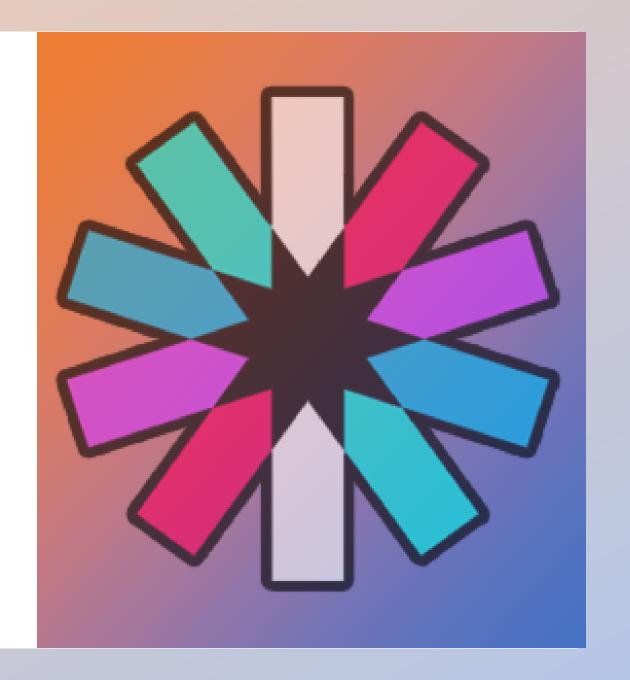
Node Auth Tutorial (JWT)

#1 - Intro & Setup



In this Node.js Auth tutorial series we'll learn how to implement an authentication system using Node.js, Express, MongoDB & JSON web tokens (JWT).





express

Terminal

npm init --yes npm install mongoose npm install express npm install ejs

- Create folder "public"
- Put the image "Smoothie.png" in the folder "public"
- Create file "styles.css"



styles.css

/* google fonts */	• • •	• • •	• • •
<pre>@import url('https://fonts.googleapis.co</pre>	nav li a{	button{	header .btn{
m/css2?family=Quicksand:wght@300;400;500	text-transform: uppercase;	margin-top: 30px;	margin-top: 20px;
;600;700&display=swap');	font-weight: 700;	border-radius: 36px;	padding: 12px 18px;
,	font-size: 0.8em;	background: #FEE996;	text-transform: uppercase;
body{	display: block;	border:0;	<pre>font-weight: bold;</pre>
margin: 20px 40px;	padding: 10px 16px;	text-transform: uppercase;	<pre>display: inline-block;</pre>
font-size: 1.2rem;	letter-spacing: 2px;	font-weight: 700;	font-size: 0.8em;
letter-spacing: 1px;	}	font-size: 0.8em;	1
background: #fafafa;	.btn{	display: block;	.recipes{
1	border-radius: 36px;	padding: 10px 16px;	display: grid;
h1, h2, h3, h4, ul, li, a, p, input, lab	background: #FEE996;	letter-spacing: 2px;	grid-template-columns: 1fr 1fr 1fr;
el, button, div, footer{	Dackground. #FEE990,	recter-spacing. 2px,	column-gap: 30px;
	form h2()	
margin: 0;	form h2{	.error{	row-gap: 80px;
<pre>padding: 0; font-family: 'Quicksand', sans-serif;</pre>	font-size: 2.4em;	color: #ff0099;	margin: 80PX AUTO;
	font-weight: 900;	margin: 10px 2px;	max-width: 1200px;
color: #444;	margin-bottom: 40px;	font-size: 0.8em;	}
}	}	font-weight: bold;	.recipe{
ul{	form{	}	<pre>display: inline-block;</pre>
list-style-type: none;	width: 360px;	header{	border-radius: 20px;
}	margin: 0 auto;	<pre>display: flex;</pre>	<pre>background: white;</pre>
a{	<pre>padding: 30px;</pre>	align-items: center;	position: relative;
text-decoration: none;	box-	}	text-align: center;
}	<pre>shadow: 1px 2px 3px rgba(0,0,0,0.1);</pre>	header img{	<pre>padding: 0 20px 30px 20px</pre>
nav{	border-radius: 10px;	width: 250px;	}
<pre>display: flex;</pre>	<pre>background: white;</pre>	margin-right: 40px;	<pre>.recipe img{</pre>
<pre>justify-content: space-between;</pre>	}	}	width: 100px;
align-items: flex-end;	<pre>input{</pre>	header h2{	margin: -30px auto 20px;
margin-bottom: 120px;	<pre>padding: 10px 12px;</pre>	<pre>font-size: 3em;</pre>	}
}	border-radius: 4px;	<pre>margin-bottom: 10px;</pre>	footer{
nav ul{	<pre>border: 1px solid #ddd;</pre>	}	text-align: center;
<pre>display: flex;</pre>	<pre>font-size: 1em;</pre>	header h3{	<pre>margin-top: 120px;</pre>
align-items: center;	width: 100%;	<pre>font-size: 1.6em;</pre>	color: #aaa;
}	}	<pre>margin-bottom: 10px;</pre>	}
nav li{	label{	margin-left: 2px;	
margin-left: 20px;	<pre>display: block;</pre>	color: #999;	
}	margin: 20px 0 10px;	}	
	}		

