



OPENCLASSROOMS

## Présentation Projet 8

«Déployez un modèle dans  
le *cloud*»

---

Denis Desoubzdanne

31 Janvier 2024

Formation *Data Scientist*

# Sommaire

1. Présentation du sujet
2. Jeu de données et environnement local
3. Prise en main d'un script *PySpark*
4. Déploiement de la solution dans le *cloud*
5. Démonstration de *Databricks*
6. Livrables et conclusion

# 01

## Présentation du sujet

Objectif, tâches

# Data Scientist



## Fruits!

Start-up de l'AgriTech "Fruits!" souhaitant développer des solutions innovantes pour la récolte des fruits, dont une application mobile pour obtenir des informations des fruits à partir de leurs images photographiques

- Mon objectif : **Développer un modèle** de classification (images de fruits) et le déployer dans une **architecture Big Data**
- Mes tâches :
  - S'approprier et compléter **un script en PySpark** pour traiter quelques images en **local**
  - Définir et construire une **architecture Big Data** sur une plateforme *cloud*
  - **Démonstration** de l'exécution du script et de l'architecture choisie

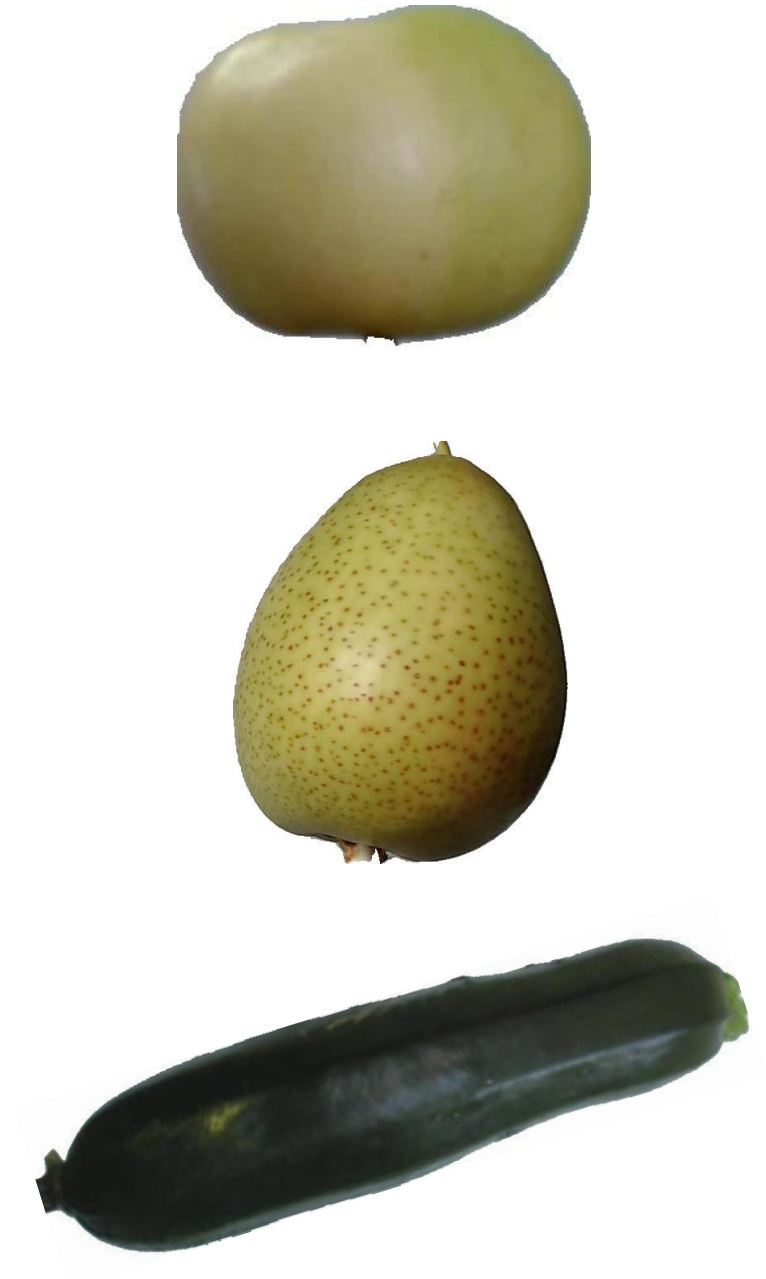
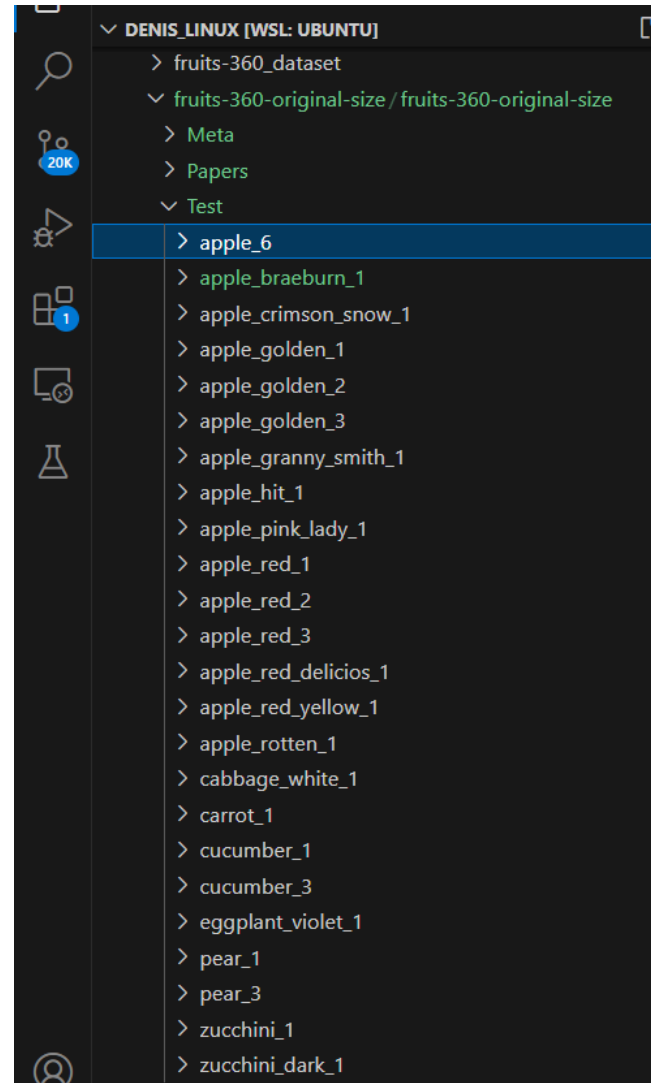
# 02

