

Bases de données NOSQL et Big Data

TP1 : initiation à Hadoop

Partie 1 : Installation et configuration

Pour déployer le framework Hadoop, nous allons utiliser des conteneurs Docker [<https://www.docker.com/>]. L'utilisation des conteneurs va garantir la consistance entre les environnements de développement et permettra de réduire considérablement la complexité de configuration des machines (dans le cas d'un accès natif) ainsi que la lourdeur d'exécution (si on opte pour l'utilisation d'une machine virtuelle).

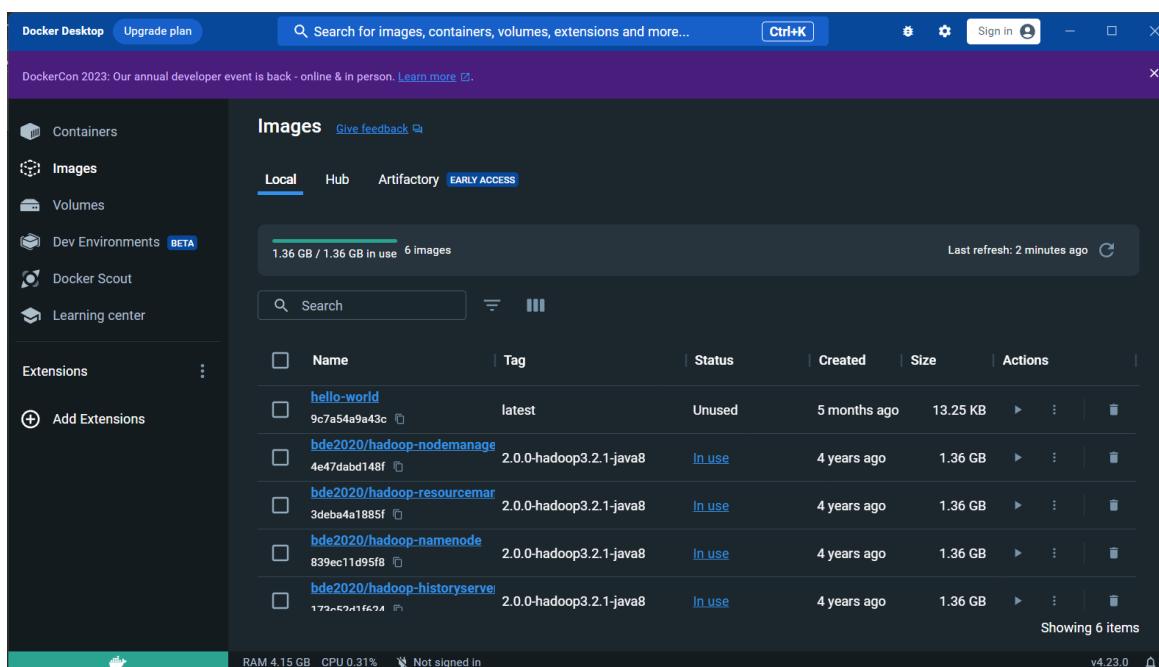
1. Cloner l'image du cluster Hadoop : Pour commencer, vous devez cloner l'image Docker du cluster Hadoop, disponible à l'adresse suivante : <https://github.com/big-data-europe/docker-hadoop>

```
git clone https://github.com/big-data-europe/docker-hadoop
```

2. Utiliser Docker Compose: Une fois l'image du cluster Hadoop clonée, vous pouvez utiliser Docker Compose pour simplifier le déploiement. Il vous suffit de lancer la commande suivante dans votre terminal :

```
docker-compose up -d
```

3. Démarrer le cluster Hadoop : Après avoir exécuté la commande précédente, tous les conteneurs du cluster Hadoop seront démarrés automatiquement. Cela inclut les composants essentiels tels que le namenode, le datanode, le resourcemanager, le nodemanager, et le historyserver.



Partie 2: Commande de HDFS

Connectez-vous au container du **namenode** en utilisant la commande Docker suivante.

docker exec -it namenode hash

Le résultat de cette exécution sera le suivant:

```
root@7ea027891999-
```

Créer le fichier "bonjour.txt" en utilisant la commande suivante:

```
echo "Bonjour Hadoop et HDFS">bonjour.txt
```

Toutes les commandes interagissant avec le système Hadoop commencent par **hdfsdfs** ou **hadoop fs** (**hdfs dfs** est utilisé avec les versions Hadoop les plus récentes). Ensuite, les options rajoutées sont très largement inspirées des commandes Unix standard.

