**HADOOP**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ce document expliquera ce qu’est hadoop, à quoi il sert et comment mettre cette environnement en place depuis le système d’exploitation windows.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

Depuis l’essor du BIG DATA. Beaucoup d’outils ont vu le jours. Tous destiné à répondre à un problème, la gestion de données massive. Les sources données nombreuses et nombreux ce sont aujourd’hui qui les gèrent tels que les GAFA. Le problème de beaucoup d’outils de gestion de base de données est que dès que la source de données devient trop grande, les performances sans voit augmenter également. C’est alors que Hadoop voit le jours.

Apache hadoop est un outil permettant de paralléliser les traitements en les stockant sur plusieurs unités de durs. D'une part, il propose un système de stockage distribué via son système de fichier HDFS (Hadoop Distributed File System) et ce dernier offre la possibilité de stocker la donnée en la dupliquant, une grappe de serveurs Hadoop n'a donc pas besoin d'être configuré avec un système RAID qui devient inutile.

Le RAID est un ensemble de techniques de virtualisation du stockage permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit les performances, soit la sécurité ou la tolérance aux pannes de l'ensemble du ou des systèmes.

|  |  |
| --- | --- |
|  | *Le RAID est un ensemble de techniques de virtualisation du stockage permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit les performances, soit la sécurité ou la tolérance aux pannes de l'ensemble du ou des systèmes.*  *Cf. wikipedia* |

D'autre part, Hadoop fournit un système d'analyse des données appelé MapReduce. Ce dernier officie sur le système de fichiers HDFS pour réaliser des traitements sur des gros volumes de données.

Hadoop est beaucoup plus utile dans un environnement multi-machine mais il n'a de sens dans un environnement mono-machine que pour tester la configuration de l'installation ou fournir un environnement de développement MapReduce (prochain article). Comme nous allons le faire avec l’api **SPARKS** dans un prochain tutoriel.

Avant de rentrer dans la vive du sujet, certaines choses ont besoin d’être défini :

* Un namenode : Un namenode est un service qui s’occupera de gérer l’état du système de fichiers. Il stocke l'emplacement de tous les fichiers dans le système de fichiers local.
* Un datanode : Les datanode sont des workers. C’est-à-dire qu’il effectue les opérations demandées par le namenode. Ils vont se charger de transmettre au client les blocs correspondant au fichier à transmettre et en écriture vont retourner l’emplacement des blocs créés.
* HDFS : HDFS (Hadoop Distributed File System) est un système de fichier distribué permettant de stocker et de récupérer des fichiers rapidement. Il s’agit de l’un des composants basiques du framework Hadoop Apache, et plus précisément de son système de stockage.

**Installation**

L’installation d’HADOOP s’effectuera sur le système d’exploitation windows parce que celui-ci permet une installation simple et sans trop de configuration par rapport à linux. Pour pouvoir installer HADOOP, nous allons avoir besoin installer les outils suivants :

* Le JDK (de préférence le 1.8)
* Apache hadoop (dans notre cas la version 2.7.1 pesant 200 MO)
* hadooponwindows-master

Tout cela est disponible via le lien suivant :

<https://github.com/sardetushar/hadooponwindows>

Une fois les différents outils installés puis extrait, nous allons dans un premier temps déclarer nos variables d’environnements :

* JAVA\_HOME (pointant sur le dossier bin du JDK)
* HADOOP\_HOME (pointant vers le bin du dossier hadoop2.7.1 extrait)
* Puis dans le PATH déclarer le chemin vers c’est 2 composants.

Après cela, nous allons devoir remplacer les fichiers contenus dans le dossier /etc et /bin du dossier d’hadoop 2.7.1 par ceux contenu dans le dossier hadooponwindows-master.

Une fois cela réalisé, nous allons procéder à la configuration de 4 fichiers :

* *core-site.xml*
* *hdfs-site.xml*
* *mapred-site.xml*
* *hadoop-env.cmd*

Ainsi, nous allons inscrire les lignes suivants dans les fichiers correspondants.

