Fruits!

Déployer un modèle dans le cloud

OC DS P8 / Massimo Bruni

Rappel de la problématique

Architecture de principe

Configuration AWS

Traitement de données

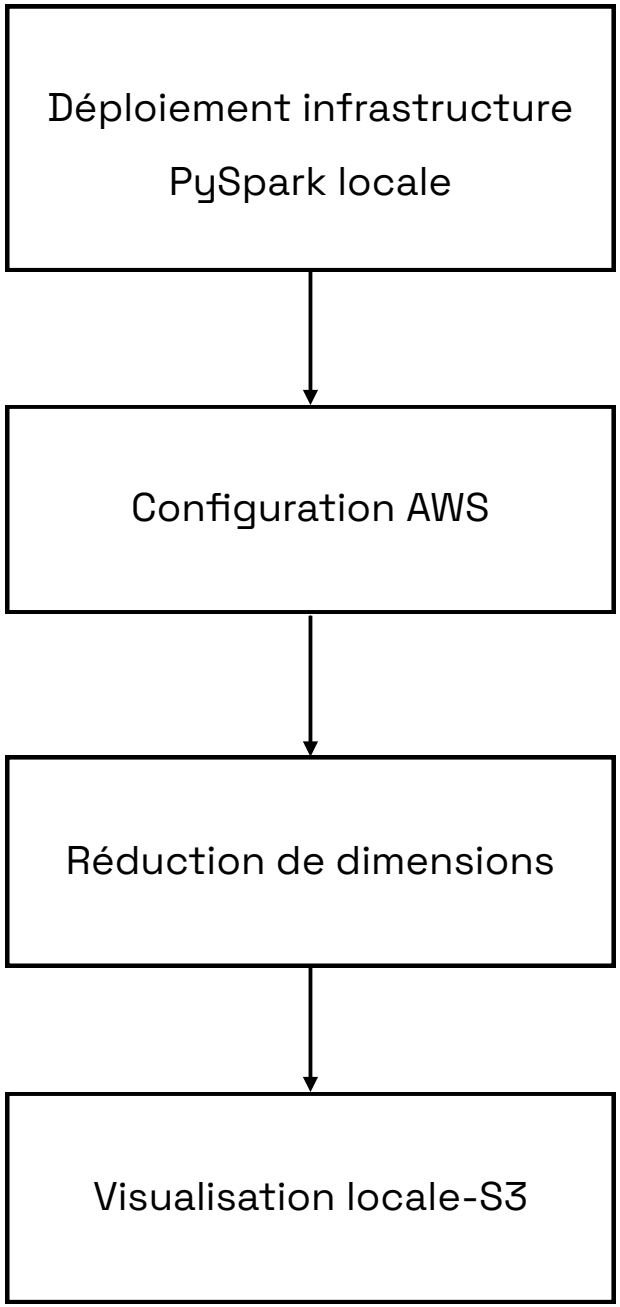
Perspectives d’amélioration

Rappel de la problématique

Fruits est une start-up de l'Agri-Tech qui souhaite proposer une application mobile qui permettrait à ses utilisateurs et utilisatrices de prendre une photo d'un fruit pour obtenir des informations liées a ce fruit.

On reprend les travaux d'un alternant qui a commencé la mise en place d'une architecture scalable sur AWS.

On doit s'assurer de pouvoir remonter une architecture similaire sur la base de ces documents, puis mettre en oeuvre une ACP distribuée sur les images vectorisées.



