

MongoDB



En esta guía vamos a instalar y poner en funcionamiento una base de datos de MongoDB.

Primero procedemos a instalar Mongo accediendo a alguno de los siguientes enlaces y siguiendo los pasos indicados, dependiendo de tu sistema operativo deberás seguir distintos pasos:

- MongoDB Windows:

➔ [Instalación](#)

1 Download the installer.

Download the MongoDB Community `.msi` installer from the following link:

➤ [MongoDB Download Center](#)

- In the **Version** dropdown, select the version of MongoDB to download.
- In the **Platform** dropdown, select **Windows**.
- In the **Package** dropdown, select **msi**.
- Click **Download**.

- MongoDB Mac:

➔ [Instalación de Brew \(Permite instalar paquetes\)](#)

➔ [Instalación de Mongo](#)

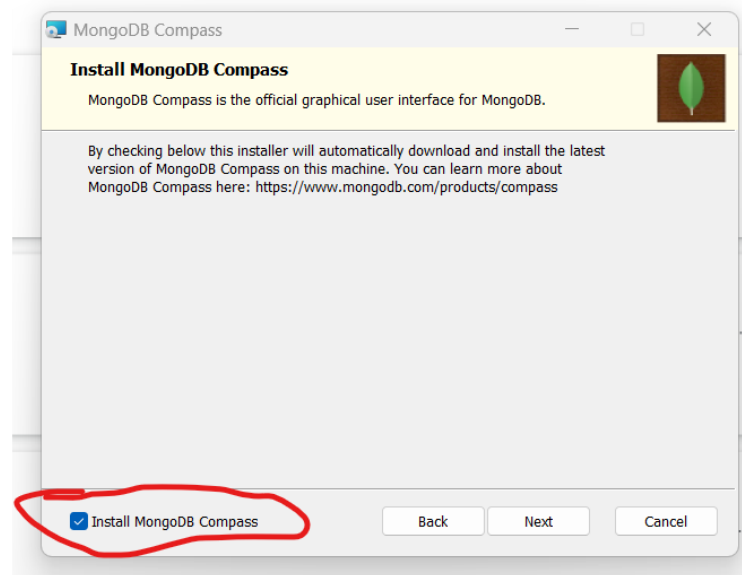
- MongoDB Ubuntu:

➔ [Instalación](#) (18.04)

➔ [Instalación](#) (20.04)

➔ [Instalación](#) (22.04)

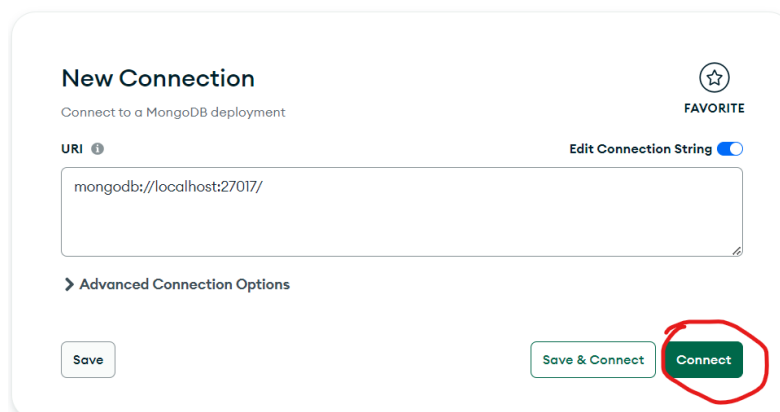
En el caso de Windows y Mac, MongoDB Compass, interfaz gráfica para gestionar bases de datos de MongoDB se instala automáticamente, si se queda trabado a la hora de instalar MongoDB deberás desmarcar la opción de instalar MongoDB Compass automáticamente e instalarlo después de instalar Mongo.



