



# Project 8

***Déployez un modèle dans le cloud***

***Tetiana Lemishko***

# Sommaire:

- Contexte, mission et jeu de données
- Le Big Data
- Architecture retenue et chaîne de traitement
- Conclusion

# Contexte et mission de projet



## Fruits!

"**Fruits!**" est une très jeune start-up de l'AgriTech, qui cherche à proposer des solutions innovantes pour la récolte des fruits.

La start-up souhaite dans un premier temps 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.

Pour la start-up, cette application permettrait 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.

De plus, le développement de l'application mobile permettra de construire une première version de l'architecture Big Data nécessaire.

### Mission:

Développer dans un environnement Big Data une première chaîne de traitement des données qui comprendra le preprocessing et une étape de réduction de dimension.

# Jeu de données

**Origine:** Kaggle <https://www.kaggle.com/moltean/fruits>

- Images de 131 variétés de fruits et légumes labélisés
- Plusieurs variétés du même fruit (exemple : Strawberry Wedge)

## Caractéristiques :

- Images 100x100 JPEG RGB
- Photos sur fond blanc centrée sur le fruit
- Photos sous tous les angles (rotation tri-axiales)



- Total : 90 483 images
- Jeu d'entraînement : 67 692 images
- Jeu de Test : 22 688 images
- Jeu multi fruits non labellisé : 103 images

# Le Big Data

## Qu'est-ce que le Big Data ?

- En Français : 'les données massives'
- La définition du Big Data est la suivante :
  - des données plus **variées**
  - arrivant dans des **volumes croissants**
  - à une **vitesse plus élevée**

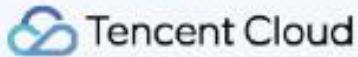
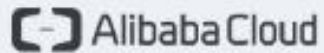
*(C'est ce que l'on appelle les trois « V »)*



Comment traiter et à analyser les données volumineuses (Big Data) plus rapidement?



