



Projet P8 OC DS : Déployez un modèle dans le cloud

Candidat : David Capelle

Date de soutenance : 04/02/2022

Evaluateur : Emmanuel Haback Li

Mentor : Nicolas Michel



Plan de la soutenance

- Présentation du projet
- Présentation du jeu de données
- Présentation générale du Big Data et Apache Spark
- Panorama de l'architecture Big Data AWS pour le machine learning
- Présentation des composants AWS utilisées dans le projet
- Présentation de la chaîne de traitement
- Conclusion et recommandations



Présentation du projet

- **Contexte :**

- la société **Fruits!** propose des solutions innovantes pour la récolte des fruits (comme des robots cueilleurs)
- volonté de proposer une application mobile permettant de reconnaître les fruits à partir d'une photo

- **Objectifs :**

- développement d'une chaîne de traitement (couvrant pre-processing et réduction de dimensions) dans un environnement Big Data sur le cloud permettant :
 - la prise en compte des considérations de scaling
 - la mise à disposition de configurations serveurs à la demande

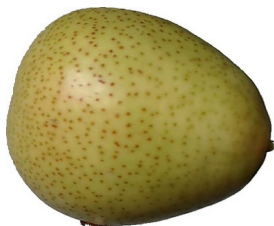
Présentation du jeu de données

- Données en chiffres sur la base de données « Fruits 360 » :

- 90483 images représentant des fruits et légumes, répartis en 131 catégories
- jeu de données d'entraînement avec 67692 images et 22688 pour le jeu de test

- Caractéristiques des images :

- images au format jpeg, classées par catégories
- présence de plusieurs variétés par fruit/légume
- Rotation du fruit sur 3 axes, à 360°





Pourquoi le Big Data ?

- **Définition** : stratégies et technologies mises en œuvre pour rassembler, organiser, stocker et analyser de vastes ensembles de données.
- **Utilité du Big Data** : à partir du moment où la quantité de données excède la faculté d'une machine à les stocker et les analyser en un temps acceptable.
- Outre le stockage des données, nécessité de distribuer les calculs d'analyse sur plusieurs machines avec ses problématiques (stratégies de distribution des données, maîtrise des coûts de l'architecture,...).



Big Data - Les enjeux des 3V

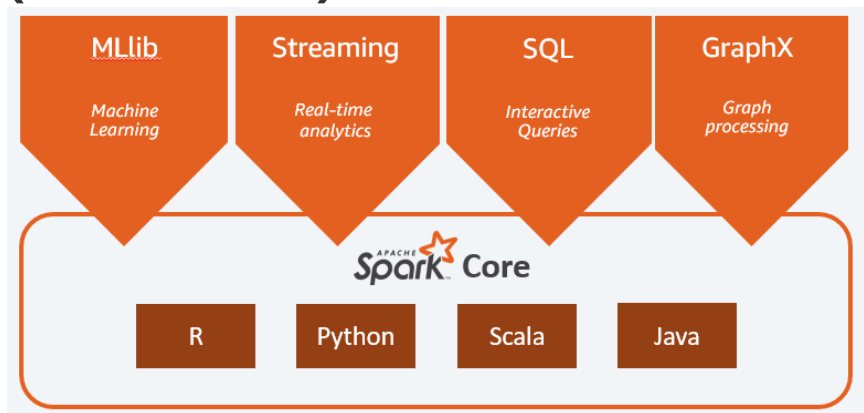
- **Volume (volume de données)** : la quantité de données générées est en constante augmentation. L'architecture Big Data doit être capable de traiter un nombre conséquent de données et d'informations.
- **Vélocité (traitement en temps réel)** : les données sont générées plus rapidement et dans des temps beaucoup plus courts. Les entreprises sont obligées de les collecter et de les partager en temps réel.
- **Variété (hétérogénéité des données)** : les types de données et leurs sources sont de plus en plus diversifiés (géolocalisation, réseaux sociaux, texte, web, images, vidéos, tweets, audio,...) avec des données structurées/non structurées.

Nécessité de mettre en place des outils de stockage/base de données appropriés.

Apache Spark



- Framework de traitement distribué open source utilisé pour les traitements Big Data
- Utilisation de la mise en cache en mémoire et l'exécution optimisée des requêtes
- API de développement en Java, Scala, Python (PySpark), R et l'accès à différents module (workloads)



