MongoDB GridFS

GridFS is a frame work to store & access large set of data. It divides the data into chunks and store then into different documents.

-API Provided by MongoDb for storing large files such as audio, video and images.

-Package that can be plucked into any application to make storing large files easier

Provides a way for storing large files in database instead of in the file system.

MongoDB GridFS

Problem: In MongoDB document size is limited to 16 MB.

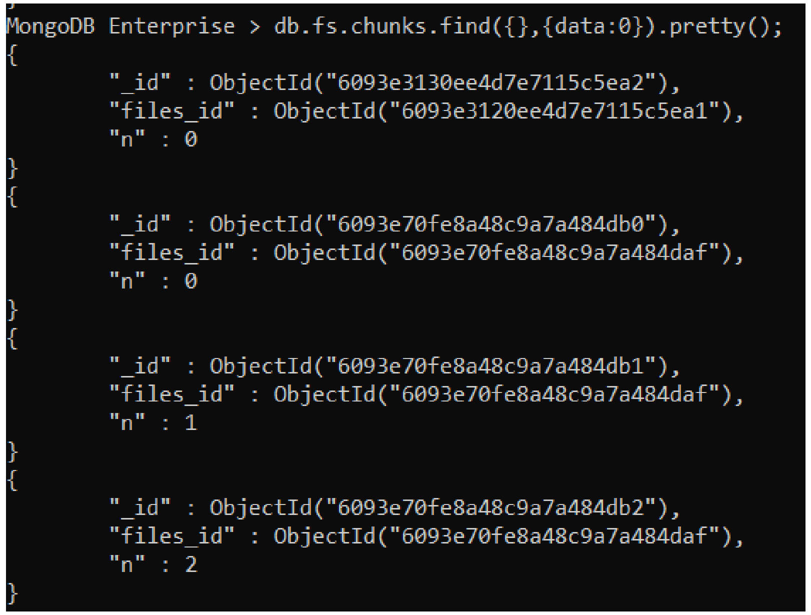
Gridfs Solves the size limitation problem

1. Breaks the files to smaller managable chunks
2. Stores these chunks of data in one collection called **fs.chunks**
3. Stores the information about the whole file itself in another collection called **fs.files**
4. Connects these documents by properties that are references to each other

MongoDB GridFS

fs.chunks collection

1. The size of each chunk is 255KB
2. No. of chunks created depends on the file size
3. Chunks stores the actual data.
4. Each chunk is linked to the fille information by “files_id” property.
5. The “files_id” points to a document that is stored in fs.files collection.

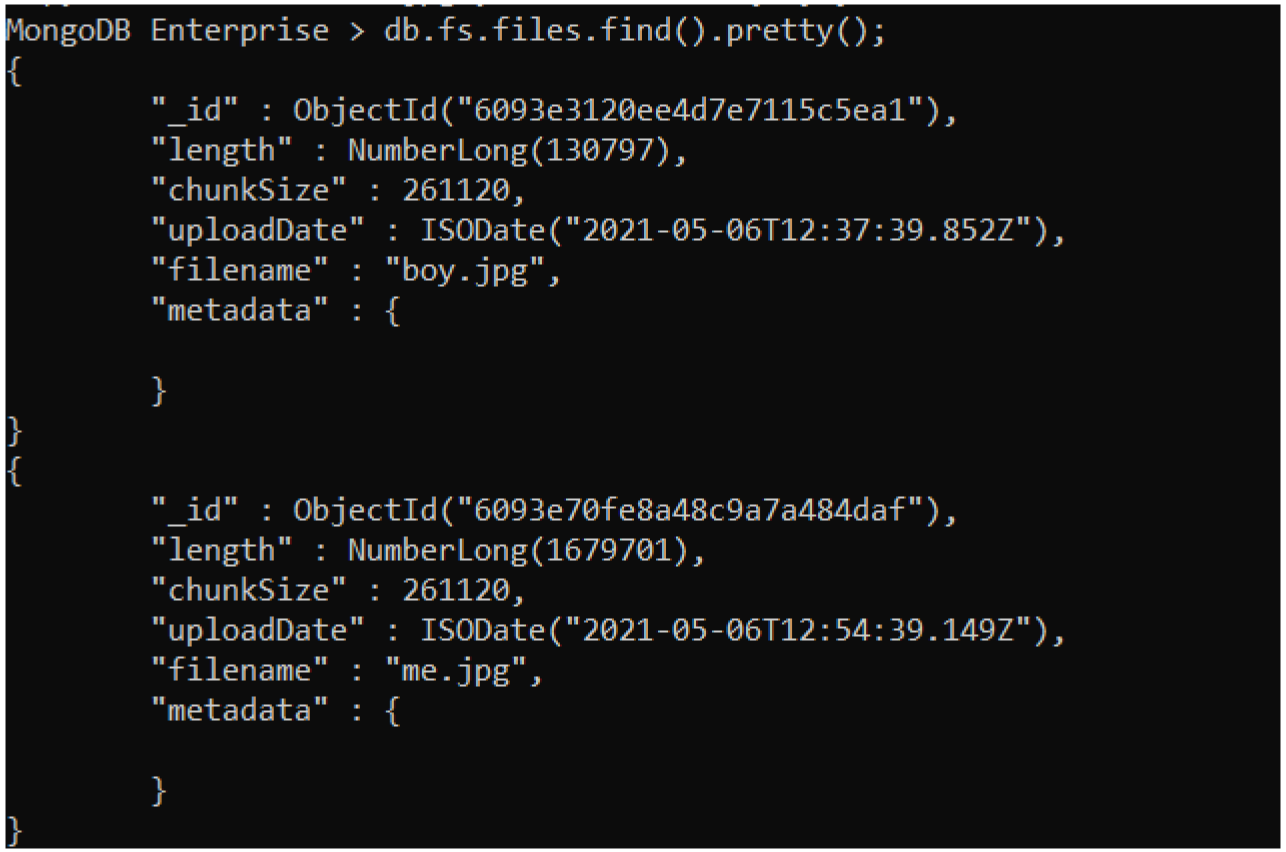


```
MongoDB Enterprise > db.fs.chunks.find({},{data:0}).pretty();
{
        "_id" : ObjectId("6093e3130ee4d7e7115c5ea2"),
        "files_id" : ObjectId("6093e3120ee4d7e7115c5ea1"),
        "n" : 0
}
{
        "_id" : ObjectId("6093e70fe8a48c9a7a484db0"),
        "files_id" : ObjectId("6093e70fe8a48c9a7a484daf"),
        "n" : 0
}
{
        "_id" : ObjectId("6093e70fe8a48c9a7a484db1"),
        "files_id" : ObjectId("6093e70fe8a48c9a7a484daf"),
        "n" : 1
}
{
        "_id" : ObjectId("6093e70fe8a48c9a7a484db2"),
        "files_id" : ObjectId("6093e70fe8a48c9a7a484daf"),
        "n" : 2
}
```

MongoDB GridFS

fs.files collection contains the information about the file

1. File name
2. Average size of each chunk
3. Upload date
4. Size of file (in bytes)
5. File metadata



```
MongoDB Enterprise > db.fs.files.find().pretty();
{
        "_id" : ObjectId("6093e3120ee4d7e7115c5ea1"),
        "length" : NumberLong(130797),
        "chunkSize" : 261120,
        "uploadDate" : ISODate("2021-05-06T12:37:39.852Z"),
        "filename" : "boy.jpg",
        "metadata" : {

        }
}
{
        "_id" : ObjectId("6093e70fe8a48c9a7a484daf"),
        "length" : NumberLong(1679701),
        "chunkSize" : 261120,
        "uploadDate" : ISODate("2021-05-06T12:54:39.149Z"),
        "filename" : "me.jpg",
        "metadata" : {

        }
}
```

MongoDB GridFS

The **mongofiles** utility makes it possible to manipulate files stored in your MongoDB instance in GridFS objects from the command line.

The mongofiles tool is part of the MongoDB Database Tools package.

https://www.mongodb.com/try/download/database-tools

Run mongofiles from the system command line, not the mongo shell.

MongoDB GridFS

```
mongofiles <options> <connection-string> <command> <filename or _id>
```

Options. You may use one or more of these options to control the behaviour of mongofiles.

Connection String. The connection string of the mongod to connect to with mongofiles.

Command - Use one of these commands to determine the action of mongofiles.

Filename - name of the file to be saved in the data base

```
C:\....\bin>mongofiles put me.jpg --db=img
```

```
C:\....\bin>mongofiles --help
```

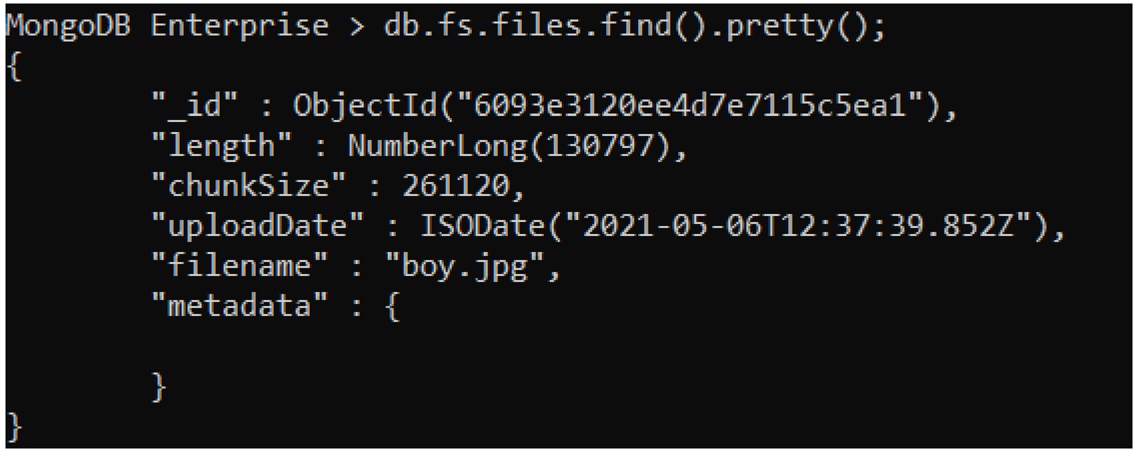
```
C:\....\bin>mongofiles get me.jpg --db=img
```

```
files.find().pretty();
```

MongoDB GridFS

Once the file is stored in the database fs.files & fs.chunks collection can be used to get the information about the file.

