



Fruits!

Déployer un modèle dans le cloud

1 sur 30

Sommaire

1. Problématique
2. Jeu de données
3. Création de l'environnement Big Data, S3 et EMR
4. Chaîne de traitement des images dans un environnement Big Data dans le cloud
5. Exécution du script PYSpark sur le Cloud
6. Synthèse et conclusion

1.Problématique

Projet de déploiement sur le cloud

- Start-up de l'AgriTech « Fruits! » de solutions innovantes pour la récolte des fruits
- Objectif : préserver la biodiversité des fruits
- Etape : Création d'une application mobile de reconnaissance d'images de légumes et fruits
- Contexte de la mission : le volume de données va augmenter rapidement, il faut donc un environnement Big Data
- Mission :
 - ➔ Développer les 1ères briques de cette application mobile en se servant d'un document déjà existant
 - ➔ S'approprier le notebook de l'alternant
 - ➔ Ajouter une étape de réduction de dimension des features
 - ➔ Création d'un environnement cloud sur AWS pour déployer le code

2. Présentation du jeu de données

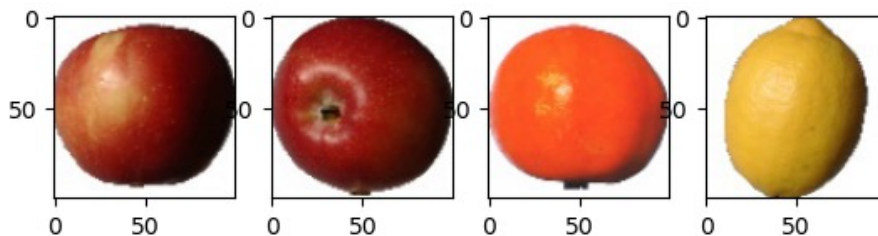
Jeu de données : 2 dossiers

fruits-360_dataset

- ✓ **Sous-dossier de même nom**
- ✓ Training set avec 67692 images
- ✓ Test set avec 22688 images
- ✓ 131 classes
- ✓ Taille 100x100 pixels
- ✓ Images de fruits et légumes
- ✓ Plusieurs variétés du même fruit
- ✓ Photos sous différents angles

Fruits-360-original-size

- ✓ **Sous-dossier de même nom**
- ✓ Nouvelle version du dataset



3. Création de l'environnement Big Data, S3 et EMR

Choix du prestataire AWS : Amazon Web Service

- Offre la plus large, la plus connue, et parfaitement adaptée ici
- Objectif 1 : louer de la puissance de calcul à la demande
- Objectif 2 : avoir suffisamment de puissance de calcul quand le volume de données augmentera, donc pouvoir faire évoluer les services
- Objectif 3 : moins cher qu'une location de matériel sur durée fixe (1 an par exemple)

Choix de la solution technique : EMR

2 possibilités :

- **Solution IAAS** (Infrastructure As A Service) :

Avantages : liberté totale, facilité de mise en œuvre comme en local

Inconvénients : chronophage, problèmes techniques possibles, non pérenne dans le temps (mises à jour des outils)

- **Solution PAAS** (Plateforme As A Service) :

Le service EMR d'AWS : location d'INSTANCES EC2 avec application préinstallées et configurées (Spark, Tensorflow, JupyterHub, packages complémentaires sur le serveur et l'ensemble des machines)

Avantages : Facilité et rapidité de mise en œuvre, solutions matérielles et logicielles optimisées par les ingénieurs d'AWS, donc stabilité et évolution de la solution, plus sécurisé (mise à jour des patches de sécurité)