## Jeux de données & env. local

*Datasets, Linux*

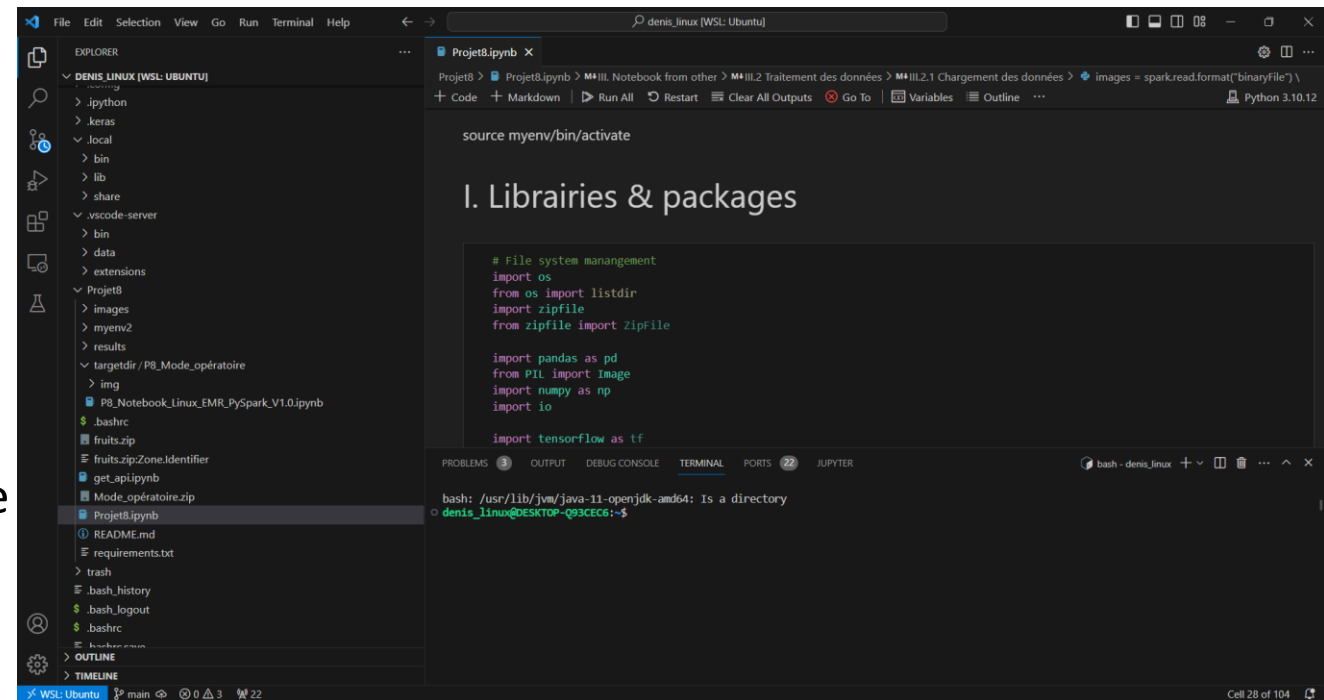
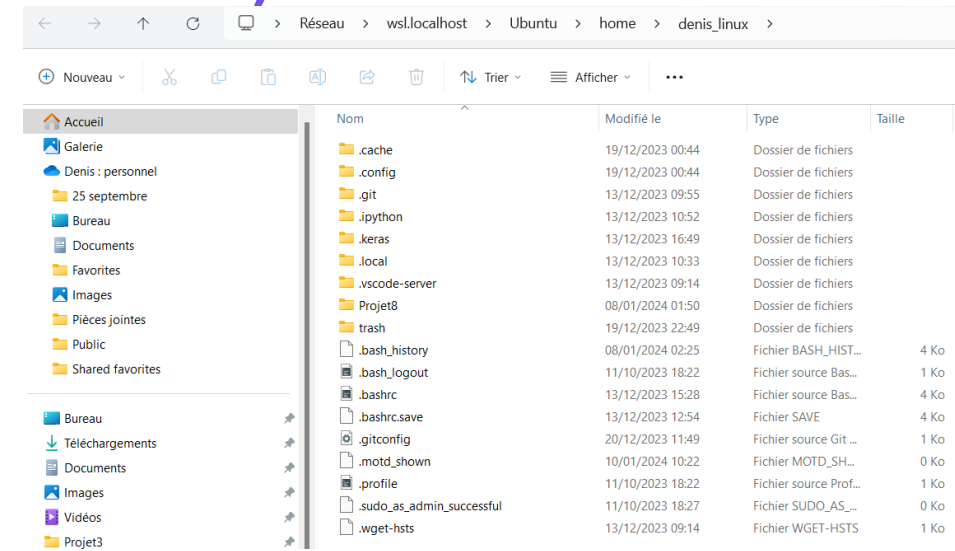
# Datasets

- 1 fichier 'fruits.zip' à télécharger
  - 2 dossiers : *fruits-360\_dataset* & *fruits-360-original-size*
- Données utilisées : **Test**
  - **24 catégories** de fruits/légumes : « apple », « carrot », « pear »...
  - **3110 images** (< 100 Ko/image)



# Développement en local (Linux)

- Installation d'un environnement Linux sous Windows avec **WSL (Windows Subsystem for Linux)**
- Distribution choisie : **Ubuntu**
- **/!\ éviter Windows**, car :
  - Plus difficile de configurer PySpark
  - Plus lent
  - Problèmes de déploiement sur une plateforme *cloud* (Linux)



# 03

## *Notebook PySpark*

*Utilisation d'un notebook fourni*

*SparkSession, Transfer Learning et PCA*



# Avantages de *PySpark* (vs *Pandas*)

- Association Python + (Apache) Spark



- Spark est écrit en **Scala**

- **API Python pour Spark**

- Spark :

- Créée en **2009** par des chercheurs de Berkeley puis vendu à Apache en **2013**

- Framework open source pour du **calcul distribué de données massives**

- Exécute la totalité des opérations **d'analyse de données en mémoire et en temps réel**



- « **Lazy evaluation** » : transformation des datasets pas effectuées immédiatement

# SparkSession & calcul distribué

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

La **SparkSession** est point d'entrée permettant d'interagir avec Spark

```
sc = spark.sparkContext
```

Le **SparkContext** permet à l'application Spark d'accéder aux ressources du cluster et de coordonner les tâches de traitement des données.