# Le Big Data. Les principaux fournisseurs de cloud

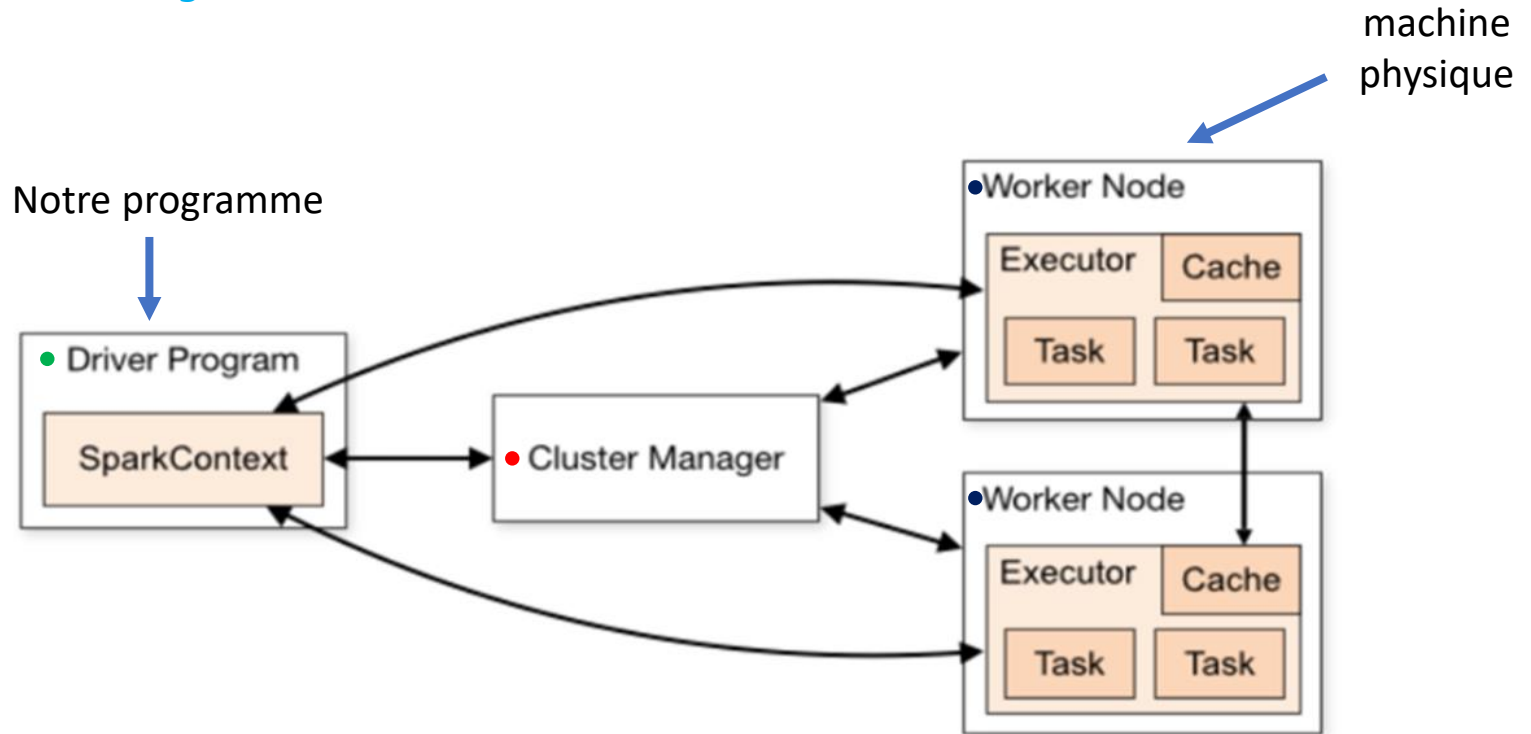


**Quels sont les principaux avantages de l'utilisation du cloud AWS ?**

- Simplicité d'utilisation (convenant pour les débutants)
- Coût abordable

# Le Big Data. Architecture de Spark

*Spark est un moteur de traitement distribué à usage général qui peut être utilisé pour plusieurs scénarios big data.*

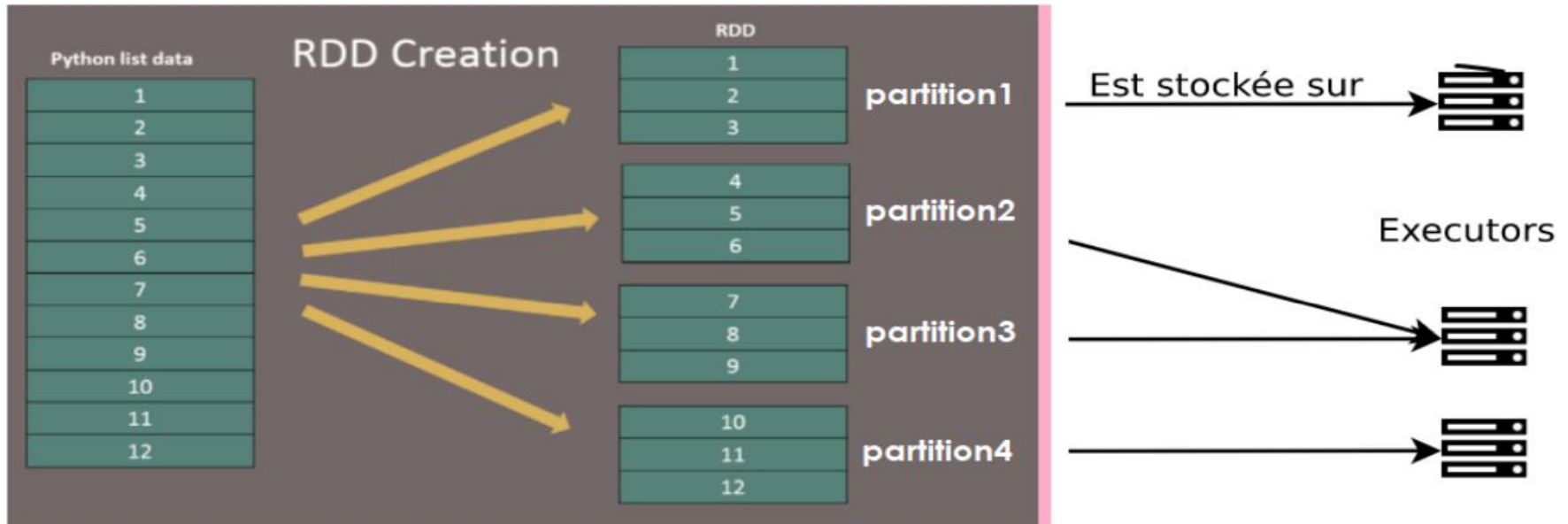


- Driver program: répartit les tâches sur les différents executors
- Cluster manager: instancie et supervise différents workers
- Worker nodes: un worker instancie un executor qui exécute des tâches

# Le Big Data. La structure de données essentiel de Spark (le RDD)

**RDD (Resilient Distributed Datasets):** principale innovation de Spark

RDD est une structure de données essentiel utilisée dans Spark pour exécuter les opérations plus rapidement et efficacement.



