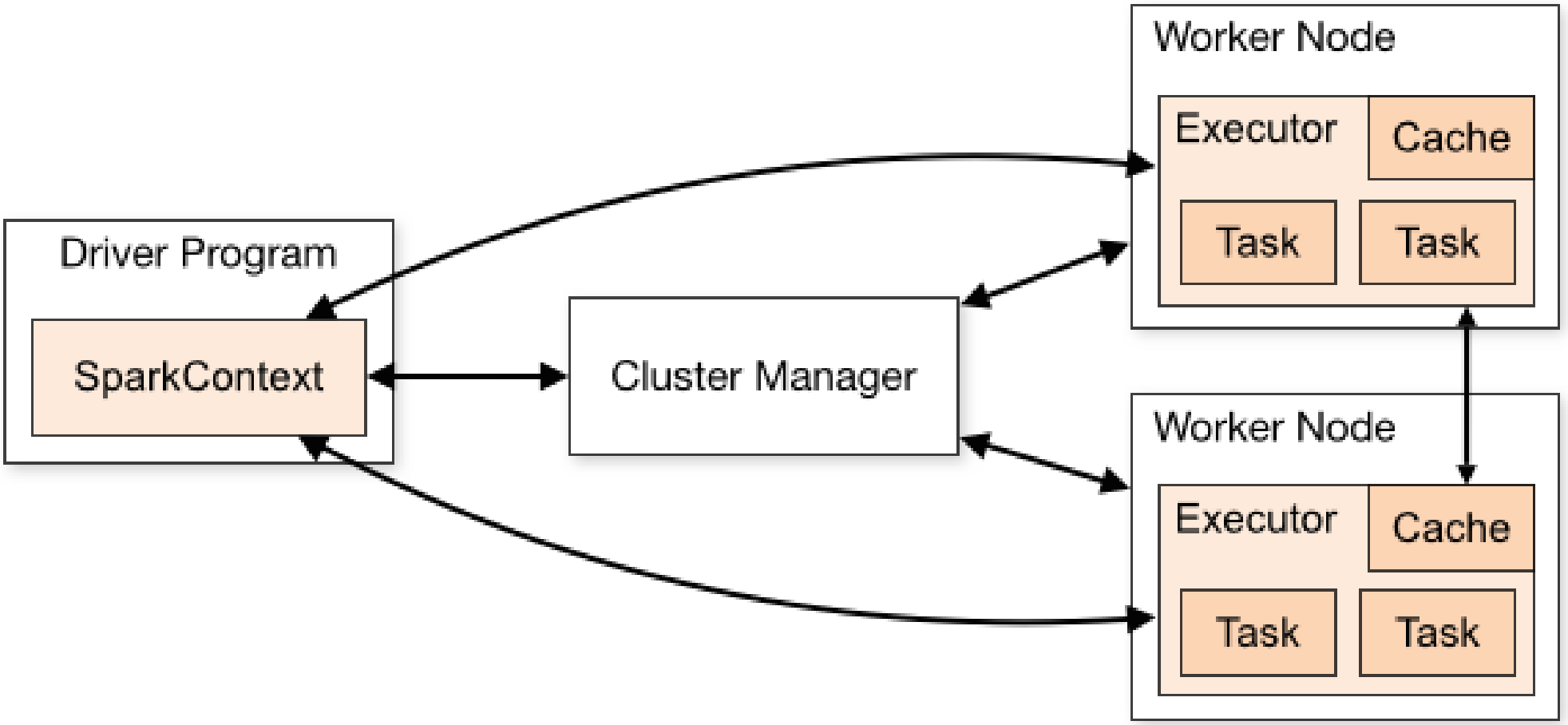
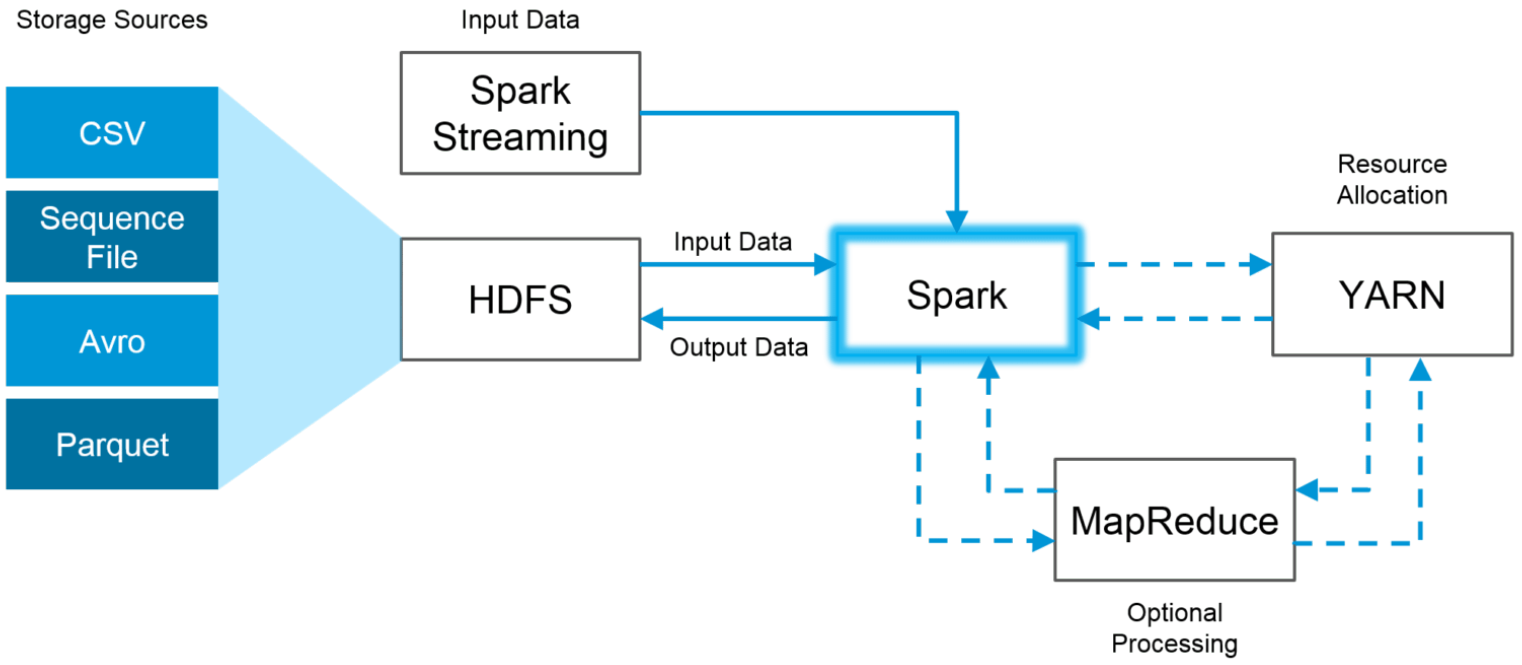

