# Steps to follow

Hey! Thanks for checking the source code. There are a few things you need to do first. If you skip these steps, you won't be able to see the site in action. Let's get started!

Node Version Used: **node v23.10.0 (npm v11.2.0)**

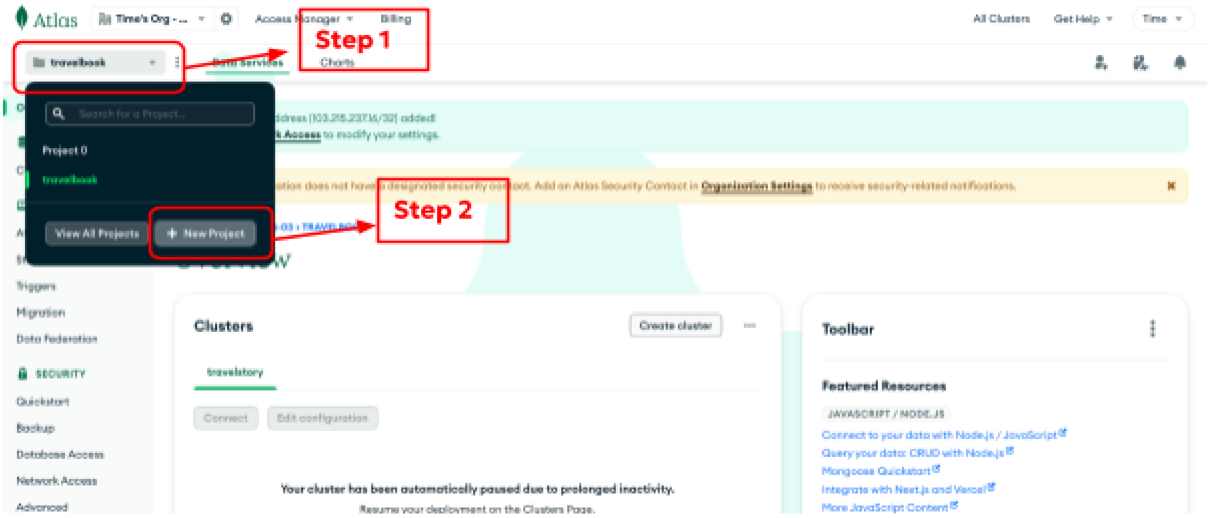
## Running the Task Manager App Project

Backend (Express.js)

1. Navigate to the `backend` folder in your terminal.
2. Run the following command to install the required dependencies:

```
npm install
```

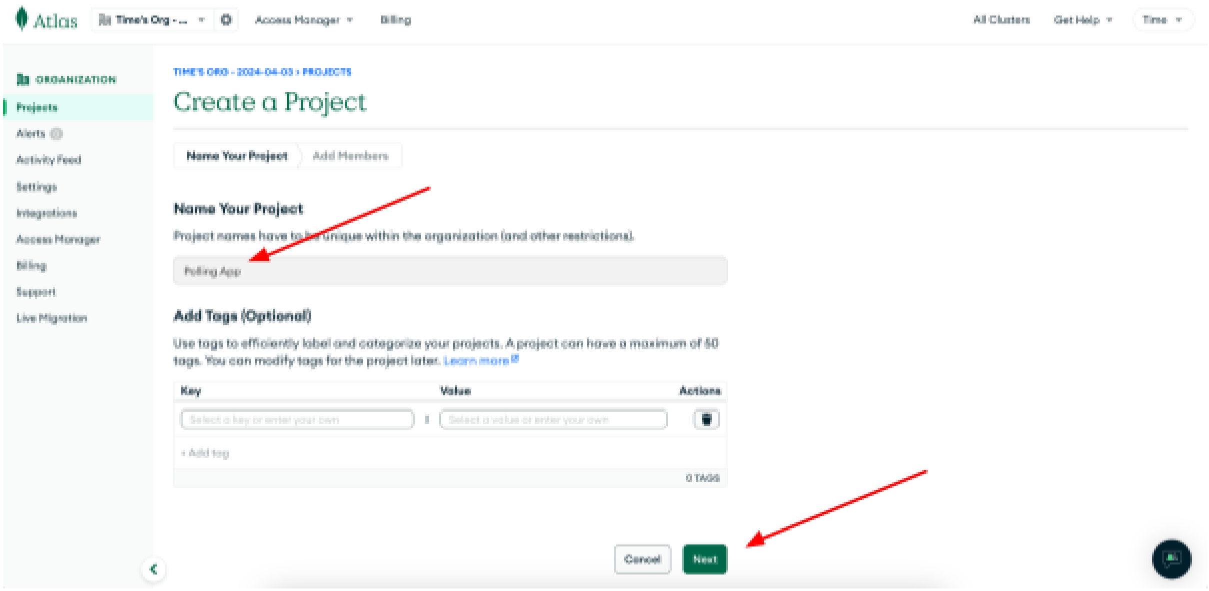
3. Once the dependencies are installed, Let's connect MongoDB
4. Go to [https://www.mongodb.com/](https://www.mongodb.com/)
5. Login or Create an Account
6. Now let's create a project by clicking on the “New Project” button