Architecture Big Data AWS pour ML

Workflow Services



Amazon SageMaker



Deep Learning AMI



Deep Learning Containers



AWS Batch



AWS ParallelCluster



Elastic Kubernetes Service



Elastic Container Service



Amazon EMR

Frameworks



TensorFlow

PyTorch



Keras



Compute, Networking, and Storage



EC2 P4 instances



EC2 P3 instances



EC2 G4 instances



EC2 Inf1 instances



Elastic Inference



AWS Outposts



Elastic Fabric Adapter



Amazon S3



Amazon EBS

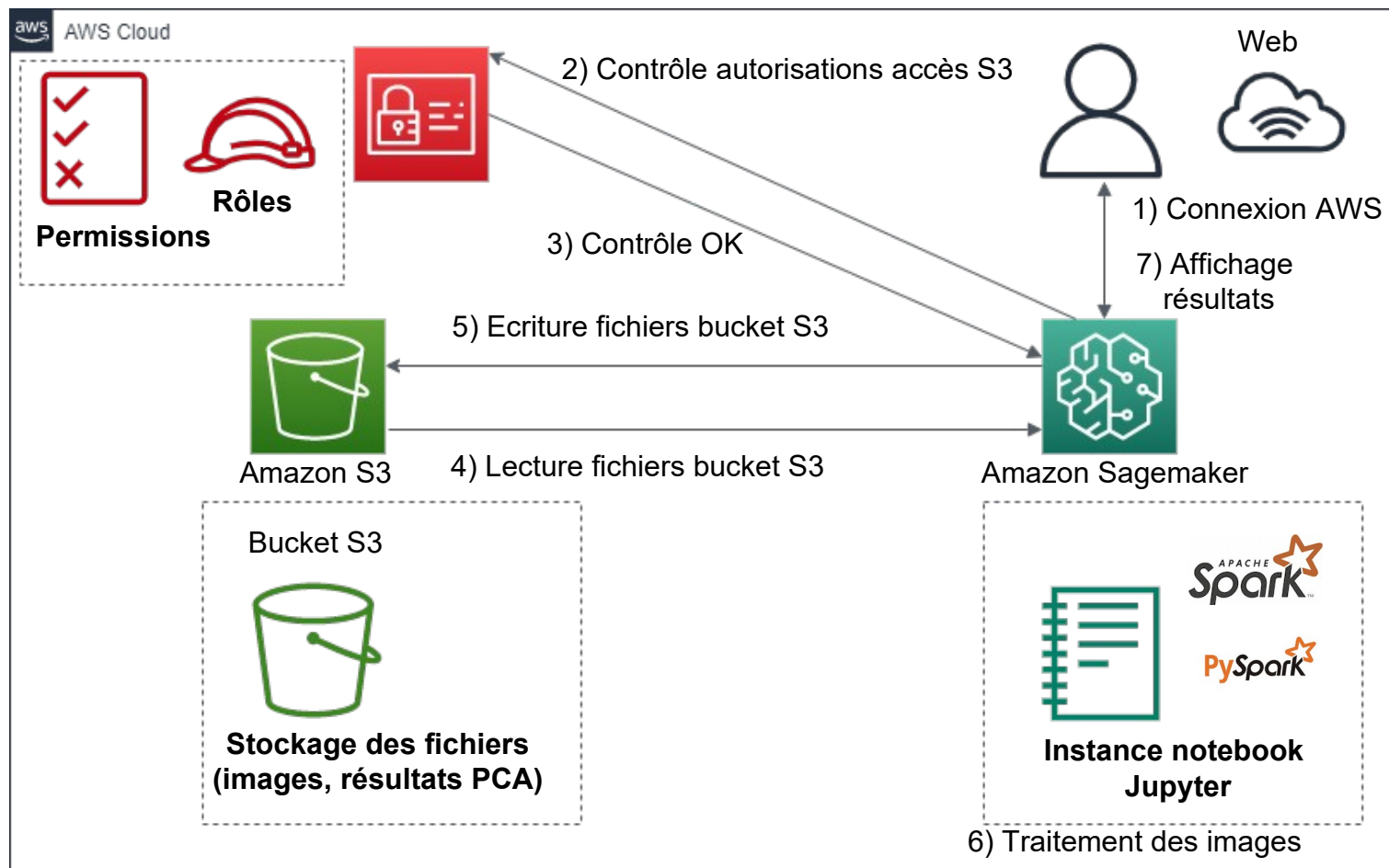


Amazon FSx



Amazon EFS

Composants Big Data AWS du projet





AWS IAM (Identity and Access Management)

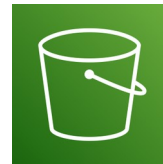
- Permet de gérer les services AWS accessibles à un compte IAM (utilisateur).
- Distinction utilisateur racine / utilisateur IAM
- Création de rôles à partir de stratégies d'autorisations
- Possibilité de donner des droits très fins à l'utilisateur pour l'accès à une ressource AWS.
- Ce que n'est pas IAM :
 - Ne permet pas de gérer les permissions des applications (application web ou de ML,...)
 - N'est pas un OS de management d'identités type LDAP ou active directory

AWS Sagemaker



- Service AWS pour la préparation, la création, l'entraînement et le déploiement de modèles ML.
- Composants :
 - préparation des données (Data wrangler, Ground Truth, Feature Store,...)
 - développement de modèles (Studio, notebook Jupyter,...)
 - entraînement de modèles (Debugger, bibliothèque d'entraînement distribué,...)
 - déploiement (Pipeline, Model monitor,...)
- Avantages :
 - facilité d'industrialisation
 - modèles pré-packagés AWS et customisation
 - scalabilité des modèles
 - monitoring de modèles en production,...