PySpark

Tour d’horizon : qu’est-ce que Spark et qu’apporte-t-il par rapport aux jobs MapReduce ?

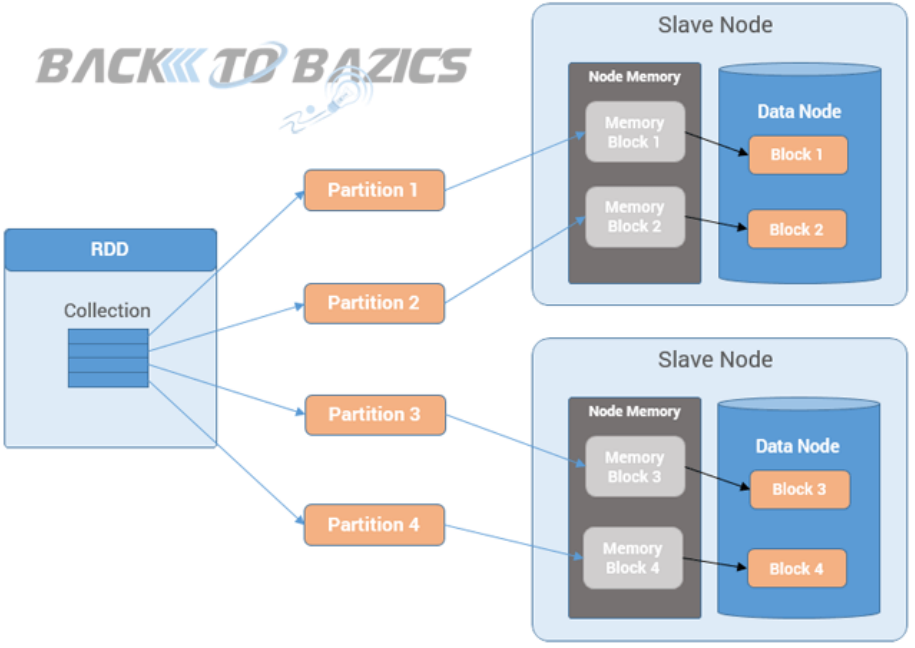
Spark est un framework de calcul distribué



