

Projet 8 : Déployez un modèle dans le Cloud

Etudiant: Eric Wendling

Mentor: Julien Heiduk

Date: 04/12/2020

---



**Fruits!**

Déploiement d'un modèle de classification d'images dans le Cloud



## Contexte

Développement de systèmes innovants pour la préservation de la biodiversité des fruits

- Robots cueilleurs intelligents
- Application mobile pour la reconnaissance de fruits

## Objectif

Développement d'un moteur de classification de fruits

- Augmentation rapide du volume de données
- Prototype architecture Big Data

## Livrable

Notebook Jupyter

- PySpark
- Etapes de traitement des données

# 1 Présentation du projet

Déploiement d'un modèle dans le Cloud



Fruits!

## Données

### Images de fruits

- Propriétés
  - Dimensions (pixels): 100 X 100
  - Profondeur: 24

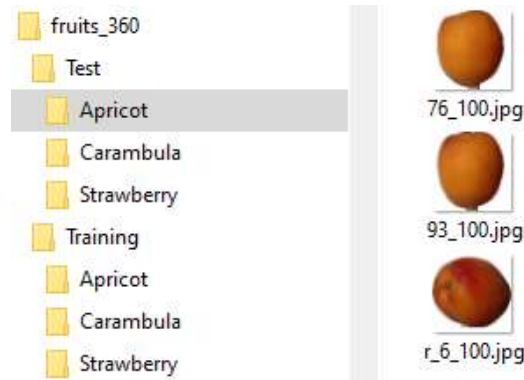
### Stockage

#### JEU DE TEST

131 répertoires (catégories)  
22 688 images  
152 Mo

#### JEU D'ENTRAÎNEMENT

131 répertoires (catégories)  
67 692 images  
467 Mo



Label	Number of training images	Number of test images
Banana Lady Finger	450	152
Banana Red	490	166
Beetroot	450	150
Blueberry	462	154
Cactus fruit	490	166
Cantaloupe 1	492	164
Cantaloupe 2	492	164
Carambula	490	166
Cauliflower	702	234
Cherry 1	492	164
Cherry 2	738	246
Cherry Rainier	738	246
Cherry Wax Black	492	164
Cherry Wax Red	492	164
Cherry Wax Yellow	492	164
Chestnut	450	153
Clementine	490	166

Source: Horea Muresan, Mihai Oltean. Fruit recognition from images using deep learning.

### Techniques de classification d'images

Réseaux de neurones convolutifs

- Identification auto des variables explicatives

Autres techniques d'apprentissage supervisé

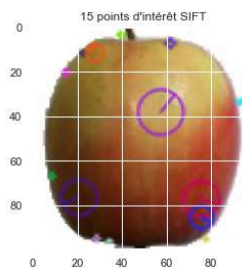
- Détermination des variables explicatives

## Variables explicatives

### Points d'intérêts et descripteurs

- OpenCV
  - SIFT, ORB, SURF

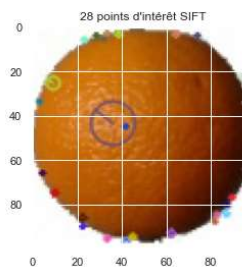
image 0



Points d'intérêt

...

image 241



### Descripteurs

	0	1	2	3	4	5	6	7	8	9	...	119	120	121	122	123	124	125	126	127	128
0	0.0	6.0	0.0	0.0	0.0	0.0	0.0	1.0	7.0	163.0	...	0.0	35.0	23.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0
1	0.0	6.0	0.0	0.0	0.0	0.0	2.0	8.0	9.0	173.0	...	0.0	84.0	16.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0
2	0.0	0.0	0.0	2.0	110.0	27.0	10.0	0.0	0.0	13.0	...	9.0	103.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	18.0	14.0	167.0	...	0.0	79.0	17.0	1.0	1.0	0.0	0.0	0.0	0.0	3.0
4	0.0	21.0	31.0	0.0	0.0	0.0	5.0	58.0	12.0	150.0	...	14.0	27.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7135	241.0	7.0	1.0	0.0	0.0	3.0	1.0	0.0	1.0	172.0	...	0.0	30.0	21.0	2.0	0.0	0.0	0.0	0.0	1.0	2.0
7136	241.0	2.0	0.0	0.0	0.0	4.0	2.0	0.0	0.0	163.0	...	0.0	85.0	24.0	3.0	1.0	0.0	0.0	0.0	0.0	6.0
7137	241.0	64.0	12.0	0.0	3.0	4.0	0.0	0.0	0.0	195.0	...	0.0	19.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7138	241.0	1.0	1.0	0.0	4.0	6.0	0.0	0.0	0.0	153.0	...	0.0	80.0	29.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0
7139	241.0	1.0	1.0	0.0	10.0	10.0	0.0	0.0	0.0	174.0	...	0.0	20.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



## 2 Modélisation

### Variables explicatives

#### Visual Words

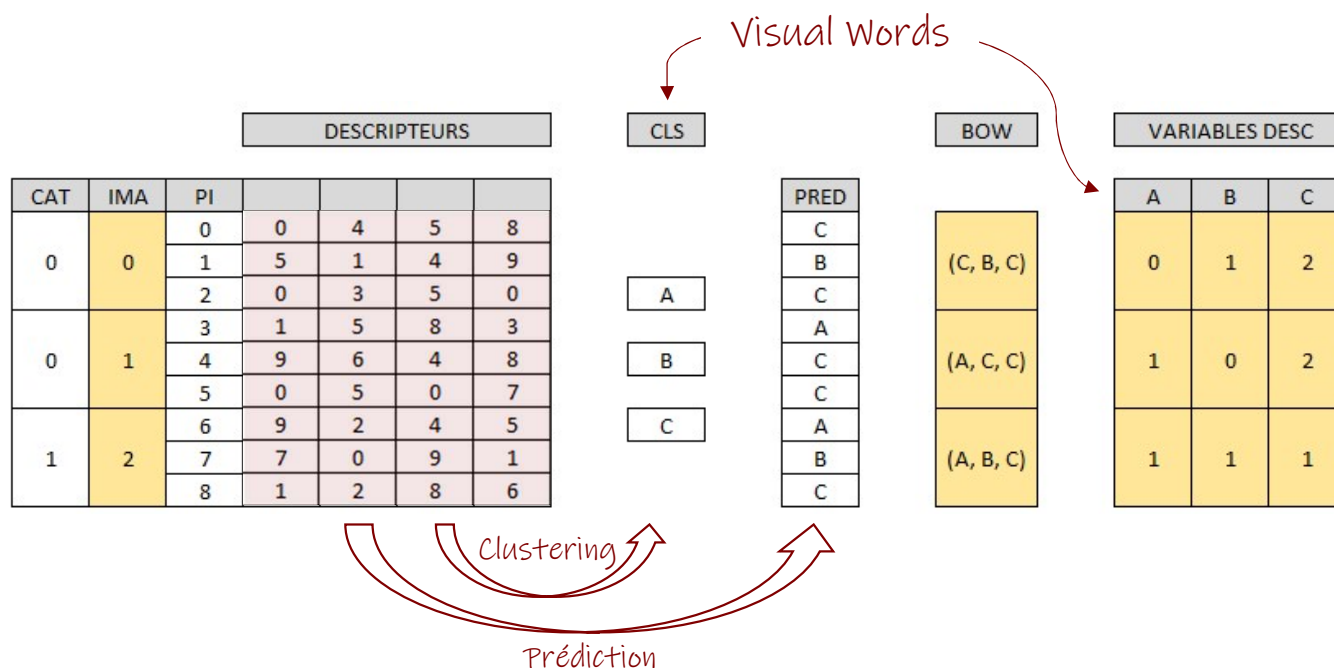
- Clustering des descripteurs
  - Modèle K-Means
    - K clusters (centroïdes)

#### Variables

- Classification des descripteurs
  - Prédictions modèle K-Means
- Bag of Words
- Historisation
  - Variables = clusters (visual words)
  - Valeurs = nombre de cluster par image

#### Poids

- Matrice des descripteurs



Jeu de données	Nombre d'images	Nombre de points d'intérêt par image	Nombre de descripteurs total	Poids d'un descripteur (octets)	Poids total (Gb)
Entraînement	67692	50	3384600	977	3,3
Test	22688	50	1134400	977	1,1



## Big Data

### Important volume de données

- Scalabilité
  - Augmentation des ressources (serveurs, RAM...)
- Limites
  - Lorsque les solutions classiques de stockage, de gestion et de traitement sont insuffisantes

### Les 3V du Big Data

- Volume
  - Stockage
- Vélocité
  - Traitement temps réel
- Variété
  - Données structurées / non structurées



### Calcul distribué

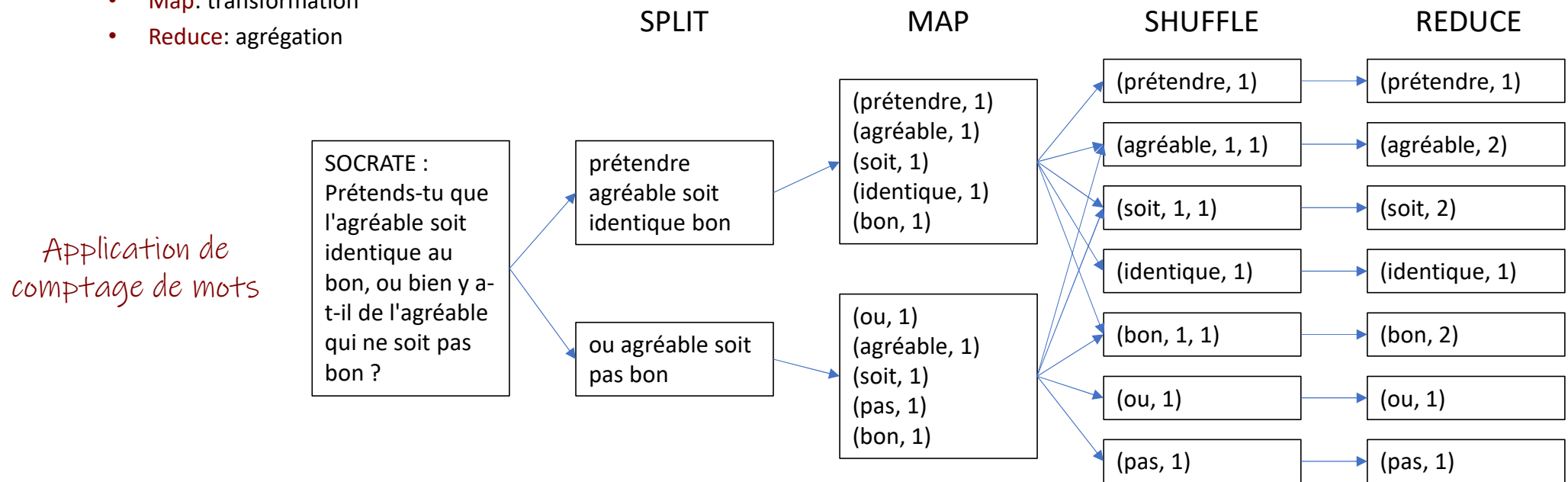
- Clusters de calcul
- Passage à l'échelle horizontal
- Plus grande tolérance aux pannes
  - Transfert de tâches entre nœuds du cluster
  - Recréer l'état du nœud en échec



## MapReduce

Cadre générique pour le calcul distribué

- Diviser pour régner
  - Problème → Sous-problèmes → Résolutions → Combinaison des résultats
- Combinaison de 2 fonctions simples
  - **Map**: transformation
  - **Reduce**: agrégation



