## Week 2 Assignment: Dive into HDFS and MapReduce

### Objective: Familiarize with the core functionalities of HDFS and get a practical understanding of MapReduce.

#### **1. Environment Initialization**

* Start by navigating to the required directory and initiating the Docker containers:
* cd bellevue-bigdata  
  cd hadoop-hive-spark-hbase  
  docker-compose up -d
* If you’re using Google Cloud, remember to set up port forwarding as outlined in the Week 1 assignment.
* Access the master container:
* docker-compose exec master bash

#### **2. Deep Dive into HDFS**

* Check the HDFS report:
* hdfs dfsadmin -report

**Deliverable:** Screenshot of the output.

* Load the grades.csv into HDFS:
* hdfs dfs -put /data/grades.csv /
* Verify that the data has been loaded:
* hdfs dfs -ls /

**Deliverable:** Screenshot proving the data has been loaded.

* Exit the master Docker container:
* CTRL+D or exit
* SSH into each of the 3 worker nodes and verify the data:
* docker-compose exec worker1 bash  
  hdfs dfs -ls /  
  CTRL+D or exit  
    
  docker-compose exec worker2 bash  
  hdfs dfs -ls /  
  CTRL+D or exit  
    
  docker-compose exec worker3 bash  
  hdfs dfs -ls /  
  CTRL+D or exit
* All worker nodes should display the grades.csv file.
* Re-enter the master container:
* docker-compose exec master bash
* Explore more HDFS commands:
* hdfs dfs -help
* Execute three other HDFS commands of your choice and observe their outputs.

**Deliverable:** Screenshots of the three chosen HDFS command outputs.

#### **3. Introduction to YARN**

* Inside the master container, inspect the YARN nodes:
* yarn node -list

**Deliverable:** Screenshot of the results.

* Understand the yarn.scheduler.maximum-allocation-mb property. This is the maximum memory capacity available for a single container.
* Modify the maximum memory allocation:
* sed -i "/<name>yarn.scheduler.maximum-allocation-mb<\/name>/,/<\/property>/s/<value>.\*<\/value>/<value>2048<\/value>/" /usr/program/hadoop/etc/hadoop/yarn-site.xml
* Restart the ResourceManager:
* yarn --daemon stop resourcemanager  
  yarn --daemon start resourcemanager

**Deliverable:** Screenshot from the YARN UI showing the updated maximum memory (2048 MB).

#### **4. Experimenting with MapReduce**

* Run the example MapReduce Pi job:
* libjars=$(find /usr/program/hadoop/share/hadoop/mapreduce -name "\*.jar" | tr '\n' ',')
* hadoop jar /usr/program/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar pi -libjars ${libjars} 2 10

**Deliverable:** A summary of the result and its significance.

## Shutting Down

Ensure all Docker containers are turned off with docker-compose down for each directory. If you’re using google cloud, please shut down your virtual machine to preserve cloud costs.