Practical No: 8

Date: 19/04/2023

Hadoop Installation:

Aim: Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python

Requirements

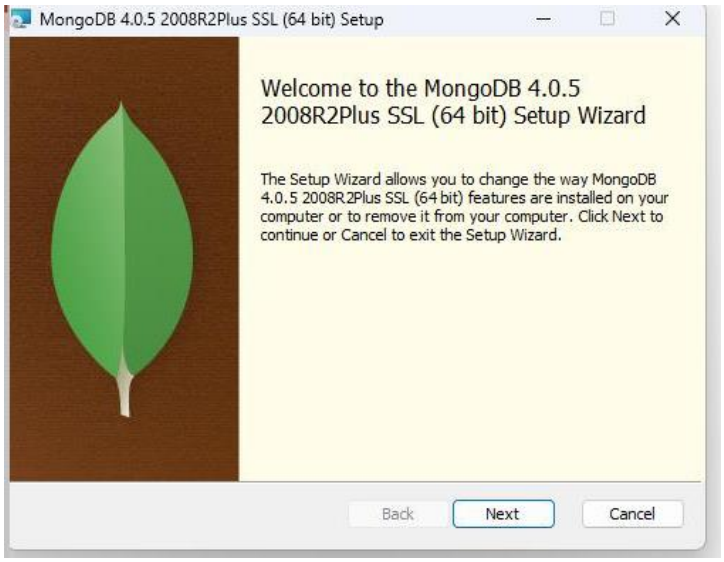
a. PyMongo

b. Mongo Database

Step A: Install Mongo database

Step 1) Go to (https://www.mongodb.com/download-center/community) and Download MongoDB Community Server. We will install the 64-bit version for Windows.

Step 2) Once download is complete open the msi file. Click Next in the start up screen



Step 3)

1. Accept the End-User License Agreement

2. Click Next

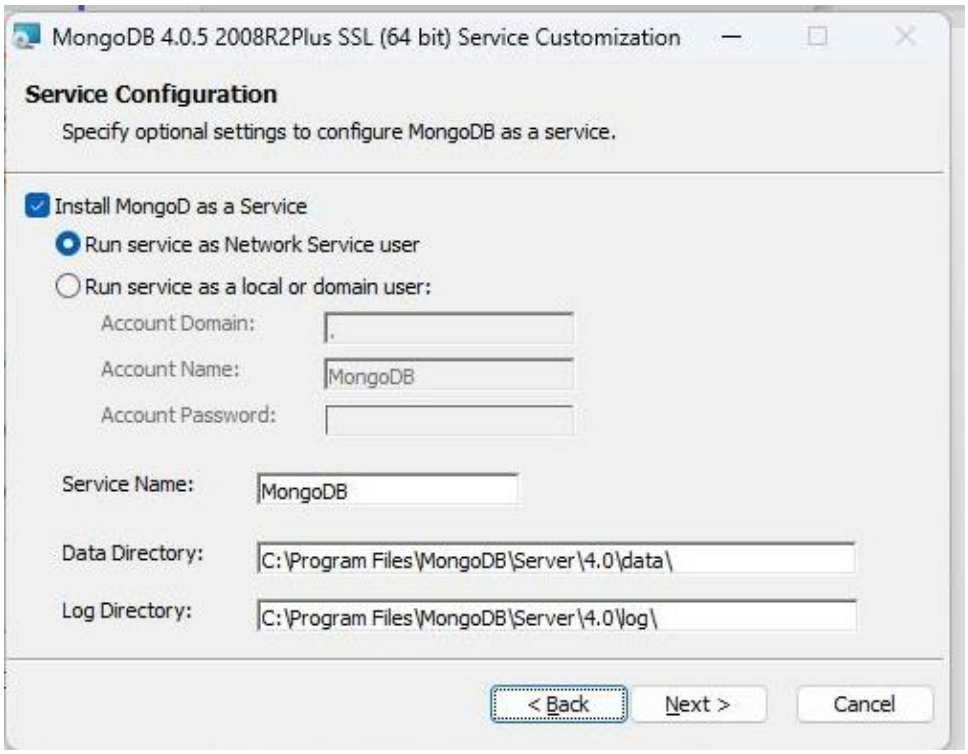
Step 4) Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.

Step 5)

1. Select “Run service as Network Service user”. make a note of the data directory,

we”ll need this later.

2. Click Next



Step 6) Click on the Install button to start the installation.

Step 7) Installation begins. Click Next once completed.

Step 8) Click on the Finish button to complete the installation.

Test Mongodb

Step 1) Go to " C:\Program Files\MongoDB\Server\4.0\bin" and double click on **mongo.exe.** Alternatively, you can also click on the MongoDB desktop icon.

Create the directory where MongoDB will store its files.

Open command prompt window and apply following commands

C:\users\admin> cd\

C:\>md data\db

Step 2) Execute mongodb

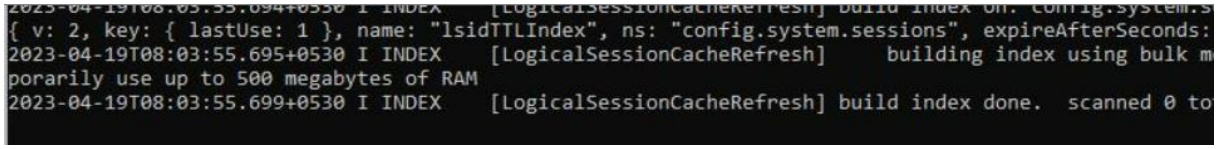
Open another command prompt window.

