



Fruits!

Déployer un modèle dans le cloud

Felipe PEREIRA DE LIMA

<https://github.com/feliplim/fruit-classification>

Date : 21/02/2024



Table de matières

- 1) Problématique et jeux de données
- 2) Processus de création de l'environnement *Big Data*
- 3) Chaîne de traitement d'images
- 4) Démonstration d'exécution du script PySpark sur le cloud

Environnement technique

- Python 3.11.6
- VS Code 1.84.2
- Librairies
 - Pandas & Numpy
 - PIL
 - Tensorflow
 - PySpark
- AWS Service
- JupyterHub
- FoxyProxy





1

Problématique et jeu de données



Fruits!

Problématique

- Fruits! est une jeune start-up de l'AgriTech qui cherche à proposer des **solutions innovantes pour la récolte des fruits**.
- Son idée : développer des **robots cueilleurs intelligents** qui permettraient des traitements spécifiques pour chaque espèce de fruits, afin de **préserver leur biodiversité**.
- Dans un premier temps, Fruits! Souhaite se faire connaître grâce à une **application mobile** qui permettrait aux utilisateurs de prendre en photos un fruit et d'obtenir des informations à son sujet
- Cette application permettrait de :
 - ◆ Sensibiliser le grand public à la **biodiversité** des fruits;
 - ◆ Mettre en place une première version d'un **moteur de classification** des images de fruits;
 - ◆ Construire une première version de **l'architecture Big Data** nécessaire.

La mission :

- 1) Mettre en place **l'architecture Big Data** nécessaire pour le passage à l'échelle en terme de volume de données.
- 2) Compléter la **chaîne de traitement** réalisée par un alternant, en utilisant un jeu de données constitué d'images de fruits et de labels associés.

Jeu de données

- Base de données d'**images 360** sur Kaggle

(source : <https://www.kaggle.com/datasets/moltean/fruits>)

- Jeu de test comprenant : **22.668 images** de fruits,
un fruit par image.
- **131 classes** : *Apple Braeburn, Banana,
Clementine,...*
- Un dossier par classe, ayant **plusieurs photos du
même fruit sous différents angles**
- Taille des images : **100x100 pixels**
- Sur fond **blanc uniformisé**

| | | | |
|-----------------------|--|--|--|
| Apple Braeburn |  |  |  |
| Banana |  |  |  |
| Clementine |  |  |  |

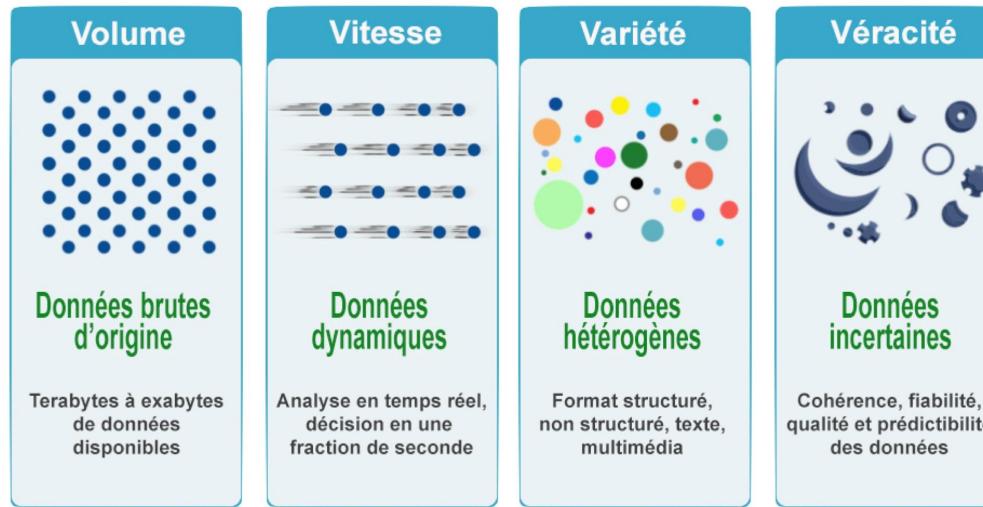


2

Processus de création de
l'environnement *Big Data*

Pourquoi un environnement *Big Data* ?

Big data (ou **données massives**) est composé de jeux de données **variées**, provenant de nombreuses sources, arrivant dans des **volumes** croissants et à une **vitesse** élevée. L'ensemble des données est si volumineux qu'un **logiciel de traitement traditionnel** ne peut pas les gérer.



Pourquoi un environnement *Big Data* ?

Volume

Le **volume** des données générées fait repenser la manière dont les données sont stockées.

Le big data permet de **stocker, gérer et analyser** les données de manière **évolutive**.

Vitesse

Les méthodes de traitement traditionnel sont limitées sur les données qui arrivent aux **vitesses** très élevées.

Le big data offre des outils nécessaires pour **traiter et analyser** les données à **grande vitesse**.

04

01

03

02

Véracité

La **véracité** est un enjeu majeur de l'exploitation du big data.

Il est très difficile de savoir si les données n'ont pas été **usurpées** ou **corrompues** ou si elles proviennent de **sources inattendues**.

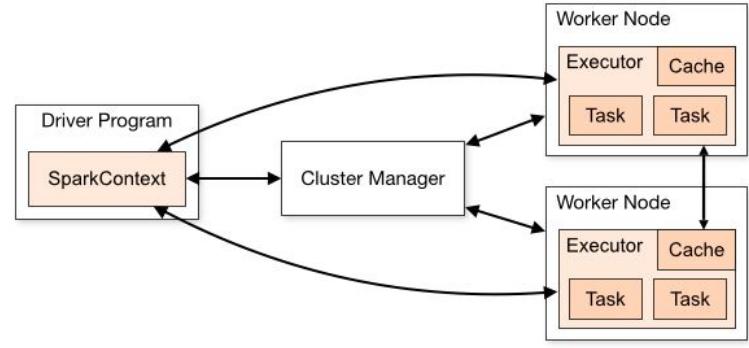
Variété

La **variété** est liée à la diversification des usages d'internet et du numérique.

Le big data permet de **traiter et d'analyser** les données hétérogènes (structurée, semi ou non structurée) issues de multiples sources.

Les outils du *Big Data*

- **Calcul distribué** : distribution du stockage et du traitement des données sur plusieurs unités de calcul réparties en clusters, au profit d'une seule machine afin de diminuer le temps d'exécution.
- **Apache Spark** : framework open-source qui permet de traiter des bases de données massives en utilisant le calcul distribué (in-memory), cet outil permet également de gérer et de coordonner l'exécution de tâches sur des données à travers un groupe d'ordinateurs.
- **Algorithme MapReduce** :
 - ◆ Largement utilisé pour le traitement parallèle et distribué de grandes quantités de données.
 - ◆ Permet de diviser les données en ensembles plus petits, de traiter indépendamment (Map) et de les agréger pour obtenir le résultat final (Reduce).
- Développement de scripts en **PySpark**, librairie python (similaire à Pandas), permettant la communication avec Spark
- **Avantages** :
 - ◆ Évolutivité (ajout de ressources supplémentaires)
 - ◆ Performance (accélération du temps de calcul)
 - ◆ Tolérance aux pannes (plus résilients aux pannes et erreurs)

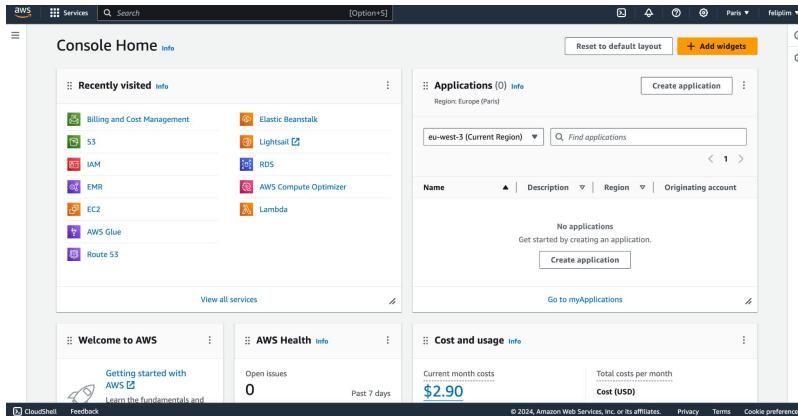


Application Spark

Le **driver** distribue et planifie les tâches entre les différents **exécuteurs** (worker) qui les exécutent et permettent un traitement réparti. Il est le responsable de l'exécution du code sur différentes machines. Le **cluster manager** assure le suivi des ressources disponibles.