AWS S3 (Simple Storage Service)



- Service de stockage des données
- Organisation des données (images, fichiers texte, ...) sous forme d'objets dans des buckets (compartiments)
- Gestion des accès aux buckets
- Différents types de stockage (standard, Standard - accès peu fréquent, Glacier,...)
- Avantages :
 - gestion fine des droits d'accès aux objets du bucket.
 - versioning des objets
 - élasticité (pas de limite en termes de capacité de stockage)
 - disponibilité (réduction du risque de pannes)

Etapes de la chaîne de traitement

Pré-requis – Création d'un utilisateur IAM

- Création d'un utilisateur IAM (david_oc) à partir de l'utilisateur racine
- Création d'un rôle IAM pour l'accès à S3 depuis Sagemaker basé sur la stratégie « AmazonSageMakerFullAccess »
- Récupération de la clé d'accès AWS et de la clé secrète.

Informations d'identification de connexion

Récapitulatif • Lien de connexion à la console : <https://863289295452.signin.aws.amazon.com/console>

Mot de passe de la console Activé (dernière connexion à Hier) | [Gestion](#)

Appareil MFA attribué Non attribué | [Gestion](#)

Certificats de signature Aucun

Clés d'accès

Utilisez des clés d'accès pour effectuer des appels par programmation vers AWS à partir de l'interface de ligne de commande AWS (AWS CLI), des outils pour PowerShell, des kits SDK AWS ou des appels d'API AWS directs. Vous pouvez disposer d'un maximum de deux clés d'accès (actives ou inactives) à la fois.

Pour votre protection, vous ne devez jamais partager vos clés secrètes avec quiconque. Comme bonne pratique, nous recommandons une rotation fréquente des clés.

La clé secrète peut être consultée ou téléchargée uniquement au moment de la création. Veuillez créer une clé d'accès si vous avez mal placé votre clé secrète existante. [En savoir plus](#)

[Créer une clé d'accès](#)

ID de clé d'accès	Créé	Dernière utilisation	Statut	
AKIA4SAANIJOFGBBCGUH	2022-01-26 17:00 UTC+0100	2022-01-30 15:40 UTC+0100 avec s3 dans us-...	Actif	Rendre inactif x

ARN de rôle am:aws:iam::863289295452:role/AWSGlueServiceSageMakerNotebookRole-Default

Description du rôle Allows SageMaker notebook instances, training jobs, and models to access S3, ECR, and CloudWatch on your behalf. | [Modifier](#)

ARN des profils d'instance

Chemin /

Heure de création 2022-01-26 16:15 UTC+0100

Dernière activité 2022-02-01 09:11 UTC+0100 (Aujourd'hui)

Durée maximale de la session 1 heure [Modifier](#)

Autorisations Relations d'approbation Balises Access Advisor Révoquer les sessions

▼ Permissions policies (1 stratégie appliquée)

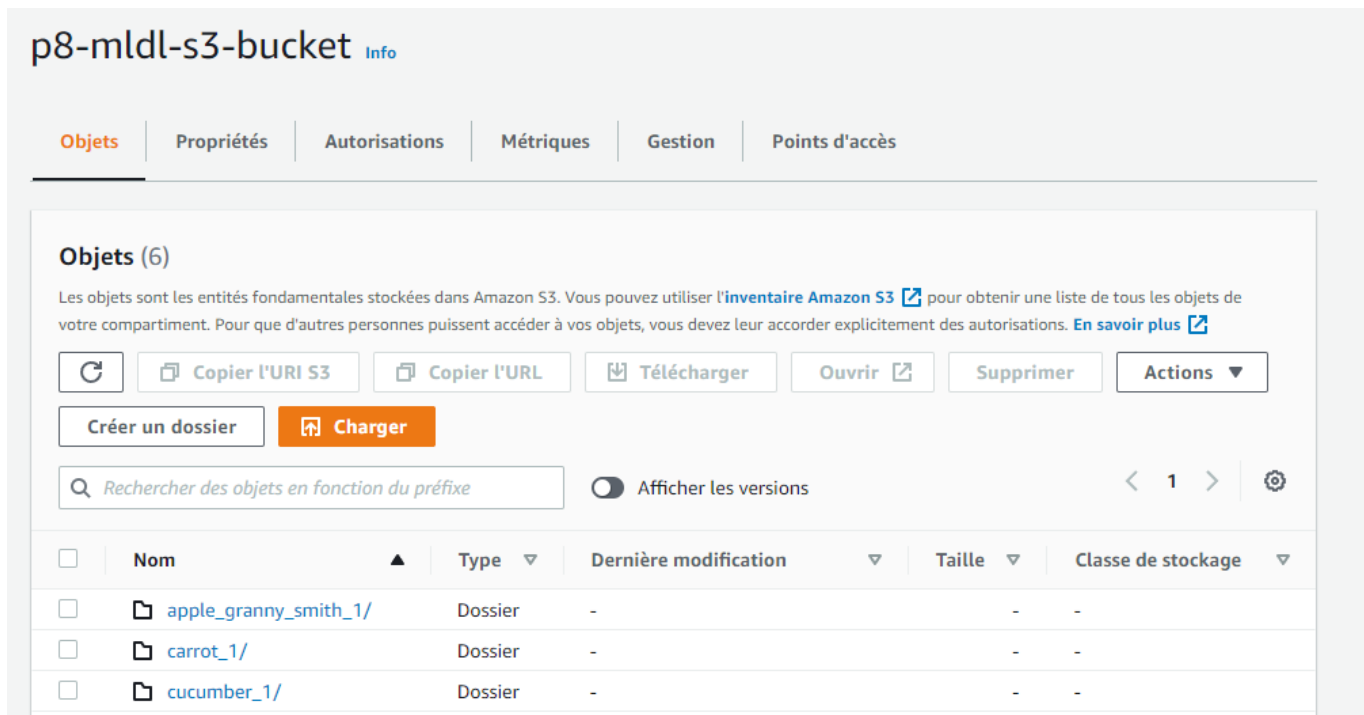
[Attacher des stratégies](#) [Ajouter une stratégie en ligne](#)

