# TP1 BIG DATA

# Partie 1

## 1. Création des machines

Créer une machine virtuelle par poste de travail physique, ce seront les nœuds de notre cluster.

## 2. Configurer les ip statiques

Éditer le fichier */etc/dhcp/dhclient.conf*  pour configurer une ip statique.



Et répétez l’opération pour les 2 autres nœuds.

## 3. définition des noms d’hôtes

Pour utiliser le cluster plus facilement, nous allons définir des noms d’hôtes pour les machines. Éditer les fichiers */etc/hosts*

Une image contenant texte

Description générée automatiquement

Éditer le fichier */etc/hostname*

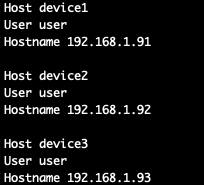


A présent votre console doit ressembler à ça :



## 4. Simplification SSH

Pour passer d’une machine à l’autre, il va être gênant de toujours taper une série de commandes. Éditez donc le fichier *.ssh/config* pour y faire figurer les informations de connexion des autres machines



Après cette manipulation, la connexion à un appareil du cluster depuis un autre doit pouvoir se faire comme suit :



Il faut tout de même entrer le mot de passe, ce qui peut être simplifié en configurant les paires de clés publiques/privées

Sur chaque appareil, exécutez :



Cela va générer une paire de clé publique/privée dans le repertoire *.ssh*, cela permettra d’utiliser SSH en toute sécurité sans entrer de mot de passe. L’un de ces fichiers s’appellera id\_ed25519, il s’agit de votre clé privée, l’autre, id\_ed25519 sera la clé publique. Aucune phrase secrète n’est nécessaire. La clé privée ne doit jamais quitter la machine sur laquelle elle a été générée.

Chaque clé publique devra être ajouter au fichier *.ssh/authorized\_keys* sur les autres appareils.  
  
Supposons que le device 1 contiendra l’enregistrement « master » et qu’on copiera ensuite sur les autres machines.

Executez la commande suivante sur les appareil 2 et 3:



Cela va concaténer le fichier de clé publique du device 2 à la liste des clés autorisée du device 1, donnant ainsi la permission au device 2 de se connecter en ssh au device 1 sans mot de passe. Il faut répéter l’opération pour chaque machine.

Il faut aussi le faire pour le device 1 afin que toutes les machines puissent se connecter au device 1



## 5. répliquer la configuration SSH

Pour répliquer la configuration sur tous les appareils, copiez simplement les deux fichiers *authorized keys* et config en utilisant la commande scp :



## 6. Faciliter les actions groupées

Modifiez le fichier *~/.bashrc* et ajouter les commandes ci-dessous. Ne pas oublier de sourcer le fichier pour pouvoir utiliser les modifications

### Obtenir le nom d’hôte de tous les autres appareils

Une image contenant texte

Description générée automatiquement

### Envoyer la même commande à toutes les machines

Une image contenant texte

Description générée automatiquement

### Envoyer le même fichier à toutes les machines

Une image contenant texte

Description générée automatiquement

Une fois que ces modifications sont effectuées, sourcez et envoyez le fichier à tous les appareils du cluster.



# TP BIG DATA

# Partie 2

## Installation d’hadoop

Avant d’installer hadoop, il faut vérifier que les machines ont une version acceptable de Java, nous allons créer une configuration à noeud unique sur le nœud maitre (device1).  
Commencer par télécharger hadoop





Supprimer maintenant le fichier archive téléchargé, il n’est plus utile.

Renommer le dossier hadoop



Modifier les autorisations sur le répertoire



Modifier le fichier bashrc

Une image contenant texte

Description générée automatiquement

Éditer le /opt/hadoop/etc/hadoop/hadoop-env.sh

Et ajouter export JAVA\_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")

Vérifier qu’hadoop s’est installé correctement



## Installation de spark

Télécharger spark-hadoop de la même manière



Décompresser le fichier dans le répertoire /opt



Renommer le dossier spark



Changer les permissions :



Ajouter le dossier spark au path

Une image contenant texte

Description générée automatiquement

On vérifie que spark a été correctement installé en vérifiant la version :



Une image contenant texte

Description générée automatiquement

HDFS (Hadoop File Distributed System)

Pour que le système de partage de fichier soit opérationnel, il faut modifier certains fichiers de configuration, ces fichiers se trouvent dans le dossier */opt/hadoop/etc/hadoop*

Ouvrir avec nano le fichier *core-site.xml* et ajouter la configuration suivante en modifiant éventuellement avec les noms que vous avez choisis dans les configurations précédentes

Une image contenant texte

Description générée automatiquement

Modifier le fichier *hdfs-site.xml* comme suit :

Une image contenant texte

Description générée automatiquement

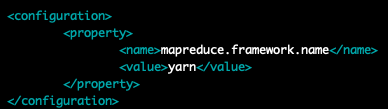
Créer les repertoire datanode et namenode dans un dossier hadoop\_tmp



Changer les droits



Modifier à présent le fichier *mapred-site.xml*



Modifier le fichier *yarn-site.xml*

Une image contenant texte

Description générée automatiquement

Une fois les fichiers modifiés, nous pouvons formater le système de fichiers :



Démarrer le HDFS avec les commandes suivantes :





Vérifier le fonctionnement en créant un répertoire temporaire :



Voir les répertoires du HDFS :



Voir les processus java :

Une image contenant texte

Description générée automatiquement

Notre HDFS est maintenant opérationnel sur le device 1

Pour vérifier que spark et hadoop fonctionnent ensemble, exécuter la commande suivante pour ajouter un fichier au hdfs :



Ensuite, ouvrir le spark-shell avec l’invite scala :



Une image contenant texte

Description générée automatiquement

A présent nous avons un cluster avec un nœud unique qui est à la fois le nœud maitre et le nœud de travail.

