Name: Harsh Chheda

Roll Number: 22-15405

Class: Msc. Computer Science (Part 2)

Subject: Big Data

Year: 2022-23

INDEX

Practical 1

Q1) Show existing Databases and create a new DB MSC.

→

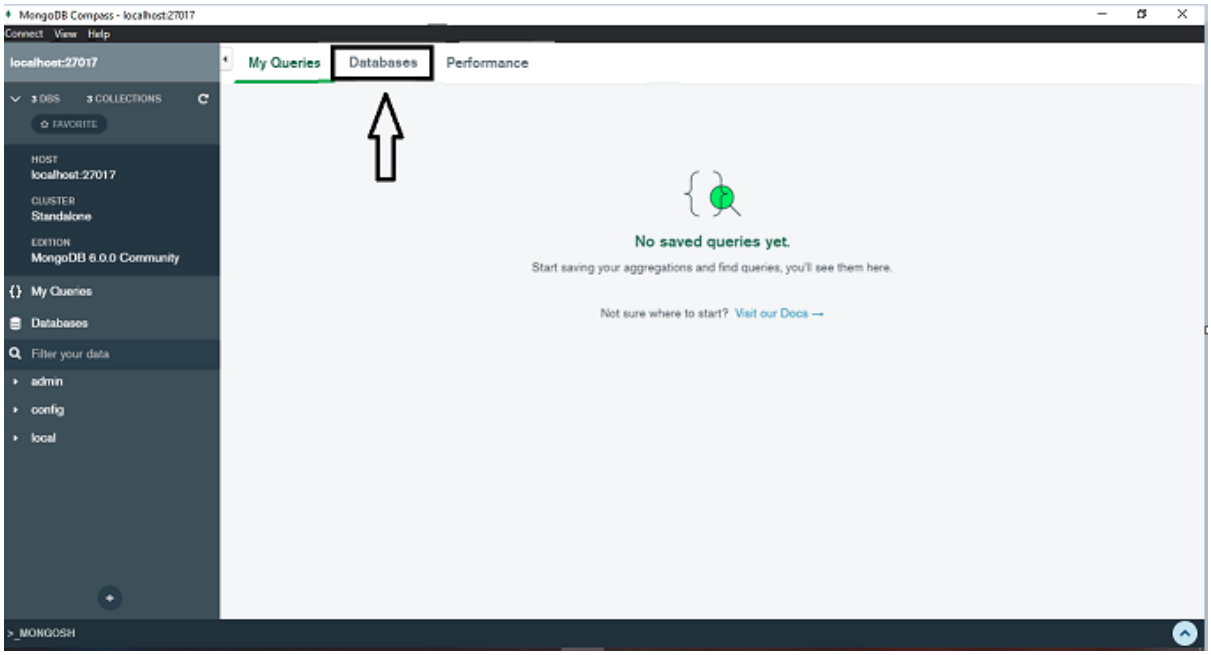
Showing existing Databases

```
show databases
```

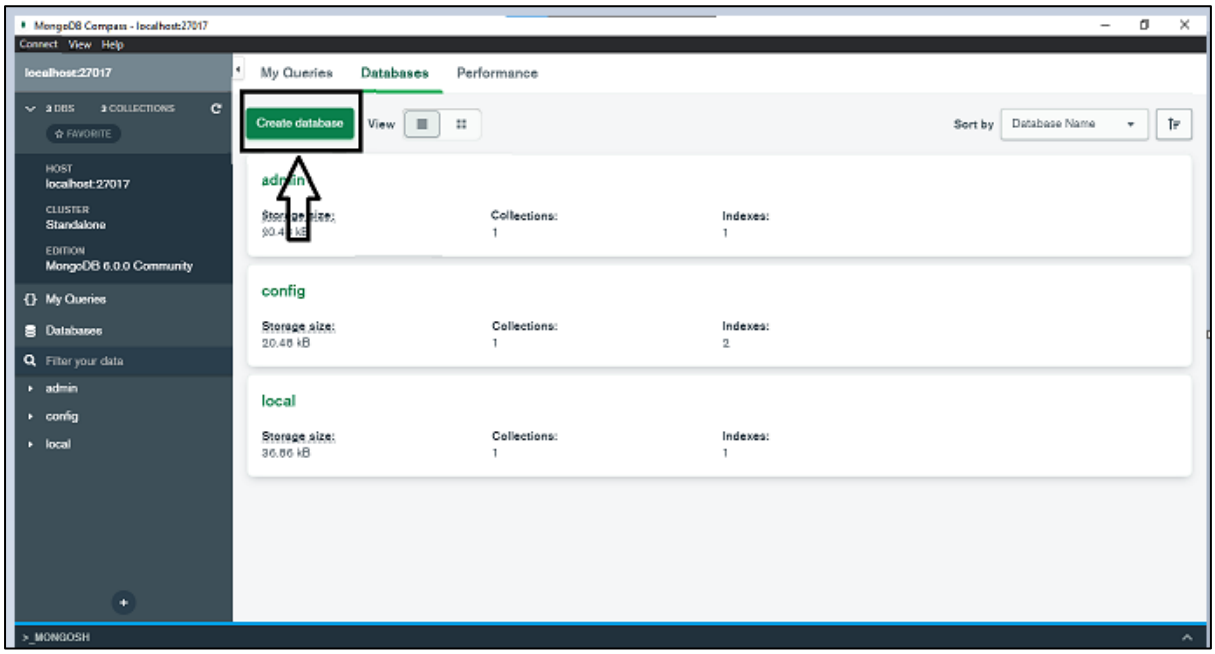
```
>_MONGOSH
> show databases
< admin   40.00 KiB
  config  60.00 KiB
  local   72.00 KiB
test>
```

Creating New Database

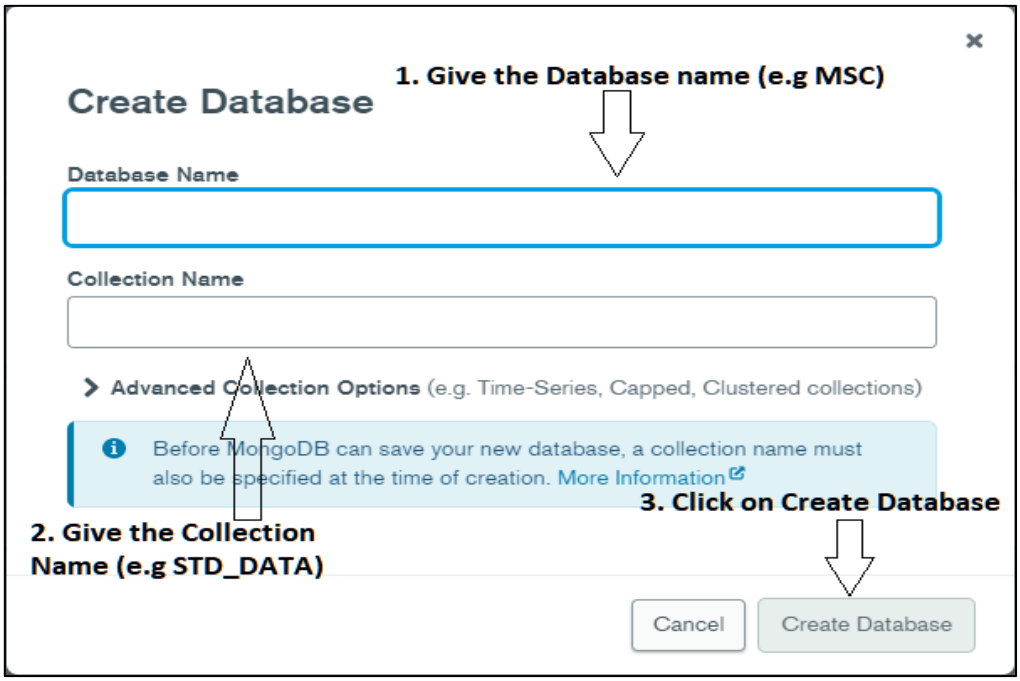
Step 1: Click on the Databases Tab



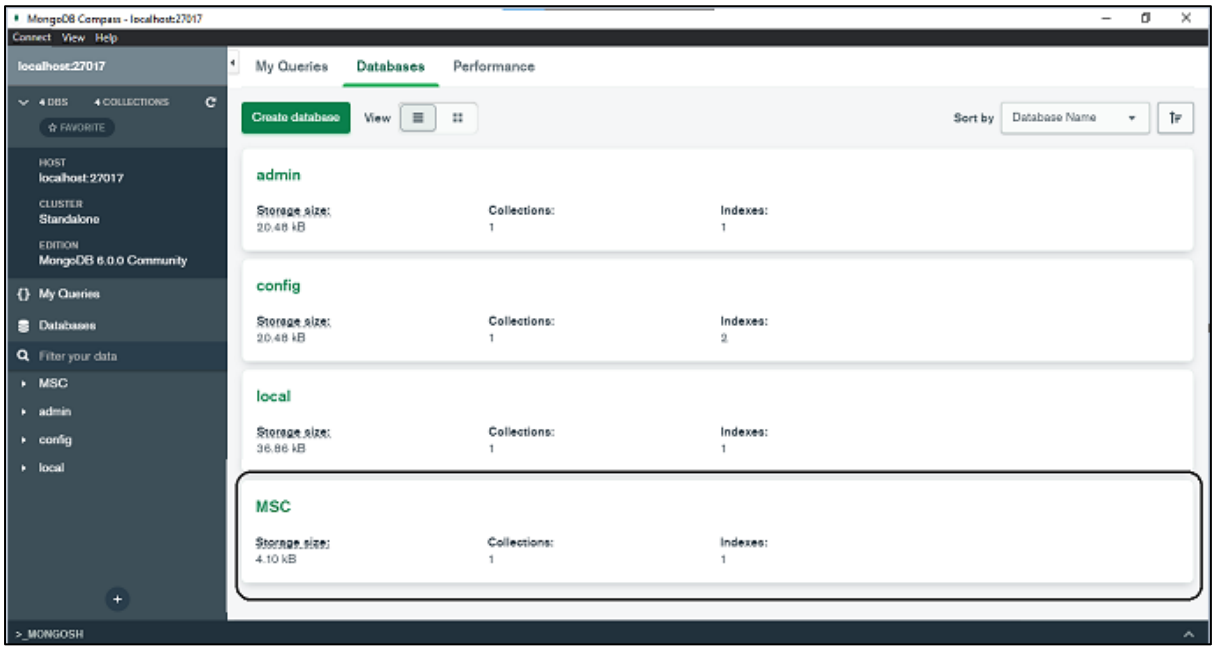
Step 2: Click on Create Database



Step 3: Assign the database name and the collection name and click on the Create Database



Step 4: After the Create Database you will be able to see the database



Switching database to MSC

```
use MSC
```

```
>_MONGOSH
> use MSC
< 'switched to db MSC'
MSC>
```

Q2. Create a new collection Students and add values and display all records.

→

Inserting Records into the collection STD_DATA

```
db.STD_DATA.insertMany([{Name: "Harsh", Age: 22, Hobby:
["Reading","Cricket"]},{Name: "Ashok", Age: 23, Hobby:
["Singing","Cricket"]},{Name: "ASHA", Age: 21, Hobby: ["Singing"]},{Name:
"John", Age: 22, Hobby: ["Reading","Painting"]}])
```

```
> MONGOSH
> use MSC
< 'switched to db MSC'
> db.STD_DATA.insertMany([{Name: "Harsh", Age: 22, Hobby: ["Reading","Cricket"]},{Name: "Ashok", Age: 23, Hobby: ["Singing","Cricket"]},{Name: "ASHA", Age: 21, Hobby: ["Sin
< { acknowledged: true,
    insertedIds:
     { '0': ObjectId("62df6bb1962590bd333bfe66"),
       '1': ObjectId("62df6bb1962590bd333bfe67"),
       '2': ObjectId("62df6bb1962590bd333bfe68"),
       '3': ObjectId("62df6bb1962590bd333bfe69") } }
MSC>
```

Displaying all the records

```
db.STD_DATA.find()
```

```
> db.STD_DATA.find()
< { _id: ObjectId("62df6bb1962590bd333bfe66"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Reading', 'Cricket' ] }
  { _id: ObjectId("62df6bb1962590bd333bfe67"),
    Name: 'Ashok',
    Age: 23,
    Hobby: [ 'Singing', 'Cricket' ] }
  { _id: ObjectId("62df6bb1962590bd333bfe68"),
    Name: 'ASHA',
    Age: 21,
    Hobby: [ 'Singing' ] }
  { _id: ObjectId("62df6bb1962590bd333bfe69"),
    Name: 'John',
    Age: 22,
    Hobby: [ 'Reading', 'Painting' ] }
MSC>
```

Q3. Display details of Ashok

→

```
db.STD_DATA.find({Name:"Ashok"})
```

```
>_MONGOSH
> db.STD_DATA.find({Name:"Ashok"})
< { _id: ObjectId("62df6bb1962590bd333bfe67"),
    Name: 'Ashok',
    Age: 23,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q4. Update age of John to 20 keep rest of the data same

→

Updating the record of John

```
db.STD_DATA.findOneAndUpdate({Name:"John"},{$set :{Age:20}})
```

```
>_MONGOSH
> db.STD_DATA.findOneAndUpdate({Name:"John"},{$set :{Age:20}})
< { _id: ObjectId("62df6e42962590bd333bfe6d"),
    Name: 'John',
    Age: 22,
    Hobby: [ 'Reading', 'Painting' ] }
MSC>
```

Displaying the updated record for John