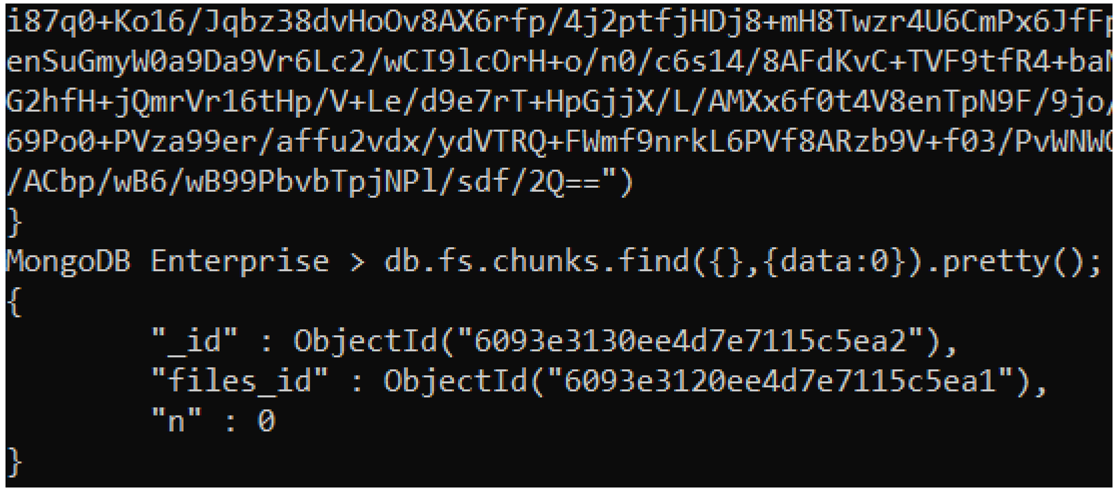
>db.fs.files.find().pretty();



```
MongoDB Enterprise > db.fs.files.find().pretty();
{
        "_id" : ObjectId("6093e3120ee4d7e7115c5ea1"),
        "length" : NumberLong(130797),
        "chunkSize" : 261120,
        "uploadDate" : ISODate("2021-05-06T12:37:39.852Z"),
        "filename" : "boy.jpg",
        "metadata" : {

        }
}
```

>db.fs.chunks.find().pretty();



```
i87q0+Ko16/Jqbz38dvHoOv8AX6rfp/4j2ptfjHDj8+mH8Twzr4U6CmPx6JfFp
enSuGmyW0a9Da9Vr6Lc2/wCI9lcOrH+o/n0/c6s14/8AFdKvC+TVF9tfR4+baM
G2hfH+jQmrVr16tHp/V+Le/d9e7rT+HpGjjX/L/AMXx6f0t4V8enTpN9F/9jo/
69Po0+PVza99er/affu2vdx/ydVTRQ+FWmf9nrkL6PVf8ARzb9V+f03/PvWNWC
/ACbp/wB6/wB99PbvbTpjNPl/sdf/2Q==")
}
MongoDB Enterprise > db.fs.chunks.find({},{data:0}).pretty();
{
        "_id" : ObjectId("6093e3130ee4d7e7115c5ea2"),
        "files_id" : ObjectId("6093e3120ee4d7e7115c5ea1"),
        "n" : 0
}
```

MongoDB GridFS

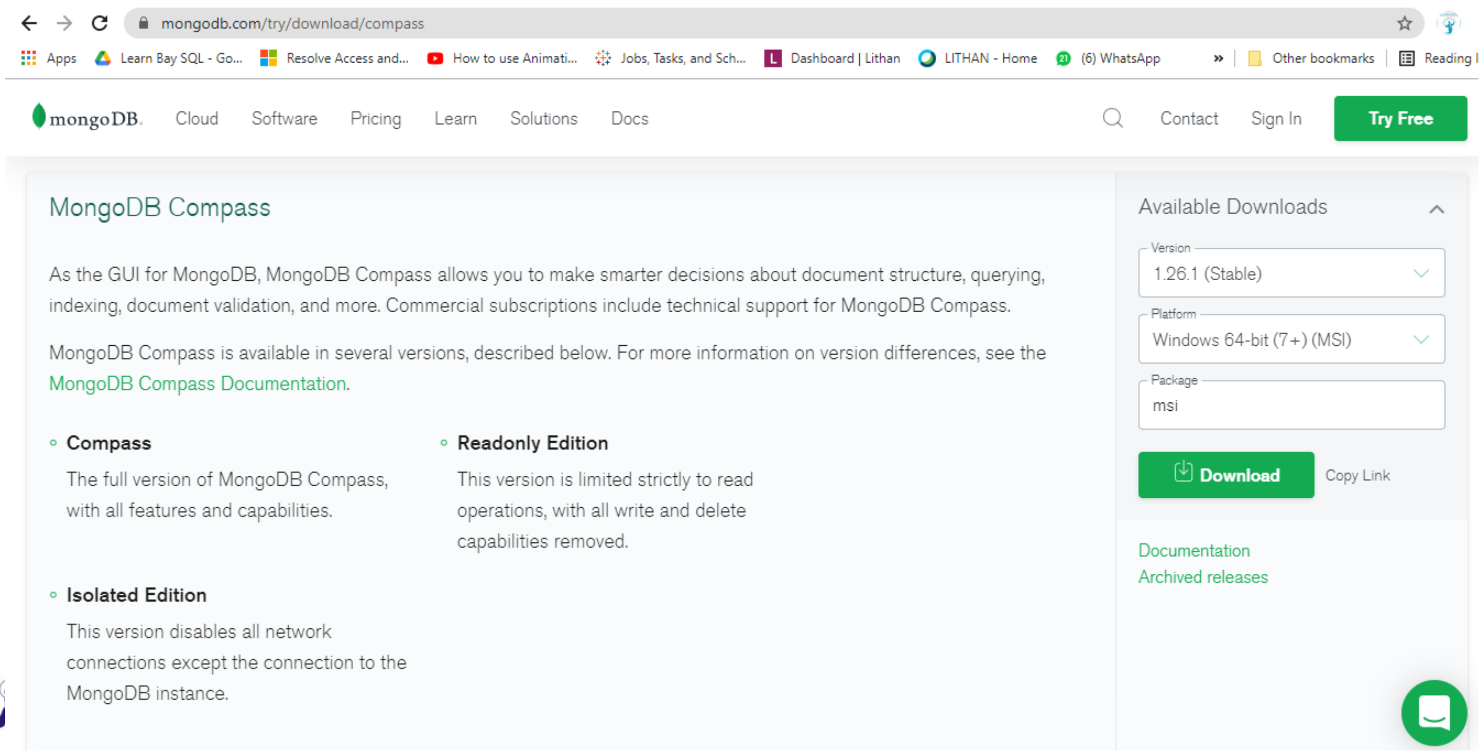
Getting the no of chunks created for each file:

```
>db.fs.chunks.find(
                    {"files_id" : ObjectId("64a7932d07bbfa87bef08aa6")}
          ).count()
```

MongoDB Compass

It’s a GUI Interface for handling MongoDb database. It is a convenient tool for performing all CRUD operations without manually writing queries. It helps in many activities such as indexing, document validation, etc.

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



MongoDB Compass

As the GUI for MongoDB, MongoDB Compass allows you to make smarter decisions about document structure, querying, indexing, document validation, and more. Commercial subscriptions include technical support for MongoDB Compass.

MongoDB Compass is available in several versions, described below. For more information on version differences, see the MongoDB Compass Documentation.

- **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 the MongoDB instance.

Available Downloads

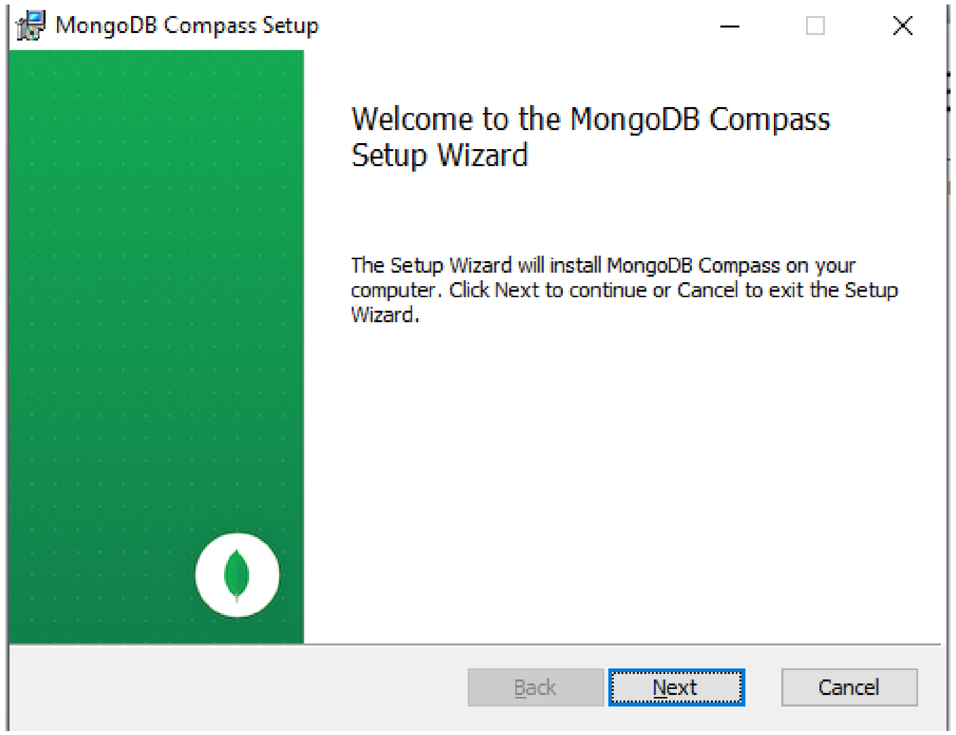
Version: 1.26.1 (Stable)

