DÉPLOYEZ UN MODÈLE DANS LE CLOUD

KONAN KOFFI

PLAN DE PRÉSENTATION

I/ Problématique et jeu de données

II/ Étapes de la chaîne de traitement

- *Preprocessing et réduction de dimensions*

III/ Différentes briques d'architecture choisies sur le cloud et leur rôle dans l'architecture Big Data

IV/ Conclusion et recommandation

I PROBLÉMATIQUE ET JEU DE DONNÉES

Start-up de l'AgriTech

Fruits!

Objectifs

- **I Reconnaissance de fruit**

ORANGE

Riche en vitamine C

- *action antioxydante*
- *favorise l'absorption du fer*
- ...

- **II Introduire de l'IA dans la récolte de fruit tout en respectant la biodiversité des fruits**

I PROBLÉMATIQUE ET JEU DE DONNÉES

Problématique

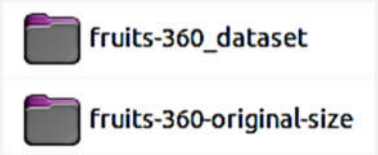
- ***Comment pourrait-on traiter de gros volumes de données (volumétrie extrême) ?***

Mission

- *Mise en place de premières briques de traitement des images*
- *Développement de scripts adaptés en termes de volume de données et de vélocité dans un environnement Big Data*

Jeu de données*

- *Un jeu de données avec 90380 images de 131 fruits et légumes*



- ***Plus d'1 Go de Données***

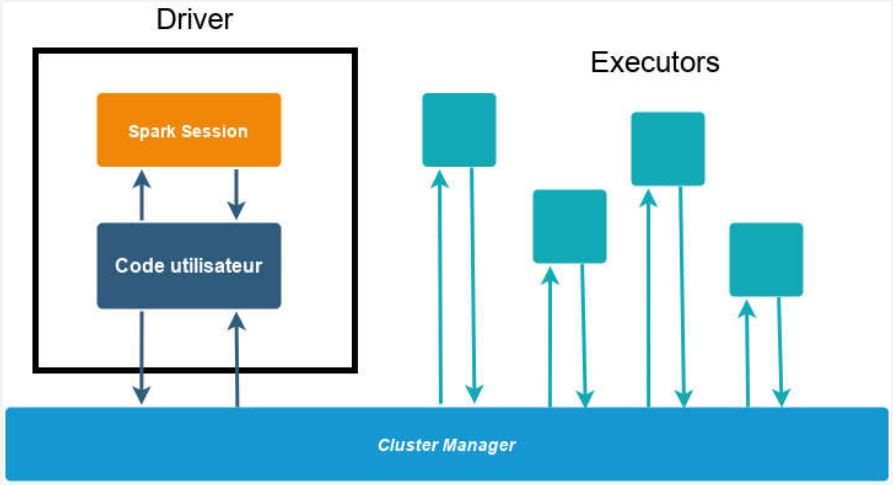
* https://www.kaggle.com/datasets/moltean/fruits

II ÉTAPES DE LA CHAÎNE DE TRAITEMENT

- *Une infrastructure de calcul de cluster open-source avec des analyses **en mémoire** **(in-memory)***

- *Plus rapide que certains systèmes informatiques en cluster (Hadoop)*
- *100 fois plus vite en mémoire et 10 fois plus vite même sur le disque*
- *Idéale pour l'analyse de données à grande échelle*

Architecture globale Apache Spark



https://meritis.fr/larchitecture-framework-spark/

https://www.lebigdata.fr/apache-spark-tout-savoir https://datascientest.com/apprendre-a-utiliser-lapi-python-pour-spark http://b3d.bdpedia.fr/spark-batch.html

II ÉTAPES DE LA CHAÎNE DE TRAITEMENT

Mise en place de la SparkSession dans le notebook Jupyter

API Python pour Spark

SparkSession : commencer à programmer avec
Spark SQL & DataFrame,
Spark Datasets,
Spark MLlib.

DataFrame

```
+--------------------+------+-----+---------+----+--------------------+
|              origin|height|width|nChannels|mode|                data|
+--------------------+------+-----+---------+----+--------------------+
|s3a://dmc-in/Trai...|   798|  324|        3|  16|[FF FF FF FF FF F...|
|s3a://dmc-in/Trai...|   797|  325|        3|  16|[FF FF FF FF FF F...|
|s3a://dmc-in/Trai...|   793|  335|        3|  16|[FF FF FF FF FF F...|
|s3a://dmc-in/Trai...|   796|  328|        3|  16|[FF FF FF FF FF F...|
|s3a://dmc-in/Trai...|   787|  339|        3|  16|[FF FF FF FF FF F...|
+--------------------+------+-----+---------+----+--------------------+
only showing top 5 rows

root
 |-- image: struct (nullable = true)
 |    |-- origin: string (nullable = true)
 |    |-- height: integer (nullable = true)
 |    |-- width: integer (nullable = true)
 |    |-- nChannels: integer (nullable = true)
 |    |-- mode: integer (nullable = true)
 |    |-- data: binary (nullable = true)
```