Inconvénients : liberté sur la version des packages à confirmer

→ Le service EMR (solution PAAS) répond pleinement à notre problématique

Choix de la solution de stockage : S3

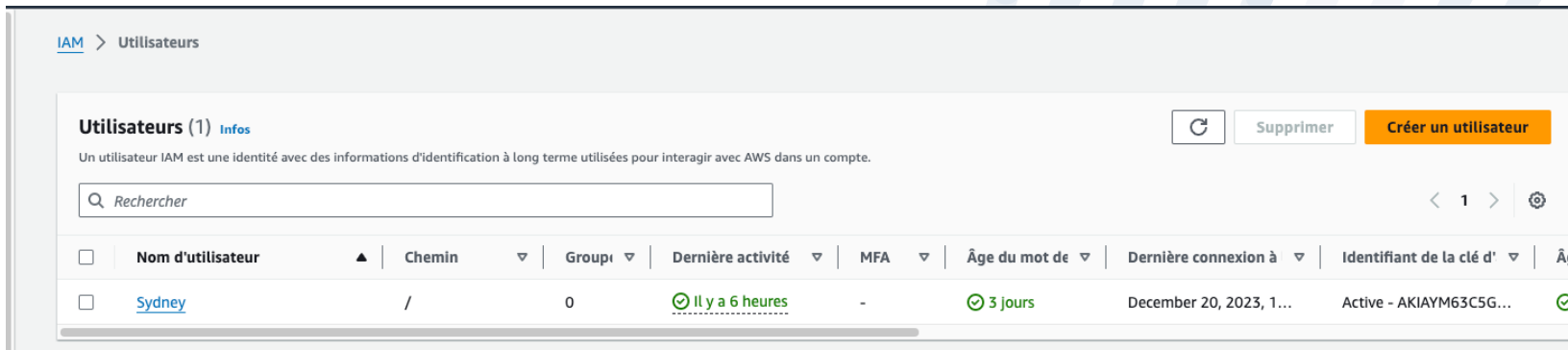
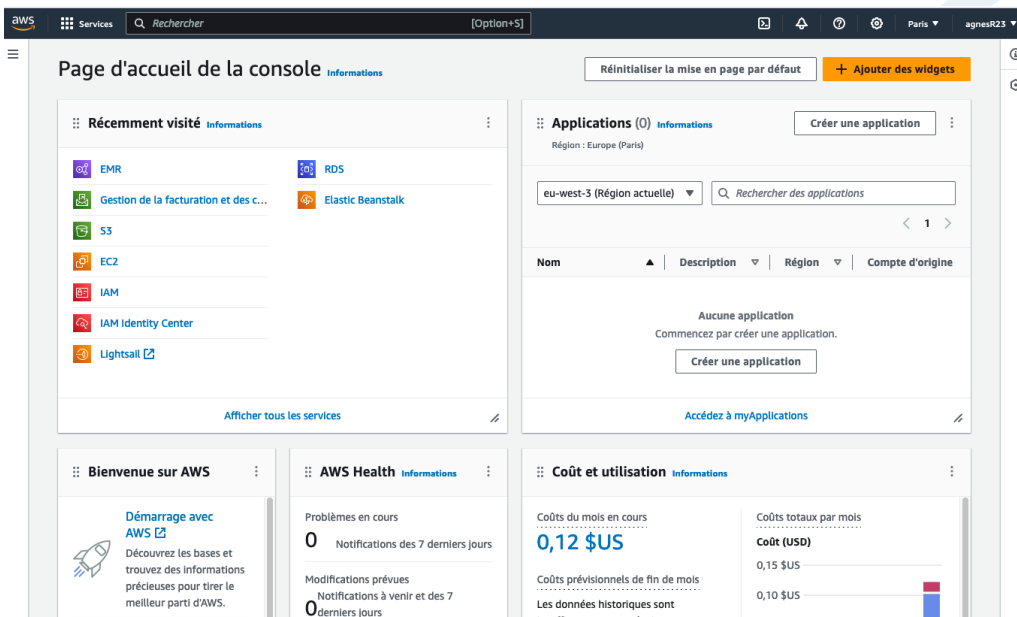
Si stockage sur l'espace alloué par le serveur EC2 alors :

- données perdues quand serveur résilié donc données à sauvegarder ensuite sur un autre support
- possibilité de saturation de l'espace disponible de nos serveurs

Si stockage sur S3 :

- espace disque disponible illimité, indépendant des serveurs EC2
- accès aux données très rapide en choisissant la même région pour nos serveurs EC2 et S3
- données faciles d'accès comme en local

1. Création d'un espace AWS et d'un utilisateur « Sydney » avec contrôle total sur le service S3



2. Upload des données sur S3 : dossier Test d'images, les features des images et les features réduites par PCA seront par la suite enregistrées dedans

Amazon S3

Compartiments

Access Grants [Nouveau](#)

Points d'accès

Points d'accès de l'objet Lambda

Points d'accès multi-région

Opérations par lot

IAM Access Analyzer pour S3

Paramètres de blocage de l'accès public pour ce compte

▼ Storage Lens

Tableaux de bord

Groupes Storage Lens

Paramètres AWS Organizations

Fonctionnalité spot 7







Amazon S3 > Compartiments > p8-data-2023

p8-data-2023 [Info](#)

Objets | Propriétés | Autorisations | Métriques | Gestion | Points d'accès

Objets (5) [Info](#)






Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'[inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autr accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)

  Copier l'URI S3  Copier l'URL  Télécharger  Ouvrir  Supprimer

Actions ▼

Créer un dossier

☐ Afficher les versions

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de s
<input type="checkbox"/>	 bootstrap-emr.sh	sh	23 Dec 2023 07:42:36 PM CET	333.0 o	Standard
<input type="checkbox"/>	 data/	Dossier	-	-	-
<input type="checkbox"/>	 jupyter/	Dossier	-	-	-
<input type="checkbox"/>	 Results/	Dossier	-	-	-
<input type="checkbox"/>	 Test/	Dossier	-	-	-

3. Configuration du serveur EMR

3.1 Nom du cluster, version emr (ici version moins récente sinon problèmes avec Tensorflow et keras) , logiciels de base

The screenshot displays the AWS Management Console interface for configuring an Amazon EMR cluster. The top navigation bar shows the 'Services' menu, a search bar, and the current region 'Paris' (highlighted with a yellow circle and a yellow arrow). The breadcrumb trail indicates the path: Amazon EMR > EMR sur EC2: Clusters > Créer un cluster.

The main content area is titled 'Cloner « P8_Fruits »' and is divided into two main sections: 'Nom et applications' and 'Récapitulatif'.

