

Projet 8 : Déployez un modèle dans le cloud

Ilham NOUMIR | Parcours Data Science | Date: 25/03/2022

Sommaire

1. Présentation de la problématique et du jeu de données
2. Pourquoi un environnement Big Data ?
3. Les éléments de l'architecture choisie et leurs rôles
4. les étapes de la chaîne de traitement
5. Conclusion



Fruits!

1. Présentation de la problématique et du jeu de donnée

Description de l'organisme :

La start-up "Fruits!" est une jeune entreprise qui travaille dans le domaine de l'AgriTech

Contexte :

L'entreprise souhaite 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

Missions :

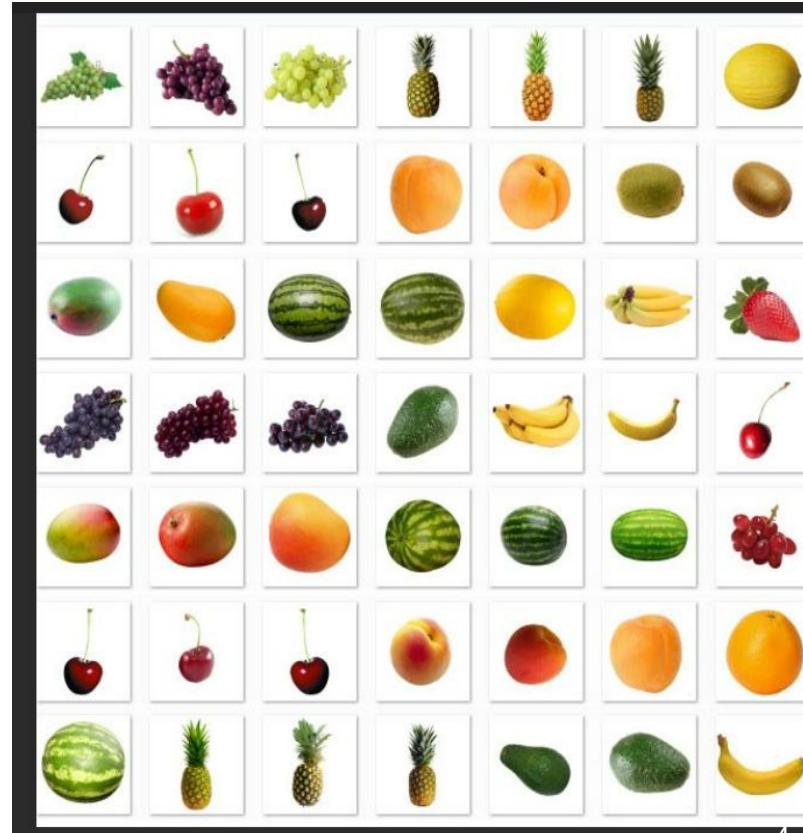
- Développement d'un environnement Big Data
- Réalisation d'une première chaîne de traitement des données :
(preprocessing + réduction de dimension)

1. Présentation de la problématique et du jeu de donnée

Jeu de données constitué des images de fruits et des labels associés

Propriétés de l'ensemble de données:

- Le nombre total d'images : 90483
- Taille de l'ensemble d'entraînement : 67692 images
- Taille de l'ensemble de test : 22688 images
- Le nombre de classes : 131 (fruits et légumes).
- Taille de l'image : 100x100 pixels.



2. Aperçu sur le Big Data :

Contexte actuel :

- les données sont générées rapidement par plusieurs sources distribuées et hétérogènes.
- La majorité des technologies traditionnelles ne sont plus adéquates pour prendre en charge cette masse de données

Le Big Data est né au moyen de la fusion de diverses sources de données telles que :

- L'utilisation d'Internet sur les mobiles
- Les réseaux sociaux
- La géolocalisation
- Le cloud
- La mesure des données vitales
- Le streaming des médias

