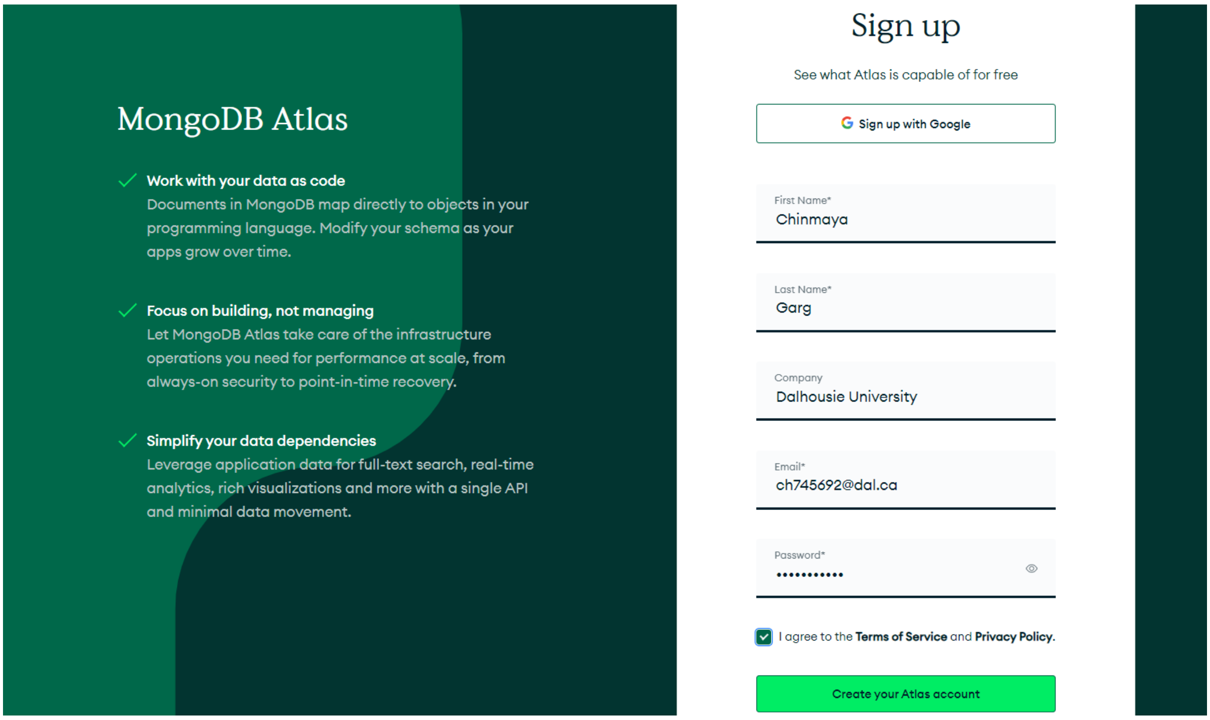
DALHOUSIE UNIVERSITY

CSCI 5709 - Advanced Web Services
Summer 2024

Tutorial 6

Submitted By:
Chinmaya Garg
B00925398

MongoDB [1] Account Creation Created an account in MongoDB (Figure 1).



MongoDB Atlas

✓ **Work with your data as code**
Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.

✓ **Focus on building, not managing**
Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.

✓ **Simplify your data dependencies**
Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal data movement.

Sign up

See what Atlas is capable of for free

Sign up with Google

First Name*: Chinmaya

Last Name*: Garg

Company: Dalhousie University

Email*: ch745692@dal.ca

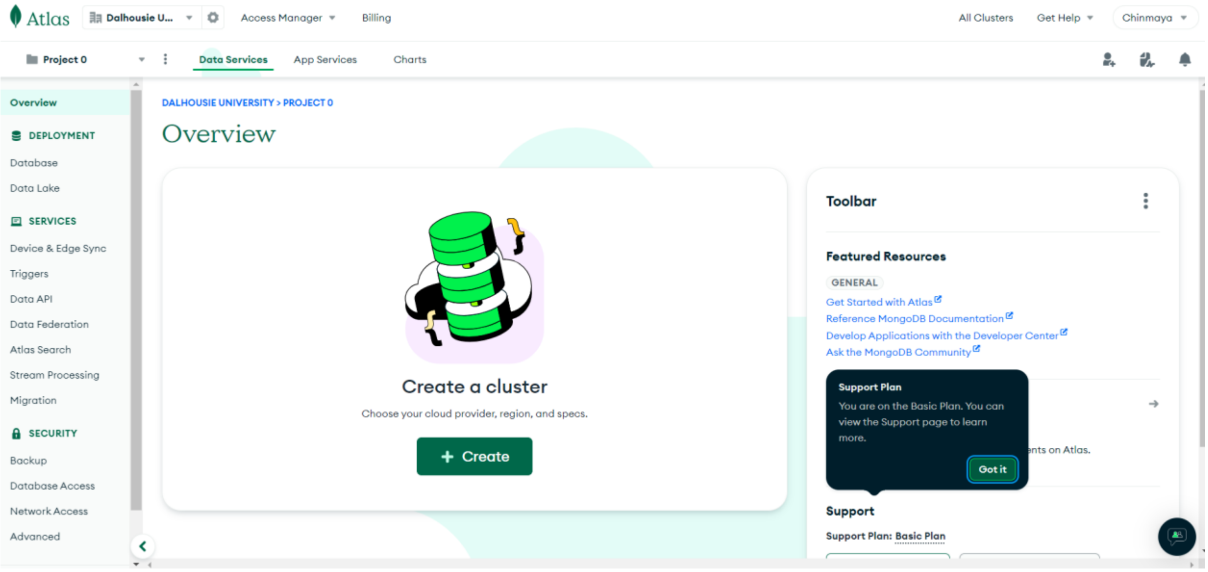
Password*: •••••••••••

☑ I agree to the **Terms of Service** and **Privacy Policy**.

Create your Atlas account

Figure 1: MongoDB Signup Page

Account Confirmation: Account successfully created (Figure 2).



DALHOUSIE UNIVERSITY > PROJECT 0

Overview

Create a cluster

Choose your cloud provider, region, and specs.

+ Create

Toolbar

Featured Resources

GENERAL

Get Started with Atlas
Reference MongoDB Documentation
Develop Applications with the Developer Center
Ask the MongoDB Community

Support Plan

You are on the Basic Plan. You can view the Support page to learn more.