**core-site.xml :**

<!-- C'est à cette adresse que nous pourrons accéder à notre BDD depuis l’API spark avec JAVA -->

<configuration>

*<property>*

*<name>fs.defaultFS</name>*

*<value>hdfs://localhost:9000</value>*

*</property>*

*</configuration>*

**hdfs-site.xml :**

<!-- Création du namenode et datanode contenu dans un dossier data que nous créerons au préalable -->

*<configuration>*

*<property>*

*<name>dfs.replication</name>*

*<value>1</value>*

*</property>*

*<property>*

*<name>dfs.namenode.name.dir</name>*

*<value>D:\hadoop\hadoop-2.7.1\data\namenode</value>*

*</property>*

*<property>*

*<name>dfs.datanode.data.dir</name>*

*<value>D:\hadoop\hadoop-2.7.1\data\datanode</value>*

*</property>*

*</configuration>*

**mapred-site.xml :**

*<configuration>*

*<property>*

*<name>mapreduce.framework.name</name>*

*<value>yarn</value>*

*</property>*

*</configuration>*

**hadoop-env.cmd:**

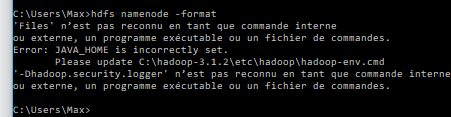
<!-- On déclare notre JAVA\_HOME -->

*set JAVA\_HOME=CheminVersLeDossierJDKSansLeBin*

Une fois que la configuration des différents fichiers est effectuée, nous allons depuis le terminal de command lancer la commande suivante :

***hdfs namenode –format***

Si l’erreur suivante se présente :

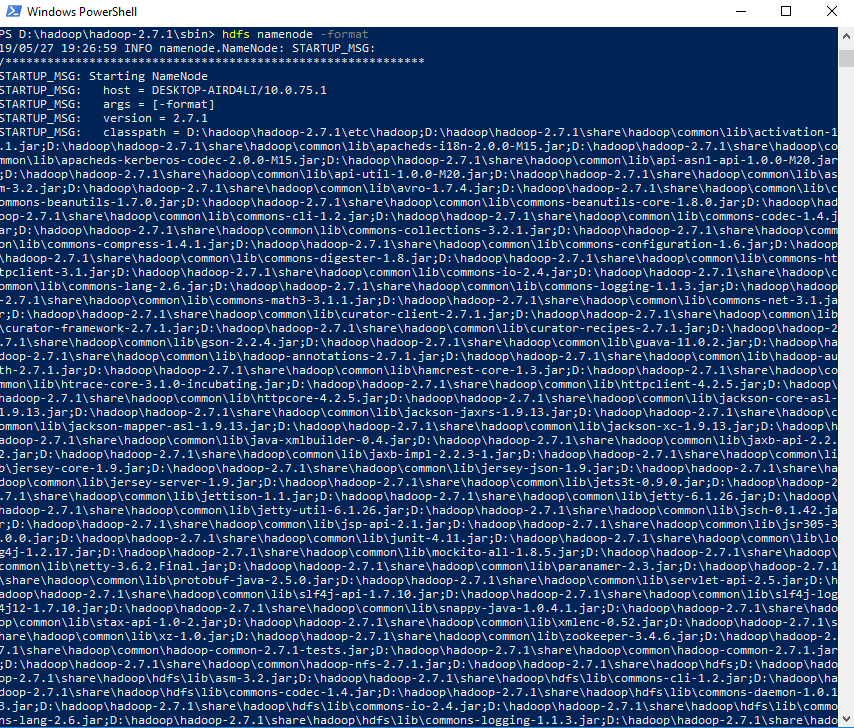


**Annexe 1: Erreur possible suite à l'exécution de la commande *hdfs namenode –format***

Vérifié que le **JAVA\_HOME et HADOOP\_HOME pointe bien vers leur bin respective** et que l’on ne pointe pas vers le dossier bin mais le **dossier racine du jdk** dans hadoop-env.cmd.

|  |  |
| --- | --- |
|  | *A noter que sur windows, lorsque l’on veut spécifier dans le chemin d’accès le dossier “Program Files”, nous écrirons* ***PROGRA~1*** *à la place.* |

Si tout va bien, une suite d’instructions est censée se lancer.