Chaque jeu de données dans RDD est divisé en partitions logiques, qui peuvent être calculées sur différents nœuds du cluster.



# Architecture Big Data

Cloud



## Stockage



Amazon S3

*Stockage des images initiales et du resultat de la réduction de dimension*



## Sécurité



AWS IAM

*Clés d'accès + rôle IAM  
(Identity and Access Management)  
Politique IAM qui autorise l'accès aux objets dans S3*



## Traitements



Amazon EC2

*(Elastic Compute Cloud)  
Exécution des scripts dans un notebook Jupyter*



## Accès SSH



PowerShell

*Accès sécurisé SSH à la console du serveur EC2 (des commandes pour les installations)*

# Architecture Big Data. S3

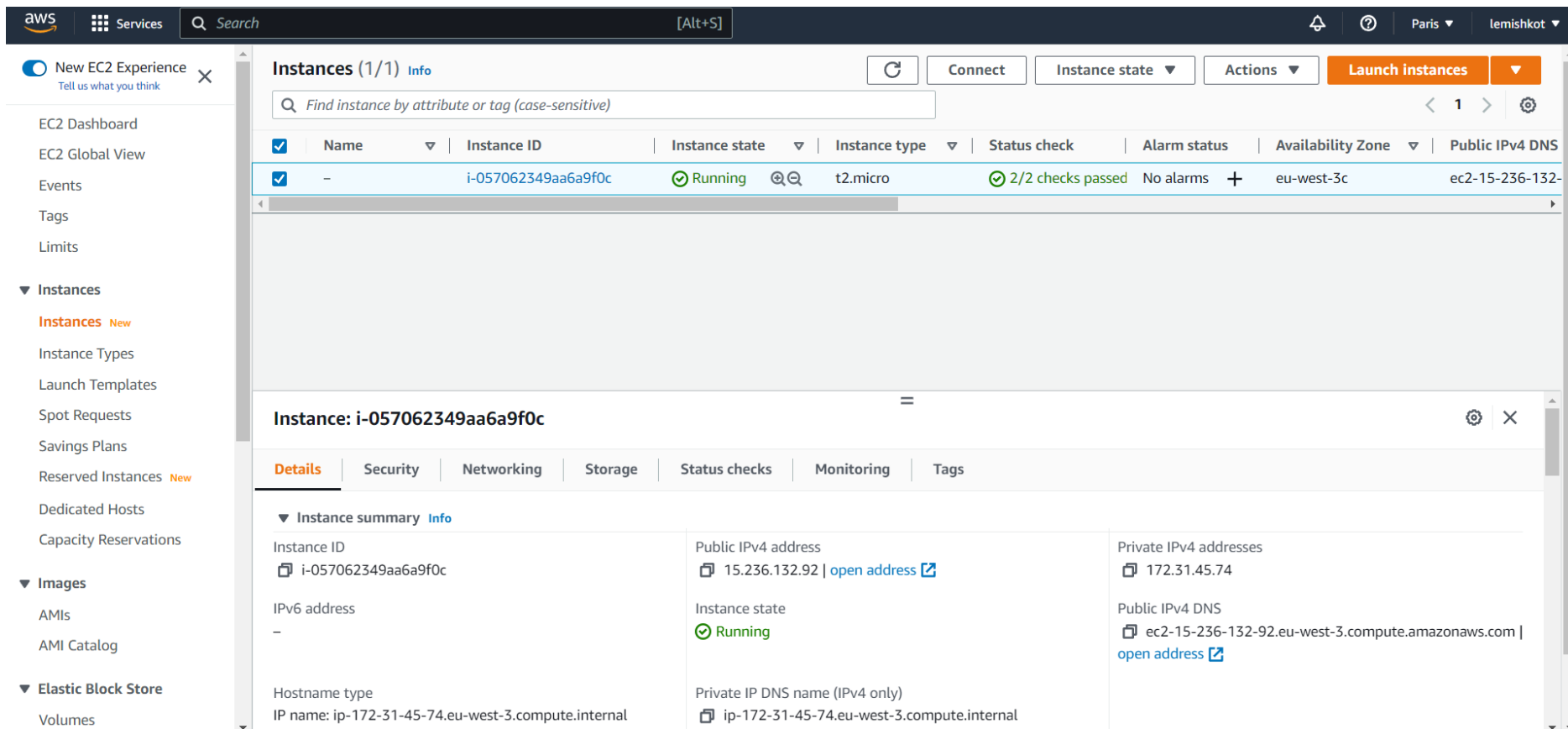
1. Tout d'abord, on a créé le bucket S3 sur AWS appelé "test-bucket-lemishkot", qui contient 2 objets :
  - le dossier 'sample', qui est utilisé pour stocker les images initiales
  - le dossier 'results\_parquet', qui contient le jeu de données résultant