2. Pourquoi un environnement Big Data ?

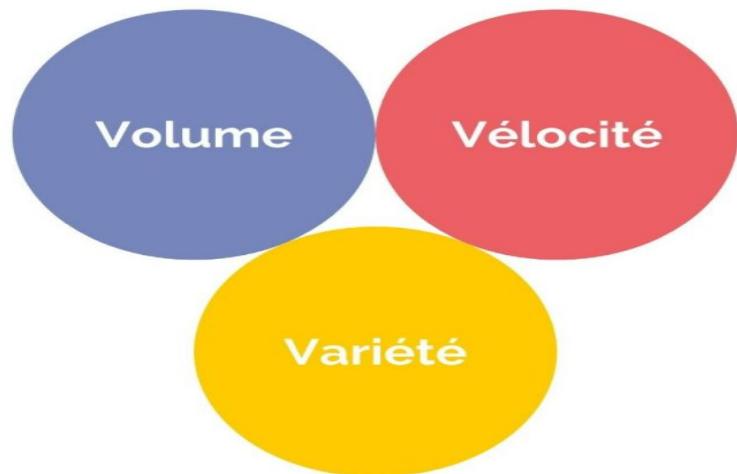
le passage à l'échelle s'accompagne quasiment toujours d'une transformation des usages que l'on résume par les 3V du big data : **Volume, Vélocité, Variété**.

Volume des données générées nécessite de repenser la manière dont elles sont stockées.

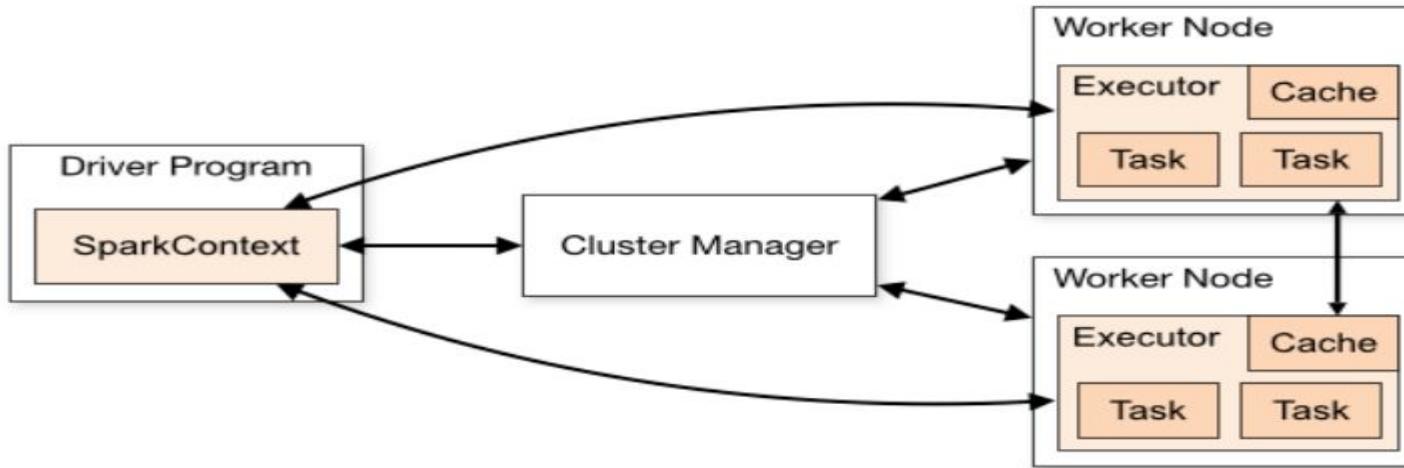
Vélocité à laquelle nous parvennent ces données implique de mettre en place des solutions de traitement en temps réel qui ne paralysent pas le reste de l'application.

Variété de données sous différents formats :

- ❖ structurées (documents JSON),
- ❖ semi-structurées (fichiers de log)
- ❖ non structurées (textes, images)



