

Task No: 4	Implement Map Reduce concept using apache Hadoop	CO2
Date:	Tools: Docker, Windows	

Task 4.1: Apache Hadoop Installation

Aim:

To download, install and configure the Apache Hadoop in windows operating system,

Procedure:

1. Download the software docker desktop from the url <https://www.docker.com/products/docker-desktop/>
2. Download the software GIT for windows from the url <https://git-scm.com/download/win>
3. Install the docker desktop and Git windows
4. Start the docker desktop engine and then open the command prompt
5. Clone the docker-hadoop using the git command

```
git clone https://github.com/big-data-europe/docker-hadoop.git
```

6. Execute the following command to start the Hadoop sever using docker

```
cd docker-hadoop/  
docker compose up
```

7. Enter into the bash mode and execute the hdfs commands
8. Execute the commands to check the containers, ip address and port number of the Hadoop server

```
docker container ls
```

```
ipconfig
```

9. Shut down the Hadoop server

```
docker compose down
```

HDFS COMMANDS

Enter into the bash mode and execute the following commands

1. List files - ls

hdfs dfs -ls /

2. Make dir

hdfs dfs -mkdir /techcoreeasy

3. create empty file

hdfs dfs -touchz /techcoreeasy/test1.txt

4. cp file from local file system to hdfs

hdfs dfs -put tech.txt /techcoreeasy/

5. see contents - cat

hdfs dfs -cat /techcoreeasy/tech.txt

6. copy - cp

hdfs dfs -cp /techcoreeasy/test_cp

7. get file to local

hdfs dfs -get /techcoreeasy/test.txt .

8. Remove file

hdfs dfs -rmr /techcoreeasy/test.txt

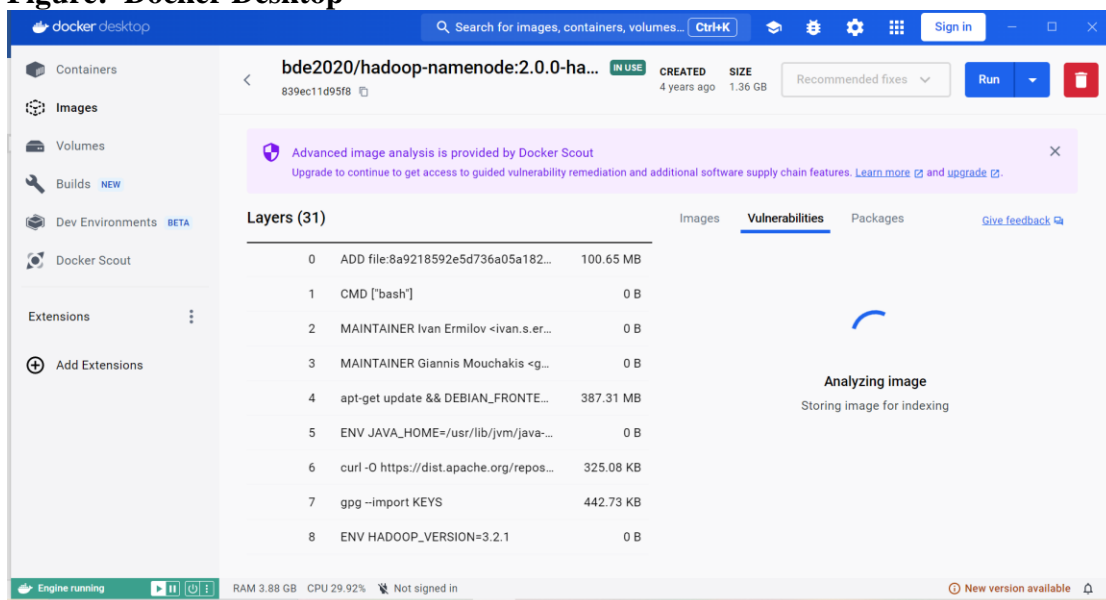
9. stst of a file

hdfs dfs -stat /path

10. exit command, switch from bash mode to command prompt

Output:

Figure: Docker Desktop



For docker compose up command

```
Administrator: Command Prompt - docker compose up
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd docker-hadoop

C:\Windows\System32\docker-hadoop>docker compose up
[+] Running 2/28
 - historyserver 3 layers [0/0] 0B/0B Pulling 39.1s
   - b2dc88cebe05 Waiting 30.4s
   - 13b908760168 Waiting 30.4s
   - 0991d53828a1 Waiting 30.4s
 - resourcemanager 12 layers [0/0] 19.55MB/564.5MB Pulling 39.1s
   - 3192219afd04 Downloading [=====] 11.47MB/45.38... 30.5s
   - 7127a1d8cccd Downloading [=] 4.318MB/159.4... 30.5s
   - 883a89599900 Download complete 4.7s
   - 77920a3e82af Download complete 10.0s
   - 92329e81aec4 Downloading [>] 3.757MB/359.7... 30.5s
   - f373218fec59 Waiting 30.5s
   - aa53513fe997 Waiting 30.5s
   - 8b1800105b98 Waiting 30.5s
   - c3a84a3e49c8 Waiting 30.5s
   - a65640a64a76 Waiting 30.5s
   - b0d764123f3e Waiting 30.5s
   - b04394ddb35d Waiting 30.5s
 - nodemanager1 2 layers [0/0] 0B/0B Pulling 39.1s
   - beaal71f32f6 Waiting 30.1s
   - 50dda04de8a9 Waiting 30.1s
 - datanode 3 layers [0/0] 0B/0B Pulling 39.1s
   - 3ca2ec07878c Waiting 30.1s
   - 26c2dd45430e Waiting 30.1s
   - 13c9c87a46cb Waiting 30.1s
 - namenode 3 layers [0/0] 0B/0B Pulling 39.1s
   - facffb3a6de3 Waiting 30.3s
   - c71a6df73788 Waiting 30.3s
   - 73b8c0ccb707 Waiting 30.3s
```

Figure: Start the Hadoop server

Open the browser and enter the url <http://172.17.205.161:9870/> or <http://your ip address:9870>

[Hadoop](#) [Overview](#) [Datanodes](#) [Datanode Volume Failures](#) [Snapshot](#) [Startup Progress](#) [Utilities](#)

Overview 'hadoop.tecadmin.com:9000' (active)

Started:	Sat Feb 01 13:42:11 +0530 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 21:26:00 +0530 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-c014b59a-461e-481a-add9-5f98d542b4bd
Block Pool ID:	BP-450651673-45.58.38.202-1580544640120

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 53.91 MB of 118 MB Heap Memory. Max Heap Memory is 443 MB.

Non Heap Memory used 47.6 MB of 48.56 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	79.99 GB
----------------------	----------



All Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved
0	0	0	0	0 B	8 GB	0 B	

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation												
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>												
Show 20 ▼ entries															
ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB
No data available in table															
Showing 0 to 0 of 0 entries															

RESULT:

Thus the Apache Hadoop installation was successfully completed.

Task 4.2: Find the Word Count using Map reduce

Aim:

To implement the word count with Map Reduce Using Apache Hadoop

Procedure:

1. Start the Docker engine and Hadoop server
2. Check the docker containers using the command
docker container ls
3. Download the jar file for the word count from the url
<https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-mapreduce-examples/2.7.1/hadoop-mapreduce-examples-2.7.1-sources.jar>
4. Copy the jar file, input1.txt into the tmp folder and enter into the bash mode

```
C:\Users\abc> docker cp hadoop-mapreduce-examples-2.7.1-sources.jar namenode:/tmp
```

```
C:\Users\abc> docker cp input1.txt namenode:/tmp/
```

Commands

Execute the following commands in bash mode to execute the jar file

```
ls
```

```
cd /tmp/
```

```
cat input1.txt
```

```
hdfs dfs -mkdir /user
```

```
hdfs dfs -mkdir /user/root
```

```
hdfs dfs -mkdir /user/root/input
```

```
hdfs dfs -put input1.txt /user/root/input/
```

```
hdfs dfs -cat /user/root/input/input1.txt
```

```
hadoop jar hadoop-mapreduce-examples-2.7.1-sources.jar  
org.apache.hadoop.examples.WordCount input output
```