Déploiement de la solution dans le cloud

- Louer la puissance de calcul à la demande permet, quel que soit la charge de travail, d'obtenir suffisamment de puissance de calcul pour pouvoir traiter les données, même si le volume de données augmente fortement.
- Le cloud permet de diminuer les coûts, si l'on compare les coûts de location d'un serveur complet sur une durée fixe (1 mois, 1 an).
- Le prestataire le plus connu et qui offre, à ce jour, le plus de solutions cloud est **Amazon Web Service (AWS)**.



Amazon Maintains Cloud Lead as Microsoft Edges Closer

Worldwide market share of leading cloud infrastructure service providers in Q4 2023*



* Includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group



statista

Architecture *Big Data* avec AWS



IAM (Identity and Access Management)



S3 (Simple Storage Service)

Stockage

- Images
- Notebooks
- Résultats
- Bootstrap
- Structure



EMR (Elastic MapReduce)

Cluster de calculs distribués

- Traitement des images

Configuration de l'environnement

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar shows various navigation options like Dashboard, User groups, Roles (which is selected), Policies, Identity providers, etc. The main content area is titled 'My security credentials' for the 'feliplim' user. It displays account details (Account name: feliplim, Email address: lima_felipe@rocketmail.com, AWS account ID: 036294471229, Canonical user ID: d58ec081678dab53513f751d6e46f258d22cfb9e9b34a67a28b580197805a671). Below this is a section for Multi-factor authentication (MFA) with a 'Assign MFA device' button. The 'Access keys' section shows one key (Access key ID: AKIAQQ42RVY6VX5WUQUW, Created on: 17 days ago, Last used: 2 days ago, Region: eu-west-3, Service: s3) with a 'Create access key' button. At the bottom, there's a 'CloudFront key pairs' section with a 'Create CloudFront key pair' button.

→ Service IAM

- ◆ Gestion de droits (contrôle S3)
- ◆ Crédation de clé d'accès qui permet la connection de la machine locale sans devoir saisir systématiquement le login / mot de passe

→ Installation et configuration de AWS CLI (Command Line Interface)

Configuration de l'environnement

The screenshot shows the AWS IAM Roles page. The left sidebar is collapsed, and the main area displays the 'Roles' section. A red box highlights the 'Create role' button at the top right of the table header. Another red box highlights two specific roles: 'EMR_DefaultRole' and 'EMR_EC2_DefaultRole'. Below the table, there's a section titled 'Roles Anywhere' with three options: 'Access AWS from your non AWS workloads', 'X.509 Standard', and 'Temporary credentials'.

IAM > Roles

Roles (7) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Create role

Role name

AWS Service: autoscaling (Service-Link)

AWS Service: compute-optimizer (Service-Link)

AWS Service: elasticmapreduce (Service-Link)

AWS Service: support (Service-Link)

AWS Service: trustedadvisor (Service-Link)

EMR_DefaultRole

AWS Service: elasticmapreduce

EMR_EC2_DefaultRole

AWS Service: ec2

Roles Anywhere

Authenticate your non AWS workloads and securely provide access to AWS services.

Manage

Access AWS from your non AWS workloads

X.509 Standard

Temporary credentials

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

→ Service IAM

- ◆ Gestion de droits (contrôle S3)
- ◆ Crédit de clé d'accès qui permet la connection de la machine locale sans devoir saisir systématiquement le login / mot de passe

→ Installation et configuration de AWS CLI (Command Line Interface)

→ Création des rôles

- ◆ Crédit des identités ayant des permissions spécifiques

Configuration de l'environnement

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (with sub-links: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), and a New link. The main area is titled "Resources" and displays a summary of Amazon EC2 resources in the Europe (Paris) Region. It includes counts for Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs (highlighted with a red box), Load balancers, Placement groups, Security groups, Snapshots, and Volumes.

The screenshot shows the AWS EC2 Key pairs page. The sidebar is identical to the first dashboard. The main area shows a table for "Key pairs (1)" with columns: Name, Type, Created, Fingerprint, and ID. A single row is listed: "fruit-classification" (rsa, 2024/01/3..., a1:08:49:5..., key-027eb...). Below the table is a search bar and a "Create key pair" button.

- **Service EC2 (Elastic Compute Cloud)**
- ◆ Création et gestion d'instances (machines)
 - ◆ Création de paire de clé EC2 pour connecter en SSH aux instances sans devoir entrer login / mot de passe

The screenshot shows the "Create key pair" dialog box. It has fields for "Name" (containing "fruit-classification"), "Key pair type" (set to RSA), and "Private key file format" (set to .pem). There are also sections for "Tags - optional" and "Add new tag". At the bottom right is a large orange "Create key pair" button.

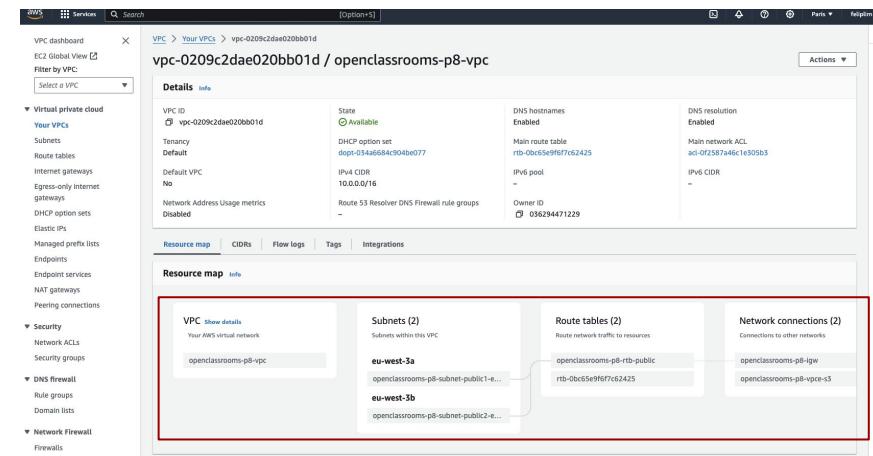
Configuration de l'environnement

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar for 'Virtual private cloud' with options like 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', and 'Egress-only internet gateways'. The main area has sections for 'Resources by Region' (VPCs, Subnets, NAT Gateways, VPC Peering Connections) and 'Settings' (Zones, Console Experiments). A prominent orange button at the top says 'Create VPC'. Below it, a note says 'Note: Your Instances will launch in the Europe region.' The 'Additional Information' section includes links to 'VPC Documentation' and 'All VPC Resources'.

