

Mount the drive to google drive and save the UUD.txt on the Gdrive

```
from google.colab import drive
drive.mount('/content/drive')
```

↗ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount()

Initiate Hadoop

- Installing all the necessary JDK and library
- Adding all required property to the file

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!java -version
```

```
!update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
!update-alternatives --set javac /usr/lib/jvm/java-8-openjdk-amd64/bin/javac
!update-alternatives --set jps /usr/lib/jvm/java-8-openjdk-amd64/bin/jps
!java -version
```

```
#Finding the default Java path
!readlink -f /usr/bin/java | sed "s:bin/java::"
!apt-get install openssh-server -qq > /dev/null
!service ssh start
```

```
!grep Port /etc/ssh/sshd_config
```

```
#Creating a new rsa key pair with empty password
!ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa <<< y
```

```
# See id_rsa.pub content
!more /root/.ssh/id_rsa.pub
```

hdfs-wordcount.txt ✕

...

```
1 abadi 1
2 adil 2
3 allah 1
4 atas 2
5 bagi 1
6 bahwa 1
7 bangsa 3
8 bebas 1
9 beradab 1
10 berbahagia 1
11 berdasar 1
12 berdasarkan 1
13 berdaulat 1
14 berkat 1
15 berkedaulatan 1
16 berkehidupan 1
17 bersatu 1
18 dalam 3
19 dan 11
20 darah 1
21 daripada 1
22 dasar 1
23 dengan 6
24 depan 1
25 di 1
26 didorongkan 1
27 dihapuskan 1
28 dipimpin 1
29 disusunlah 1
30 dunia 2
31 esa 1
32 gerbang 1
33 hak 1
34 harus 1
35 hikmat 1
36 ialah 1
37 ikut 1
38 indonesia 12
39 ini 1
40 itu 4
```

```

#Copying the key to authorized keys
!cat $HOME/.ssh/id_rsa.pub > $HOME/.ssh/authorized_keys
#Changing the permissions on the key
!chmod 0600 ~/.ssh/authorized_keys

#Conneting with the local machine
!ssh -o StrictHostKeyChecking=no localhost uptime

#Downloading Hadoop 3.2.3
!wget -q https://archive.apache.org/dist/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz

#Untarring the file
!sudo tar -xzf hadoop-3.2.3.tar.gz
#Removing the tar file
!rm hadoop-3.2.3.tar.gz

#Copying the hadoop files to user/local
!cp -r hadoop-3.2.3/ usr/local/
#-r copy directories recursively

#Adding JAVA_HOME directory to hadoop-env.sh file
!sed -i '/export JAVA_HOME=/a export JAVA_HOME=\usr\lib\jvm\java-8-openjdk-amd64' usr/local

import os
#Creating environment variables
#Creating Hadoop home variable

os.environ["HADOOP_HOME"] = "/usr/local/hadoop-3.2.3"
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["JRE_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64/jre"
os.environ["PATH"] += f'{os.environ["JAVA_HOME"]}/bin:{os.environ["JRE_HOME"]}/bin:{os.environ["PATH"]}'

#Downloading text example to use as input
!wget -q https://www.mirrorservice.org/sites/ftp.ibiblio.org/pub/docs/books/gutenberg/1/0/101/1

```



```

openjdk version "11.0.24" 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu322.04)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu322.04, mixed mode, sharing)
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java to provide /usr/
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javac to provide /usr/bin/
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jps to provide /usr/bin/j
openjdk version "1.8.0_422"

```

```
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~22.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
/usr/lib/jvm/java-8-openjdk-amd64/jre/
* Starting OpenBSD Secure Shell server sshd
  ...done.
#Port 22
#GatewayPorts no
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zBJIBHKfwlTHYbaUa5KLccwYWj5Z2+srxAOlHW/3q08 root@1920d4359aa0
The key's randomart image is:
+---[RSA 3072]-----+
| . +++. *o          |
| =oo+Bo.           |
| +o@+*o.           |
| . O.O  *+.         |
|   B =.oS.         |
|   . = .. .        |
|   . o   E.         |
|   . .  ..         |
|   ....O.          |
+----[SHA256]-----+
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC/Dq8xxLQhQNMzvqU38wDAUwjZRTok4bxAouHfitpmAsTGONx8cY
5Sxg0X0Y0+sd5b/VYI5iRDCqRuGP07gm5z5wtLD3WaYqVSMc/0EcbbZaRy5LXK00ujyF1IaZ2vClTL+LDD6UmAhrdh
1wZIPBK42Mi7arVI6JNk7Cp+P7zn8Edo8vx89Pxx59faqBBvt9dcIMtpkA+tY4X3yOp1SVLbt08fuwfBTCKb9YXpEd
9BjCg+0K9vksBXle+u7z4ps4bAGh7gFaWzHJ0jcptaXwswkCtSXRWGqV0BjfA5etzg7irJb7CAEFo6amPNgjAHHAau
+TjCDNUZzLBLaVkBroy2tPiFNosBMFoSIrM99xnj1/x8jemXWaeQ16rhV+GaaDRl+CkrIZRMbFcjNruxf4+2VLhhkO
OekcwnPCXz0fWDYkMunomyTPcJxRHqQatdmKctKx5CUIryUYTXM= root@1920d4359aa0
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
15:19:49 up 2 min,  0 users,  load average: 3.28, 1.28, 0.48
```