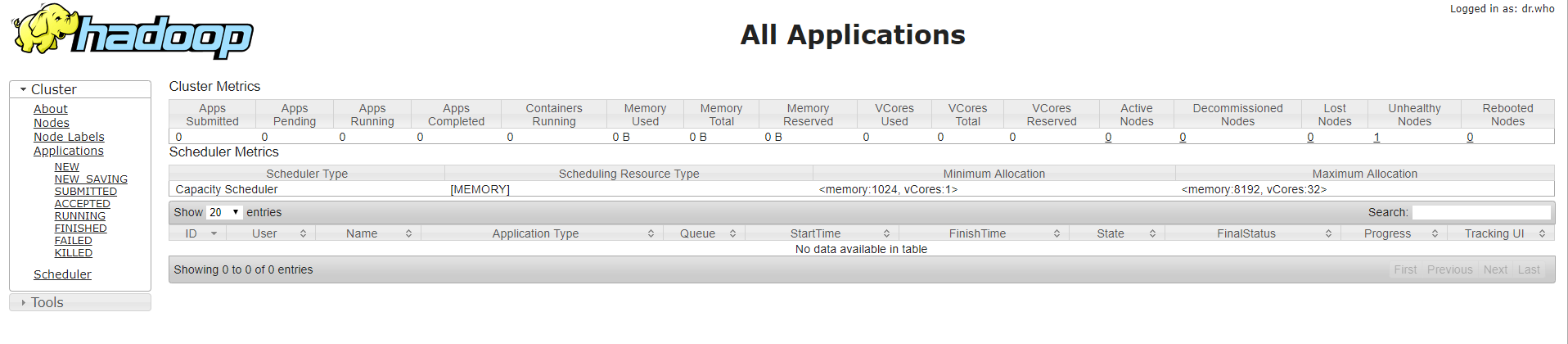


**Annexe 2: Résultat obtenu suite à l'exécution de la commande hdfs namenode –format**

Il nous restera plus qu’à nous rendre dans le dossier **sbin** de hadoop puis de lancer l’exécutable **start-all.cmd**.

4 terminaux sont censés s’ouvrir confirmant le bon fonctionnement de l’outil. Ainsi nous pourront accéder à diverse composant via l’URL de notre navigateur. Si la configuration/port et semblable à celle effectué dans ce document :

* Localhost:8088 :



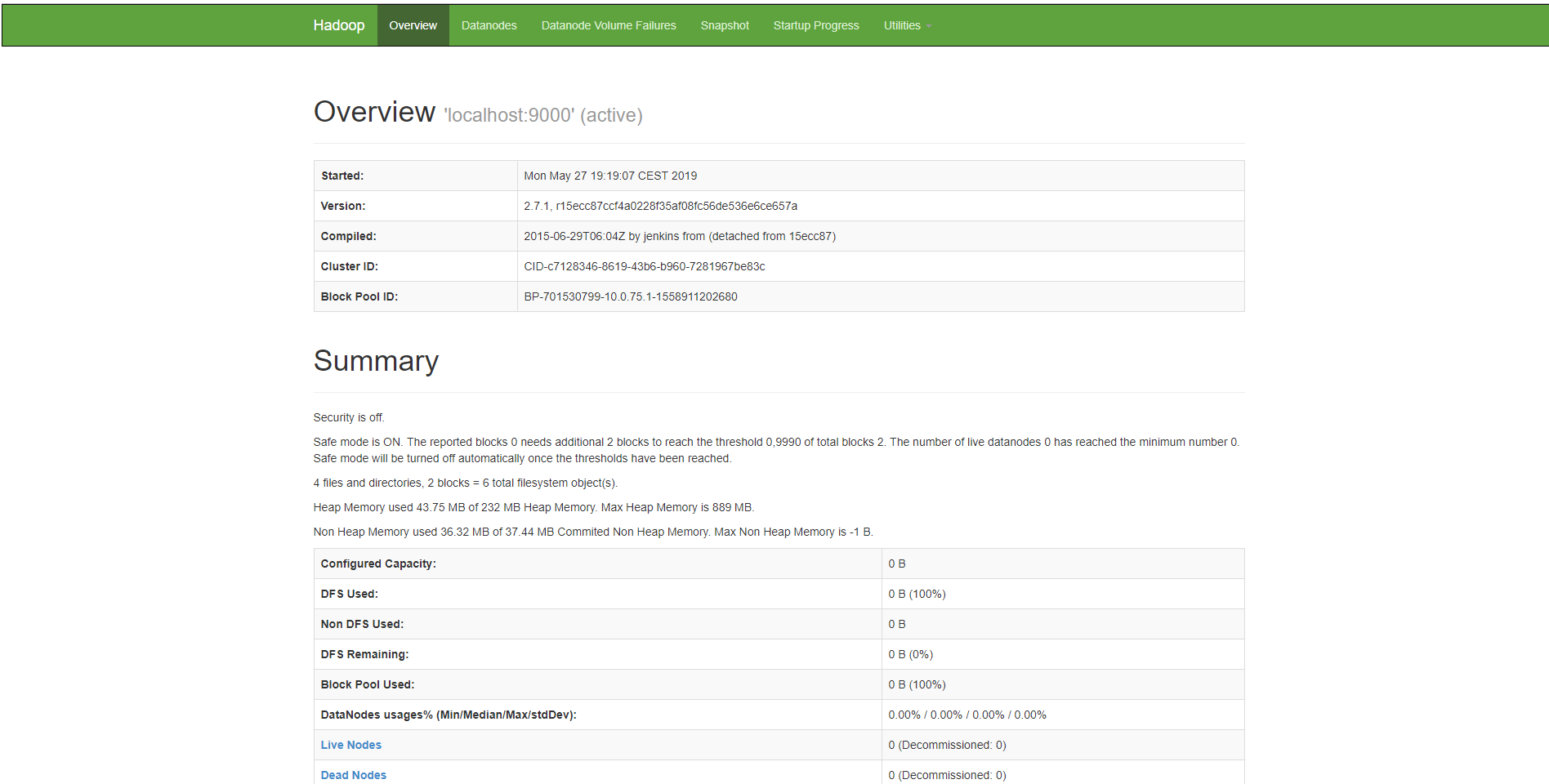
**Annexe 3: Erreur possible suite à l'exécution de la commande *hdfs namenode –format***