Développer et tester des scripts hors d'AWS

- ➕ coût nul
- ➖ ressources limitées

Mettre en place un environnement de production grande disponibilité et sécurisé

- ➕ ressources importantes
- ➖ facturation *uptime*

Réaliser un modèle de réduction de dimension sur AWS

- ➕ calculs distribués
- ➖ facturation *uptime*

Connecter l'IDE local a S3 pour visualisation

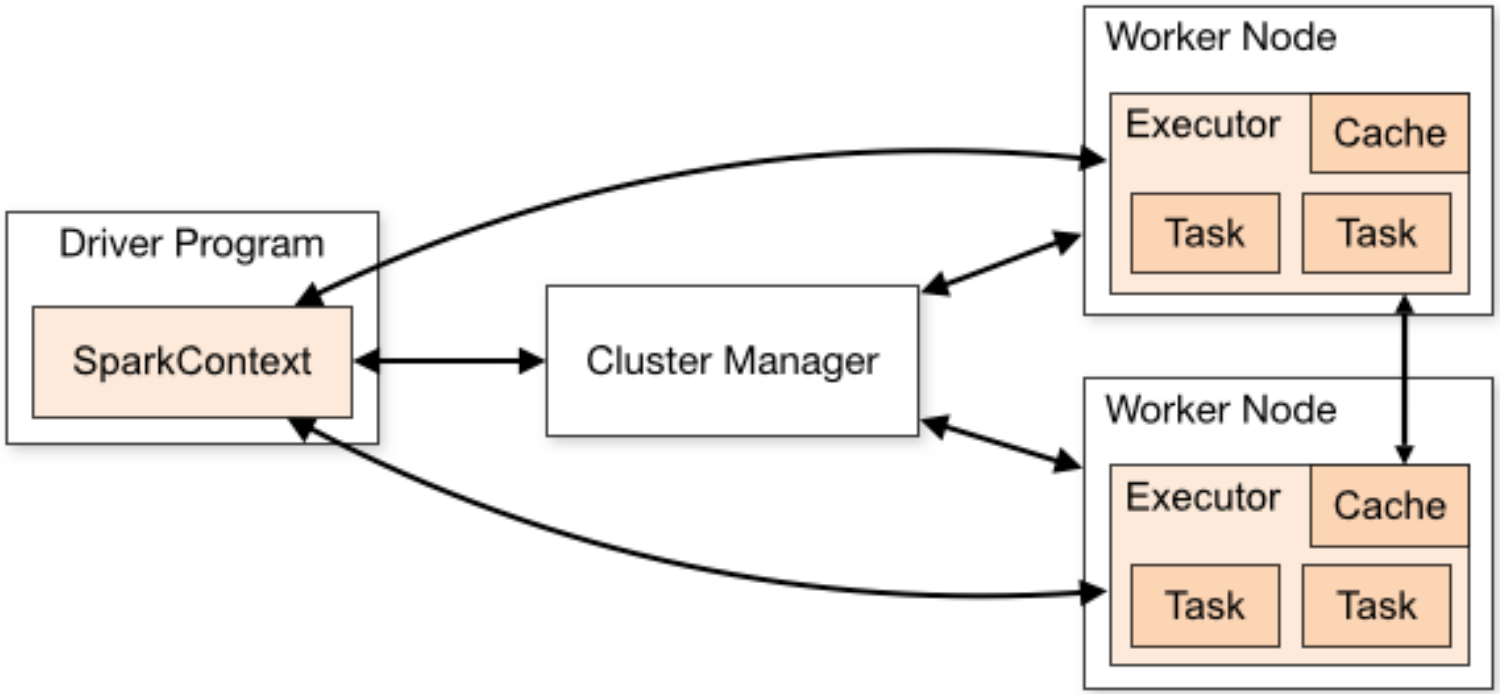
- ➕ Environnement maîtrisé
- ➖ configuration IDE-AWS

