


Json Web Token - Parte 2



terminal

 Copiar código

```
// terminar, ver sesión 25
```

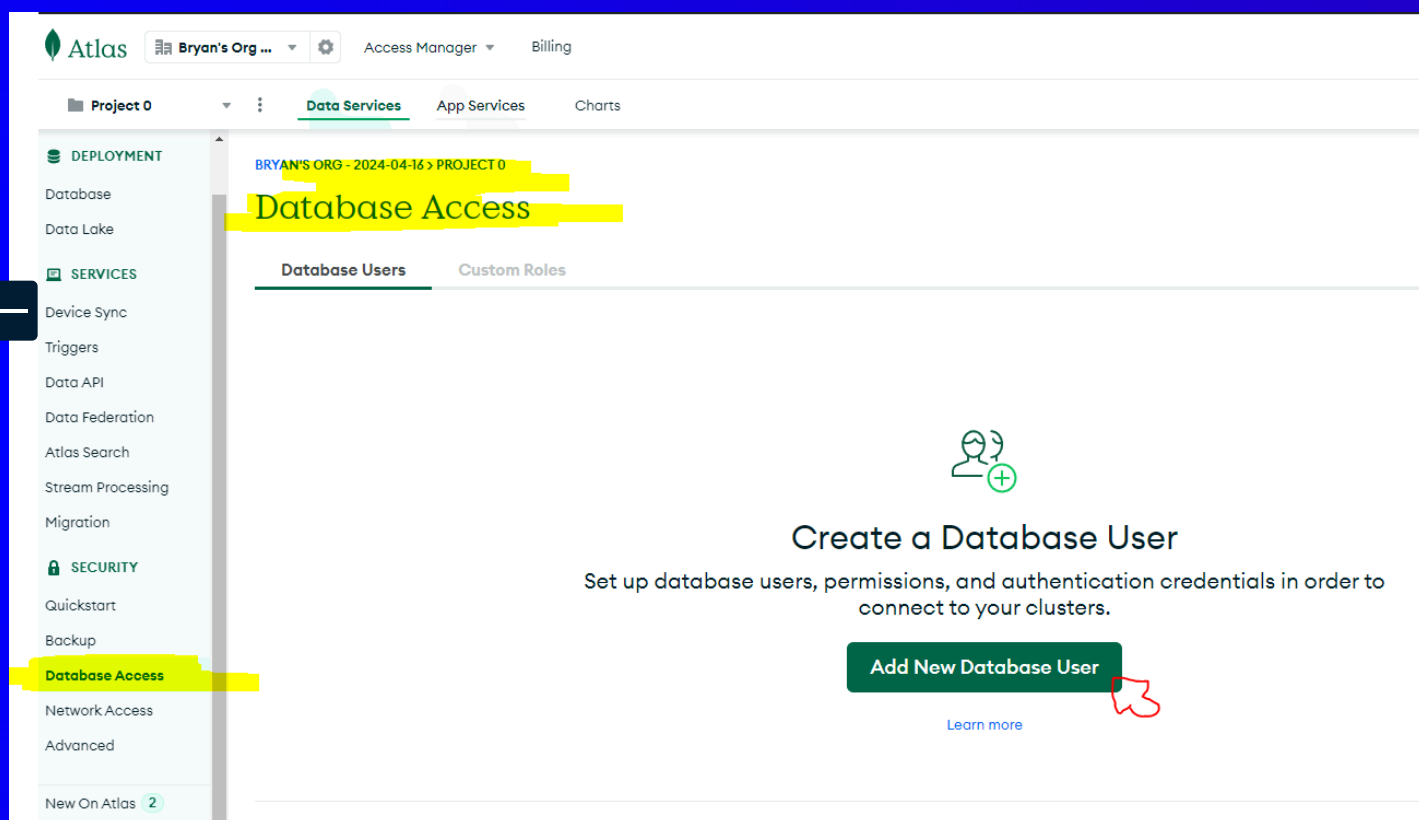
CRUD / Express y MongoDB - Parte 1

MongoDB logueo y configuración con Compass

Para este proyecto se trabajara con una base de datos no relacional, **MongoDb** y para la integración se usará **Express**

Lo primero sera la creación de la cuenta de [MongoDb desde su página](#)

Luego de crearse la cuenta, crea tu **usuario** y **contraseña**:



Add New Database User

13.04.2024 5:25 2

Create a database user to grant an application or user access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#)

Authentication Method

Password

Certificate

AWS IAM
(MongoDB 4.4 and up)

PREVIEW
Federated Auth
(MongoDB 7.0 and up)

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

user

.....

