

# TP1 - Installation Hadoop

## 1. Installation de Hadoop

Vous allez installer Hadoop en local sur votre machine.

Téléchargez et décompressez hadoop-2.7.3

<https://dist.apache.org/repos/dist/release/hadoop/common/hadoop-2.7.3/>

## 2. Configuration de Hadoop

Nous allons utiliser Hadoop sur 1 seule machine locale

## 3. Variables d'environnement

Dans votre fichier `~/.bashrc` ajoutez les lignes suivantes :

```
export JAVA_HOME = <Repertoire Java>/usr/lib/jvm/java-7-  
openjdk-i386  
export HADOOP_HOME =Repertoire Hadoop  
export PATH = $HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

La commande `source ~/.bashrc` vous permet de lire les variables qui y sont décrites.

Vérifiez que la commande `bin/hadoop` vous affiche une aide sur cette commande.

## 4. Lancement de HDFS

On démarre les démons (NameNode et DataNode)

```
start-dfs.sh
```

En cas de problème avec HDFS, il faut arrêter le serveur hdfs avec la commande `stop-dfs.sh`

En cas de problème d'autorisation avec le port ssh, il faut installer open ssh avec la commande :

```
sudo apt-get install openssh-server openssh-client
```

sur votre navigateur web testez, pour accéder au NameNode :

<http://localhost:50070/>

## 5. Lancement de YARN

YARN est le nom donné à l'architecture MapReduce v2. Son fonctionnement sépare d'une part le gestionnaire de ressources et d'autre part l'ordonnanceur de jobs.

Les commandes pour démarrer/arrêter YARN sont:

```
start-yarn.sh
```

```
stop-yarn.sh
```

sur votre navigateur web testez, <http://localhost:8088/>

## 6. Sources

[http://www.science.smith.edu/dftwiki/index.php/Hadoop\\_Tutorial\\_1\\_-\\_Running\\_WordCount](http://www.science.smith.edu/dftwiki/index.php/Hadoop_Tutorial_1_-_Running_WordCount)

[http://www.fil.univ-lille1.fr/~caronc/SGBD/TP\\_Hadoop.html](http://www.fil.univ-lille1.fr/~caronc/SGBD/TP_Hadoop.html)

<https://tecadmin.net/hadoop-running-a-wordcount-mapreduce-example/>

## TP 2 - Exercices

### 1. Exécution de WordCount

Le programme que nous allons tester permet de compter le nombre d'occurrences de mots dans des textes.

Le programme [WordCount.java](#) est disponible sur moodle.

### 2. Recherche d'anagrammes

Ecrire un programme MapReduce qui détermine les anagrammes des mots lus dans un dictionnaire. Comme fichier dictionnaire on utilisera le fichier des mots de la langue anglaise [dico.txt](#).

On veut tous les mots en résultats, même ceux qui n'ont pas d'anagramme.

On ne conserve que les mots du dictionnaire qui ont un autre mot comme anagramme.

### 3. Min, Max, Count

Ecrire un programme MapReduce qui à partir du fichier [bank-data.csv](#) calcule pour chaque âge le salaire minimum, maximum et le nombre de personnes de cet âge.