MONGO DB

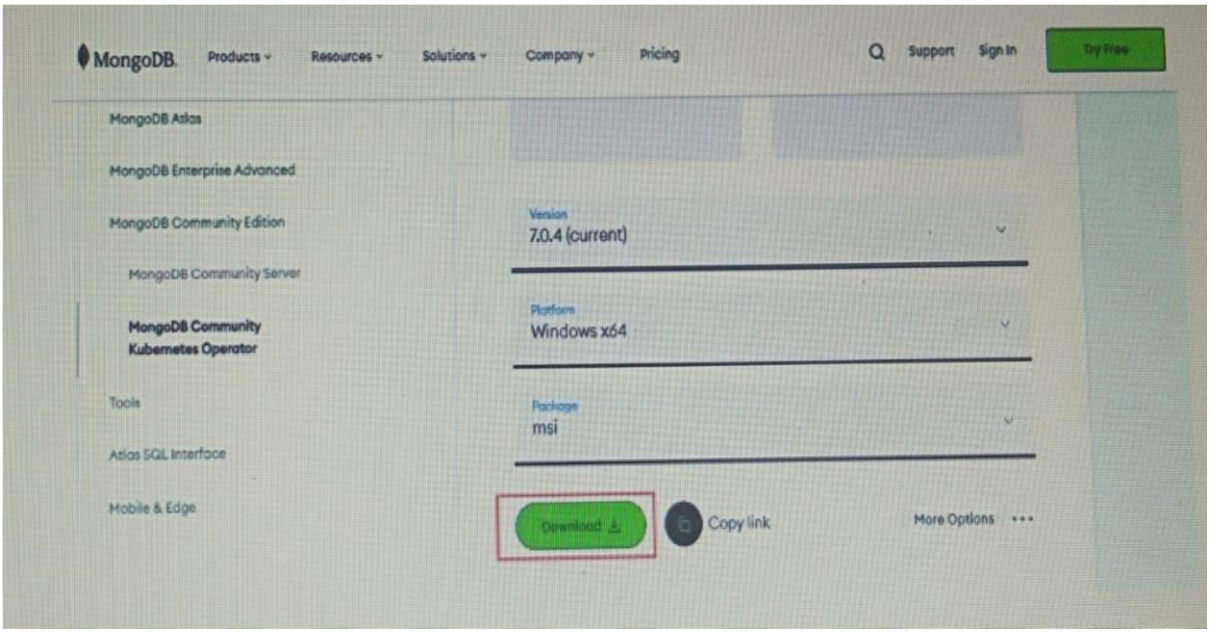
Steps to install Mongo DB:-

To install MongoDB on Windows, first, download the MongoDB server and then install the MongoDB shell. The Steps below explain the installation process in detail and provide the required resources for the smooth **download and install MongoDB**.

Step 1: Go to the MongoDB Download Center to download the MongoDB Community Server.

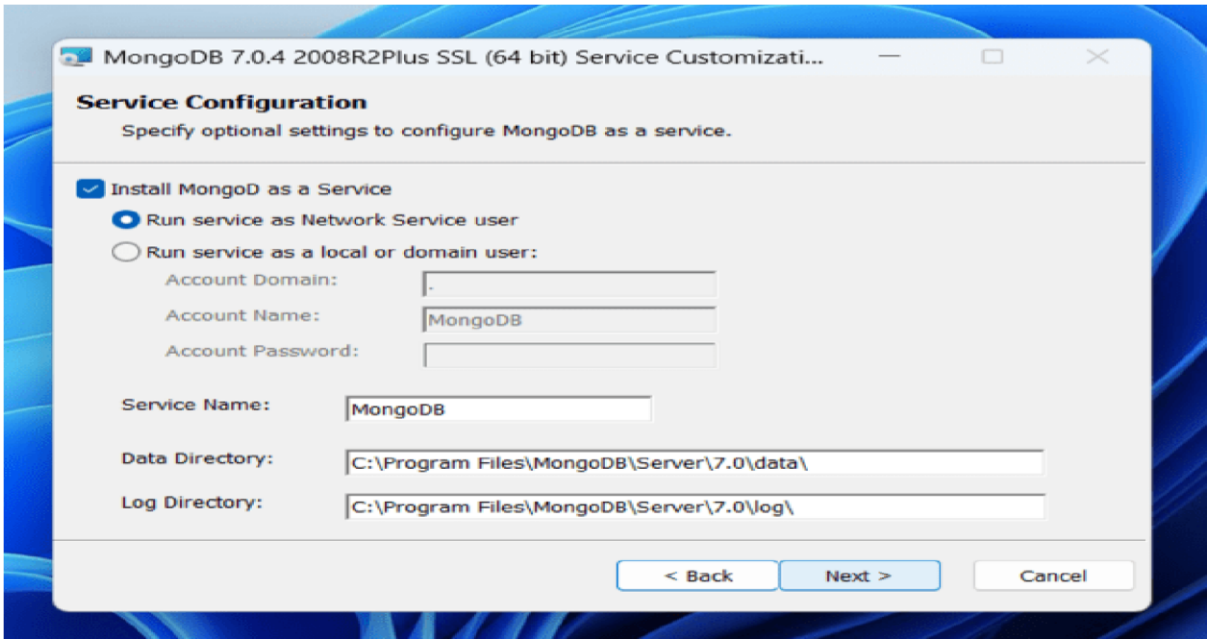
Here, You can select any version, Windows, and package according to your requirement. For Windows, we need to choose:

- Version: 7.0.4
- OS: Windows x64
- Package: msi

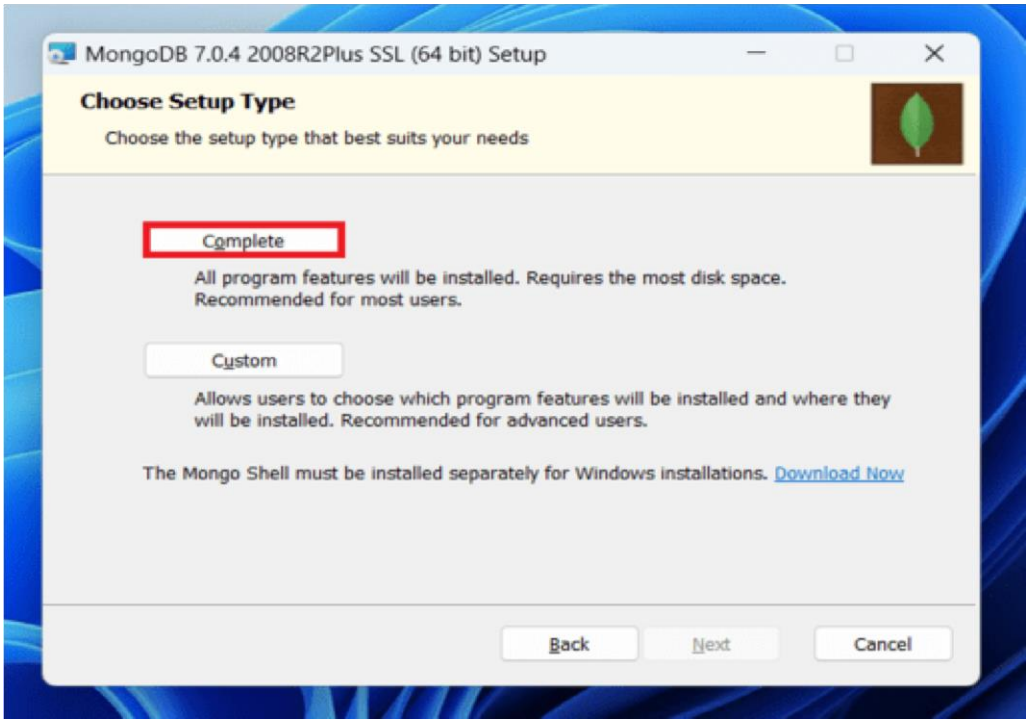


Step 2: When the download is complete open the msi file and click the *next button* in the startup screen:

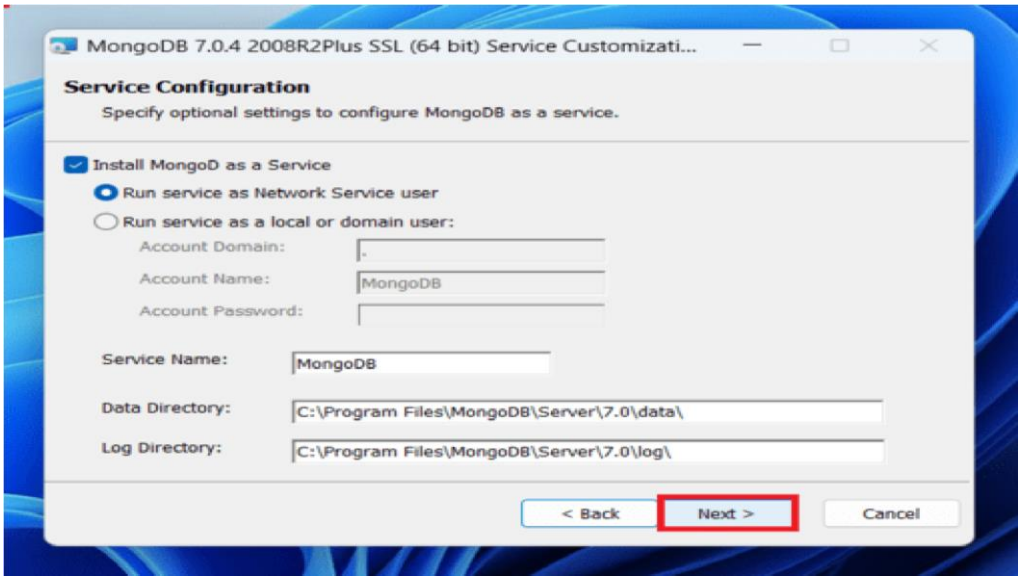
Step 3: Now accept the End-User License Agreement and click the next button:



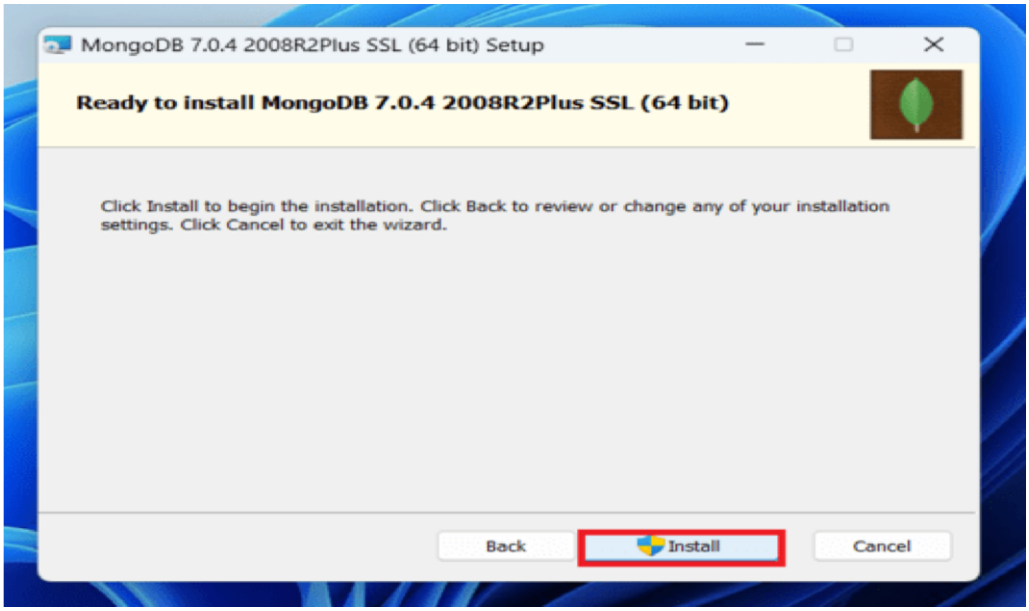
Step 4: Now select the *complete option* to install all the program features. Here, if you can want to install only selected program features and want to select the location of the installation, then use the *Custom option:*



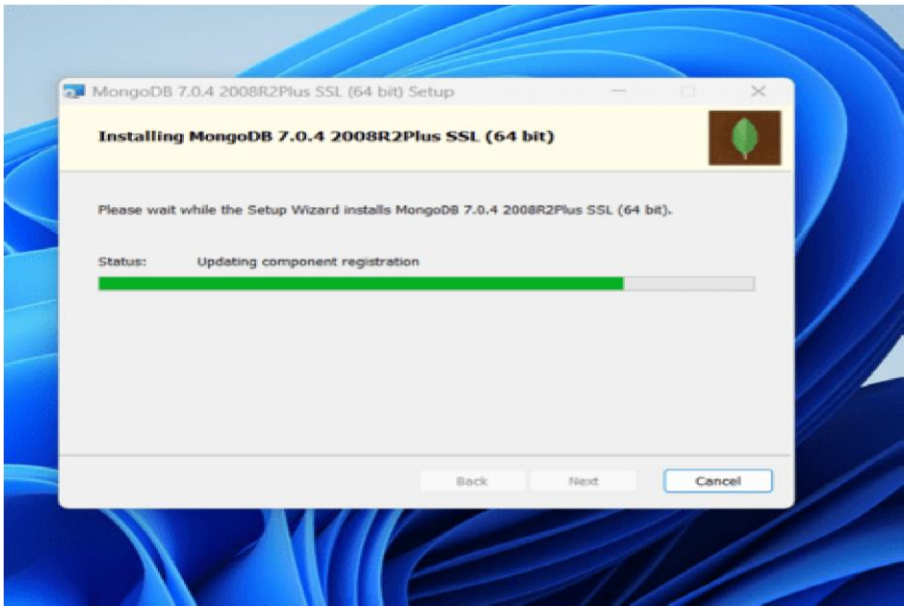
Step 5: Select “Run service as Network Service user” and copy the path of the data directory. Click Next:



Step 6: Click the *Install button* to start the MongoDB installation process:

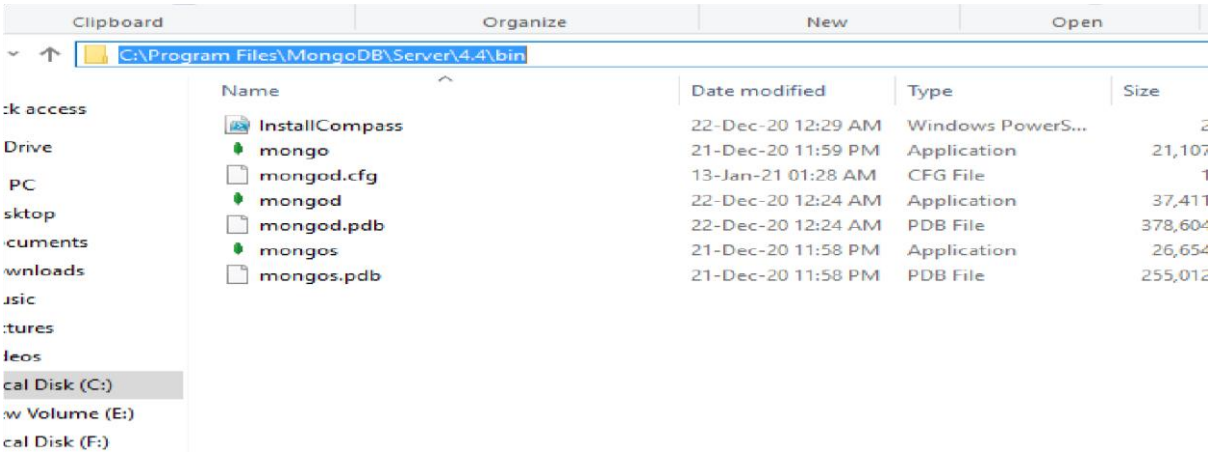


Step 7: After clicking on the install button installation of MongoDB begins:

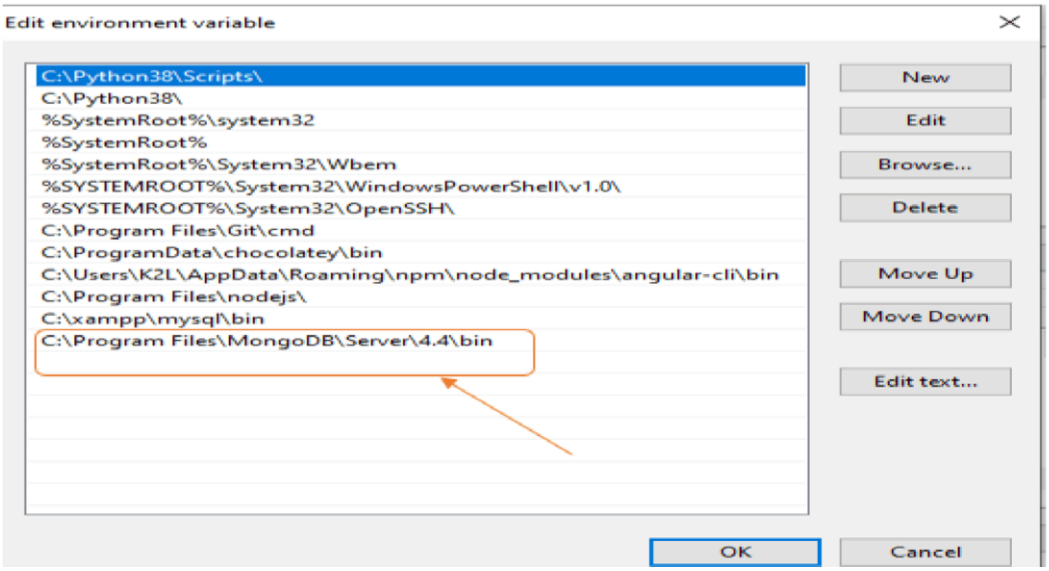


Step 8: Now click the ***Finish button*** to complete the MongoDB installation process:

Step 9: Now we go to the location where MongoDB installed in step 5 in your system and copy the bin path:



Step 10: Now, to create an environment variable open system properties >> Environment Variable >> System variable >> path >> Edit Environment variable and paste the copied link to your environment system and click Ok:



Step 11: After setting the environment variable, we will run the MongoDB server, i.e. mongod. So, open the command prompt and run the following command:

```
mongod
```

When you run this command you will get an error i.e. *C:/data/db/ not found*.

