Fruits!

Déployez un modèle dans le Cloud

Création le 18/02/2020

Julien Di Giulio

SOMMAIRE

I - PRESENTATION

PRESENTATION DU PROJET

Fruits! : StartUp faisant partie de l'Agritech

Fruits!

Agritech : Mettre la technologie au profit de l'agriculture

Projet final de Fruits! : Développer des robots cueilleurs intelligents.

Première phase du projet : Se faire connaître du grand public

1. Création d'une application mobile permettant de reconnaître un fruit sur une photo prise par l'utilisateur.

2. Afficher les informations disponibles sur ce fruit

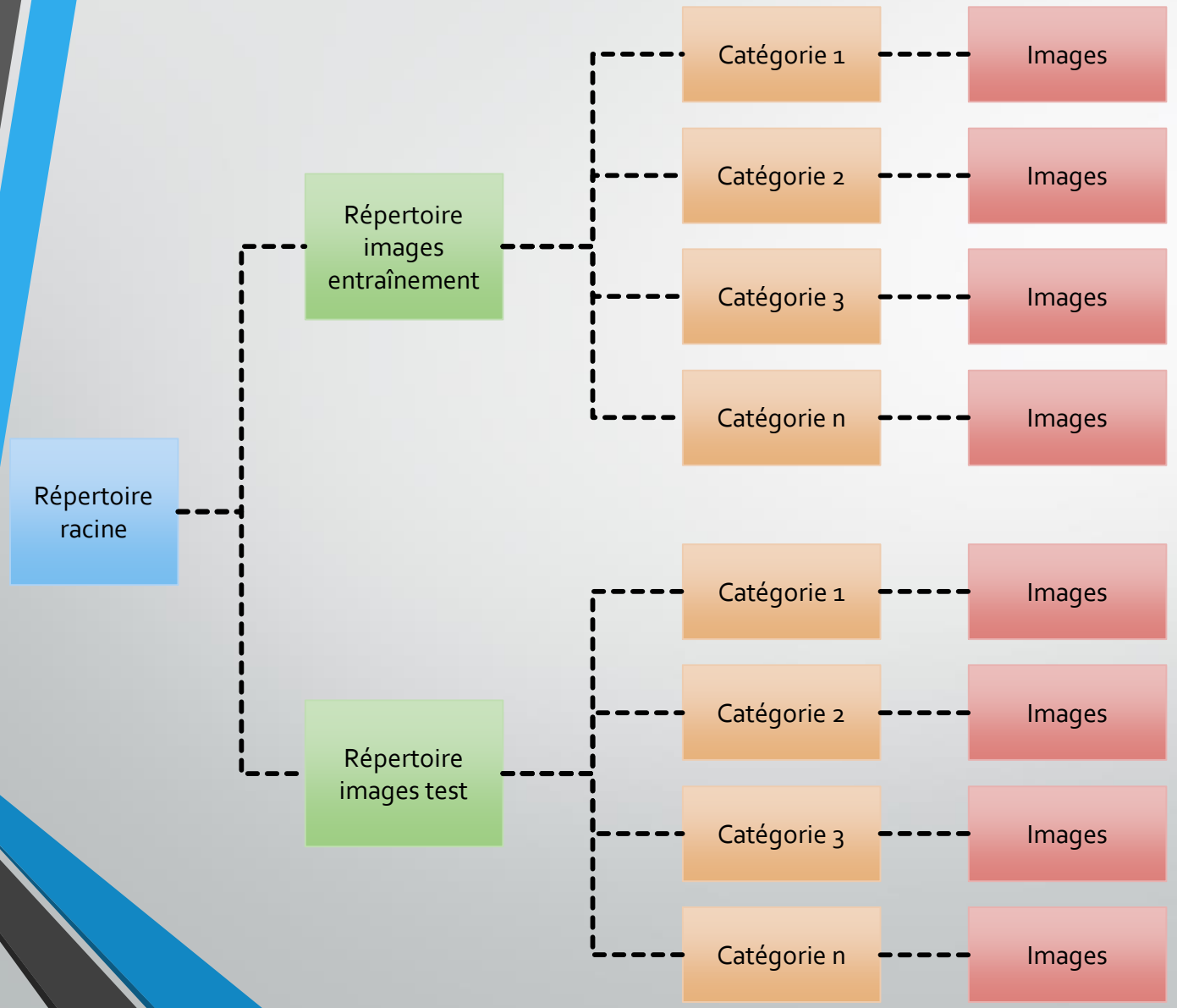
Mission : Développer une première chaîne de traitement des données comprenant le preprocessing et une étape de réduction de dimension.

- Prendre en compte le passage à l'échelle des calculs du fait que le volume de données va augmenter très rapidement après la livraison du projet.
- Ecriture des scripts en Pyspark.
- Déploiement du modèle sur le Cloud.

Données : Banques d'images de fruits avec les labels associés

https://www.kaggle.com/moltean/fruits

PRESENTATION DES DONNEES



- Taille set d'entraînement : **61488** images

Chaque image ne représente qu'un seul fruit ou légume

- Nombre de catégories : **120** (fruits et légumes)
- Taille d'une image : **100x100** pixels
- Images en couleurs avec fruits centrés

- Taille set de test : **20622** images

Chaque image ne représente qu'un seul fruit ou légume

- Nombre de catégories : **120** (fruits et légumes)
- Taille d'une image : **100x100** pixels
- Images en couleurs avec fruits centrés

II – LE BIG DATA

LE BIG DATA, C'EST QUOI?

