

Paris, samedi 14 décembre 2019

Ecole Supérieure de Génie Informatique

Global AI Bootcamp

14 DECEMBER 2019





Créateur de réussites numériques

Explorer la donnée en grimpant à l'échelle



AZEO
talents & technology

Merci à l'ESGI, aux communautés
et aux sponsors



ESGI
école supérieure de
génie informatique

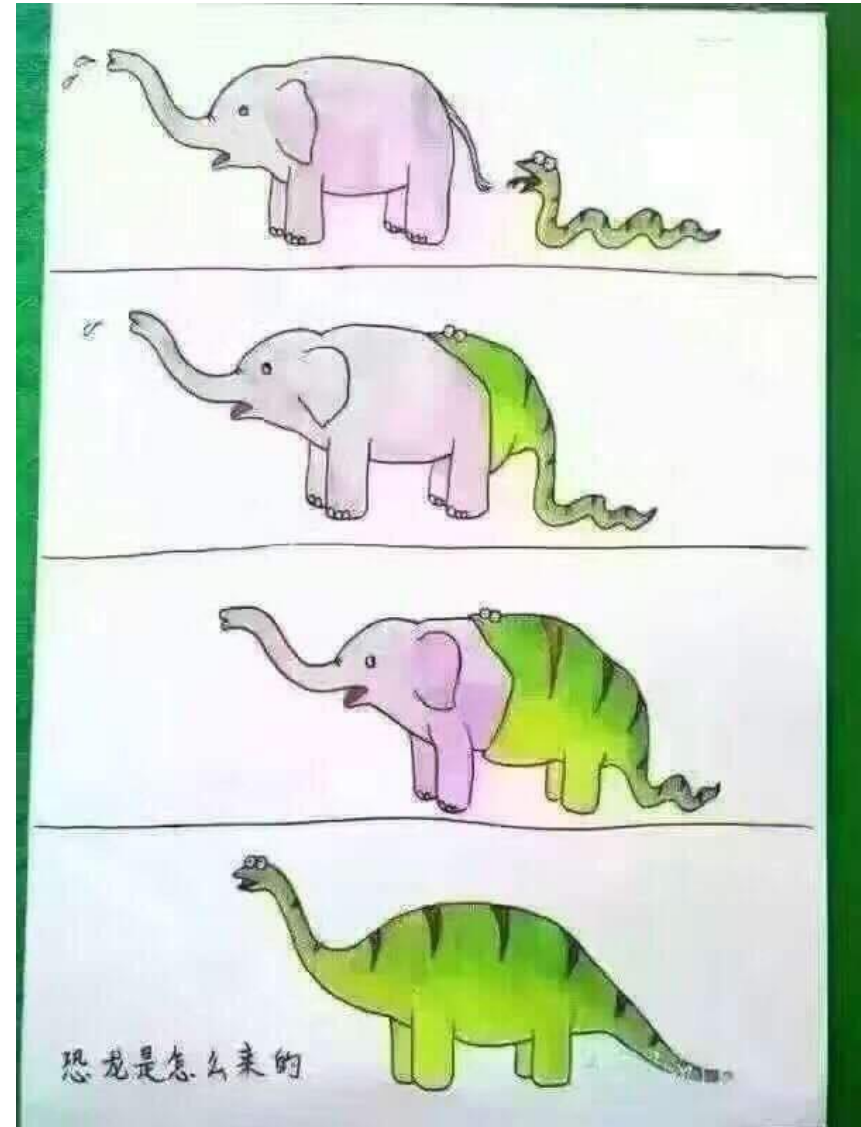
GUS

Local sponsors

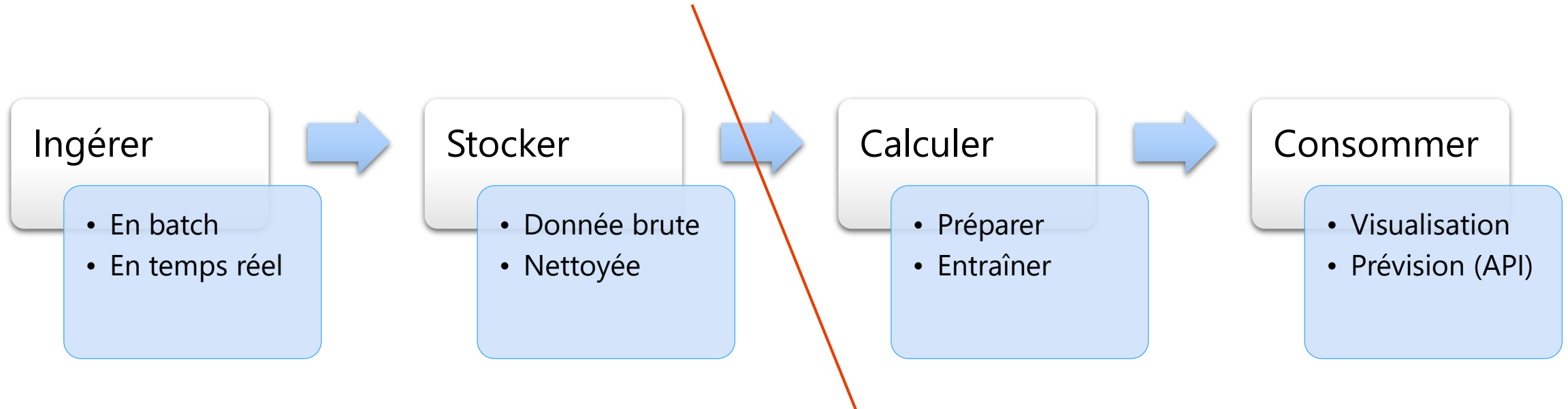
AZEO
talents & technology

cellenza
Does IT better

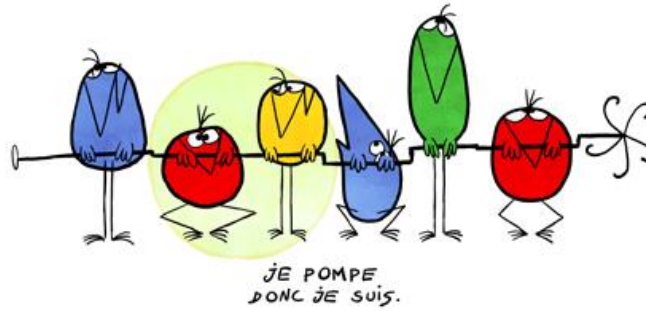
A la fin de cette
session, vous
pourrez rire à
cette blague.



Rappel : les quatre étapes d'un projet data



« *Séparer stockage et traitement* »,
le « *diviser pour mieux régner* »
de l'architecture data



Bien choisir ses outils

Anticiper le passage à l'échelle



— Surtout ! ne parlons pas de l'affaire Dreyfus !



R ou Python pour la Data Science ?

Par Caran d'Ache — Cette image provient de la Bibliothèque en ligne Gallica
sous l'identifiant ARK bpt6k2842896/f3.item, Domaine public,
<https://commons.wikimedia.org/w/index.php?curid=438435>

Objectifs de la session :

Bien débiter

Choisir le bon langage selon ses objectifs

Monter à l'échelle du volume de données

Sommaire :

Python, Anaconda, Pandas, etc.

Jupyter & Jupyter Lab

Azure Notebooks

Data Science Virtual Machine

Azure Databricks



Paul PETON

Samedi 14 décembre 2019



Python, Anaconda, Pandas, etc.



Description du langage Python

Python est un langage de programmation

objet

multi-plateformes

non compilé

multiparadigmes

Il favorise la programmation

impérative structurée

fonctionnelle

orientée objet

Il est doté

d'un typage dynamique fort (« *duck type* »)

d'une gestion automatique de la mémoire par ramasse-miettes

d'un système de gestion d'exceptions



```
import this
"""The Zen of Python, by Tim Peters. (poster by Joachim Jablon)"""

1 Beautiful is better than ugly.
2 Explicit is better than impl..
3 Simple is better than complex.
4 Complex is better than c0mpl@c@ted.
5 Flat is better than nested.
6 Sparse is better than dense.
7 Readability counts.
8 Special cases aren't special enough to break the rules.
9 Although practicality beats purity.
10 raise PythonicError("Errors should never pass silently.")
11 # Unless explicitly silenced.
12 In the face of ambiguity, refuse the temptation to guess.
13 There should be one-- and preferably only one --obvious way to do it.
14 # Although that way may not be obvious at first unless you're Dutch.
15 Now is better than ... never.
16 Although never is often better than rightnow.
17 If the implementation is hard to explain, it's a bad idea.
18 If the implementation is easy to explain, it may be a good idea.
19 Namespaces are one honking great idea -- let's do more of those!
```



Quelques librairies (indispensables) pour la Data Science

numpy : calcul numérique (codé en C et Fortran)

scipy : calcul scientifique

pandas : pour manipulation d'un « dataframe »

