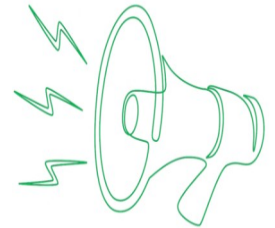
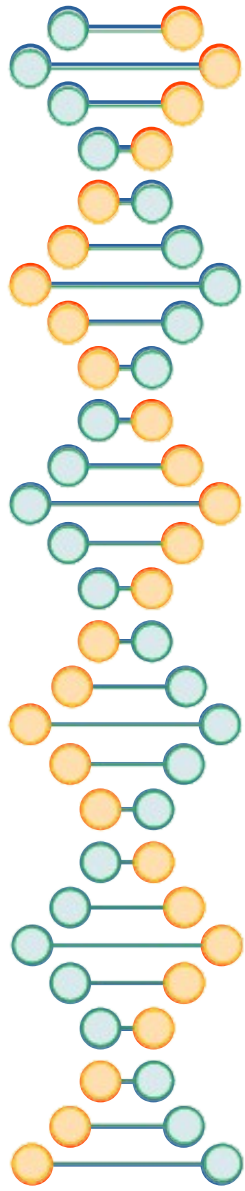


# Déployez un modèle dans le cloud

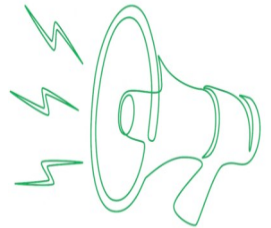
Projet 7  
Sofia Velasco

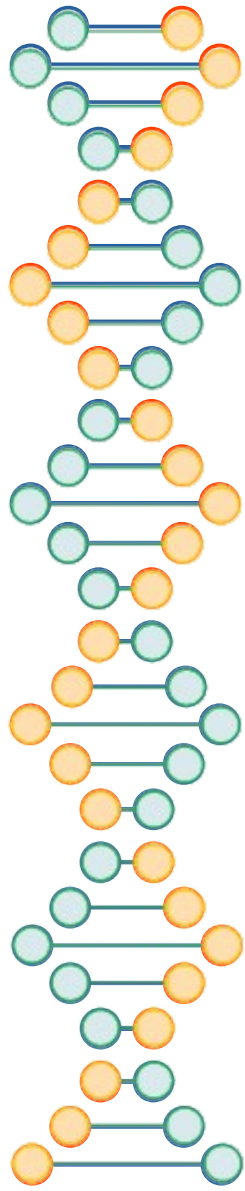




## Objectif:

1. S'approprier des travaux préalables et les compléter (PCA, inférence distribuée).
2. Développement dans Databricks (Pyspark) et en utilisant le cloud AWS pour le stockage.





# Création d'un compte AWS

(Stockage dans le Cloud)

- A. Création d'un compte AWS
- B. Connexion au compte AWS
- C. Configuration du compte AWS
  - Création User
  - Création du Bucket sur S3



# A. Création d'un compte AWS

(Option essais gratuit → valide 12 mois par défaut)

- Sur: <https://aws.amazon.com/fr/free/>

**1**

aws

re:Invent Produits Solutions Tarification Documentation Apprendre Réseau de partenaires

Offre gratuite d'AWS Présentation Questions fréquentes (FAQ) Conditions

## Offre gratuite d'AWS

Testez gratuitement la plateforme, les produits et les services AWS

En savoir plus sur l'offre gratuite d'AWS ⓘ

**Créer un compte gratuit**

**2**

aws

### Connexion

☒ **Utilisateur racine**  
Propriétaire du compte qui effectue des tâches requérant un accès illimité. [En savoir plus](#)

☐ **Utilisateur IAM**  
Utilisateur au sein d'un compte qui effectue des tâches quotidiennes. [En savoir plus](#)

Adresse e-mail de l'utilisateur racine

username@example.com

**Suivant**

En continuant, vous acceptez le [contrat client AWS](#) ou tout autre accord pour l'utilisation des services AWS, ainsi que la [politique de confidentialité](#). Ce site utilise des cookies essentiels. Consultez notre [avis sur les cookies](#) pour plus d'informations.

Nouveau sur AWS ?

**Créer un nouveau compte AWS**

**3**

### Sign up for AWS

Root user email address  
Used for account recovery and some administrative functions

AWS account name  
Choose a name for your account. You can change this name in your account settings after you sign up.

**Verify email address**

OR

**Sign in to an existing AWS account**

Note:


L'adresse mail devra être confirmée par un code.  
Informations personnelles dont **région (choisir France)**  
et coordonnées bancaires demandées.



# A. Création d'un compte AWS

(Option essais gratuit → valide 12 mois par défaut)

4/5




## Sign up for AWS

### Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)  
☒ You can change your plan anytime in the AWS Management Console.


☒ **Basic support - Free**

- Recommended for new users just getting started with AWS
- 24x7 self-service access to AWS resources
- For account and billing issues only
- Access to Personal Health Dashboard & Trusted Advisor




☐ **Developer support - From \$29/month**


- Recommended for developers experimenting with AWS
- Email access to AWS Support during business hours
- 12 (business)-hour response times



☐ **Business support - From \$100/month**

- Recommended for running production workloads on AWS
- 24x7 tech support via email, phone, and chat
- 1-hour response times
- Full set of Trusted Advisor best-practice recommendations







**Need Enterprise level support?**

From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more](#)

**Complete sign up**

6



## Congratulations

Thank you for signing up for AWS.

We are activating your account, which should only take a few minutes. You will receive an email when this is complete.

**Go to the AWS Management Console**

[Sign up for another account or contact sales.](#)

6

5

## B. Connexion au compte AWS (Une fois le compte crée).

- Sur: <https://aws.amazon.com/fr/free/>



1ère connexion on arrive directement ici  
après création de compte.



## C. Configuration du compte AWS (Console → Création User sur IAM).

1

Console Home

Introducing 4 new widgets  
Now you can view the Secu

Recently visited

- IAM
- WorkSpaces
- CloudFormation
- AWS Budgets
- S3

2

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers

IAM dashboard

Security recommendations 1

- Add MFA for root user**  
Add MFA for root user - Enable multi-factor authentication (MFA) for this account.
- Root user has no active access keys**  
Using access keys attached to an IAM user instead of the root user.

IAM resources

User groups	Users	Roles

3

Introducing the new Users list experience  
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

IAM > Users

**Users (0)** Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

User name	Groups	Last activity	MFA	Password age	Active key age
No resources to display					

Add users



## C. Configuration du compte AWS (Console → Création User).

**Add user**

1 2 3 4 5

### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\* toto1 1

[Add another user](#)

### Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type\* 2

☒ **Access key - Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **Password - AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel **Next: Permissions** 3

**Add user**

1 2 3 4 5

### Set permissions

[Add user to group](#) [Copy permissions from existing user](#) 1 [Attach existing policies directly](#)

[Create policy](#)

Filter policies 2 S3 Showing 9 results

	Policy name	Type	Used as
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	AWS managed	None
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	None
<input type="checkbox"/>	AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	None
<input checked="" type="checkbox"/>	AmazonS3OutpostsFullAccess 3	AWS managed	None
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AWSBackupServiceRolePolicyForS3Backup	AWS managed	None
<input type="checkbox"/>	AWSBackupServiceRolePolicyForS3Restore	AWS managed	None
<input type="checkbox"/>	QuickSightAccessForS3StorageManagementAnalyticsReadOnly	AWS managed	None

4

### Set permissions boundary

Cancel Previous **Next: Tags**



## C. Configuration du compte AWS (Console → Création User).

**Add user** 1 2 3 4 5

**Add tags (optional)**

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
Name	databricks 1	×
Use	User from databricks	×
<a href="#">Add new key</a>		

You can add 48 more tags.

