RÉALISEZ UN TRAITEMENT DANS UN ENVIRONNEMENT BIG DATA SUR LE CLOUD

Samira MAHJOUB – 10/2024

SOMMAIRE

- Contexte
- Environnement Big Data
- Chaine de traitement des images
- Démonstration
- Conclusion

Problématique :
Migrer vers une architecture Cloud pour gérer l'explosion des volumes de données tout en optimisant les coûts et respectant le RGPD.

Contexte :
Jeune start-up AgriTech “Fruits!” propose une app mobile de reconnaissance de fruits, sensibilisant à la biodiversité et alimentant des modèles d'IA

Objectif :

- Déployer une architecture Big Data scalable dans le Cloud pour traiter les images de fruits avec Py Spark et AWS EMR.

- Sensibiliser le grand public à la biodiversité des fruits via une application mobile.

CONTEXTE

JEU DE DONNÉES

Dataset Kaggle : Fruits-360

94 110 images contenues dans 141 dossiers de fruits et légumes

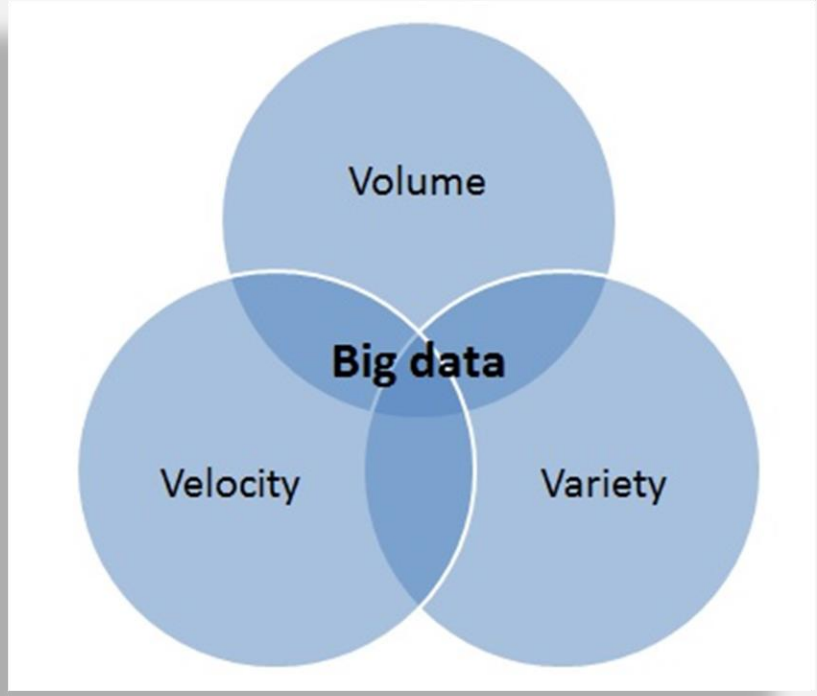
Taille des images : 100x100 pixels Format : jpg

Taille du jeu de test utilisé: 23 619 images (un fruit ou légume par image).

ENVIRONNEMENT BIG DATA

I. Qu'est-ce que le Big Data ?

➢ Le Big Data se réfère à des ensembles de données **massifs** et **complexes**, qui **dépassent les capacités des outils traditionnels pour le stockage**, le traitement et l'analyse. Ces données sont souvent volumineuses, variées et produites rapidement.



Les 3 V du Big Data :

• **Volume :** Quantité massive de données à traiter, provenant de sources variées comme des images, des transactions, des réseaux sociaux, etc.
• **Variété :** Les données peuvent être structurées (tableaux), semi-structurées (fichiers XML), ou non-structurées (images, vidéos, textes).
• **Vélocité :** Les données sont générées à une vitesse élevée et doivent être traitées en temps réel pour en tirer de la valeur.

Pourquoi le Big Data dans mon projet ?

Dans notre projet avec AWS EMR et S3, le Big Data est utilisé pour traiter et analyser des volumes massifs d'images de fruits, avec une infrastructure scalable et efficace qui s'adapte à la croissance des données.

ENVIRONNEMENT BIG DATA

II. Choix du Prestataire

Choix de Amazon Web Services (AWS)