2. Calculs distribués

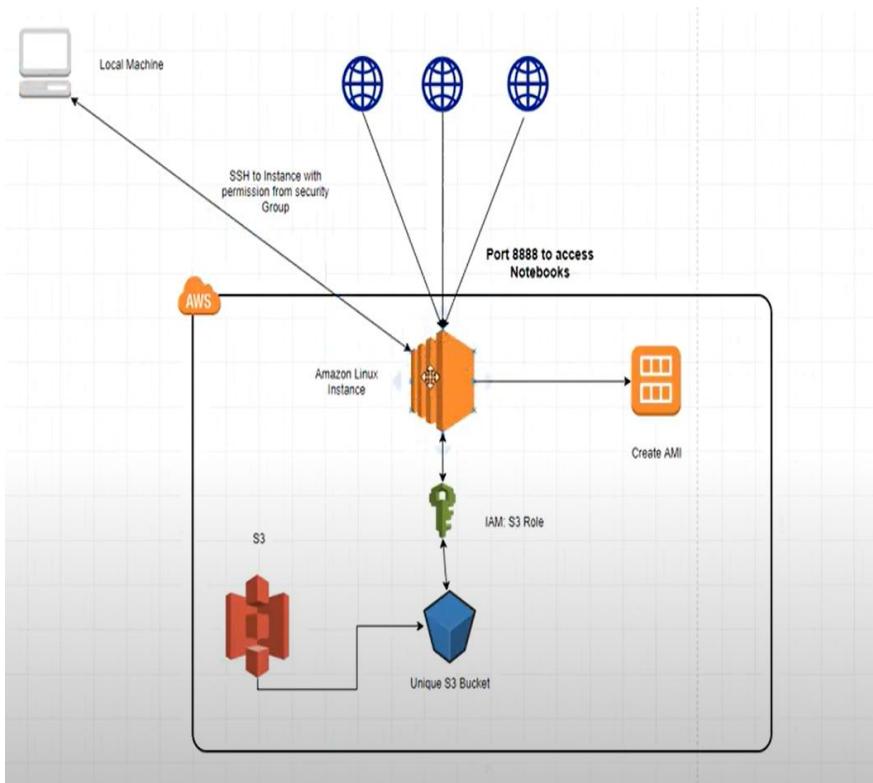


-
- **Application maître :**
- Configuration /
- Initialisation /
- Agrégation des calculs
-

-
- **Cluster Manager :**
- Gestion des ressources
- Distribution des calculs
- entre les workers
-

-
- **Workers :**
- Exécution des tâches en
- parallèle
-

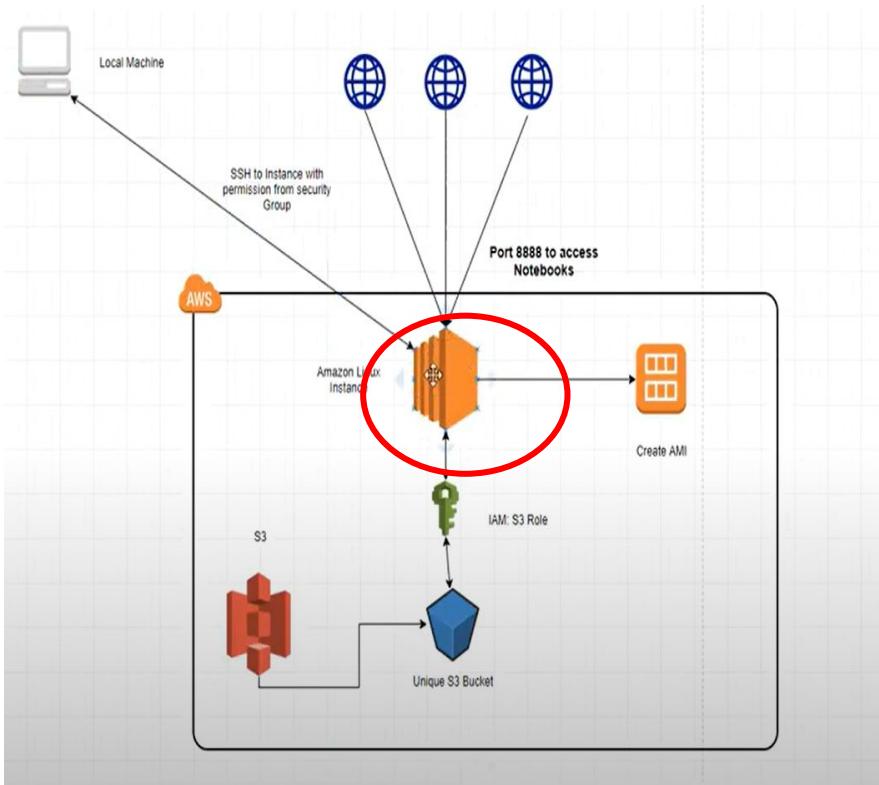
3. Les éléments de l'architecture choisie et leurs rôles



