Fruits!

Mission

A partir d’une base de photos de fruits numérisés

- Mettre en place les étapes du traitement des images, afin de fournir aux utilisateurs un service de reconnaissance de fruits
  - Traiter les images, effectuer une réduction de dimensions, sauvegarder.
  - Appliquer les étapes précédentes sur des images soumises par les utilisateurs.
- La montée en puissance du service va amener à traiter un grand nombre d’images
  - Déployer la chaine de traitement sur une architecture Big Data afin de répondre à la montée en charge de l’application

Données

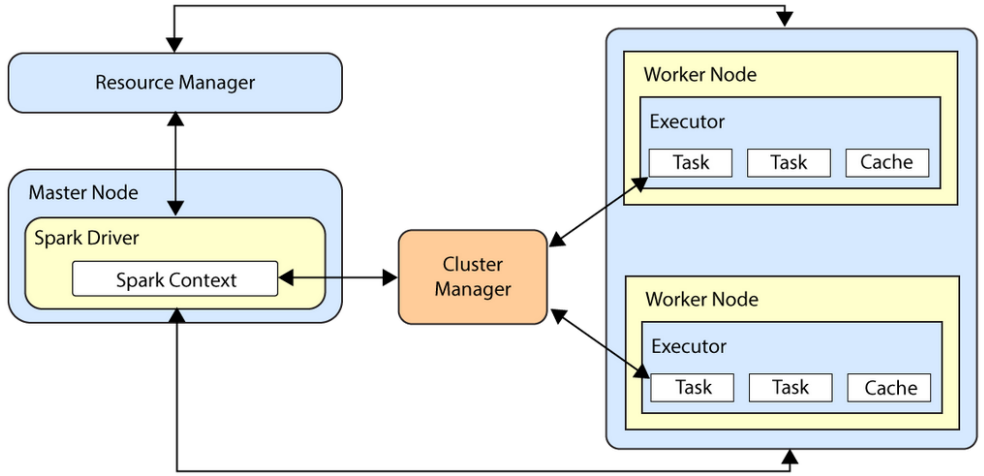
- Données issues de la compétition Kaggle Fruits360
- 131 fruits ont été photographiés:
  - Images couleur
  - Taille 100 x 100 pixels
  - Format .jpg
  - Environ 400 images par fruit

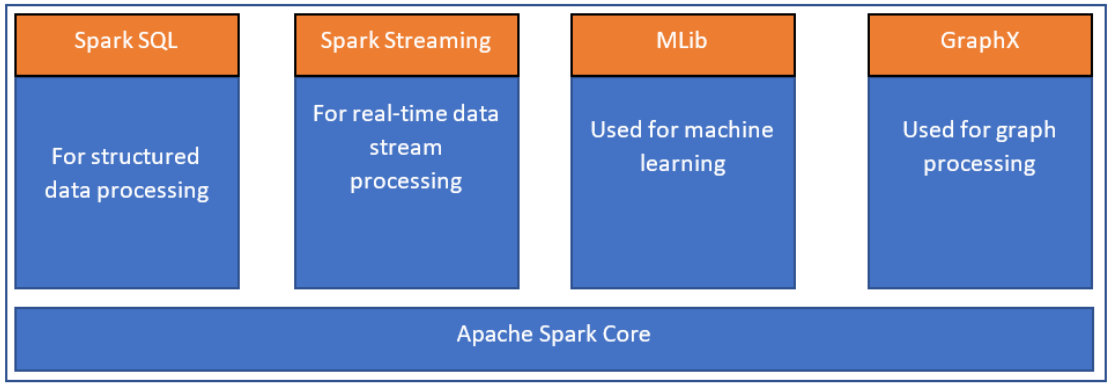
- Deux datasets:
  - Training : 67692 images
  - Test : 22688 images

Architecture Big Data

La contrainte de montée en volume des données à traiter impose l'utilisation d'un framework adapté:

- Rapide
- Distribué
- Simple d'usage
- Polyvalent





Infrastructure as a Service (IaaS)

