The Document Database

```
private TsMongoContext()
{
        connString = ConfigurationManager.AppSettings[Constants.MongoConnectionString];
         rl = MongoUrl.Create(connString);
         lient = new MongoClient(url);
          client.GetDatabase(url.DatabaseName);
}

99+ re  ences
publi  static TsMongoContext Instance()
{
```

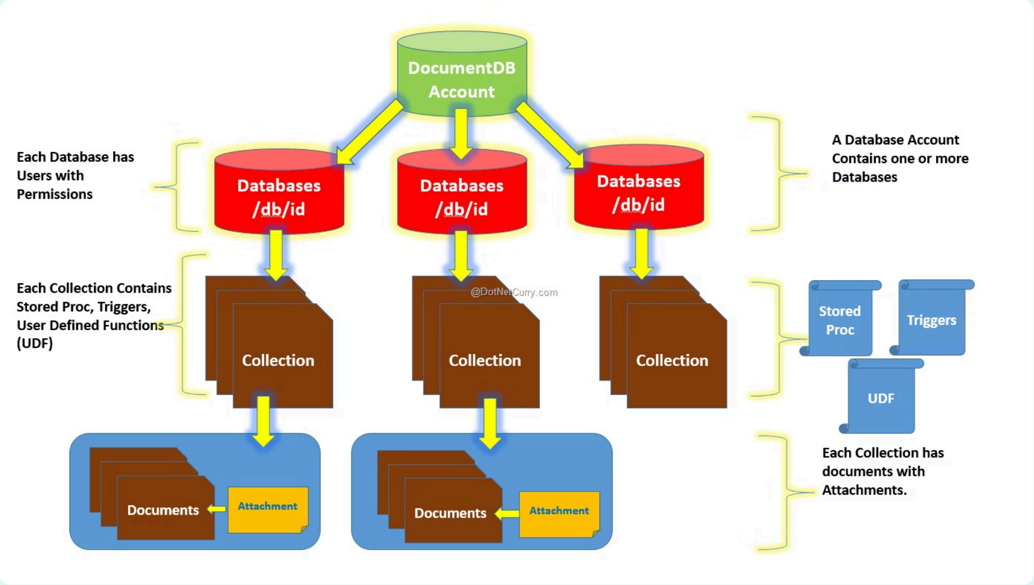
Ngoc My Nguyen

May 2025

Agenda

- 1. Documents and Document Databases
- 2. Company Overview
- 3. Licensing Evolution
- 4. Product Portfolio
- 5. IT Architecture
- 6. The Data Models
- 7. Client Libraries and Programming
- 8. Real-World Applications
- 9. Installation Options
- 9.1. Example: Easy installation and configuration with Docker Desktop (GUI)
- 9.2. Example: Easily explore and manipulate database with MongoDB Compass (GUI)
- 9.3. Example: Create a database and a collection with MongoDB Compass (GUI)
- 10. Manage the database

1. Documents and Document Databases


DocumentDB Account
Each Database has Users with Permissions
Databases /db/id
Databases /db/id
Databases /db/id
A Database Account Contains one or more Databases
Each Collection Contains Stored Proc, Triggers, User Defined Functions (UDF)
@DotNetCurry.com
Collection
Collection
Collection
Stored Proc
Triggers
UDF
Documents
Attachment
Documents
Attachment
Each Collection has documents with Attachments.

Document Databases

Non-relational databases that store data in the form of documents, typically JSON or BSON formats.

Flexible Data Models

Designed to handle semi-structured or unstructured data with dynamic schemas.

Scalability and Performance

Support rich querying capabilities, indexing, and horizontal scaling for modern applications.

2. Company Overview

2007

Founded as 10gen in New York City by former DoubleClick executives (Dwight Merriman, Eliot Horowitz, and Kevin Ryan)

2009

First production release of MongoDB database

2013

Company rebranded from 10gen to MongoDB

2014

Dev Ittycheria joined as CEO, leading the cloud transformation

2017

Launch of Atlas, MongoDB's cloud database service

Successful Initial public offering (IPO)

3. Licensing Evolution

Open Source Beginnings

MongoDB initially adopted the Affero General Public License (AGPL), ensuring that anyone modifying the code would need to share those modifications if distributed publicly.

