

Déployez un modèle dans le cloud

Projet 8 du parcours Data Scientist

25 janvier 2024

Méric Manuel Kucukbas



Contexte



Environnement Big Data



Traitement images



Architecture de développement



Conclusions et perspectives



Annexes

"Fruits!", jeune start-up de l'AgriTech, cherche à proposer des **solutions innovantes pour la récolte des fruits.**

La volonté de l'entreprise est de préserver la biodiversité des fruits en permettant des traitements spécifiques pour chaque variété de fruits en développant des robots cueilleurs intelligents.

Nous souhaitons dans un premier temps **nous faire connaître** en mettant à disposition du grand public une **application mobile** qui permettant aux utilisateurs de **prendre en photo un fruit et d'obtenir des informations sur ce fruit.**

Cette application permettra de sensibiliser le grand public à la biodiversité des fruits et de **mettre en place une première version du moteur de classification des images de fruits.**

De plus, le développement de l'application mobile permettra de construire une **première version de l'architecture Big Data nécessaire.**



Un alternant a formalisé un document dans lequel il teste une première approche dans un environnement Big Data. Le notebook réalisé par l'alternant servira de point de départ pour construire une partie de la chaîne de traitement des données.

Objectif :

Réviser la chaîne de traitement proposée par l'alternant pour le développement du moteur de classification, il n'est **pas nécessaire d'entraîner un modèle** pour le moment, et la **compléter** avec une étape de réduction **de dimensions**.

Mettre en place les premières briques de traitement qui serviront lorsqu'il faudra passer à l'échelle en termes de volume de données.



Données : 90483 images
100x100 pixels
131 fruits

Training dataset : 67692 images

Test dataset : 22688 images

Test multiple fruits : 103 images

On ne prendra que le dataset test et pour des raisons de coût, on ne prend que :

Plum 3	304
Pear Stone	237
Pear Forelle	234
Pear 2	232
Pear Red	222
Apple Red Yellow 2	219
Pear Monster	166
Banana	166
Banana Red	166
Apple Red Delicious	166
Pear Williams	166
Pear Abate	166
Apple Golden 2	164
Apple Granny Smith	164
Apple Braeburn	164
Pear	164
Apple Red Yellow 1	164
Apple Red 1	164
Apple Red 2	164
Apple Golden 3	161
Apple Golden 1	160
Banana Lady Finger	152
Apple Pink Lady	152
Plum	151
Apple Crimson Snow	148
Apple Red 3	144
Plum 2	142
Pear Kaiser	102

- 28 fruits
- 4 catégories:
 - Pomme
 - Poire
 - Banane
 - Prune
- 4904 images

 Ananas



 Blueberry



 Pear Kaiser





Contexte



Environnement Big Data



Traitement images



Architecture de développement

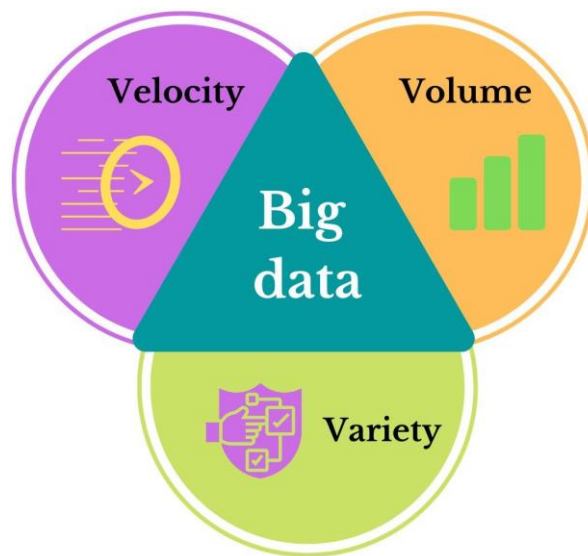


Conclusions



Annexes

Enjeux
























Volume	Données récoltées de plus en plus massives
Variété	Données de forme et type très variées
Vitesse	Lecture de données en continue
Véracité	Besoin de fiabilité sur les données

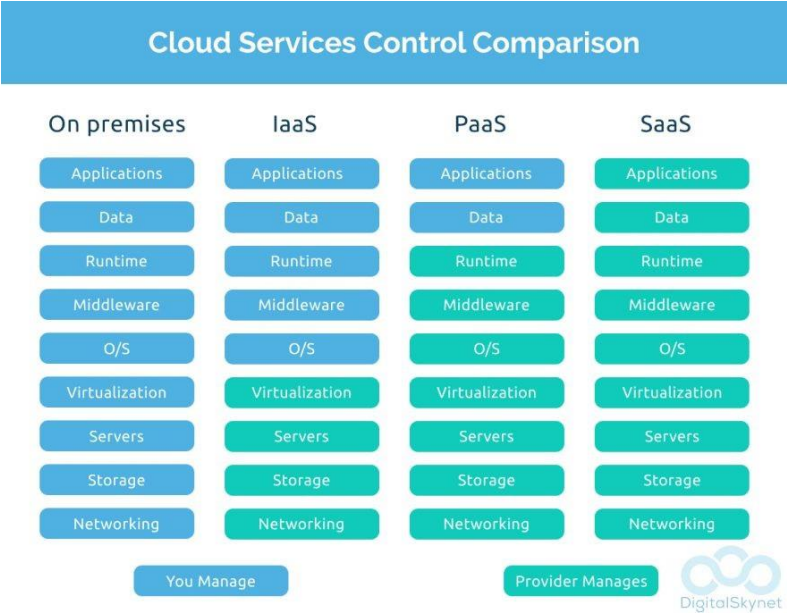
La **tolérance aux pannes** : loi de Murphy stipule, la probabilité qu'un composant tombe en panne tend vers 1 avec le temps

Une **bonne maintenabilité** : architecture facile à maintenir et à modifier.