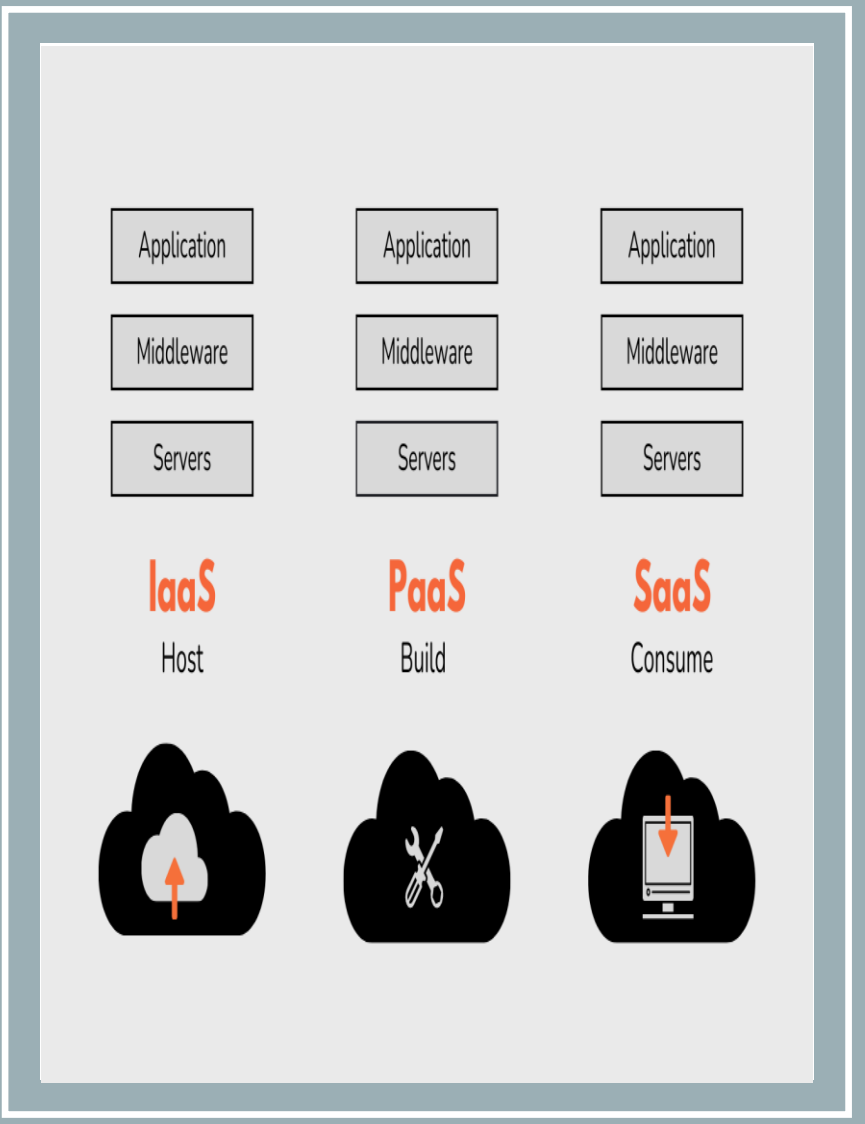
Cloud Computing à la demande :

- Puissance de calcul modulable selon les besoins du projet.
- AWS propose des solutions adaptées pour gérer des données massives, avec une infrastructure flexible.

Pourquoi AWS ?

- **Scalabilité** : Ajustement automatique des ressources pour s'adapter à la montée en charge.
- **Efficacité** : Optimisation des coûts grâce à un modèle de paiement à l'usage, évitant les dépenses liées à une infrastructure fixe.
- **Sécurité** : AWS est reconnu pour ses solutions robustes en termes de sécurité et de conformité avec les réglementations (ex. RGPD).

ENVIRONNEMENT BIG DATA



III. Choix de la Solution Technique

Choix d'une solution de type PaaS (Platform as a Service) avec AWS EMR permet de louer des instances EC2 préconfigurées pour le traitement des données massives.

Solution avec mise en œuvre facile et rapide (mise en place en moins de 15 minutes).

Offre robustesse et scalabilité pour répondre aux besoins d'un projet Big Data évolutif.

ENVIRONNEMENT BIG DATA

IV. Services AWS utilisés

Instance EC2

Serveur virtuel évolutif et flexible dans le cloud, permettant de gérer les applications et d'exécuter des charges de travail variées.

-- Région : eu-west-3c (Paris)

Stockage S3

Service de stockage d'objets dans le cloud, offrant une scalabilité illimitée et une haute disponibilité pour stocker et récupérer des données de manière sécurisée.

Gestion des accès IAM

Service de gestion des identités et des accès pour sécuriser les ressources AWS, en définissant des politiques et des permissions pour les utilisateurs, groupes et rôles.

Cluster EMR (Elastic MapReduce)

Service de gestion des clusters Hadoop/Spark pour le traitement de grandes quantités de données, optimisé pour l'analyse évolutive et le traitement par lots dans le cloud.

Version : emr-6.4.0

Applications : Hadoop 3.2.1, Spark 3.1.2

Instances : m5.xlarge

ENVIRONNEMENT BIG DATA

Intégration de Hadoop et Spark avec AWS pour le Big Data

Hadoop

Framework open-source permettant le traitement de grandes quantités de données via le modèle MapReduce.

Avantages:

- Parallélisation des tâches grâce au modèle de traitement par lots.
- Très utilisé dans les solutions Big Data pour sa robustesse.
- Système distribué, idéal pour le stockage massif (HDFS).