Cancel Previous **Next: Review** 2

**Add user** 1 2 3 4 5

**Review**

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

**User details**

<b>User name</b>	toto1
<b>AWS access type</b>	Programmatic access - with an access key
<b>Permissions boundary</b>	Permissions boundary is not set

**Permissions summary**

The following policies will be attached to the user shown above.

Type	Name
Managed policy	<a href="#">AmazonS3OutpostsFullAccess</a>

**Tags**

The new user will receive the following tags

Key	Value
Name	databricks
Use	User from databricks

Cancel Previous **Create user**

# C. Configuration du compte AWS (Console → Création User).

123458

✓ Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://949926556363.signin.aws.amazon.com/console>

Download .csv

1

⚠

ATTENTION : Downloader et bien garder le fichier cvs avec les keys.

User	Access key ID	Secret access key
▶ ✓ toto1	AKIA52LAI6LFQNX7AASO	***** Show

2

Close

✓ The user toto1 have been created.

IAM > Users

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Find users by username or access key

User name

Groups

Last activity

MFA

Password age

Active key age

toto1

None

Never

None

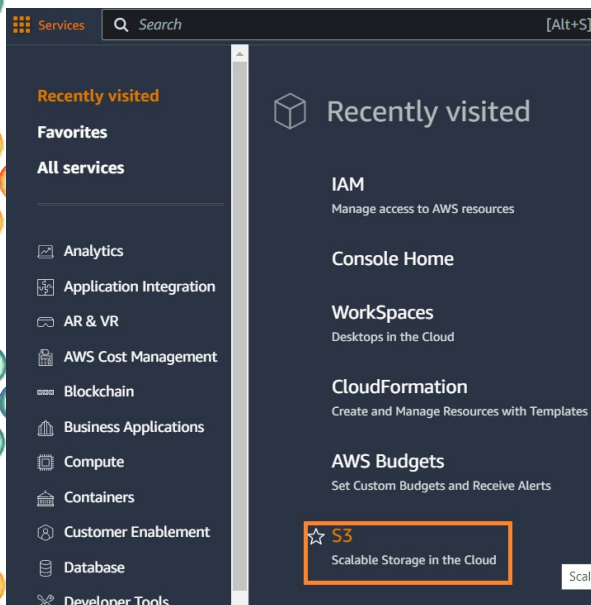
None

✓ 3 minutes ago

10

# C. Configuration du compte AWS (Console → Création du Bucket sur S3).

Bucket = 'contenant' où se trouvent les fichiers et dossier à appeler avec Databricks.



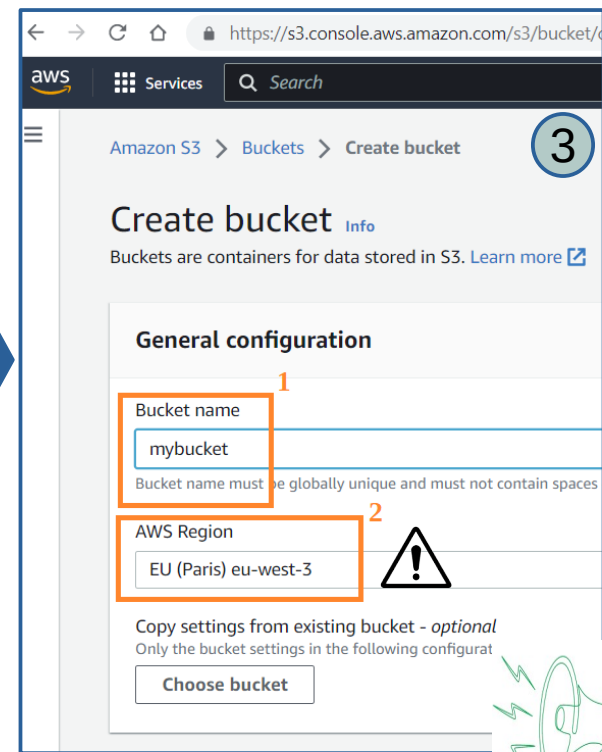
Aller sur S3

2

**Create a bucket**

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

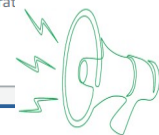
**Create bucket**



ATTENTION :  
Noms des Buckets → uniques  
Choisir **AWS Region** → **EU(Paris) eu-west3**



Laisser tous les autres  
choix par défaut



## C. Configuration du compte AWS (Console → Création du Bucket sur S3).

Amazon S3 > Buckets

► **Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

**Buckets (1)** [Info](#)  
Buckets are containers for data stored in S3. [Learn more](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

< 1 > [Settings](#)

	Name	AWS Region	Access	Creation date
<input type="radio"/>	mybucketsvg	EU (Paris) eu-west-3	Bucket and objects not public	December 29, 2022, 17:01:46 (UTC+01:00)



Dans ce bucket:

- On **uploadera notre data**
- On retrouvera nos **résultats** obtenus en utilisant **Databricks**.

mybucketsvg [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

**Objects (0)**  
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

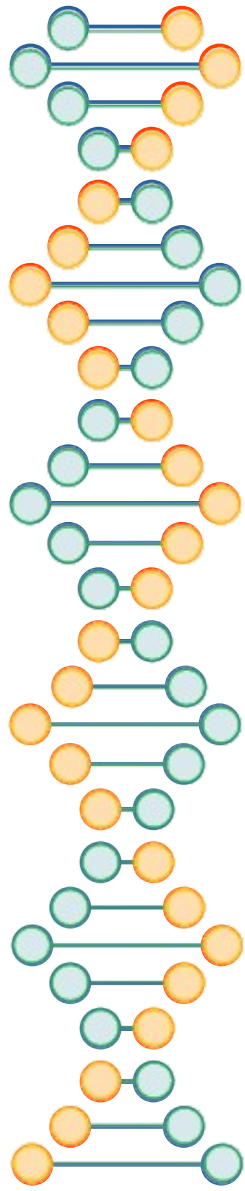
[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

< 1 > [Settings](#)

	Name	Type	Last modified	Size	Storage class
No objects					
You don't have any objects in this bucket.					

[Upload](#)





# Création d'un compte Databricks.

Databricks → Plate-forme Web,

- pour: travailler avec Spark,
- avec: gestion automatisée des clusters et notebooks style IPython

A. Création d'un compte Databricks

B. Liaison avec le compte AWS

(Création automatique du Workspace )

C. Configuration du compte Databricks

- Création Cluster
- Création Notebook



# A. Création d'un compte Databricks

(Option essais gratuit → valide 14 jours)

- Sur: <https://www.databricks.com/try-databricks#account>

The diagram illustrates the four steps of creating a Databricks account:

- Step 1: Create your Databricks account**  
This screen shows a form with fields for First name, Last Name, Email, Company, Title, Phone (Optional), and Country. The 'Country' dropdown is highlighted with an orange box and labeled '1'. Below the form is a 'Continue' button, also highlighted with an orange box.
- Step 2: Choose a cloud provider**  
This screen shows three options: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. The 'Continue' button at the bottom is highlighted with an orange box and labeled '2'.
- Step 3: Check your email to start your trial.**  
This is a simple instruction screen with the Databricks logo and a blue arrow pointing down to the next step.
- Step 4: Set your password**  
This screen shows a 'Set your password' form with fields for Password and Confirm Password. The 'Set Password' button at the bottom is highlighted with an orange box and labeled '2'.

