TP 02 Cluster Hadoop de N Machines

A. Objectifs:

- Configurer un cluster Hadoop avec trois machines.
- Déployer et exécuter des programmes MapReduce sur ce cluster.
- Dans ce TP, nous utilisons l'image Hadoop de Lilia Sfaxi de l'Université de Tunis, Tunisie

  https://registry.hub.docker.com/r/liliasfaxi/hadoop-cluster

➔ Ce Tp est noté, n'oubliez pas de soumettre votre compte rendu.

B. Installer un cluster Hadoop avec trois machines:

1. **Télécharger l'image docker uploadée sur dockerhub:**

   docker pull liliasfaxi/hadoop-cluster:latest

   et voir les images disponibles avec : docker images

2. **Créer un réseau Docker :**

   docker network create --driver=bridge hadoop

   → docker network create: Cette commande est utilisée pour créer un nouveau réseau Docker.

   → driver=bridge: Cette option spécifie le type de réseau à créer. bridge est le type de réseau par défaut pour Docker. Il permet aux conteneurs sur le même réseau de communiquer entre eux tout en isolant ce réseau du réseau hôte.

   → hadoop: C'est le nom de ce réseau.

3. **Pour vérifier que le réseau a bien été créé, vous pouvez utiliser la commande suivante :** *docker network ls*
4. **Créer et de démarrer un cluster Hadoop :**

- Démarrage du conteneur hadoop-master

docker run -itd --net=hadoop -p 9870:9870 -p 8088:8088 -p 7077:7077 -p 16010:16010 --name hadoop-master --hostname hadoop-master liliasfaxi/hadoop-cluster:latest

→ docker run -itd: Démarre un conteneur en mode interactif (-i), avec un pseudo-terminal (-t), et en arrière-plan (-d).

→ --net=hadoop: Connecte le conteneur au réseau Docker nommé hadoop.

→ -p 9870:9870 -p 8088:8088 -p 7077:7077 -p 16010:16010: Expose les ports spécifiques du conteneur sur l'hôte :

– 9870:9870: Port pour l'interface web de Hadoop NameNode.

– 8088:8088: Port pour l'interface web du ResourceManager de YARN.

– 7077:7077: Port pour Spark.

– 16010:16010: Port pour l'interface web de HBase Master.

→ --name hadoop-master: Attribue le nom hadoop-master au conteneur.

→ --hostname hadoop-master: Définit le nom d'hôte du conteneur à hadoop-master.

→liliasfaxi/hadoop-cluster:latest: Utilise l'image Docker liliasfaxi/hadoop-cluster avec la dernière version

- Démarrage du conteneur hadoop-worker1

  docker run -itd -p 8040:8042 --net=hadoop --name hadoop-worker1 --hostname hadoop-worker1 liliasfaxi/hadoop-cluster:latest

  → -p 8040:8042: Expose le port 8042 du conteneur sur le port 8040 de l'hôte. Le port 8042 est utilisé par l'interface web du NodeManager de YARN

- Démarrage du conteneur hadoop-worker2:

  docker run -itd -p 8041:8042 --net=hadoop --name hadoop-worker2 --hostname hadoop-worker2 liliasfaxi/hadoop-cluster:latest

5. **Ouvrir un terminal interactif dans le conteneur hadoop-master, et travailler directement à l'intérieur du conteneur**

   docker exec -it hadoop-master bash

6. **Afficher le contenu du fichier start-hadoop.sh**
   a) Expliquez ce que fait ce script en détail.
   b) Pourquoi est-il important de démarrer HDFS avant YARN dans un cluster Hadoop ?
   c) Lancer ce script et vérifier que les services sont en cours d'exécution en accédant aux interfaces web suivantes : HDFS NameNode : http://localhost:9870, YARN ResourceManager : http://localhost:8088
   d) voir les fichiers HDFS : *hdfs dfs -ls /*

```
Starting resourcemanager
Starting nodemanagers
hadoop-worker2: Warning: Permanently added 'hadoop-worker2' (ED25519) to the list of known hosts.
hadoop-worker1: Warning: Permanently added 'hadoop-worker1' (ED25519) to the list of known hosts.


root@hadoop-master:~#
```

7. **Créer un répertoire nommé input dans le système de fichiers distribué HDFS :** *hdfs dfs -mkdir -p /input*

   Lorsqu'on travaille avec HDFS, il est crucial de comprendre la différence entre chemins relatifs et chemins absolus, car cela peut affecter l'endroit où vos répertoires ou fichiers sont créés.

