# Guide d'Installation de Hadoop YARN et MapReduce sur Docker

**Introduction**

Ce guide est conçu pour aider les étudiants à installer et tester Hadoop YARN et MapReduce sur Docker. En suivant les étapes progressivement, vous allez créer un environnement Hadoop fonctionnel.

**Prérequis**

**1. Configuration Système Requise**

- **RAM :** Minimum 8 Go (Recommandé : 16 Go)
- **Disque :** Au moins 20 Go d'espace libre
- **Système d'exploitation :** Windows 10/11, macOS, ou Linux

**2. Logiciels Requis**

- Docker Desktop (Windows/Mac) ou Docker Engine (Linux)
- Docker Compose
- Un éditeur de texte (VS Code, Notepad++, etc.)

**3. Vérification de l'Installation de Docker**

Ouvrez-le terminal/invite de commandes et exécutez ces commandes :

docker --version
docker-compose --version

Les deux commandes doivent retourner les informations de version. Si vous obtenez une erreur, installez d'abord Docker.

---

**Étape 1 : Création du Dossier de Projet**

1. Créez un nouveau dossier sur votre bureau ou à l'emplacement souhaité:

   ```
   mkdir hadoop-docker-lab
   cd hadoop-docker-lab
   ```

2. Nous allons travailler dans ce dossier. Tous les fichiers seront ici.

---

**Étape 2 : Création du Fichier Docker Compose**

Dans le dossier `hadoop-docker-lab`, créez un fichier nommé `docker-compose.yml` et écrivez le contenu suivant :

```
version : '3'

services :
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    restart: always
    ports:
      - 9870:9870
      - 9000:9000
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
    environment:
      - CLUSTER_NAME=test
    env_file:
      - ./hadoop.env

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode
    restart: always
    volumes:
      - hadoop_datanode:/hadoop/dfs/data
    environment:
      SERVICE_PRECONDITION: "namenode:9870"
    env_file:
      - ./hadoop.env

  resourcemanager:
    image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
    container_name: resourcemanager
    restart: always
    ports:
      - 8088:8088
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
    env_file:
      - ./hadoop.env

  nodemanager:
    image: bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
    container_name: nodemanager
    restart: always
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864
resourcemanager:8088"
    env_file:
      - ./hadoop.env
```

```
  historyserver:
    image: bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8
    container_name: historyserver
    restart: always
    ports:
      - 8188:8188
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864
resourcemanager:8088"
    volumes:
      - hadoop_historyserver:/hadoop/yarn/timeline
    env_file:
      - ./hadoop.env

volumes:
  hadoop_namenode:
  hadoop_datanode:
  hadoop_historyserver:
```