Le Big Data très grossièrement définit :

- On fait du Big Data à partir du moment où la quantité de données excède la faculté d'une machine à les stocker et les analyser en un temps acceptable.
- Exemple : Si les données à traiter sont trop grosses pour être stockées dans la RAM de la machine, nous sommes confrontés à un problème de Big Data.

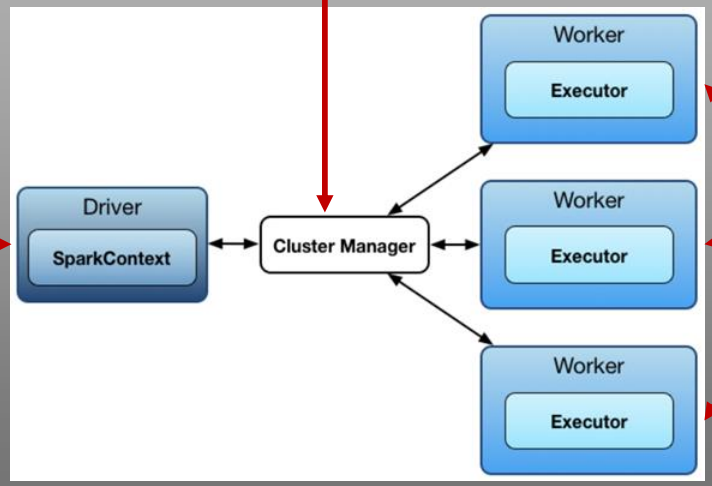
Comment faire pour traiter ces données :

- La solution à ce problème consiste à paralléliser les calculs sur plusieurs machines différentes. Cela pose un certain nombre de questions qui ne sont pas simples à résoudre :
  - La stratégie de distribution des calculs entre les machines
  - Comment distribuer les calculs entre les machines
  - Comment agréger les résultats des différentes machines
  - Gérer les pannes des machines lors de l'exécution des calculs
  - Maîtriser les coûts de l'architecture mis en place.

Logiciel de calculs distribués :

L'objet Sparkcontext va créer une machine virtuelle responsable de la mise en œuvre de l'application et de la distribution des calculs

Application Python exécutée dans une JVM dans laquelle est instanciée et configurée l'objet Sparkcontext et/ou Sparksession



Les Workers ou Executeurs sont les machines virtuelles qui sont en charge de l'exécution des calculs.

Spark peut lancer un traitement sur une machine locale ou sur une machine en ligne.
En mode local, le parallélisme des calculs se fera en fonction du nombre de cœurs disponibles sur la machine

CODE AVEC PYSPARK

Liste des images dans l'arborescence

Création d'un Spark Dataframe (RDD) avec les chemins de tous les fichiers images à traiter

Extraction de la catégorie

Création d'une nouvelle colonne contenant la catégorie de chaque image

Chargement des images

Création d'une nouvelle colonne contenant le vecteur initial de chaque image

Réduction dimensionnelle

Création d'une nouvelle colonne contenant l'extraction des descripteurs de chaque image

Création du moteur Spark

```
try:
    sc = SparkContext.getOrCreate()
    # On stoppe tous les messages INFO et WARN dans la console
    sc.setLogLevel("WARN")

    spark = SparkSession.builder.appName("name").getOrCreate()

except:
    print("Erreur à la construction du moteur Spark")
```

Définition RDD

Resilient Distributed Datasets (RDDs) : Un RDD permet de réutiliser efficacement les données. C'est une collection partitionnée d'enregistrements.

- *Ils sont créés par des opérations appelés « Transformations » (Map, filter, join, …).*
- *Ils sont utilisés dans des opérations de calculs appelées « Actions » (count, collect, show, …).*

Exemple : Chargement des images

TRANSFORMATION

Création colonne Dataframe

Une solution pour créer une colonne en pySpark est de créer une UDF

```
df = df.withColumn("descriptors", udf_desc("path_img"))
```

UDF (User Defined Function)

Permet de distribuer le traitement demandé à chaque ligne du Dataframe

```
udf_image = udf(read_image, ArrayType(IntegerType()))
```

Fonction de calcul

C'est dans cette fonction que sont réalisés les calculs voulus par le développeur

```
def read_image(img):
    """
    Cette fonction permet de charger les images avec Pillow
    afin d'en extraire le vecteur
    :param img:
    :return vecteur au format liste:
    """
    return image
```

ACTION

Affichage Dataframe

C'est l'action .show() ici qui permet de lancer toute la procédure ci-dessus

```
df.show(5)
```

III – ARCHITECTURE AWS

SERVICES UTILISES

Amazon IAM

Configuration des « rôles » sur notre compte AWS pour permettre à nos différentes instances de communiquer entres elle.

Amazon S3

Espace de stockage dans lequel sont placés :

- Les BDD
- L'application
- Les fichiers de configuration de l'architecture

Amazon EMR

Construction d'un cluster EMR (ElasticMapReduce) qui permet la distribution des calculs sur différentes instances EC2.

Notebook EMR

Création d'un Notebook EMR pour analyser les résultats obtenus.

Amazon EC2 maitre

L'instance EC2 est fournit par défaut avec Hadoop YARN qui permet d'effectuer la distribution des tâches.