Fonctionnalités similaires à un map reduce

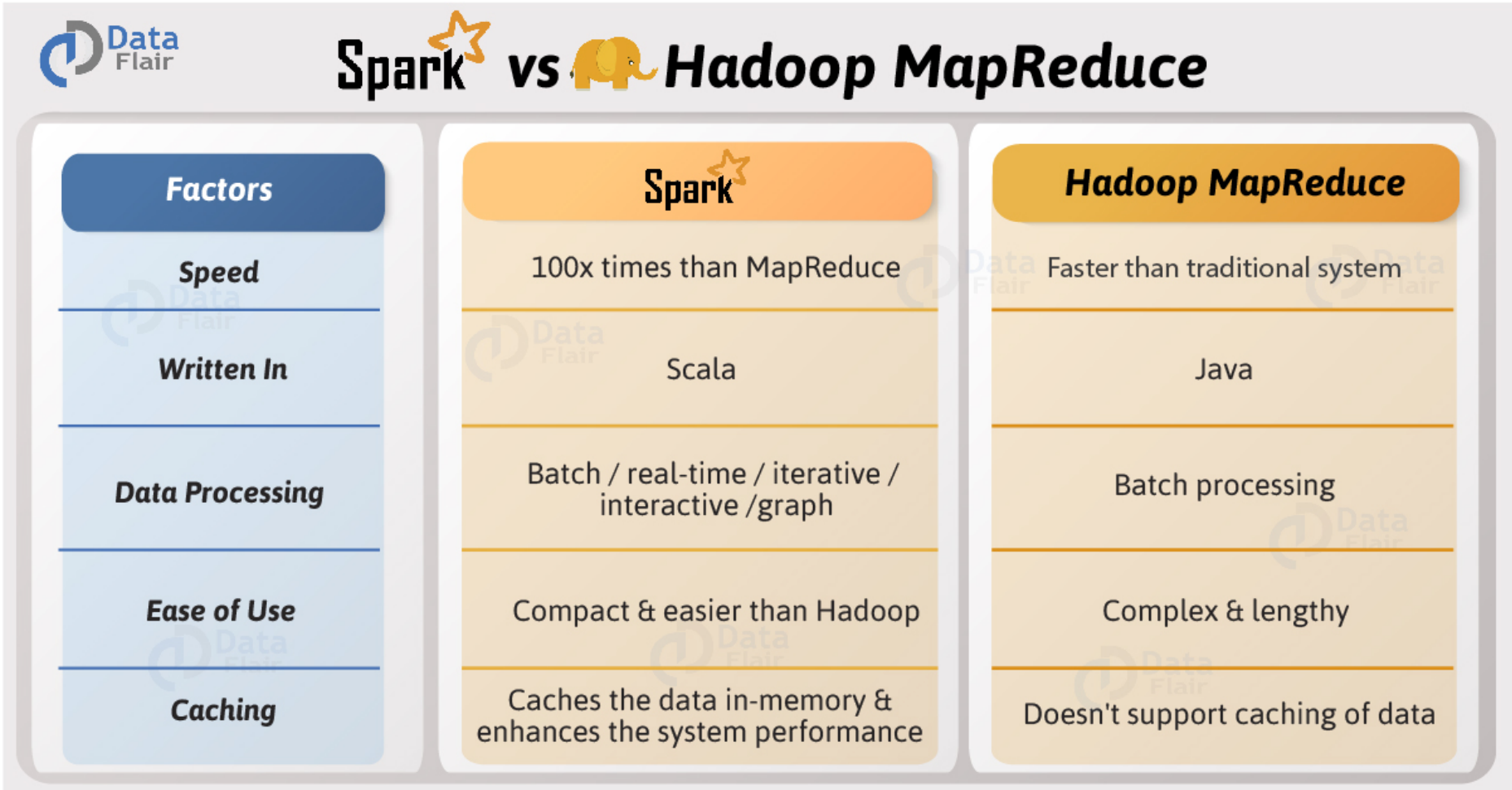


Mais utilisation particulière de la RAM des workers

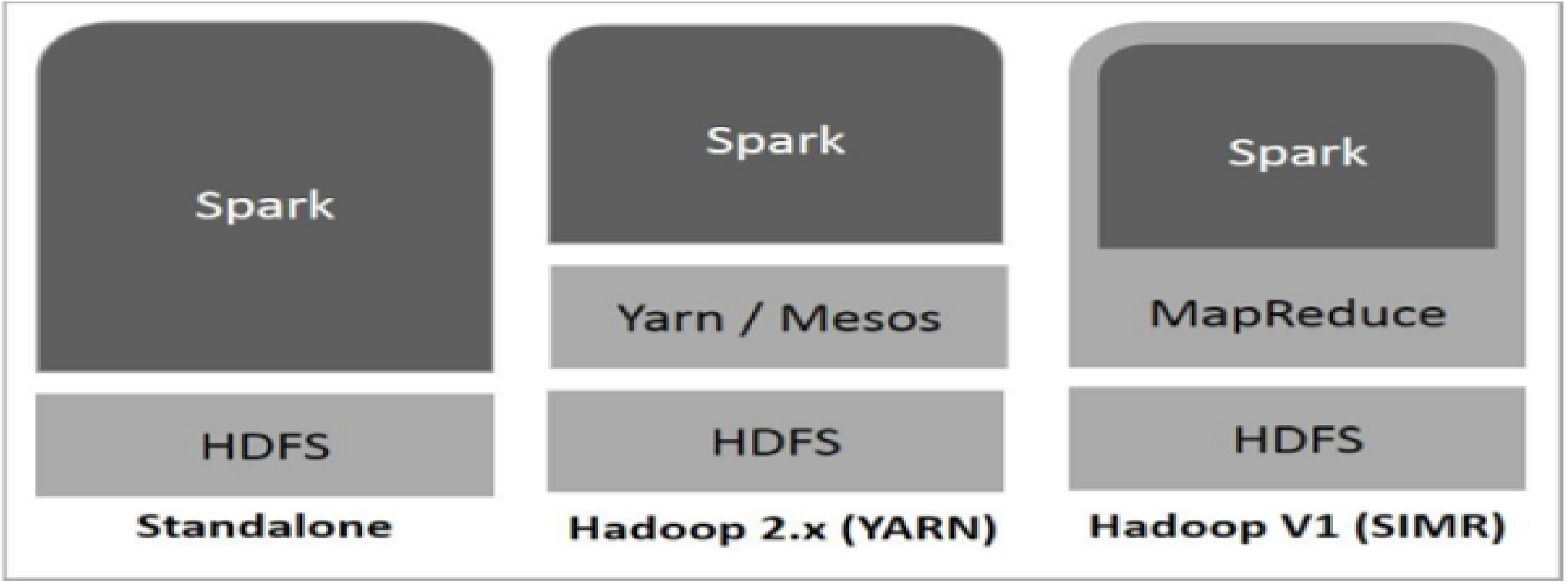


- Caching
- Temps réel (vs Batch processing)