C:\> cd C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin> mongod

In case if it gives an error then run the following command:

C:\Program Files\MongoDB\Server\4.0\bin> mongod –repair

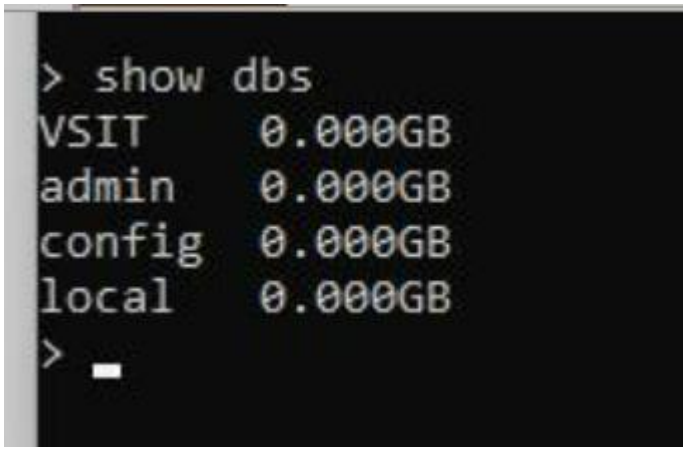


Step 3) Connect to MongoDB using the Mongo shell

Let the MongoDB daemon to run.

Open another command prompt window and run the following commands:

C:\users\admin> cd C:\Program Files\MongoDB\Server\4.0\bin
C:\Program Files\MongoDB\Server\4.0\bin>mongo



Step 4) Install PyMongo

Open another command prompt window and run the following commands:

Check the python version on your desktop / laptop and copy that path from window explorer

C:\users\admin>cd C:\Program Files\Python311\Scripts
C:\Program Files\<Python38>\Scripts > python -m pip install pymongo

```
C:\Windows\System32>cd C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>python -m pip install pymongo
Requirement already satisfied: pymongo in c:\users\admin\appdata\local\packages\pythonsoftwarefoundation.pytho
5n2kfra8p0\localcache\local-packages\python310\site-packages (4.2.0)

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>
```

Note: # `-m` option is for <module-name>

Now you have downloaded and installed a mongoDB driver.

Step 5) Test PyMongo

Run the following command from python command prompt

import pymongo

Now, either create a file in Python IDLE or run all commands one by one in sequence on Python cell

Program 1: Creating a Database: create_dp.py

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["mybigdata"]

print(myclient.list_database_names())
```

```
========== RESTART: D:/NK.py ==========
['VSIT', 'admin', 'config', 'local', 'mybigdata', 'mybigdataNK']
['student']
```

Progam 2: Creating a Collection: create_collection.py

```
import pymongo

myclient = pymongo.MongoClient("mongodb://localhost:27017/")

mydb = myclient["mybigdata"]

mycol=mydb["student"]

print(mydb.list_collection_names())
```

Progam 3: Insert into Collection: insert_into_collection.py

```
import pymongo
```

```
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol=mydb["student"]
mydict={"name":"Beena", "address":"Mumbai"}
x=mycol.insert_one(mydict) # insert_one(containing the name(s) and value(s) of each field
```

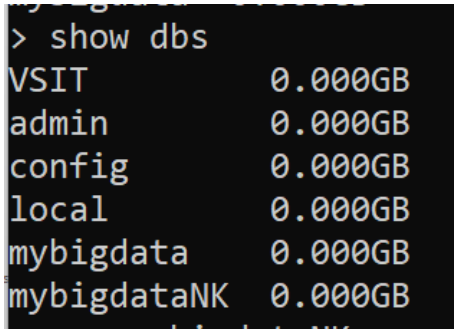
Program 4: Insert Multiple data into Collection: insert_many.py

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdataNK"]
print(myclient.list_database_names())
mycol=mydb["student"]
print(mydb.list_collection_names())
mydict={"name":"Ninad", "address":"Mumbai"}
x=mycol.insert_one(mydict) # insert_one(containing the name(s) and value(s) of each field
 mylist=[{"name":"Vighnesh", "address":"Mumbai"}, {"name":"Sarthak", "address":"Mumbai"},
{"name":"Nidhi", "address":"Pune"}, {"name":"Komal", "address":"Pune"},]
x=mycol.insert_many(mylist)
```

Step 6) Test in Mongodb to check database and data inserted in collection

a. If you want to check your database list, use the command show dbs in mongo command prompt

```
> show dbs
```



b. If you want to use a database with name mybigdata, then use database

statement would be as follow:

> use mybigdataNK

```
> use mybigdataNK
switched to db mybigdataNK
> show collections
```

c. If you want to check collection in mongodb use the command show collections

> show collections

d. If you want to display the first row from collection: db.collection_name.find()

> db.student.findOne()

e. If you want to display all the data from collection: db.collection_name.find()

> db.student.find()

f. count number of rows in a collection

> db.student.count()

```
> show collections
student
> db.student.findOne()
{
        "_id" : ObjectId("643f57fd927b349355491923"),
        "name" : "Ninad",
        "address" : "Mumbai"
}
> db.student.count()
5
```

```
> db.student.find()
{ "_id" : ObjectId("643f57fd927b349355491923"), "name" : "Ninad", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491924"), "name" : "Vighnesh", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491925"), "name" : "Sarthak", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491926"), "name" : "Nidhi", "address" : "Pune" }
{ "_id" : ObjectId("643f57fd927b349355491927"), "name" : "Komal", "address" : "Pune" }
>
```