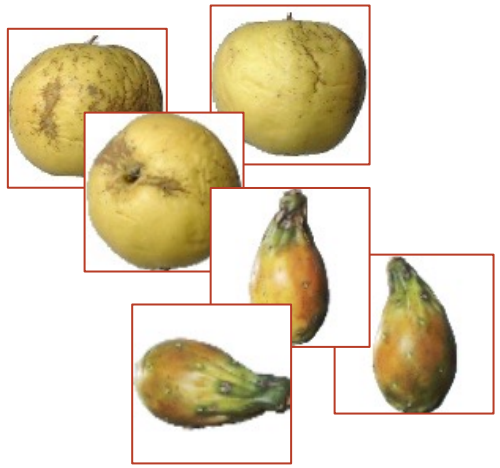
Fruits!

Préparation à la reconnaissance d'image

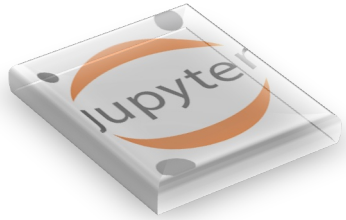
Déploiement du modèle dans le cloud

Introduction

- Objectifs :

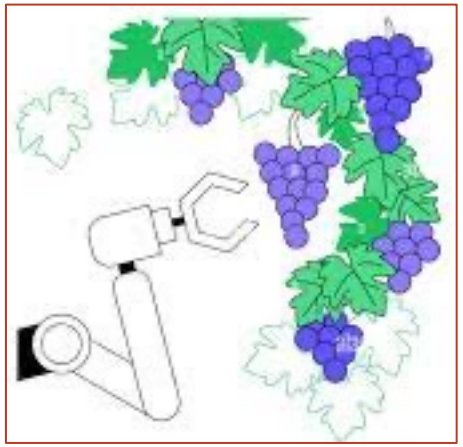


Base initiale



État initial

Accroissement de la base

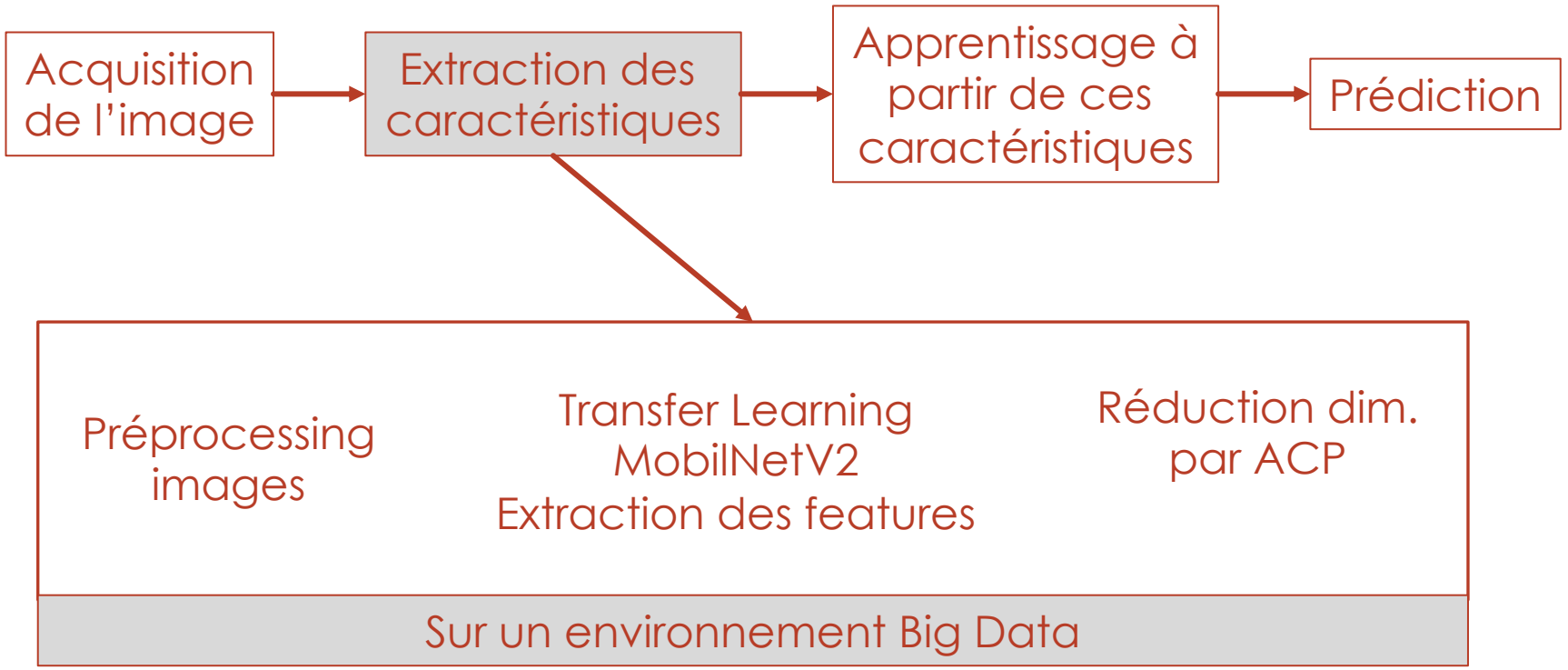


Utilisation finale

Fruits!