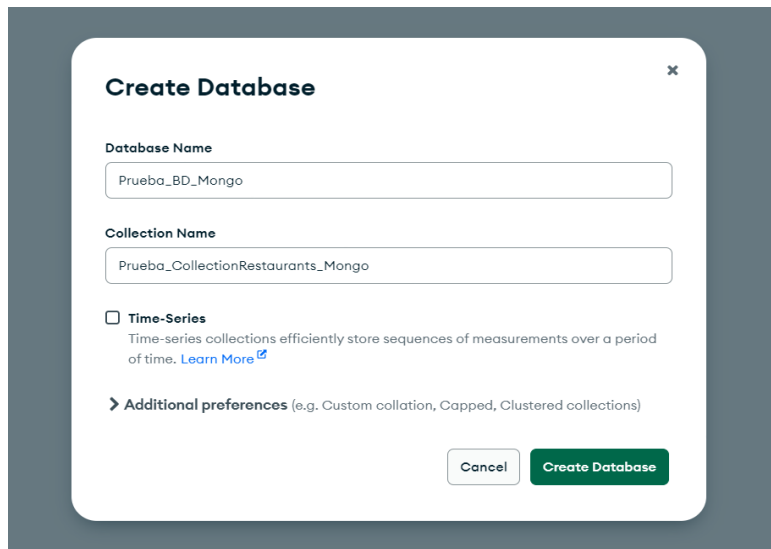
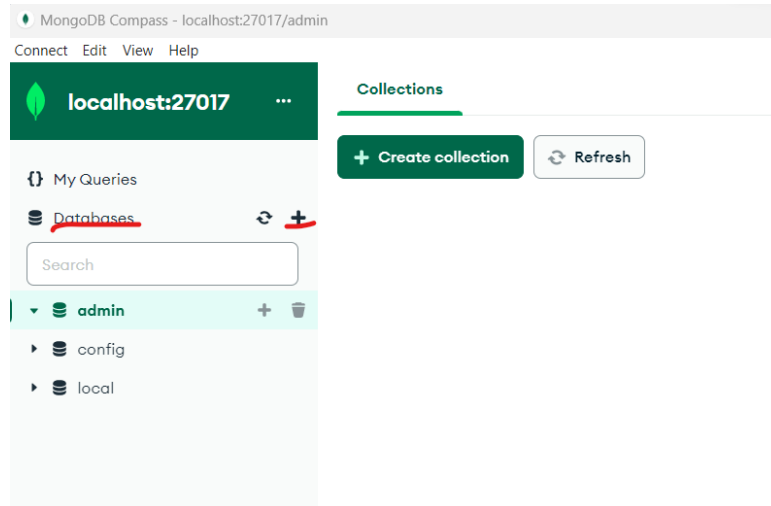
Para instalar MongoDB Compass aparte, [aquí](#).

MongoDB Compass incluye la posibilidad de utilizar el Shell de mongo (Mongosh), si quieres utilizar Mongosh fuera de Compass puedes instalarlo [aquí](#).

Segundo vamos a desplegar la base de datos de Mongo **en local**, simplemente abrimos MongoDB Compass y en el cuadrado de New Connection damos click en “Connect”.





En este caso nos estamos conectando a un servidor de MongoDB en local, podremos agregar datos no estructurados (.json por ejemplo) y ejecutar comandos de Mongo para filtrar o hacer queries. Tendremos por defecto admin, config y local, vamos a crear una base de datos nueva en el apartado de “Databases”, dando click en el símbolo “+”.




Luego podemos importar datos, en este caso importaremos la data de “restaurants.json” para probar.

Prueba_MongoDB.Prueba_Collection_Restaurants

[Documents](#) [Aggregations](#) [Schema](#) [Indexes](#) [Validation](#)

[Filter](#)   Type a query: { field: 'value' }



[+ ADD DATA](#) [EXPORT DATA](#)



Si todo va bien, deberíamos tener ya nuestra data cargada.