Un **coût faible** : déployer des composants simples, ajustés aux besoins pour minimiser ces coûts.

Services Cloud

SaaS Software as a service		 Office 365	 zendesk	
FaaS Function as a service		 APACHE OpenWhisk		Google Cloud Functions
DBaaS Database as a service			 ORACLE DATA CLOUD	 CockroachDB
PaaS Platform as a service		 CLOUD FOUNDRY		 salesforce platform
STaaS Storage as a service		 OneDrive	 Dropbox	 Google Drive
IaaS Infrastructure as a service		 openstack.	 apachecloudstack open source cloud computing	 Google Compute Engine



Simple Storage Service

- Données
- Code chaîne de traitement des images
- Informations configuration EMR, bootstrapping
- Résultats



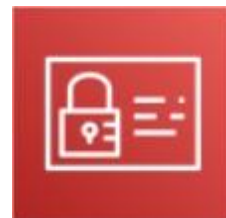
EMR



Elastic Map Reduce

- Gestion instances EC2 (Master & slaves)
- Installation packages complémentaires (Spark, Tensorflow, JupyterHub)

IAM



Identity and Access Manager

- Service de sécurité
- Gestion des accès

Avantages

- Essai gratuit + *Pay as you go*
- Régulation automatique puissance loué
- Résolution des problèmes techniques par AWS

EMR

Elastic Map Reduce

- Gestion instances EC2 (Master & slaves)
- Installation packages complémentaires (Spark, Tensorflow, JupyterHub)

Informations sur le cluster

ID de cluster

j-1ODUO7A2ZTDJK

Configuration de cluster

Groupes d'instances

Capacité

1 primaire(s) | 1 unité(s) principale(s) | 2 tâche(s)

Applications

Version d'Amazon EMR

emr-6.15.0

Applications installées

Hadoop 3.3.6, JupyterHub 1.5.0, Spark 3.4.1,
TensorFlow 2.11.0

Installation Bootstrap : action d'amorçage

```
#!/bin/bash
sudo python3 -m pip install -U setuptools
sudo python3 -m pip install -U pip
sudo python3 -m pip install wheel
sudo python3 -m pip install pillow
sudo python3 -m pip install pandas
sudo python3 -m pip install pyarrow
sudo python3 -m pip install boto3
sudo python3 -m pip install s3fs
sudo python3 -m pip install fsspec
sudo python3 -m pip install pyspark
sudo python3 -m pip install tensorflow
```

Paramétrisation du cluster:



Connexion via tunnel SSH

avec Putty



puis Proxy SwitchyOmega



Travail du script sur :





Contexte



Environnement Big Data



Traitement images



Architecture de développement



Conclusions



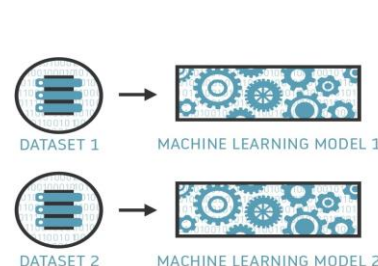
Annexes

Objectif : Moteur classification données

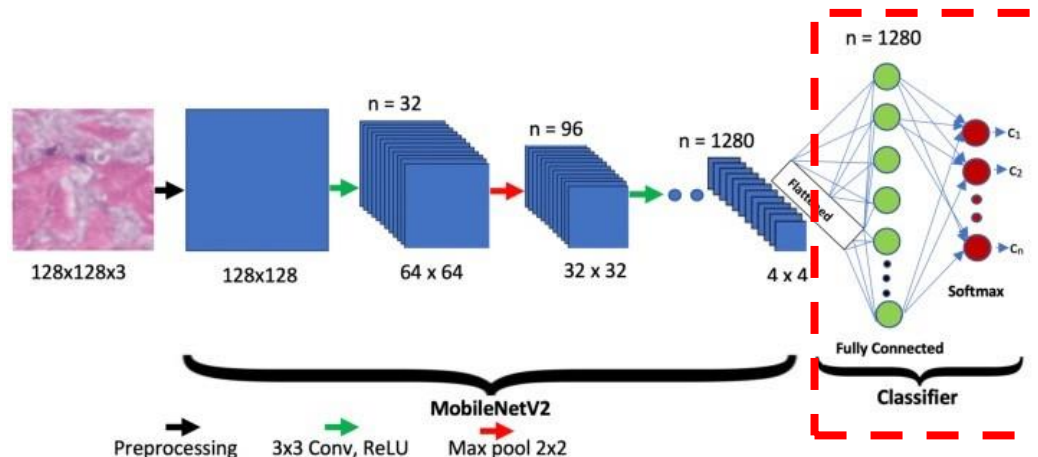
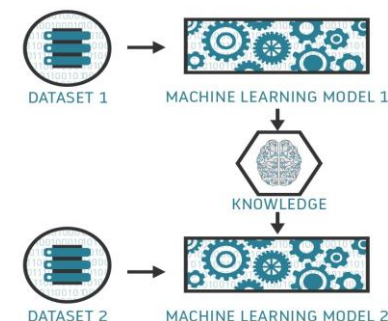
Solution : Transfer learning >> **MobileNetV2**