Introduction

- Reconnaissance d'image :



Fruits!

Environnement Big Data

- Briques de l'environnement Big Data sur **Amazon Web Service**

- **IAM** : Gestion des utilisateurs et de leurs droits

- **EC2** : Instanciation des serveurs virtuels pour le calcul
  Création de la paire de clé rsa pour le tunnel SSH
  Ouverture du port 22 vers l'extérieur pour le tunnel SSH (groupe de sécurité ElasticMapReduce-master)

- **S3** : Création du bucket elisepoupi-p8
  Stockage du Notebook, du fichier de bootstrapping et des données

- **EMR** : Création du cluster pour les calculs distribués par PySpark sur des serveurs EC2
  Accès à JupyterHub par le tunnel SSH vers EC2 pour lancer les calculs du Notebook stocké sur S3

Fruits!

Environnement Big Data

- **IAM :** Deux utilisateurs créés

Services	Elise	Lecteur
S3	Contrôle total	Lecture seule
EC2	Contrôle total	Lecture seule
EMR	Contrôle total	Lecture seule
IAM	Contrôle total	

- **Lecteur_credentials** : identifiants pour la connexion à la console

Nom d'utilisateur	Mot de passe	URL de connexion
Lecteur	]%1m*A5$	https://105083061881.signin.aws.amazon.com/console

Fruits!

Environnement Big Data

- **EC2** :

EC2 > Groupes de sécurité > sg-0045d4508eb4efe04 - ElasticMapReduce-master

sg-0045d4508eb4efe04 - ElasticMapReduce-master

Règles entrantes (9) | Gérer les balises | Modifier les règles entrantes

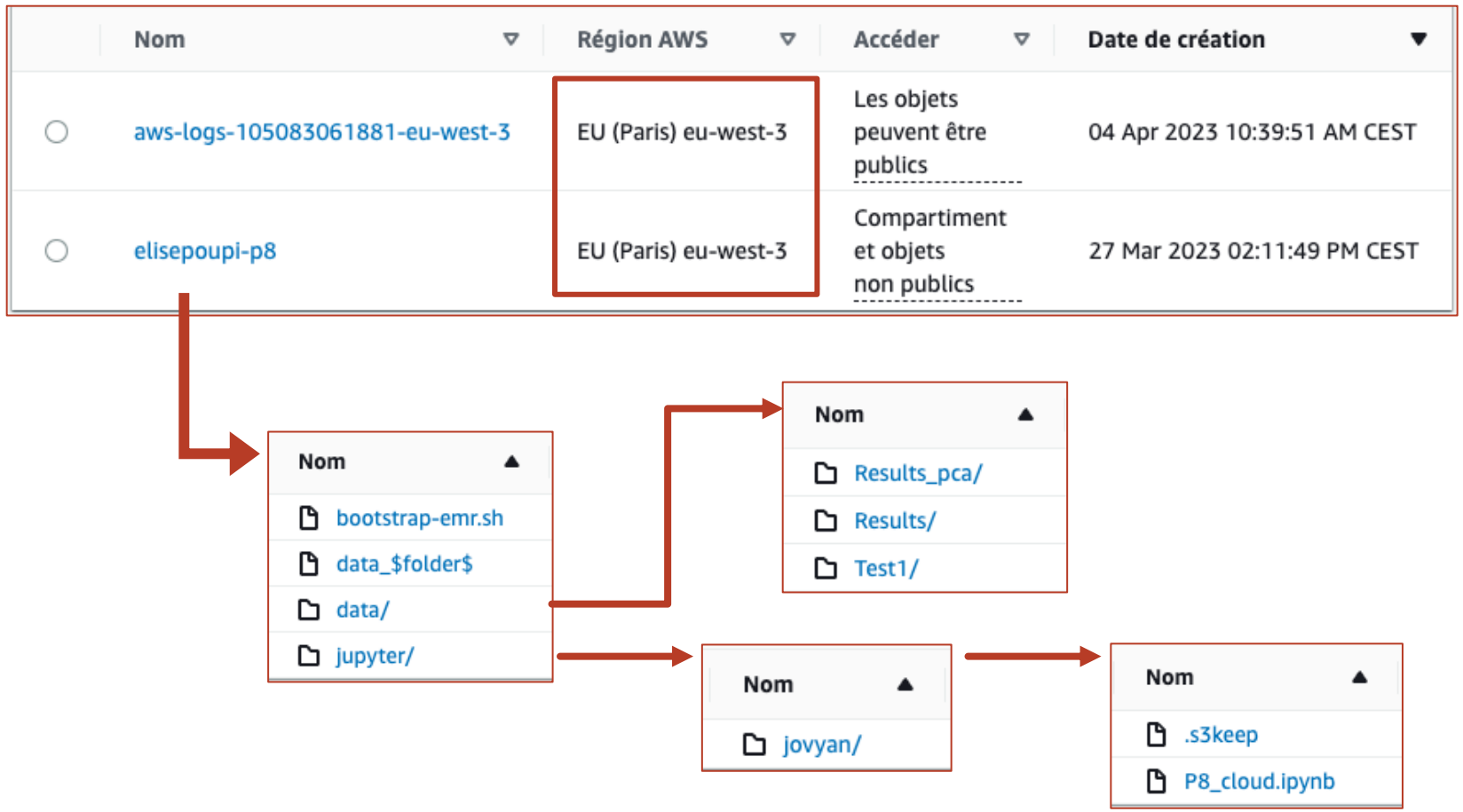
Filtrer les règles des groupes de sécurité | 1