EC2 : Elastic Compute Cloud est un service de calcul élastique dans le cloud

S3: Simple Storage Service est un service de stockage et de distribution des fichiers

3. Les éléments de l'architecture choisie et leurs rôles



Instance t2.large avec un noyau Ubuntu Server 18

Capacité du disque : 30 GB

Configuration du rôle AMI

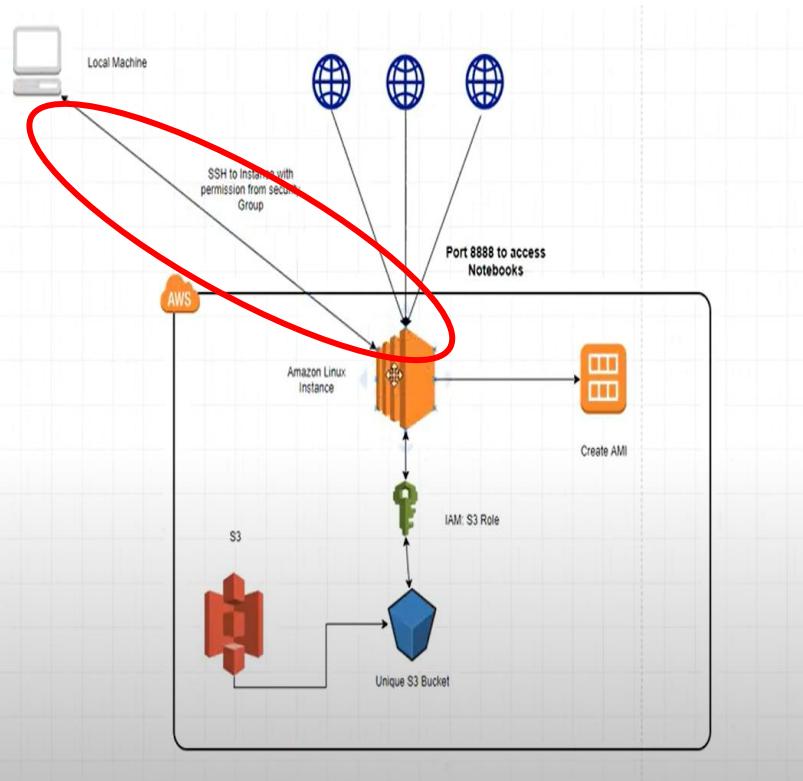
Configuration du groupe de sécurité

Port : 22

Port : 8888

Port : 4040

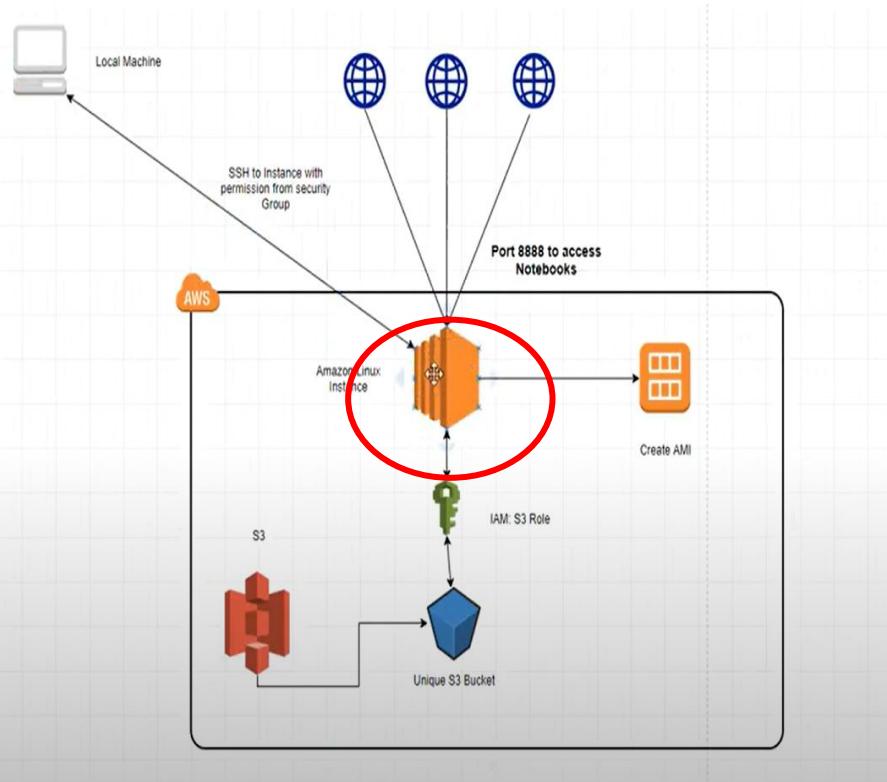
3. Les éléments de l'architecture choisie et leurs rôles