Nom de la stratégie ▼	Type de stratégie ▼	
AmazonSageMakerFullAccess	Stratégie gérée par AWS	x

Etapes de la chaîne de traitement

Pré-requis – Création d'un bucket S3

- Création d'un bucket S3 (p8-mldl-s3-bucket)
- Chargement des images depuis un répertoire en local



The screenshot shows the Amazon S3 console interface for a bucket named 'p8-mldl-s3-bucket'. The 'Objets' tab is selected, displaying a list of 6 objects. The interface includes navigation tabs, action buttons, a search bar, and a table of objects.

Objets (6)

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](#)

Buttons: [Copier l'URI S3](#), [Copier l'URL](#), [Télécharger](#), [Ouvrir](#), [Supprimer](#), [Actions](#)

Buttons: [Créer un dossier](#), [Charger](#)

Search: Afficher les versions

<input type="checkbox"/>	Nom	Type	Dernière modification	Taille	Classe de stockage
<input type="checkbox"/>	apple_granny_smith_1/	Dossier	-	-	-
<input type="checkbox"/>	carrot_1/	Dossier	-	-	-
<input type="checkbox"/>	cucumber_1/	Dossier	-	-	-

Etapes de la chaîne de traitement

Pré-requis – Création instance notebook Jupyter - Sagemaker

- Création d'une instance notebook Jupyter dans Sagemaker de type ml.t2.xlarge reliée à S3
- Autorisations : application du rôle créé dans IAM pour accès S3 depuis Sagemaker

Paramètres d'instances de blocs-notes

Nom de l'instance de bloc-notes

SageMakerInstance-p8-oc

Maximum de 63 caractères alphanumériques. Puis incluent les traits d'union (-), mais pas d'espaces. Doit être unique au sein de votre compte dans une région AWS.

Type d'instance de bloc-notes

ml.t2.xlarge

Prédiction élastique [En savoir plus](#)

aucun

ⓘ L'instance de bloc-notes Amazon SageMaker met fin à sa prise en charge standard sur l'AMI Amazon Linux (AL1). [En savoir plus](#)

Identifiant de plateforme [En savoir plus](#)

notebook-al1-v1

► Configuration supplémentaire

Autorisations et chiffrement

Rôle IAM

Les instances de bloc-notes nécessitent des autorisations pour appeler d'autres services, y compris SageMaker et S3. Choisissez un rôle ou laissez-nous en créer un avec la **AmazonSageMakerFullAccess** stratégie IAM attachée.

AWSGlueServiceSageMakerNotebookRole-Default

Accès racine - *facultatif*

- ☒ Enable (Activer): accorder aux utilisateurs l'accès racine au bloc-notes
- ☐ Disable (Désactiver): ne pas accorder aux utilisateurs l'accès racine au bloc-notes
- Les configurations de cycle de vie ont toujours un accès racine.