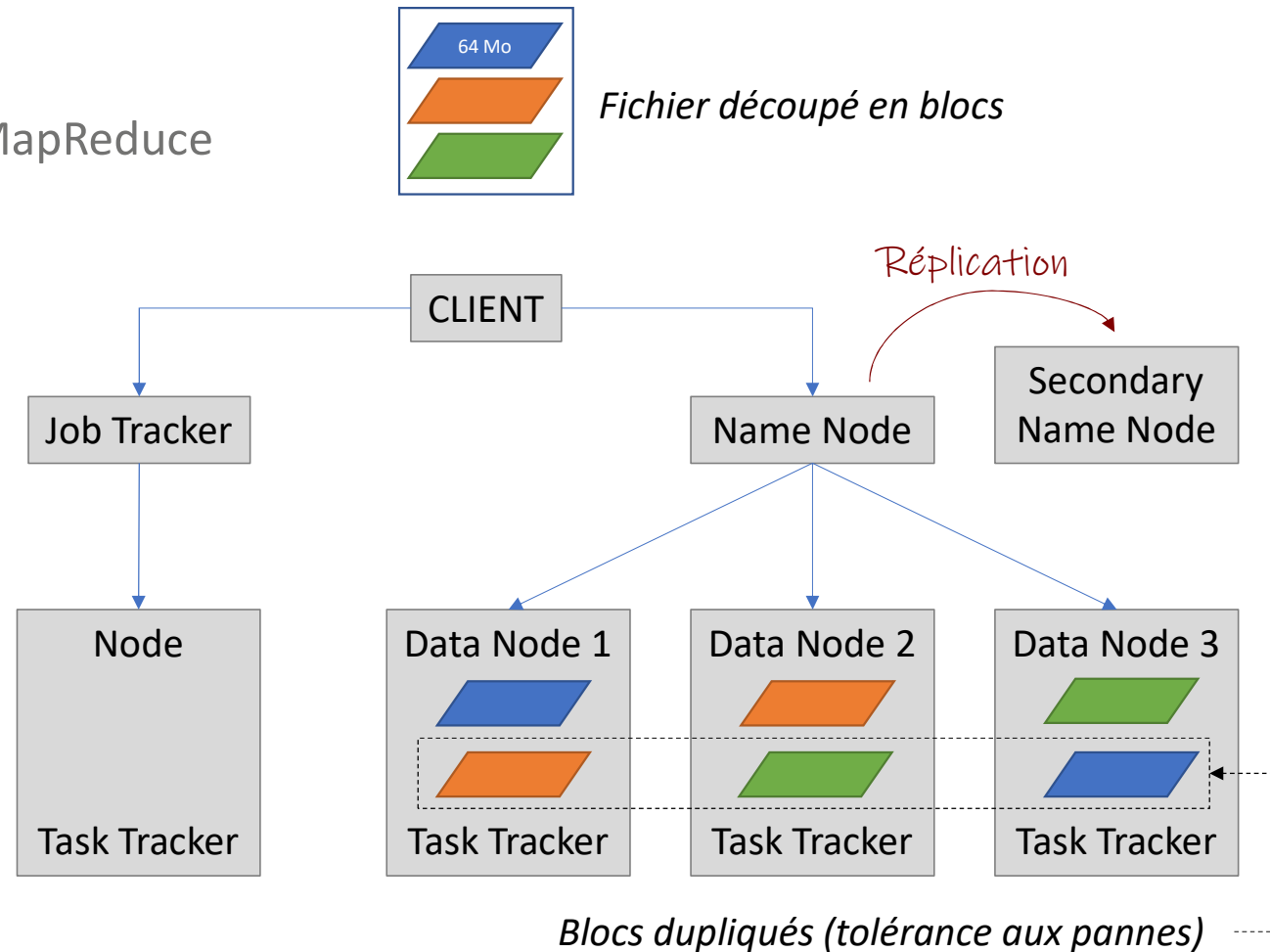




## Hadoop 1.0 (Apache)

### Infrastructure de référence pour MapReduce

- Socle technique
  - HDFS (Hadoop Distributed File System)
  - Framework MapReduce
- Architecture HDFS
  - Type maître / esclaves
  - Distribution des fichiers
  - Réplication des fichiers
  - Colocalisation données/traitements
- Framework MapReduce
  - Type maître / esclaves
  - Ordonnancement traitements
  - Distribution de l'exécution
  - Localisation des fichiers





### Hadoop 2.0 (Apache)

#### Optimisation de l'architecture

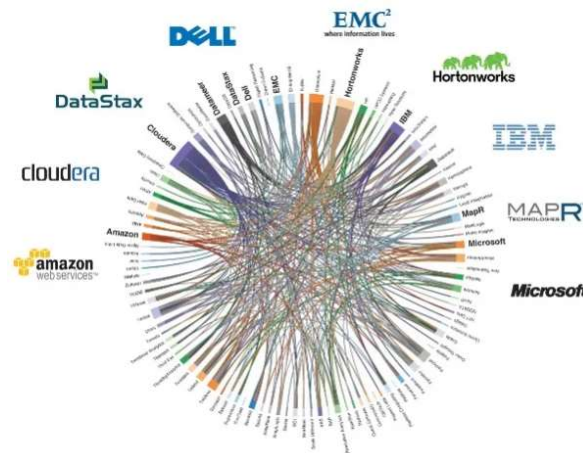
- YARN
  - Yet Another Ressource Negociator
  - Exécution de tout type d'applications
- Hadoop Streaming
  - Utilisation d'autres langages que Java

#### Installation / Distributions

- Installation manuelle
  - Paquets
- Distribution intégrée
  - Services
- Cloud
  - Services

#### Limites

- Ecriture sur disque
  - Ecriture sur disque des données entre 2 étapes (Map Reduce)
    - Lenteur d'exécution
- Jeu d'instructions limité
  - Map et Reduce
    - Difficulté de réaliser des opérations complexes



### Spark (Apache)

#### Description

- Framework open source de calcul distribué

#### Avantages

- Données stockées en RAM
  - Rapidité d'exécution (X10 à X100 par rapport à Hadoop)
- Jeu d'instructions optimisé
  - Nombreuses opérations en mode distribué
  - Réduction automatique niveau Map / Reduce

#### Langages de programmation

- Java
- Scala (natif)
- Python
  - API PySpark



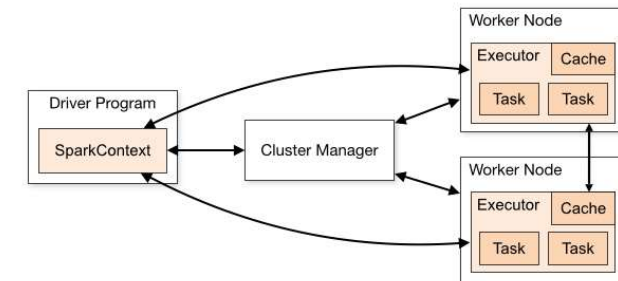
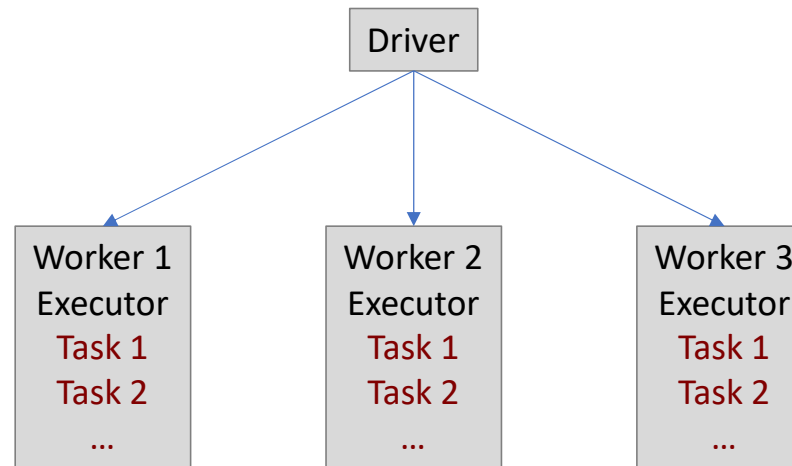
Temps réel

Machine Learning

## Spark (Apache)

### Architecture

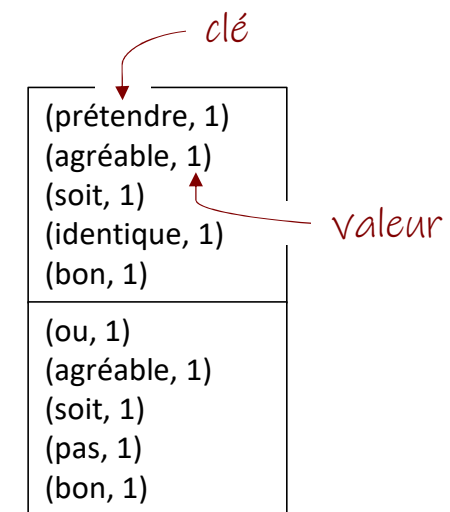
- Type Maître / Esclaves
- Gestion des fichiers: HDFS
- Hadoop Map Reduce



Source: Documentation de Spark

### Distribution des calculs

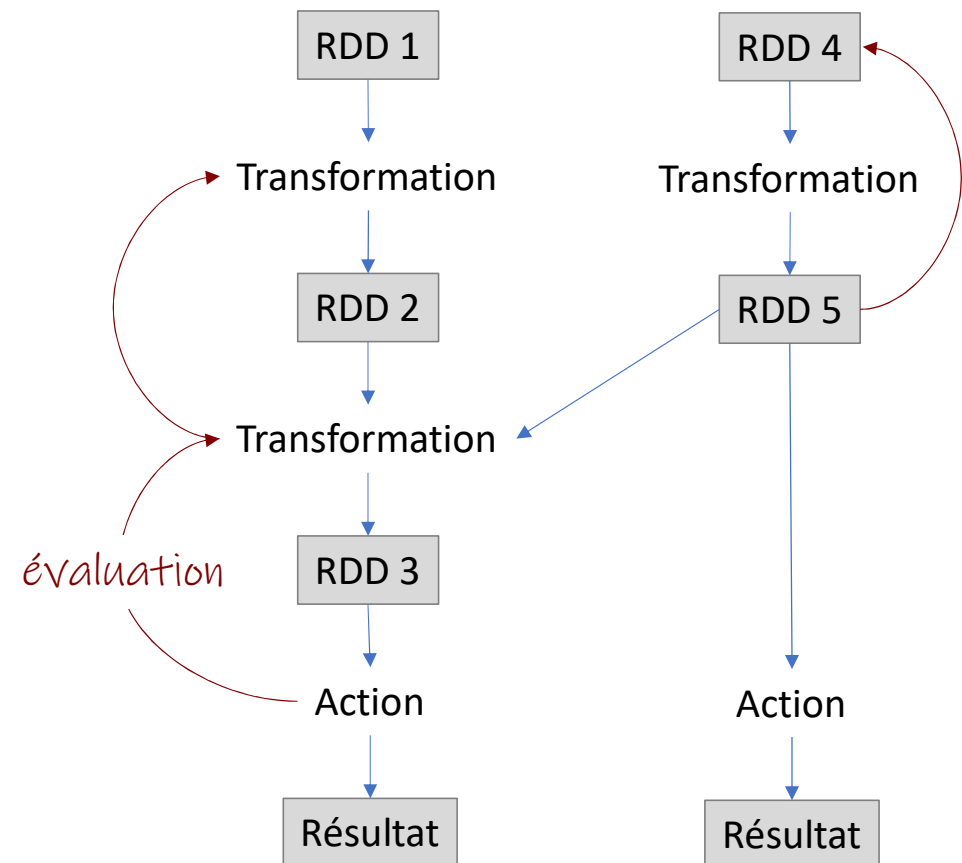
- Classe SparkContext
  - Configuration de l'application
  - Lecture des données
- Création d'un objet de type RDD (Resilient Distributed Dataset)
  - Format permettant la distribution des calculs
  - Optimisé pour la tolérance aux pannes
  - Type clé / valeur



