

Projet 8 :

Déployez un modèle dans le cloud

Nom : TRABIS
Prénom : Mohamed

Table des matières

1. Introduction
2. Le jeu de données
3. Introduction « Big Data »
4. AWS – SPARK
5. Architecture « Big Data » - Chaîne de traitement
6. Conclusion et recommandations

Introduction

Introduction

► Contexte :

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

La volonté de l'entreprise est de préserver la biodiversité des fruits en permettant des traitements spécifiques pour chaque espèce de fruits en développant des robots cueilleurs intelligents.

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.

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



Fruits!

Introduction

● 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.

Il n'est pas nécessaire d'entraîner un modèle pour le moment.

● Contraintes:

Ils existent plusieurs contraintes:

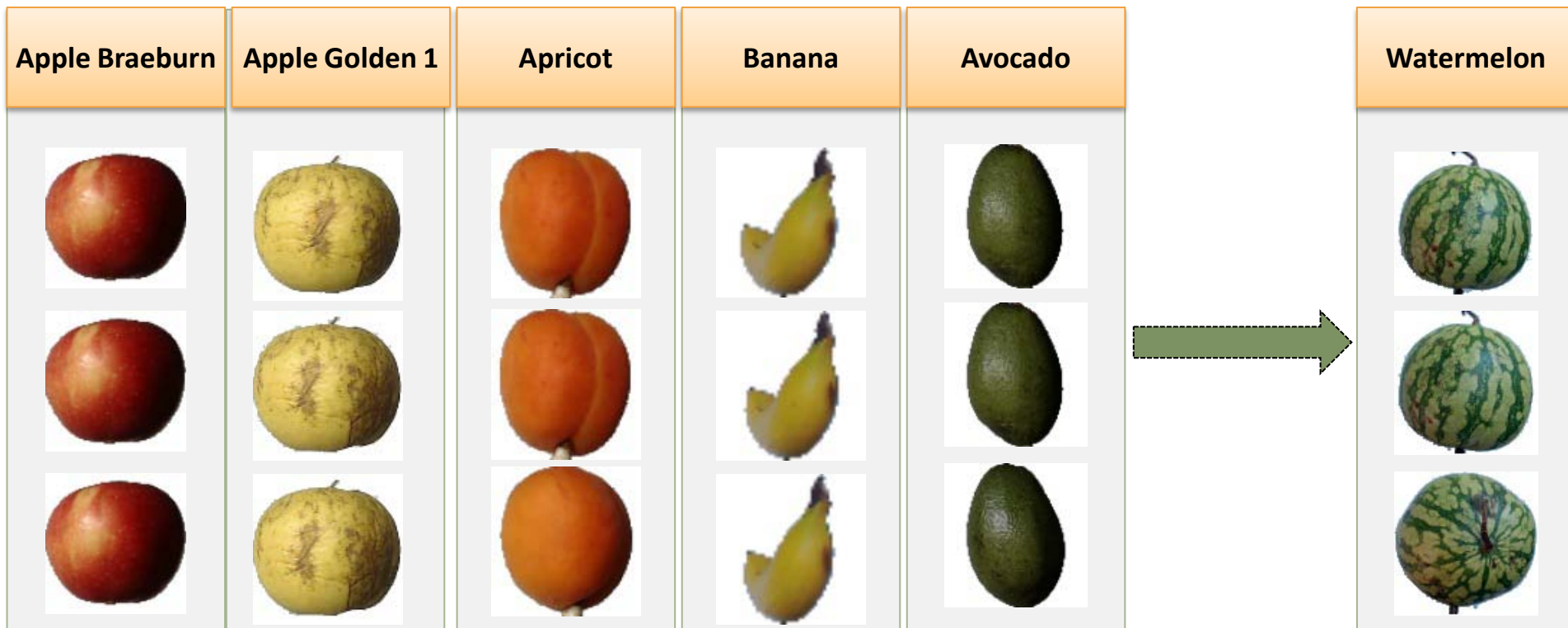
1. IL faut tenir compte dans les développements du fait que le volume de données va augmenter très rapidement après la livraison de ce projet. Vous développerez donc des scripts en Pyspark et utiliserez par exemple le cloud AWS pour profiter d'une architecture Big Data (EC2, S3, IAM), basée sur un serveur EC2 Linux.
2. La mise en œuvre d'une architecture Big Data sous AWS peut nécessiter une configuration serveur plus puissante que celle proposée gratuitement (EC2 = t2.micro, 1 Go RAM, 8 Go disque serveur).