Amazon EC2 esclave 1

Chaque instance EC2 est une machine louée à AWS pour effectuer les calculs

Amazon EC2 esclave 2

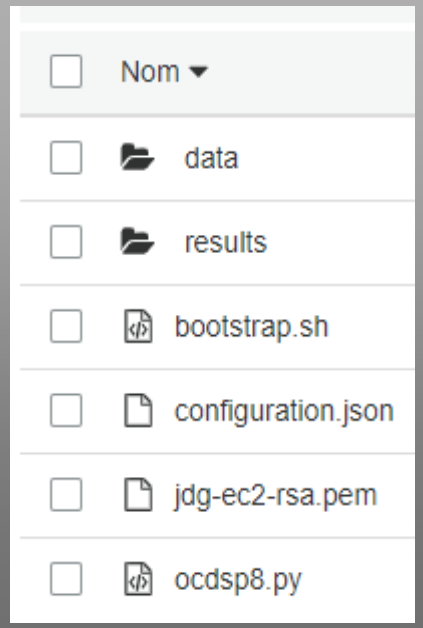
Chaque instance EC2 est un machine louée à AWS pour effectuer les calculs

Amazon EC2 esclave n

Chaque instance EC2 est un machine louée à AWS pour effectuer les calculs

STOCKAGE SUR S3

Aperçu fichiers stockés sur S3 :



Data : BDD d'images

Results : Résultats sauvegardés au format parquet

Jdg-ec2-rsa.pem : Clé privée pour pouvoir se connecter en SSH aux instances esclaves

Ocdsp8.py : Application du développeur

Bootstrap.sh :

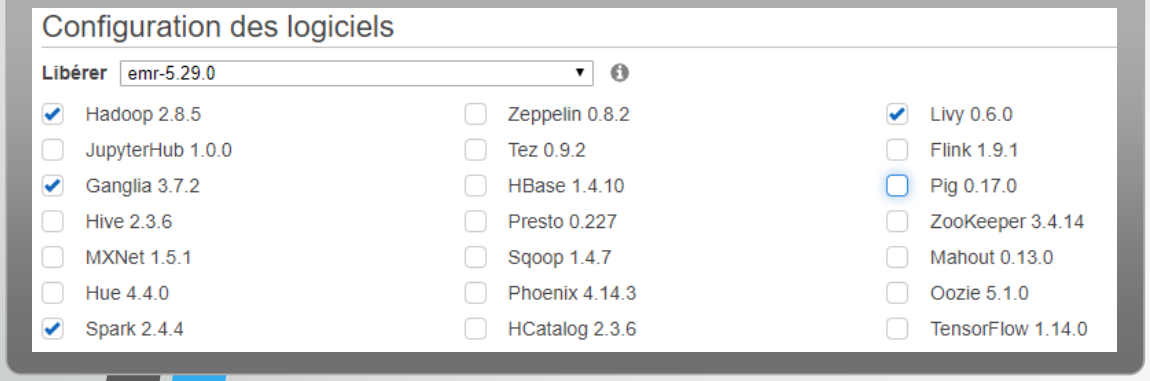
```
#! /bin/bash
sudo pip install opencv-python
sudo pip install boto3
sudo pip install pillow
sudo python3 -m pip install opencv-python
sudo python3 -m pip install boto3
sudo python3 -m pip install pillow
```

Configuration.json :

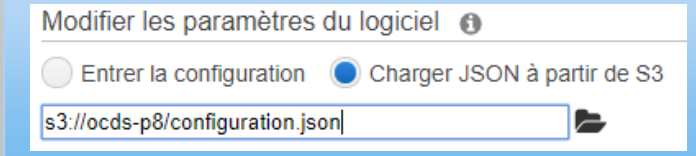
```
[
  {
    "Classification": "spark-env",
    "Configurations":
    [
      {
        "Classification": "export",
        "Properties":
        {
          "PYSPARK_PYTHON": "/usr/bin/python3"
        }
      }
    ]
  }
]
```

CREATION CLUSTER EMR

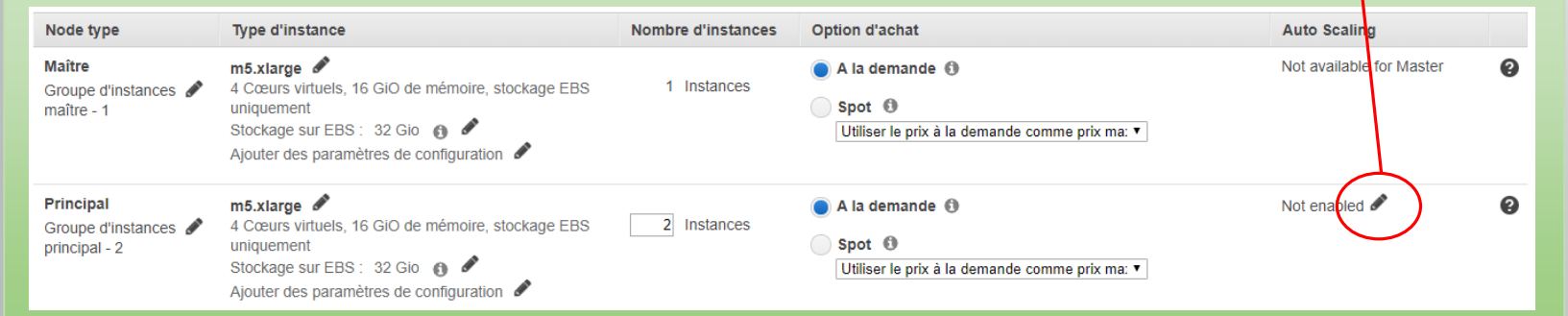
Etape 1: Sélection des logiciels à installer sur les machines