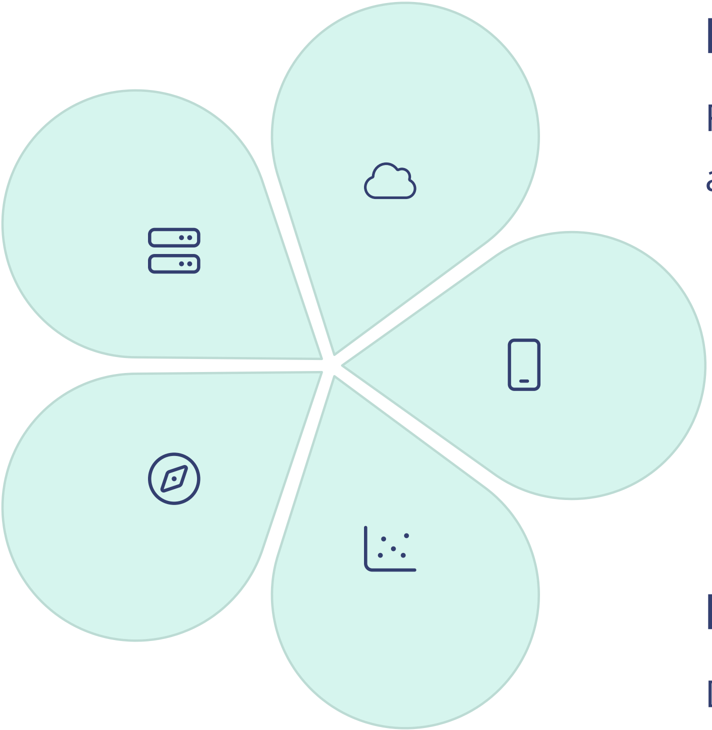
SSPL Introduction

In 2018, MongoDB shifted to the Server Side Public License (SSPL), designed specifically to prevent cloud providers from offering MongoDB as a service without contributing back to the community.

Current Model

MongoDB offers both a free community edition as well as an enterprise version with additional features, enhanced security, monitoring, and support for on-premises deployments