Le jeu de données

Le jeu de données - Évaluation et découverte des données

• Base de données :

- ☐ Le jeu de données constitué des images de fruits et des labels associés, qui pourra servir de point de départ pour construire une partie de la chaîne de traitement des données.
- ☐ Ci-dessous un exemple des images par catégorie (environ 131 catégories):

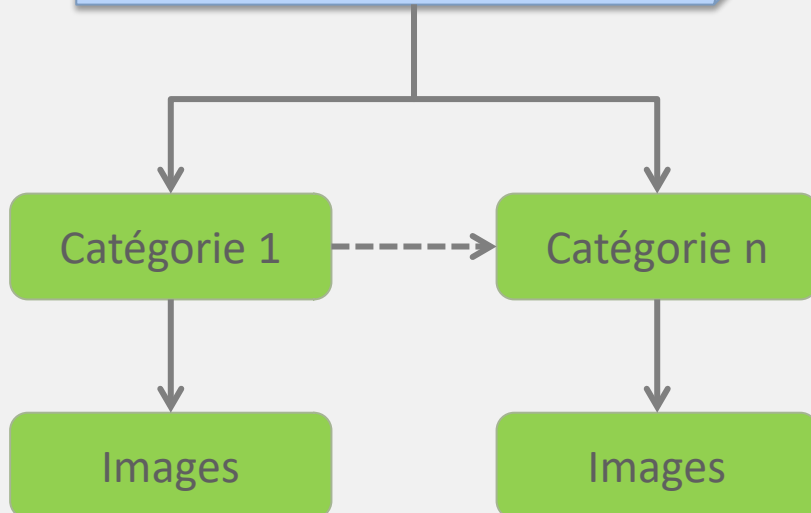


Le jeu de données - Évaluation et découverte des données

Le schéma des images :

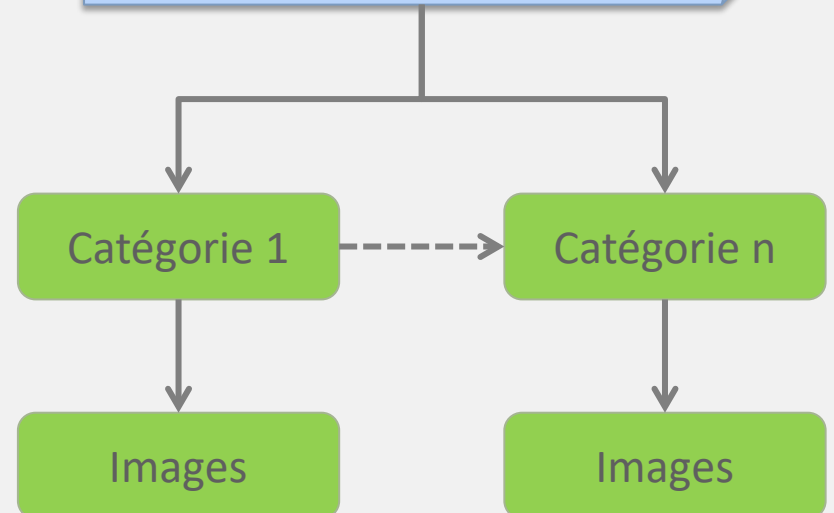
Images test

Nombre d'images : 22688
Nombre de catégories: 131
Taille de l'image : 100x100



Images d'entraînement

Nombre d'images : 90483
Nombre de catégories : 131
Taille de l'image : 100x100



Introduction « Big Data »

Introduction Big Data – Spark vs Hadoop

• Big Data :

Pour analyser les données massives (Big Data), il est nécessaire de s'équiper de meilleurs outils analytiques.