Note:  
Bien sélectionner 'Country' → 'France'.

# A. Création d'un compte Databricks

(Option essais gratuit → valide 14 jours)

5

Select a subscription plan

1

Standard

Basic platform for your data analytics and ML workloads

- ✗ Databricks SQL
- ✗ Autoscaling
- ✗ Role based access control

Selected

Premium

Databricks SQL, cl... and autoscaling

- ✓ Databricks SQL
- ✓ Autoscaling
- ✓ Role based acc...

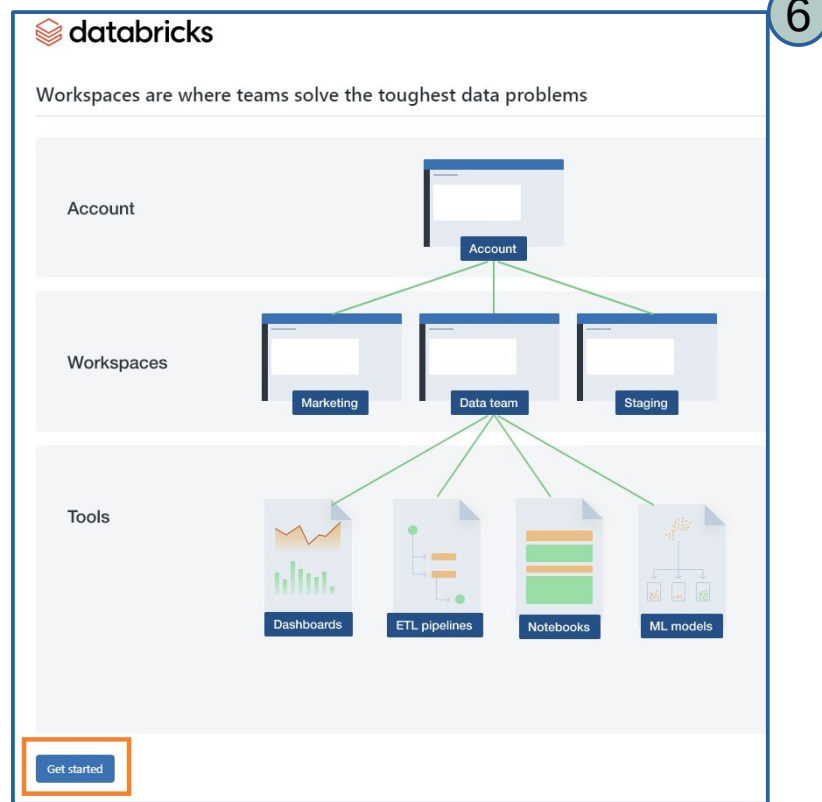
Select

Your 14-day free trial starts when you click Continue. Thereafter, you will be

2

Continue

Note:  
Prendre la version standard!






# A. Création d'un compte Databricks

(Option essais gratuit → valide 14 jours)

**7**



To proceed, please confirm you have the following

- ☒ An AWS account
- ☒ The password you used to setup your Databricks account
- ☒ A friendly name for your workspace


**Confirm**

## ATTENTION:

**Vous devez être connectés à votre compte AWS, pour lancer le processus de 'quickstart' qui se chargera de lier le compte Databricks au compte AWS.**



**8**



Let's setup your first workspace

We're going to send you to your AWS Console to configure your account.

We'll then pre-populate a CloudFormation template that creates an IAM role

Workspace name

toto1 **1**

AWS region

Frankfurt (eu-central-1) **2**

☒ I have data in S3 that I want to query with Databricks

**3** You will be prompted for your S3 bucket name on the next page. An instance profile will be created to let you query data in the bucket.

**4** **Start quickstart**

After clicking "Start quickstart" follow the steps on AWS to create a stack, then return to this tab.

If you encounter any errors during the process, visit the [Databricks Community](#) for troubleshooting guidance.

## ATTENTION:

**Prendre 'AWS Region' → Frankfurt (eu-central-1)  
Compatible avec la France pour le respect des  
contraintes du RGPD**





**Stack name**

Stack name  
databricks-workspace-stack  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**Databricks Account Credentials**

Databricks account email address  
Your case-sensitive Databricks account email address.  
sofavelasdigas@hotmail.com

Databricks account password  
This is the password you set for your Databricks account.  
1

Databricks account ID  
Find your account ID at <https://accounts.cloud.databricks.com>  
90d24396-bf54-4597-a95c-aa5dc25e5810

**Workspace configuration**

Workspace name  
Human-readable name for this workspace.  
toto1

AWS Region of the Databricks workspace  
AWS Region where the workspace will be created.  
eu-central-1

Data bucket name  
The S3 bucket where your data is stored. By entering \* you can access data in any of your S3 buckets. We recommend to restrict this policy later by going to IAM > Roles  
mybucketsvg 2

**Required IAM role and S3 bucket configuration**

Cross-account IAM role name  
Specify a unique cross-account IAM role name. For naming rules, see [https://docs.aws.amazon.com/IAM/latest/APIReference/API\\_CreateRole.html](https://docs.aws.amazon.com/IAM/latest/APIReference/API_CreateRole.html).  
db-7952847b46d3078c72081c2c1eae040-iam-role

Root S3 bucket name  
Specify a unique name for the S3 bucket where Databricks will store metadata for your workspace. Use only alphanumeric characters. For naming rules, see <https://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html>.  
db-7952847b46d3078c72081c2c1eae040-s3-root-bucket

**Capabilities**

**The following resource(s) require capabilities: [AWS::IAM::Role]**  
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. Learn more [here](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names. 3

Cancel Create change set **Create stack** 4

9

## B. Liaison avec le compte AWS (Création automatique du Workspace)

### ATTENTION:

Remplir uniquement:

- le 'Databricks password' et
- le 'Data bucket name' → celui de notre AWS (où se trouvent les données)



