Agenda:

- What is MongoDB?
- What advantages does MongoDB have?
- Familiarization with BSON
- MongoDB data example and storing of data in JSON
- Installation of MongoDB Community Server
- Installation of MongoDB Compass
- How to create and drop DB?
- How to create and drop a Collection?

Starting from this session, we are going to learn all about MongoDB starting from scratch. So, first thing that comes to mind is that what is MongoDB, s
let's get started:

What is MongoDB ?

MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relation
databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDI
Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into lig
around the mid-2000s.

Now, let's see some features of MongoDB:

1. Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other.

2. The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs.

3. The rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly.

4. The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.

5. Scalability – The MongoDB environments are very scalable. Companies across the world have defined clusters with some of them running 100+ nodes with around millions of documents within the database.

Advantages of using MongoDB:

1. **Document-oriented** – Since MongoDB is a NoSQL type database, instead of having data in a relational type format, it stores the data in documents. This makes MongoDB very flexible and adaptable to real business world situation and requirements.

2. **Ad hoc queries** – MongoDB supports searching by field, range queries, and regular expression searches. Queries can be made to return specific fields within documents.

3. **Indexing** – Indexes can be created to improve the performance of searches within MongoDB. Any field in a MongoDB document can be indexed.

4. **Replication** – MongoDB can provide high availability with replica sets. A replica set consists of two or more mongo DB instances. Each replica set member may act in the role of the primary or secondary replica at any time. The primary replica is the main server which interacts with the client and performs all the read/write operations. The Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically switches over to the secondary and then it becomes the primary server.

5. **Load balancing** – MongoDB uses the concept of sharding to scale horizontally by splitting data across multiple MongoDB instances. MongoDB can run over multiple servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure.

Familiarization with BSON:

Before we get into BSON, let's first quickly see that what is JSON

JSON:

JavaScript Object Notation, more commonly known as JSON, was defined as part of the JavaScript language in the early 2000s by JavaScript creat
Douglas Crockford, though it wasn't until 2013 that the format was officially specified.

JavaScript objects are simple associative containers, wherein a string key is mapped to a value (which can be a number, string, function, or even anoth object). This simple language trait allowed JavaScript objects to be represented remarkably simply in text as shown below:

```
{
  "_id": 1,
  "name" : { "first" : "John", "last" : "Backus" },
  "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards" : [
    {
      "award" : "W.W. McDowell Award",
      "year" : 1967,
      "by" : "IEEE Computer Society"
    }, {
      "award" : "Draper Prize",
      "year" : 1993,
      "by" : "National Academy of Engineering"
    }
  ]
}
```

As JavaScript became the default language of client-side web development, JSON began to take on a life of its own. JSON quickly moved beyond th web page, and into software everywhere.

JSON shows up in many different cases:

- APIs
- Configuration files
- Log messages
- Database storage

MongoDB and JSON connection:

MongoDB was designed from its inception to be the ultimate data platform for modern application development. JSON's ubiquity made it the obviou choice for representing data structures in MongoDB's innovative document data model. In order to make MongoDB JSON-first, but still high-performanc and general-purpose, BSON was invented to bridge the gap: a binary representation to store data in JSON format, optimized for speed, space, ar flexibility.

So, now let's learn about BSON:

BSON:

BSON simply stands for "Binary JSON," and that's exactly what it was invented to be. BSON's binary structure encodes type and length informatio which allows it to be parsed much more quickly.

Since its initial formulation, BSON has been extended to add some optional non-JSON-native data types, like dates and binary data, without whic MongoDB would have been missing some valuable support.

Languages that support any kind of complex mathematics typically have different sized integers (ints vs longs) or various levels of decimal precisio (float, double, decimal128, etc.).

Not only is it helpful to be able to represent those distinctions in data stored in MongoDB, it also allows for comparisons and calculations to happe directly on data in ways that simplify consuming application code.

Now, a question arises that whether MongoDB uses BSON or JSON:

MongoDB stores data in BSON format both internally, and over the network, but that doesn't mean you can't think of MongoDB as a JSON databas Anything you can represent in JSON can be natively stored in MongoDB, and retrieved just as easily in JSON.

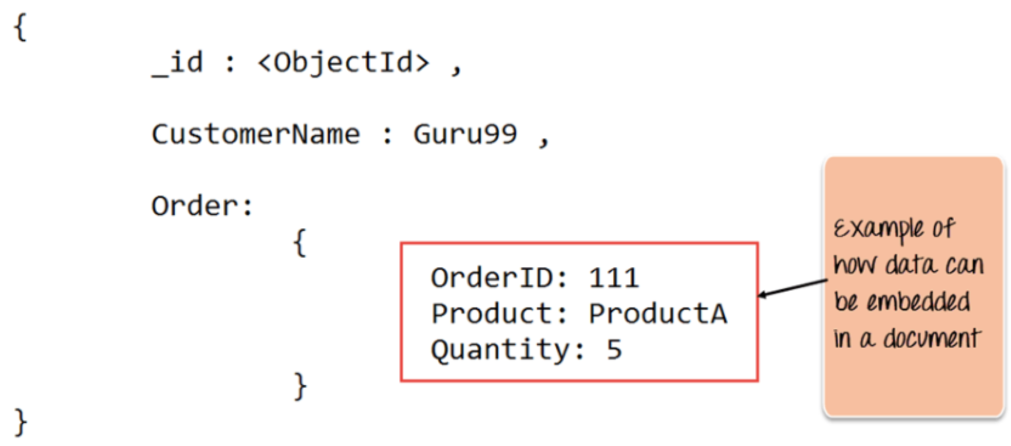
Unlike systems that simply store JSON as string-encoded values, or binary-encoded blobs, MongoDB uses BSON to offer the industry's most powerf indexing and querying features on top of the web's most usable data format.

Let's try to get a gist about MongoDB through an example:

The below example shows how a document can be modeled in MongoDB.

1. The _id field is added by MongoDB to uniquely identify the document in the collection.

2. What you can note is that the Order Data (OrderID, Product, and Quantity) which in RDBMS will normally be stored in a separate table, while in MongoDB it is actually stored as an embedded document in the collection itself. This is one of the key differences in how data is modeled in MongoDB.



```
{
        _id : <ObjectId> ,

        CustomerName : Guru99 ,

        Order:
                {     OrderID: 111
                      Product: ProductA
                      Quantity: 5
                }
}
```

Key Components of MongoDB Architecture

Below are a few of the common terms used in MongoDB

1. **_id** – This is a field required in every MongoDB document. The _id field represents a unique value in the MongoDB document. The _id field is like the document's primary key. If you create a new document without an _id field, MongoDB will automatically create the field. So for example, if we see the example of the above customer table, Mongo DB will add a 24 digit unique identifier to each document in the collection.

_Id	CustomerId	CustomerName	OrderId
563479cc8a8a4246bd27d784	11	Guru99	111
563479cc7a8a4246bd47d784	22	Trevor Smith	222
563479cc9a8a4246bd57d784	33	Nicole	333

An **ObjectId** is a 12-byte BSON type having the following structure –

- The first 4 bytes representing the seconds since the unix epoch
- The next 3 bytes are the machine identifier
- The next 2 bytes consists of **process id**
- The last 3 bytes are a random counter value

MongoDB uses ObjectIds as the default value of **_id** field of each document, which is generated while the creation of any document. The comple combination of ObjectId makes all the _id fields unique.