SparkSession - in-memory

SparkContext

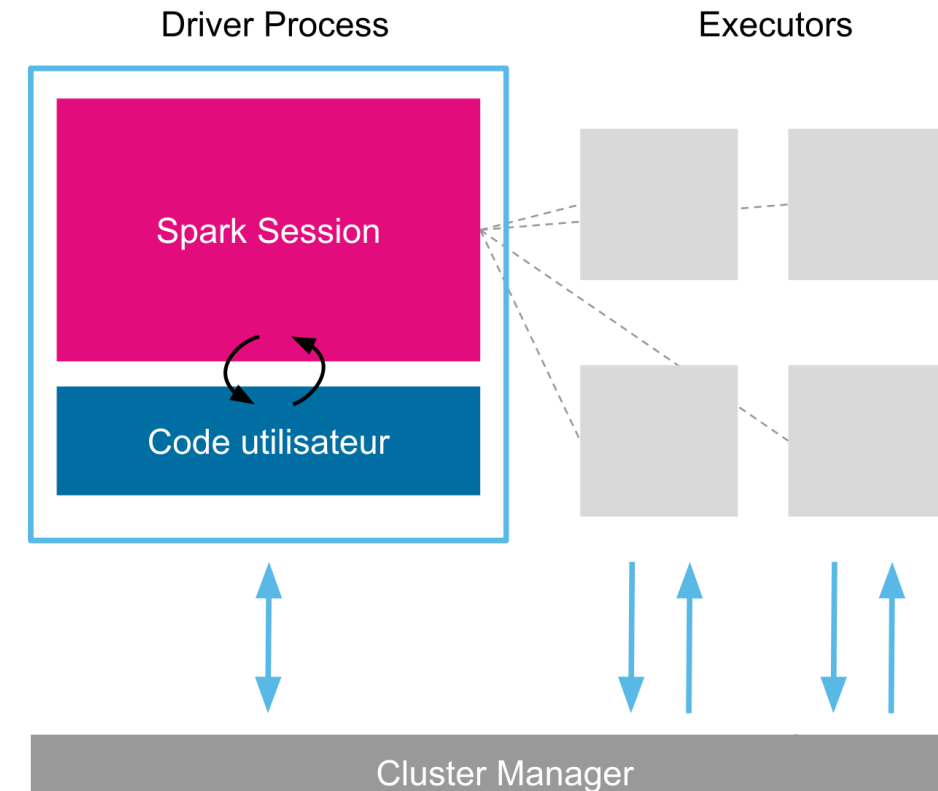
[Spark UI](#)

Version  
v3.5.0

Master  
local

AppName  
P8

Affichage des informations de la **SparkSession** en cours d'exécution



# Lecture des données images

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

```
images = images.withColumn('label', element_at(split(images['path'], '/'), -2))
print(images.printSchema())
print(images.select('path', 'label').show(5, False))
```

Python

root

```
-- path: string (nullable = true)
-- modificationTime: timestamp (nullable = true)
-- length: long (nullable = true)
-- content: binary (nullable = true)
-- label: string (nullable = true)
```

None

path	label
file:/home/denis_linux/Projet8/images/fruits-360-original-size/fruits-360-original-size/Test/apple_hit_1/r0_115.jpg	apple_hit_1
file:/home/denis_linux/Projet8/images/fruits-360-original-size/fruits-360-original-size/Test/apple_hit_1/r0_119.jpg	apple_hit_1
file:/home/denis_linux/Projet8/images/fruits-360-original-size/fruits-360-original-size/Test/apple_hit_1/r0_107.jpg	apple_hit_1
file:/home/denis_linux/Projet8/images/fruits-360-original-size/fruits-360-original-size/Test/apple_hit_1/r0_143.jpg	apple_hit_1
file:/home/denis_linux/Projet8/images/fruits-360-original-size/fruits-360-original-size/Test/apple_hit_1/r0_111.jpg	apple_hit_1

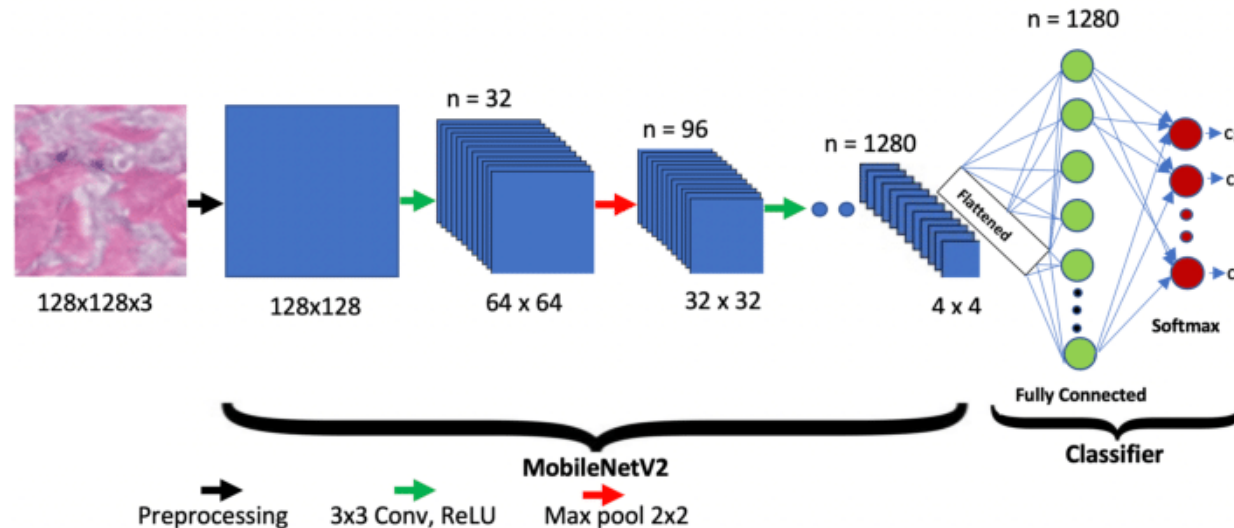
- Lecture des images
- Affichage des 5 premières images (**DataFrame Spark**)