En résumé


Data Flair
Spark vs Hadoop MapReduce
Factors
Speed
Written In
Data Processing
Ease of Use
Caching
Spark
100x times than MapReduce
Scala
Batch / real-time / iterative / interactive /graph
Compact & easier than Hadoop
Caches the data in-memory & enhances the system performance
Hadoop MapReduce
Faster than traditional system
Java
Batch processing
Complex & lengthy
Doesn't support caching of data

Intégration dans l’écosystème



Anatomie d’une application Spark

NB : (« application » | Spark) = (« job » | MapReduce)

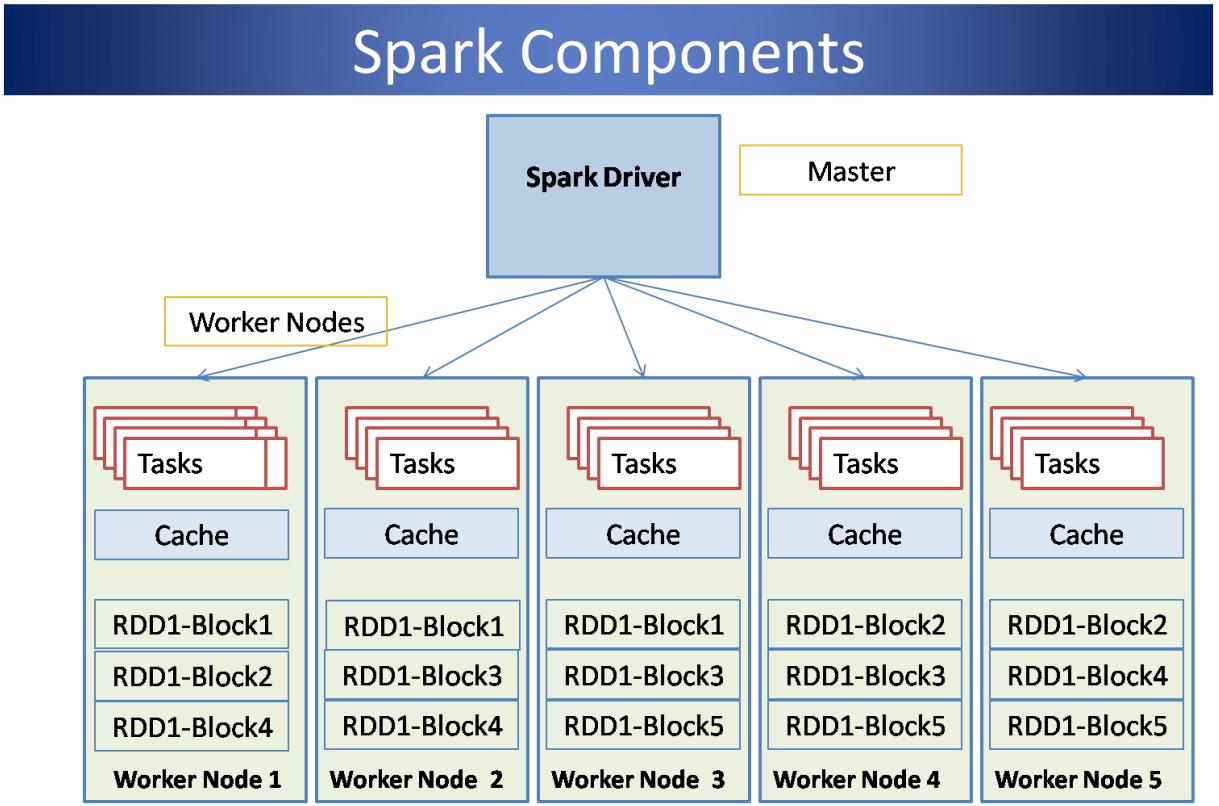
Une application Spark commence toujours par l’instanciation d’un « **spark context** »

```
from pyspark import SparkContext
sc = SparkContext("local", "First App")
```