Prueba_MongoDB.Prueba_Collection_Restaurants

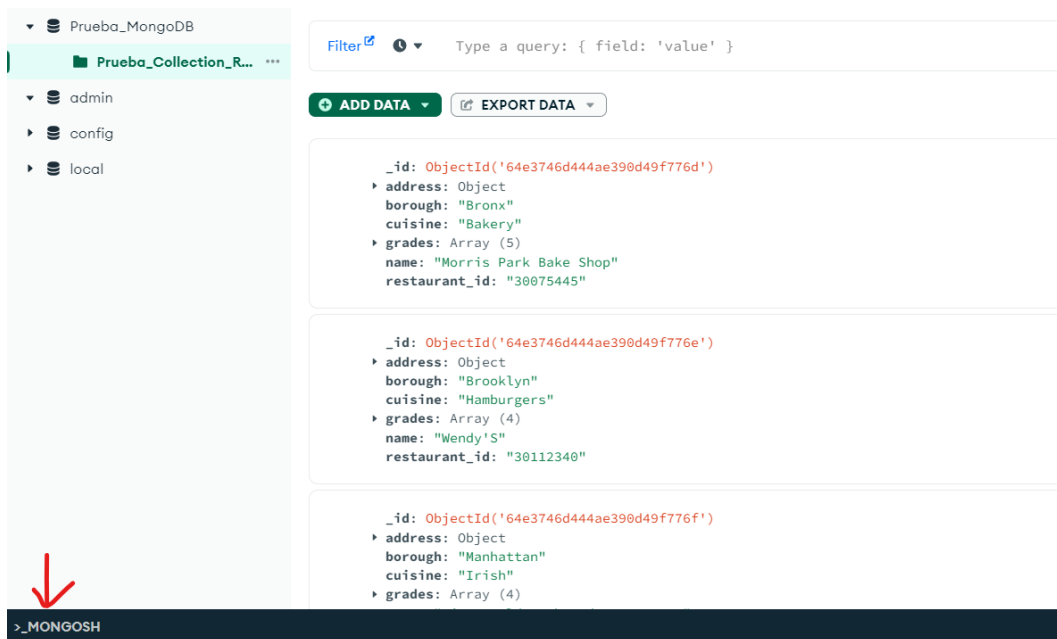
[Documents](#) [Aggregations](#) [Schema](#) [Indexes](#) [Validation](#)

[Filter](#)   Type a query: { field: 'value' }

[+ ADD DATA](#) [EXPORT DATA](#)

```
_id: ObjectId('64e3746d444ae390d49f776d')
address: Object
  borough: "Bronx"
  cuisine: "Bakery"
```

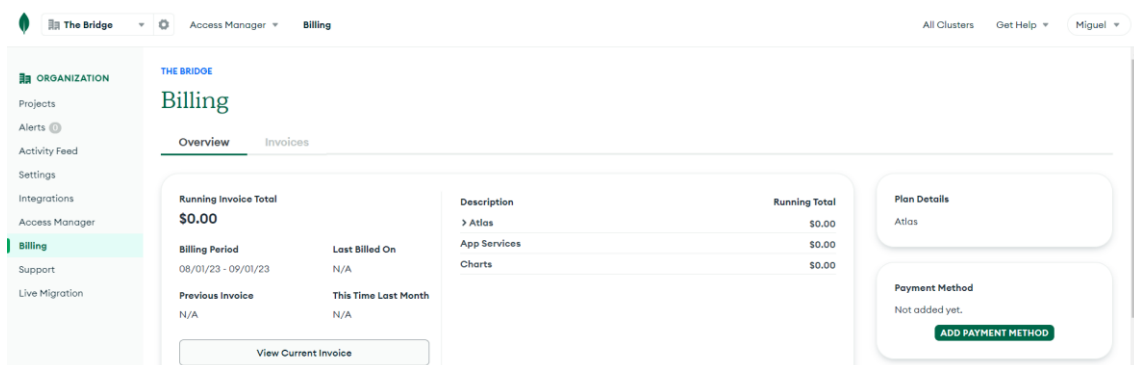
Ya tenemos una base de datos de Mongo creada, con una colección (homólogo de tabla en SQL) creada, con sus respectivos documentos (homólogo a registros en SQL), así que podremos hacer queries para obtener información, probamos desde la Shell de MongoDB Compass a hacer algunas queries. Accedemos a la Shell en la esquina inferior izquierda.