4. Product Portfolio



Core Database

Community & Enterprise Editions

MongoDB Atlas

Fully-managed cloud database service across major cloud providers

MongoDB Realm

Mobile database and synchronization platform

MongoDB Charts

Data visualization tools

MongoDB Compass

GUI for exploring and managing data

5. IT Architecture

Application Layer

Client applications using MongoDB drivers

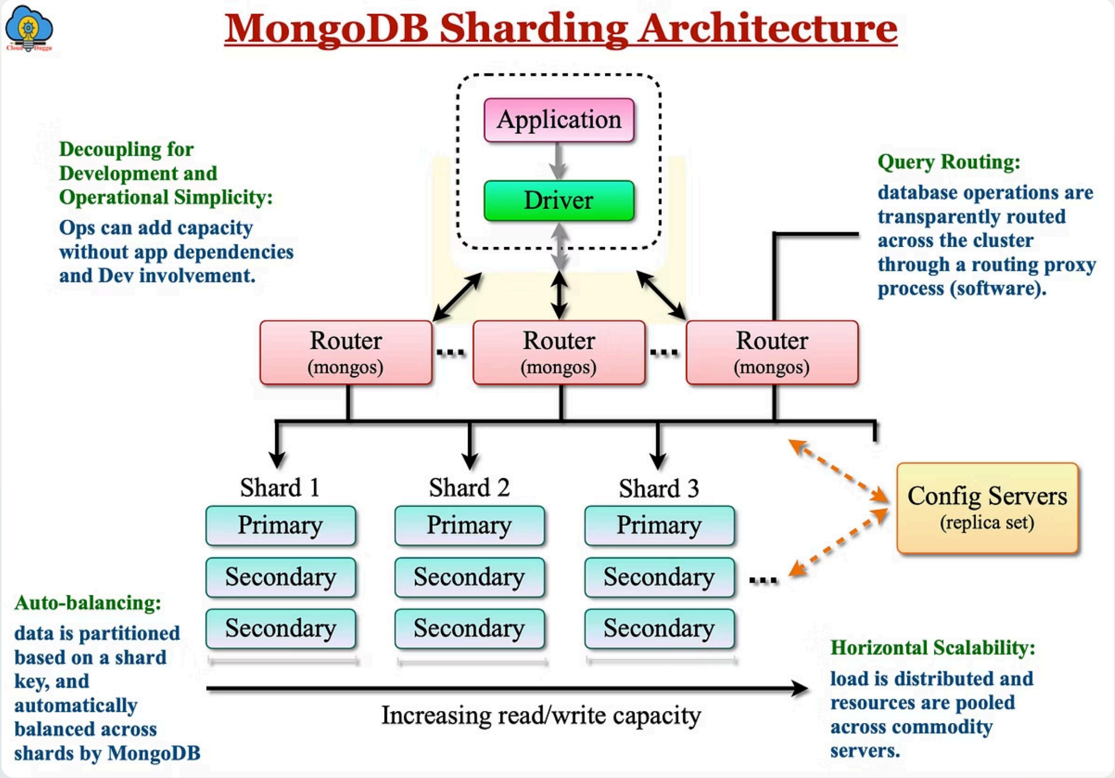
Database Layer

Distributed architecture with automatic sharding

Storage Layer

WiredTiger storage engine (replaced memory-mapped storage)

MongoDB uses replica sets for high availability. Primary-secondary replication ensures data redundancy.



MongoDB Sharding Architecture

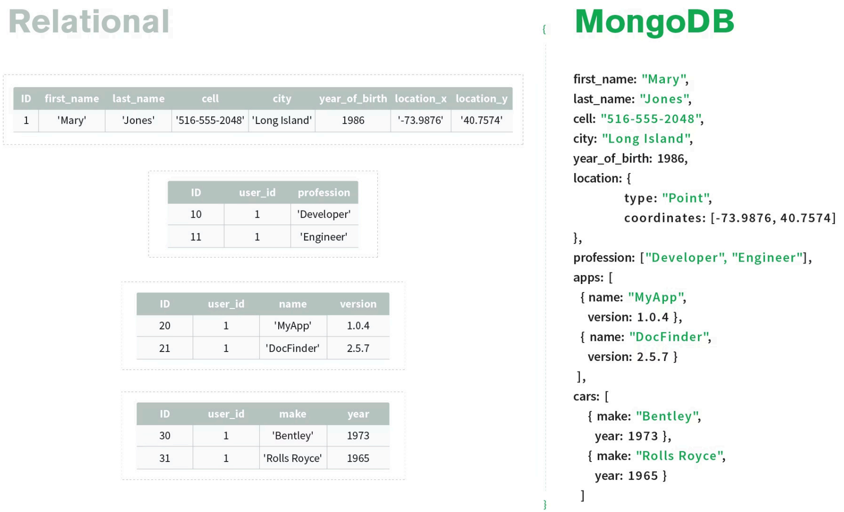
Decoupling for Development and Operational Simplicity: Ops can add capacity without app dependencies and Dev involvement.

Query Routing: database operations are transparently routed across the cluster through a routing proxy process (software).

Auto-balancing: data is partitioned based on a shard key, and automatically balanced across shards by MongoDB

Horizontal Scalability: load is distributed and resources are pooled across commodity servers.

6. The Data Models



Relational

ID	first_name	last_name	cell	city	year_of_birth	location_x	location_y
1	'Mary'	'Jones'	'516-555-2048'	'Long Island'	1986	'-73.9876'	'40.7574'

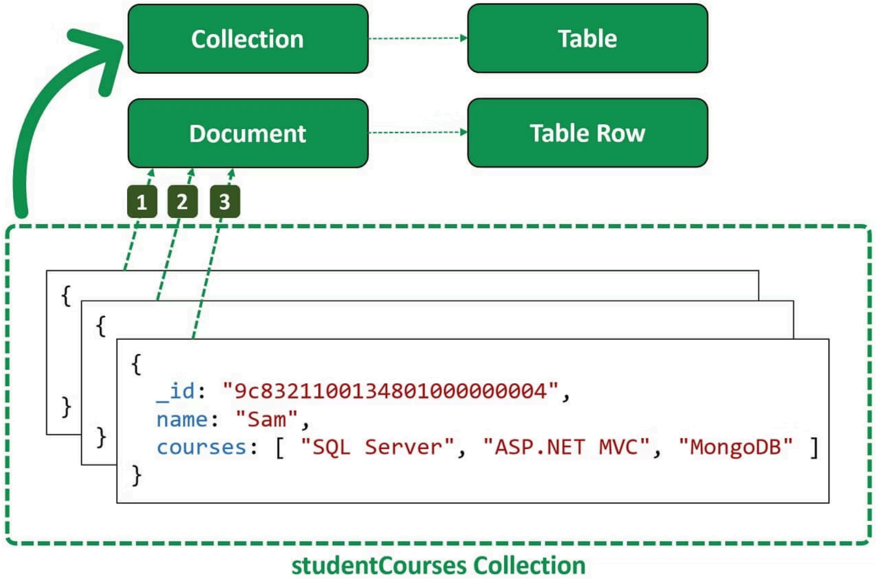
ID	user_id	profession
10	1	'Developer'
11	1	'Engineer'