Conversion des code de couleurs et modification de la taille de nos images à 224 × 224

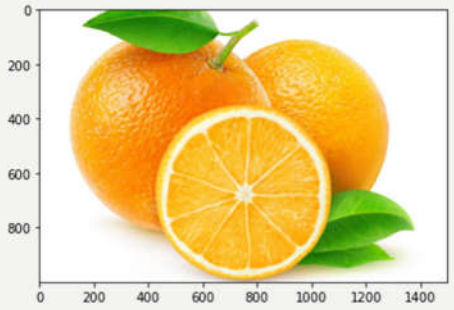


Image réelle (RGB)

Spark

Entrée attendue : **BGRA**

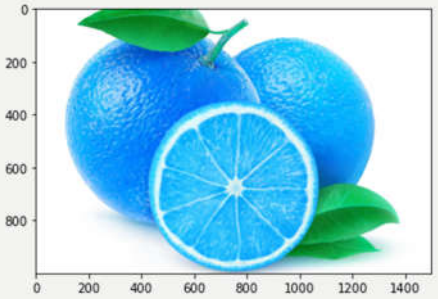


Image bleue après traitement Spark

Transformations

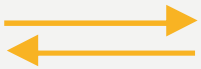
- RGB ➔ BGR
- Échelle réduite

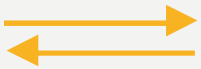
224 × 224

https://www.lebigdata.fr/apache-spark-tout-savoir

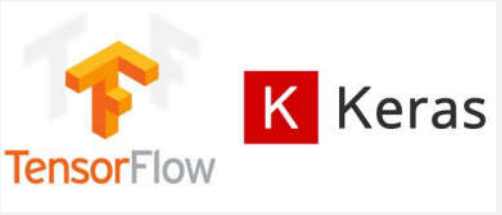
II ÉTAPES DE LA CHAÎNE DE TRAITEMENT

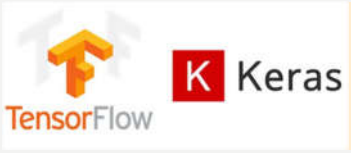
Outils de ML* :
Sckit-Learn
PCA





DL*





Extraction de features

Vectorisation des features d'images

Transfer Learning via ResNet50 model

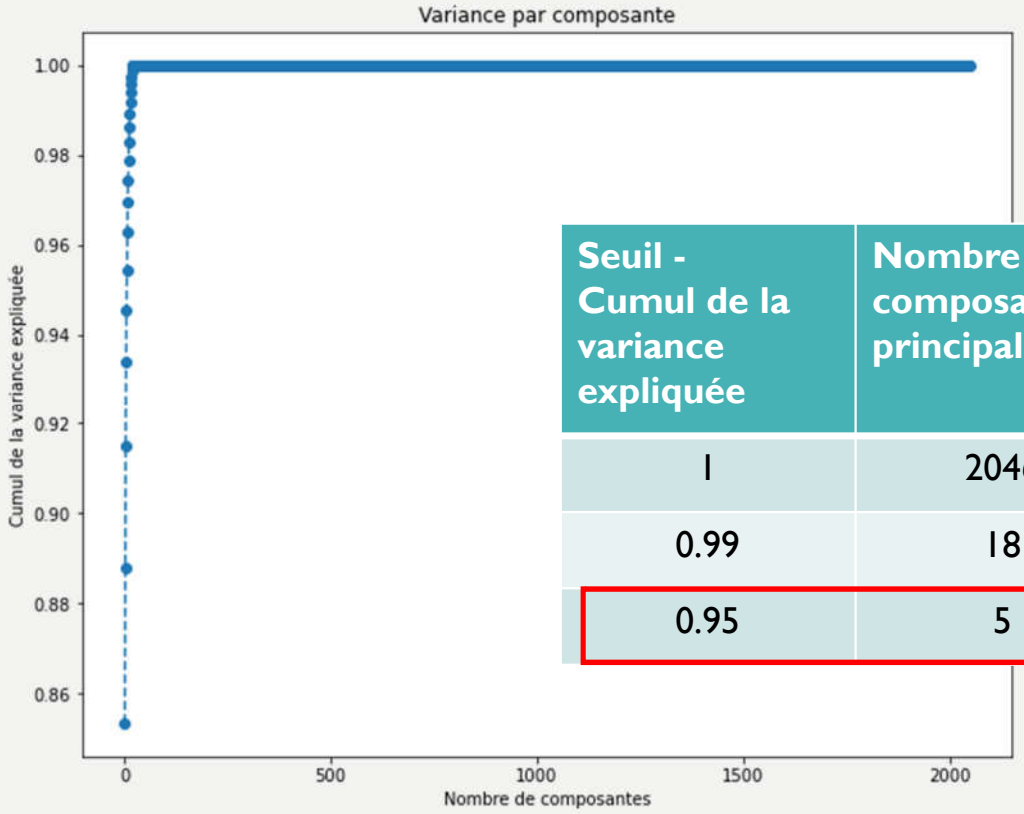
- *Vecteurs denses*
- *Nécessaire pour MLlib*