Version IP	Type	Protocole	Plage de ports	Source
–	Tous les UDP	UDP	0 - 65535	sg-001d8fa3410f320a...
IPv4	SSH	TCP	22	0.0.0.0/0
–	Tous les ICMP - IPv4	ICMP	Tous	sg-0045d4508eb4efe0...
–	Tous les TCP	TCP	0 - 65535	sg-0045d4508eb4efe0...
–	Tous les TCP	TCP	0 - 65535	sg-001d8fa3410f320a...
IPv4	TCP personnalisé	TCP	8443	52.46.82.0/23
IPv6	SSH	TCP	22	::/0
–	Tous les ICMP - IPv4	ICMP	Tous	sg-001d8fa3410f320a...
–	Tous les UDP	UDP	0 - 65535	sg-0045d4508eb4efe0...

Fruits!

Environnement Big Data

- **S3** :



	Nom	Région AWS	Accéder	Date de création
○	aws-logs-105083061881-eu-west-3	EU (Paris) eu-west-3	Les objets peuvent être publics	04 Apr 2023 10:39:51 AM CEST
○	elisepoupi-p8	EU (Paris) eu-west-3	Compartiment et objets non publics	27 Mar 2023 02:11:49 PM CEST

Fruits!

Environnement Big Data

- **EMR** :

Détails de configuration

Étiquette de version : emr-6.9.0
Distribution Hadoop : Amazon 3.3.3
Applications : JupyterHub 1.4.1, Spark 3.3.0
URI de connexion : s3://aws-logs-105083061881-eu-west-3/elasticmapreduce/
Vue cohérente EMRFS : Désactivé
ID d'AMI personnalisée : --
Version d'Amazon Linux : 2.0.20230404.0 En savoir plus

Retrait de Tensorflow suite bug et insertion de Tensorflow dans le fichier bootstrap-emr.sh

Réseau et matériel

Zone de disponibilité eu-west-3a
ID de sous-réseau (subnet) : subnet-03c843f28b79cc8d7
Maître : Action d'amorçage 1 m5.xlarge
Principal : Action d'amorçage 2 m5.xlarge
Tâche : --
Cluster scaling: Custom policy
Résiliation automatique : Arrêter en cas d'inactivité pour 1 heure, 30 minutes

m5 : adapté pour les charges de travail EMR
m5.xlarge : 4 vCPU – mémoire 16 Gio

Sécurité et accès

Nom de clé : elise-ec2
Profil d'instance EC2 : EMR_EC2_DefaultRole
Rôle EMR : EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole

Fruits!