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

Package: msi

MongoDB Compass

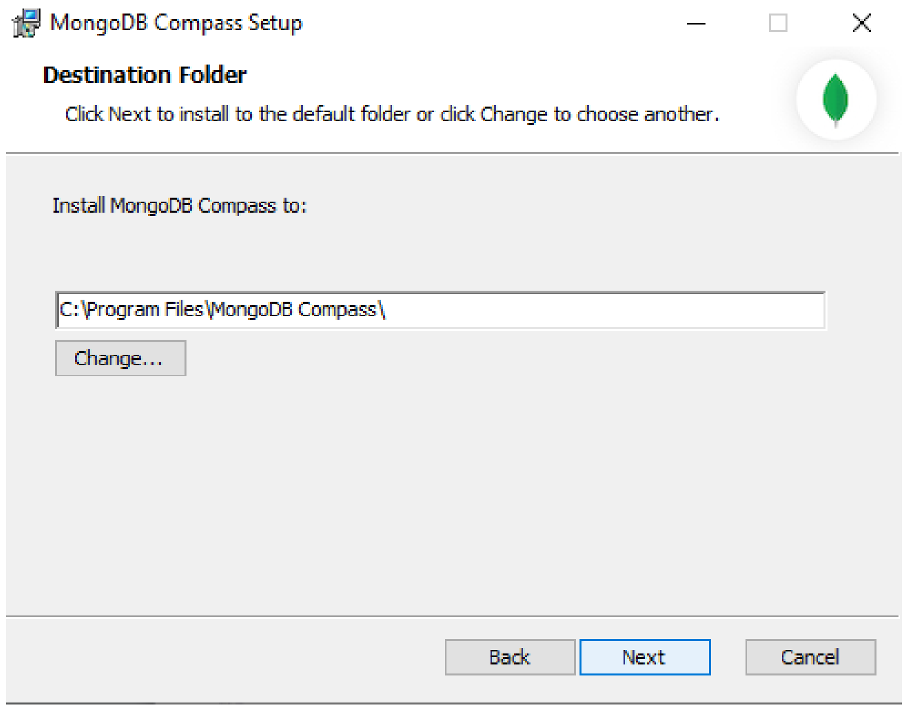
Double click on the downloaded file to initiate the installation:



MongoDB Compass Setup

Welcome to the MongoDB Compass Setup Wizard

The Setup Wizard will install MongoDB Compass on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



MongoDB Compass Setup

Destination Folder

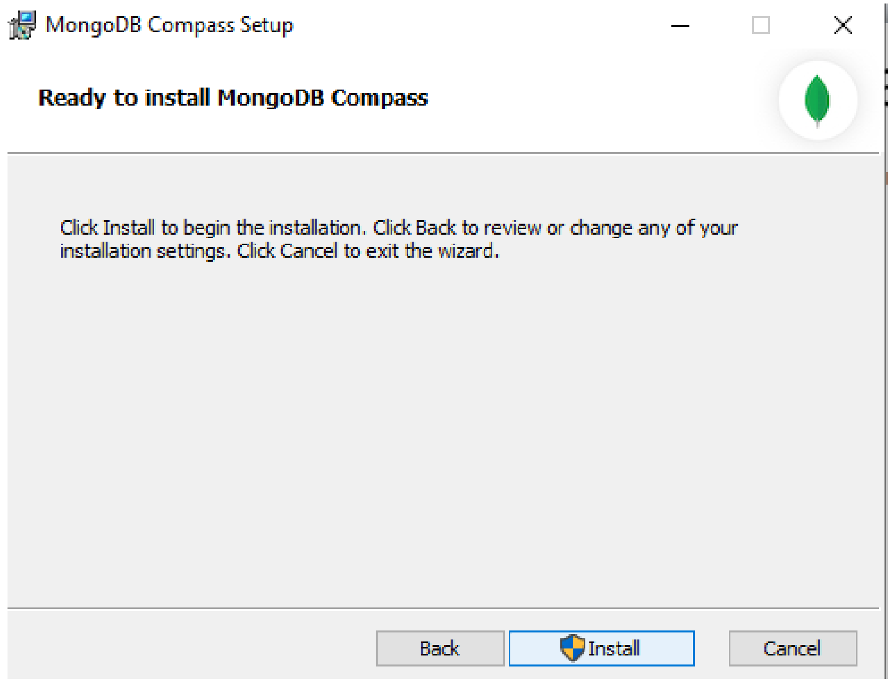
Click Next to install to the default folder or click Change to choose another.

Install MongoDB Compass to:

C:\Program Files\MongoDB Compass\

MongoDB Compass

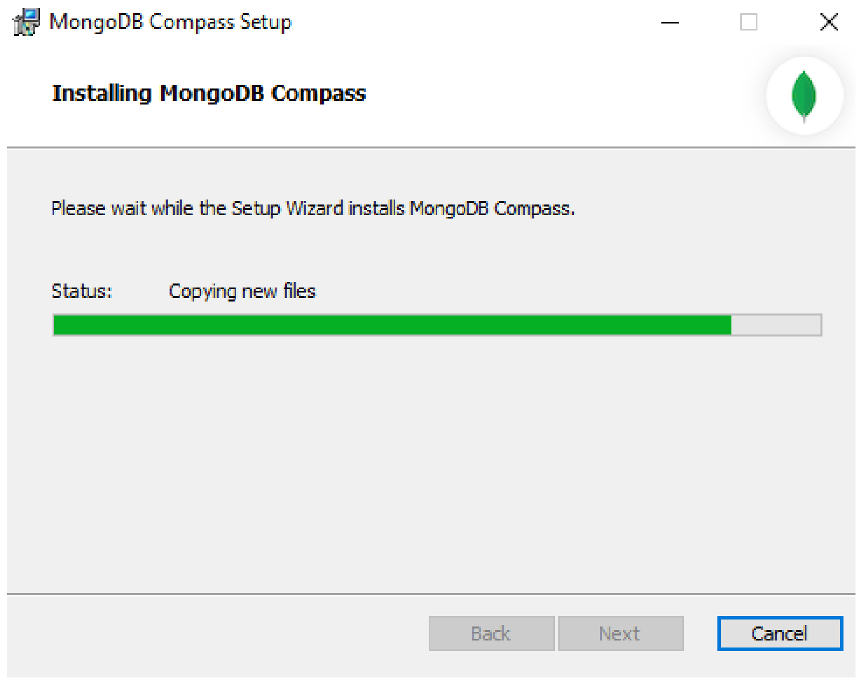
Click on Install proceed with the installation:



MongoDB Compass Setup

Ready to install MongoDB Compass

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



MongoDB Compass Setup

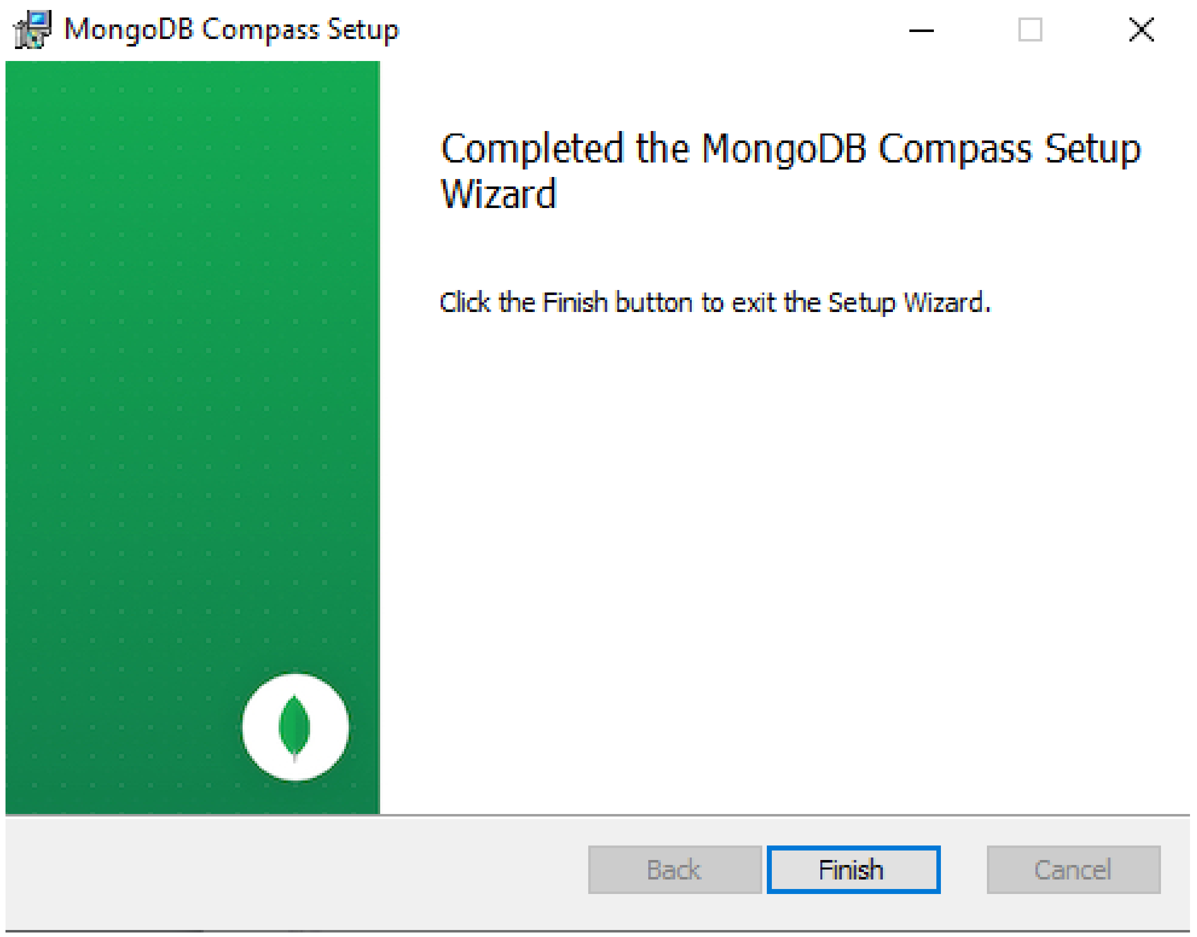
Installing MongoDB Compass

Please wait while the Setup Wizard installs MongoDB Compass.