ID	user_id	name	version
20	1	'MyApp'	1.0.4
21	1	'DocFinder'	2.5.7

ID	user_id	make	year
30	1	'Bentley'	1973
31	1	'Rolls Royce'	1965

MongoDB

```
{
    first_name: "Mary",
    last_name: "Jones",
    cell: "516-555-2048",
    city: "Long Island",
    year_of_birth: 1986,
    location: {
            type: "Point",
            coordinates: [-73.9876, 40.7574]
    },
    profession: ["Developer", "Engineer"],
    apps: [
     { name: "MyApp",
       version: 1.0.4 },
     { name: "DocFinder",
       version: 2.5.7 }
     ],
    cars: [
       { make: "Bentley",
         year: 1973 },
       { make: "Rolls Royce",
         year: 1965 }
      ]
}
```



```
{
  _id: "9c8321100134801000000004",
  name: "Sam",
  courses: [ "SQL Server", "ASP.NET MVC", "MongoDB" ]
}
```

JSON-Like Documents

MongoDB stores data in flexible BSON format (binary JSON), which supports a rich variety of data types and nested structures.

- No rigid schema requirements
- Natural mapping to object-oriented code
- Intuitive data representation

Schema Flexibility

Documents in the same collection can have different fields, allowing applications to evolve naturally over time.

- Easy schema evolution
- No migration downtime
- Support for polymorphic data

7. Client Libraries and Programming

Python

Popular for data science and web applications with frameworks like Django and Flask.

Node.js

Excellent for JavaScript-based full-stack development with Express.js and MEAN/MERN stacks.

Java

Enterprise-grade driver with strong Spring Boot integration for robust backend systems.

C#/.NET

Deep integration with Microsoft ecosystem for building enterprise Windows applications.

Go

Lightweight, high-performance driver ideal for cloud-native and microservice architectures.

Ruby

Popular among web developers utilizing Rails, with a straightforward driver for MongoDB integration.

PHP

Commonly used in web development, with robust support for MongoDB in popular frameworks like Laravel.

and more ...

Official Drivers

Consistent API across all major programming languages

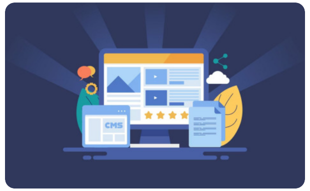
CRUD Operations

Simple, intuitive interface for database operations

Aggregation Framework

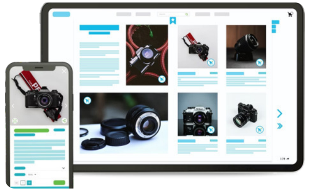
Powerful data processing and analysis capabilities

8. Real-World Applications



Content Management

MongoDB supports flexible content structures, making it ideal for dynamic content management systems that require rapid iteration and rich media support.



Catalogs

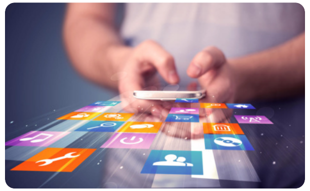
The document model efficiently handles diverse product catalogs with complex attributes and frequent updates.

Personalization

Document model excels at storing user profiles and preferences for customized experiences.

Real-Time Analytics

MongoDB's aggregation framework and time series collections enable powerful real-time dashboards and analytics applications.



Mobile Applications

The Realm SDK and synchronization capabilities make MongoDB ideal for offline-first mobile apps with seamless cloud sync.

Internet of Things

Flexible schema and time series collections handle diverse sensor data and high-volume telemetry ingestion.

9. Installation Options

Community Edition

- These documents offer guidance on installing MongoDB Community Edition, which is maintained by MongoDB.

Install on Linux

Install on macOS

Install on Windows

Install on Docker

- A community-managed Docker version of MongoDB is also available (**mongo - Official Image | Docker Hub**

Enterprise

MongoDB Enterprise is available for MongoDB Enterprise subscribers and includes several additional features including LDAP authentication, Kerberos authentication, and System Event Auditing.