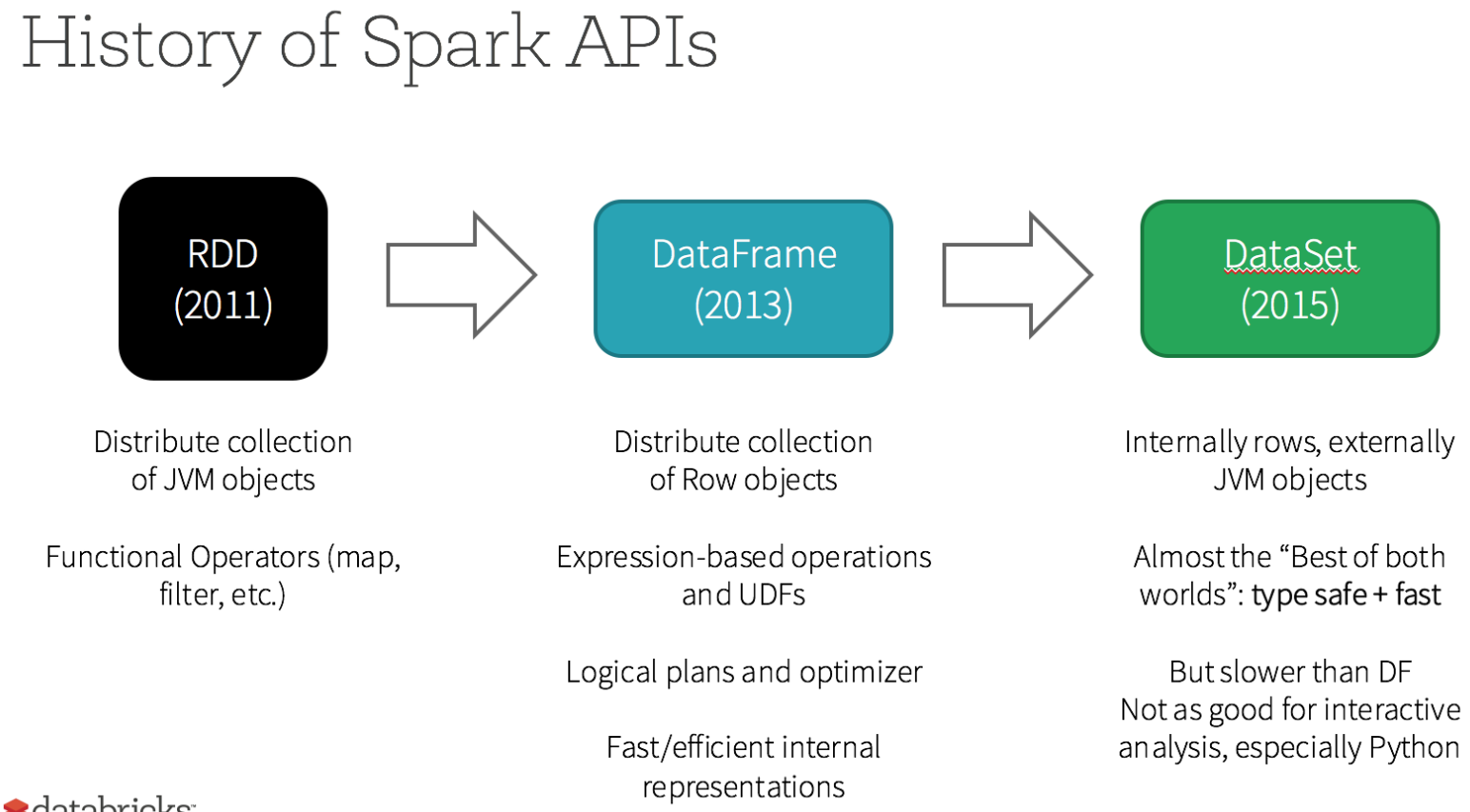
(indique au driver l’existence d’une application pour qu’il lui alloue des workers)

Mais que sont ces RDD (Résilent distributed datasets) ?

En un mot, des collections distribuées sur la ram des workers



Nous rencontrerons d'autres structures de données propres à Spark



Mise en place de l’environnement

Environnement technique

Hortonworks sandbox (virtualbox)

https://fr.hortonworks.com/tutorial/hortonworks-sandbox-guide/

Ajout du nom d’hôte

Ajouter la ligne suivante :

```
127.0.0.1 sandbox-hdp.hortonworks.com sandbox-hdf.hortonworks.com
```