Status: Copying new files

MongoDB Compass

Click on Finish to complete the installation:



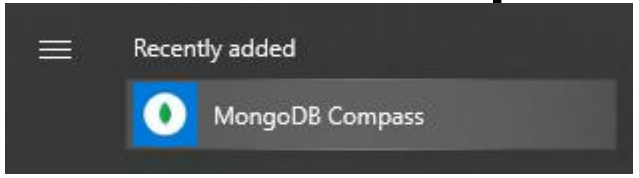
MongoDB Compass Setup

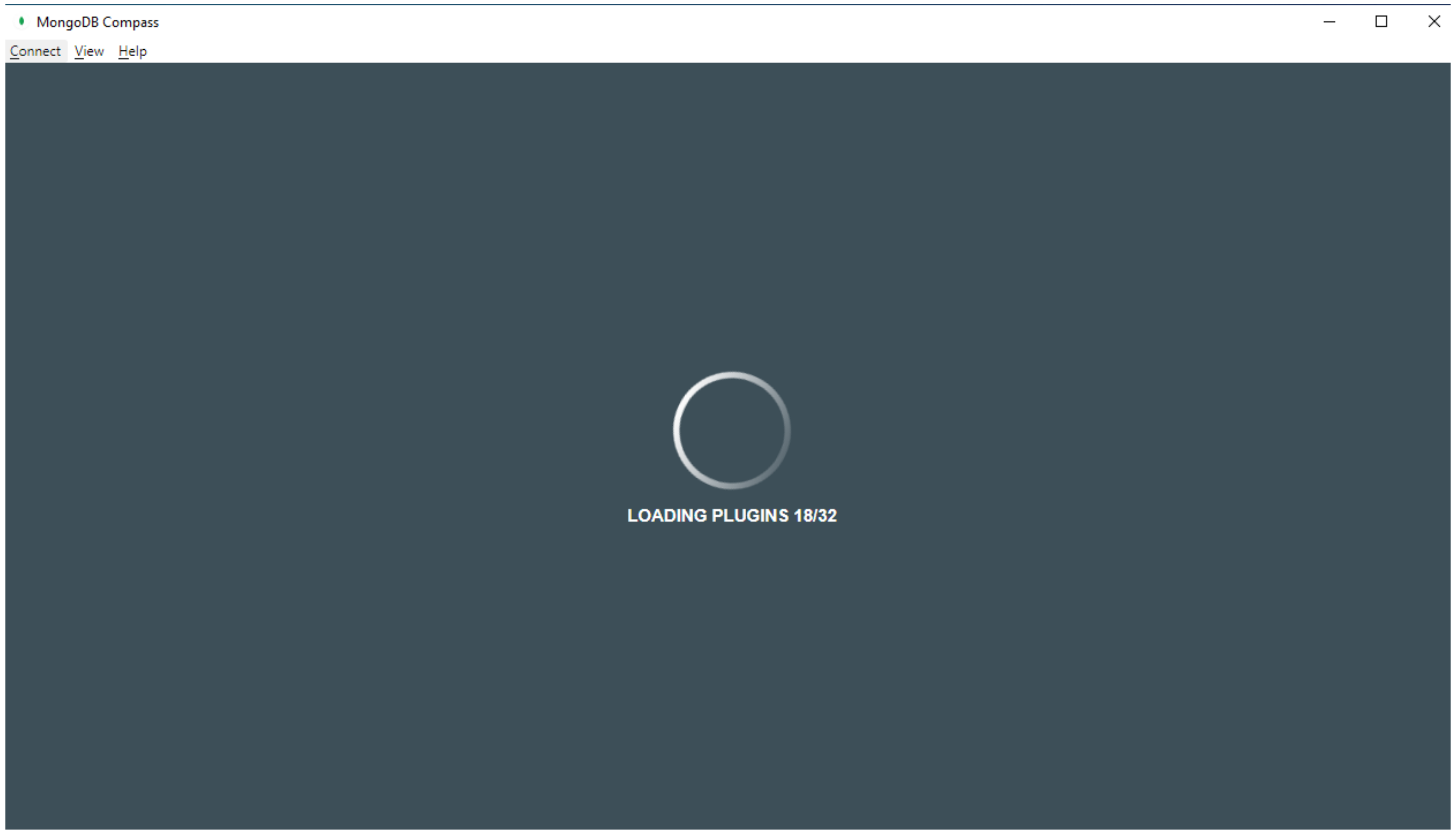
Completed the MongoDB Compass Setup Wizard

Click the Finish button to exit the Setup Wizard.

MongoDB Compass

Go to Start Menu and click on



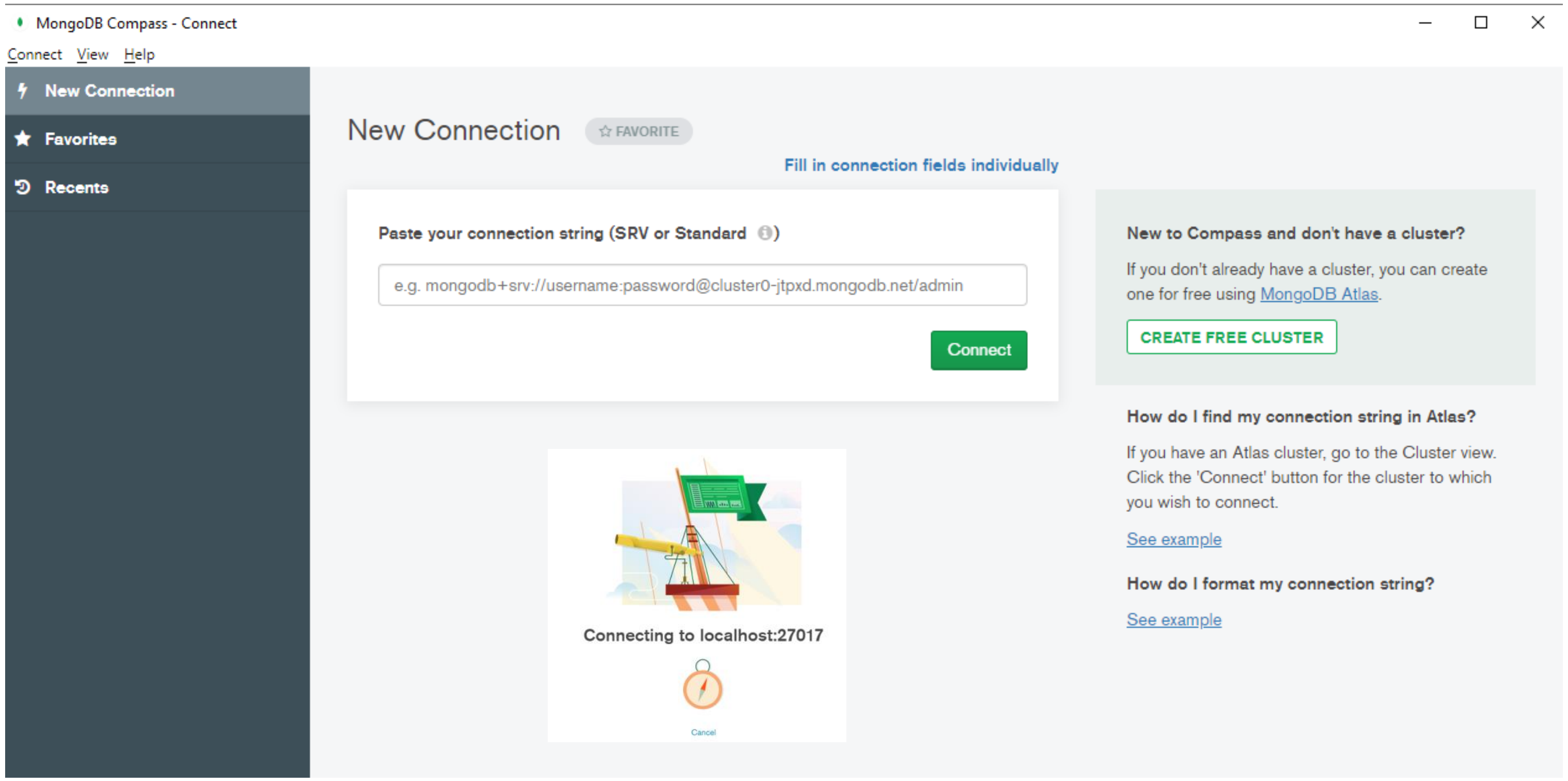


LOADING PLUGINS 18/32

MongoDB Compass

Connect to the cluster (server). We will be connecting to the local host.