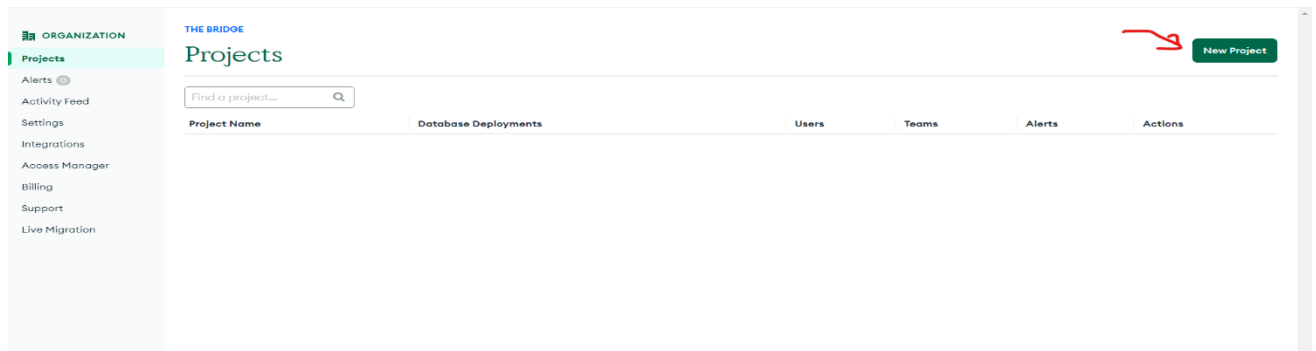
Utilizamos el comando “use Prueba_MongoDB” para acceder a la base de datos y ya podremos hacer queries dentro de la base de datos seleccionada.

ejemplo de query: “db.Prueba_Collection_Restaurants.find()” sin comillas, muestra todos los documentos de la colección seleccionada.

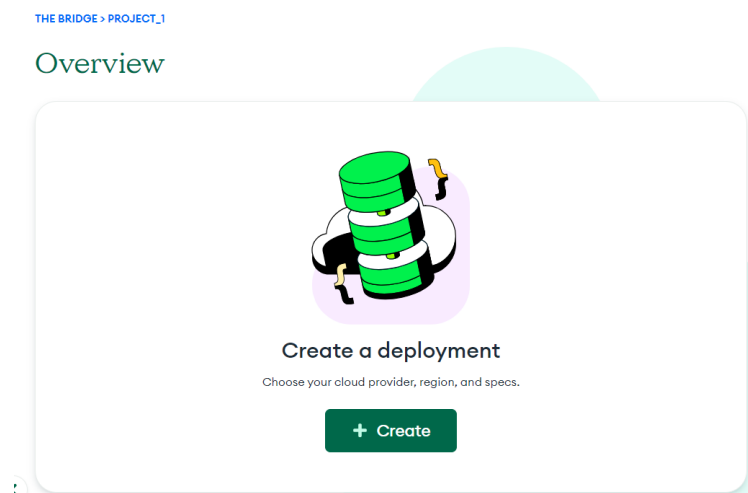
Tercero vamos a desplegar la base de datos de Mongo **en remoto** desde MongoDB Compass, para ello vamos a crear la base de datos en remoto utilizando MongoDB Atlas con una configuración sencilla y gratuita, para configuraciones más complejas consultar documentación. Para ello deberemos registrarnos en MongoDB Atlas.



Vamos al apartado de “Projects” y creamos un nuevo proyecto.



Creamos un nuevo despliegue en “Create a Deployment”.



Utilizamos la plantilla M0 (porque es gratis). Luego entramos a configuraciones “Security Quickstart”, configuraremos un usuario y password (1), además de agregar nuestra IP (2).

(1)

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password

Certificate

i We autogenerated a username and password for your first database user in this project using your MongoDB Cloud registration information. **x**

Create a database user using a username and password. Users will be given the *read and write to any database* privilege by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

Username

miguel

Password **👁**

1234

🔍 Autogenerate Secure Password

📋 Copy

Create User

(2)

✓ Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.

My Local Environment

Use this to add network IP addresses to the IP Access List. This can be modified at any time.

Cloud Environment

Use this to configure network access between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

ADVANCED

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

Enter IP Address

Description

Enter description

Add My Current IP Address



Luego podremos crear una base de datos.