- Réseau de neurones CNN (Convolution Neural Network)
- Spécialisé en computer vision pour les systèmes embarqués
- On enlève l'avant dernière couche pour fitter sur nos données

TRADITIONAL MACHINE LEARNING



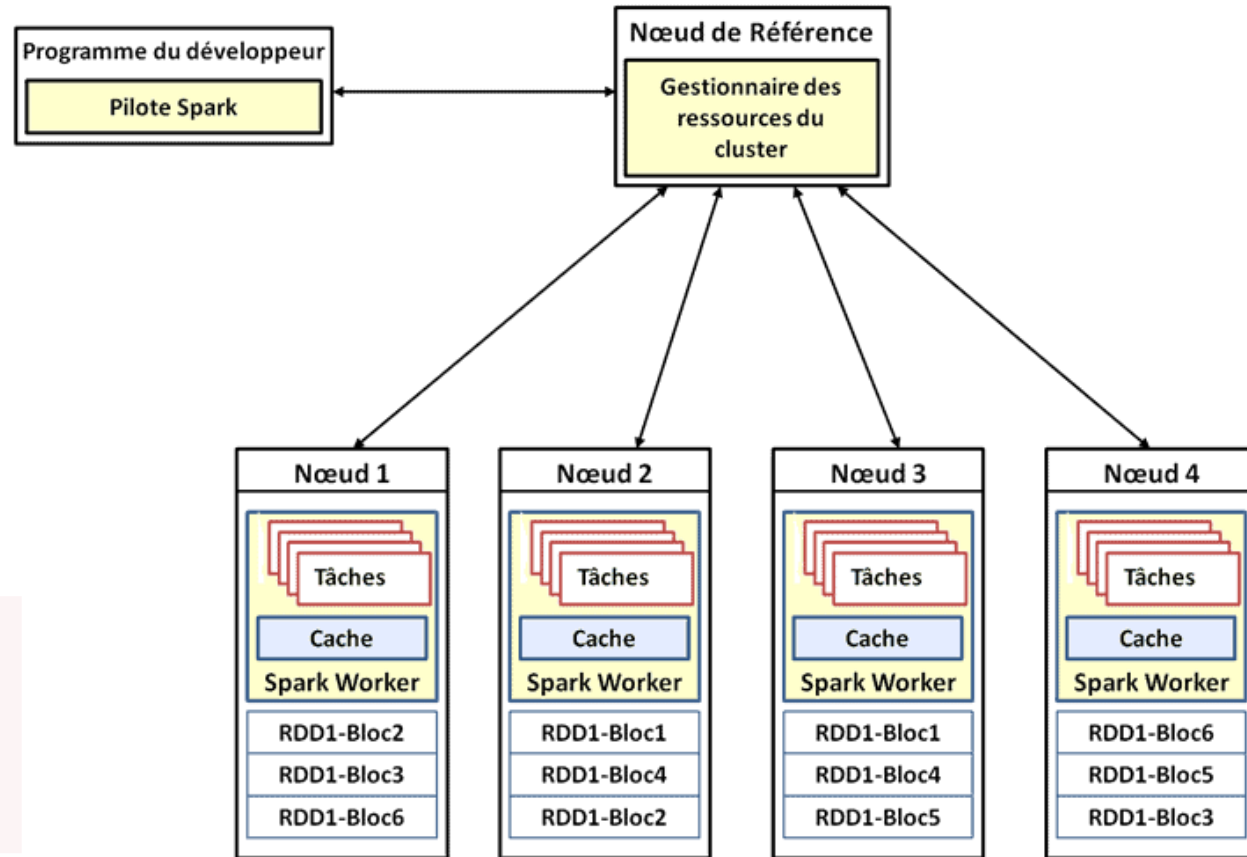
TRANSFER LEARNING



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Représentation (downscaling) dans sous-espace latent des images

Calcul distribué



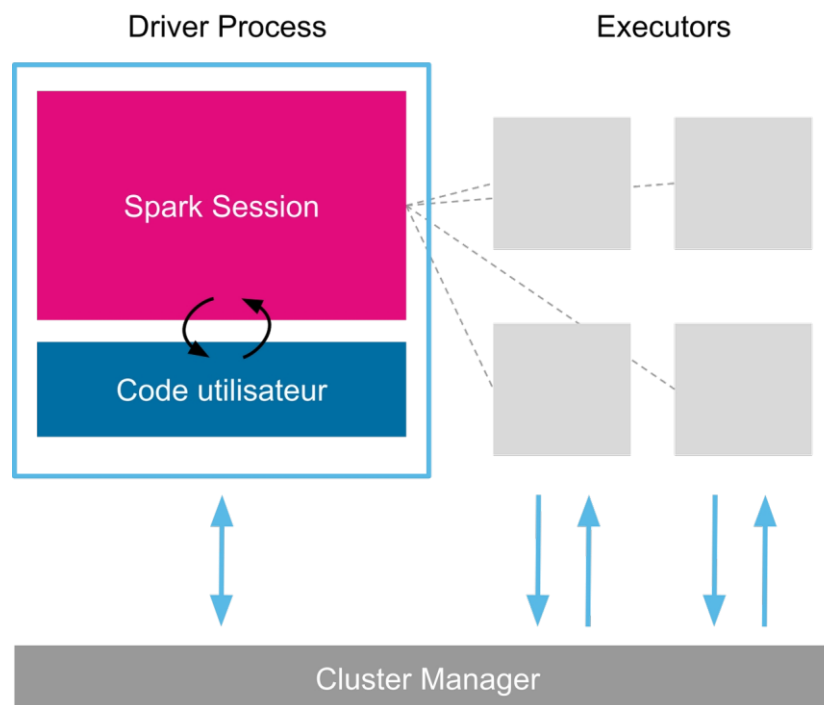
Avantages :