Creating New ObjectId

To generate a new ObjectId use the following code –

```
>newObjectId = ObjectId()
```

The above statement returned the following uniquely generated id –

```
ObjectId("5349b4ddd2781d08c09890f3")
```

Instead of MongoDB generating the ObjectId, you can also provide a 12-byte id –

```
>myObjectId = ObjectId("5349b4ddd2781d08c09890f4")
```

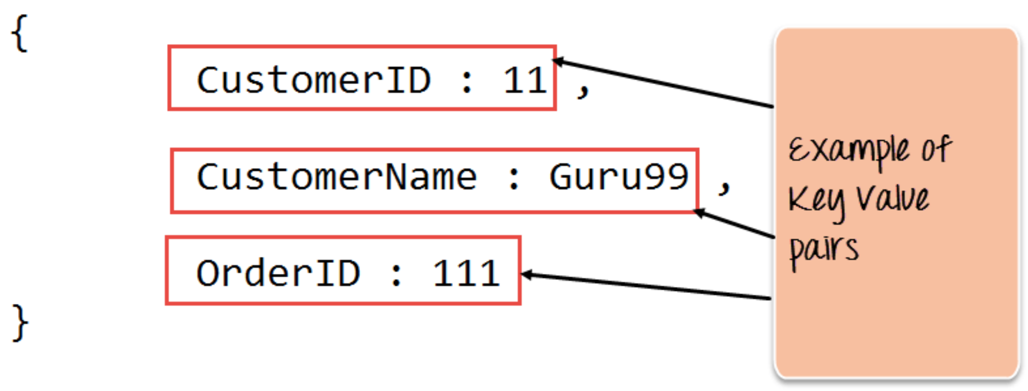
2. **Collection** – This is a grouping of MongoDB documents. A collection is the equivalent of a table which is created in any other RDMS such as Oracle or MS SQL. A collection exists within a single database. As seen from the introduction collections don't enforce any sort of structure.

3. **Cursor** – This is a pointer to the result set of a query. Clients can iterate through a cursor to retrieve results.

4. **Database** – This is a container for collections like in RDMS wherein it is a container for tables. Each database gets its own set of files on the file system. A MongoDB server can store multiple databases.

5. **Document** – A record in a MongoDB collection is basically called a document. The document, in turn, will consist of field name and values.

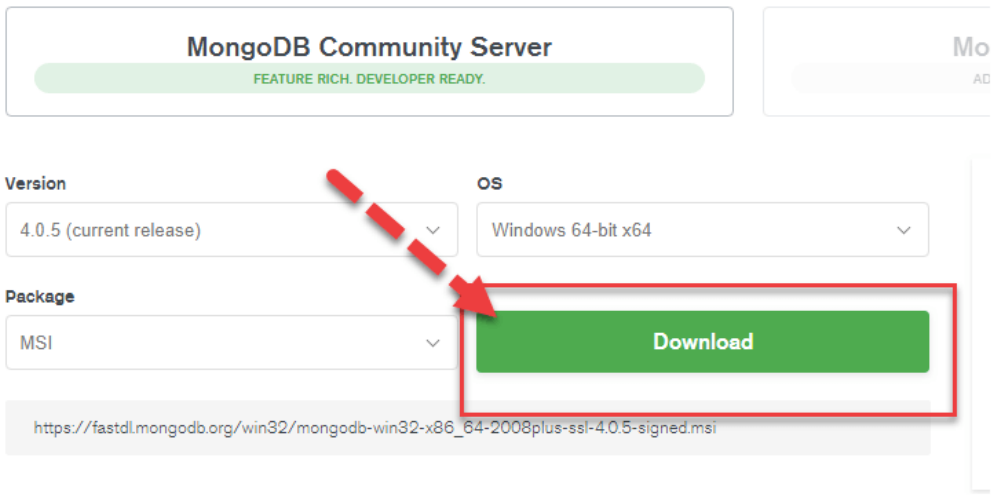
6. **Field** – A name-value pair in a document. A document has zero or more fields. Fields are analogous to columns in relational databases.The following diagram shows an example of Fields with Key value pairs. So in the example below CustomerID and 11 is one of the key value pair's defined in the document.



7. JSON – This is known as JavaScript Object Notation. This is a human-readable, plain text format for expressing structured data. JSON is currently supported in many programming languages.

Installation of MongoDB Community Server:

1. Download MongoDB Community Server Go to this link: https://www.mongodb.com/try/download/community and download MongoDB Community Server.



MongoDB Community Server

FEATURE RICH. DEVELOPER READY.

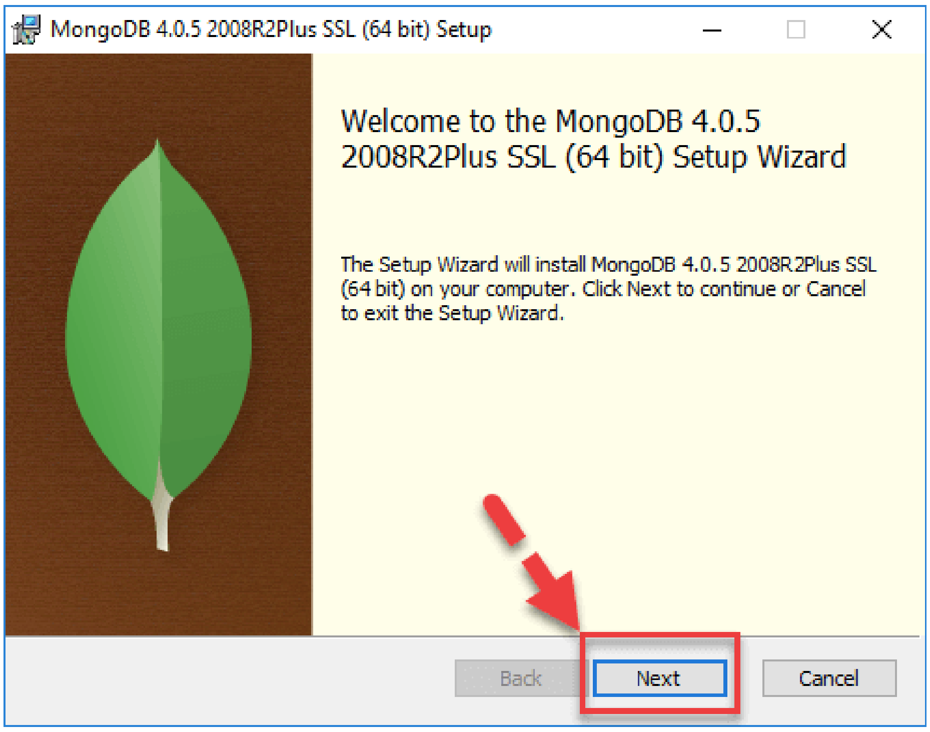
Version: 4.0.5 (current release)

OS: Windows 64-bit x64

Package: MSI

https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.5-signed.msi

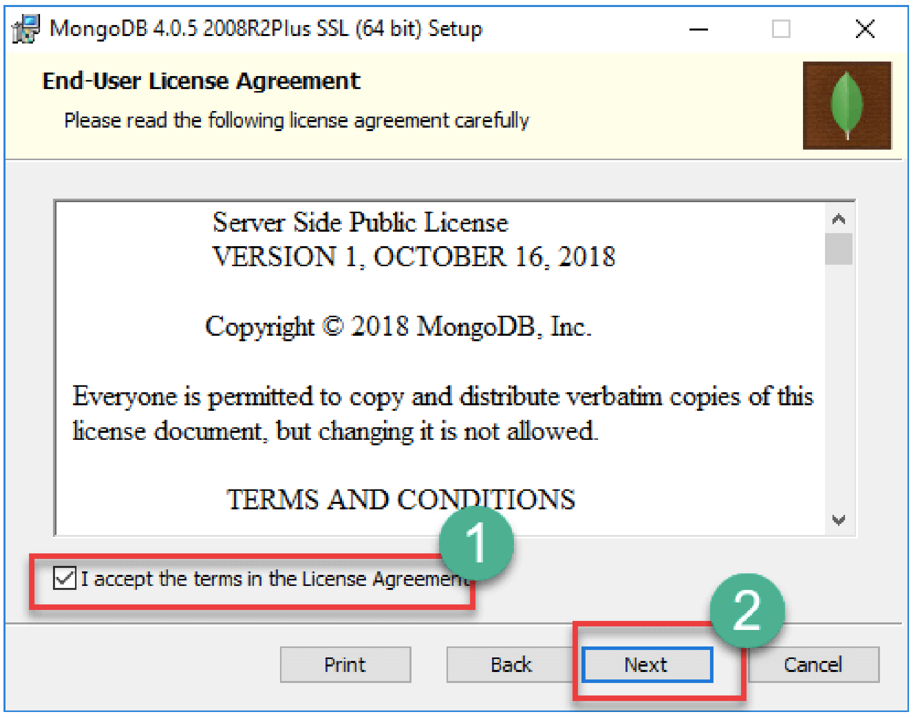
2. Once download is complete open the msi file. Click Next in the start up screen