Environnement Big Data

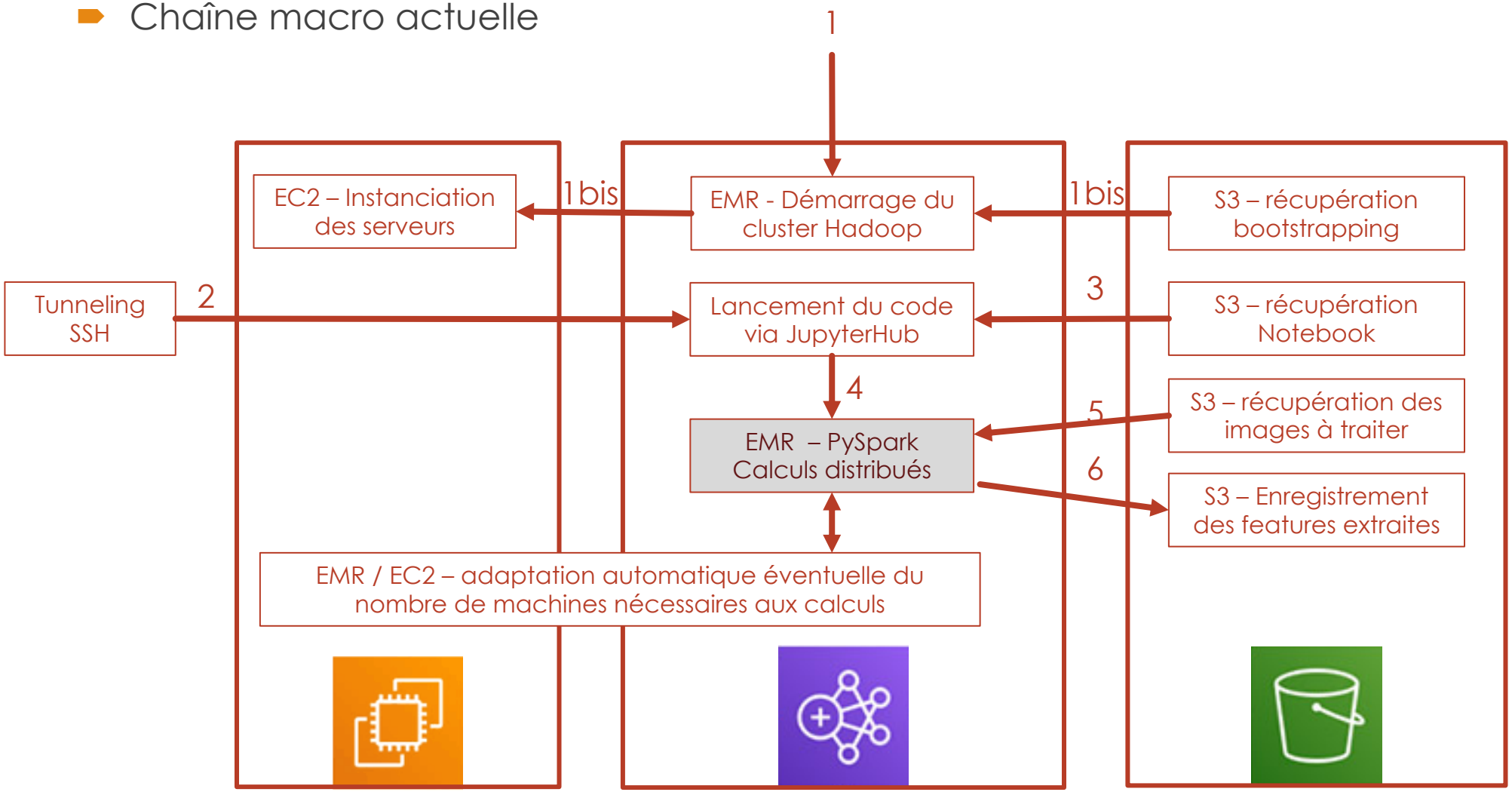
- **S3 et EMR** : choix de la région AWS Europe (Paris)
  - Latence réduite au maximum pour les utilisateurs français
  - Données conservées en France, région conforme aux normes ISO 27001, ISO 27017, ISO 27018, SOC 1 (SAS 70), SOC2 et SOC3 en matière de sécurité et disponibilité, PCI DSS niveau 1

Région	Coût de l'instance m5.xlarge
Europe (Stockholm) eu-north-1	0,204 USD/h -8,9%
Europe (Irlande) eu-west-1	0,214 USD/h -4,5%
Europe (Espagne) eu-south-2	0,214 USD/h -4,5%
Europe (Londres) eu-west-2	0,222 USD/h -0,9%
Europe (Paris) eu-west-3	0,224 USD/h -
Europe (Milan) eu-south-1	0,224 USD/h -
Europe (Francfort) eu-central-1	0,230 USD/h +2,7%
Europe (Zurich) eu-central-2	0,253 USD/h +12,9%

Fruits!

