

# Episode-13 | Creating a database & mongodb

There are two main ways to use MongoDB:

## 1. Install MongoDB Locally

You can download the MongoDB package, install it on your system, and use it directly from your local environment.

## 2. Use MongoDB Managed Services (MongoDB Atlas)

Instead of installing MongoDB locally, you can use MongoDB Atlas, a cloud-based solution where MongoDB manages the database for you.

## Types of MongoDB Editions

### 1. Community Edition:

This is a free version, ideal for developers and personal projects.

### 2. Enterprise Edition:

This version is designed for companies and offers advanced features and support.

---

## Why Use MongoDB Atlas?

Using MongoDB Atlas has several advantages:

- **No Server Management:** MongoDB handles all the database infrastructure and scaling for you.
  - **Easy Transition to Production:** You can easily move from a local development environment to production with minimal effort.
- 

## How to Set Up MongoDB Atlas

### 1. Visit the MongoDB Website:

Go to

[mongodb.com](https://mongodb.com) and click on the "Try Atlas Free" button.

Sign up if you don't have an account.

### 2. Choose the Free Plan (M0):

After signing up, you'll see different plans. Select the **M0 plan**, which is free forever.

### 3. Configure Your Cluster:

- **Cluster Name:** You can give a name to your cluster.
- **Automatic Security Setup:** Enable this option to automatically configure security.
- **Sample Dataset:** Preload a sample dataset to experiment with.

### 4. Cloud Provider and Region:

- You can choose between AWS, Google Cloud, or Azure as your provider.
- For users in India, select the **Mumbai region**. *Note:* In organizations like NamasteDev, the region is often set to Mumbai because the majority of students are from India. However, for someone in Canada, this may result in slightly slower APIs.

## 5. Create a Deployment:

- MongoDB Atlas will provide you with a **username** and **password** for accessing the cluster. Save these credentials.
- You can create multiple database users if needed.

## Connecting to Your Cluster

### 1. Connection Method:

After creating your cluster, choose a connection method. MongoDB Atlas will provide several options.

### 2. Get the Connection String:

- Click the **"Connect"** button to get your connection string.
- Replace the placeholder `[password]` in the connection string with the password you copied earlier.

### 3. Create a `database.js` File:

In your project, create a

`database.js` file and paste your connection string.

The screenshot shows the MongoDB Atlas cluster creation interface. At the top, there are three deployment options: M10 (\$0.08/hour), Serverless, and M0 (Free). The M0 option is selected and highlighted with a green border. Below the options, a green banner states: "Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime." The "Name" field is set to "Cluster0". There are two checkboxes: "Automate security setup" (checked) and "Preload sample dataset" (checked). The "Provider" section shows "aws" selected, with "Google Cloud" and "Azure" as alternatives. The "Region" is set to "Mumbai (ap-south-1)". At the bottom, there are two buttons: "I'll do this later" and "Create Deployment".

Deployment Option	Price	Use Case
M10	\$0.08/hour	For production applications with sophisticated workload requirements.
Serverless	-	For application development and testing, or workloads with variable traffic.
M0	Free	For learning and exploring MongoDB in a cloud environment.

**Cluster Configuration:**

- Name: Cluster0
- Automate security setup: ☒
- Preload sample dataset: ☒
- Provider: aws
- Region: Mumbai (ap-south-1)

**Buttons:** I'll do this later, Create Deployment

## Connect to Cluster4224



You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

### 1. Add a connection IP address

✓ Your current IP address (106.219.70.166) has been added to enable local connectivity. Add another later in [Network Access](#).

### 2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

**i** You'll need your database user's credentials in the next step. Copy the database user password.

Username

Password

 HIDE

Copy

Create Database User

Close

Choose a connection method

## Connect to Cluster4224



1

Set up connection security

2

Choose a connection method

3

Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

### 1. Add a connection IP address

✓ Your current IP address (106.219.70.166) has been added to enable local connectivity. Add another later in [Network Access](#).

### 2. Create a database user

✓ A database user has been added to this project. Create another user later in [Database Access](#).

You'll need your database user's credentials in the next step.