Nous allons à présenter créer les répertoires requis sur les autres machines à l’aide des commandes configurées dans les étapes précédentes

Une image contenant texte

Description générée automatiquement

A présent, copier les configurations dans */opt/hadoop* en utilisant *rsync*:



Vous pouvez vérifier le fonctionnement en demandant la version d’hadoop dans le cluster.

Pour que hadoop s’exécute sur le cluster, il faut à présent modifier les fichiers de configuration que nous avons déjà modifié.

*Mapred-site.xml*

Une image contenant texte

Description générée automatiquement

*Yarn-site.xml*

Une image contenant texte

Description générée automatiquement

Nettoyer les anciens fichiers des machines



Il faut indiquer à Hadoop quel machine utiliser comme nœuds de travail et comme nœud maitre

Créer un fichier *master* dans le répertoire */opt/hadoop/etc/hadoop*

Ajouter une ligne *device1*

Créer un fichier workers et y ajouter les deux autres nœuds

Éditer le fichier */etc/hosts* et supprimer (ou commenter) la ligne *127.0.1.1 DeviceX*

Puis copier le fichier sur le reste du cluster.

Redémarrer le cluster



Démarrer le HDFS avec les commandes suivantes :

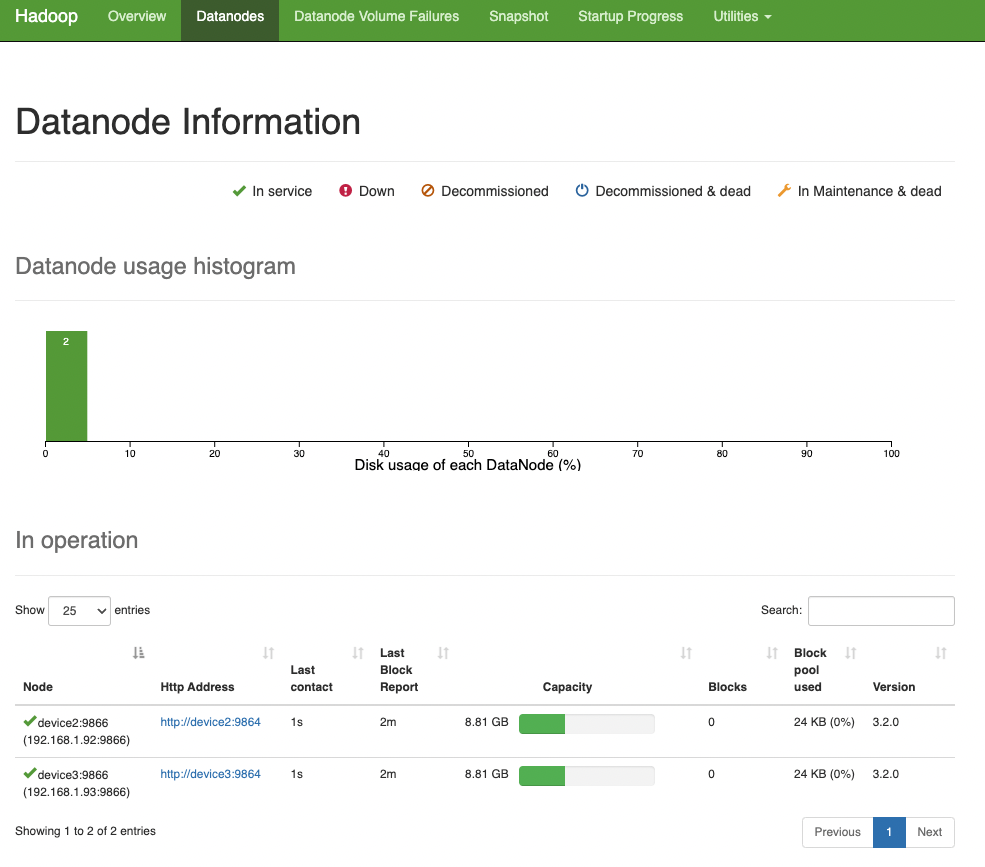


Nous pouvons maintenant tester le cluster en envoyant des fichiers dans HDFS à partir de l’importe quel machine (avec *hadoop fs -put*).

Vérifier que le cluster est bien opérationnel en allant avec votre navigateur à l’adresse : [*http://ip\_du\_master:9870/*](http://ip_du_master:9870/)

Une image contenant texte

Description générée automatiquement



## Configuration de SPARK sur le cluster

Pour que *spark* puisse communiquer avec YARN, il faut rajouter des variables d’environnement dans le fichier *.bashrc* du device1. Pour le moment nous avons :



Il faut à présent ajouter :



HADOOP\_CONF\_DIF est le répertoire qui contient tous les fichiers de configuration xml que nous avons modifié précédemment

Créer le fichier de configuration spark



Ajouter une configuration :

Une image contenant texte

Description générée automatiquement

Quelques commandes HADOOP :