Got it

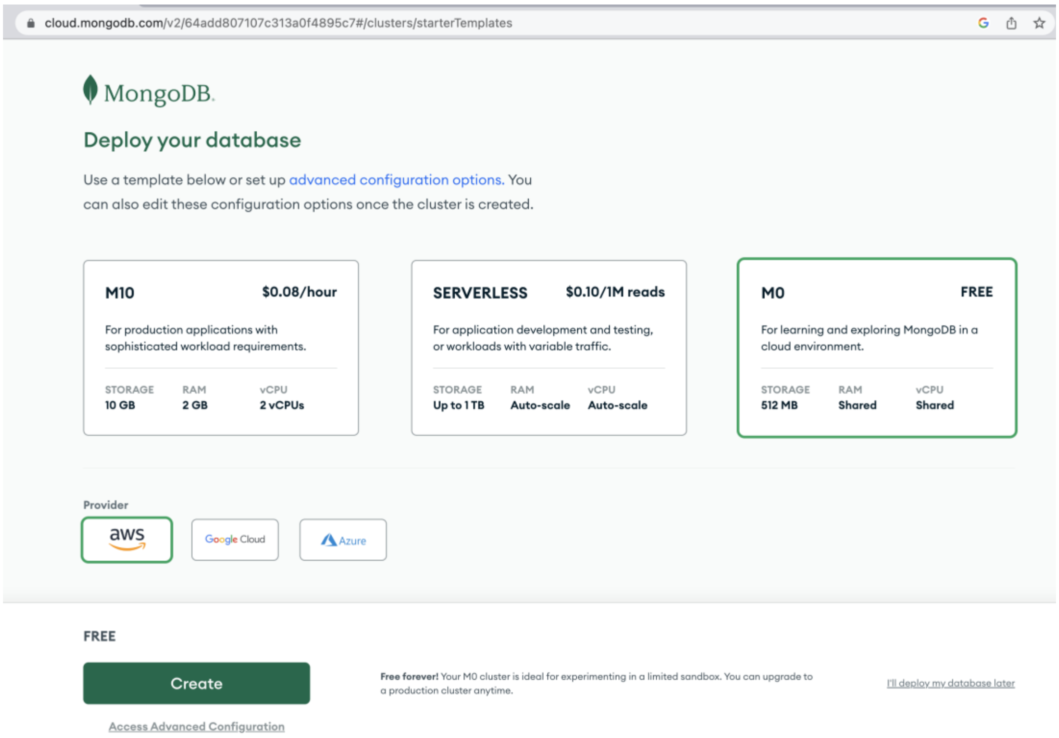
ents on Atlas.

Support

Support Plan: Basic Plan

Figure 2: MongoDB Home Page

Cluster Selection: Selected the cluster type (Figure 3).



cloud.mongodb.com/v2/64add807107c313a0f4895c7#/clusters/starterTemplates

MongoDB.

Deploy your database

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

M10	$0.08/hour		
For production applications with sophisticated workload requirements.			
STORAGE	RAM	vCPU	
10 GB	2 GB	2 vCPUs	

SERVERLESS	$0.10/1M reads		
For application development and testing, or workloads with variable traffic.			
STORAGE	RAM	vCPU	
Up to 1 TB	Auto-scale	Auto-scale	

M0	FREE		
For learning and exploring MongoDB in a cloud environment.			
STORAGE	RAM	vCPU	
512 MB	Shared	Shared	

Provider

aws | Google Cloud | Azure

FREE

Create

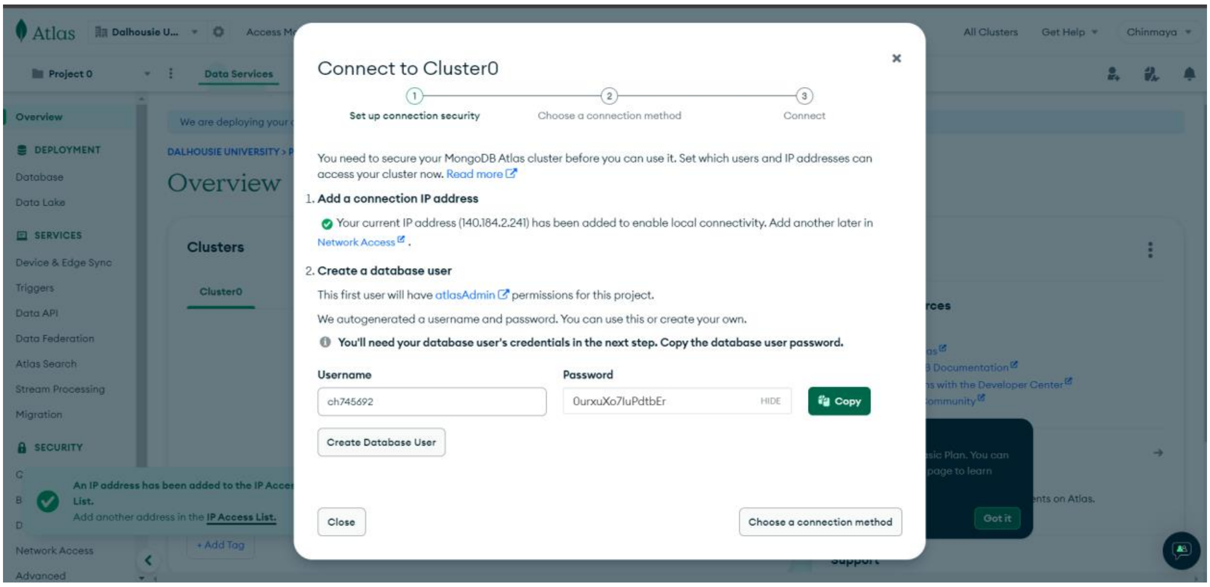
Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

I'll deploy my database later

Access Advanced Configuration

Figure 3: Cluster Creation

Cluster Credentials Setup: Created credentials to access the cluster (Figure 4).



Connect to Cluster0

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 (140.184.2.241) 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.

