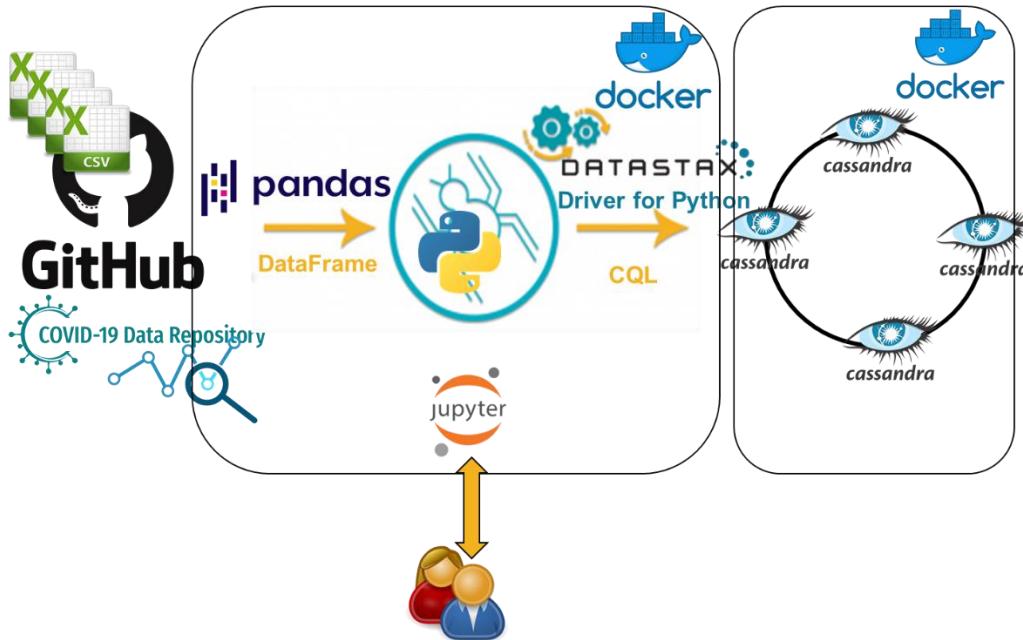


BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

Atelier 2 : Mise en place de l'environnement de Développement Jupyter & Cassandra



1. Objectifs

Après avoir terminé cet atelier, vous serez en mesure de :

- Lancer Cassandra avec Docker
- Lancer un CQLShell via Docker
- Créer un notebook avec Jupyter contenant le processus d'ingestion de données vers Cassandra avec Python (voir document PDF)

2. Installation de Cassandra avec Docker

- Créez un répertoire «/home/ubuntu/cassandra_abdata/» pour l'utiliser comme dépôt pour vos conteneurs

```
sudo mkdir /home/ubuntu/cassandra_abdata/
```

- Lancez un shell et taper la commande suivante

```
sudo docker run --rm -p 7000:7000 -p 7001:7001 -p 7199:7199 -p 9042:9042 -p 9160:9160 -v
"$PWD":/home/ubuntu/cassandra_abdata/ -d cassandra:latest
```

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

- Vérifier que l'instance est bien lancée

```
sudo docker ps
```

```
ubuntu@cloudera:~$ sudo docker run --name abdata-cassandra -p 7000:7000 -p 7001:7001 -p 7199:7199 -p 9042:9042 -p 9160:9160 -d
assandra:latest
6b623dc4eb44f3b5a594b72b64915bc1e2ba86dfcf310241262ed6aa4d4517f8
ubuntu@cloudera:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
6b623dc4eb44        cassandra:latest   "docker-entrypoint.s..."   5 hours ago       Up 5 hours         abdata-cassandra
7001/tcp, 0.0.0.0:7199->7199/tcp, 0.0.0.0:9042->9042/tcp, 0.0.0.0:9160->9160/tcp
ubuntu@cloudera:~$
```

- Lancer le Shell CQLSH pour accéder à l'instance

```
sudo docker exec -it 6b623dc4eb44 /bin/bash
```

```
ubuntu@cloudera:~/abdata_cassandra$ sudo docker exec -it 6b623dc4eb44 /bin/bash
root@6b623dc4eb44:/# cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.9 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh>
```