Your cluster has been automatically paused due to prolonged inactivity.

Resume your deployment on the Clusters Page.

7. Now, Enter the project name and Click “Next”



TIME'S ORG - 2024-04-03 › PROJECTS

Create a Project

Name Your Project

Project names have to be unique within the organization (and other restrictions).

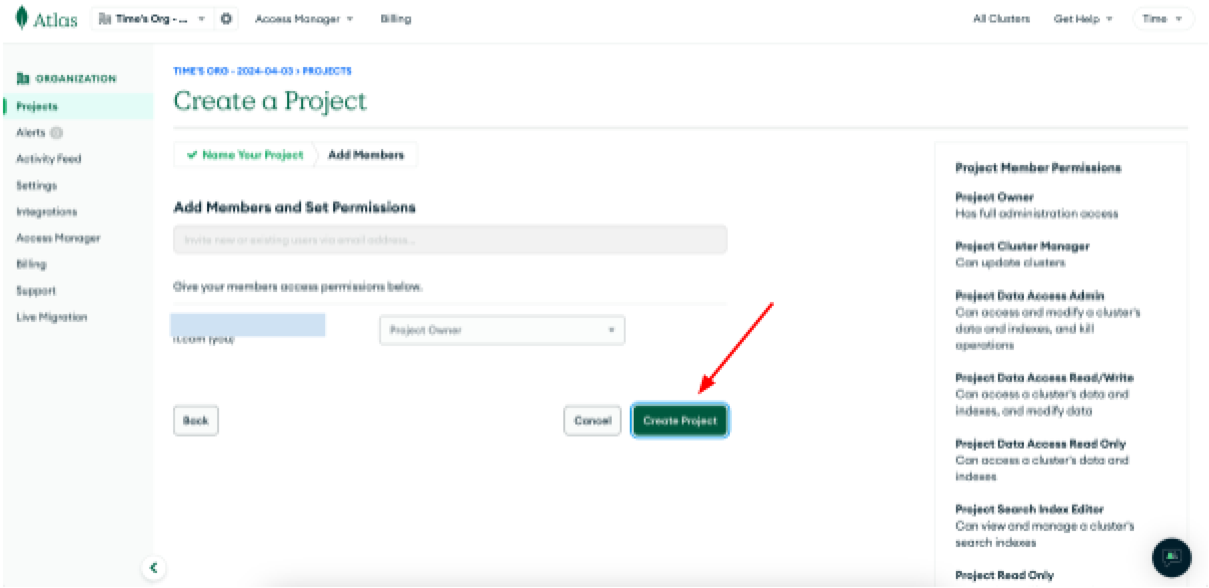
Polling App

Add Tags (Optional)

Use tags to efficiently label and categorize your projects. A project can have a maximum of 50 tags. You can modify tags for the project later. Learn more

Select a key or enter your own : Select a value or enter your own

8. Add Member if needed. Then click on “Create Project”



TIME'S ORG - 2024-04-03 › PROJECTS

Create a Project

Add Members and Set Permissions

Invite new or existing users via email address...

Give your members access permissions below.

.com (you) Project Owner

Project Member Permissions

Project Owner
Has full administration access

Project Cluster Manager
Can update clusters

Project Data Access Admin
Can access and modify a cluster's data and indexes, and kill operations

