

Running MongoDB service for macOS and Windows

Here are the instructions for starting and stopping the MongoDB service on both macOS and Windows operating systems:

Service on macOS:

To start, stop, restart, and list MongoDB services on macOS, you can use the following commands:

1. Start MongoDB service:

...

```
brew services start mongodb-community@5.0
```

or

```
sudo brew services start mongodb-community
```

...

2. Stop MongoDB service:

...

```
brew services stop mongodb-community@5.0
```

or

```
sudo brew services stop mongodb-community
```

...

3. Restart MongoDB service:

...

```
brew services restart mongodb-community@5.0
```

or

```
sudo brew services restart mongodb-community
```

...

4. List MongoDB services:

...

```
brew services list
```

...

5. Install MongoDB compass:

- Restart device

...

```
brew install --cask mongodb-compass
```

...

The above commands assume that you have MongoDB installed using Homebrew and the MongoDB service is managed by Homebrew services. The version specified in the commands is 5.0, but you can replace it with your installed MongoDB version if different.

Service on Windows:

To start, stop, restart, and list MongoDB services on Windows, you can use the following commands:

1. Start MongoDB service:

Open the Command Prompt as an administrator and run the command:

...

```
net start MongoDB
```

...

2. Stop MongoDB service:

Open the Command Prompt as an administrator and execute the command:

...

```
net stop MongoDB
```

...

3. Restart MongoDB service:

Open the Command Prompt as an administrator and enter the command:

...

```
net stop MongoDB && net start MongoDB
```

...

4. List MongoDB services:

Open the Command Prompt as an administrator and use the command:

```
...
```

```
sc query MongoDB
```

```
...
```

These commands assume that you have MongoDB installed as a Windows service. The "net start" command starts the MongoDB service, "net stop" stops the MongoDB service, and "sc query" lists the MongoDB service information.

Please note that you might need to adjust the command if your MongoDB service has a different name or if you have multiple MongoDB services installed.

Hands-on Walkthrough:

MongoDB Compass

1. Visual Database Navigation:

- Launch MongoDB Compass and connect to your MongoDB server.
- Explore the list of databases and collections in the left sidebar.
- Click on a collection to view its documents.

2. Schema Exploration:

- Select a collection and click on the "Schema" tab.
- Observe the fields, their data types, and the distribution of values.
- Use this information to understand the structure of your data.

3. Query Building:

- Switch to the "Query" tab.
- Use the query builder interface to visually define filters, projections, and sorting criteria.
- Observe the generated MongoDB query code on the right side.
- Click "Find" to execute the query and view the results.

4. Document Editing and Creation:

- Select a document from the collection view.
- Click the "Edit Document" button.

- Modify field values, add new fields, or remove existing fields.
- Save the changes to update the document.

5. Aggregation Pipeline Visualization:

- Switch to the "Aggregations" tab.
- Click "New Pipeline" to create a new aggregation pipeline.
- Add stages to the pipeline by clicking the "+" button.
- Configure each stage with the desired parameters.
- Observe the intermediate results and adjust the pipeline as needed.

6. Performance Monitoring:

- Go to the "Performance" tab.
- View real-time metrics such as CPU usage, memory usage, and network traffic.
- Analyze query performance statistics and identify potential performance issues.
- Utilize the information to optimize your database operations.

MongoDB Shell

1. Interactive JavaScript Environment:

- Open the MongoDB Shell in your terminal or command prompt.
- Start by connecting to your MongoDB server using the appropriate connection string.

2. Command Execution:

- Execute administrative commands like ``show databases`` to list available databases.
- Switch to a specific database using the ``use`` command.
- Perform CRUD operations using commands like ``insertOne``, ``find``, ``updateOne``, ``deleteOne``, etc.

3. Querying and Aggregation:

- Write queries using the MongoDB Query Language (MQL) to retrieve specific documents.
- Utilize operators like ``$match``, ``$project``, ``$sort``, ``$limit``, etc., to filter and manipulate data.
- Construct aggregation pipelines using stages such as ``$match``, ``$group``, ``$sort``, ``$project``, etc.

4. Database Administration:

- Use administrative commands like ``db.createCollection``, ``db.dropDatabase``, ``db.createIndex``, etc., to manage databases and collections.

- Perform backup and restore operations using commands like ``mongodump`` and ``mongorestore``.

5. Scripting and Automation:

- Create JavaScript scripts using your preferred text editor or IDE.
- Write code to automate tasks like data imports, migrations, or database provisioning.
- Execute scripts in the MongoDB Shell using the ``load`` command or by running the script file directly.

6. Diagnostic and Debugging Tools:

- Utilize built-in commands and methods to diagnose issues and optimize performance.
- Access server logs using the ``show log`` command.
- Monitor server statistics using the ``db.stats()`` method.
- Analyze query performance using the ``explain()`` method.

By following this walkthrough, you can gain hands-on experience with MongoDB Compass and the MongoDB Shell. These tools provide a user-friendly GUI and a powerful command-line interface, respectively, for managing, querying, and interacting with MongoDB databases.

Managing databases, collections, and documents

Next, we have a walkthrough on using MongoDB tools to create, modify, and delete databases, collections, and documents while emphasizing the importance of proper database management:

1. Start by launching the MongoDB Shell or MongoDB Compass, depending on your preferred interface.

2. Creating a Database:

- Use the command ``use <database_name>`` in the MongoDB Shell to switch to a specific database or select the database from the MongoDB Compass interface.
- To create a new database, execute the command ``db.createCollection("<collection_name>")`` in the MongoDB Shell or use the GUI option in MongoDB Compass.

3. Modifying a Database:

- To modify a database, you can use commands like ``db.<collection_name>.insertOne(<document>)`` to insert a new document,

`db.<collection_name>.updateOne(<filter>, <update>)`` to update an existing document, or
``db.<collection_name>.deleteOne(<filter>)`` to delete a document.

4. Creating and Modifying Collections:

- In MongoDB Compass, navigate to the selected database and use the GUI options to create a new collection or modify an existing one.
- In the MongoDB Shell, use the ``db.createCollection("<collection_name>")`` command to create a new collection.

5. Deleting Databases and Collections:

- In MongoDB Compass, select the database or collection you want to delete and use the GUI options to delete it.
- In the MongoDB Shell, use the ``db.<collection_name>.drop()`` command to delete a collection, and the ``db.dropDatabase()`` command to delete a database.

6. Proper Database Management:

It is essential to follow best practices for database management to ensure data integrity, performance, and scalability.

- Regularly review and optimize indexes to improve query performance.
- Implement data backups and disaster recovery plans to safeguard against data loss.
- Use access control mechanisms to secure the database and limit user privileges.
- Consider data modeling techniques to create efficient and maintainable database schemas.
- Monitor database performance and usage to identify bottlenecks and optimize resource utilization.

Proper database management is crucial to ensure the reliability, security, and efficiency of your MongoDB deployments. By using the appropriate tools and following best practices, you can create, modify, and delete databases, collections, and documents effectively while maintaining the health of your MongoDB environment.