(voir aussi Koalas pour une optimisation sous Databricks)

matplotlib : visualisations graphiques

plotly, seaborn : visualisations adaptées aux dataframes Pandas

bokeh : interface interactive de visualisation (équivalent de R Shiny)

statsmodels : méthodes statistiques traditionnels

scikit-learn : algorithmes de Machine Learning

Installation par la commande **pip install <package-name>**

Notion de Pandas DataFrame

Un Pandas DataFrame est un objet *mutable*, une structure tabulaire de données potentiellement hétérogènes (des types différents) à deux dimensions (lignes et colonnes) avec des axes labellisés (index de ligne et nom de colonne)

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

```
# Import pandas package
import pandas as pd

# Define a dictionary containing employee data
data = {'Name': ['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age': [27, 24, 22, 32],
        'Address': ['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification': ['Msc', 'MA', 'MCA', 'Phd']}

# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# select two columns
print(df[['Name', 'Qualification']])
```



« The World's Most Popular Data Science Platform »

Simplifie le management et le déploiement des packages

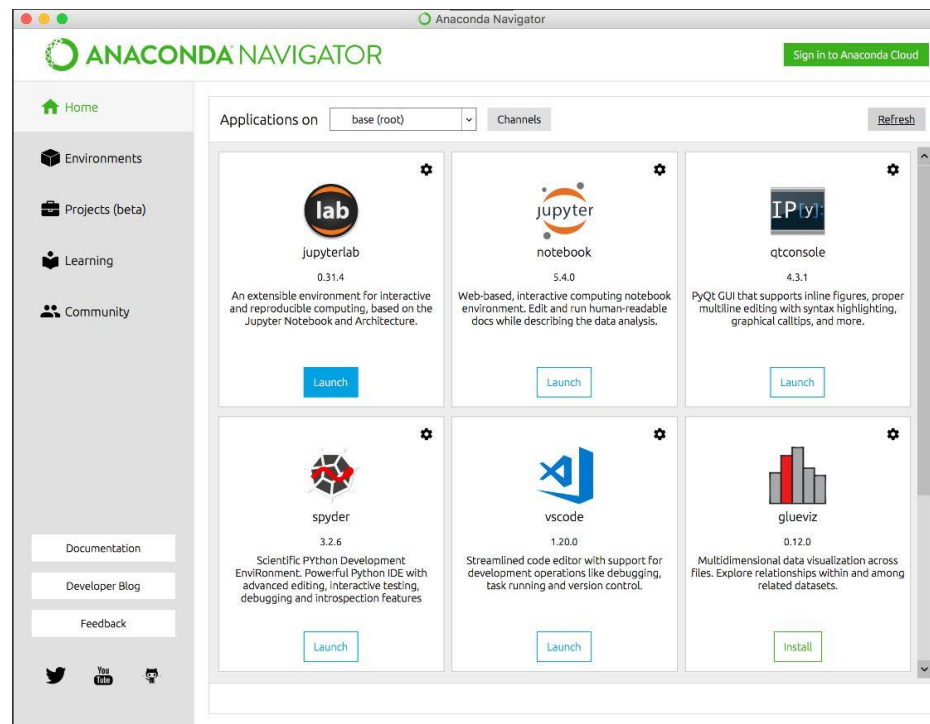
conda analyses the current environment including everything currently installed, and, together with any version limitations specified, works out how to install a compatible set of dependencies, warning if this cannot be done.

Anaconda Prompt

`conda install <package-name>`

Anaconda Navigator

Anaconda Cloud



Choisir la bonne solution selon l'échelle

Solution	Environnement	Licence	Passage à l'échelle	Coût
Python (Anaconda)	Local	Libre / Anaconda		0 + coût du laptop

Jupyter et Jupyter Lab

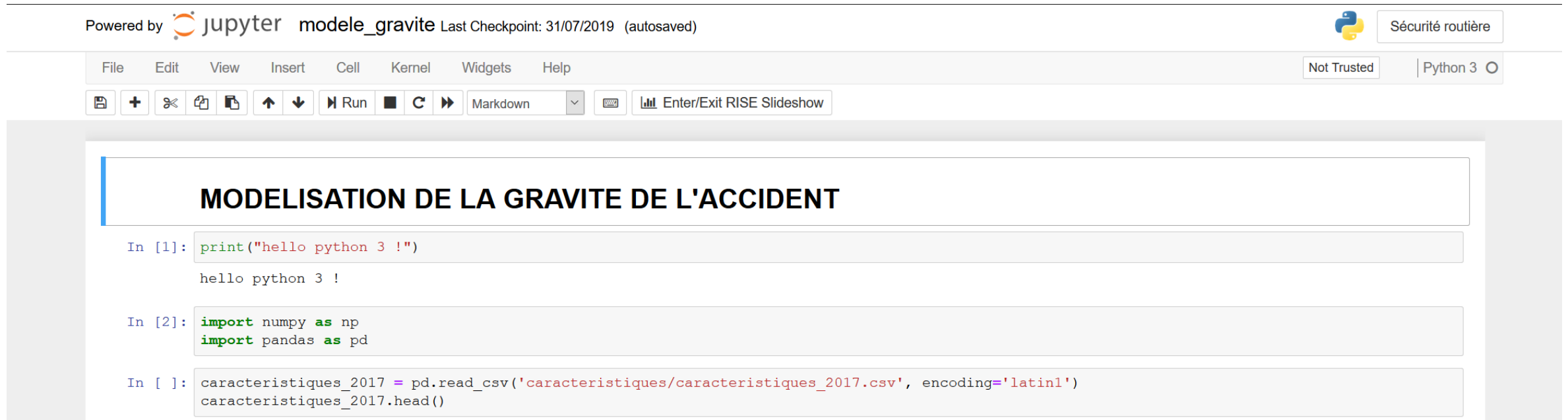
Interpréteur interactif de code

Affichage simultané de code « live », d'images, d'équations... de manière interactive

Une autre approche du mode debug

Permet le partage et le travail collaboratif

Permet d'utiliser une multitude de langages. Pour Jupyter : Julia, Python & R



The screenshot shows a Jupyter Notebook interface. At the top, it says "Powered by jupyter modele_gravite Last Checkpoint: 31/07/2019 (autosaved)". On the right, there's a "Sécurité routière" logo and a "Not Trusted" warning. Below the menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), there's a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area displays a notebook titled "MODELISATION DE LA GRAVITE DE L'ACCIDENT". It contains three code cells:

```
In [1]: print("hello python 3 !")
hello python 3 !

In [2]: import numpy as np
import pandas as pd

In [ ]: caracteristiques_2017 = pd.read_csv('caracteristiques/caracteristiques_2017.csv', encoding='latin1')
caracteristiques_2017.head()
```




— Surtout ! ne parlons pas de l'affaire Dreyfus !



Pour ou contre les notebooks ?

Un notebook intègre du code, ce n'est pas un outil de développement de code. L'outil d'élaboration des codes est un environnement de développement intégré (IDE). Importer des librairies dans les notebooks.

Jupyter Lab, le multi-onglets et bien plus

« Sur-couche » au-dessus de Jupyter

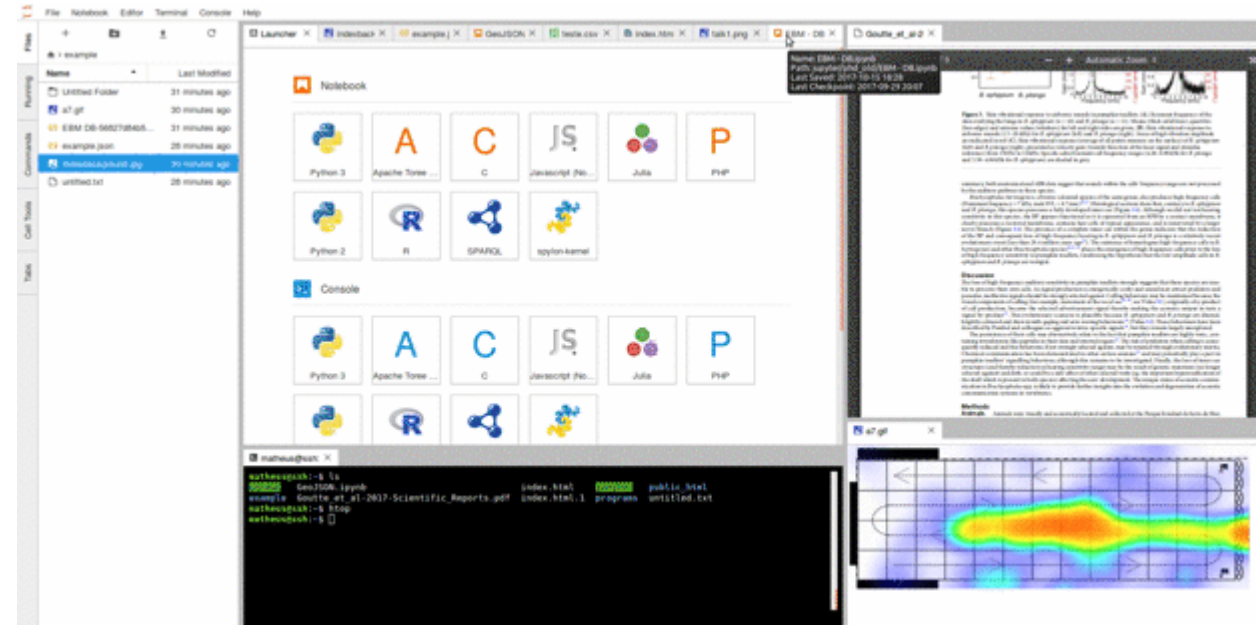
Facilite la navigation par l'explorateur de fichiers

Permet d'ouvrir d'autres types de fichiers que .ipynb

Permet de visualiser plusieurs notebooks sous forme d'onglets

Affichage des résultats dans une vue « output »

Utilisation d'un console dans la même fenêtre



<https://www.stat4decision.com/fr/jupyterlab-python/>

Markdown : mieux que des commentaires !

