Technologies BigData

2019

hétic
La grande école de l'Internet

Plan du cours

- **Concepts Hadoop (0.5 jours)**
  - Histoire
  - MapReduce
  - Architecture Hadoop
  - Single Node Cluster (Marchine Virtuelle)
- **Atelier : commandes Shell (0.5 jours)**
  - Lancer les outils Hadoop / Hive / spark / Python
  - Commandes shell Hadoop de base
- **Atelier : Données Structurées avec HIVE (0.5 jours)**
  - Concepts
  - Pratique
- **Atelier : InMemory avec SPARK (1.5 jours)**
  - Concepts SPARK / Python Spark
  - Pratique SparkSQL : régression linéaire simple sur données de séquences
  - Pratique SparkMLLIB : classification avec une Random Forest

hétic
La grande école de l'Internet

Ouvrages et liens

- **Livres :**
  - Hadoop The definitive guide, Tom White, O'REILLY
  - Learning Spark, Holden Karau, O'REILLY
  - Python for Data Analysis, Wes McKinney, O'REILLY

- **Pdf :**
  - Forêts aléatoires : https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

- **Liens :**
  - ***Documentation Hadoop : https://hadoop.apache.org/***
  - ***Documentation Hive : https://cwiki.apache.org/confluence/display/Hive/LanguageManual***
  - ***Documentation SPARK : https://spark.apache.org/docs/2.0.0/api/python/index.html***
  - ***Documentation iPython : http://ipython.org/documentation.html***
  - ***Documentation Python scikit-learn : http://scikit-learn.org/stable/documentation.html***

hétic
La grande école de l'Internet

Un peu d'histoire

- L'estimation par IDC (international data corporation) du volume de données stockées est en croissance exponentielle : **2006:** 0,18 ZB, **2011:** 1,81 ZB, **2020:** 40 ZB (1 ZB = 1 Milliard de TB)

- Le Big Data est initialement une technologie informatique créée pour gérer ces volumes de données :

Google
2003

hadoop
YAHOO!
2006

hadoop
The Apache Software Foundation
http://www.apache.org/
2008

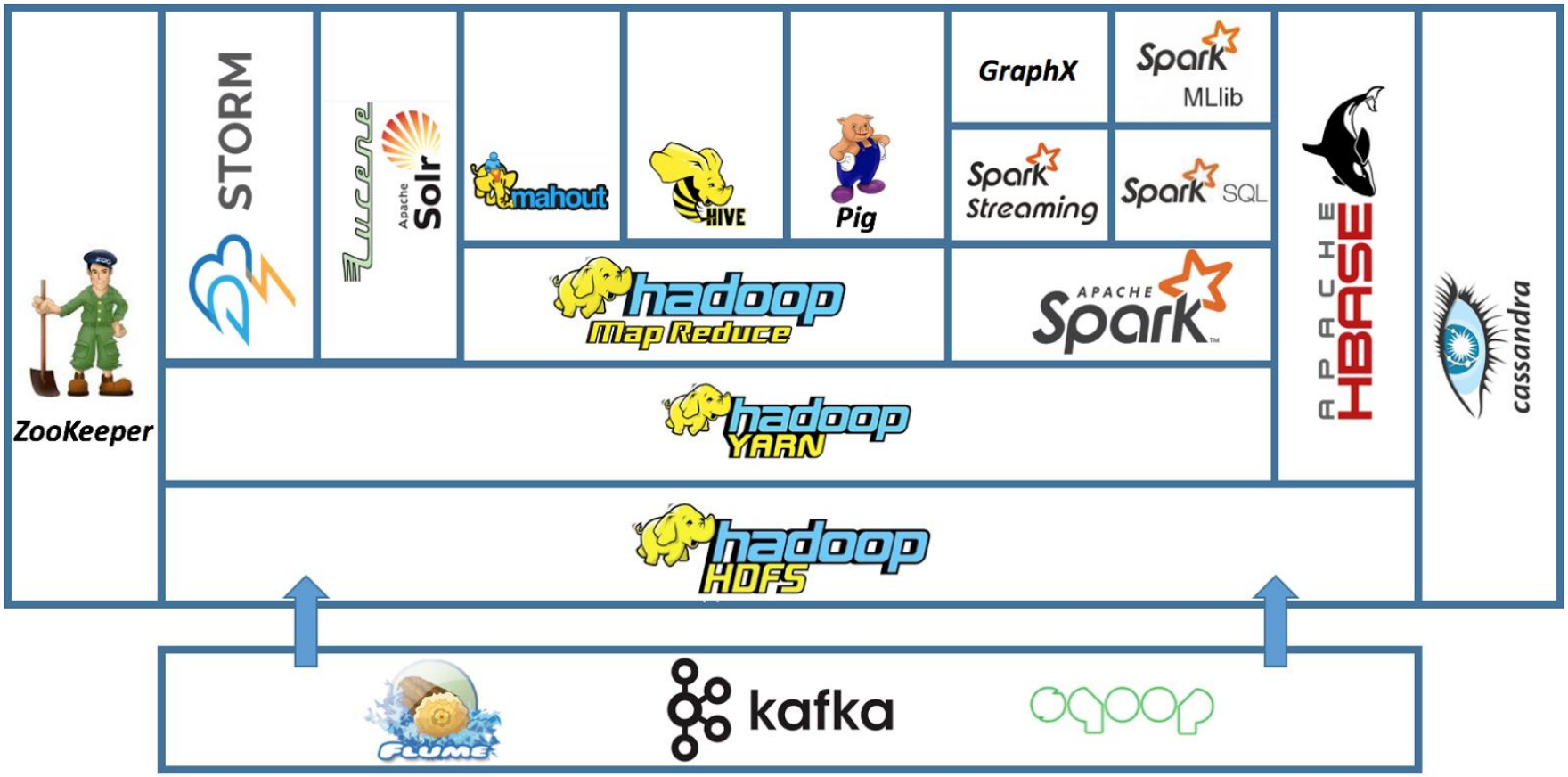
- Cette technologie repose sur le principe de parallélisation des tâches à très grande échelle

- L'adoption du projet Hadoop par la communauté OpenSource a permis le développement d'un écosystème d'outils capable d'interagir avec cette technologie (base de données, data visualisation, data analytics…)