```
#Adding required property to core-site.xml file
%%bash
cat <<EOF > $HADOOP_HOME/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
    <description>Where HDFS NameNode can be found on the network</description>
  </property>
</configuration>
EOF
```

```
%%bash
cat <<EOF > $HADOOP_HOME/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

</configuration>
EOF
```

```
%%bash
cat <<EOF > $HADOOP_HOME/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
  <name>mapreduce.application.classpath</name>
  <value>$HADOOP_HOME/share/hadoop/mapreduce/*:$HADOOP_HOME/share/hadoop/mapreduce/lib/*</va
</property>

</configuration>
EOF
```

```

%%bash
cat <<EOF > $HADOOP_HOME/etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<property>
  <description>The hostname of the RM.</description>
  <name>yarn.resourcemanager.hostname</name>
  <value>localhost</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DIS
</property>

<!-- Site specific YARN configuration properties -->

</configuration>
EOF

```

✓ Formating the HDFS

```
!$HADOOP_HOME/bin/hdfs namenode -format
```

```
#Creating other necessary enviroment variables before starting nodes
```

```
os.environ["HDFS_NAMENODE_USER"] = "root"  
os.environ["HDFS_DATANODE_USER"] = "root"  
os.environ["HDFS_SECONDARYNAMENODE_USER"] = "root"  
os.environ["YARN_RESOURCEMANAGER_USER"] = "root"  
os.environ["YARN_NODEMANAGER_USER"] = "root"
```

```
#Launching hdfs daemons  
!$HADOOP_HOME/sbin/start-dfs.sh
```

```
#Launching yarn daemons  
#nohup causes a process to ignore a SIGHUP signal  
!nohup $HADOOP_HOME/sbin/start-yarn.sh
```

```
#Listing the running daemons  
!jps
```




```
2024-08-17 15:20:34,118 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-08-17 15:20:34,118 INFO util.GSet: VM type          = 64-bit
2024-08-17 15:20:34,119 INFO util.GSet: 0.029999999329447746% max memory 2.8 GB = 886.4
2024-08-17 15:20:34,119 INFO util.GSet: capacity        = 2^17 = 131072 entries
2024-08-17 15:20:34,152 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1270534210-
2024-08-17 15:20:34,218 INFO common.Storage: Storage directory /tmp/hadoop-root/dfs/name
2024-08-17 15:20:34,257 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hado
2024-08-17 15:20:34,369 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-root
2024-08-17 15:20:34,391 INFO namenode.NNStorageRetentionManager: Going to retain 1 image
2024-08-17 15:20:34,459 INFO namenode.FSNamesystem: Stopping services started for active
2024-08-17 15:20:34,459 INFO namenode.FSNamesystem: Stopping services started for standb
2024-08-17 15:20:34,467 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 whe
2024-08-17 15:20:34,468 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at 1920d4359aa0/172.28.0.12
*****/

Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [1920d4359aa0]
1920d4359aa0: Warning: Permanently added '1920d4359aa0' (ED25519) to the list of known h
nohup: ignoring input and appending output to 'nohup.out'
2181 NodeManager
1816 SecondaryNameNode
1626 DataNode
2074 ResourceManager
2302 Jps
1518 NameNode
```

```
#Report the basic file system information and statistics
!$HADOOP_HOME/bin/hdfs dfsadmin -report
```