Install on Linux

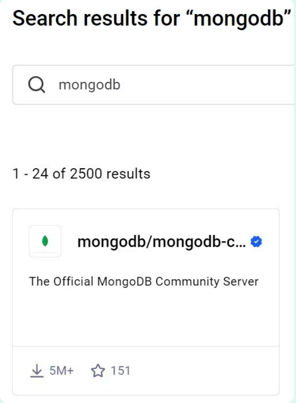
Install on macOS

Install on Windows

Install with Docker

9.1. Example: Easy installation and configuration with Docker Desktop (GUI)

1. Search for "MongoDB Community Server" on Docker Hub and click on the result.



Search results for “mongodb”

mongodb

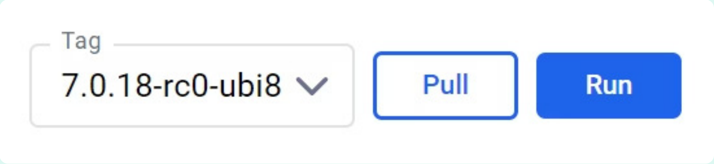
1 - 24 of 2500 results

mongodb/mongodb-c...

The Official MongoDB Community Server

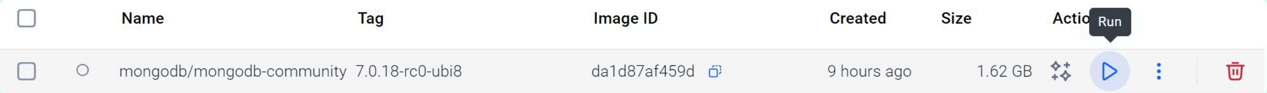
5M+ 151

2. On the top right side, select and pull the desired version.



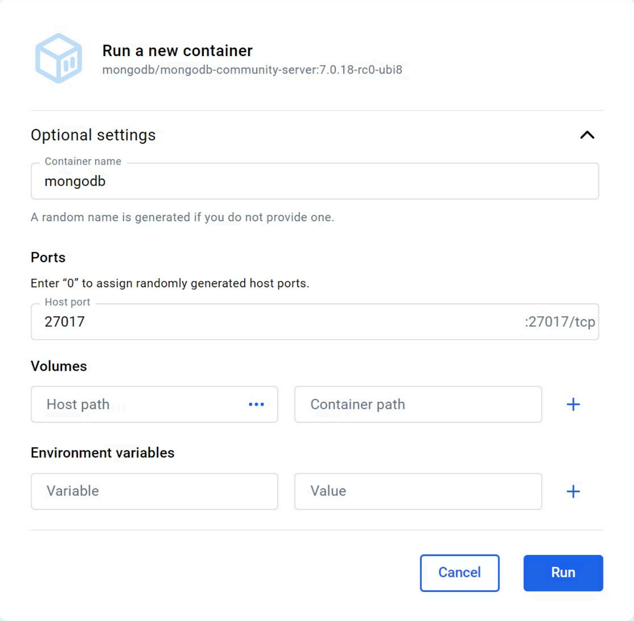
Tag: 7.0.18-rc0-ubi8

3. Locate the downloaded image and execute it to start running it.



□		Name	Tag	Image ID	Created	Size	Actio
□	○	mongodb/mongodb-community	7.0.18-rc0-ubi8	da1d87af459d	9 hours ago	1.62 GB	

4. Customize the container settings by modifying the fields provided below.



Run a new container

mongodb/mongodb-community-server:7.0.18-rc0-ubi8

Optional settings

Container name: mongodb

A random name is generated if you do not provide one.

Ports

Enter “0” to assign randomly generated host ports.