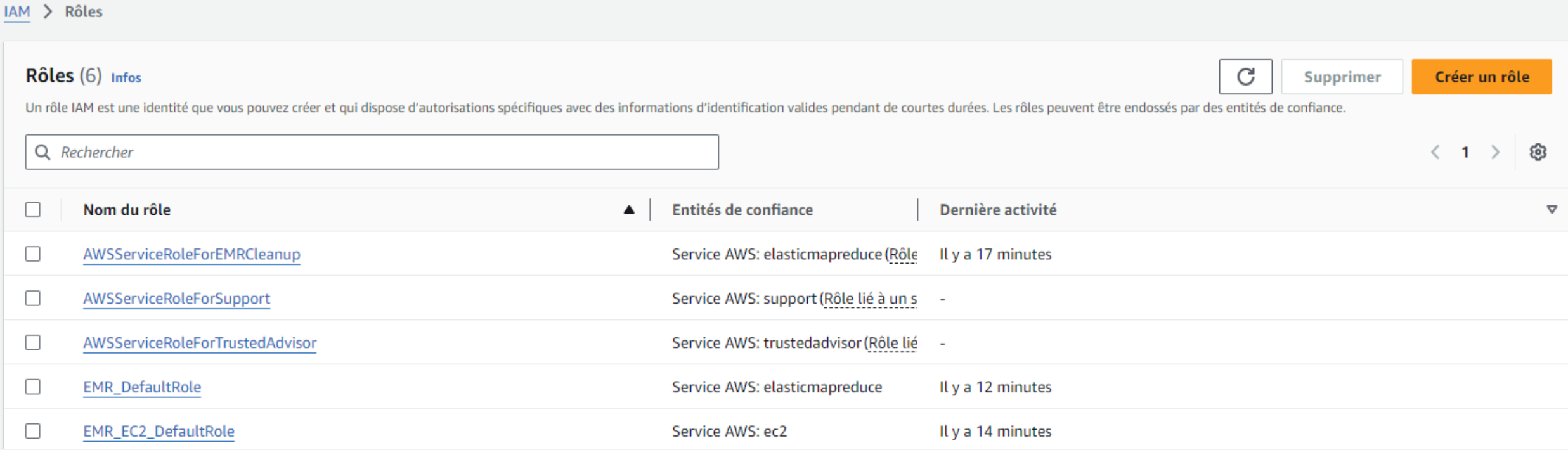
Spark

Framework conçu pour un traitement rapide des données en mémoire.

Avantages :

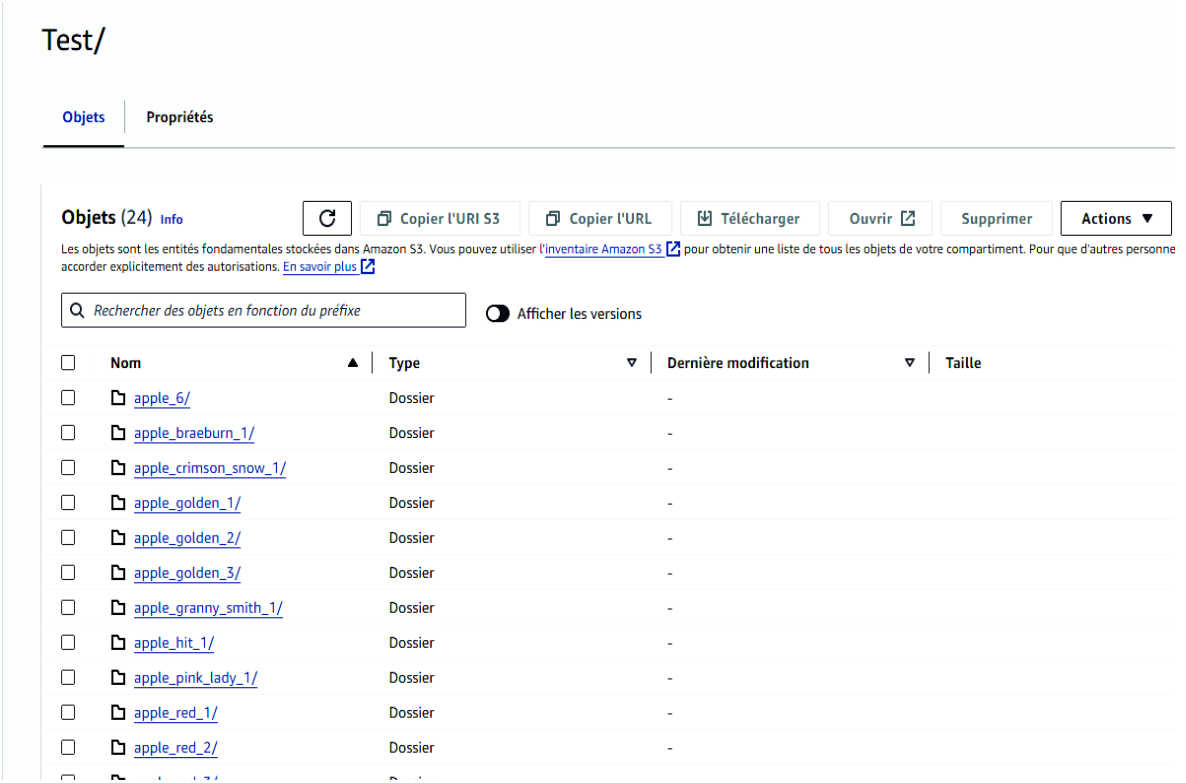
- Traitement des données en mémoire, plus rapide que Hadoop.
- Flexibilité pour le traitement batch et streaming.
- Intégration facile avec d'autres outils, y compris sur AWS (EMR).