SHOW

Autogenerate Secure Password

Copy

Database User Privileges

Configure role based access control by assigning database user a mix of one built-in role, multiple custom roles, and multiple specific privileges. A user will gain access to all actions within the roles assigned to them, not just the actions those roles share in common. **You must choose at least one role or privilege.** [Learn more about roles.](#)

Built-in Role

Select one [built-in role](#) for this user.

0 SELECTED



Built-in RoleSelect one [built-in role](#) for this user.

1 SELECTED



Read and write to any database

Custom RolesSelect your [pre-defined custom role\(s\)](#). Create a custom role in the [Custom Roles](#) tab.**Specific Privileges**

Select multiple privileges and what database and collection they are associated with. Leaving collection blank will grant this role for all collections in the database.

**Restrict Access to Specific Clusters/Federated Database Instances/Stream Processing Instances**

Enable to specify the resources this user can access. By default, all resources in this project are accessible.

**Temporary User**

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week.



Cancel

Add User

Luego de crearse un usuario, verifica que el acceso a la red no tenga restricción alguna, la mejor forma para evitar problemas es la siguiente:

The screenshot shows the MongoDB Atlas interface for a project named "Bryan's Org". The left sidebar contains a navigation menu with sections: DEPLOYMENT, SERVICES, and SECURITY. Under SERVICES, "Network Access" is highlighted. The main content area shows the "Network Access" configuration for "PROJECT 0". It includes tabs for "IP Access List", "Peering", and "Private Endpoint". A yellow banner states: "You will only be able to connect to your cluster from the following list of IP Addresses:". Below this is a table with the following data:

IP Address	Comment	Status	Actions
179.33.67.42/32 (includes your current IP address)	Created as part of the Auto Setup process	Active	EDIT DELETE

At the bottom, the system status is "All Good" and the footer includes copyright information for MongoDB, Inc. and links to Status, Terms, Privacy, Atlas Blog, and Contact Sales.

Edit IP Access List Entry

Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. [Learn more.](#)

ALLOW ACCESS FROM ANYWHERE

Access List Entry:

0.0.0.0/0

Comment:

ok

Cancel

Confirm

Ahora genera una nueva conexión usando **Compass**.

The screenshot shows the MongoDB Atlas 'Database Deployments' page. The left sidebar contains navigation links for Overview, DEPLOYMENT, Database, Data Lake, SERVICES, Device Sync, Triggers, Data API, Data Federation, Atlas Search, Stream Processing, Migration, SECURITY, Backup, Database Access, Network Access, and Advanced. The main content area shows a deployment for 'Cluster0' with a status message: 'Sample dataset successfully loaded. Access it in Collections or by connecting with the MongoDB Shell.' Below this, there are buttons for 'Connect', 'View Monitoring', and 'Browse Collections'. A red arrow points to the 'Connect' button. The 'Connect' button is also highlighted with a yellow box. Below the deployment details, there are charts for 'Visualize Your Data' showing read/write rates, connections, and data size. At the bottom, a table lists deployment details like version, region, cluster tier, type, backups, and linked app services.

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SQL	ATLAS SEARCH
7.0.8	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	Unable to load application data	Connect	Create Index

15.04.2024 8:55:12



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



Data Explorer

Browse your Atlas collections without leaving the UI



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



Atlas SQL

Connect to Cluster0



Connecting with MongoDB Compass

I don't have MongoDB Compass installed

I have MongoDB Compass installed

1. Select your operating system and download MongoDB Compass

Windows 64-bit (10+)

Download Compass (1.42.5) or Copy download URL

Compass is an interactive tool for querying, optimizing, and analyzing your MongoDB data.

2. Copy the connection string, then open MongoDB Compass

mongodb+srv://user:<password>@cluster0.jlljz kf.mongodb.net/

Replace **<password>** with the password for the **user** user. Ensure any options are [URL encoded](#).

RESOURCES

[Connect with Compass](#)

[Access your Database Users](#)

[Import and Export Data](#)