3. Préparation du Schéma

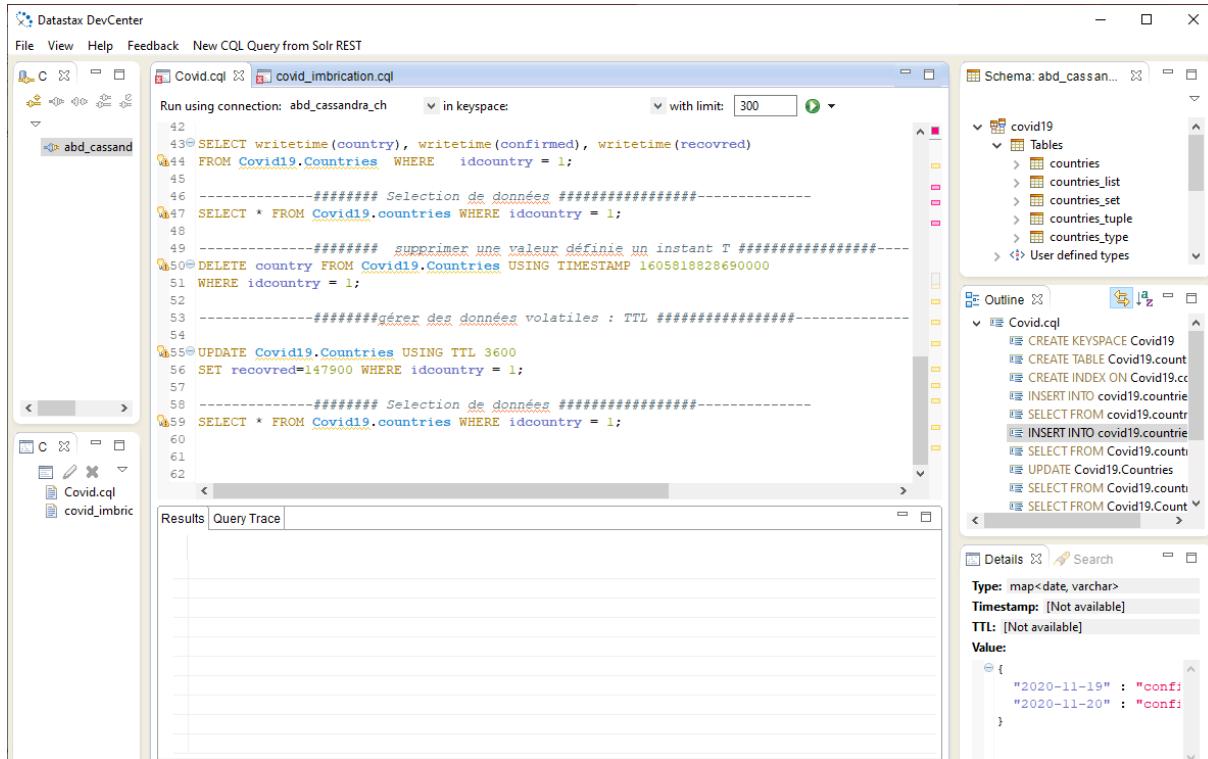
- Crédation du Schéma du Keyspace Covid19App