- Équilibrage de la charge
- Optimisation des transferts
- Scalabilité
- Tolérance aux pannes

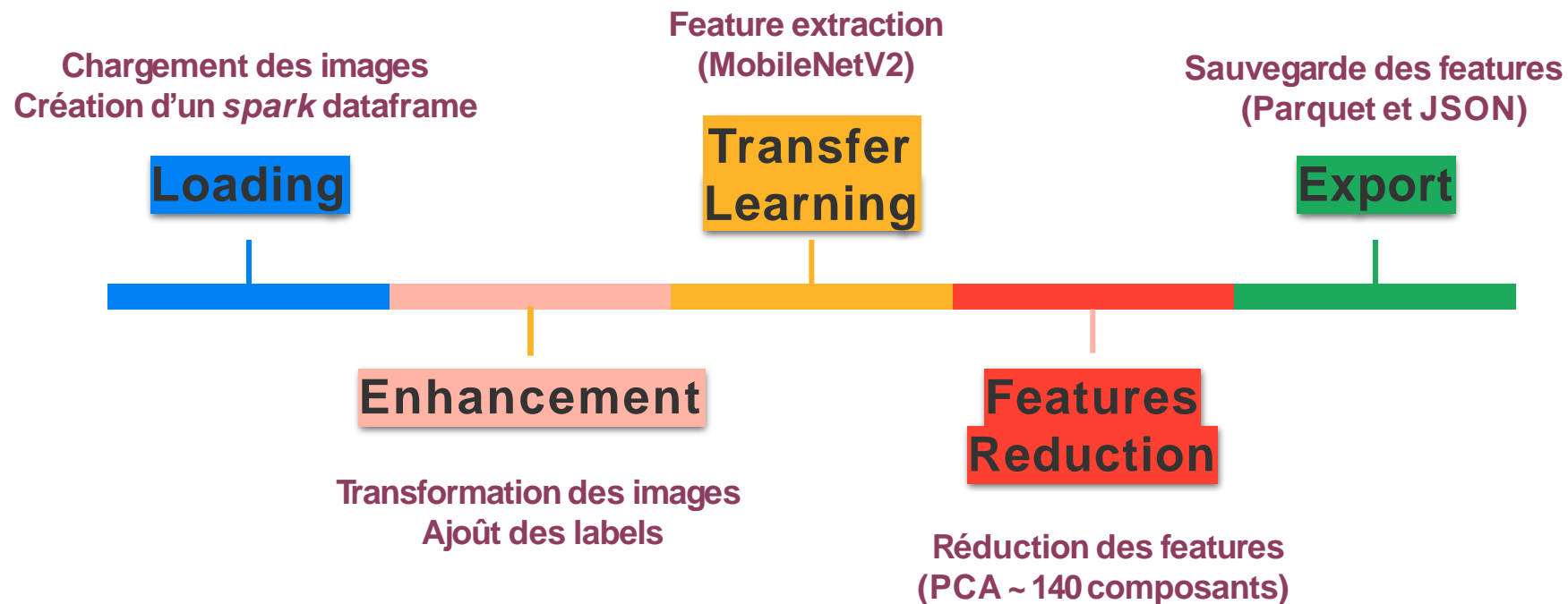
Objectif : solution performante et scalable, capable de traiter un gros volume de données

Utilisation de la technologie **Spark** pour distribuer les calculs sur plusieurs cœurs / machines

Les traitements sont donc développés en **pyspark**



Les étapes suivantes ont été implémentées :



Les étapes suivantes ont été implémentées :

1. Création d'une **SparkSession** (driver process) et la variable **sparkContext**

Jupyter notebook_cluster_fruits_oc8 Dernière Sauvegarde : il y a 3 heures (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help Non fiable Python 3 (ipykernel)

1.4. [NOTE](#)

Afin de limiter les coûts, le jeu de données a été restreint. Ici il ne sera utilisé que 4904 photos de 28 fruits différents

2. [Démarrage de la session Spark et importation des librairies](#)

2.1. [Démarrage de la session Spark](#)

Entrée [2]: `# L'exécution de cette cellule démarre l'application Spark`

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

Entrée [3]: `%info`

Current session configs: {'driverMemory': '1000M', 'executorCores': 2, 'proxyUser': 'jovyan', 'kind': 'pyspark'}

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
0	application_1705835948839_0001	pyspark	idle	Link	Link	None	✓

2. Charger les images et les associer aux images leur label

3. Définition des PATH pour le chargement des images et l'enregistrement des résultats

Définition des chemins d'accès pour les images et le fichier résultats dans s3:

```
Entrée [7]: PATH = 's3://oc8-data-mkucukba/data'
PATH_Data = PATH+'/Test1'
PATH_Result = PATH+'/Results'
print('PATH: '+\
      PATH+'\nPATH_Data: '+\
      PATH_Data+'\nPATH_Result: '+PATH_Result)
```