Utilisation d'un tunnel SSH : Logiciel Putty pour se connecter en SSH

Utilisation de PuttyGen pour transformer la clé .pem donné par AWS lors de la création du serveur en clé .ppk pour accéder en local et faire les installations nécessaires

3. Les éléments de l'architecture choisie et leurs rôles



Anaconda 3



python : 3.9.7

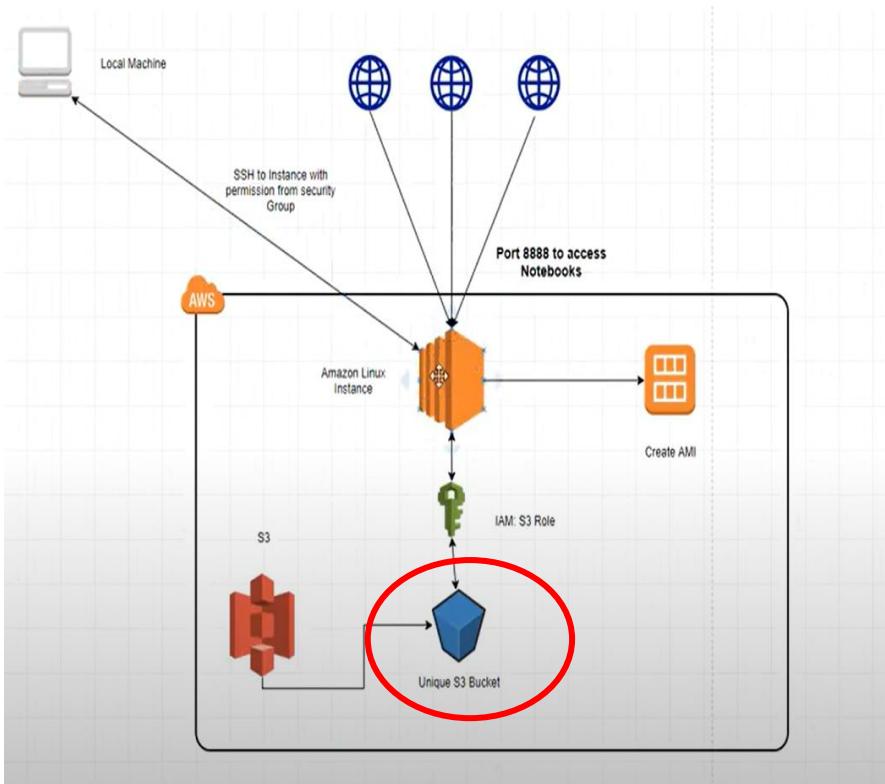


Java : 1.8



3.0.3

3. Les éléments de l'architecture choisie et leurs rôles



Input : Téléchargement des dossiers des images
Un dossier ↔ Un fruit

Output : Fichier CSV contenant la sortie de la réduction de dimension

Résumé des étapes de mise en oeuvre :