- Create folder "views"
- In the folder "views" create file "home.ejs"
- In the folder "views" create file "smoothies.ejs"
- In the folder "views" create folder "partials"
- In the folder "partials" create file "footer.ejs"
- In the folder "partials" create file "header.ejs"



```
<%- include('partials/header'); -%>
<header>
  <div class="smoothie">
   <img src="/smoothie.png" alt="">
 </div>
  <div class="headings">
    <h2>Smoothie Recipes</h2>
    <h3>By Master Chef For Your</h3>
    <a href="/smoothies" class="btn">View Recipes</a>
 </div>
</header>
<%- include('partials/footer'); -%>
```

```
<%- include('partials/header'); -%>
<img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Banana Boost</h4>
   Banana, Vanilla ice cream, Milk
 class="recipe">
   <img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Tropical Twist</h4>
   Peach, Pinapple, Apple juice
 class="recipe">
   <img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Protein Packer</h4>
   Oats, Peanut butter, Milk, Banana, Blueberries
 class="recipe">
   <img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Banana Boost</h4>
   Banana, Vanilla ice cream, Milk
 class="recipe">
   <img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Tropical Twist</h4>
   Peach, Pinapple, Apple juice
 <img src="/smoothie.png" alt="smoothie recipe icon">
   <h4>Protein Packer</h4>
   Oats, Peanut butter, Milk, Banana, Blueberries
 <%- include('partials/footer'); -%>
```

```
<footer>Copyright 2021 Master Chef Smoothies</footer>
</body>
</html>
```

Create file "app.js"

Change file package.json

("main": "app.js")



```
const express = require('express');
const mongoose = require('mongoose');
const app = express();
// middleware
app.use(express.static('public'));
                                                                       You need to create
                                                                       your own DataBase
// view engine
app.set('view engine', 'ejs');
// database connection
const dbURI = 'mongodb+srv://marizza:demo1234@cluster0.del96.mongodb.net/node-auth';
mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true, useCreateIndex:true
  .then((result) => app.listen(3000))
  .catch((err) => console.log(err));
// routes
app.get('/', (req, res) => res.render('home'));
app.get('/smoothies', (req, res) => res.render('smoothies'));
```

Go to https://www.mongodb.com/ and Sign In (with Google account) Create organization and project

← Organizations

Create Organization

Name and Service

Add Members

Next

Name Your Organization

demo

DEMO > JWT

Database Deployments

Q Find a database deployment...



Create a database

Choose your cloud provider, region, and specs.

Build a Database

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



Deploy a cloud database

Experience the best of MongoDB on AWS, Azure, and Google Cloud. Choose a deployment option to get started.

PREVIEW



For serverless applications that aren't critical with variable traffic. Minimal configuration required.