IAM

Identity and Access Management

- Utilisateurs
- Autorisations d’accès aux données et au code

S3

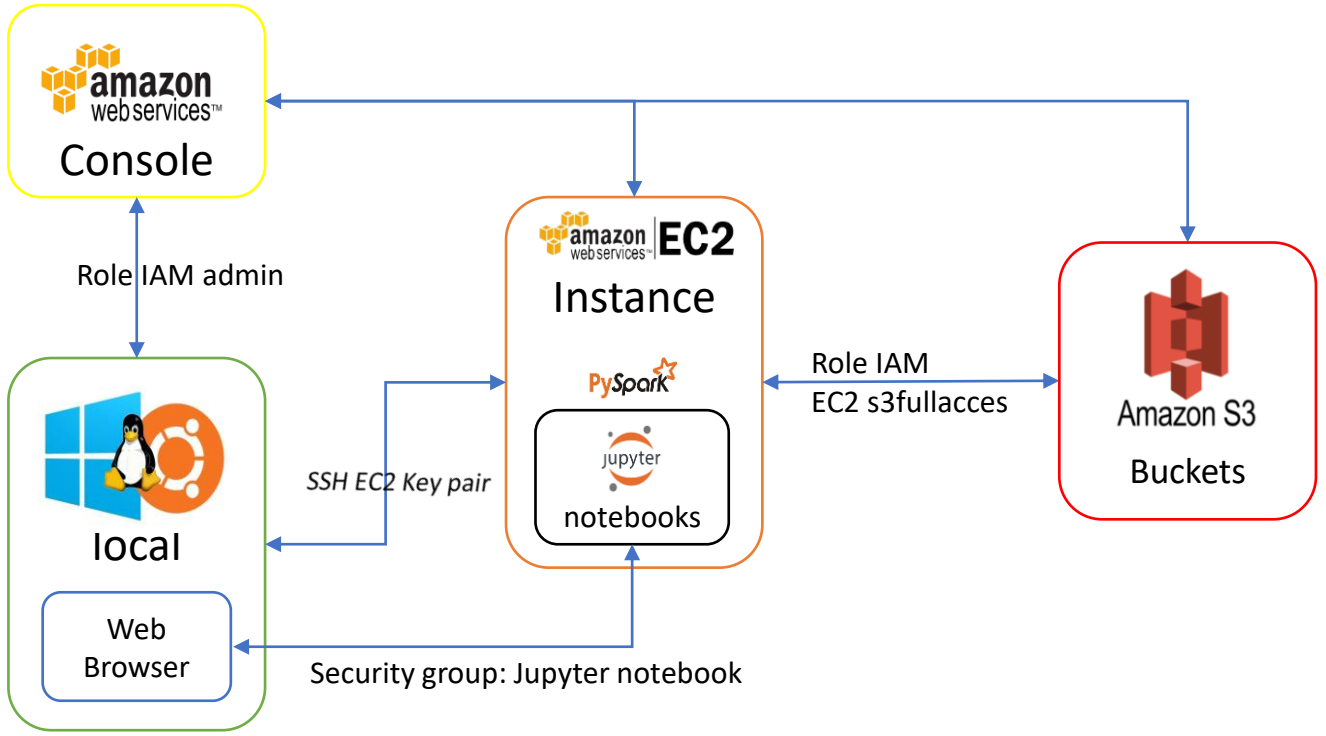
Simple Storage Service

- Capacité de stockage dynamique
- Disponibilité