- *Suppression de la dernière couche du réseau de neurones*
- *Récupération des poids*

ML: *Machine Learning*, DL: *Deep Learning*

II ÉTAPES DE LA CHAÎNE DE TRAITEMENT

Réduction de dimensionnalité PCA (intégré à Spark)



Seuil - Cumul de la variance expliquée	Nombre de composantes principales
1	2046
0.99	18
0.95	5

DataFrame avec **PCA(features) + Label**

pca_features	label
[14.5507688986928...	cucumber_1
[14.6421758454457...	cucumber_1
[14.0862501664891...	cucumber_1
[15.6812748471574...	cucumber_1
[12.4223389816891...	cucumber_1

only showing top 5 rows

Stockage de DataFrame

III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

Architecture Big Data AWS

- *basée sur un serveur EC2 Linux*

S3

- ***Stockage de données***

EC2

- ***Calculs***
- ***Déploiement de modèle***

IAM

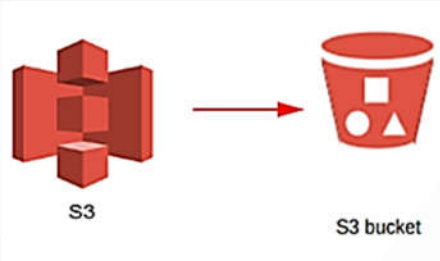
- ***Control d'accès***

III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

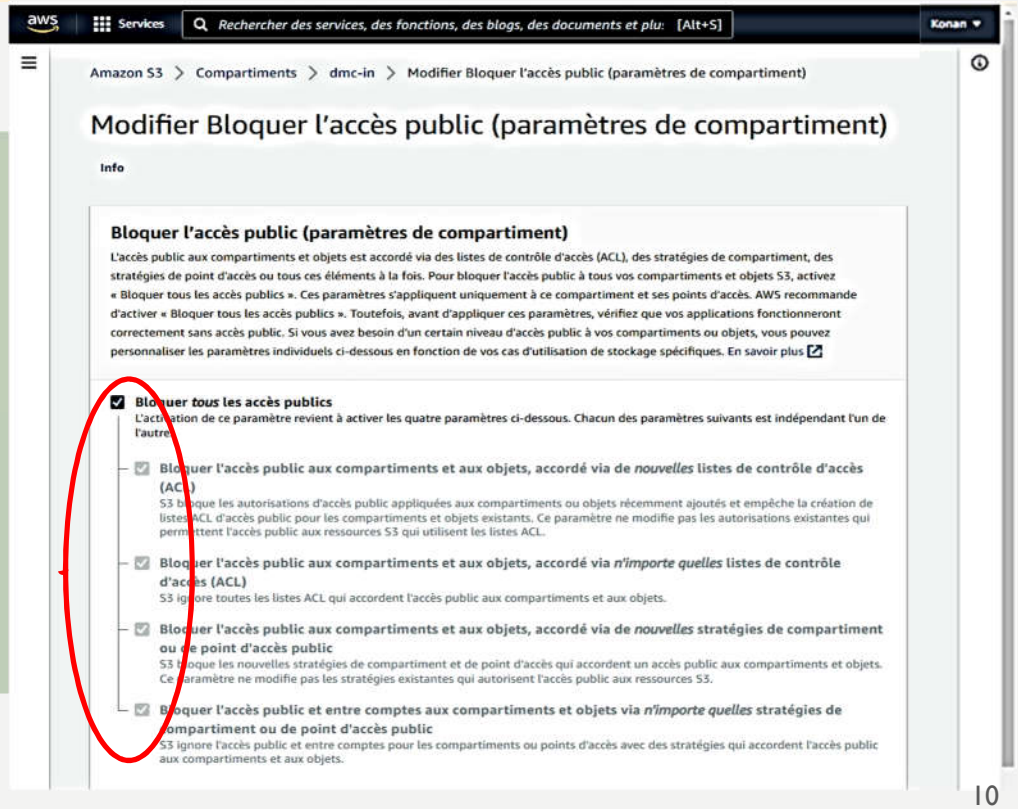
S3 (Simple Storage Service)