Step 12: Now, Open C drive and create a folder named “data” inside this folder create another folder named “db”. After creating these folders. Again open the command prompt and run the following command:

```
mongod
```

Now, this time the MongoDB server(i.e., mongod) will run successfully.

```
C:\Users\NIkhil Chhipa>mongod
{"t":{"$date":"2021-01-31T00:56:54.081+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx"
ify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2021-01-31T00:56:54.087+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx"
}
{"t":{"$date":"2021-01-31T00:56:54.088+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx"
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx"
bPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-L9MUQ7N"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx"
rgetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx"
gitVersion":"913d6b62acfbb344dde1b116f4161360acd8fd13","modules":[],"allocator":"tcmalloc","
}}}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx"
ndows 10","version":"10.0 (build 14393)"}}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx"
{"t":{"$date":"2021-01-31T00:56:54.157+05:30"},"s":"I",  "c":"STORAGE",  "id":22270,   "ctx"
:{"dbpath":"C:/data/db/","storageEngine":"wiredTiger"}}
{"t":{"$date":"2021-01-31T00:56:54.158+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx"
ize=1491M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statist
le_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statisti
ess],"}}
{"t":{"$date":"2021-01-31T00:56:54.395+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx"
95788][3708:140713908197088], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 20 thr
{"t":{"$date":"2021-01-31T00:56:54.631+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx"
```

Run mongo Shell:

Step 13: Now we are going to connect our server (mongod) with the mongo shell. So, keep that mongod window and open a new command prompt window and write **mongo.** Now, our mongo shell will successfully connect to the mongod.

Important Point: Please do not close the mongod window if you close this window your server will stop working and it will not able to connect with the mongo shell.

```
C:\Users\NIkhil Chhipa>mongo
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("96cca5da-dc9f-4a40-aabb-732ee37600c0") }
MongoDB server version: 4.4.3
---
The server generated these startup warnings when booting:
        2021-01-28T20:56:52.570+05:30: Access control is not enabled for the database. Read and write access
configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

MongoDB, a popular NoSQL database, has several characteristics that distinguish it from traditional relational databases like MySQL or PostgreSQL. Here are some key characteristics of MongoDB:

1. **Schema-less Design**: MongoDB is schema-less, meaning you don't need to predefine the structure of your data. Each document can have its own unique schema, providing flexibility in data modeling.
2. **Document-Oriented**: MongoDB stores data in flexible, JSON-like documents called BSON (Binary JSON). Each document can contain nested data structures, arrays, and other BSON data types.
3. **Highly Scalable**: MongoDB is designed to scale horizontally across multiple servers, allowing you to handle large volumes of data and high traffic loads by distributing data across a cluster of machines.
4. **High Performance**: MongoDB's architecture and indexing features provide high performance for both read and write operations. It supports various types of indexes, including compound indexes and geospatial indexes, to optimize query performance.
5. **Automatic Sharding**: MongoDB supports automatic sharding, which partitions data across multiple servers to distribute the load and improve scalability. Sharding allows you to horizontally scale your database while maintaining high availability and performance.
6. **Flexible Query Language**: MongoDB provides a rich query language with support for ad-hoc queries, range queries, regular expressions, and more. It also supports aggregation pipelines for performing complex data aggregation and transformation operations.
7. **Replication and High Availability**: MongoDB supports replica sets, which are self-healing clusters of MongoDB nodes. Replica sets provide data redundancy and high availability by maintaining multiple copies of data across different servers.
8. **Horizontal Data Consistency**: MongoDB sacrifices strong consistency for scalability and availability. By default, reads may not always reflect the most recent writes, but MongoDB provides

configurable consistency levels to balance consistency requirements with performance and scalability needs.

9. **Built-in Data Redundancy**: MongoDB's replica sets automatically maintain multiple copies of data across different nodes, providing built-in data redundancy and fault tolerance. In case of node failures, replica sets can elect a new primary node and continue serving read and write requests.
10. **Rich Ecosystem and Community Support**: MongoDB has a rich ecosystem with a wide range of tools, libraries, and frameworks that integrate seamlessly with MongoDB. It also has an active community providing support, resources, and frequent updates and improvements.

These characteristics make MongoDB a popular choice for a variety of applications, including web applications, mobile apps, real-time analytics, and content management systems.