This screenshot shows the 'Create VPC' wizard. It's on the 'Preview' step. The 'VPC settings' section shows 'Name tag auto-generation' is checked, and the name 'openclassrooms-p8-vpc' is highlighted in red. Other fields include 'CIDR block' (10.0.0.0/16), 'Number of Availability Zones (AZs)' (3), and 'Tenancy' (Default). The 'Resource to create' section has 'VPC only' selected. The 'Preview' section shows the VPC structure with subnets in 'eu-west-2' and 'eu-west-3' regions.

→ Service VPC (Virtual Private Cloud)

- ◆ Création d'une partie isolée du cloud AWS pour y créer les instances.



Stockage des données sur S3

→ Service S3

- ◆ Stockage d'une grande variété d'objets (images, fichiers)
- ◆ Évolutivité avec espace disponible en illimité
- ◆ Indépendant des serveurs EC2
- ◆ Accès aux données très rapide
- ◆ Possibilité de définir des politiques d'accès IAM

→ Mise en oeuvre :

- ◆ Création d'un compartiment (bucket)
- ◆ Choisir la même région (i.e. eu-west-3)
- ◆ Chargement des données sur le bucket :
 - Fichier de configuration avec bootstrap (EN)
 - Fichier avec configuration du software (EMR)
 - Dossier avec les données
 - Test
 - Résultats
 - Dossier avec les logs

```
felipeleima@MacBook-Air-de-Felipe fruits-360_dataset % cd fruits-360
felipeleima@MacBook-Air-de-Felipe fruits-360 % ls
LICENSE           Test          Training      papers      readme.md    test-multiple_fruits
felipeleima@MacBook-Air-de-Felipe Test % ls
Apple Braeburn   Cantaloupe 1  Grape Blue   Mangostan  Pear Monster Potato White
Apple Crimson Snow  Cantaloupe 2  Grape Pink   Maracuja  Pear Red Quince
Apple Golden 1    Carambula   Grape White  Melon Piel de Sapo Pear Stone Rambutan
Apple Golden 2    Cauliflower  Grape White 2 Mulberry  Pear Williams Raspberry
Apple Golden 3    Cherry 1     Grape White 3 Nectarine Pepino Redcurrant
Apple Granny Smith  Cherry 2     Grape White 4 Nectarine Flat Pepper Green Salak
Apple Pink Lady   Cherry Rainier  Grapefruit Pink Nut Forest Pepper Orange Strawberry
Apple Red 1       Cherry Wax Black  Grapefruit White Pepper Red Strawberry Wedge
Apple Red 2       Cherry Wax Red   Guava   Onion Red Tamarillo
Apple Red 3       Cherry Wax Yellow Hazelnut Onion Red Peeled Physalis Tangelo
Apple Red Delicious  Chestnut   Huckleberry Onion White Physalis with Husk Tomato 1
Apple Red Yellow 1 Clementine  Kaki   Orange   Pineapple Tomato 2
Apple Red Yellow 2 Cocos     Kiwi   Papaya  Pineapple Mini Tomato 3
Apricot          Corn       Kohlrabi  Passion Fruit Pitahaya Red Tomato 4
Avocado          Corn Husk   Kumquats Peach   Plum Tomato Cherry Red
Avocado ripe     Cucumber Ripe Lemon   Peach 2   Plum 2 Tomato Heart
Banana           Cucumber Ripe 2 Lemon Meyer Peach Flat Plum 3 Tomato Maroon
Banana Lady Finger Dates   Limes   Pear   Pomegranate Tomato Yellow
Banana Red       Eggplant   Lychee   Pear 2   Pomegranate Tomato not Ripened
Beetroot          Fig       Mandarine  Pear Abate Potato Red Walnut
Blueberry         Ginger Root Mango   Pear Forelle Potato Red Washed Watermelion
Cactus Fruit     Granadilla  Mango Red  Pear Kaiser Potato Sweet
felipeleima@MacBook-Air-de-Felipe Test % clear
```

```
felipeleima@MacBook-Air-de-Felipe Test % aws sync . s3://openclassrooms-p8-fruits-data/data/test/
upload: ./DS_Store to s3://openclassrooms-p8-fruits-data/data/test/.DS_Store
upload: Apple Braeburn/323_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/323_100.jpg
upload: Apple Braeburn/322_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/322_100.jpg
upload: Apple Braeburn/326_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/326_100.jpg
upload: Apple Braeburn/324_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/324_100.jpg
upload: Apple Braeburn/321_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/321_100.jpg
upload: Apple Braeburn/325_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/325_100.jpg
upload: Apple Braeburn/328_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/328_100.jpg
upload: Apple Braeburn/33_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/33_100.jpg
upload: Apple Braeburn/32_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/32_100.jpg
upload: Apple Braeburn/34_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/34_100.jpg
upload: Apple Braeburn/34_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/34_100.jpg
upload: Apple Braeburn/3_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/3_100.jpg
upload: Apple Braeburn/35_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/35_100.jpg
upload: Apple Braeburn/39_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/39_100.jpg
upload: Apple Braeburn/41_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/41_100.jpg
upload: Apple Braeburn/48_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/48_100.jpg
upload: Apple Braeburn/37_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/37_100.jpg
upload: Apple Braeburn/38_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/38_100.jpg
upload: Apple Braeburn/42_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/42_100.jpg
upload: Apple Braeburn/43_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/43_100.jpg
upload: Apple Braeburn/44_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/44_100.jpg
upload: Apple Braeburn/45_100.jpg to s3://openclassrooms-p8-fruits-data/data/test/Apple Braeburn/45_100.jpg
```



Cluster de calcul distribué avec EMR

- EMR est une plateforme qui permet l'exécution de **traitements de données distribuées à grande échelle**, en utilisant de frameworks tels que **Hadoop et Spark**.
- EMR utilise des instances **EC2**, avec des applications préinstallées et configurées pour créer et gérer le cluster de calcul distribué.
- Le service est entièrement géré par AWS, ce qui garantit :
 - ◆ Évolutilité
 - ◆ Flexibilité
 - ◆ Gestion simplifiée

Étapes de création d'un cluster :

- 1) Configuration logiciel
- 2) Configuration matériel
- 3) Actions d'amorçage
- 4) Options de sécurité
- 5) Choix de network
- 6) Choix de log

EMR - 1) Configuration logiciel

- Choix de logiciels :
 - ◆ Hadoop et Spark : calculs distribués
 - ◆ Tensorflow : import du modèle et transfert learning
 - ◆ JupyterHub : exécution des notebooks PySpark
- Paramétrage de la persistance des notebooks créés et ouverts via JupyterHub (configuration au format JSON sur S3)

▼ Software settings - optional [Info](#)

Enter configuration Load JSON from Amazon S3

Amazon S3 location View

▼ Software settings - optional [Info](#)

Enter configuration Load JSON from Amazon S3

```
1 [{"2": [{"3": "classification": "jupyter-s3-conf", "4": "properties": {"5": "s3.persistence.bucket": "openclassrooms-p8-fruits-data", "6": "s3.persistence.enabled": "true"}, "7": }}, {"8": }, {"9": }]
```

Amazon EMR > [EMR on EC2: Clusters](#) > Create cluster

Create cluster [Info](#)

Name and applications [Info](#)

Name: EMR-fruit-cluster

Amazon EMR release: [Info](#)
A release contains a set of applications which can be installed on your cluster.
emr-6.7.0