Architecture de principe

PaaS scalable - Stockage

High Availability
Distributed File
System

PaaS scalable - Calcul distribué



1. Le maître va scinder le code en étapes à exécuter, chaque étape étant constituée de tâches.

2. Le Cluster Manager suit le bon déroulement des étapes auprès des esclaves et du maître.

3. Le stockage des données sert la mise à disposition des données d'entrées et la sauvegarde des résultats.

Mise en oeuvre d’une infrastructure scalable

S3 EMR IAM

Configuration AWS

1. Identity and Access Management (IAM)

Sécurisation des accès et authentification

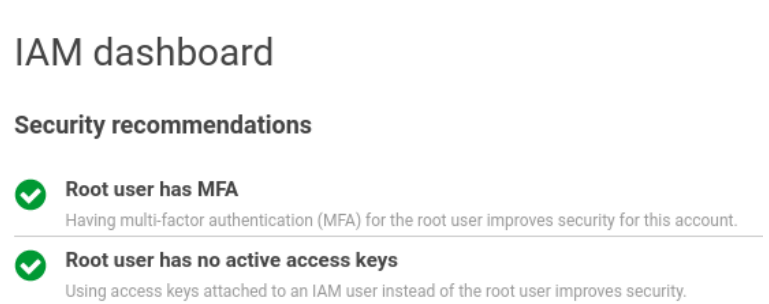
(users- groups - policies)

Se créer un compte sur le site AWS et activer le MFA

Choisir eu-west-3 (Paris) comme région par défaut

Créer un utilisateur non-root autorisé à gérer les espaces S3 et EMR

Configurer une clé d'accès pour le nouvel utilisateur et l'utiliser pour AWS CLI local



Configuration AWS

2. Simple Storage Service (S3)

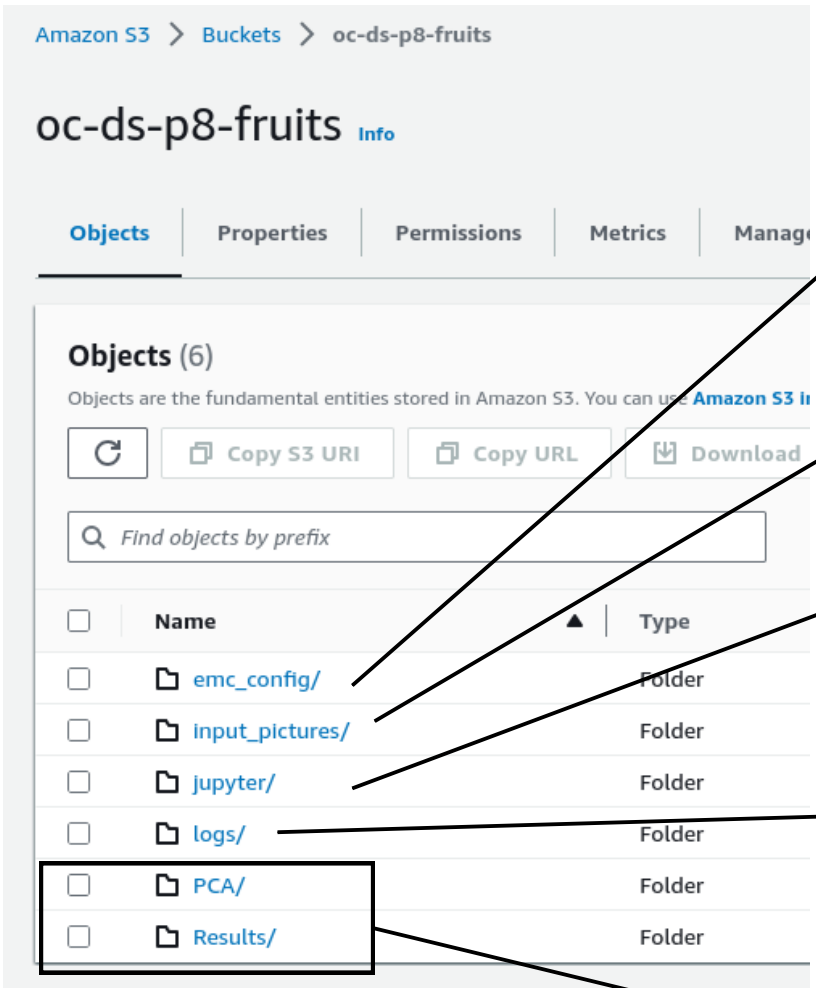
Assure la persistence de nos données et leur accès non localisé

