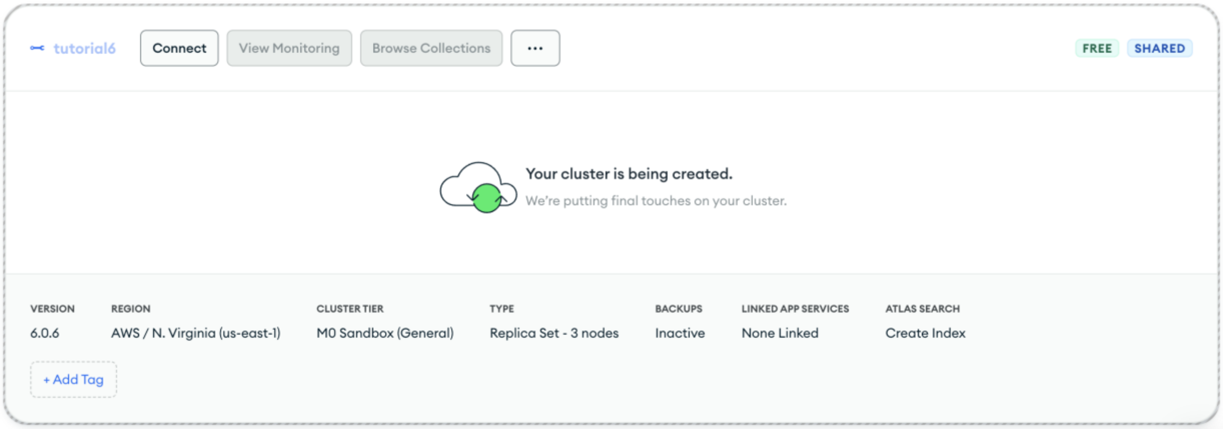
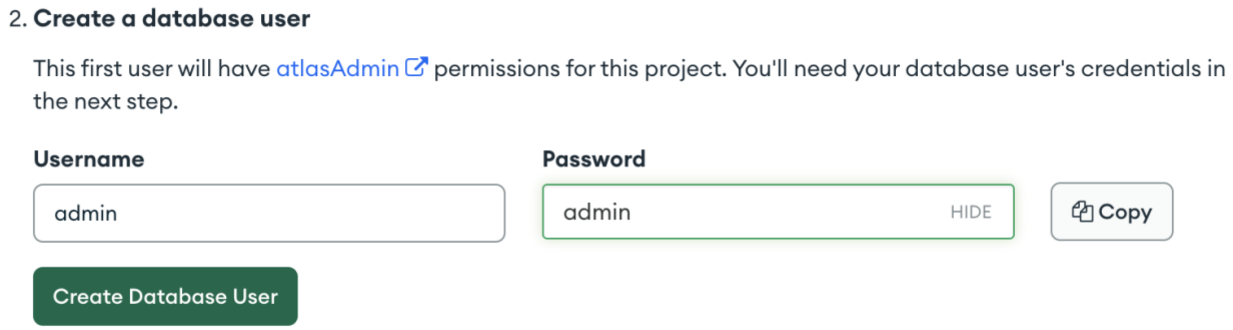


Your cluster is being created.

We're putting final touches on your cluster.

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
6.0.6	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

Figure 1: Creating Cluster (Instance Hosted)



2. **Create a database user**

This first user will have atlasAdmin permissions for this project. You'll need your database user's credentials in the next step.