Application bundle:

| | | | | | |
|-------|-------------|-------|--------|-------|--------|
| Spark | Core Hadoop | HBase | Presto | Trino | Custom |
| | | | | | |

Flink 1.14.2 Ganglia 2.7.0 HBase 2.4.4
 HCatalog 3.1.3 Hadoop 3.2.1 Hive 3.1.3
 Hue 4.10.0 JupyterEnterpriseGateway 2.1.0 Oozie 5.2.1
 Livy 0.7.1 MXNet 1.8.0 Pig 0.17.0 Presto 0.272
 Phoenix 5.1.2 Phoenix 5.1.2 Sqoop 1.4.7 TensorFlow 2.4.1
 Spark 3.2.1 Spark 3.2.1 ZooKeeper 3.5.7 Trino 378 Zeppelin 0.10.0

AWS Glue Data Catalog settings
Use the AWS Glue Data Catalog to provide an external metastore for your application.
 Use for Spark table metadata

Operating system options [Info](#)

Amazon Linux release Custom Amazon Machine Image (AMI)
 Automatically apply latest Amazon Linux updates

Summary [Info](#)

Name: EMR-fruit-cluster

Amazon EMR release: emr-6.7.0

Application bundle: Custom (Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1)

Cluster configuration

Uniform instance groups
Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)

Cluster scaling and provisioning

Provisioning configuration
Core size: 1 instance
Task size: 1 instance

Configure IAM roles
You must choose a service role and instance profile before you create this cluster.

EMR - 2) Configuration matériel

→ Choix des instances

- ◆ 1 instance nœud **primaire** (driver)
- ◆ 2 instances nœuds **principaux** (workers)
- ◆ Instances de **type M5** (instances équilibrés) et **xlarge** (la moins onéreuse)



vCPU : 4

Mémoire (GiO) : 16

Bandé passante réseau (Gbit/s) : jusqu'à 10

Bandé passante EBS (Mbit/s) : Jusqu'à 4 750

Coût (instance/heure): 0,224\$

Cluster configuration Info
Choose a configuration method for the primary, core, and task node groups for your cluster.

Uniform instance groups
Choose the same EC2 instance type and purchasing option (On-Demand or Spot) for all nodes in your node group. [Learn more](#)

Flexible instance fleets
Choose from the widest variety of provisioning options for the EC2 instances in your cluster. Diversify instance types and purchasing options, and use an allocation strategy. [Learn more](#)

Uniform instance groups

Primary
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: \$0.224 per instance/hour
Lowest Spot price: \$0.068 (eu-west-3b)

Use high availability
Launch highly available, more resilient cluster with three primary nodes on On-Demand Instances. This configuration applies for the lifetime of your cluster. [Learn more](#)

► Node configuration - optional

Core
Choose EC2 instance type
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: \$0.224 per instance/hour
Lowest Spot price: \$0.068 (eu-west-3b)

Remove instance group

► Node configuration - optional

Cluster scaling and provisioning Info
Set up scaling and provisioning configurations for the core and task node groups for your cluster.

Choose an option

Set cluster size manually
Use this option if you know your workload patterns in advance.

Use EMR-managed scaling
Monitor key workload metrics so that EMR can optimize the cluster size and resource utilization.

Use custom automatic scaling
To programmatically scale core and task nodes, create custom automatic scaling policies.

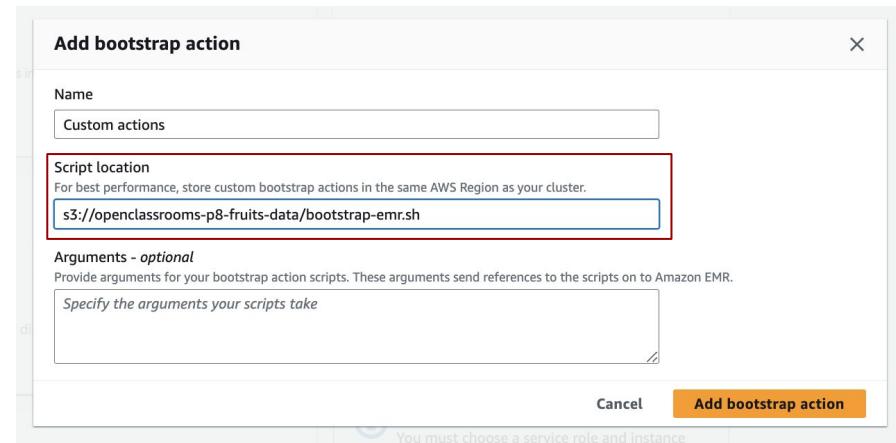
Provisioning configuration
Set the size of your core instance group. Amazon EMR attempts to provision this capacity when you launch your cluster.

| Name | Instance type | Instance(s) size | Use Spot purchasing option |
|------|---------------|------------------|----------------------------|
| Core | m5.xlarge | 2 | <input type="checkbox"/> |

EMR - 3) Actions d'amorçage

- Choix de **packages manquants à installer**, utiles pour l'exécution du notebook.
- **À l'initialisation du serveur**, les packages seront installés sur l'ensemble des machines du cluster (pas seulement sur le driver).
- Création du fichier **bootstrap.sh** contenant des commandes **pip install** (chargement du fichier sur S3).
- Ajout du script dans les **actions d'amorçage**.

```
$ bootstrap-emr.sh
$ bootstrap-emr.sh
1  #! /bin/bash
2  sudo python3 -m pip install -U setuptools
3  sudo python3 -m pip install -U pip
4  sudo python3 -m pip install wheel
5  sudo python3 -m pip install pillow
6  sudo python3 -m pip install pandas==1.2.5
7  sudo python3 -m pip uninstall numpy
8  sudo python3 -m pip install numpy==1.23
9  sudo python3 -m pip install pyarrow
10 sudo python3 -m pip install boto3
11 sudo python3 -m pip install s3fs
12 sudo python3 -m pip install fsspec
13 sudo python3 -m pip uninstall tensorflow
14 sudo python3 -m pip install tensorflow==2.6.0
```



| ▼ Bootstrap actions - optional (1) <small>Info</small> | | | |
|---|---|-----------|--|
| Use bootstrap actions to install software or customize your instance configuration. | | | |
| Name | Amazon S3 location | Arguments | |
| Custom actions | s3://openclassrooms-p8-fruits-data/bootstrap-emr.sh | - | <small>Remove</small> <small>Edit</small> <small>Add</small> |

EMR - 4) Sécurité

- Choix de la **paire de clés EC2** créée précédemment, ce qui permet de connecter aux instances EC2 en SSH sans devoir saisir login / mot de passe.
- Choix des **rôles** pour **EMR** et pour les instances **EC2** :
 - ◆ Service role : permet à **EMR** d'appeler des services AWS tels que EC2.
 - ◆ Instance profile: permet aux instances **EC2** d'appeler des services AWS tels que S3.

Security configuration and EC2 key pair - optional [Info](#)

Security configuration
Select your cluster encryption, authentication, authorization, and instance metadata service settings.

Amazon EC2 key pair for SSH to the cluster [Info](#)

EMR_DefaultRole [Info](#)
Allows Elastic MapReduce to call AWS services such as EC2 on your behalf.

EMR_EC2_DefaultRole [Info](#)
Allows EC2 instances in an Elastic MapReduce cluster to call AWS services such as S3 on your behalf.

