# 

# **Configurando Node.js y MongoDB Atlas**

Una guía rápida para iniciar un entorno con Node.js y MongoDB

Objetivo: Crear una aplicación Node.js simple después de configurar un entorno de desarrollo local, probar las API REST a través de Postman y establecer una conexión de base de datos a MongoDB Atlas.

Requisitos:

* Visual Studio Code: <https://code.visualstudio.com/>
* Git: <https://git-scm.com/downloads>
* Node.js: <https://nodejs.org/en/>
* npm: Normalmente incluido en la instalación de node
* Postman: <https://www.postman.com/>
* MongoDB Atlas: <https://www.mongodb.com/>

Si aún no los instalaste, sigue el enlace de cada uno y continúa con las instrucciones que indique para su instalación.

Revisa que node y npm estén instalados en su entorno:

node -v && npm -v

mostrándole algo similar a (Las versiones pueden cambiar):



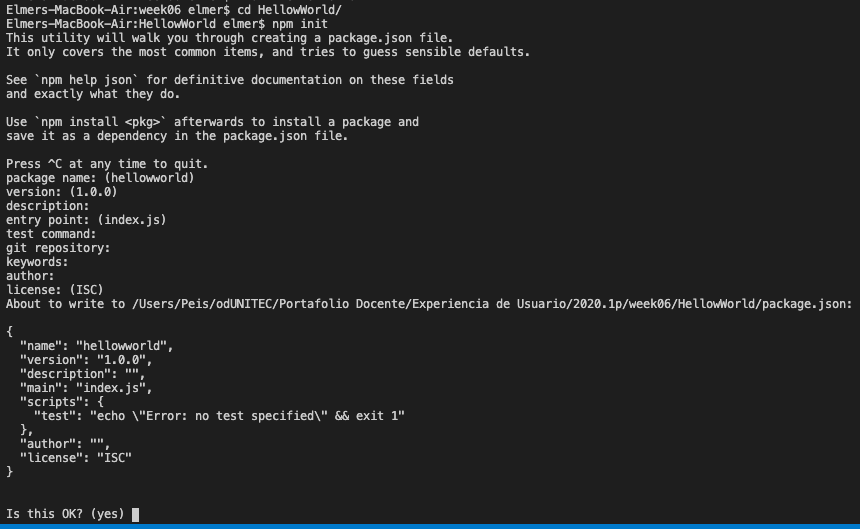
# **App Node.js HelloWorld**

Crea un directorio en algún lugar de su computadora para una aplicación HelloWorld:



Cambia a la carpeta e inicialice la aplicación Node.js:

npm init



Crea un archivo app.js

touch app.js

y agrega el siguiente código a dicho archivo app.js:

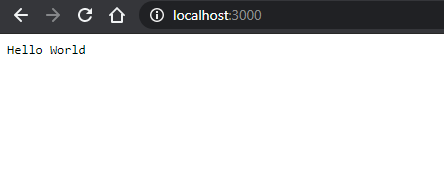
| const http = require("http");  const hostname = "127.0.0.1";  const port = 3000;  const server = http.createServer((req, res) => {  res.statusCode = 200;  res.setHeader("Content-Type", "text/plain");  res.end("Hello World\n");  });  server.listen(port, hostname, () => {  console.log(`Server running at http://${hostname}:${port}/`);  }); |
| --- |

Lo que estamos haciendo aquí es especificar el nombre de host y el puerto y luego creamos nuestro propio servidor HTTP. Adicionalmente estamos definiendo un servicio que al recibir una solicitud, si tiene éxito, veremos «Hello World» impreso en la pantalla.