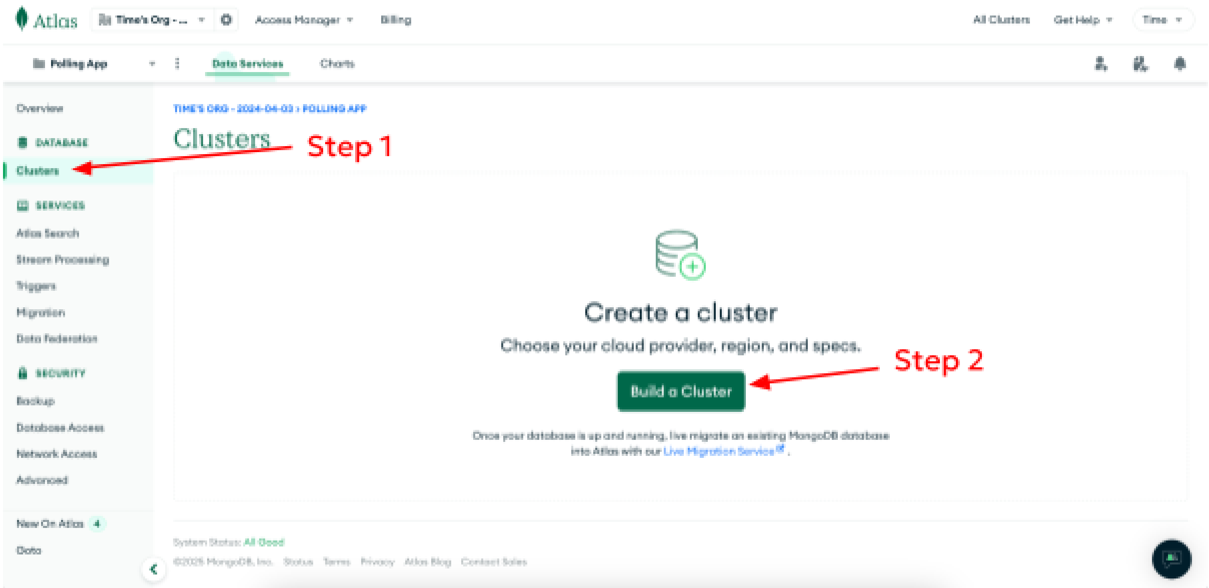
Project Data Access Read/Write
Can access a cluster's data and indexes, and modify data

Project Data Access Read Only
Can access a cluster's data and indexes

Project Search Index Editor
Can view and manage a cluster's search indexes

Project Read Only

9. Now click on “Clusters” option in the side menu and click on “Build a Cluster” button



TIME'S ORG - 2024-04-03 › POLLING APP

Clusters

Create a cluster

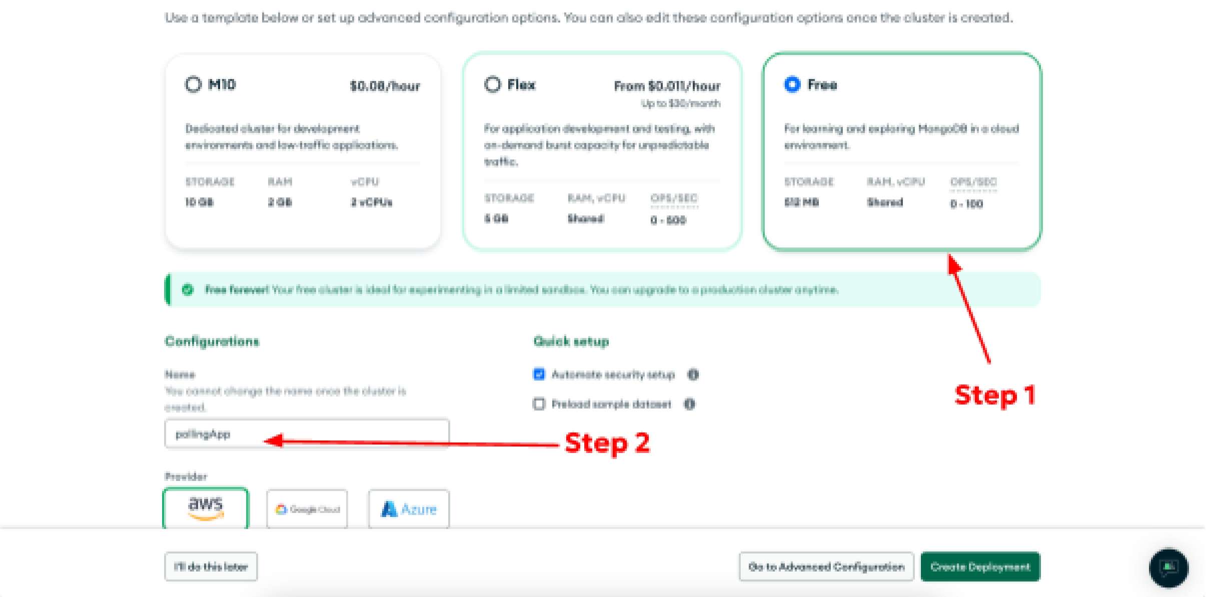
Choose your cloud provider, region, and specs.