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

Username: ch745692

Password: 0urxuXo7luPdtbEr HIDE

Copy

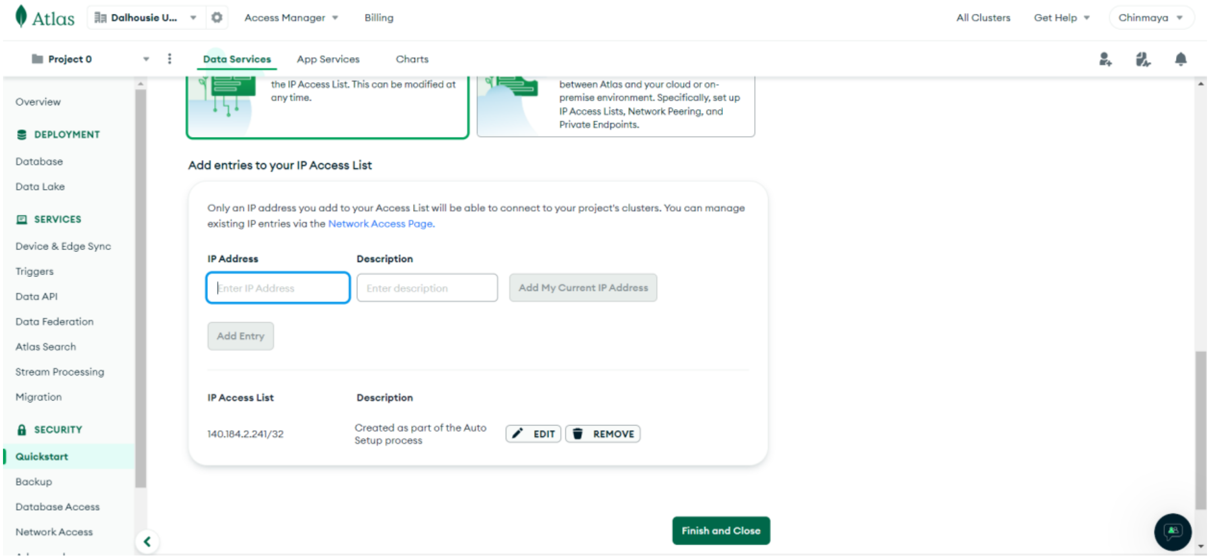
Create Database User

Close

Choose a connection method

Figure 4: Credentials Page

IP Whitelisting: Provided access to my IP (Figure 5).



the IP Access List. This can be modified at any time.

between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the Network Access Page.

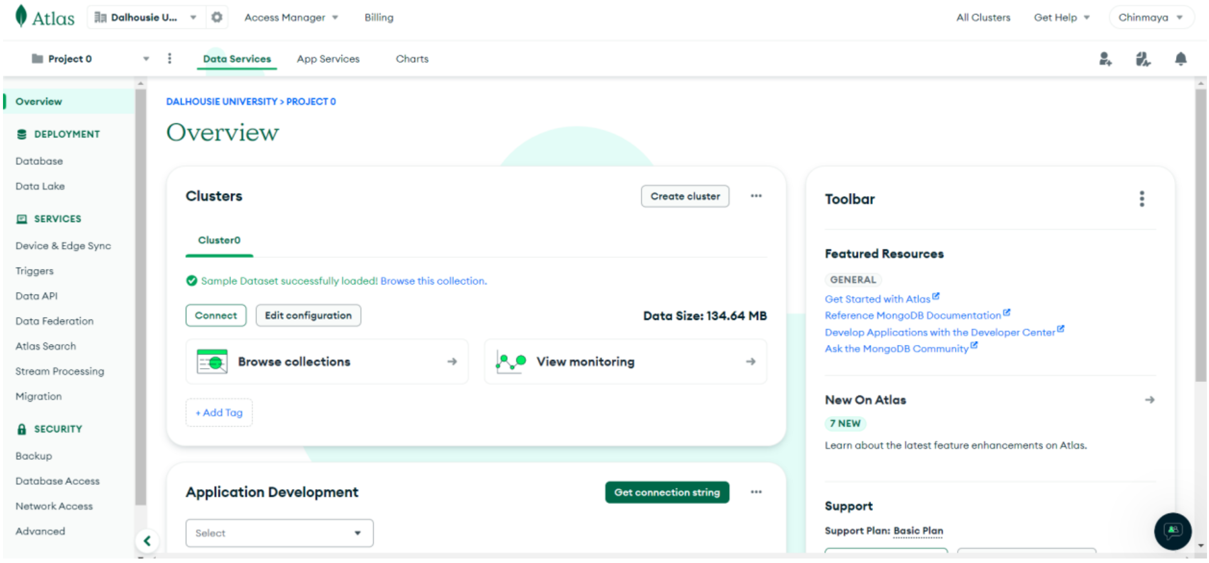
IP Address

Description

IP Access List	Description
140.184.2.241/32	Created as part of the Auto Setup process

Figure 5: Whitelisted IP for Cluster Access

Cluster Overview: Displaying the Cluster Overview page (Figure 6).



Sample Dataset successfully loaded! Browse this collection.

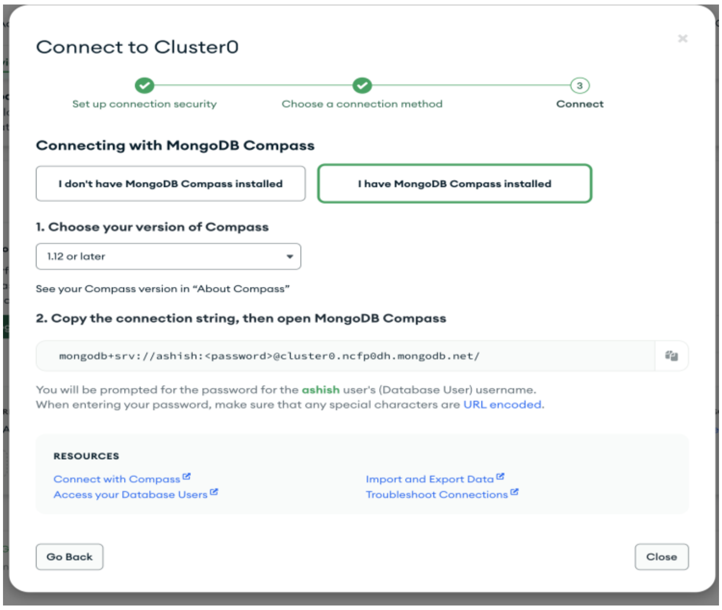
Data Size: 134.64 MB

Learn about the latest feature enhancements on Atlas.

Support Plan: Basic Plan

Figure 6: Cluster Overview

Cluster Connection Details: Connection details of the cluster (Figure 7).