## Spark (Apache)

### RDD (Resilient Distributed Dataset)

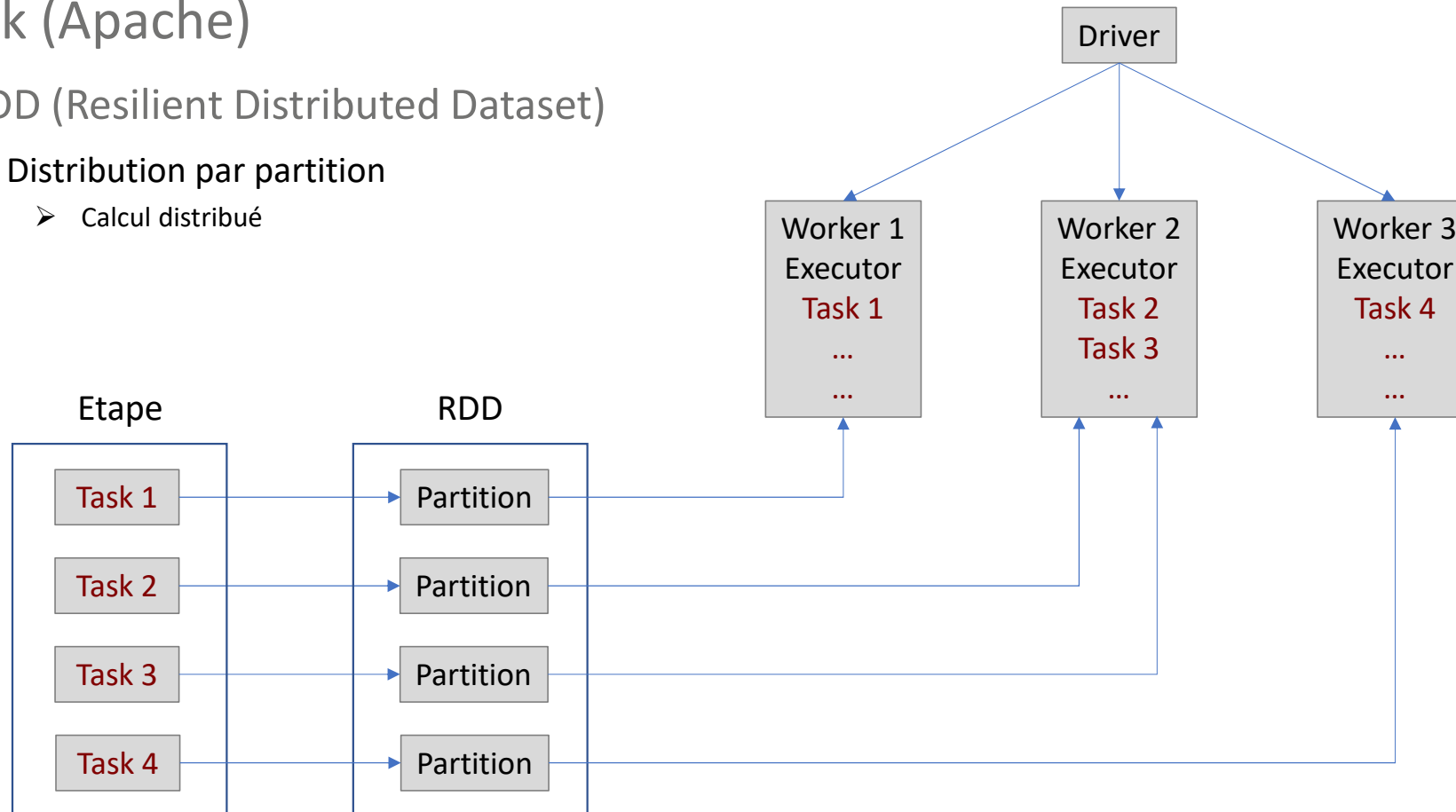
- Types d'opérations
  - Transformation
  - Action
- DAG (Directed Acyclic Graph)
  - Tolérance aux pannes
- Lazy evaluation
  - Evaluation des transformations au moment utile
  - Lors d'une action



## Spark (Apache)

### RDD (Resilient Distributed Dataset)

- Distribution par partition
  - Calcul distribué

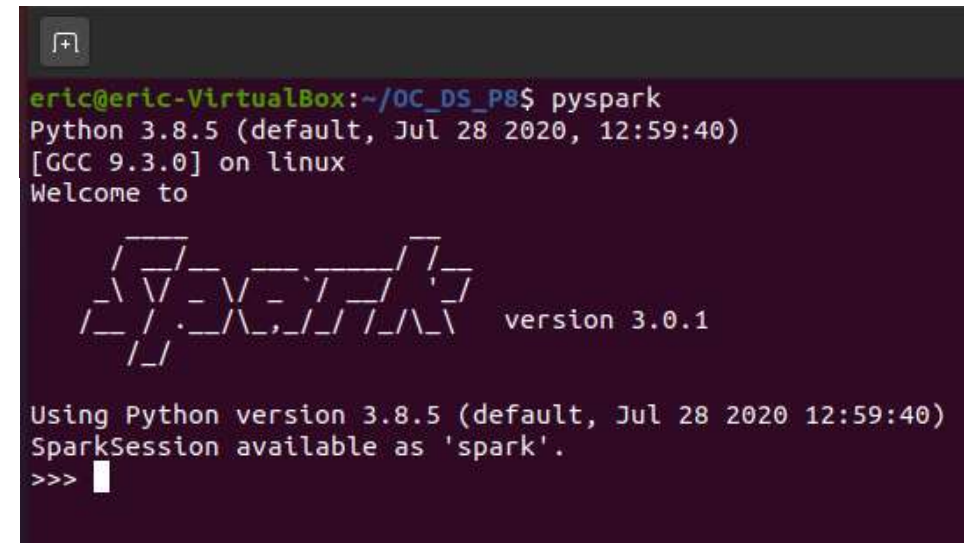


## Spark - Langage Python

### Installation (LOCAL)

- Packages
  - Java
  - Python3
  - Jupyter Notebook
  - Spark / PySpark
- Variables d'environnement

```
export SPARK_HOME=/opt/spark
export PATH=$SPARK_HOME/bin:$PATH
export PYSARK_PYTHON=python3
```



```
eric@eric-VirtualBox:~/OC_DS_P8$ pyspark
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Welcome to

      _ _ _ _ _
     / _ _ _ _ \   version 3.0.1
    / _ _ _ _ \
   / _ _ _ _ \
  / _ _ _ _ \
 / _ _ _ _ \
/_ _ _ _ _ \

Using Python version 3.8.5 (default, Jul 28 2020 12:59:40)
SparkSession available as 'spark'.
>>>
```

Console Spark - Python

eric@eric-VirtualBox:~/OC\_DS\_P8\$ **spark-submit --master local[2] P8\_01\_spark.py True data/fruits\_360...**



## Spark - Langage Python

### Installation (LOCAL)

- Packages

- Java
- Python3
- Jupyter Notebook
- Spark / PySpark

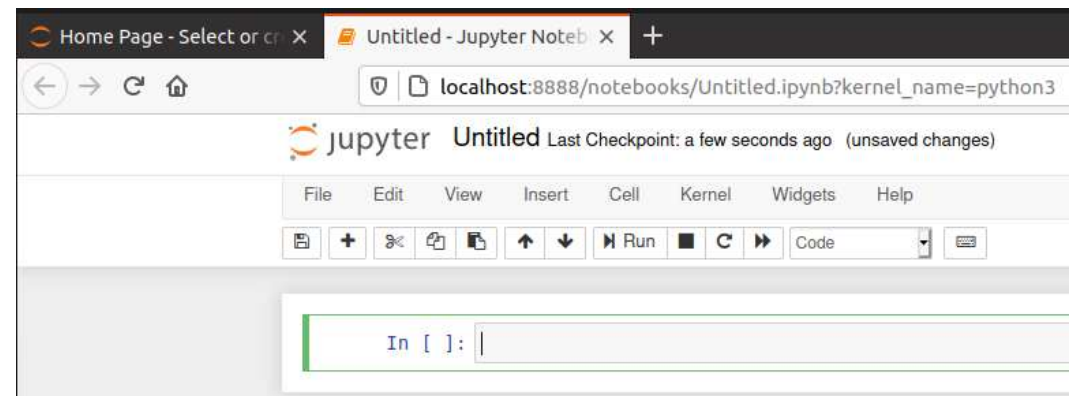
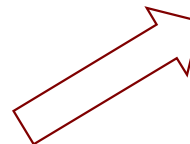
```
eric@eric-VirtualBox:~/OC_DS_P8$ pyspark
[I 18:32:22.066 NotebookApp] Serving notebooks from local directory: /home/eric/OC_DS_P8
[I 18:32:22.066 NotebookApp] The Jupyter Notebook is running at:
[I 18:32:22.067 NotebookApp] http://localhost:8888/?token=f7f0bcbe79395ca952b1d8a5e1339236d49d9f66a5199a23
[I 18:32:22.067 NotebookApp] or http://127.0.0.1:8888/?token=f7f0bcbe79395ca952b1d8a5e1339236d49d9f66a5199a23
[I 18:32:22.067 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:32:22.185 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/eric/.local/share/jupyter/runtime/nbserver-14058-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=f7f0bcbe79395ca952b1d8a5e1339236d49d9f66a5199a23
or http://127.0.0.1:8888/?token=f7f0bcbe79395ca952b1d8a5e1339236d49d9f66a5199a23
```

- Variables d'environnement

```
export SPARK_HOME=/opt/spark
export PATH=$SPARK_HOME/bin:$PATH
export PYSARK_PYTHON=python3

export PYSARK_DRIVER_PYTHON=jupyter
export PYSARK_DRIVER_PYTHON_OPTS='notebook'
```



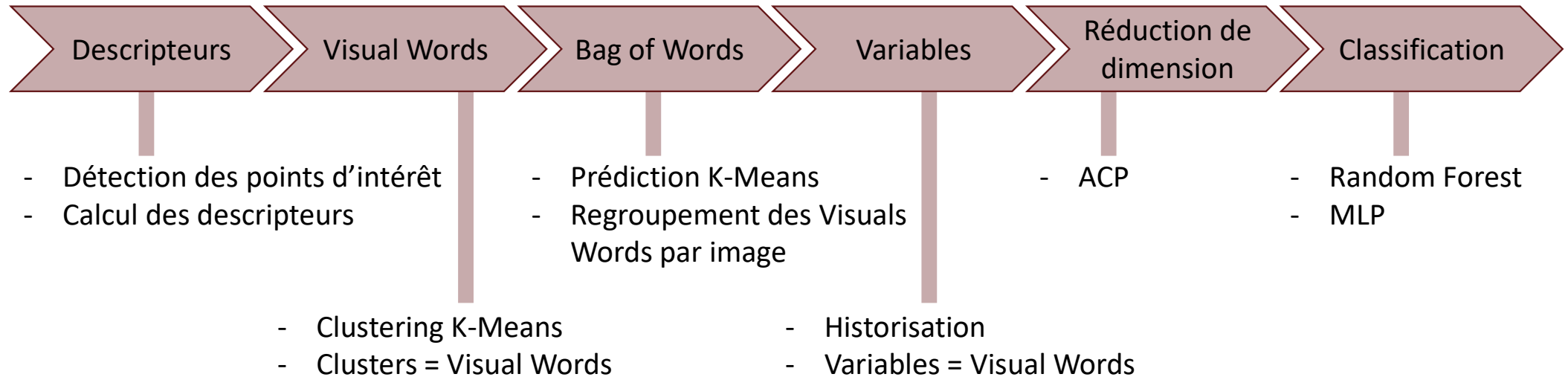


# 4 Conception

Déploiement d'un modèle dans le Cloud



Fruits!