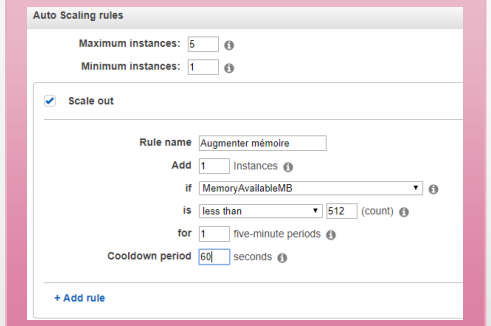
Etape 2: Modification des paramètres du logiciel



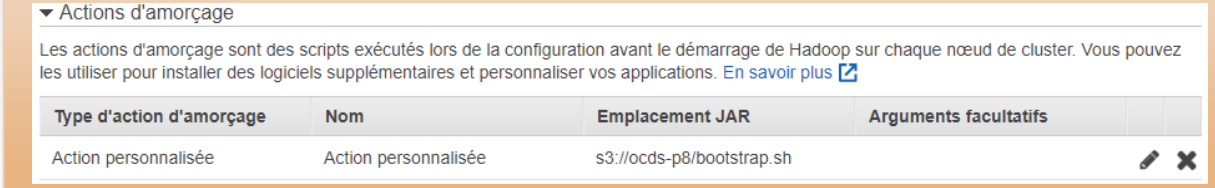
Etape 3: Configuration Hardware de l'architecture



Mise à l'échelle régulée



Etape 4: Initialisation de l'environnement



Etape 5: Configuration de la sécurité

Logiciels :

- Hadoop : AWE EMR implémente un cluster Hadoop
- Ganglia : Logiciel de mesure de performances des clusters
- Spark : Pour la parallélisation des calculs
- Livy : Nous permet d'utiliser les Notebooks EMR

AJOUT D'ETAPE

On renseigne le type d'étape au préalable, ce qui nous amène sur ce formulaire

On nomme l'étape

On choisit d'exécuter l'application sur la machine maître

<u>On peut configurer la ligne de commande d'exécution de Spark :</u>

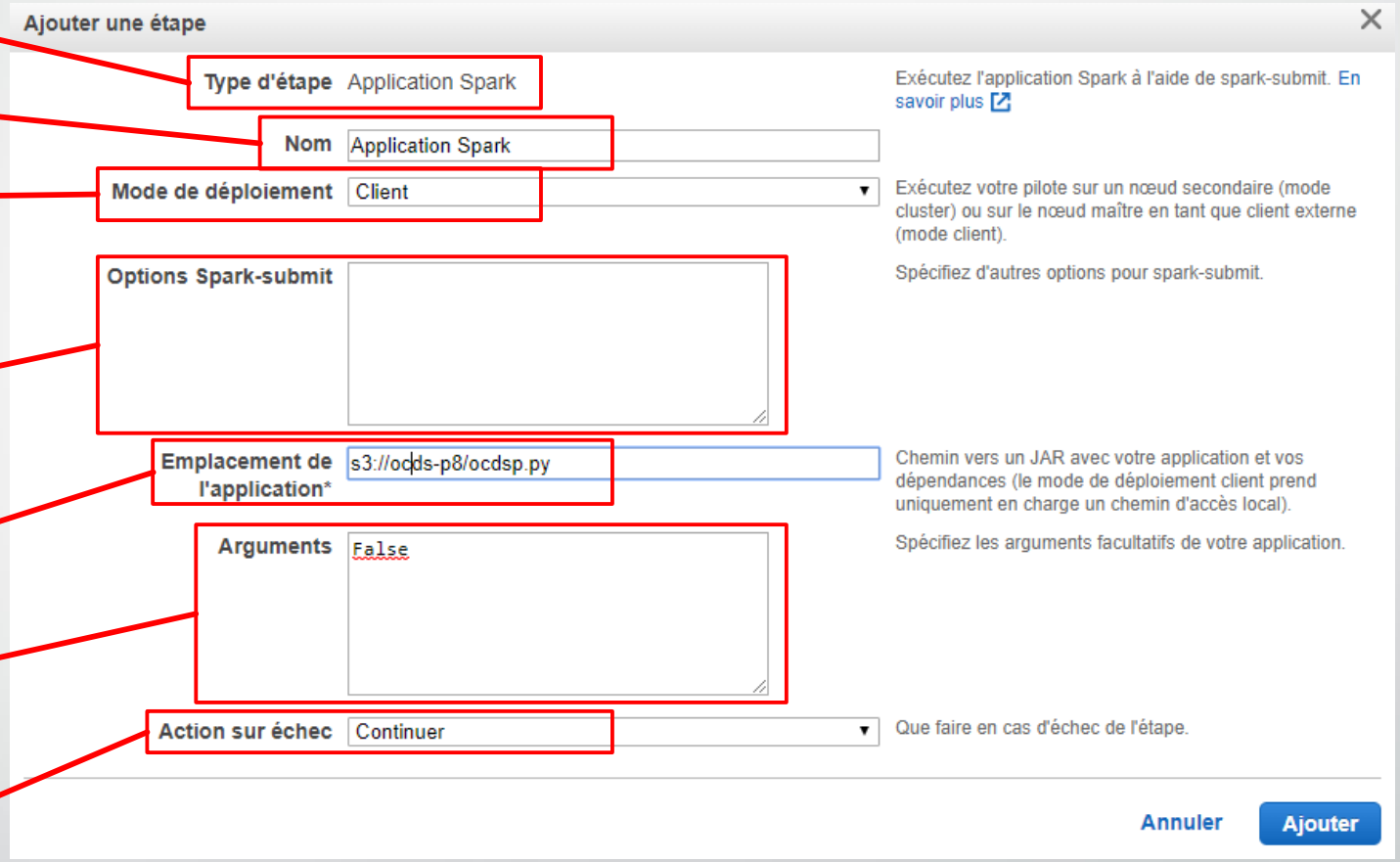
- Espace mémoire alloué, nombre de cœurs utilisés par machine, etc.
- Ne rien mettre correspond à la configuration par défaut de Spark