1

Création de la zone
de stockage sur S3

2

Configuration de
l'environnement de
travail

3

Preprocessing et
réduction
dimensionnelle

4

Stockage de la sortie
de la réduction
dimensionnelle sur
s3

4. Les étapes de la chaîne de traitement :

Création du
spark
Context

Chargement
des données

Récupération
des labels des
fruits à partir
du chemin
d'accès

Preprocessing
sur les images

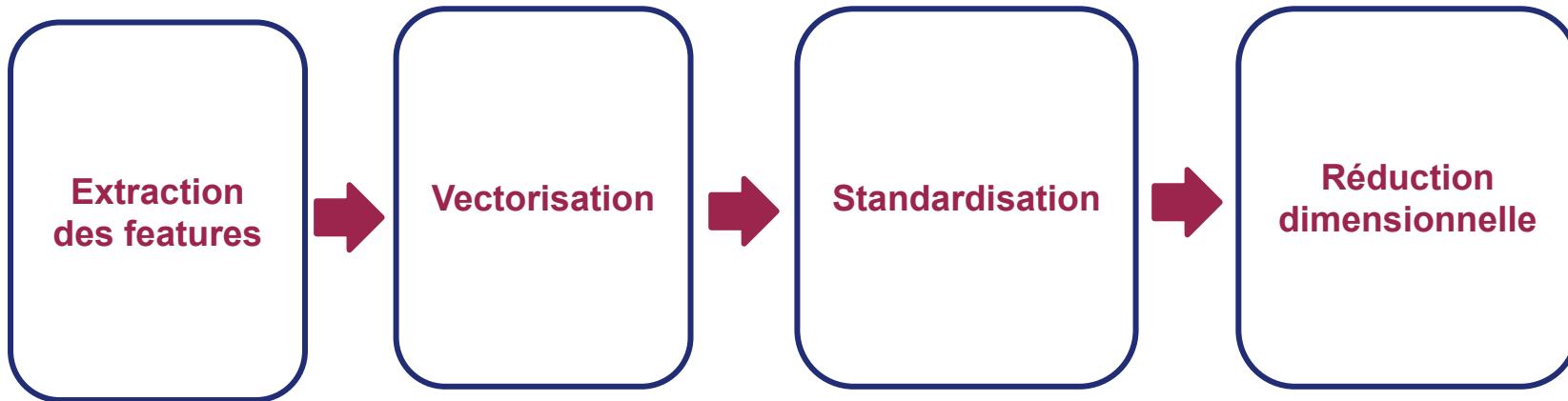
sparkContext

spark.read.format
Création d'un
RDD

Utilisation de la
fonction split()