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our Live Migration Service.

System Status: All Good

©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

10. Now, select free tier and give a cluster name



Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

○ M10 $0.08/hour

Dedicated cluster for development environments and low-traffic applications.

| STORAGE | RAM | vCPU |
|---|---|---|
| 10 GB | 2 GB | 2 vCPUs |

○ Flex From $0.011/hour Up to $30/month

For application development and testing, with on-demand burst capacity for unpredictable traffic.

| STORAGE | RAM, vCPU | OPS/SEC |
|---|---|---|
| 5 GB | Shared | 0 - 500 |

● Free

For learning and exploring MongoDB in a cloud environment.

| STORAGE | RAM, vCPU | OPS/SEC |
|---|---|---|
| 512 MB | Shared | 0 - 100 |

Free forever! Your free cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

**Configurations**

Name

You cannot change the name once the cluster is created.

pollingApp

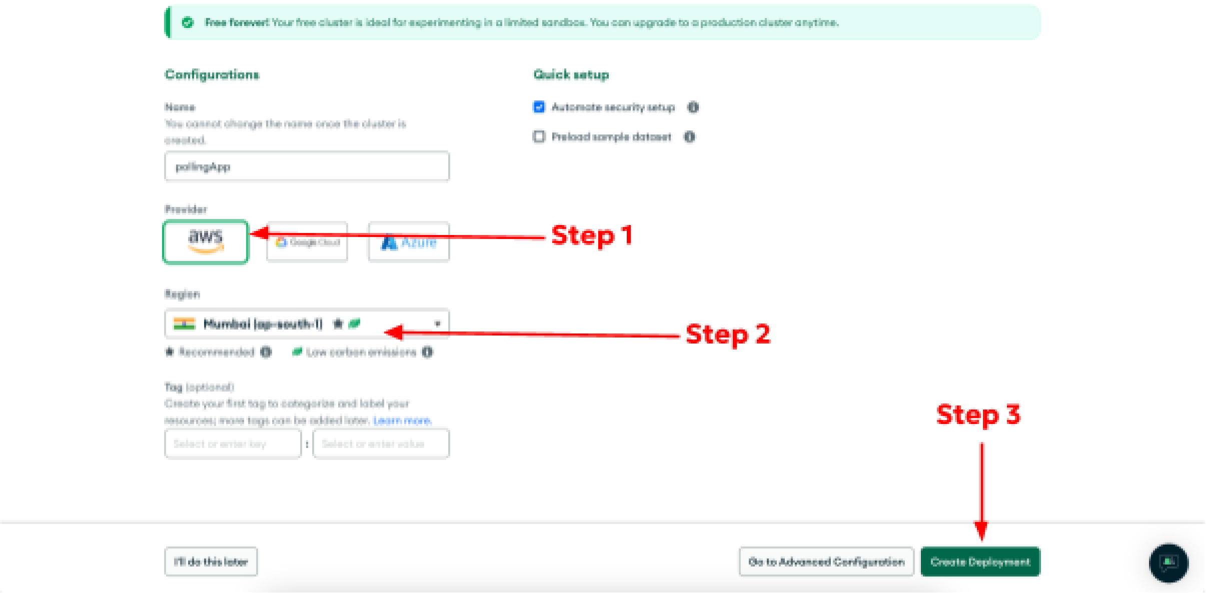
Provider

**Quick setup**

☑ Automate security setup

□ Preload sample dataset

11. Select a server provider, select a region that's near you, and click on “Create Deployment”



Free forever! Your free cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Configurations