- Le « BigData » devient aujourd'hui un domaine d'activité qui s'appuie sur l'exploitation des données numériques pour développer de nouvelles opportunités pour les entreprises

hétic
La grande école de l'Internet

L'ecosystème Hadoop Apache



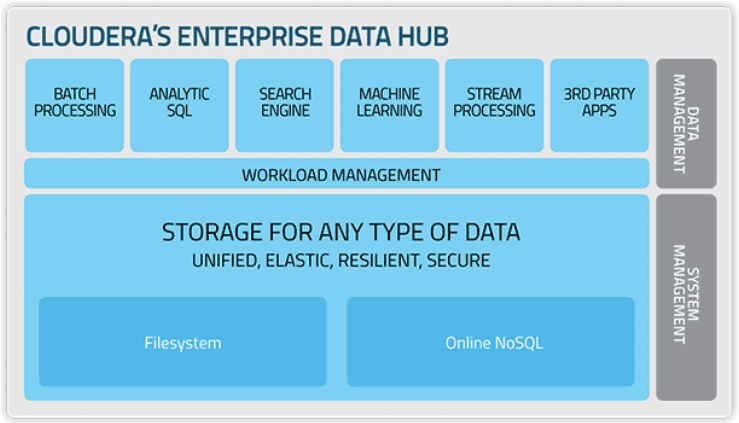
hétic

La grande école de l'Internet

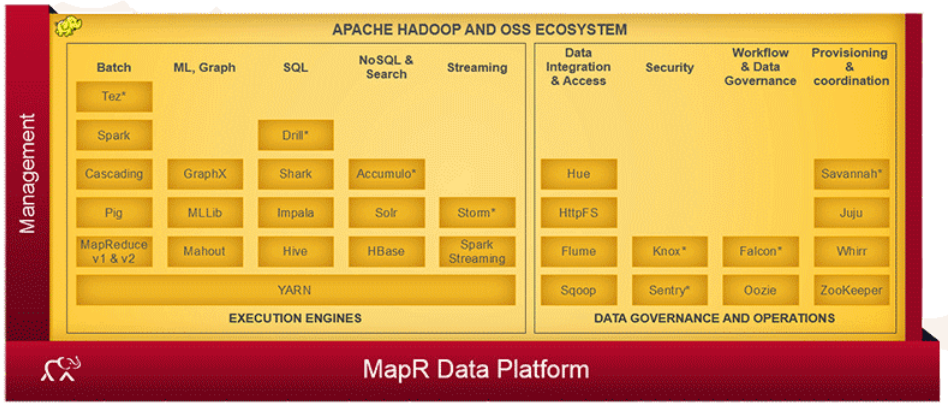
Les distributions payantes

- 3 acteurs principaux distribuent les briques Hadoop, garantissent la bonne configuration et installation des différents outils et proposent une interface unifiée de management (par le biais d'interface comme Hue notamment) :

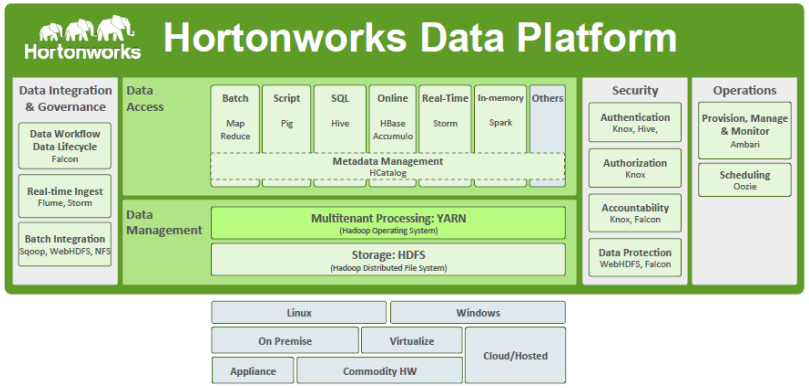
CLOUDERA



MAPR

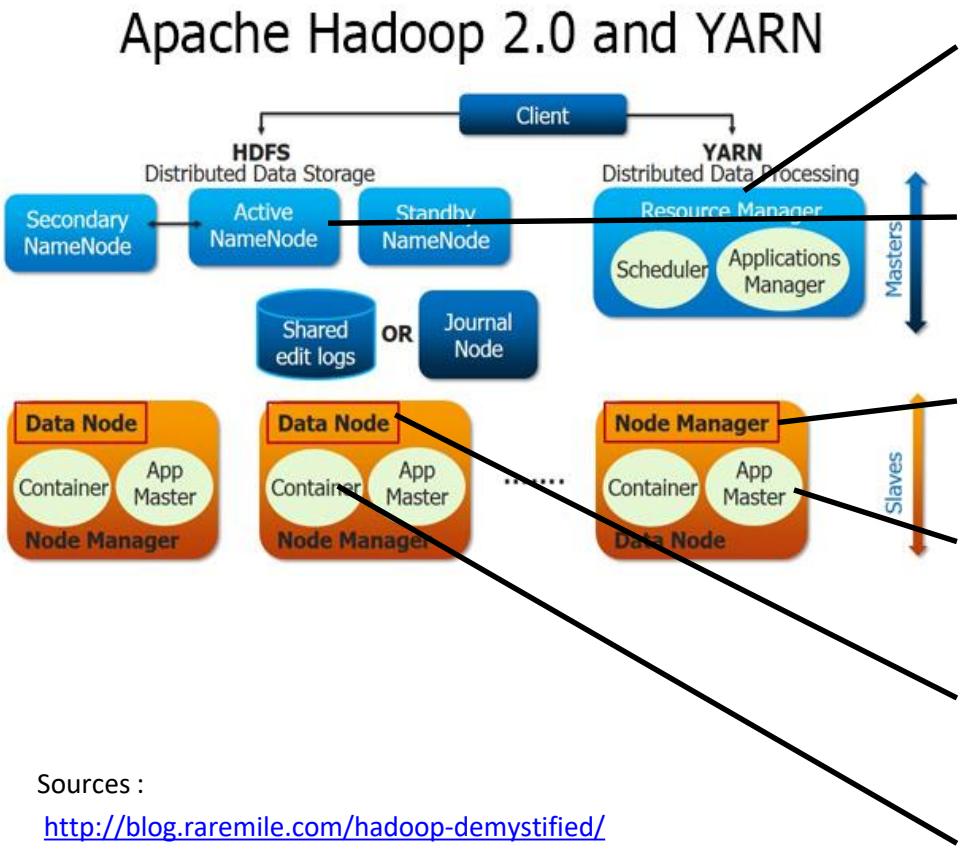


HORTONWORKS



Note : illustrations au 10/2015

Architecture Hadoop



Le Ressouce Manager est le maître qui organise les tâches et alloue les ressources disponibles sur le cluster. Il travaille avec chaque Node Manager et chaque Application Master

Le NameNode est l'élément principal du système de fichier HDFS. Il historise toutel'arborescence des fichiers et trace leur emplacement sur le cluster (il ne stock par contre aucune donnée)

Le node Manager reçoit mes instructions du Ressource manager. Il gère les ressources diposnibles sur un noeud.