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

```
PATH:      s3://oc8-data-mkucukba/data
PATH_Data: s3://oc8-data-mkucukba/data/Test1
PATH_Result: s3://oc8-data-mkucukba/data/Results
```

4.1. Chargement des données

```
Entrée [8]: # Chargement des images avec l'extension .jpg sous format binaire présentes dans les répertoires et sous-répertoires
images = spark.read.format("binaryFile") \
    .option("pathGlobFilter", "*.jpg") \
    .option("recursiveFileLookup", "true") \
    .load(PATH_Data)
```

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

```
Entrée [9]: # Affichage de 5 images
images.show(5)
```

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

path	modificationTime	length	content
s3://oc8-data-mku...	2024-01-19 10:37:51	6328	[FF D8 FF E0 00 1...
s3://oc8-data-mku...	2024-01-19 10:37:48	6322	[FF D8 FF E0 00 1...
s3://oc8-data-mku...	2024-01-19 10:37:47	6308	[FF D8 FF E0 00 1...
s3://oc8-data-mku...	2024-01-19 10:38:15	6304	[FF D8 FF E0 00 1...
s3://oc8-data-mku...	2024-01-19 10:37:47	6300	[FF D8 FF E0 00 1...

only showing top 5 rows

3. Préparation du modèle. Importer le modèle **MobileNetV2**

Créer un **nouveau modèle dépourvu de la dernière couche** de MobileNetV2

Broadcast des “weights” du modèle

4.2. Préparation du modèle

Nous appliquons également ici, une diffusion des poids du modèles (broadcasting) à travers les différents noeuds de calcul. Cela permet l'accélération de l'entraînement du modèle sur de grands ensembles de données.

```
Entrée [12]: # Création du modèle MobileNetV2 avec l'ensemble des couches :
base_model = MobileNetV2(weights='imagenet',
                        include_top=True,
                        input_shape=(224, 224, 3),
                        )

# Création du modèle spécifique (retrait de la dernière couche):
model = Model(inputs=base_model.input,
              outputs=base_model.layers[-2].output)

# Diffusion des poids du modèle :
broadcast_weights = sc.broadcast(model.get_weights())

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

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_kernels_1.0_224.h5
14536120/14536120 [=====] - 1s 0us/step
```

```
Entrée [13]: # Résumé du modèle
model.summary()

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

Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	input_1[0][0]
Conv1 (Conv2D)	(None, 112, 112, 32)	864	input_1[0][0]
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 112, 112, 32)	0	bn_Conv1[0][0]
expanded_conv_depthwise (DepthwiseConv2D)	(None, 112, 112, 32)	288	Conv1_relu[0][0]
•			
•			
•			
block_16_project (Conv2D)	(None, 7, 7, 320)	307200	block_16_depthwise_relu[0][0]
block_16_project_bn (BatchNormalization)	(None, 7, 7, 320)	1280	block_16_project[0][0]
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409600	block_16_project_bn[0][0]
Conv_1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0	out_relu[0][0]

```

Total params: 2,257,984
Trainable params: 2,223,872
Non-trainable params: 34,112
```

4. Définition du processus de chargement des images et application de leur featurisation

Redimensionner les images pour qu'elles soient compatibles avec le modèle

Extraction de features à travers l'utilisation de pandas UDF

```
In [23]: # Resize images to 224x224
def preprocess(content):
    """
    Preprocesses raw image bytes for prediction.
    """
    img = Image.open(io.BytesIO(content)).resize([224, 224])
    arr = img_to_array(img)
    return preprocess_input(arr)

# Featurize images and return a series of vectors (flattened tensors)
def featurize_series(model, content_series):
    """
    Featurize a pd.Series of raw images using the input model.
    :return: a pd.Series of image features
    """
    input = np.stack(content_series.map(preprocess))
    preds = model.predict(input)
    # For some layers, output features will be multi-dimensional tensors.
    # We flatten the feature tensors to vectors for easier storage in Spark DataFrames.
    output = [p.flatten() for p in preds]
    return pd.Series(output)

@pandas_udf('array<float>', PandasUDFType.SCALAR_ITER)
def featurize_udf(content_series_iter):
    """
    This method is a Scalar Iterator pandas UDF wrapping our featurization function.
    The decorator specifies that this returns a Spark DataFrame column of type ArrayType(FloatType).