On sélectionne l'application stockée dans le S3

On rentre les arguments utiles au bon fonctionnement de l'application

<u>On sélectionne la réaction du Cluster en cas d'erreur dans l'application :</u>

- Laisser le cluster actif pour relancer une étape après correction
- Le résilier



<u>Ligne de commande pour executer l'application :</u>

spark-submit –deploy-mode client s3://ocds-p8/ocdsp8.py False

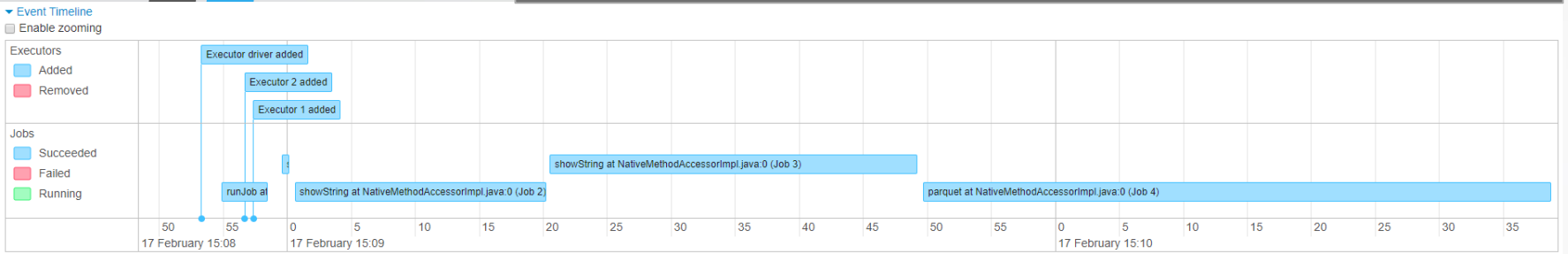
- True -> Travail en local
- False -> Travail sur AWS

IV – ANALYSE

HISTORIQUE SPARK

AWS EMR met à disposition une interface utilisateur du service d'historique Spark

Avantage : Pas besoin de connexion SSH



Completed Jobs (5)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4	parquet at NativeMethodAccessorImpl.java:0 parquet at NativeMethodAccessorImpl.java:0	2020/02/17 15:09:49	49 s	1/1	2/2
3	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2020/02/17 15:09:20	29 s	1/1	1/1
2	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2020/02/17 15:09:00	20 s	1/1	1/1
1	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2020/02/17 15:08:59	0.6 s	1/1	1/1
0	runJob at PythonRDD.scala:153 runJob at PythonRDD.scala:153	2020/02/17 15:08:54	4 s	1/1	1/1

Logout de l'application sauvegardé automatiquement sur S3

```
Nombre d'images chargÃ©es : 491
Temps d execution : 4.532979488372803 secondes ---
+----------------------------------------+
|path_img                                |
+----------------------------------------+
|s3://ocds-p8/data/applebraeburn/0_100.jpg   |
|s3://ocds-p8/data/applebraeburn/100_100.jpg|
|s3://ocds-p8/data/applebraeburn/101_100.jpg|
|s3://ocds-p8/data/applebraeburn/102_100.jpg|
|s3://ocds-p8/data/applebraeburn/103_100.jpg|
+----------------------------------------+
only showing top 5 rows

-----Extraction catÃ©gories images-----
-----Chargement images-----
+--------------------+------------+--------------------+
|            path_img|       categ|               image|
+--------------------+------------+--------------------+
|s3://ocds-p8/data...|applebraeburn|[254, 255, 255, 2...|
|s3://ocds-p8/data...|applebraeburn|[254, 255, 251, 2...|
|s3://ocds-p8/data...|applebraeburn|[253, 255, 250, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 251, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 250, 2...|
+--------------------+------------+--------------------+
only showing top 5 rows

-----Extraction descripteurs-----
+--------------------+------------+--------------------+--------------------+
|            path_img|       categ|               image|         descriptors|
+--------------------+------------+--------------------+--------------------+
|s3://ocds-p8/data...|applebraeburn|[254, 255, 255, 2...|[222, 254, 255, 2...|
|s3://ocds-p8/data...|applebraeburn|[254, 255, 251, 2...|[244, 172, 189, 1...|
|s3://ocds-p8/data...|applebraeburn|[253, 255, 250, 2...|[252, 124, 253, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 251, 2...|[220, 253, 189, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 250, 2...|[244, 100, 157, 1...|
+--------------------+------------+--------------------+--------------------+
only showing top 5 rows

-----Enregistrement rÃ©sultats-----
Enregistrement rÃ©ussi
Temps d execution : 49.74844026565552 secondes ---
Programme terminÃ©.
```