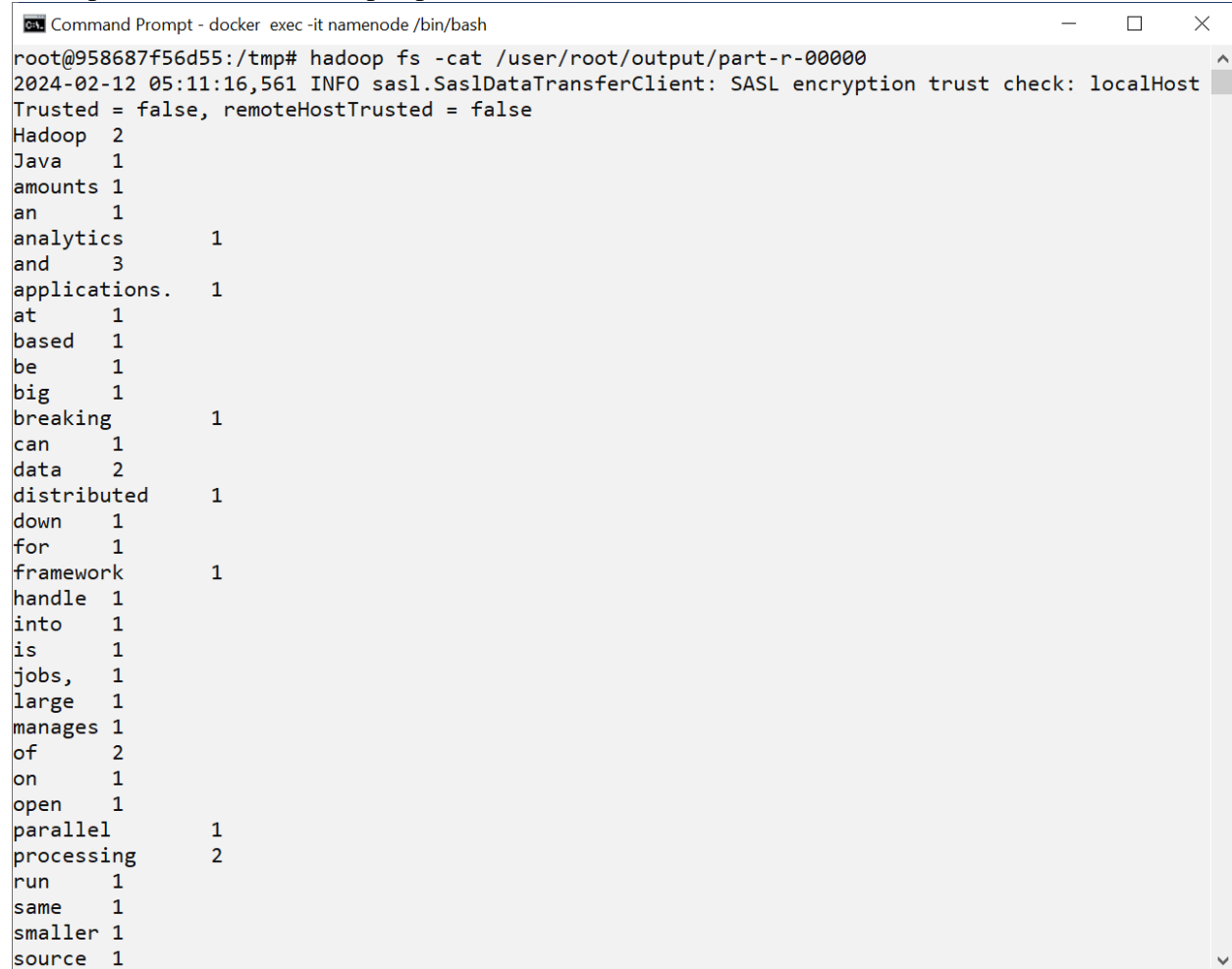
Input File: input1.txt

Hadoop is an open source framework based on Java that manages the storage and processing of large amounts of data for applications. Hadoop uses distributed storage and parallel processing to handle big data and analytics jobs, breaking workloads down into smaller workloads that can be run at the same time.

Output:

Execute the command in the bash mode

`hadoop fs -cat /user/root/output/part-r-00000`



```
Command Prompt - docker exec -it namenode /bin/bash
root@958687f56d55:/tmp# hadoop fs -cat /user/root/output/part-r-00000
2024-02-12 05:11:16,561 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhost
Trusted = false, remoteHostTrusted = false
Hadoop 2
Java 1
amounts 1
an 1
analytics 1
and 3
applications. 1
at 1
based 1
be 1
big 1
breaking 1
can 1
data 2
distributed 1
down 1
for 1
framework 1
handle 1
into 1
is 1
jobs, 1
large 1
manages 1
of 2
on 1
open 1
parallel 1
processing 2
run 1
same 1
smaller 1
source 1
```

Result:

Thus the Map reduce concept was implement for word count using the Hadoop and Docker engine.