Host port: 27017 :27017/tcp

Volumes

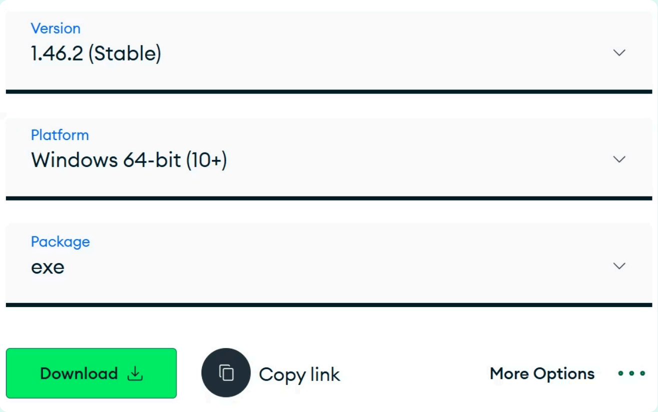
Host path | Container path

Environment variables

Variable | Value

9.2. Example: Easily explore and manipulate database with MongoDB Compass (GUI)

1. Download MongoDB Compass **here**

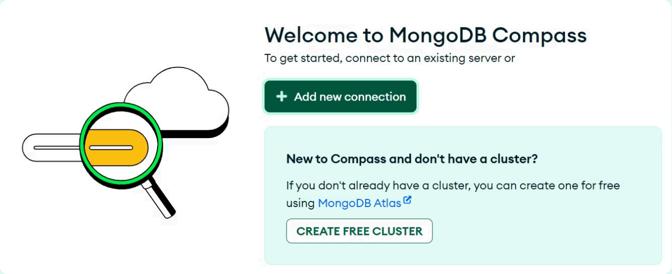


Version
1.46.2 (Stable)

Platform
Windows 64-bit (10+)

Package
exe

2. Click "Add New Connection" to connect to the downloaded MongoDB Community Edition instance



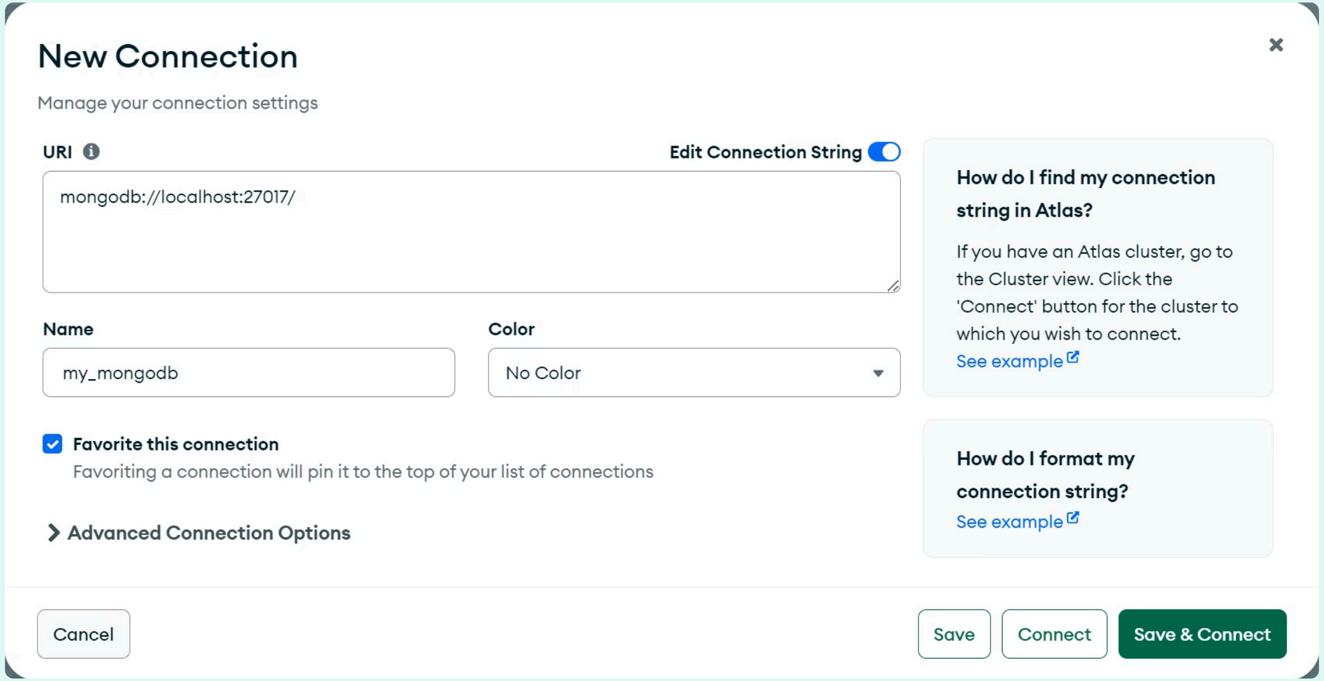
Welcome to MongoDB Compass

To get started, connect to an existing server or

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas

3. Modify the fields provided to complete the configuration, then click "Save & Connect."



New Connection

Manage your connection settings

URI

Edit Connection String

mongodb://localhost:27017/

Name
my_mongodb

Color
No Color

☑ Favorite this connection

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

> Advanced Connection Options

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.