- Sécurité

Cf annexe 1

Elastic Cloud Computing

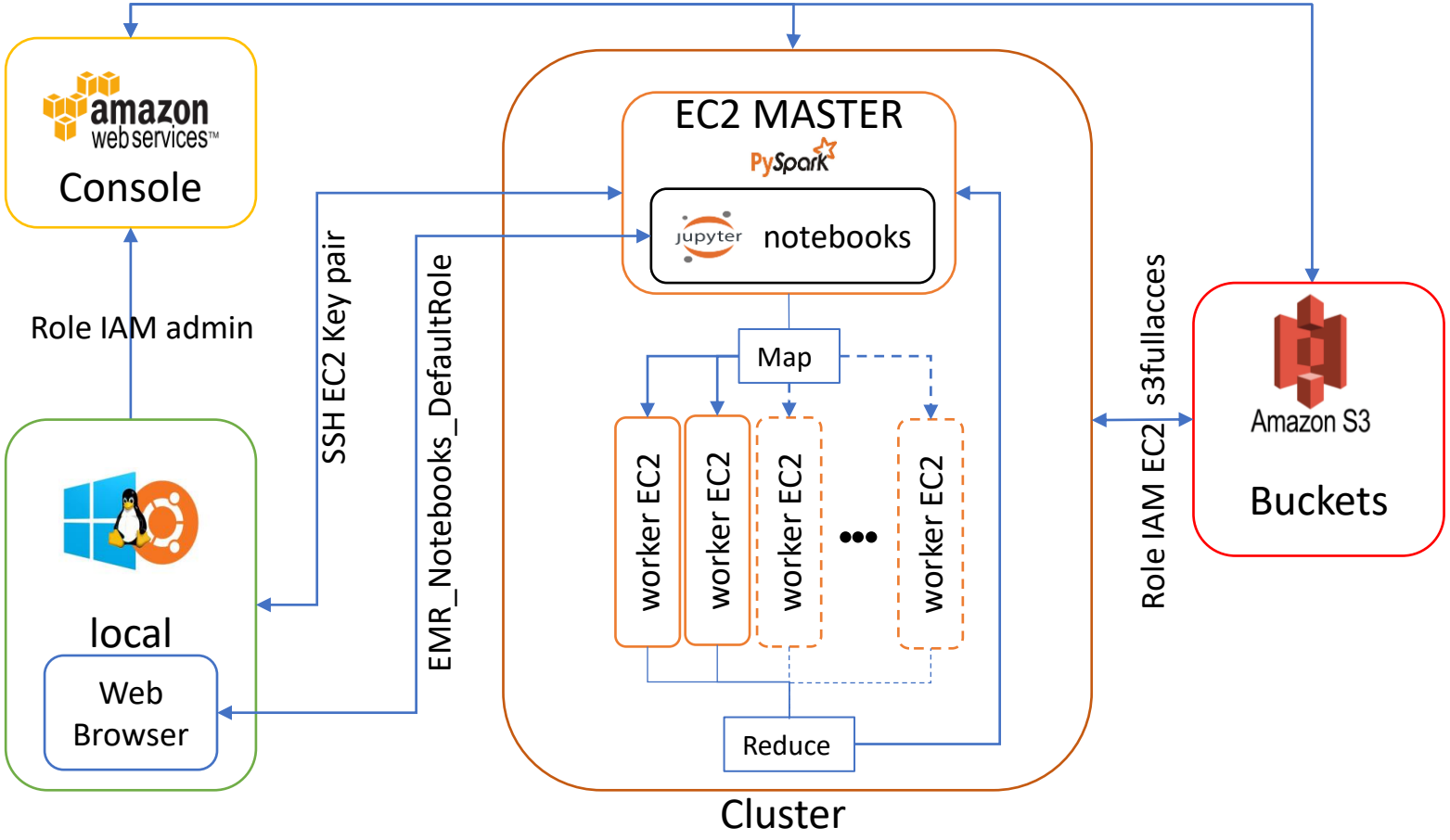
Capacité de calcul redimensionnable
Grand choix de :
- Processeur
- Volumes de stockage
- Mise en réseau
- Systèmes d'exploitation