- Pay only for the operations you run
- Resources scale seamlessly to meet your workload
- Always-on security and backups

Create

\$0.30/1M reads

Dedicated

For production applications with sophisticated workload requirements. Advanced configuration controls.

- Network isolation and finegrained access controls
- ✓ On-demand performance advice
- Multi-region and multi-cloud options available

Create

Starting at \$0.08/hr*

*estimated cost \$56.94/month

FREE



Shared

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

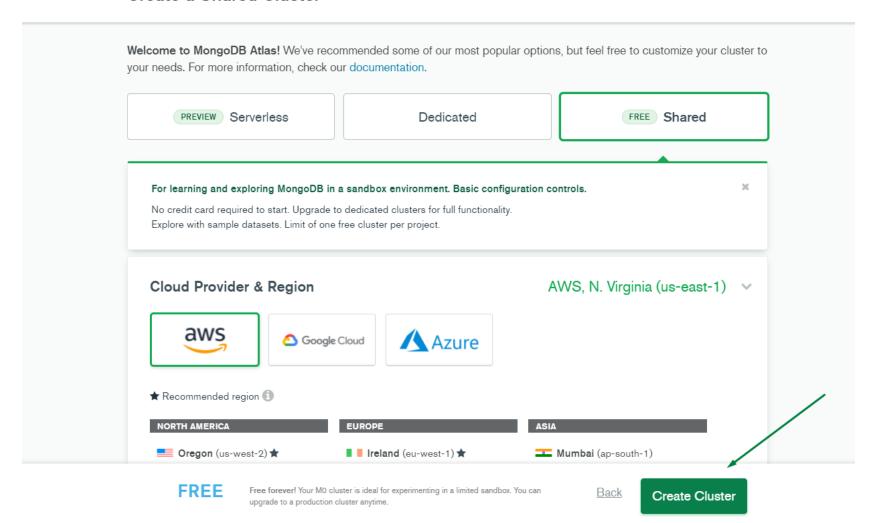
- ✓ No credit card required to start
- Explore with sample datasets
- Upgrade to dedicated clusters for full functionality

