

BIBD MINI PROJECT

MongoDB

What is MongoDB?

MongoDB is a **cross-platform document-oriented database program**. Classified as a NoSQL database program, MongoDB uses JSON -like documents with schema. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License.

Installation of MongoDB

Step 1 — Download the MongoDB MSI Installer Package

- ☐ Download the current version of MongoDB.
- ☐ Make sure you select **MSI** as the package you want to download.

Select the server you would like to run:

MongoDB Community Server
FEATURE RICH. DEVELOPER READY.

Version: 4.0.3 (current release) ▼

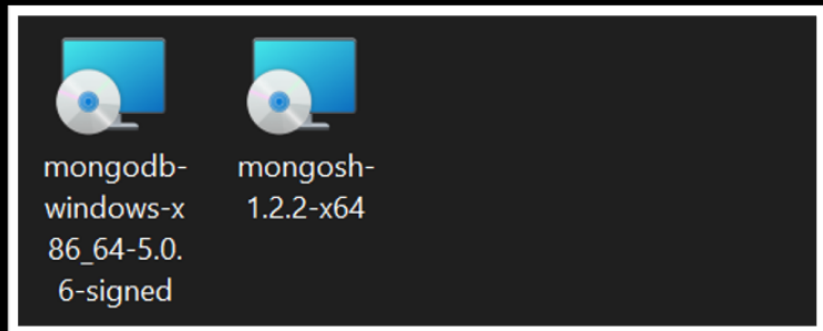
OS: Windows 64-bit x64 ▼

Package: MSI ▼

Download

Step 2 — Install MongoDB with the Installation Wizard

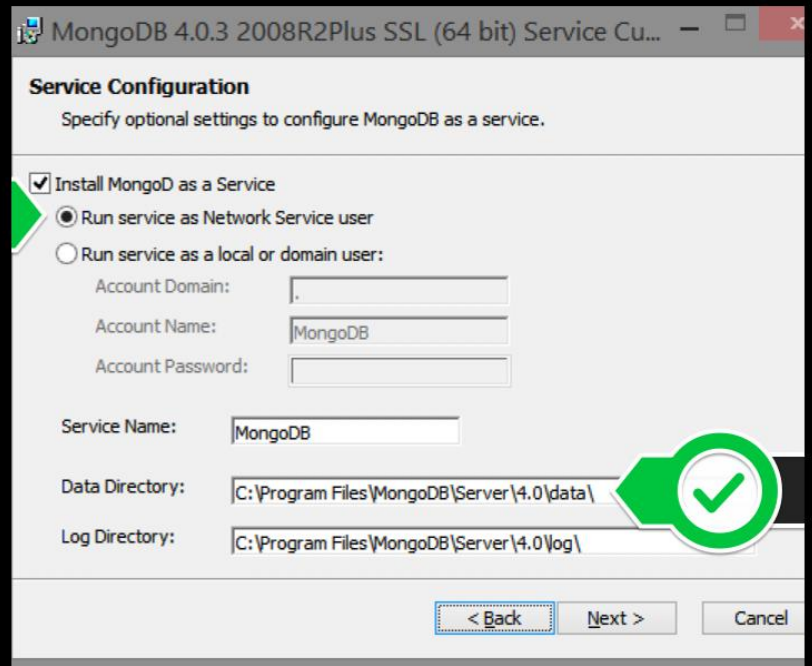
- ☐ Make sure you are logged in as a user with Admin privileges.
- ☐ Then navigate to your downloads folder and double click on the .msi package you just downloaded.
- ☐ This will launch the installation wizard.