---

**Étape 3 : Création du Fichier de Variables d'Environnement**

Dans le même dossier, créez un fichier nommé `hadoop.env` et ajoutez ce contenu :

```
CORE_CONF_fs_defaultFS=hdfs://namenode:9000
CORE_CONF_hadoop_http_staticuser_user=root
CORE_CONF_hadoop_proxyuser_hue_hosts=*
CORE_CONF_hadoop_proxyuser_hue_groups=*
CORE_CONF_io_compression_codecs=org.apache.hadoop.io.compress.SnappyCodec

HDFS_CONF_dfs_webhdfs_enabled=true
HDFS_CONF_dfs_permissions_enabled=false
HDFS_CONF_dfs_replication=1

YARN_CONF_yarn_log___aggregation___enable=true
YARN_CONF_yarn_log_server_url=http://historyserver:8188/applicationhistory
/logs/
YARN_CONF_yarn_resourcemanager_recovery_enabled=true
YARN_CONF_yarn_resourcemanager_store_class=org.apache.hadoop.yarn.server.r
esourcemanager.recovery.FileSystemRMStateStore
YARN_CONF_yarn_resourcemanager_scheduler_class=org.apache.hadoop.yarn.serv
er.resourcemanager.scheduler.capacity.CapacityScheduler
YARN_CONF_yarn_scheduler_capacity_root_default_maximum___allocation___mb=8
192
YARN_CONF_yarn_scheduler_capacity_root_default_maximum___allocation___vcor
es=4
YARN_CONF_yarn_resourcemanager_fs_state___store_uri=/rmstate
YARN_CONF_yarn_resourcemanager_system___metrics___publisher_enabled=true
YARN_CONF_yarn_resourcemanager_hostname=resourcemanager
YARN_CONF_yarn_resourcemanager_address=resourcemanager:8032
YARN_CONF_yarn_resourcemanager_scheduler_address=resourcemanager:8030
```

```
YARN_CONF_yarn_resourcemanager_resource__tracker_address=resourcemanager:8
031
YARN_CONF_yarn_timeline___service_enabled=true
YARN_CONF_yarn_timeline___service_generic___application___history_enabled=
true
YARN_CONF_yarn_timeline___service_hostname=historyserver
YARN_CONF_mapreduce_map_output_compress=true
YARN_CONF_mapred_map_output_compress_codec=org.apache.hadoop.io.compress.S
nappyCodec
YARN_CONF_yarn_nodemanager_resource_memory___mb=4096
YARN_CONF_yarn_nodemanager_resource_cpu___vcores=4
YARN_CONF_yarn_nodemanager_disk___health___checker_max___disk___utilizatio
n___per___disk___percentage=98.5
YARN_CONF_yarn_nodemanager_remote___app___log___dir=/app-logs
YARN_CONF_yarn_nodemanager_aux___services=mapreduce_shuffle

MAPRED_CONF_mapreduce_framework_name=yarn
MAPRED_CONF_mapred_child_java_opts=-Xmx4096m
MAPRED_CONF_mapreduce_map_memory_mb=4096
MAPRED_CONF_mapreduce_reduce_memory_mb=8192
MAPRED_CONF_mapreduce_map_java_opts=-Xmx3072m
MAPRED_CONF_mapreduce_reduce_java_opts=-Xmx6144m
MAPRED_CONF_yarn_app_mapreduce_am_env=HADOOP_MAPRED_HOME=/opt/hadoop-
3.2.1/
MAPRED_CONF_mapreduce_map_env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
MAPRED_CONF_mapreduce_reduce_env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
```

---

**Étape 4 : Démarrage du Cluster Hadoop**

1. Dans le terminal/invite de commandes, accédez au dossier `hadoop-docker-lab`.
2. Exécutez la commande suivante pour démarrer les conteneurs :

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

3. Vérifiez si les conteneurs sont en cours d'exécution :

   ```
   docker-compose ps
   ```

   Tous les services doivent être dans l'état "Up".

4. Pour consulter les logs:

   ```
   docker-compose logs -f
   ```

   (Appuyez sur Ctrl+C pour quitter)

---

**Étape 5: Vérification des Interfaces Web**

Ouvrez les adresses suivantes dans votre navigateur pour vérifier que votre cluster Hadoop fonctionne :

- **Interface NameNode:** http://localhost:9870
- **Interface YARN ResourceManager:** http://localhost:8088
- **Interface History Server:** http://localhost:8188

Dans chaque interface, vous pouvez voir les informations du cluster, les nœuds et leurs états.

---

**Étape 6: Téléchargement de Fichiers vers HDFS**

1. Connectez-vous au conteneur NameNode : `docker exec -it namenode bash`
2. Créez un dossier dans HDFS : `hdfs dfs -mkdir -p /user/root/input`
3. Créez un fichier texte de test :

```
echo "Bonjour le monde Hadoop" > test.txt
echo "MapReduce fonctionne" >> test.txt
echo "YARN est en cours d'exécution" >> test.txt
```

4. Téléchargez le fichier vers HDFS : `hdfs dfs -put test.txt /user/root/input/`
5. Vérifiez que le fichier a été téléchargé :

```
hdfs dfs -ls /user/root/input/
hdfs dfs -cat /user/root/input/test.txt
```