(inputs - outputs- notebooks)

EU (Paris)		USD 0.20
Amazon Simple Storage Service EUW3-Requests-Tier1		USD 0.15
$0.00 per request - PUT, COPY, POST, or LIST requests under the monthly global free tier	2,000 Requests	USD 0.00
$0.0053 per 1,000 PUT, COPY, POST, or LIST requests	27,867 Requests	USD 0.15
Amazon Simple Storage Service EUW3-Requests-Tier2		USD 0.05
$0.00 per request - GET and all other requests under the monthly global free tier	20,000 Requests	USD 0.00
$0.0042 per 10,000 GET and all other requests	120,703 Requests	USD 0.05
Amazon Simple Storage Service EUW3-TimedStorage-ByteHrs		USD 0.00
$0.000 per GB - storage under the monthly global free tier	0.091 GB-Mo	USD 0.00
US East (N. Virginia)		USD 0.00

→ Paiement en fonction du nombre de fichiers déposés par mois.

→ Paiement en fonction du nombre de requêtes sur les fichiers stockés



Configuration VM (bootstrap)

input data

IDE - notebooks

EMR logs

models' outputs

Configuration AWS

3. Elastic Map Reduce (EMR)

Mise à disposition de plateformes de calculs distribués

Description	Usage Quantity	
Elastic Compute Cloud		**USD 16.12**
EU (Paris)		USD 14.94
Amazon Elastic Compute Cloud running Linux/UNIX		USD 14.94
$0.224 per On Demand Linux m5.xlarge Instance Hour	66.689 Hrs	USD 14.94
EBS		USD 0.00
$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier	7.011 GB-Mo	USD 0.00
US East (N. Virginia)		USD 1.18
Amazon Elastic Compute Cloud running Linux/UNIX		USD 1.18
$0.192 per On Demand Linux m5.xlarge Instance Hour	6.16 Hrs	USD 1.18
EBS		USD 0.00
$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier	0.629 GB-Mo	USD 0.00
Elastic MapReduce		**USD 3.19**
EU (Paris)		USD 2.94
Amazon Elastic MapReduce EUW3-BoxUsage:m5.xlarge		USD 2.94
$0.048 per hour for EMR m5.xlarge	61.232 Hrs	USD 2.94
US East (N. Virginia)		USD 0.25

Facturation dépendante de la puissance demandée et de la durée durant laquelle les instance sont Up.

Amazon EMR release | Info

A release contains a set of applications which can be installed on your cluster.

emr-6.3.0

Application bundle

Spark | Core Hadoop | HBase | Presto | PrestoSQL | Custom

Customize your application bundle

non necessaire

Applications included in bundle

- [ ] Flink 1.12.1
- [ ] HCatalog 3.1.2
- [ ] Hue 4.9.0
- [ ] Livy 0.7.0
- [ ] Phoenix 5.0.0
- [ ] PrestoSQL 350
- [x] TensorFlow 2.4.1
- [ ] ZooKeeper 3.4.14
- [ ] Ganglia 3.7.2
- [x] Hadoop 3.2.1
- [ ] JupyterEnterpriseGateway 2.1.0
- [ ] MXNet 1.7.0
- [ ] Pig 0.17.0
- [x] Spark 3.1.1
- [ ] Tez 0.9.2
- [ ] HBase 2.2.6
- [ ] Hive 3.1.2
- [x] JupyterHub 1.2.0
- [ ] Oozie 5.2.1
- [ ] Presto 0.245.1
- [ ] Sqoop 1.4.7
- [ ] Zeppelin 0.9.0