```
db.STD_DATA.find({Name:"John"})
```

```
>_MONGOSH
> db.STD_DATA.find({Name:"John"})
< { _id: ObjectId("62df6e42962590bd333bfe6d"),
    Name: 'John',
    Age: 20,
    Hobby: [ 'Reading', 'Painting' ] }
MSC>
```

Q5. Update hobby of Harsh as Dancing instead of Reading.

→

Updating the record of Harsh

```
db.STD_DATA.findOneAndUpdate({"Name":"Harsh"}, { $set:
{"Hobby.$[element]":"Dancing"}},{ arrayFilters: [{ element: "Reading" }]})
```

```
>_MONGOSH
> db.STD_DATA.findOneAndUpdate({"Name":"Harsh"}, { $set: {"Hobby.$[element]":"Dancing"}},{ arrayFilters: [{ element: "Reading" }]})
< { _id: ObjectId("62df6e42962590bd333bfe6a"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Reading', 'Cricket' ] }
MSC>
```

Displaying Records of Harsh

```
>_MONGOSH
> db.STD_DATA.find({Name:"Harsh"})
< { _id: ObjectId("62df6e42962590bd333bfe6a"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Dancing', 'Cricket' ] }
MSC>
```

Q6. Display name whose age is 22.

→

```
db.STD_DATA.find({Age:20},{Name:1})
```

```
>_MONGOSH
> db.STD_DATA.find({Age:20},{Name:1})
< { _id: ObjectId("62df6e42962590bd333bfe6d"), Name: 'John' }
MSC>
```

Q7. Delete record of John.

→

```
db.STD_DATA.deleteOne({Name:"John"})
```

```
>_MONGOSH
> db.STD_DATA.deleteOne({Name:"John"})
< { acknowledged: true, deletedCount: 1 }
MSC>
```

Q8. Update Age of Ashok first occurrence as 19.

→

Updating Record

```
db.STD_DATA.findOneAndUpdate({Name:"Ashok"},{$set :{Age:19}})

db.STD_DATA.find({Name:"Ashok"})
```

```
>_MONGOSH
> db.STD_DATA.findOneAndUpdate({Name:"Ashok"},{$set :{Age:19}})
< { _id: ObjectId("62df6e42962590bd333bfe6b"),
    Name: 'Ashok',
    Age: 23,
    Hobby: [ 'Singing', 'Cricket' ] }
> db.STD_DATA.find({Name:"Ashok"})
< { _id: ObjectId("62df6e42962590bd333bfe6b"),
    Name: 'Ashok',
    Age: 19,
    Hobby: [ 'Singing', 'Cricket' ] }
  { _id: ObjectId("62df758a962590bd333bfe6e"),
    Name: 'Ashok',
    Age: 20,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q9. Update Age of Ashok, all Occurrences as 15.

→

```
db.STD_DATA.updateMany({Name:"Ashok"},{$set :{Age:15}})

db.STD_DATA.find({Name:"Ashok"})
```

```
>_MONGOSH
> db.STD_DATA.updateMany({Name:"Ashok"},{$set :{Age:15}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 2,
    modifiedCount: 2,
    upsertedCount: 0 }
> db.STD_DATA.find({Name:"Ashok"})
< { _id: ObjectId("62df6e42962590bd333bfe6b"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
  { _id: ObjectId("62df758a962590bd333bfe6e"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q10. Add mobile no. of Harsh.

→

```
db.STD_DATA.updateMany({Name:"Harsh"},{$set :{Mobile:9372685907}})

db.STD_DATA.find({Name:"Harsh"})
```

```
>_MONGOSH
> db.STD_DATA.updateMany({Name:"Harsh"},{$set :{Mobile:9372685907}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.STD_DATA.find({Name:"Harsh"})
< { _id: ObjectId("62df6e42962590bd333bfe6a"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Dancing', 'Cricket' ],
    Mobile: 9372685907 }
MSC>
```

Q11. Display Record whose age is 22 and hobby as Dancing.

→

```
db.STD_DATA.find({$and:[{Age:22},{Hobby: "Dancing"}]})
```

```
> db.STD_DATA.find({$and:[{Age:22},{Hobby: "Dancing"}]})
< { _id: ObjectId("62df6e42962590bd333bfe6a"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Dancing', 'Cricket' ],
    Mobile: 9372685907 }
MSC>
```

Q12. Display Record whose age is 15 or hobby as Singing

→

```
db.STD_DATA.find({$or: [{Age:15},{ Hobby:["Singing"]}]})
```

```
>_MONGOSH
> db.STD_DATA.find({$or: [{Age:15},{ Hobby:["Singing"]}]})
< { _id: ObjectId("62df6e42962590bd333bfe6b"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
  { _id: ObjectId("62df6e42962590bd333bfe6c"),
    Name: 'ASHA',
    Age: 21,
    Hobby: [ 'Singing' ] }
  { _id: ObjectId("62df758a962590bd333bfe6e"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q13. Display records from the collection by skipping first 2 records.

→

```
db.STD_DATA.find({}).skip(2).limit(2)
```

```
>_MONGOSH
> db.STD_DATA.find({}).skip(2).limit(2)
< { _id: ObjectId("62df6e42962590bd333bfe6c"),
    Name: 'ASHA',
    Age: 21,
    Hobby: [ 'Singing' ] }
  { _id: ObjectId("62df758a962590bd333bfe6e"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q14. Display records by sorting names.

→

```
db.STD_DATA.find({}).sort({name:-1})
```

```
>_MONGOSH
> db.STD_DATA.find({}).sort({name:-1})
< { _id: ObjectId("62df6e42962590bd333bfe6a"),
    Name: 'Harsh',
    Age: 22,
    Hobby: [ 'Dancing', 'Cricket' ],
    Mobile: 9372685907 }
  { _id: ObjectId("62df6e42962590bd333bfe6b"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
  { _id: ObjectId("62df6e42962590bd333bfe6c"),
    Name: 'ASHA',
    Age: 21,
    Hobby: [ 'Singing' ] }
  { _id: ObjectId("62df758a962590bd333bfe6e"),
    Name: 'Ashok',
    Age: 15,
    Hobby: [ 'Singing', 'Cricket' ] }
MSC>
```

Q15. Count the number of records into the collection

→

```
db.STD_DATA.find({}).count({})
```

```
>_MONGOSH
> db.STD_DATA.find({}).count({})
< 4
MSC>
```

Q16) Show existing Databases and create a new DB MSC.

→

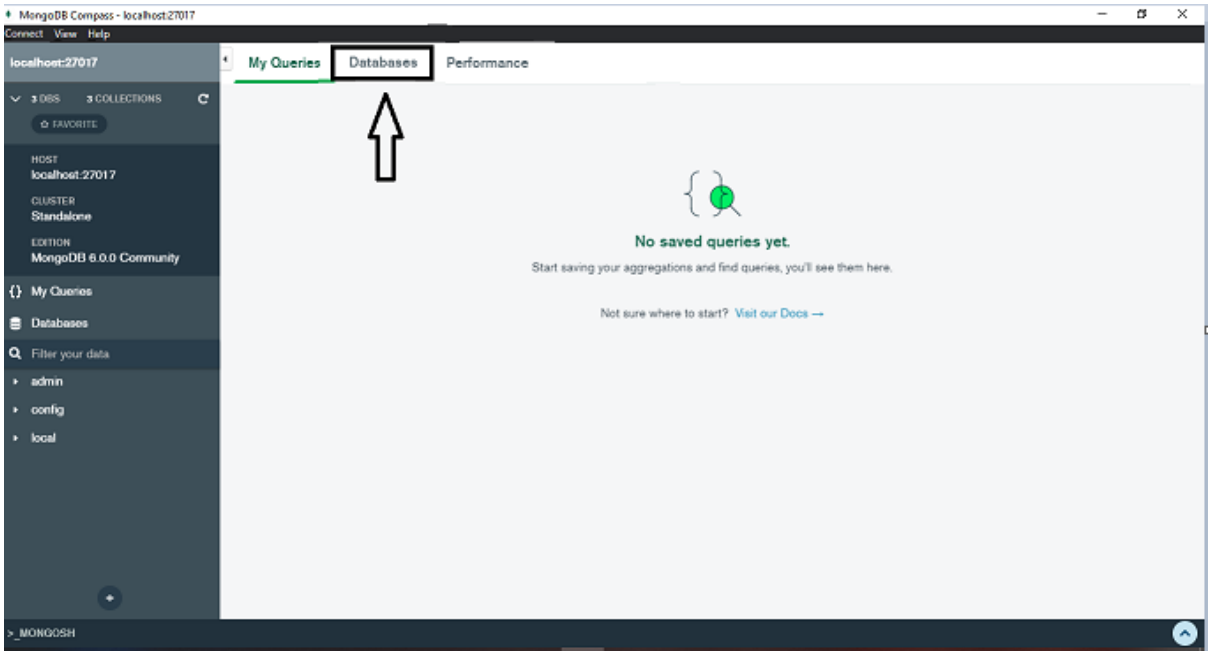
Showing existing Databases

```
show databases
```

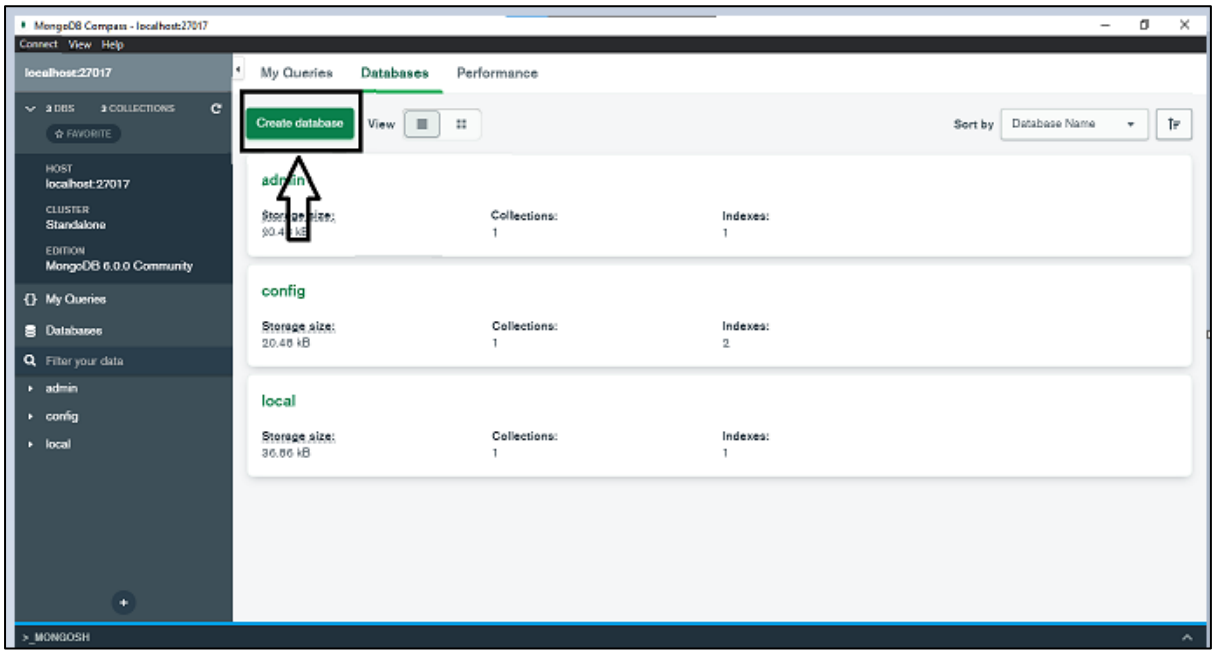
```
>_MONGOSH
> show databases
< admin   40.00 KiB
  config  60.00 KiB
  local   72.00 KiB
test >
```

Creating New Database

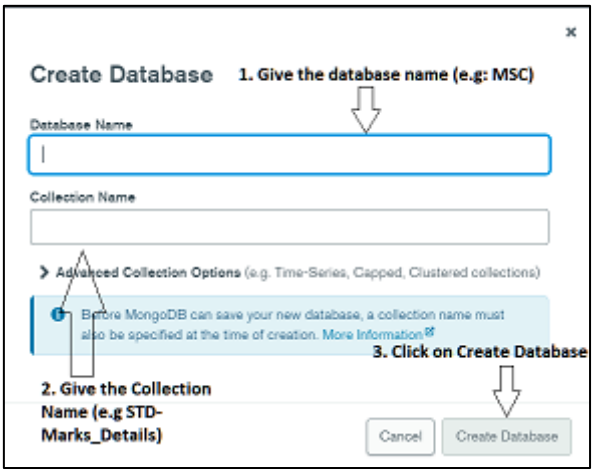
Step 1: Click on the Databases Tab



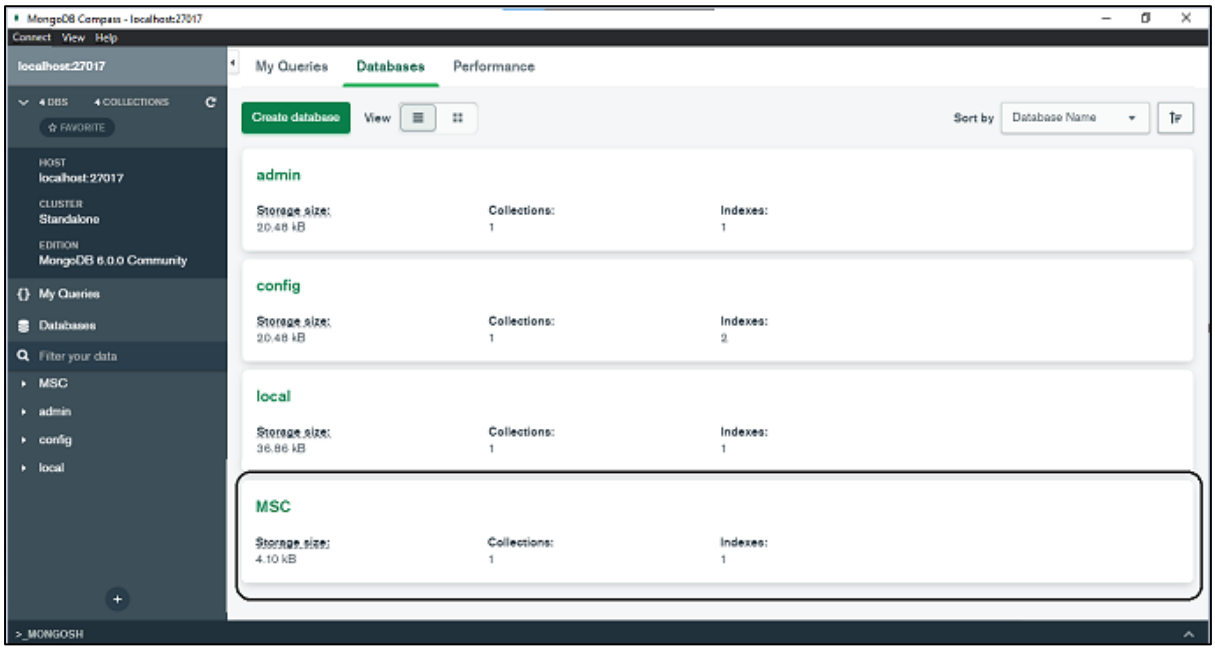
Step 2: Click on Create Database



Step 3: Assign the database name and the collection name and click on the Create Database



Step 4: After the Create Database you will be able to see the database



Switching database to MSC