# 4 Conception

Descripteurs

Visual Words

Bag of Words

Variables

Réduction de  
dimension

Classification

```
=====
Identification des chemins d'accès aux répertoires d'images
=====
```

Nombre d'images par catégorie (sous-répertoire):

	Catégorie	Nombre d'images
0	Corn	50
1	Raspberry	50
2	Orange	50

Nombre total d'images: 150      dataset\_path = data/fruits\_360\_v3b/Training/

```
sdf_images = spark.read.format("binaryFile") \
    .option("pathGlobFilter", "*.jpg") \
    .option("recursiveFileLookup", "true") \
    .load(dataset_path) \
    .select("path", "content")

rdd_images = sdf_images.rdd
rdd_cat_ima_desc = rdd_images.map(lambda img: get_descriptors(img))
rdd_cat_ima_desc_f = rdd_cat_ima_desc.filter(lambda x: x[2] is not None).cache()
rdd_desc = rdd_cat_ima_desc_f.flatMap(lambda x: x[2])
```

```
=====
Calcul des descripteurs
=====
```

```
=====
Chargement des images (rdd_images)
=====
```

MapPartitionsRDD[4] at javaToPython

Nombre de partitions: 5  
Dimension: 150

```
=====
Descripteurs (rdd_desc)
=====
```

PythonRDD[16] at RDD at PythonRDD.scala:53

Nombre de partitions: 5  
Dimension: 11133

# 4 Conception

Déploiement d'un modèle dans le Cloud



Fruits!



```
Descripteurs (rdd_desc)
=====
```

```
PythonRDD[16] at RDD at PythonRDD.scala:53
```

```
Nombre de partitions: 5
Dimension: 11133
```

```
km_model = KMeans.train(rdd_desc, nb_clusters, maxIterations=2000, initializationMode="random")
```

```
=====  
Classification non supervisée des descripteurs avec K-Means  
=====
```

```
Modèle K-Means (km_model)
=====
```

```
<pyspark.mllib.clustering.KMeansModel object at 0x7fd9a5a06130>
```

```
Nombre de clusters: 30 ← Visual Words
```



# 4 Conception



```
rdd_km_pred = km_model.predict(rdd_desc)
```

```
=====
Prédictions des descripteurs avec K-Means
=====
```

```
Prédictions (rdd_km_pred)
```

```
=====
```

```
PythonRDD[226] at RDD at PythonRDD.scala:53
```

```
Nombre de partitions: 5
```

```
Dimension: 11133
```

```
Collecte des prédictions (list_km_pred)
```

```
=====
```

```
[3, 21, 11, 27, 7, 11, 13, 11, 20, 1]
```

Concaténation des identifiants  
des images (encodés) et des  
prédictions (clusters K-Means)

```
rdd_ima_pred = sdf_ima_pred.rdd.map(lambda x:x)
rdd_words = rdd_ima_pred.reduceByKey(lambda a,b: str(a) + ',' + str(b))
```

```
Nombre de partitions: 2
```

```
Dimension: 79
```

```
(sdf_ima_pred)
+---+-----+
| id|prediction|
+---+-----+
|110|      3|
|110|     21|
|110|     11|
|110|     27|
|110|      7|
|110|     11|
|110|     13|
|110|     11|
|110|     20|
|110|      1|
+---+-----+
```

ReduceByKey

```
(sdf_words)
+---+-----+
| image_id|      words|
+---+-----+
|    110|[3, 21, 11, 27, 7...|
|    112|[3, 3, 3, 27, 11,...|
|    108|[19, 13, 27, 3, 7...|
|    106|[3, 1, 3, 27, 27,...|
|    114|[19, 3, 7, 3, 11,...|
|    116|[21, 3, 7, 23, 21...|
|    140|[27, 3, 8, 15, 27...|
|    124|[13, 7, 0, 3, 3, ...|
|    102|[13, 3, 21, 27, 3...|
|    136|[3, 19, 27, 8, 27...|
+---+-----+
```

```
sdf_words = rdd_words.map(lambda tupl_words: (tupl_words[0], str(tupl_words[1]).split(','))) \
.toDF(['image_id', 'words'])
```



# 4 Conception



```

vectorizer = CountVectorizer(inputCol="words", outputCol="bag_of_words")
vectorizer_transformer = vectorizer.fit(sdf_words)
sdf_bow = vectorizer_transformer.transform(sdf_words).select('image_id', 'bag_of_words')
  
```

(sdf\_words)

image_id	words
110	[3, 21, 11, 27, 7...]
112	[3, 3, 3, 27, 11,...]
108	[19, 13, 27, 3, 7...]
106	[3, 1, 3, 27, 27,...]
114	[19, 3, 7, 3, 11,...]
116	[21, 3, 7, 23, 21...]
140	[27, 3, 8, 15, 27...]
124	[13, 7, 0, 3, 3, ...]
102	[13, 3, 21, 27, 3...]
136	[3, 19, 27, 8, 27...]

CountVectorizer

Bag of words (sdf bow)

image_id	bag_of_words
110	(30, [0,1,2,3,4,5,...])
112	(30, [0,1,2,3,4,5,...])
108	(30, [0,1,2,3,4,5,...])
106	(30, [0,1,2,3,4,5,...])
114	(30, [0,1,2,3,4,5,...])
116	(30, [0,1,2,3,4,5,...])
140	(30, [0,1,2,3,4,5,...])
124	(30, [0,1,2,3,4,5,...])
102	(30, [0,1,2,3,4,5,...])
136	(30, [0,1,2,3,4,5,...])

transformation

DataFrame des  
variables explicatives

	ima	cat	0	1	2	3	4	...
0	Raspberry_19_100.jpg	Raspberry	3.0	9.0	4.0	15.0	4.0	...
1	Raspberry_14_100.jpg	Raspberry	2.0	8.0	4.0	12.0	3.0	...
2	Raspberry_43_100.jpg	Raspberry	2.0	8.0	24.0	11.0	2.0	...
3	Raspberry_45_100.jpg	Raspberry	2.0	7.0	18.0	9.0	2.0	...
4	Raspberry_20_100.jpg	Raspberry	2.0	8.0	6.0	13.0	2.0	...

Dimensions du jeu de données: (150, 32)

Nombre de visuals  
words uniques (30)





# 4 Conception



```

pca_dim = int(nb_clusters-(nb_clusters*0.3))
pca = PCA(k=pca_dim, inputCol="bag_of_words", outputCol="features")
model = pca.fit(sdf_bow)
sdf_features = model.transform(sdf_bow).select("features")
  
```

Encodage de la variable catégories (sdf\_lab\_features)

=====

```

root
|-- label: double (nullable = false)
|-- features: vector (nullable = true)
  
```

Bag of words (sdf\_bow)

image_id	bag_of_words
110	(30, [0,1,2,3,4,5,...]
112	(30, [0,1,2,3,4,5,...]
108	(30, [0,1,2,3,4,5,...]
106	(30, [0,1,2,3,4,5,...]
114	(30, [0,1,2,3,4,5,...]
116	(30, [0,1,2,3,4,5,...]
140	(30, [0,1,2,3,4,5,...]
124	(30, [0,1,2,3,4,5,...]
102	(30, [0,1,2,3,4,5,...]
136	(30, [0,1,2,3,4,5,...]

PCA

label	features
1.0	[-1.9381004478848...
1.0	[-0.5822993253601...
1.0	[-1.3910691745482...
1.0	[-1.3747704461391...
2.0	[26.6684862365683...
0.0	[2.23326783378635...
0.0	[1.16132418870859...
0.0	[3.17218229130792...
0.0	[2.81897891583134...
1.0	[-0.8462483779417...

Bag of words après réduction de dimension (df\_lab\_features)

	label	0	1	2	3	4
0	1.0	-1.938100	4.373330	7.230668	-6.354446	6.046578
1	1.0	-0.582299	3.418589	5.567492	-7.003518	3.878954
2	1.0	-1.391069	3.003477	5.123474	-6.696992	4.103136
3	1.0	-1.374770	3.939350	5.026782	-4.804279	1.135001
4	2.0	26.668486	7.078681	-0.799438	-4.180231	5.872323
5	0.0	2.233268	-2.347001	-2.344602	-3.261785	3.304597
6	0.0	1.161324	-1.819158	0.087453	-6.264102	5.091056
7	0.0	3.172182	0.122427	0.194158	-9.281862	8.285937
8	0.0	2.818979	-4.847871	-2.529770	-4.401651	4.233620

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (150, 22)



# 4 Conception



label	features
1.0	[-1.9381004478848...
1.0	[-0.5822993253601...
1.0	[-1.3910691745482...
1.0	[-1.3747704461391...
2.0	[26.6684862365683...
0.0	[2.23326783378635...

Apprentissage

```

layers = [pca_dim, pca_dim, pca_dim, nb_cat]

mlp = MultilayerPerceptronClassifier(layers=layers, seed=123)
mlp.setMaxIter(200)
mlp.setBlockSize(128)
mlp_model = mlp.fit(result_ima_4)
mlp_model.setFeaturesCol("features")
  
```

label	features
2.0	[26.4668964038766...
1.0	[-0.4043484642786...
2.0	[28.6641195896339...
0.0	[5.18647680516223...
1.0	[-0.0956747951576...
2.0	[24.8632082783237...

Test

```
test_pred_results = mlp_model.transform(result).select("features", "prediction")
```

ima	cat	label	prediction	features
Raspberry_87_100.jpg	Raspberry	2.0	2.0	[26.4668964038766...
Orange_3_100.jpg	Orange	1.0	1.0	[-0.4043484642786...
Raspberry_82_100.jpg	Raspberry	2.0	2.0	[28.6641195896339...
Corn_2_100.jpg	Corn	0.0	2.0	[5.18647680516223...
Orange_43_100.jpg	Orange	1.0	1.0	[-0.0956747951576...
Raspberry_100_100...	Raspberry	2.0	2.0	[24.8632082783237...
Raspberry_98_100.jpg	Raspberry	2.0	2.0	[25.6584718118259...

Evaluation

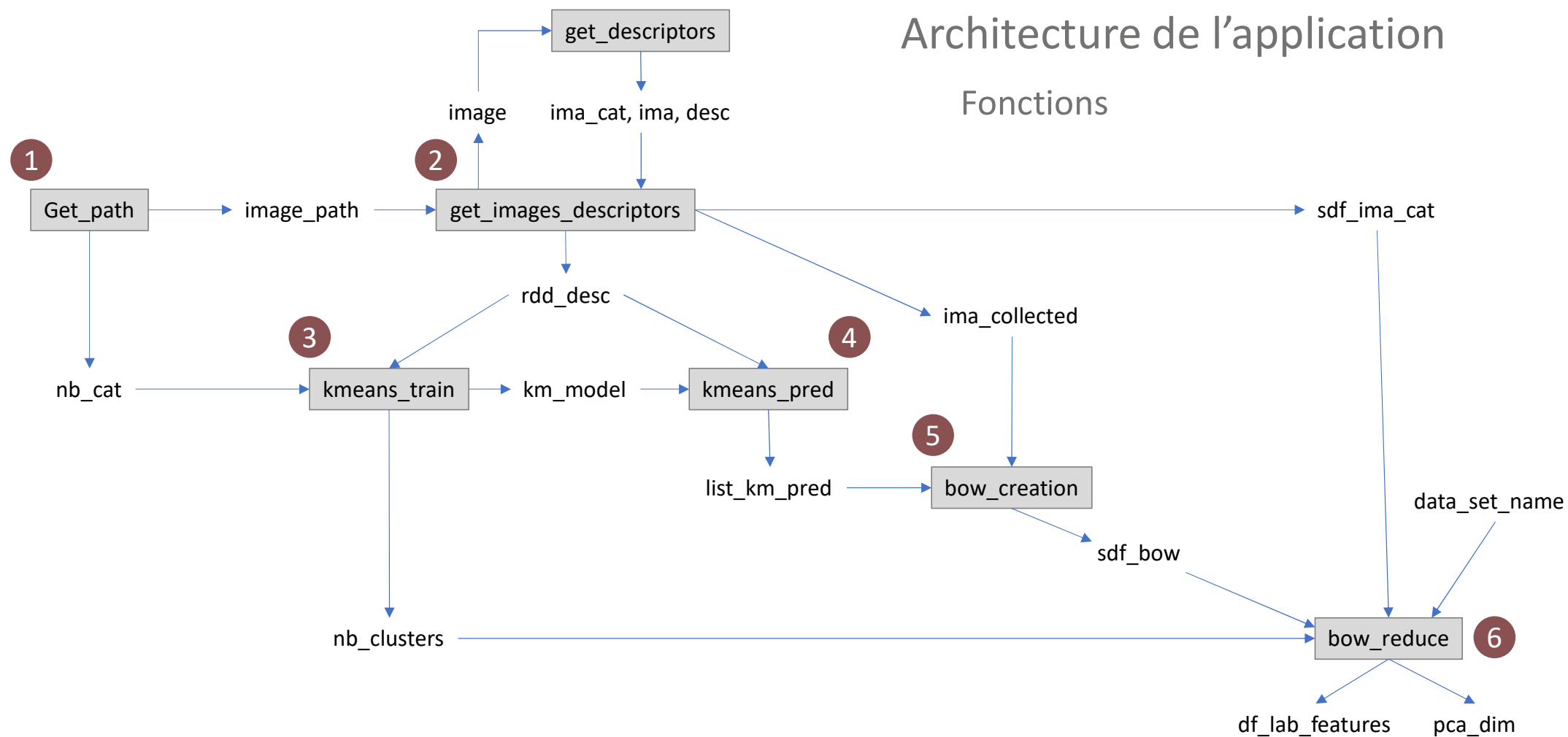
prediction	0.0	1.0	2.0
label			
0.0	16.0	56.0	28.0
1.0	0.0	100.0	0.0
2.0	0.0	0.0	100.0