m5.xlarge

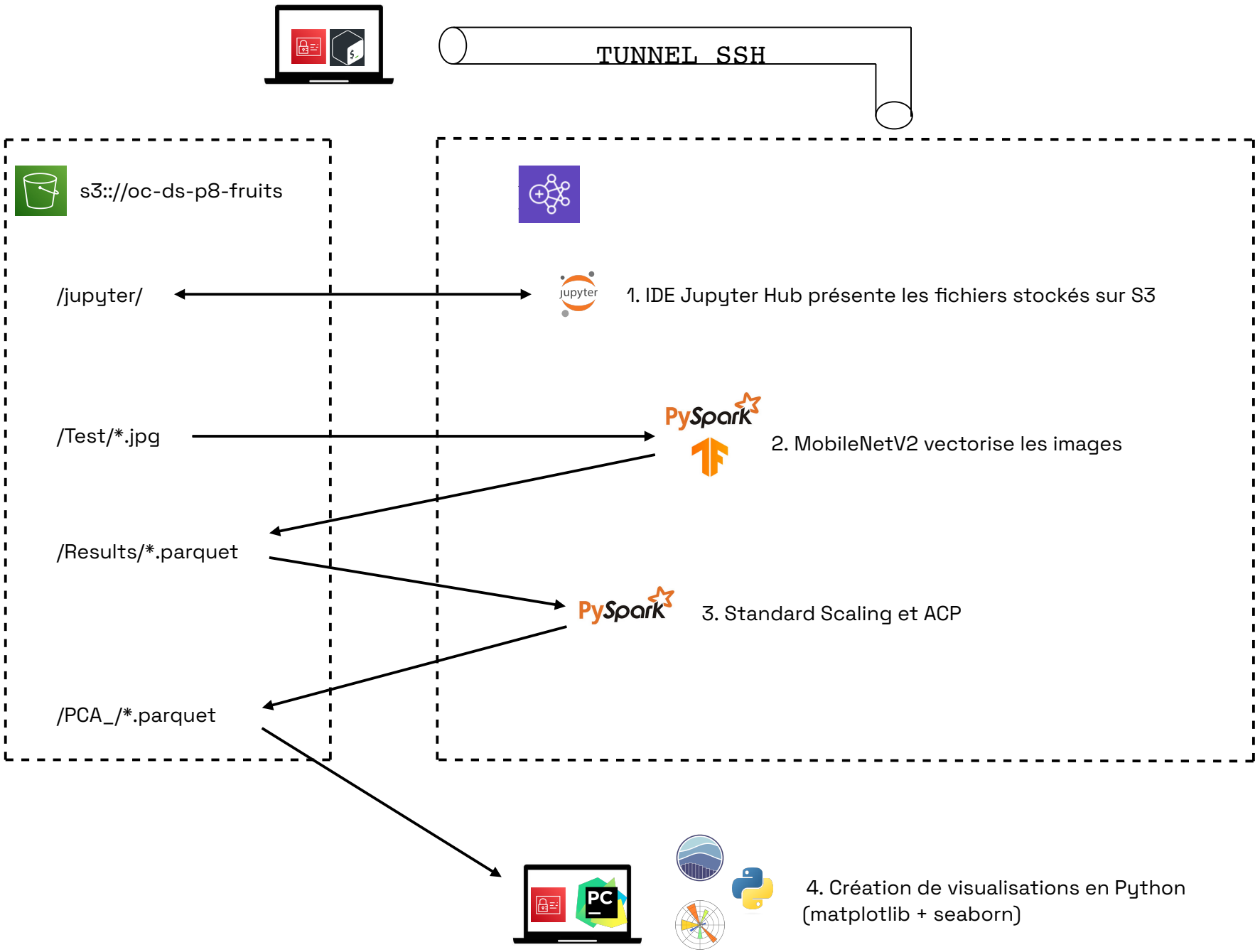
4 vCore 16 GiB memory EBS only storage

On-Demand price: $0.224 per instance/hour

Lowest Spot price: $0.064 (eu-west-3b)