```
use MSC
```

```
>_MONGOSH
> use MSC
< 'switched to db MSC'
MSC>
```

Q17) Create a MongoDB containing marks of students for subjects like English, Maths and Computer.

Insert 3 documents where

1) **First document is having marks of Maths and Computer**
2) **Second document is having marks of Maths and English**
3) **Third document is having marks of all subjects**

→

```
db.STD_MARKS_DETAILS.insertMany([{Name:"John",Roll_No:01,Subject:{'Maths':91,'
Computer':85}},{Name:"Tom",Roll_No:02,Subject:{'Maths':83,'English':45}},{Name
:'Bob',Roll_No:03,Subject:{'Maths':76,'English':70,'Computer':85}}])
```

```
>_MONGOSH
> db.STD_MARKS_DETAILS.insertMany([{Name:"John",Roll_No:01,Subject:{'Maths':91,'Computer':85}},{Name:"Tom",Roll_No:02,Subject:{'Maths':83,'English':45}},{Name:'Bob',Roll_No
< { acknowledged: true,
  insertedIds:
   { '0': ObjectId("62e0b55705c69448b5713918"),
     '1': ObjectId("62e0b55705c69448b5713919"),
     '2': ObjectId("62e0b55705c69448b571391a") } }
MSC>
```

Q18) Update marks of 1st document English as 85.

→

```
db.STD_MARKS_DETAILS.updateOne({Name:'John'},{$set:{'Subject.English':85}})

db.STD_MARKS_DETAILS.find({Name:'John'})
```

```
>_MONGOSH
> db.STD_MARKS_DETAILS.updateOne({Name:'John'},{$set:{'Subject.English':85}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.STD_MARKS_DETAILS.find({Name:'John'})
< { _id: ObjectId("62e0b55705c69448b5713918"),
    Name: 'John',
    Roll_No: 1,
    Subject: { Maths: 91, Computer: 85, English: 85 } }
MSC>
```

Q19) Retrieve does contain marks of English as 85 and Maths as 91.

→

```
db.STD_MARKS_DETAILS.find({$and:[{'Subject.Maths':91},{'Subject.English':85}]}
)
```