Test set accuracy (MLP) = 0.72

# 4 Conception

## Architecture de l'application

### Fonctions

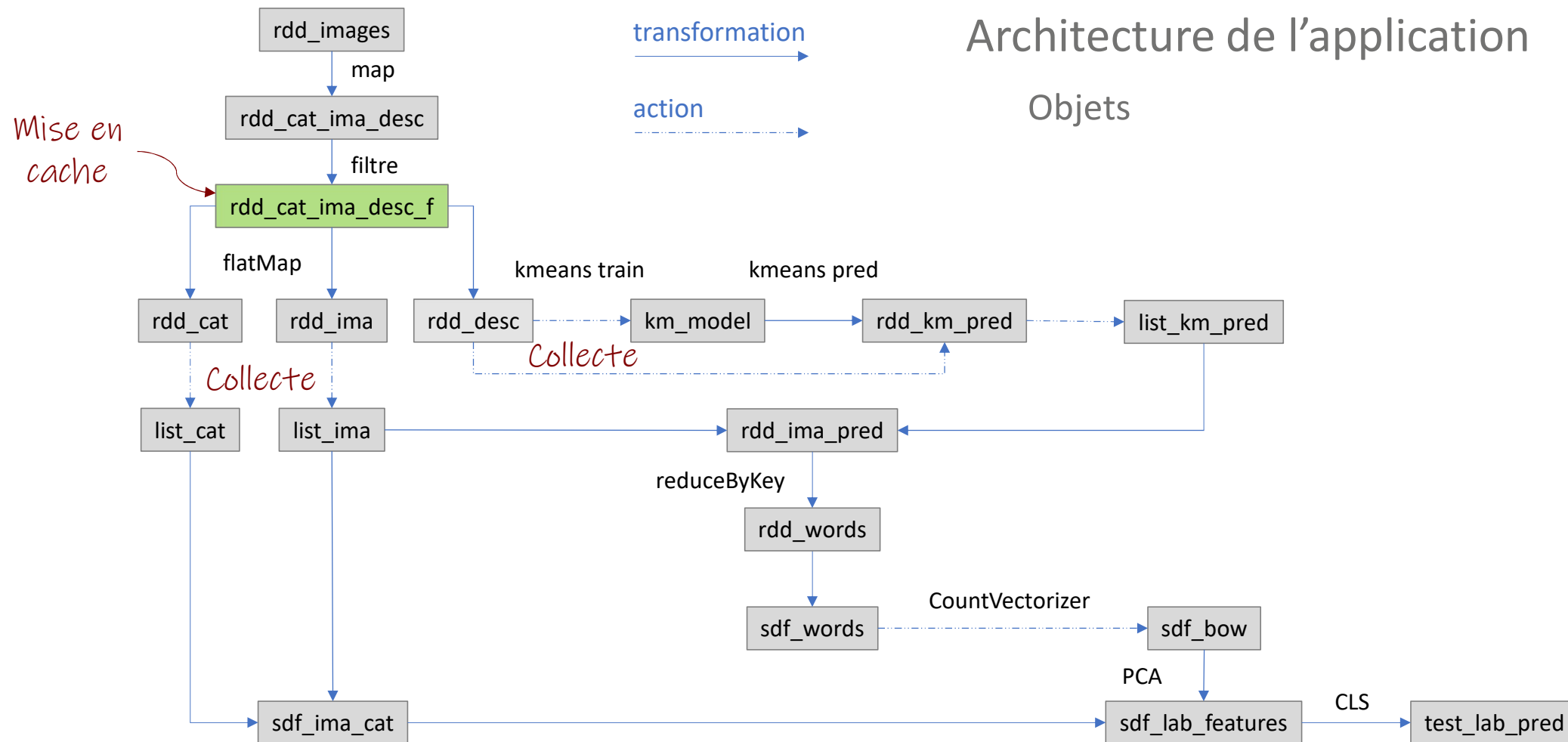




# 4 Conception

## Architecture de l'application

### Objets



## Plateforme pour le Big Data

### Le Cloud

- Accès à des ressources distantes
  - Calcul (CPU, mémoire...)
  - Stockage (espace disque)
- Elasticité
  - Modification des capacités
- Gestion des coûts
  - Facturation à l'utilisation

### Solutions



### AWS (Amazon Web Service)

- Serveurs de calcul
  - Elastic Compute Cloud (EC2)
- Clusters
  - Elastic Map Reduce (EMR)
    - Framework Hadoop hébergé
- Stockage
  - Simple Storage Service (S3)
    - Connecteur HDFS

# 5 Déploiement dans le Cloud



Déploiement d'un modèle dans le Cloud



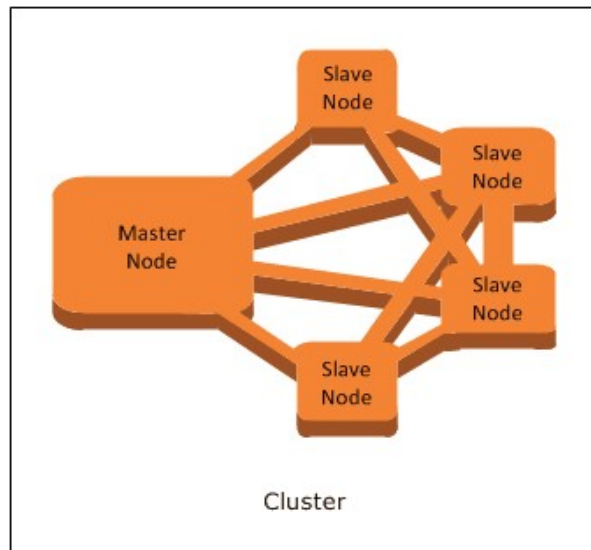
Fruits!



amazon  
EMR



EC2  
instances



Données



S3

Objets

- Images (fruits)
- Bag of words
- Logs



bucket

## Création d'un Cluster avec EMR

### Configuration générale

Nom du cluster

☒ Journalisation ⓘ

Dossier S3

Mode de lancement ☒ Cluster ⓘ ☐ Exécution d'étape ⓘ

### Configuration des logiciels

Libérer  ⓘ

Applications

- ☐ Core Hadoop: Hadoop 2.10.0, Hive 2.3.7, Hue 4.7.1, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- ☐ HBase: HBase 1.4.13, Hadoop 2.10.0, Hive 2.3.7, Hue 4.7.1, Phoenix 4.14.3, and ZooKeeper 3.4.14
- ☐ Presto: Presto 0.238.3 with Hadoop 2.10.0 HDFS and Hive 2.3.7 Metastore
- ☒ Spark: Spark 2.4.6 on Hadoop 2.10.0 YARN and Zeppelin 0.8.2

☐ Utiliser AWS Glue Data Catalog pour les métadonnées de table ⓘ

### Configuration du matériel

Type d'instance  ⓘ Le type d'instance sélectionné ajoute un volume EBS GP2 par défaut de 64 GiO par instance. [En savoir plus](#)

Nombre d'instances  (1 nœud maître et 2 nœuds principaux)

Cluster scaling ☐ scale cluster nodes based on workload

### Sécurité et accès

Paire de clés EC2  ⓘ [Apprenez à créer une paire de clés EC2.](#)

Autorisations ☒ Par défaut ☐ Personnalisé

Utilisez les rôles IAM par défaut. Si des rôles sont absents, ils seront créés automatiquement pour vous avec des stratégies gérées pour les mises à jour automatiques de stratégies.

Rôle EMR [EMR\\_DefaultRole](#) ⓘ

Profil d'instance EC2 [EMR\\_EC2\\_DefaultRole](#) ⓘ



## Création d'un Cluster avec EMR

Cluster : oc-ds-p8 **En attente** Cluster ready after last step completed.

Récapitulatif	Historique de l'application	Surveillance	Matériel	Configurations	Événements	Étapes	Actions d'amorçage
<p><b>Récapitulatif</b></p> <p>ID : j-HVWO2J54XAYC</p> <p>Date de création : 28-11-2020 23:46 (UTC+1)</p> <p>Temps écoulé : 5 minutes</p> <p>Résiliation automatique : Cluster waits</p> <p>Protection de la résiliation : Désactivé <a href="#">Modification</a></p> <p>Balises : -- <a href="#">Afficher tout/Modifier</a></p> <p>DNS public principal : ec2-54-247-18-29.eu-west-1.compute.amazonaws.com </p> <p><a href="#">Connect to the Master Node Using SSH</a></p>	<p><b>Détails de configuration</b></p> <p>Étiquette de version : emr-5.31.0</p> <p>Distribution Hadoop : Amazon</p> <p>Applications : Spark 2.4.6, Zeppelin 0.8.2</p> <p>URI de connexion : s3://aws-logs-383023238722-eu-west-1/elasticmapreduce/ </p> <p>Vue cohérente EMRFS : Désactivé</p> <p>ID d'AMI personnalisée : --</p>						
<p><b>Application user interfaces</b></p> <p>Service d'historique :  <a href="#">Spark history server, YARN timeline server</a></p> <p>Connexions :  Not Enabled <a href="#">Activer la connexion Web</a></p>	<p><b>Réseau et matériel</b></p> <p>Zone de disponibilité : eu-west-1a</p> <p>ID de sous-réseau (subnet) : <a href="#">subnet-42d7aa18</a> </p> <p>Maître : En cours d'exécution 1 m5.xlarge</p> <p>Principal : En cours d'exécution 2 m5.xlarge</p> <p>Tâche : --</p> <p>Cluster scaling: Not enabled</p>						



## Création d'un Cluster avec EMR

Cluster : oc-ds-p8 **En attente** Cluster ready after last step completed.