Close

Choose a connection method

## Connect to Cluster4224



### Connect to your application



#### Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)



### Access your data through tools



#### Compass

Explore, modify, and visualize your data with MongoDB's GUI



#### Shell

Quickly add & update data using MongoDB's Javascript command-line interface



#### MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment



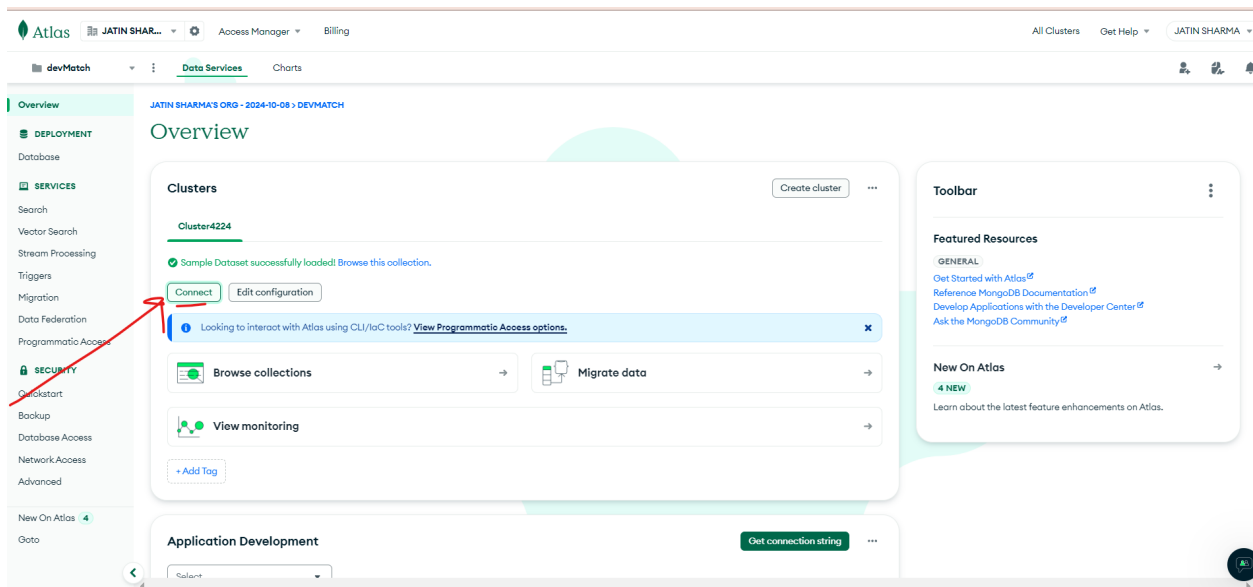
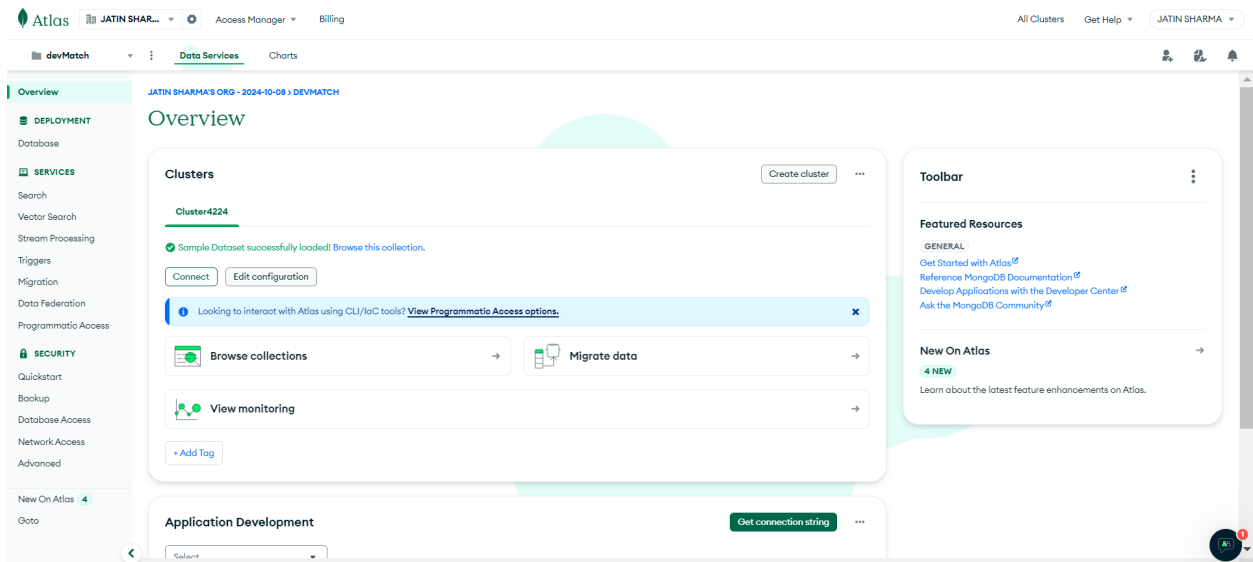
#### Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization



Go Back

Close



## Connect to Cluster4224



### Connect to your application



#### Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)



### Access your data through tools



#### Compass

Explore, modify, and visualize your data with MongoDB's GUI



#### Shell

Quickly add & update data using MongoDB's Javascript command-line interface



#### MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment



#### Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization



Go Back

Close



## Connect to Cluster4224



### Connecting with MongoDB Compass

I don't have MongoDB Compass installed

I have MongoDB Compass installed

#### 1. Select your operating system and download MongoDB Compass

macOS arm64 (M1) (11.0+)

Download Compass (1.44.4)

or

Copy download URL

Compass is an interactive tool for querying, optimizing, and analyzing your MongoDB data.

#### 2. Copy the connection string, then open MongoDB Compass

Use this connection string in your application

mongodb+srv://jatin4224:<db\_password>@cluster4224.cctoa.mongodb.net/



Replace **<db\_password>** with the password for the **jatin4224** user. Ensure any options are [URL encoded](#).

#### RESOURCES

[Connect with Compass](#)

[Import and Export Data](#)

[Access your Database Users](#)

[Troubleshoot Connections](#)

Go Back

Done

## Using MongoDB Compass

To visually interact with your MongoDB databases, you can use **MongoDB Compass**, which is a graphical interface for your databases.

1. **Download MongoDB Compass:** [Download MongoDB Compass](#) from the official website.

## 2. Add a New Connection:

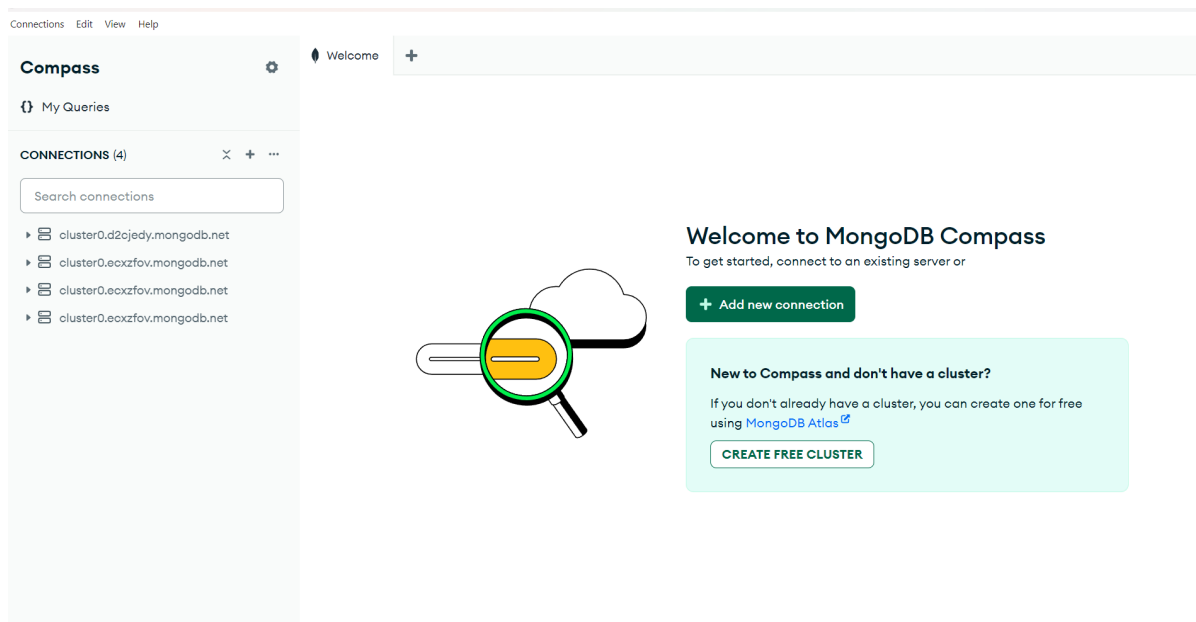
- Open MongoDB Compass and add a new connection using the connection string.
- You can now create databases and collections directly from Compass or via code.