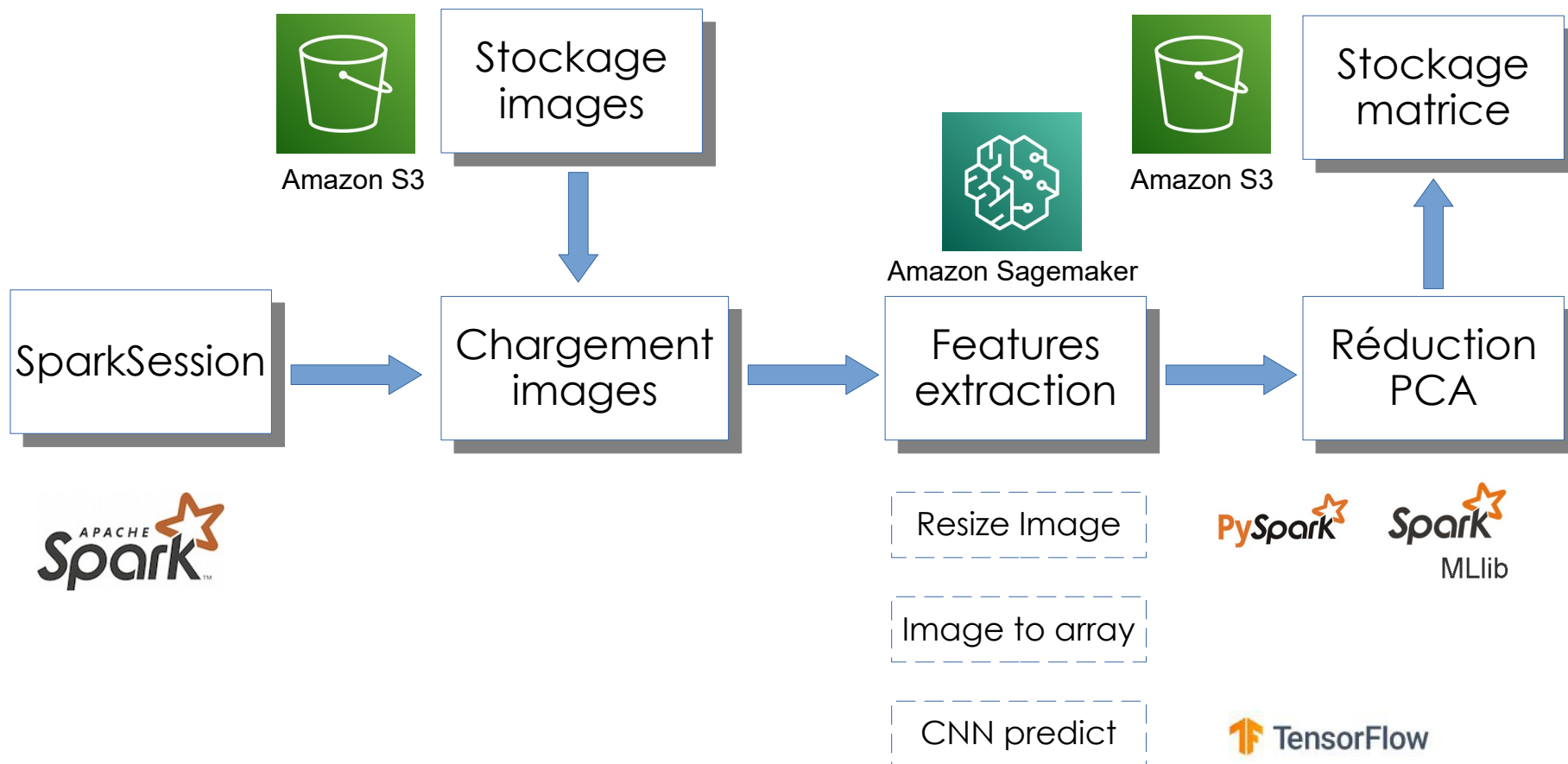
Clé de chiffrement - *facultatif*

Chiffrez vos données de bloc-notes. Choisissez une clé KMS existante ou saisissez un ARN de clé.

Aucun chiffrement personnalisé

Etapes de la chaîne de traitement

Schéma du traitement des images dans notebook Jupyter



Etapes de la chaîne de traitement

Création d'une instance SparkSession

- Point d'entrée vers Spark pour travailler avec RDD, Dataframe et Dataset
- Chargement des fichiers Sagemaker Jar, extraction des clés d'identification par botocore pour accès S3

```
session = boto3.session.get_session()
credentials = session.get_credentials()

conf = (SparkConf()
        .set("spark.driver.extraClassPath", ":%s".join(sagemaker_pyspark.classpath_jars()))

spark = (
    SparkSession
    .builder
    .config(conf=conf) \
    .config('fs.s3a.access.key', credentials.access_key) \
    .config('fs.s3a.secret.key', credentials.secret_key) \
    .config("spark.driver.memory", "15g") \
    .master('local[*]') \
    .appName('P8_OC_dc') \
    .getOrCreate()
)

sc = spark.sparkContext
```

Entrée [14]: # Affichage du contexte pyspark

sc

Out[14]: SparkContext

[Spark UI](#)

Version

v2.4.0

Master

local[*]