Nom et applications

- Nom:** A text input field containing 'P8_Fruits'.
- Version Amazon EMR:** A dropdown menu set to 'emr-6.7.0'. Below it, a note states: 'Une version contient un ensemble d'applications susceptibles d'être installées sur votre cluster.'
- Offre d'applications:** A grid of application stacks: Spark, Core Hadoop, HBase, Presto, Trino, and Custom (selected). Each stack has a corresponding logo.
- Logiciels de base (Base Software):** A list of software packages with checkboxes:
 - ☐ Flink 1.14.2
 - ☐ HCatalog 3.1.3
 - ☐ Hue 4.10.0
 - ☐ Livy 0.7.1
 - ☐ Phoenix 5.1.2
 - ☒ Spark 3.2.1
 - ☐ Tez 0.9.2
 - ☐ ZooKeeper 3.5.7
 - ☐ Ganglia 3.7.2
 - ☒ Hadoop 3.2.1
 - ☐ JupyterEnterpriseGateway 2.1.0
 - ☐ MXNet 1.8.0
 - ☐ Pig 0.17.0
 - ☐ Sqoop 1.4.7
 - ☐ Trino 378
 - ☐ HBase 2.4.4
 - ☐ Hive 3.1.3
 - ☒ JupyterHub 1.4.1
 - ☐ Oozie 5.2.1
 - ☐ Presto 0.272
 - ☒ TensorFlow 2.4.1
 - ☐ Zeppelin 0.10.0

Récapitulatif

- Nom et applications:** A summary of the configuration:
 - Nom:** P8_Fruits
 - Version Amazon EMR:** emr-6.7.0
 - Offre d'applications:** Custom (Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1, TensorFlow 2.4.1)
 - Version Amazon Linux:** 2.0.20231206.0
- Configuration de cluster:**
 - Groupes d'instances:** Primaire (m5.xlarge), Unité principale (m5.xlarge), Tâche (m5.xlarge)
 - Dimensionnement et mise en service du cluster:** (Link to further configuration)

3.2 Paramétrage de JupyterHub : enregistrement et ouverture des notebooks directement sur S3

▼ Paramètres logiciels - facultatif [Info](#)

☒ Entrer la configuration ☐ Charger JSON à partir d'Amazon S3

```
1 {  
2   {  
3     "Classification": "jupyter-s3-conf",  
4     "Properties": {  
5       "s3.persistence.bucket": "p8-data-2023",  
6       "s3.persistence.enabled": "true"  
7     }  
8   }  
9 }
```

3.3 Configuration matériel : sélection des instances

Configuration de cluster Info

Choisissez une méthode de configuration pour les groupes de nœuds primaires, principaux et de tâches de votre cluster.

☒ **Groupes d'instances**
Choisir un type d'instance par groupe de nœuds

☐ **Flottes d'instances**
Choisir une combinaison de types d'instance au sein de chaque groupe de nœuds

Groupes d'instances

Primaire

Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire EBS uniquement stockage
Prix à la demande : 0.224 USD par instance/heure
Prix Spot le plus bas : \$0.070 (eu-west-3b)

Actions ▼

Dimensionnement et mise en service du cluster Info

Configurez des configurations de dimensionnement et de provisionnement pour les groupes de nœuds principaux et de tâches de votre cluster.

Choisir une option

☒ **Définir manuellement la taille du cluster**
Utilisez cette option si vous connaissez vos modèles de charge de travail à l'avance.

☐ **Utiliser la mise à l'échelle gérée par EMR**
Surveillez les principales métriques de charges de travail afin qu'EMR puisse optimiser la taille du cluster et l'utilisation des ressources.

☐ **Utiliser un autoscaling personnalisé**
Pour dimensionner de manière programmatique les unités principales et les nœuds de tâches, créez des politiques d'autoscaling personnalisées.

Configuration de mise en service

Définissez la taille de votre noyau et tâchegroupes d'instance. Amazon EMR tente de fournir cette capacité lorsque vous lancez votre cluster.

Nom	Type d'instance	Taille de l'instance(s)	Utiliser l'option d'achat Spot
Tâche - 1	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>
Unité principale	m5.xlarge	<input type="text" value="2"/>	<input type="checkbox"/>

Unité principale

Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire EBS uniquement stockage
Prix à la demande : 0.224 USD par instance/heure
Prix Spot le plus bas : \$0.070 (eu-west-3b)

Actions ▼

► Configuration de nœud - facultatif

Tâche 1 sur 1

Retirer le groupe d'instances

Nom

Choisir un type d'instance EC2

m5.xlarge
4 vCore 16 GiB mémoire EBS uniquement stockage
Prix à la demande : 0.224 USD par instance/heure
Prix Spot le plus bas : \$0.070 (eu-west-3b)

Actions ▼

3.4 Configuration résiliation du cluster

Résiliation du cluster [Info](#)

- ☐ Résilier manuellement le cluster
- ☐ Résilier automatiquement le cluster à la fin de la dernière étape
- ☒ Résilier automatiquement le cluster après le temps d'inactivité (Recommandé)

Temps d'inactivité

Saisissez la durée avant la résiliation de votre cluster.

0 jour 01:00:00

Choisissez une durée supérieure à 1 minute (00:01:00) et inférieure à 7 jours. L'heure est au format hh:mm:ss (24 heures).

- ☐ Utiliser la protection contre la résiliation
Protégez vos instances EC2 de la résiliation accidentelle.

3.5 Ajout d'une action d'amorçage pour installer les packages manquants

▼ Actions d'amorçage – facultatif [Info](#)

Utilisez les actions d'amorçage pour installer des logiciels ou personnaliser la configuration de votre instance.