De l'environnement au traitement

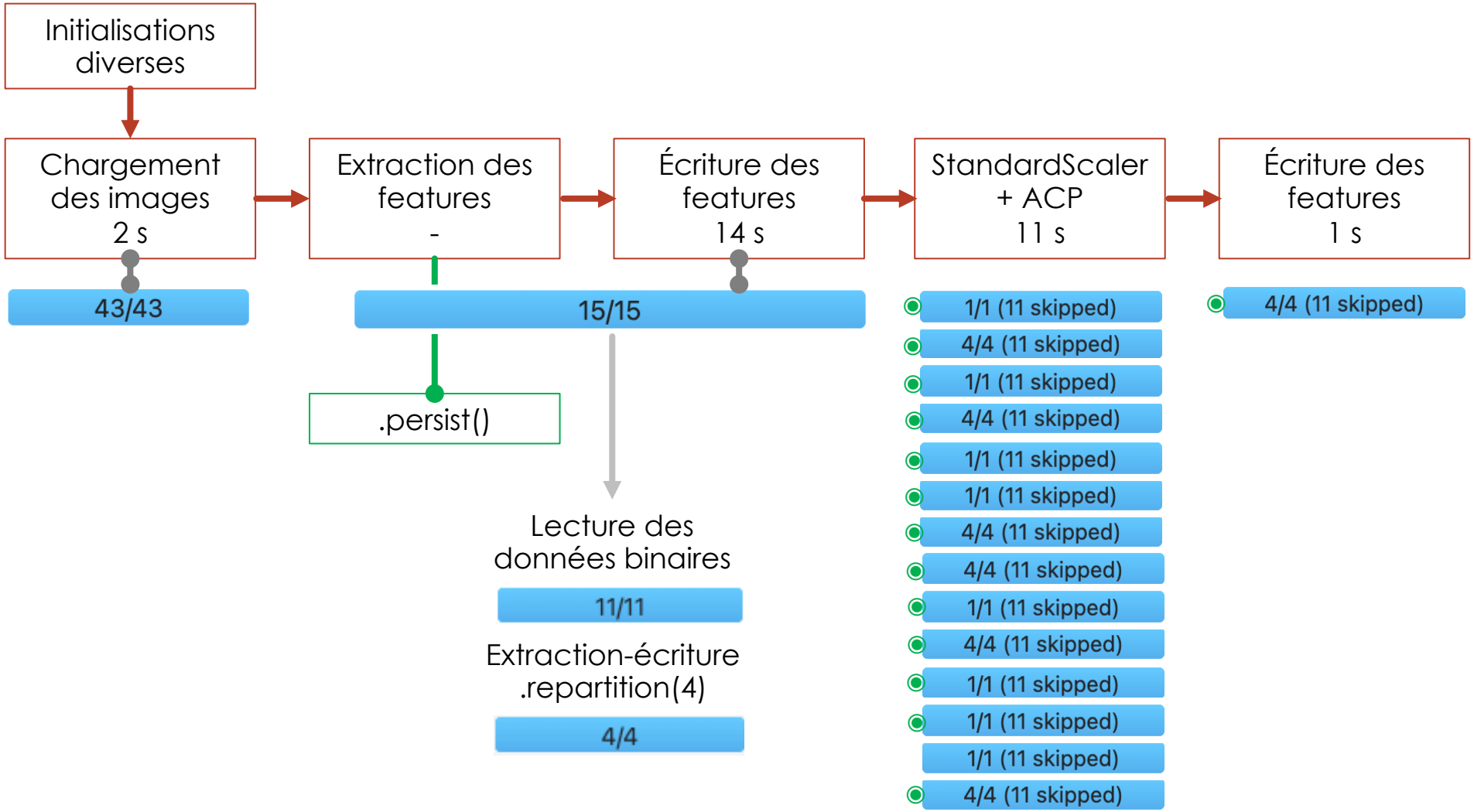
- Chaîne macro actuelle



Fruits!

Chaîne de traitement des images

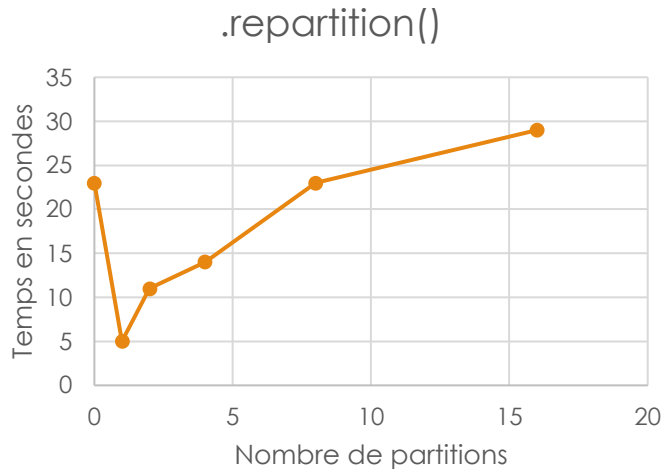
- Chaine de traitement dans le code



Fruits!