Summary

| | | |
|--|---|--|
| Creation date January 31, 2024, 14:21 (UTC+01:00) | ARN arn:aws:iam::036294471229:role/EMR_EC2_DefaultRole | Instance profile ARN arn:aws:iam::036294471229:instance-profile/EMR_EC2_DefaultRole |
| Last activity 2 days ago | Maximum session duration 1 hour | |

Permissions **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

Permissions policies (1) [Info](#)
You can attach up to 10 managed policies.

| Policy name | Type | Attached entities |
|--|-------------|-------------------|
| AmazonElasticMapReduceforEC2Role | AWS managed | 1 |

EC2 instance profile for Amazon EMR
The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

Choose an existing instance profile
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

Instance profile

Custom automatic scaling role - optional
When a custom automatic scaling rule triggers, Amazon EMR assumes this role to add and terminate EC2 instances. [Learn more](#)

Custom automatic scaling role

EMR - 5) Choix de network

Networking [Info](#)

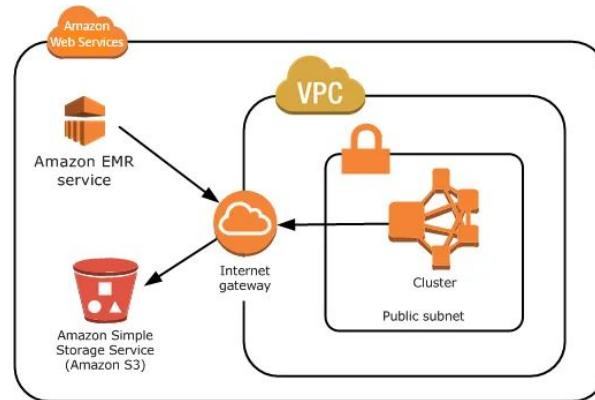
Virtual private cloud (VPC) [Info](#)
vpc-0209c2dae020bb01d [Browse](#) [Create VPC](#)

Subnet [Info](#)
subnet-051f9ef194f0dadb3 [Browse](#) [Create subnet](#)

▶ EC2 security groups (firewall)

→ Attribution d'un **VPC** au cluster afin d'avoir :

- ◆ Plus de flexibilité
- ◆ Plus de contrôle de la sécurité
- ◆ Meilleur routage de trafic
- ◆ Disponibilité



EMR - 6) Choix de log

▼ Cluster logs - optional [Info](#)

i We automatically archive your log files to Amazon S3. You can specify your own S3 location, or use the default S3 location for Amazon EMR. The default log location is pre-populated in the **Amazon S3 location** field.

Publish cluster-specific logs to Amazon S3

Amazon S3 location

X View Browse S3

Format: Use s3://bucket/prefix

Encrypt cluster-specific logs



Attribution d'un dossier sur **S3** pour stocker les logs

Lancement du cluster EMR

AWS Services Search [Option+S]

Amazon EMR > EMR on EC2: Clusters > Create cluster

Create cluster Info

Name and applications Info

Name: EMR-fruit-cluster

Amazon EMR release: Info Info
A release contains a set of applications which can be installed on your cluster.

emr-6.7.0

Application bundle:

| | | | | | |
|-------|------|-------|--------|-------|--------|
| Spark | Core | HBase | Presto | Trino | Custom |
| | | | | | |

Flink 1.14.2 Ganglia 3.7.2 Hadoop 3.2.1 HBase 2.4.4 Hive 3.1.3 Hue 4.10.0 JupyterHub 1.4.1 MXNet 1.8.0 Presto 0.272 TensorFlow 2.4.1 Tez 0.9.2 Zeppelin 0.10.0

Flink 1.14.2 Ganglia 3.7.2 Hadoop 3.2.1 HBase 2.4.4 Hive 3.1.3 Hue 4.10.0 JupyterHub 1.4.1 MXNet 1.8.0 Presto 0.272 TensorFlow 2.4.1 Tez 0.9.2 Zeppelin 0.10.0

Flink 1.14.2 Ganglia 3.7.2 Hadoop 3.2.1 HBase 2.4.4 Hive 3.1.3 Hue 4.10.0 JupyterHub 1.4.1 MXNet 1.8.0 Presto 0.272 TensorFlow 2.4.1 Tez 0.9.2 Zeppelin 0.10.0

Summary Info

Amazon S3 location: s3://openclassrooms...

Software settings - optional

Configuration: s3://openclassrooms...

Security configuration and EC2 key pair - optional

Amazon EC2 key pair: fruit-classif...

Identity and Access Management (IAM) roles

Service role: EMR_DefaultRole

Instance profile: EMR_EC2_DefaultRole

AWS Glue Data Catalog settings

Use the AWS Glue Data Catalog to provide an external metastore for your application.

Use for Spark table metadata

Operating system options Info

Amazon Linux release: Info Info
Custom Amazon Machine Image (AMI)

Automatically apply latest Amazon Linux updates

Cluster configuration Info

Choose a configuration method for the primary, core, and task node groups for your cluster.

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EMR > EMR on EC2: Clusters > EMR-fruit-cluster

EMR-fruit-cluster

Updated 7 minutes ago

Summary

| Cluster info | Applications | Cluster management | Status and time |
|--|---|--|--|
| Cluster ID: j-17KL8PPV3A703 | Amazon EMR version: emr-6.7.0 | Log destination in Amazon S3: openclassrooms-p8-fruits-data/logs | Status: Waiting |
| Cluster configuration: Instance groups | Installed applications: Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1 | Persistent application UIs: Spark History Server, YARN timeline server | Creation time: February 14, 2024, 11:39 (UTC+01:00) |
| Capacity: 1 Primary, 2 Core, 0 Task | Primary node public DNS: ec2-13-36-237-163.eu-west-3.compute.amazonaws.com | Elapsed time: 14 minutes | Connect to the Primary node using SSH: Connect to the Primary node using SSM |

Lancement du cluster EMR

Amazon EMR > EMR on EC2: Clusters > EMR-fruit-cluster

EMR-fruit-cluster

Updated 7 minutes ago [C](#) [Terminate](#) [Clone in AWS CLI](#) [Clone](#)

▼ Summary

| Cluster info | Applications | Cluster management | Status and time |
|-------------------------------|--|--|--|
| Cluster ID j-17KL8PPV3A7O3 | Amazon EMR version emr-6.7.0 | Log destination in Amazon S3 openclassrooms-p8-fruits-data/logs | Status Waiting |
| Cluster configuration | Installed applications | Persistent application UIs | Creation time |
| Instance groups | Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1 | Spark History Server YARN timeline server | February 14, 2024, 11:39 (UTC+01:00) |
| Capacity | | Primary node public DNS | Elapsed time |
| 1 Primary 2 Core 0 Task | | ec2-13-36-237-163.eu-west-3.compute.amazonaws.com | 14 minutes |
| | | Connect to the Primary node using SSH | |
| | | Connect to the Primary node using SSM | |

Création du tunnel SSH à l'instance EC2

Objectif : accéder aux **applications** en créant un tunnel SSH vers le **driver**

- Modification du **groupe de sécurité EC2** du driver
 - ◆ Autorisation sur les connexions entrantes du driver : **ouverture du port 22** (port d'écoute du serveur SSH)

The screenshot shows the AWS EC2 Security Groups page. A new security group named "ElasticMapReduce-master" has been created and selected. The "Inbound rules" section is highlighted, showing a rule allowing SSH traffic (TCP port 22) from the IP address 0.0.0.0/0.