    :param: content_series_iter, This argument is an iterator over batches of data, where each batch
           is a pandas Series of image data.
    """
    # With Scalar Iterator pandas UDFs, we can load the model once and then re-use it
    # for multiple data batches. This amortizes the overhead of loading big models.
    model = model_fn()
    for content_series in content_series_iter:
        yield featurize_series(model, content_series)
```

Pandas UDF :

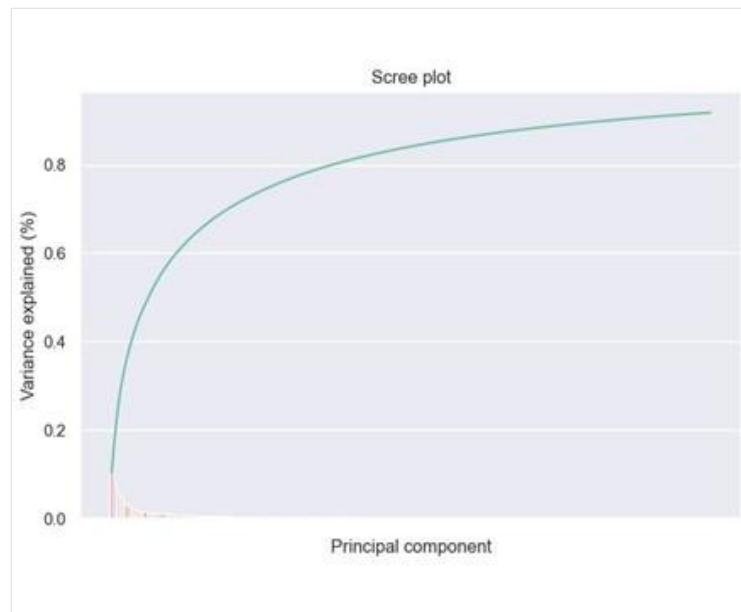
(User-Defined Function) dans Spark sont des fonctions définies par l'utilisateur qui utilisent la bibliothèque Pandas pour effectuer des opérations sur des données dans Spark.

path	label	features
file:/home/raquel...	Raspberry	[0.29254088, 1.06...
file:/home/raquel...	Strawberry	[2.20994, 0.09814...
file:/home/raquel...	Strawberry	[1.6356167, 0.0, ...]
file:/home/raquel...	Peach	[0.13757613, 0.0, ...]
file:/home/raquel...	Tomato not Ripened	[0.0, 0.4384215, ...]

only showing top 5 rows

5. Réduction des dimensions via un PCA

Recherche nombre optimal composantes expliquant 95 % variance



5.2. Réduction dimensionnelle

A l'aide d'une PCA avec 138 composantes.

Les 138 composantes ayant été définies lors du test local, permettant d'atteindre 95% de la variance expliquée.

```
Entrée [19]: # Création d'une fonction de conversion de la colonne 'features' en vecteur :
features_to_vector_udf = udf(lambda arr: Vectors.dense(arr), VectorUDT())

# Application de la fonction au DataFrame et création d'une nouvelle colonne :
features_df = features_df.withColumn("features_vector", features_to_vector_udf("features"))

# Création d'un modèle PCA avec les 138 composantes principales pour atteindre 95% de la variance :
pca = PCA(k=138, inputCol="features_vector", outputCol="vectorized_components_pca_features")

# Application de la PCA sur le DataFrame :
pca = pca.fit(features_df)
features_df = pca.transform(features_df)
```

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

```
Entrée [20]: # Affichage des 5 premières lignes :
features_df.show(5, truncate=True)
```

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

path	label	features	features_vector	vectorized_components_pca_features
s3://oc8-data-mku...	Apple Golden 1	[0.007712948, 6.0...	[0.00771294813603...	[0.62169322696219...
s3://oc8-data-mku...	Apple Golden 1	[0.0, 0.073598616...	[0.0, 0.0735986158...	[0.63665490126447...
s3://oc8-data-mku...	Pear Red	[0.0, 0.0, 0.0, 0.0...	[0.0, 0.0, 0.0, 0.0...	[-3.3915057090228...
s3://oc8-data-mku...	Pear Red	[5.2549403E-6, 0.0...	[5.25494033354334...	[-3.1520013943988...
s3://oc8-data-mku...	Pear 2	[0.0, 0.0, 0.0, 0.0...	[0.0, 0.0, 0.0, 0.0...	[-2.5845975905324...

only showing top 5 rows

6. Sauvegarde du résultat de nos actions

6.3. Sauvegarde des résultats

Sauvegarde du DataFrame au format CSV dans le bucket s3

```
Entrée [27]: # Enregistrement du DataFrame en tant que fichier CSV sur S3
df.to_csv(PATH_Result + '/df_results_aws_emr.csv', index=False)