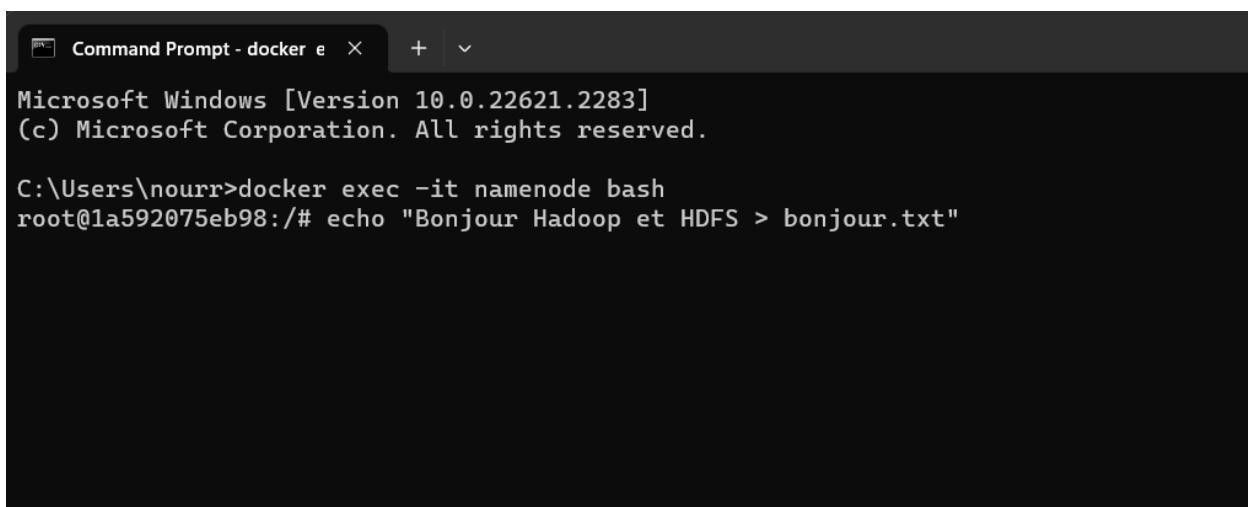
Dans ce qui suit, vous allez manipuler deux systèmes de fichiers.

-les fichiers et dossiers disponibles sur le disque local du conteneur **namenode**, visibles avec **ls**.

-les fichiers et dossiers HDFS, visibles en faisant **hdfsdfs-ls**.

Voici quelques commandes à essayer rapidement :

hdfs dfs -ls affiche ce qu'il y a à la racine HDFS. Vous pouvez descendre inspecter les dossiers que vous voyez. Exemple **hdfs dfs ls /user**. Il n'y a pas de commande équivalente à **cd**, parce qu'il n'y a pas de notion de dossier courant dans HDFS, donc à chaque fois, il faut remettre le chemin complet. C'est une habitude à prendre



```
Command Prompt - docker e  X  +  ▾
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nourr>docker exec -it namenode bash
root@1a592075eb98:/# echo "Bonjour Hadoop et HDFS" > bonjour.txt
```

```
root@1a592075eb98:/# echo "Bonjour Hadoop et HDFS" > bonjour.txt
root@1a592075eb98:/# ls
KEYS  bonjour.txt  dev  etc  hadoop-data  lib  media  opt  root  run.sh  srv  tmp  var
bin  boot  entrypoint.sh  hadoop  home  lib64  mnt  proc  run  sbin  sys  usr
  121  500075  120  115  150  100  100  100  100  100  100  100  100  100
```

```
root@1a592075eb98:/# hdfs dfs -ls /
Found 1 items
drwxr-xr-x  - root supergroup          0 2023-10-03 21:35 /rmstate
root@1a592075eb98:/# |
```