Permet de réaliser une mise en forme avancée de blocs de texte

Choisir une cellule de type Markdown et l'exécuter une fois renseignée.

Exemples :

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3



Powered by  jupyter modele_gravite Last Checkpoint: 31/07/2019 (autosaved)

 Sécurité routière

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Run Enter/Exit RISE Slideshow

MODELISATION DE LA GRAVITE DE L'ACCIDENT

```
In [1]: print("hello python 3 !")
```



Précéder la commande magique de :

% pour une ligne

%% pour une cellule

%run : exécuter un autre notebook

%load : charger un fichier (par exemple, un .py contenant des librairies ou des fonctions)

%store : passer une variable d'un notebook à un autre

%who : liste toutes les variables du scope globale

%%time : temps d'exécution détaillé de la cellule

%%writefile nomFichier : sauvegarder le contenu d'une cellule dans un fichier

```
from IPython.core.interactiveshell
```

```
import InteractiveShell InteractiveShell.ast_node_interactivity = "all"
```

Pour afficher tous les objets lors de l'exécution d'une cellule (et non uniquement le dernier)



Les dérives classiques du développement sous notebook

Code souvent dupliqué entre cellules

Fonctionnalités similaires implémentées dans plusieurs notebooks

Manque d'abstraction au moyen de fonctions

Mélange de code pour l'exploration et de code pour le traitement voulu

Des cellules dédiées au débogage

Risque de ne pas exécuter toutes les cellules et dans l'ordre

Nécessité de faire un restart du kernel et RunAll à chaque modification

Bonnes pratiques pour le développement sous notebook

« *One notebook, one focus* »

Développer dans un IDE, importer des librairies dans le notebook

Limiter le nombre de cellules (entre 4 et 10)

Chaque cellule devrait avoir un but unique.

Les fondamentaux du développement :

- Garder le code propre
- Ne commenter que pour expliquer la logique du développement
- Utiliser des fonctions
- Appliquer une stratégie de tests
- Faire des commits fréquents et un faible nombre de modifications

Typical structure of the ipynb

1. Imports

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

2. Get Data

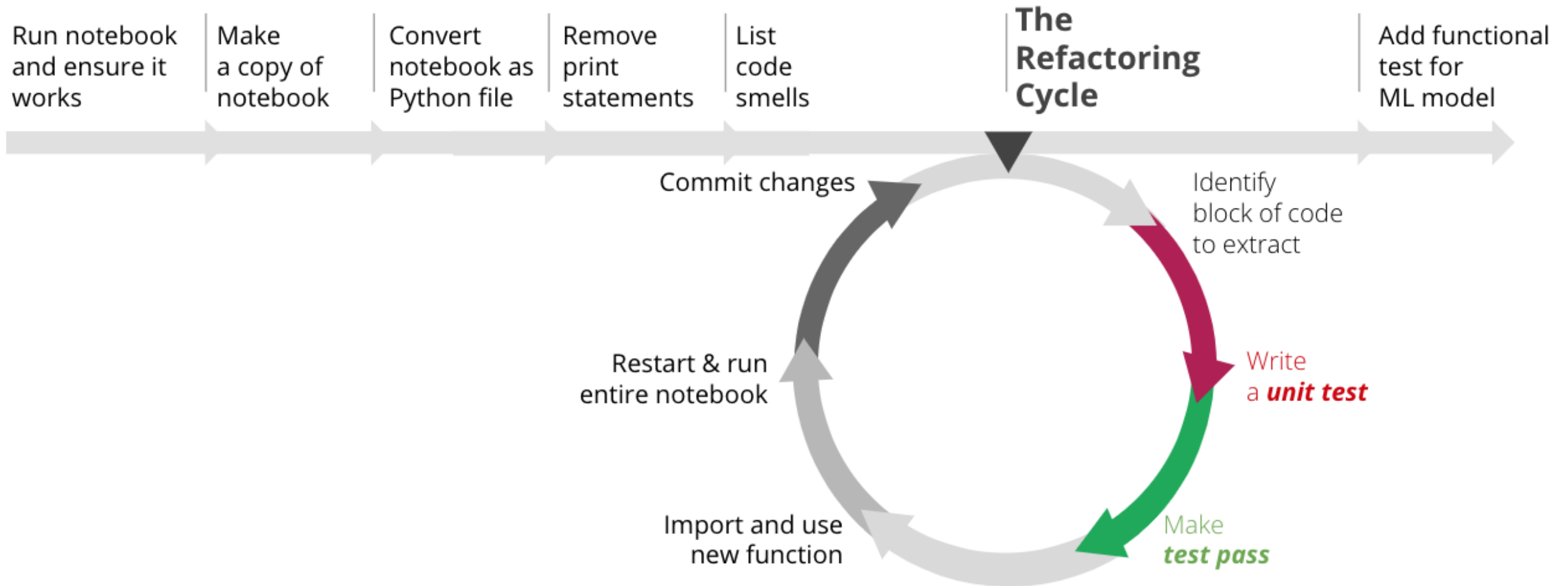
3. Transform Data

4. Modelling

5. Visualisation

6. Making sense of the data

Préparer un notebook pour la mise en production



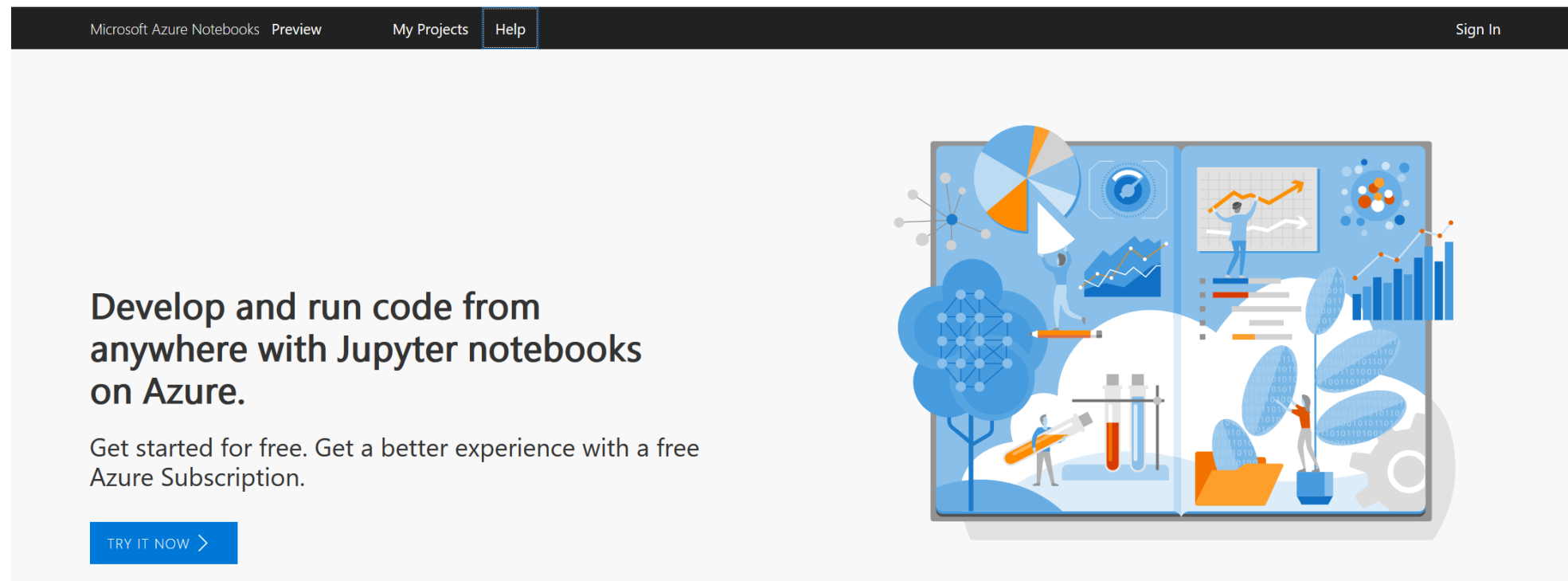
Azure Notebooks



Objectif : travailler dans un notebook Jupyter gratuitement et sans installation

En utilisant :

<https://notebooks.azure.com/> (Preview)





Nouveau projet / upload GitHub repo

A vous de jouer : upload du notebook ci-dessous

<https://github.com/methodidacte/aznycitibike/blob/master/New%20York%20citibike%20prep%20%26%20model.ipynb>



Home > My Projects

My Projects

Run Download Delete

Search Projects



Terminal

+ New Project

Upload GitHub Repo

✓ 👁 Name ▼

Status

Stars

Clones

Modified On

Created On

🌐 Case Studies-Mayed

Stopped

0

0

Sep 13, 2018

Sep 13, 2018



Travailler au sein d'un nouveau projet

Créer des éléments

Charger des éléments (From URL, From Computer)

Définir la ressource de calcul et la démarrer

Project settings (public)

Environnement setup ?