Possibilité de visualiser le partage des tâches sur les instances exclaves

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(3)	0	0.0 B / 12.6 GB	0.0 B	8	0	0	6	6	2.3 min (4 s)	0.0 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(3)	0	0.0 B / 12.6 GB	0.0 B	8	0	0	6	6	2.3 min (4 s)	0.0 B	0.0 B	0.0 B	0

Executors

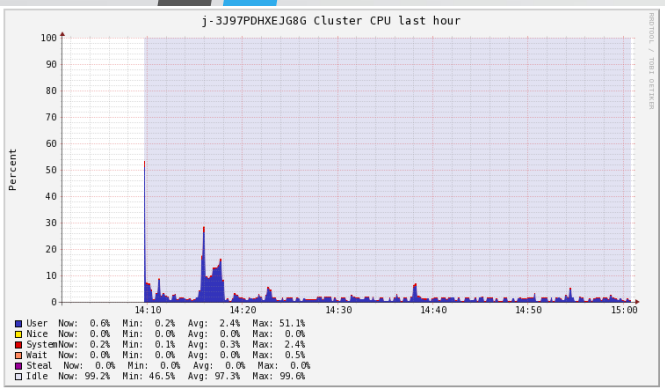
Show 20 entries Search:

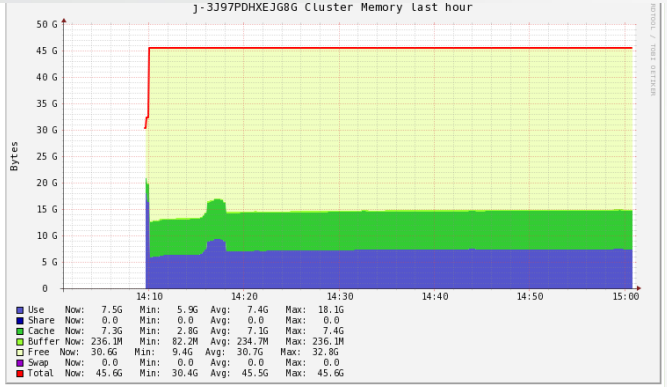
Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	ip-172-31-23-14.us-west-2.compute.internal:32809	Active	0	0.0 B / 1.1 GB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	
1	ip-172-31-16-20.us-west-2.compute.internal:43305	Active	0	0.0 B / 5.8 GB	0.0 B	4	0	0	2	2	57 s (2 s)	0.0 B	0.0 B	0.0 B	stdout stderr
2	ip-172-31-16-202.us-west-2.compute.internal:45005	Active	0	0.0 B / 5.8 GB	0.0 B	4	0	0	4	4	1.3 min (2 s)	0.0 B	0.0 B	0.0 B	stdout stderr

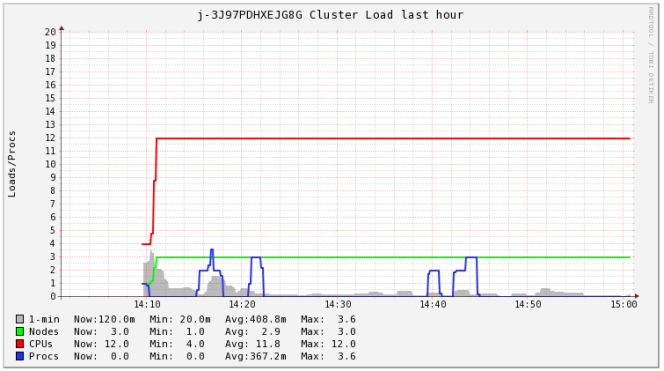
GANGLIA

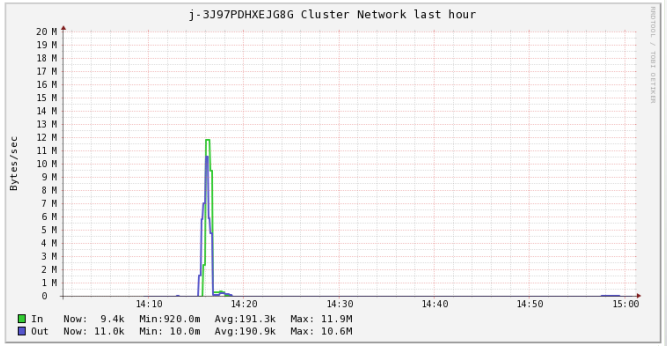
Ganglia est un outil de monitoring

Récemment , AWS a simplifié la procédure pour s'y connecter sans tunnel SSH



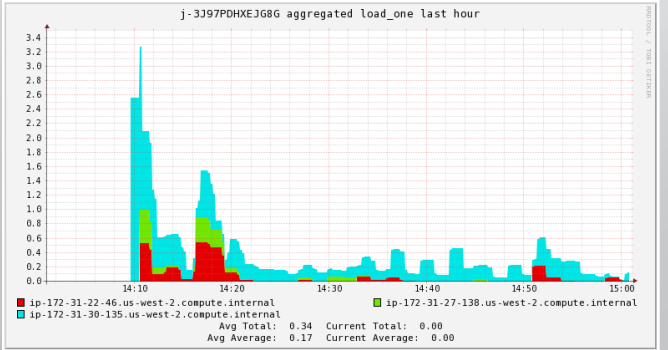






CPUs Total: **12**
Hosts up: **3**
Hosts down: **0**

Current Load Avg (15, 5, 1m):
1%, 1%, 1%
Avg Utilization (last hour):
3%



Ganglia permet d'analyser l'utilisation de chaque instance du cluster tant que celui-ci est en fonctionnement.