L'application Master négocie les ressources avec le ressource manager et exécute et track la tâche avec le node manager pour remonter l'état d'avancement au ressource manager.

Datanode : Un datanode stock les données. Il y en a plusisurs et les données sont répliquées entre ces noeuds.

Container : C'est une collection de l'ensemble des ressources nécessaires pour exécuter une tâche pour une application (code, librairies etc…). Lorsqu'une application est lancée sur le cluster, le Ressource Manager planifie les tâches et lance l'éxécution de la tâche (container) sur les noeuds esclaves (slaves)

Sources :

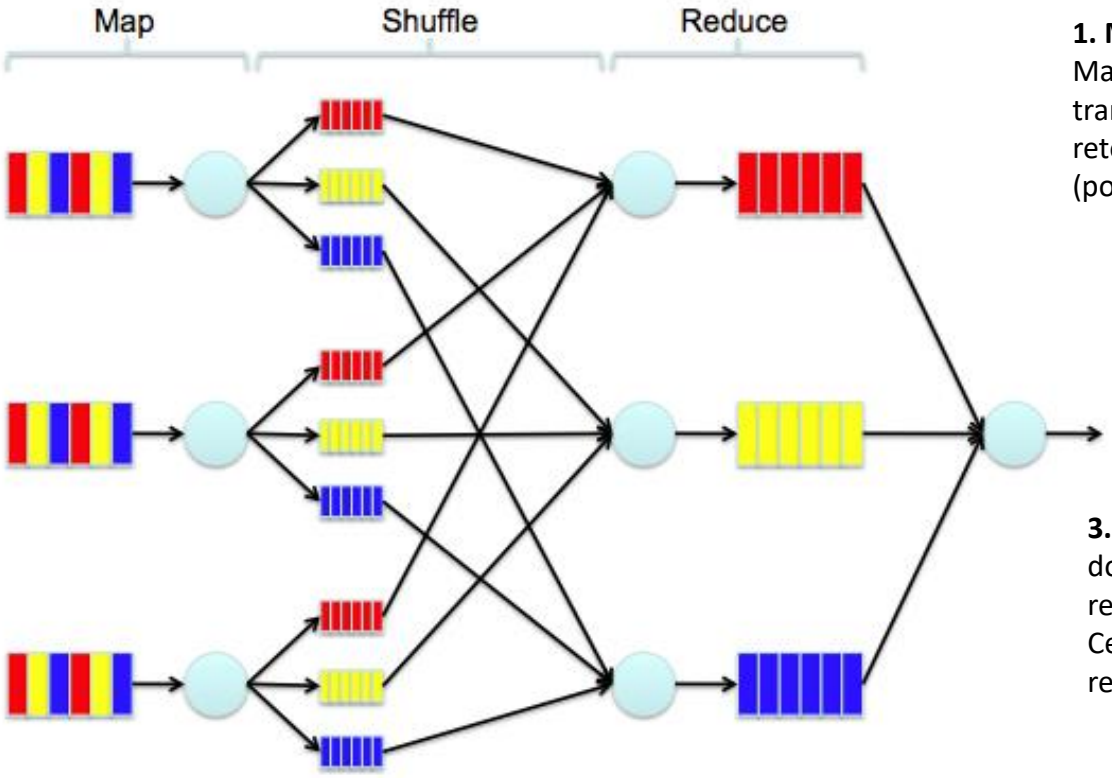
http://blog.raremile.com/hadoop-demystified/

http://www.edureka.co/blog/apache-hadoop-2-0-and-yarn/

hétic

La grande école de l'Internet

L'algorithme MapReduce



1. MAP : C'est la première étape d'un programme MapReduce (mapping). Les données sont transmises une à une fonction (mapper) qui retourne pour chaque entrée une nouvelle donnée (potentiellement transformée)

2. SHUFFLE : cette étape transfert les données issues des mappers (N) vers les X reducers. Les données sont triées durant cette étape pour simplifier/accélérer le traitement par les reducers.

3. REDUCE : L'étape Reduce permet d'aggréger les données élémentaires. Une fonction reduce reçoit un iterateur sur une liste de données. Cette fonction combine les valeurs pour ne retourner qu'une seule valeur en sortie.

hétic
La grande école de l'Internet

DAG (directed acyclic graph)

- **MapReduce :** La mécanique mapReduce implique un traitement séquentielle des trois étapes (map, shuffle, reduce). Cela génère des temps de latence entre chaque étape , ce phénomène est amplifié si la tâches doit exécuter plusieurs sous-tâches mapReduce : les tâches étant bloquées tant que les tâches précédentes ne sont pas terminées.

- **DAG :** Une nouvelle génération d'outils visent à réduire ces temps de réponse. Dans ce contexte un graphe acyclique orienté (DAG) est une modèle de planification des tâches ou les jobs sont représentés comme des sommets dans un graphe dont l'ordre d'exécution est défini par la direction des arêtes dans le graphe. L'idée étant de trouver le chemin direct le plus cours pour exécuter la tâche. Chaque sommet indépendant peut être exécuté en parallèle plutôt que séquentiellement. Dans cette mécanique qui repose tout de même sur des étapes mapReduce, le problème de la latence entre les étapes demeure. Pour réduire cette latence les outils faisant appel au principe DAG pour organiser les tâches utilisent de la mémoire vive (Spark, Tez…). Les données sont chargées en mémoire ce qui améliore les temps d'accès. Un autre avantage de du DAG est la tolérance aux échecs. En cas d'erreur d'un job, il est facile de revenir en arrière dans le graphe et de ré-executer l'ensemble des jobs en échec.

- **Apache Spark et TEZ :** ces 2 framework font partie des outils qui utilisent ce principe , ils ont aussi recours à l'utilisation de la mémoire vive pour réduire les accès I/O sur les disques. Tez permet l'exécution des tâches Hive et Pig et Spark permet d'exécuter du code (scala, python, java) sur le cluster selon ce principe.