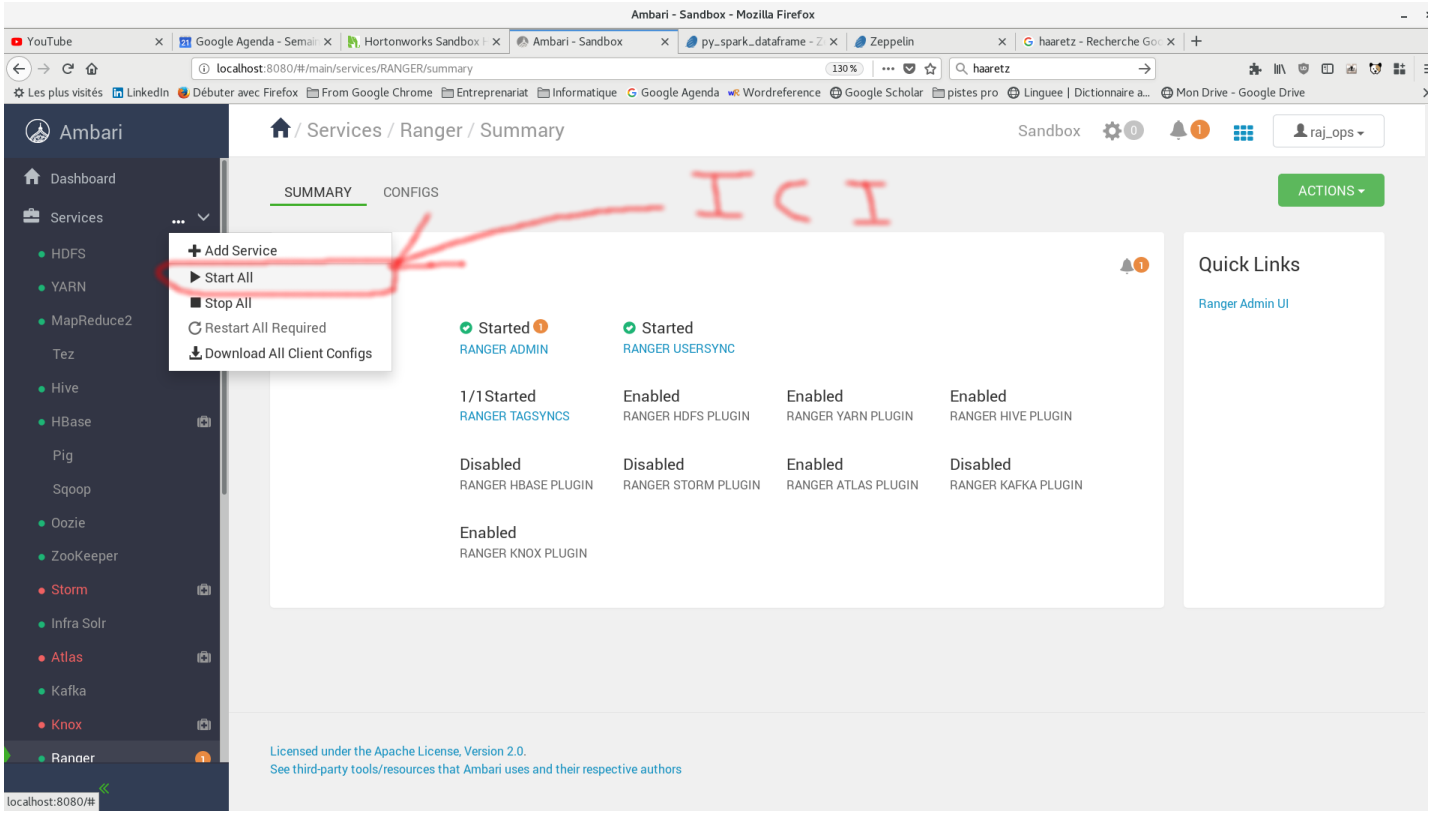
- Sous linux/MacOSX dans /etc/hosts
- Sous Windows C:\Windows\System32\Drivers\etc

Démarrage des services

Étape 1 se rendre sur **Ambari** (port 8080)