THE BRIDGE > PROJECT_1

[← Back](#)

Add Data Options

Deployment: Cluster0 **▼**

Create Dataset

Load Sample Data

Migrate Database

Import File

[← Create Database on Atlas](#)

Database name

Enter a name for your new database

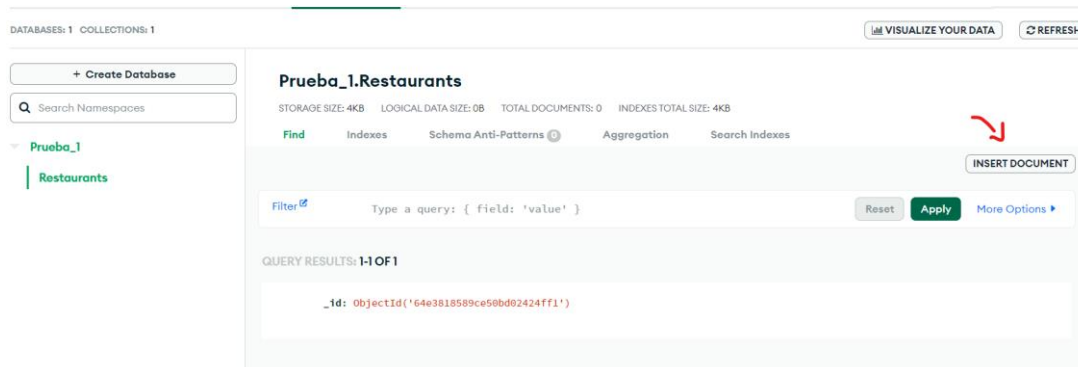
Prueba_1

Collection name

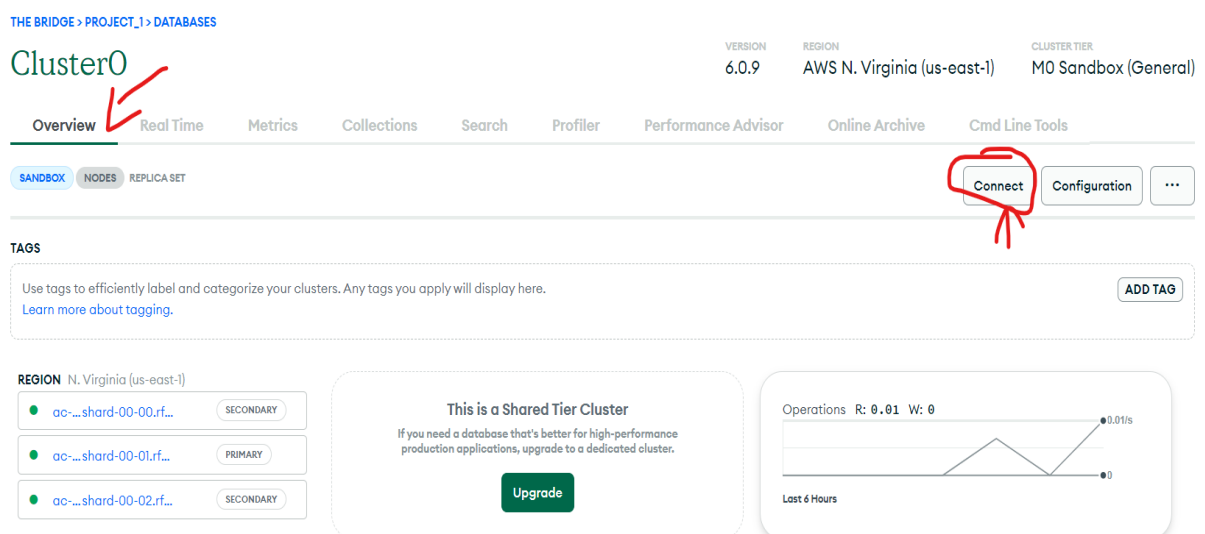
Enter a name for your new collection

Restaurants

En “Créate Dataset” colocamos nombre de la base de datos y la colección, lo demás lo dejamos por defecto. Luego insertamos un documento cualquiera para probar, podemos insertar el documento que crea por defecto. Las acciones de insertar documentos y manipular la base de datos las haremos normalmente desde Python.



Vamos a probar a conectarnos desde el local, vamos a la sección de “Overview” y hacemos click en “Connect”.