**Databricks account password**  
This is the password you set for your Databricks account.  
1

**Data bucket name**  
The S3 bucket where your data  
mybucketsvg 2

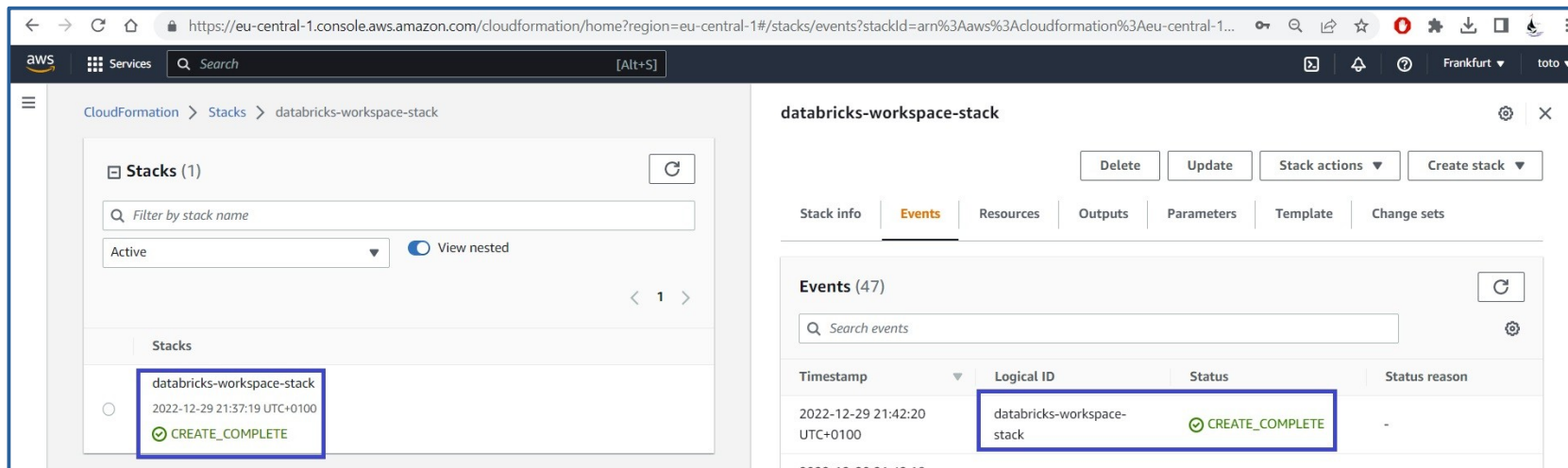
☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names. 3 4

Cancel Create change set **Create stack**

Ceci lie automatiquement le 'ID databricks' et  
le 'IAM role' de AWS

## B. Liaison avec le compte AWS (Création automatique du Workspace)

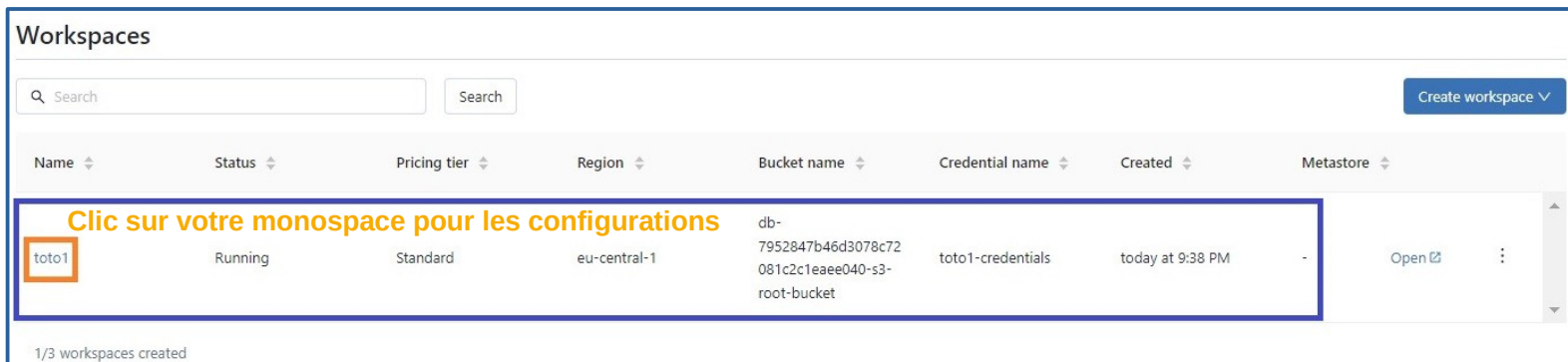
Sur notre AWS on verra alors:



The screenshot shows the AWS CloudFormation console for the 'databricks-workspace-stack' in the eu-central-1 region. The left pane shows a list of stacks with 'databricks-workspace-stack' highlighted, indicating a 'CREATE\_COMPLETE' status. The right pane shows the 'Events' tab for this stack, with a table listing events. The first event, 'databricks-workspace-stack', is highlighted with a blue box and shows a 'CREATE\_COMPLETE' status.

Timestamp	Logical ID	Status	Status reason
2022-12-29 21:42:20 UTC+0100	databricks-workspace-stack	CREATE_COMPLETE	-

Sur notre Databricks on verra alors:



The screenshot shows the Databricks Workspaces page. A table lists the workspaces. The 'toto1' workspace is highlighted with a blue box. An orange box highlights the 'toto1' name, and a blue box highlights the entire row. A text overlay points to the 'toto1' name.

**Clic sur votre monospace pour les configurations**

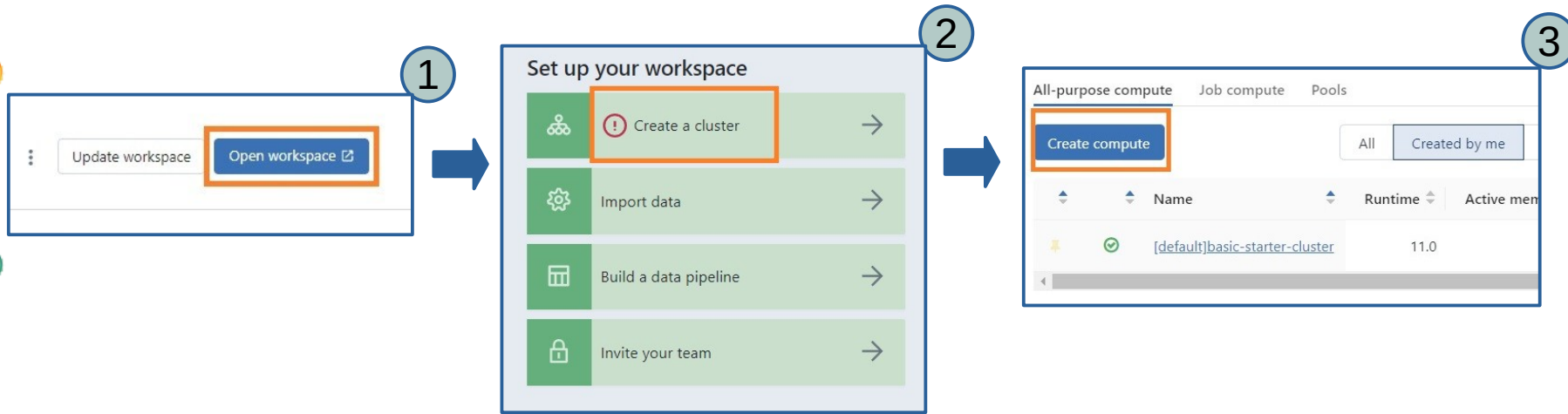
Name	Status	Pricing tier	Region	Bucket name	Credential name	Created	Metastore
toto1	Running	Standard	eu-central-1	db-7952847b46d3078c72081c2c1eae040-s3-root-bucket	toto1-credentials	today at 9:38 PM	-

1/3 workspaces created



# A. Configuration du compte Databricks (Création Cluster)

Suite à cliquer sur notre Workspace:



# A. Configuration du compte Databricks (Création Cluster)

**Sofia Velasco's Cluster**

☒ Multi node ☐ Single node

**Access mode** **Single user access**

Single user | Sofia Velasco (sofiavelascigas@h...)