Username

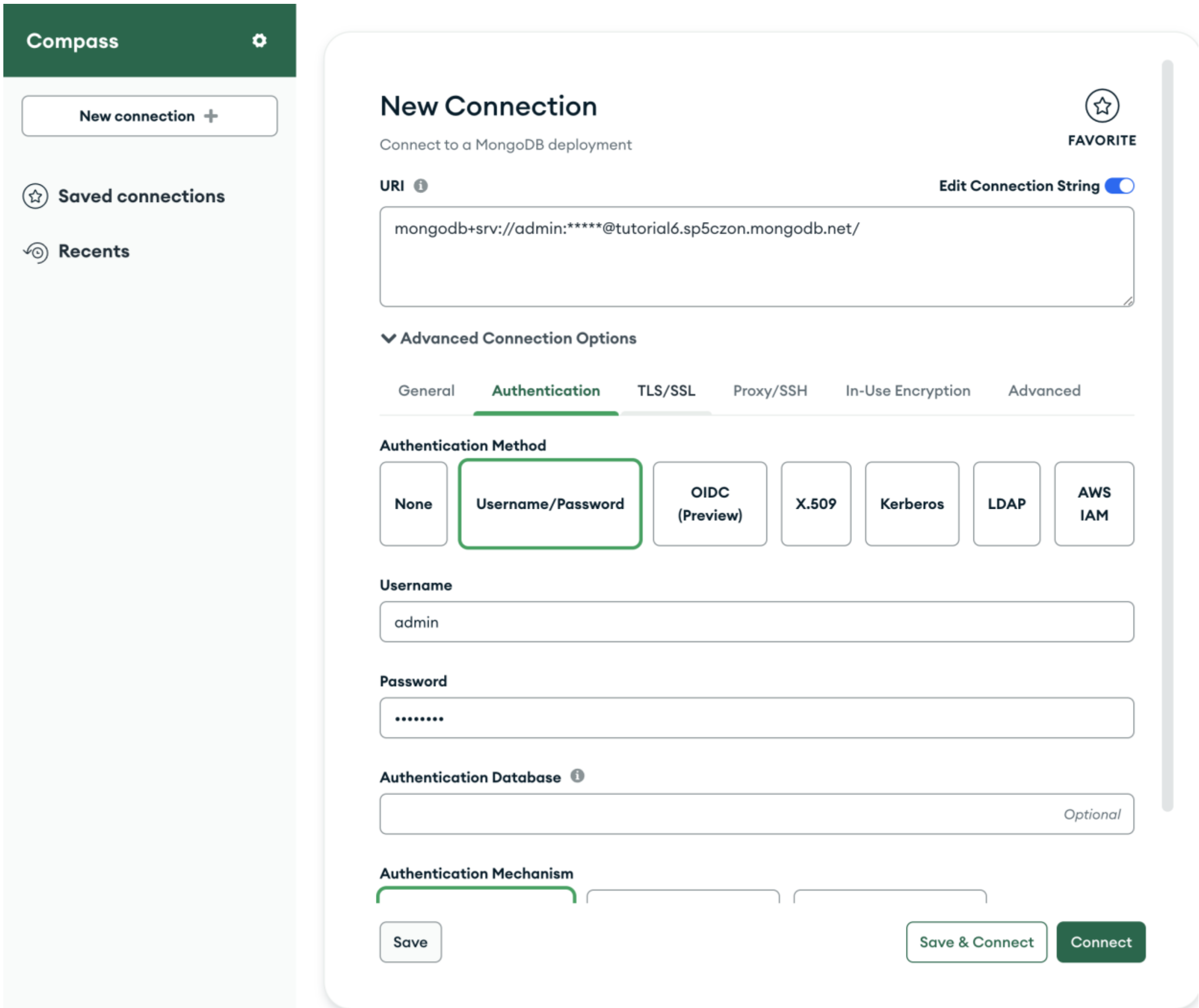
admin

Password

admin

Figure 2: Create a database user

Figure 1 & Figure 2 represents steps to follow on MongoDB Atlas website (https://cloud.mongodb.com/).



New Connection

Connect to a MongoDB deployment

URI

mongodb+srv://admin:*****@tutorial6.sp5czon.mongodb.net/

Authentication Method: ○ None ● Username/Password ○ OIDC (Preview) ○ X.509 ○ Kerberos ○ LDAP ○ AWS IAM

