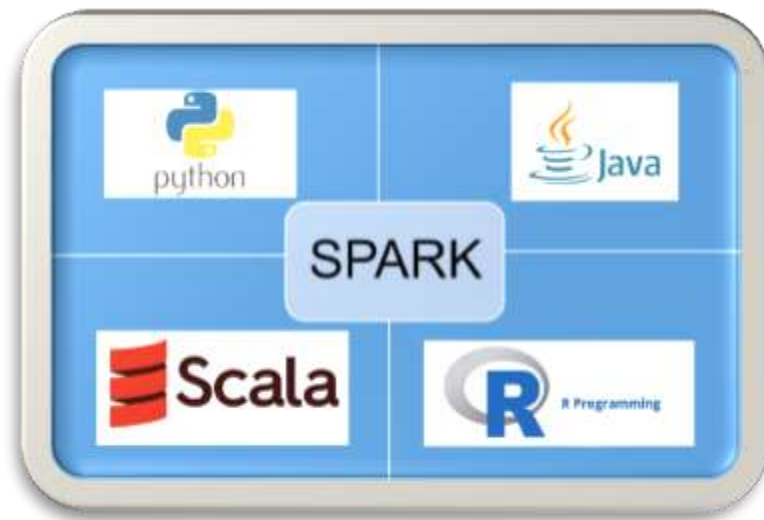
Application Python exécutée JVM l'objet Sparkcontext et/ou Sparksession est instancié



Workers ,machines virtuelles pour l'exécution des calculs

L'objet Sparkcontext crée une machine virtuelle pour la mise en œuvre de l'application et de la distribution des calculs

Quel langage de programmation pour utiliser SPARK?



Qu'est-ce que PySpark?



- PySpark , une interface pour Apache Spark en Python
- Ecrire des applications Spark à l'aide d'API Python
- *Shell PySpark* pour analyser interactivement les données dans un environnement distribué
- PySpark supporte la plupart des fonctionnalités de Spark telles que Spark SQL, Data Frame, Streaming, MLlib (Machine Learning) et Spark Core.

Executer *Spark* en local

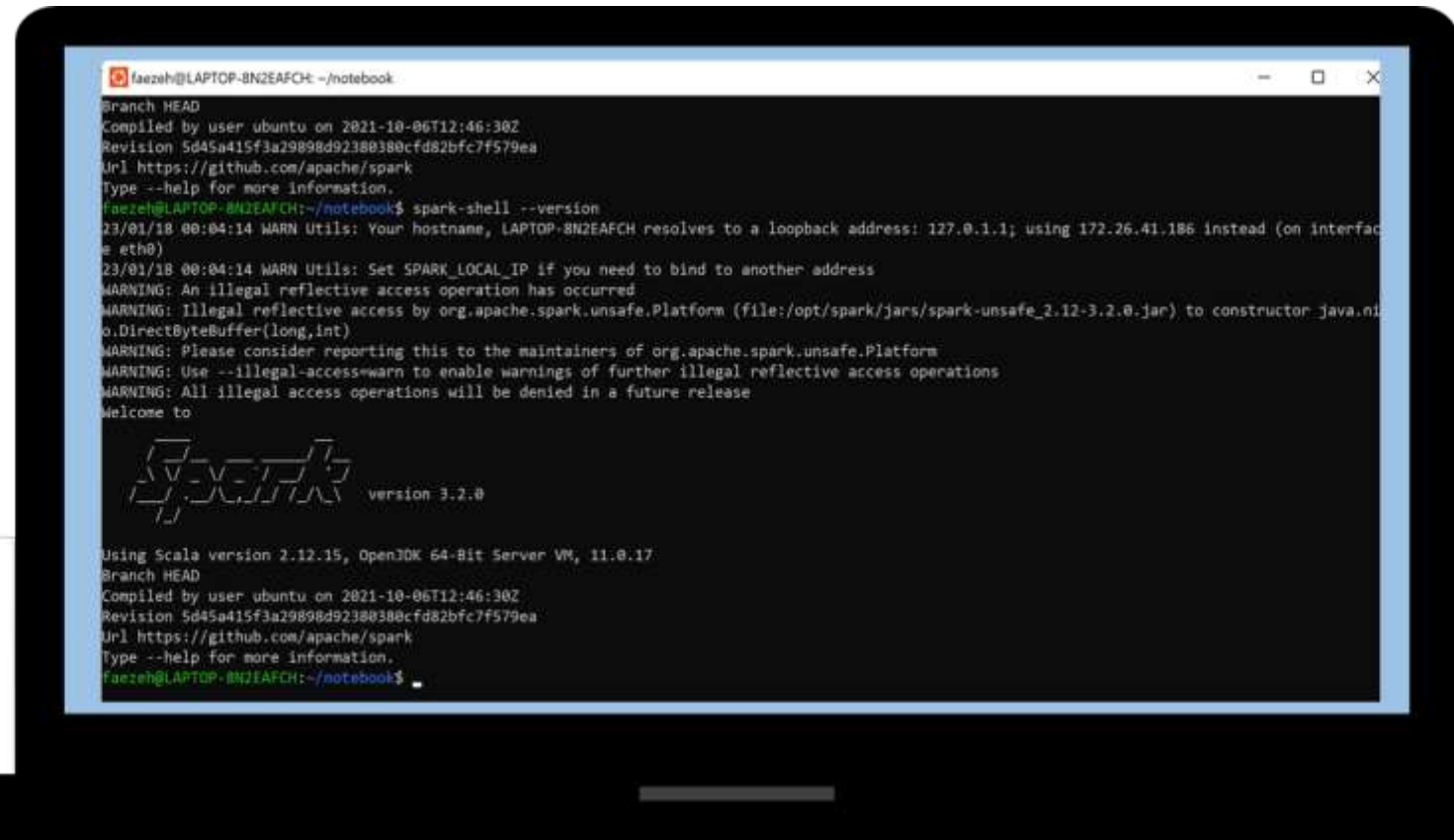
Configurations requises

- Virtual Machine **UBUNTU**
- **JAVA 8+**
- Télécharger le fichier TAR de Spark
- Installer le *PySpark* avec PyPi
- Spécifier des variables d'environnement dans *.bashrc*
- Lancer un *shell spark*



Download Apache Spark™

1. Choose a Spark release:
2. Choose a package type:
3. Download Spark: `spark-3.3.1-bin-hadoop2.tgz`
4. Verify this release using the 3.3.1 signatures, checksums and project release KEYS by following these