Name	Instance type	Instance(s) size
Core	m5.xlarge	1
Task - 1	m5.xlarge	2

Architecture du projet

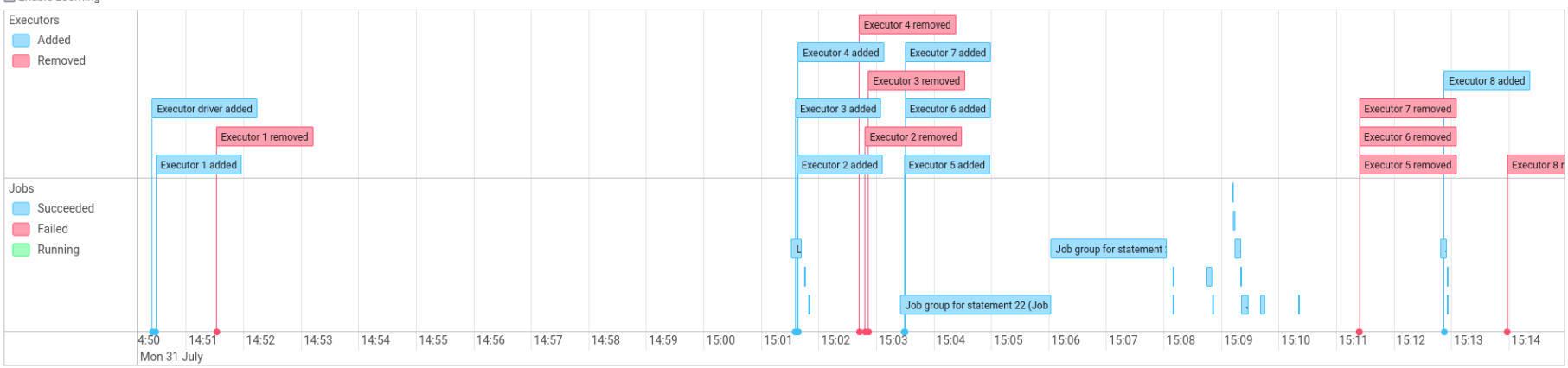

TUNNEL SSH
s3://oc-ds-p8-fruits
/jupyter/
1. IDE Jupyter Hub présente les fichiers stockés sur S3
/Test/*.jpg
2. MobileNetV2 vectorise les images
/Results/*.parquet
3. Standard Scaling et ACP
/PCA_/*.parquet
4. Création de visualisations en Python
(matplotlib + seaborn)

Spark Jobs (?)