- Notebook
- Folder
- Blank File
- Markdown

Create New Notebook

Finish creating a new notebook by filling in the name and selecting the language.

Notebook Name

nycitibike

Select Language

- ☐ Python 2.7
- ☐ Python 3.5
- ☒ Python 3.6
- ☐ R
- ☐ F#

Status: **Stopped**

Run on Free Co...

search...

Free Compute

Azure Notebooks Compute

Direct Compute

Specify by an Azure IP Address

Microsoft Azure Notebooks Preview

My Projects

Help

Paul PETON - AI MVP



My Projects > Sécurité routière

Sécurité routière

Clone

0

Star

0

Status: **Stopped**

Project Settings

Download Project

Share

Run on Free Co...



Search files, notebooks...



Show hidden items



✓	Name	File Type	Modified On	Created On
	ANOVA décès.ipynb	Notebook	Sep 24, 2019	
	caracteristiques	Folder		

Les notebooks Azure proposent une capacité de traitement gratuite et partagée.

Il est possible d'associer sa propre capacité de traitement au travers d'une VM provisionnée sur sa souscription Azure.

⇒ Choisir une Data Science VM Linux Ubuntu

Vérifier la connexion au Jupyter Lab

⇒ <https://your-vm-ip:8000/user/your-username/lab>

On associe alors la VM en fournissant les informations suivantes :

- Nom de la machine
- IP publique & port 8000
- User name & password

Run on dsvmlinux

Any Jupyter service running in Azure can be used as a compute target for Azure Notebooks. Please configure the connection below.

① You need to validate the credentials for Azure Notebooks to connect to dsvmlinux.

Name

dsvmlinux

Enter an IP

13.74

Port

8000

User Name

ppeton

Password (will not be saved)

●●●●●●●●

Validate

Your DSVM configuration has been validated and cached for 8 hours. Click Run to continue connecting to 'dsvmlinux'.

Run

Cancel

Choisir la bonne solution selon l'échelle

Solution	Environnement	Licence	Passage à l'échelle	Coût
Python (Anaconda)	Local	Libre / Anaconda		0 + coût du laptop
Azure Notebooks	Cloud Azure	Microsoft	Direct compute	0 / Coût de la VM

Data Science Virtual Machine



Azure Data Science Virtual Machine

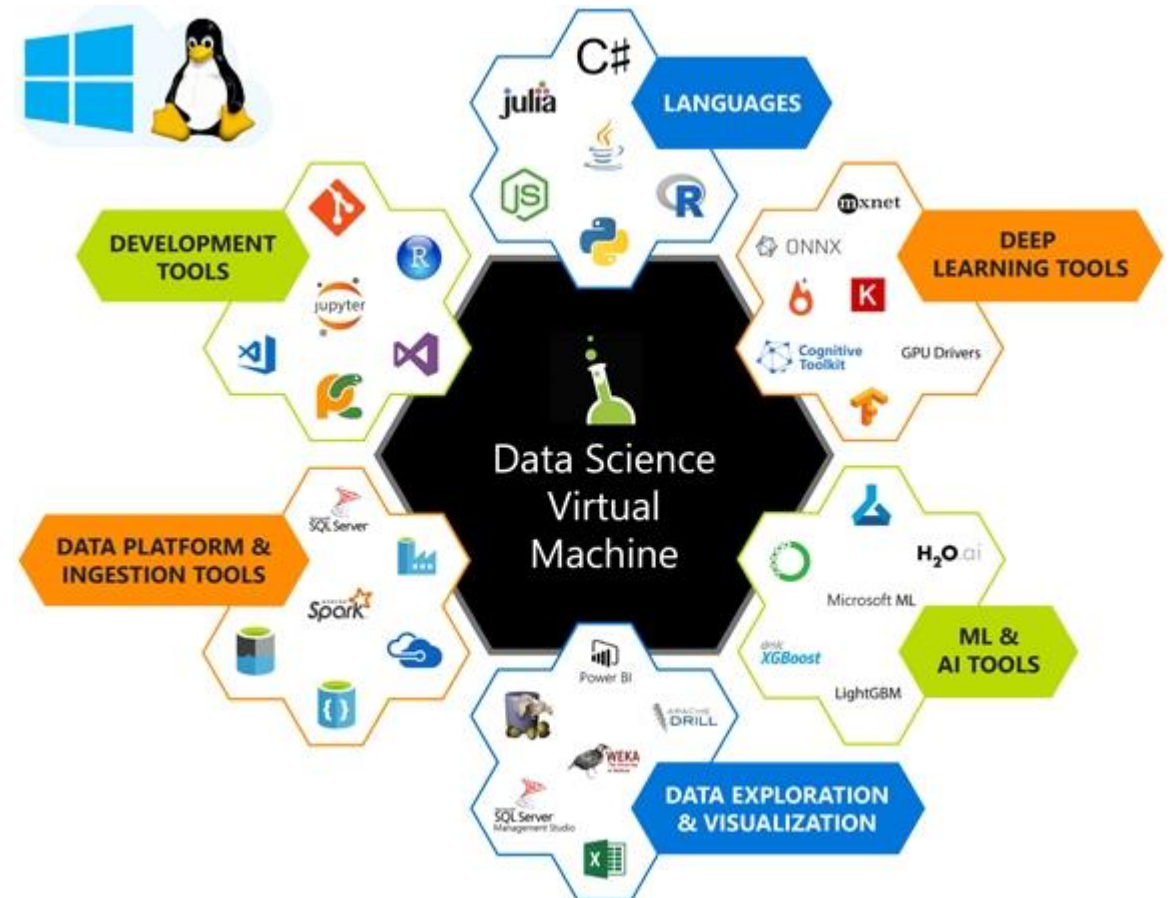
Machine virtuelle avec des outils déjà installés

- SQL Server
- R Server
- R Studio
- Anaconda
- Packages de Data Science
- Modèles pré-entraînés
- Etc.

OS Windows Server ou Linux

Retired : Azure Batch AI (version pour la production)

Deprecated : Deep Learning VM (avec GPU)



Choisir une DSVM sous Azure

Créer la ressource sur Azure

Choisir le système d'exploitation :

- Linux Ubuntu
- Linux CentOS
- Windows 2016

Choisir la série de la VM :

- Ls-Series (Storage optimized virtual machines)
- N-Series (GPU enabled virtual machines)



Data Science Virtual Machine for Linux (Ubuntu)

Microsoft

Virtual machine image with deep learning frameworks and tools for machine learning and data science.



Geo AI Data Science VM with ArcGIS

Microsoft

A geo-spatial analytics and AI extension to the Microsoft Data Science Virtual Machine



Free trial

Data Science Virtual Machine for Linux (CentOS)

Microsoft

Virtual machine with tools for data science and machine learning



Intel Optimized Data Science VM for Linux

Intel Software

A pre-configured Data Science Virtual Machine with CPU-optimized TensorFlow, MXNet and



Data Science Virtual Machine - Windows 2016

Microsoft

Development and modeling tools for AI, data science and analytics






Utiliser les notebooks Jupyter sur une DSVM Linux


Démarrer la machine virtuelle


(Ajouter éventuellement d'autres utilisateurs sur la machine)

Se connecter à <https://your-vm-ip:8000/> pour accéder à Jupyter Hub










 Logout Control Panel

Files Running Clusters Conda

Select items to perform actions on them. Upload New ▾ 

☐ 0 ▾  /

Name ▾ Last Modified File size

<input type="checkbox"/>  AzureML	il y a 5 mois	
<input type="checkbox"/>  BatchAI	il y a 10 mois	
<input type="checkbox"/>  catboost	il y a 10 mois	
<input type="checkbox"/>  dataset	il y a 14 jours	
<input type="checkbox"/>  deep_water	il y a 10 mois	
<input type="checkbox"/>  h2o	il y a 10 mois	
<input type="checkbox"/>  julia	il y a 10 mois	
<input type="checkbox"/>  MMLSpark	il y a 10 mois	
<input type="checkbox"/>  nycb	il y a 14 jours	
<input type="checkbox"/>  SparkML	il y a 10 mois	
<input type="checkbox"/>  DocumentDBSample.ipynb	il y a 10 mois	7.72 kB
<input type="checkbox"/>  Introduction to Azure ML R notebooks.ipynb	il y a 10 mois	32.7 kB
<input type="checkbox"/>  Introduction to Microsoft R Operationalization.ipynb	il y a 10 mois	28.1 kB