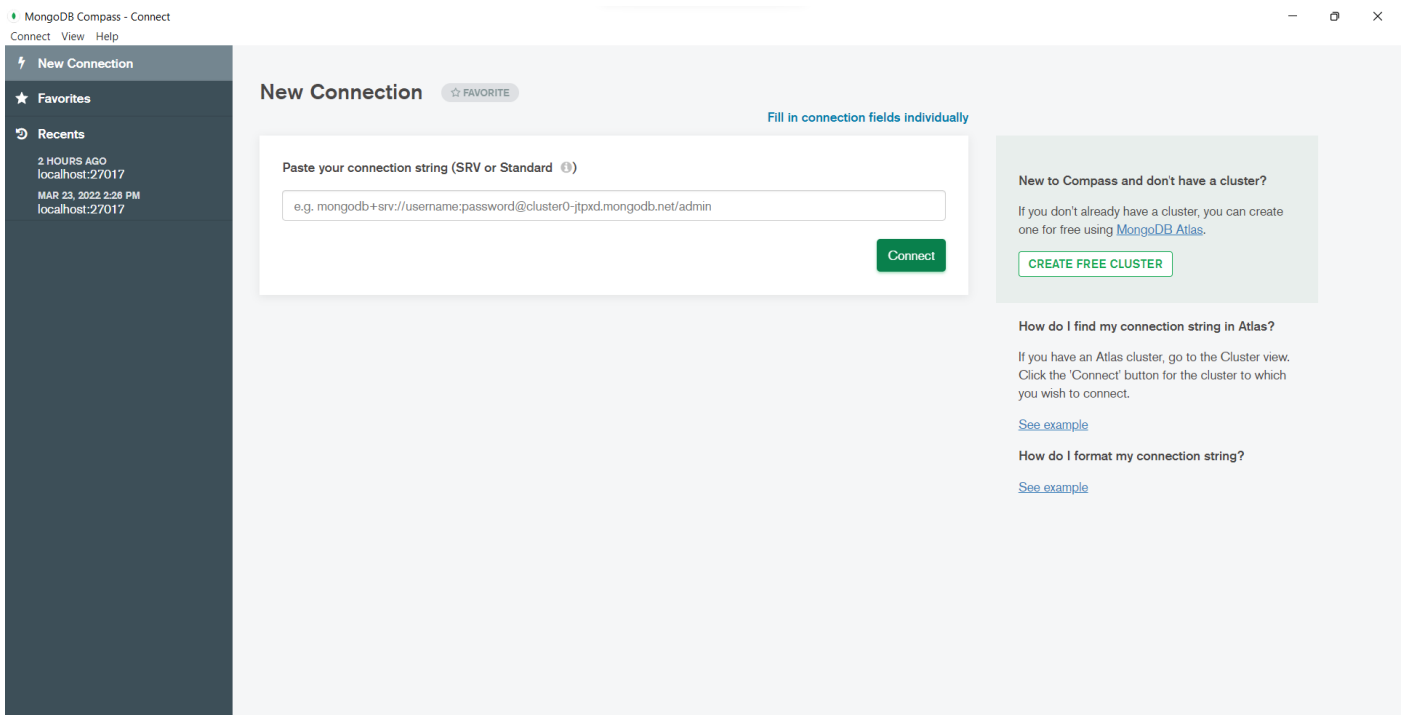
- ☐ Click Next to start installation.
- ☐ Accept the license agreement then click Next.
- ☐ Select the "Complete Setup".



- ❑ Select “Run service as Network Service user” which is given by default.



- After installing need to connect MongoDB Compass with Localhost so we can access the database



- After clicking on connect, localhost get connected then we can create database, collection & can also execute command by compass.
- But we will do the command on PowerShell.
- Open PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Mahesh>
```

➤ Starting with MongoDB Command on powershell

View all databases

show dbs

```
> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
mydb     0.000GB
>
```

Create a new or switch databases

use dbName

```
> use mydb
switched to db mydb
```

View current Database

db

```
> db
mydb
```

Create a collection named 'comments'

```
db.createCollection('comments')
```

```
> db.createCollection('comments')  
{ "ok" : 1 }
```

Insert One Row

```
db.comments.insert({  
  'name': 'Mack',  
  'lang': 'JavaScript',  
  'member_since': 5  
})
```