Évolutivité de stockage d'objets, Disponibilité des données

Sécurité et des performances de pointe



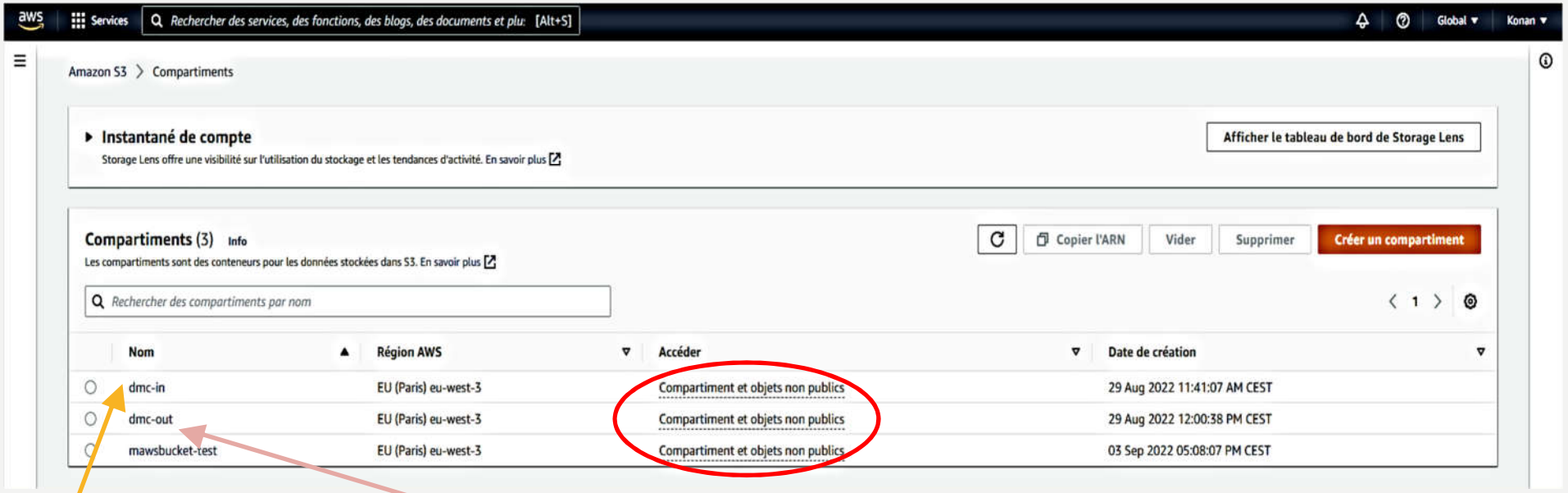
- Applications mobiles
- Sauvegardes et restaurations
- Archives
- Applications métier
- Appareils IoT
- Analyses de Big Data



III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

S3, Buckets et Objets

- *Fonctions de gestion pour **optimisation**, **organisation** et pour **configuration** de l'accès aux données (des structurées aux non-structurées)*
- *Répondre à des exigences spécifiques*



Stockage de Données d'images

Stockage Features d'images après la réduction de dimension PCA

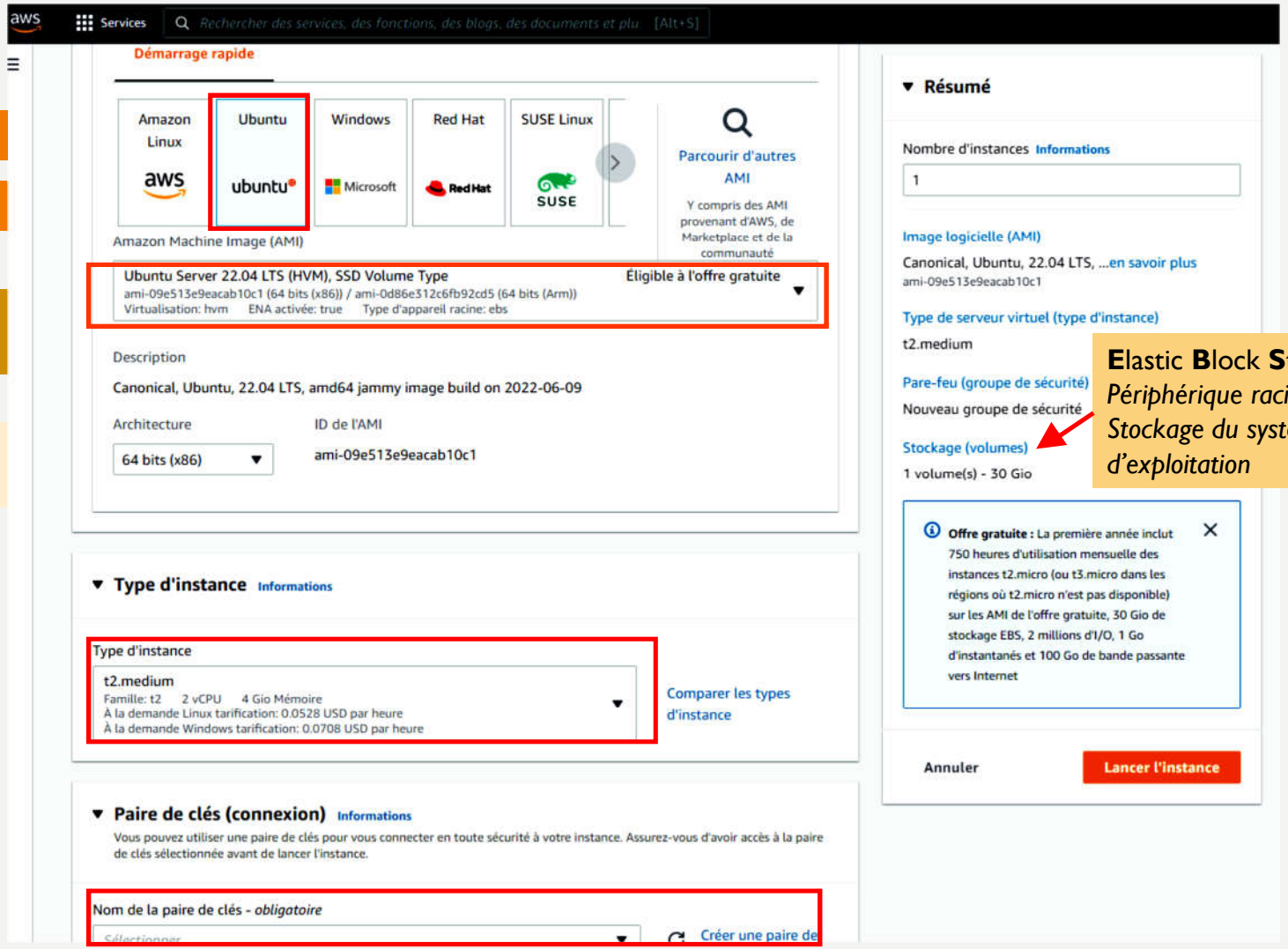
III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