The screenshot shows the Amazon S3 console interface for the bucket 'test-bucket-lemishkot'. The breadcrumb navigation at the top reads 'Amazon S3 > Buckets > test-bucket-lemishkot'. Below the bucket name, there are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected, showing a list of objects. Above the list, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar with the placeholder 'Find objects by prefix' is also present. The object list has columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. Two folders are listed: 'results\_parquet/' and 'sample/'.

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	results_parquet/	Folder	-	-	-
<input type="checkbox"/>	sample/	Folder	-	-	-

# Architecture Big Data. S3 <=> EC2

2. On a créé une instance EC2 sur AWS. Une instance EC2 est un serveur virtuel dans Elastic Compute Cloud (EC2) d'Amazon pour exécuter des applications sur l'infrastructure Amazon Web Services (AWS).



The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information for 'Paris' and 'lemishkot'. The left sidebar shows the 'Instances' section expanded, with 'Instances' highlighted. The main content area shows a list of instances with one instance, 'i-057062349aa6a9f0c', in a 'Running' state. Below the list, the details for this instance are shown, including its ID, public IPv4 address (15.236.132.92), private IPv4 address (172.31.45.74), and public IPv4 DNS name (ec2-15-236-132-92.eu-west-3.compute.amazonaws.com).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-057062349aa6a9f0c	Running	t2.micro	2/2 checks passed	No alarms	eu-west-3c	ec2-15-236-132-

Instance: i-057062349aa6a9f0c		
Details	Security	Networking
<strong>Instance summary</strong>		
Instance ID i-057062349aa6a9f0c	Public IPv4 address 15.236.132.92   <a href="#">open address</a>	Private IPv4 addresses 172.31.45.74
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-15-236-132-92.eu-west-3.compute.amazonaws.com   <a href="#">open address</a>
Hostname type IP name: ip-172-31-45-74.eu-west-3.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-45-74.eu-west-3.compute.internal	