```
>_MONGOSH
> db.STD_MARKS_DETAILS.find({$and:[{'Subject.Maths':91},{'Subject.English':85}]})
< { _id: ObjectId("62e0b55705c69448b5713918"),
    Name: 'John',
    Roll_No: 1,
    Subject: { Maths: 91, Computer: 85, English: 85 } }
MSC>
```

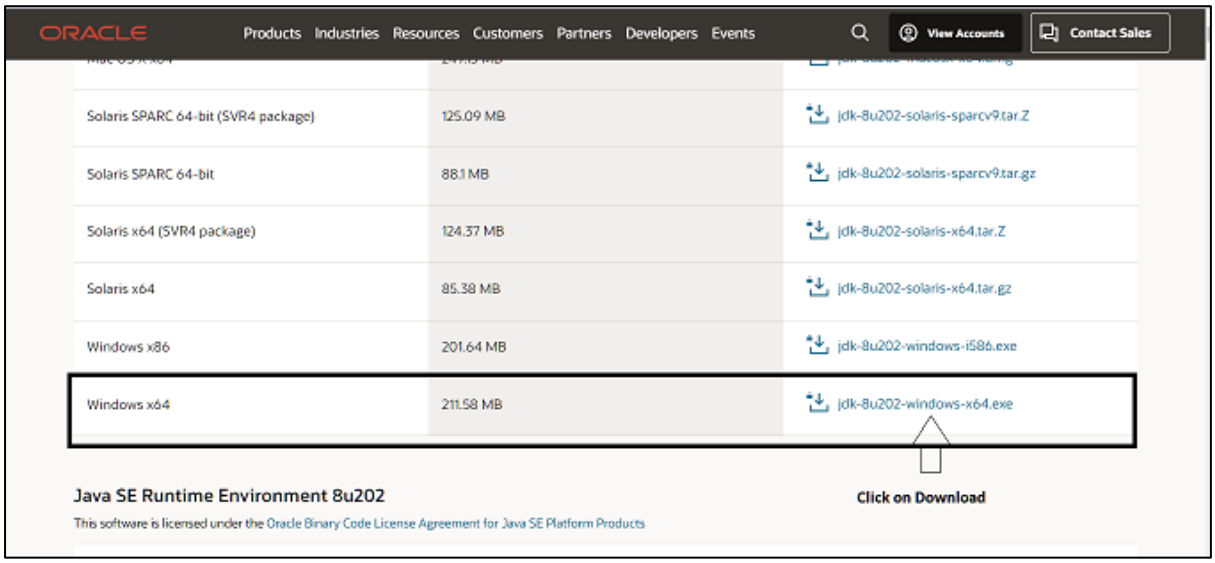
Practical 2

Aim: Installation of Hadoop and java in windows.

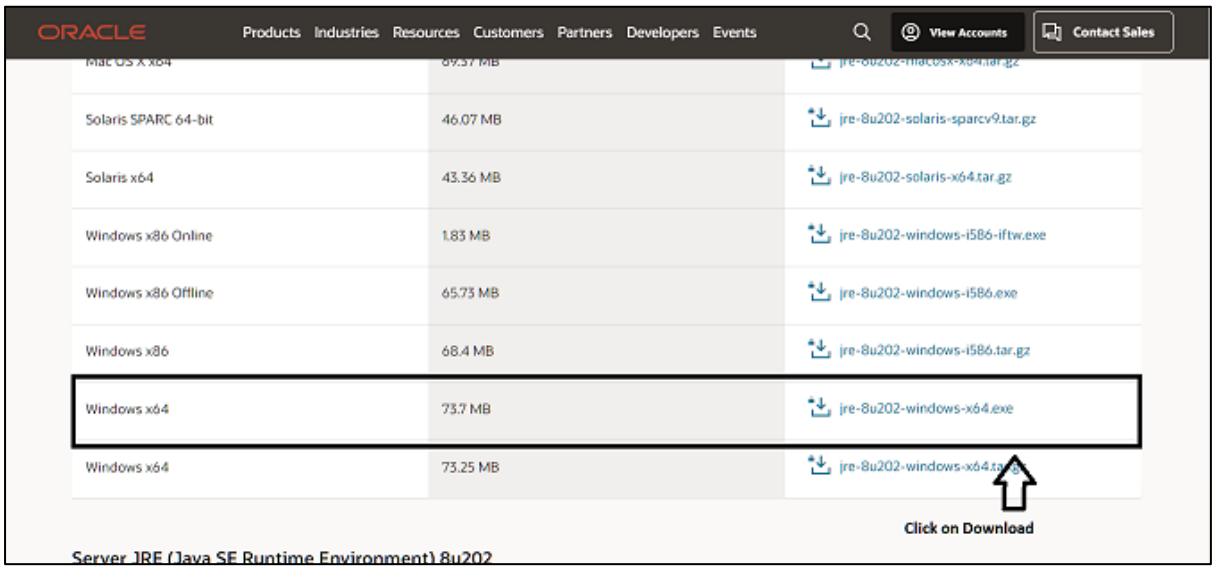
→

Installation on java jdk and jre

Step 1: Download SE Development Kit from the given link
https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html



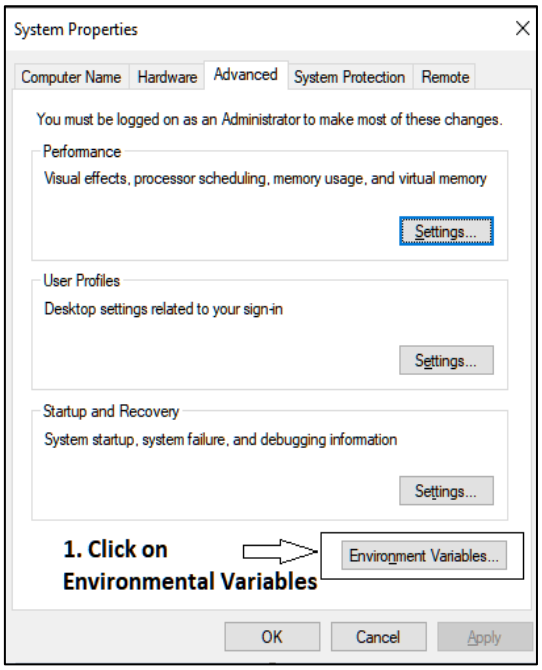
Step 2: Download Java SE Runtime Environment 8u202 from the given link
https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html



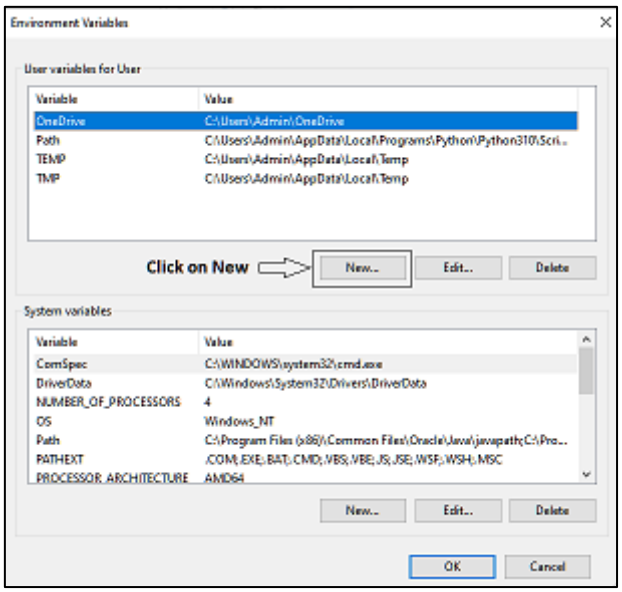
Setting up the JDK in Environmental Variable

Step 1: Start → Edit the system environment variables

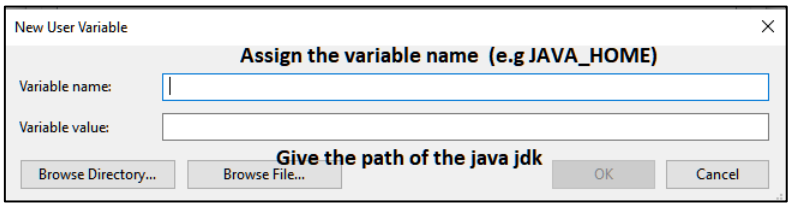
Step 2: Click on the environment Variables

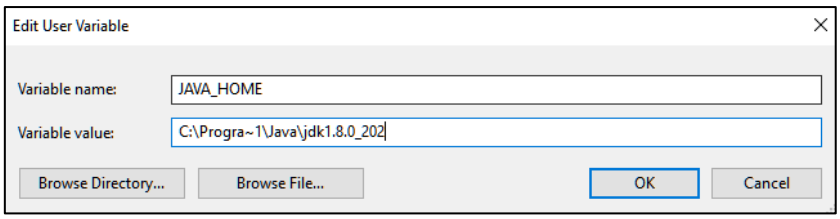


Step 3: Create the new user variables

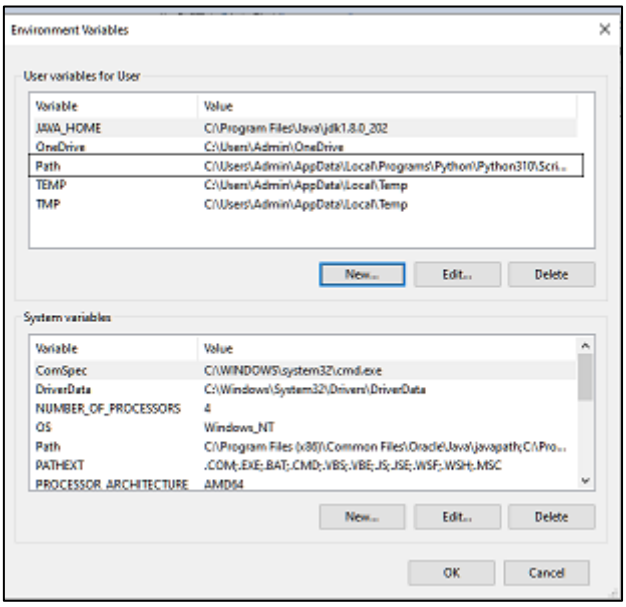


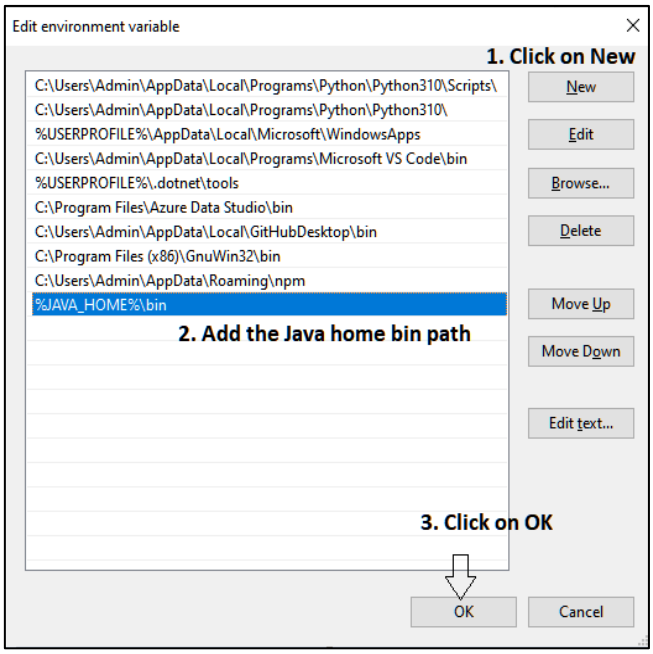
Step 4: Give the Variable Name and java jdk path (NOTE: Rename the path C:\Program files\Java\jdk1.8.0_202 to C:\Progra~1\Java\jdk1.8.0_202)



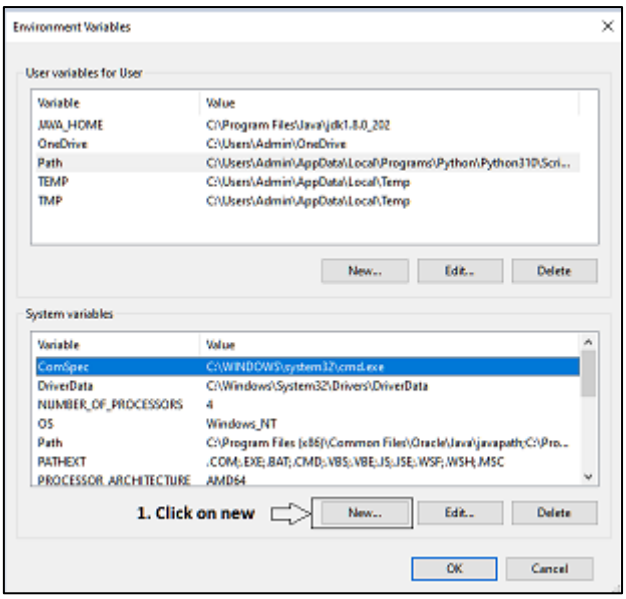


Step 5: Add the Variable into the Path

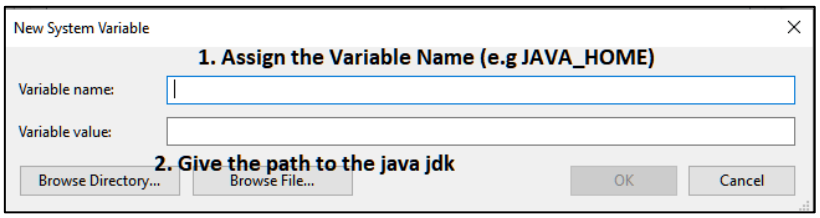


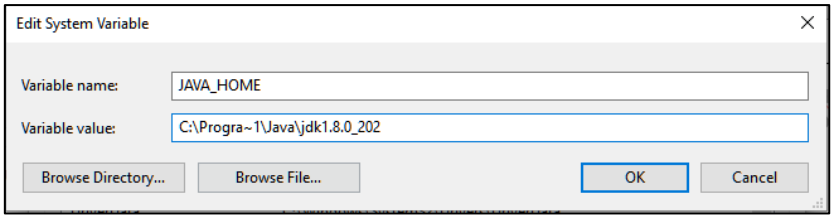


Step 6: Create the new system variables

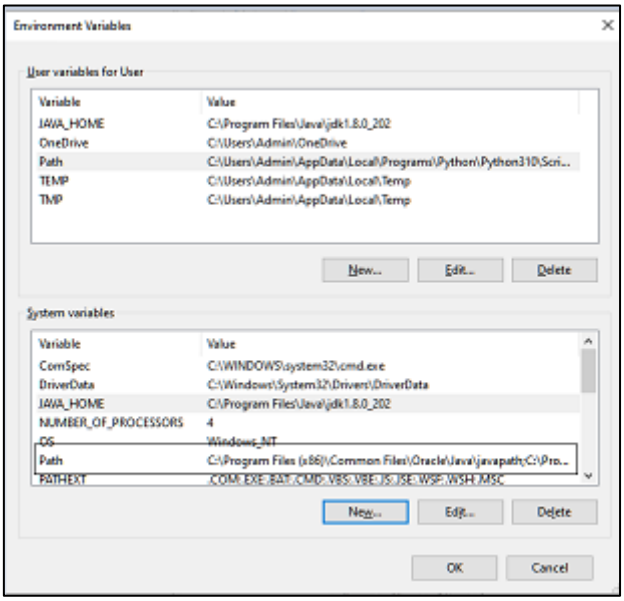


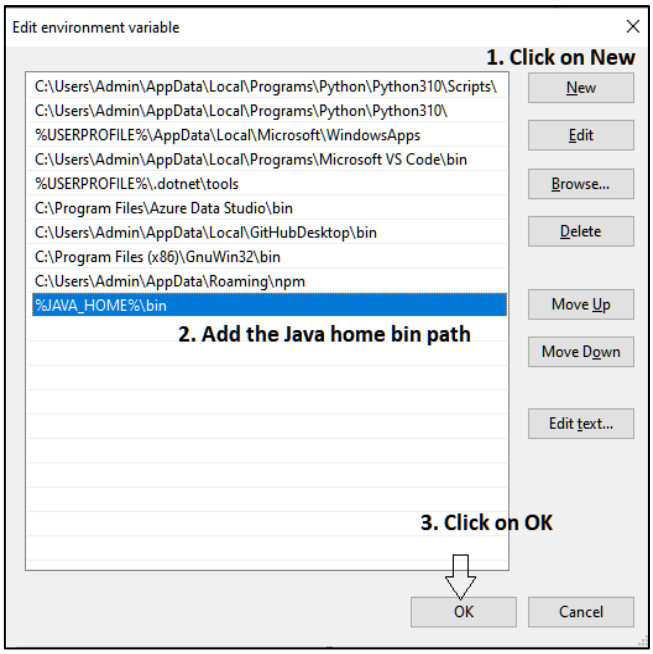
Step 7: Give the Variable Name and java jdk path (NOTE: Rename the path C:\Program files\Java\jdk1.8.0_202 to C:\Progra~1\Java\jdk1.8.0_202)





Step 8: Add the Variable into the Path

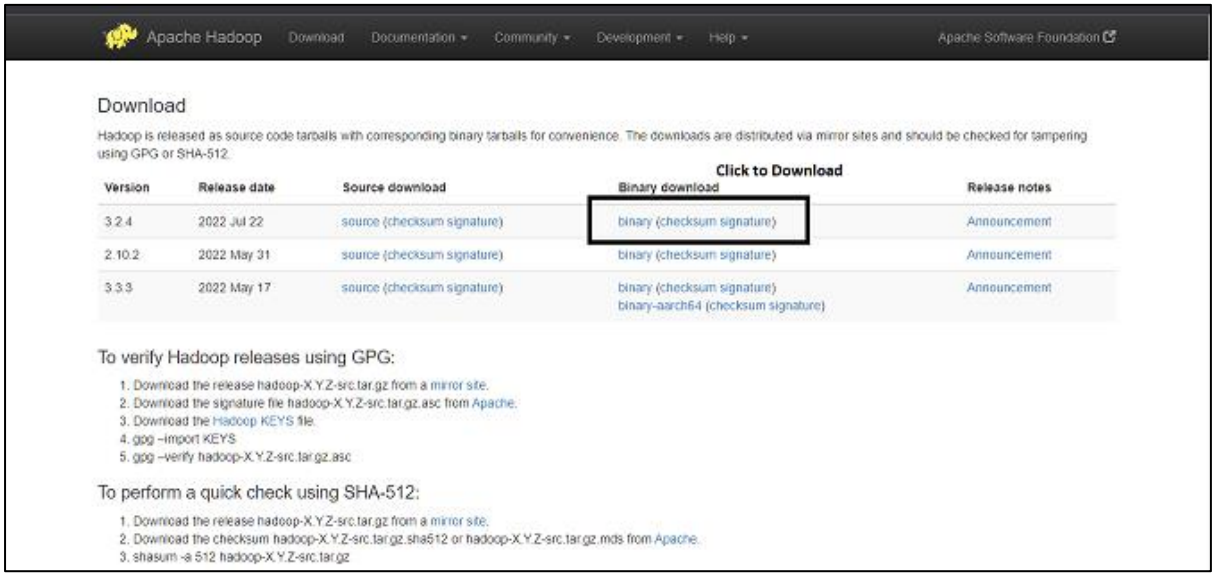




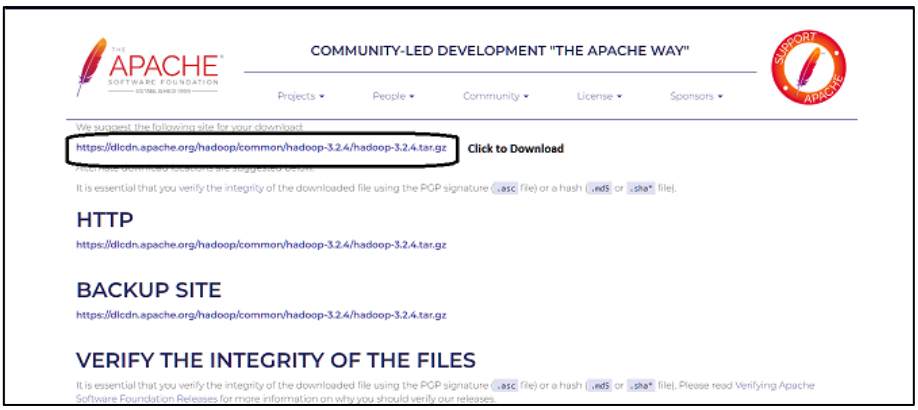
Installation of Hadoop

Step 1: Download Apache Hadoop from the given link and click on the binary

https://hadoop.apache.org/releases.html



Step 2: Click the link to download the zip

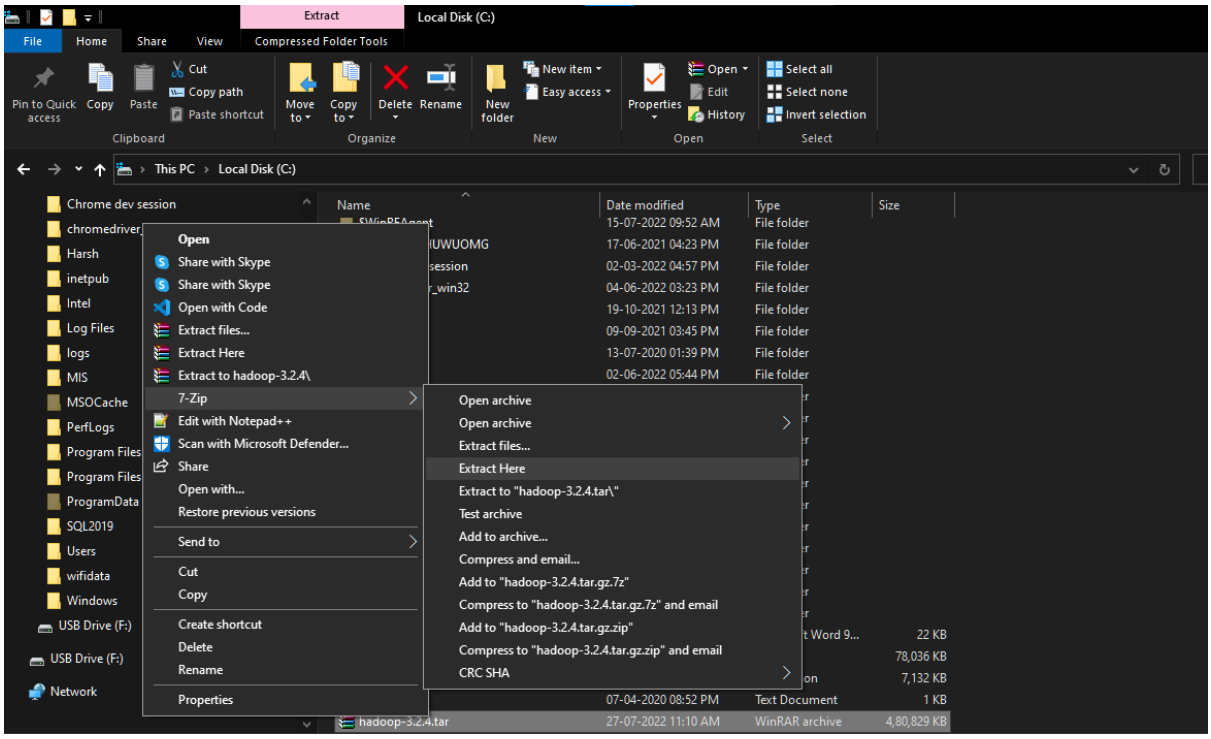


Installation of the 7zip

Step 1: Download the 7zip and install into the system https://www.7-zip.org/download.html

Extracting Hadoop zip using 7Zip

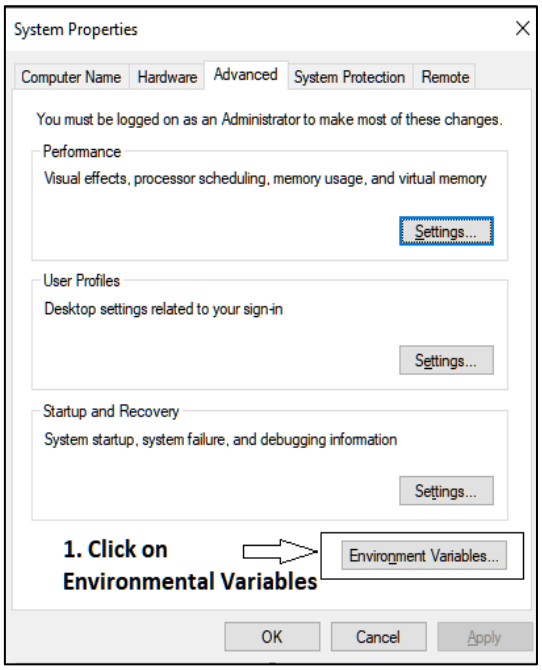