(EC2)

(Elastic Compute Cloud)

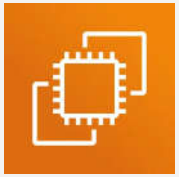
Servers pour exécuter des applications web, ...

Offre IaaS d'Amazon Web Services à l'usage courant.



Elastic **B**lock **S**tore :
Périphérique racine
Stockage du système d'exploitation

III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA



(EC2)

Création de l'instance EC2 server Linux

Création obligatoire d'une paire de clés publique/privée SSH ?

Clé privée

Clé publique

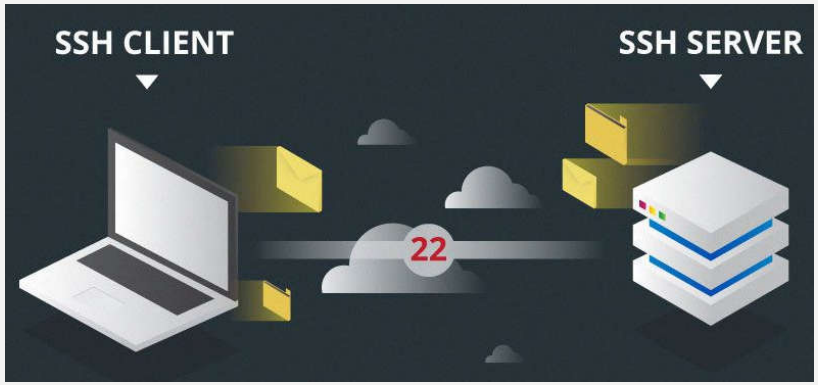
- *Protocole de **connexions** sécurisées entre deux systèmes*