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

```
Entrée [28]: # Vérification de l'enregistrement :
df = pd.read_csv(PATH_Result + '/df_results_aws_emr.csv')

# Affichage des 5 premières lignes :
df.head()

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

	path	...	pca_feature_138
0	s3://oc8-data-mkucukba/data/Test1/Apple Golden...	...	0.742611
1	s3://oc8-data-mkucukba/data/Test1/Pear Red/r2_...	...	-0.343838
2	s3://oc8-data-mkucukba/data/Test1/Pear Red/r2_...	...	0.888846
3	s3://oc8-data-mkucukba/data/Test1/Pear Red/r2_...	...	0.889252
4	s3://oc8-data-mkucukba/data/Test1/Pear 2/r_264...	...	0.722700

[5 rows x 140 columns]



Contexte



Environnement Big Data



Traitement images



Architecture de développement



Conclusions



Annexes

Nom

Fruits_P8

Version Amazon EMR


Info

Une version contient un ensemble d'applications susceptibles d'être installées sur votre cluster.


emr-6.15.0

Offre d'applications


Spark Interactive




Core Hadoop




Flink




HBase




Presto



Trino



Custom



☐ Flink 1.17.1

☐ HCatalog 3.1.3

☐ Hue 4.11.0

☐ Livy 0.7.1

☐ Phoenix 5.1.3

☒ Spark 3.4.1

☐ Tez 0.10.2

☐ ZooKeeper 3.5.10

☐ Ganglia 3.7.2

☒ Hadoop 3.3.6

☐ JupyterEnterpriseGateway 2.6.0

☐ MXNet 1.9.1

☐ Pig 0.17.0

☐ Sqoop 1.4.7

☐ Trino 426

☐ HBase 2.4.17

☐ Hive 3.1.3

☒ JupyterHub 1.5.0

☐ Oozie 5.2.1

☐ Presto 0.283

☒ TensorFlow 2.11.0

☐ Zeppelin 0.10.1

Paramètres du catalogue de données AWS Glue

Utilisez le catalogue de données AWS Glue pour fournir un metastore externe à votre application.

☐ Utiliser pour les métadonnées de table Spark

Options du système d'exploitation

Info

☒ Version Amazon Linux :

☐ Amazon Machine Image (AMI) personnalisée

☒ Appliquez automatiquement les dernières mises à jour Amazon Linux

▼ Actions d'amorçage – facultatif (1)


Info

Supprimer

Modifier

Ajouter

Utilisez les actions d'amorçage pour installer des logiciels ou personnaliser la configuration de votre instance.

	Nom	Emplacement Amazon S3	Arguments
<input type="radio"/>	Booty	s3://oc8-data-mkucukba/bootstrap-emr.sh	 Boot de cluster (master/slave) nodes

Dimensionnement et mise en service du cluster

Info

Configurez des configurations de dimensionnement et de provisionnement pour les groupes de nœuds principaux et de tâches de votre cluster.

Choisir une option

☒ Définir manuellement la taille du cluster

Utilisez cette option si vous connaissez vos modèles de charge de travail à l'avance.

☐ Utiliser la mise à l'échelle gérée par EMR

Surveillez les principales métriques de charges de travail afin qu'EMR puisse optimiser la taille du cluster et l'utilisation des ressources.

☐ Utiliser un autoscaling personnalisée

Pour dimensionner de manière programmatique les unités principales et les nœuds de tâches, créez des politiques d'autoscaling personnalisées.

Configuration de mise en service

Définissez la taille de votre noyau et tâchegroupes d'instance. Amazon EMR tente de fournir cette capacité lorsque vous lancez votre cluster.

Nom	Type d'instance	Taille de l'instance(s)	Utiliser l'option d'achat Spot
Unité principale	m5.xlarge	<input type="text" value="1"/>	<input type="checkbox"/>
Slave	m5.xlarge	<input type="text" value="2"/>	<input type="checkbox"/>

oc8-data-mkucukba

Info

Objets

Propriétés

Autorisations

Métriques

Gestion

Points d'accès

Objets (3)

Info

🔄

Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Actions

Créer un dossier

Charger

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 personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. En savoir plus

Rechercher des objets en fonction du préfixe

< 1 > ⚙

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	bootstrap-emr.sh	sh	21 Jan 2024 11:30:16 AM CET	405.0 o	Standard
<input type="checkbox"/>	data/	Dossier	-	-	-
<input type="checkbox"/>	jupyter/	Dossier	-	-	-

Points d'accès de l'objet Lambda

Points d'accès multi-région

Opérations par lot

IAM Access Analyzer pour S3

Paramètres de blocage de l'accès public pour ce compte

▼ Storage Lens

Tableaux de bord

Groupes Storage Lens

Paramètres AWS Organizations

Fonctionnalité spot

► AWS Marketplace pour S3

Objets (22)

Info

🔄

Copier l'URI S3

Copier l'URL

Télécharger

Ouvrir

Supprimer

Actions

Créer un dossier

Charger

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 personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. En savoir plus

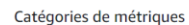
Rechercher des objets en fonction du préfixe

< 1 > ⚙

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	_SUCCESS	-	21 Jan 2024 12:51:50 PM CET	0 o	Standard
<input type="checkbox"/>	df_results_cloud.csv	csv	21 Jan 2024 12:52:56 PM CET	7.3 Mo	Standard
<input type="checkbox"/>	part-00000-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:06 PM CET	158.3 Ko	Standard
<input type="checkbox"/>	part-00001-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:06 PM CET	134.0 Ko	Standard
<input type="checkbox"/>	part-00002-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:10 PM CET	130.7 Ko	Standard
<input type="checkbox"/>	part-00003-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:09 PM CET	130.1 Ko	Standard
<input type="checkbox"/>	part-00004-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:14 PM CET	135.0 Ko	Standard
<input type="checkbox"/>	part-00005-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:15 PM CET	137.2 Ko	Standard
<input type="checkbox"/>	part-00006-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:19 PM CET	135.1 Ko	Standard
<input type="checkbox"/>	part-00007-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:20 PM CET	134.0 Ko	Standard

24

[illegible]