| Name | Security group rule ID | Type | Protocol | Port range |
|------------------------------|------------------------|-----------------|------------|------------|
| sgr-072a6954a18cf683f | - | All ICMP - IPv4 | ICMP | All |
| sgr-061921f49365d3f5 | - | All UDP | UDP | 0 - 65535 |
| sgr-0fe61ffd700bdcc3e9 | - | All TCP | TCP | 0 - 65535 |
| sgr-08a0868fdc9602ccb | IPv4 | SSH | TCP | 22 |
| sgr-0493963077edeb... | - | All TCP | TCP | 0 - 65535 |
| sgr-0d7781741d899b... | - | All ICMP - IPv4 | ICMP | All |
| sgr-0b4bf210c9a07c30f | IPv6 | SSH | TCP | 22 |
| sgr-032cb7e0d809755bf | - | Custom TCP | TCP | 8443 |

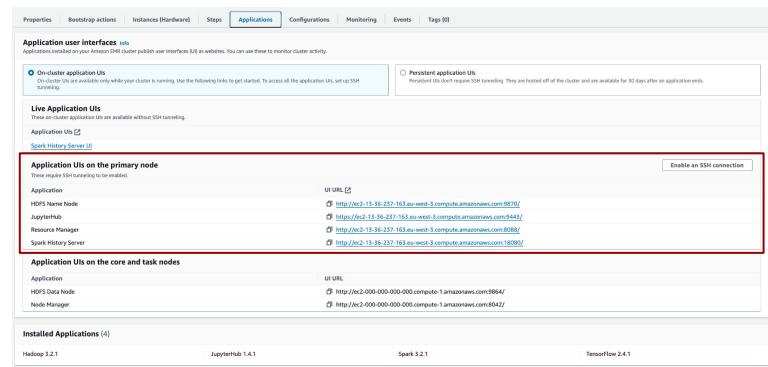
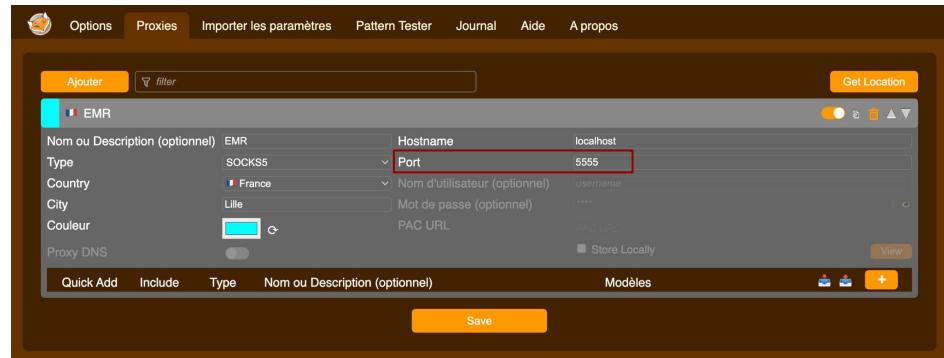
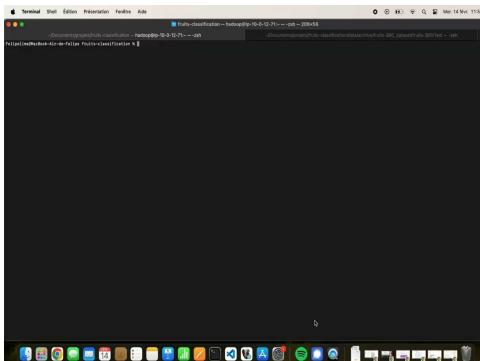
The screenshot shows the AWS EC2 Instances page. The "ElasticMapReduce-master" instance is selected, indicated by a red border around its row in the table.

| Name | Security group ID | Security group name |
|----------|-----------------------------|--------------------------------|
| - | sg-0b744c2d09434efd6 | default |
| - | sg-03e9c5e72e1278a80 | ElasticMapReduce-master |
| - | sg-08b31396439981227 | ElasticMapReduce-slave |

Création du tunnel SSH à l'instance EC2

Objectif : accéder aux **applications** en créant un tunnel SSH vers le **driver**

- Modification du **groupe de sécurité EC2** du driver
 - ◆ Autorisation sur les connexions entrantes du driver : **ouverture du port 22** (port d'écoute du serveur SSH)
 - Établissement du **tunnel SSH**
 - Configuration de **FoxyProxy** pour que le **navigateur** emprunte le tunnel SSH
 - Accès aux **applications** du serveur EMR via le **navigateur**



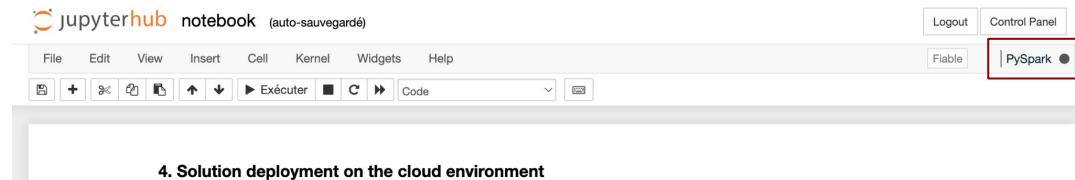


3

Chaîne de traitement d'images
dans un environnement *Big Data*
dans le cloud

Chaîne de traitement d'images

- Exécution du notebook depuis **JupyterHub**, hébergé sur le serveur EMR.
- Utilisation d'un **kernel PySpark**.
- Lancement de la session **Spark** à l'exécution de la première ligne.



4. Solution deployment on the cloud environment



- | | | | |
|--|--|--|---|
| → Images stockées dans un compartiment S3. | → Utilisation de la librairie PIL. | → Modèle MobileNetV2, pré-entraîné sur la base ImageNet. | → Écriture des résultats dans des fichiers Parquet. |
| → Chargement des images dans un DataFrame Spark. | → Redimensionnement des images (100, 100, 3) vers (224, 224, 3). | → Couche de sortie : avant dernière couche (extraction de features). | → Stockage dans le compartiment S3. |
| | → Fonction de processing adapté au modèle. | → Extraction de features par batch à l'aide de Pandas UDF. | |

Chargement de données

- Chargement des données avec **spark.read()** :
 - ◆ Traitement des fichiers en tant que **données binaires**.
 - ◆ À l'emplacement spécifié (le compartiment S3), recherche récursive dans les sous-répertoires des fichiers avec l'extension **.jpeg**.
 - ◆ Chargement des images dans un **DataFrame Spark**.

```
root
|-- path: string (nullable = true)
|-- modificationTime: timestamp (nullable = true)
|-- length: long (nullable = true)
|-- content: binary (nullable = true)
|-- label: string (nullable = true)
```

Schéma du DataFrame Spark

- Ajout de la colonne **label** issu du chemin d'accès du fichier :
 - ◆ **Label** représente la catégorie de l'image (nom du fruit), avant dernier élément (-2) du path.

| path | modificationTime | length | content |
|----------------------|---------------------|--------|-----------------------|
| s3://openclassroo... | 2024-02-14 09:41:46 | 7353 | [FF D8 FF E0 00 1...] |
| s3://openclassroo... | 2024-02-14 09:41:46 | 7350 | [FF D8 FF E0 00 1...] |
| s3://openclassroo... | 2024-02-14 09:41:46 | 7349 | [FF D8 FF E0 00 1...] |
| s3://openclassroo... | 2024-02-14 09:41:46 | 7348 | [FF D8 FF E0 00 1...] |
| s3://openclassroo... | 2024-02-14 09:41:47 | 7328 | [FF D8 FF E0 00 1...] |