hdfs dfs -ls-R-h/var: affiche les fichiers des sous-dossiers, avec une taille arrondie en Ko, Mo ou Go (l'option -h-human units existe aussi sur Unix).

```
root@1a592075eb98:/# hdfs dfs -ls -R -h /
drwxr-xr-x  - root supergroup          0 2023-10-03 21:35 /rmstate
drwxr-xr-x  - root supergroup          0 2023-10-10 14:01 /rmstate/FSRMStateRoot
drwxr-xr-x  - root supergroup          0 2023-10-03 21:35 /rmstate/FSRMStateRoot/AMRMTokenSecretManagerRoot
-rw-r--r--  3 root supergroup          17 2023-10-03 21:35 /rmstate/FSRMStateRoot/AMRMTokenSecretManagerRoot/AMRMTokenSecretManagerNode
-rw-r--r--  3 root supergroup          2 2023-10-10 14:01 /rmstate/FSRMStateRoot/EpochNode
drwxr-xr-x  - root supergroup          0 2023-10-03 21:35 /rmstate/FSRMStateRoot/RMAppRoot
drwxr-xr-x  - root supergroup          0 2023-10-10 14:01 /rmstate/FSRMStateRoot/RMDTSecretManagerRoot
-rw-r--r--  3 root supergroup          17 2023-10-03 21:35 /rmstate/FSRMStateRoot/RMDTSecretManagerRoot/DelegationKey_1
-rw-r--r--  3 root supergroup          17 2023-10-03 21:35 /rmstate/FSRMStateRoot/RMDTSecretManagerRoot/DelegationKey_2
-rw-r--r--  3 root supergroup          17 2023-10-10 14:01 /rmstate/FSRMStateRoot/RMDTSecretManagerRoot/DelegationKey_3
-rw-r--r--  3 root supergroup          17 2023-10-10 14:01 /rmstate/FSRMStateRoot/RMDTSecretManagerRoot/DelegationKey_4
-rw-r--r--  3 root supergroup          4 2023-10-03 21:35 /rmstate/FSRMStateRoot/RMVersionNode
drwxr-xr-x  - root supergroup          0 2023-10-03 21:35 /rmstate/FSRMStateRoot/ReservationSystemRoot
```

hdfs dfs-mkdir dossier: crée un dossier dans votre espace HDFS. Notez que la taille d'un dossier sera toujours 0.

```
root@1a592075eb98:/# hdfs dfs -mkdir -p dossier
root@1a592075eb98:/# |
```

Reprenez le fichier appelé "bonjour.txt", vérifiez qu'il n'est pas vide.

- Copiez ce fichier sur HDFS par : **hdfs dis-put bonjour.txt**. Utilisez **hdfs dfs -ls-R** pour vérifier.

```
bin  boot  entrypoint.sh  hadoop  home      lib64  mnt      proc  run  sbin      sys  usr
root@1a592075eb98:/# hdfs dfs -put /bonjour.txt
2023-10-10 14:28:23,358 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@1a592075eb98:/# |
```

```
2023-10-10 14:28:23,358 INFO sasl.SaslDataTransferClient: SASL encryption tr
root@1a592075eb98:/# hdfs dfs -ls -R
-rw-r--r--  3 root supergroup          23 2023-10-10 14:28 bonjour.txt
drwxr-xr-x  - root supergroup          0 2023-10-10 14:18 dossier
root@1a592075eb98:/# |
```

hdfs dfs -cat bonjour.txt: affiche le contenu. Il n'y a pas de commande more mais vous pouvez faire **hdfs dfs -cat bonjour.txt | more**

```
root@1a592075eb98:/# hdfs dfs -cat bonjour.txt | more
2023-10-10 14:36:33,917 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Bonjour Hadoop et HDFS
root@1a592075eb98:/#
```

hdfs dfs-tail bonjour.txt: affiche le dernier Ko du fichier

```
root@1a592075eb98:/# hdfs dfs -tail bonjour.txt
2023-10-10 14:38:47,130 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Bonjour Hadoop et HDFS
root@1a592075eb98:/#
```

Supprimez ce fichier de HDFS par **hdfs dfs -rm bonjour.txt**.

```
Bonjour Hadoop et HDFS
root@1a592075eb98:/# hdfs dfs -rm bonjour.txt
Deleted bonjour.txt
root@1a592075eb98:/#
```

Remettez à nouveau ce fichier par **hdfs dfs -copyFromLocal bonjour.txt** (vérifier avec **hdfs dfs -ls**). Cette commande est similaire à **hdfs dfs -put**

```
root@1a592075eb98:/# hdfs dfs -copyFromLocal bonjour.txt
2023-10-10 14:41:40,471 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@1a592075eb98:/#
```

vérifier avec **hdfs dfs -ls**

```
root@1a592075eb98:/# hdfs dfs -ls
Found 2 items
-rw-r--r--  3 root supergroup          23 2023-10-10 14:41 bonjour.txt
drwxr-xr-x  - root supergroup          0 2023-10-10 14:18 dossier
root@1a592075eb98:/#
```