Connect to Cluster0
Set up connection security
Choose a connection method
3
Connect
I don't have MongoDB Compass installed
I have MongoDB Compass installed
RESOURCES
Connect with Compass
Access your Database Users
Import and Export Data
Troubleshoot Connections
Go Back
Close

Connecting with MongoDB Compass

1. Choose your version of Compass

1.12 or later

See your Compass version in "About Compass"

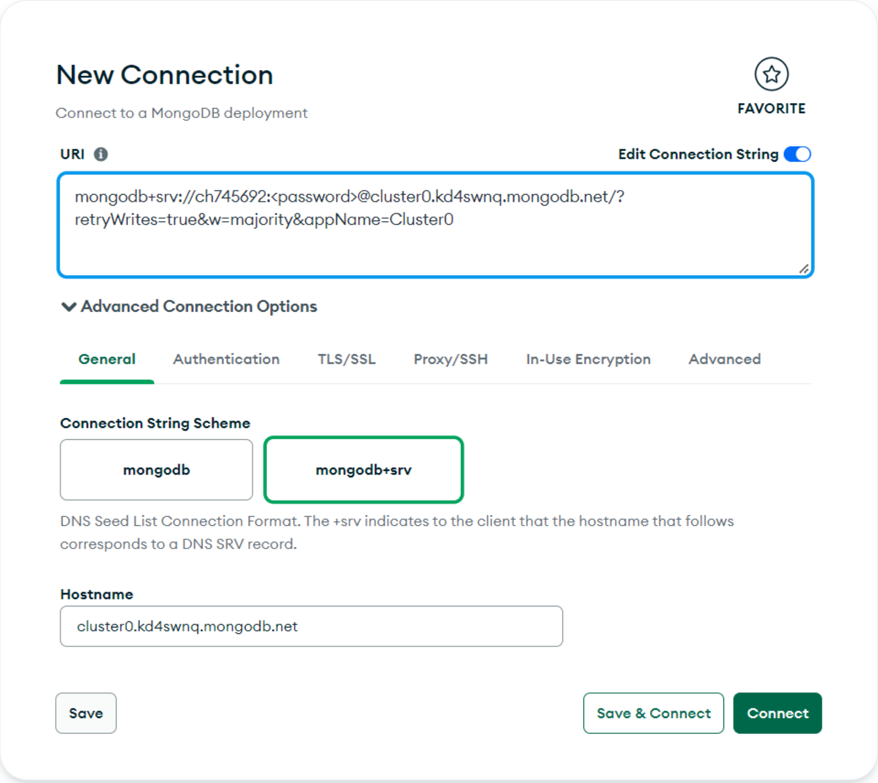
2. Copy the connection string, then open MongoDB Compass

```
mongodb+srv://ashish:<password>@cluster0.ncfp0dh.mongodb.net/
```

You will be prompted for the password for the ashish user's (Database User) username.
When entering your password, make sure that any special characters are URL encoded.

Figure 7: Connection Details

Connecting with MongoDB Compass [2]: Connected to the cluster using MongoDB Compass, with the URI entered in the connection field (Figure 8).



URI

Edit Connection String

```
mongodb+srv://ch745692:<password>@cluster0.kd4swnq.mongodb.net/?
retryWrites=true&w=majority&appName=Cluster0
```

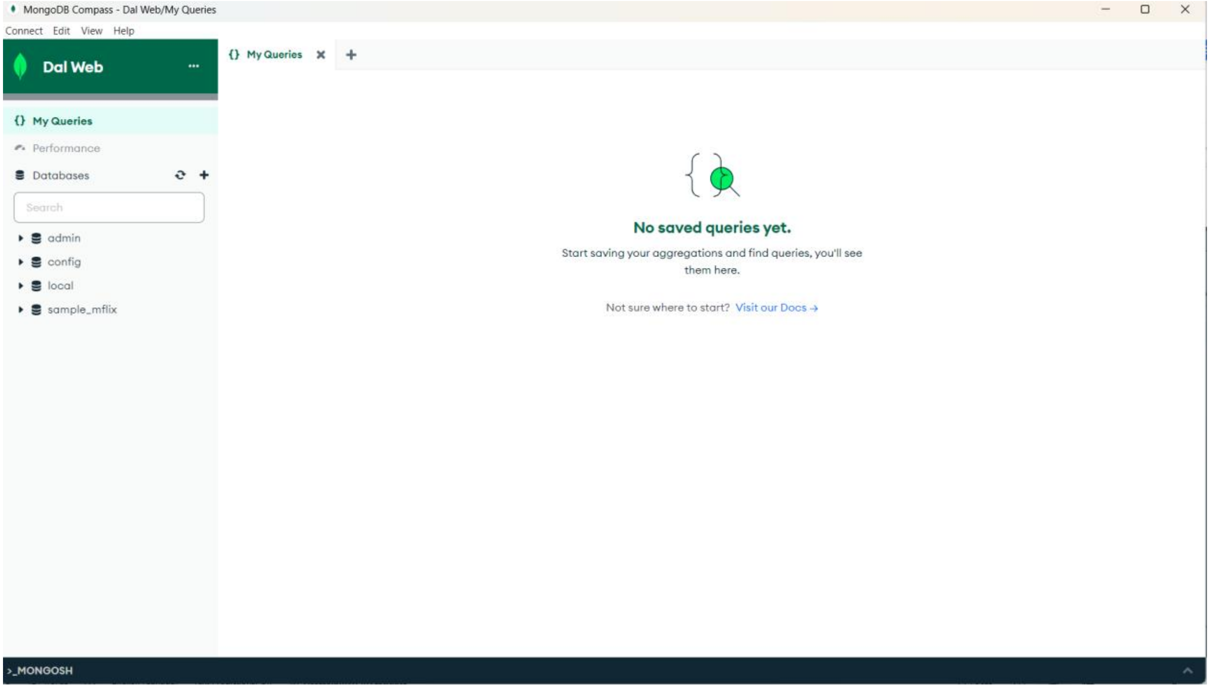
Connection String Scheme: mongodb+srv

DNS Seed List Connection Format. The +srv indicates to the client that the hostname that follows corresponds to a DNS SRV record.

Hostname: cluster0.kd4swnq.mongodb.net

Figure 8: Connecting to the cluster from MongoDB Compass

Figure 9 shows the Compass dashboard after connecting to the cluster.



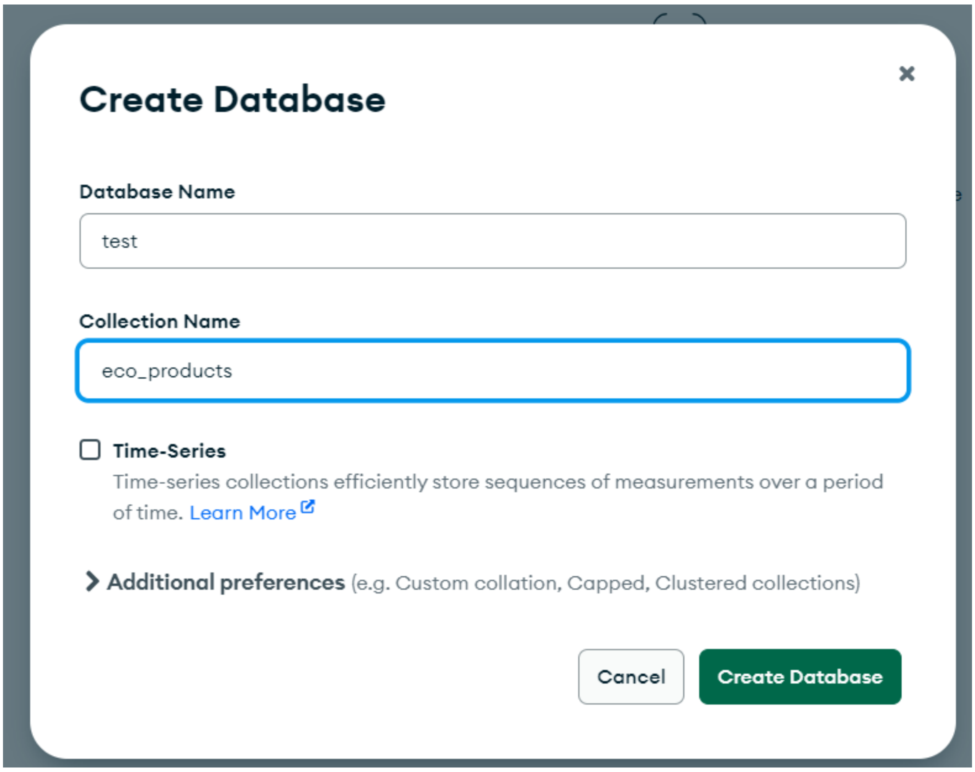
No saved queries yet.