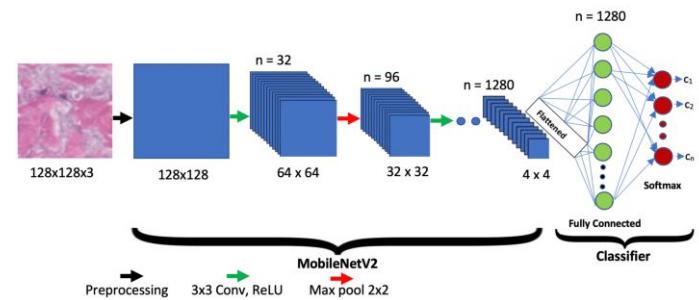
only showing top 5 rows

| path | label |
|---|------------|
| s3://openclassrooms-p8-fruits-data/data/test/Watermelon/r_106_100.jpg | Watermelon |
| s3://openclassrooms-p8-fruits-data/data/test/Watermelon/r_109_100.jpg | Watermelon |
| s3://openclassrooms-p8-fruits-data/data/test/Watermelon/r_108_100.jpg | Watermelon |
| s3://openclassrooms-p8-fruits-data/data/test/Watermelon/r_107_100.jpg | Watermelon |
| s3://openclassrooms-p8-fruits-data/data/test/Watermelon/r_95_100.jpg | Watermelon |

only showing top 5 rows

MobileNetV2 avec le Transfer Learning

- Choix du modèle **MobileNetV2** :
 - ◆ Modèle de réseau de neurones convolutifs (CNN), pré-entraîné sur la base ImageNet pour la détection de features et la classification d'images.
 - ◆ Spécialement conçu pour **appareils mobiles avec ressources limitées** :
 - Rapidité d'exécution (adapté pour le traitement d'un gros volume de données).
 - Faible dimensionnement du vecteur de sortie (1, 1, 1280).
- **Transfer Learning** :
 - ◆ Consiste à utiliser la connaissance déjà acquise par un modèle entraîné en l'adaptant à la problématique.
 - ◆ Crédit d'une instance du modèle pré-entraîné avec les poids du jeu de données ImageNet, incluant la couche de classification finale.
- **Préparation du modèle** :
 - ◆ Crédit d'un nouveau modèle ayant pour couche de sortie l'avant-dernière couche (extraction de features) du modèle pré-entraîné.
 - ◆ Dimension du vecteur de sortie (1, 1, 1280).
 - ◆ **Diffusion des poids** avec `sparkContext.broadcast()` de PySpark:
 - Chargement du modèle sur le driver puis diffusion des poids aux workers.
 - Distribution d'une variable à travers le cluster pour qu'elle soit disponible pour tous les nœuds de calcul.



Pré-processing

- Dimension **d'origine** des images : **(100, 100, 3)** = (100 * 100 pixels + 3 canaux de couleur RVB).
- Dimension **attendue** des images **en entrée du modèle MobileNetV2** : **(224, 224, 3)**.
 - ◆ Il est nécessaire de redimensionner les images avant de les envoyer au modèle.
- Utilisation de la librairie **PIL (Python Imaging Library)** :
 - ◆ Ouverture des données binaires de l'image en tant qu'image.
 - ◆ Redimensionnement de l'image à une taille (224, 224, 3).
- Application de la fonction **preprocess_input** de Tensorflow, une **fonction de prétraitement spécifique** pour prétraiter les images avant de les passer en entrée du modèle MobileNetV2.



Traitement de données et stockage

→ Extraction des features :

- ◆ À partir des images pré-traitées, répartition de données et application itérative du modèle aux batches de données (images) pour en extraire les features, en utilisant Pandas UDF.
- ◆ Résultat : un DataFrame avec colonnes d'origine + features
- ◆ Les données sont traitées en parallèle sur différents nœuds, ce qui permet d'employer la puissance du calcul distribué.
- ◆ Le modèle est chargé une seule fois et réutilisé pour tous les batches de données, ce qui évite les coûts de recharge et réduit la consommation de mémoire.

→ Réduction de dimension avec PCA :

- ◆ Application de l'analyse en composantes principales pour réduire la dimensionnalité tout en préservant un maximum d'informations.

→ Stockage des résultats :

- ◆ Données du DataFrame écrites dans un fichier Parquet (format de stockage optimisé pour le *Big Data*).
- ◆ Mode *overwrite* : si le fichier existe déjà, il sera écrasé.
- ◆ Sauvegarde dans le dossier **Data/Results** du compartiment S3.



4

Exécution du script PySpark sur le
cloud

Démonstration d'exécution dans le cloud

Properties > EMR-fruit-cluster

eu-west-3.console.aws.amazon.com/emr/home?region=eu-west-3#clusterDetails/j-17KL8PPV3A703

Terminer la mise à jour

aws Services Search Option+S

Your cluster "EMR-fruit-cluster" has been successfully created.

Amazon EMR > EMR on EC2 Clusters > EMR-fruit-cluster

EMR-fruit-cluster

Summary

| Cluster info | Applications | Cluster management | Status and time |
|-------------------------------|--|--|---|
| Cluster ID j-17KL8PPV3A703 | Amazon EMR version emi-6.7.0 | Log destination in Amazon S3 openclassrooms-p8-fruits-data/logs | Status Waiting |
| Cluster configuration | Installed applications | Persistent application URLs Spark History Server | Creation date February 14, 2024, 11:39 (UTC+01:00) |
| Instance groups | Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1 | YARN timeline server | Elapsed time 19 minutes, 23 seconds |
| Capacity | 1 Primary node public DNS ec2-13-36-237-163.eu-west-3.compute.amazonaws.com | Primary node public DNS | |
| | Connect to the Primary node using SSH | Connect to the Primary node using SSH | |

Properties Bootstrap actions Instances (Hardware) Steps Applications Configurations Monitoring Events Tags (0)

Operating system Info Cluster logs Info Cluster termination Info Edit cluster termination

| Amazon Linux release 2.0.20240131.0 | Archive log files to Amazon S3 Turned on | Encryption for logs Turned off | Termination option Automatically terminate cluster after idle time | Termination protection Turned on |
|--|---|-----------------------------------|---|-------------------------------------|
| | | | idle time 2 hours | |

Network and security Info

| Network | Security configuration | Permissions |
|--|--------------------------------|--|
| Virtual Private Cloud (VPC) vpc-02092c2ea020b601d | Security configuration None | Service role for Amazon EMR EMR_DefaultRole |

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences

Spark 3.2.1-amzn-0 Jobs Stages Storage Environment Executors SQL

ivy-session-0 application UI

Spark Jobs

User: ivy
Duration: 20 min
Scheduling Mode: FIFO
Completed Jobs: 20

Completed Jobs (20)