Step 1: Right Click on the zip → Click on 7-zip → Extract Here



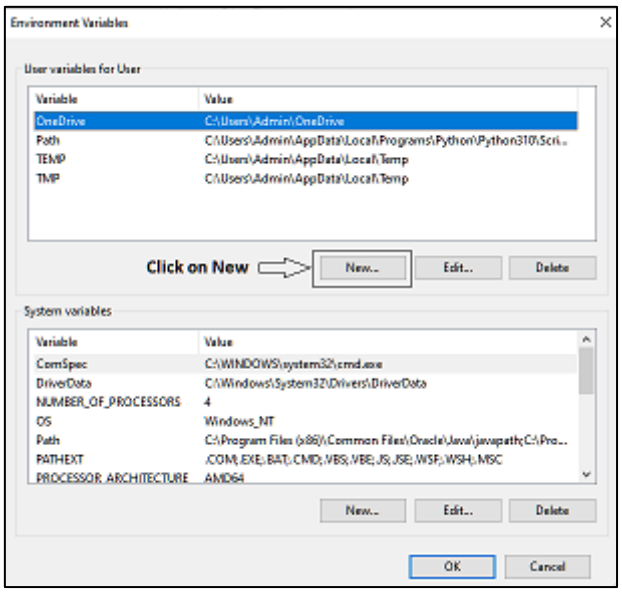
Setting up the Hadoop in Environmental Variable

Step 1: Start → Edit the system environment variables

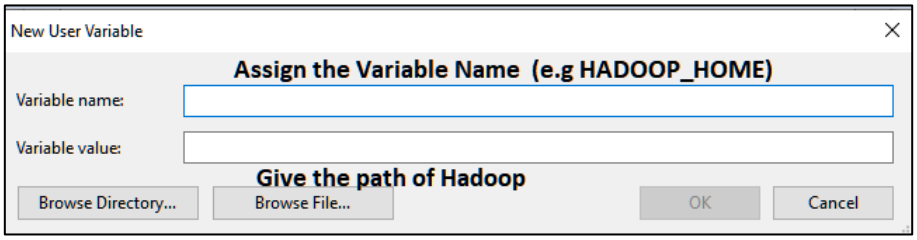
Step 2: Click on the environment Variables

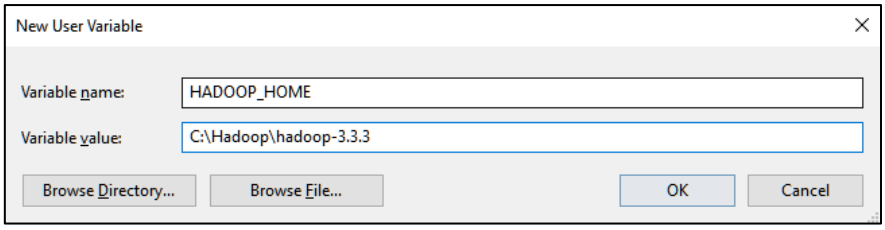


Step 3: Create the new user variables

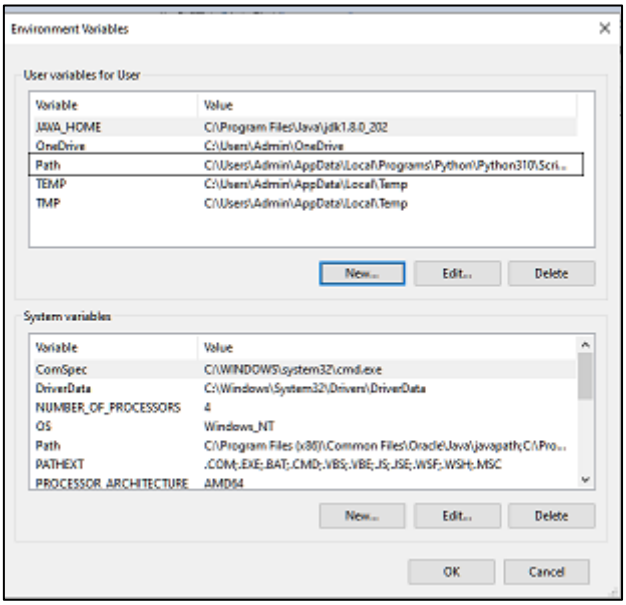


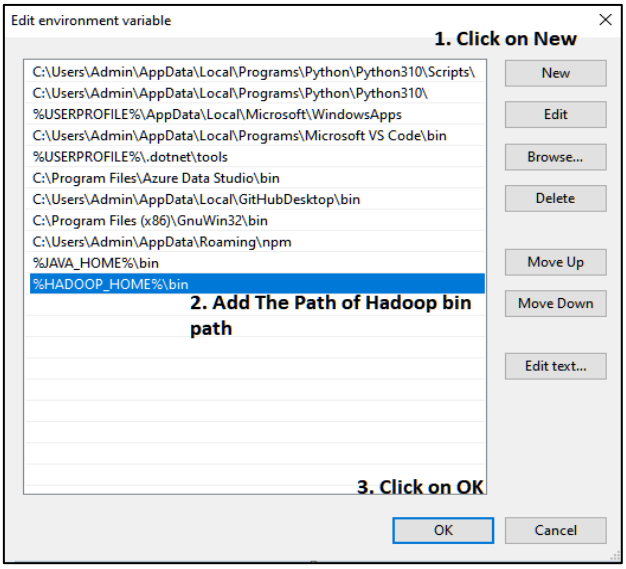
Step 4: Give the Variable Name and java Hadoop path



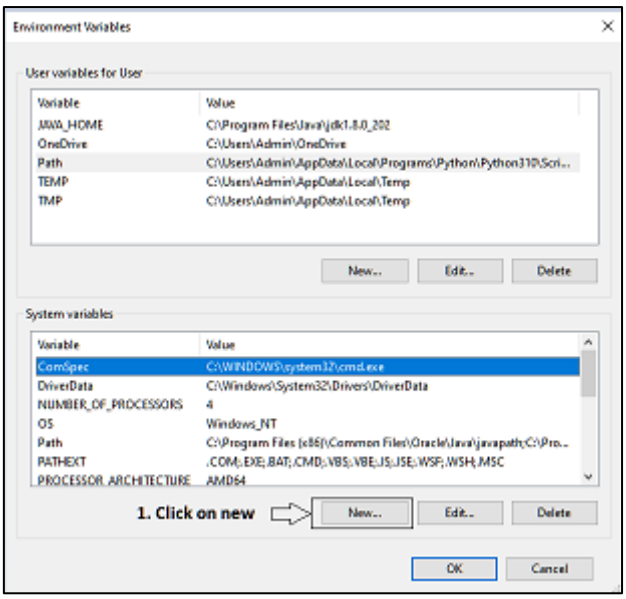


Step 5: Add the Variable into the Path

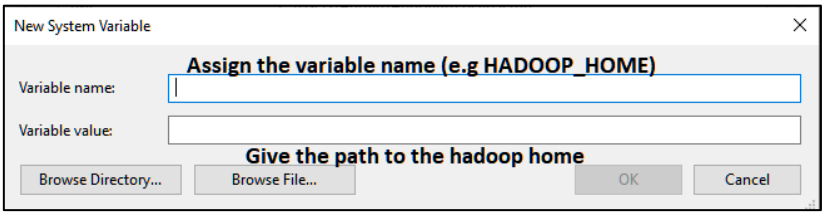


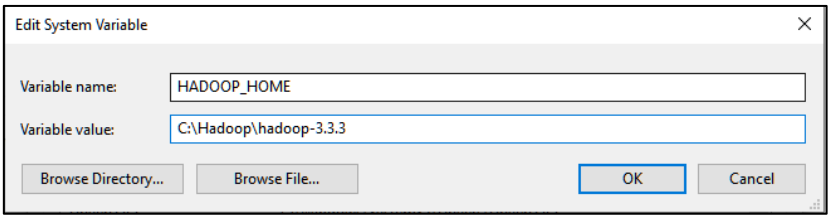


Step 6: Create the new system variables

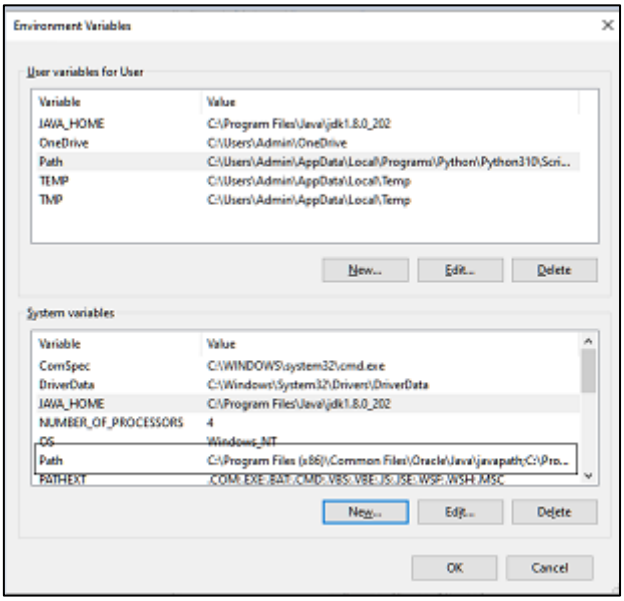


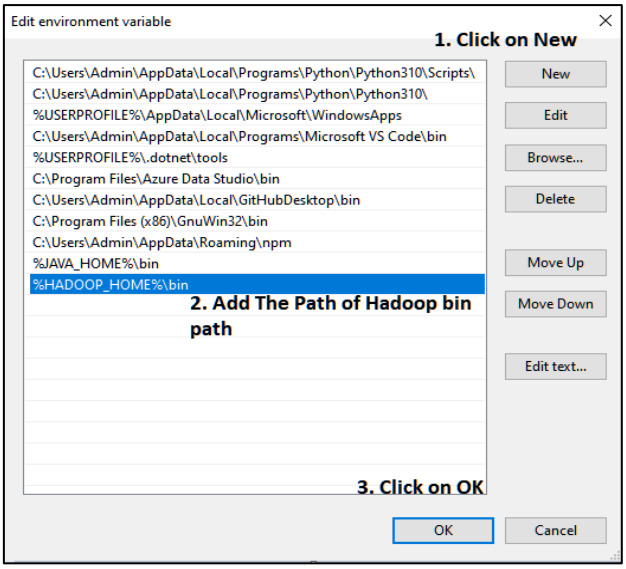
Step 7: Give the Variable Name and java Hadoop path





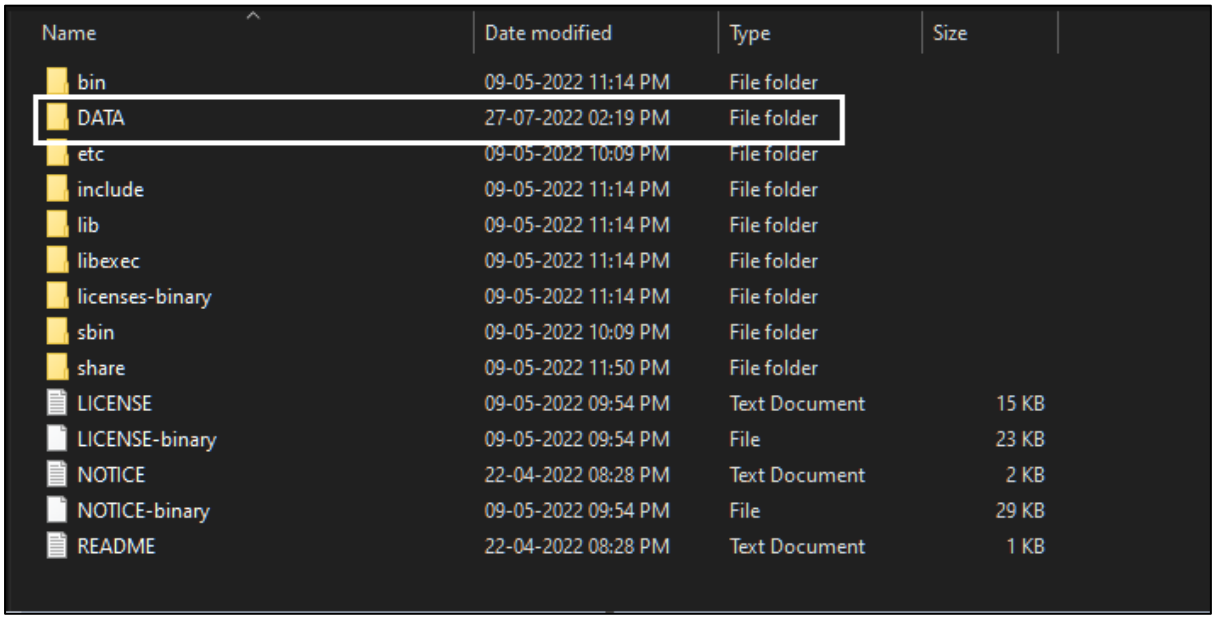
Step 8: Add the Variable into the Path





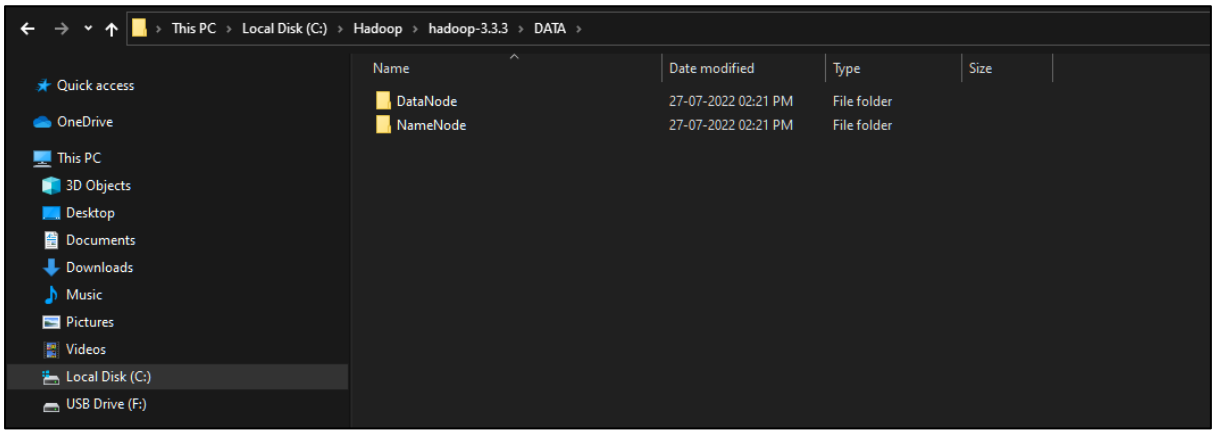
Setting up Hadoop

Step 1: Create the new Folder with the name DATA



Name	Date modified	Type	Size
bin	09-05-2022 11:14 PM	File folder	
DATA	27-07-2022 02:19 PM	File folder	
etc	09-05-2022 10:09 PM	File folder	
include	09-05-2022 11:14 PM	File folder	
lib	09-05-2022 11:14 PM	File folder	
libexec	09-05-2022 11:14 PM	File folder	
licenses-binary	09-05-2022 11:14 PM	File folder	
sbin	09-05-2022 10:09 PM	File folder	
share	09-05-2022 11:50 PM	File folder	
LICENSE	09-05-2022 09:54 PM	Text Document	15 KB
LICENSE-binary	09-05-2022 09:54 PM	File	23 KB
NOTICE	22-04-2022 08:28 PM	Text Document	2 KB
NOTICE-binary	09-05-2022 09:54 PM	File	29 KB
README	22-04-2022 08:28 PM	Text Document	1 KB

Step 2: Go inside the data folder and create 2 folder i.e. NameNode and DataNode



Step 3: open the file **hdfs-site.xml** C:\Hadoop\hadoop-3.3.3\etc\hadoop

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>


<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
<name>dfs.namenode.name.dir</name>
<value>C:\Hadoop\hadoop-3.3.3\DATA\NameNode</value>
</property>

<property>
<name>dfs.datanode.data.dir</name>
<value>C:\Hadoop\hadoop-3.3.3\DATA\DataNode</value>
</property>

<property>
  <name>dfs.permissions</name>
  <value>false</value>
</property>
</configuration>
```