AppName

P8_OC_dc

Etapes de la chaîne de traitement

Chargement des images à partir du bucket S3

- Création d'une dataframe Spark pour récupérer les images et leurs chemins
- Ajout des colonnes catégorie, des caractéristiques de l'image (largeur, hauteur, nombre canaux), vecteur initial de l'image

```
# Fonction pour Le chargement des données
def load_data(path_img):
    '''Chargement des dataframes:
    Entrée:
    - path_image: répertoire qui contient les sous répertoires contenant les images
      string

    Retour:
    - df_img: un spark dataframe contenant les images et le nom des fruits associés
    '''

    # Compteur
    start = time.time()

    # Chargement dataframe spark avec Les images
    df_img = spark.read.format("image").load(path_img, inferSchema=True)
    print('Chargement des images effectué')

    df_img = df_img.withColumn("fileName", regexp_replace('image.origin', 'dbfs:/mnt/images/', ''))
    split_col = split(df_img['fileName'], '/')
    df_img = df_img.withColumn('categorie', split_col.getItem(3))

    df_img_disp = df_img.select('image', 'image.origin', "image.height", "image.width", "image.nChannels", "image.mode", "image.data")
    df_img_trait = df_img.select('image.origin', "image.height", "image.width", "image.nChannels", "image.mode", "image.data", 'cate')

    print('Temps de chargement des images : {} secondes'.format(time.strftime('%S', time.gmtime(time.time()-start))))

    return df_img_trait, df_img_disp
```

```
# Affichage des 20 premières images
images_trait.show(20)
```

origin	height	width	nChannels	mode	data	categorie
s3a://p8-mld1-s3-...	435	540	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	434	532	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	526	421	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	527	411	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	435	538	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	527	421	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	434	538	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	435	537	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	527	417	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	433	525	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	435	539	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	798	323	3	16	[FF FF FF FF FF F...]	cucumber_1
s3a://p8-mld1-s3-...	434	534	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	803	325	3	16	[FF FF FF FF FF F...]	cucumber_1
s3a://p8-mld1-s3-...	527	418	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	526	435	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	433	554	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	434	535	3	16	[FF FF FF FF FF F...]	pear_1
s3a://p8-mld1-s3-...	798	324	3	16	[FF FF FF FF FF F...]	cucumber_1
s3a://p8-mld1-s3-...	435	541	3	16	[FF FF FF FF FF F...]	pear_1

only showing top 20 rows

Etapes de la chaîne de traitement

Extraction des features par modèle pré-entraîné VGG16

- Initialisation d'un modèle VGG16 pré-entraîné sur un base Imagenet
- Chargement, redimensionnement image 224*224 pixels, conversion image en array, preprocess et extraction features (fonction predict)

```
model = VGG16(include_top=False, weights='imagenet', pooling='max', input_shape=(224, 224, 3))
model.summary()

# Récupération des ressources sur le service AWS S3
s3_client = boto3.client("s3")
s3 = boto3.resource('s3')
bucket = s3.Bucket(bucket_name)

vgg16_features=[]

for my_bucket_object in bucket.objects.all():
    if my_bucket_object.key.endswith('.jpg'):
        file_byte_string = s3_client.get_object(Bucket=bucket_name, Key=my_bucket_object.key)['Body'].read()

        # Chargement de l'image
        img = Image.open(BytesIO(file_byte_string))

        # Redimensionnement de l'image en 224*224 pixels
        img_redim = img.resize((224, 224))

        # Conversion de l'image en array
        img_array = image.img_to_array(img_redim).reshape((-1,224,224,3))
        img_array = np.array(img_array)

        # Pre-processing des images
        img_array = preprocess_input(img_array)

        # Extraction de features pour un image
        feature = model.predict(img_array).ravel().tolist()
```