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 4.0.5 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

3. Accept the End-User License Agreement and click Next.


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

End-User License Agreement

Please read the following license agreement carefully

Server Side Public License
VERSION 1, OCTOBER 16, 2018

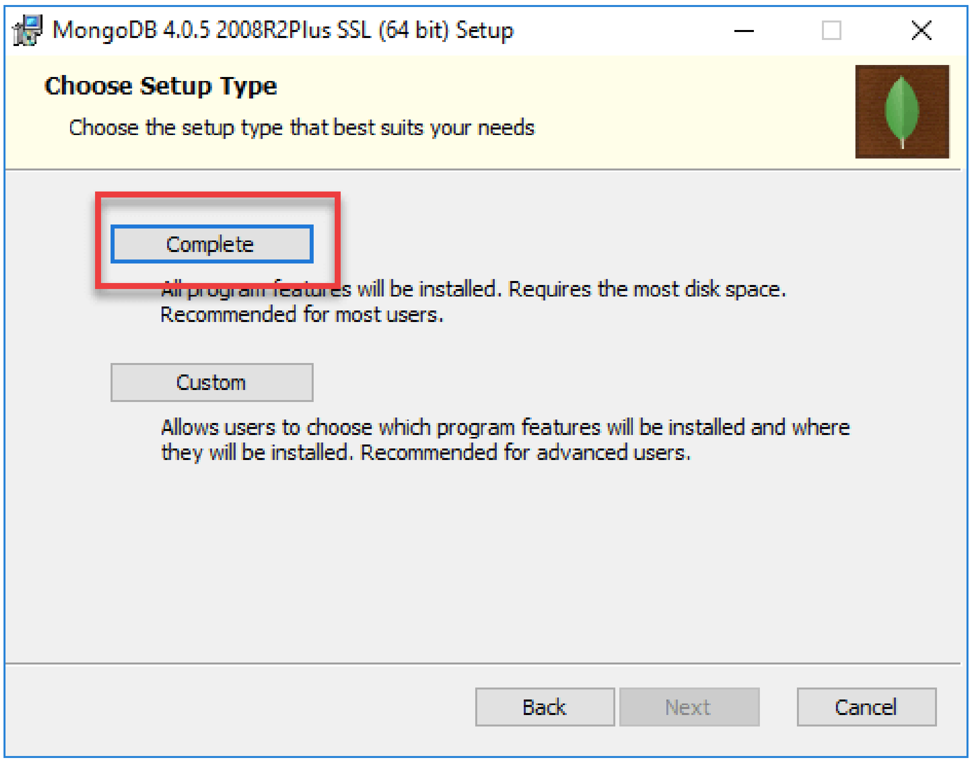
Copyright © 2018 MongoDB, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

TERMS AND CONDITIONS

☑ I accept the terms in the License Agreement

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.


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

Choose the setup type that best suits your needs

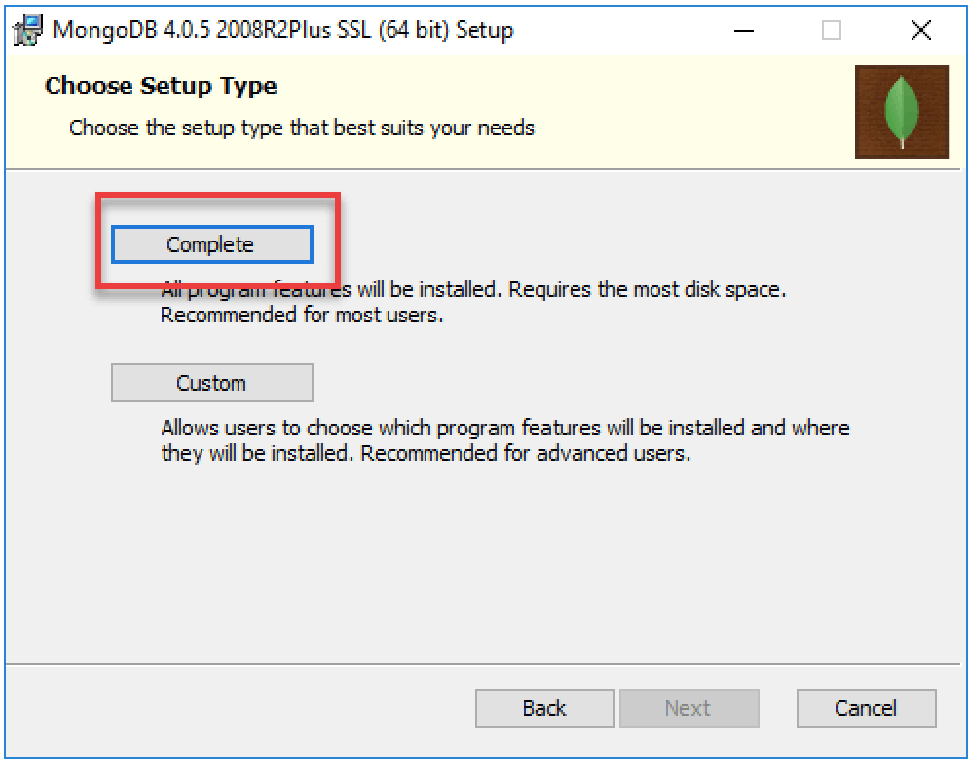
Complete

All program features will be installed. Requires the most disk space. Recommended for most users.

Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

5. Select “Run service as Network Service user”. make a note of the data directory, we’ll need this later and click Next.