## 3. Create a Database:

- Click on **"Create Database"** in Compass.
- Provide a **database name** and a **collection name** (a collection is similar to a table in SQL databases).
- Example: If you want to store user data, name your collection `users` and click "Create Database."
- **Note:** Do not check the **time-series** option unless you need it.

## 4. Insert Data:

- Click **"Add Data"** and select **"Insert Document"** to add data to your collection.
- MongoDB will automatically generate an `_id` for each document.



CONNECTIONS (4)

Search connections

cluster0.d2oje

cluster0.ecxz

cluster0.ecxz

cluster0.ecxz

## New Connection

Manage your connection settings

URI ⓘ

mongodb://localhost:27017/

Edit Connection String ⓘ

write same connection string

Name

Color

No Color

☐ Favorite this connection

Favoriting a connection will pin it to the top of your list of connections

Cancel

Save

Connect

### 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?

mongodb.compass - USER@MONGO/CLUSTER4224

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connections

cluster4224

admin

config

local

sample\_mflix

cluster4224

Open MongoDB shell

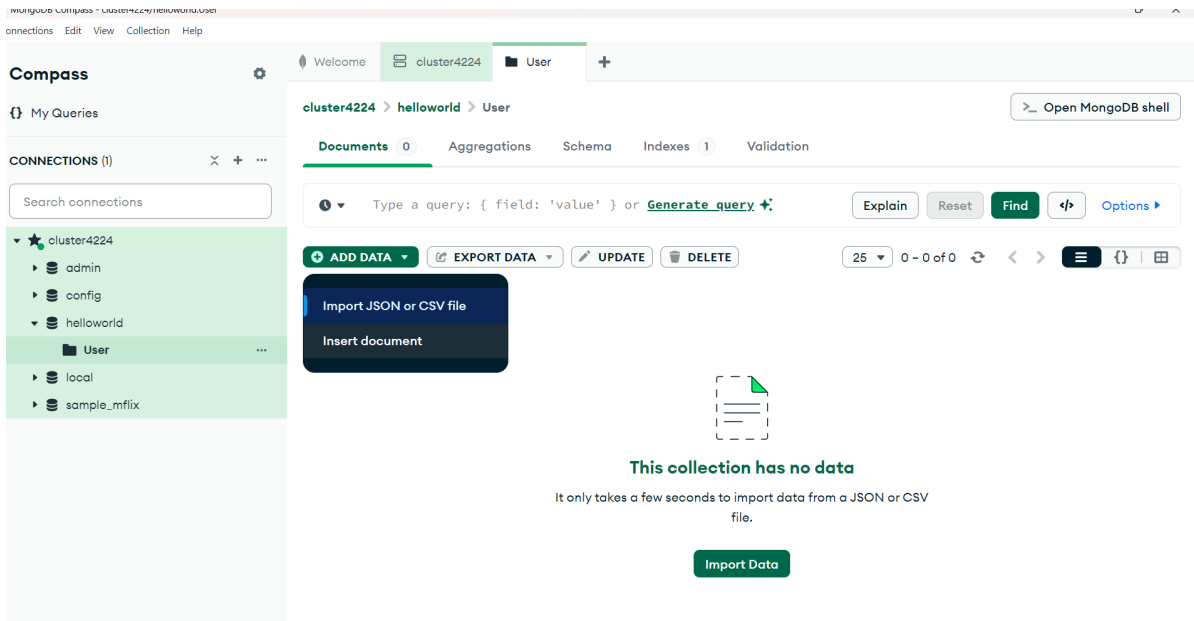
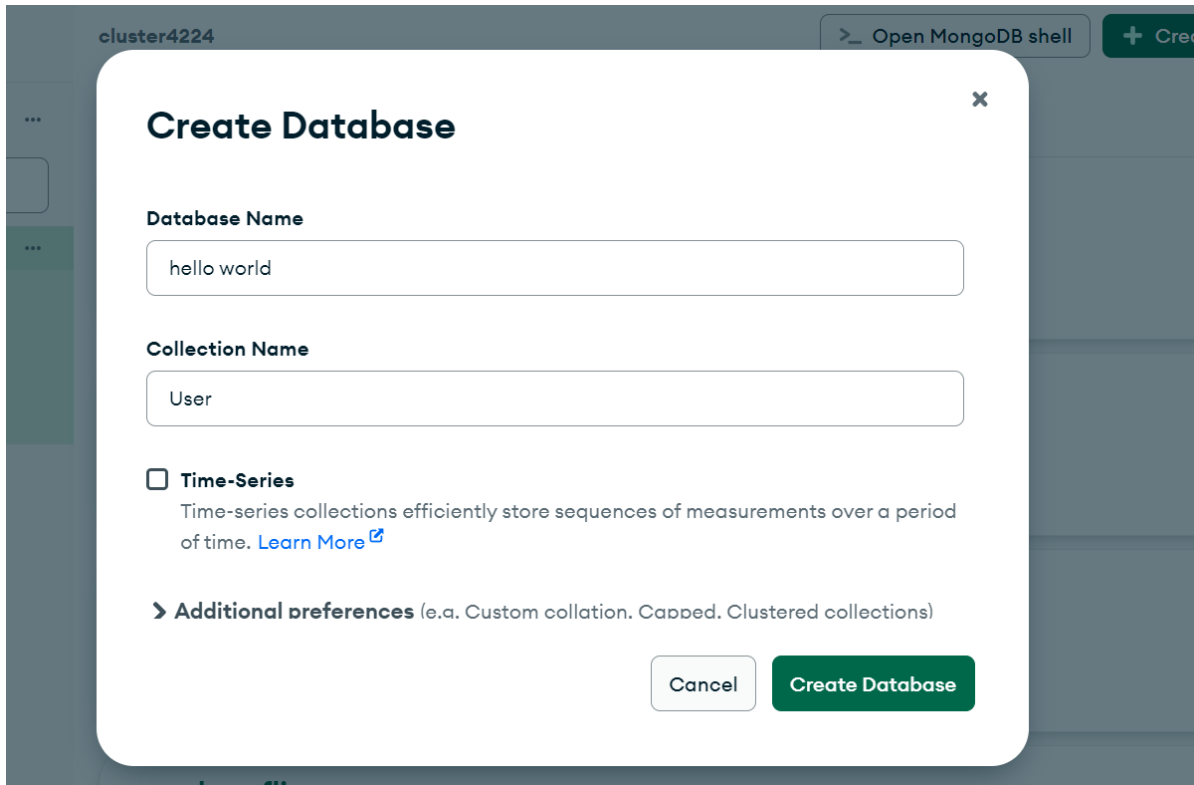
Create database