[Récapitulatif](#)[Historique de l'application](#)[Surveillance](#)[Matériel](#)[Configurations](#)[Événements](#)[Étapes](#)[Actions d'amorçage](#)[Ajouter un groupe d'instances de tâches](#)

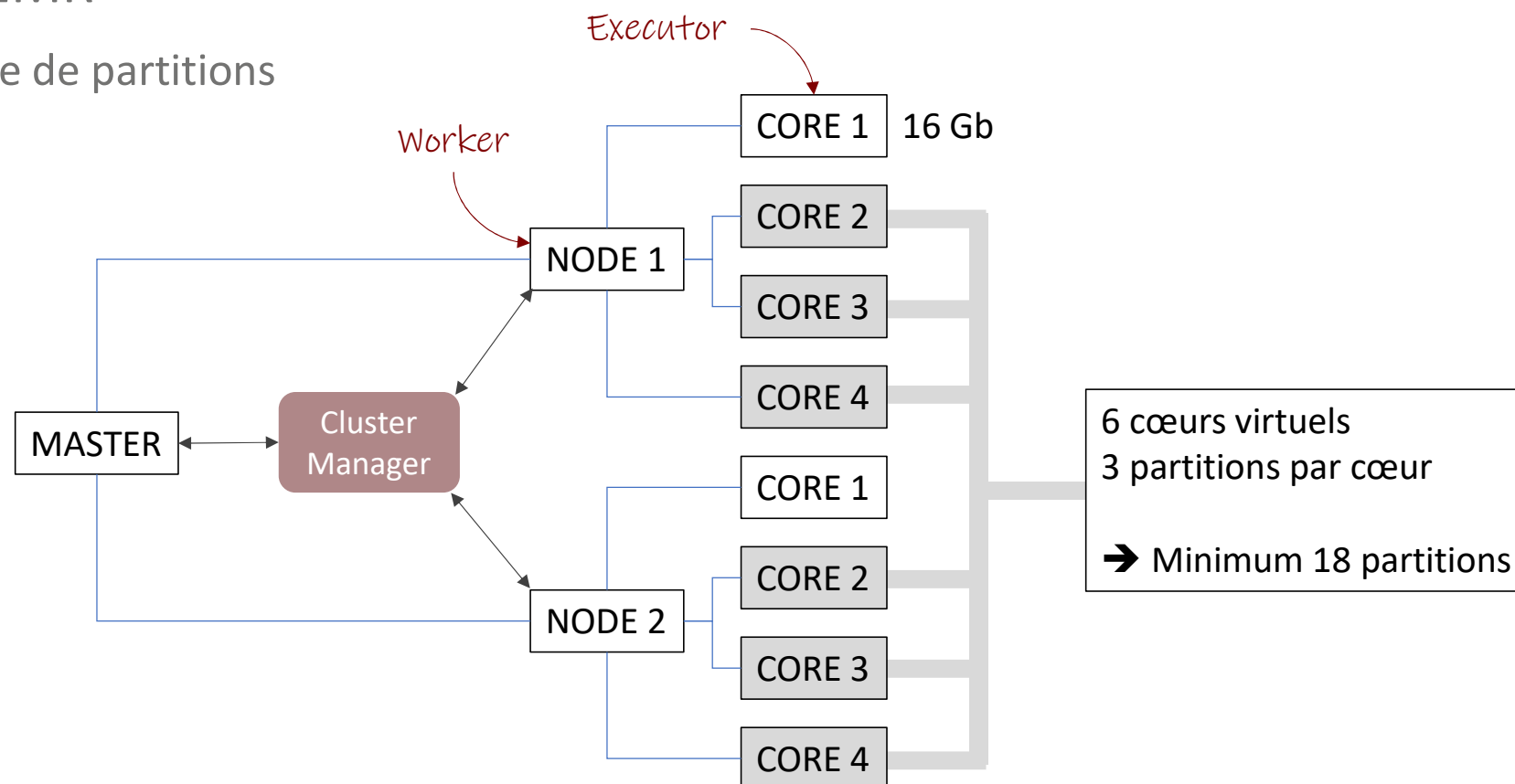
Groupes d'instances

Filtre :  2 groupes d'instances (tous chargés) 

ID	Status	Nom et type de nœud	Type d'instance	Nombre d'instances
▶ <a href="#">ig-WA66DB11HVHC</a>	En cours d'exécution	<b>CORE</b> Core Instance Group	<b>m5.xlarge</b> 4 Cœurs virtuels, 16 GiO de mémoire, stockage EBS uniquement Stockage sur EBS : 64 Gio	2 Instances <a href="#">Redimensionner</a>
▶ <a href="#">ig-2FYSG6LRE40DT</a>	En cours d'exécution	<b>MASTER</b> Master Instance Group	<b>m5.xlarge</b> 4 Cœurs virtuels, 16 GiO de mémoire, stockage EBS uniquement Stockage sur EBS : 64 Gio	1 Instances

## Cluster EMR

Nombre de partitions





# 5 Déploiement dans le Cloud



Déploiement d'un modèle dans le Cloud



## Cluster EMR

Cluster : oc-ds-p8 **En attente** Cluster ready after last step completed.

Récapitulatif

Historique de l'application

Surveillance

Matériel

### Récapitulatif

ID : j-HVWO2J54XAYC

Date de création : 28-11-2020 23:46 (UTC+1)

Temps écoulé : 5 minutes

Résiliation automatique : Cluster waits

Protection de la résiliation : Désactivé [Modification](#)

Balises : -- [Afficher tout/Modifier](#)

DNS public principal :

ec2-54-247-18-29.eu-west-1.compute.amazonaws.com

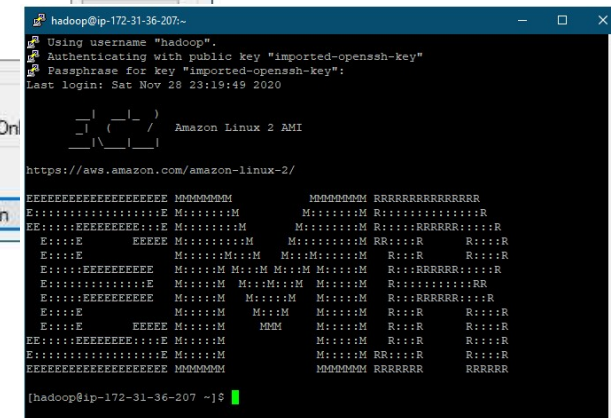
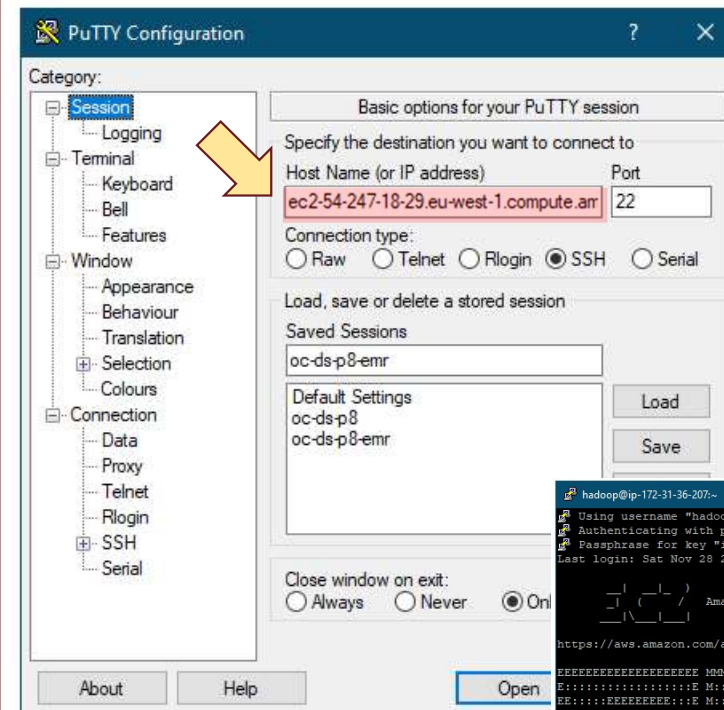
[Connect to the Master Node Using SSH](#)

### Application user interfaces

Service d'historique : [Spark history server, YARN timeline server](#)

Connexions : [Not Enabled](#) [Activer la connexion Web](#)

## Accès au driver via SSH





# 5 Déploiement dans le Cloud



Déploiement d'un modèle dans le Cloud



## Cluster EMR

Cluster : oc-ds-p8 **En attente** Cluster ready after last step completed.

Récapitulatif

Historique de l'application

Surveillance

Matériel

### Récapitulatif

ID : **j-HVWO2J54XAYC**

Date de création : 28-11-2020 23:46 (UTC+1)

Temps écoulé : 5 minutes

Résiliation automatique : Cluster waits

Protection de la résiliation : Désactivé [Modification](#)

Balises : -- [Afficher tout/Modifier](#)

DNS public principal :

ec2-54-247-18-29.eu-west-1.compute.amazonaws.com

[Connect to the Master Node Using SSH](#)

### Application user interfaces

Service d'historique : [Spark history server](#), [YARN timeline server](#)

Connexions : [Not Enabled](#) [Activer la connexion Web](#)

## Connecter un notebook Jupyter

Bloc-notes : oc-ds-p8 **Prêt** Workspace(notebook) is ready to run jobs on **cluster j-HVWO2J54XAYC.**

Ouvrir dans JupyterLab

Ouvrir dans Jupyter

Arrêter

Supprimer

### Bloc-notes

ID de bloc-notes e-6DXBFIF3FS8APKKVH1M8LRBC1

Description : --

Dernière modification : il y a 3 secondes

Dernière modification par : ...root

Créé le : 10-11-2020 17:26 (UTC+1)

Créé par : ...root

Rôle de service IAM : [EMR\\_Notebooks\\_DefaultRole](#)

Groupes de sécurité pour l'instance principale : [sg-084d47618c8cec778](#)

Groupes de sécurité pour l'instance de blocs-notes : [sg-026a905ce64d353d9](#)

Balises de bloc-notes : creatorUserId = 383023238722 [Afficher tout/Modifier](#)

Emplacement du bloc-notes : s3://oc-ds-p8/

# 5 Déploiement dans le Cloud



Déploiement d'un modèle dans le Cloud



## Notebook Jupyter (noyau spark)

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer shows a file named 'oc-ds-p8.ipynb' with a last modified time of 'a minute ago'. The code editor shows the following code:

```
[*]: sc.install_pypi_package("pip==20.2.4")
sc.install_pypi_package("opencv-python")
sc.install_pypi_package("resize-image")
sc.install_pypi_package("boto3")
sc.install_pypi_package("pandas")
sc.install_pypi_package("scikit-learn")
sc.install_pypi_package("Pillow==7.0.0")
sc.install_pypi_package("pyquickhelper")
sc.list_packages()
```

Below the code editor, the 'Spark Job Progress' section is visible, showing the status of the Spark application. A yellow arrow points to the 'Starting Spark application' section.

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
1	application_1606603801331_0002	pyspark	idle	<a href="#">Link</a>	<a href="#">Link</a>	✓

SparkSession available as 'spark'.

Progress:

```
[ ]: #####
### PARAMETRES ###
#####

LOCAL = False # Serveur Local (True) - Cloud (False)
LOC_EXP = False # Export de RDD en fichier texte (uniquement si LOCAL=True)
FOLDER = 'data/fruits_360_v3b/' # Chemin du jeu de données
MIN_PARTITION = 18 # Nombre minimum de partitions (RDD)
OPENCV = 'sift' # Méthode de calcul des descripteurs: 'sift' ou 'orb'
CLS = True # Lancement du processus de classification (True)
```

# 5 Déploiement dans le Cloud



Déploiement d'un modèle dans le Cloud



## Notebook Jupyter (noyau PySpark)

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The code editor contains the following PySpark code:

```
title("> > > Traitements finalisés < < <",1)

if LOCAL:
    title("> > > PAUSE < < <",1)
    input("press ctrl+c to exit")
```

Below the code editor, the Spark Job Progress is displayed. It shows the progress of three jobs:

- Job [214]: count at <stdin>:116**
  - Progress for count at <stdin>:116: Job Progress: 13/13 Tasks Complete
  - Stage [625]: coalesce at Nati...java:0: COMPLETE, Task Progress: 1/1, Elapsed Time (seconds): 9.463
  - Stage [626]: count at <stdin>:116: COMPLETE, Task Progress: 12/12, Elapsed Time (seconds): 0.158
- Job [215]: count at <stdin>:116**
  - Progress for count at <stdin>:116: Job Progress: 12/12 Tasks Complete
  - Stage [627]: coalesce at Nati...java:0: SKIPPED, Task Progress: 0/1, Elapsed Time (seconds): n/a
  - Stage [628]: count at <stdin>:116: COMPLETE, Task Progress: 12/12, Elapsed Time (seconds): 1.259
- Job [216]: collect at <stdin>:327**

A yellow arrow points to the Spark Job Progress section.