Source : http://mammothdata.com/dag-vs-mapreduce/

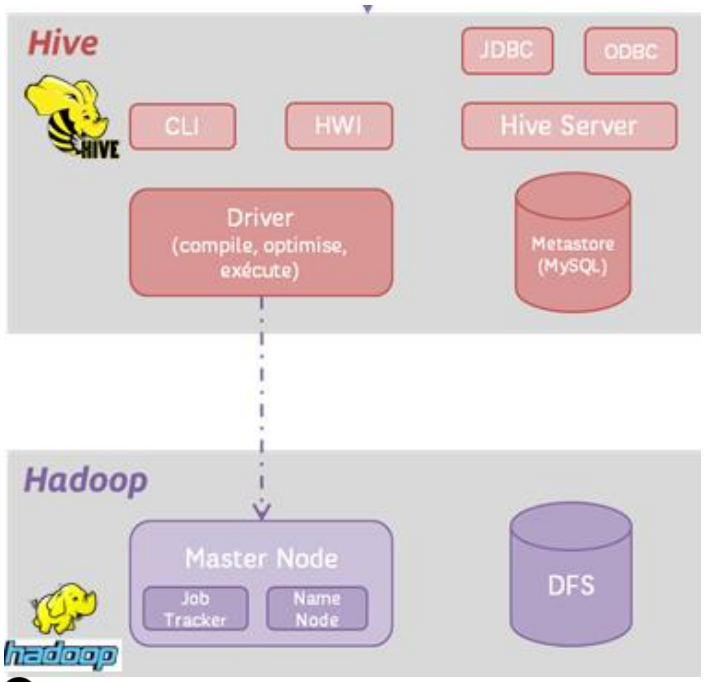
Plus d'infos : https://www.youtube.com/watch?v=A5BwUNuR_o8&list=PLAXcUSglh5YhN5PRgJRyqITPO5R21Hkff

hétic

La grande école de l'Internet

Hive

Hive est une solution de datawarehousing qui reposer sur l'infrastructure Hadoop. Les données y sont structurées et représentées comme en tables. Les données sont manipulées en HQL une syntaxe très proche du langage SQL.



Les composants Hive

Driver Hive **:** Permet d'interpréter, de compiler, d'optimiser et d'exécuter des commandes HQL.

Metastore **:** composants indispensable à Hive. Il s'agit d'une base de données (généralement MySQL) annexe permettant de stocker la structure, l'emplacement, le schéma et les métadonnées des tables Hive.

Les interfaces EXTERNES

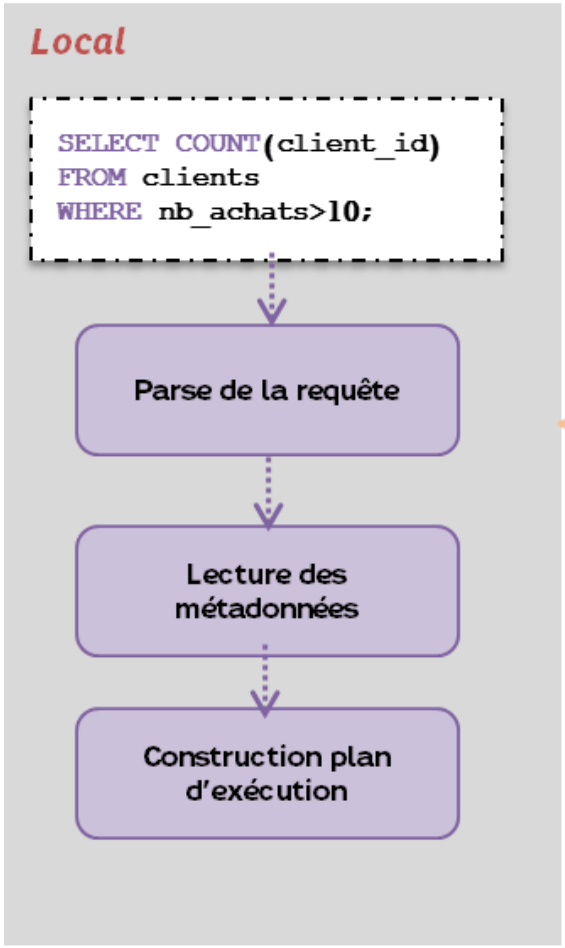
Le CLI (Command Line Interface) est le moyen le plus simple d'interagir avec Hive. Il s'agit d'une interface de type commande, dans laquelle il est possible d'entrer des requêtes Hive.

HWI (Hive Web Interface) est une interface Web simple permettant d'accéder à Hive à distance.

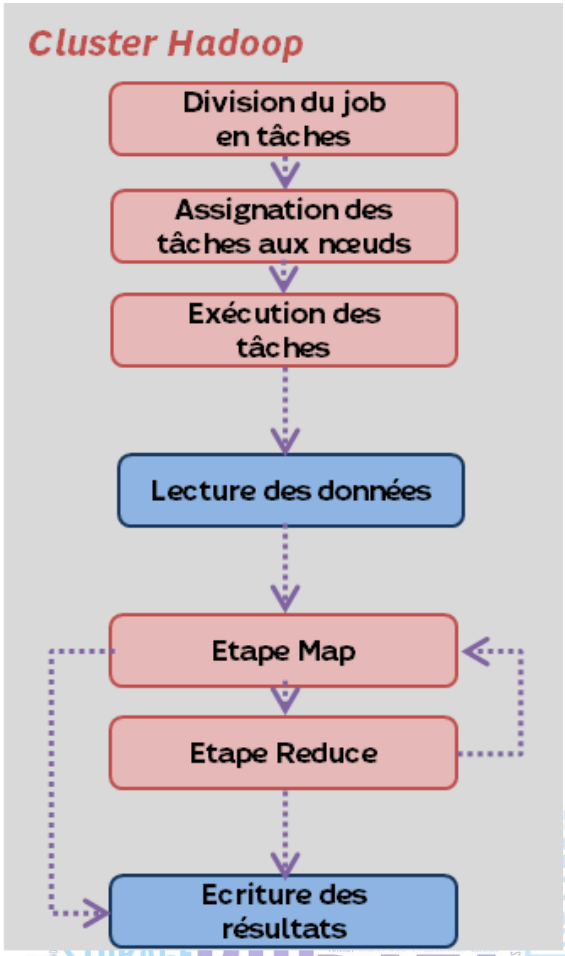
JDBC (Java DataBase Connectivity) et ***ODBC*** (Open Database Connectivity) servent à de lancer des requêtes Hive à partir d'autres applications (par exemple R, SAS), grâce à leurs drivers respectifs.