Architecture client/server

Clé privée ⇄ **Clé publique**

Échange de clés sécurisées

Local

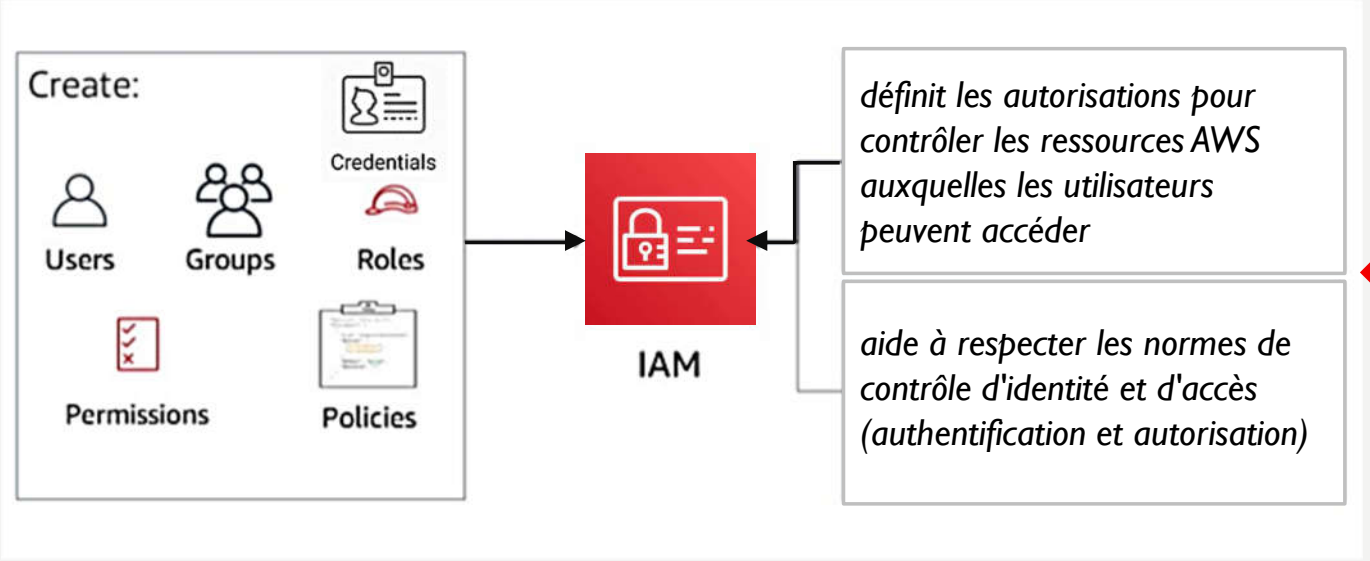


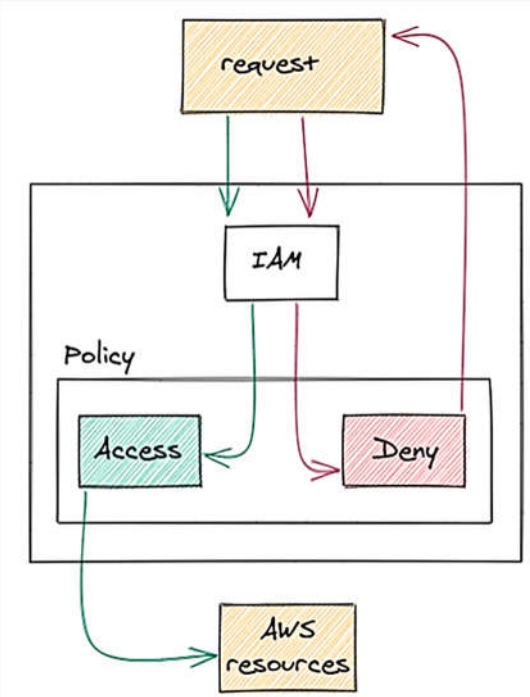
Systèmes hôte servers AWS EC2 Linux

III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

IAM (Identity and Access Management)