```
> db.comments.insert({  
... 'name': 'Mack',  
... 'lang': 'JavaScript',  
... 'member_since': 5})  
WriteResult({ "nInserted" : 1 })
```

Show all Rows in a Collection

&

Show all Rows in a Collection (Prettified)

```
db.comments.find()
```

```
db.comments.find().pretty()
```

```
> db.comments.find()  
{ "_id" : ObjectId("623ddd37e49c8c312d52e187"), "name" : "Mack", "lang" : "JavaScript", "member_since" : 5 }  
> db.comments.find().pretty()  
{  
  "_id" : ObjectId("623ddd37e49c8c312d52e187"),  
  "name" : "Mack",  
  "lang" : "JavaScript",  
  "member_since" : 5  
}
```

Insert many Rows

```
db.comments.insertMany([{\n  'name': 'Harry',\n  'lang': 'JavaScript',\n  'member_since': 5\n},\n  {'name': 'Rohan',\n  'lang': 'Python',\n  'member_since': 3\n},\n  {'name': 'Lovish',\n  'lang': 'Java',\n  'member_since': 4\n}])
```

```
> db.comments.insertMany([{\n...   'name': 'Harry',\n...   'lang': 'JavaScript',\n...   'member_since': 5\n... },\n...   {'name': 'Rohan',\n...   'lang': 'Python',\n...   'member_since': 3\n... },\n...   {'name': 'Lovish',\n...   'lang': 'Java',\n...   'member_since': 4\n... }])\n{\n  "acknowledged" : true,\n  "insertedIds" : [\n    ObjectId("623ddd62e49c8c312d52e188"),\n    ObjectId("623ddd62e49c8c312d52e189"),\n    ObjectId("623ddd62e49c8c312d52e18a")\n  ]\n}
```

Search in a MongoDB Database

`db.comments.find({lang:'Python'})`

```
> db.comments.find({lang:'Python'})
{
  "_id" : ObjectId("623ddd62e49c8c312d52e189"), "name" : "Rohan", "lang" : "Python", "member_since" : 3 }
>
```

Limit the number of rows in output

`db.comments.find().limit(2)`

```
> db.comments.find().limit(2)
{
  "_id" : ObjectId("623ddd37e49c8c312d52e187"), "name" : "Mack", "lang" : "JavaScript", "member_since" : 5 }
  "_id" : ObjectId("623ddd62e49c8c312d52e188"), "name" : "Harry", "lang" : "JavaScript", "member_since" : 5 }
> db.comments.find().limit(2).pretty()
{
  "_id" : ObjectId("623ddd37e49c8c312d52e187"),
  "name" : "Mack",
  "lang" : "JavaScript",
  "member_since" : 5
}

  "_id" : ObjectId("623ddd62e49c8c312d52e188"),
  "name" : "Harry",
  "lang" : "JavaScript",
  "member_since" : 5
}
>
```

Count the number of rows in the output

`db.comments.find().count()`

```
>
> db.comments.find().count()
4
>
```

Update a row & if upsert is true than new row is created

```
db.comments.update({name: 'Shubham'},  
{name: 'Shubham',  
  'lang': 'JavaScript',  
  'member_since': 51  
}, {upsert: true})
```

```
>  
>  
> db.comments.update({name: 'Shubham'},  
... {'name': 'Shubham',  
...   'lang': 'JavaScript',  
...   'member_since': 51  
... }, {upsert: true})  
WriteResult({  
  "nMatched" : 0,  
  "nUpserted" : 1,  
  "nModified" : 0,  
  "_id" : ObjectId("623ddf355b429b6f457ce311")  
})
```


Mongodb Increment Operator

```
db.comments.update({name: 'Rohan'},  
{$inc:{  
  member_since: 2  
}})
```

Mongodb Rename Operator

```
db.comments.update({name: 'Rohan'},  
{$rename:{  
  member_since: 'member'  
}})
```

Delete Row

