1️⃣

Lab 1 – Hadoop & MapReduce (Word Count)

1️⃣ Create Docker Network

We create a bridge network to allow all Hadoop containers to communicate.

```
docker network create --driver bridge hadoop

docker network ls → To check available networks
```

2️⃣ Start Hadoop Master Container

Start the master container and map its Hadoop UI ports to host machine ports.

```
docker run -itd --network hadoop `
  --hostname hadoop-master `
  --name hadoop-master `
  -p 8090:50070 `
  -p 8091:8088 `
  -p 16011:16010 `
  liliasfaxi/spark-hadoop:hv-2.7.2
```

- **Ports mapping**:
  - `50070 → 8090`: NameNode Web UI
  - `8088 → 8089`: YARN ResourceManager Web UI
  - `16010 → 16011`: JobHistory Server

3️⃣ Start Hadoop Slave Containers

Start two slave nodes for HDFS and YARN.

```
docker run -itd --network hadoop --hostname hadoop-slave1 --name hadoop
-slave1 -p 8042:8042 liliasfaxi/spark-hadoop:hv-2.7.2

docker run -itd --network hadoop --hostname hadoop-slave2 --name hadoop
-slave2 -p 8043:8042 liliasfaxi/spark-hadoop:hv-2.7.2
```

- **Port 8042** → NodeManager

To start the containers if they already exist :

```
docker start hadoop-master
docker start hadoop-slave1
docker start hadoop-slave2
```

4️⃣ Access Hadoop Master Container

```
docker exec -it hadoop-master bash
```

- You are now inside the master container shell.

5️⃣ Start Hadoop Services

```
./start-hadoop.sh
```

> If this script is missing, alternatively use:
>
> ```
> hdfs namenode -format    # only first time
> start-dfs.sh
> ```

```
start-yarn.sh
jps
```

- `jps` confirms running daemons: NameNode, DataNode, ResourceManager, NodeManager, SecondaryNameNode.

6️⃣ Prepare HDFS Input

Create input directory:

```
hdfs dfs -mkdir /input
```

Upload input file:

```
hdfs dfs -put purchases.txt /input
```

Verify uploaded files:

```
hdfs dfs -ls /input
```

Check last few lines of the file:

```
    tail purchases.txt
```

7️⃣ Copy WordCount JAR to Container

From **host machine**, copy the JAR file:

```
docker cp "C:\Users\raouf\OneDrive - esi-sba.dz\Desktop\3CS\Big Data\TP01\untitled\target\wordcount-1.0-SNAPSHOT.jar" hadoop-master:/root/
```

Inside the container, verify:

```
ls /root
```

- You should see `wordcount-1.0-SNAPSHOT.jar`.

8️⃣ Run WordCount MapReduce Job

```
hadoop jar /root/wordcount-1.0-SNAPSHOT.jar tp1.WordCount /input /output
```

> If /output exists from a previous run:
>
> ```
> hdfs dfs -rm -r /output
> ```

9️⃣ Check Output

List output directory:

```
hdfs dfs -ls /output
```

- `_SUCCESS` → job completed successfully
- `part-r-00000` → contains WordCount results

View first 20 lines of output:

```
hdfs dfs -cat /output/part-r-00000 | head -n 20
```

Cleanup / Rerun

- Hadoop cannot overwrite existing output, so you must delete it before rerunning:

```
hdfs dfs -rm -r /output
```

- To completely reset the lab:

```
stop-yarn.sh
stop-dfs.sh
docker rm -f hadoop-master hadoop-slave1 hadoop-slave2
docker network rm hadoop
```