Actions d'amorçage (1)

Supprimer

Modifier

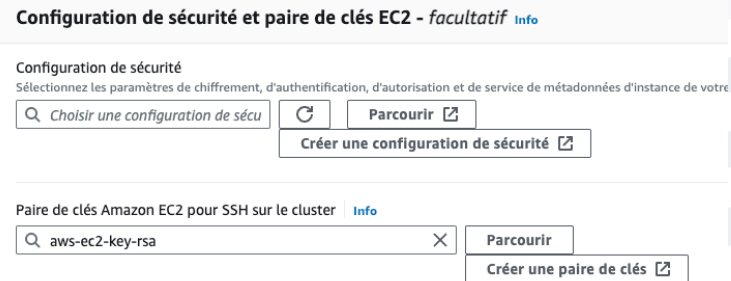
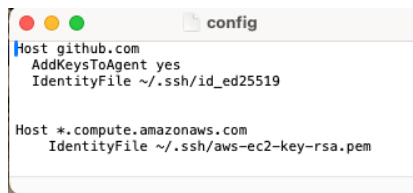
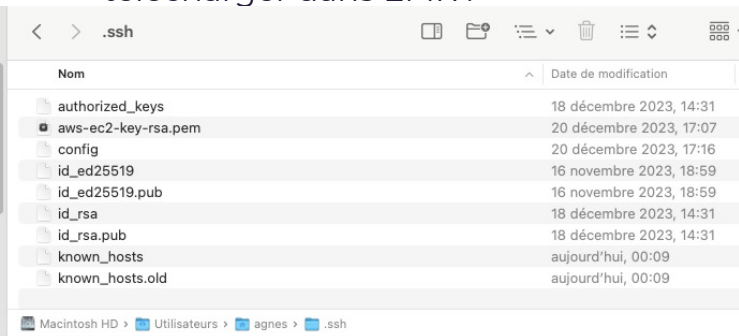
Ajouter

	Nom	Emplacement Amazon S3	Arguments
<input type="radio"/>	Packages_install	s3://p8-data-2023/bootstrap-emr.sh	-

```
bootstrap-emr.sh
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
```


3.6 Ajout de la paire de clés de sécurité EC2 créée précédemment

Cliquer sur "Paires de clés" sous l'onglet "Réseau et sécurité » dans EC2 d'AWS. En cliquant sur "Créer une paire de clés" vous générez une clé privée que vous pouvez nommer et télécharger sur votre ordinateur puis à télécharger dans EMR :



3.7 Choix des rôles automatique puis « Créer cluster »

Rôle Identity and Access Management (IAM) [Info](#)

Choisissez ou créez une fonction du service et un profil d'instance pour les instances EC2 de votre cluster.

Fonction du service Amazon EMR [Info](#)

La fonction du service est un rôle IAM assumé par Amazon EMR pour mettre en service des ressources et effectuer des actions au niveau du service avec d'autres services AWS.

☐ Choisir une fonction du service existant

Sélectionnez une fonction du service par défaut ou un rôle personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec d'autres services AWS.

☒ Créez une fonction du service

Laissez Amazon EMR créer une nouvelle fonction du service afin que vous puissiez accorder et restreindre l'accès aux ressources d'autres services AWS.

Profil d'instance EC2 pour Amazon EMR

Le profil d'instance attribue un rôle à chaque instance EC2 d'un cluster. Le profil d'instance doit spécifier un rôle qui peut accéder aux ressources pour vos étapes et actions d'amorçage.

☐ Choisir un profil d'instance existant

Sélectionnez un rôle par défaut ou un profil d'instance personnalisé avec des stratégies IAM attachées afin que votre cluster puisse interagir avec vos ressources dans Amazon S3.

☒ Choisir un profil d'instance

Laissez Amazon EMR créer un profil d'instance afin de pouvoir spécifier un ensemble personnalisé de ressources auquel il peut accéder dans Amazon S3.

Accès au compartiment S3 [Info](#)

☐ Compartiments S3 ou préfixes spécifiques de votre compte [Info](#)

Choisissez les compartiments ou préfixes auxquels vous voulez que ce profil d'instance accède.

☒ Tous les compartiments S3 de ce compte avec accès en lecture et en écriture

Accordez au profil d'instance l'accès à tous les compartiments pour lesquels l'accès en lecture et en écriture est activé dans votre compte.

Récapitulatif, instantiation du serveur

[Amazon EMR](#) > [EMR sur EC2: Clusters](#) > P8_Fruits

P8_Fruits

Mise à jour il y a moins d'une minute



Résilier

Cloner dans AWS CLI

▼ Récapitulatif

Informations sur le cluster

ID de cluster
j-2YI7DP8NAUG4

Configuration de cluster
Groupes d'instances

Capacité
1 primaire(s) | 2 unité(s) principale(s) | 1 tâche(s)

Applications

Version d'Amazon EMR
emr-6.7.0

Applications installées
Hadoop 3.2.1, JupyterHub 1.4.1, Spark 3.2.1,
TensorFlow 2.4.1

Gestion des clusters

Destination des journaux dans Amazon S3
[aws-logs-577595697594-eu-west-3/elasticmapreduce](#)

Interfaces utilisateur d'application persistantes
[Serveur d'historique Spark](#)
[Serveur de chronologie YARN](#)

DNS public du nœud primaire
 ec2-15-188-62-211.eu-west-3.compute.amazonaws.com

[Connexion au nœud primaire à l'aide de SSH](#)
[Connexion au nœud primaire à l'aide de SSM](#)