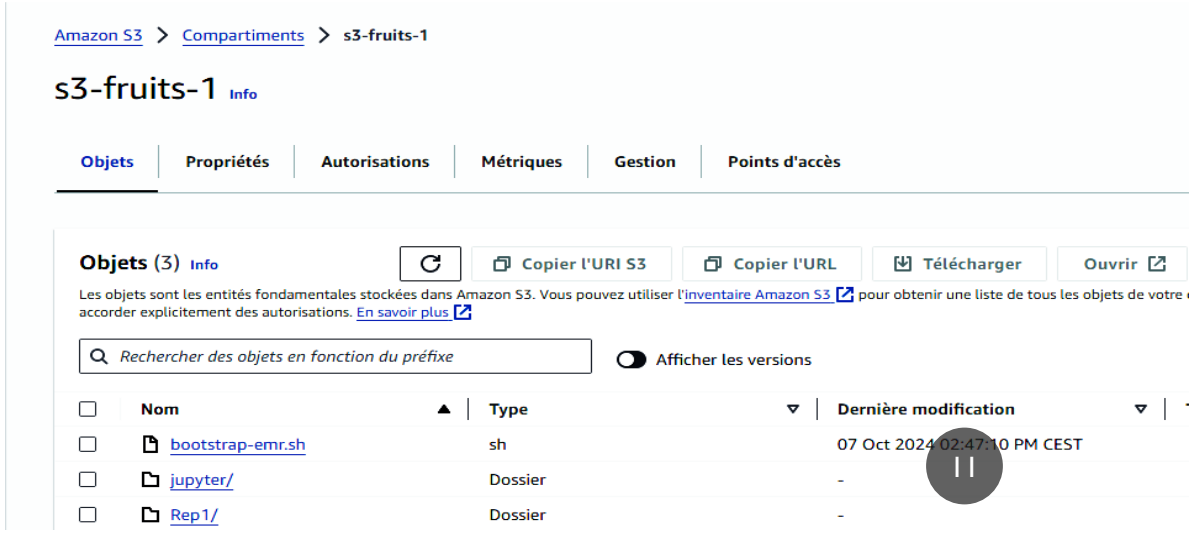
GESTION DES ACCES IAM

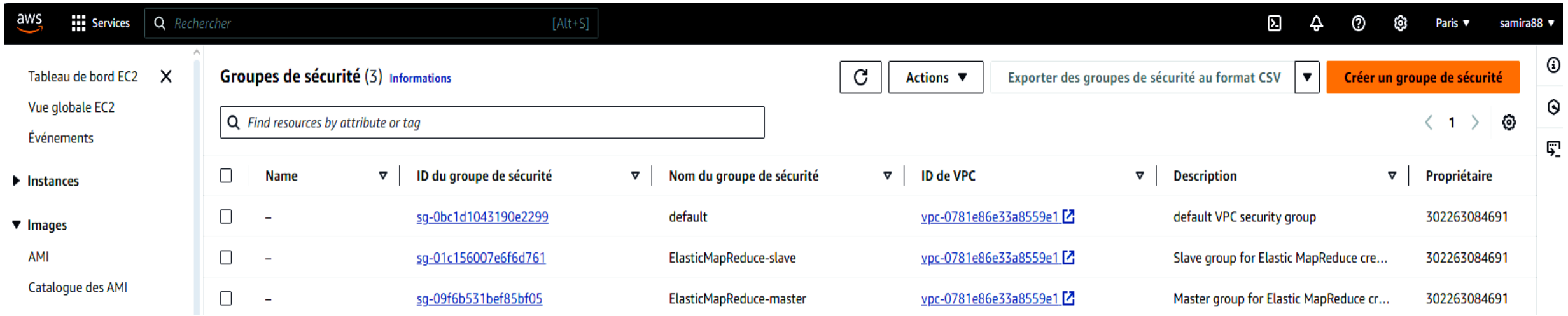


➢ ***IAM (Identity and Access Management) permet de gérer les permissions et l'accès aux ressources AWS.***
Ici, différents rôles sont attribués aux services AWS (EMR, EC2, support) pour garantir une sécurité et un contrôle d'accès spécifiques à chaque service.