Les credentials sont

- User : **raj_ops**
- Mdp : **raj_ops**

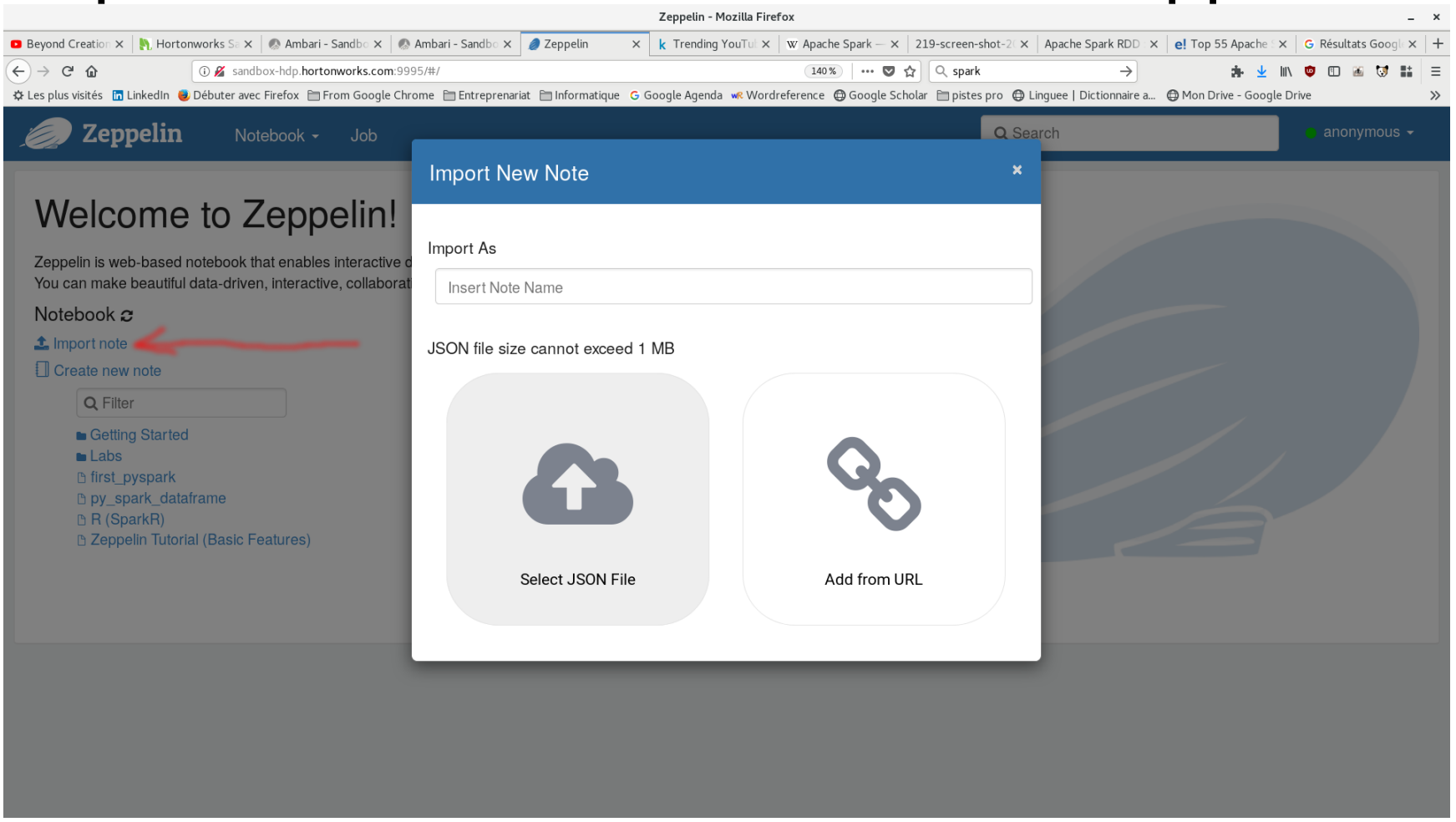


Démarrage des services

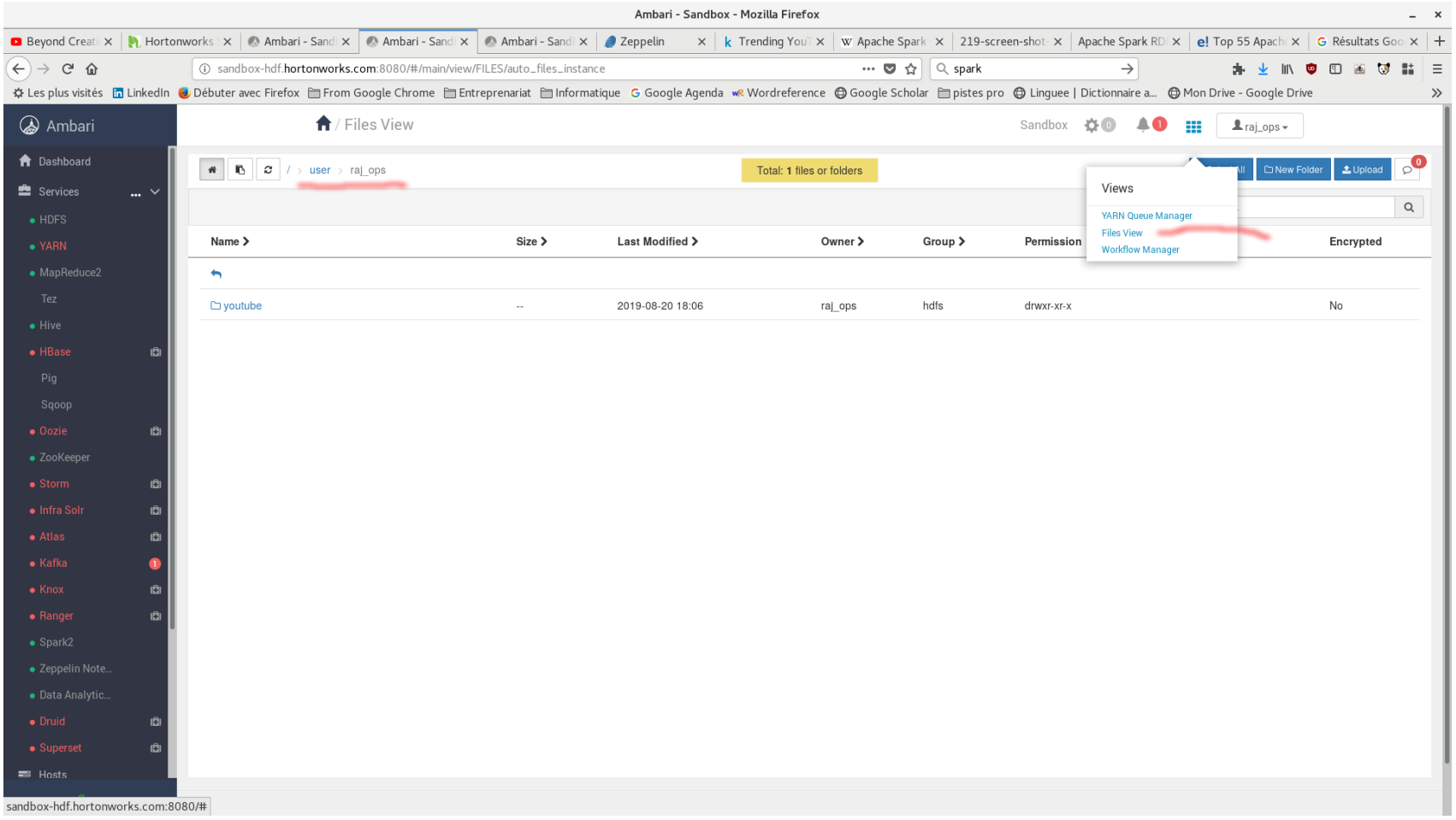
Assurez-vous que les services suivants fonctionnent (le démarrage de l'ensemble de ces services est un peu long, c'est tout à fait normal)