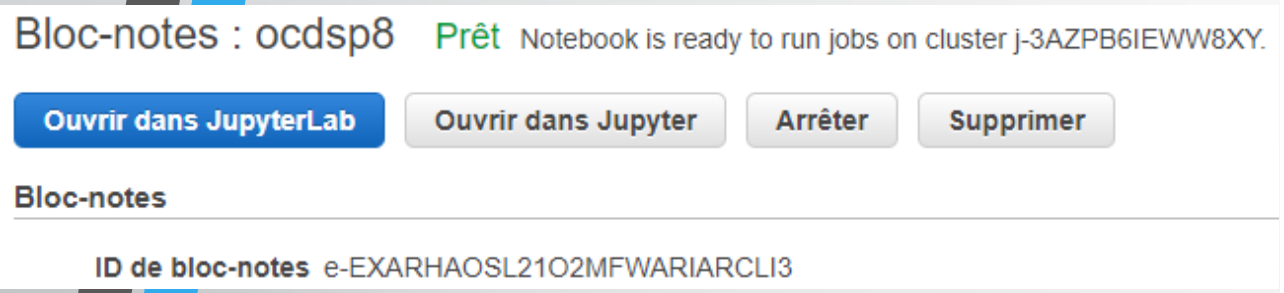
Procédure d'accès simplifiée à Ganglia

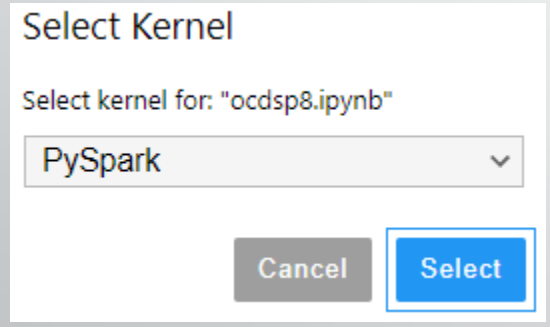
- Créer un tunnel SSH sur le port par défaut de Ganglia (8157) en tapant la ligne de commande : **ssh –i {Cle-rsa.pem} –ND 8157 {Adresse-publique-DNS-Master}**
- A l'aide d'un serveur proxy, autoriser l'accès au port de Ganglia
- Dans le navigateur web, tapez l'adresse : **http://{adresseDNS-master}:8157**

NOTEBOOK EMR

Qu'est-ce qu'un Notebook EMR :

- Un Notebook EMR permet d'interagir directement avec Spark
- Il est rattaché à un cluster en cours de fonctionnement
- Cette option n'est pas disponible dans toutes les régions






```
[1]: df = sqlContext.read.parquet("s3://ocds-p8/results")
```

▸ Spark Job Progress

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1582121362438_0002	pyspark	idle	Link	Link	✔

```
SparkSession available as 'spark'.
```

```
[16]: df.show(5)
```

▸ Spark Job Progress

```
+--------------------+------------+--------------------+--------------------+
|            path_img|       categ|               image|         descriptors|
+--------------------+------------+--------------------+--------------------+
|s3://ocds-p8/data...|applebraeburn|[254, 255, 255, 2...|[222, 254, 255, 2...|
|s3://ocds-p8/data...|applebraeburn|[254, 255, 251, 2...|[244, 172, 189, 1...|
|s3://ocds-p8/data...|applebraeburn|[253, 255, 250, 2...|[252, 124, 253, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 251, 2...|[220, 253, 189, 2...|
|s3://ocds-p8/data...|applebraeburn|[255, 255, 250, 2...|[244, 100, 157, 1...|
+--------------------+------------+--------------------+--------------------+
only showing top 5 rows
```

```
[5]: img = df.collect()[0]
```

•••

```
[14]: vect_mg_origine = len(img[2])

print("L'image originale a un vecteur de taille :", vect_mg_origine)
```

```
L'image originale a un vecteur de taille : 30000
```