Provide the connection string and click on connect



MongoDB Compass - Connect

Connect View Help

New Connection
Favorites
Recents

New Connection ☆ FAVORITE

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

e.g. mongodb+srv://username:password@cluster0-jtpxd.mongodb.net/admin

Connect

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas.

CREATE FREE CLUSTER

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

See example

How do I format my connection string?

See example

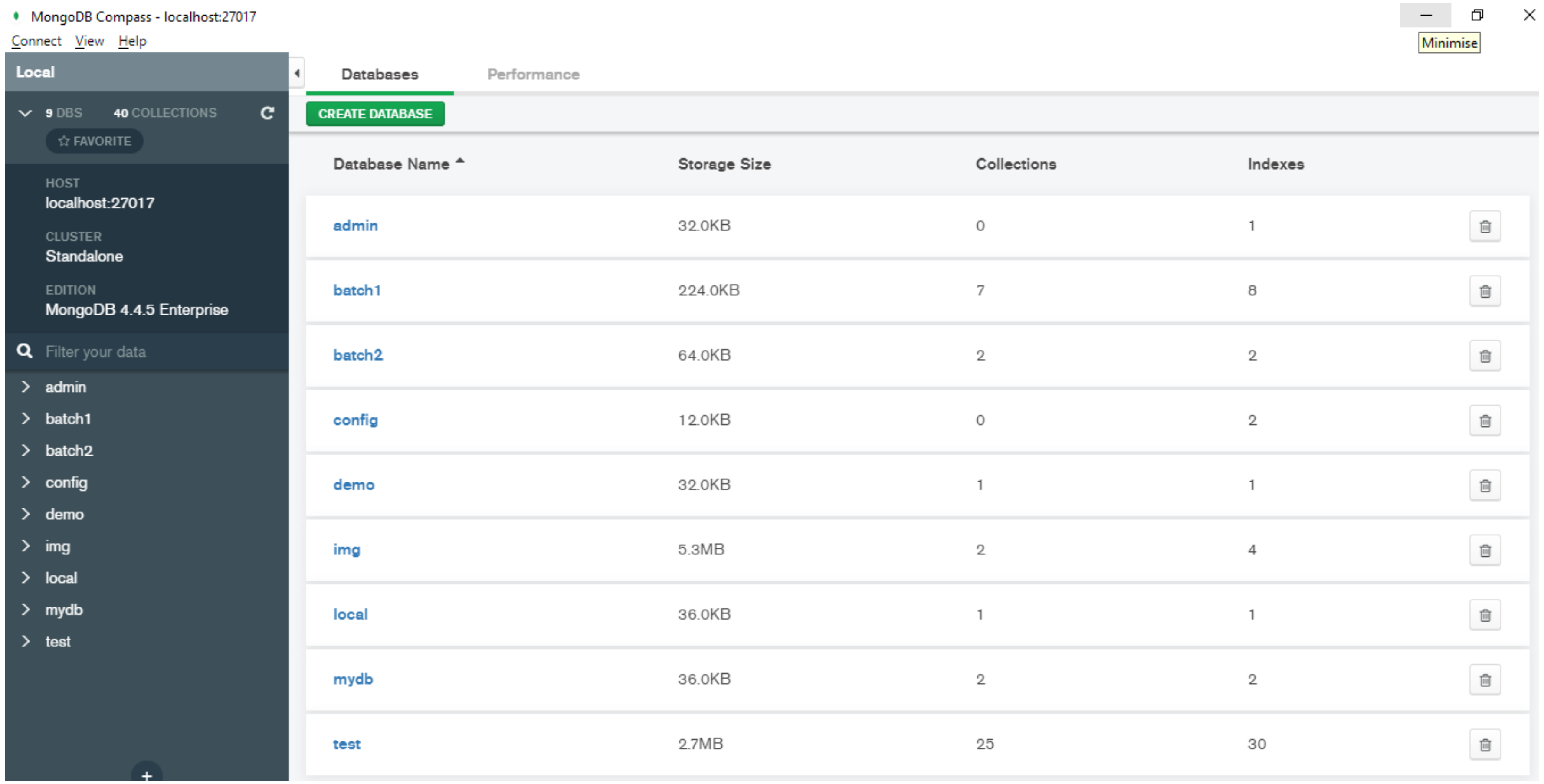
Connecting to localhost:27017

Cancel

NOTE: Mongo server must be running on your system to connect to the host

MongoDB Compass

Once connected you will be able to see all the databases in the mongo



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.5 Enterprise

- admin
- batch1
- batch2
- config
- demo
- img
- local
- mydb
- test

Database Name	Storage Size	Collections	Indexes
admin	32.0KB	0	1
batch1	224.0KB	7	8
batch2	64.0KB	2	2
config	12.0KB	0	2
demo	32.0KB	1	1
img	5.3MB	2	4
local	36.0KB	1	1
mydb	36.0KB	2	2
test	2.7MB	25	30

MongoDB Compass

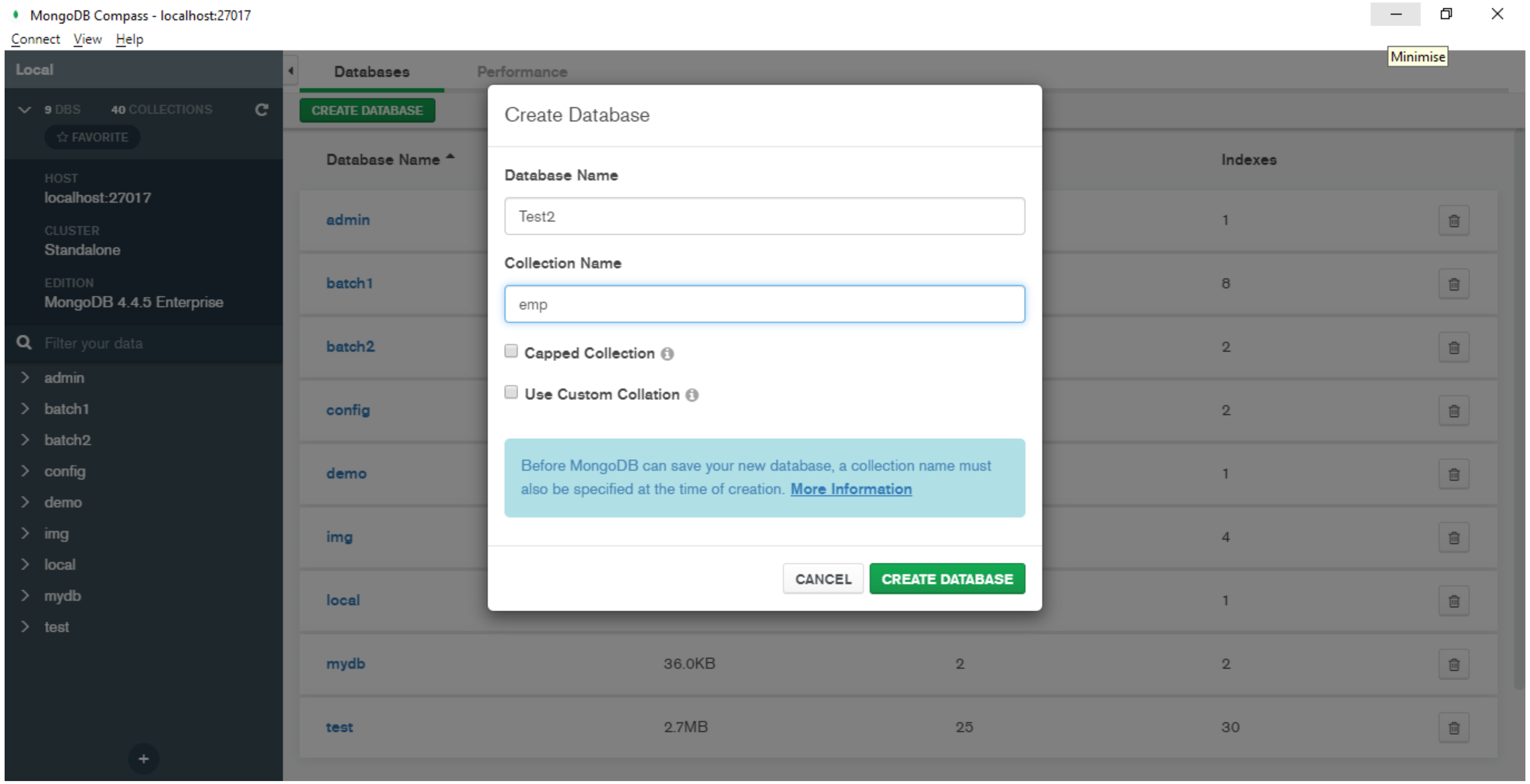
MongoDb Compass allows us to perform below operations:

- Create Database
- Create Collection
- Perform CRUD operations
- Add data to collection
  - ➢ Import File
  - ➢ Insert Document
- Use Options
  - ➢ Filter
  - ➢ Project
  - ➢ Sort
- Aggregations
- Indexes

dation

MongoDB Compass

Create Database



MongoDB Compass - localhost:27017

Connect View Help

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.4.5 Enterprise

admin
batch1
batch2
config
demo
img
local
mydb
test

Database Name			Indexes
admin			1
batch1			8
batch2			2
config			2
demo			1
img			4
local			1
mydb	36.0KB	2	2
test	2.7MB	25	30