Start saving your aggregations and find queries, you'll see them here.

Not sure where to start? Visit our Docs →

Figure 9: Compass dashboard

Database and Collection Creation: Created a database named 'test' and a collection named ‘eco_products’ in the cluster (Figure 10).


Create Database

Database Name: test

Collection Name: eco_products

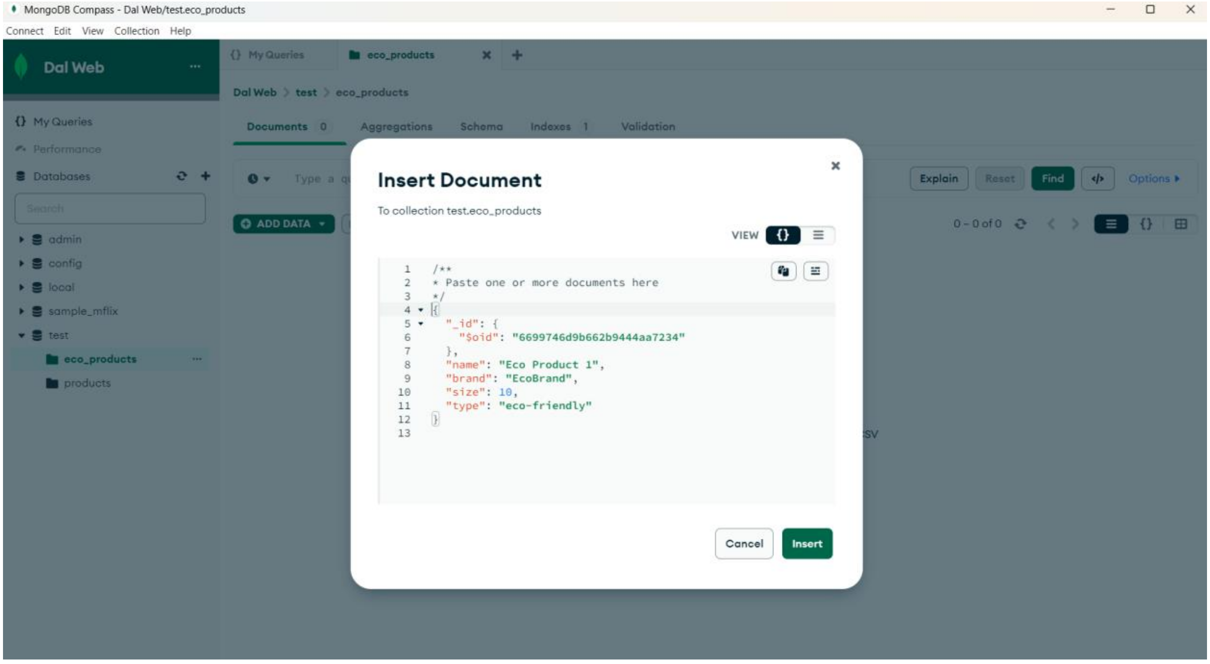
□ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. Learn More

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)
Cancel
Create Database

Figure 10: Database and Collection Creation

First Entry Insertion: Inserted the first entry manually into the cluster (Figure 11).



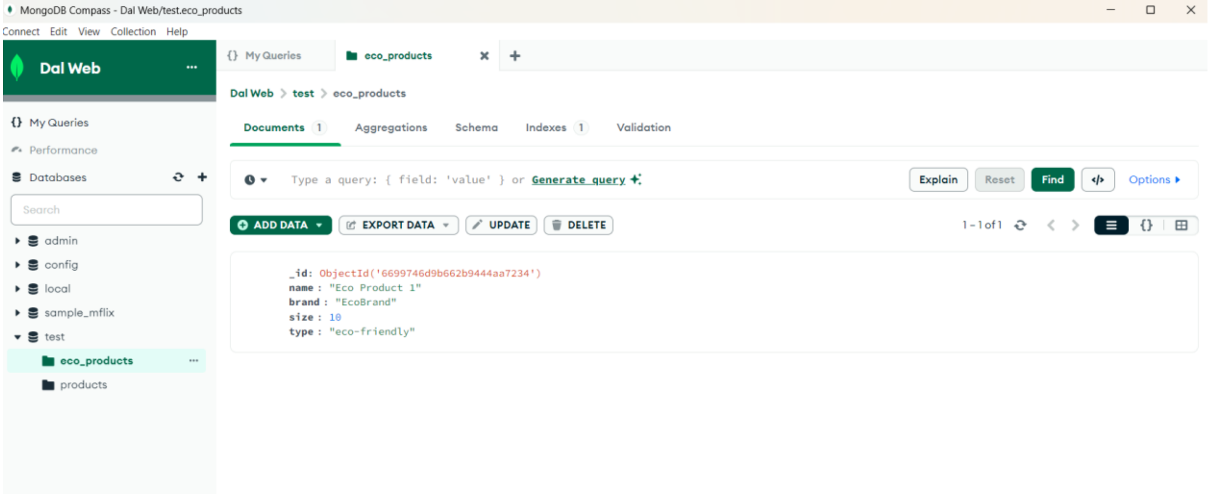
Insert Document

To collection test.eco_products

```
/**
* Paste one or more documents here
*/
{
  "_id": {
    "$oid": "6699746d9b662b9444aa7234"
  },
  "name": "Eco Product 1",
  "brand": "EcoBrand",
  "size": 10,
  "type": "eco-friendly"
}
```