Nos aparecerán varias opciones de conexión como “Connect to your application” o “Access your data through tools”, la primera es para conectarnos a través de nuestro lenguaje de programación u otros, luego queda seguir los pasos indicados, por ahora vamos a usar la segunda para conectarnos con Compass y probar la base de datos desplegada.

Connect to Cluster0




Connect to your application


**Drivers**
Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)

>


Access your data through tools

**Compass**
Explore, modify, and visualize your data with MongoDB's GUI

>

**Shell**
Quickly add & update data using MongoDB's Javascript command-line interface

>

**MongoDB for VS Code**
Work with your data in MongoDB directly from your VS Code environment

>

Seguimos los pasos que nos indiquen, que será copiar un string y pegarlo en la zona de "New Connection" de Mongo Compass para conectarnos, sustituyendo <password> por la clave que hayamos creado anteriormente con nuestro usuario, sin los símbolos de ("<", ">").

Connect Edit View Help

Compass

New connection +

Saved connections



Recents

localhost:27017
21 ago 2023, 16:07

New Connection

Connect to a MongoDB deployment

FAVORITE

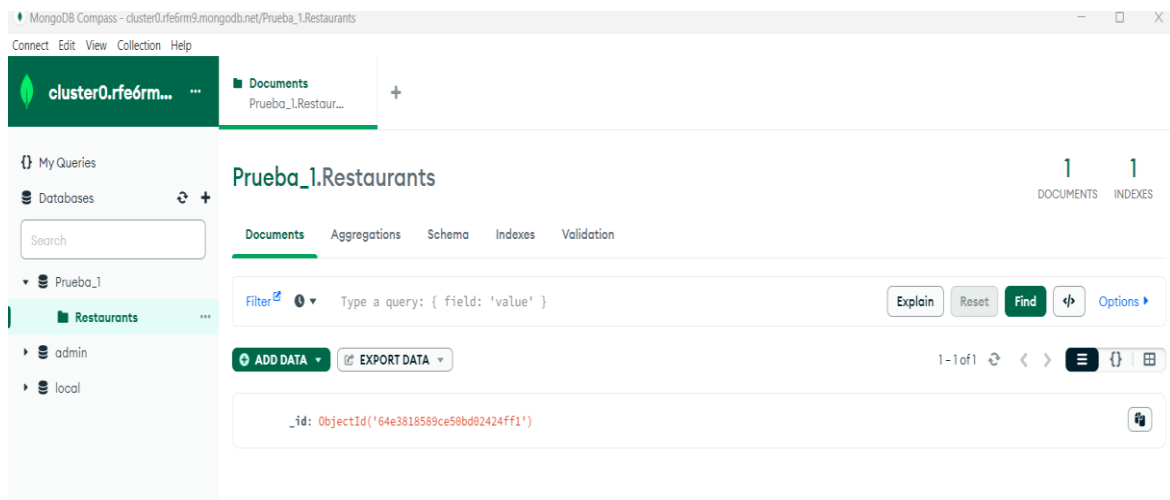
URI  Edit Connection String 

mongodb+srv://miguel:*****@cluster0.rfe6rm9.mongodb.net/

> Advanced Connection Options

Save Save & Connect Connect

Si todo está bien, podremos acceder a través de Compass a la base de datos creada y podremos hacer queries.



Comandos comunes de MongoDB:

- **use <nombre_de_la_base_de_datos>** -- Seleccionar una base de datos
- **db.usuarios.insert({nombre: "Ejemplo", edad: 30, correo: "ejemplo@example.com"})** -- Insertar documento
- **show dbs** -- Ver bases de datos
- **show collections** -- Ver colecciones
- **mongoexport --db nombre_de_la_base_de_datos --collection nombre_de_la_coleccion --out archivo_salida.json** -- Exportar a .json
- **db.<nombre_de_la_coleccion>.find()** -- Realiza una consulta para recuperar documentos de una colección.
- **db.<nombre_de_la_coleccion>.insertOne(<documento>):** Inserta un documento en una colección.
- **db.<nombre_de_la_coleccion>.updateOne(<filtro>, <actualizacion>)** -- Actualiza un documento en una colección.
- **db.<nombre_de_la_coleccion>.deleteOne(<filtro>)** -- Elimina un documento de una colección.
- **db.dropDatabase()** -- Borrar base de datos.