Create Database

Database Name: Test2

Collection Name: emp

☐ Capped Collection

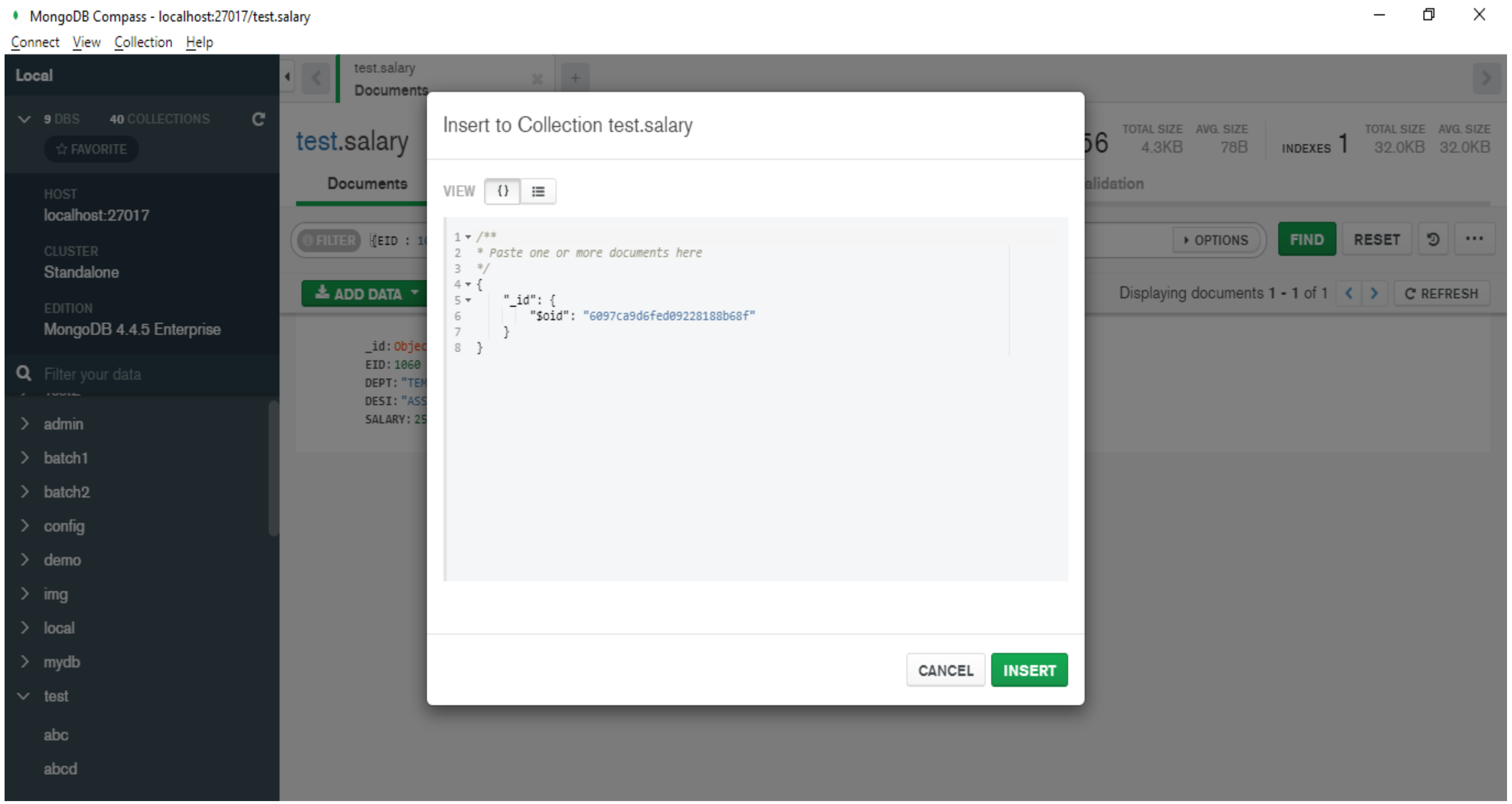
☐ Use Custom Collation

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information

CANCEL CREATE DATABASE

MongoDB Compass

Add data to collection

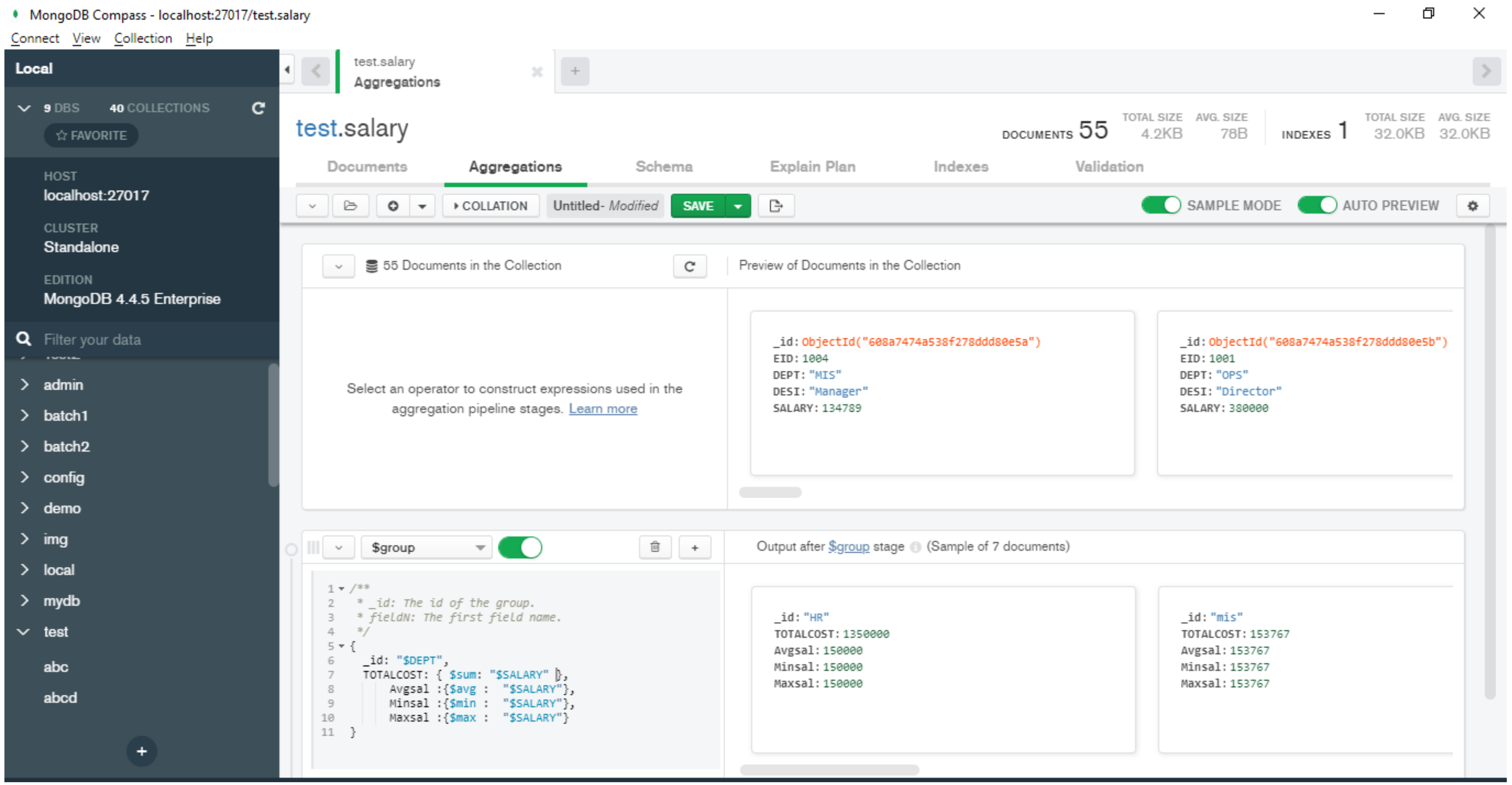

MongoDB Compass - localhost:27017/test.salary

Insert to Collection test.salary

```
/**
 * Paste one or more documents here
 */
{
    "_id": {
        "$oid": "6097ca9d6fed09228188b68f"
    }
}
```

MongoDB Compass

Aggregations



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.5 Enterprise

test.salary

DOCUMENTS 55 TOTAL SIZE 4.2KB AVG. SIZE 78B

INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

55 Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. Learn more

Preview of Documents in the Collection

```
_id: ObjectId("608a7474a538f278ddd80e5a")
EID: 1004
DEPT: "MIS"
DESI: "Manager"
SALARY: 134789
```

```
_id: ObjectId("608a7474a538f278ddd80e5b")
EID: 1001
DEPT: "OPS"
DESI: "Director"
SALARY: 380000
```