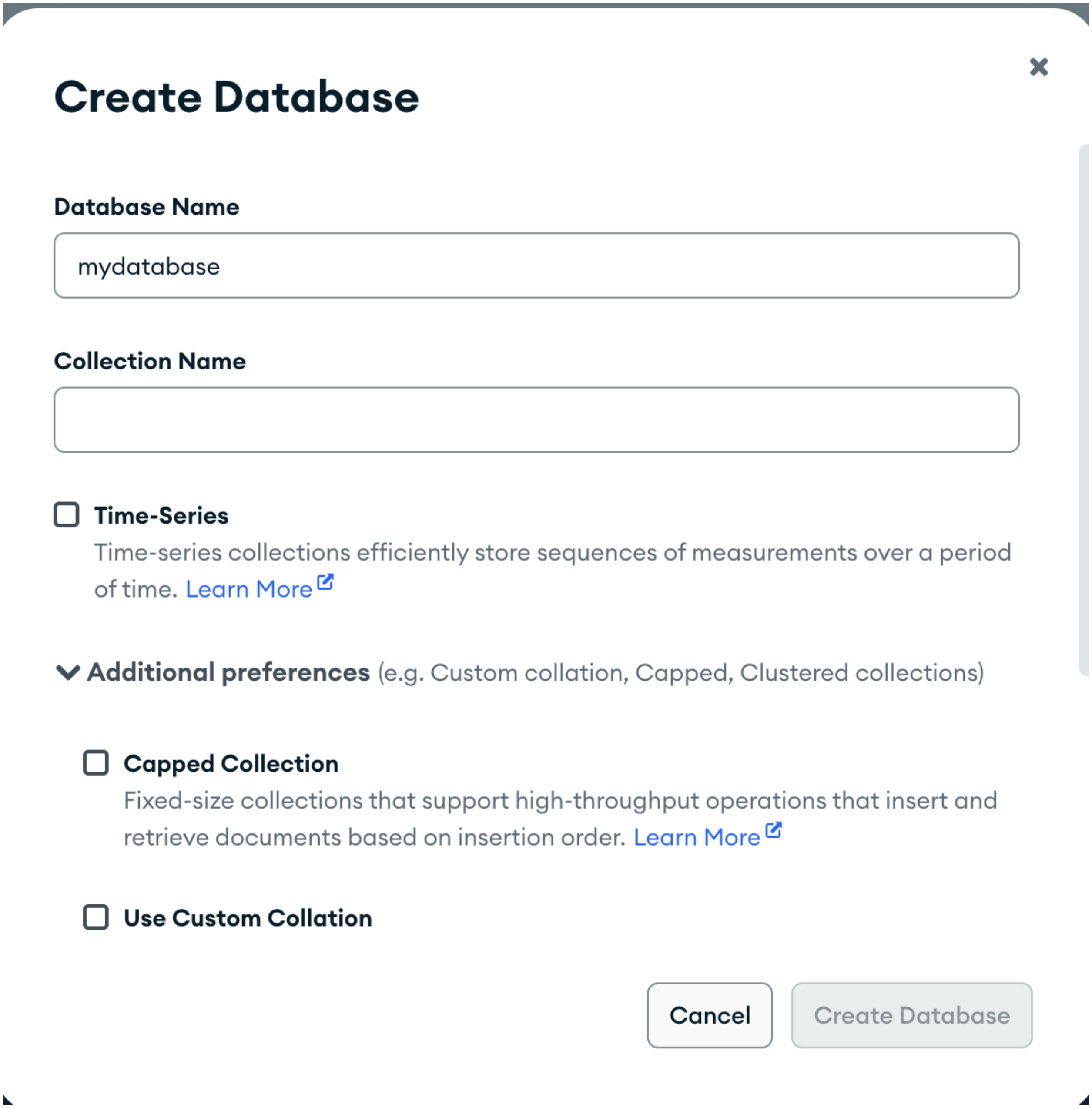
Username: admin

Password: ••••••••

Authentication Database: Optional

Authentication Mechanism

Figure 3: Connecting to instance



Create Database

Database Name

mydatabase

Collection Name

□ Time-Series

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

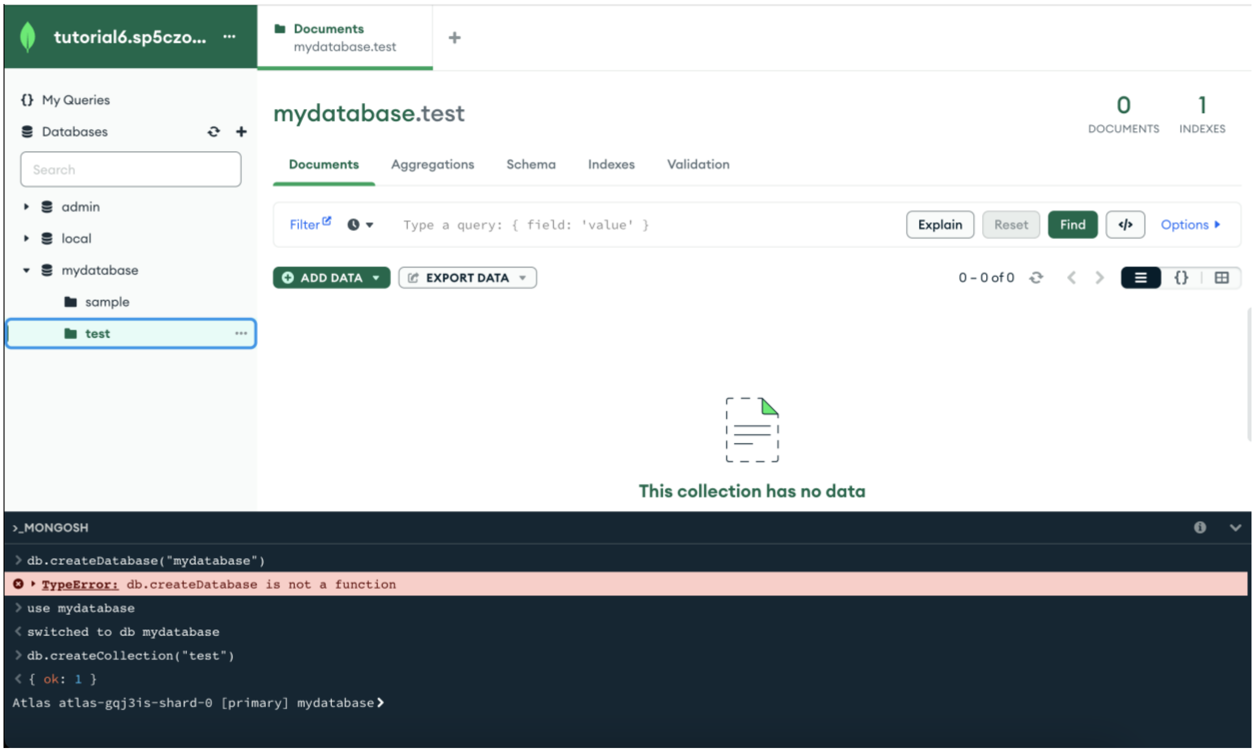
Additional preferences (e.g. Custom collation, Capped, Clustered collections)

□ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. Learn More

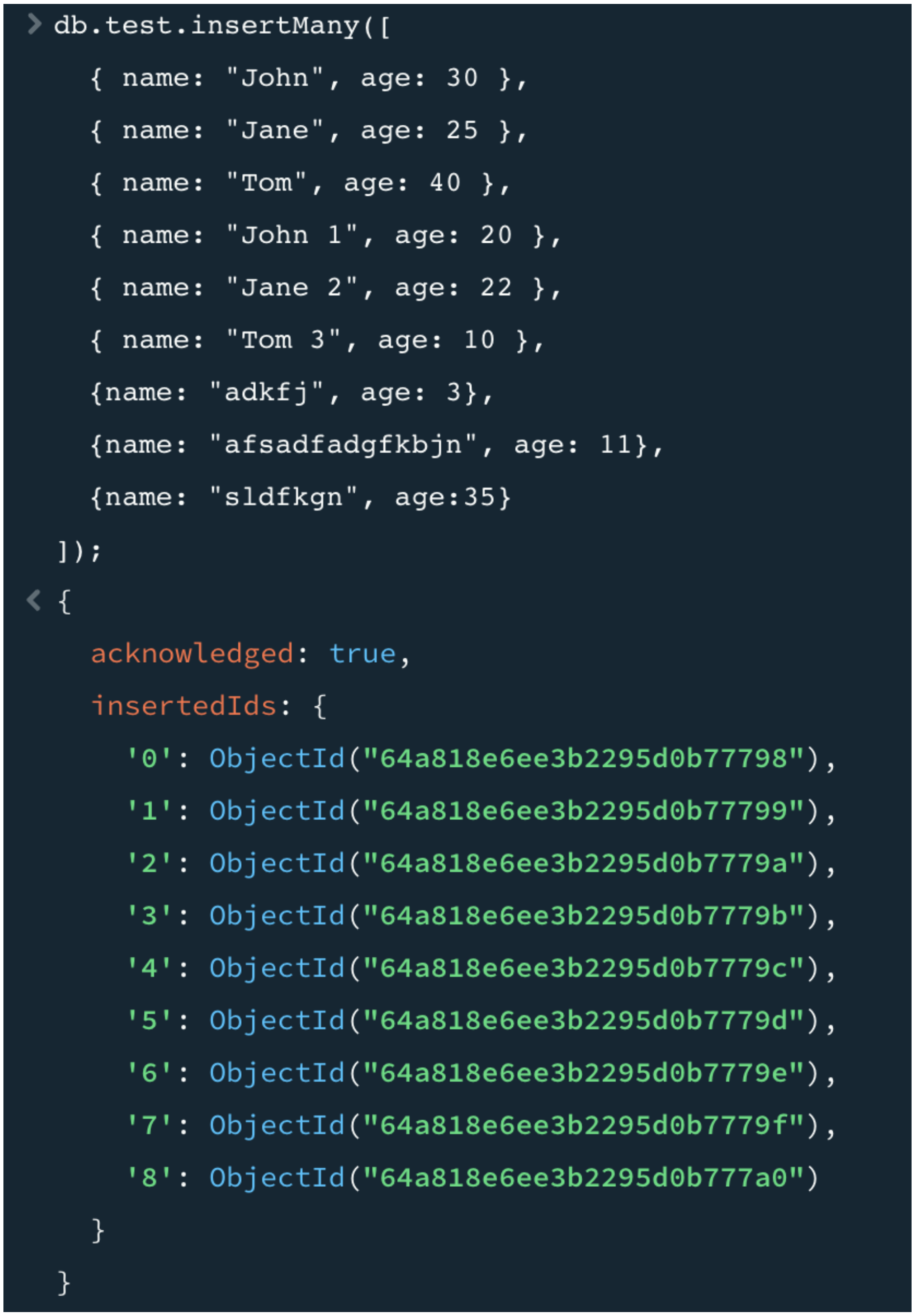
□ Use Custom Collation

Figure 4: Create Database and initial collection (Create a simple database schema and attach to the instance)