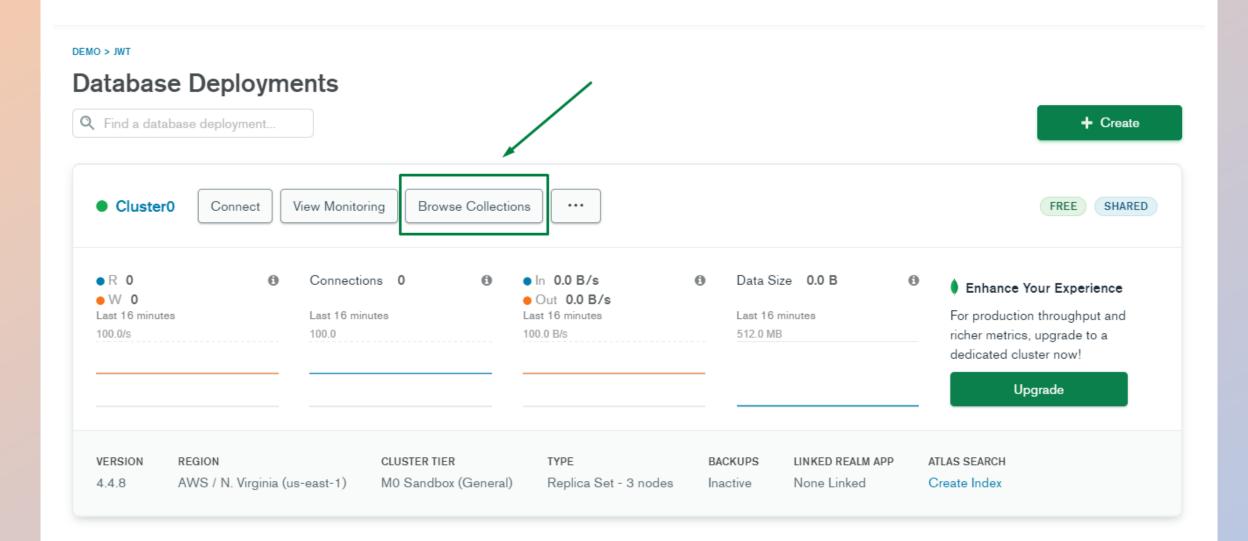
Create

Starting at

FREE

Create a Shared Cluster





DEMO > JWT > DATABASES



VERSION REGION

4.4.8 AWS N. Virginia (us-east-1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Command Line

DATABASES: 0 COLLECTIONS: 0

Idal VISUALIZE YOUR DATA

REFRESH

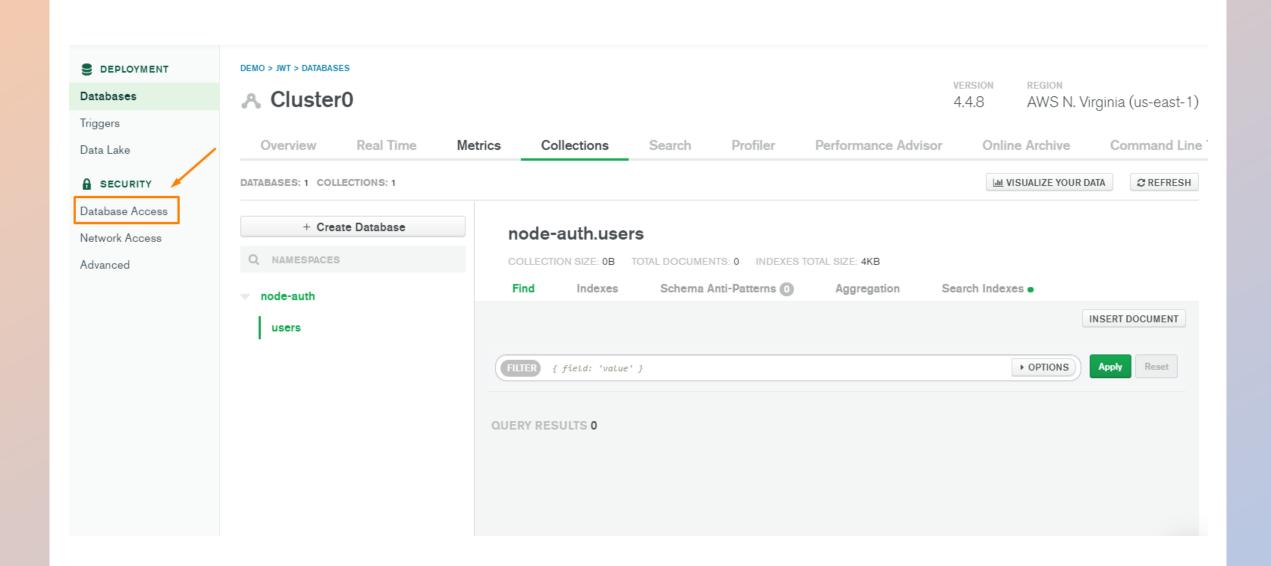


- Find: run queries and interact with documents
- Indexes: build and manage indexes
- Aggregation: test aggregation pipelines
- Search: build search indexes

Load a Sample Dataset

Add My Own Data

X Create Database Database name @ node-auth Collection name @ □ Capped Collection Before MongoDB can save your new database, a collection name must be specified at the time of creation. Create Cancel





Databases

Triggers

Data Lake



♠ SECURITY

Database Access

Network Access

Advanced

DEMO > JWT

Database Access

Database Users

Custom Roles



Create a Database User

Set up database users, permissions, and authentication credentials in order to connect to your clusters.