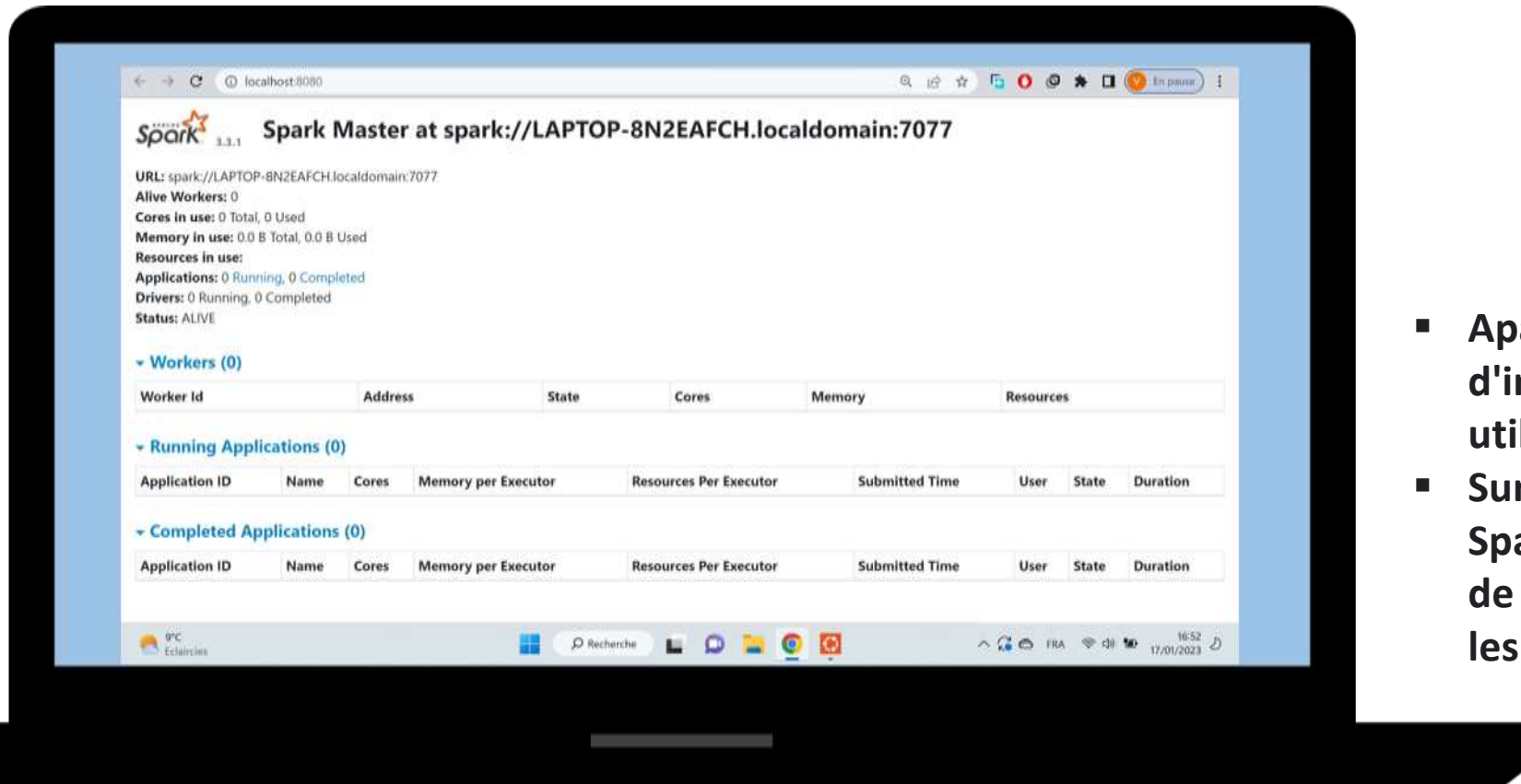


```
faezeh@LAPTOP-8N2EAFCH: ~/notebook
Branch HEAD
Compiled by user ubuntu on 2021-10-06T12:46:30Z
Revision 5d45a415f3a29898d92380380cfd82bfc7f579ea
Url https://github.com/apache/spark
Type --help for more information.
faezeh@LAPTOP-8N2EAFCH:~/notebook$ spark-shell --version
23/01/18 00:04:14 WARN Utils: Your hostname, LAPTOP-8N2EAFCH resolves to a loopback address: 127.0.1.1; using 172.26.41.186 instead (on interface eth0)
23/01/18 00:04:14 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.2.0.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Welcome to

  _ _ _ _ _
 / _ _ _ _ \   version 3.2.0
( _ _ _ _ _ )
  \ _ _ _ _ /
   _ _ _ _ _

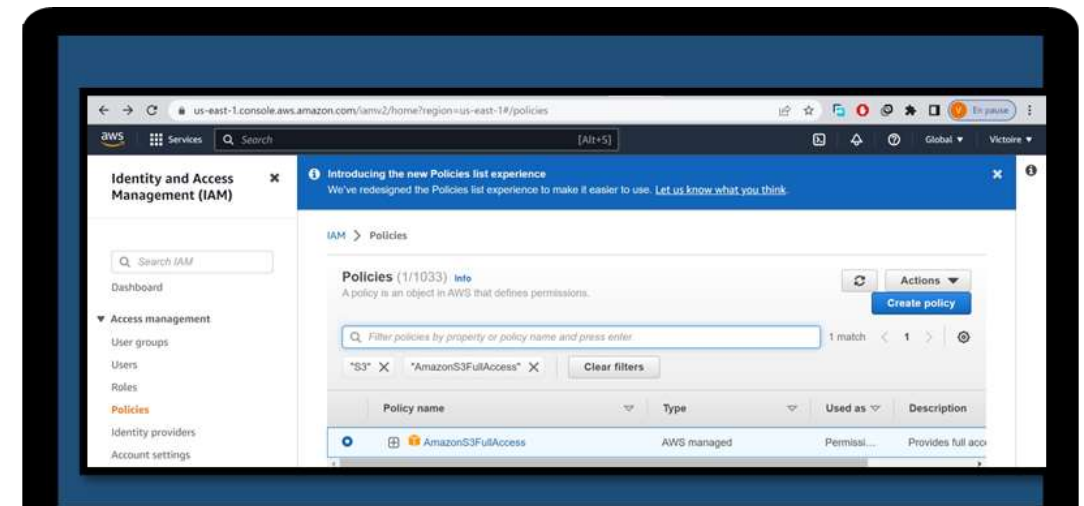
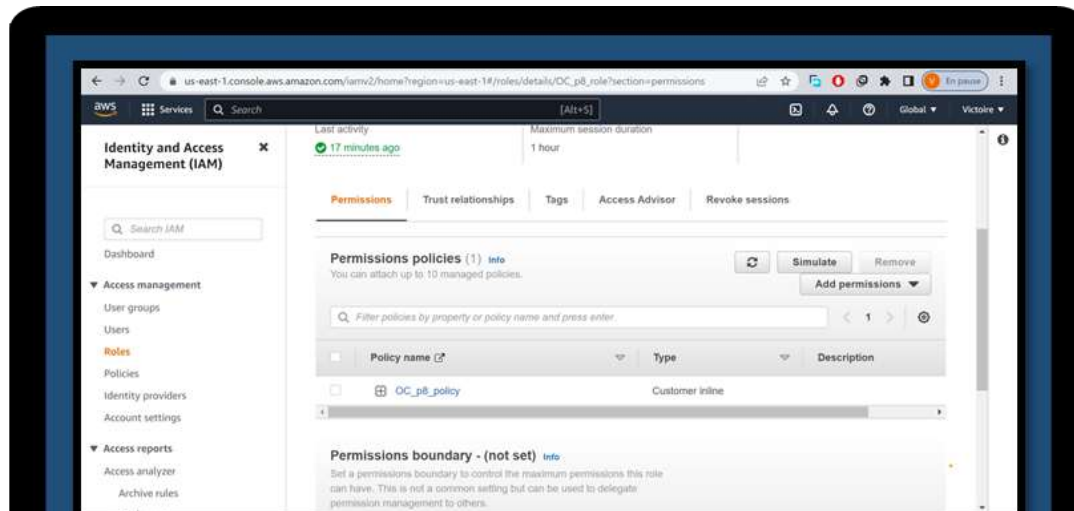
Using Scala version 2.12.15, OpenJDK 64-Bit Server VM, 11.0.17
Branch HEAD
Compiled by user ubuntu on 2021-10-06T12:46:30Z
Revision 5d45a415f3a29898d92380380cfd82bfc7f579ea
Url https://github.com/apache/spark
Type --help for more information.
faezeh@LAPTOP-8N2EAFCH:~/notebook$
```