Lancer DevCenter et connecter au cluster Cassandra

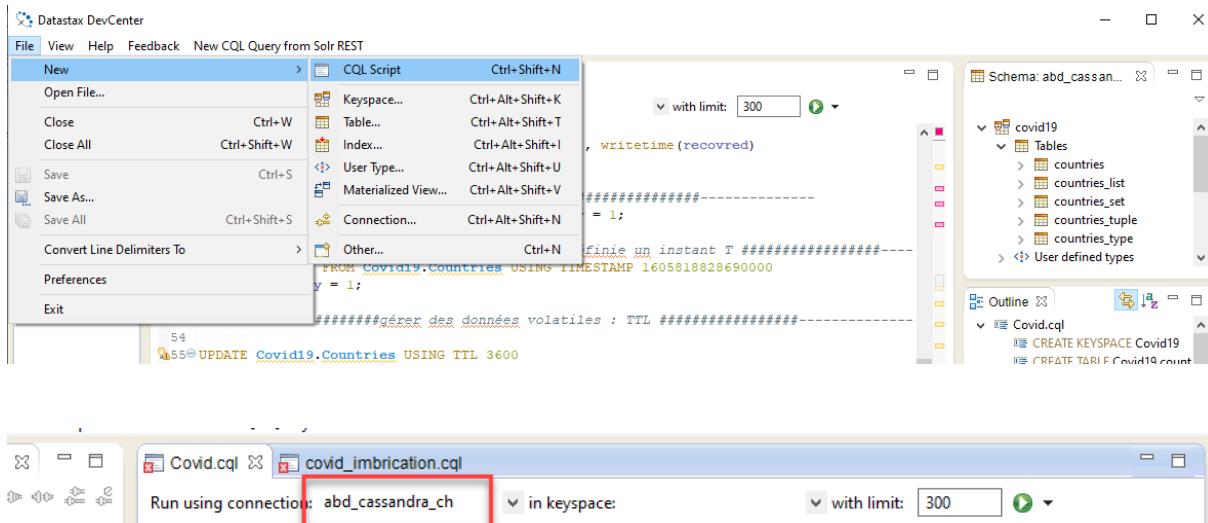
```
./DevCenter
```

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



Créer un nouveau Fichier CovidApp.cql et choisir une connexion



Ce Script permettant de créer un Keyspace avec une table countries utilisant une imbrication MAP et UDT – TYPE CovidType

```
-----#####Creation d'un keyspace#####
CREATE KEYSPACE IF NOT EXISTS Covid19App
WITH REPLICATION={ 'class':'SimpleStrategy', 'replication_factor':3 };

use Covid19App;
```

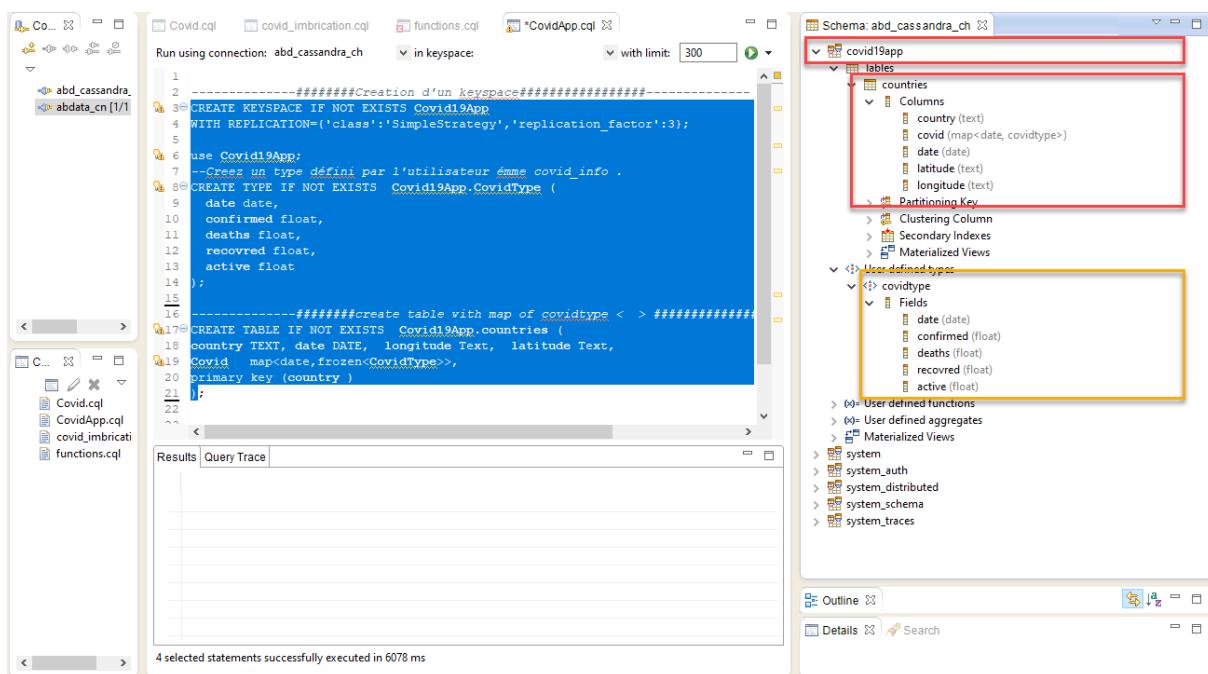
BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

```
--Creez un type défini par l'utilisateur émme covid_info .
CREATE TYPE IF NOT EXISTS Covid19App.CovidType (
    date date,
    confirmed float,
    deaths float,
    recovered float,
    active float
);

-----#####create table with map of covidtype < >
#####
CREATE TABLE IF NOT EXISTS Covid19App.countries (
    country TEXT, date DATE, longitude Text, latitude Text,
    Covid map<date,frozen<CovidType>>,
    primary key (country )
);
```