Page: 1 of 1 Pages. Jump to 1 . Show 1000 items in a page. Go

| Job Id (Job Group) | Description | Submitted | Duration | Stages: Successed/Total | Tasks (for all stages): Successed/Total |
|--------------------|--|---------------------|----------|-------------------------|---|
| 0 (4) | Listing leaf files and directories for 131 paths: s3://openclassrooms-p8-fruits-data/test/Apple Braeburn, ... load at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:02:19 | 7 s | 1/1 | 131/131 |
| 1 (5) | Job group for statement 5 showString at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:02:30 | 1 s | 1/1 | 1/1 |
| 2 (6) | Job group for statement 6 showString at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:02:32 | 0.1 s | 1/1 | 1/1 |
| 3 (16) | Job group for statement 16 print at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:02:38 | 0.1 min | 1/1 | 709/709 |
| 4 (16) | Job group for statement 16 parquet at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:12:38 | 5.7 min | 1/1 (1 skipped) | 24/24 (709 skipped) |
| 5 (18) | Job group for statement 18 first at PCA.scala:44 | 2024/02/14 11:18:22 | 10 min | 2/2 | 710/710 |
| 6 (18) | Job group for statement 18 first at RowMatrix.scala:62 | 2024/02/14 11:28:27 | 18 s | 1/1 (1 skipped) | 1/1 (709 skipped) |
| 6 (18) | Job group for statement 18 first at RowMatrix.scala:62 | 2024/02/14 11:28:27 | 18 s | 1/1 (1 skipped) | 1/1 (709 skipped) |
| 7 (18) | Job group for statement 18 treeAggregate at Statistics.scala:58 | 2024/02/14 11:28:45 | 5.8 min | 2/2 (1 skipped) | 28/28 (709 skipped) |
| 8 (18) | Job group for statement 18 isEmpty at RowMatrix.scala:426 | 2024/02/14 11:34:30 | 17 s | 1/1 (1 skipped) | 1/1 (709 skipped) |
| 9 (18) | Job group for statement 18 treeAggregate at RowMatrix.scala:166 | 2024/02/14 11:34:47 | 5.8 min | 2/2 (1 skipped) | 28/28 (709 skipped) |
| 10 (19) | Job group for statement 19 parquet at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:40:43 | 9.7 min | 1/1 | 709/709 |
| 11 (19) | Job group for statement 19 parquet at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:50:24 | 5.8 min | 1/1 (1 skipped) | 24/24 (709 skipped) |
| 12 (21) | Job group for statement 21 parquet at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:17 | 0.3 s | 1/1 | 1/1 |
| 13 (22) | Job group for statement 22 head at cassandra-1 | 2024/02/14 11:56:17 | 0.6 s | 1/1 | 1/1 |
| 14 (24) | Job group for statement 24 count at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:18 | 0.7 s | 1/1 | 3/3 |
| 15 (24) | Job group for statement 24 count at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:19 | 30 ms | 1/1 (1 skipped) | 1/1 (3 skipped) |
| 16 (26) | Job group for statement 26 parquet at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:20 | 0.1 s | 1/1 | 1/1 |
| 17 (27) | Job group for statement 27 head at cassandra-1 | 2024/02/14 11:56:20 | 0.2 s | 1/1 | 1/1 |
| 18 (30) | Job group for statement 30 count at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:21 | 0.4 s | 1/1 | 3/3 |
| 19 (30) | Job group for statement 30 count at NativeMethodAccessorsImpl.java:0 | 2024/02/14 11:56:22 | 15 ms | 1/1 (1 skipped) | 1/1 (3 skipped) |

Terminer le cluster EMR

Properties > EMR-fruit-cluster

eu-west-3.console.aws.amazon.com/emr/home?region=eu-west-3#clusterDetails/j-17KL8PPV3A703

Your cluster "EMR-fruit-cluster" has been successfully created.

Amazon EMR > EMR on EC2: Clusters > EMR-fruit-cluster

EMR-fruit-cluster

Summary

| Cluster info | Applications | Cluster management | Status and time |
|-------------------------------|---|--|---|
| Cluster ID j-17KL8PPV3A703 | Amazon EMR version emr-6.7.0 | Log destination in Amazon S3 openclassrooms-p8-fruits-data/logs | Status Waiting |
| Cluster configuration | Installed applications | Persistent application UIs Spark History Server | Creation time February 14, 2024, 11:39 (UTC+01:00) |
| Instance groups | Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1 | YARN timeline server | Elapsed time 1 hour, 30 minutes |
| Capacity | Primary node 2 Core 0 Task | Primary node public DNS ec2-13-36-237-163.eu-west-3.compute.amazonaws.com | |
| | | Connect to the Primary node using SSH | |
| | | Connect to the Primary node using SSH | |

Properties Bootstrap actions Instances (Hardware) Steps Applications Configurations Monitoring Events Tags (0)

Operating system Info

Amazon Linux release 2.0.20240131.0

Cluster logs Info

Archive log files to Amazon S3
Turned on

Encryption for logs
Turned off

Cluster termination Info

Edit cluster termination

Termination option
Automatically terminate cluster after idle time

Termination protection
Turned on

idle time
2 hours

Network and security Info

Network

Virtual Private Cloud (VPC)
vpc-02092c2e0020b601d

Security configuration

Security configuration
None

Permissions

Service role for Amazon EMR
EMR_DefaultRole

https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#ConnectedInstances?instanceId=i-00524879b3fb4f446

© 2024 Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EMR-fruit-cluster

Updated less than a minute ago

Terminate Clone in AWS CLI Clone

Summary

| Cluster info | Applications | Cluster management | Status and time |
|-------------------------------|---|--|---|
| Cluster ID j-17KL8PPV3A703 | Amazon EMR version emr-6.7.0 | Log destination in Amazon S3 openclassrooms-p8-fruits-data/logs | Status Terminated |
| Cluster configuration | Installed applications | Persistent application UIs Spark History Server | Creation time February 14, 2024, 11:39 (UTC+01:00) |
| Instance groups | Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1 | YARN timeline server | Elapsed time 1 hour, 52 minutes |
| Capacity | Primary node 2 Core 0 Task | Primary node public DNS ec2-13-36-237-163.eu-west-3.compute.amazonaws.com | End time February 14, 2024, 13:11 (UTC+01:00) |
| | | Connect to the Primary node using SSH | |

Sauvegarde des résultats

Amazon S3 > Buckets > openclassrooms-p8-fruits-data > data/ > results/

results/

Objects Properties

Objects (26) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Show versions

| <input type="checkbox"/> | Name | Type | Last modified | Size | Storage class |
|--------------------------|---|---------|---|--------|---------------|
| <input type="checkbox"/> | __SUCCESS | - | February 14, 2024, 12:18:22 (UTC+01:00) | 0 B | Standard |
| <input type="checkbox"/> | part-00000-8c3ffd48-e448-4a35-9113-482cfddd55b2-c000.snappy.parquet | parquet | February 14, 2024, 12:13:16 (UTC+01:00) | 3.5 MB | Standard |
| <input type="checkbox"/> | part-00001-8c3ffd48-e448-4a35-9113-482cfddd55b2-c000.snappy.parquet | parquet | February 14, 2024, 12:13:16 (UTC+01:00) | 3.5 MB | Standard |
| <input type="checkbox"/> | part-00002-8c3ffd48-e448-4a35-9113-482cfddd55b2-c000.snappy.parquet | parquet | February 14, 2024, 12:13:44 (UTC+01:00) | 3.5 MB | Standard |
| <input type="checkbox"/> | part-00003-8c3ffd48-e448-4a35-9113-482cfddd55b2-c000.snappy.parquet | parquet | February 14, 2024, 12:13:43 (UTC+01:00) | 3.5 MB | Standard |
| <input type="checkbox"/> | part-00004-8c3ffd48-e448-4a35-9113-482cfddd55b2-c000.snappy.parquet | parquet | February 14, 2024, 12:14:11 (UTC+01:00) | 3.5 MB | Standard |



Fruits!

Conclusion

- Mise en place d'une **architecture *Big Data*** avec AWS (EMR + S3 + IAM).
- Appropriation de la **chaîne de traitement d'images**.
- L'utilisation d'un environnement *Big Data* offre des avantages en termes de **performance**, d'**évolutivité** et de **préparation pour l'avenir**.

Merci pour votre attention

Des questions ?



Fruits!