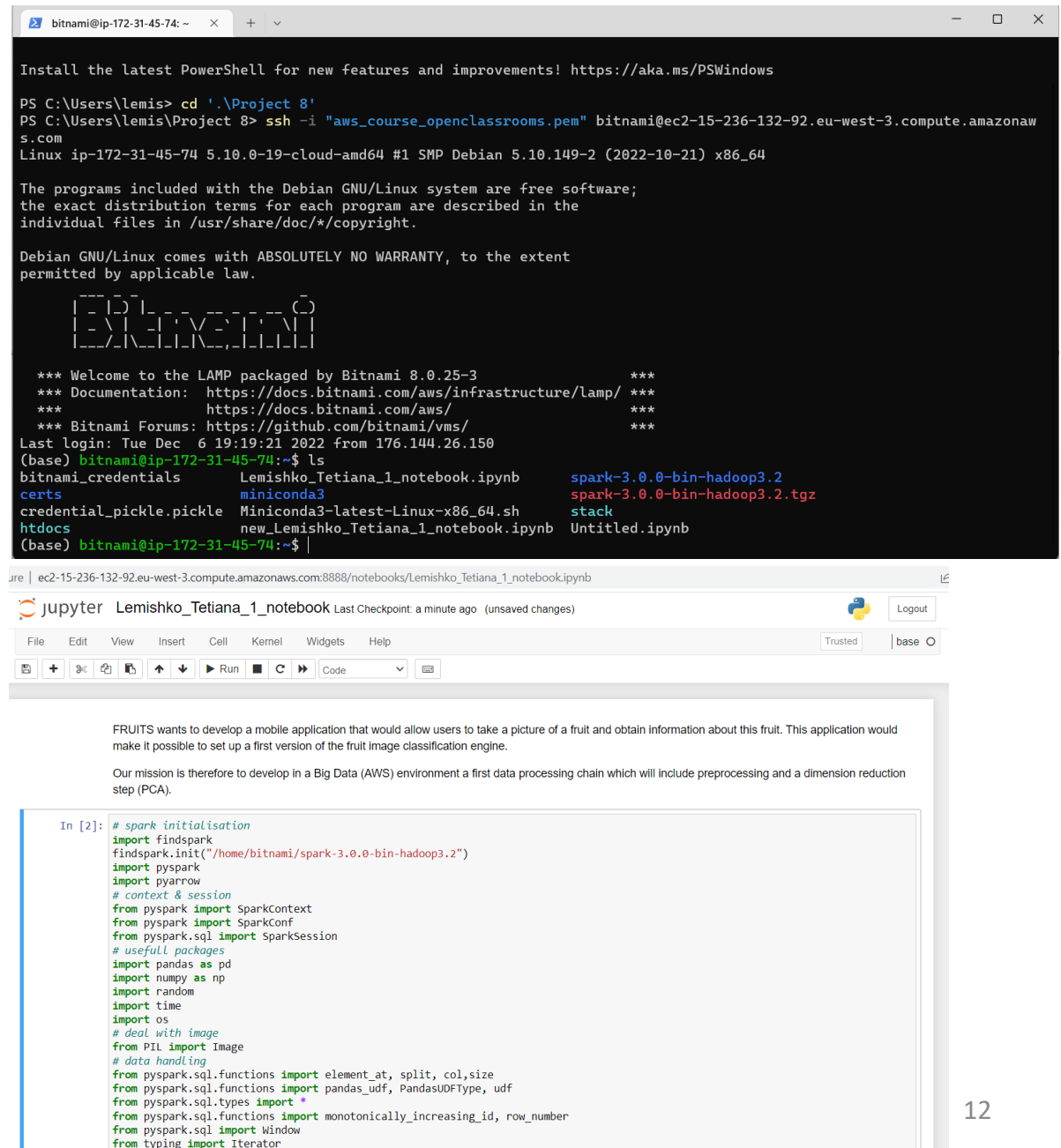
3. Ensuite, on a créé le rôle IAM et les clés (le fichier contenant les clés a été enregistré sur l'EC2) afin de fournir à EC2 l'accès à S3

# Architecture Big Data. PowerShell => EC2

4. On a utilisé la console PowerShell afin de se connecter à cette instance EC2

5. Puis, on a installé plusieurs packages pour créer l'environnement de travail (Miniconda (Anaconda), Jupyter notebook, Spark etc.)

6. On a créé le notebook jupyter sur EC2 pour exécuter le code du traitement des données



The image shows a terminal window and a Jupyter Notebook interface. The terminal window displays the command prompt for a Bitnami Linux instance on an Amazon EC2 machine. It shows the installation of PowerShell, the connection to the instance via SSH, and the installation of various packages including Miniconda, Jupyter, and Spark. The Jupyter Notebook interface shows the code for Spark initialization and data handling.

```
bitnami@ip-172-31-45-74: ~  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
PS C:\Users\lemis> cd '.\Project 8'  
PS C:\Users\lemis\Project 8> ssh -i "aws_course_openclassrooms.pem" bitnami@ec2-15-236-132-92.eu-west-3.compute.amazonaws.com  
Linux ip-172-31-45-74 5.10.0-19-cloud-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
  
*** Welcome to the LAMP packaged by Bitnami 8.0.25-3 ***  
*** Documentation: https://docs.bitnami.com/aws/infrastructure/lamp/ ***  
*** https://docs.bitnami.com/aws/ ***  
*** Bitnami Forums: https://github.com/bitnami/vms/ ***  
Last login: Tue Dec 6 19:19:21 2022 from 176.144.26.150  
(base) bitnami@ip-172-31-45-74:~$ ls  
bitnami_credentials Lemishko_Tetiana_1_notebook.ipynb spark-3.0.0-bin-hadoop3.2  
certs miniconda3 spark-3.0.0-bin-hadoop3.2.tgz  
credential_pickle.pickle Miniconda3-latest-Linux-x86_64.sh stack  
htdocs new_Lemishko_Tetiana_1_notebook.ipynb Untitled.ipynb  
(base) bitnami@ip-172-31-45-74:~$
```