STOCKAGE S3


Test/
Objets
Propriétés
Objets (24) Info
Copier l'URI S3
Copier l'URL
Télécharger
Ouvrir
Supprimer
Actions
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 personne accorder explicitement des autorisations. En savoir plus
Rechercher des objets en fonction du préfixe
Afficher les versions
Nom
Type
Dernière modification
Taille
apple_6/ Dossier -
apple_braeburn_1/ Dossier -
apple_crimson_snow_1/ Dossier -
apple_golden_1/ Dossier -
apple_golden_2/ Dossier -
apple_golden_3/ Dossier -
apple_granny_smith_1/ Dossier -
apple_hit_1/ Dossier -
apple_pink_lady_1/ Dossier -
apple_red_1/ Dossier -
apple_red_2/ Dossier -


Amazon S3 > Compartiments > s3-fruits-1
s3-fruits-1 Info
Objets
Propriétés
Autorisations
Métriques
Gestion
Points d'accès
Objets (3) Info
Copier l'URI S3
Copier l'URL
Télécharger
Ouvrir
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
accorder explicitement des autorisations. En savoir plus
Rechercher des objets en fonction du préfixe
Afficher les versions
Nom
Type
Dernière modification
bootstrap-emr.sh sh 07 Oct 2024 02:47:10 PM CEST
jupyter/ Dossier -
Rep1/ Dossier -



INSTANCE EC2

CLUSTER EMR

Amazon EMR > EMR sur EC2: Clusters > Mon-cluster-fruits

Mon-cluster-fruits

Mise à jour il y a moins d'une minute | Résilier | Cloner dans AWS CLI | Cloner

▼ **Récapitulatif**

Informations sur le cluster

ID de cluster
j-2LLPUHXBGGG5Z

Configuration de cluster
Groupes d'instances

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

Applications

Version d'Amazon EMR
emr-6.4.0

Applications installées
Hadoop 3.2.1, JupyterEnterpriseGateway 2.1.0, JupyterHub 1.4.1, Livy 0.7.1, Spark 3.1.2

Gestion des clusters

Destination des journaux dans Amazon S3
aws-logs-302263084691-eu-west-3/elasticmapreduce

Interfaces utilisateur d'application persistantes
Serveur d'historique Spark
Serveur de chronologie YARN

DNS public du nœud primaire
ec2-13-38-100-47.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
7 octobre 2024 15:42 (UTC+02:00)

Temps écoulé
2 jours, 19 heures

Interfaces utilisateur d'application sur le nœud primaire

Activer une connexion SSH

Celles-ci nécessitent l'activation du tunneling SSH.

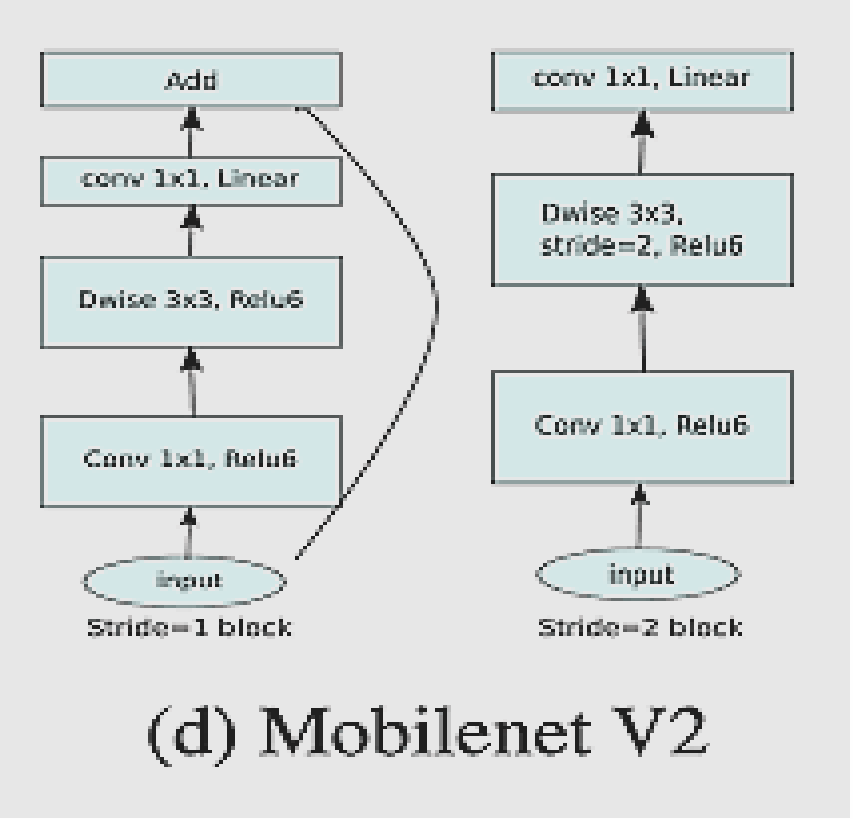
Application	URL de l'interface utilisateur
Gestionnaire de ressources	http://ec2-13-38-100-47.eu-west-3.compute.amazonaws.com:8088/
JupyterHub	https://ec2-13-38-100-47.eu-west-3.compute.amazonaws.com:9443/
Livy	http://ec2-13-38-100-47.eu-west-3.compute.amazonaws.com:8998/
Nom du nœud HDFS	http://ec2-13-38-100-47.eu-west-3.compute.amazonaws.com:9870/
Serveur d'historique Spark	http://ec2-13-38-100-47.eu-west-3.compute.amazonaws.com:18080/