- HDFS
- YARN
- HIVE (spark.sql va en avoir besoin)
- SPARK2
- ZEPPELIN NOTEBOOK (un genre de jupyter)

Importation des notebooks dans Zeppelin

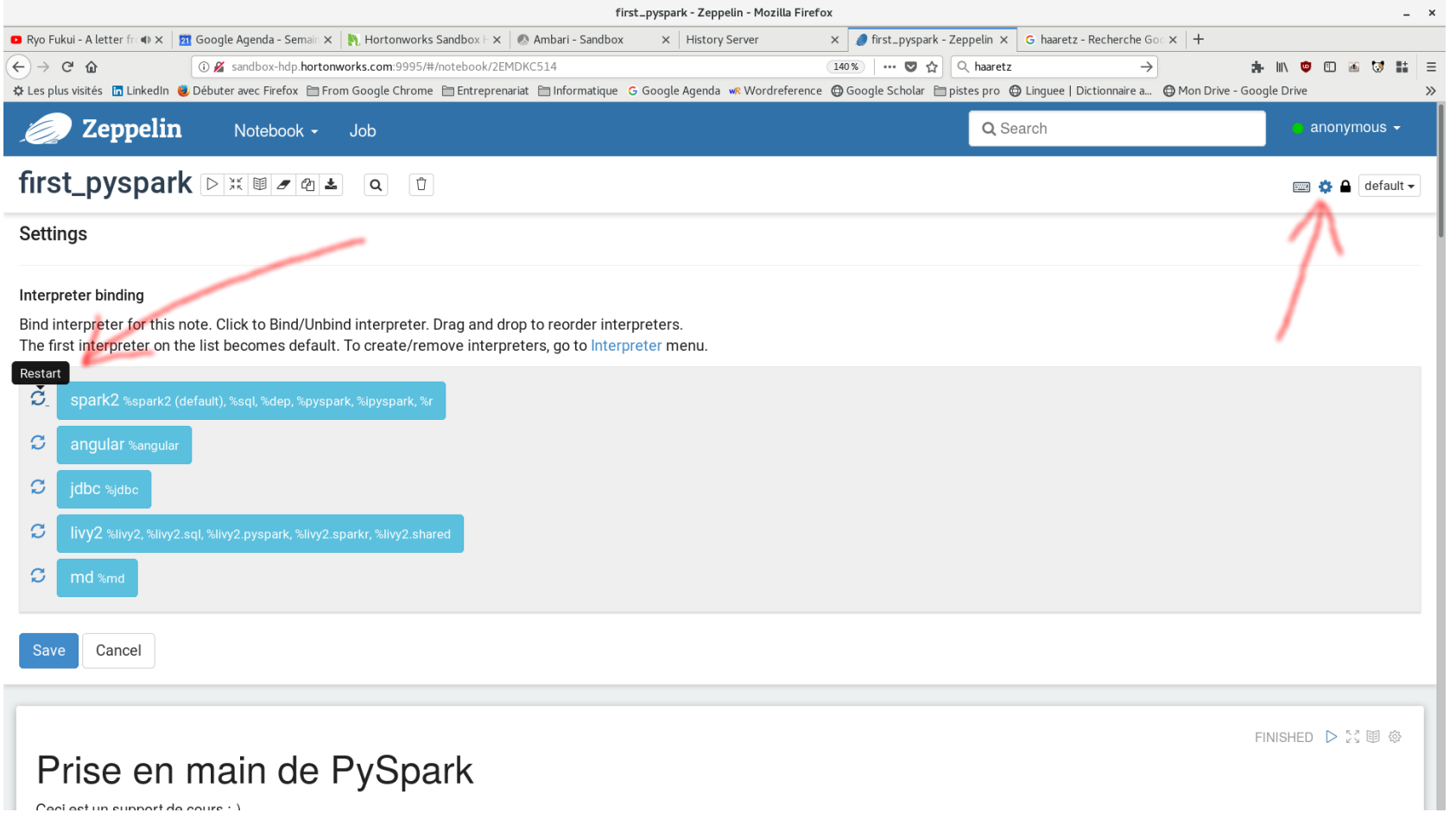


Importation des données dans l'HDFS



Troubleshooting

Erreur Zeppelin : interpreter not found



Les jobs ne se lancent pas => YARN

Il peut être intéressant de regarder si **trop de jobs** ne sont pas en cours sur la plateforme et les KILL au besoin

http://sandbox-hdp.hortonworks.com:8088