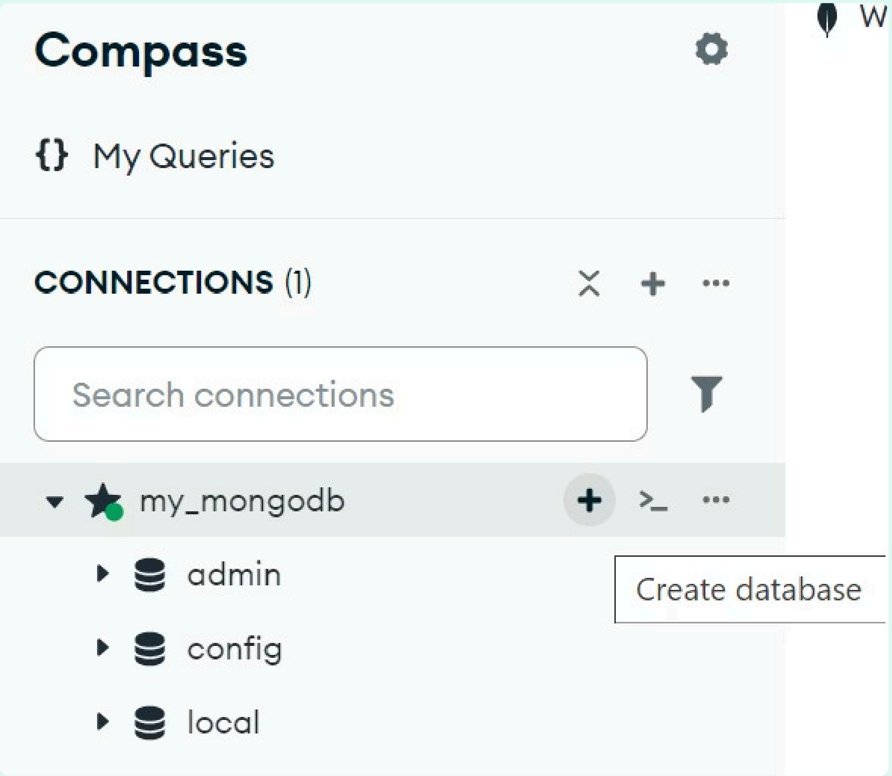
See example

How do I format my connection string?