CHAINE DE TRAITEMENT DES IMAGES

Etape 1 : Test du notebook en local

➢ **Exécution du notebook laissé par l'alternant sur Jupyter :**

- Définition des chemins en local
- Création d'une session **Spark**
- Construction du modèle **MobileNetV2**
- Extraction des caractéristiques (features) des images
- Enregistrement des résultats au format **Parquet**



CHAINE DE TRAITEMENT DES IMAGES

Etape 2 : Création de l'architecture sur le Cloud

Déploiement d'une instance EC2

Région : eu-west-3 (Paris) pour assurer la conformité au RGPD et optimiser la latence pour les utilisateurs européens.

Sécurité : Création et configuration d'une paire de clés SSH au format .ppk pour des connexions sécurisées.

Configuration du stockage S3

Création du Bucket : "nom_du_bucket" pour stocker les données liées au projet de classification des fruits.

Téléchargement des données : Upload du dossier **"Test"** contenant les images du dataset à analyser.

Permissions : Définition des règles d'accès pour sécuriser les données et limiter les accès.

Avantages de cette architecture

Scalabilité : Les ressources peuvent facilement être ajustées selon les besoins.

Sécurité : Respect strict des réglementations européennes, avec des mesures de sécurité renforcées.

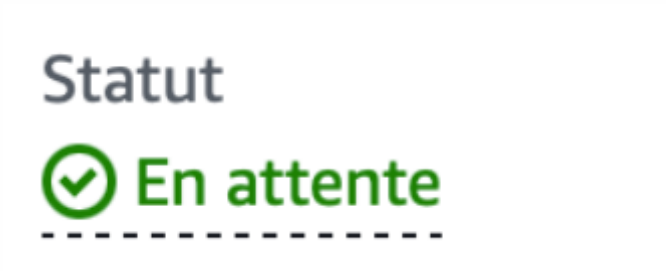
CHAINE DE TRAITEMENT DES IMAGES

Etape 3 : Création de l'architecture sur le Cloud

•Paramétrage du cluster :

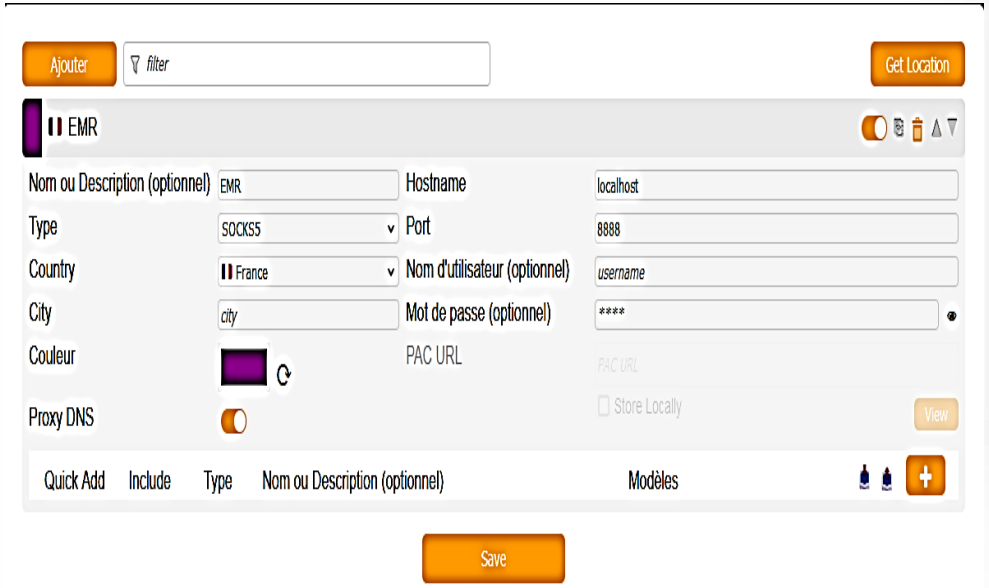
- **Version :** emr-6.4.0
- **Applications :** Hadoop 3.2,1, JupyterHub 1.4.1, Spark 3.1.2
- **Configuration JSON :** Enregistrement des configurations sur S3
- **Script d'amorçage :** Script Shell uploadé sur S3 pour automatiser les configurations initiales
- **Sécurité :** Paire de clé SSH pour l'accès sécurisé à l'instance EC2
- **Rôles IAM :** Création de rôles pour gérer les accès à EC2, EMR, et S3