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Choose Setup Type

Choose the setup type that best suits your needs

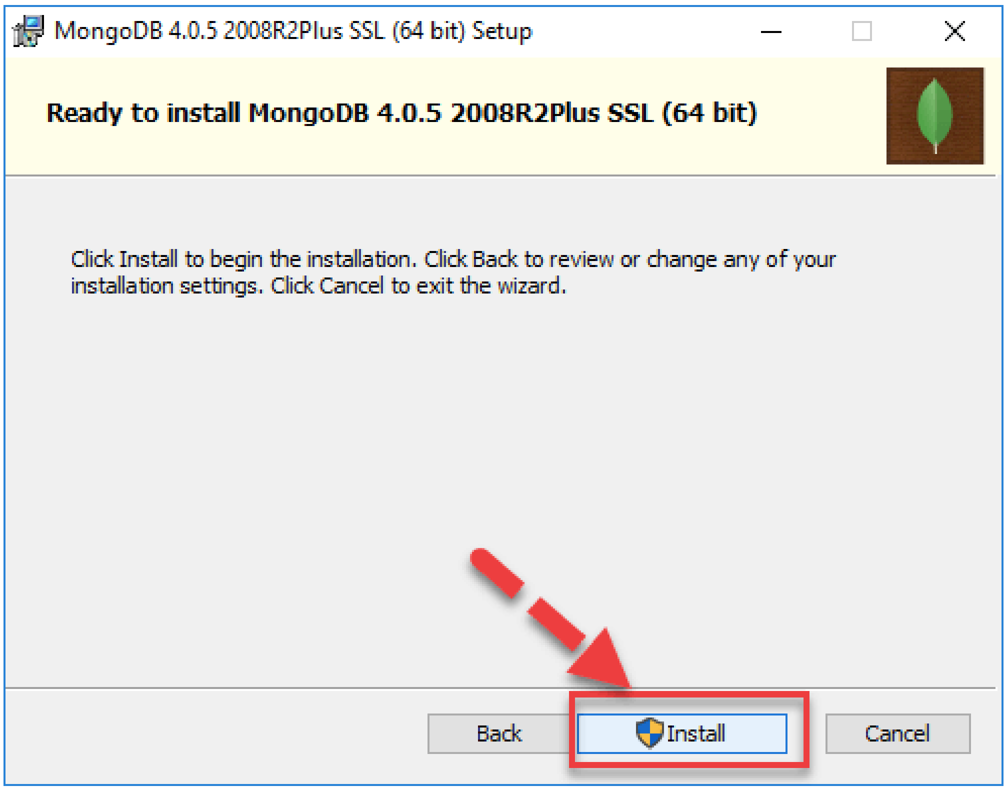
Complete

All program features will be installed. Requires the most disk space. Recommended for most users.

Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

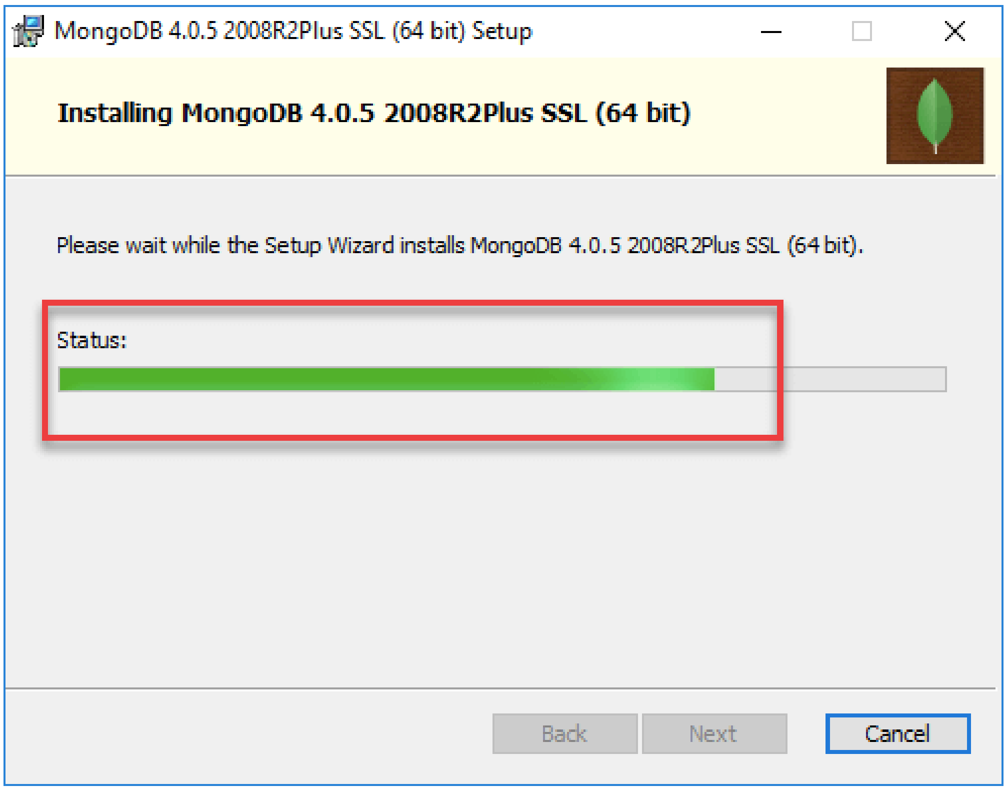
6. Click on the Install button to start the installation.


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 4.0.5 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

7. Installation begins. Click Next once completed

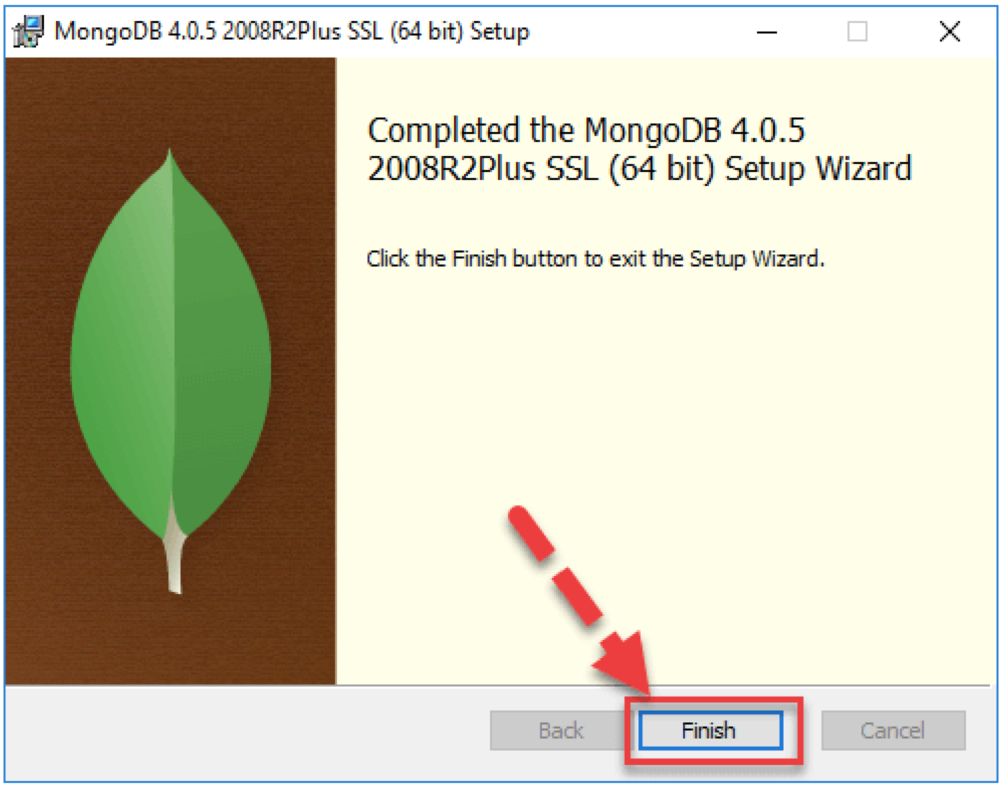

MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Installing MongoDB 4.0.5 2008R2Plus SSL (64 bit)

Please wait while the Setup Wizard installs MongoDB 4.0.5 2008R2Plus SSL (64 bit).