Chaîne de traitement des images

- Focus sur .repartition(n)
  - Opération dite par PySpark coûteuse en ressources et à éviter dans le cadre de l'optimisation des performances



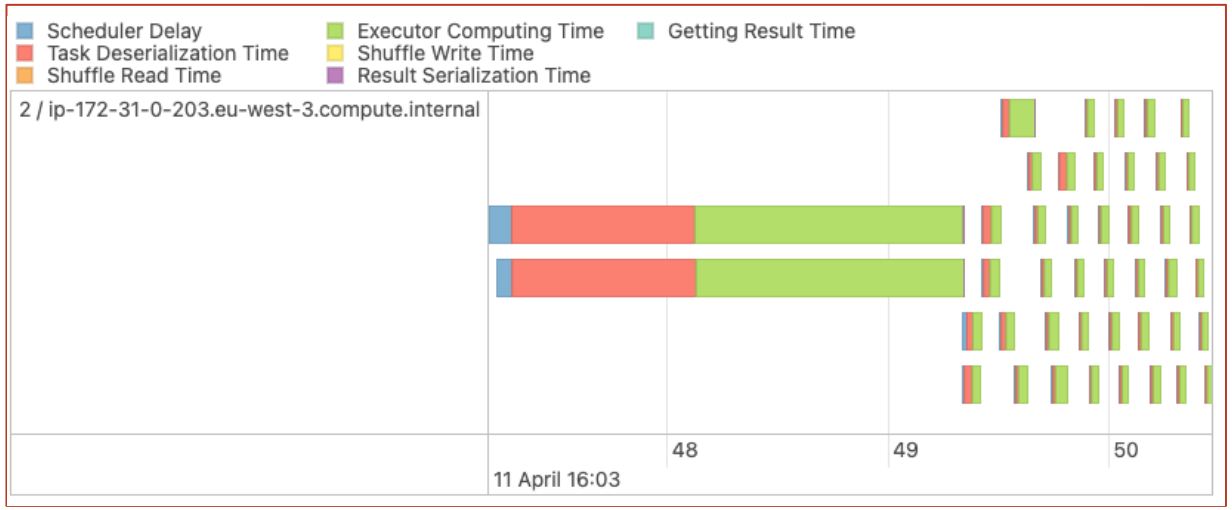
  - Ici elle reste intéressante avec peu de données, mais à surveiller par la suite et voir s'il faut laisser PySpark gérer le découpage

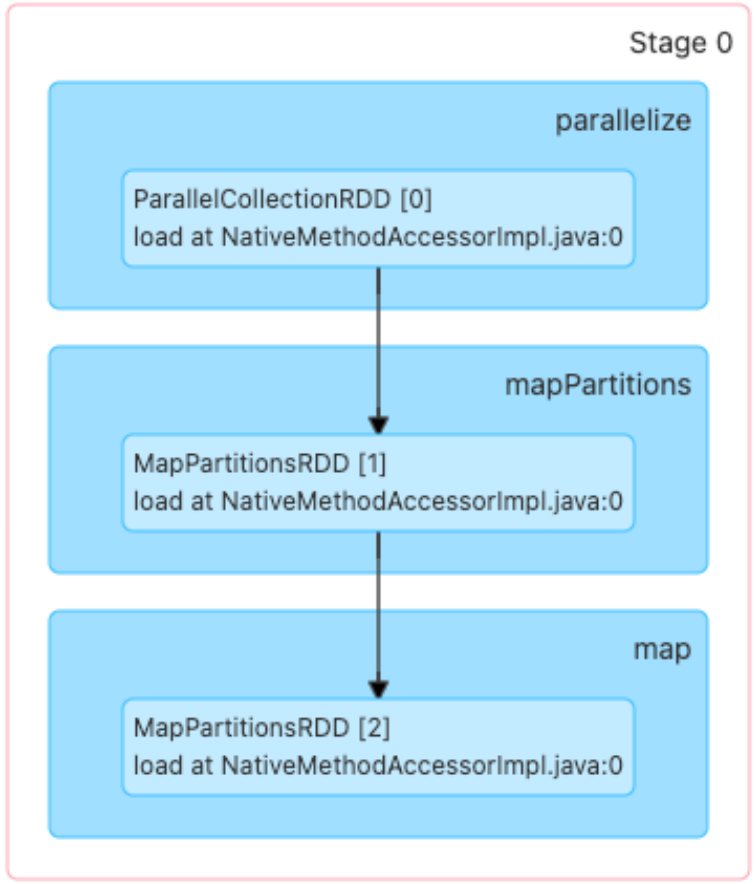
Fruits!