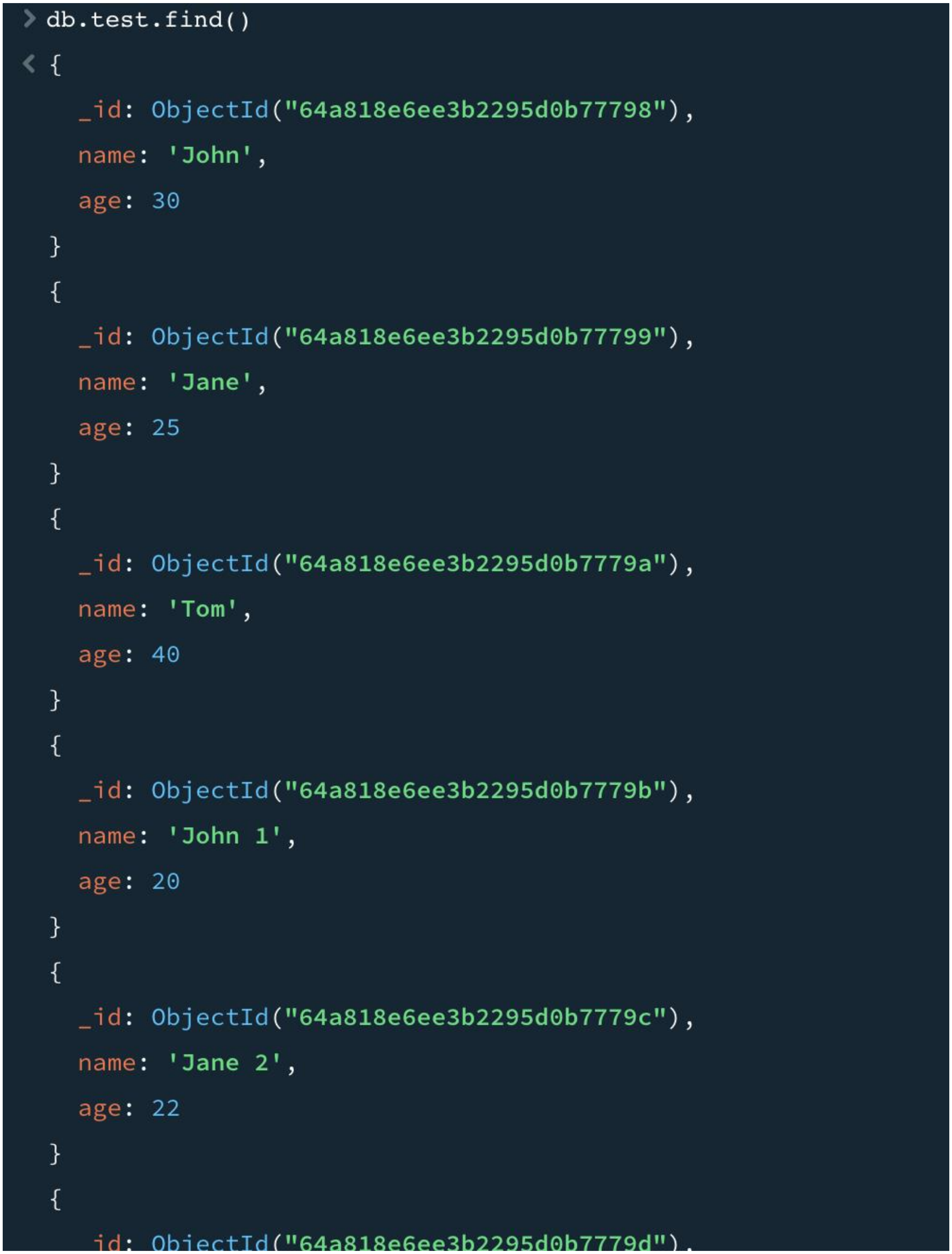
```
> db.createDatabase("mydatabase")
✖ ▸ TypeError: db.createDatabase is not a function
> use mydatabase
< switched to db mydatabase
> db.createCollection("test")
< { ok: 1 }
Atlas atlas-gqj3is-shard-0 [primary] mydatabase>
```