Checkpoint: En su consola, ejecute el comando node app.js:



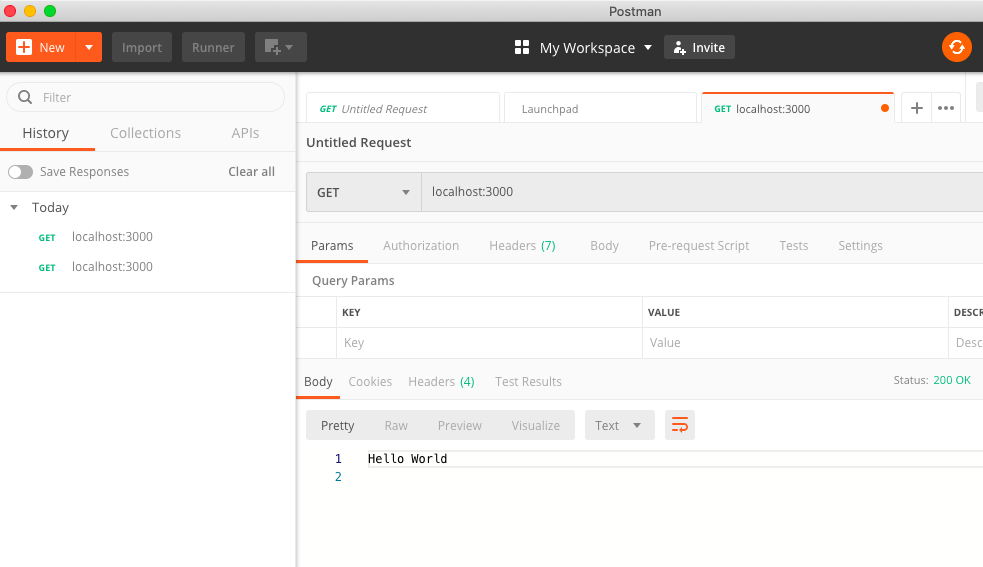
Abrí el navegador y andá a localhost:3000



## **Postman**

Para probar nuestra API, necesitaremos algún tipo de cliente. Podríamos enviar un comando curl a través de la línea de comando cada vez, pero eso se vuelve tedioso. Es bueno tener una GUI (interfaz gráfica de usuario) para simplificar este proceso.

A medida que nuestro proyecto crezca, estos comandos curl solo se volverán cada vez más complicados. Entonces, usaremos Postman. Vaya a <https://www.getpostman.com/downloads/> y seleccione la descarga para su sistema operativo (No necesita una cuenta para comenzar. Simplemente creá una nueva pestaña junto a «Launchpad»).