Utiliser Jupyter Lab par défaut

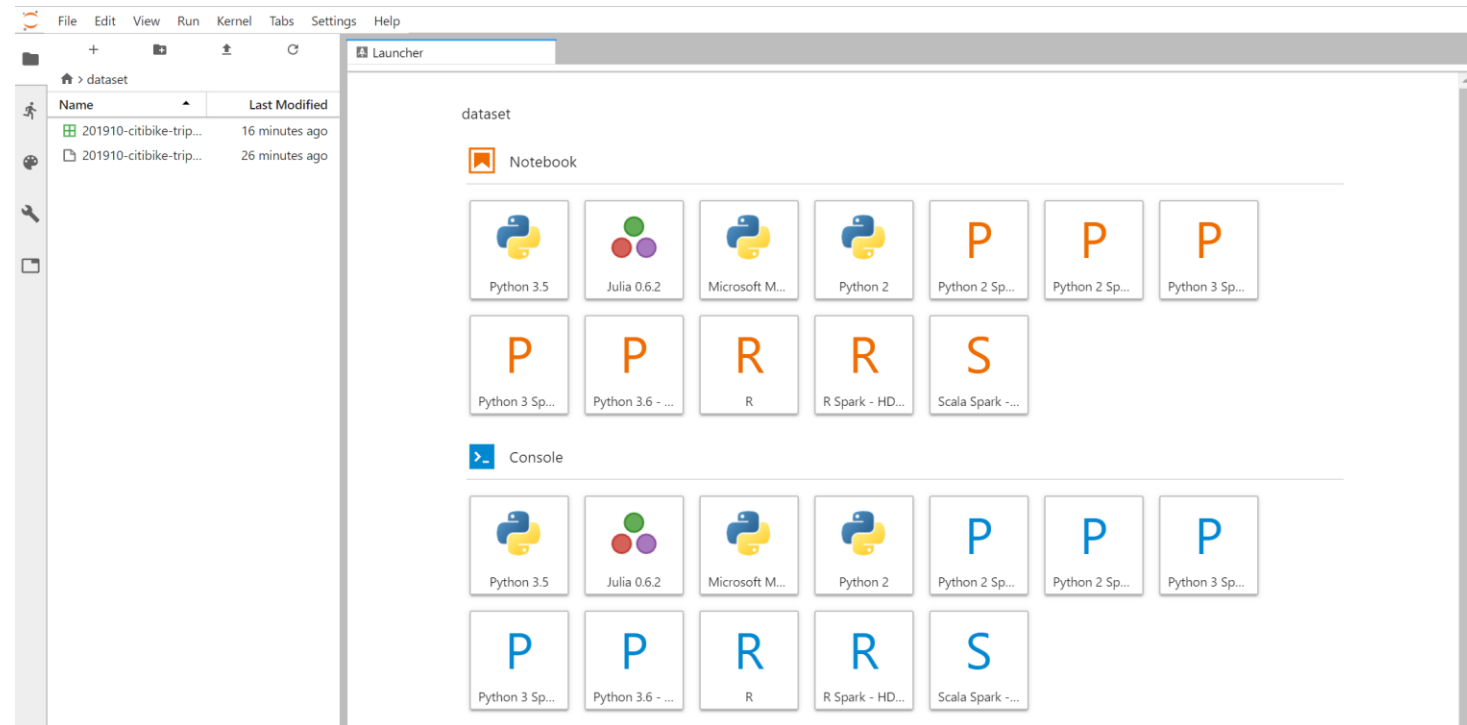
Se connecter <https://your-vm-ip:8000/user/your-username/lab> pour accéder à Jupyter Lab

Pour accéder par défaut à Jupyter Lab, modifier le fichier de configuration :

```
sudo vi /etc/jupyterhub/jupyterhub_config.py
```

Insérer la ligne suivante :

```
c.Spawner.default_url = '/lab'
```



Exploiter le contexte Spark local


Une installation locale de l'environnement Spark est déjà configurée sur la DSVM.

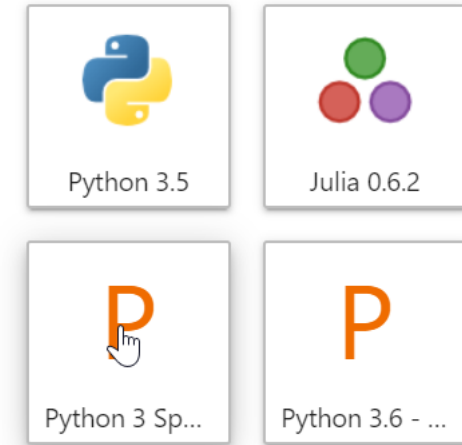
Il est donc possible de choisir le kernel « Python 3 Spark – local » pour exécuter un notebook.

Les commandes pyspark sont alors disponibles.

Il s'agit bien sûr de l'exécution sur un « cluster mono-nœud » mais cela permet de valider la syntaxe sans utiliser de ressource de cluster supplémentaire facturée.

dataset

 Notebook



Select Kernel

Select kernel for: "nycb_prep_sparklocal.ipynb"

Python 3 Spark - local



CANCEL

SELECT

Choisir la bonne solution selon l'échelle

Solution	Environnement	Licence	Passage à l'échelle	Coût
Python (Anaconda)	Local	Libre / Anaconda		0 + coût du laptop
Azure Notebooks	Cloud Azure	Microsoft	Direct compute	0 / Coût de la VM
DSVM + Spark context	Cloud Azure	Libre / Ubuntu / Microsoft	Vertical (perf de la VM)	Coût de la VM

Azure Databricks



D'Apache Spark à Azure Databricks

Suite à l'essor d'Hadoop (HDFS + MapReduce), les besoins de performances poussent à travailler « *in-memory* » plutôt qu'avec des I/O coûteux entre les nœuds du cluster.

Plusieurs chercheurs de l'université de Berkeley lancent en 2009 le projet Spark
open-source distributed computing framework built atop [Scala](#)

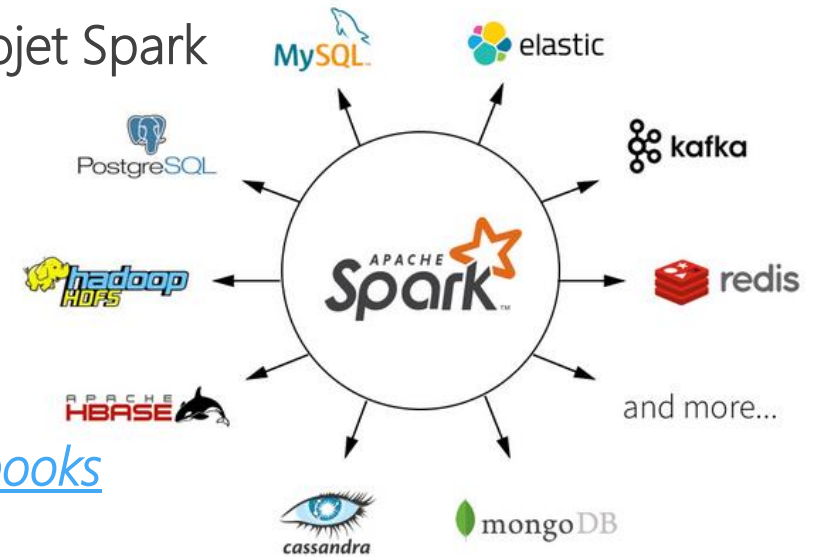
... puis fondent la société Databricks en 2013 :

*Databricks develops a web-based platform for working with Spark,
that provides automated cluster management and [IPython](#)-style [notebooks](#)*

Déploiement de Databricks en tant que service managé sur le cloud public :