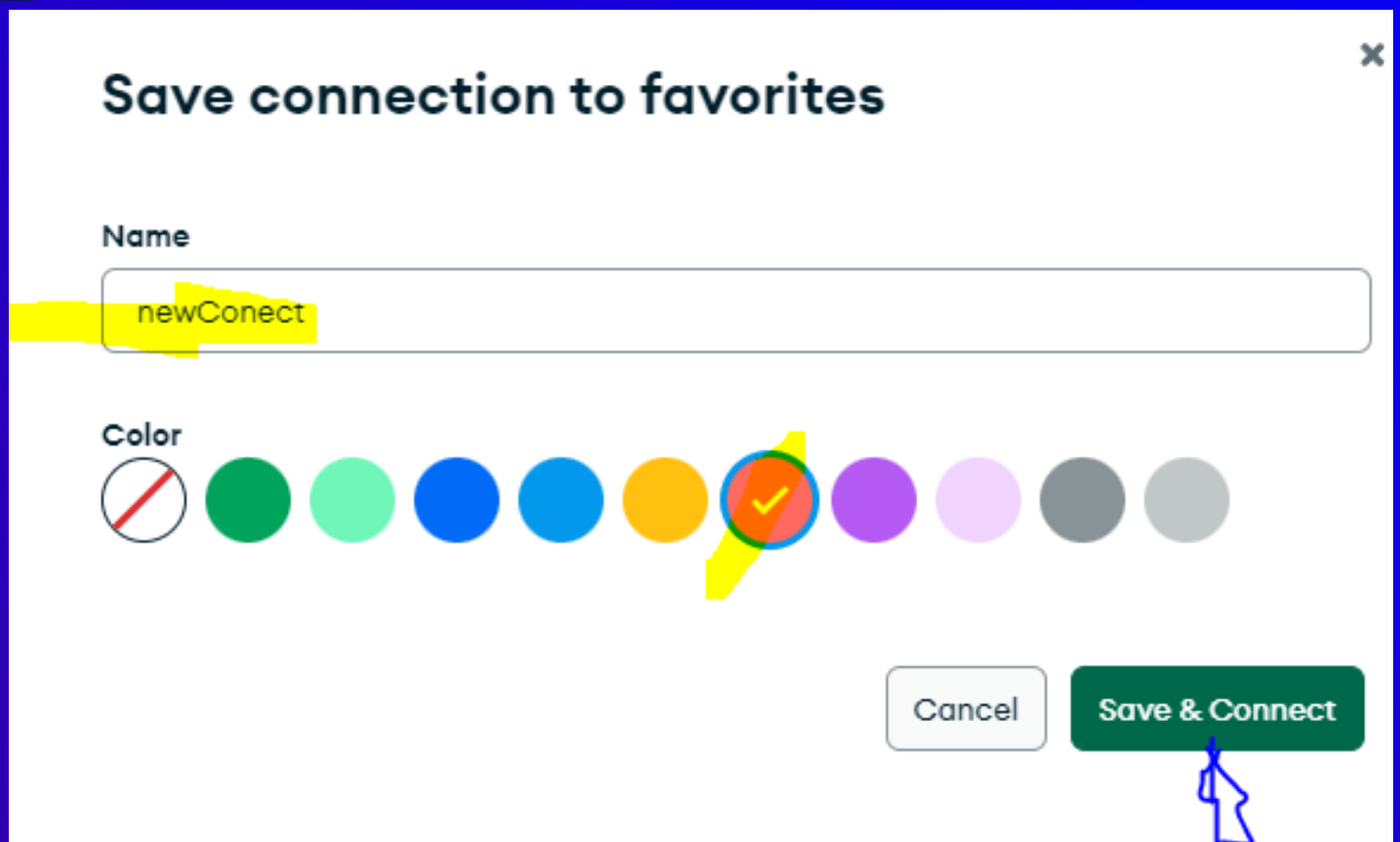
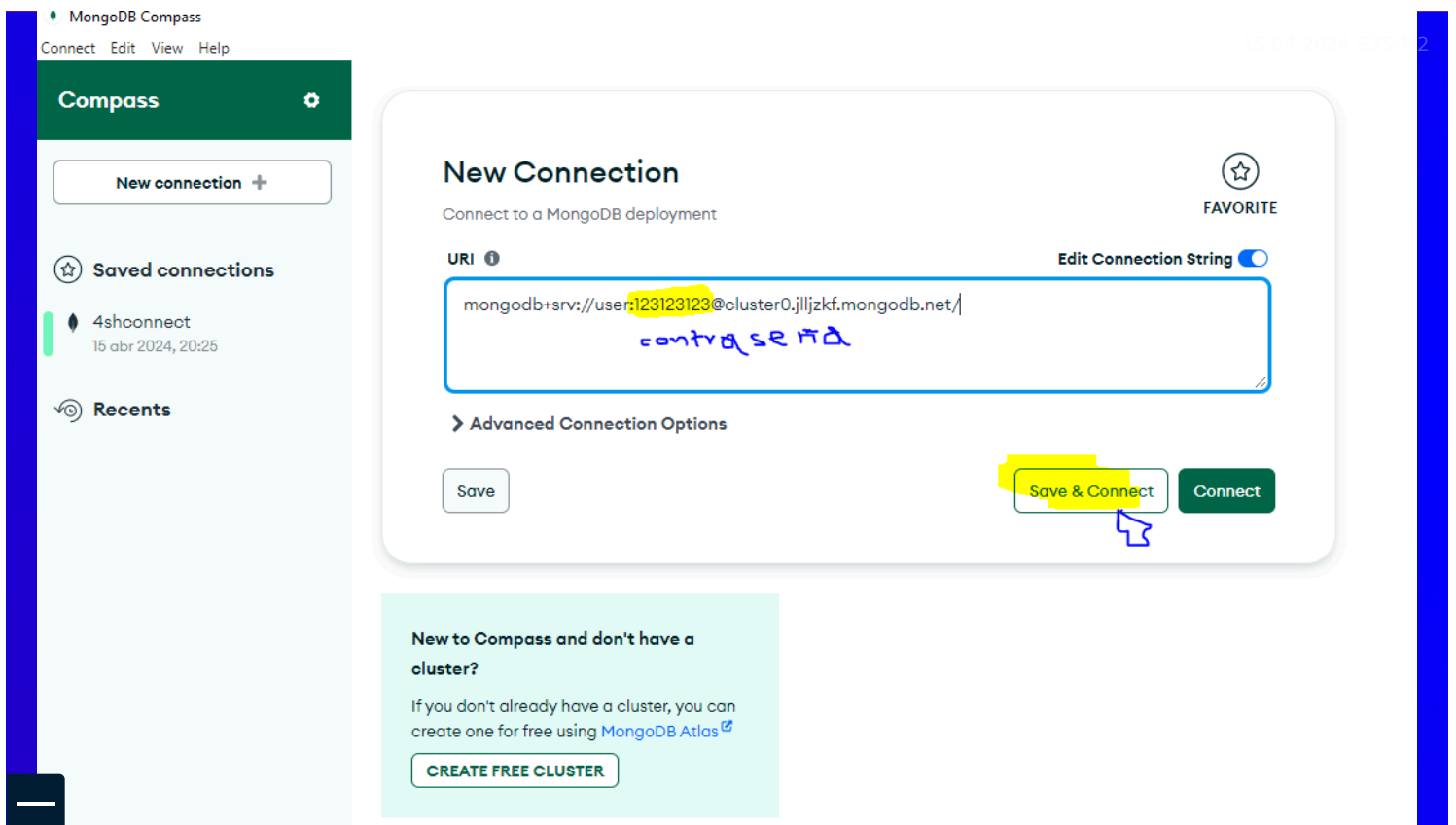
[Troubleshoot Connections](#)