Figure 5: Create a collection



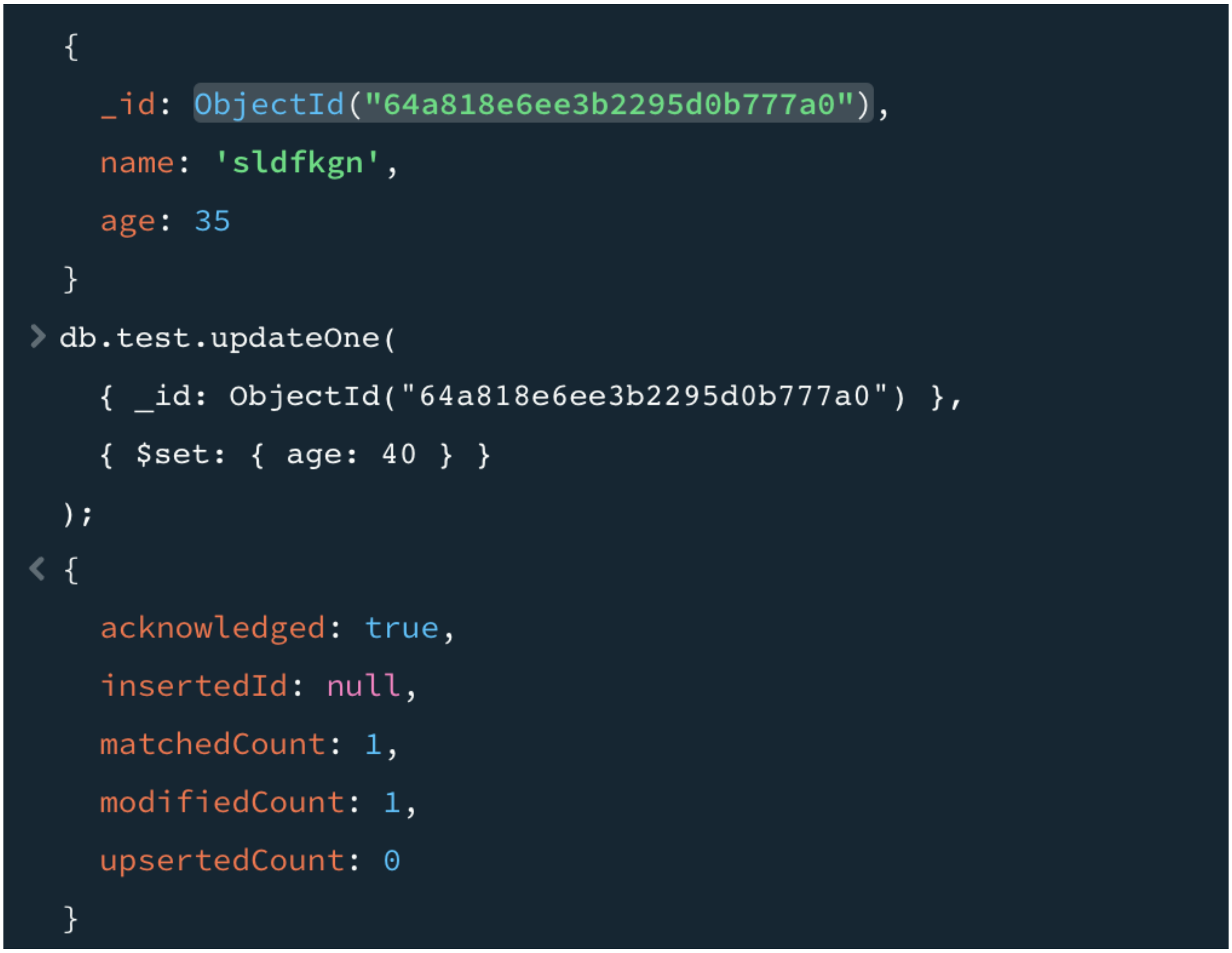
```
> db.test.insertMany([
    { name: "John", age: 30 },
    { name: "Jane", age: 25 },
    { name: "Tom", age: 40 },
    { name: "John 1", age: 20 },
    { name: "Jane 2", age: 22 },
    { name: "Tom 3", age: 10 },
    {name: "adkfj", age: 3},
    {name: "afsadfadgfkbjn", age: 11},
    {name: "sldfkgn", age:35}
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("64a818e6ee3b2295d0b77798"),
      '1': ObjectId("64a818e6ee3b2295d0b77799"),
      '2': ObjectId("64a818e6ee3b2295d0b7779a"),
      '3': ObjectId("64a818e6ee3b2295d0b7779b"),
      '4': ObjectId("64a818e6ee3b2295d0b7779c"),
      '5': ObjectId("64a818e6ee3b2295d0b7779d"),
      '6': ObjectId("64a818e6ee3b2295d0b7779e"),
      '7': ObjectId("64a818e6ee3b2295d0b7779f"),
      '8': ObjectId("64a818e6ee3b2295d0b777a0")
    }
  }
```

Figure 6: Insert Query (A query to insert multiple records and result)