Name
You cannot change the name once the cluster is created.
pollingApp

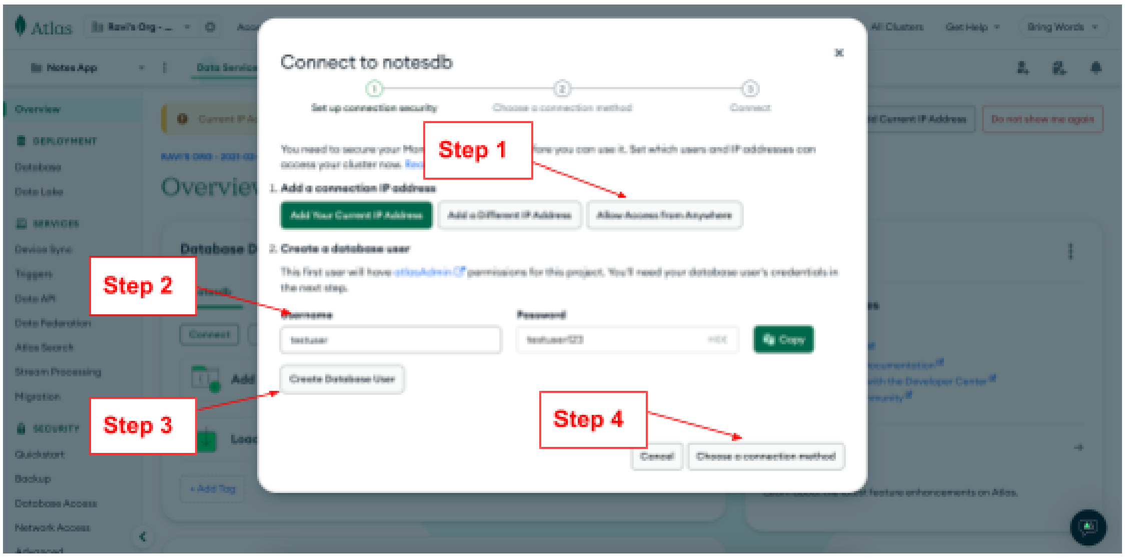
Provider

Region
Mumbai (ap-south-1)

Tag (optional)
Create your first tag to categorize and label your resources; more tags can be added later. Learn more.

Quick setup
☑ Automate security setup
□ Preload sample dataset

12. Now we will be directed to the connection steps page.
13. Here, we need to add an IP address for the connection. I usually select the \`Allow Access from Where\` option and create a database user. And click on the “Choose a connection method” button



Connect to notesdb

You need to secure your Mon[...]fore you can use it. Set which users and IP addresses can access your cluster now.

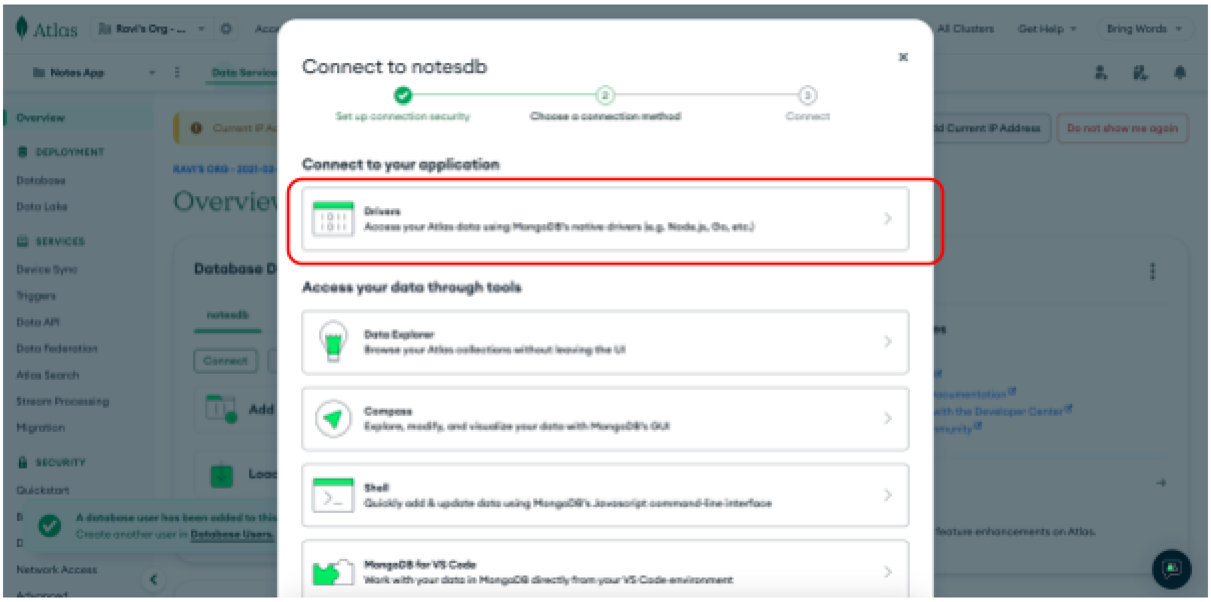
1. Add a connection IP address

2. Create a database user

This first user will have atlasAdmin permissions for this project. You'll need your database user's credentials in the next step.