Spark UI



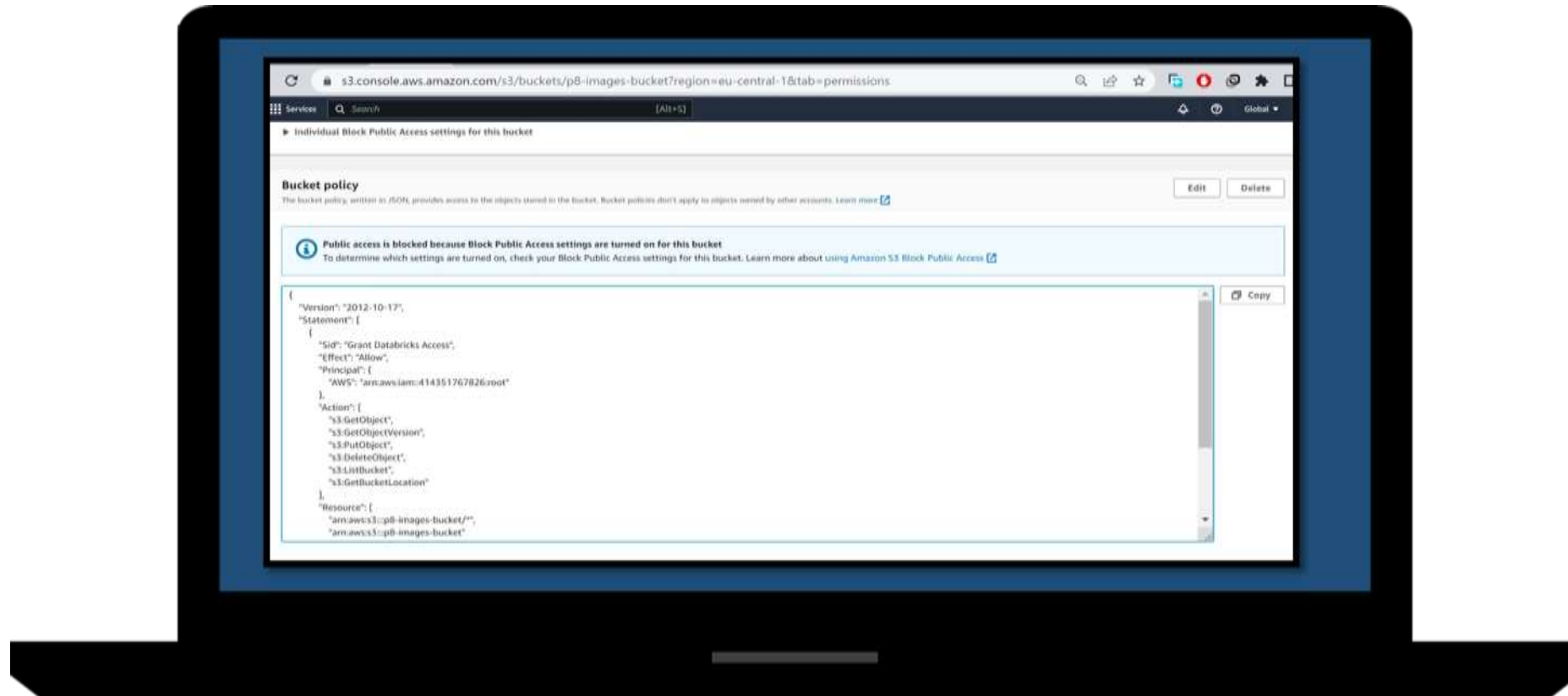
- Apache Spark fournit une suite d'interfaces utilisateur/interface utilisateur
- Surveiller l'état de l'application Spark/PySpark, la consommation de ressources du cluster Spark et les configurations Spark

CONFIGURATIONS REQUIRES



Amazon S3, AWS Identity and Access IAM sont entièrement conformes au Code RGPD

CONFIGURATIONS REQUIRES

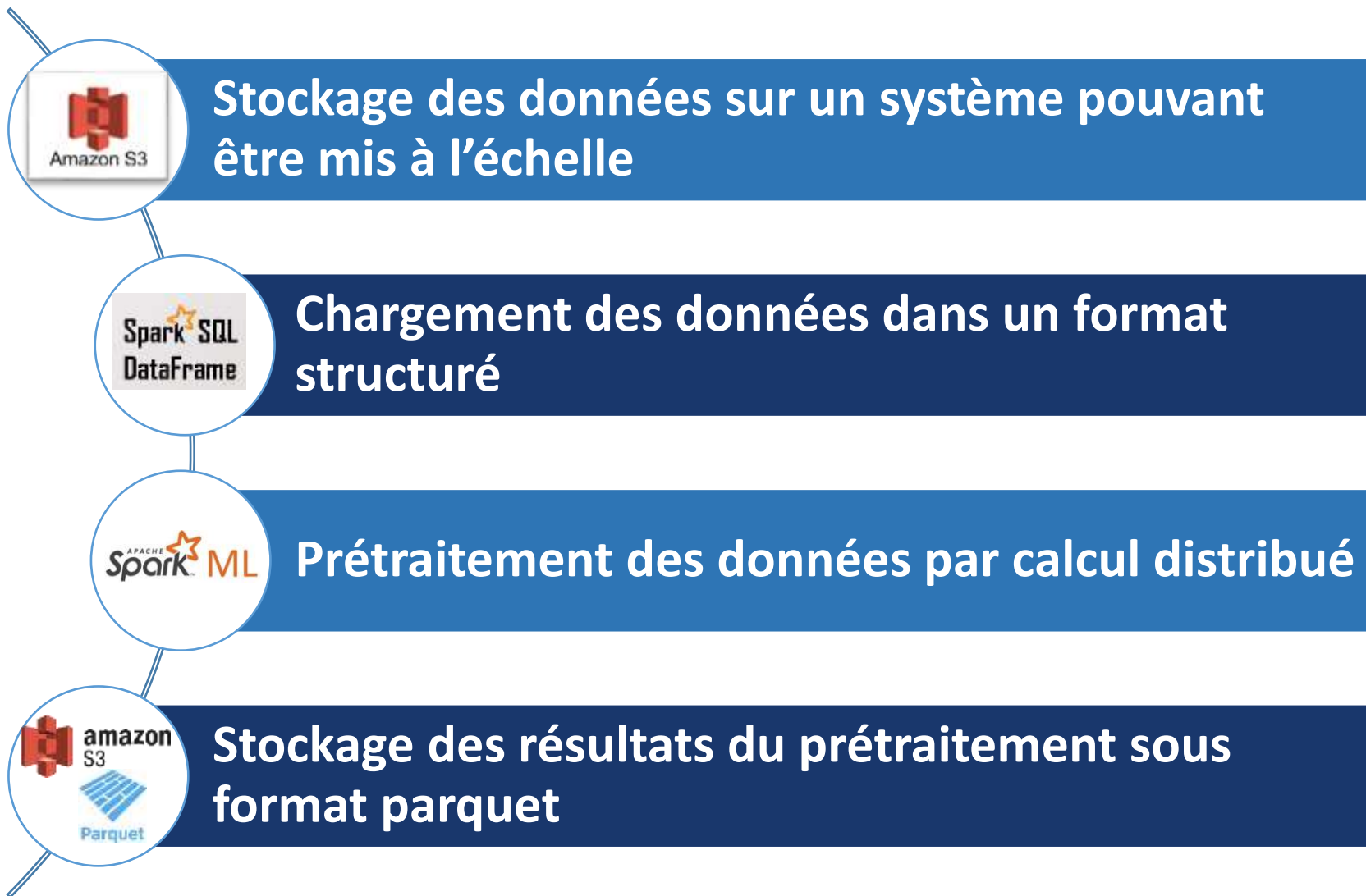


- Les stratégies de compartiment et les stratégies utilisateur sont deux options de stratégie d'accès disponibles pour accorder des autorisations aux ressources Amazon S3
- Les deux utilisent le langage d'accès Policy basé sur JSON.