Vérifier que la création est bien terminée avec succès



The screenshot shows the DataStax Studio interface with two main panes. On the left, the 'Code' pane displays CQL code for creating a keyspace, a user-defined type, and a table. The code includes comments explaining the steps: 'Creation d'un keyspace', 'Creez un type défini par l'utilisateur émme covid_info .', and 'create table with map of covidtype'. The right pane, titled 'Schema: abd_cassandra_ch', shows the resulting schema. It highlights the 'countries' table and the 'CovidType' type definition. The 'countries' table is defined with columns for 'country', 'date', 'longitude', 'latitude', and a 'Covid' column of type 'map<date,frozen<CovidType>>'. The 'CovidType' type is defined with fields for 'date', 'confirmed', 'deaths', 'recovered', and 'active'. Both definitions are highlighted with red and yellow boxes respectively.

4. Un Editeur web pour le développement Python

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala2. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces calepins sont utilisés en science des données pour explorer et analyser des données.

4.1. Exécuter un conteneur Jupyter

L'utilisation de l'une des Jupyter Docker Stacks nécessite deux choix:

1. Quelle image Docker vous souhaitez utiliser
2. Comment vous souhaitez démarrer les conteneurs Docker à partir de cette image

Cette section fournit des détails sur le second.

4.2. Utilisation de la CLI Docker

Vous pouvez lancer un conteneur Docker local à partir des Jupyter Docker Stacks à l'aide de l' [interface de ligne de commande Docker](#) . Il existe de nombreuses façons de configurer les conteneurs à l'aide de l'interface de ligne de commande. Voici quelques modèles courants.

4.2.1. Lancement du conteneur jupyter

Cette commande extrait l' image `jupyter/scipy-notebook` balisée `2c80cf3537ca` de Docker Hub si elle n'est pas déjà présente sur l'hôte local. Il démarre ensuite un conteneur exécutant un serveur Jupyter Notebook et expose le serveur sur le port hôte 8888. Les journaux du serveur apparaissent dans le terminal et incluent une URL vers le serveur Notebook.

```
$ sudo docker run --rm -p 8888:8888 -e JUPYTER_ENABLE_LAB=yes -v  
"$PWD":/home/ubuntu/cassandra_abdata/ jupyter/scipy-  
notebook:2c80cf3537ca
```

Le fait d'appuyer sur `Ctrl-C` arrête le serveur notebook mais laisse le conteneur intact sur le disque pour un redémarrage ultérieur ou une suppression permanente à l'aide de commandes telles que les suivantes :

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com

```
# list containers
docker ps -a
CONTAINER ID        IMAGE               COMMAND
CREATED              STATUS              PORTS
d67fe77f1a84        jupyter/base-notebook   "tini -- start-noteb..."  44
seconds ago         Exited (0) 39 seconds ago
cocky_mirzakhani

# start the stopped container
docker start -a d67fe77f1a84
Executing the command: jupyter notebook
[W 16:45:02.020 NotebookApp] WARNING: The notebook server is listening
on all IP addresses and not using encryption. This is not recommended.
...
.

# remove the stopped container
docker rm d67fe77f1a84
d67fe77f1a84
```

Lancer Jupyter en cliquant sur le lien dans le Shell

```
Digest: sha256:1457325a4df1803427042686b7b8c99261ec0ae75c6af1d4acbf2df9279c3668
Status: Downloaded newer image for jupyter/scipy-notebook:2c80cf3537ca
Executing the command: jupyter notebook
[I 20:18:59.153 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_secret
[W 20:18:59.635 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 20:18:59.676 NotebookApp] JupyterLab alpha preview extension loaded from /opt/conda/lib/python3.6/site-packages/jupyterlab
[I 20:18:59.676 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 20:18:59.682 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 20:18:59.683 NotebookApp] 0 active kernels
[I 20:18:59.683 NotebookApp] The Jupyter Notebook is running at:
[I 20:18:59.683 NotebookApp] http://[all ip addresses on your system]:8888/?token=aa9cbf81a8cdaadc789a01f09decb64cb1091c7626720f5
7
[I 20:18:59.683 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:18:59.683 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=aa9cbf81a8cdaadc789a01f09decb64cb1091c7626720f57
```