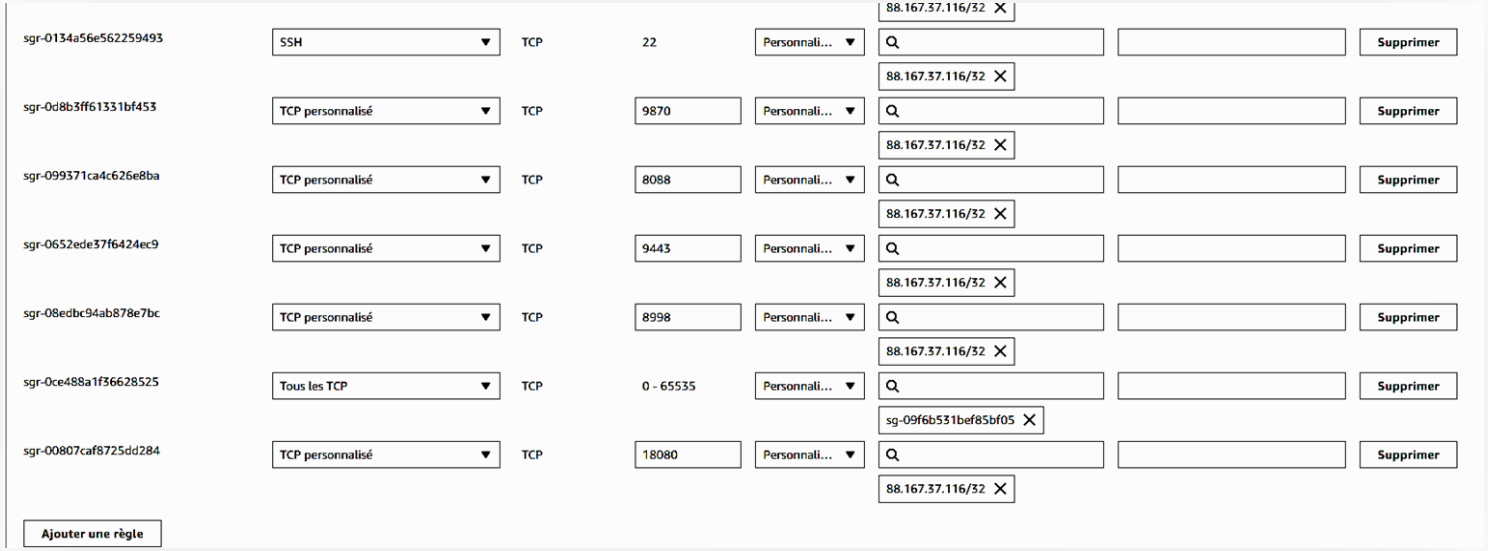
```
#!/bin/bash

python3 -m pip install -U setuptools
python3 -m pip install -U pip
python3 -m pip install wheel
python3 -m pip install matplotlib seaborn
python3 -m pip install pandas
python3 -m pip install pyspark
python3 -m pip install boto3
python3 -m pip install s3fs
python3 -m pip install jupyterlab
```



CHAINE DE TRAITEMENT DES IMAGES

Etape 4 : Connexion sécurisée à JupyterHub via SSH et Proxy SOCKS





1.Configuration du groupe de sécurité AWS :

- **Port 22 (SSH)** : Ouvrir pour permettre la connexion SSH.
- **Port personnalisé (9443)** : Pour accéder à JupyterHub

2. Connexion SSH avec Tunnel SOCKS
3. Configuration du Proxy SOCKS dans le navigateur

- **Serveur** : localhost
- **Port** : 8888

EXÉCUTION DU SCRIPT PY SPARK

2. Chargement des images et enregistrement des résultats

[3]:
```
# Chemins d'accès à Amazon S3
PATH = 's3://s3-fruits-1/Rep1/'
PATH_Data = PATH + '/Test'
PATH_Results = PATH + '/Results'

# Afficher les chemins
print('PATH:         ' + PATH)
print('PATH_Data:    ' + PATH_Data)
print('PATH_Results: ' + PATH_Results)
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
42	application_1728309150675_0043	pyspark	idle	Link	Link	None	✓

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…

SparkSession available as 'spark'.

FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…

PATH:         s3://s3-fruits-1/Rep1/
PATH_Data:    s3://s3-fruits-1/Rep1//Test
PATH_Results: s3://s3-fruits-1/Rep1//Results
```