# Transfer Learning : configuration du modèle

```
def model_fn():  
    """  
    Returns a MobileNetV2 model with top layer removed  
    and broadcasted pretrained weights.  
    """  
    model = MobileNetV2(weights='imagenet',  
                        include_top=True,  
                        input_shape=(224, 224, 3))  
    for layer in model.layers:  
        layer.trainable = False  
    new_model = Model(inputs=model.input,  
                    outputs=model.layers[-2].output)  
    new_model.set_weights(broadcast_weights.value)  
    return new_model
```

```
broadcast_weights = sc.broadcast(new_model.get_weights())
```

- Chargement du modèle **MobileNetV2** avec les poids précalculés issus d'**Imagenet**
- Création d'**un nouveau modèle** en retirant la dernière couche (de classification) => récupération d'**arrays** (dim = 1280)
- **Distribution des poids** du modèle à travers plusieurs nœuds (instances de traitement)



# Transfer Learning : featurisation

```
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)

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)
```

***Prétraiter les images + transformation***

***Featuriser une série d'images***

***Itération sur l'ensemble des images***

```
features_df = images.repartition(20).select(col("path"),
| | | | | | | | | | col("label"),
| | | | | | | | | | featurize_udf("content").alias("features")
| | | | | | | | | | )
```

***Featurisation de l'ensemble des données fractionnées en 20 parties (tâches)***

# Enregistrement au format parquet

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

Python

```
2023-12-19 22:55:37.786330: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different r
2023-12-19 22:55:37.790019: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU wil
2023-12-19 22:55:37.836343: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Att
2023-12-19 22:55:37.836424: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Att
2023-12-19 22:55:37.839934: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: A
2023-12-19 22:55:37.849011: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU wil
2023-12-19 22:55:37.849247: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use avail
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate
2023-12-19 22:55:38.596019: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
5/5 [=====] - 3s 302ms/step
5/5 [=====] - 2s 316ms/step (1 +
5/5 [=====] - 2s 335ms/step (2 + 1)
5/5 [=====] - 2s 271ms/step (3 + 1)
5/5 [=====] - 3s 279ms/step (4 +
5/5 [=====] - 2s 305ms/step (5 +
5/5 [=====] - 2s 287ms/step (6 +
5/5 [=====] - 2s 266ms/step (7 + 1)
```

**Ecriture des données en  
format parquet**



- Stockage des données par col.
- Données compressées
- Optimisé pour les syst. distribués...

```
type(features_df)
```

```
pyspark.sql.dataframe.DataFrame
```

# ACP => reduction de dimensions

```
# fct utilisateur pour convertir ArrayType en VectorUDT
def array_to_vector_udf(array_col):
    return Vectors.dense(array_col)
```

```
#enregistrement de la fct utilisateur en tant qu'UDF
array_to_vector = udf(array_to_vector_udf, VectorUDT())
features_df = features_df.withColumn('vector_features', array_to_vector('features'))
```

Conversion des données **arrays** en données **vectorielles UDT** (User-Defined Type)

**/!\ Particularité des ACPs sous PySpark!!**

```
features_df.show()
```

```
+-----+-----+-----+-----+
|      path|      label|      features|      vector_features|
+-----+-----+-----+-----+
|file:/home/denis_...| apple_hit_1|[0.2560484, 0.313...|[0.25604841113090...|
|file:/home/denis_...| apple_hit_1|[0.28270248, 0.36...|[0.28270247578620...|
|file:/home/denis_...| apple_hit_1|[0.32698685, 0.81...|[0.32698684930801...|
|file:/home/denis_...| apple_hit_1|[0.002047996, 0.2...|[0.00204799603670...|
|file:/home/denis_...|cabbage_white_1|[0.0, 0.6773124, ...|[0.0,0.6773123741...|
|file:/home/denis_...| apple_hit_1|[0.34845182, 0.02...|[0.34845182299613...|
|file:/home/denis_...| apple_hit_1|[0.5632592, 0.0, ...|[0.56325918436050...|
|file:/home/denis_...|cabbage_white_1|[0.0, 0.59185535, ...|[0.0,0.5918553471...|
|file:/home/denis_...|cabbage_white_1|[0.0, 1.7062128, ...|[0.0,1.7062127590...|
|file:/home/denis_...| apple_hit_1|[0.09217532, 0.48...|[0.09217531979884...|
|file:/home/denis_...| apple_hit_1|[0.011136882, 0.0...|[0.01113688200712...|
|file:/home/denis_...|cabbage_white_1|[0.0, 0.3722937, ...|[0.0,0.3722937107...|
|file:/home/denis_...| apple_hit_1|[0.89603764, 0.0...|[0.89603763818740...|
|file:/home/denis_...| apple_hit_1|[0.4900368, 0.459...|[0.49003678560256...|
|file:/home/denis_...| pear_3|[0.54662186, 0.21...|[0.54662185907363...|
|file:/home/denis_...| pear_3|[0.50064385, 0.0...|[0.50064384937286...|
|file:/home/denis_...| pear_3|[0.5533866, 0.026...|[0.55338662862777...|
|file:/home/denis_...| pear_3|[0.17466481, 0.00...|[0.17466481029987...|
|file:/home/denis_...| apple_red_3|[0.75871104, 0.12...|[0.75871104001998...|
|file:/home/denis_...| apple_red_3|[0.041301727, 0.2...|[0.04130172729492...|
+-----+-----+-----+-----+
only showing top 20 rows
```

Affichage du **nouveau PySpark Dataframe**

# PCA : choix du nombre de composantes

```
num_elements
✓ 0.0s
1280
```

```
pca = PCA(k=num_elements, inputCol='vector_features')
pca_model = pca.fit(features_df)
```

} *Modèle ajusté aux données pour 1280 composantes*

```
pca_exp = pca_model.explainedVariance
```