III. PRÉSENTATION DE L'ENVIRONNEMENT BIG DATA DANS LE CLOUD

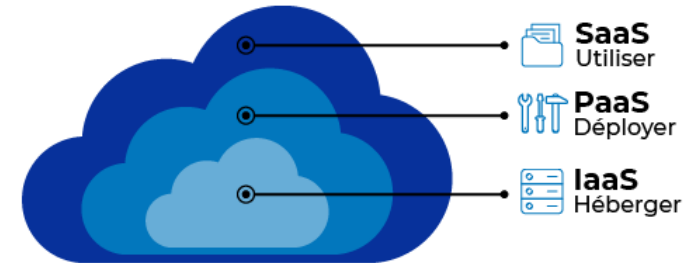
Décomposition du problème



CLOUD COMPUTING

■ Trois principaux types de services Cloud :

- Infrastructure en tant que service (IaaS),
- logiciel en tant que service (SaaS)
- plate-forme en tant que service (PaaS)



■ Caractéristiques du modèle Cloud :

- Plus besoin d'ordinateurs hyper performants
- Plus besoin d'organiser sa sauvegarde sur CD, clé USB ou autre.
- Possibilité de partager ses contenus et des ressources avec son réseau personnel.
- Passage du stockage local (concret) au Cloud,

• Différents fournisseurs principaux de cloud :

- Ex. Amazon Web Service, AZURE, Google cloud, IBM Cloud

■ Rapport du cloud avec le big data et le calcul distribué :