Statut et heure

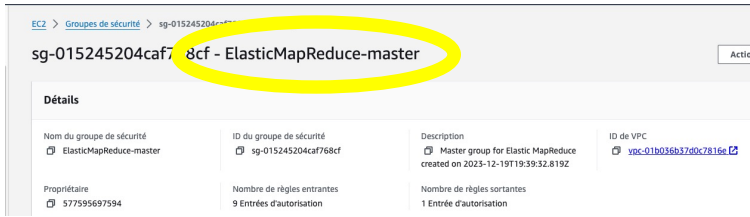
Statut
 En attente

Heure de création
23 décembre 2023 23:55 (UTC+01:00)

Temps écoulé
1 heure, 12 minutes

4 Création du tunnel SSH à l'instance EC2 (Maître)

4.1 Autorisations sur les connexions entrantes



4.2 Création du tunnel ssh vers le driver

Ligne de commande :

```
(env_p8) agnes@MacStudio Projet_8 % ssh -i ~/aws-ec2-key-rsa.pem -D 8157 hadoop@ec2-15-188-62-211.eu-west-3.compute.amazonaws.com
```

Règles entrantes (9)

	Name	ID de règle de grou...	Version IP	Type	Protocole	Plage de ports
<input type="checkbox"/>	-	sgr-037458d7606004...	-	Tous les ICMP - IPv4	ICMP	Tous
<input type="checkbox"/>	-	sgr-0b4af518774a3a438	-	Tous les UDP	UDP	0 - 65535
<input type="checkbox"/>	-	sgr-00a4dc5e4d40...	-	Tous les TCP	TCP	8443
<input type="checkbox"/>	-	sgr-09fdaa9d981a2356b	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0e0db7c368a6b461...	IPv6	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0290183a2af1b0c0e	-	Tous les TCP	TCP	0 - 65535
<input type="checkbox"/>	-	sgr-0a8dbc04989a7c3e8	-	Tous les TCP	TCP	0 - 65535
<input type="checkbox"/>	-	sgr-097340f877d08c42a	-	Tous les ICMP - IPv4	ICMP	Tous
<input type="checkbox"/>	-	sgr-07f4df7310cf2ab5d	-	Tous les TCP	TCP	0 - 65535

Activer une connexion SSH

Les applications EMR publient des interfaces utilisateur sous forme de sites Web hébergés sur le nœud primaire. Pour des raisons de sécurité, ces sites Web ne sont disponibles que sur le serveur Web local du nœud primaire.

Pour accéder aux interfaces Web, vous devez établir un tunnel SSH avec le nœud primaire à l'aide d'une redirection de port dynamique ou locale. Si vous utilisez la redirection de port dynamique, vous devez également configurer un serveur proxy pour afficher les interfaces Web. [En savoir plus](#)

Étape 1: Ouvrez un tunnel SSH vers le nœud primaire Amazon EMR.

Windows | **Mac/Linux**

- Ouvrez une fenêtre de terminal. Sur Mac OS X, sélectionnez **Applications > Utilitaires > Terminal**. Sur les autres distributions Linux, le terminal se trouve généralement dans **Applications > Accessoires > Terminal**.
- Pour établir un tunnel SSH avec le nœud primaire à l'aide de la redirection de port dynamique, entrez la commande suivante. Remplacez `~/aws-ec2-key-rsa.pem` par l'emplacement et le nom de fichier du fichier de clé privée (.pem) utilisé pour lancer le cluster.

```
ssh -i ~/aws-ec2-key-rsa.pem -D 8157 hadoop@ec2-15-188-62-211.eu-west-3.compute.amazonaws.com
```

Remarque : le port 8157 utilisé dans la commande est un port local non utilisé sélectionné au hasard.

3. Saisissez yes (oui) pour ignorer l'avertissement de sécurité.

Résultat :
tunnel ssh correctement
établi port 8157 mais clé à
protéger avec chmod 400

```
(env_p8) agnes@MacBook-Projet_8 % ssh -i /aws-ec2-key-rsa.pem -D 8157 hadoop@ec2-15-188-62-211.eu-west-3.compute.amazonaws.com
The authenticity of host 'ec2-15-188-62-211.eu-west-3.compute.amazonaws.com (15.188.62.211)' can't be established.
ED25519 key fingerprint is SHA256:g9CR9qewjUbHmZrQTQdRMv3DyZuGnLg3yn1u8MtGpJ0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-15-188-62-211.eu-west-3.compute.amazonaws.com' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/Users/agnes/aws-ec2-key-rsa.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/agnes/aws-ec2-key-rsa.pem": bad permissions
Last login: Sat Dec 23 23:04:48 2023

      #_
     _\_\_ #####
    ~~~ \_\_#####\
         \###|
         \#/
        ~~~ V~' i-->

                                     Amazon Linux 2

                                     AL2 End of Life is 2025-06-30.

                                     A newer version of Amazon Linux is available!

                                     Amazon Linux 2023, GA and supported until 2028-03-15.
                                     https://aws.amazon.com/linux/amazon-linux-2023/

3 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEEEE MMMMMMMM          MMMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::E M:::M M:::M R:::R
EE:::::EEEEEEEE::E M:::M M:::M R:::R
E:::E EEEE M:::M M:::M RR::R R:::R
E:::E M:::M M:::M M:::M R::R R:::R
E:::::EEEEEEEE M:::M M:::M M:::M R::RRRRR::R
E::::::::::::E M:::M M:::M M:::M R:::RRRR::RR
E:::::EEEEEEEE M:::M M:::M M:::M R::RRRRR:::R
E:::E M:::M M:::M M:::M R::R R:::R
E:::E EEEE M:::M MM R:::R R:::R
EE:::::EEEEEEEE::E M:::M M:::M R:::R R:::R
E::::::::::::E M:::M M:::M RR::R R:::R
EEEEEEEEEEEEEEEEEEEE MMMMMMMM RRRRRRR RRRRRR
```

5 Configuration SwitchyOmega sur Google Chrome

Accès aux applications du serveur EMR via le tunnel ssh :
Connexion au notebook JupyterHub :

Example: Configure SwitchyOmega for chrome

The following example demonstrates how to set up the SwitchyOmega extension for Google Chrome. SwitchyOmega lets you configure, manage, and switch between multiple proxies.