Add New Database User

Learn more

Authentication Method

Password

Certificate

AWS IAM

(MongoDB 4.4 and up)

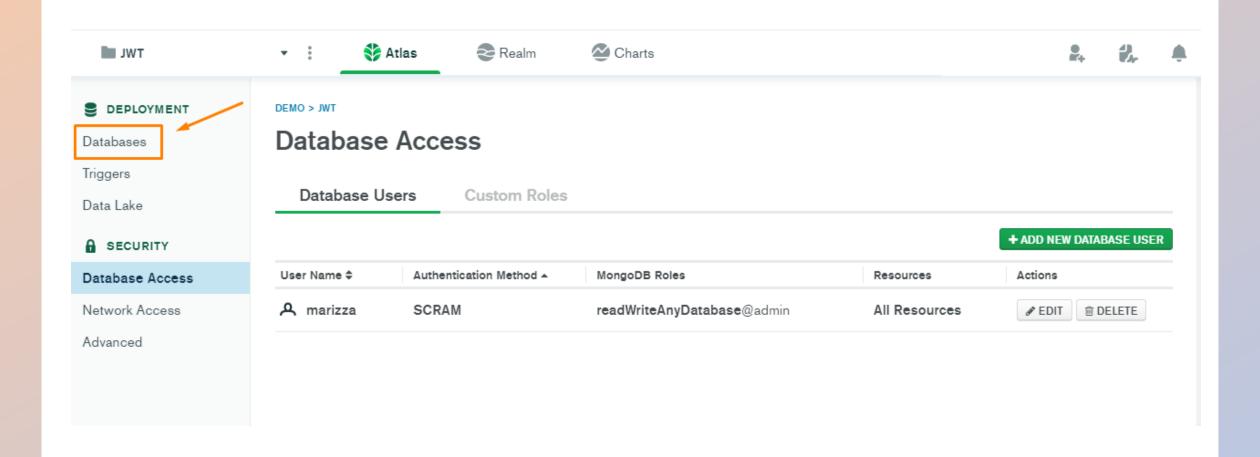
MongoDB uses SCRAM as its default authentication method.