- Louer des tiers des ressources matérielles pour une durée déterminées

PLATEFORMES UTILISÉES POUR LE DÉPLOIEMENT SUR LE CLOUD

Amazon WEB SERVICE



AWS- IAM
AWS Identity and Access
Management



AWS IAM

AWS- S3
Amazon Simple
Storage Service



Amazon S3

Databricks
Outil d'ingénierie
de données basé
sur le cloud



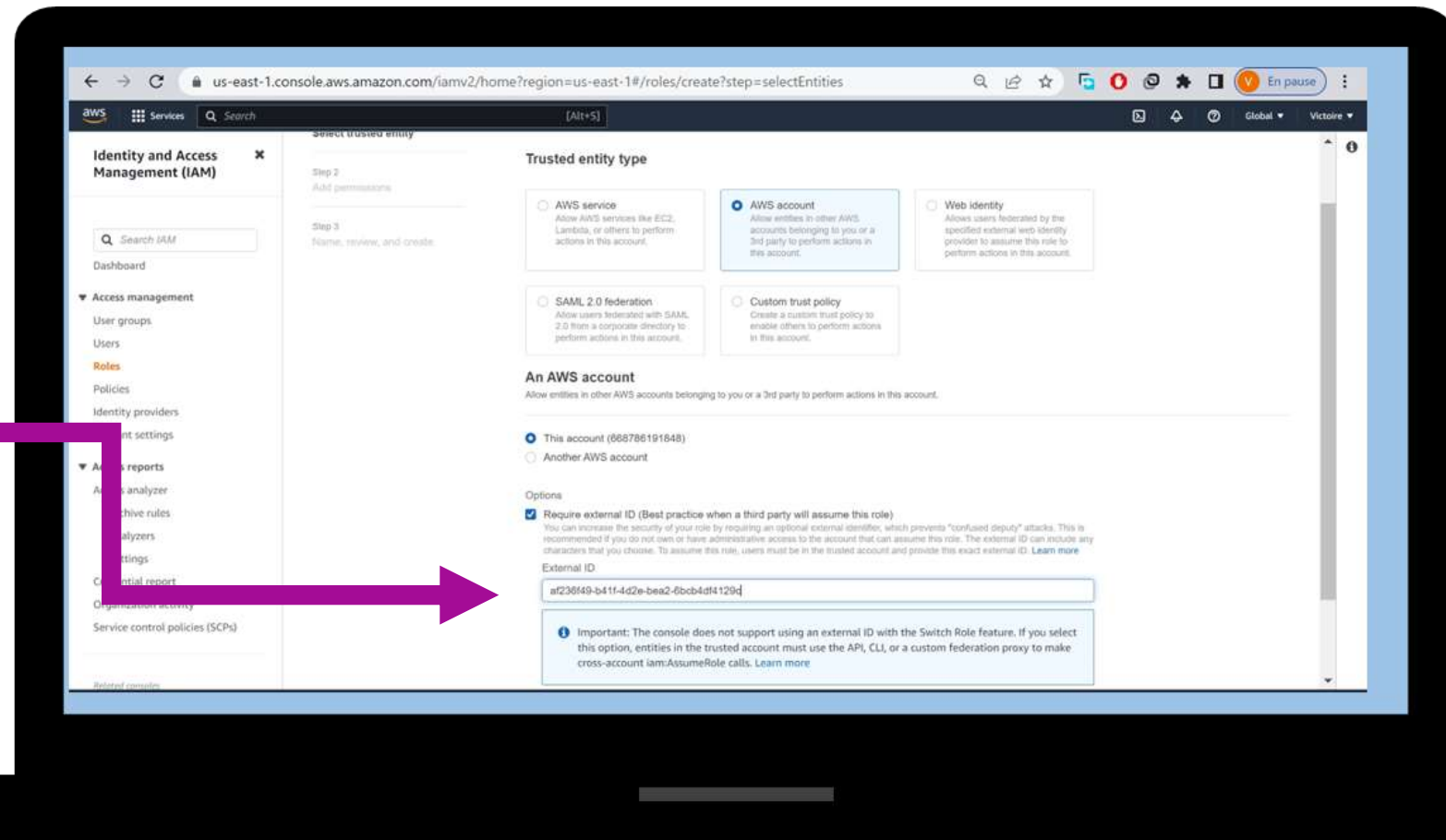
databricks

CONFIGURATIONS REQUIRES



Cross-account IAM Rôle

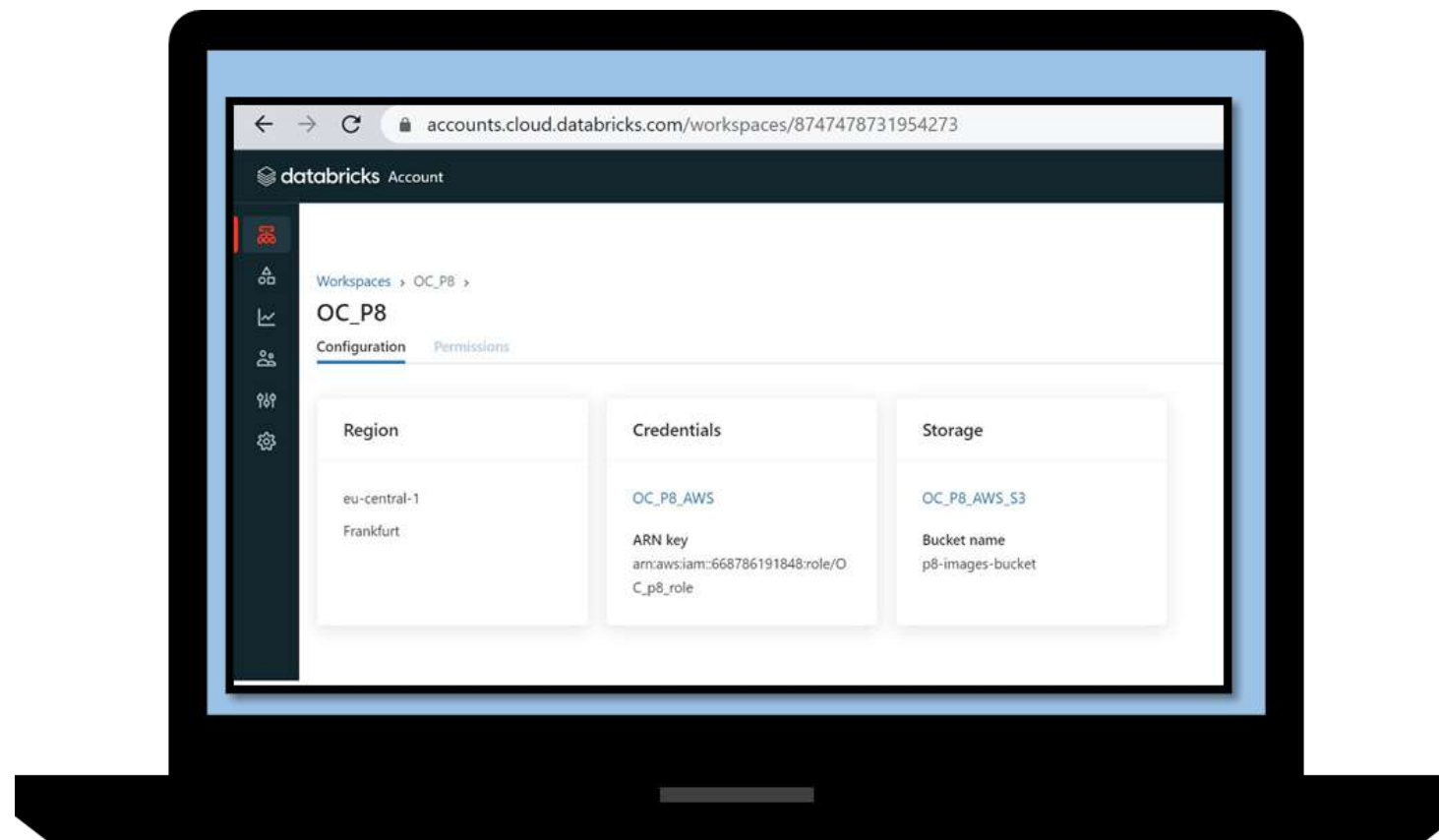
Databricks ID
account



CONFIGURATIONS REQUISES



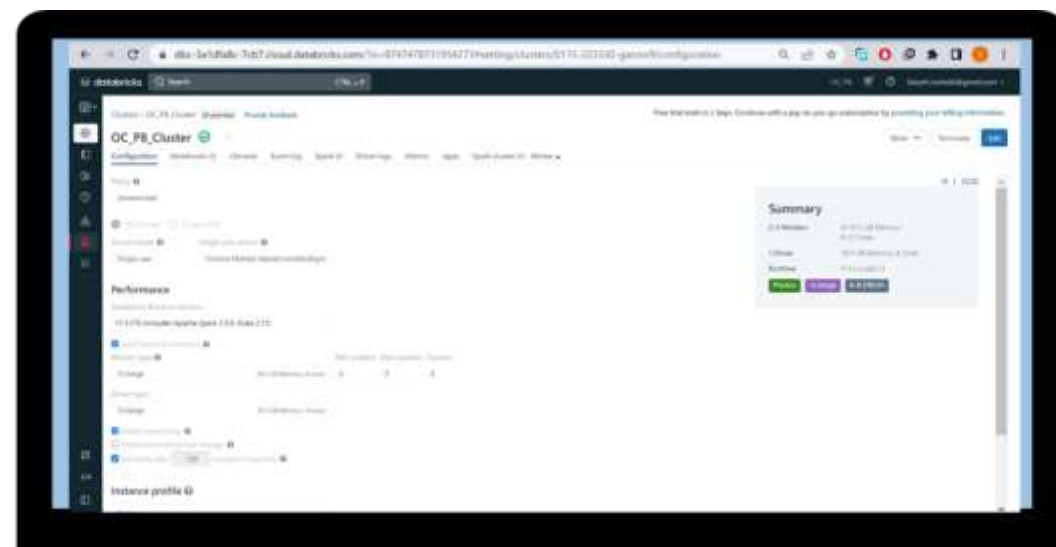
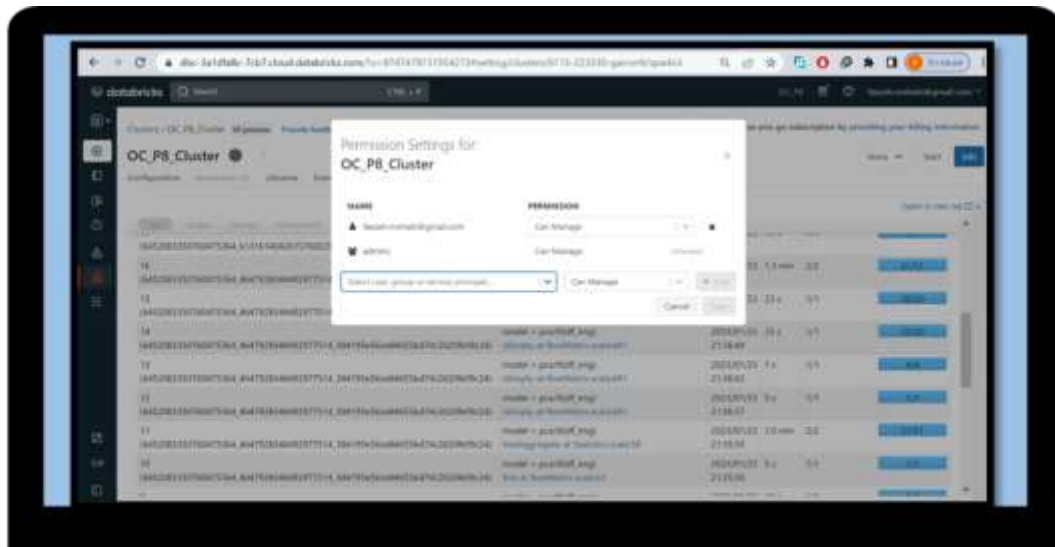
Création de Workplace



CONFIGURATIONS REQUISES



Création et configuration du cluster



A close-up, low-angle shot of a hand typing on a dark keyboard. The background is heavily blurred, showing bokeh light effects in shades of blue and white, suggesting an indoor setting with artificial lighting. The overall mood is professional and focused.

