Creating and configuring MongoDB involves setting up a database server, creating databases and collections, and customizing the configuration to suit your application needs. Here's a step-by-step guide:

---

1. Install MongoDB

On Linux:

- Add MongoDB's official repository to your package manager.
- Install MongoDB with `sudo apt install mongodb` or `sudo yum install mongodb-org` depending on your distribution.
- Start the MongoDB service using `sudo systemctl start mongod`.

On Windows:

- Download the MSI installer from the official MongoDB website.
- Follow the installation wizard steps.
- Start the MongoDB service from the Windows Services panel or use the command prompt.

On macOS:

Use Homebrew:
 brew tap mongodb/brew
brew install mongodb-community@<version>
brew services start mongodb/brew/mongodb-community

-

---

2. Run MongoDB

Ensure the MongoDB server (`mongod`) is running. By default:

- **Port**: 27017
- **Data Directory**: `/data/db` (Linux/macOS) or `C:\Program Files\MongoDB\Server\<version>\data` (Windows)
- **Log Directory**: `/var/log/mongodb` (Linux/macOS) or equivalent for Windows.

---

3. Connect to MongoDB

Use the MongoDB shell (`mongosh`):

```
mongosh
```

Once connected, you can start creating databases, collections, and documents.

---

4. Basic Configuration

Configuration File

The primary configuration file is typically located at:

- `/etc/mongod.conf` (Linux/macOS)
- `C:\Program Files\MongoDB\Server\<version>\bin\mongod.cfg` (Windows)

Common configurations:

Bind IP: To make MongoDB accessible remotely, edit:

```
net:
  bindIp: 0.0.0.0
```

-

Authentication: Enable access control:

```
security:
  authorization: enabled
```

- Restart MongoDB to apply changes.

---

5. Create a Database and Collections

In the shell:

```
// Switch to or create a new database
use myDatabase;

// Create a collection
db.createCollection("myCollection");

// Insert a document
db.myCollection.insertOne({ name: "Alice", age: 25 });
```

---

6. Create a User

To secure MongoDB:

Switch to the `admin` database:

```
use admin;
```

1.

Create a user with necessary roles:

```
db.createUser({
   user: "myUser",
   pwd: "myPassword",
   roles: [{ role: "readWrite", db: "myDatabase" }]
});
```

2.

---

7. Backup and Restore

Backup:

```
mongodump --db myDatabase --out /path/to/backup
```

-

Restore:

```
mongorestore --db myDatabase /path/to/backup/myDatabase
```

-

---

8. Monitoring

Use tools like:

- **MongoDB Compass**: A graphical interface.
- **Monitoring Services**: Integrate MongoDB Atlas or other third-party monitoring tools.

Let me know if you need assistance with a specific aspect of MongoDB setup!