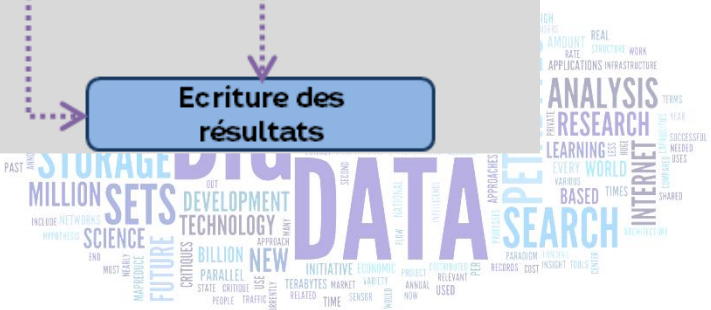
hétic
La grande école de l'Internet

Hive : Exécution d'une requête



Soumission MapReduce au cluster





hétic
La grande école de l'Internet

SPARK

Spark est un framework de calcul distribué (sur un cluster) open source développé en scala initialement parle AMPLab à l'universtié de Californie (Berkeley). Il diffère du principe de base d'Hadoop (map reduce basé sur des I/O disques) en utilisant la mémoire vive du cluster pour réaliser des tâches complexes (à plusieurs niveaux). C'est aujourd'hui l'un des projets Apache les plus actifs. Ces composants :

Spark Core : c'est le cœur du projet, il permet de planifier, distribuer des tâche sur un cluster. Il utilise le concept de RDD (resilient distributed datasets) qui sont des données partitionnées entre les différentes machines disponibles (en mémoire, mais aussi sur disque si nécessaire). Les RDD peuvent être manipulés par 3 langages de programmation : Java, Scala et Python

Spark SQL : c'est un composant de Spark qui permet de gérer des données structurés à l'aide de dataframe (tableaux nommés) et d'un langage proche du SQL.

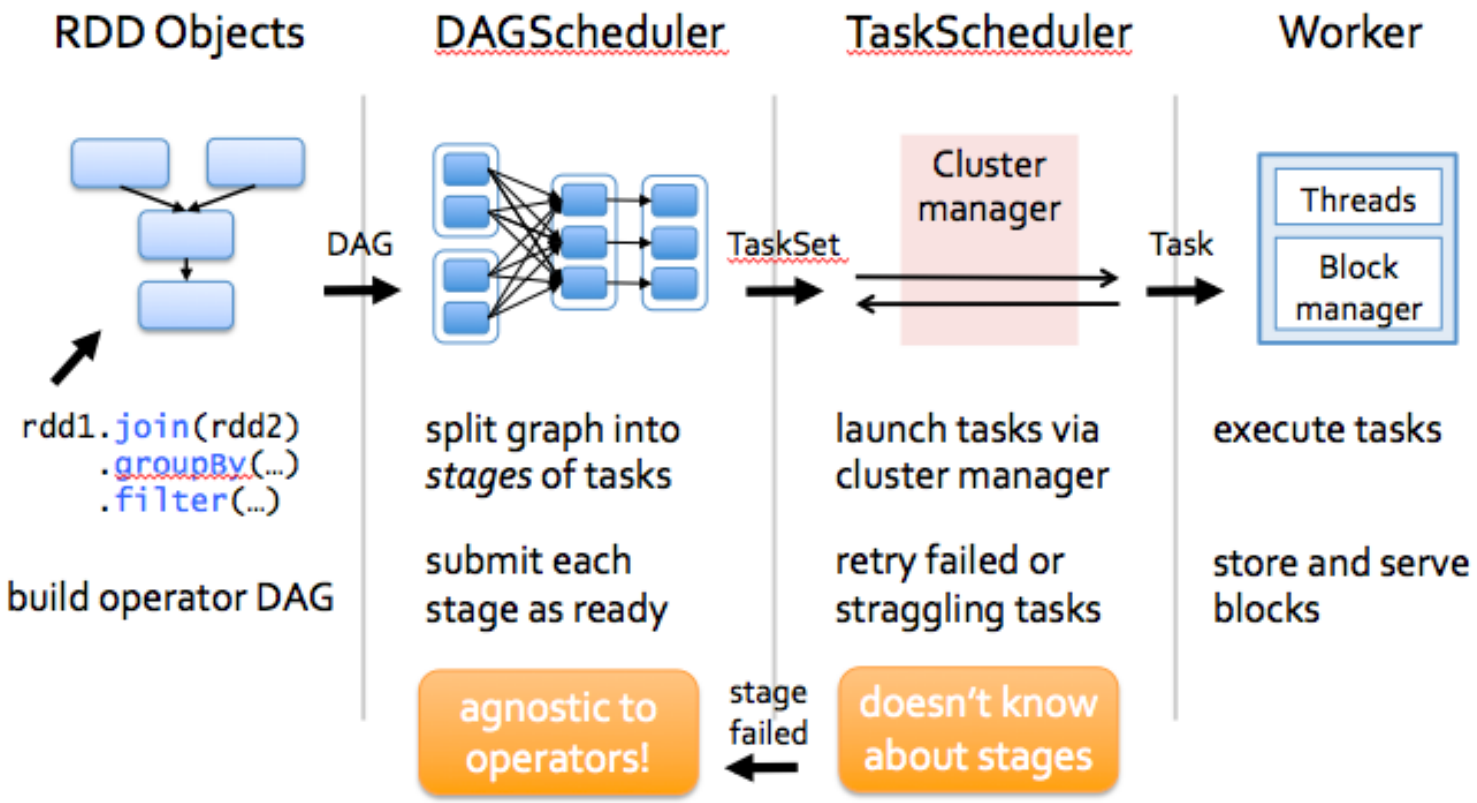
Spark MLLIB : c'est une librairie qui contient des fonctions de machine learning qui s'exécutent en parallèle sur un cluster.

Spark Streaming : un composant qui permet de gérer des flux de données / jobs en quasi temps réel (on parle de microbash)

Spark GraphX : un environnement dédié au traitement des graphs (représentation d'entités et de leurs relations) (plus d'infos : http://www.dummies.com/how-to/content/graph-processing-in-hadoop.html)

hétic
La grande école de l'Internet

DAG SPARK



hétic
La grande école de l'Internet

Environnement de travail

- ***Virtual Machine Single Node cluster :***
  - Ubuntu server 16.04.03 (64 bits)
  - Hadoop 2.6.5
  - Hive 1.2.2
  - Spark 1.6.3
  - Jupyter Notebook (module matplotlib, scipy, numpy, statsmodels, pandas, scikit-learn)
- ***Ressources (minimales)de la VM pour les ateliers :***
  - 1512 mo ou 2G RAM
  - 2 CPUs
  - 30 Go