```
➤ Configured Capacity: 115658190848 (107.72 GB)
Present Capacity: 79997550592 (74.50 GB)
DFS Remaining: 79997526016 (74.50 GB)
DFS Used: 24576 (24 KB)
DFS Used%: 0.00%
Replicated Blocks:
    Under replicated blocks: 0
    Blocks with corrupt replicas: 0
    Missing blocks: 0
    Missing blocks (with replication factor 1): 0
    Low redundancy blocks with highest priority to recover: 0
    Pending deletion blocks: 0
Erasure Coded Block Groups:
    Low redundancy block groups: 0
    Block groups with corrupt internal blocks: 0
```

Missing block groups: 0
Low redundancy blocks with highest priority to recover: 0
Pending deletion blocks: 0

Live datanodes (1):

Name: 127.0.0.1:9866 (localhost)
Hostname: 1920d4359aa0
Decommission Status : Normal
Configured Capacity: 115658190848 (107.72 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 35643871232 (33.20 GB)
DFS Remaining: 79997517824 (74.50 GB)
DFS Used%: 0.00%
DFS Remaining%: 69.17%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Aug 17 15:21:10 UTC 2024
Last Block Report: Sat Aug 17 15:20:50 UTC 2024
Num of Blocks: 0

#Exploring Hadoop utilities available
!ls \$HADOOP_HOME/share/hadoop/tools/lib/

➡ aliyun-java-sdk-core-4.5.10.jar hadoop-gridmix-3.2.3.jar
aliyun-java-sdk-kms-2.11.0.jar hadoop-kafka-3.2.3.jar
aliyun-java-sdk-ram-3.1.0.jar hadoop-openstack-3.2.3.jar
aliyun-sdk-oss-3.13.0.jar hadoop-resourceestimator-3.2.3.jar
aws-java-sdk-bundle-1.11.901.jar hadoop-rumen-3.2.3.jar
azure-data-lake-store-sdk-2.2.9.jar hadoop-sls-3.2.3.jar
azure-keyvault-core-1.0.0.jar hadoop-streaming-3.2.3.jar
azure-storage-7.0.0.jar ini4j-0.5.4.jar
hadoop-aliyun-3.2.3.jar jdom2-2.0.6.jar
hadoop-archive-logs-3.2.3.jar kafka-clients-2.8.1.jar
hadoop-archives-3.2.3.jar lz4-java-1.7.1.jar
hadoop-aws-3.2.3.jar ojalgo-43.0.jar
hadoop-azure-3.2.3.jar opentracing-api-0.33.0.jar
hadoop-azure-datalake-3.2.3.jar opentracing-noop-0.33.0.jar
hadoop-datajoin-3.2.3.jar opentracing-util-0.33.0.jar

hadoop-distcp-3.2.3.jar
hadoop-extras-3.2.3.jar
hadoop-fs2img-3.2.3.jar

org.jacoco.agent-0.8.5-runtime.jar
wildfly-openssl-1.0.7.Final.jar
zstd-jni-1.4.9-1.jar

✓ Hadoop Streaming using Python

1. Make a directory files (similar with linux command)
2. Copy file from local to HDFS
3. Create mapper.py
4. Create reducer.py
5. Changing the permission of the files
6. Running the MapReduce programs
7. Check the output file
8. Copy the output file to local

```
#Creating directory in HDFS
```

```
!$HADOOP_HOME/bin/hdfs dfs -mkdir /word_count_with_python
```


```
#Copying the file from local file system to Hadoop distributed file system (HDFS)
```

```
!$HADOOP_HOME/bin/hdfs dfs -put /content/drive/MyDrive/ColabNotebooks/FTDE-Hadoop/UUD.txt /wor
```

```
# Check directory in HDFS
```

```
!$HADOOP_HOME/bin/hdfs dfs -ls /word_count_with_python
```

```
📁 Found 1 items
-rw-r--r--  1 root supergroup      1423 2024-08-17 15:21 /word_count_with_python/UUD.txt
```



```
%%writefile mapper.py

#!/usr/bin/env python

###!' is known as shebang and used for interpreting the script

# import sys because we need to read and write data to STDIN and STDOUT
import sys

# reading entire line from STDIN (standard input)
for line in sys.stdin:
    # to remove leading and trailing whitespace
    line = line.strip()
    # split the line into words, output data type list
    words = line.split()
    # delete [; , : . /]
    words = [word.replace('[', '').replace(';', '').replace(',', '').replace(':', '').replace('.', '').replace(' ', '') for word in words]
    # lower all words
    words = [word.lower() for word in words]

    # we are looping over the words array and printing the word
    # with the count of 1 to the STDOUT
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        print('%s\t%s' % (word, 1))
```

⇒ Writing mapper.py

```
%%writefile reducer.py
```

```
#!/usr/bin/env python
```

```
from operator import itemgetter
import sys
```

```
current_word = None
current_count = 0
word = None
```

```
# read the entire line from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # splitting the data on the basis of tab we have provided in mapper.py
    word, count = line.split('\t', 1)
    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue
```