Chaîne de traitement des images

- Chargement des images (43 dossiers différents ici)
  en local : 0,3 secondes
  sur AWS : 4,0 secondes

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





Fruits!

Conclusion

- Solution AWS avec EC2, EMR et S3 :
  - ✓ Tolérance aux pannes : Cloud et PySpark
  - ✓ Maintenabilité : EMR et sa pré-configuration, séparation EMR et S3
  - ✓ Coût faible : Cloud, EC2 et instanciations à la demande, possibilité de savings plan sur un an
  - ✓ Augmentation du volume de données : passage à l'échelle horizontal permis par Spark et EMR/EC2, grande rapidité d'actions avec retrait et ajout de machines possibles en quelques minutes en fonction des besoins

Fruits!

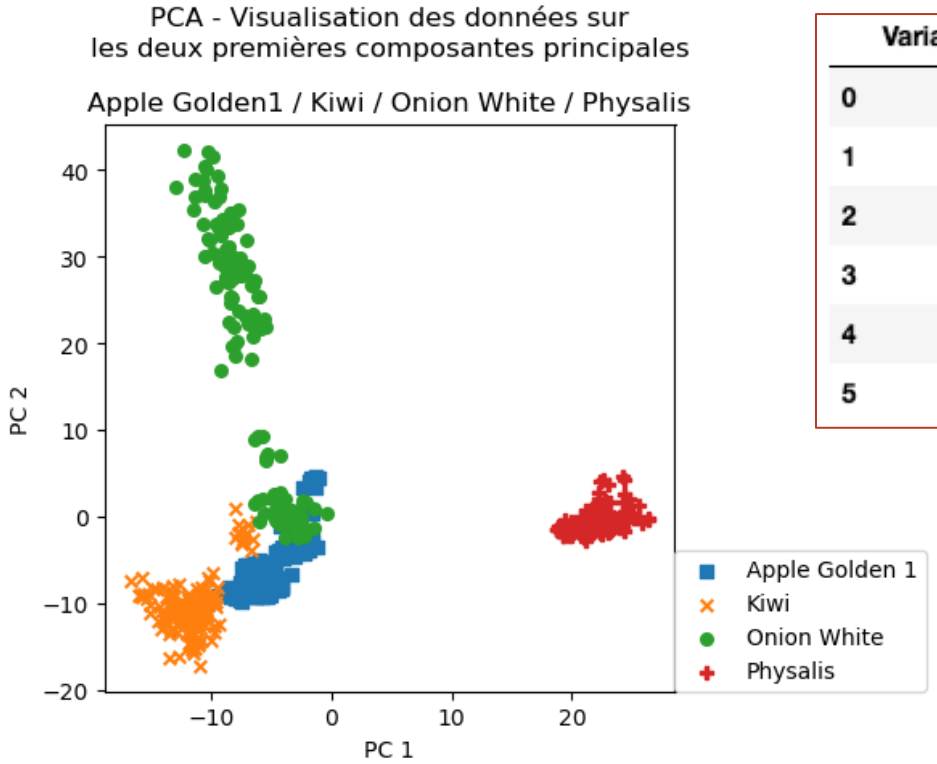
Fruits!

Merci pour votre attention

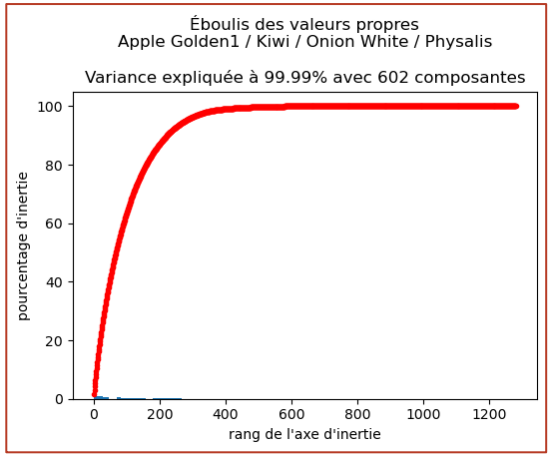
Place aux questions!