hétic
La grande école de l'Internet

Commandes Linux sur la VM

- Serveur Ubuntu commandes SHELL :
  - Login **hadoop**, mot de passe **hadoop**
  - Se déloguer : **logout**
  - Partage entre la VM et le poste hôte : **/home/hadoop/public/**
  - Obtenir l'IP de la VM : **ip addr show**
  - Obtenir sa position dans l'arborescence : **pwd**
  - Se déplacer dans les dossiers : **cd « chemin ou l'on souhaite aller »**
  - Lancer les services hadoop : **start-dfs.sh && start-yarn.sh**
  - Stopper les services hadoop : **stop-dfs.sh && stop-yarn.sh**
  - Lancer hive shell : **hive**
  - Lancer le serveur ODBC hive : **$HIVE_HOME/bin/hiveserver2**
  - Lancer Spark shell : **$SPARK_HOME/bin/pyspark –master (local[*] ou yarn-client)**
  - Lancer Spark SQL : **$SPARK_HOME/bin/spark-sql –master (local[*] ou yarn-client)**
  - Lancer un Master Spark : **$SPARK_HOME/sbin/start-all.sh**
  - Stopper le Master Spark : **$SPARK_HOME//sbin/stop-all.sh**
  - Lancer le notebook iPython : **jupyter notebook**
  - Lister les processus (java): **ps (jps)**
  - Killer un processus : **kill xxxxxx (xxxxxx = numéro du processus)**

hétic
La grande école de l'Internet

Commandes Linux sur la VM

- Commandes de base SHELL HADOOP :
  - Parcourir le HDFS : **hadoop fs –ls**
  - Créer un dossier : **hadoop fs mkdir**
  - Copier un fichier sur le HDFS : **hadoop fs –put sourcelocale desthdfs**
  - Copier un fichier à partir du HDFS : **hadoop fs –get sourcehdfs sourcelocale**
  - Effacer un fichier sur le HDFS : **hadoop fs –rm nomdufichier**
  - Effacer un dossier sur le HDFS : **hadoop fs –rm r nomdudossier**
  - Contrôle des données sur le HDFS : **hdfs fsck /**
  - Killer un job hadoop : **yarn application –kill « application-id »**
  - Se déplacer dans un dossier : **hadoop fs –cd « chemin »**
  - Déplacer des éléments : **hadoop fs –mv « origine » « destination »**

Liste des commandes hadoop Shell : https://hadoop.apache.org/docs/r2.6.5/hadoop-project-dist/hadoop-common/FileSystemShell.html

hétic
La grande école de l'Internet

URL du cluster Single Node

Après avoir récupérer l'IP de la Machine Virtuelle (xxx.xxx.xxx.xxx), il est possible d'accéder aux interfaces web en utilisant les adresses suivantes :

- Web UI Hadoop du namenode : xxx.xxx.xxx.xxx:50070
- Web UI Hadoop du moniteur d'applications : xxx.xxx.xxx.xxx:8088
- Web UI du moniteur d'application SPARK : xxx.xxx.xxx.xxx:4040
- Web UI du Master Spark : xxx.xxx.xxx.xxx:8080
- Notebook Ipython : xxx.xxx.xxx.xxx:9091

Pour accéder au dossier partagé sur le Machine Virtuelle, tapez dans l'explorateur de fichier l'adresse suivante : \\xxx.xxx.xxx.xxx\public\ (il faudra s'authentifier avec le user hadoop pour accéder à ce dossier)

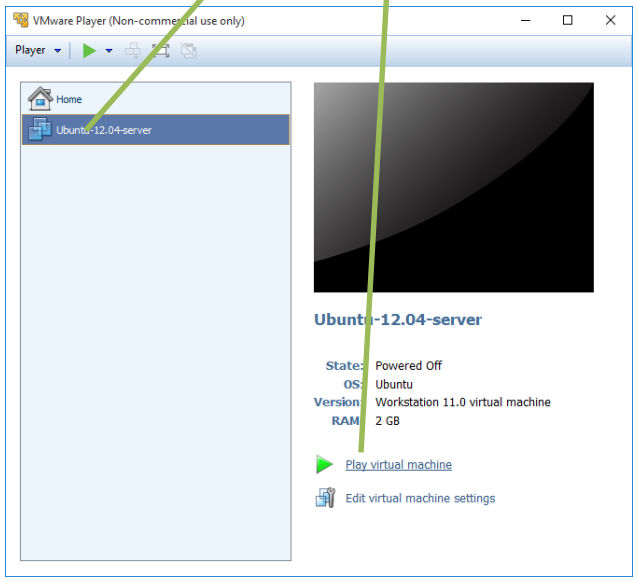
hétic
La grande école de l'Internet

Démarrer la VM

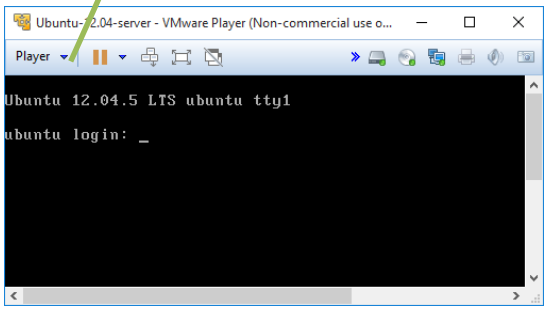
1. Cliquer sur l'icône du VmPlayer sur le bureau de l'ordinateur

2. Sélectionner la machine virtuelle Ubuntu 12.04 Server

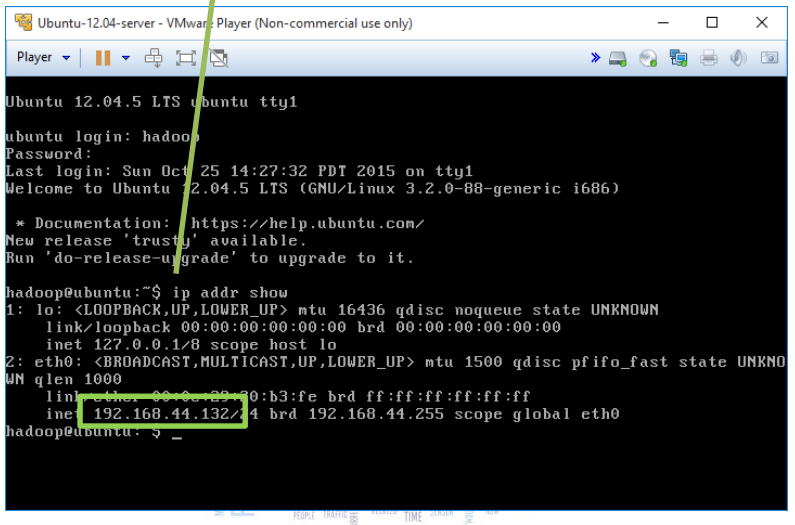
3. Démarrer le serveur



3.Une fois le serveur démarré vous pouvez vous authentifier sur la machine avec le user hadoop et la mot de passe hadoop