```
db.comments.remove({name: 'Harry'})
```

```

>
> db.comments.update({name: 'Rohan'},
... {$inc:{
...   member_since: 2
... }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
>
> db.comments.update({name: 'Rohan'},
... {$rename:{
...   member_since: 'member'
... }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
>
> db.comments.remove({name: 'Harry'})
WriteResult({ "nRemoved" : 1 })
>

```

Less than/Greater than/ Less than or Eq/Greater than or Eq command

```
db.comments.find({member_since: {$lt: 90}})
```

```
db.comments.find({member_since: {$lte: 90}})
```

```
db.comments.find({member_since: {$gt: 90}})
```

```
db.comments.find({member_since: {$gte: 90}})
```

```

>
> db.comments.find({member_since: {$lt: 90}})
{"_id" : ObjectId("623ddd37e49c8c312d52e187"), "name" : "Mack", "lang" : "JavaScript", "member_since" : 5 }
{"_id" : ObjectId("623ddd62e49c8c312d52e18a"), "name" : "Lovish", "lang" : "Java", "member_since" : 4 }
{"_id" : ObjectId("623ddf355b429b6f457ce311"), "name" : "Shubham", "lang" : "JavaScript", "member_since" : 5 }
> db.comments.find({member_since: {$lt: 90}}).pretty()
{
  "_id" : ObjectId("623ddd37e49c8c312d52e187"),
  "name" : "Mack",
  "lang" : "JavaScript",
  "member_since" : 5
}

{
  "_id" : ObjectId("623ddd62e49c8c312d52e18a"),
  "name" : "Lovish",
  "lang" : "Java",
  "member_since" : 4
}

{
  "_id" : ObjectId("623ddf355b429b6f457ce311"),
  "name" : "Shubham",
  "lang" : "JavaScript",
  "member_since" : 5
}

```

```
> db.comments.find({member_since: {$gte: 90}}).pretty()
{
  "_id" : ObjectId("623ddf355b429b6f457ce311"),
  "name" : "Shubham",
  "lang" : "JavaScript",
  "member_since" : 90
}
```

➤ Final collection data in the database

```
> db.comments.find().pretty()
{
  "_id" : ObjectId("623ddd37e49c8c312d52e187"),
  "name" : "Mack",
  "lang" : "JavaScript",
  "member_since" : 5
}
{
  "_id" : ObjectId("623ddd62e49c8c312d52e189"),
  "name" : "Rohan",
  "lang" : "Python",
  "member" : 5
}
{
  "_id" : ObjectId("623ddd62e49c8c312d52e18a"),
  "name" : "Lovish",
  "lang" : "Java",
  "member_since" : 4
}
{
  "_id" : ObjectId("623ddf355b429b6f457ce311"),
  "name" : "Shubham",
  "lang" : "JavaScript",
  "member_since" : 60
}
```

Merin Kurian Roll no - 20

mydb.commentsDocuments

+

mydb.comments

DOCUMENTS 4STORAGE SIZE20.5KBAVG. SIZE75BTOTAL SIZE20.5KBAVG. SIZE20.5KBINDEXES 1

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

FILTER

{ field: 'value' }

OPTIONS

FIND

RESET

↺

⋮

ADD DATA

↓

VIEW

☰

{ }

☒

Displaying documents 1 - 4 of 4

◀

▶

REFRESH

_id: ObjectId("623de70d98701baa78f5bdd4")

lang: "JavaScript"

member_since: "5"

name: "Mack"

_id: ObjectId("623de70d98701baa78f5bdd5")

lang: "Python"

member: "5"

name: "Rohan"

_id: ObjectId("623de70d98701baa78f5bdd6")

lang: "Java"

member_since: "4"

name: "Lovish"

_id: ObjectId("623de70d98701baa78f5bdd7")

lang: "JavaScript"

member_since: "60"

name: "Shubham"