To install and configure SwitchyOmega using Google Chrome

1. Go to <https://chrome.google.com/webstore/category/extensions>, search for **Proxy SwitchyOmega**, and add it to Chrome.
2. Choose **New profile** and enter `emr-socks-proxy` as the profile name.
3. Choose **PAC profile** and then **Create**. [Proxy Auto-Configuration \(PAC\)](#) files help you define an allow list for browser requests that should be forwarded to a web proxy server.
4. In the **PAC Script** field, replace the contents with the following script that defines which URLs should be forwarded through your web proxy server. If you specified a different port number when you set up your SSH tunnel, replace `8157` with your port number.

```
function FindProxyForURL(url, host) {  
    if (shExpMatch(url, "*ec2*.compute*.amazonaws.com*")) return 'SOCKS5 localhost:  
    if (shExpMatch(url, "*ec2*.compute*")) return 'SOCKS5 localhost:  
    if (shExpMatch(url, "http://10.*")) return 'SOCKS5 localhost:8157  
    if (shExpMatch(url, "*10*.compute*")) return 'SOCKS5 localhost:  
    if (shExpMatch(url, "*10*.amazonaws.com*")) return 'SOCKS5 localhost:  
    if (shExpMatch(url, "*compute.internal*")) return 'SOCKS5 localhost:  
    if (shExpMatch(url, "*ec2.internal*")) return 'SOCKS5 localhost:  
    return 'DIRECT';  
}
```

The screenshot shows the JupyterHub web interface. At the top, there's a navigation bar with the JupyterHub logo and links for 'Logout' and 'Control Panel'. Below this, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser interface. It includes a search bar, a list of items (currently empty), and buttons for 'Upload', 'New', and 'Refresh'. Below the file browser, there's a table of notebooks. The table has columns for 'Name', 'Last Modified', and 'File size'. One notebook is listed: 'Regaud_Agnès_1_notebook_122023.ipynb'. The status bar at the bottom indicates 'Actif il y a une heure'.

4. Chaîne de traitement des images dans un environnement Big Data dans le cloud

Méthode :

- 1^{ère} étape de test de la solution en local, puis sur le cloud
- Exécution du code sur JupyterHub avec kernel pyspark
- Une session spark est créée à l'exécution de la 1^{ère} cellule, donc pas de script « SparkSession »

jupyterhub Regaud_Agnès_1_notebook_122023 (auto-sauvegardé) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Exécuter Code

3. Déploiement de la solution sur le cloud

3.1 Exécution du code

3.1.1 Démarrage de la session Spark

Entrée [1]: `# L'exécution de cette cellule démarre l'application Spark`

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1703372446295_0001	pyspark	idle	Link	Link	None	✓

SparkSession available as 'spark'.

Affichage des informations sur la session en cours et liens vers Spark UI :

Entrée [2]: `%info`

Current session configs: {'driverMemory': '1000M', 'executorCores': 2, 'proxyUser': 'joyvan', 'kind': 'pyspark'}

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1703372446295_0001	pyspark	idle	Link	Link	None	✓

3.1.3 Import des librairies

Entrée [3]:

```
import pandas as pd
import numpy as np
import io
import os
import tensorflow as tf
from PIL import Image
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras import Model
from pyspark.sql.functions import col, pandas_udf, PandasUDFType, element_at, split
```

Entrée [4]:

```
#sur le cloud :
PATH = 's3://p8-data-2023'
PATH_Data = PATH+'/Test'
PATH_Result = PATH+'/Results'
PATH_Result_PCA = PATH+'/data/Result_PCA'
print('PATH: '+\
      PATH+'\nPATH_Data: '+\
      PATH_Data+'\nPATH_Result: '+PATH_Result
      +'\nPATH_Result_PCA: '+PATH_Result_PCA)
```

```
PATH:      s3://p8-data-2023
PATH_Data: s3://p8-data-2023/Test
PATH_Result: s3://p8-data-2023/Results
PATH_Result_PCA: s3://p8-data-2023/data/Result_PCA
```

1. Chargement des données

- Chargées au format binaire
- Seulement extension .jpg
- A l'intérieur des sous-dossiers
- Ajout d'une colonne avec label de l'image
- Redimensionnement des images pour qu'elles soient compatibles avec notre modèle (configuration du modèle pour avoir des images. De dimension (224, 224, 3) et non (100, 100, 3))

3.1.5.1 Chargement des données

Entrée [5]:

```
images = spark.read.format("binaryFile") \  
    .option("pathGlobFilter", "*.jpg") \  
    .option("recursiveFileLookup", "true") \  
    .load(PATH_Data)
```