```
# this IF-switch only works because Hadoop sorts map output
# by key (here: word) before it is passed to the reducer
```

```
if current_word == word:
    current_count += count
else:
    if current_word: #to not print current_word=None
        # write result to STDOUT
        print('%s\t%s' % (current_word, current_count))
    current_count = count
    current_word = word
```

```
# do not forget to output the last word if needed!
if current_word == word:
    print('%s\t%s' % (current_word, current_count))
```

➦ Writing reducer.py

```
#Testing our MapReduce job locally (Hadoop does not participate here)
!cat '/content/drive/MyDrive/ColabNotebooks/FTDE-Hadoop/UUD.txt' | python mapper.py | sort -k1
#We apply sorting after the mapper because it is the default operation in MapReduce architectu
```

```
⇒ abadi 1
   adil 2
   allah 1
   atas 2
   bagi 1
   bahwa 1
   bangsa 3
   bebas 1
   beradab 1
   berbahagia 1
```

```
#Changing the permissions of the files
!chmod 777 /content/mapper.py /content/reducer.py
#Setting 777 permissions to a file or directory means that it will be readable, writable and e
```

```
#Running MapReduce programs
!$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.2.3.jar \
  -input /word_count_with_python/UUD.txt \
  -output /word_count_with_python/output \
  -mapper "python /content/mapper.py" \
  -reducer "python /content/reducer.py"
```

```
⇒ packageJobJar: [/tmp/hadoop-unjar4216440069373083778/] [] /tmp/streamjob4864525865896900
2024-08-17 15:21:28,448 INFO client.RMPProxy: Connecting to ResourceManager at localhost/
2024-08-17 15:21:28,775 INFO client.RMPProxy: Connecting to ResourceManager at localhost/
2024-08-17 15:21:29,117 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for
2024-08-17 15:21:29,488 INFO mapred.FileInputFormat: Total input files to process : 1
2024-08-17 15:21:30,034 INFO mapreduce.JobSubmitter: number of splits:2
2024-08-17 15:21:30,364 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1723
2024-08-17 15:21:30,366 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-08-17 15:21:30,653 INFO conf.Configuration: resource-types.xml not found
2024-08-17 15:21:30,653 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'
2024-08-17 15:21:31,162 INFO impl.YarnClientImpl: Submitted application application_1723
2024-08-17 15:21:31,202 INFO mapreduce.Job: The url to track the job: http://1920d4359aa
2024-08-17 15:21:31,203 INFO mapreduce.Job: Running job: job_1723908063797_0001
2024-08-17 15:21:44,509 INFO mapreduce.Job: Job job_1723908063797_0001 running in uber m
2024-08-17 15:21:44,510 INFO mapreduce.Job: map 0% reduce 0%
2024-08-17 15:21:52,667 INFO mapreduce.Job: map 50% reduce 0%
2024-08-17 15:21:53,674 INFO mapreduce.Job: map 100% reduce 0%
2024-08-17 15:22:00,776 INFO mapreduce.Job: map 100% reduce 100%
2024-08-17 15:22:01,800 INFO mapreduce.Job: Job job_1723908063797_0001 completed success
```

2024-08-17 15:22:01,901 INFO mapreduce.Job: Counters: 54

File System Counters

FILE: Number of bytes read=2110
FILE: Number of bytes written=719602
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=2343
HDFS: Number of bytes written=1141
HDFS: Number of read operations=11
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0

Job Counters

Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=13128
Total time spent by all reduces in occupied slots (ms)=5730
Total time spent by all map tasks (ms)=13128
Total time spent by all reduce tasks (ms)=5730
Total vcore-milliseconds taken by all map tasks=13128
Total vcore-milliseconds taken by all reduce tasks=5730
Total megabyte-milliseconds taken by all map tasks=13443072
Total megabyte-milliseconds taken by all reduce tasks=5867520

Map-Reduce Framework

Map input records=7
Map output records=178
Map output bytes=1748
Map output materialized bytes=2116
Input split bytes=208
Combine input records=0
Combine output records=0
Reduce input groups=108
Reduce shuffle bytes=2116
Reduce input records=178
Reduce output records=108
Spilled Records=356

!\$HADOOP_HOME/bin/hdfs dfs -ls /word_count_with_python/output



Found 2 items

-rw-r--r--	1	root	supergroup	0	2024-08-17 15:21	/word_count_with_python/output/
-rw-r--r--	1	root	supergroup	1141	2024-08-17 15:21	/word_count_with_python/output/

#Printing out first 50 lines

!\$HADOOP_HOME/bin/hdfs dfs -cat /word_count_with_python/output/part-00000 | head -50