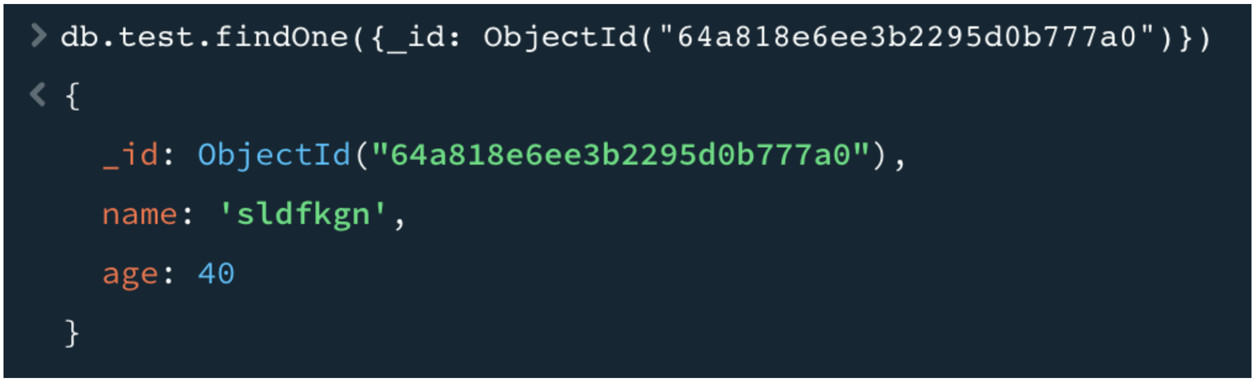
```
> db.test.find()
< {
    _id: ObjectId("64a818e6ee3b2295d0b77798"),
    name: 'John',
    age: 30
  }
  {
    _id: ObjectId("64a818e6ee3b2295d0b77799"),
    name: 'Jane',
    age: 25
  }
  {
    _id: ObjectId("64a818e6ee3b2295d0b7779a"),
    name: 'Tom',
    age: 40
  }
  {
    _id: ObjectId("64a818e6ee3b2295d0b7779b"),
    name: 'John 1',
    age: 20
  }
  {
    _id: ObjectId("64a818e6ee3b2295d0b7779c"),
    name: 'Jane 2',
    age: 22
  }
  {
    _id: ObjectId("64a818e6ee3b2295d0b7779d"),
```

Figure 7: Retrieval query to get all records. (A query to retrieve multiple records and result)



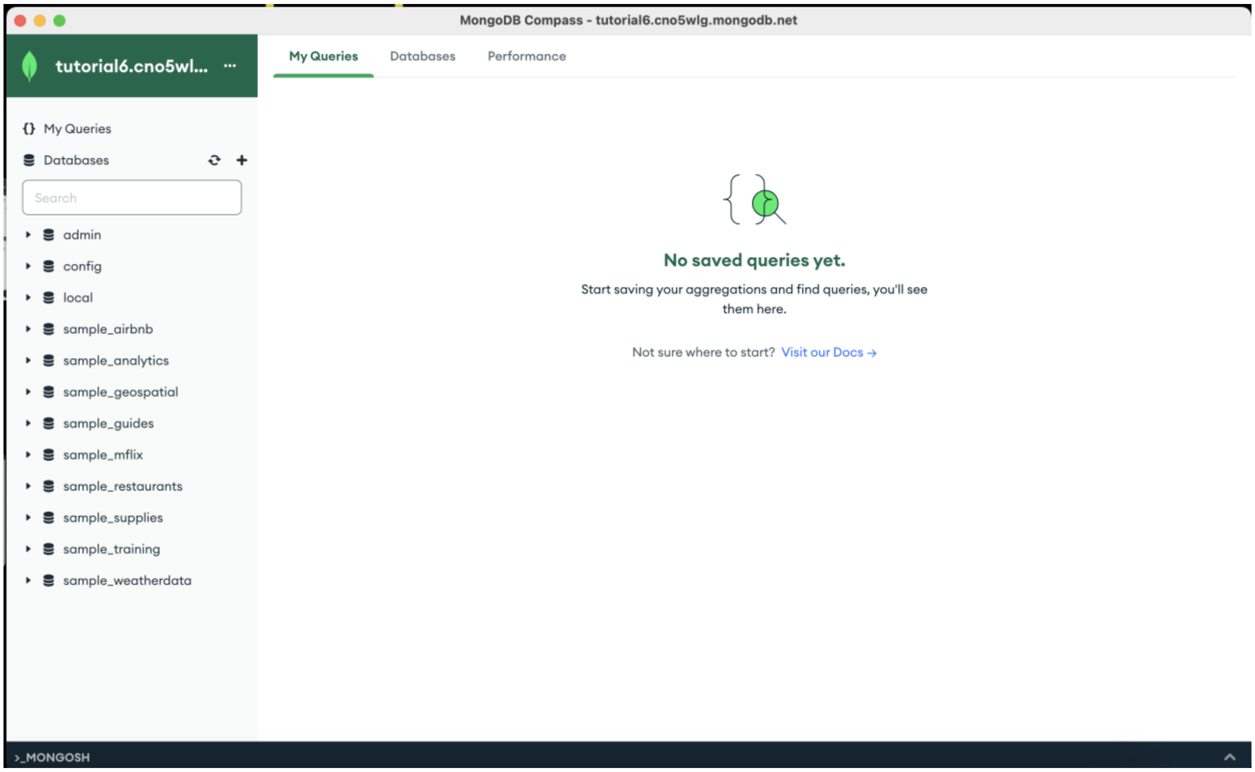
```
  {
    _id: ObjectId("64a818e6ee3b2295d0b777a0"),
    name: 'sldfkgn',
    age: 35
  }
> db.test.updateOne(
    { _id: ObjectId("64a818e6ee3b2295d0b777a0") },
    { $set: { age: 40 } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

Figure 8: Update query with old record (A query to update a record and result)