Entrée [6]:

```
images.show(5)
```

path	modificationTime	length	content
s3://p8-data-2023...	2023-12-23 14:05:11	7353	[FF D8 FF E0 00 1...
s3://p8-data-2023...	2023-12-23 14:05:12	7350	[FF D8 FF E0 00 1...
s3://p8-data-2023...	2023-12-23 14:05:11	7349	[FF D8 FF E0 00 1...
s3://p8-data-2023...	2023-12-23 14:05:11	7348	[FF D8 FF E0 00 1...
s3://p8-data-2023...	2023-12-23 14:05:12	7328	[FF D8 FF E0 00 1...

only showing top 5 rows

Je ne conserve que le **path** de l'image et j'ajoute une colonne contenant les **labels** de chaque image :

Entrée [7]:

```
images = images.withColumn('label', element_at(split(images['path'], '/'),-2)) \  
print(images.printSchema()) \  
print(images.select('path','label').show(5,False))
```

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

None

path	label
s3://p8-data-2023/Test/Watermelon/r_106_100.jpg	Watermelon
s3://p8-data-2023/Test/Watermelon/r_109_100.jpg	Watermelon
s3://p8-data-2023/Test/Watermelon/r_108_100.jpg	Watermelon
s3://p8-data-2023/Test/Watermelon/r_107_100.jpg	Watermelon
s3://p8-data-2023/Test/Watermelon/r_95_100.jpg	Watermelon

only showing top 5 rows

None

2. Préparation du modèle

Contexte :

- Technique du transfert learning pour extraire features
- Modèle MobileNetV2 plus rapide que VGG16
- Sa dernière couche sert à classifier les images selon 1000 catégories : pas besoin ici
- Faible dimensionnalité du vecteur de caractéristique en sortie (1,1,1280)

Etapes :

- Chargement du modèle MobileNetV2 avec poids précalculés issus d'imagenet avec format des images en entrée
- Création nouveau modèle,
 - ✓ en entrée : l'entrée du modèle MobileNetV2
 - ✓ en sortie : avant-dernière couche du modèle MobileNetV2

3.1.5.2 Préparation du modèle

```
Entrée [8]: model = MobileNetV2(weights='imagenet',
                                include_top=True,
                                input_shape=(224, 224, 3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weight
s_tf_dim_ordering_tf_kernels_1.0_224.h5
14540800/14536120 [=====] - 1s 0us/step
```

```
Entrée [9]: new_model = Model(inputs=model.input,
                              outputs=model.layers[-2].output)
```

```
Entrée [10]: broadcast_weights = sc.broadcast(new_model.get_weights())
```

```
Entrée [11]: new_model.summary()
```

block_16_depthwise_BN (BatchNor (None, 7, 7, 960))	3840	block_16_depthwise[0][0]
block_16_depthwise_relu (ReLU (None, 7, 7, 960))	0	block_16_depthwise_BN[0][0]

global_average_pooling2d (Globa (None, 1280))	0	out_relu[0][0]
---	---	----------------

=====

Total params: 2,257,984
Trainable params: 2,223,872
Non-trainable params: 34,112

```
Entrée [12]: def model_fn():
    """
    Returns a MobileNetV2 model with top layer removed
    and broadcasted pretrained weights.
    """
    model = MobileNetV2(weights='imagenet',
                        include_top=True,
                        input_shape=(224, 224, 3))

    for layer in model.layers:
        layer.trainable = False
    new_model = Model(inputs=model.input,
                      outputs=model.layers[-2].output)
    new_model.set_weights(broadcast_weights.value)
    return new_model
```

3. Chargement des images et application de leur featurisation à travers l'utilisation de pandas UDF

```
[13]: def preprocess(content):  
    """  
    Preprocesses raw image bytes for prediction.  
    """  
    img = Image.open(io.BytesIO(content)).resize([224, 224])  
    arr = img_to_array(img)  
    return preprocess_input(arr)  
  
def featurize_series(model, content_series):  
    """  
    Featurize a pd.Series of raw images using the input model.  
    :return: a pd.Series of image features  
    """  
    input = np.stack(content_series.map(preprocess))  
    preds = model.predict(input)  
    # For some layers, output features will be multi-dimensional tensors.  
    # We flatten the feature tensors to vectors for easier storage in Spark DataFrames.  
    output = [p.flatten() for p in preds]  
    return pd.Series(output)  
  
@pandas_udf('array<float>', PandasUDFType.SCALAR_ITER)  
def featurize_udf(content_series_iter):  
    """  
    This method is a Scalar Iterator pandas UDF wrapping our featurization function.  
    The decorator specifies that this returns a Spark DataFrame column of type ArrayType(FloatType).  
  
    :param content_series_iter: This argument is an iterator over batches of data, where each batch  
                                is a pandas Series of image data.  
    """  
    # With Scalar Iterator pandas UDFs, we can load the model once and then re-use it  
    # for multiple data batches. This amortizes the overhead of loading big models.  
    model = model_fn()  
    for content_series in content_series_iter:  
        yield featurize_series(model, content_series)
```

4. Chargement des images et application de leur featurisation

3.1.5.4 Exécutions des actions d'extractions de features et enregistrement des données

Entrée [14]: `spark.conf.set("spark.sql.execution.arrow.maxRecordsPerBatch", "1024")`

Entrée [15]: `features_df = images.repartition(24).select(col("path"),
col("label"),
featurize_udf("content").alias("features")
)`