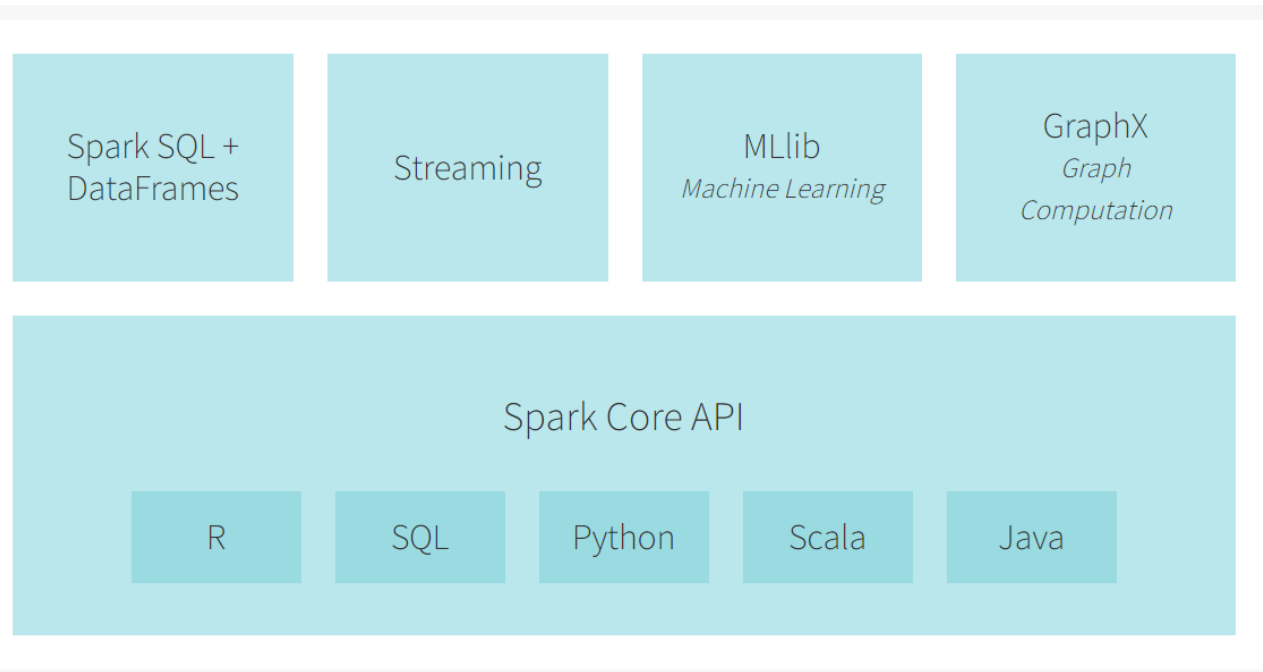
AWS en juin 2015

Azure en octobre 2017



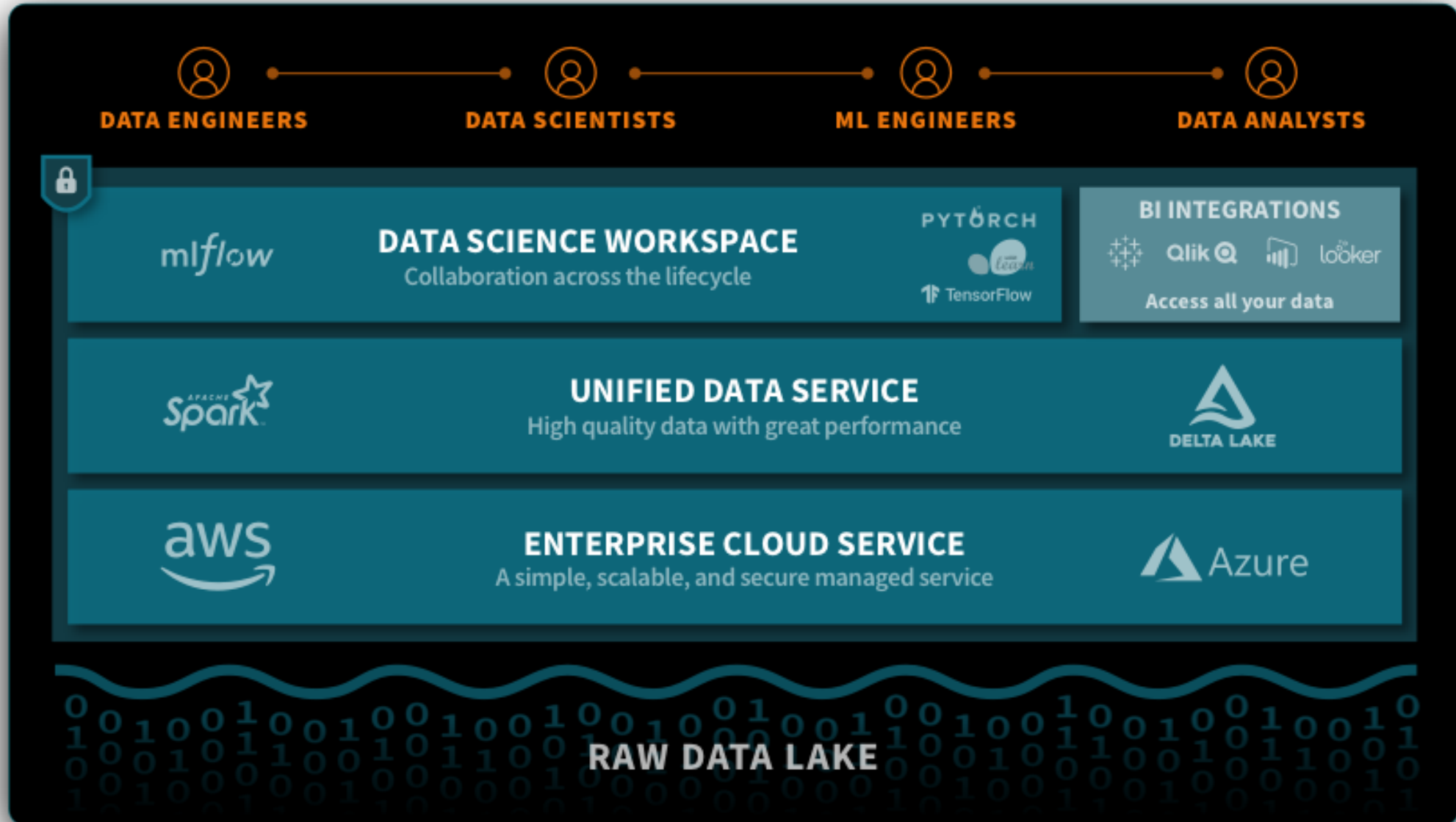


Apache Spark Ecosystem



- Polyglot:
 - APIs in Java, Scala, Python and R pour l'écriture du code Spark.
- Speed:
 - up to 100 times faster than Hadoop MapReduce for large-scale data processing.
- Multiple Formats:
 - Spark supports multiple data sources such as Parquet, JSON, Hive and Cassandra apart from the usual formats such as text files, CSV and RDBMS tables. The Data Source API provides a pluggable mechanism for accessing structured data through Spark SQL.
- Real Time Computation:
 - Spark's computation is real-time and has low latency because of its in-memory computation.
- Hadoop Integration:
 - Apache Spark provides smooth compatibility with Hadoop, using YARN for resource scheduling.
- Lazy Evaluation:
 - Apache Spark delays its evaluation till it is absolutely necessary. This is one of the key factors contributing to its speed. For transformations, Spark adds them to a DAG (Directed Acyclic Graph) of computation and only when the driver requests some data, does this DAG actually get executed.
- Machine Learning:
 - Spark's MLlib is the machine learning component which is handy when it comes to big data processing.
 - Use spark ml since Spark 3.0

Databricks : « Unified Data Analytics Platform »



Se connecter à Databricks community

Community : version gratuite limitée à un cluster mononoeud de 6Go de RAM

Single cluster limited to 6GB and no worker nodes

Basic notebook without collaboration

Limited to 3 max users

Public environment to share your work

Des fonctionnalités absentes (création de token, redémarrage du cluster, etc.)

S'inscrire sur le portail : <https://databricks.com/try-databricks>

Se connecter : <https://community.cloud.databricks.com/login.html>

Créer un cluster

Importer le notebook depuis l'URL

<https://github.com/methodidacte/aznycitibike/blob/master/New%20York%20citibike%20prep%20%26%20mode%20pyspark.ipynb>

Attacher le cluster au notebook

COMMUNITY EDITION

For students and educational institutions just getting started with
Apache Spark

- Single cluster limited to 6GB and no worker nodes
- Basic notebook without collaboration
- Limited to 3 max users
- Public environment to share your work

GET STARTED

Choisir la bonne solution selon l'échelle

Solution	Environnement	Licence	Passage à l'échelle	Coût
Python (Anaconda)	Local	Libre / Anaconda		0 + coût du laptop
Azure Notebooks	Cloud Azure	Microsoft	Direct compute	0 / Coût de la VM
DSVM + Spark context	Cloud Azure	Libre / Ubuntu / Microsoft	Vertical (perf de la VM)	Coût de la VM
Databricks Community	Cloud	Databricks	Non	0



Les apports de Azure Databricks

(versus Databricks Community)

Multi-utilisateurs (ceux présents dans Azure AD et autorisés dans les Access controls)

Choix du mode de cluster (standard ou high concurrency)

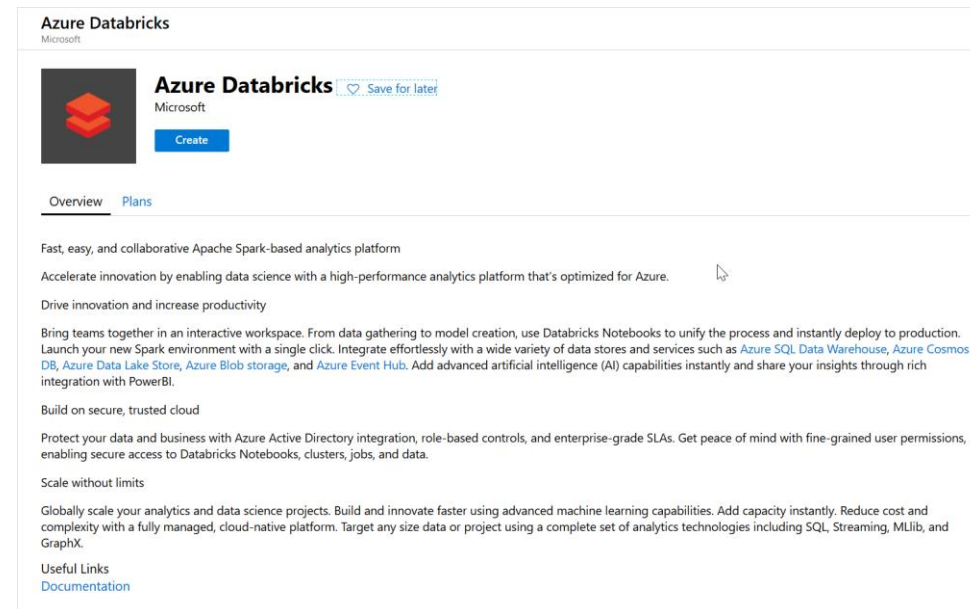
Gestion du cluster (interactif ou automatique)