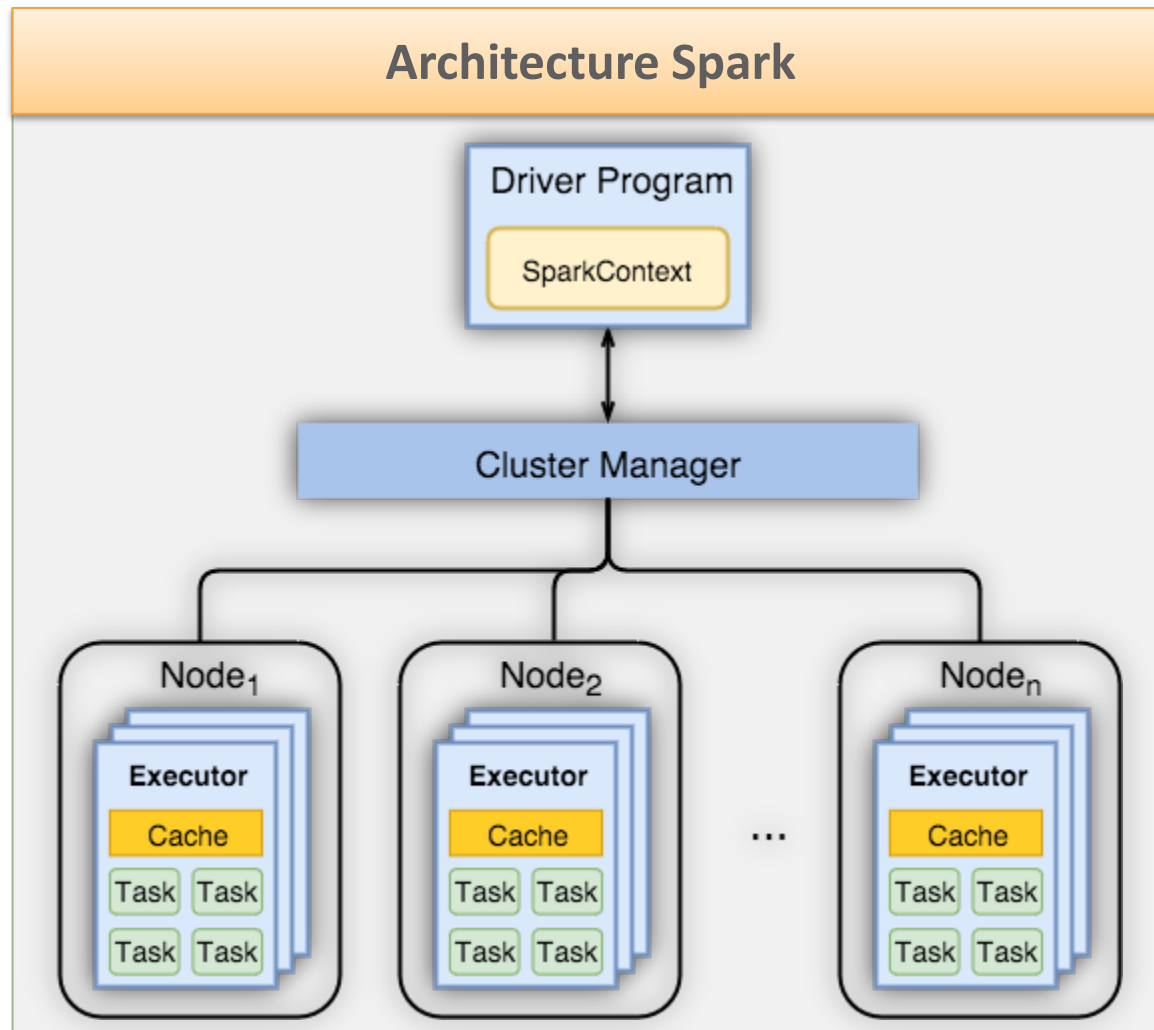
Parmi les outils les plus populaires nous allons comparer **Hadoop** et **Spark** :

		
Traitement des données	Traitement en mode batch (Latence élevée)	Traitement interactif (Latence Faible)
Rapidité	Rapide	10 à 100 fois plus rapide
Facilité d'utilisation	Facile	Très facile
Ressource Manager	ex : YARN	ex : Standalone
Planificateur de tâches	Externe(Oozie)	En mémoire

• Remarque : Notre choix va se porter sur « **Spark** », vu que ses performances sont meilleures que celles d'« **Hadoop** ».