Entrée [16]: `print(PATH_Result)`
`s3://p8-data-2023/Results`

Entrée [17]: `features_df.cache()
features_df.count()`

22688

Entrée [18]: `features_df.write.mode("overwrite").parquet(PATH_Result)`

5. PCA

Entrée [19]: `from pyspark.ml.feature import PCA
from pyspark.ml.linalg import Vectors, VectorUDT
from pyspark.sql.functions import udf`

Entrée [20]: `def array_to_vector(array):
 return Vectors.dense(array)

array_to_vector_udf = udf(array_to_vector, VectorUDT())

features_df = features_df.withColumn("features", array_to_vector_udf(features_df["features"]))`

Entrée [21]: `num_components = 50

pca = PCA(k=num_components, inputCol="features", outputCol="pca_features")
model_pca = pca.fit(features_df)
df_pca = model_pca.transform(features_df)`

Entrée [22]: `df_pca.show()`

path	label	features	pca_features
s3://p8-data-2023...	Watermelon	[0.02950825542211...	[-2.4517777256345...
s3://p8-data-2023...	Watermelon	[0.01487588509917...	[-1.8770516056425...
s3://p8-data-2023...	Pineapple Mini	[0.0,5.0234093666...	[-5.8802616017680...
s3://p8-data-2023...	Watermelon	[0.0,0.1364074498...	[-3.1825366756191...
s3://p8-data-2023...	Watermelon	[0.35576733946800...	[-2.5328938265756...
s3://p8-data-2023...	Cauliflower	[0.0,0.6837162971...	[-4.2470413505814...
s3://p8-data-2023...	Cauliflower	[0.0,0.3231029212...	[-5.9401509492805...
s3://p8-data-2023...	Cauliflower	[0.0,0.9533805847...	[-4.2638977170783...
s3://p8-data-2023...	Pineapple	[0.0,4.4910292625...	[-6.1131896207153...
s3://p8-data-2023...	Pineapple	[0.0,4.2289829254...	[-5.9483262939193...
s3://p8-data-2023...	Raspberry	[0.10317493230104...	[-0.2236674863237...
s3://p8-data-2023...	Cauliflower	[0.0,0.1326977014...	[-4.5731684251708...
s3://p8-data-2023...	Cauliflower	[0.0,0.9293980598...	[-4.8568615386459...
s3://p8-data-2023...	Pineapple Mini	[0.0,4.8620567321...	[-4.6354282873043...
s3://p8-data-2023...	Pineapple Mini	[0.02071469463407...	[-5.3803464537752...
s3://p8-data-2023...	Pineapple Mini	[0.0,3.7098712921...	[-6.5133351055317...
s3://p8-data-2023...	Apple Golden 1	[0.02250090986490...	[-3.8393572980781...
s3://p8-data-2023...	Onion White	[0.00932506751269...	[0.62060279861376...
s3://p8-data-2023...	Apple Golden 1	[0.0,0.0517689697...	[-3.2684578026181...
s3://p8-data-2023...	Lychee	[1.15856051445007...	[-3.0941282645856...

only showing top 20 rows

Entrée [23]: `df_pca.write.mode("overwrite").parquet(PATH_Result_PCA)`

Entrée [24]: `first_row_pca = df_pca.limit(1).toPandas()`
`print(first_row_pca)`

```
          path  ...          pca_features
0  s3://p8-data-2023/Test/Watermelon/r_87_100.jpg  ...  [-2.4517777256345616, 6.344217905351185, -5.50...
```

[1 rows x 4 columns]

Entrée [25]: `features_length = len(first_row_pca['features'])[0]`
`pca_features_length = len(first_row_pca['pca_features'])[0]`

`print(f"Length of 'features' vector: {features_length}")`
`print(f"Length of 'pca_features' vector: {pca_features_length}")`

Length of 'features' vector: 1280
Length of 'pca_features' vector: 50

Entrée [26]: `explained_variances = model_pca.explainedVariance`
`cumulative_variance = explained_variances.toArray().cumsum()`

`print(f"Cumulative variance explained by the first 50 components: {cumulative_variance[49]:.4f}")`

Cumulative variance explained by the first 50 components: 0.7318

Result_PCA/

Copier l'URI S3

Objets

Propriétés

Objets (25) [Info](#)

Les objets sont les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser [l'inventaire Amazon S3](#) pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. [En savoir plus](#)



Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Actions

Créer un dossier

Charger

Rechercher des objets en fonction du préfixe

Afficher les versions

< 1 > ⚙

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	_SUCCESS	-	24 Dec 2023 12:26:04 AM CET	0 o	Standard
<input type="checkbox"/>	part-00000-a286b095-64c5-4988-aea9-0f27bf5a275e-c000.snappy.parquet	parquet	24 Dec 2023 12:25:56 AM CET	5.4 Mo	Standard
<input type="checkbox"/>	part-00001-a286b095-64c5-4988-aea9-0f27bf5a275e-c000.snappy.parquet	parquet	24 Dec 2023 12:25:57 AM CET	5.4 Mo	Standard
<input type="checkbox"/>	part-00002-a286b095-64c5-4988-aea9-0f27bf5a275e-c000.snappy.parquet	parquet	24 Dec 2023 12:25:56 AM CET	5.4 Mo	Standard

**5. Exécution du script
PYSpark sur le Cloud**

6. Synthèse et conclusion