**Performance**

**Databricks runtime version**

Runtime: 11.3 LTS (Scala 2.12, Spark 3.3.0)

☒ Use Photon Acceleration

**Worker type** **Min workers** **Max workers**

i3.xlarge 30.5 GB Memory, 4 Cores | 2 | 8

**Driver type**

Same as worker 30.5 GB Memory, 4 Cores

☒ Enable autoscaling

☐ Enable autoscaling local storage

☒ Terminate after 120 minutes of inactivity

**Instance profile**

None

**Tags**

Create Cluster Cancel

4

**Databricks runtime version**

Runtime: 11.3 LTS (Scala 2.12, Spark 3.3.0)

Standard	>	12.0 ML	GPU, Scala 2.12, Spark 3.3.1
ML	>	12.0 ML	Scala 2.12, Spark 3.3.1
		11.3 LTS ML	GPU, Scala 2.12, Spark 3.3.0
		11.3 LTS ML	Scala 2.12, Spark 3.3.0
		11.2 ML	GPU, Scala 2.12, Spark 3.3.0
		11.2 ML	Scala 2.12, Spark 3.3.0
		11.1 ML	GPU, Scala 2.12, Spark 3.3.0
		11.1 ML	Scala 2.12, Spark 3.3.0
		11.0 ML	GPU, Scala 2.12, Spark 3.3.0
		11.0 ML	Scala 2.12, Spark 3.3.0
		10.4 LTS ML	GPU, Scala 2.12, Spark 3.2.1
		10.4 LTS ML	Scala 2.12, Spark 3.2.1

- Choisir un cluster ML pour avoir les packaging Ipython
- GPU non nécessaire!



# A. Configuration du compte Databricks (Création Notebook)

1

2

**ATTENTION:**

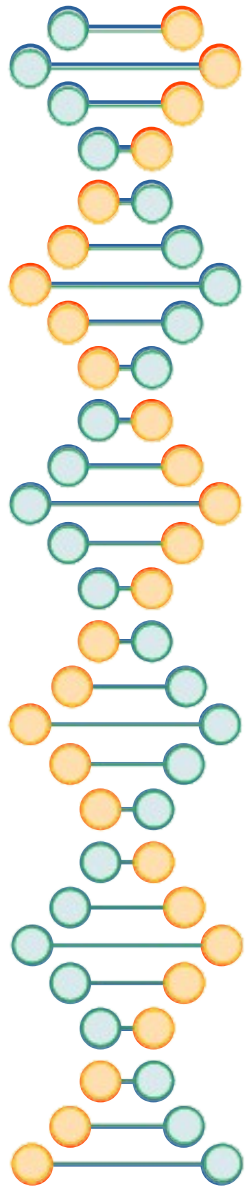
Mount à faire une seule fois!  
Après on peut coder!

3

```
Cmd 1
1  ### MOUNT AND READ S3 FILES
2  AWS_BUCKET_NAME = "mybucketsvg"
3  MOUNT_NAME = "bridge"
4  dbutils.fs.mount("s3a://%s" % AWS_BUCKET_NAME, "/mnt/%s" % MOUNT_NAME)
5  display(dbutils.fs.ls("/mnt/%s" % MOUNT_NAME))

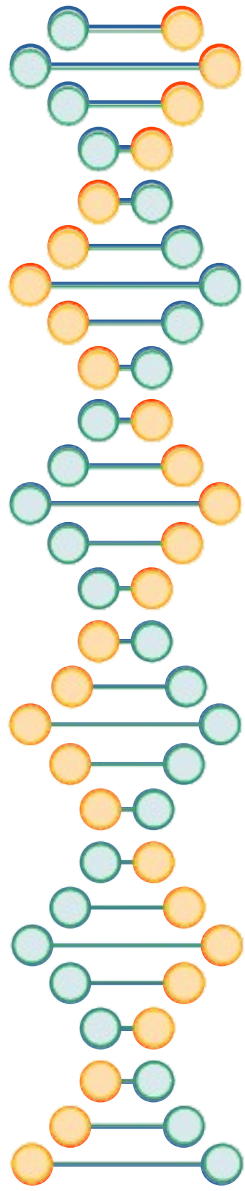
Shift+Enter to run
```





# Étapes de la chaîne de traitement (PySpark/Databricks)





## A. PySpark (Atouts de Spark face à Pandas).



### Pourquoi PySpark et non pas Pandas?

**Pandas** → exécute des opérations sur **une seule machine**

**PySpark** → exécute des opérations sur **plusieurs machines.**



Pour faire du **Machine Learning** où traiter des **datasets volumineux**, **PySpark** est la solution idéale:

**Traitement des opérations plusieurs fois (100x) plus rapidement que Pandas, et sans crash.**





## B. Étapes de la chaîne de traitement (Spark et le modèle).

- Création de **Session Spark et Contexte** → Nécessaires pour l'utilisation de Spark.

```
spark = (SparkSession
    .builder
    .appName('P8')
    .master('local')
    .config("spark.sql.parquet.writeLegacyFormat", 'true')
    .getOrCreate()
)
```

### ATTENTION:

Une fois Spark lancé → tout se fait par Spark



```
SparkSession - hive
SparkContext
Spark UI
Version
  v3.3.1
Master
  spark://10.75.188.178:7077
AppName
  Databricks Shell
```

- Le modèle:**
  - 'MobileNetV2' de TensorFlow (poids pré-calculés d'imagenet) → rapide et efficace.  
`model = MobileNetV2(weights='imagenet', include_top=True, input_shape=(224, 224, 3))`
  - Sans deuxième couche.  
`new_model = Model(inputs=model.input, outputs=model.layers[-2].output)`
  - Avec broadcast de ses poids (sur tous les processeurs de Spark) → plus rapide  
`broadcast_weights = sc.broadcast(new_model.get_weights())`
  - Récupération du vecteur des features de dimensions (1,1,1280) → mis dans des  
moteur de classification reconnaîtra les différents fruits





## B. Étapes de la chaîne de traitement (Les images).

- Chargement des **images**:  
Format binaire → permet de séparer facilement un objet de l'arrière-plan.

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



```
▶ (1) Spark Jobs  
▶ images: pyspark.sql.dataframe.DataFrame = [path: string, modificationTime: timestamp ... 3 more fields]  
  
root  
 |-- path: string (nullable = true)  
 |-- modificationTime: timestamp (nullable = true)  
 |-- length: long (nullable = true)  
 |-- content: binary (nullable = true)  
 |-- label: string (nullable = true)  
  
None  
Deux colonnes: 'path' et 'label'  
+-----+-----+  
|path|label|  
+-----+-----+  
|dbfs:/mnt/bridge/Test/apple_hit_1/r0_167.jpg|apple_hit_1|  
|dbfs:/mnt/bridge/Test/apple_hit_1/r1_51.jpg|apple_hit_1|  
|dbfs:/mnt/bridge/Test/cabbage_white_1/r0_139.jpg|cabbage_white_1|  
|dbfs:/mnt/bridge/Test/cabbage_white_1/r0_71.jpg|cabbage_white_1|  
|dbfs:/mnt/bridge/Test/cabbage_white_1/r0_103.jpg|cabbage_white_1|  
+-----+-----+  
only showing top 5 rows
```

### ATTENTION:

- On n'entraîne pas le modèle → **uniquement dossier 'TEST'**.
- **Base de données très réduite** (3 images par fruits) du dossier 'TEST' → ne pas dépasser les versions moins chères de AWS, et test de Databricks.