* Localhost:9000 : le port permettant d’accéder à notre BDD, soit l’HDFS

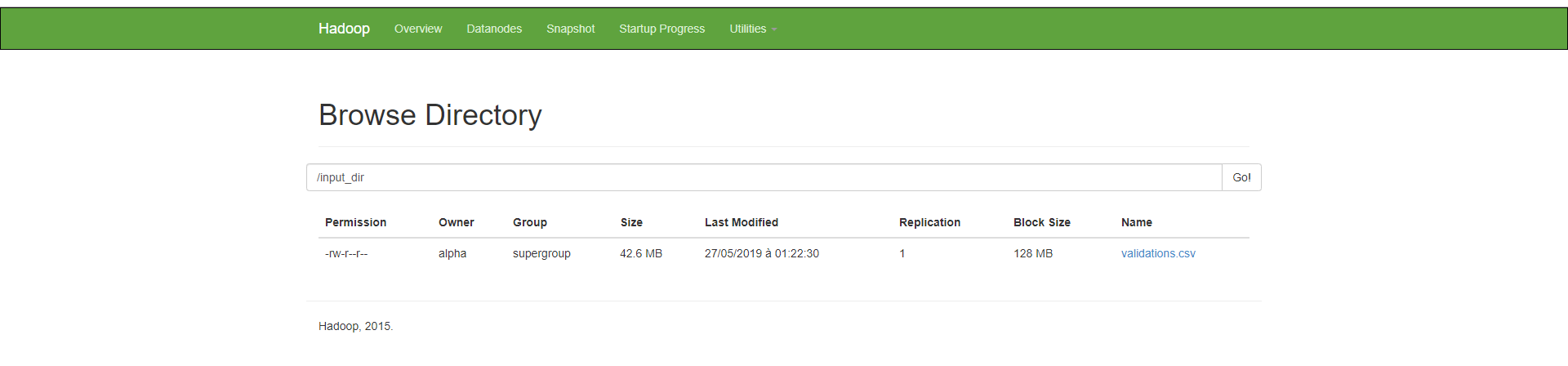


**Annexe 4: Résultat de l’URL**

* Localhost:50070 : l’environnement on l’on stocke notre BDD.



**Annexe 5: Environnement de stockage de la BDD**



**Annexe 6: Fichier .csv contenu dans hadoop**

Voici différentes commande utile quant à la bonne utilisation de hadoop:

* hadoop fs -mkdir /dossierDestinataire : création de dossier.
* hadoop fs -put cheminVersMonFichier /dossierDestinataire : Ajout d’un fichier (txt, csv etc..) dans notre base.