Figure 11: Manual First Entry Insertion

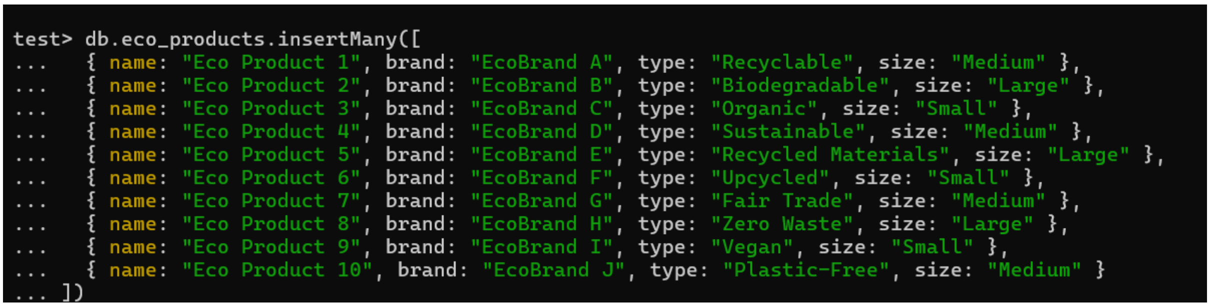
Inserted document in the collection (Figure 12).



```
_id: ObjectId('6699746d9b662b9444aa7234')
name : "Eco Product 1"
brand : "EcoBrand"
size : 10
type : "eco-friendly"
```

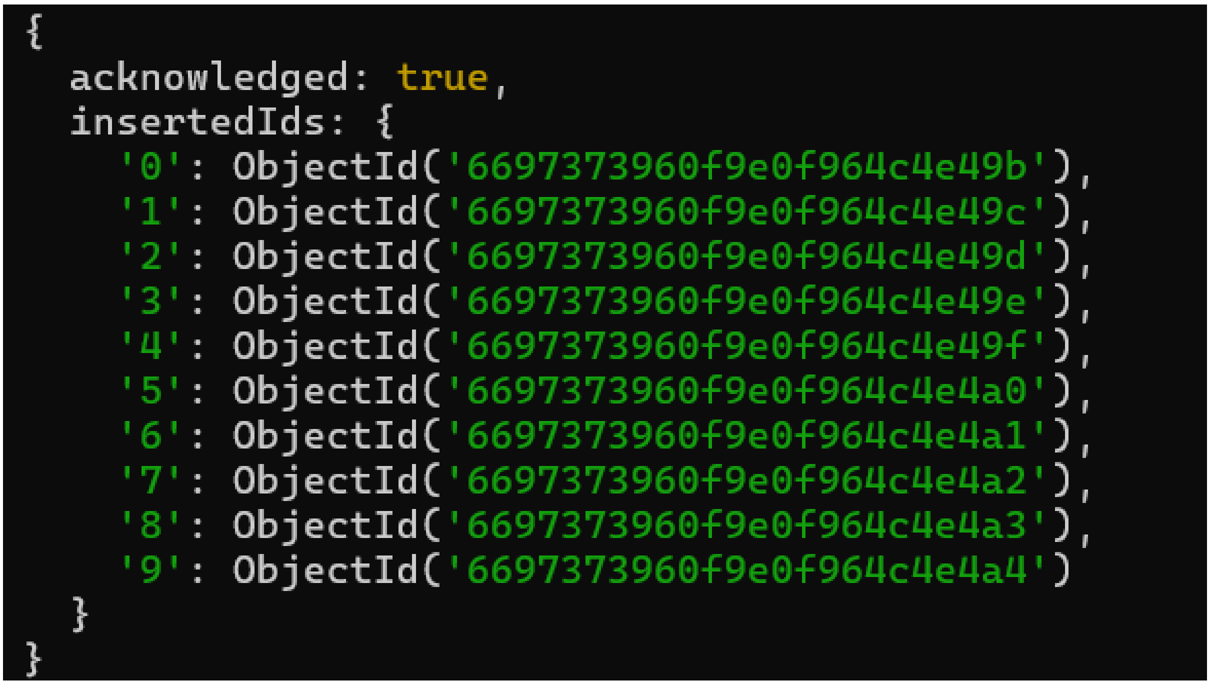
Figure 12: Manually Entered Document

Inserting Multiple Records: Query to insert multiple records into the collection (Figure 13) and its result (Figure 14).



```
test> db.eco_products.insertMany([
...   { name: "Eco Product 1", brand: "EcoBrand A", type: "Recyclable", size: "Medium" },
...   { name: "Eco Product 2", brand: "EcoBrand B", type: "Biodegradable", size: "Large" },
...   { name: "Eco Product 3", brand: "EcoBrand C", type: "Organic", size: "Small" },
...   { name: "Eco Product 4", brand: "EcoBrand D", type: "Sustainable", size: "Medium" },
...   { name: "Eco Product 5", brand: "EcoBrand E", type: "Recycled Materials", size: "Large" },
...   { name: "Eco Product 6", brand: "EcoBrand F", type: "Upcycled", size: "Small" },
...   { name: "Eco Product 7", brand: "EcoBrand G", type: "Fair Trade", size: "Medium" },
...   { name: "Eco Product 8", brand: "EcoBrand H", type: "Zero Waste", size: "Large" },
...   { name: "Eco Product 9", brand: "EcoBrand I", type: "Vegan", size: "Small" },
...   { name: "Eco Product 10", brand: "EcoBrand J", type: "Plastic-Free", size: "Medium" }
... ])
```

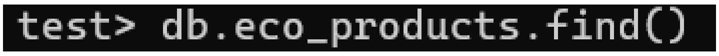
Figure 13: Query to Insert Multiple Records



```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6697373960f9e0f964c4e49b'),
    '1': ObjectId('6697373960f9e0f964c4e49c'),
    '2': ObjectId('6697373960f9e0f964c4e49d'),
    '3': ObjectId('6697373960f9e0f964c4e49e'),
    '4': ObjectId('6697373960f9e0f964c4e49f'),
    '5': ObjectId('6697373960f9e0f964c4e4a0'),
    '6': ObjectId('6697373960f9e0f964c4e4a1'),
    '7': ObjectId('6697373960f9e0f964c4e4a2'),
    '8': ObjectId('6697373960f9e0f964c4e4a3'),
    '9': ObjectId('6697373960f9e0f964c4e4a4')
  }
}
```