Go Back

Close

Review setup steps

Ahora para finalizar **abre Compass** y crea la nueva conexión, pegando el código que acabas de copiar




Para finalizar con esta parte crea una nueva base de datos

Creación de proyecto e integración con Express

Primero vamos a estructurar nuestro proyecto, crea las siguientes carpetas:


estructura de carpetas

 Copiar código

```
// Raíz Proyecto
Proyecto
  backend-mongo
```

Ahora inicializamos nuestro proyecto backend node.js con express, y algunas otras dependencias en el directorio **backend-mongo**:


terminal

 Copiar código

```
npm i -y // Inicializamos proyecto node.js con parámetros por default
npm i express
npm i mongoose dotenv cors
npm i -D nodemon // Trae los datos del servidor y los actualiza en tiempo real
```

Añadimos al proyecto las siguientes carpetas y archivos:


terminal

 Copiar código

```
→ config // New
→ controllers // New
→ models // New
→ node_modules
→ routes // New
↓ src // New
  index.js // New
.env // New
package-lock.json
package.json
```

Por el momento nuestro **index.js** presenta la siguiente configuración:

src/index.js

 Copiar código

```
// Aquí iniciaremos express
const express = require('express');


const app = express();
const port = 5000;

app.listen(port, () => console.log('Nuestro servidor se encuentra conectado 😊 http://localhost:5000'));

app.get('/', (req, res) =>{
  res.send('Bienvenido, nuestro servidor esta configurado');
});
```

A nuestro **package.json** se lo ha modificado para que inicie nuestra app desde *src/index.js*


package.json

 Copiar código

```
{
  "name": "backend-mongo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "nodemon src/index.js" // Modificado
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^16.4.5",
    "express": "^4.19.2",
    "mongoose": "^8.3.1"
  },
  "devDependencies": {
    "nodemon": "^3.1.0"
  }
}
```

Por ahora en nuestras variables de entorno tenemos la uri de conexión con MongoDB

`.env`

 Copiar código


```
MONGO_URL = mongodb+srv://4USER:4MIn4C_.ON@cluster0.4bxgl6w.mongodb.net/apiclientes
```

NOTA

'**apiclientes**' es el nombre de la base de datos a la cual nos vamos a conectar, la cual creaste por **Compass**

Finalmente para terminar con esta parte y ejecutar la app usa el siguiente código:

`Ejecutar app`

 Copiar código

```
npm run start
```