IV. PRÉTRAITEMENT DES IMAGES

PRÉPARER LES IMAGES POUR LA MODÉLISATION

Extraction
d'information
des images
(features
extraction)

Réduction de
dimensions

Solution envisageable :

Egalisation
histogramme et
redimensionne
ment

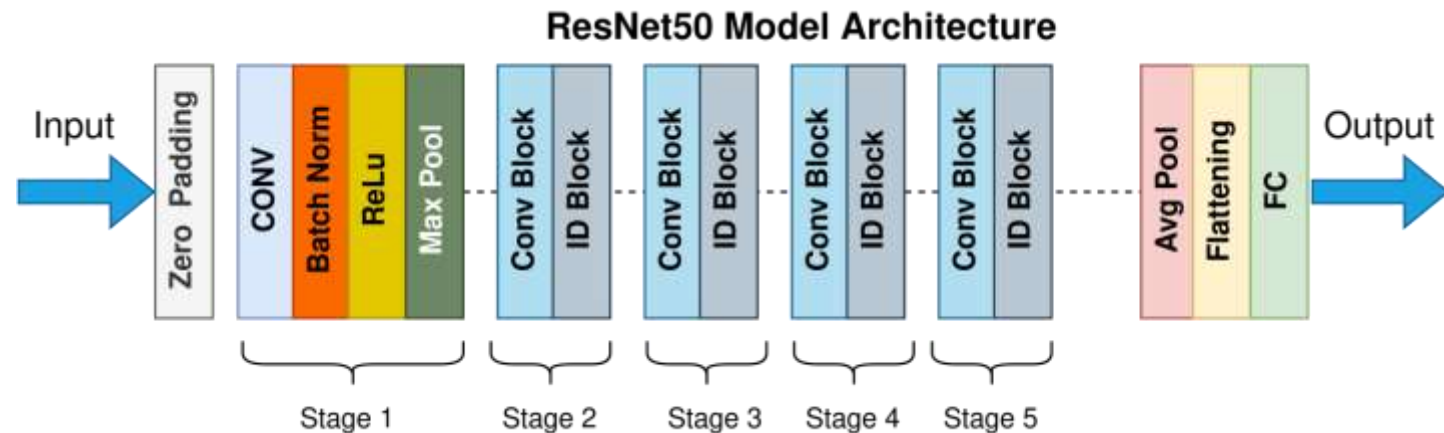
Traitement
d'image +
extraction de
features (ORB,
SURF, SIFT, etc.)

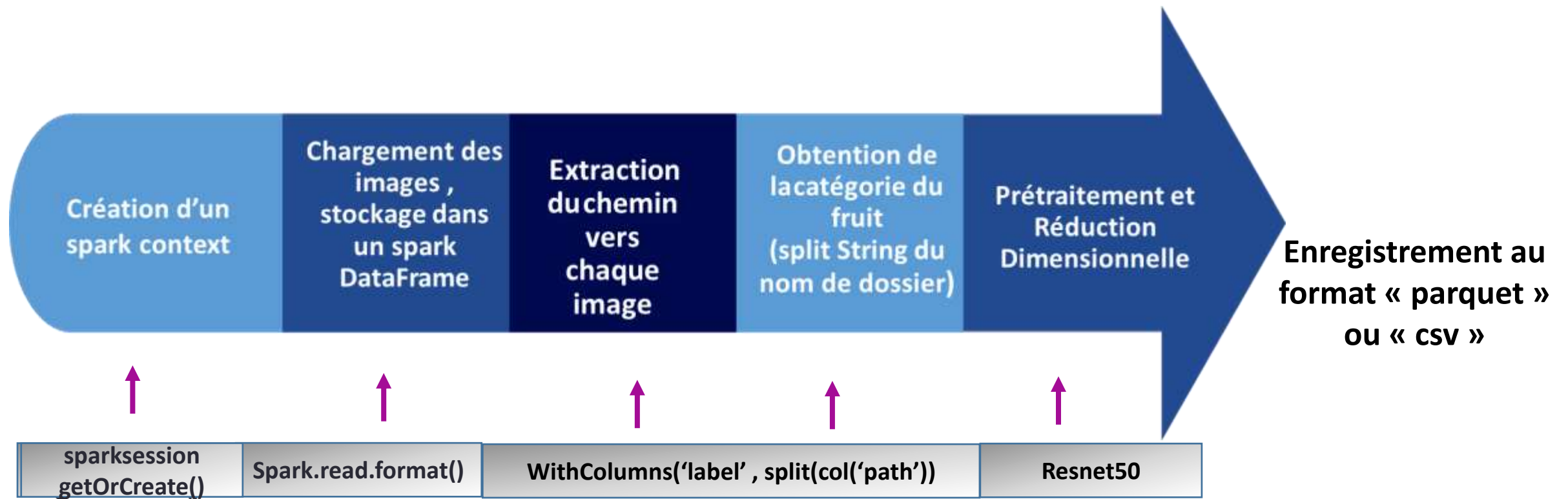
Algorithme
préentraînés
(Transfer
Learning)



FEATURE EXTRACTION AVEC *RESNET50*

- Le modèle Resnet50(Residual Network)
- Resnet50 : pré-entraîné sur la base de données *Imagenet* (plus de 14 millions d'images classées en plus de 20 000 groupes)
- Resnet50 est composé de 50 couches
- La particularité : introduire des connexions résiduelles. Contrairement aux réseaux de neurones convolutifs ave l'architecture linéaire
- Le réseau résiduel, la sortie des couches précédentes est reliée à la sortie de nouvelles couches pour les transmettre toutes les deux à la couche suivante.