Rappel : *Service de gestion des identités et des accès d'AWS*

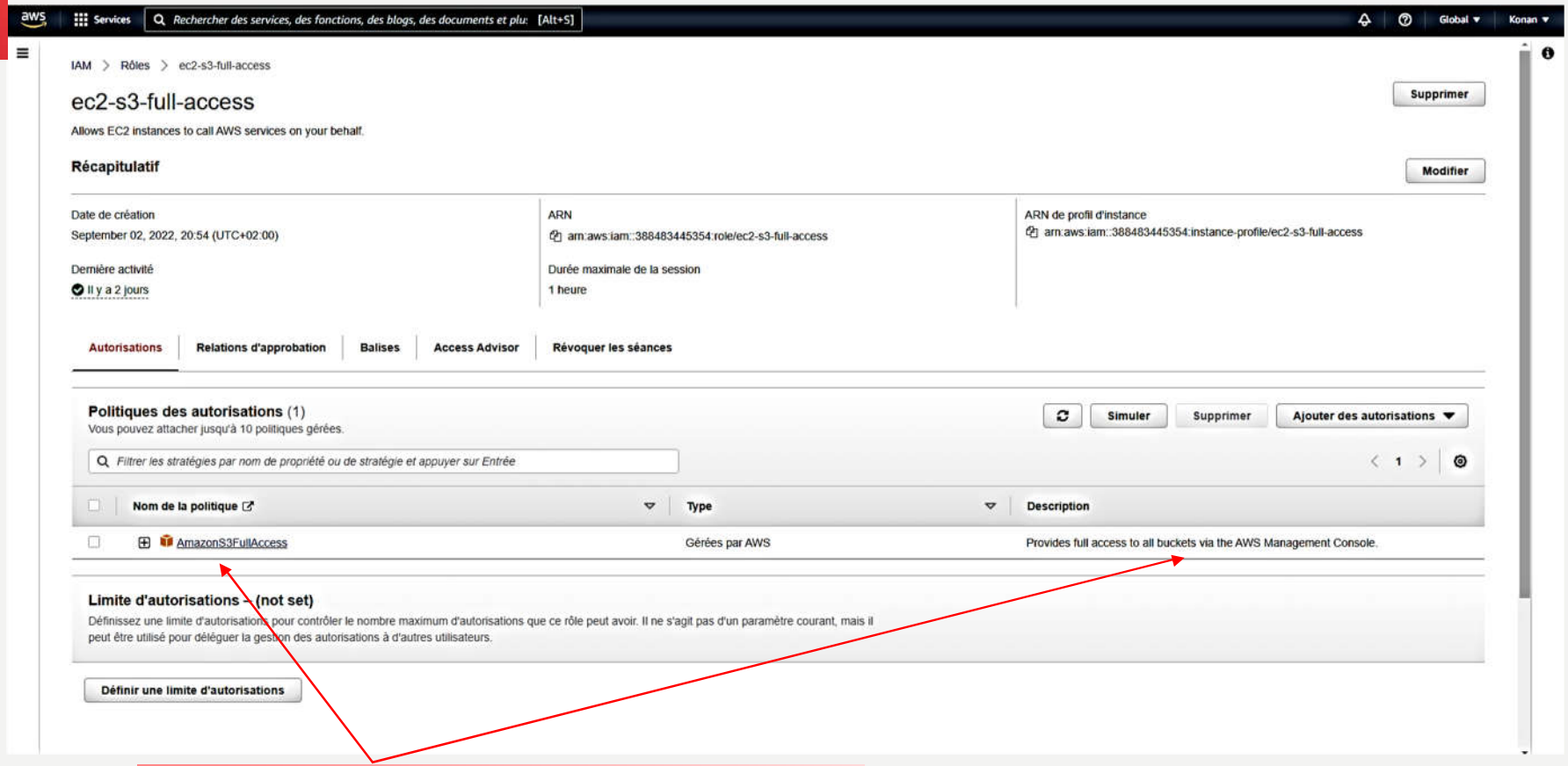




III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

IAM Role

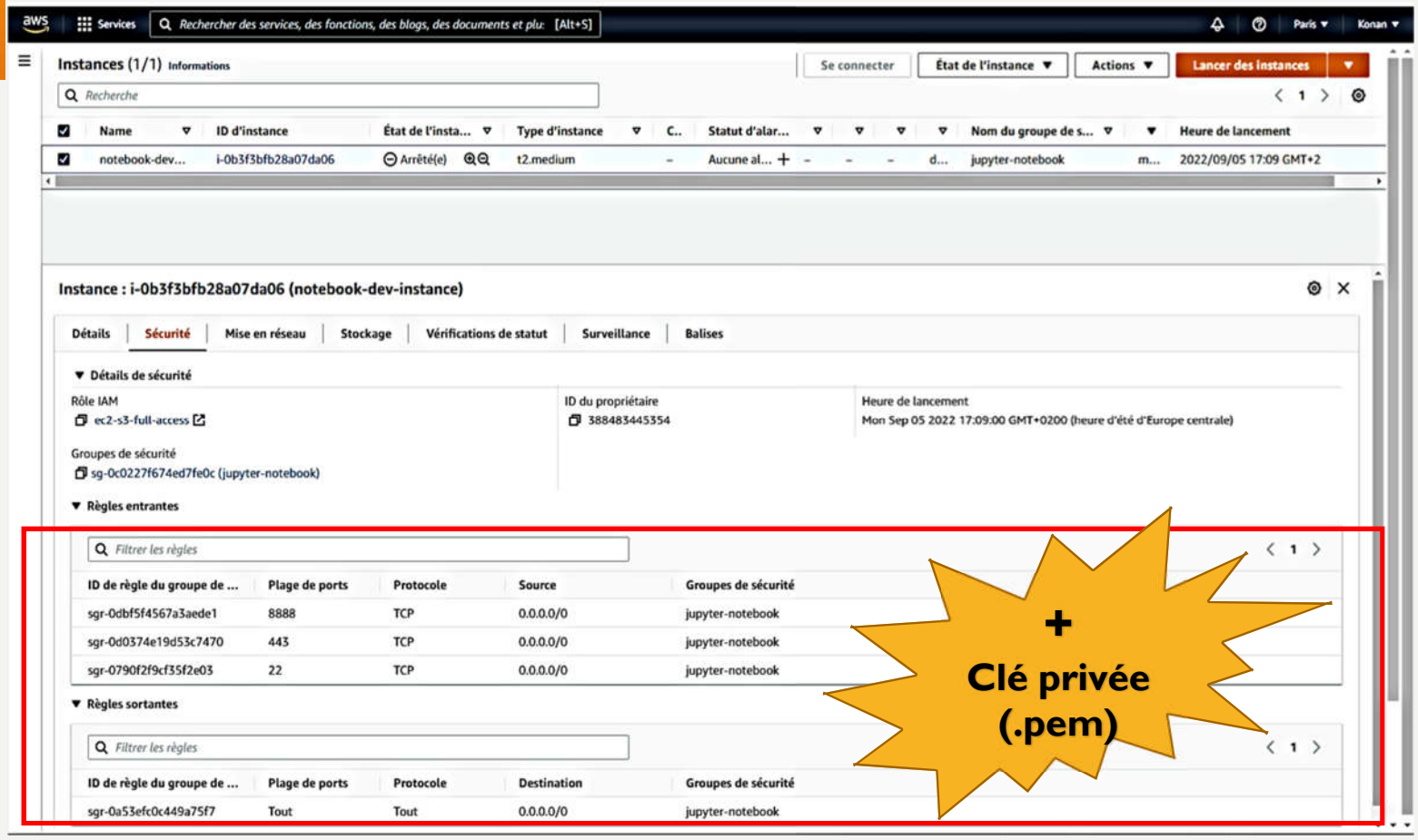
Tableau de bord de la stratégie attachée au rôle IAM



Autorisation d'un accès total aux ressources S3

III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

Tableau de bord de notre instance EC2 et configuration du notebook Jupyter sur AWS