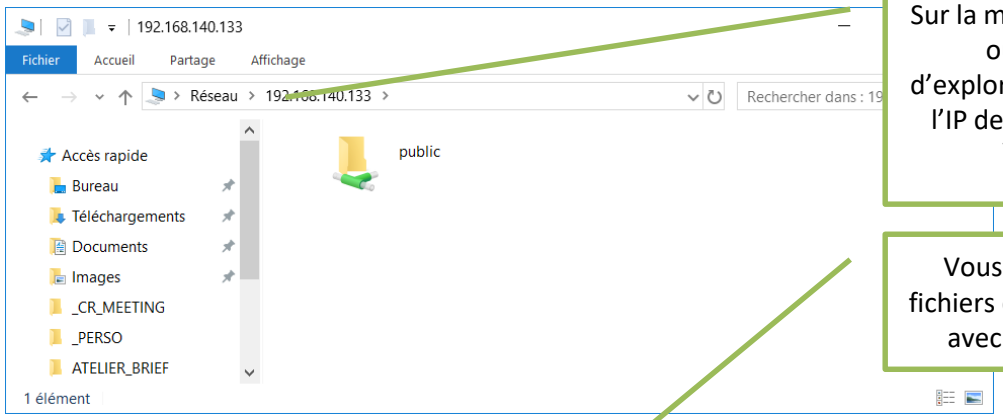


Récupérer l'IP de la machine virtuelle : ip addr show



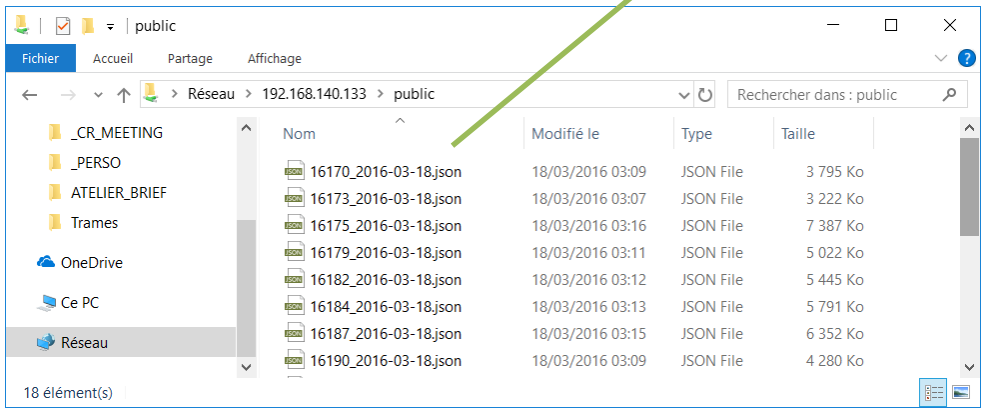
hétic
La grande école de l'Internet

Accéder au Partage

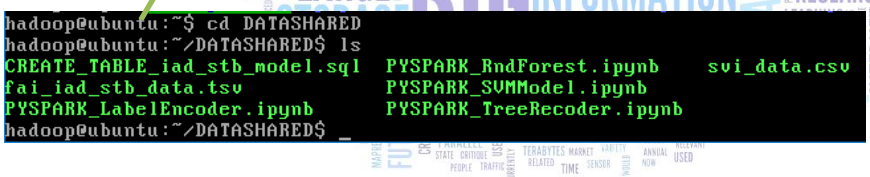


Sur la machine hôte (votre PC), ouvrir une fenêtre d'exploration windows et saisir l'IP de la machine virtuelle : \\xxx.xxx.xx.xxx

Vous pouvez déposer des fichiers dans ce dossier partagé avec la machine virtuelle



Retrouvez vos fichiers sur la VM :
→cd public
→ ls



hétic
La grande école de l'Internet

Lancer les services Hadoop

- Lancer le data file system : start-dfs.sh (pour arrêter le dfs : stop-dfs.sh)

```
hadoop@ubuntu:~$ start-dfs.sh
15/10/25 14:09:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java class
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hadoop/hadoop-2.7.0/logs/hadoop-hadoop-namenode-ubuntu.out
localhost: starting datanode, logging to /home/hadoop/hadoop-2.7.0/logs/hadoop-hadoop-datanode-ubuntu.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/hadoop/hadoop-2.7.0/logs/hadoop-hadoop-secondarynamenode-ubuntu.out
15/10/25 14:09:42 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java class
hadoop@ubuntu:~$
```

- Lancer le ressource manager Yarn : start-yarn.sh (pour arrêter yarn : stop-yarn.sh)

```
hadoop@ubuntu:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-2.7.0/logs/yarn-hadoop-resourcemanager-ubuntu.out
localhost: starting nodemanager, logging to /home/hadoop/hadoop-2.7.0/logs/yarn-hadoop-nodemanager-ubuntu.out
hadoop@ubuntu:~$
```

- Contrôler les processus Java : jps

```
hadoop@ubuntu:~$ jps
2659 SecondaryNameNode
3289 Jps
2505 DataNode
2986 NodeManager
2876 ResourceManager
2398 NameNode
hadoop@ubuntu:~$
```