C:\Hadoop\hadoop-3.3.3\etc\hadoop\hdfs-site.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>


<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
<name>dfs.namenode.name.dir</name>
<value>C:\Hadoop\hadoop-3.3.3\DATA\NameNode</value>
</property>

<property>
<name>dfs.datanode.data.dir</name>
<value>C:\Hadoop\hadoop-3.3.3\DATA\DataNode</value>
</property>

<property>
  <name>dfs.permissions</name>
  <value>false</value>
</property>
</configuration>
```

Step 4: open the file **core-site.xml** C:\Hadoop\hadoop-3.3.3\etc\hadoop

```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.
-->
```

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

<configuration>
<property>
<name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
<!-- <value>hdfs://localhost:9820</value> -->
</property>
</configuration>
```

hdfs-site.xml core-site.xml

core-site.xml

```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.default.name</name>
  <value>hdfs://localhost:9000</value>
<!-- <value>hdfs://localhost:9820</value> -->
</property>
</configuration>
```

Step 5: open the file **mapred-site.xml** C:\Hadoop\hadoop-3.3.3\etc\hadoop

```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.cluster.local.dir</name>
<value>${hadoop.tmp.dir}/mapred/local</value>
</property>

</configuration>
```

hdfs-site.xml | core-site.xml | mapred-site.xml

mapred-site.xml

```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.cluster.local.dir</name>
<value>${hadoop.tmp.dir}/mapred/local</value>
</property>

</configuration>
```

Step 6: open the file **yarn-site -site.xml** C:\Hadoop\hadoop-3.3.3\etc\hadoop

```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>

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

```xml
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>
```

hdfs-site.xml | core-site.xml | mapred-site.xml | yarn-site.xml

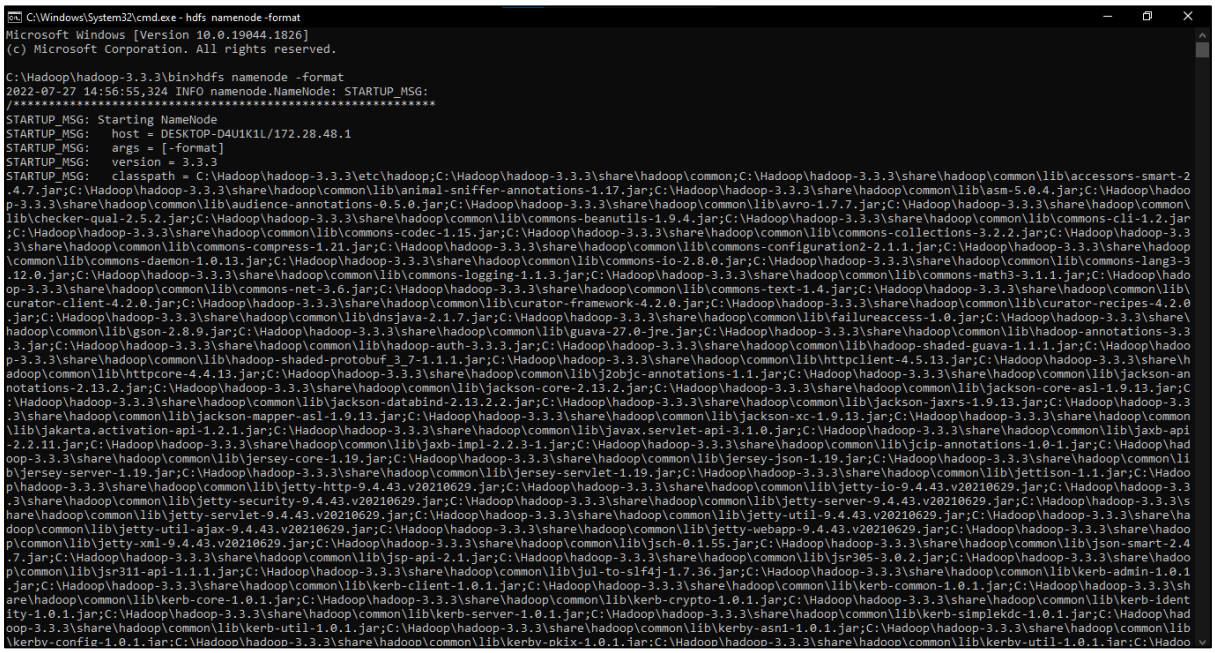
yarn-site.xml

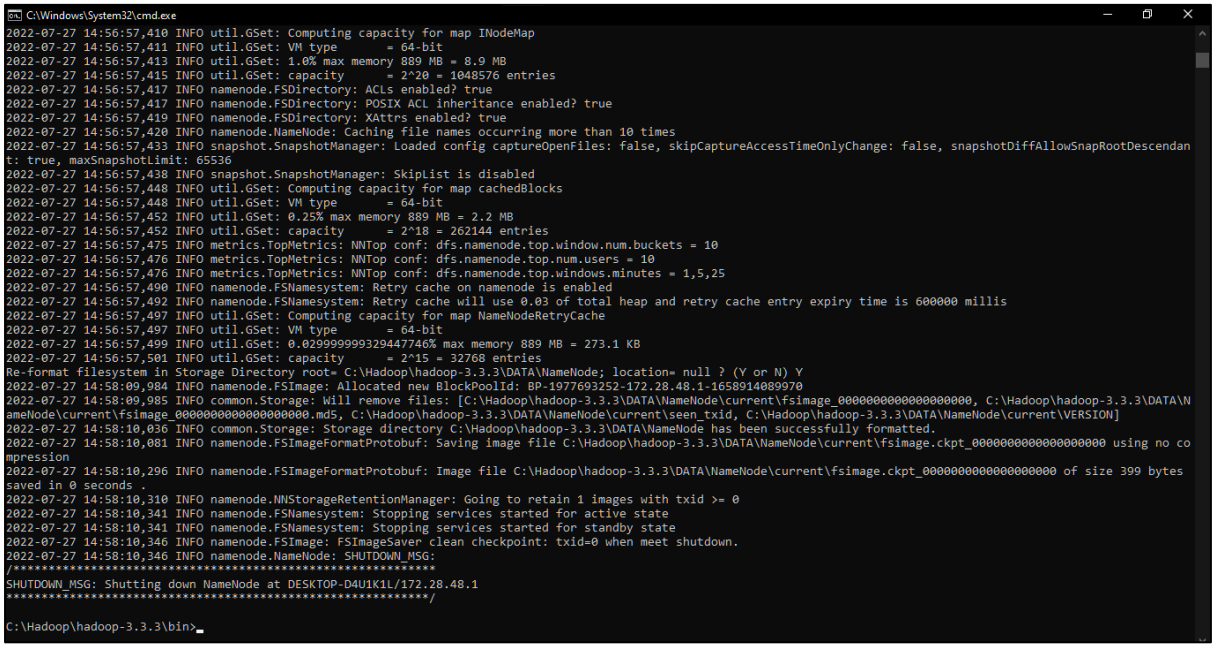
```xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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>