```
Windows PowerShell
PS C:\Users\mkucukba> aws s3 ls s3://oc8-data-mkucukba/data/Results/
PRE Results/
PS C:\Users\mkucukba> aws s3 ls s3://oc8-data-mkucukba/data/Results/
2024-01-21 12:51:50 0 _SUCCESS
2024-01-21 12:52:56 7637264 df_results_cloud.csv
2024-01-21 12:51:06 141665 part-00000-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:06 137238 part-00001-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:10 133831 part-00002-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:09 133271 part-00003-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:14 138276 part-00004-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:15 140523 part-00005-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:19 138346 part-00006-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:20 137210 part-00007-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:24 134961 part-00008-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:24 140002 part-00009-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:29 141161 part-00010-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:34 141102 part-00011-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:33 141093 part-00012-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:34 139454 part-00013-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:44 143412 part-00014-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:46 142818 part-00015-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:42 141681 part-00016-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:43 140568 part-00017-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:49 141086 part-00018-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
2024-01-21 12:51:49 143367 part-00019-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet
PS C:\Users\mkucukba>
```

Copier l'URI S3

Propriétés

(22) Info

Copier l'URI S3 Copier l'URL Télécharger Ouvrir Supprimer Actions Créer un dossier Charger

sort les entités fondamentales stockées dans Amazon S3. Vous pouvez utiliser l'interface Amazon S3 pour obtenir une liste de tous les objets de votre compartiment. Pour que d'autres personnes puissent accéder à vos objets, vous devez leur accorder explicitement des autorisations. En savoir plus

Rechercher des objets en fonction du préfixe

Nom	Type	Dernière modification	Taille	Classe de stockage
_SUCCESS	-	21 Jan 2024 12:51:50 PM CET	0 o	Standard
df_results_cloud.csv	csv	21 Jan 2024 12:52:56 PM CET	7.3 Mo	Standard
part-00000-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:06 PM CET	138.3 Ko	Standard
part-00001-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:06 PM CET	134.0 Ko	Standard
part-00002-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:10 PM CET	130.7 Ko	Standard
part-00003-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:09 PM CET	130.1 Ko	Standard
part-00004-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:14 PM CET	135.0 Ko	Standard
part-00005-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:15 PM CET	137.2 Ko	Standard
part-00006-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:19 PM CET	135.1 Ko	Standard
part-00007-48782f06-d65f-45e9-a4c9-4d9821dfcdea-c000.snappy.parquet	parquet	21 Jan 2024 12:51:20 PM CET	134.0 Ko	Standard



Contexte



Environnement Big Data



Traitement images



Architecture de développement



Conclusions



Annexes

- ✓ Premier approche **traitement des images révisé**
- ✓ Implémentation réduction des dimensions via **PCA** (de 1280 à 128)
- ✓ **Chaîne de traitement des images validée**
- ✓ Architecture Big Data : **EMR, S3, IAM**
 - ✓ 15 minutes pour instanciation des clusters
 - ✓ Traitement des images rapide
 - ✓ Les coûts évoluent en parallèle au volume de données
- ✓ Les briques mises en place peuvent évoluer en fonction du volume de données à traiter en augmentant le nombre de workers et / ou leur configuration



Contexte



Environnement Big Data



Traitement images



Architecture de développement



Conclusions



Annexes

AWS: EC2

EC2 : Elastic Compute Cloud

Ce service permet de gérer des serveurs sous forme de machines virtuelles dans le cloud. En gros, vous pouvez lancer des serveurs et faire ce que vous voulez avec. Vous avez accès à la ligne de commande, donc vous pouvez les piloter à distance.



Amazon Elastic Compute Cloud (Amazon EC2)

New EC2 Experience

Groupes de sécurité (1/6) Informations

Filter les groupes de sécurité

	Name	ID du groupe de sécu...	Nom du groupe de sécurité	ID de VPC	Description	Propriétaire	Nom
<input checked="" type="checkbox"/>	-	sg-02e3d9b9055f75932	ElasticMapReduce-master	vpc-889e88e1	Master group for Elasti...	553358241472	9 Ent
<input type="checkbox"/>	-	sg-0513291aefc2bde70	ElasticMapReduceEditors-Editor	vpc-889e88e1	Security group that all...	553358241472	0 Ent
<input type="checkbox"/>	-	sg-039e93236e94060bd	ElasticMapReduce-slave	vpc-889e88e1	Slave group for Elastic ...	553358241472	7 Ent
<input type="checkbox"/>	-	sg-0471903c18153bde9	GroupeDeSécurité	vpc-889e88e1	Pour Les notebooks Ju...	553358241472	3 Ent
<input type="checkbox"/>	-	sg-049a39d2bc9917cff	ElasticMapReduceEditors-Liv...	vpc-889e88e1	Security group that all...	553358241472	1 Ent
<input type="checkbox"/>	-	sg-c83149a7	default	vpc-889e88e1	default VPC security gr...	553358241472	1 Ent

AWS: S3

S3 : Simple Storage Service

Amazon S3 (Simple Storage Service) est un service de stockage et de distribution de fichiers. C'est une sorte d'entrepôt de fichiers à très bas coût qui garantit de ne jamais perdre vos données. Utilisez-le pour faire télécharger des fichiers sur votre site, ou pour y stocker des images.



Amazon Simple Storage Service (Amazon S3)

AWS: IAM

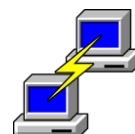
IAM : Gestion des identités

AWS IAM (Identity and Access Management) est LE service de sécurité par excellence. On y définit les règles d'accès des utilisateurs aux services de la galaxie AWS. Si vous souhaitez autoriser votre comptable à télécharger la facture mais pas à éteindre vos serveurs, c'est là que ça se passe.

Service de sécurité Gestion des accès



AWS Identity and Access Management (IAM)



Configuration de sécurité et paire de clés EC2 - facultatif [Info](#)

Configuration de sécurité
Sélectionnez les paramètres de chiffrement, d'authentification, d'autorisation et de service de métadonnées d'instance de votre cluster.

Paire de clés Amazon EC2 pour SSH sur le cluster [Info](#)

PuTTY Configuration

Category:

- Session
 - Logging
- Terminal
 - Keyboard
 - Bell
 - Features
- Window
 - Appearance
 - Behaviour
 - Translation
- Selection
 - Colours
- Connection
 - Data
 - Proxy
 - SSH
 - Kex
 - Host keys
 - Cipher
 - Auth
 - Credentials
 - GSSAPI
 - TTY
 - X11
 - Tunnels

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

Connection type: ☒ SSH ☐ Serial ☐ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:

☐ Always ☐ Never ☒ Only on clean exit