Status:

8. Final step, Once complete the installation, Click on the Finish button


MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup

Completed the MongoDB 4.0.5 2008R2Plus SSL (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

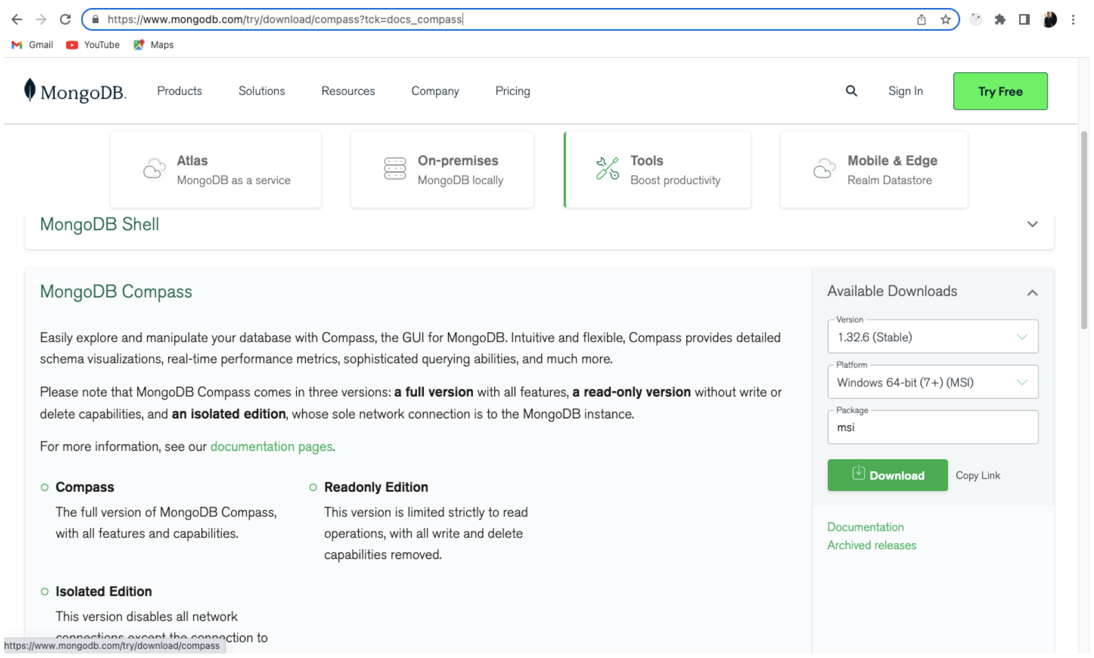
Now, after installing MongoDB on our computer we will install MongoDB Compass - a management tool for MongoDB.

Installation of MongoDB Compass:

There are tools in the market which are available for managing MongoDB. One such non-commercial tool is MongoDB Compass.

Some of the features of Compass are given below:

1. Full power of the Mongoshell
2. Multiple shells
3. Multiple results
4. Go to the link: https://www.mongodb.com/try/download/compass?tck=docs_compass and download MongoDB Compass as per your OS.


https://www.mongodb.com/try/download/compass?tck=docs_compass

MongoDB Compass

Easily explore and manipulate your database with Compass, the GUI for MongoDB. Intuitive and flexible, Compass provides detailed schema visualizations, real-time performance metrics, sophisticated querying abilities, and much more.

Please note that MongoDB Compass comes in three versions: **a full version** with all features, **a read-only version** without write or delete capabilities, and **an isolated edition**, whose sole network connection is to the MongoDB instance.

For more information, see our documentation pages.

Compass
The full version of MongoDB Compass, with all features and capabilities.

Readonly Edition
This version is limited strictly to read operations, with all write and delete capabilities removed.

Isolated Edition
This version disables all network connections except the connection to

Available Downloads

Version: 1.32.6 (Stable)

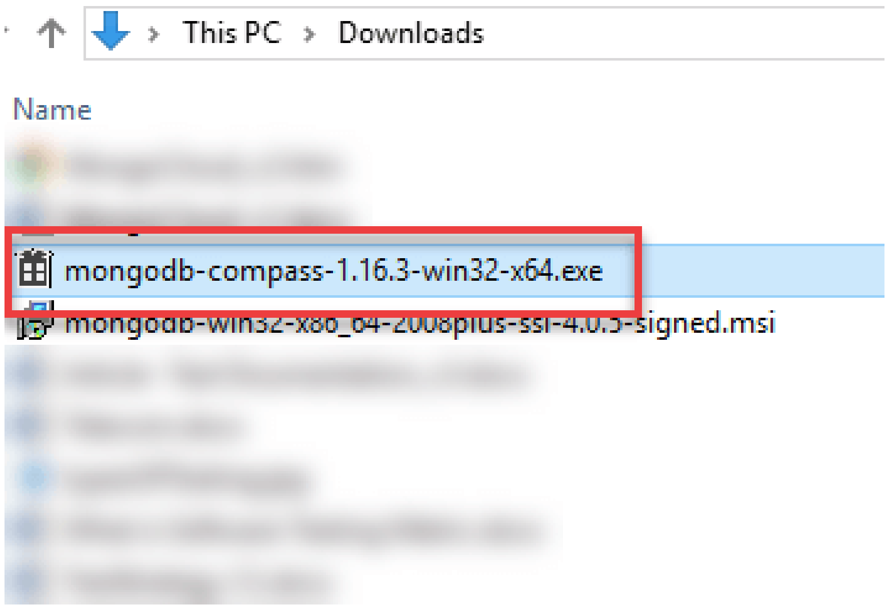
Platform: Windows 64-bit (7+) (MSI)

Package: msi

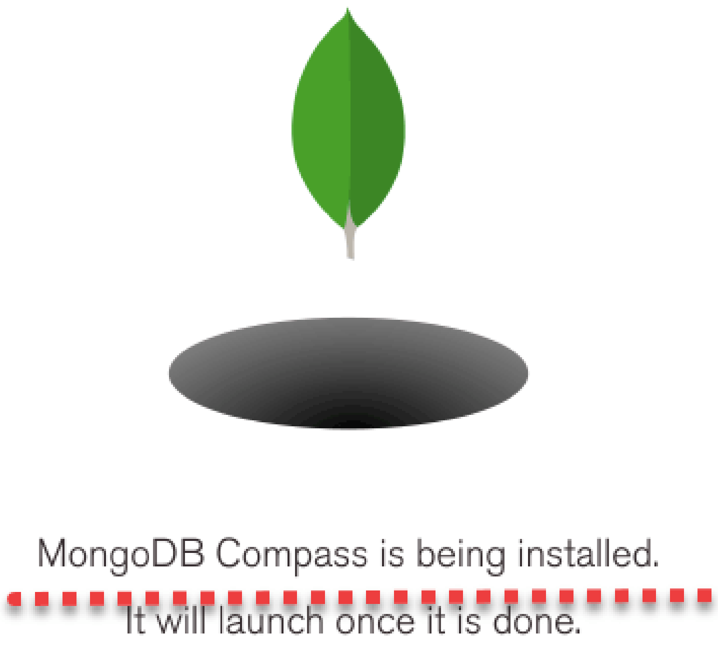
https://www.mongodb.com/try/download/compass

Note: While downloading it might ask for details so fill all the details and click submit, then downloading will start.