User: livy
Total Uptime: 30 min
Scheduling Mode: FIFO
Completed Jobs: 19

▾ Event Timeline

☐ Enable zooming



Traîtement des données

Table 1. MobileNet Body Architecture

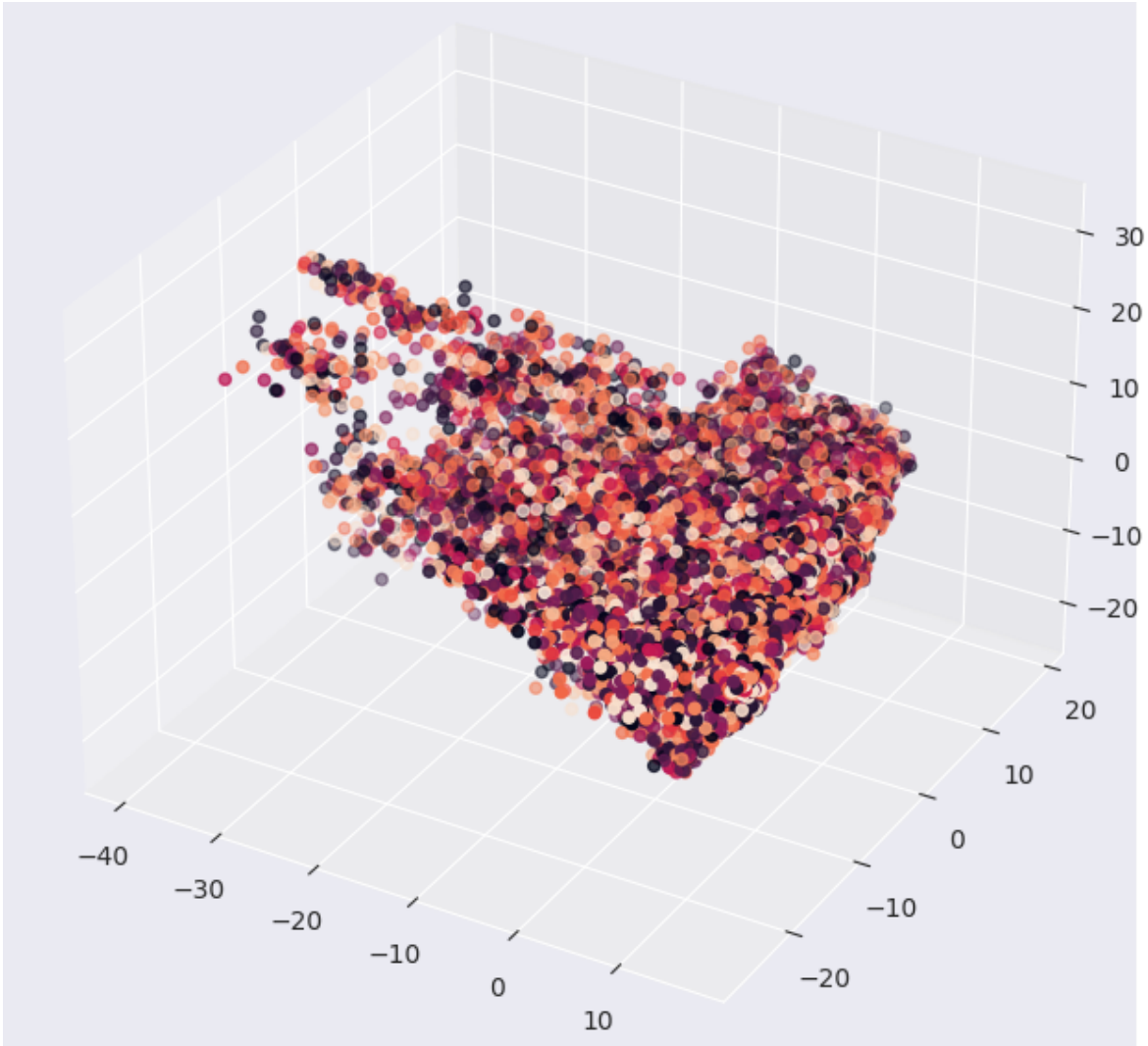
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
~~Softmax / s1~~	~~Classifier~~	~~$1 \times 1 \times 1000$~~

Network Architecture: MobileNet

”La V2 des séries MobileNet introduit les résiduels inversés (inverted residuals) et les goulots d’étranglement linéaires (linear bottlenecks) pour améliorer les performances de MobileNets.” (source)

Retrait de la couche softmax

”La dernière couche softmax est remplacée par notre catégorisation spécifique.”



Visualisation des points dans le référentiel des trois premières composantes principales (coloration par catégorie de fruits)

Question

L'ACP peut servir plusieurs objectifs : elle peut permettre de réduire le nombre de dimensions à quelques composantes principales ; et elle peut faciliter l'explicabilité d'un phénomène par interprétation des plans factoriels et visualisation.

Or,

La vectorisation des images et la détection de features constitue déjà une réduction de dimension.

L'interprétation des plans factoriels en sortie d'un réseau de neurones convolutifs nécessite un compréhension profonde dudit réseau de neurones.

Quel est l'objectif du travail réalisé ? Trouver la quantité minimale d'information qu'on pourrait stocker pour réduire nos coûts ?