## B. Étapes de la chaîne de traitement (Traitement des images et Featurize).

- 3 Fonctions auxiliaires (liées entre elles):
- **Traitement des images.**  
Size: (224,224) → besoins du modèle. Array: conversion RVB en BGR → compatible TensorFlow.

```
def preprocess(content):  
    img = Image.open(io.BytesIO(content)).resize([224, 224])  
    arr = img_to_array(img)  
    return preprocess_input(arr)
```

- **Featurize d'une série pandas d'images, en utilisant un modèle.**  
Prédiction avec modèle → obtention des features.  
En retour → série pandas des features des images.

```
def featurize_series(model, content_series):  
  
    input = np.stack(content_series.map(preprocess)) #on stack (ajoute des arries) et applique la focntion process définie auparavant.  
    preds = model.predict(input) #on predit avec le modèle choisi.  
  
    output = [p.flatten() for p in preds] #on applatit les tenseur.  
    return pd.Series(output)
```

Note: les features peuvent être des tenseur multi-dimensionnels → 'flatten', supprimer toute leurs dimensions sauf une.



## B. Étapes de la chaîne de traitement (Pandas UDF).

- **Pandas UDF** (batches et définition du modèle).
  - **Scalar Iterator pandas UDFs** → load modèle une fois et le réutilise pour plusieurs 'batches' (lots) de données → amortit coût en temps et mémoire.
  - En retour → une colonne 'Spark DataFrame' de type `ArrayType(FloatType)`.
  - 'content\_series\_iter' → itérateur sur des 'batches' (lots) de data. Chaque 'batch' est un 'pandas Series' d'images.

```
@pandas_udf('array<float>', PandasUDFType.SCALAR_ITER)  
def featurize_udf(content_series_iter):  
    model = model_fn()  
    for content_series in content_series_iter:  
        yield featurize_series(model, content_series)
```

Batches + Modèle aux poids broadcastés



'Distributed model inference using TensorFlow Keras'.



## B. Étapes de la chaîne de traitement (Extraction des Features).

- Extraction des Features sur nos images.

```
features_df = images.repartition(24).select(col("path"),  
                                             col("label"),  
                                             featurize_udf("content").alias("features")  
                                             )
```

### ATTENTION:

- '.repartition()' → crée des 'batches' d'images. On en crée 24 → 24 fichier parquet à la fin.
- Le nombre de lots choisit car 24 types de fruits (ie. 24 dossiers).

### Résultats:

```
f_pandas=features_df.toPandas()  
f_pandas.loc[0,'features'].shape  
  
Out[112]: (1280,)  
f_pandas.shape  
Out[114]: (72, 3)
```



On a bien:

- Des vecteurs de 1280 features
- 72 lignes (3 fruits x 24 types)
- 3 colonnes: 'path', 'label' (initiales) + features



## B. Étapes de la chaîne de traitement (PCA).

- **PCA** → Réduction de dimension des Feature Vectors.

PCA en PySpark → `from pyspark.ml.feature import PCA`

Pour appliquer le PCA 'features' doit être: 'VectorUDT' et non 'ArrayType'.

```
from pyspark.ml.linalg import Vectors, VectorUDT
```

```
list_to_vector_udf = udf(lambda l: Vectors.dense(l), VectorUDT())
```

```
features_df2 = features_df2.select(col("*"), list_to_vector_udf(features_df2["features"]).alias("features_2"))
```

```
pca = PCA(k=50, inputCol="features_2", outputCol="pcaFeatures")
```

```
model_pca = pca.fit(features_df2)
```

```
result = model_pca.transform(features_df2)
```

**ATTENTION:** on fixe 50 features (ie. k=50)  
car ça explique aux alentours de 95 % de la  
variance.

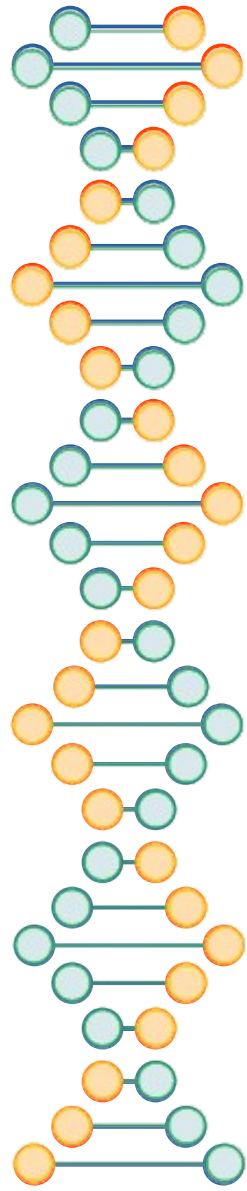
Résultats:

```
f_pandas_PCA=result.toPandas()
f_pandas_PCA.loc[0, 'pcaFeatures'].shape
Out[157]: (50,)
f_pandas_PCA.shape
Out[159]: (72, 5)
```

On a bien:

- Des vecteurs de 50 features
- 72 lignes (3 fruits x 24 types)
- 5 colonnes: 'path', 'label' (initiales) + features  
+features\_2 (vecteur) +pcaFeatures (réduite)

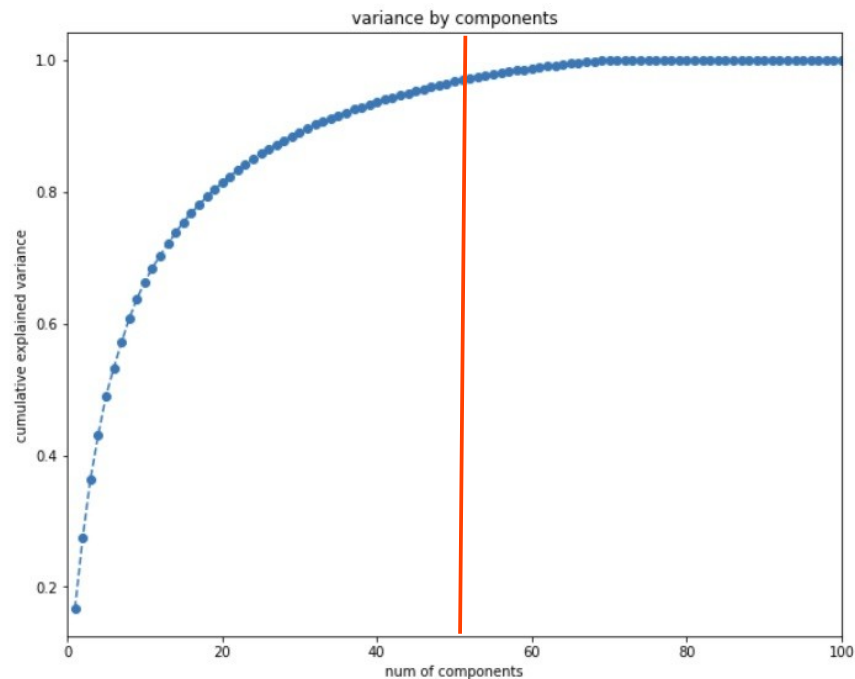




## B. Étapes de la chaîne de traitement (PCA).

### ATTENTION:

on fixe 50 features (ie.  $k=50$ ) car ça explique aux alentours de 95 % de la variance.