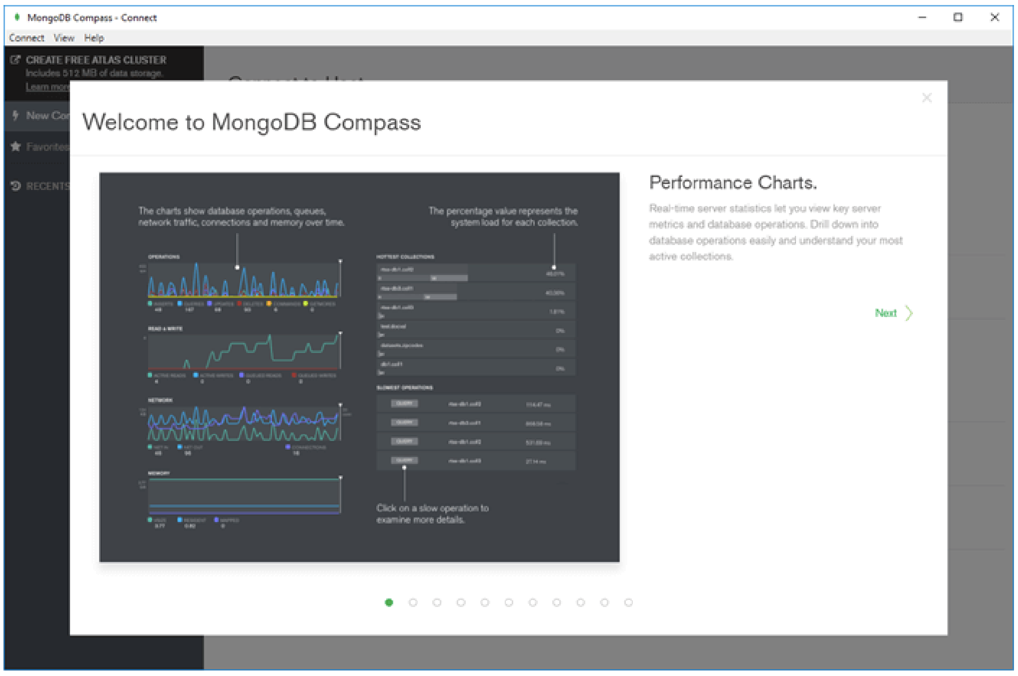
2. Double click on the downloaded file


This PC › Downloads
Name
mongodb-compass-1.16.3-win32-x64.exe
mongodb-win32-x86_64-2008plus-ssl-4.0.5-signed.msi

3. Installation will auto-start


MongoDB Compass is being installed.
It will launch once it is done.

4. Compass will launch with a Welcome screen

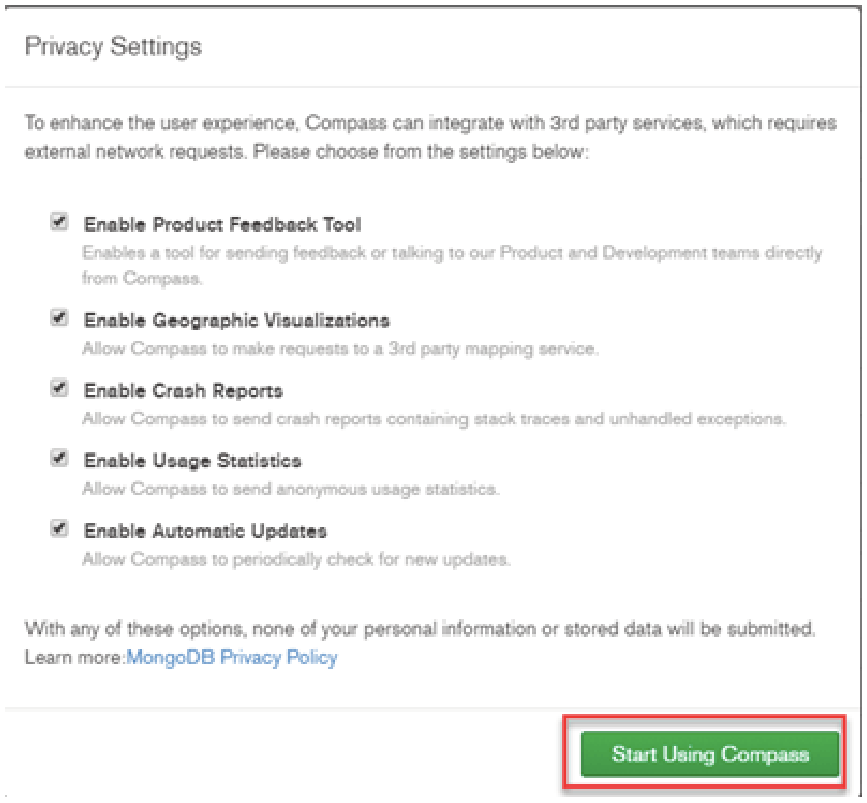

MongoDB Compass - Connect
Connect View Help
CREATE FREE ATLAS CLUSTER
The charts show database operations, queues, network traffic, connections and memory over time.
The percentage value represents the system load for each collection.
Click on a slow operation to examine more details.
Next

Welcome to MongoDB Compass

Performance Charts.

Real-time server statistics let you view key server metrics and database operations. Drill down into database operations easily and understand your most active collections.

5. Keep the privacy settings as default and Click “Start Using Compass”



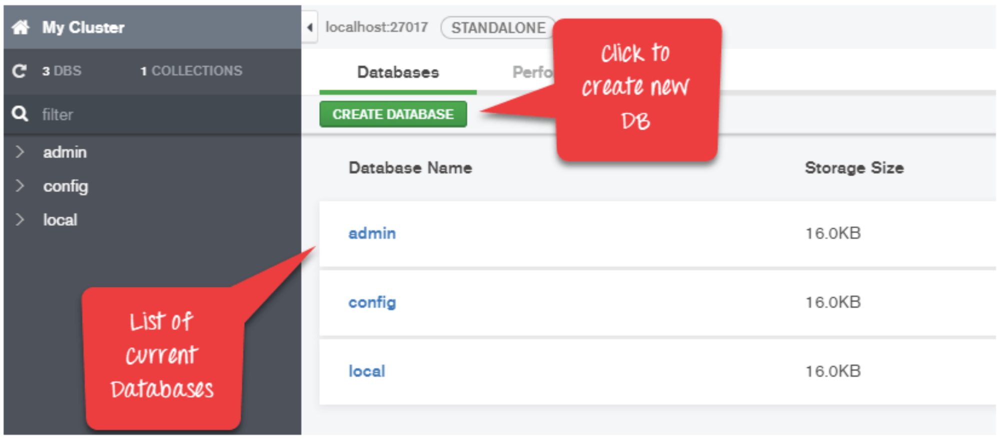
Privacy Settings

To enhance the user experience, Compass can integrate with 3rd party services, which requires external network requests. Please choose from the settings below:

- ✓ **Enable Product Feedback Tool**
  Enables a tool for sending feedback or talking to our Product and Development teams directly from Compass.
- ✓ **Enable Geographic Visualizations**
  Allow Compass to make requests to a 3rd party mapping service.
- ✓ **Enable Crash Reports**
  Allow Compass to send crash reports containing stack traces and unhandled exceptions.
- ✓ **Enable Usage Statistics**
  Allow Compass to send anonymous usage statistics.
- ✓ **Enable Automatic Updates**
  Allow Compass to periodically check for new updates.

With any of these options, none of your personal information or stored data will be submitted.
Learn more:MongoDB Privacy Policy

6. You will see homescreen with list of current databases.



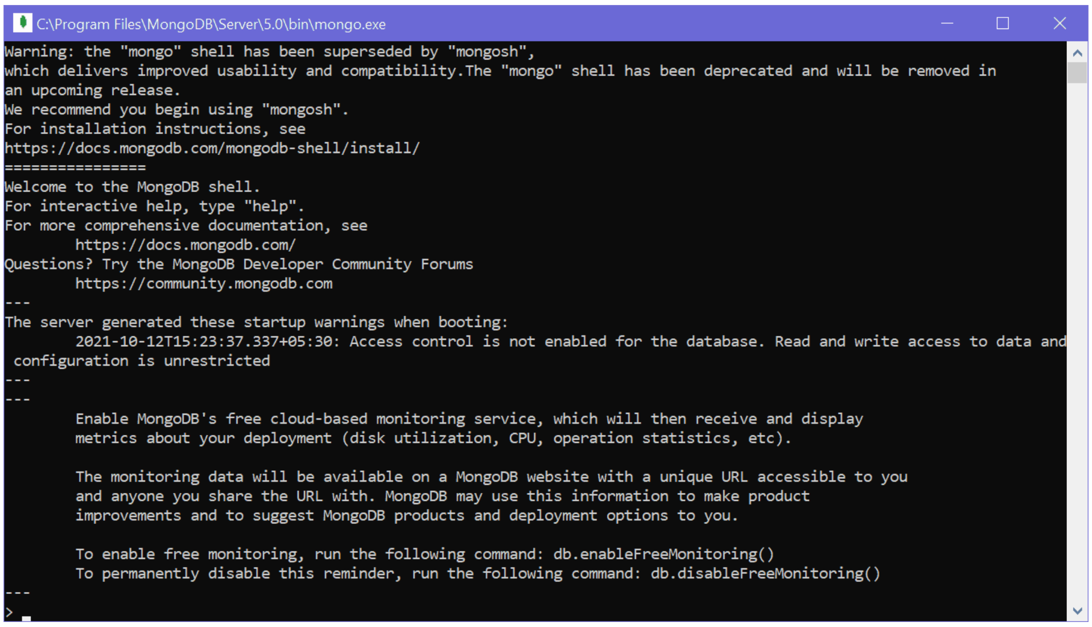
Database Name	Storage Size
admin	16.0KB
config	16.0KB
local	16.0KB