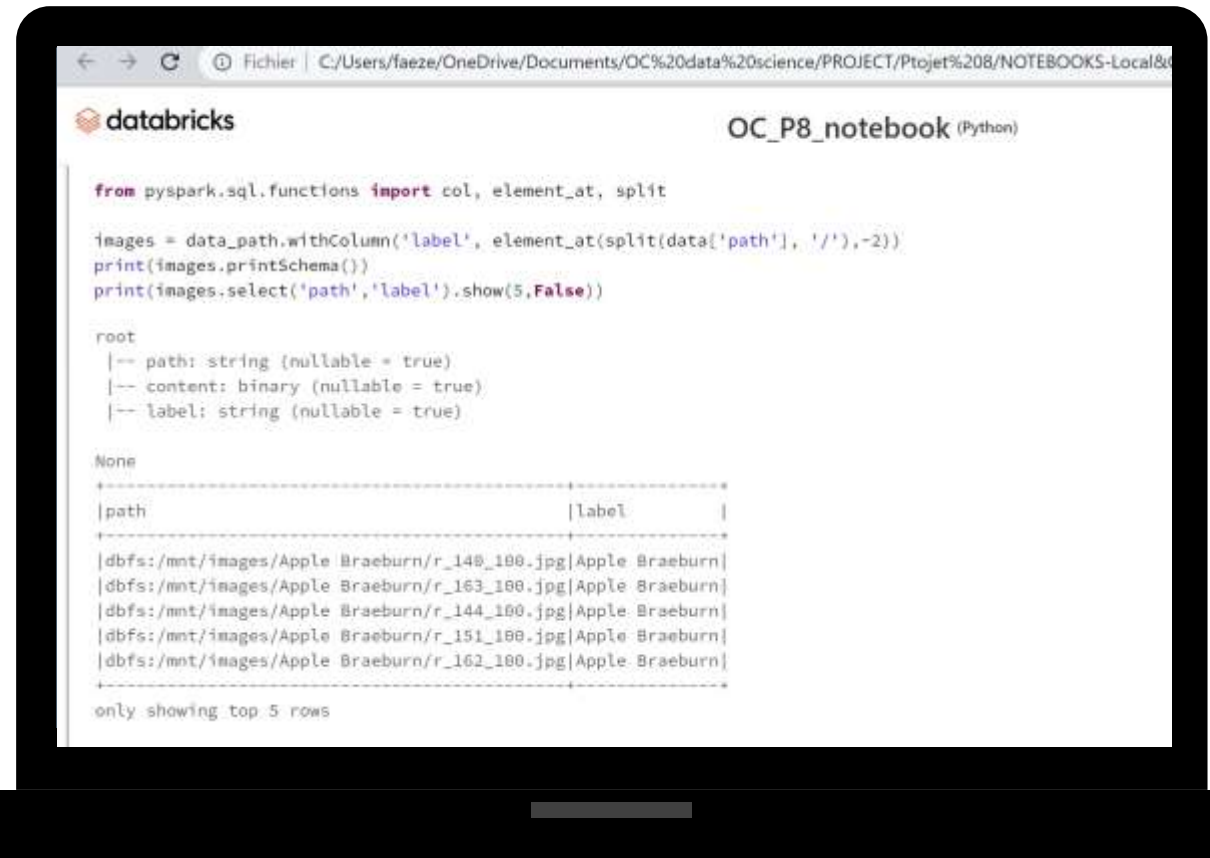


■ LE FORMAT PARQUET?

- Alternative au stockage CSV
- Parquet est un format de fichier open source largement utilisé par l'écosystème Hadoop et SPARK
- Combinaison de formats de stockage en ligne et en colonnes (hybrides)

INSTANCE SPARK

- Extraction du chemin vers chaque fichier
- Obtention de la catégorie de fruit



The screenshot shows a Databricks notebook titled "OC_P8_notebook (Python)". The code defines a Spark DataFrame with columns 'path' and 'label'. The output is a table showing the first 5 rows of data.

```
from pyspark.sql.functions import col, element_at, split

images = data_path.withColumn('label', element_at(split(data['path'], '/'), -2))
print(images.printSchema())
print(images.select('path', 'label').show(5, False))

root
 |-- path: string (nullable = true)
 |-- content: binary (nullable = true)
 |-- label: string (nullable = true)

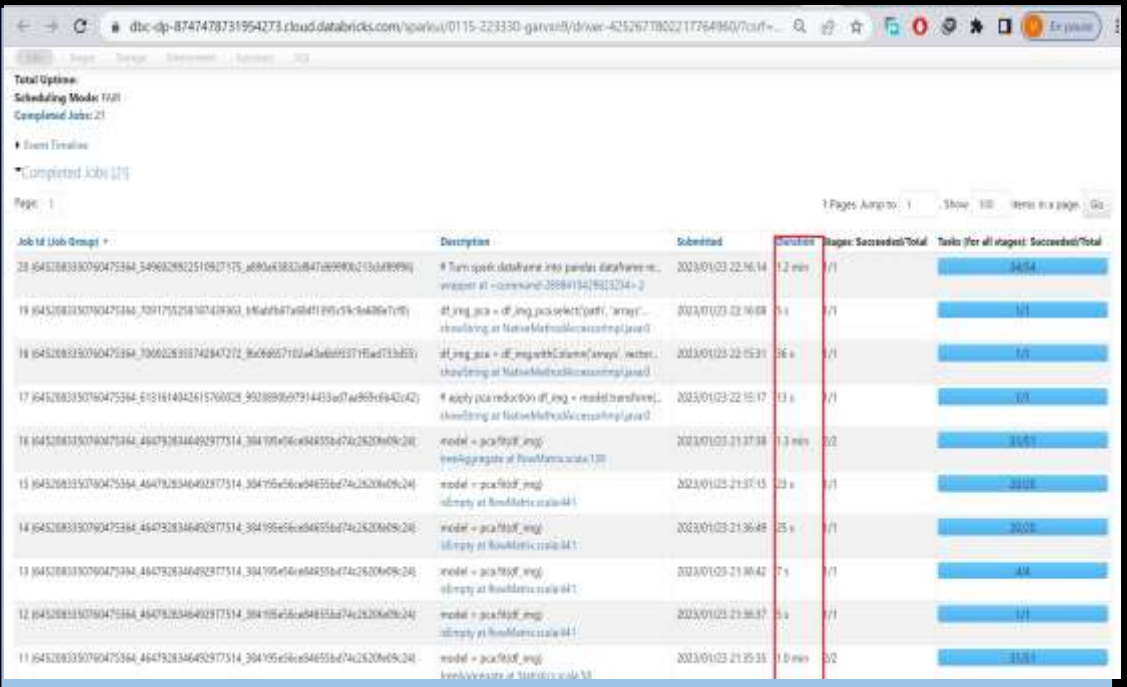
None
```

path	label
dbfs:/mnt/images/Apple Braeburn/r_148_100.jpg	Apple Braeburn
dbfs:/mnt/images/Apple Braeburn/r_163_100.jpg	Apple Braeburn
dbfs:/mnt/images/Apple Braeburn/r_144_100.jpg	Apple Braeburn
dbfs:/mnt/images/Apple Braeburn/r_151_100.jpg	Apple Braeburn
dbfs:/mnt/images/Apple Braeburn/r_162_100.jpg	Apple Braeburn

only showing top 5 rows

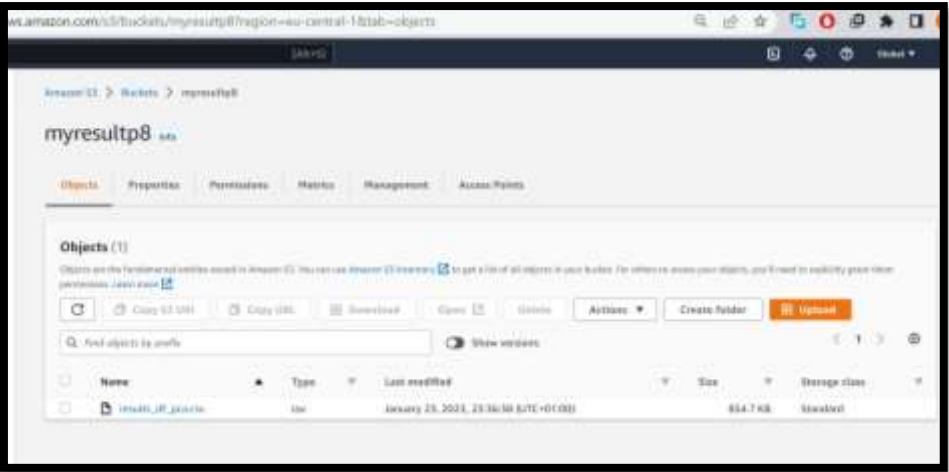
SPARK UI dans le DATABRICKS

- SPARK UI (interface utilisateur Spark) est l'interface Web
- Surveiller et inspecter les exécutions de tâches Spark dans un navigateur Web
- Le cluster a mis moins de 5 min pour traiter les 2000 images dans le compartiment S3 !