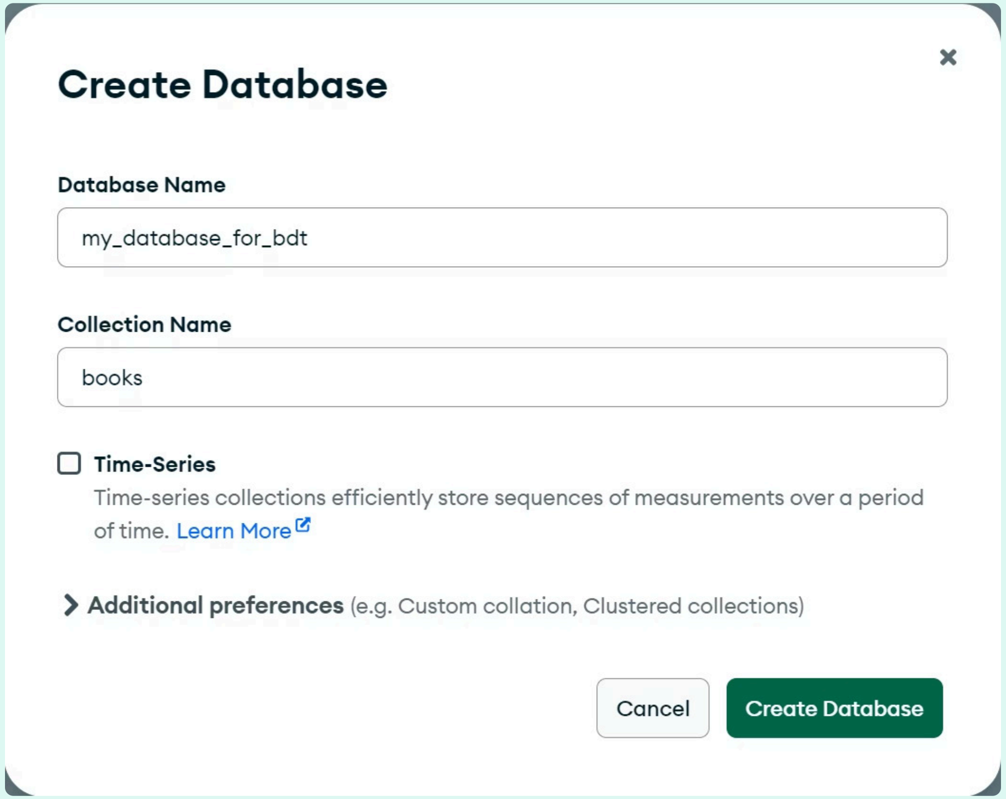
See example

9.3. Example: Create a database and a collection with MongoDB Compass (GUI)

4. The connection will appear under "Connections." Click the plus symbol next to your connection to create a new database.



5. Enter the database and collection name. Then click "Create Database" to finalize the setup.



Create Database

Database Name: my_database_for_bdt

Collection Name: books

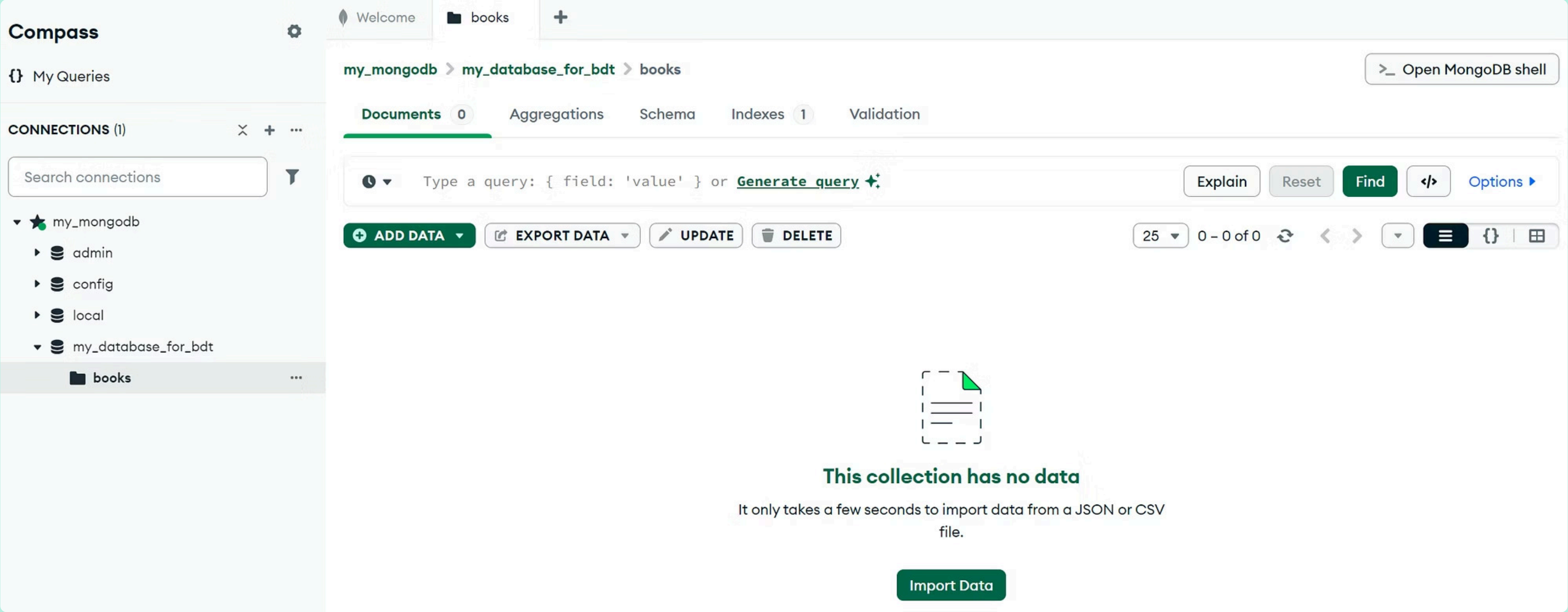
☐ Time-Series

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

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

10. Manage the database

The database and collections can be managed directly within MongoDB Compass, and they can also be accessed directly in your programming environment.



This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

For programming examples, please refer to the Jupyter Notebook that has been provided.