<!-- Site specific YARN configuration properties -->
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>
```

Step 7: Go to GitHub https://github.com/Selfgrowth/Apache-hadoop-3.1.1-winutils and download the bin folder and replace all the files with the C:\Hadoop\hadoop-3.3.3\bin

Step 8: Open the command prompt and change the location to C:\Hadoop\hadoop-3.3.3\bin

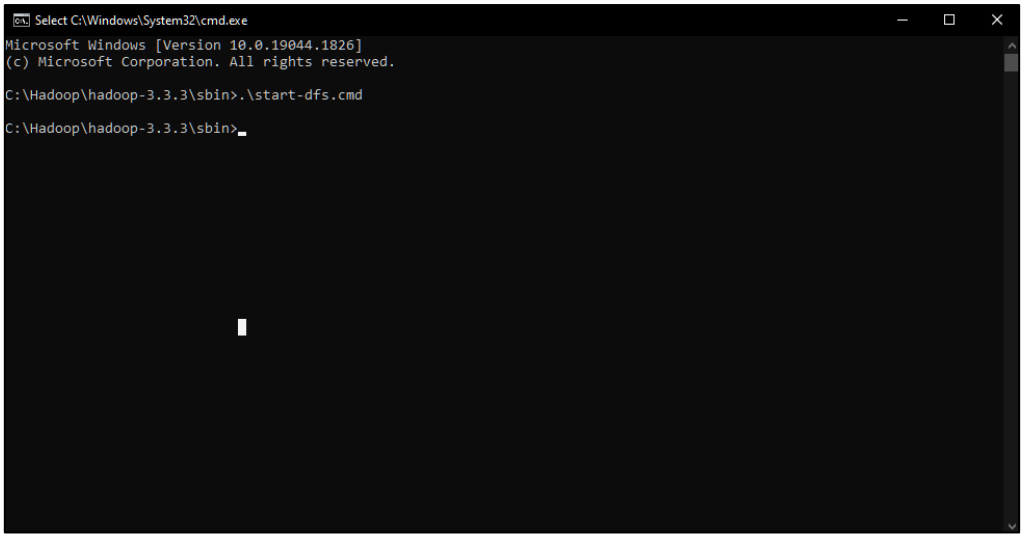
Step 9: hdfs namenode -format



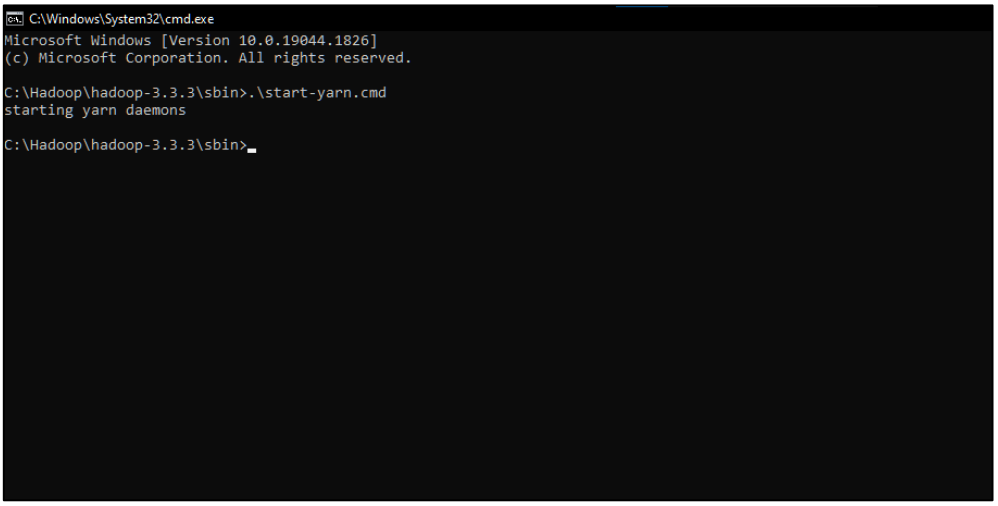


Step 10: Go to sbin folder. Type cmd in the address bar and 2 different open command prompt

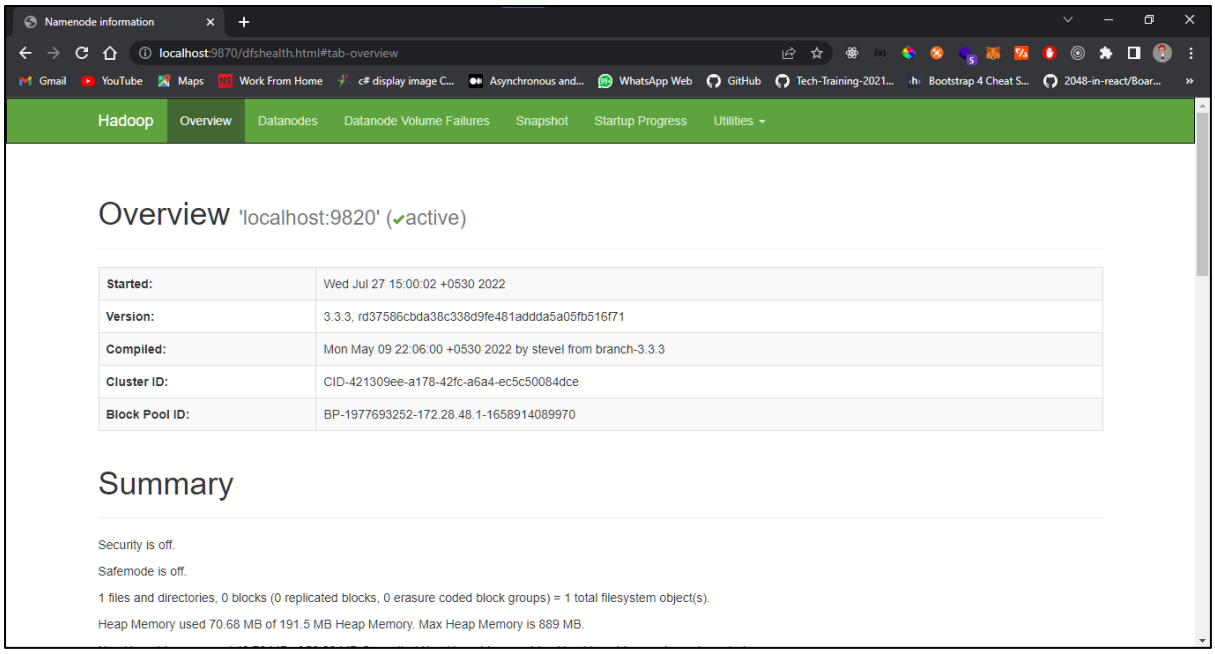
First cmd: **.\start-dfs.cmd**



Second cmd: **.\start-yarn.cmd**



Step 11: now open chrome or any other browser and type http://localhost:9870/



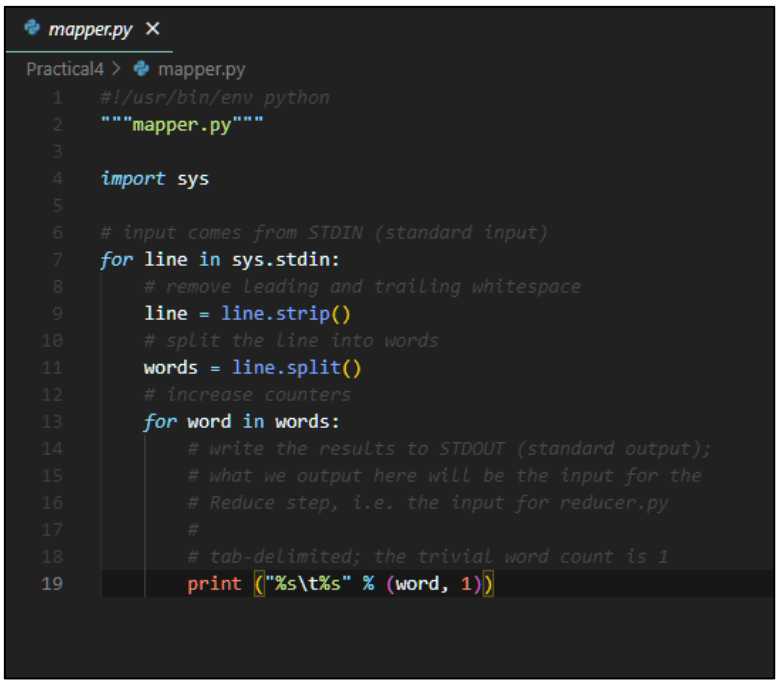
Practical 3

Aim: Write an Hadoop MapReduce Program in Python

→

Create the mapper.py

```python
#!/usr/bin/env python
"""mapper.py"""

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    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
        #
        # tab-delimited; the trivial word count is 1
        print ("%s\t%s" % (word, 1))
```



Create the reducer.py

```python
#!/usr/bin/env python
"""reducer.py"""

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from 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:
            # 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))
```

reducer.py

Practical4 > reducer.py

```python
#!/usr/bin/env python
"""reducer.py"""

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from 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:
            # 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))
```

Running mapper and reducer without Hadoop HDFS

Step 1: Open **Command Prompt** where the mapper.py and reducer.py is located

PROBLEMS OUTPUT TERMINAL JUPYTER COMMENTS DEBUG CONSOLE cmd

```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

F:\Practicals\BigData>cd Practical4

F:\Practicals\BigData\Practical4>
```

Step 2: To Execute the program create one file in the same location with name sample.txt

sample.txt

Practical4 > sample.txt

```
foo foo quux labs foo bar quux
```

Step 3: Now run the following command to get the output

```
type .\sample.txt | python .\mapper.py | sort |python .\reducer.py
```

```
PROBLEMS   OUTPUT   TERMINAL   JUPYTER   COMMENTS   DEBUG CONSOLE

F:\Practicals\BigData\Practical4>type .\sample.txt | python .\mapper.py | sort |python .\reducer.py
bar     1
foo     3
labs    1
quux    2

F:\Practicals\BigData\Practical4>
```

Running the Python Code on Hadoop

Step 1: Download example input data

We will use three eBooks from Project Gutenberg for this example:

- The Outline of Science, Vol. 1 (of 4) by J. Arthur Thomson
- The Notebooks of Leonardo Da Vinci
- Ulysses by James Joyce

Download each eBook as text files in `Plain Text UTF-8` encoding and store the files in a local temporary directory of choice.

Copy local example data to HDFS

Before we run the actual MapReduce job, we must first copy the files from our local file system to Hadoop's HDFS.

Step 1: Open Command Prompt in Administration Mode and change the present working directory to the C:\Hadoop\hadoop-3.3.3\sbin

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..

C:\Windows> cd ..

C:\> cd Hadoop

C:\Hadoop>cd hadoop-3.3.3

C:\Hadoop\hadoop-3.3.3> cd sbin

C:\Hadoop\hadoop-3.3.3\sbin>
```

Step 2: Now run the command **.\start-all.cmd**

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..

C:\Windows> cd ..

C:\> cd Hadoop

C:\Hadoop>cd hadoop-3.3.3

C:\Hadoop\hadoop-3.3.3> cd sbin

C:\Hadoop\hadoop-3.3.3\sbin>.\start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Hadoop\hadoop-3.3.3\sbin>
```

Step3: Now change the present working directory to **C:\Hadoop\hadoop-3.3.3\bin** and run the command

hadoop dfs –copyFromLocal 'path of the downloaded sample file' 'path to store on the hdfs'

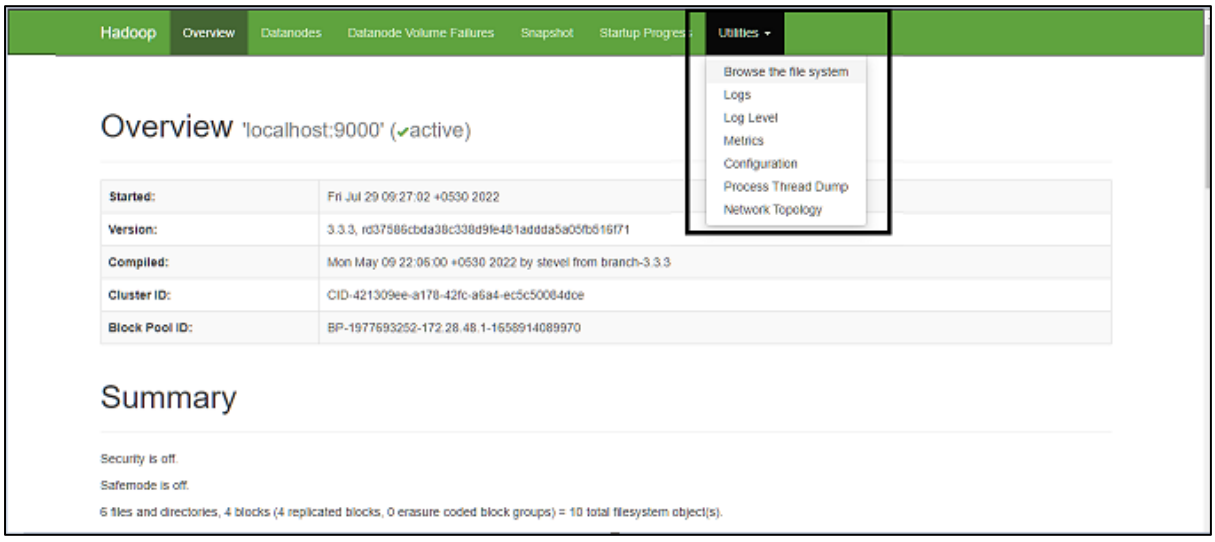
```
hadoop dfs -copyFromLocal
"F:\Practicals\BigData\Practical4\Data"  hdfs://localhost:9000/Harsh
```

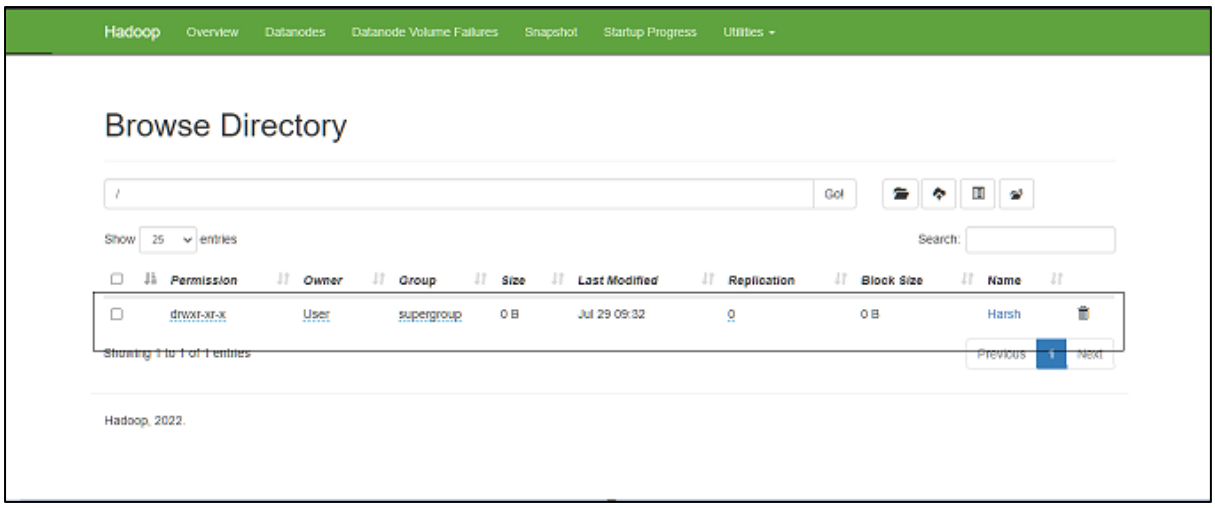
```
Administrator: Command Prompt

C:\Hadoop\hadoop-3.3.3\sbin>cd ..