Jupyter Notebook: Lemishko\_Tetiana\_1\_notebook  
Last Checkpoint: a minute ago (unsaved changes)  
File Edit View Insert Cell Kernel Widgets Help  
Trusted base O  
Run Code  
In [2]: # spark initialisation  
import findspark  
findspark.init("/home/bitnami/spark-3.0.0-bin-hadoop3.2")  
import pyspark  
import pyarrow  
# context & session  
from pyspark import SparkContext  
from pyspark import SparkConf  
from pyspark.sql import SparkSession  
# usefull packages  
import pandas as pd  
import numpy as np  
import random  
import time  
import os  
# deal with image  
from PIL import Image  
# data handling  
from pyspark.sql.functions import element\_at, split, col, size  
from pyspark.sql.functions import pandas\_udf, PandasUDFType, udf  
from pyspark.sql.types import \*  
from pyspark.sql.functions import monotonically\_increasing\_id, row\_number  
from pyspark.sql import Window  
from typing import Iterator

# Technologies utilisées

- **SPARK (Pyspark)**



Framework open source de calcul distribué (pour la parallélisation des calculs – Pyspark = API python)

- **Boto3**



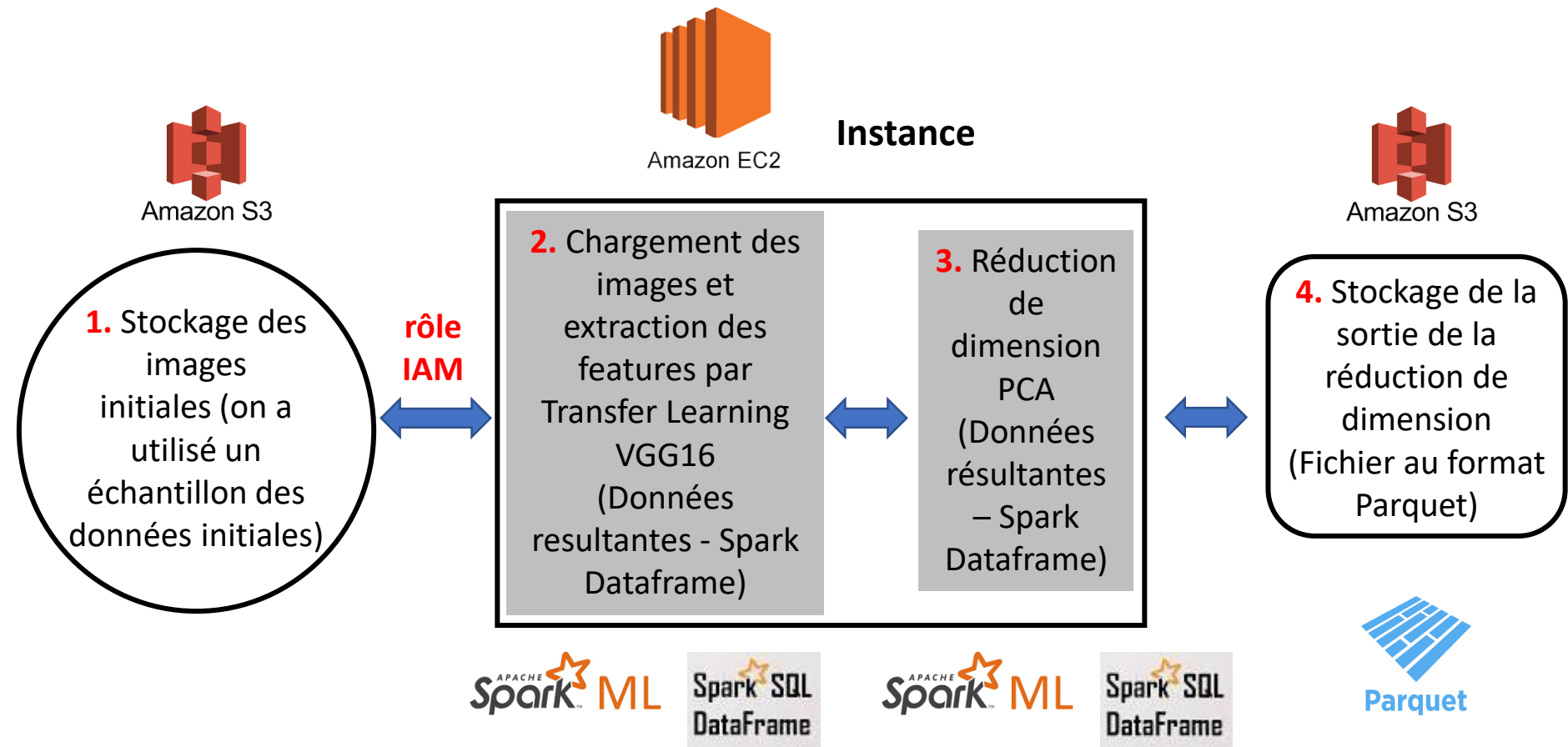
SDK (software development kit) pour accéder au bucket S3 afin d'effectuer des opérations de lecture et écriture de fichiers