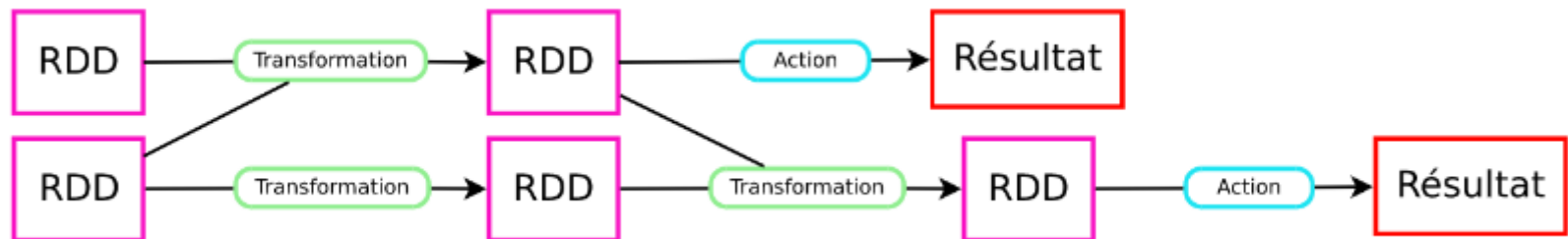
Introduction Big Data – Architecture Spark

- Spark utilise l'architecture Maître / Esclave :



Introduction Big Data – Architecture Spark

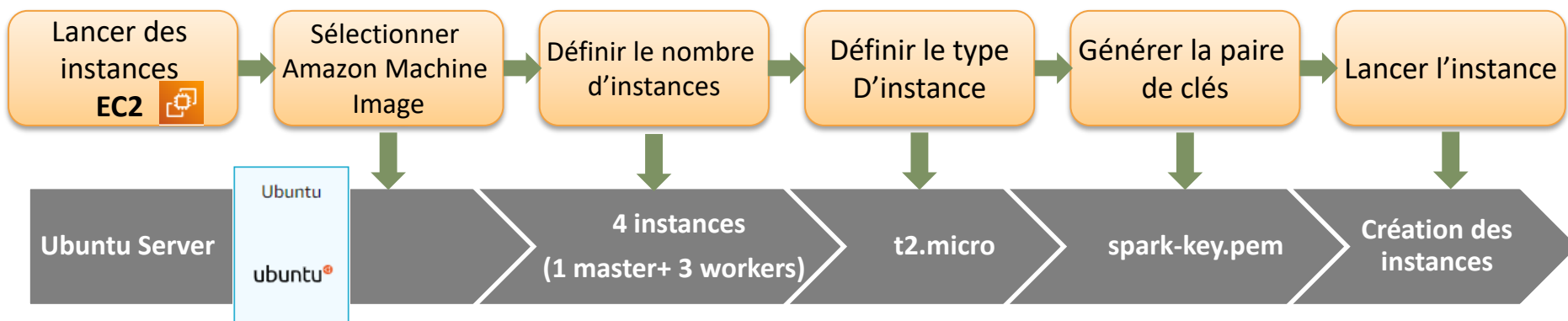
- **Caractéristiques de Spark** : Spark est connu pour avoir plusieurs caractéristiques
 - **Performance de traitement** (jusqu'à 100x meilleur que Hadoop Map Reduce)
 - **Tolérance aux Fautes.**
 - **Traitements à la volée.**
 - **Support de plusieurs langages.**
 - **Intégration avec Hadoop.**
- **Resilient Distributed Datasets (RDD)** : est le cœur de Spark pour réaliser des opérations MapReduce plus rapides et efficaces.