origin	height	width	nChannels	mode	data	categorie	features
s3a://p8-m1d1-s3-...	435	540	3	16	[FF FF FF FF FF F...]	pear_1	[60.2613334655761...
s3a://p8-m1d1-s3-...	434	532	3	16	[FF FF FF FF FF F...]	pear_1	[70.5976409912109...
s3a://p8-m1d1-s3-...	526	421	3	16	[FF FF FF FF FF F...]	pear_1	[56.7357063293457...
s3a://p8-m1d1-s3-...	527	411	3	16	[FF FF FF FF FF F...]	pear_1	[56.5463790893554...
s3a://p8-m1d1-s3-...	435	538	3	16	[FF FF FF FF FF F...]	pear_1	[56.8080062866210...
s3a://p8-m1d1-s3-...	527	421	3	16	[FF FF FF FF FF F...]	pear_1	[59.5471496582031...
s3a://p8-m1d1-s3-...	434	538	3	16	[FF FF FF FF FF F...]	pear_1	[46.6975746154785...
s3a://p8-m1d1-s3-...	435	537	3	16	[FF FF FF FF FF F...]	pear_1	[47.8209075927734...
s3a://p8-m1d1-s3-...	527	417	3	16	[FF FF FF FF FF F...]	pear_1	[53.1864929199218...
s3a://p8-m1d1-s3-...	433	525	3	16	[FF FF FF FF FF F...]	pear_1	[52.2564659118652...
s3a://p8-m1d1-s3-...	435	539	3	16	[FF FF FF FF FF F...]	pear_1	[51.8170127868652...
s3a://p8-m1d1-s3-...	798	323	3	16	[FF FF FF FF FF F...]	cucumber_1	[48.6805343627929...
s3a://p8-m1d1-s3-...	434	534	3	16	[FF FF FF FF FF F...]	pear_1	[61.0049324035644...
s3a://p8-m1d1-s3-...	803	325	3	16	[FF FF FF FF FF F...]	cucumber_1	[50.0950622558593...
s3a://p8-m1d1-s3-...	527	418	3	16	[FF FF FF FF FF F...]	pear_1	[55.3948707580566...
s3a://p8-m1d1-s3-...	526	435	3	16	[FF FF FF FF FF F...]	pear_1	[55.1200256347656...
s3a://p8-m1d1-s3-...	433	554	3	16	[FF FF FF FF FF F...]	pear_1	[55.2945518493652...
s3a://p8-m1d1-s3-...	434	535	3	16	[FF FF FF FF FF F...]	pear_1	[46.9456710815429...
s3a://p8-m1d1-s3-...	798	324	3	16	[FF FF FF FF FF F...]	cucumber_1	[51.0811386108398...
s3a://p8-m1d1-s3-...	435	541	3	16	[FF FF FF FF FF F...]	pear_1	[43.2085037231445...

only showing top 20 rows

Etapes de la chaîne de traitement

Réduction de dimensions PCA avec SparkMLib

- Transformation des features en vecteurs dense par une fonction utilisateur avec udf
- Réalisation d'une PCA avec 100 composantes principales (SparkMLib)

```
# Fonction pour réduction de dimensions PCA
def reduc_dim_pca(features):
    '''Réduction des dimensions PCA sur les feature
    Entrée:
    - features de l'image

    Retour:
    - pca
    - pca_matrix: résultat de la réduction de dimensions
    ...

    # Conversion d'un tableau (array) en vecteur dense
    # Création d'une fonction utilisateur avec udf
    vector_dense = udf(lambda x: Vectors.dense(x), VectorUDT())
    img_vd_df = features.select('origin', 'categorie', 'features', vector_dense("features").alias("features_vd"))

    # PCA
    pca_spark = PCA(inputCol="features_vd", outputCol="features_pca", k=100)
    pca = pca_spark.fit(img_vd_df)
    pca_matrix = pca.transform(img_vd_df)

    return pca, pca_matrix
```

Etapes de la chaîne de traitement

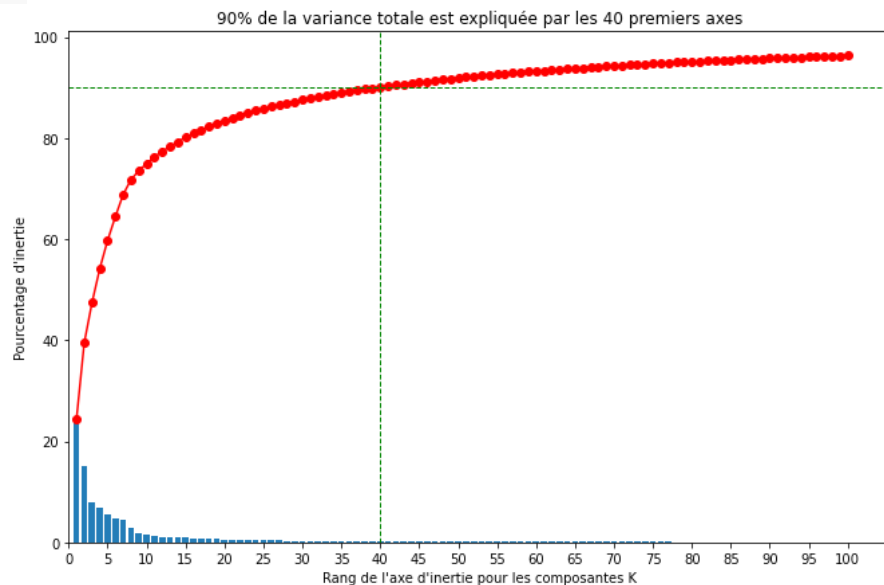
Résultats – Matrice avec sortie réduction de dimensions