Username: testuser
Password: testuser123

14. In the Next step, Select the \`Drivers\` option to access the atlas database using our Node.js project



Connect to notesdb

Set up connection security · Choose a connection method · Connect

**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**

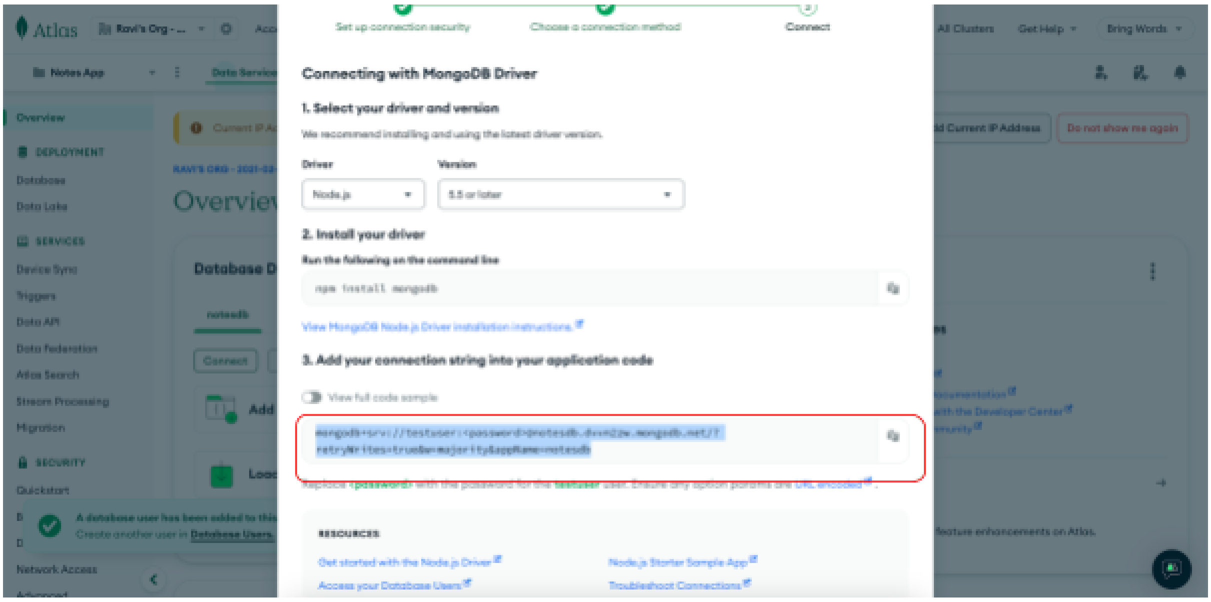
**Data Explorer**
Browse your Atlas collections without leaving the UI

**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

15. Now, copy the connection string



Set up connection security · Choose a connection method · Connect

**Connecting with MongoDB Driver**

**1. Select your driver and version**

We recommend installing and using the latest driver version.

Driver: Node.js
Version: 5.5 or later

**2. Install your driver**

Run the following on the command line

```
npm install mongodb
```

View MongoDB Node.js Driver installation instructions.

**3. Add your connection string into your application code**

View full code sample

```
mongodb+srv://testuser:<password>@notesdb.dvn2zw.mongodb.net/?retryWrites=true&w=majority&appName=notesdb
```

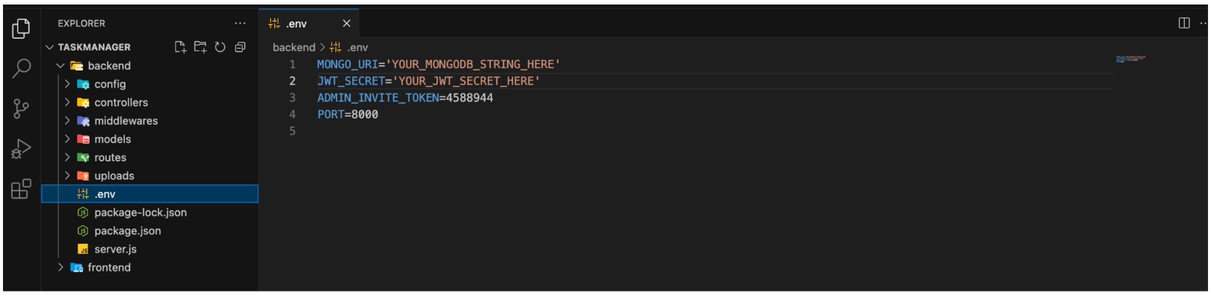
Replace <password> with the password for the testuser user. Ensure any option params are URL encoded.