# 5 Déploiement dans le Cloud



vs

localhost

Déploiement d'un modèle dans le Cloud

## Résultats comparatifs

Durée totale de traitement: 00 h 09 m 36 s

Durée des opérations

	Opération	Durée
0	Récupération des images	0.09
1	Extraction des descripteurs des images	186.44
2	Clustering K-Means	133.76
3	Prédiction K-Means	30.34
4	Création du bag of words	27.72
5	Réduction de dimension	82.06
6	Récupération des images - Test	0.08
7	Extraction des descripteurs des images - Test	61.59
8	Prédiction K-Means - Test	10.25
9	Création du bag of words - Test	10.35
10	Réduction de dimension - Test	33.51
11	Fin des traitements	0.00

Bag of words après réduction de dimension (df\_lab\_features)

	label	0	1	2	3	4	5
0	1.0	6.024672	-0.337976	-4.901770	-2.729916	-1.297712	-3.308183
1	1.0	6.371121	-0.153607	-2.852417	-3.309263	-3.896041	-2.565207
2	0.0	3.488277	3.316177	1.005654	0.826375	0.506713	-1.856883
3	1.0	19.223620	-4.594866	-1.225558	2.574097	-7.756241	-6.299376
4	0.0	19.091069	-2.171380	-0.814867	2.847041	-6.719943	-4.418533
5	1.0	15.015931	18.144498	-1.183434	0.501897	-3.000527	-6.548748
6	3.0	27.552906	0.646730	11.393523	-3.724111	1.212250	-7.387620
7	0.0	3.121793	2.597990	-0.193969	-7.074767	-6.144737	-9.125464
8	1.0	11.660417	17.784483	2.943873	0.971977	-6.908611	-5.676366

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (876, 36)

Durée totale de traitement: 00 h 25 m 32 s

Durée des opérations

	Opération	Durée
0	Récupération des images	0.03
1	Extraction des descripteurs des images	70.93
2	Clustering K-Means	962.98
3	Prédiction K-Means	198.57
4	Création du bag of words	40.38
5	Réduction de dimension	96.95
6	Récupération des images - Test	0.02
7	Extraction des descripteurs des images - Test	20.79
8	Prédiction K-Means - Test	72.04
9	Création du bag of words - Test	15.43
10	Réduction de dimension - Test	54.81
11	Fin des traitements	0.00

Bag of words après réduction de dimension (df\_lab\_features)

	label	0	1	2	3	4	5
0	4.0	2.310509	1.823638	2.864982	-3.205206	-2.444131	0.527192
1	4.0	9.568033	3.034005	-1.136817	-2.721742	-3.005855	-3.177016
2	0.0	16.699686	-0.891379	0.004849	-11.215860	2.667583	-2.249117
3	0.0	27.952051	3.035560	10.499386	-1.128175	2.487680	-1.243211
4	4.0	3.990037	4.013742	0.944926	-2.764598	-1.226985	-2.875845
5	1.0	5.225382	0.385623	1.045589	-2.130795	2.702250	-2.184826
6	2.0	15.204800	15.778906	0.060568	-4.207583	-0.553880	1.574485
7	0.0	26.378407	3.498933	8.153155	0.605576	3.552638	-0.156810
8	0.0	20.058290	-0.906470	0.237750	-7.645144	-2.660374	-2.944669

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (876, 36)



## Redimensionnement du Cluster EMR

Cluster : oc-ds-p8 **En attente** Cluster ready after last step completed.

[Récapitulatif](#)[Historique de l'application](#)[Surveillance](#)[Matériel](#)[Configurations](#)[Événements](#)[Étapes](#)[Actions d'amorçage](#)[Ajouter un groupe d'instances de tâches](#)

Groupes d'instances

Filtre :  2 groupes d'instances (tous chargés) 

ID	Status	Nom et type de nœud	Type d'instance	Nombre d'instances
▶ <a href="#">ig-WA66DB11HVHC</a>	Redimensionnement (4 demandées)	<b>CORE</b> Core Instance Group	<b>m5.xlarge</b> 4 Cœurs virtuels, 16 GiO de mémoire, stockage EBS uniquement Stockage sur EBS : 64 Gio	2 Instances <a href="#">Redimensionner</a>   <a href="#">Arrêter</a>
▶ <a href="#">ig-2FYSG6LRE40DT</a>	En cours d'exécution	<b>MASTER</b> Master Instance Group	<b>m5.xlarge</b> 4 Cœurs virtuels, 16 GiO de mémoire, stockage EBS uniquement Stockage sur EBS : 64 Gio	1 Instances



# 5 Déploiement dans le Cloud



## Déploiement d'un modèle dans le Cloud

Durée totale de traitement: 00 h 09 m 36 s

Durée des opérations

	Opération	Durée
0	Récupération des images	0.09
1	Extraction des descripteurs des images	186.44
2	Clustering K-Means	133.76
3	Prédiction K-Means	30.34
4	Création du bag of words	27.72
5	Réduction de dimension	82.06
6	Récupération des images - Test	0.08
7	Extraction des descripteurs des images - Test	61.59
8	Prédiction K-Means - Test	10.25
9	Création du bag of words - Test	10.35
10	Réduction de dimension - Test	33.51
11	Fin des traitements	0.00

Bag of words après réduction de dimension (df\_lab\_features)

	label	0	1	2	3	4	5
0	1.0	6.024672	-0.337976	-4.901770	-2.729916	-1.297712	-3.308183
1	1.0	6.371121	-0.153607	-2.852417	-3.309263	-3.896041	-2.565207
2	0.0	3.488277	3.316177	1.005654	0.826375	0.506713	-1.856883
3	1.0	19.223620	-4.594866	-1.225558	2.574097	-7.756241	-6.299376
4	0.0	19.091069	-2.171380	-0.814867	2.847041	-6.719943	-4.418533
5	1.0	15.015931	18.144498	-1.183434	0.501897	-3.000527	-6.548748
6	3.0	27.552906	0.646730	11.393523	-3.724111	1.212250	-7.387620
7	0.0	3.121793	2.597990	-0.193969	-7.074767	-6.144737	-9.125464
8	1.0	11.660417	17.784483	2.943873	0.971977	-6.908611	-5.676366

2 instances  
(8 cœurs)

4 instances  
(16 cœurs)

Durée totale de traitement: 00 h 08 m 20 s

Durée des opérations

	Opération	Durée
0	Récupération des images	0.08
1	Extraction des descripteurs des images	187.22
2	Clustering K-Means	83.63
3	Prédiction K-Means	16.96
4	Création du bag of words	30.09
5	Réduction de dimension	74.94
6	Récupération des images - Test	0.08
7	Extraction des descripteurs des images - Test	61.18
8	Prédiction K-Means - Test	5.99
9	Création du bag of words - Test	10.50
10	Réduction de dimension - Test	30.33
11	Fin des traitements	0.00

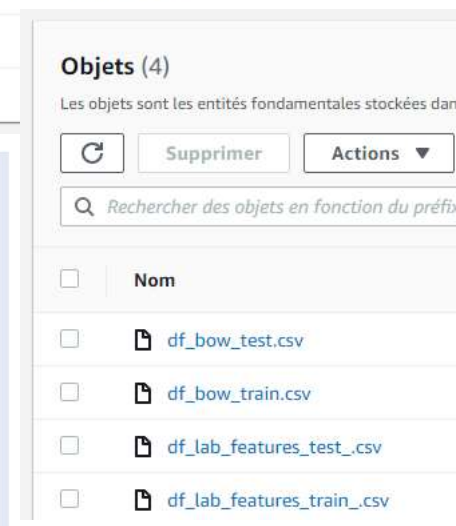
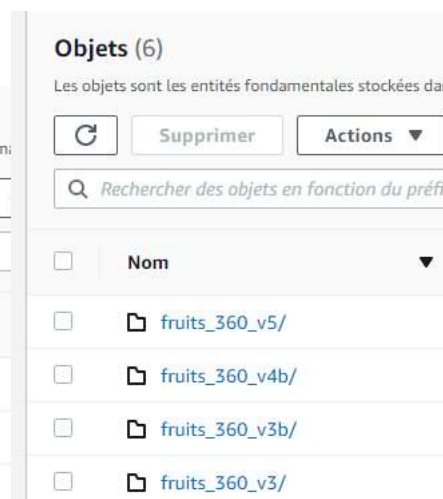
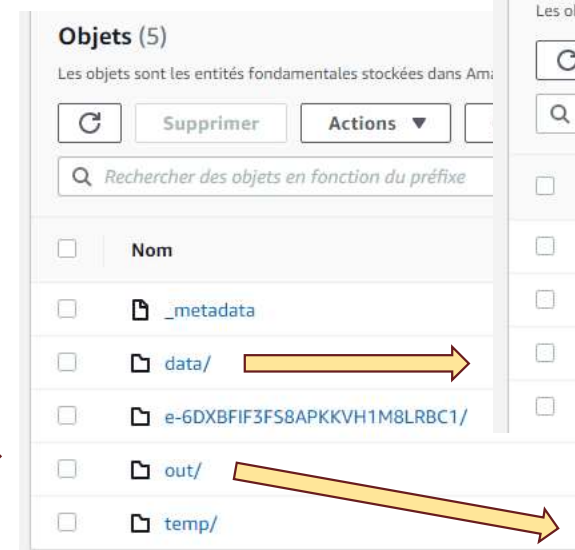
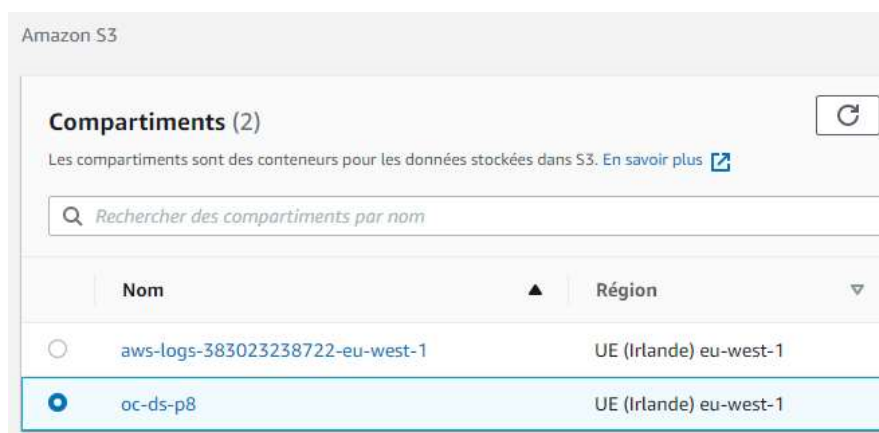
Bag of words après réduction de dimension (df\_lab\_features)

	label	0	1	2	3	4	5
0	2.0	25.386380	1.323169	11.145485	3.024961	-0.307836	-5.314542
1	2.0	15.018282	18.741854	7.108276	-4.578584	1.092393	-2.340866
2	2.0	-3.533843	3.123953	5.946621	-2.436330	3.546601	-6.404101
3	3.0	7.418447	1.077152	-3.034161	-1.225032	3.069555	-3.419146
4	0.0	22.732754	-1.599416	4.438273	-4.354162	0.923390	-1.317509
5	0.0	19.515657	-1.485339	1.508824	-5.699662	1.745299	-3.071305
6	2.0	21.782525	-1.878628	3.117757	-6.511164	-0.090578	-6.319000
7	0.0	-3.922658	2.620455	7.455179	-4.533826	-0.530551	-3.939859
8	1.0	13.567361	21.039295	3.744701	-5.634244	4.886197	-0.780661

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (876, 36)

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (876, 36)

## AWS S3 (Simple Storage Service)



```
df_lab_features = sdf_lab_features.toPandas()

# On exporte le DataFrame Pandas
if LOCAL:
    df_lab_features.to_csv(os.path.abspath("out/df_lab_features_" + data_set_name + ".csv"), index=False)
else:
    csv_buffer = StringIO()
    df_lab_features.to_csv(csv_buffer)
    s3_resource = boto3.resource('s3')
    s3_resource.Object('oc-ds-p8', "out/df_lab_features_" + data_set_name + ".csv") \
        .put(Body=csv_buffer.getvalue(), ACL='public-read')
```





## AWS S3 (Simple Storage Service)

```
image_path = s3://oc-ds-p8/data/fruits_360_v3b/Training/Corn/,
             s3://oc-ds-p8/data/fruits_360_v3b/Training/Orange/,
             s3://oc-ds-p8/data/fruits_360_v3b/Training/Raspberry/
```

### Affichage console

Bag of words après réduction de dimension (df\_lab\_features)

```
=====
label      0      1      2      3      4      5
0    2.0    0.330376 -4.698919 -2.556379 -1.753178 -1.902630 1.298467
1    0.0    0.153717 -3.672065  0.463361 -1.445074 -0.231556 0.819848
2    2.0    3.441013  2.309796 -1.730758 -3.248047 -4.133691 -0.994865
3    1.0   34.514508  0.356139  7.945553 -5.846689 -5.059411  2.386031
4    0.0    4.573013  7.235424  0.762018 -5.468033 -1.932407  2.238392
5    2.0   -0.516230 -3.932645 -2.017393 -2.830665 -3.392697  0.640334
6    1.0    4.137105  0.086613 -2.316639 -5.521217 -4.984313 -0.406035
7    1.0   31.670925 -0.758153 -6.523613 -0.356006 -0.143185  0.033581
8    0.0   27.778097 -0.274003 -2.831368  2.885279 -8.110952  3.122179
```