MongoDB Shell:

MongoDB Shell is the quickest way to connect, configure, query, and work with your MongoDB database. It acts as a command-line client of th MongoDB server.

The MongoDB Shell is a standalone, open-source product and developed separately from the MongoDB Server under the Apache 2 license. It is a ful functional JavaScript and Node.js 14.x REPL for interacting with MongoDB servers.

MongoDB Shell is already installed with MongoDB. You can find it in the installation directory where you installed MongoDB. By default, it is "C:\Progra Files\MongoDB\Server". Open the installation folder and appropriate version folder and go to the "bin" folder. Here, "mongo.exe" is MongoDB shell. Cli on it to open the MongoDB shell, as shown below.



```
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2021-10-12T15:23:37.337+05:30: Access control is not enabled for the database. Read and write access to data and
 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()
---
> _
```

Now, as we are in the MongoDB Shell, now we are ready to create our new database:

Creating a Database:

A database is a place where data is stored in an organized way. In MongoDB, databases are used to store collections. A single MongoDB server ca have multiple databases and a single MongoDB database can have multiple collections.

You can use MongoDB Shell or MongoDB Compass to create a new database.

Using MongoDB Shell:

MongoDB provides the `use <database-name` command to connect with the database. If the specified database name does not exist then it creates and set it as a current database.

Syntax

Basic syntax of **use DATABASE** statement is as follows −

```
use DATABASE_NAME
```

Example

If you want to use a database with name , then **use DATABASE** statement would be as follows −

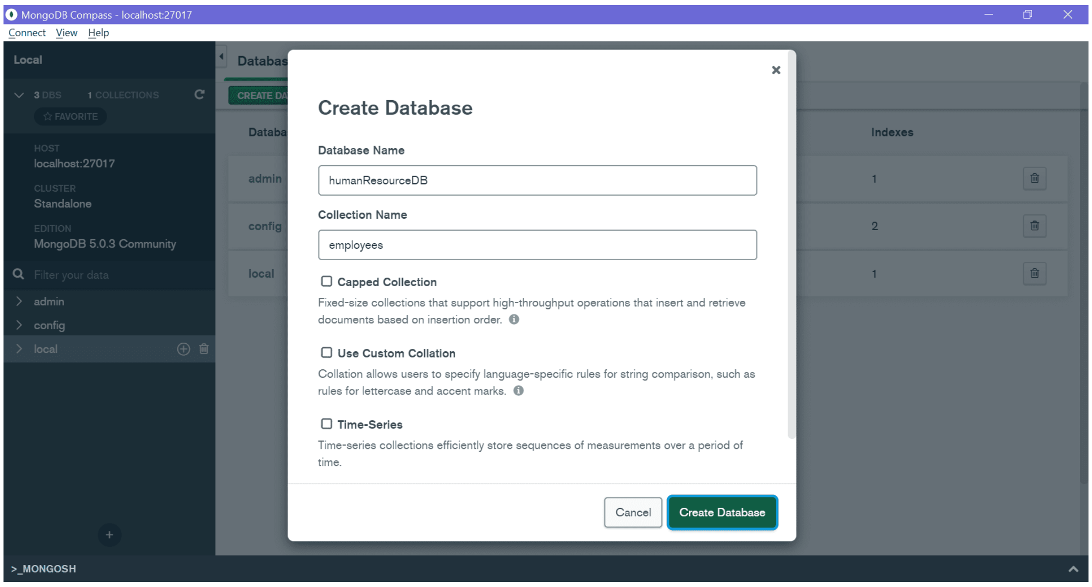
```
>use humanResourceDb
switched to db humanResourceDb
```

To check your currently selected database, use the command **db**

```
>db
humanResourceDb
```

Using MongoDB Compass:

You can create a new database using MongoDB Compass. For that, open Compass and connect with your local or remote database. Once it connec with the MongoDB server, click on the top "CREATE DATABASE" button which will open the popup window, as shown below.



MongoDB Compass - localhost:27017

Create Database

Database Name: humanResourceDB

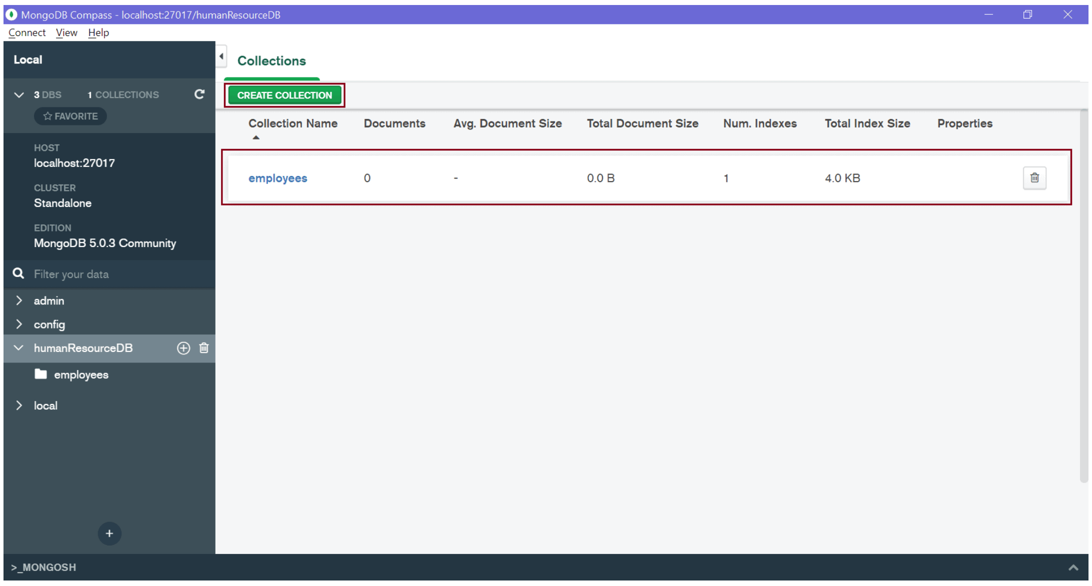
Collection Name: employees

☐ Capped Collection
Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order.

☐ Use Custom Collation
Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks.

☐ Time-Series
Time-series collections efficiently store sequences of measurements over a period of time.

Enter your database name and collection name and click Create Database. This will create a new database humanResourceD with th new employees collection shown below.



MongoDB Compass - localhost:27017/humanResourceDB

Collections

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
employees	0	-	0.0 B	1	4.0 KB	

Thus, you can create a new database in MongoDB.

Droping a Database:

MongoDB **db.dropDatabase()** command is used to drop a existing database.

Syntax

Basic syntax of **dropDatabase()** command is as follows −

```
db.dropDatabase()
```

This will delete the selected database. If you have not selected any database, then it will delete default 'test' database.