Hors propos

- **PCA** : sur 4 types de fruits
- 602 variables pour une variance expliquée à 99,99%



	Variance expliquée	k
0	90	223
1	95	278
2	98	342
3	99	396
4	99.5	455
5	99.99	602

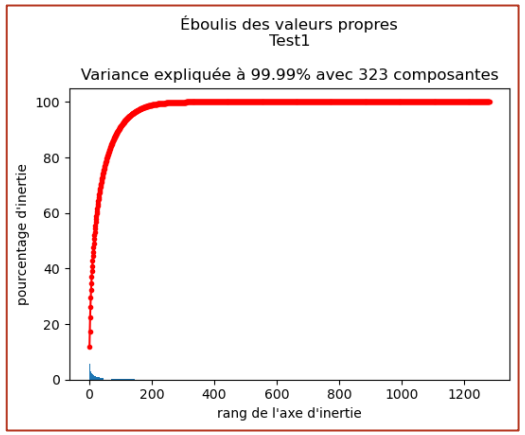


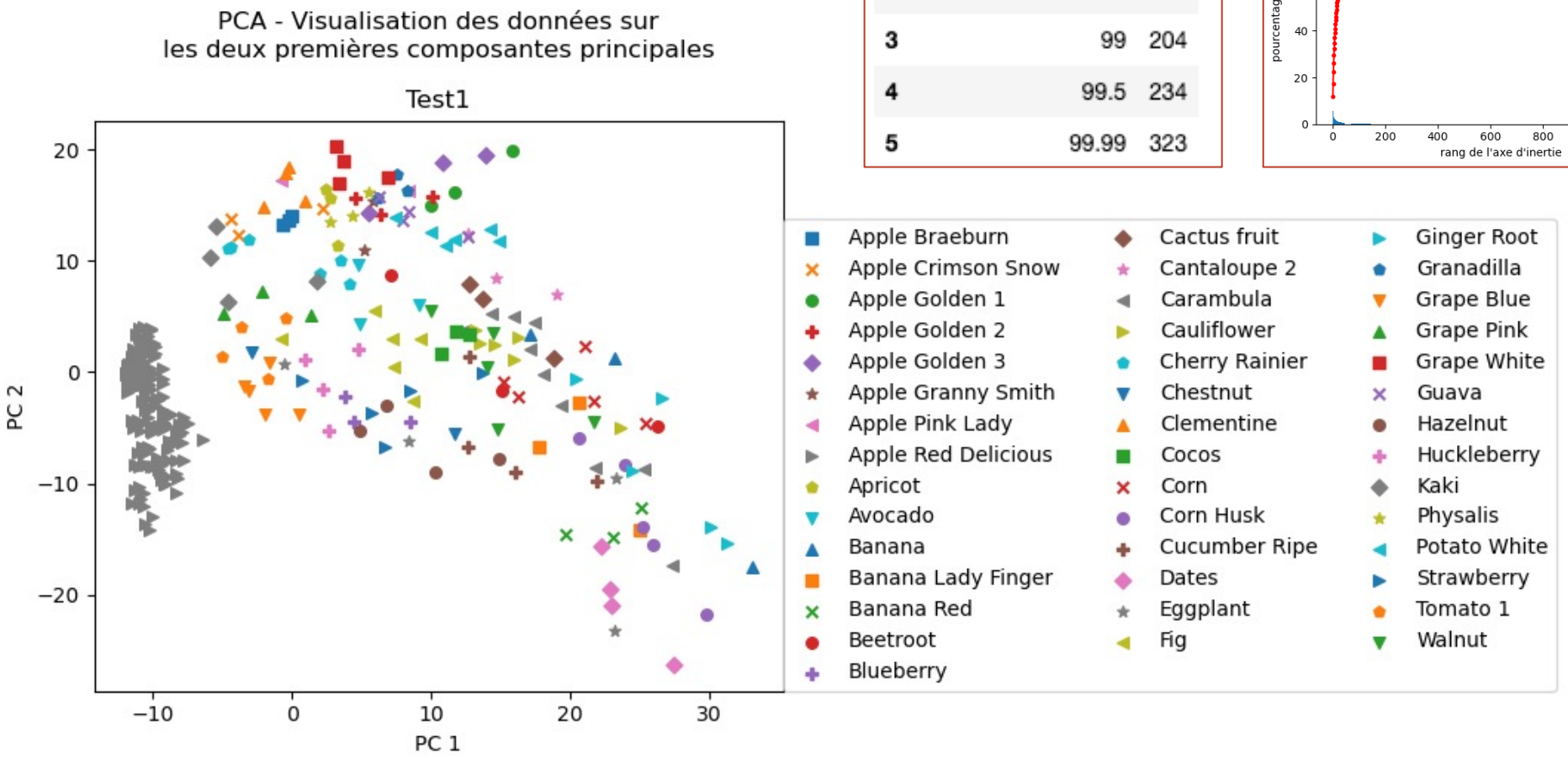
Fruits!

Hors propos

- **PCA** : sur 43 types de fruits
- 323 variables pour une variance expliquée à 99,99%

	Variance expliquée	k
0	90	94
1	95	131
2	98	174
3	99	204
4	99.5	234
5	99.99	323





Fruits!