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id: "$DEPT",
  TOTALCOST: { $sum: "$SALARY" },
      Avgsal :{$avg :  "$SALARY"},
      Minsal :{$min :  "$SALARY"},
      Maxsal :{$max :  "$SALARY"}
}
```

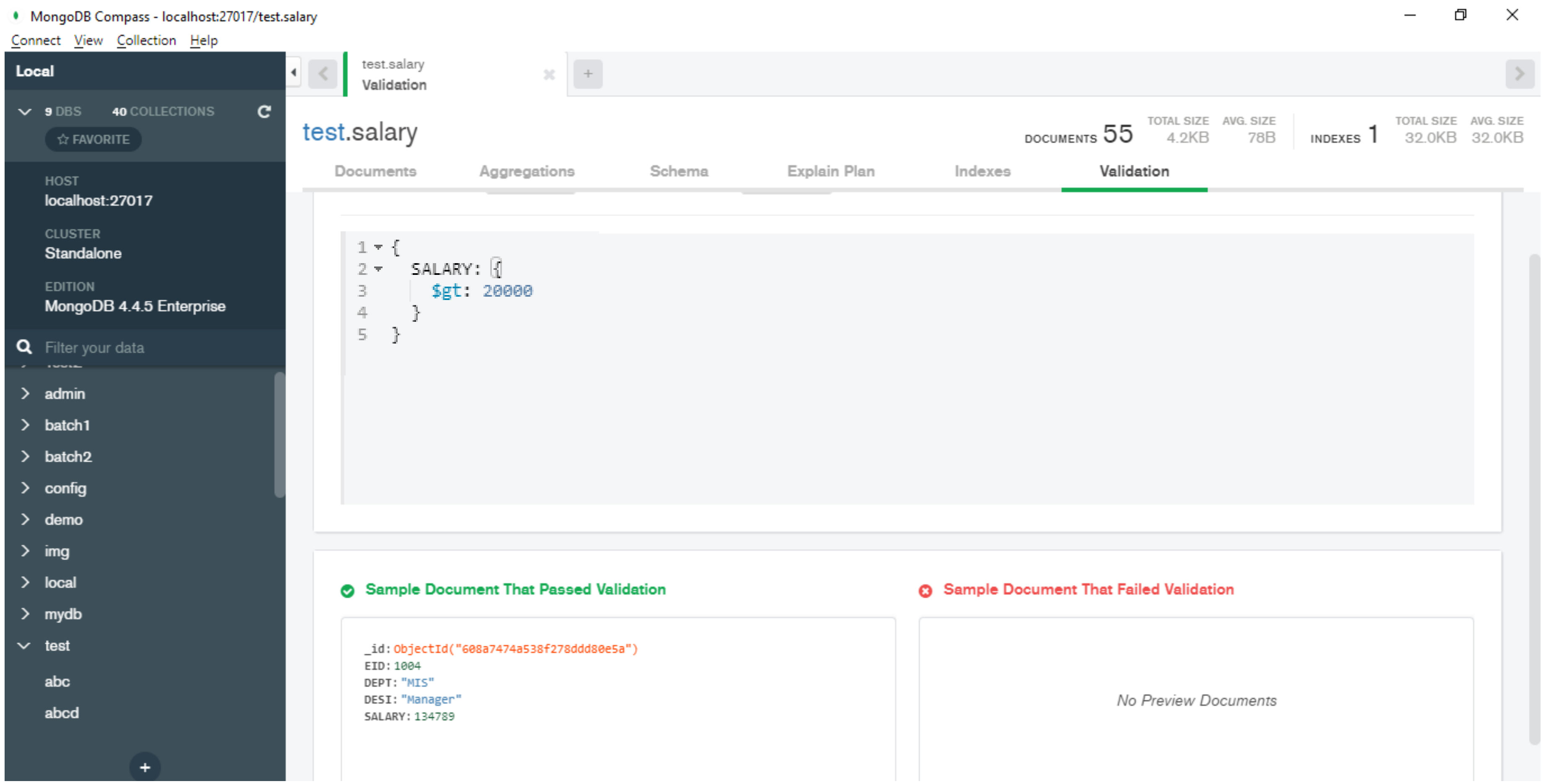
Output after $group stage (Sample of 7 documents)

```
_id: "HR"
TOTALCOST: 1350000
Avgsal: 150000
Minsal: 150000
Maxsal: 150000
```

```
_id: "mis"
TOTALCOST: 153767
Avgsal: 153767
Minsal: 153767
Maxsal: 153767
```

MongoDB Compass

Validations



MongoDB Compass - localhost:27017/test.salary

Connect View Collection Help

Local

9 DBS 40 COLLECTIONS

FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.5 Enterprise

Filter your data

- admin
- batch1
- batch2
- config
- demo
- img
- local
- mydb
- test
  - abc
  - abcd

test.salary
Validation

test.salary

DOCUMENTS 55 TOTAL SIZE 4.2KB AVG. SIZE 78B INDEXES 1 TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