## B. Étapes de la chaîne de traitement (Enregistrement sur S3 → 'parquet').

- **Enregistrement des Features obtenus en format 'parquet' sur S3.**

'Apache Parquet' → format pour **stocker de très gros volumes de données** ayant une structure «complexe». Facilite distribution de la charge de traitement, et sa structuration exclue rapidement des pans entiers du jeu de données → **réduit temps de traitement**.

```
#K.1 Résultats avant PCA.
```

```
features_df.write.mode("overwrite").parquet(PATH_Result)
```

```
#K.2 Résultats après PCA.
```

```
result.write.mode("overwrite").parquet(PATH_Result_PCA)
```

- **Lecture des fichiers 'parquet' obtenus et sauvegardés en S3.**

Avec 'spark.read.parquet', en pointant sur le répertoire principal on charge tous les fichiers du répertoire de type parquet.

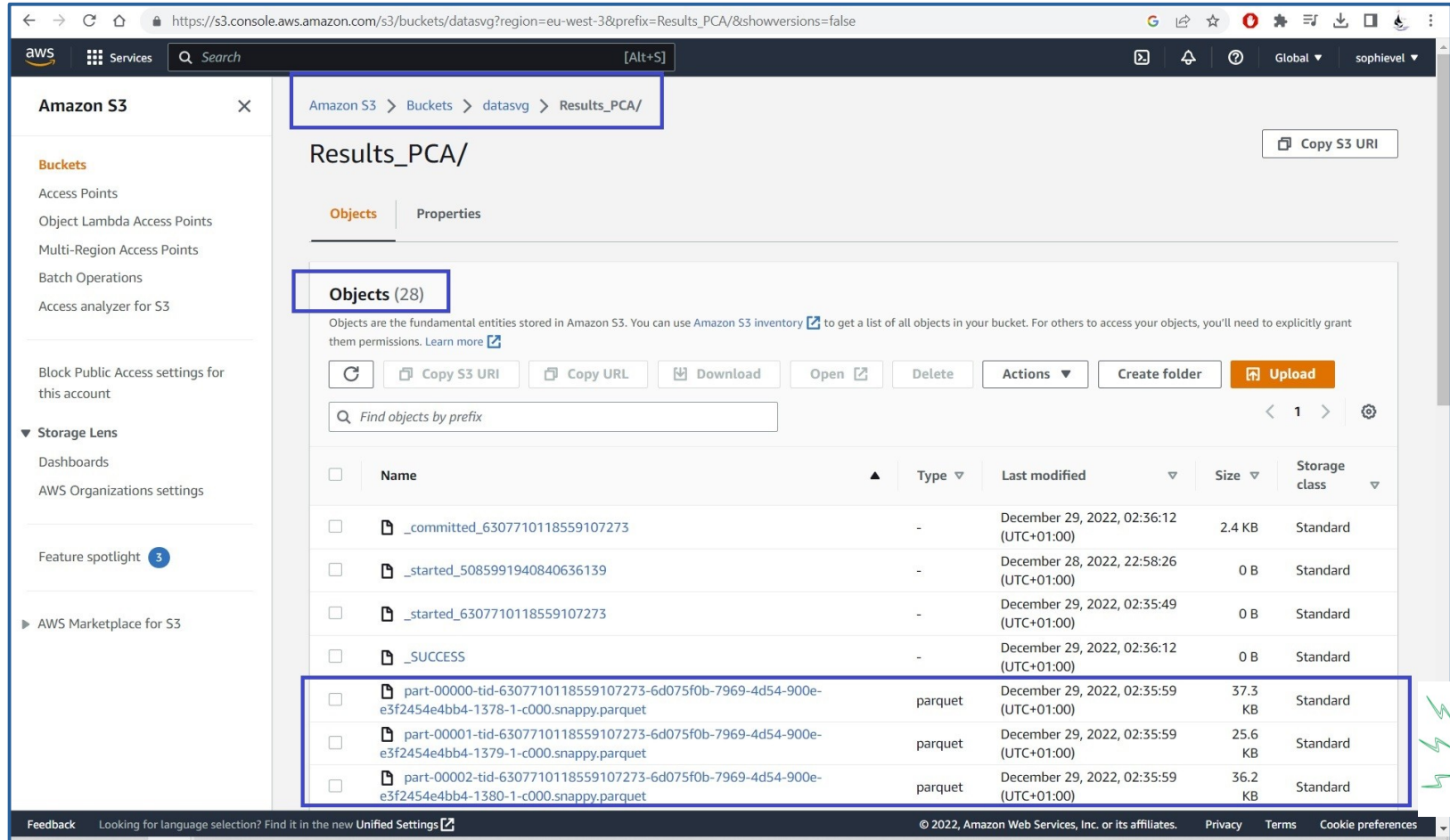
```
df = spark.read.parquet(PATH_Result, engine='pyarrow')  
display(df)
```

Table ▾ +

path	label	features
dbfs/mnt/bridge/Test/apple_granny_smith_1/r1_251.jpg	apple_granny_smith_1	[0.78509045, 0.94276524, 0, 0, 0, 0.18496965, 0, 0.0694279, 0.103255376, 0, 0, 0, 0, 0, 0.03737079, 0, 0, 0.09759391, 0, 0, 0.22270946, 0.050179053, 0, 0, 0.52550423, 1.2778711, 0, 0, 0.5197245, 0.0020650844, 0, 0, 0.44645616, 0, 0.13031834, 0.14059882, 2.1999779, 0, 0, 0.047193915, 0, 0, 1.5839534, 0, 0, 0.4349056, 0.23649508, 0.6606921, 0, 0.015158449, 0.00024021888, 0, 0, 0, 0.1891966, 0, 0.031277373, 0, 0.02731763, 0, 0, 0, 0, 0.3602988, 0.06701653, 0.124102786, 0, 0, 0, 0, 0, 0.11493514, 0.3177375, 0, 0, 0.11075862, 0.039396074, 0.1271352, 0, 0.2318707, 0.12300039, 0, 0, 0.4530867, 0, 0, 0.013674882, 0.18436135, 0.021185292, 0.49742788, 0.30888566, 0, 0.014879548, 0.014159601, 0, 0, 0.01825428, 0, 0.11093818, 0.030685054,



# C. Visualiser les fichiers 'parquet' sur AWS



The screenshot shows the AWS S3 console interface. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > datasvg > Results\_PCA/. The left sidebar shows the 'Amazon S3' service with various options like Buckets, Access Points, and Storage Lens. The main content area displays the 'Results\_PCA/' bucket with a tab for 'Objects (28)'. Below the tab, there's a search bar and a list of objects. Three objects are highlighted in a blue box:

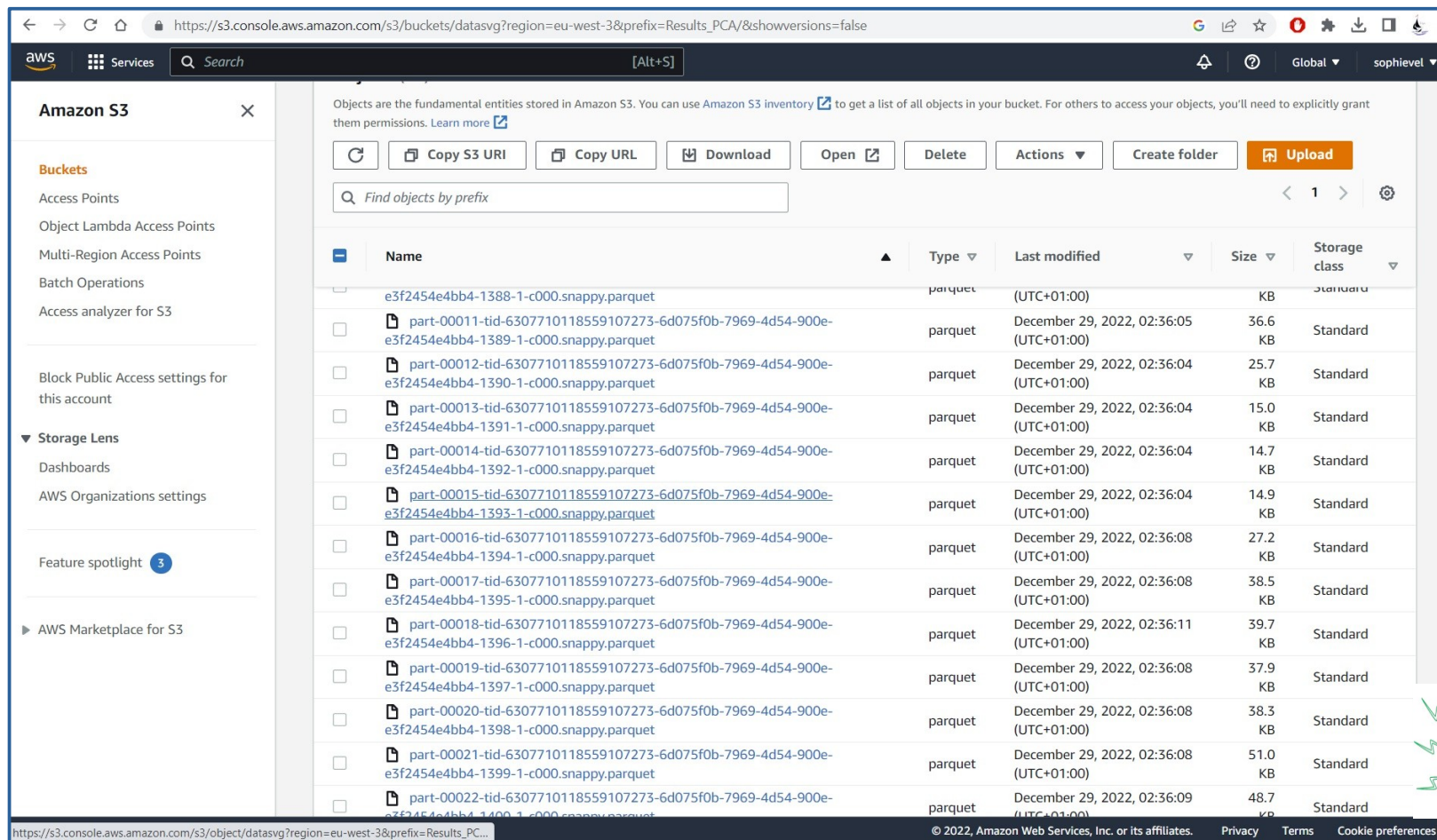
Name	Type	Last modified	Size	Storage class
part-00000-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1378-1-c000.snappy.parquet	parquet	December 29, 2022, 02:35:59 (UTC+01:00)	37.3 KB	Standard
part-00001-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1379-1-c000.snappy.parquet	parquet	December 29, 2022, 02:35:59 (UTC+01:00)	25.6 KB	Standard
part-00002-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1380-1-c000.snappy.parquet	parquet	December 29, 2022, 02:35:59 (UTC+01:00)	36.2 KB	Standard

Other objects in the list include '\_committed\_6307710118559107273', '\_started\_5085991940840636139', '\_started\_6307710118559107273', and '\_SUCCESS'. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', '© 2022, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.





# C. Visualiser les fichiers 'parquet' sur AWS



The screenshot displays the AWS S3 console interface. On the left, the 'Amazon S3' sidebar is visible, showing navigation options like 'Buckets', 'Access Points', and 'Storage Lens'. The main content area shows a bucket named 'datasvg' with a list of objects. The objects are parquet files, each with a unique ID and a size. The table columns are: Name, Type, Last modified, Size, and Storage class. The files are listed in descending order of size.

Name	Type	Last modified	Size	Storage class
e3f2454e4bb4-1388-1-c000.snappy.parquet	parquet	(UTC+01:00)	KB	Standard
part-00011-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1389-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:05 (UTC+01:00)	36.6 KB	Standard
part-00012-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1390-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:04 (UTC+01:00)	25.7 KB	Standard
part-00013-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1391-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:04 (UTC+01:00)	15.0 KB	Standard
part-00014-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1392-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:04 (UTC+01:00)	14.7 KB	Standard
part-00015-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1393-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:04 (UTC+01:00)	14.9 KB	Standard
part-00016-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1394-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:08 (UTC+01:00)	27.2 KB	Standard
part-00017-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1395-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:08 (UTC+01:00)	38.5 KB	Standard
part-00018-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1396-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:11 (UTC+01:00)	39.7 KB	Standard
part-00019-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1397-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:08 (UTC+01:00)	37.9 KB	Standard
part-00020-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1398-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:08 (UTC+01:00)	38.3 KB	Standard
part-00021-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1399-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:08 (UTC+01:00)	51.0 KB	Standard
part-00022-tid-6307710118559107273-6d075f0b-7969-4d54-900e-e3f2454e4bb4-1400-1-c000.snappy.parquet	parquet	December 29, 2022, 02:36:09 (UTC+01:00)	48.7 KB	Standard

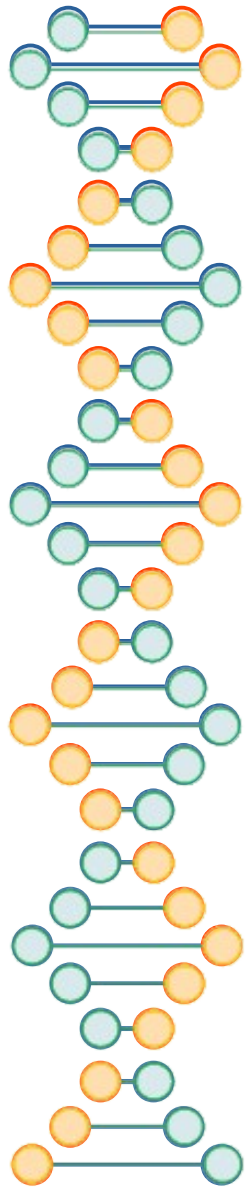




## Limites et améliorations

- L'utilisation de Spark, accélère beaucoup le processus des calculs, c'est extrêmement rapide comparé à pandas.
- Databricks est un environnement simple d'utilisation qui étant lié à AWS nous facilite beaucoup la tâche tout en pouvant stocker de grosses bases de données sur le web.
- AWS est très puissant. Il permet de réaliser plein de choses au-delà de sa liaison avec Databricks. L'unique soucis c'est la question des coups.
- L'utilisation de ces outils est donc franchement recommandée, mais il convient d'utiliser une stratégie adapté aux rapport besoin/coups de la boîte pour ne pas se retrouver avec quelque chose qui rapporte moins que ce qu'il coûte





MERCI