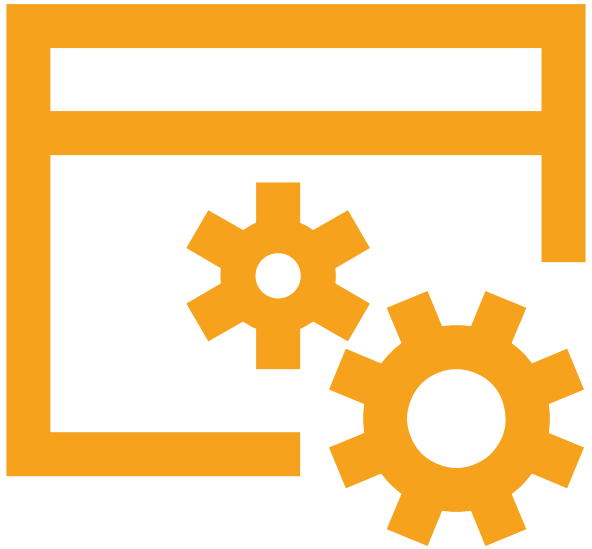
2.2 Ajout du label des images et sélection de colonnes

Entrée [6]:
```
images = images.withColumn('label', element_at(split(images['path'], '/'),-2))
print(images.printSchema())
print(images.select('path','label').show(5,False))
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),…

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://s3-fruits-1/Rep1/Test/apple_hit_1/r0_115.jpg|apple_hit_1|
|s3://s3-fruits-1/Rep1/Test/apple_hit_1/r0_119.jpg|apple_hit_1|
|s3://s3-fruits-1/Rep1/Test/apple_hit_1/r0_107.jpg|apple_hit_1|
|s3://s3-fruits-1/Rep1/Test/apple_hit_1/r0_143.jpg|apple_hit_1|
|s3://s3-fruits-1/Rep1/Test/apple_hit_1/r0_111.jpg|apple_hit_1|
+----------------------------------------------+----------+
only showing top 5 rows

None
```

SUIVI DES JOBS SPARK

Aperçu de la chaîne entière de traitement avec SparkUI



LIEN EMR AWS (SERVER SUR PARIS)

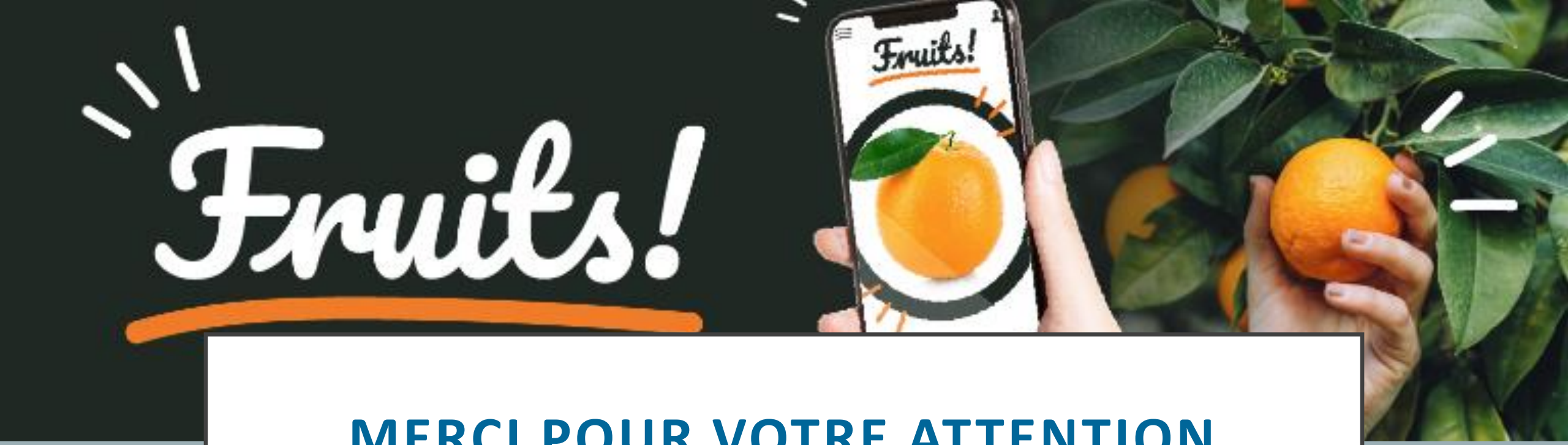
https://eu-west-3.console.aws.amazon.com/emr/home?region=eu-west-3#/clusters

CONCLUSION

- Utilisation des services AWS et Apache Spark pour le traitement de données massives, avec une architecture scalable.
- Stockage des données sur S3 et exécution des tâches distribuées sur EMR, garantissant performance et flexibilité.
- Application de l'ACP (Analyse en Composantes Principales) pour une réduction dimensionnelle efficace.
- Surveillance des tâches et optimisation des ressources via SparkUI, permettant une visualisation détaillée des processus distribués.

➢ **Perspectives d'amélioration:**

- Exploration de modèles de classification avancés (SVM, Random Forest) pour améliorer les performances.
- Enrichissement des visualisations avec des techniques comme le t-SNE en 2D ou 3D.
- Optimisation des coûts et des ressources sur AWS tout en assurant la stabilité avec la gestion des versions logicielles.



MERCI POUR VOTRE ATTENTION