Password Authentication

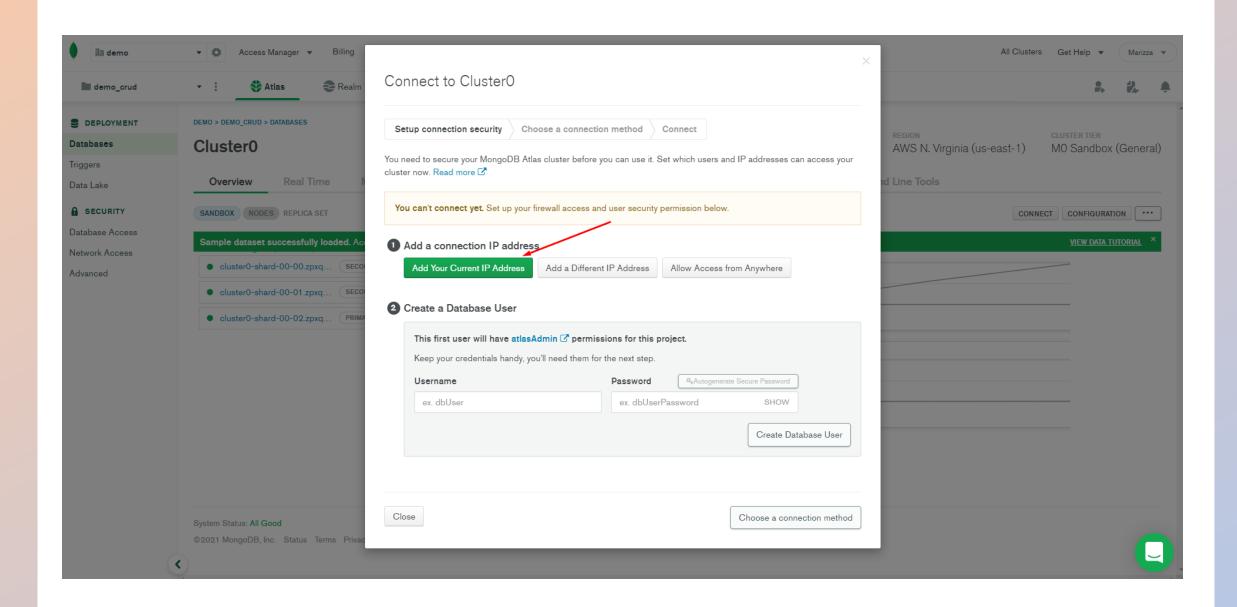
marizza

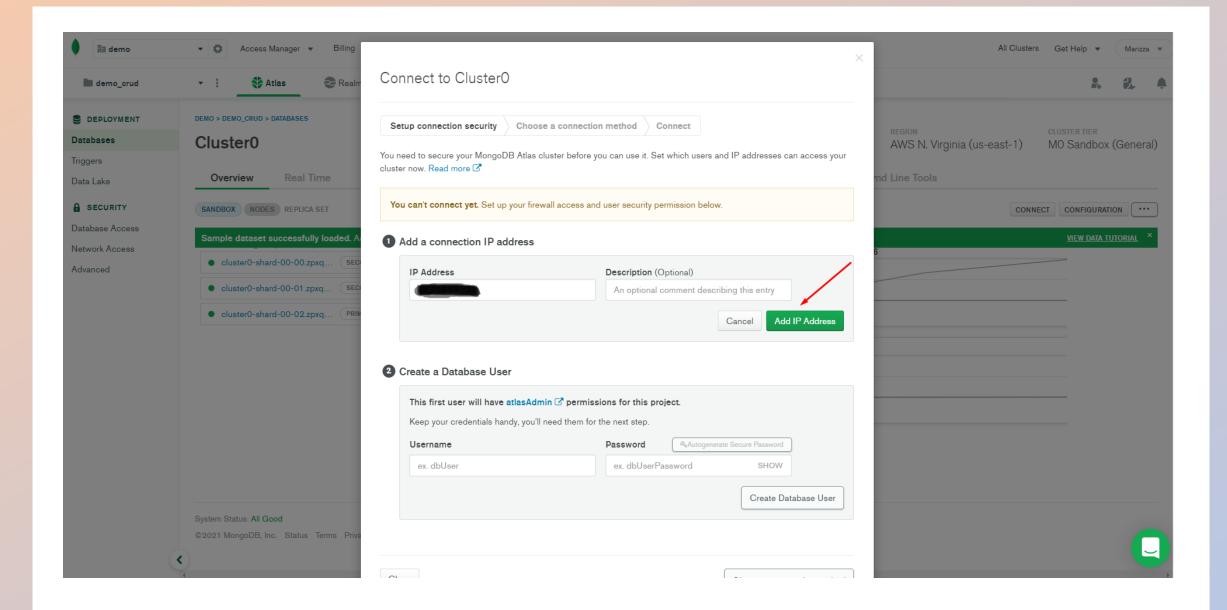
demo1234

HIDE



DEMO > JWT **Database Deployments** + Create Q Find a database deployment... Cluster0 Browse Collections View Monitoring FREE SHARED Connect ... • R 0 Connections 0 ● In 0.0 B/s Data Size 819.2 B Enhance Your Experience W 0 Out 0.0 B/s Last 46 minutes Last 46 minutes Last 46 minutes Last 46 minutes For production throughput and 1.0 512.0 MB 100.0/s 291.5 B/s richer metrics, upgrade to a dedicated cluster now! Upgrade VERSION REGION CLUSTER TIER TYPE BACKUPS LINKED REALM APP ATLAS SEARCH AWS / N. Virginia (us-east-1) Replica Set - 3 nodes None Linked Create Index 4.4.8 M0 Sandbox (General) Inactive





Connect to Cluster0

Setup connection security Choose a connection method

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. Read more 🗷

Connect

You can't connect yet. Set up your firewall access below.