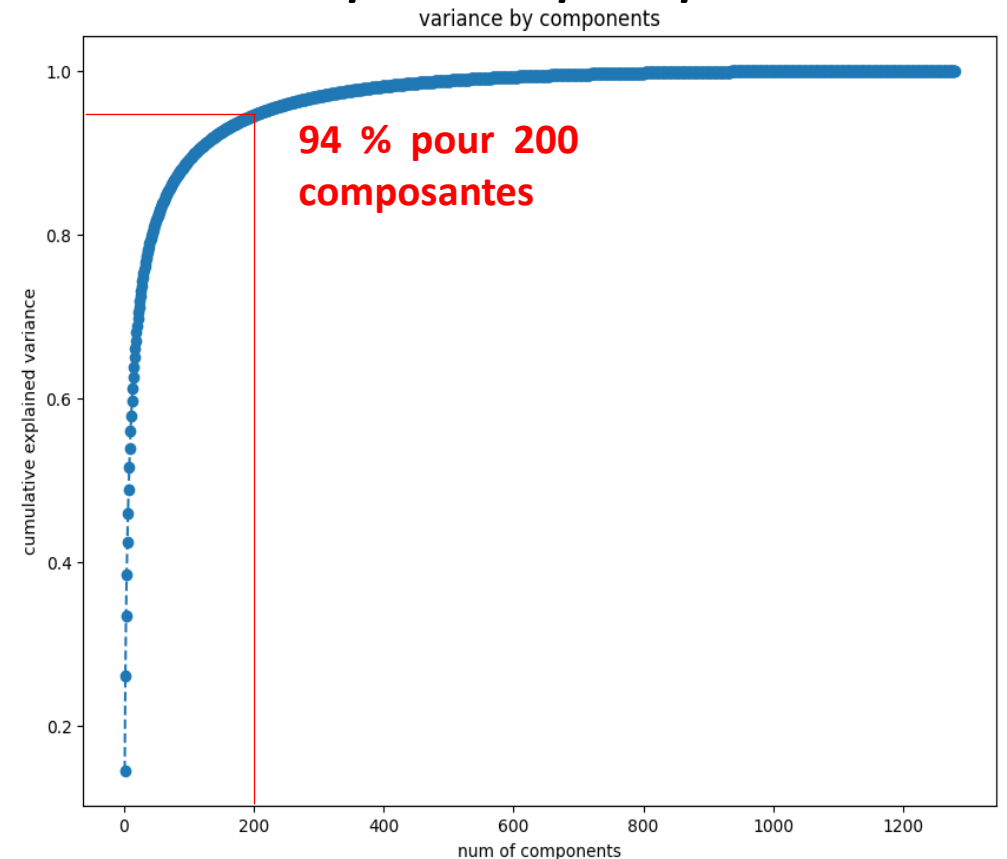
```
cumValues = pca_exp.cumsum() # get the cumulative values
```

```
cumValues[199] #selection of the 200 st components (new_features)
```

```
0.9456581695702824
```

```
# plot the graph
plt.figure(figsize=(10,8))
plt.plot(range(1,num_elements+1), cumValues, marker = 'o', linestyle='--')
plt.title('variance by components')
plt.xlabel('num of components')
plt.ylabel('cumulative explained variance')
plt.show()
```

*Variance expliquée cumulée en fct du nb de composantes principales*





# PCA : transformation et sauvegarde des données

```
pca_features = pca_model.transform(features_df)
```

```
# extract the 200 st components
# define an user fct to extract the 2 first elts
def extract_first_n_udf(features, n):
    return Vectors.dense(features[:n])

extract_first_n = udf(extract_first_n_udf, VectorUDT()) #enregistrer la fct utilisateur en tant q'UDF

#specify the number to extract (ex:)
n_elements = 200

#appliquer l'UDF pour extraire les n premiers elts
pca_features = pca_features.withColumn('pca200', extract_first_n(col('pca_features'), lit(n_elements)))
```

```
features_df = features_df.join(pca_features.select('path', 'pca200'), 'path')
```

```
# Save the DataFrame to a Parquet file
features_df.write.mode("overwrite").parquet('mypca')
```

**Transformation des données sur 1280 composantes**

**Récupération des 200 premières valeurs (composantes)**

**Jointure de table pour incorporer les nouvelles features**

**Ecriture des données en format parquet**

# 04

## *Déploiement de la solution dans le cloud*

*Solution de stockage*

*Environnement de travail*

# Outils utilisés



Solution	Description
Databricks	Plateforme d'analyse interactive
GitHub	Plateforme de <i>versioning</i>
Microsoft Azure	Plateforme informatique <i>Cloud</i> (solution de stockage)

# Présentation de Databricks



- Développé par les créateurs d'Apache Spark (2013)
  - Plateforme web pour travailler avec Spark
  - Solution de stockage et d'analyse (ML, dashboard, SQL...)
  - Fourni une gestion automatique de *clusters* et l'utilisation de *Notebooks iPython*
- Intégration aux environnements de *cloud* distribués

➤ *Microsoft Azure*



➤ AWS (Amazon)