Elastic Map Reduce

Calcul distribué sur plusieurs instances EC2

Composition du cluster en fonction des besoins

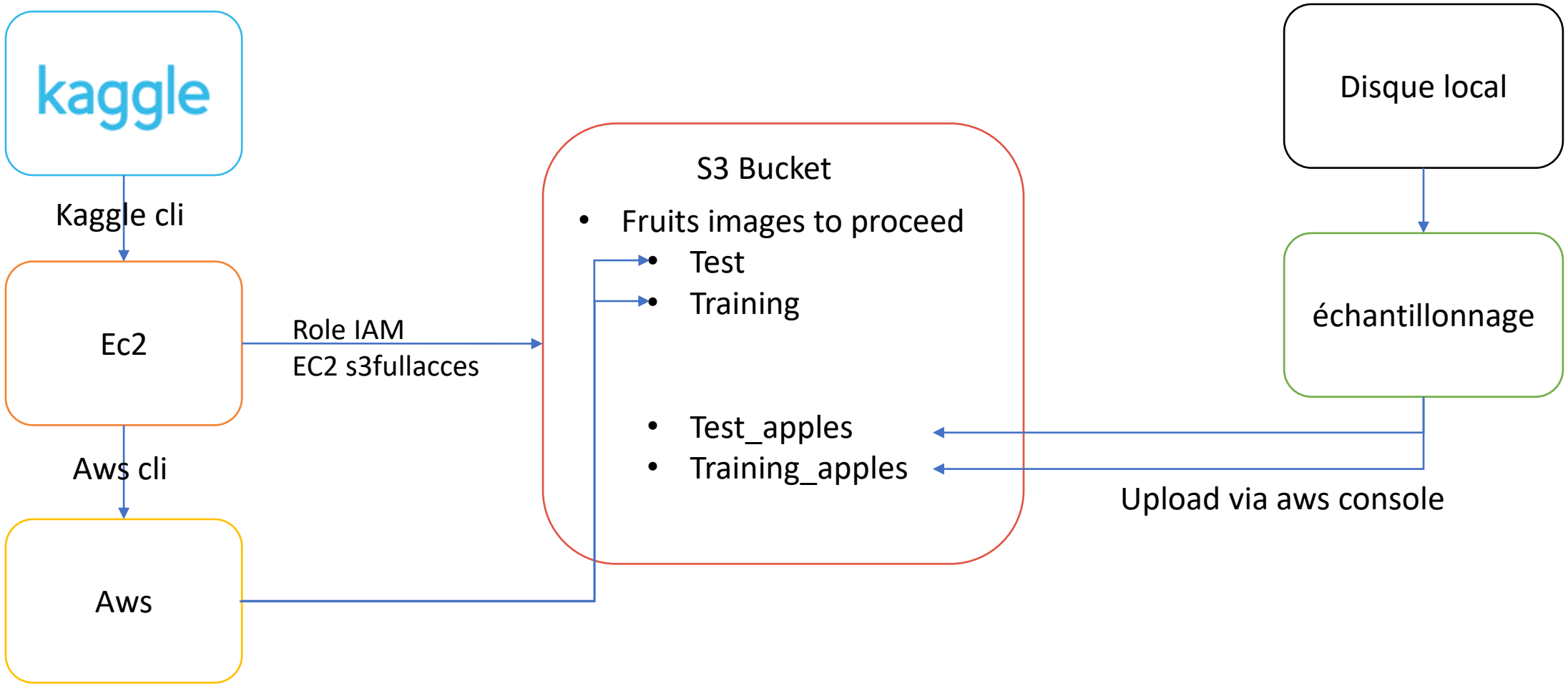


Chaine de traitement des images

Chargement récursif des images au format binary	A partir du path de l'image	Amélioration des images	Features extraites par CNN resnet50 ➢ Vecteur de dimension 2048	Nombre de composantes principales conservées en fonction du pourcentage de la variance totale expliquée désirée	PCA transform Feat_array ➢ Vecteur de dimension < 2048	Ecriture de • Path • Label • Feat_array Au format parquet Partitionné par Label
Spark SQL	pillow	K + TensorFlow			Spark ML	

Parallélisé (UDF)

Upload des données sur S3



Développement du code sur un échantillon réduit

Local

Os: wsl2 Ubuntu 20,04
Stockage: local ssd
Sparkcontext: local avec 2 workers
Arrow maxRecordsPerBatch=128

EC2 : t3.xlarge

Os: Aws linux 2
Stockage: volume EBS 12Go
Sparkcontext: local avec 2 workers
Arrow maxRecordsPerBatch=128

Training_apples

Echantillonnage
- 5 types de pommes
- 1700 images

Test_apples

Echantillonnage
- 5 types de pommes
- 805 images

Librairies:

Miniconda
Python 3.8
Java 8
Pyspark 3.1.2
Hadoop 3.2.0
Pyarrow 2.0.0
Numpy 1.19.5
Pandas 1.2.5
Pillow 8.3.2
Tensorflow 2.4.1

Passage à l'échelle sur un Cluster EMR

Cluster

Release: emr-6.3.0
Stockage: volume EBS 12Go
Hadoop distribution : Amazon 3.2.1

Master : m5.xlarge
5 Cores : m5.xlarge

4 vCore, 16 GiB memory, EBS only storage
EBS Storage: 64 GiB

Arrow maxRecordsPerBatch=512

Training

- 131 fruits
- 67692 images

Test

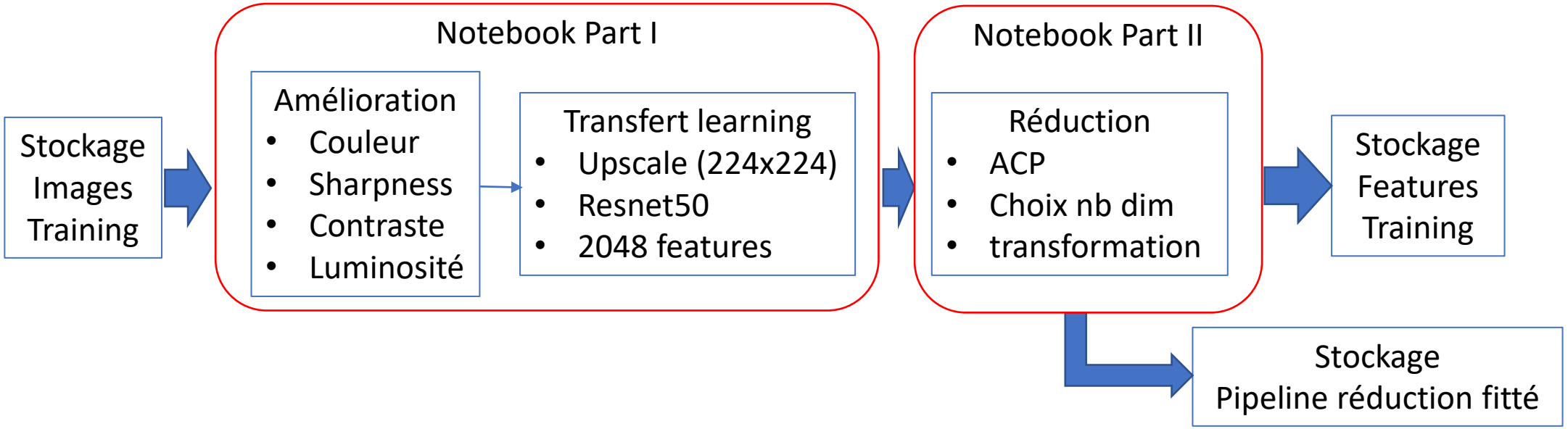
- 131 fruits
- 22688 images

Librairies:

Spark 3.1.1
Hadoop 3.2.1
Hue 4.9.0
Tensorflow 2.4.1
JupyterEnterpriseGateway 2.1.0
Pyarrow 2.0.0
Numpy
Pandas 1.2.5
Pillow 8.3.2