RESOURCES

Get started with the Node.js Driver · Node.js Starter Sample App · Access your Database Users · Troubleshoot Connections

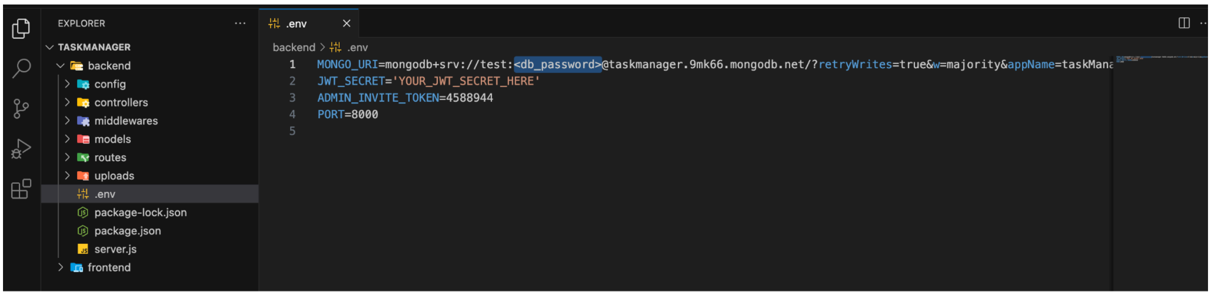
16. Paste the connection string inside the `.env` file:

Before:



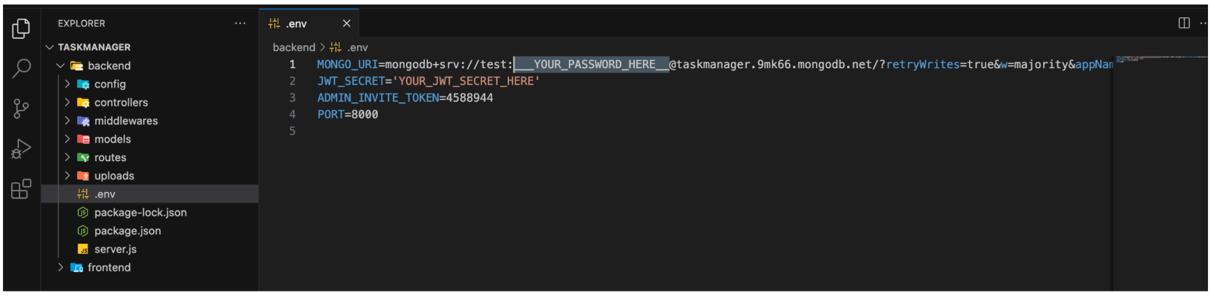
```
MONGO_URI='YOUR_MONGODB_STRING_HERE'
JWT_SECRET='YOUR_JWT_SECRET_HERE'
ADMIN_INVITE_TOKEN=4588944
PORT=8000
```

After:



```
MONGO_URI=mongodb+srv://test:<db_password>@taskmanager.9mk66.mongodb.net/?retryWrites=true&w=majority&appName=taskMana
JWT_SECRET='YOUR_JWT_SECRET_HERE'
ADMIN_INVITE_TOKEN=4588944
PORT=8000
```

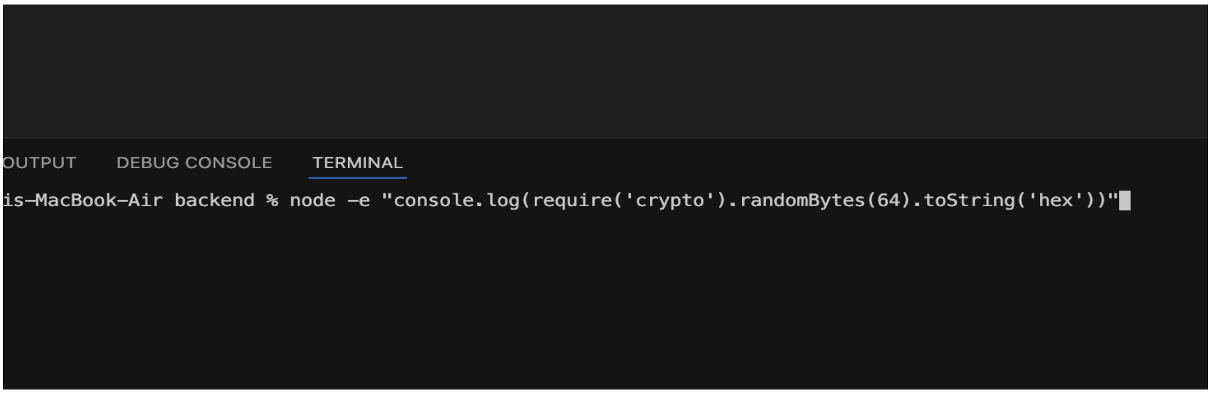
17. Now replace `<password>` in the connection string with the user's password that we have created in Step 13