AWS - SPARK

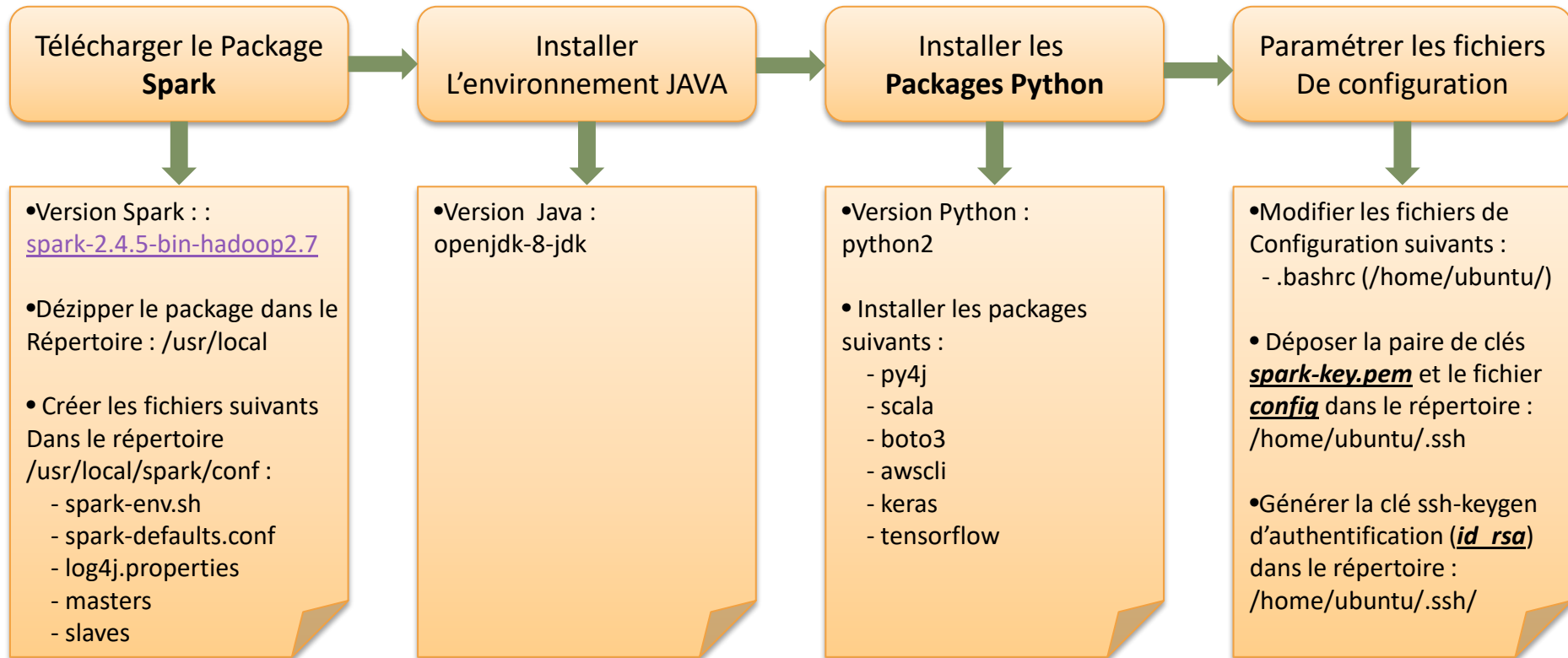
AWS – SPARK – Instance EC2

- Les étapes effectuées pour la création des instances EC2 dans AWS :



AWS – SPARK – Installation de Spark

Le processus d'installation de « Spark » dans EC2 de AWS :