- Production d'une matrice de sortie sous la forme d'une dataframe Spark
- Affichage du diagramme des éboulis (on peut envisager une compression des images d'un facteur 2,5).

```
pca_matrix.show(20)
```

origin	categorie	features	features_vd	features_pca
s3a://p8-ml-dl-s3-...	pear_1	[60.2613334655761...	[60.2613334655761...	[244.967693724988...
s3a://p8-ml-dl-s3-...	pear_1	[70.5976409912109...	[70.5976409912109...	[237.879241083890...
s3a://p8-ml-dl-s3-...	pear_1	[56.7357063293457...	[56.7357063293457...	[310.678845815459...
s3a://p8-ml-dl-s3-...	pear_1	[56.5463790893554...	[56.5463790893554...	[314.179557151492...
s3a://p8-ml-dl-s3-...	pear_1	[56.8080062866210...	[56.8080062866210...	[312.563467567600...
s3a://p8-ml-dl-s3-...	pear_1	[59.5471496582031...	[59.5471496582031...	[316.151559872676...
s3a://p8-ml-dl-s3-...	pear_1	[46.6975746154785...	[46.6975746154785...	[298.142309540999...
s3a://p8-ml-dl-s3-...	pear_1	[47.8209075927734...	[47.8209075927734...	[329.555360633738...
s3a://p8-ml-dl-s3-...	pear_1	[53.1864929199218...	[53.1864929199218...	[327.358049823398...
s3a://p8-ml-dl-s3-...	pear_1	[52.2564659118652...	[52.2564659118652...	[335.160848007300...
s3a://p8-ml-dl-s3-...	pear_1	[51.8170127868652...	[51.8170127868652...	[330.895021606815...
s3a://p8-ml-dl-s3-...	cucumber_1	[48.6805343627929...	[48.6805343627929...	[338.827480819587...
s3a://p8-ml-dl-s3-...	pear_1	[61.0049324035644...	[61.0049324035644...	[244.562301630886...
s3a://p8-ml-dl-s3-...	cucumber_1	[50.0950622558593...	[50.0950622558593...	[343.564745116220...
s3a://p8-ml-dl-s3-...	pear_1	[55.3948707580566...	[55.3948707580566...	[345.283913990602...
s3a://p8-ml-dl-s3-...	pear_1	[55.1200256347656...	[55.1200256347656...	[339.028451265808...
s3a://p8-ml-dl-s3-...	pear_1	[55.2945518493652...	[55.2945518493652...	[341.213892528840...
s3a://p8-ml-dl-s3-...	pear_1	[46.9456710815429...	[46.9456710815429...	[347.539314296890...
s3a://p8-ml-dl-s3-...	cucumber_1	[51.0811386108398...	[51.0811386108398...	[353.319801894126...
s3a://p8-ml-dl-s3-...	pear_1	[43.2085037231445...	[43.2085037231445...	[359.858289349568...

only showing top 20 rows



Etapes de la chaîne de traitement

Sauvegarde des résultats sur un bucket S3

- Création d'un nouveau bucket (p8-mldl-s3-bucket-matrix)
- Transformation d'une dataframe Spark en dataframe Pandas et enregistrement au format csv
- Sauvegarde dans le bucket avec le nom p8_result_with_pca.csv

```
# Fonction pour la sauvegarde d'un fichier csv sur un bucket s3
def save_csv_bucket_s3(pca_matrix, file_name, bucket_name):
    '''Sauvegarde résultats dans bucket S3 sous forme d'un fichier csv
    Entrée:
    - pca_matrix
      dataframe pyspark
    - nom du fichier csv à sauvegarder
      string
    - bucket_name: nom du bucket_s3

    Retour:
    - N/A
    ...

    s3_resource = boto3.resource('s3')

    # Création d'un buffer
    csv_buffer = StringIO()

    # Transformation dans une structure dataframe pandas
    pca_matrix.toPandas().to_csv(csv_buffer)

    # Ecriture du fichier csv dans le bucket s3
    s3_resource.Object(bucket_name, file_name).put(Body=csv_buffer.getvalue())
```



Conclusion - Limites

- Le projet a permis de déployer un développement ML sur le cloud en environnement Big Data, en utilisant :
 - Apache Spark et Pyspark pour les traitements distribués
 - AWS IAM pour la gestion des utilisateurs et des autorisations
 - AWS S3 pour le stockage des données
 - AWS Sagemaker pour le développement de modèles
 - avec peu de packages supplémentaires (sauf TensorFlow)
 - le notebook Jupyter permet de travailler directement sur le cloud
- Limites de Sagemaker :
 - outil payant
 - problèmes ponctuels de plantage du notebook Jupyter
 - nécessité d'ajuster le niveau de RAM pour la Java Heap size (Spark session)



Perspectives

- Utilisation des résultats en sortie pour mise en œuvre d'un moteur de classification des images
- Utilisation d'autres services AWS en production :
 - service AWS EMR (clusters) pour le scaling horizontal
 - module Model monitor de SageMaker pour surveiller la qualité de prédictions du modèle (analyse dérive)
- Etude pour maîtriser les coûts de déploiement et de maintenance sur AWS