Planification par jobs

Pérennisation des données sur le cluster au moyen d'un metastore

Accès à un gestionnaire de version

Deux licences : standard ou premium



Lier Azure Databricks à un gestionnaire de version

Git integration

Databricks supports notebook version control integration with GitHub, Bitbucket Cloud, or Azure DevOps Services. To connect to git repositories and make commits, we need a personal access token (for GitHub) or an app password (for Bitbucket Cloud). Azure DevOps Services requires no extra authentication.

Generating tokens

To generate a GitHub personal access token, follow the [GitHub documentation](#). When using GitHub, the token must have the “repo” permission.

To generate a Bitbucket Cloud app password, follow the [Bitbucket Cloud documentation](#). When using Bitbucket Cloud, the app password must have “read” and “write” permission under repository.

For more information on Git integration, see the Databricks documentation on [GitHub integration](#), [Bitbucket Cloud integration](#), or [Azure DevOps Services integration](#).

Git provider

Change settings

GitHub

Git Preferences

Status ☐ Link ☒ Unlink

Link

Link to file in repository (e.g. [https://\[GITPROVIDER\].com/repo](#))

Branch

master

Path in Git
Repo

notebooks/Shared/sandbox/Datalake Mounts.py

Tip: You can also import/export multiple notebooks or an entire folder through [Workspace API](#) to your computer and check-in to your favorite version control system.

Close

Save



Interactive or automated cluster

Un cluster interactif est défini manuellement ou au moyen d'un fichier JSON de configuration.

Les VMs provisionnées s'éteignent lorsque le cluster n'est plus utilisé mais restent présentes dans les ressources Azure.

Conservation des points de montages
Conservation des tables matérialisées



La facturation est plus élevée pour ce type de cluster (voir tarifs « Analyse de données »).

Clusters

Clusters Pools

+ Create Cluster

▼ Interactive Clusters

Name
  nycbcluster

▼ Automated Clusters

--

Un cluster automatique correspond à la création de nouvelles VMs lors du lancement d'un job planifié.

A l'arrêt du cluster, les VMs ne sont pas conservées dans les ressources Azure.

La facturation est moins élevée pour ce type de cluster (voir tarifs « Engineering données »).

Le mode « Engineering Light » correspond à l'utilisation d'un cluster Spark dans sa version Open Source.



ANALYSE DE DONNÉES		ENGINEERING DONNÉES	ENGINEERING DONNÉES LIGHT
Charges de travail interactives pour analyser les données de façon collaborative avec des bloc-notes		Charges de travail automatisées qui exécutent des travaux robustes et rapides via des API ou une interface utilisateur	Charges de travail automatisées qui exécutent des travaux robustes via des API ou l'interface utilisateur
CHARGE DE TRAVAIL	TARIF DES UNITÉS DBU - NIVEAU STANDARD	TARIF DES UNITÉS DBU - NIVEAU PREMIUM	
Analyse de données	0,34 €/DBU-heure	0,464 €/DBU-heure	
Engineering données	0,13 €/DBU-heure	0,253 €/DBU-heure	
Engineering données Light	0,06 €/DBU-heure	0,186 €/DBU-heure	

*En plus des machines virtuelles, Azure Databricks facturera également les disques managés, le stockage blob et les adresses IP publiques.

Fonctionnalités spécifiques à la licence Premium

FONCTIONNALITÉ	ANALYSE DE DONNÉES	ENGINEERING DONNÉES	ENGINEERING DONNÉES LIGHT
	Charges de travail interactives pour analyser les données de façon collaborative avec des bloc-notes	Charges de travail automatisées qui exécutent des travaux robustes et rapides via des API ou une interface utilisateur	Charges de travail automatisées qui exécutent des travaux robustes via des API ou l'interface utilisateur
	Inclut des fonctionnalités Standard	Inclut des fonctionnalités Standard	Inclut des fonctionnalités Standard
Contrôle d'accès en fonction du rôle pour les notebooks, les clusters, les travaux et les tables	✓	✓	✓
Authentification du point de terminaison JDBC/ODBC	✓	✓	✓
Journaux d'audit (en préversion)	✓	✓	✓

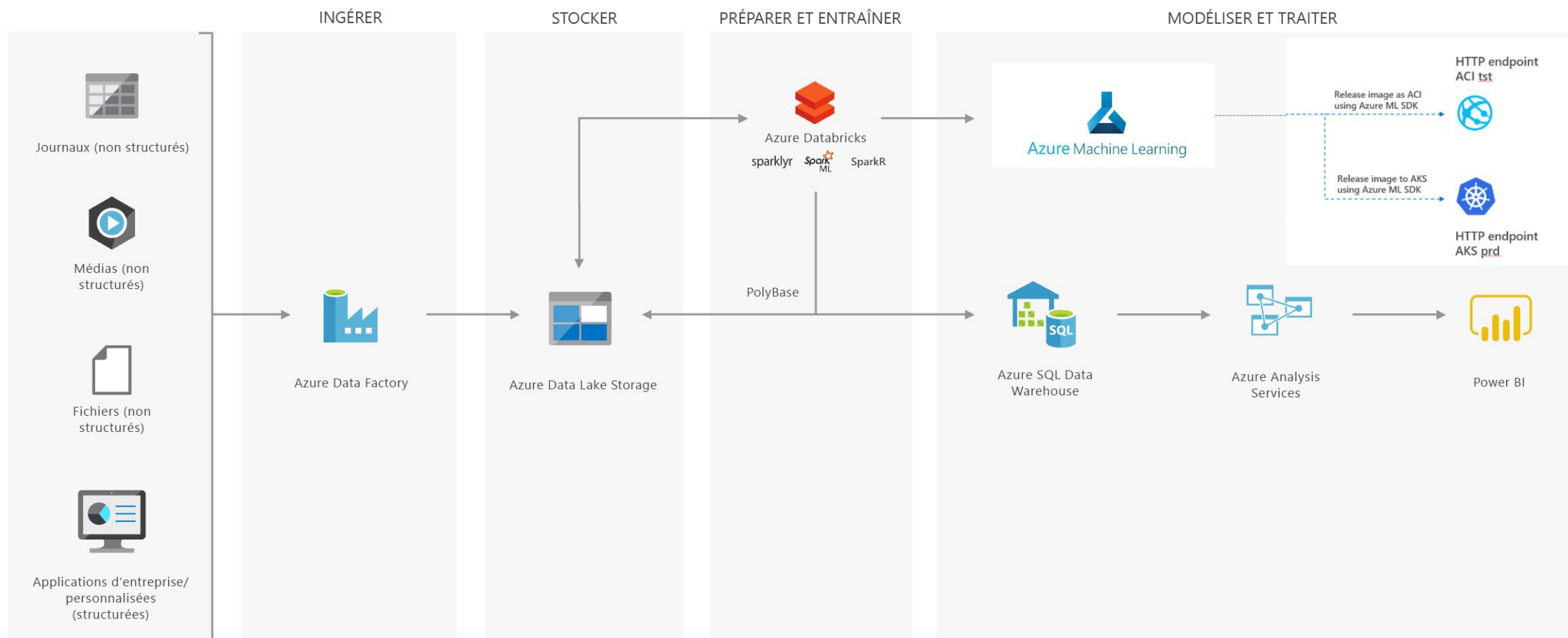
Choisir la bonne solution selon l'échelle

Solution	Environnement	Licence	Passage à l'échelle	Coût
Python (Anaconda)	Local	Libre / Anaconda		0 + coût du laptop
Azure Notebooks	Cloud Azure	Microsoft	Direct compute	0 / Coût de la VM
DSVM + Spark context	Cloud Azure	Libre / Ubuntu / Microsoft	Vertical (perf de la VM)	Coût de la VM
Databricks Community	Cloud	Databricks	Non	0
Azure Databricks	Cloud Azure	Azure / Databricks	Horizontal par le nb et type de VM du cluster	Mesuré en DBU (voir calculatrice)



Architecture Advanced Analytics for Data Science (Viz & ML)

Analytique avancée pour Big Data



De Python à pyspark





Pyspark est une **API** permettant d'utiliser la puissance de Spark par le biais du langage Python.