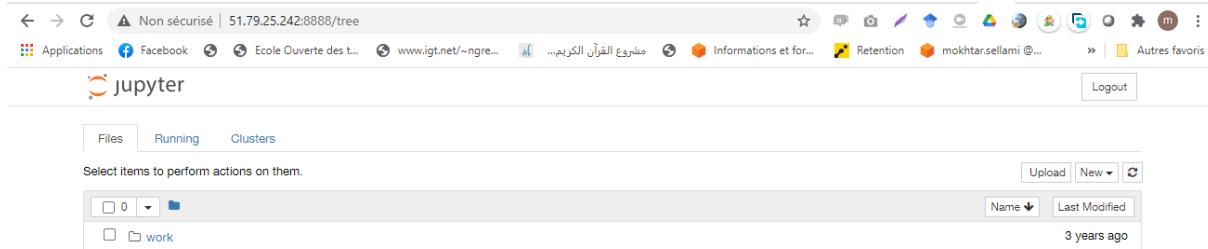
4.2.2. Manipulation de Jupyter notebook¹

Le notebook est constitué d'une succession de cellules comportant soit du texte en Markdown comme ici, soit du code comme dans la cellule suivante (Python pour nous):

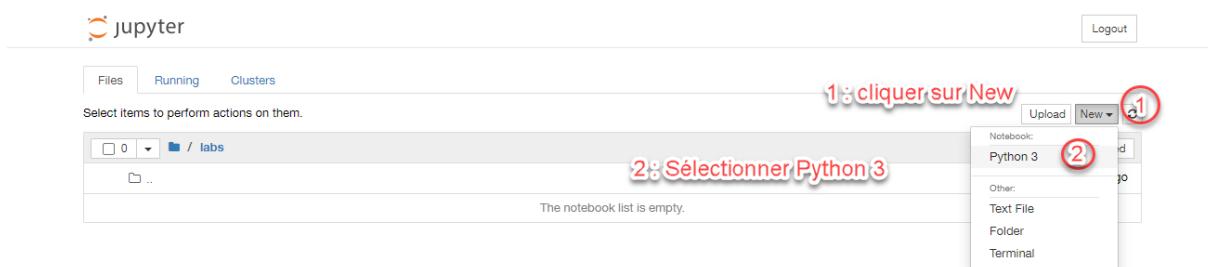
¹ <http://faccanoni.univ-tln.fr/user/enseignements/20182019/M62-CM1-M.pdf>
<https://jupyter-notebook.readthedocs.io/en/4.x/changelog.html>

BASE DE DONNEES NO SQL Cassandra

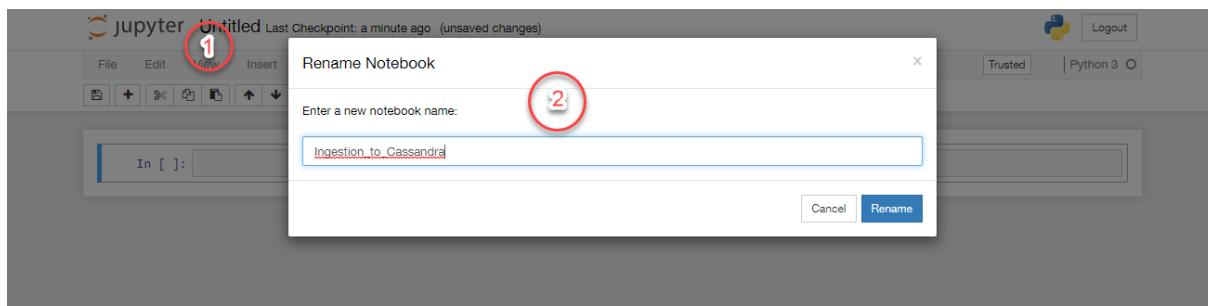
Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