If you want to delete new database , then **dropDatabase()** command would be as follows −

```
>use humanResourceDb
switched to db humanResourceDb
>db.dropDatabase()
>{ "dropped" : "humanResourceDb", "ok" : 1 }
>
```

Creating a Collection in MongoDB:

Creation of collection can be done using **db.createCollection(name, options)**. But, typically you will not require building or constructing a collection your own. MongoDB does this creation task automatically as you start to insert some documents for making a database. Here is a syntax that will tell yc how to create your collection in MongoDB:

Syntax:

```
db.createCollection(collection_name, options)
```

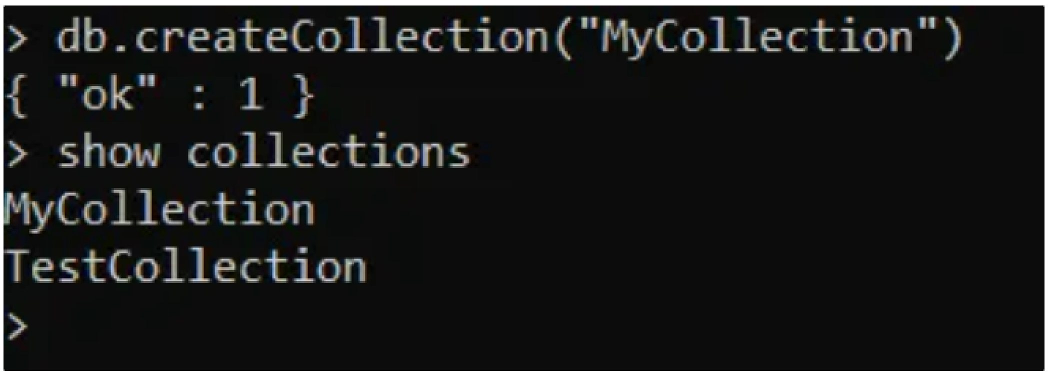
Here, db.createCollection() is the method used; "name" is a data type - string which is specifying the name of the collection to be formed. "options" is a added document type which helps in specifying the size of memory along with indexing the collection in the database. This parameter is an optional one

Let us take an example to see how to implement the command in MongoDB:

Example:

```
db.createCollection("MyCollection")
```

Output:



```
> db.createCollection("MyCollection")
{ "ok" : 1 }
> show collections
MyCollection
TestCollection
>
```

MongoDB can create collections automatically as documents are inserted automatically.

Let us take a situation where you want to insert a document using **insert()** in a collection named "movie"; you can type the command something like this

Example:

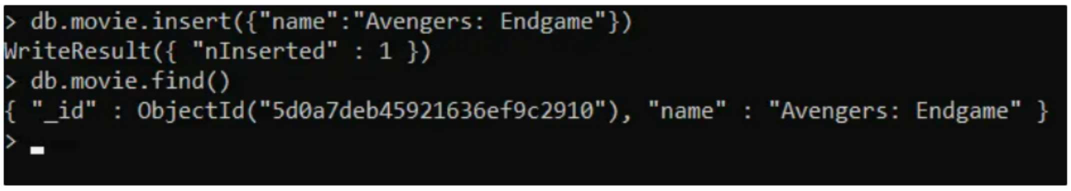
```
db.movie.insert({"name":"Avengers: Endgame"})
```

The above operation will automatically create a collection if the collection with this name does not currently exist. Also, if you want to check an inserte document, you can use the **find()** method. Its syntax is:

Syntax:

```
db.collection_name.find()
```

Output:



```
> db.movie.insert({"name":"Avengers: Endgame"})
WriteResult({ "nInserted" : 1 })
> db.movie.find()
{ "_id" : ObjectId("5d0a7deb45921636ef9c2910"), "name" : "Avengers: Endgame" }
> 
```

Dropping a Collection in MongoDB:

For dropping a collection in MongoDB, you have to make use of the **collection.drop()** method. This will eliminate the collection from the databas completely and will not leave any indexes that are connected to this drop collection. It is also to be noted that, this collection dropping method has r argument in it and will pop up with errors in case arguments are passed. All the indexes associated with the collection gets dropped once this method executed.

The syntax for using this method is:

Syntax:

```
db.collection_name.drop()
```

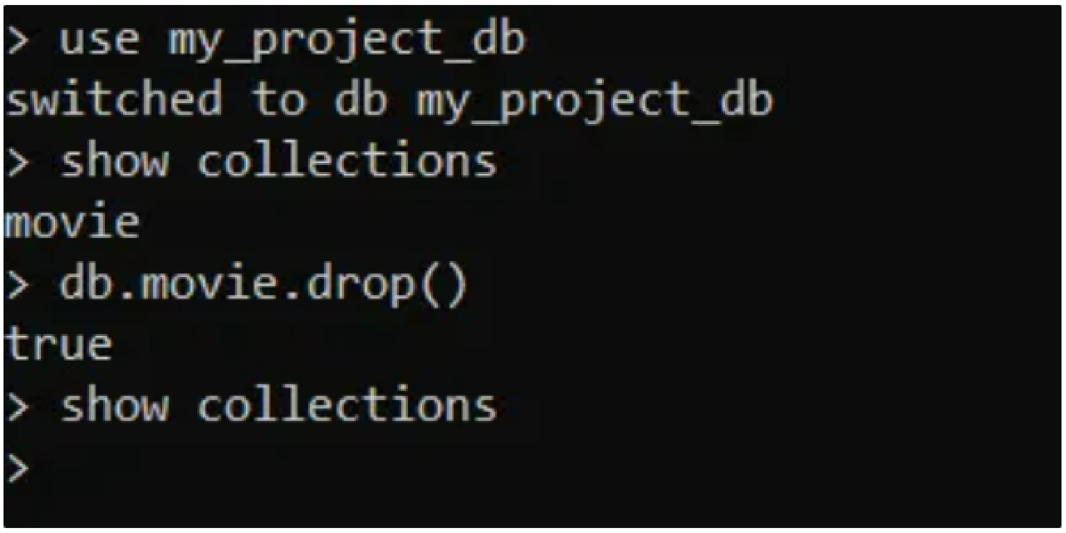
Here is an example that is showing the use of the drop() method:

Continuing from the example in the previous lesson where you have created a collection inside the **my_project_db** database. Now use this database remove or drop the **movie** collection and see the changes:

Example:

```
use my_project_db
db.movie.drop()
show collections
```

Output:



```
> use my_project_db
switched to db my_project_db
> show collections
movie
> db.movie.drop()
true
> show collections
>
```

Conclusion:

In this session, we have learnt about:

- What is MongoDB ?
- Advantages of using MongoDB
- Installing MongoDB Community Server and MongoDB Compass on our computers
- Creating and Deleting Database
- Creating and Dropping

Interview Questions:

Is MongoDB better than SQL databases ? If yes then how ?

```
MongoDB is better than other SQL databases because it allows a highly flexible and scalable document structure.

**For example:**

- One data document in MongoDB can have five columns and the other one in the same collection can have ten columns.
- MongoDB database are faster than SQL databases due to efficient indexing and storage techniques.
```

List out the important features of MongoDB.

```
Some of the important features of MongoDB are:

- Uses a schema-less database
- No complex joins
- Faster access to data because of the presence of the working set (internal memory)
- Features like aggregation, sharding, and replication make it easy to use
- Cross-platform and document-based
- Automatic fail-over and high-availability
```

Does MongoDB uses BSON or JSON ?

```
MongoDB stores data in BSON format both internally, and over the network, but that doesn't mean you can't think of Mongo

Unlike systems that simply store JSON as string-encoded values, or binary-encoded blobs, MongoDB uses BSON to offer the
```

Thank You