Perspectives d’amelioration

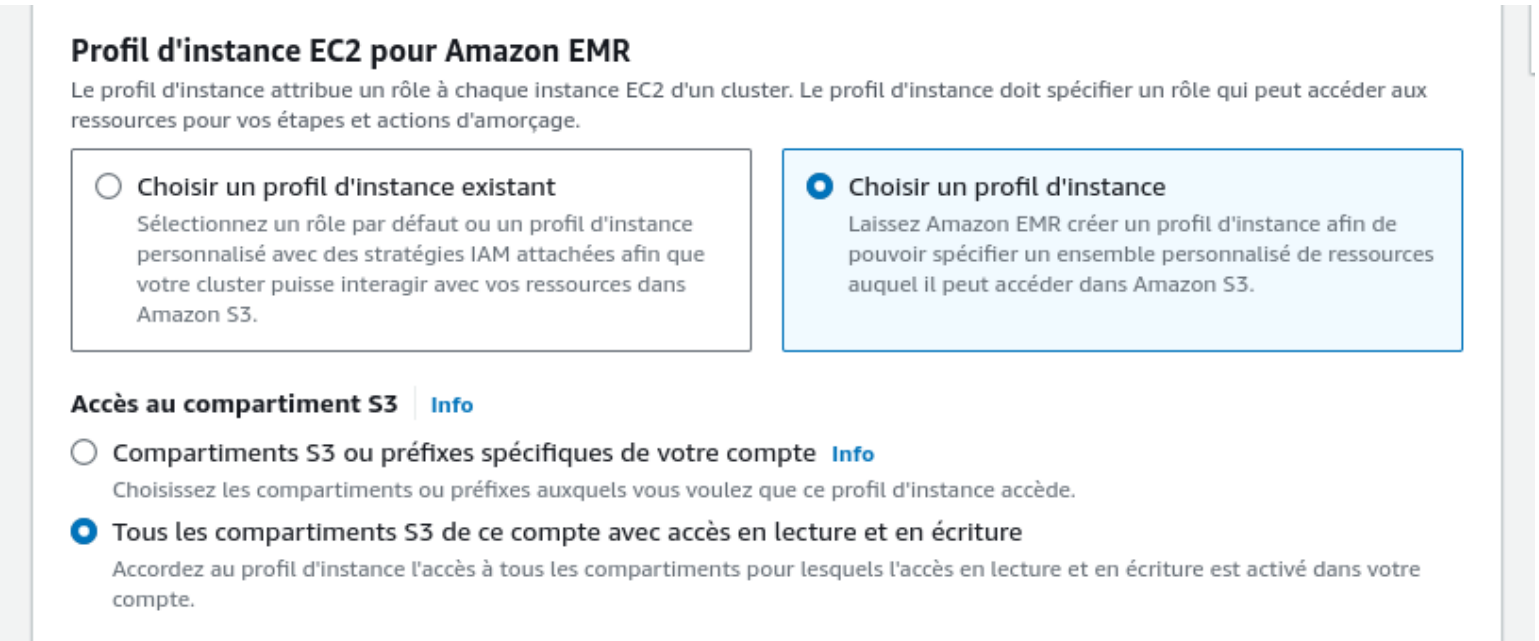
En l'état, on ne traîte aucune donnée à caractère personnel, il n'est donc pas nécessaire de tout stocker en Europe.

Déporter la vectorisation des images sur les terminaux clients (*edge computing*) pourrait offrir des économies de stockage et serait cohérent avec MobileNetV2.

Avant de penser architecture de scalabilité pour le projet : Identifier les sources de données qu'on aimerait agréger, puis concevoir et tester un modèle.

Ma politique de gestion des groupes de sécurité entre EMR et S3 pourrait-être améliorée

Actuellement j’ai configuré les accès au plus simple :



Mais la sécurisation de l’infrastructure nécessiterait une révision de cette configuration.

Merci !