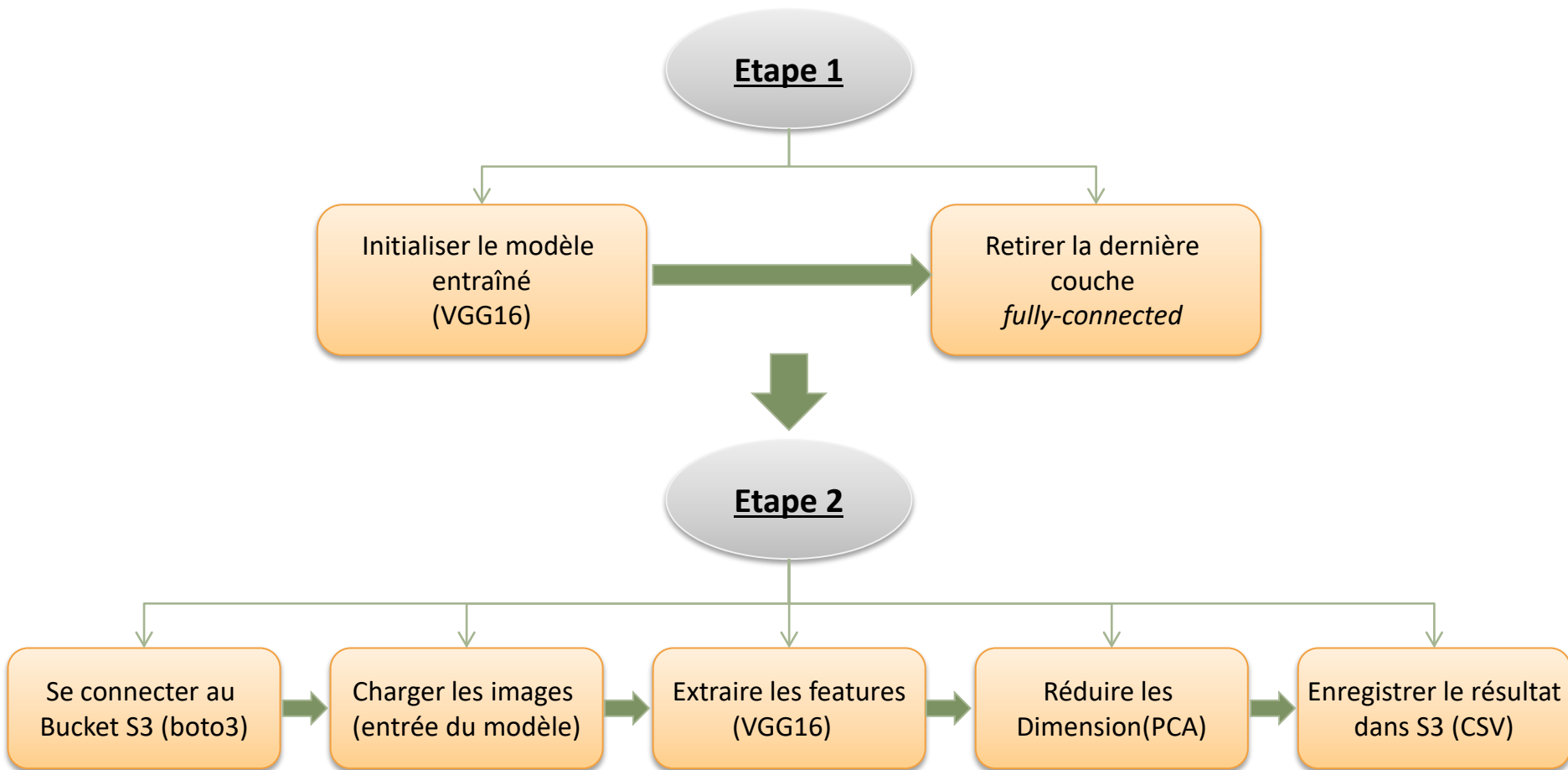


• **Remarque** : Ce processus d'installation doit être effectué pour chaque instance EC2 (1 master + 3 workers)

Architecture « Big Data » - Chaîne de traitement

Architecture « Big Data » – Chaîne de traitement

- Le processus de la chaîne de traitement des données comprend plusieurs étapes :



Architecture « Big Data » – Exemple de traitement

- Ci-dessous un exemple des résultats générés lors du traitement en lançant le script python :

Extraction des VGG16 Features		path_img	label	vgg_features
		data2/Test/Apple ...	Apple Braeburn	[0.37030035257339...
		data2/Test/Apple ...	Apple Braeburn	[0.37430453300476...
		data2/Test/Avocad...	Avocado	[0.54048901796340...
		data2/Test/Avocad...	Avocado	[0.53512614965438...
		data2/Training/Ap...	Apple Braeburn	[0.38282635807991...
		data2/Training/Ap...	Apple Braeburn	[0.36382997035980...
		data2/Training/Ap...	Apple Braeburn	[0.36087590456008...
		data2/Training/Av...	Avocado	[0.57190960645675...
		data2/Training/Av...	Avocado	[0.56644451618194...
Réduction de Dimension (PCA)		path_img	label	pca_vectors
		data2/Test/Apple ...	Apple Braeburn	[-4.1404424, -0.5...
		data2/Test/Apple ...	Apple Braeburn	[-4.276082, -0.87...
		data2/Test/Avocad...	Avocado	[-6.214579, -0.51...
		data2/Test/Avocad...	Avocado	[-6.2330575, -0.5...
		data2/Training/Ap...	Apple Braeburn	[-4.1702805, -0.4...
		data2/Training/Ap...	Apple Braeburn	[-4.148466, -0.48...
		data2/Training/Ap...	Apple Braeburn	[-4.144001, -0.49...
		data2/Training/Av...	Avocado	[-6.2382393, -0.6...
		data2/Training/Av...	Avocado	[-6.195516, -0.60...