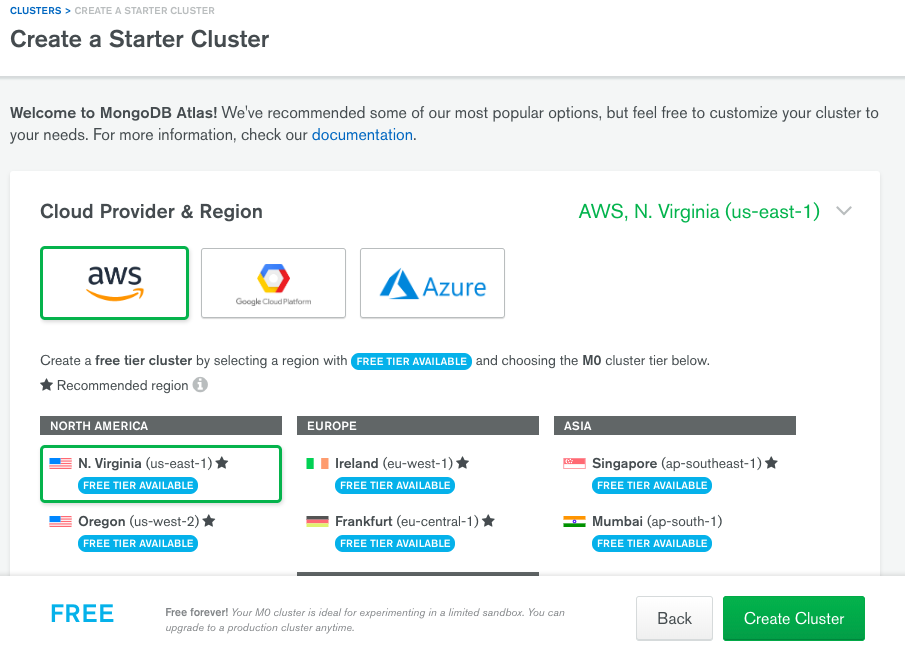


## **MongoDB Atlas**

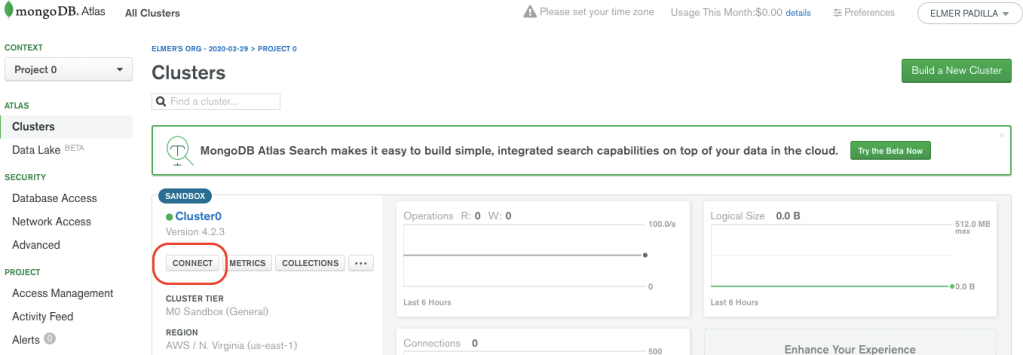
Para esta sección solo necesitarás crear una cuenta. Estaremos dentro del nivel gratuito de esta cuenta.

Hacé clic en el botón «Try Free» en <https://www.mongodb.com/> e ingrese algunos detalles de la cuenta. Luego sugiero «Starter Clusters» para mantenerlo libre.

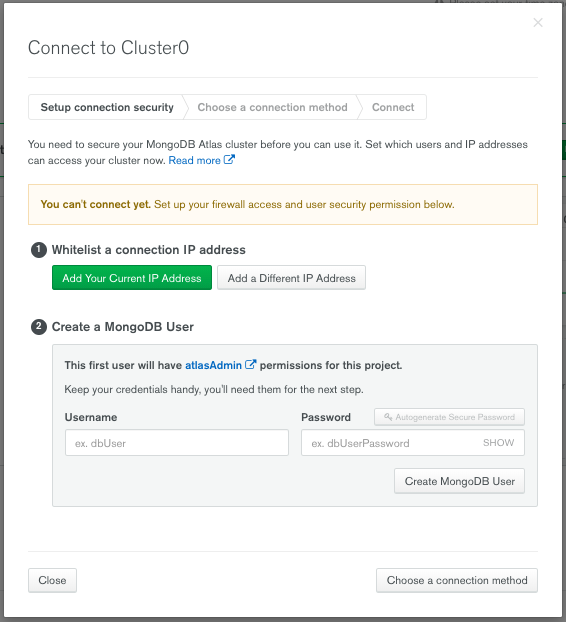
Entonces llegarás a esta página. Seleccione un proveedor y ubicación en la capa free.



Asegúrate de que cuando crea este clúster diga «FREE» en la parte inferior. Luego, dale a su nuevo clúster un par de minutos para que se cree.