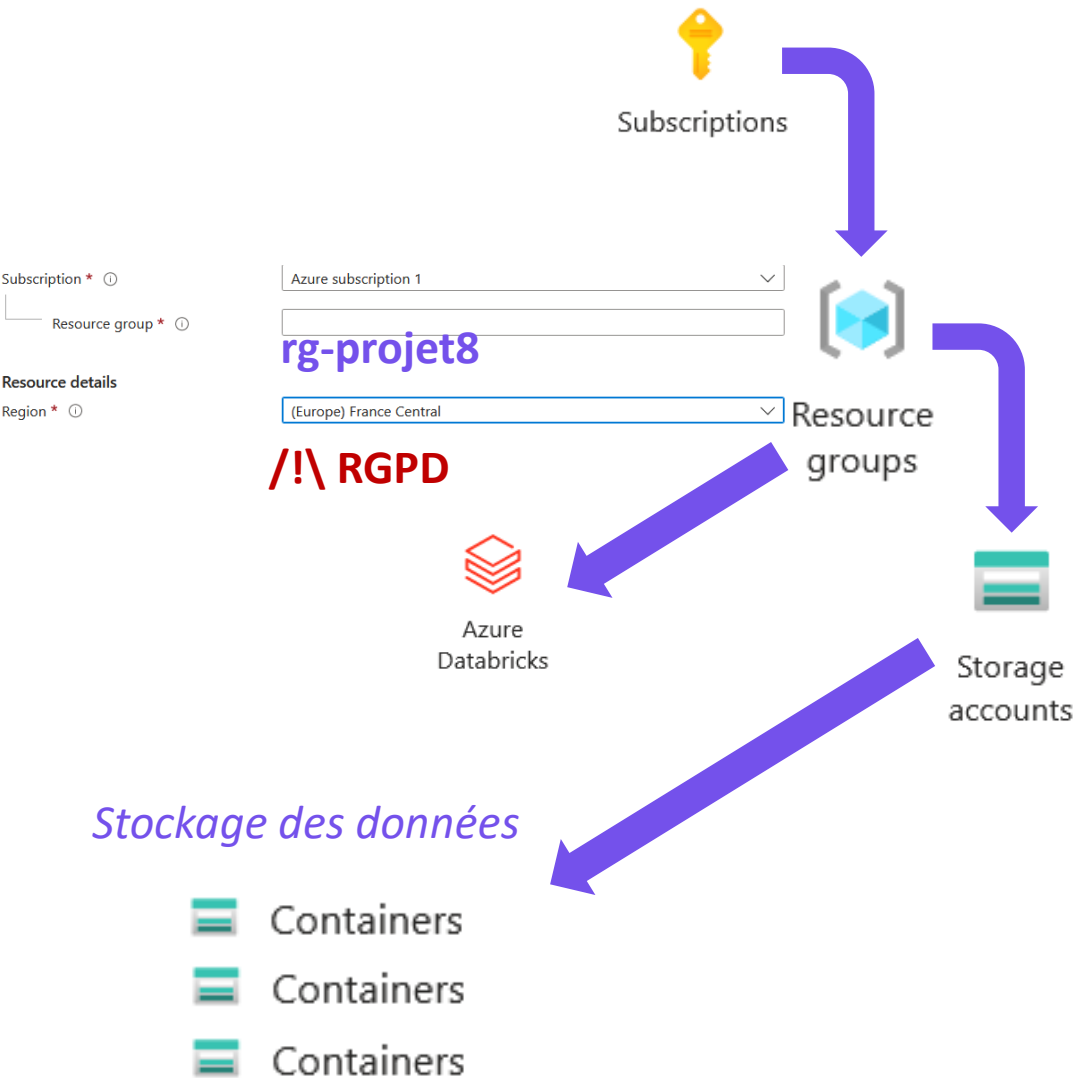


➤ Google Cloud Platform



*Solution choisie !*

# Environnement Azure



## Azure services



## Resources

Recent Favorite

Name	Type	Last Viewed
projet8images	Storage account	31 minutes ago
databdd	Azure Databricks Service	an hour ago
rg-projet8	Resource group	an hour ago
data-shared-account	Data Share	an hour ago
MyStreamlit	App Service	2 weeks ago
BasicwebappVL	App Service	2 weeks ago
Azure subscription 1	Subscription	a month ago
rg-projet7	Resource group	2 months ago

[See all](#)

## projet8images

Storage account

Search

Overview

- Activity log
- Tags
- Diagnose and solve problems
- Access Control (IAM)
- Data migration
- Events
- Storage browser
- Storage Mover

Data storage

Containers

**/!\ Uploader ou downloader des données**

Upload Open in Explorer Delete Move Refresh Open in Explorer

Essentials

Resource group (move) : rg-projet8

Location : francecentral

Subscription (move) : Azure subscription 1

Subscription ID : 06b74d93-ffa5-4415-9d97-51b6f6702be1

Disk state : Available

Tags (edit) : storage : ok

Properties

Monitoring

Capabilities (7)

Recommendations (0)