- **PARQUET**



Format de fichier pour les données massives

# Chaîne de traitement



# Les principales étapes du traitement des images

## Extraction des features par Transfer Learning VGG16

VGG16 est un réseau neuronal convolutif, qui comprend 16 couches profondes. On peut charger une version pré-entraînée du réseau formée sur plus d'un million d'images de 1000 catégories différentes à partir de la base de données ImageNet.

path	modificationTime	length	content	label	features
s3a://test-bucket...	2022-11-28 14:18:56	3643	[FF D8 FF E0 00 1...	Banana	[0.49548423290252...
s3a://test-bucket...	2022-11-28 14:18:57	3583	[FF D8 FF E0 00 1...	Banana	[0.49772876501083...
s3a://test-bucket...	2022-11-28 14:18:57	3574	[FF D8 FF E0 00 1...	Banana	[0.49324735999107...
s3a://test-bucket...	2022-11-28 14:18:57	3484	[FF D8 FF E0 00 1...	Banana	[0.49358439445495...
s3a://test-bucket...	2022-11-28 14:18:57	3425	[FF D8 FF E0 00 1...	Banana	[0.52460342645645...
s3a://test-bucket...	2022-11-28 14:18:58	3408	[FF D8 FF E0 00 1...	Banana	[0.52716338634490...
s3a://test-bucket...	2022-11-28 14:18:58	3348	[FF D8 FF E0 00 1...	Banana	[0.53755146265029...
s3a://test-bucket...	2022-11-28 14:18:58	3348	[FF D8 FF E0 00 1...	Banana	[0.54704099893569...
s3a://test-bucket...	2022-11-28 14:18:58	3235	[FF D8 FF E0 00 1...	Banana	[0.56857973337173...
s3a://test-bucket...	2022-11-28 14:18:58	3233	[FF D8 FF E0 00 1...	Banana	[0.59479212760925...

## Réduction de dimension PCA

Méthode utilisée en réduction de dimension, qui cherche à représenter les données dans un sous-espace de plus petite dimension de sorte à conserver au maximum la variance du nuage de données.

Out[40]:

path	modificationTime	length	content	label	features	vectors	pca_vectors	pca_features
s3a://test-bucket...	2022-11-28 14:18:56	3643	[FF D8 FF E0 00 1...	Banana	[0.49548423290252...	[0.49548423290252...	[-1.8084739556665...	[-1.808474, -0.32...
s3a://test-bucket...	2022-11-28 14:18:57	3583	[FF D8 FF E0 00 1...	Banana	[0.49772876501083...	[0.49772876501083...	[-1.7688524128342...	[-1.7688525, -0.3...
s3a://test-bucket...	2022-11-28 14:18:57	3574	[FF D8 FF E0 00 1...	Banana	[0.49324735999107...	[0.49324735999107...	[-1.8162089152568...	[-1.816209, -0.44...
s3a://test-bucket...	2022-11-28 14:18:57	3484	[FF D8 FF E0 00 1...	Banana	[0.49358439445495...	[0.49358439445495...	[-1.8882339493855...	[-1.8882339, -0.5...
s3a://test-bucket...	2022-11-28 14:18:57	3425	[FF D8 FF E0 00 1...	Banana	[0.52460342645645...	[0.52460342645645...	[-2.0264109276818...	[-2.0264108, -0.6...

# Conclusion

On s'est familiarisé avec l'environnement Big Data. On a développé dans un environnement Big Data une première chaîne de traitement des données qui comprend le preprocessing et une étape de réduction de dimension.

- Enseignements
  - Prise en main Pyspark
  - Découverte du format distribué parquet
  - Découverte de l'écosystème AWS
- Difficultés rencontrées
  - Nombreuses possibilités techniques : choix complexes
  - Débug complexe dû à des erreurs peu explicites (superposition Spark/Java/S3)