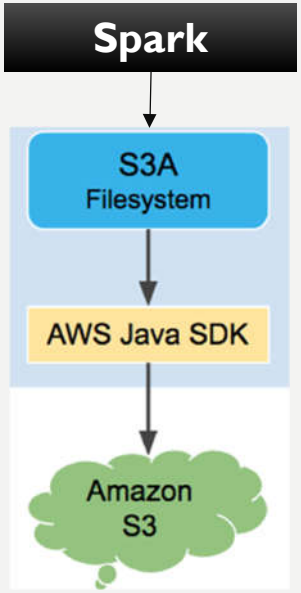


III DIFFÉRENTES BRIQUES D'ARCHITECTURE CHOISIES SUR LE CLOUD ET LEUR RÔLE DANS L'ARCHITECTURE BIG DATA

Communication entre Spark et AWS S3

Complément de dépendances dans le **répertoire Spark**

Dépendances java « *module*.jar » :



hadoop-aws-3.3.1.jar

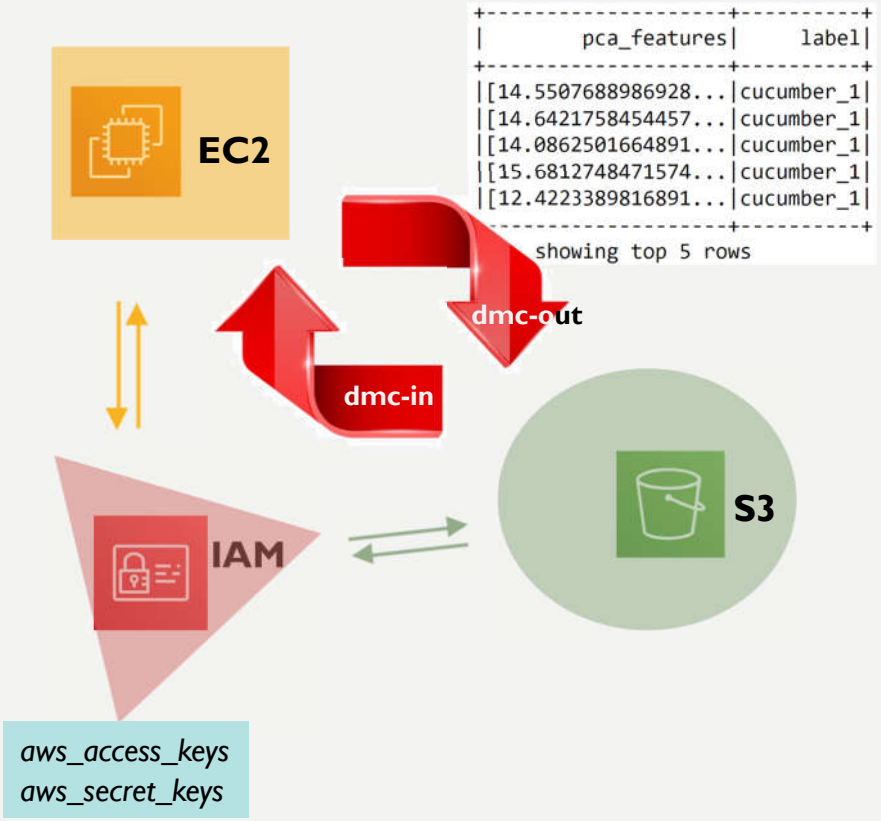
- *Intégration de Spark (et autres applications de l'écosystème Hadoop) avec **AWS***

Le connecteur S3A permet de lire et d'écrire des fichiers stockés dans Amazon S3 grâce avec une URL de préfixe : s3a://

aws-java-sdk-bundle-1.11.901.jar

- *Prise en compte du cycle de vie des API,*
- *Gestion des informations d'identification, les tentatives, l'organisation des données et la sérialisation*

Autorisation d'accès aux fichiers dans S3 à EC2




```
+--------------------+----------+
|        pca_features|     label|
+--------------------+----------+
|[14.5507688986928...|cucumber_1|
|[14.6421758454457...|cucumber_1|
|[14.0862501664891...|cucumber_1|
|[15.6812748471574...|cucumber_1|
|[12.4223389816891...|cucumber_1|
+--------------------+----------+
only showing top 5 rows
```

https://docs.cloudera.com

IV/ CONCLUSION ET RECOMMANDATION

Tout projet qui consiste à étudier des millions voire des milliards de données doit judicieusement se tourner vers des outils adaptés au Big Data.

➢ **Scripts**

Une place de choix dans le calcul distribué accélérant principalement la vitesse d'exécution de traitement sur des bases de données d'une extrême volumétrie.

➢ **Les ressources Big Data**

Recommandation
au traitement de données évolutives

➢ **Amazon EC2**
+ *Auto Scaling*
Bon complément à l'équilibrage de charge

➢ **Amazon EMR**
(Elastic Map Reduce)
mise en place automatique d'un cluster Spark qui gère naturellement l'auto scaling en fonction de son paramétrage.

Merci pour votre attention