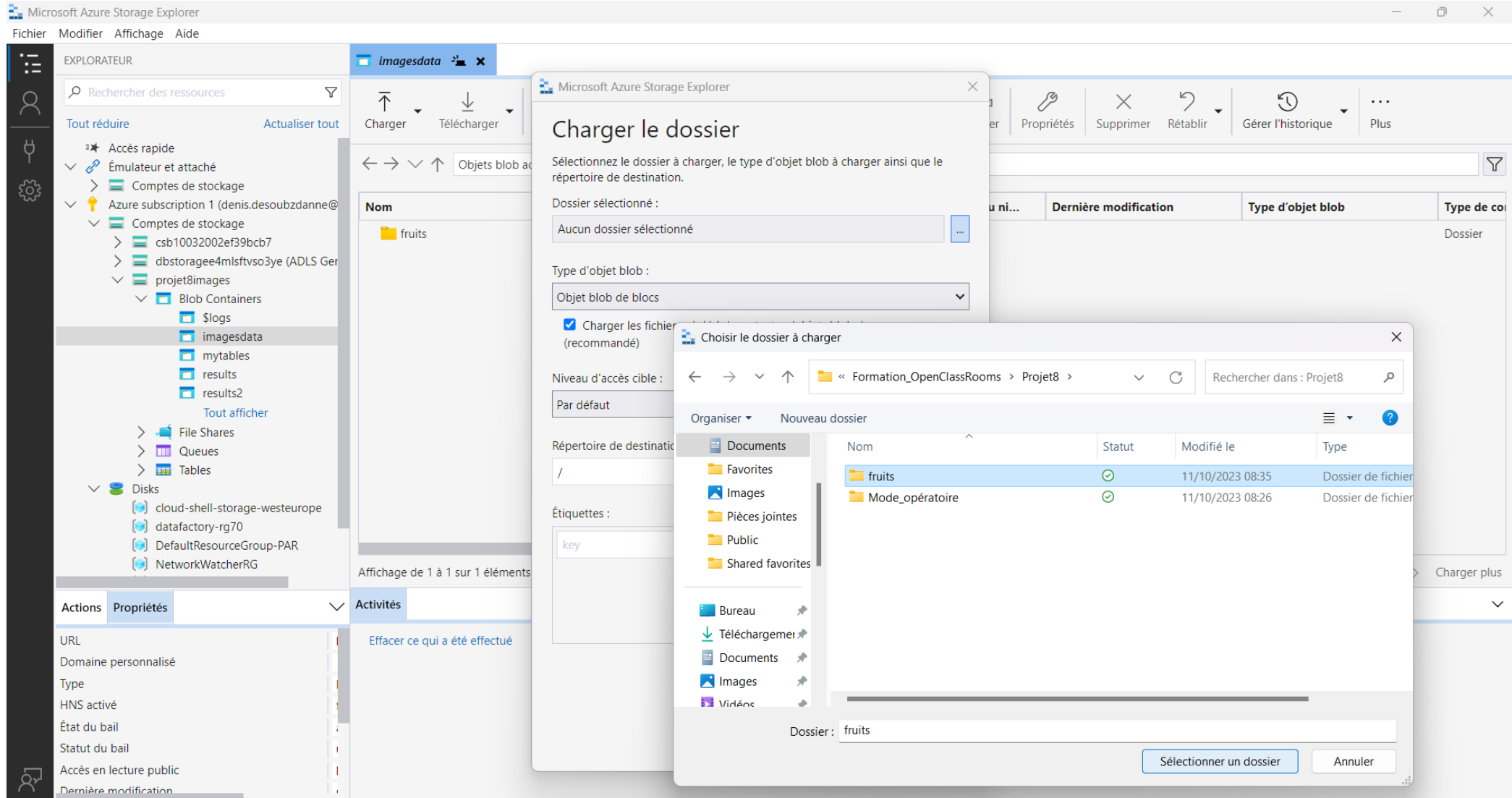
Tutorials

Blob service

Hierarchical namespace

Disabled

# Stockage et accès des données *via* Azure Storage Explorer



# Creation de containers sous Microsoft Azure

Microsoft Azure

Search resources, services, and docs (G+)

denis.desoubzdanne@g...  
RÉPERTOIRE PAR DÉFAUT

Home > projet8images

projet8images | Containers

Storage account

Search

Container Change access level Restore containers Refresh Delete Give feedback

Search containers by prefix

Name	Last modified	Anonymous
<input type="checkbox"/> \$logs	12/20/2023, 9:55:56 AM	Private
<input type="checkbox"/> imagesdata	1/10/2024, 5:17:17 PM	Blob
<input type="checkbox"/> mytables	1/23/2024, 10:29:28 PM	Container
<input type="checkbox"/> results	1/10/2024, 5:21:51 PM	Blob
<input type="checkbox"/> results2	1/23/2024, 10:26:31 PM	Container

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Storage Mover

Data storage

Containers

File shares

Queues

Tables

Security + networking

Networking

New container

Name \*  
zipfile

Anonymous access level ⓘ  
Container (anonymous read access for containers and blobs) Private (no anonymous access) Blob (anonymous read access for blobs only) Container (anonymous read access for containers and blobs)

Advanced

Create

Give feedback

# Stockage des images dans un conteneur

Microsoft Azure

Search resources, services, and docs (G+/)

denis.desoubzdanne@g...  
RÉPERTOIRE PAR DÉFAUT

Home > projet8images | Containers >

zipfile  
Container

Search

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Upload Change access level Refresh Delete Change tier Acquire lease

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: zipfile

Search blobs by prefix (case-sensitive)

Add filter

Name	Modified	Access tier	Archived
No results			

Upload blob

Uploading on blob(s)...  
Attempting to upload 1 blobs(s)

Drag and drop files here  
or  
Browse for files

☐ Overwrite if files already exist

Advanced

Upload

Give feedback

Current uploads

Dismiss: Completed All

fruits.zip 4 MiB / 1.28 GiB



# Téléchargement des images (zip file) : lien url

Microsoft Azure

Search resources, services, and docs (G+/)

denis.desoubzdanne@g...  
RÉPERTOIRE PAR DÉFAUT

Home > projet8images | Containers > zipfile >

zipfile  
Container

Search

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Upload

Change access level

...

Authentication method: Access key [\(Switch to Microsoft Entra user account\)](#)

Location: zipfile

Search blobs by prefix (case-...

Show deleted blobs

Add filter

Name

fruits.zip

fruits.zip

Blob

Save

Discard

Download

Refresh

Delete

Change tier

Acquire lease

Break lease

Give feedback

Overview

Versions

Snapshots

Download

Generate SAS

Properties

URL	<div>https://projet8images.bl ...</div>
LAST MODIFIED	1/24/2024, 10:46:07 AM
CREATION TIME	1/24/2024, 10:46:07 AM
VERSION ID	-
TYPE	Block blob
SIZE	1.28 GiB
ACCESS TIER	Hot (Inferred)
ACCESS TIER LAST MODIFIED	N/A
ARCHIVE STATUS	-
REHYDRATE PRIORITY	-
SERVER ENCRYPTED	true
ETAG	0x8DC1CC149FEC046
VERSION-LEVEL IMMUTABILITY POLICY	Disabled
CACHE-CONTROL	
CONTENT-TYPE	application/x-zip-compressed
CONTENT-MD5	

# Création d'un compte (Azure) Databricks



## Azure services

[Create a resource](#)

Azure Databricks

Resource groups

Cost Management ...

SQL databases

Microsoft Entra ID

Data factories

Reservations

Subscriptions

[More service](#)

## Resources

[Recent](#) [Favorite](#)

Name	Type	Last Viewed
projet8images	Storage account	9 minutes ago
databdd	Azure Databricks Service	2 hours ago
rg-projet8	Resource group	2 hours ago

## Create an Azure Databricks workspace ...

### Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Azure subscription 1

Resource group \* ⓘ

Create new

### Instance Details

Workspace name \*

Enter name for Databricks workspace

Region \*

France Central

Pricing Tier \* ⓘ

Premium (+ Role-based access controls)


We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.

Managed Resource Group name

Enter name for managed resource group

# Mise en place d'un cluster (machine)

Microsoft Azure

 databricks

CTRL + P

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning



Experiments

Features

Models

Serving

Compute > Preview [Send feedback](#)

**DDesouCluster1**  

[Configuration](#) [Notebooks \(0\)](#) [Libraries](#) [Event log](#) [Spark UI](#) [Driver logs](#) [Metrics](#) [Apps](#) [Spark compute UI - Master](#)

Policy ⓘ

Personal Compute

Access mode ⓘ

Single user access ⓘ

Single user

DDesou Desoubzdanne

Performance

Databricks Runtime Version

14.2 ML (includes Apache Spark 3.5.0, Scala 2.12)

☐ Use Photon Acceleration ⓘ

Node type ⓘ

Standard\_DS3\_v2 14 GB Memory, 4 Cores

☒ Terminate after 30 minutes of inactivity ⓘ

Tags ⓘ

clusterone ok

> Automatically added tags

▶ Advanced options

More

Start

Edit

[UI](#) | [JSON](#)