hdfs dfs -chmod go+w bonjour.txt (vérifiez son propriétaire, son groupe et ses droits avec **hdfs dfs -ls**)

```
root@1a592075eb98:/# hdfs dfs -chmod go+w bonjour.txt
root@1a592075eb98:/# |
```

vérifiez son propriétaire, son groupe et ses droits avec **hdfs dfs -ls**

hdfs dfs-chmod go-r bonjour.txt (vérifiez les droits)

hdfs dfs -mv bonjour.txt dossier/bonjour.txt (vérifiez avec **hdfs dfs -ls-R**)

hdfs dfs -get dossier/bonjour.txt bien.txt: transfère le fichier de HDFS vers le local en lui changeant son nom. Cette commande ne serait pas à faire avec de vraies mégadonnées!

```
drwxr-xr-x - root supergroup 0 2023-10-10 14:18 dossier
root@1a592075eb98:/# hdfs dfs -chmod go+w bonjour.txt
root@1a592075eb98:/# hdfs dfs -chmod go-r bonjour.txt
root@1a592075eb98:/# hdfs dfs -mv bonjour.txt dossier/bonjour.txt
root@1a592075eb98:/# hdfs dfs -get dossier/bonjour.txt bien.txt
2023-10-10 14:50:20,242 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@1a592075eb98:/# |
```

hdfs dfs-cp dossier/bonjour.txt dossier/bien.txt (vérifiez)

```
root@1a592075eb98:/# hdfs dfs -cp dossier/bonjour.txt dossier/bien.txt
2023-10-10 14:51:54,102 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-10-10 14:51:54,205 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@1a592075eb98:/# |
```

hdfs dfs -count -h /dossier affiche le nombre de sous-dossiers, fichiers et octets occupés dans/dossier. Cette commande correspond à peu près à du -h dans Unix.

```
root@6e8a12d8edc0:/# hdfs dfs -count -h /dossier
      1          2          46 /dossier
root@6e8a12d8edc0:/# |
```

hdfs dfs -rm dossier/bonjour.txt (vérifiez avec hdfs dfs -ls dossier).

```
root@6e8a12d8edc0:/# hdfs dfs -ls /dossier
Found 1 items
-rw--w--w- 3 root supergroup          23 2023-10-21 06:08 /dossier/bonjour.txt
root@6e8a12d8edc0:/# hdfs dfs -rm /dossier/bonjour.txt
Deleted /dossier/bonjour.txt
root@6e8a12d8edc0:/# hdfs dfs -ls /dossier
root@6e8a12d8edc0:/# |
```

hdfs dfs -rm -r dossier (vérifiez avec hdfs dfs -ls).

```
root@6e8a12d8edc0:/# hdfs dfs -rm -r /dossier
Deleted /dossier
root@6e8a12d8edc0:/# hdfs dfs -ls
drwxr-xr-x  - root supergroup          0 2023-10-21 05:57 testFolder
root@6e8a12d8edc0:/# |
```

Partie 3: Création d'une arborescence et téléchargement de fichier

1. Créez une structure de répertoires sur HDFS avec le dossier parent "TPs" et les sous-dossiers "data" et "codes". Utilisez les commandes HDFS nécessaires pour réaliser cela.

```
root@6e8a12d8edc0:/# hdfs dfs -mkdir /TPs
root@6e8a12d8edc0:/# hdfs dfs -mkdir /TPs/data
root@6e8a12d8edc0:/# hdfs dfs -mkdir /TPs/codes
root@6e8a12d8edc0:/# hdfs dfs -ls /TPs
Found 2 items
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/codes
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/data
root@6e8a12d8edc0:/# |
```

2. Télécharger le fichier **purchases.txt** à partir de <https://bitly.ws/WTSK> sur votre machine hôte et le copier vers le container **namenode** puis dans le dossier "data" en utilisant la commande docker cp dont la syntaxe est la suivante :

`docker cp container:source_path destination_path (pour copier à partir d'un container)`

`docker cp source_path container:destination_path (pour copier vers un container)`

`docker cp « path » namenode:/`

`docker exec -it namenode bash`

`hdfs dfs -put purchases.txt.gz /TPs/data`

```
root@6e8a12d8edc0:/# hdfs dfs -put purchases.txt.gz /TPs/data
2023-10-22 06:37:48,035 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@6e8a12d8edc0:/# hdfs dfs -ls -R /TPs
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/codes
drwxr-xr-x  - root supergroup          0 2023-10-22 06:37 /TPs/data
-rw-r--r--  3 root supergroup  38454568 2023-10-22 06:37 /TPs/data/purchases.txt.gz
root@6e8a12d8edc0:/#
```