```
> db.test.findOne({_id: ObjectId("64a818e6ee3b2295d0b777a0")})
< {
    _id: ObjectId("64a818e6ee3b2295d0b777a0"),
    name: 'sldfkgn',
    age: 40
  }
```

Figure 9: Retrieve modified record (A query to retrieve a specific record and result)

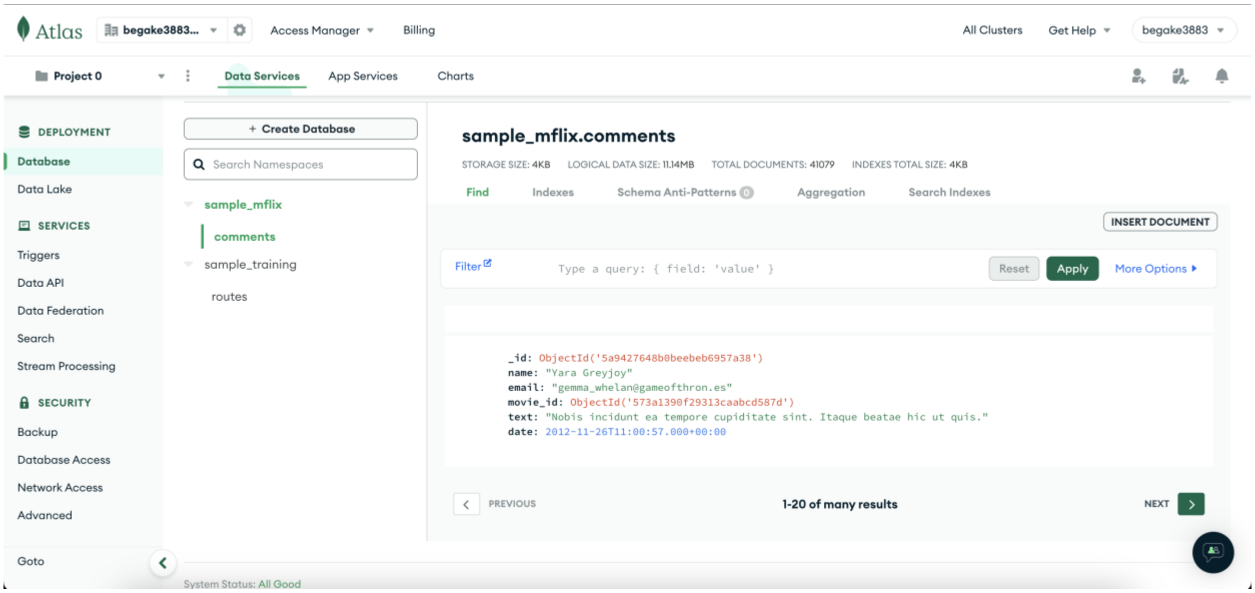


No saved queries yet.

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

Not sure where to start? Visit our Docs →

Figure 10: Database connected using created admin user



sample_mflix.comments

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 11.14MB TOTAL DOCUMENTS: 41079 INDEXES TOTAL SIZE: 4KB

```
_id: ObjectId('5a9427648b0beebeb6957a38')
name: "Yara Greyjoy"
email: "gemma_whelan@gameofthron.es"
movie_id: ObjectId('573a1390f29313caabcd587d')
text: "Nobis incidunt ea tempore cupiditate sint. Itaque beatae hic ut quis."
date: 2012-11-26T11:00:57.000+00:00
```

1-20 of many results

Figure 11: Sample Database schema

Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9 & Figure 10 is done on MongoDB Compass application.
Caption represents description of steps performed.