Les **dataframes Pyspark** sont une collection distribuée de données structurées ou semi-structurées, stockés sous forme de lignes et de colonnes.

PySpark SQL is a higher-level abstraction module over the PySpark Core. It is majorly used for processing structured and semi-structured datasets. It also provides an optimized API that can read the data from the various data source containing different files formats.

Il est possible de manipuler les données avec des instructions apparentées au langage SQL traditionnelle.

EXEMPLE

Il n'y a alors pas de différence de performance dans un contexte de notebook Python ou Scala.



Comparaison Python (pandas) versus pyspark - IMPORT

```
# import de fichier plat
df = read_csv('my_file.csv')
```

```
# structure du DataFrame
df.info()
```

```
# nb de lignes du DataFrame
df.shape[0]
```

```
# aperçu des premières lignes
df.head()
```

```
# renommer une colonne
df = df.rename({'old': 'new'})
```

```
# import de fichier plat
df = spark.read.format('csv')
    \.option('inferSchema', 'true')
    \.load('my_file.csv')
```

```
# structure du DataFrame
df.printSchema()
```

```
# nb de lignes du DataFrame
df.count()
```

```
# aperçu des premières lignes
df.show()
```

```
display(df) # DATABRICKS
```

```
# renommer une colonne
df.withColumnRenamed('old', 'new')
```




Comparaison Python (pandas) versus pyspark - PREPARATION

nb de valeurs manquantes

```
df.isnull().sum()
```

supprimer les valeurs manquantes

```
df.drop_na()
```

remplacer les valeurs manquantes

```
df.fillna('N/A', inplace = True)
```

filtrer certains valeurs

```
df[df['column'] = 'A']
```

nb de valeurs manquantes

```
df.isnull().sum()
```

supprimer les valeurs manquantes

```
df.na.drop()
```

remplacer les valeurs manquantes

```
df.na.fill('N/A')
```

filtrer certains valeurs

```
df.filter(col('column') = 'A')
```



pandas DataFrame API on top of Apache Spark

La simplicité d'écriture de pandas et la puissance de Databricks

<https://koalas.readthedocs.io/en/latest/>

```
import databricks.koalas as ks
```

```
kdf = ks.from_pandas(pdf)
```

```
type(kdf)
```

```
sdf = kdf.to_spark()
```

```
sdf.to_koalas()
```

```
Kdf_from_csv = ks.read_csv("dbfs:/koalas/data.csv")
```



Koalas

Data Scientist vs Automated ML : who's best ?

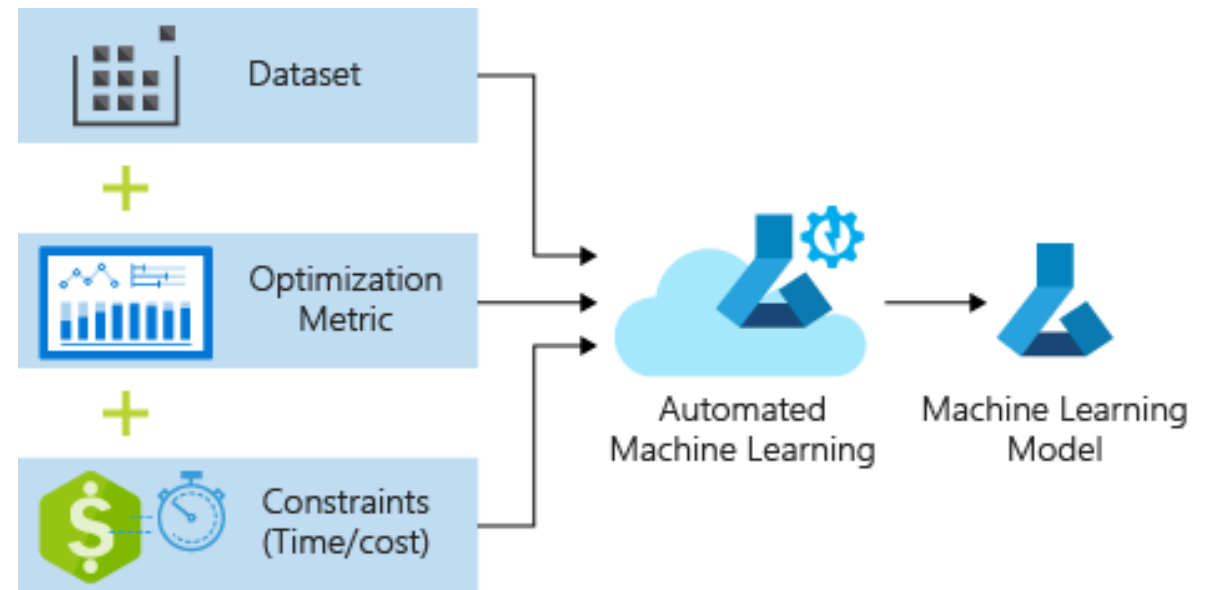
« *There is no free lunch* » = Il faudra tester toutes les méthodes algorithmiques pour trouver la meilleure.

C'est une approche de « force brute » qui ne nécessite pas de compétence particulière.

L'automated ML réalise, à la place du Data Scientist, la recherche du meilleur modèle.

C'est une fonctionnalité disponible :

- Au travers d'un SDK (il faut coder)
- Sur le portail Azure Machine Learning Studio





Configure run

Configure the experiment. Select from existing experiments or define a new name, select the target column and the training compute to use.

[Learn more on how to configure the experiment](#) 

Dataset

Diabetes [\(View dataset\)](#)

Experiment name *

diabetes-autoML



Target column *

Y



Select training compute target *

myfirstcompute



 [Create a new compute](#)  [Refresh compute](#)

Create a new automated machine learning run

- ✓ Select dataset
- ✓ Configure run
- Task type and settings**

Select task type

Select the machine learning task type for the experiment. Additional settings are available to fine tune the experiment if needed.



Classification

To predict one of several categories in the target column. yes/no, blue, red, green.



Enable deep learning (preview) [i](#)



Regression

To predict continuous numeric values



Time series forecasting

To predict values based on time



[⚙ View additional configuration settings](#) [🔍 View featurization settings](#)

Back

Finish

Cancel

Run 1  Preparing

 Refresh  Cancel

Details

Models

Data guardrails

Properties

Logs

Outputs

Run details

Task type
classification

Primary metric
Accuracy

Run status
Preparing

Experiment name
diabetes-autoML

Run ID
AutoML_4047d78d-3111-47fe-a563-f7beef530441

Run 1 ✔ Completed

[Switch to old experience](#) ?

[Refresh](#) [Cancel](#)

☐ Auto refresh every 30 seconds

Details **Models** Data guardrails Properties Logs Outputs

[Search to filter items...](#)

Algorithm name	R2 score ↓	Created	Duration	Status	Model
VotingEnsemble	0.5196837276672349	September 21, 2019 2:46 PM	00:01:17	Completed	Download
StandardScalerWrapper, LightGBM	0.4852919777734466	September 21, 2019 1:54 PM	00:00:58	Completed	Download
StandardScalerWrapper, LightGBM	0.4746368450470218	September 21, 2019 2:21 PM	00:01:01	Completed	Download
MinMaxScaler, LightGBM	0.47353390887812713	September 21, 2019 2:02 PM	00:00:59	Completed	Download
RobustScaler, LightGBM	0.47016213061020473	September 21, 2019 1:52 PM	00:01:01	Completed	Download
StandardScalerWrapper, GradientBoosting	0.44737229156006003	September 21, 2019 2:28 PM	00:00:59	Completed	Download
MinMaxScaler, LightGBM	0.43807341418747736	September 21, 2019 2:20 PM	00:01:00	Completed	Download
MaxAbsScaler, LightGBM	0.42676793245406075	September 21, 2019 1:50 PM	00:00:56	Completed	Download
StandardScalerWrapper, GradientBoosting	0.41510107470261814	September 21, 2019 2:42 PM	00:00:59	Completed	Download
RobustScaler, LightGBM	0.4130161122924981	September 21, 2019 2:40 PM	00:01:07	Completed	Download

[< Prev](#) [Next >](#)



— Surtout ! ne parlons pas de l'affaire Dreyfus !



La fin des Data Scientists ?

Par Caran d'Ache — Cette image provient de la Bibliothèque en ligne Gallica sous l'identifiant ARK bpt6k2842896/f3.item, Domaine public, <https://commons.wikimedia.org/w/index.php?curid=438435>



SIÈGE SOCIAL

52 Avenue André Morizet
92100 Boulogne-Billancourt

Paris – Bordeaux – Nantes – Toulouse



Siège Social

52, avenue André Morizet - 92100 Boulogne-Billancourt
contact@azeo.com | [01.83.62.65.54](tel:01.83.62.65.54) | www.azeo.com

AZEO

talents & technology



Paul Péton
Lead Data Scientist
paul.peton@azeo.com
06 11 10 22 01



Benjamin Benito
Consultant Data & AI
benjamin.benito@azeo.com

Gold
Microsoft Partner

[Contacter un conseiller](#)