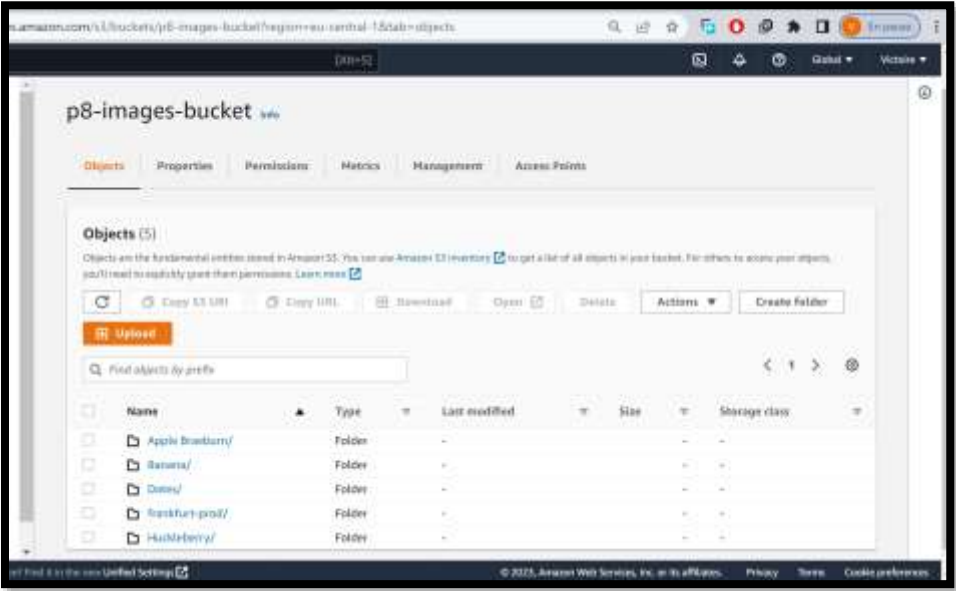


Job ID (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
28 (6452083330790475364_3496029822516927175_880a63832b847d09900213db8999f)	# Turn spark.databricks.read.parallel.dataframe.enabled at <command> (398841341923234) 2	2023/01/23 22:16:14	12 min	1/1	34/34
18 (6452083330790475364_700175528107420303_81a6b67a9841195c9b4e4867c79)	df_img_psa = df_img_psa.select('path', 'image') showString at NotebookExecutionConsole(jspid)	2023/01/23 22:16:08	5 s	1/1	1/1
18 (6452083330790475364_700022693742847272_8a06657102a3a6d937f5ad713b55)	df_img_psa = df_img_psa.select('image', 'vector') showString at NotebookExecutionConsole(jspid)	2023/01/23 22:15:31	16 s	1/1	1/1
17 (6452083330790475364_6131614042615700029_952889959714433a7a269e642a42)	# apply (pca.reduction) df_img = model.transform(showString at NotebookExecutionConsole(jspid)	2023/01/23 22:15:17	13 s	1/1	1/1
16 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) writeAggregate at RowMatrix.scala:130	2023/01/23 21:37:38	13 min	3/2	31/31
15 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) sortBy at RowMatrix.scala:44	2023/01/23 21:37:45	23 s	1/1	31/31
14 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) sortBy at RowMatrix.scala:44	2023/01/23 21:36:48	25 s	1/1	30/30
13 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) sortBy at RowMatrix.scala:44	2023/01/23 21:36:42	7 s	1/1	4/4
12 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) sortBy at RowMatrix.scala:44	2023/01/23 21:36:37	6 s	1/1	1/1
11 (6452083330790475364_4647928346492977514_304195e56e94655b474c2620a69c24)	model = pca.fit(df_img) sortBy at RowMatrix.scala:44	2023/01/23 21:35:35	18 min	3/2	31/31

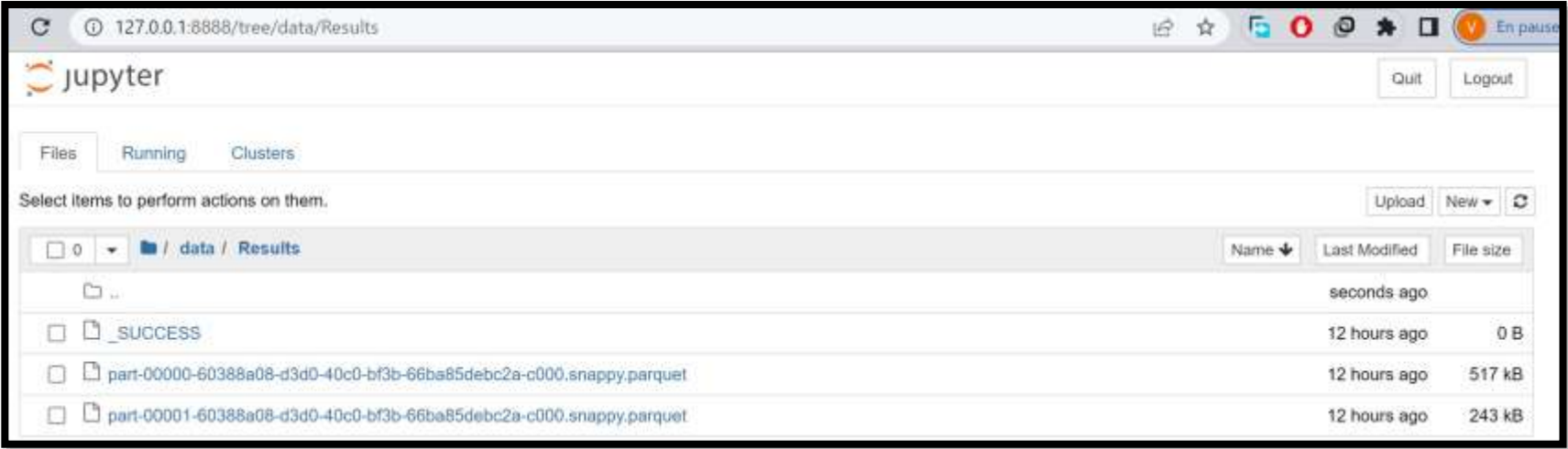
Bucket S3 contenant les résultats



Bucket S3 contenant les images initiales



Enregistrement en local des resultats sous format parquet





V. CONCLUSION

PERSPECTIVE

CONCLUSION

- Solution **PaaS** (Plateforme as a Service)
- Utilisation de pySpark pour anticiper la forte évolution de la base de données
- Prétraitement des images : Extraction des features avec TL & Réduction dimensionnelle
- Sauvegarde des résultats au format parquet ou csv
- Application utilisable en mode local ou en ligne

Déploiement sur le cloud

- Fournisseur choisit : AWS + Databricks
- Services utilisés : S3 – IAM – Databricks

Difficultés rencontrée

- Nombreuses possibilités techniques : choix complexes
- Débug complexe dû à des erreurs peu explicites (superposition Spark/Java)

PERSPECTIVE

- **Prétraitement des images pour le cas réels (recadrage, plusieurs fruits, arrière plan, etc.)**
- **Transfer Learning avec fine tuning pour meilleur accuracy pour la classification**
- **Déployer le modèle en production sur un cluster**
- **Monitoring...**
- **Tester d'autres solutions technique pour le déploiement dans le cloud , ex. service EMR d'Amazon Web Services**
- **Tester d'autre fournisseur de cloud comme Google Cloud Platform (GCP) Microsoft AZURE ou IBM Cloud**
- **Automatisation des tache de collecter des photos**



MERCI DE VOTRE ATTENTION !