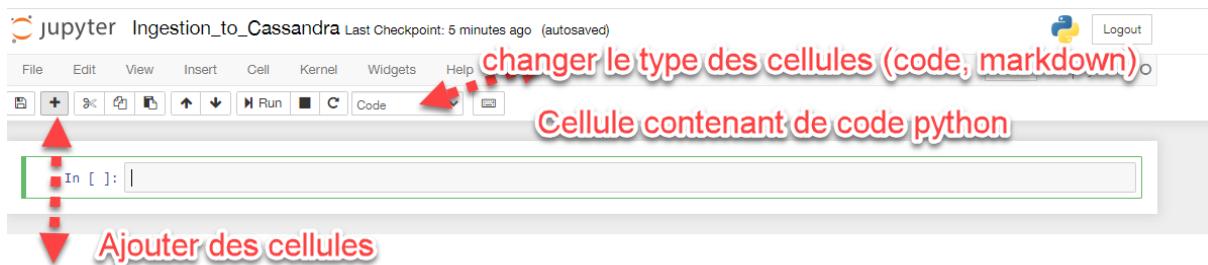
Créer un nouveau notebook



Renommer le « Ingestion_to_Cassandra »



Créer une cellule de code ou markdown



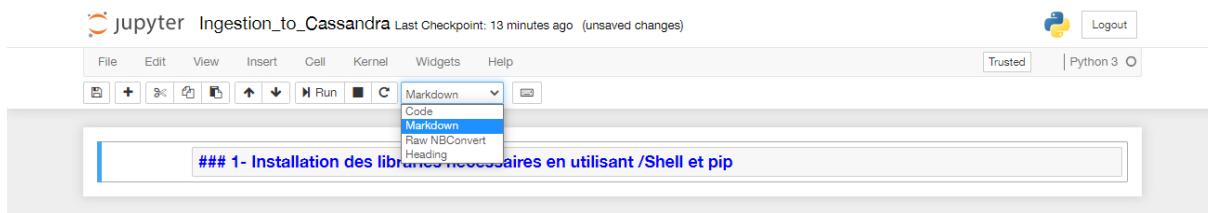
Mode édition : les cellules Markdown et sa syntaxe

Le texte de la cellule doit être rédigé en langage Markdown qui est un langage de balisage léger. La syntaxe markdown est facile à apprendre (le plus simple est d'ailleurs de regarder des exemples de documents).

Créer et formater une première cellule contenant du texte

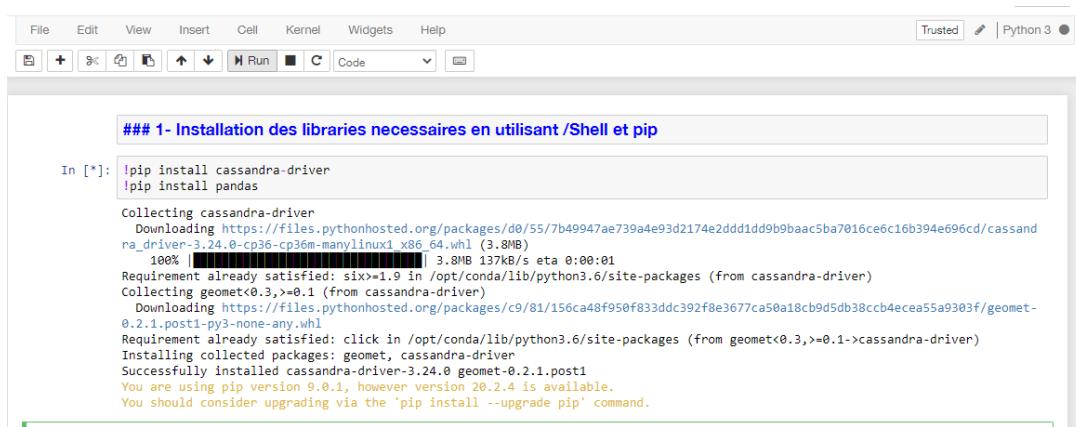
BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