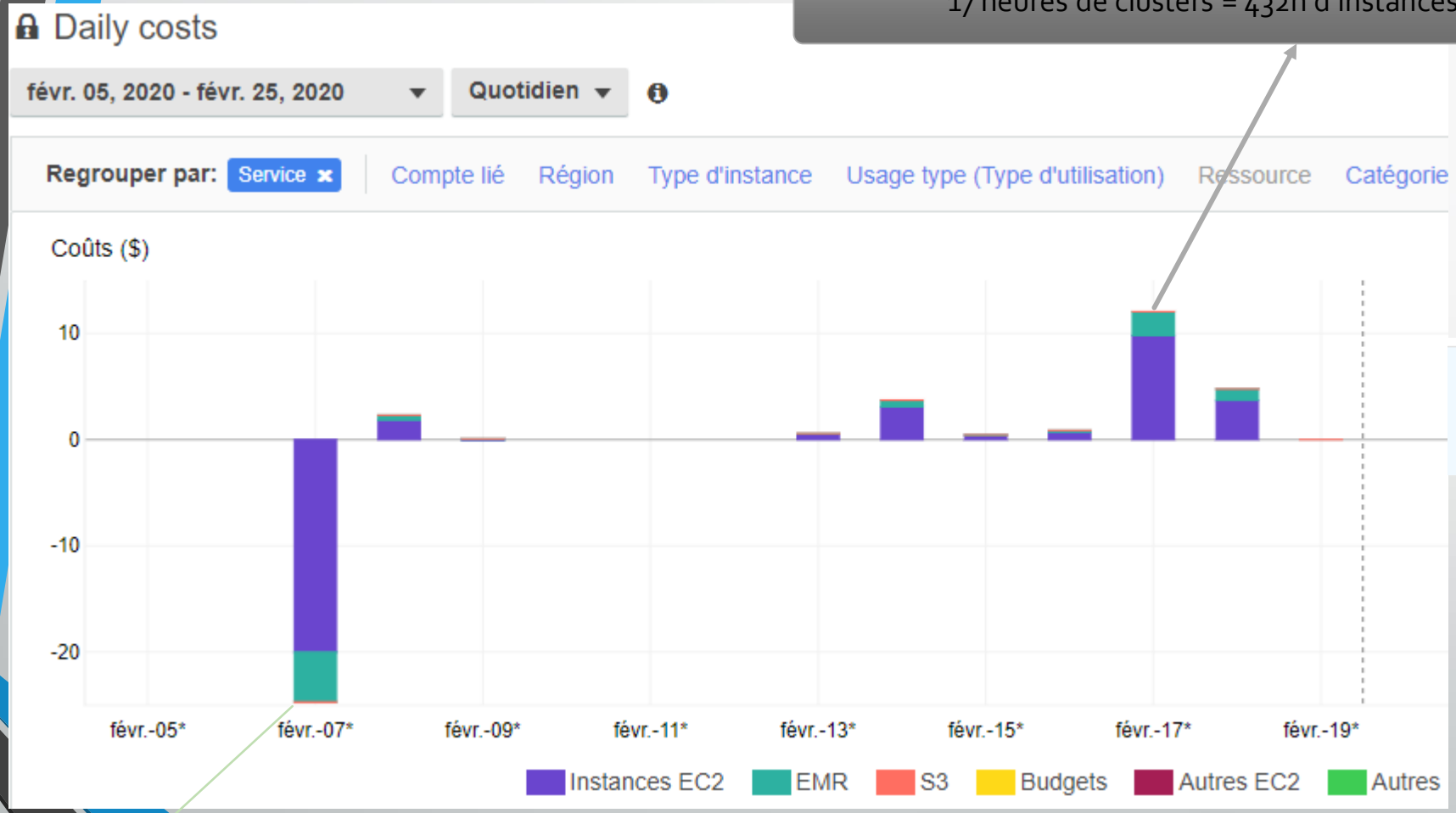
```
[13]: vect_img_desc = len(img[3])

print("Le vecteur descripteurs obtenu avec ORB a une taille de :", vect_img_desc)
```

```
Le vecteur descripteurs obtenu avec ORB a une taille de : 128
```

COUTS

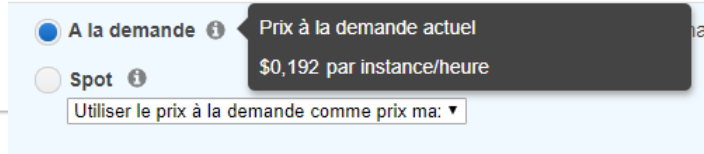
Pic de dépenses le 17/02/2020 car oubli de résiliation d'un cluster

17 heures de clusters = 432h d'instances EC2 facturées



Coût total pour tous les tests effectués sur le mois de février :

Environ 20$



Crédit suite à formation AWS

V – CONCLUSION

RESUME

APPLICATION

- Utilisation de pySpark pour anticiper la forte évolution de la base de données.
- Chargement d'images dans un Spark Dataframe.
- Réduction dimensionnelle avec l'algorithme ORB pour extraire les descripteurs.
- Sauvegarde des résultats au format parquet.
- Application utilisable en mode local ou en ligne.

DEPLOIEMENT SUR LE CLOUD

- Fournisseur choisit : AWS
- Services utilisés : S3 – EC2 – EMR – IAM – Notebook EMR
- EMR permettant une utilisation big data et une mise à l'échelle facilement
- Analyse des calculs avec les outils mis à disposition par AWS
- Analyse des coûts maîtrisés

AXES D'AMELIORATION

- **Application :**
  - Terminer le modèle de classification avec ORB :
    - Création de mots visuels par catégorie par l'utilisation d'un K-means
  - Utilisation des librairies incluses avec Spark :
    - Mllib pour le chargement d'images par exemple
  - Utilisation d'un réseau neuronal pré-entraîné (Transfer Learning) avec la librairie sparkdl
- **Architecture :**
  - Optimisation des coûts :
    - Choix de région : EMR et S3 dans la même région pour optimiser les échanges
    - Utilisation du cluster avec tarifications Spot. Les instances Spot EC2 sont moins couteuses, mais ne peuvent pas être utilisées à toute heure.

Sans tarification Spot	m5.xlarge	$0.106 par heure
Avec tarification Spot	m5.xlarge	$0.07 par heure

    - Optimiser la configuration du cluster EMR

QUESTIONS - REPONSES