Add a connection IP address

Add Your Current IP Address

Add a Different IP Address

Allow Access from Anywhere

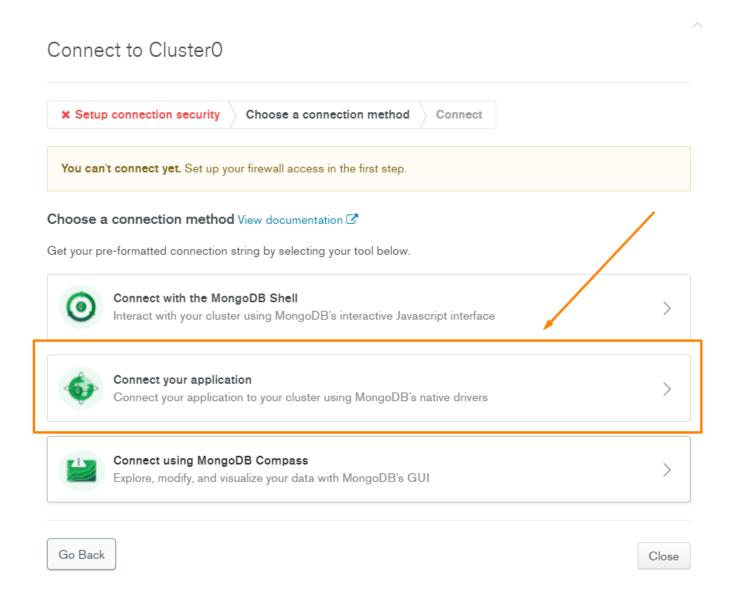
- 2 Create a Database User
 - ✓ A MongoDB user has been added to this project. Not yours? Create one in the MongoDB Users tab.

You'll need your MongoDB user's credentials in the next step.

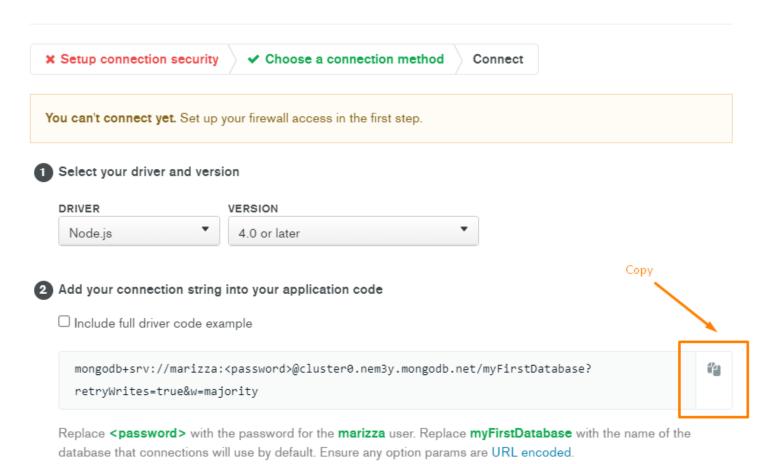
Close

Choose a connection method

 \times



Connect to ClusterO



Having trouble connecting? View our troubleshooting documentation

```
const express = require('express');
const mongoose = require('mongoose');
const app = express();
// middleware
                                                   Replace
app.use(express.static('public'));
// view engine
app.set('view engine', 'ejs');
                                                                                                      delete
                                                                      node-auth
                             marizza demo1234
// database connection
const dbURI = | mongodb+srv://<username>:<password>@cluster0.nem3y.mongodb.net/<dbname>?retryWrites=true&w=majority ' ;
mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true} \
  .then((result) => app.listen(3000))
  .catch((err) => console.log(err));
                                                                                             Paste
// routes
app.get('/', (req, res) => res.render('home'));
app.get('/smoothies', (req, res) => res.render('smoothies'));
```

```
const express = require('express');
const mongoose = require('mongoose');
const app = express();
// middleware
app.use(express.static('public'));
// view engine
app.set('view engine', 'ejs');
// database connection
const dbURI = 'mongodb+srv://marizza:demo1234@cluster0.nem3y.mongodb.net/node-auth';
mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true})
  .then((result) => app.listen(3000))
  .catch((err) => console.log(err));
// routes
app.get('/', (req, res) => res.render('home'));
app.get('/smoothies', (req, res) => res.render('smoothies'));
```

Terminal

nodemon app

Go to http://localhost:3000/