Dimensions du nouveau jeu de données avec les étiquettes (df\_lab\_features): (150, 22)

### Export fichier

df\_lab\_features\_train.csv

	A	B	C	D	E	F	G	H
1		label	0	1	2	3	4	5
2	0	2.0	0,330376	-4,698919	-2,556379	-1,753178	-1,902630	1,298467
3	1	0.0	0,153717	-3,672065	0,463361	-1,445074	-0,231556	0,819848
4	2	2.0	3,441013	2,309796	-1,730758	-3,248047	-4,133691	-0,994865
5	3	1.0	34,514508	0,356139	7,945553	-5,846689	-5,059411	2,386031
6	4	0.0	4,573013	7,235424	0,762018	-5,468033	-1,932407	2,238392
7	5	2.0	-0,516230	-3,932645	-2,017393	-2,830665	-3,392697	0,640334
8	6	1.0	4,137105	0,086613	-2,316639	-5,521217	-4,984313	-0,406035
9	7	1.0	31,670925	-0,758153	-6,523613	-0,356006	-0,143185	0,033581
10	8	0.0	27,778097	-0,274003	-2,831368	2,885279	-8,110952	3,122179





Processeur double cœur (4 processeurs logiques)

← → ↺ 🏠 localhost:4040/jobs/?&completedJob.page=4&comp

APACHE **Spark** 3.0.1

Jobs Stages Storage Environment Executors SQL

## Spark Jobs (?)

User: eric  
Total Uptime: 46 min  
Scheduling Mode: FIFO  
Completed Jobs: 380

▼ Event Timeline  
☐ Enable zooming

Executors					
<div>Added</div> <div>Removed</div>					
		Executor driver added			

Jobs					
<div>Succeeded</div> <div>Failed</div> <div>Running</div>					

## ▼ Completed Jobs (380)

Page: 1 2 3 4 &gt;

Job Id ▼	Description
379	toPandas at <ipython-input-1-3d75f7cead2d>:793 toPandas at <ipython-input-1-3d75f7cead2d>:793
378	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0
377	collect at StringIndexer.scala:204 collect at StringIndexer.scala:204
...	
4	takeSample at KMeans.scala:347 takeSample at KMeans.scala:347
3	collect at <ipython-input-1-3d75f7cead2d>:332 collect at <ipython-input-1-3d75f7cead2d>:332
2	collect at <ipython-input-1-3d75f7cead2d>:327 collect at <ipython-input-1-3d75f7cead2d>:327
1	count at <ipython-input-1-3d75f7cead2d>:116 count at <ipython-input-1-3d75f7cead2d>:116
0	count at <ipython-input-1-3d75f7cead2d>:116 count at <ipython-input-1-3d75f7cead2d>:116



```
=====
Calcul des descripteurs
=====
```

```
Chargement des images (rdd_images)
=====
```

```
MapPartitionsRDD[4] at javaToPython at NativeMethodAccessorImpl.java:0
```

```
Nombre de partitions: 5
Dimension: 150
```

```
Catégories / Images / Descripteurs (rdd_cat_ima_desc)
=====
```

```
PythonRDD[6] at RDD at PythonRDD.scala:53
```

```
Catégories / Images / Descripteurs (rdd_cat_ima_desc_f)
=====
```

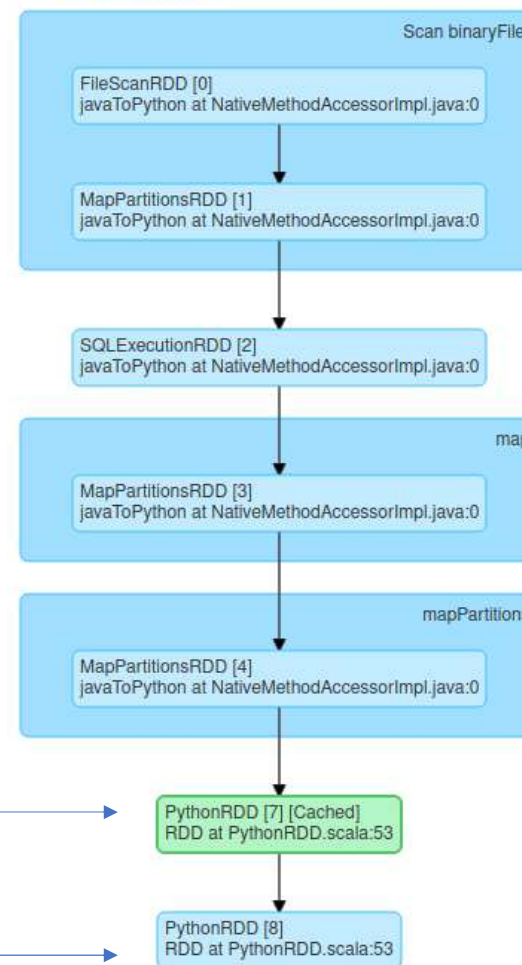
```
PythonRDD[7] at RDD at PythonRDD.scala:53
```

```
Catégories (rdd_cat)
=====
```

```
PythonRDD[8] at RDD at PythonRDD.scala:53
```

▼ DAG Visualization

Stage 2



## Details for Job 2

**Status:** SUCCEEDED**Completed Stages:** 1

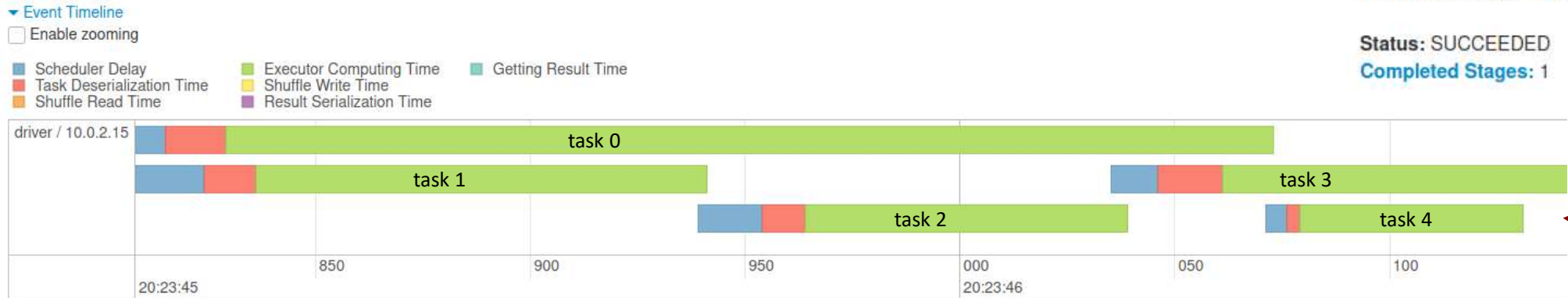


Processeur double cœur (4 processeurs logiques)

## Details for Job 2

Status: SUCCEEDED

Completed Stages: 1



1 tâche par processeur (executor) à la fois

## Aggregated Metrics by Executor

Show 20 entries

Executor ID	Logs	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks
driver		10.0.2.15:45149	0.7 s	5	0	0	5

Tasks (5)

Show 20 entries

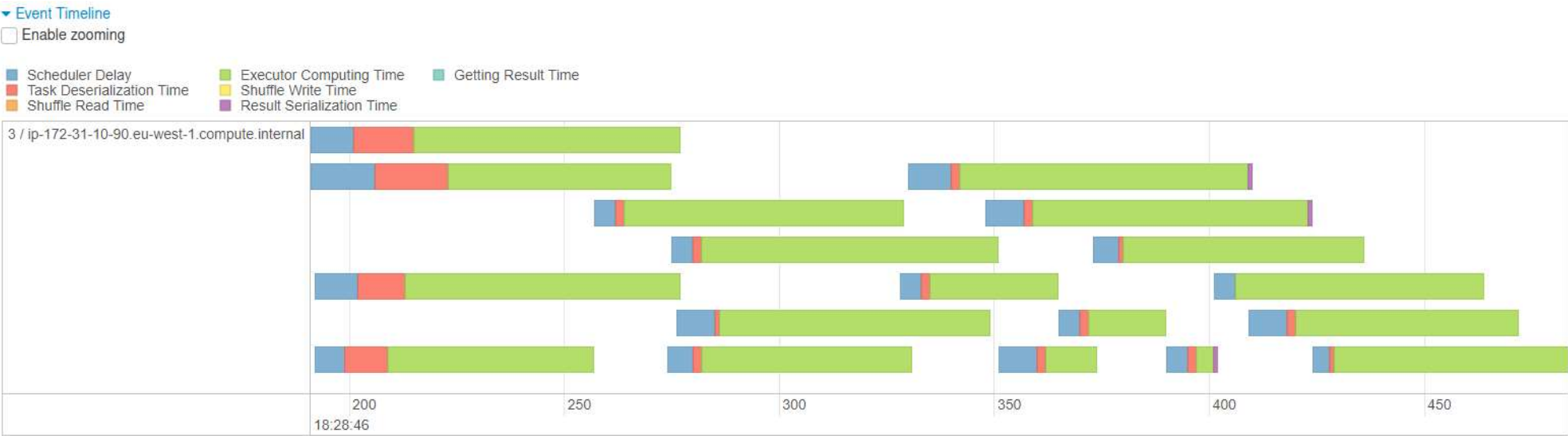
Search:

Index	Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Input Size / Records
0	20	0	SUCCESS	PROCESS_LOCAL	driver	10.0.2.15		2020-12-03 20:23:45	0.2 s		1.5 MiB / 32
1	21	0	SUCCESS	PROCESS_LOCAL	driver	10.0.2.15		2020-12-03 20:23:45	0.1 s		1005.8 KiB / 23
2	22	0	SUCCESS	PROCESS_LOCAL	driver	10.0.2.15		2020-12-03 20:23:45	75.0 ms		173.9 KiB / 7
3	23	0	SUCCESS	PROCESS_LOCAL	driver	10.0.2.15		2020-12-03 20:23:46	92.0 ms		304.7 KiB / 10
4	24	0	SUCCESS	PROCESS_LOCAL	driver	10.0.2.15		2020-12-03 20:23:46	52.0 ms		217 KiB / 7

# 6 Contrôle Web UI



2 instances (8 processeurs logiques)



▼ Tasks (18)

Index ▾	ID	Attempt	Status	Locality Level	Executor ID	Host		Launch Time	Duration
17	70	0	SUCCESS	PROCESS_LOCAL	3	ip-172-31-10-90.eu-west-1.compute.internal	<a href="#">stdout</a> <a href="#">stderr</a>	2020/12/03 18:28:46	55 ms
16	69	0	SUCCESS	PROCESS_LOCAL	3	ip-172-31-10-90.eu-west-1.compute.internal	<a href="#">stdout</a> <a href="#">stderr</a>	2020/12/03 18:28:46	52 ms
15	68	0	SUCCESS	PROCESS_LOCAL	3	ip-172-31-10-90.eu-west-1.compute.internal	<a href="#">stdout</a> <a href="#">stderr</a>	2020/12/03 18:28:46	58 ms



## Optimisation du code

- Analyse des tâches
- Optimisation

## Dimension de l'architecture

- Serveurs de calcul
  - Nombre de cores
  - Nombre de partitions
- Stockage
  - Croissance du volume des images
  - Dimension des images

## Modèle de classification

- Choix d'un modèle
  - Performances

## Configuration de l'architecture Big Data

- Complexe
  - Ressources: Data Architect...
  - Services: AWS...

## Programmation calcul distribué

- Plusieurs langages dont Python
  - Nouvelle syntaxe (PySpark)
- Optimisation
  - Liée à l'architecture

## Le Big Data: plus que des big data

- Transformation des processus métiers