Architecture « Big Data » – Spark master

- Après avoir démarré Spark « **master** » et « **slaves** » nous allons avoir accès à la page web d'administration spark (port 8080) avec l'url suivante : <http://master-public-dns-name:8080/>



Spark Master at spark://ip-172-31-19-187.eu-west-1.compute.internal:7077

URL: spark://ip-172-31-19-187.eu-west-1.compute.internal:7077

Alive Workers: 3

Cores in use: 3 Total, 0 Used

Memory in use: 3.0 GB Total, 0.0 B Used

Applications: 0 Running, 4 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory
worker-20220517124237-172.31.19.75-43999	172.31.19.75:43999	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20220517124237-172.31.31.248-36921	172.31.31.248:36921	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20220517124239-172.31.24.9-38809	172.31.24.9:38809	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications (0)

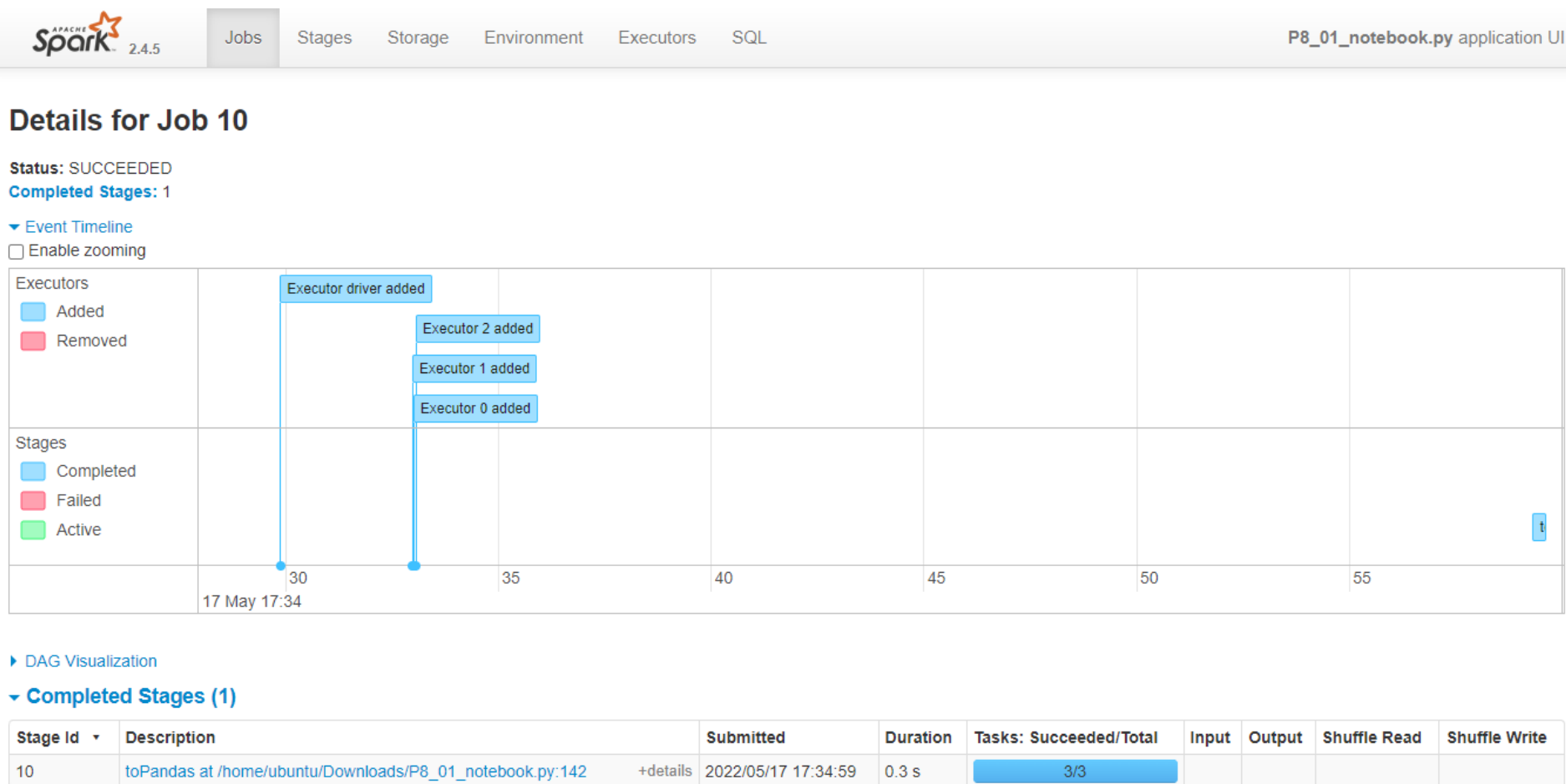
Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	----------------	------	-------	----------