Summary

1 Driver

14 GB Memory, 4 Cores

Runtime

14.2.x-cpu-ml-scala2.12

Standard\_DS3\_v2

0.75 DBU/h

Configuration de la machine

27

# Module Spark UI (Compute)

Microsoft Azure

databricks

CTRL + P

denis.desoubzdanne@gmail.com

More

Terminate

Edit

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Experiments

Features

Models

Serving

Compute > Preview

Send feedback

DDesouCluster1

Configuration

Notebooks (1)

Libraries

Event log

Spark UI

Driver logs

Metrics

Apps

Spark compute UI - Master

Jobs

Stages

Storage

Environment

Executors

SQL / DataFrame

JDBC/ODBC Server

Structured Streaming

Connect

Scheduling Mode: FAIR

Active Jobs: 2

Completed Jobs: 39

Failed Jobs: 1

Event Timeline

Enable zooming

Executors

Added

Removed

Jobs

Succeeded

Failed

Running

2024/01/23 21:00

05

10

15

20

25

30

35

40

Active Jobs (2)

Page: 1


1 Pages. Jump to 1. Show 100 items in a page. Go


Stages:

Tasks (for all stages):

# Workspace connecté à un repos GitHub

Microsoft Azure

 databricks

 Search data, notebooks, recents, and more...

CTRL + P

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Experiments

Features

Models

Serving

Marketplace

Partner Connect

Collapse menu

Settings

Workspace admin

Appearance

Identity and access

Security

Compute

Development

Notifications

Advanced

User

Profile

Preferences

Developer

Linked accounts

Notifications

Linked accounts

Connect your Databricks account to other services

Git integration

With co-versioned repo

Databricks Repos allows you to clone a remote Git repo, which you can specify when you add a repo. [Learn more](#)

With individual notebooks

Although we recommended using co-versioned repo for Git integration, Databricks supports individual notebook version control integration with [GitHub](#), [Bitbucket Cloud](#), or [Azure DevOps Services](#) (using AAD authentication only).

Set your Git provider and credentials

You can also set your Git provider credentials via API. [Learn more](#)










✔ GitHub (Linked)

DDesou

Configure in GitHub

Unlink

DDesou / Proj8

Proj8

databricks

2 branches

0 tags

Go to file

Add file

Code

About

This branch is 1 commit ahead of main.

Contribute

DDesou and DDesou v2\_notebook

17 hours ago

6 commits

Proj8.ipynb

v2\_notebook

17 hours ago

Proj8\_v2.py

v2\_notebook

17 hours ago

README.md

Update README.md

last month

README

Proj8

Here is the Readme for the Proj8 (OCR).

Readme

Activity

Stars

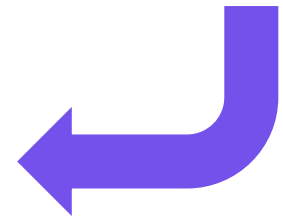
Watching

Forks

Releases

Package

Language



29

# 05

## *Démonstration de Databricks*


*Environnement de travail*

*Cluster*

*Notebook*


# Dashboard Databricks


Microsoft Azure

 databricks

CTRL + P

databdd





denis.desoubzdanne@gmail.com

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

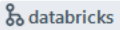
Experiments

Features

Models

Serving

Proj8\_v2

 databricks

Python

★

File

Edit

View

Run

Help

Last edit was 10 hours ago

Provide feedback

▶ Run all

■ Terminated


📅 Schedule

Share

0. Creation of the mounts

data container mount

```
1 storage_account_name = "projet8images"
2 container_name = "imagesdata"
3 storage_account_key = "zez1wQv/LGqyCCNn1LAKv0jTyT29ilq0/ccqsEHT1X4CyK01Ha7b0APu7xm5xglPmnCU1UVFvRnb+AStFf0ftw=="
4
5 dbutils.fs.mount(
6     source = f"wasbs://{container_name}@{storage_account_name}.blob.core.windows.net/",
7     mount_point = "/mnt/projet8",
8     extra_configs = {"fs.azure.account.key.projet8images.blob.core.windows.net": storage_account_key}
9 )
```

⊕ java.rmi.RemoteException: java.lang.IllegalArgumentException: requirement failed: Directory already mounted: /mnt/projet8; nested exception is:  
 Diagnose error Command took 2.04 seconds -- by denis.desoubzdanne@gmail.com at 23/01/2024 22:06:49 on DDesouCluster1

results container mount

```
1 storage_account_name = "projet8images"
2 container_name = "results2"
```

# 06

## Livrables & conclusion

*Liens url*

*Bilan*



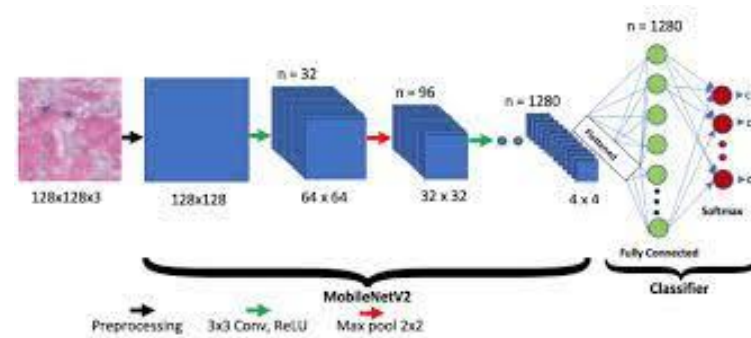
# Liens urls pour les différents livrables

- Lien vers *notebook* Databricks : <https://adb-8983437898642533.13.azuredatabricks.net/browse/folders/163836705254556?o=8983437898642533>
- Lien pour télécharger la table csv de données pca : <https://projet8images.blob.core.windows.net/mytables/table.csv>
- Lien pour télécharger les images (fichier zip) : <https://projet8images.blob.core.windows.net/zipfile/fruits.zip>
- Lien vers le repos GitHub : <https://github.com/DDesou/Projet8.git>

# Conclusion

- Partie 1 : prise en main locale

- Environnement Linux en local (WSL)
- Prise en main de Spark (*Big Data*)
- Extraction de features (*transfer learning*)
- Réduction de dimensions (ACP)



- Partie 2 : déploiement dans un environnement *cloud*

- Microsoft Azure (compte et stockage)
- Databricks (plateforme d'analyse et d'exécution des tâches)
- GitHub (versioning, partage)





OPENCLASSROOMS

Denis Desoubzdanne

31 Janvier 2024

Formation *Data Scientist*