3. Depuis votre machine hôte, téléchargez le fichier «**pg4300.txt**» du roman «**Ulysse**» disponible sur <https://bitly.ws/WTKB> directement sur le noeud namenode en utilisant la commande docker exec. Placer ensuite ce fichier sous le dossier data de HDFS.

```
root@6e8a12d8edc0:/# hdfs dfs -put pg4300.txt /TPs/data
root@6e8a12d8edc0:/# hdfs dfs -ls -R /TPs
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/codes
drwxr-xr-x  - root supergroup          0 2023-10-22 06:43 /TPs/data
-rw-r--r--  3 root supergroup          0 2023-10-22 06:43 /TPs/data/pg4300.txt
-rw-r--r--  3 root supergroup  38454568 2023-10-22 06:37 /TPs/data/purchases.txt.gz
root@6e8a12d8edc0:/#
```

4. Depuis le container **namenode**, téléchargez le fichier «**pg135.txt**» du roman «**Les misérables**» disponible à l'URL <https://bitly.ws/WTKU>. Placer ensuite ce fichier sous le dossier data deHDFS.

```
root@6e8a12d8edc0:/# hdfs dfs -put pg135.txt /TPs/data
root@6e8a12d8edc0:/# hdfs dfs -ls -R /TPs
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/codes
drwxr-xr-x  - root supergroup          0 2023-10-22 06:45 /TPs/data
-rw-r--r--  3 root supergroup          0 2023-10-22 06:45 /TPs/data/pg135.txt
-rw-r--r--  3 root supergroup          0 2023-10-22 06:43 /TPs/data/pg4300.txt
-rw-r--r--  3 root supergroup  38454568 2023-10-22 06:37 /TPs/data/purchases.txt.gz
root@6e8a12d8edc0:/# |
```

5. Utilisez les commandes HDFS pour lister le contenu du répertoire "data" sur HDFS, ainsi que pour afficher les informations sur les fichiers téléchargés.

```
root@6e8a12d8edc0:/# hdfs dfs -ls -R /TPs
drwxr-xr-x  - root supergroup          0 2023-10-22 05:59 /TPs/codes
drwxr-xr-x  - root supergroup          0 2023-10-22 06:45 /TPs/data
-rw-r--r--  3 root supergroup          0 2023-10-22 06:45 /TPs/data/pg135.txt
-rw-r--r--  3 root supergroup          0 2023-10-22 06:43 /TPs/data/pg4300.txt
-rw-r--r--  3 root supergroup  38454568 2023-10-22 06:37 /TPs/data/purchases.txt.gz
root@6e8a12d8edc0:/# hdfs dfs -ls -R /TPs/data
-rw-r--r--  3 root supergroup          0 2023-10-22 06:45 /TPs/data/pg135.txt
-rw-r--r--  3 root supergroup          0 2023-10-22 06:43 /TPs/data/pg4300.txt
-rw-r--r--  3 root supergroup  38454568 2023-10-22 06:37 /TPs/data/purchases.txt.gz
```

Partie 3: Etat du cluster

Les services Hadoop génèrent des pages web automatiquement pour permettre de suivre fonctionnement. Cliquez sur ce lien : <http://localhost:9870/explorer.html#/> pour vous y connecter.

La page que vous voyez propose plusieurs liens vers différents services Hadoop.

Page Overview: Dans le tableau Summary vous avez l'espace total, l'espace utilisé, l'espace libre, Avec des valeurs valeurs en %, des liens vers les DataNodes vivants, morts ou en train de se désactiver (decommissioning nodes).

Page Datanodes: Vous voyez la capacité et la charge de chaque DataNode. La colonne Last Contact indique le nombre de secondes depuis le précédent << battement de cœur >> (heartbeat) le Datanode au Namenode, c'est le terme employé pour un signal périodique de bon fonctionnement. Il y a un contact toutes les 3 secondes. Un Datanode est considéré comme mort lorsqu'il n'a pas donné signe de vie depuis 10 minutes. Vous pouvez rafraîchir pour voir l'évolution.

Page Utilities: un item Browse the file system vous permet de parcourir l'arbre des fichiers HDFS,. En cliquant sur le nom d'un fichier, vous aurez des informations sur les blocs et les machines contenant ce fichier. Ne cliquez sur Download que pour un petit fichier.