1- Installation des librairies nécessaires en utilisant /Shell et pip

Créer une nouvelle cellule contenant des commandes Shell pour installer les librairies nécessaires (Cassandra driver et pandas) pour note notebook



```
## 1- Installation des librairies nécessaires en utilisant /Shell et pip

In [*]: !pip install cassandra-driver
!pip install pandas

Collecting cassandra-driver
  Downloading https://files.pythonhosted.org/packages/d0/55/7b49947ae739a4e93d2174e2ddd1dd9b9baac5ba7016ce6c16b394e696cd/cassand
ra_driver-3.24.0-cp36-cp36m-manylinux1_x86_64.whl (3.8MB)
     100% |██████████| 3.8MB 137kB/s eta 0:00:01
Requirement already satisfied: six>=1.9 in /opt/conda/lib/python3.6/site-packages (from cassandra-driver)
Collecting geomet<0.3,>=0.1 (from cassandra-driver)
  Downloading https://files.pythonhosted.org/packages/c9/81/156ca48f95bf833ddc392f8e3677ca58a18cb9d5db38ccb4ece55a9303f/geomet-
0.2.1.post1-py3-none-any.whl
Requirement already satisfied: click in /opt/conda/lib/python3.6/site-packages (from geomet<0.3,>=0.1->cassandra-driver)
Installing collected packages: geomet, cassandra-driver
Successfully installed cassandra-driver-3.24.0 geomet-0.2.1.post1
You are using pip version 9.0.1, however version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Saisir le code suivant dans une nouvelle cellule pour importer les librairies nécessaires pour ce projet

2 - chargement des librairies nécessaires

```
In [3]: import pandas as pd
from cassandra.cluster import Cluster
from datetime import date
from datetime import datetime

today = date.today()
# dd/mm/YY
date_ingestion = today.strftime("%Y-%m-%d")
print("date_ingestion =", date_ingestion)

date_ingestion = 2020-11-22
```

Charger les fichiers CSV dans des DataFrames en utilisant Pandas

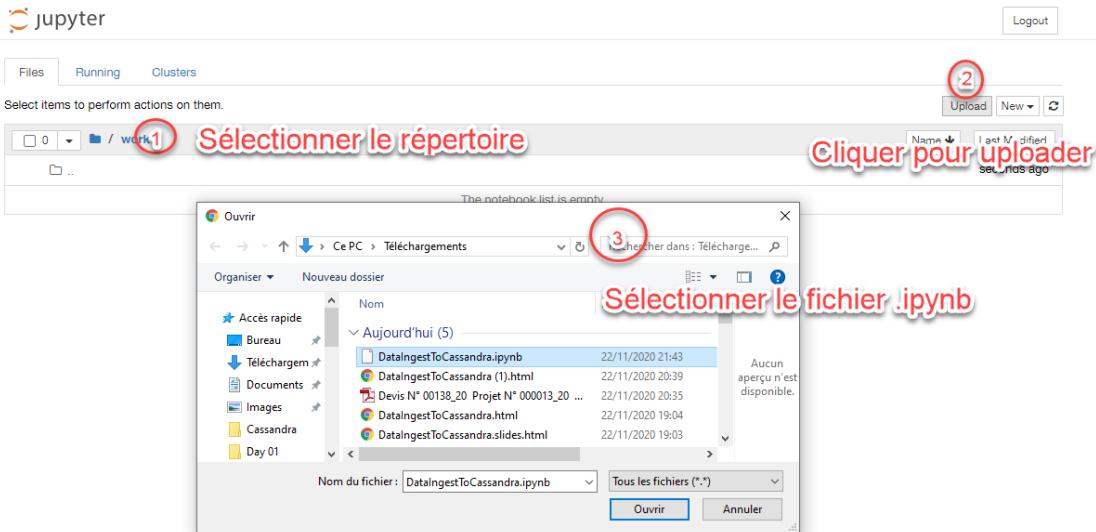
2 - Extraire les données COVID (CSV) à partir de dépôt gihub
1- Chargement de données à partir du web

```
In [1]: #specifier les urls nécessaires au données COVID-19
url_confirmed = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_
url_deaths= 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_
url_recovered='https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_
```

Charger un notebook existant sur le disque local

BASE DE DONNEES NO SQL Cassandra

Formateur : Sellami Mokhtar
mokhtar.sellami@gmail.com



Enregistrement des notebooks sous plusieurs formes (html, pdf, ipynb, python py)

File Edit View Insert Cell Kernel Widgets Help

New Notebook Open... Make a Copy... Rename... Save and Checkpoint Revert to Checkpoint Print Preview Download as Trusted Notebook Close and Halt

1- Inst

In []: `!pip install pandas==0.20.0`
`!pip install pandas==0.20.0.`

2 - chargement des librairies nécessaires

51.79.25.242:8888/notebooks/work/DatalngestToCassandra.ipynb# In []: import Cluster