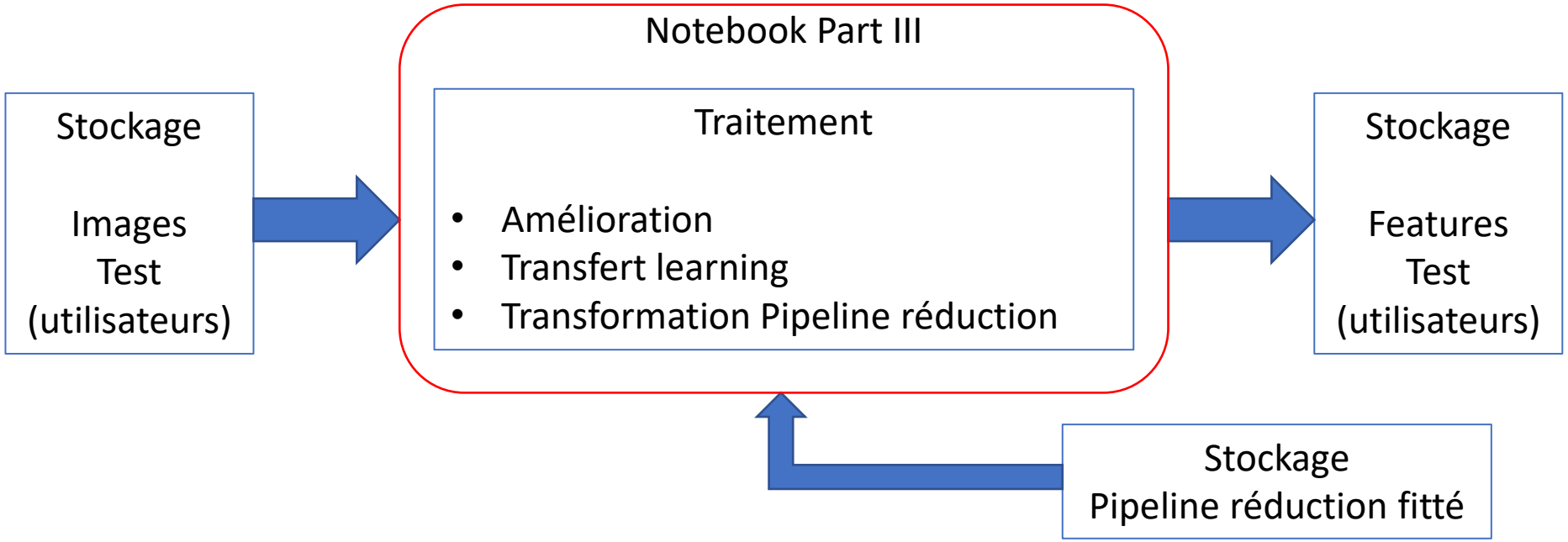
Traitement données Training

Ce jeu est ici considéré comme le set à étudier afin de déterminer les différents paramètres de la chaine de traitement:



Traitement données Test

- Ce jeu est considéré comme les images soumises par les utilisateurs de l'application:



Logistic Regression

- Sur jeu réduit (EC2):
  - PCA 183 composantes principales
  - Variance totale expliquée 90%

```
Validation f1 score : 0.971410863074718
Validation fMeasureByLabel score : 0.9296636085626913
Validation weightedFMeasure score : 0.971410863074718
Validation accuracy score : 0.9714285714285714
Validation precisionByLabel score : 0.9325153374233128
Validation weightedPrecision score : 0.9723611144790125
Validation recallByLabel score : 0.926829268292683
Validation weightedRecall score : 0.9714285714285715
Validation logLoss score : 0.06136007625303008
Validation hammingLoss score : 0.02857142857142857
```

- Sur jeu complet (EMR):
  - PCA 1468 composantes principales
  - Variance totale expliquée 99%

```
Validation f1 score : 0.9933592031893852
Validation fMeasureByLabel score : 1.0
Validation weightedFMeasure score : 0.9933592031893852
Validation accuracy score : 0.9934767277856136
Validation precisionByLabel score : 1.0
Validation weightedPrecision score : 0.9939146596956664
Validation recallByLabel score : 1.0
Validation weightedRecall score : 0.9934767277856131
Validation logLoss score : 0.0228604962993868
Validation hammingLoss score : 0.00652327221438646
```

Conclusion

- Ce projet à permis:
  - L'utilisation du framework big data pySpark-Hadoop
  - L'utilisation de services de base d'AWS
  - Le déploiement sur différentes machines en fonction du volume de données à traiter:
    - Local : mise au point du process sur un nombre de données réduit
    - EC2 : Traitement sur un échantillon réduit d'images
    - EMR : Calcul distribué sur un cluster de la totalité des données (cpu load cf. annexe2)
  - La création d'un process de modélisation simple permettant de valider la qualité du featuring des images