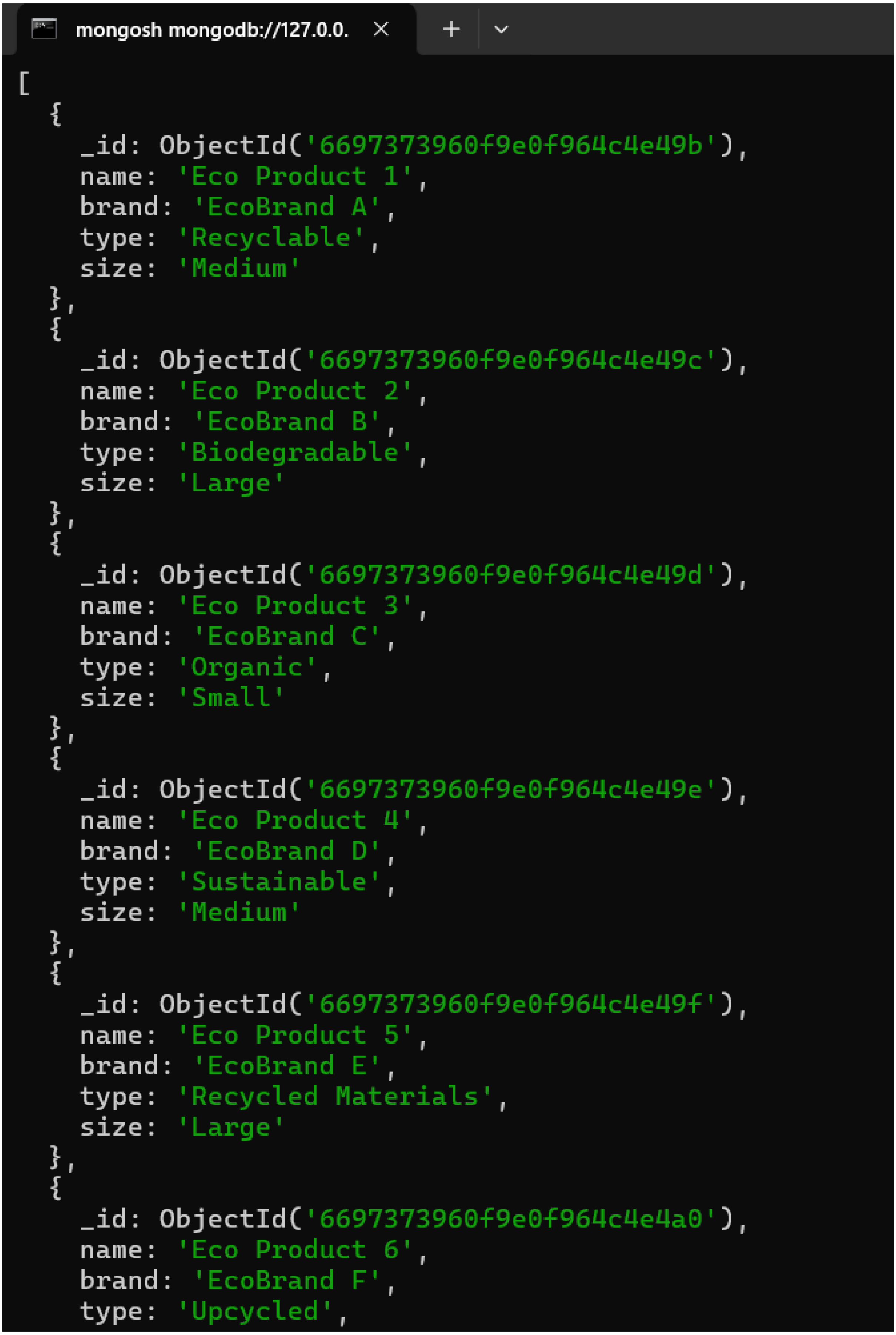
Figure 14: Result of Multiple Records Insertion

Retrieving All Records: Query to retrieve all records from the collection (Figure 15) and its result (Figures 16 and 17).