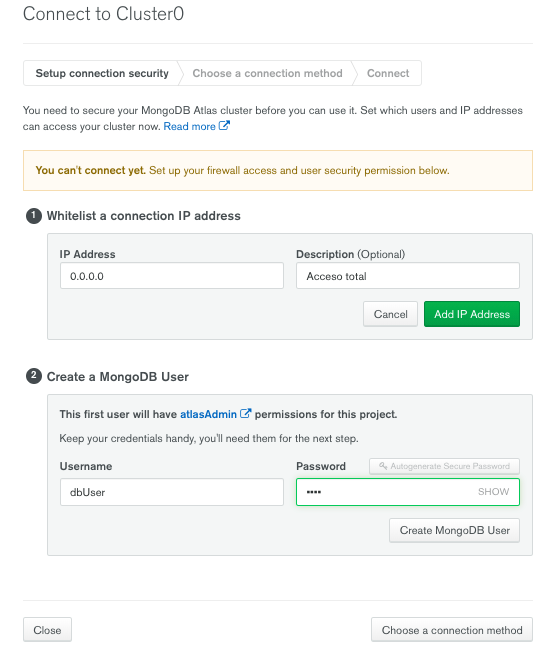


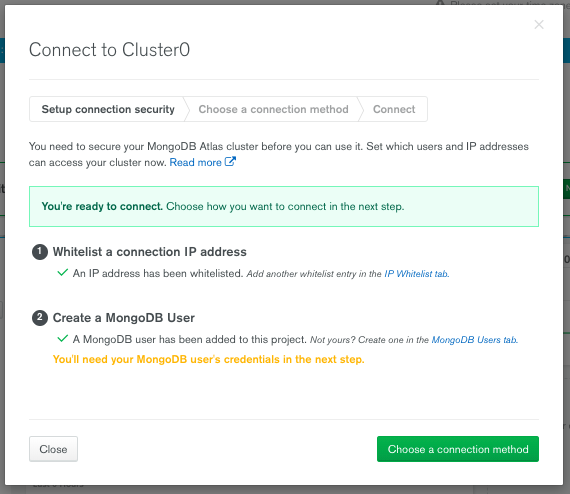
Haga clic en el botón «CONNECT» en Clusters.



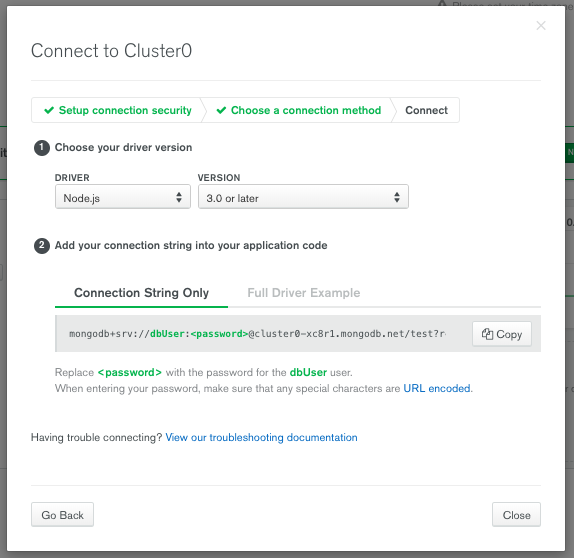
Luego ellegí “**Add Your Current IP Addres**s”. O puede agregar 0.0.0.0 para conectarse desde cualquier lugar (por cuestiones de seguridad no es recomendado).

También creemos nuestro usuario MongoDB. Esto se usará en el URI de conexión, por lo que sugiero, anotá el usuario y contraseña para usarlo luego.





Presione «Choose a connection method» y seleccioná «**Connect Your Application**»:



Copiá la cadena de conexión porque la usaremos mas adelante. También reemplazaremos <contraseña> con la contraseña que acabamos de crear.

De vuelta en nuestra aplicación Hello World, vamos a la línea de comando donde instalaremos el controlador MongoDB.

npm i --save mongodb

Ahora, agregá el siguiente código en app.js para que podamos hacer una conexión a nuestra base de datos.

| const http = require("http");  const hostname = "127.0.0.1";  const port = 3000;  const server = http.createServer((req, res) => {  res.statusCode = 200;  res.setHeader("Content-Type", "text/plain");  res.end("Hello World\n");  });  //Código nuevo  const MongoClient = require("mongodb").MongoClient;  const uri = "mongodb+srv://dbUser:123#@cluster0-xc8r1.mongodb.net/test?retryWrites=true&w=majority";  MongoClient.connect(uri, {useUnifiedTopology: true }, (err, client) => {  if (err) console.log("Error occurred connecting to MongoDB...");  console.log("Connected to MongoDB!");  });  //Fin código nuevo  server.listen(port, hostname, () => {  console.log(`Server running at http://${hostname}:${port}/`);  }); |
| --- |

Checkpoint: Al reiniciar su servidor de node (node app.js), debería ver la siguiente salida de la consola.



Hemos establecido con éxito una conexión a la base de datos. ¡Ahora estamos listos para construir una API REST usando Node.js y MongoDB!