```
{
  SALARY: {
    $gt: 20000
  }
}
```

Sample Document That Passed Validation

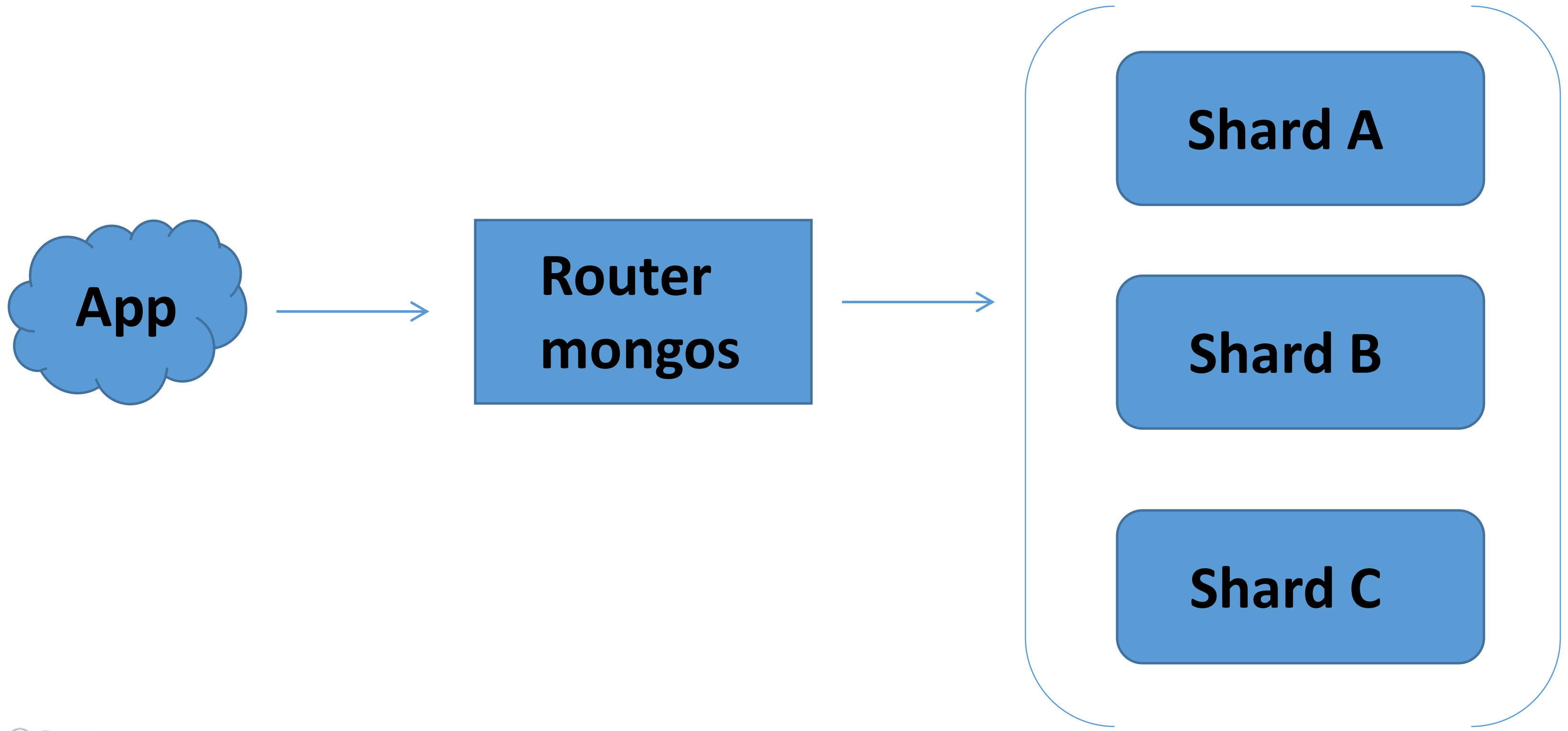
```
_id: ObjectId("608a7474a538f278ddd80e5a")
EID: 1004
DEPT: "MIS"
DESI: "Manager"
SALARY: 134789
```

Sample Document That Failed Validation

No Preview Documents

Sharding MongoDB

Sharding is the process of partitioning your data across multiple servers. It is a type of database partitioning that separates very large database into faster, smaller and more easily manageable parts called shards.



Sharding MongoDB

MongoDB uses the shard key to distribute the collection’s documents across shards. The shard key consists of a field or multiple fields in the documents.

Why Sharding ?

Scalable - data is growing continuously

High Availability

Ability to control data distribution

Application Transparent

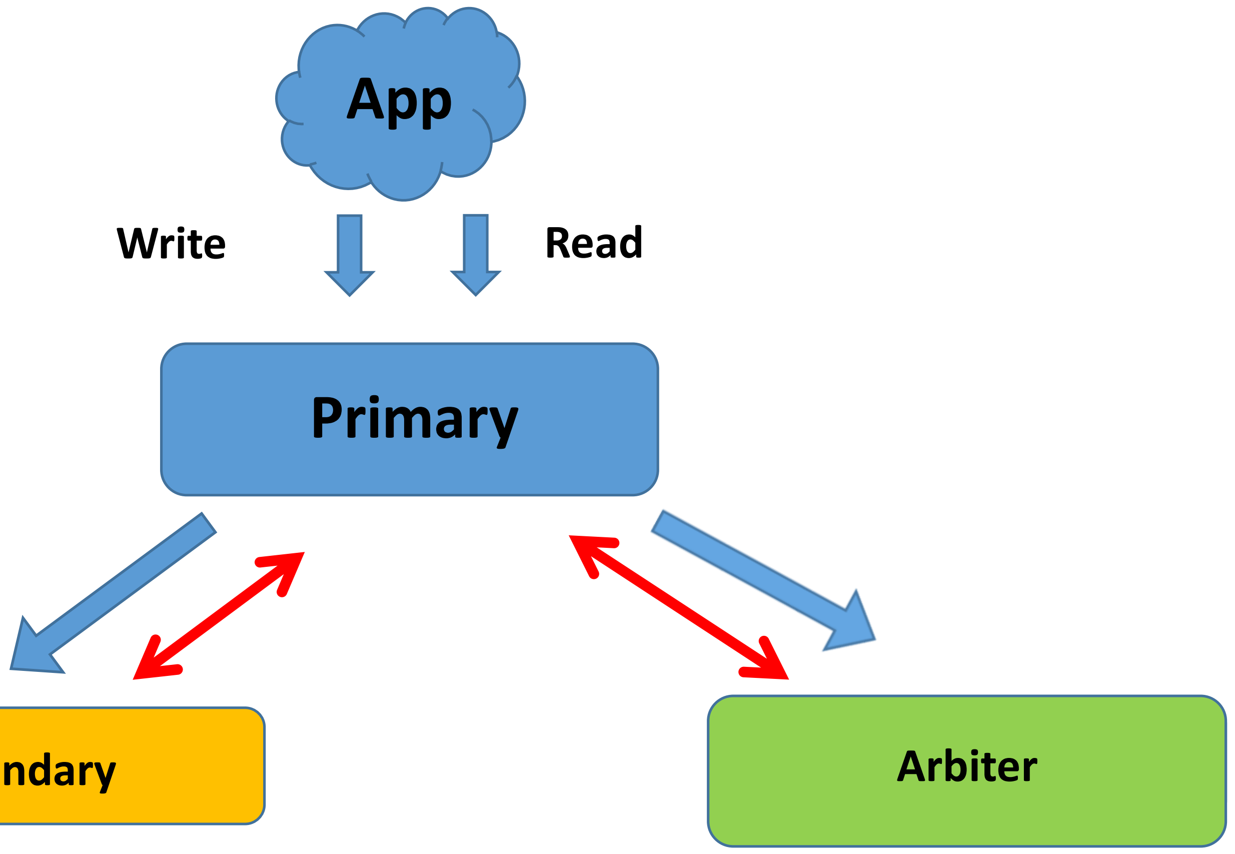
Cost effective

No database downtime

Replication in MongoDB

A replica set is a group of mongod instances (server) that maintain the same data set. A replica set contains several data bearing nodes and optionally one arbiter node. Of the data bearing nodes, one and only one member is deemed the primary node, while the other nodes are deemed secondary nodes.

The primary node receives all write operations



Replication in MongoDB

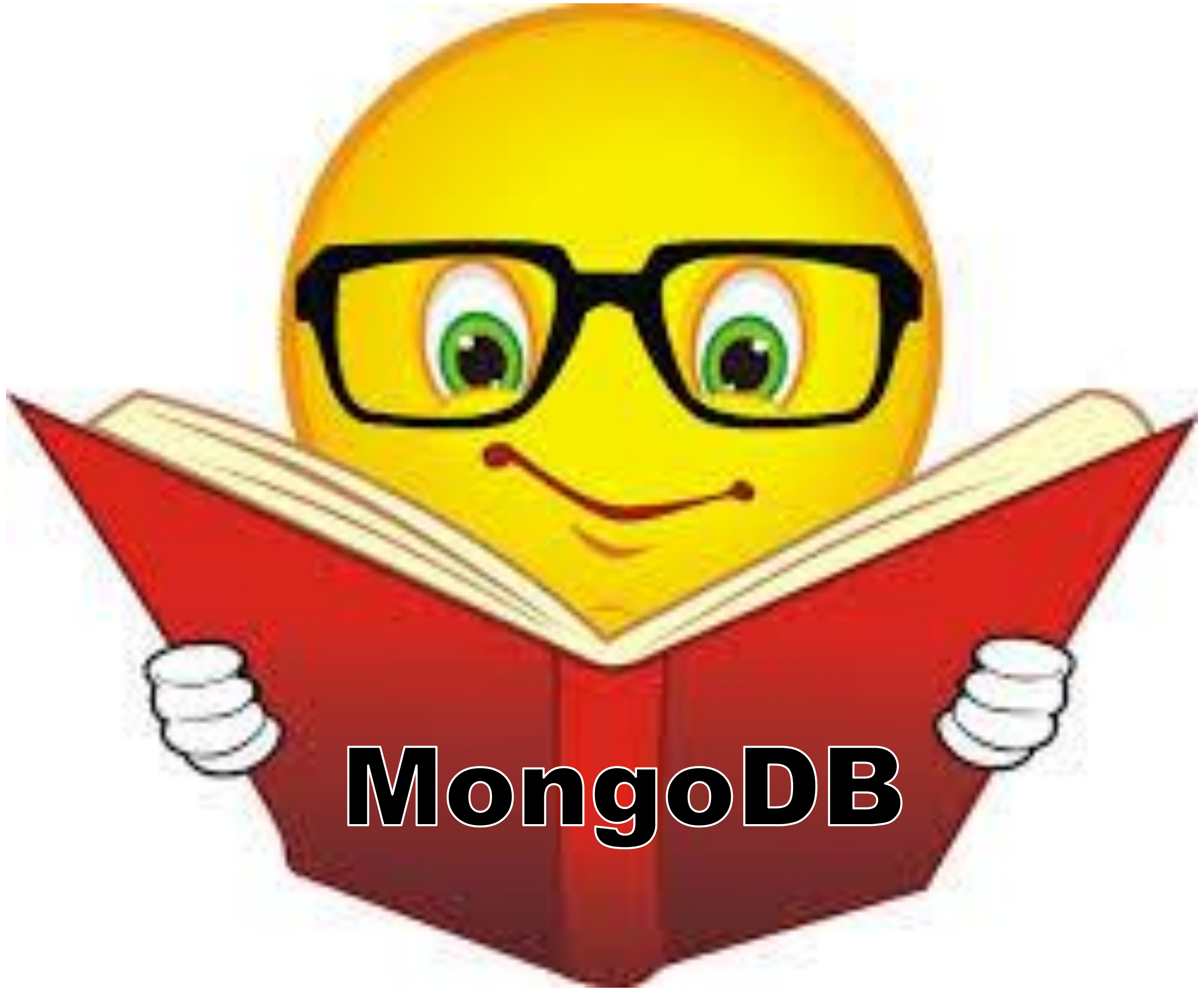
Major features of replica:

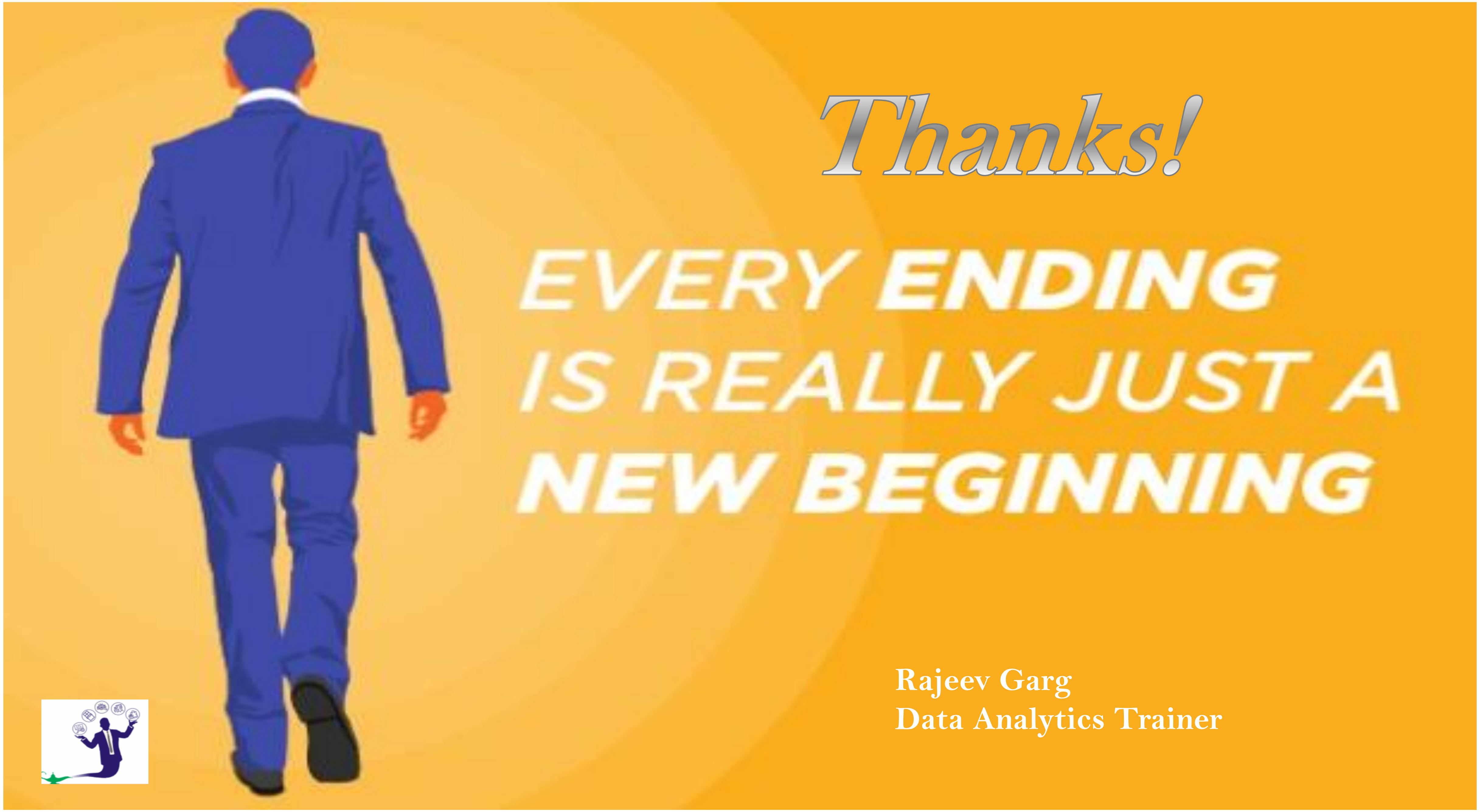
Asynchronous Replication -Secondary replicate the primary’s and apply the operations to their data sets asynchronously.

Automatic Failover (electionTimeoutMillis period (10 seconds by default))

Read Preference - can specify a read preference to send read operations to secondaries.

Mirrored Reads – operations can be in the cache of secondary.


MongoDB



Thanks!

Rajeev Garg
Data Analytics Trainer