---

**Étape 7 : Exécution d'un Job MapReduce (Exemple WordCount)**

1. Vous devez toujours être dans le conteneur NameNode. Exécutez l'exemple WordCount:

```
hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-
mapreduce-examples-3.2.1.jar wordcount /user/root/input
/user/root/output
```

2. Attendez que le job se termine. Vous verrez la progression du job MapReduce dans le terminal.
3. Vérifiez les résultats:

```
hdfs dfs -ls /user/root/output/
hdfs dfs -cat /user/root/output/part-r-00000
```

4. Le résultat devrait ressembler à ceci :

| | |
|---|---|
| Bonjour | 1 |
| Hadoop | 1 |
| MapReduce | 1 |
| YARN | 1 |
| cours | 1 |
| d'exécution | 1 |
| en | 1 |
| est | 1 |
| fonctionne | 1 |
| le | 1 |
| monde | 1 |

---

**Étape 8 : Surveillance des Jobs dans YARN**

1. Accédez à http://localhost:8088 dans votre navigateur.
2. Cliquez sur l'onglet **"Applications"** dans le menu de gauche.
3. Vous verrez le job WordCount que vous venez d'exécuter dans la liste.
4. Cliquez sur le job pour examiner ses détails, les ressources utilisées et les logs.

---

**Étape 9 : Écrire Votre Propre Code MapReduce (Optionnel)**

**Exemple Simple MapReduce en Python**

1. Créez le fichier mapper.py (voir le fichier codexemplaire.ipynb)
2. Créez le fichier reducer.py (voir le fichier codexemplaire.ipynb)
3. Copiez les fichiers dans le conteneur :

```
docker cp mapper.py namenode:/tmp/
docker cp reducer.py namenode:/tmp/
```

4. Connectez-vous au conteneur et définissez les permissions :

```
docker exec -it namenode bash
chmod +x /tmp/mapper.py /tmp/reducer.py
```

5. Exécutez avec Hadoop Streaming:

```
hadoop jar /opt/hadoop-3.2.1/share/hadoop/tools/lib/hadoop-
streaming-3.2.1.jar \
  -input /user/root/input \
  -output /user/root/python-output \
  -mapper /tmp/mapper.py \
  -reducer /tmp/reducer.py
```

---

**Étape 10 : Nettoyage et Arrêt**

1. Arrêter les Conteneurs

   ```
   docker-compose stop
   ```

2. Supprimer Complètement les Conteneurs

   ```
   docker-compose down
   ```

3. Supprimer Également les Volumes (Supprime toutes les données!)

   ```
   docker-compose down -v
   ```

4. Redémarrer

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

---

**Dépannage**

**Si les Conteneurs ne Démarrent pas**

1. Vérifiez les logs:

   ```
   docker-compose logs namenode
   docker-compose logs resourcemanager
   ```

2. Vérifiez si les ports sont utilisés :

   ```
   # Windows
   netstat -ano | findstr "9870"
   # Linux/Mac
   lsof -i :9870
   ```

3. Allouez plus de RAM à Docker (Docker Desktop Settings > Resources)

**Si les Commandes HDFS ne Fonctionnent pas**

1. Assurez-vous que le conteneur a complètement démarré (attendez 30-60 secondes)
2. Vérifiez que le NameNode est sorti du mode Safe Mode :

   ```
   hdfs dfsadmin -safemode get
   ```

**Si le Job MapReduce Échoue**

1. Vérifiez les logs du YARN ResourceManager
2. Réduisez les paramètres de mémoire (dans le fichier hadoop.env)
3. Assurez-vous que le dossier de sortie n'existe pas déjà (supprimez-le si nécessaire)

**Ressources Supplémentaires**

- **Documentation Officielle Apache Hadoop:** https://hadoop.apache.org/docs/
- **Architecture YARN:** https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
- **Tutoriel MapReduce:** https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

**Conclusion**

Félicitations! Vous avez réussi à installer Hadoop YARN et MapReduce sur Docker. Vous pouvez continuer à apprendre les techniques de traitement de big data sur cet environnement.

**Bon travail!**