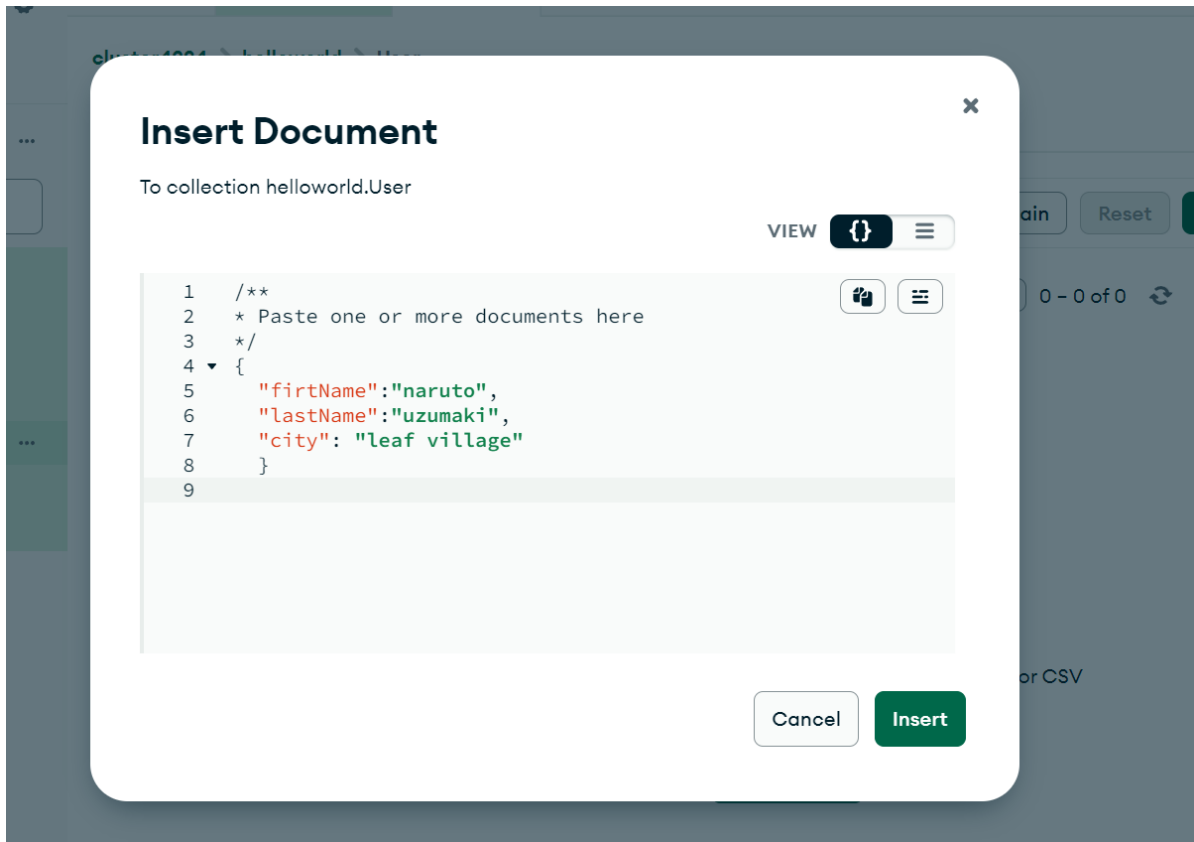
Refresh

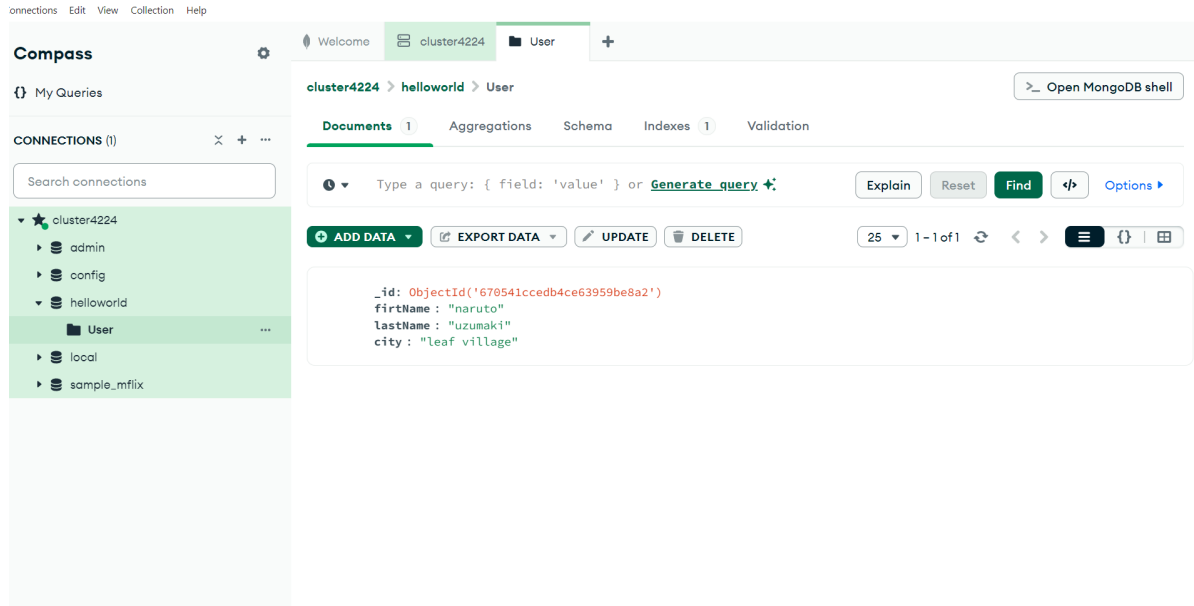
Sort by Database Name

View

admin	Storage size: 0 B	Collections: 0	Indexes: 0
config	Storage size: -	Collections: -	Indexes: -
local	Storage size: -	Collections: -	Indexes: -
sample_mflix	Storage size: -	Collections: -	Indexes: -







## Using Code to Insert Data

You can also interact with your MongoDB database using Node.js.

### 1. Install the MongoDB NPM Package:

Run the following command to install the MongoDB package:

```
npm i mongodb
```

### 2. What is NPM?

NPM (Node Package Manager) is a repository of packages (node modules) that you can use in your Node.js applications. It's essentially a package manager for Node.js.

### 3. Code Example in `database.js`:

```
const { MongoClient } = require("mongodb");

const url =
  "db connection string ";

const client = new MongoClient(url);
```

```

const dbName = "HelloWorld";

async function main() {
  await client.connect();
  console.log("Connected successfully to server");
  const db = client.db(dbName);
  const collection = db.collection("User");

  const data = {
    firstname: "Ranveer",
    lastname: "Singh",
    city: "Mumbai",
    phoneNumber: "987543210",
  };

  const insertResult = await collection.insertOne(data);
  console.log("Inserted documents =>", insertResult);

  // Read
  const findResult = await collection.find({}).toArray();
  console.log("Found documents =>", findResult);

  const countResult = await collection.countDocuments({});
  console.log("Count of documents in the User collection =
>", countResult);

  // Find all documents with a filter of firstname: Deepika
  const result = await collection.find({ firstname: "Deepika" }).count();
  console.log("result => ", result);

  return "done.";
}

```

```
main()  
  .then(console.log)  
  .catch(console.error)  
  .finally(() => client.close());
```