- Améliorations à apporter:
  - Préprocessing des images plus poussé
  - Essai de différentes solutions pour le tranfert learning
  - Etude des temps de traitements upscaling vs outscaling
  - Gestion des coûts de stockage

Annexe 1 : Sécurité et autorisations

Role IAM admin — Utilisateur principal

Role IAM EC2 s3fullacces — Accorde aux instances EC2 les droits en lecture écriture sur les buckets S3

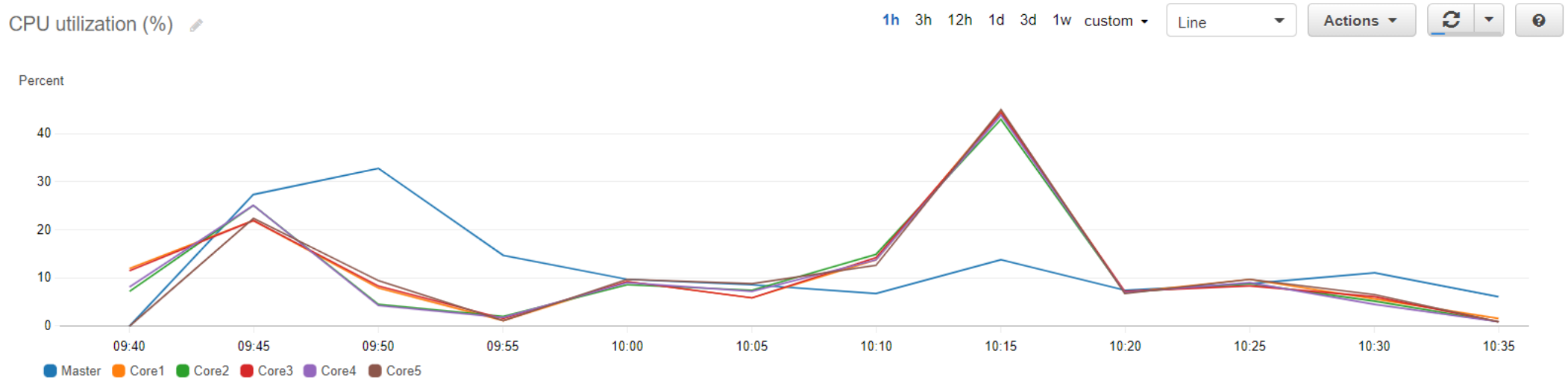
SSH Paire de clés

Name	Type	Fingerprint	ID
P8_OC…	rsa	ef:a2:d6:4e:ff:d7:b3:…	key-04fea4bc05e736f49

Security group: Jupyter notebook

Security group rule…	IP version	Type	Protocol	Port range	Source
sgr-06fa16e817bdf1b2e	IPv4	SSH	TCP	22	0.0.0.0/0
sgr-08bd0fa75da68721d	IPv4	Custom TCP	TCP	8888	0.0.0.0/0
sgr-0f2fcb133fba4b047	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-0c86664c92c152d3e	IPv4	Custom TCP	TCP	4040	0.0.0.0/0

Annexe 2 : Charge Cpu cluster exécution partie modélisation



All metrics | Graphed metrics (6) | Graph options | Source

Math expression | Dynamic labels | Statistic: Average | Period: 1 Second | Remove all

Label	Details	Statistic	Period	Y Axis	Actions
Master	EC2 • CPUUtilization • InstanceId: i-0888b8e5027fe856e	Average	1 Second		
Core1	EC2 • CPUUtilization • InstanceId: i-0040cfcfa382d7532	Average	1 Second		
Core2	EC2 • CPUUtilization • InstanceId: i-0d3f6a0db34186aef	Average	1 Second		
Core3	EC2 • CPUUtilization • InstanceId: i-04c15f312777eadae	Average	1 Second		
Core4	EC2 • CPUUtilization • InstanceId: i-024832de250ef50e4	Average	1 Second		
Core5	EC2 • CPUUtilization • InstanceId: i-0eb9f9eee185e4c94	Average	1 Second		