   - Un chemin sans / au début (ex. : input) est considéré comme relatif à votre répertoire de travail actuel dans HDFS.

     Exemple :si votre répertoire de travail actuel est /user/root, la commande : hdfs dfs -mkdir input va créer le répertoire suivant : /user/root/input.

- Un chemin qui commence par / (ex. : /input) est considéré comme absolu, c'est-à-dire qu'il spécifie directement la position à la racine de HDFS.

8. Charger le fichier purchases dans le répertoire input (de HDFS)

hdfs dfs –put purchases.txt input

9. Afficher le contenu du répertoire *input*

hdfs dfs -ls input

10. Afficher les dernières lignes du fichier purchases:

hdfs dfs -tail input/purchases.txt

```
root@hadoop-master:~# hdfs dfs -tail input/purchases.txt
31      17:59   Norfolk Toys    164.34  MasterCard
2012-12-31      17:59   Chula Vista     Music   380.67  Visa
2012-12-31      17:59   Hialeah Toys    115.21  MasterCard
2012-12-31      17:59   Indianapolis    Men's Clothing  158.28  MasterCard
2012-12-31      17:59   Norfolk Garden  414.09  MasterCard
2012-12-31      17:59   Baltimore       DVDs    467.3   Visa
2012-12-31      17:59   Santa Ana       Video Games     144.73  Visa
2012-12-31      17:59   Gilbert Consumer Electronics    354.66  Discover
2012-12-31      17:59   Memphis Sporting Goods  124.79  Amex
2012-12-31      17:59   Chicago Men's Clothing  386.54  MasterCard
2012-12-31      17:59   Birmingham      CDs     118.04  Cash
2012-12-31      17:59   Las Vegas       Health and Beauty       420.46  Amex
2012-12-31      17:59   Wichita Toys    383.9   Cash
2012-12-31      17:59   Tucson  Pet Supplies    268.39  MasterCard
2012-12-31      17:59   Glendale        Women's Clothing        68.05   Amex
2012-12-31      17:59   Albuquerque     Toys    345.7   MasterCard
2012-12-31      17:59   Rochester       DVDs    399.57  Amex
2012-12-31      17:59   Greensboro      Baby    277.27  Discover
2012-12-31      17:59   Arlington       Women's Clothing        134.95  MasterCard
2012-12-31      17:59   Corpus Christi  DVDs    441.61  Discover
```

11. Créer un Projet Maven dans VS Code

Installer les Extensions Nécessaires

Ouvrez VS Code et installez les extensions suivantes :

- Java Extension Pack : qui inclut des outils pour le développement Java.

- *Maven for Java : qui fournit des fonctionnalités spécifiques à Maven.*
- Ajouter les dépendances Hadoop, HDFS et Map Reduce dans pom.xml
- Écrire la classe *WordCount* permet de calculer le nombre de mots dans un fichier donné avec Map Reduce.
- Tester et valider votre programme Map Reduce en local
- Créer, compiler et packager le projet dans un fichier JAR

12. **Copier le fichier jar créé dans le contenaire master**

    docker cp file.jar hadoop-master:/root/file.jar

13. **Lancer le job map reduce sur master**

    hadoop jar <jar_file> <main_class> <input_path> <output_path>

14. **Afficher les dernières lignes du fichier résultat**
15. **Proposez une solution en utilisant le paradigme Map Reduce (les étapes), puis testez et exécutez votre programme sur le cluster, pour :**
    - Calculer le total des ventes par magasin
    - Trouver le montant total des ventes payées par chaque type de carte de paiement.
    - Déterminer le nombre total de transactions effectuées chaque jour.
    - Calculer le montant moyen des ventes par magasin.
    - Identifier les produits les plus vendus
    - Calculer le total des ventes par catégorie de produit