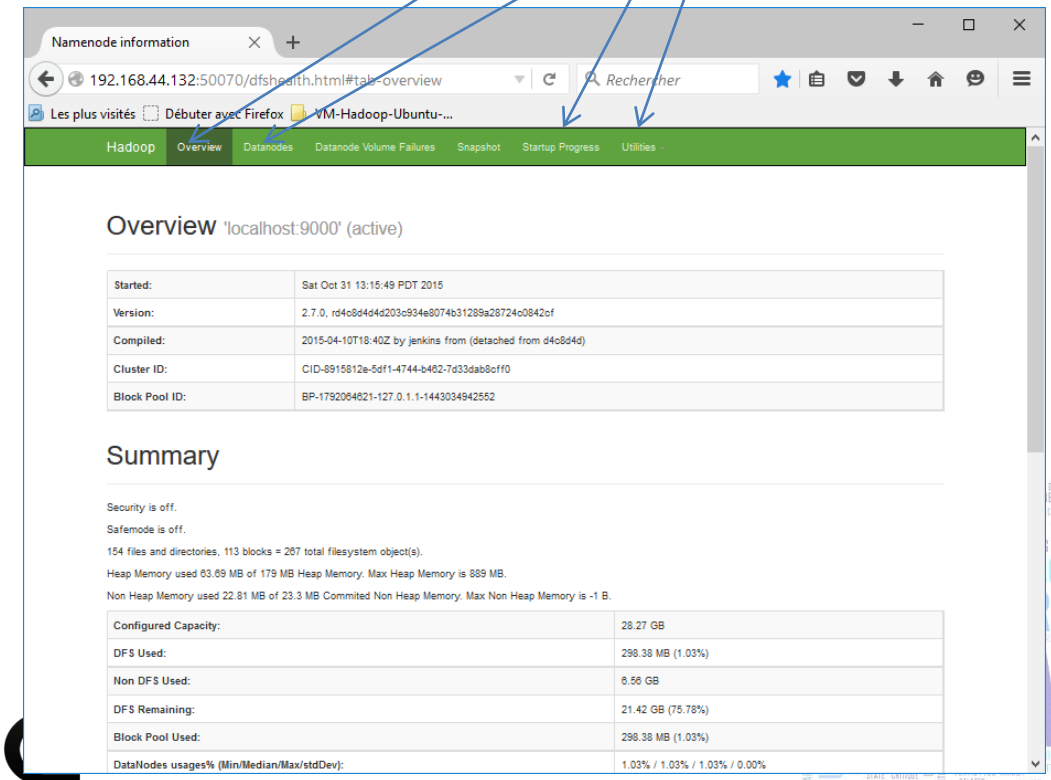
hétic
La grande école de l'Internet

Accéder aux interfaces WEB Hadoop

- HDFS : http:\\xxx.xxx.xx.xxx:50070

Cette interface permet de monitorer le HDFS :

-État des données
-Démarrage des services
-Nombre de nodes actifs
-Parcourir le HDFS

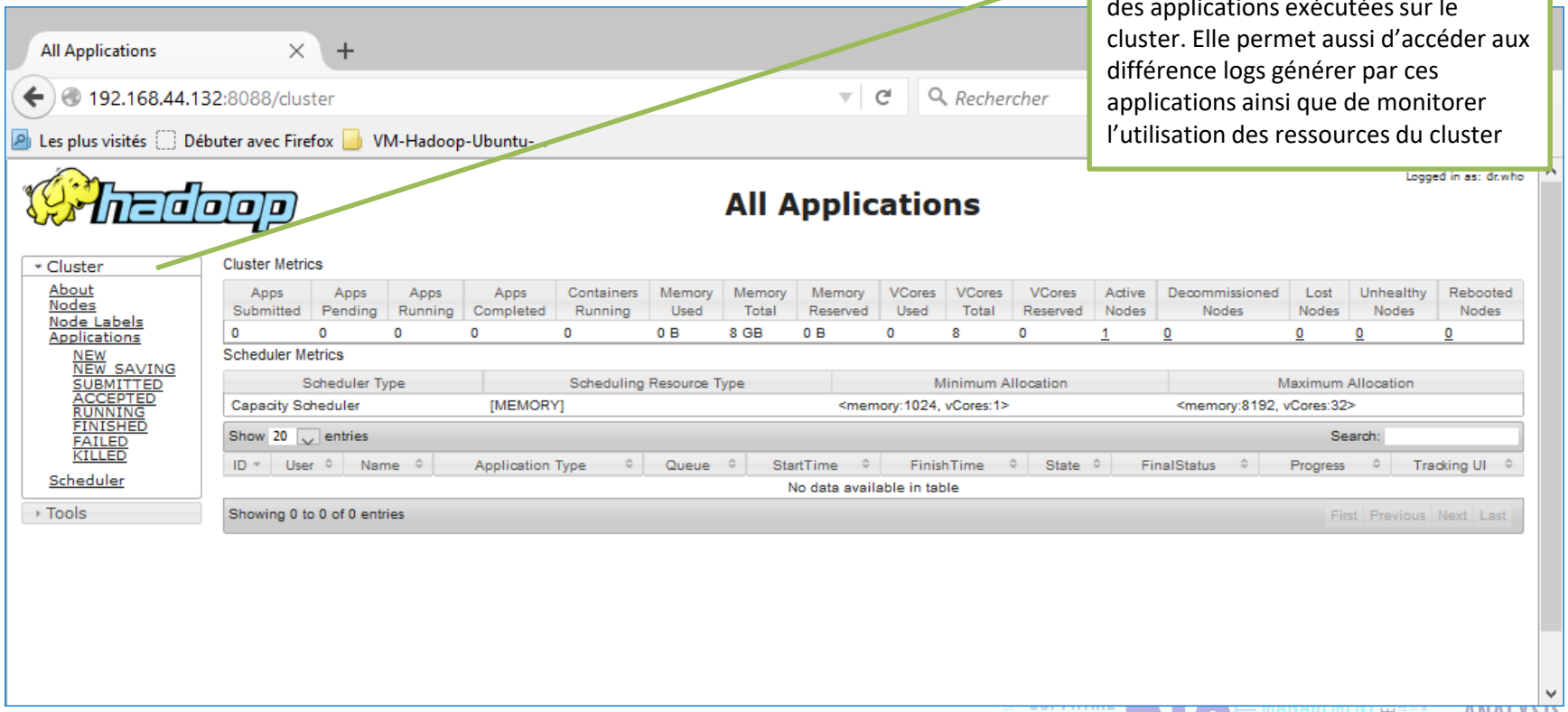


hétic
La grande école de l'Internet

Accéder aux interfaces WEB Hadoop

- Ressource Manager : http:\\xxx.xxx.xx.xxx:8088

Cette interface permet le monitoring des applications exécutées sur le cluster. Elle permet aussi d'accéder aux différence logs générer par ces applications ainsi que de monitorer l'utilisation des ressources du cluster



hétic
La grande école de l'Internet

Jupyter Notebook

- Pour lancer le notebook Jupyter :
  - Se déplacer dans le dossier notebooks : cd notebooks
  - Lancer le notebook jupyter : jupyter notebook

```
hadoop@ubuntu:~/notebooks$ jupyter notebook
[W 11:45:35.445 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. Thi
[I 11:45:35.855 NotebookApp] Serving notebooks from local directory: /home/hadoop/notebooks
[I 11:45:35.856 NotebookApp] 0 active kernels
[I 11:45:35.859 NotebookApp] The Jupyter Notebook is running at:
[I 11:45:35.862 NotebookApp] http://[all ip addresses on your system]:9090/
[I 11:45:35.870 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Le notebook est accessible via votre navigateur web : xxx.xxx.xxx.xxx:9090

  - Le mot de passe du notebook : hadoop

hétic
La grande école de l'Internet