Now, you're ready to connect your MongoDB cluster and perform operations from your Node.js code.

---

To make your life easier when connecting to MongoDB, you can use the **Mongoose** library, which is an Object Data Modeling (ODM) library for MongoDB and Node.js. It simplifies the interaction with MongoDB by providing schema-based solutions for modeling your data and handling queries.

## Steps to Connect to MongoDB Using Mongoose:

### 1. Install Mongoose:

First, you need to install the Mongoose library. Run the following command in your terminal:

```
npm install mongoose
```

### 2. Connect to MongoDB with Mongoose:

In your

`database.js` file, you can use the following code to connect to your MongoDB cluster using Mongoose.

## Code Example for `database.js` :

```
const mongoose = require("mongoose");  
  
// Connection URL (replace <username>, <password>, and <your-  
cluster-url> with your actual values)  
const mongoURI = "mongodb+srv://<username>:<password>@<your-c
```



```

cluster-url>/<dbname>?retryWrites=true&w=majority";

// Function to connect to MongoDB using Mongoose
const connectDB = async () => {
  try {
    // Mongoose connection
    await mongoose.connect(mongoURI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log("MongoDB connected successfully!");
  } catch (error) {
    console.error("Error connecting to MongoDB:", error.message);
    process.exit(1); // Exit process with failure
  }
};

module.exports = connectDB;

```

## Advantages of Using Mongoose:

- **Schema Definitions:** Mongoose allows you to define schemas for your collections, which helps enforce structure in your documents.
- **Built-in Validation:** Mongoose provides validation at the schema level, ensuring your data is clean and consistent.
- **Middleware:** You can use pre and post hooks in Mongoose for tasks like data validation, logging, or transforming data before saving.