C:\Hadoop\hadoop-3.3.3> cd bin

C:\Hadoop\hadoop-3.3.3\bin> hadoop dfs -copyFromLocal "F:\Practicals\BigData\Practical4\Data"  hdfs://localhost:9000/Harsh
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

C:\Hadoop\hadoop-3.3.3\bin>
```

```
hadoop dfs -ls /Harsh
```

```
Administrator: Command Prompt
C:\Hadoop\hadoop-3.3.3\bin>hadoop dfs -ls /Harsh
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Found 4 items
-rw-r--r--   3 User supergroup    1586331 2022-07-29 09:32 /Harsh/4300-0.txt
-rw-r--r--   3 User supergroup    1428909 2022-07-29 09:32 /Harsh/5000-8.txt
-rw-r--r--   3 User supergroup     674565 2022-07-29 09:32 /Harsh/pg20417.txt
-rw-r--r--   3 User supergroup         30 2022-07-29 09:32 /Harsh/sample.txt

C:\Hadoop\hadoop-3.3.3\bin>
```

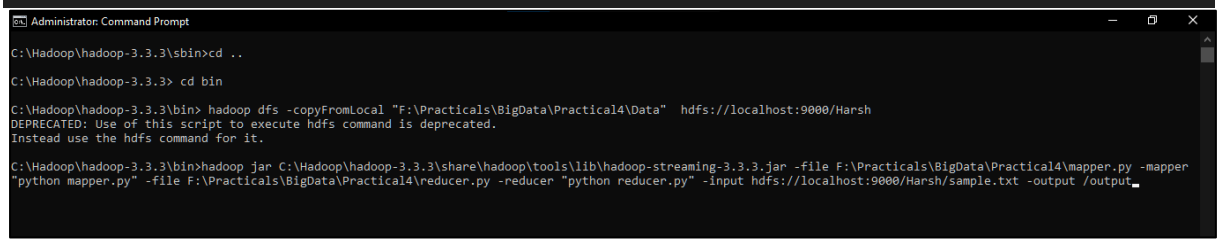
Step 4: To check the files are uploaded to the Hadoop HDFS the visit http://localhost:9870/dfshealth.html#tab-overview → go to utilities in the navigation bar and click on the Browse the file system

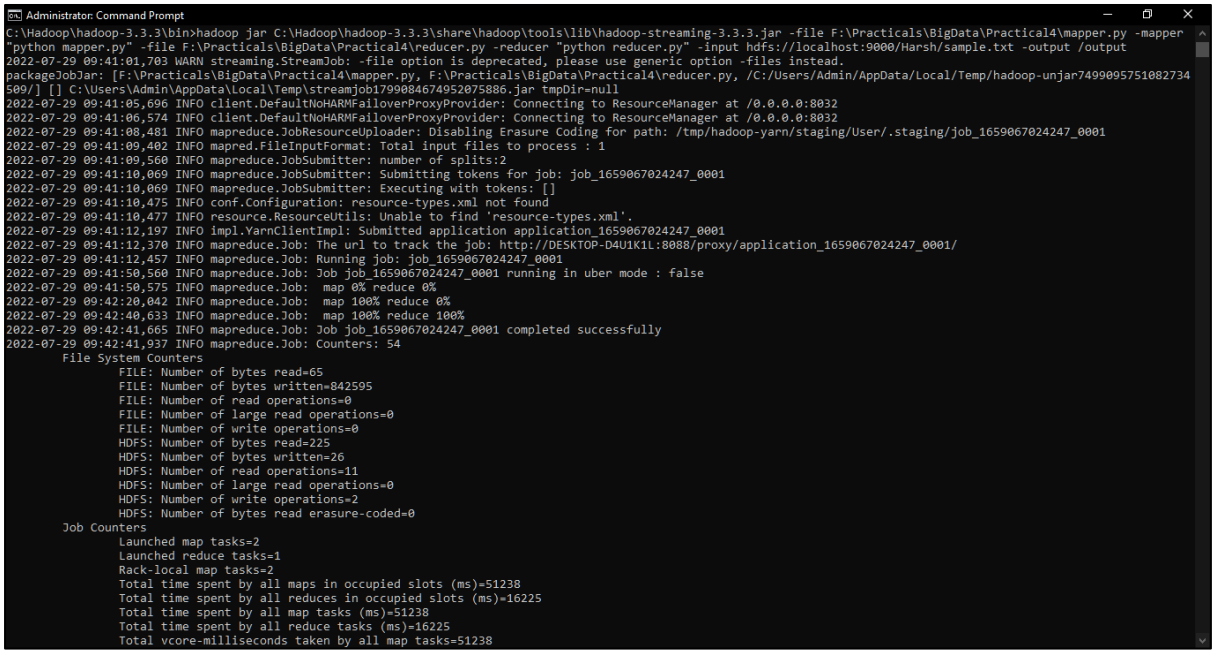




Step 5: Run the MapReduce job

```
hadoop jar C:\Hadoop\hadoop-3.3.3\share\hadoop\tools\lib\hadoop-streaming-3.3.3.jar -file F:\Practicals\BigData\Practical4\mapper.py -mapper "python mapper.py" -file F:\Practicals\BigData\Practical4\reducer.py -reducer "python reducer.py" -input hdfs://localhost:9000/Harsh/sample.txt -output /output
```





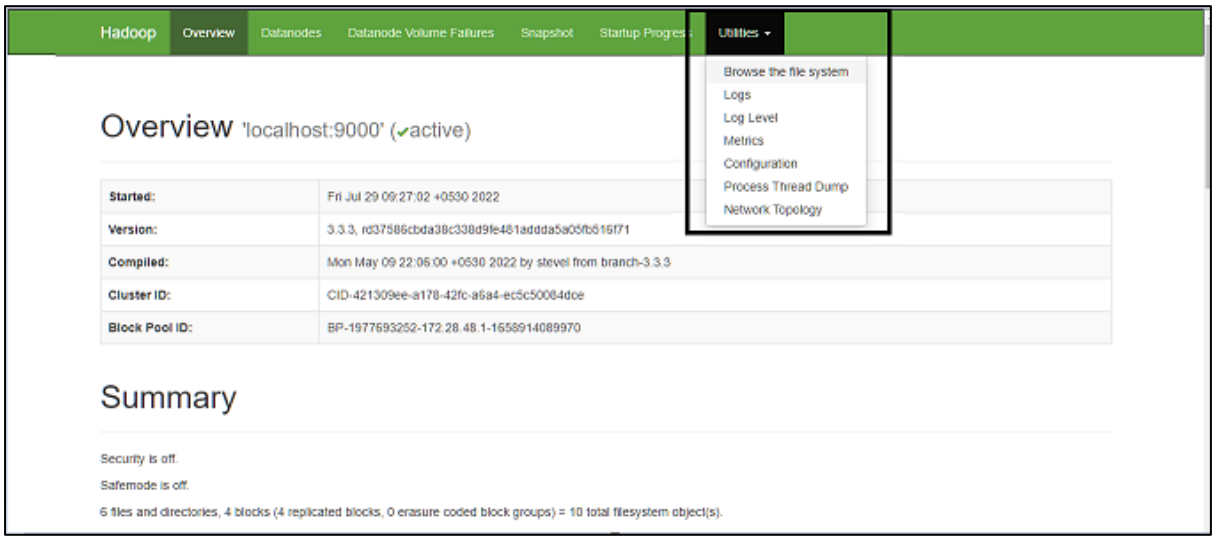
```
Administrator: Command Prompt
                Total vcore-milliseconds taken by all map tasks=51238
                Total vcore-milliseconds taken by all reduce tasks=16225
                Total megabyte-milliseconds taken by all map tasks=52467712
                Total megabyte-milliseconds taken by all reduce tasks=16614400
        Map-Reduce Framework
                Map input records=1
                Map output records=7
                Map output bytes=45
                Map output materialized bytes=71
                Input split bytes=180
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=71
                Reduce input records=7
                Reduce output records=4
                Spilled Records=14
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=368
                CPU time spent (ms)=4246
                Physical memory (bytes) snapshot=752508928
                Virtual memory (bytes) snapshot=1077448704
                Total committed heap usage (bytes)=630718464
                Peak Map Physical memory (bytes)=271990784
                Peak Map Virtual memory (bytes)=379629568
                Peak Reduce Physical memory (bytes)=209387520
                Peak Reduce Virtual memory (bytes)=321359872
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=45
        File Output Format Counters
                Bytes Written=26
2022-07-29 09:42:41,940 INFO streaming.StreamJob: Output directory: /output

C:\Hadoop\hadoop-3.3.3\bin>
```

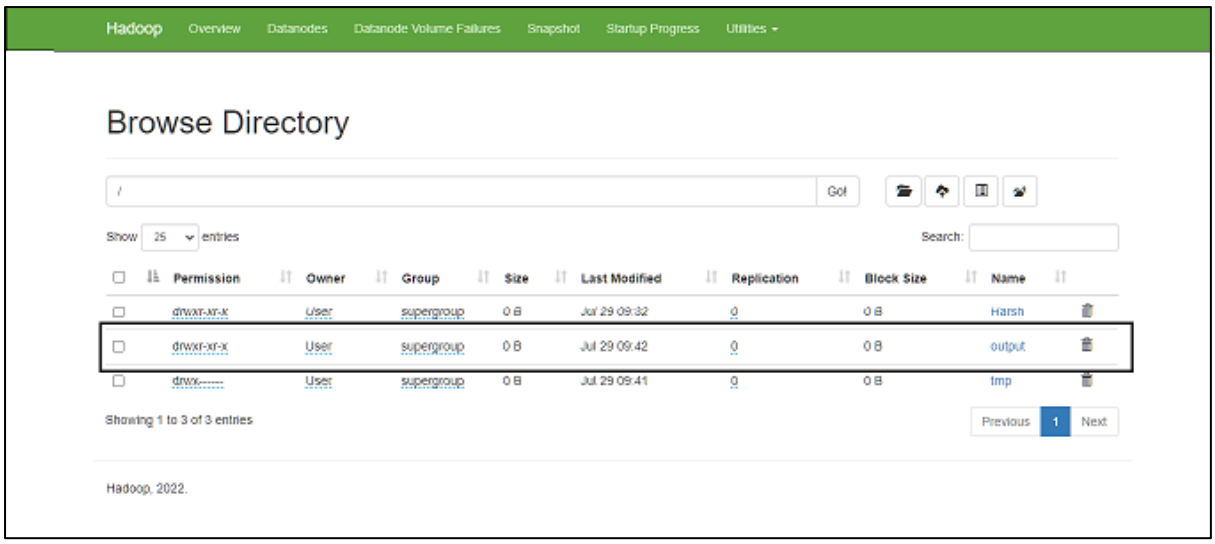
Step 6: Check if the result is successfully stored in HDFS directory `/output`

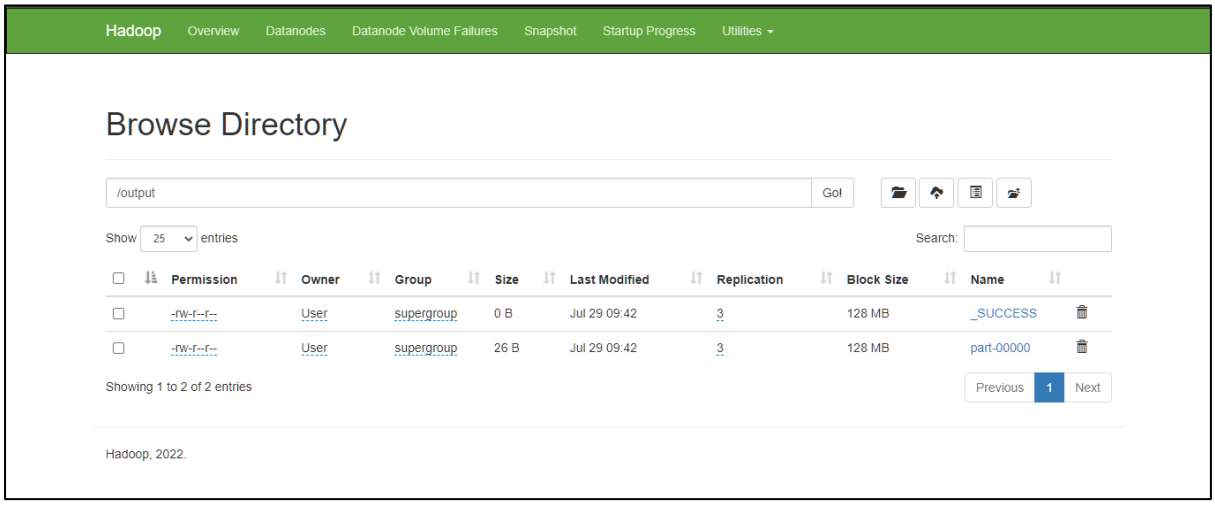
```
hadoop dfs -ls /output
```

```
Administrator: Command Prompt

C:\Hadoop\hadoop-3.3.3\bin>hadoop dfs -ls /output
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Found 2 items
-rw-r--r--   3 User supergroup          0 2022-07-29 09:42 /output/_SUCCESS
-rw-r--r--   3 User supergroup         26 2022-07-29 09:42 /output/part-00000

C:\Hadoop\hadoop-3.3.3\bin>
```

Step 7: To check the output is generated to the Hadoop HDFS the visit http://localhost:9870/dfshealth.html#tab-overview → go to utilities in the navigation bar and click on the Browse the file system







Step 8: You can then inspect the contents of the file with the `fs -cat` command:

```
hadoop fs -cat /output/part-00000
```