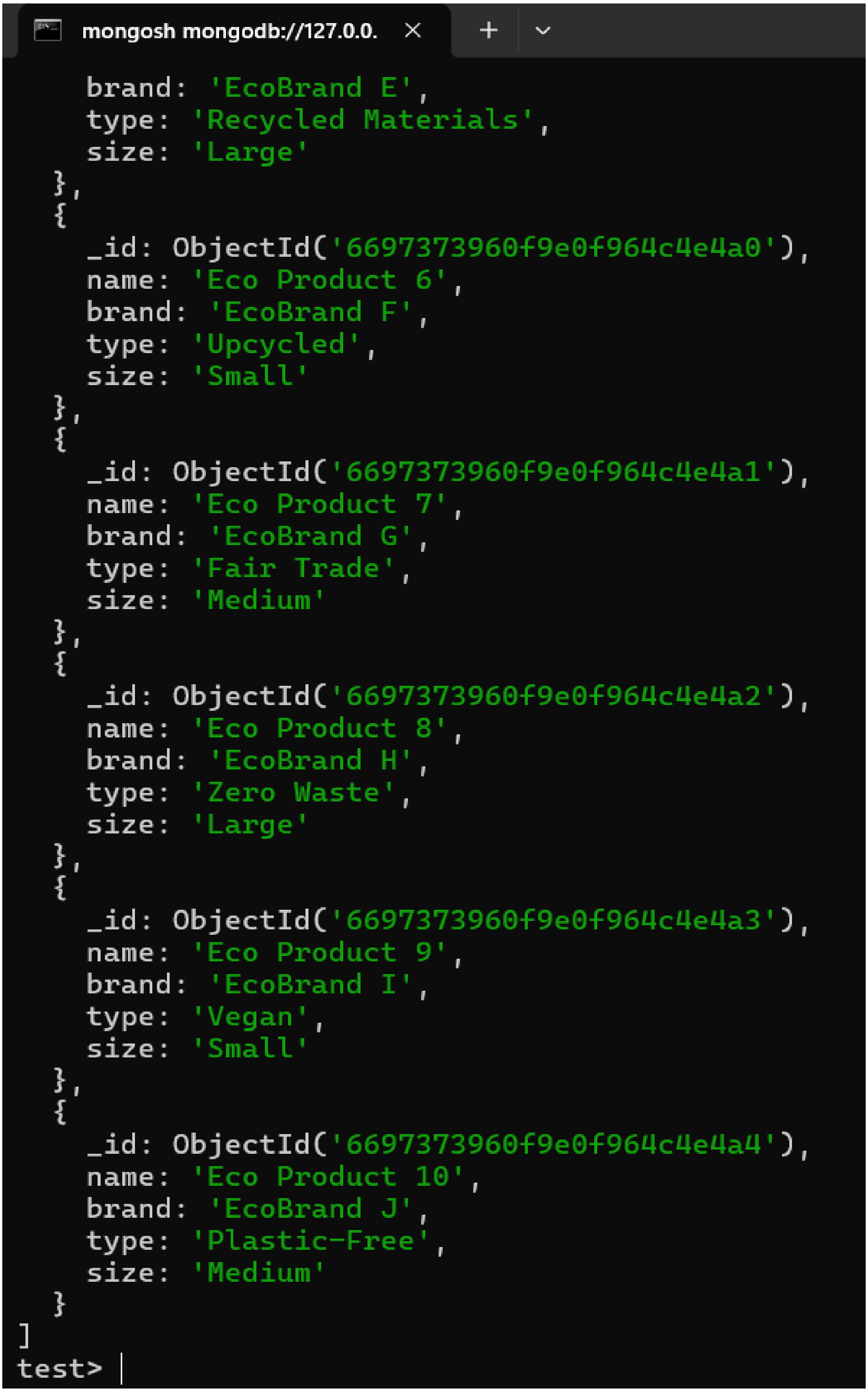
```
test> db.eco_products.find()
```

Figure 15: Query to Retrieve All Records



```
[
  {
    _id: ObjectId('6697373960f9e0f964c4e49b'),
    name: 'Eco Product 1',
    brand: 'EcoBrand A',
    type: 'Recyclable',
    size: 'Medium'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e49c'),
    name: 'Eco Product 2',
    brand: 'EcoBrand B',
    type: 'Biodegradable',
    size: 'Large'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e49d'),
    name: 'Eco Product 3',
    brand: 'EcoBrand C',
    type: 'Organic',
    size: 'Small'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e49e'),
    name: 'Eco Product 4',
    brand: 'EcoBrand D',
    type: 'Sustainable',
    size: 'Medium'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e49f'),
    name: 'Eco Product 5',
    brand: 'EcoBrand E',
    type: 'Recycled Materials',
    size: 'Large'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a0'),
    name: 'Eco Product 6',
    brand: 'EcoBrand F',
    type: 'Upcycled',
```

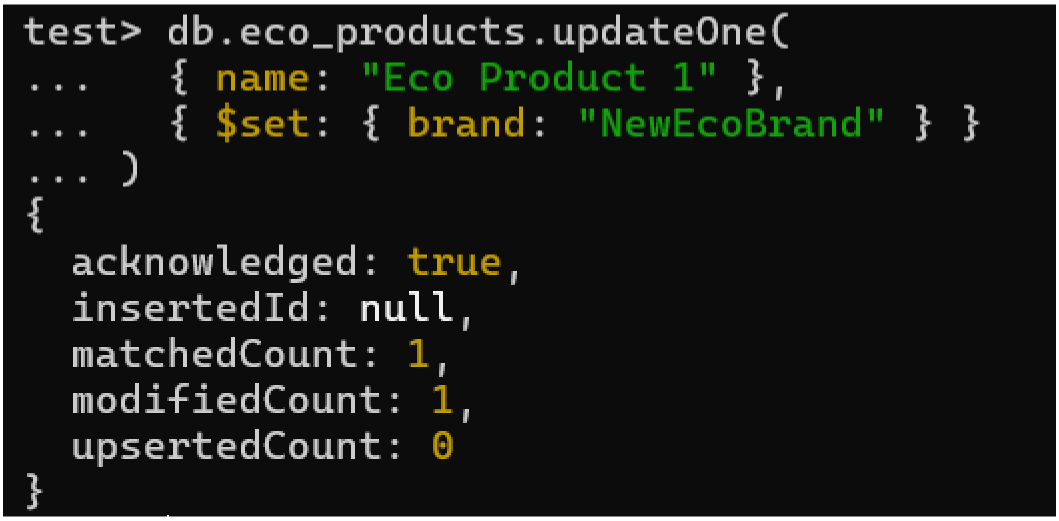
Figure 16: Result of Query to Retrieve All Records (Part 1)



```
    brand: 'EcoBrand E',
    type: 'Recycled Materials',
    size: 'Large'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a0'),
    name: 'Eco Product 6',
    brand: 'EcoBrand F',
    type: 'Upcycled',
    size: 'Small'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a1'),
    name: 'Eco Product 7',
    brand: 'EcoBrand G',
    type: 'Fair Trade',
    size: 'Medium'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a2'),
    name: 'Eco Product 8',
    brand: 'EcoBrand H',
    type: 'Zero Waste',
    size: 'Large'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a3'),
    name: 'Eco Product 9',
    brand: 'EcoBrand I',
    type: 'Vegan',
    size: 'Small'
  },
  {
    _id: ObjectId('6697373960f9e0f964c4e4a4'),
    name: 'Eco Product 10',
    brand: 'EcoBrand J',
    type: 'Plastic-Free',
    size: 'Medium'
  }
]
test>
```

Figure 17: Result of Query to Retrieve All Records (Part 2)

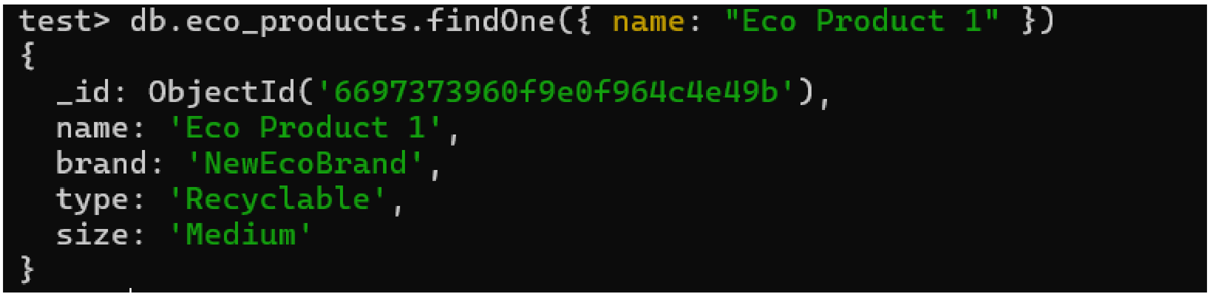
Modifying a Single Record: Query to modify a single record and its result (Figure 18).



```
test> db.eco_products.updateOne(
...   { name: "Eco Product 1" },
...   { $set: { brand: "NewEcoBrand" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Figure 18: Query to Modify a Single Record and Result

Retrieving Modified Record: Retrieval query to get the modified record (Figure 19).



```
test> db.eco_products.findOne({ name: "Eco Product 1" })
{
  _id: ObjectId('6697373960f9e0f964c4e49b'),
  name: 'Eco Product 1',
  brand: 'NewEcoBrand',
  type: 'Recyclable',
  size: 'Medium'
}
```

Figure 19: Retrieval of Modified Record

References:

[1] "MongoDB Documentation." MongoDB. Accessed: Jul. 17, 2024. [Online]. Available: https://www.mongodb.com/docs/

[2] "What is MongoDB Compass? - MongoDB Compass." MongoDB. Accessed: Jul. 17, 2024. [Online]. Available: https://www.mongodb.com/docs/compass/current/