Completed Applications (4)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20220517132229-0003	P8_01_notebook.py	3	1024.0 MB	2022/05/17 13:22:29	ubuntu	FINISHED	2.7 min
app-20220517131004-0002	P8_01_notebook.py	3	1024.0 MB	2022/05/17 13:10:04	ubuntu	FINISHED	9.1 min
app-20220517130355-0001	P8_01_notebook.py	3	1024.0 MB	2022/05/17 13:03:55	ubuntu	FINISHED	1.6 min
app-20220517130038-0000	P8_01_notebook.py	3	1024.0 MB	2022/05/17 13:00:38	ubuntu	FINISHED	19 s

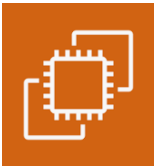



Architecture « Big Data » – Monitoring

- Ci-dessous la page de monitoring Spark (Spark History Server : <http://master-public-dns-name:18080/>) :



Architecture « Big Data » – Présentation des Outils

- Les outils utilisés pour la mise en place de l'architecture « Big Data » :

Outil	Description
	EC2 : <i>Amazon Elastic Compute Cloud</i> ou EC2 est un service proposé par Amazon permettant à des tiers de louer des serveurs sur lesquels exécuter leurs propres applications.
 Amazon S3	Amazon S3 : Amazon Simple Storage Service (Amazon S3) est un service de stockage d'objets qui offre une capacité de mise à l'échelle, une disponibilité des données, une sécurité et des performances de pointe.
	FileZilla : Permet de charger ou télécharger les fichiers sur un serveur distant. Il possède une interface utilisateur graphique intuitive.
	PuTTY : Est un émulateur de terminal pour Windows permettant la connexion à une machine distante par protocole <u>SSH</u> .

Conclusion et recommandations

Conclusion et recommandations

► Conclusions :

• Enseignements :

- Prise en main Pyspark et du MLlib
- Découverte de l'écosystème AWS (EC2 ,S3, RDS, EMR...)
- Administration et configuration d'un serveur Ubuntu par SSH
- Analyse et traitement des anomalies liées au serveur Ubuntu

• Difficultés rencontrées :

- Nombreuses possibilités techniques de configuration
- Choisir les bonnes versions des packages compatibles avec **Spark 2.4.5**
- Débug complexe dû à des erreurs peu explicites (Spark/Java/S3)

Conclusion et recommandations

► Recommandations :

• Pour aller plus loin :

- Affiner le prétraitement des images (recadrage, plusieurs fruits, arrière plan...)
- Entraîner le jeu de données avec plusieurs modèle de Transfer Learning.
- Déployer le modèle en production sur un cluster.
- Utiliser d'autres outils de Monitoring (ex : Glue)
- Utiliser AWS Auto Scaling qui ajuste automatiquement la capacité à maintenir des performances constantes et prévisibles de la manière la plus rentable possible.

Merci pour votre attention
Fin de la présentation