Passage de la
forme binaire à
une array

4. Les étapes de la chaîne de traitement :



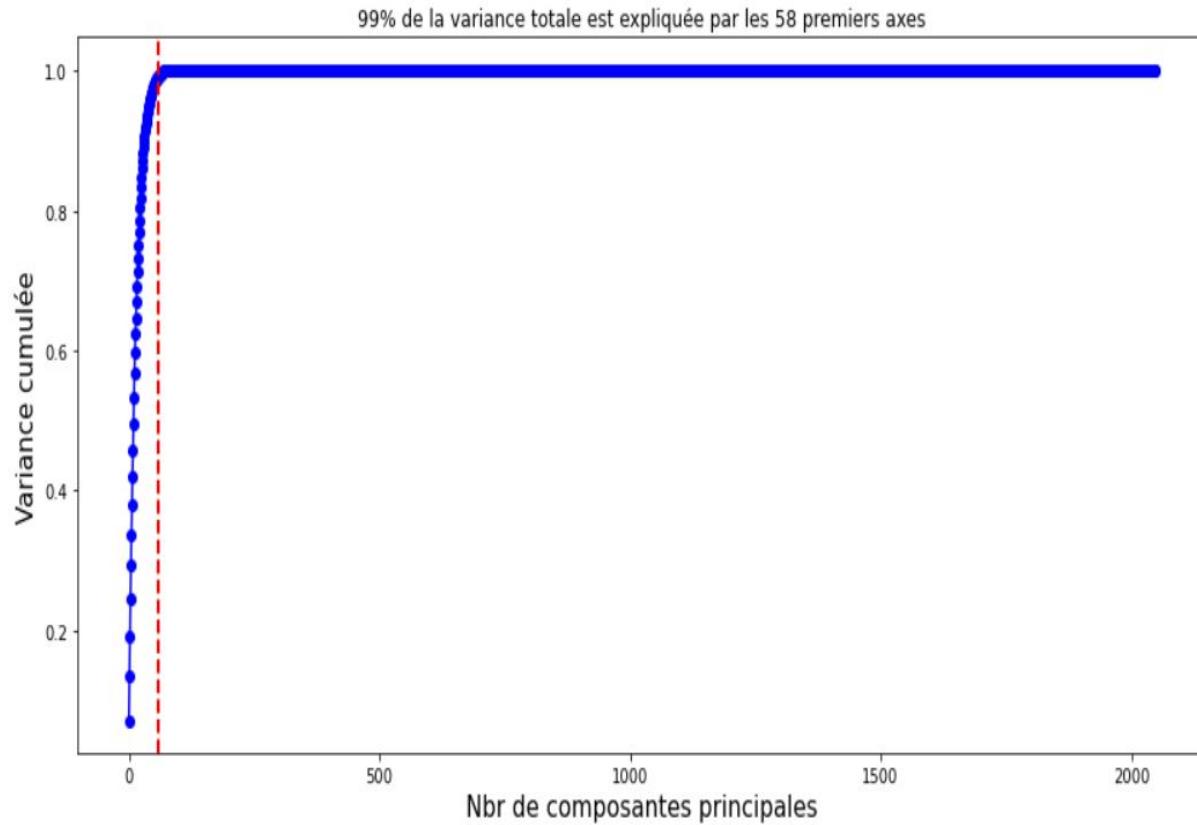
ResNet50 pré entraîné sur imagenet

VectorUDT

StandarScaler

PCA

4. Les étapes de la chaîne de traitement : Réduction dimensionnelle



Réduction dimensionnelle :
Variance expliquée à 99%
passage de 2048 features à 58
features en appliquant le PCA

4. Les étapes de la chaîne de traitement :

**Stockage sur le S3 :
sortie de la réduction de dimension**

		path	label	features_reduced
0	s3a://oc-in2-p8/data/pineable/r_1_100.jpg	pineable	[6.616042714100934, -21.302603642638083, -14.3...	
1	s3a://oc-in2-p8/data/pineable/r_99_100.jpg	pineable	[5.3196187671209785, -20.784546682094977, -15....	
2	s3a://oc-in2-p8/data/avocado/116_100.jpg	avocado	[7.6018415720586265, 27.45124192542639, -21.24...	
3	s3a://oc-in2-p8/data/avocado/r_142_100.jpg	avocado	[-0.7133216226531506, 14.988496899652775, -6.6...	
4	s3a://oc-in2-p8/data/banana/263_100.jpg	banana	[-34.79771715925994, 5.453574670287678, -11.30...	
...
75	s3a://oc-in2-p8/data/apple/r_278_100.jpg	apple	[1.9682519295549483, 2.9817454770611613, 16.75...	
76	s3a://oc-in2-p8/data/apple/177_100.jpg	apple	[3.06477380269612, 6.0157417000212945, 8.79574...	
77	s3a://oc-in2-p8/data/avocado/r_31_100.jpg	avocado	[4.343664698511755, 13.891744926758832, -4.564...	
78	s3a://oc-in2-p8/data/avocado/183_100.jpg	avocado	[12.48138100539999, 31.632679933735464, -28.60...	
79	s3a://oc-in2-p8/data/banana/89_100.jpg	banana	[-43.02436169331235, 2.467010991633207, -17.66...	

80 rows × 3 columns

5. Comment passer à l'échelle :

- Aucune modification du code Spark/Python à apporter
- Le stockage des fichiers se fera sur S3
- Nous pouvons prendre une instance EC2 de plus grande capacité RAM/Processeur
- On peut aussi utiliser plusieurs instances et fixer une instance comme le principal

Conclusion :

Difficultés rencontrées :

- Découverte de l'API pyspark
- Découverte de l'écosystème AWS
- Administration d'un serveur Linux par SSH
- Débug complexe dû à des erreurs peu explicites (superposition Spark/Java/S3)