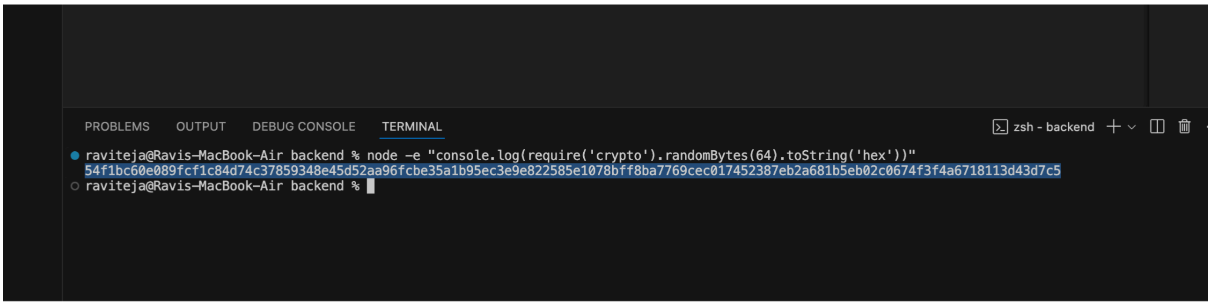
```
MONGO_URI=mongodb+srv://test:__YOUR_PASSWORD_HERE__@taskmanager.9mk66.mongodb.net/?retryWrites=true&w=majority&appNam
JWT_SECRET='YOUR_JWT_SECRET_HERE'
ADMIN_INVITE_TOKEN=4588944
PORT=8000
```

18. After updating the connection string, let's generate our JWT_SECRET. To do so, you can run the following command in the terminal.

**node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"**

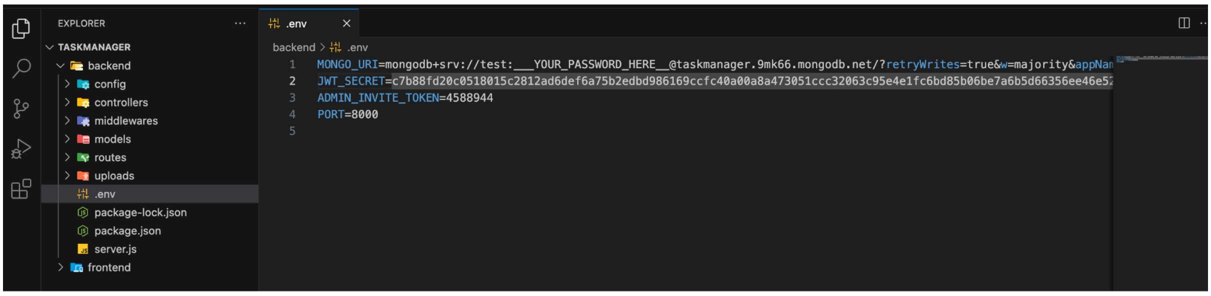


```
is-MacBook-Air backend % node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"
```



```
raviteja@Ravis-MacBook-Air backend % node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"
54f1bc60e089fcf1c84d74c37859348e45d52aa96fcbe35a1b95ec3e9e822585e1078bff8ba7769cec017452387eb2a681b5eb02c0674f3f4a6718113d43d7c5
raviteja@Ravis-MacBook-Air backend % 
```

19. Now lets update the JWT_SECRET



```
MONGO_URI=mongodb+srv://test:___YOUR_PASSWORD_HERE__@taskmanager.9mk66.mongodb.net/?retryWrites=true&w=majority&appNam
JWT_SECRET=c7b88fd20c0518015c2812ad6def6a75b2edbd986169ccfc40a00a8a473051ccc32063c95e4e1fc6bd85b06be7a6b5d66356ee46e52
ADMIN_INVITE_TOKEN=4588944
PORT=8000
```

That's it, we are done.

20. Now, start the server by running:

```
npm run dev
```

Frontend

1. Navigate to the `frontend/Task-Manager` folder.
2. Run the following command to install the required dependencies:

```
npm install
```

3. After the installation is complete, start the React development server by running:

```
npm run dev
```

This will start the frontend server and open the app in your default web browser.