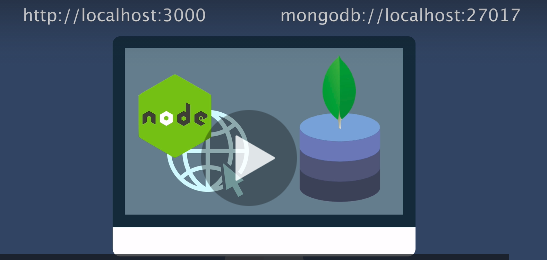
DEVELOPNG APPS WITH DATABASES!

****How to set everything together ?****

Now, we render our app with localhost of our PC and the localhost of MONGODB

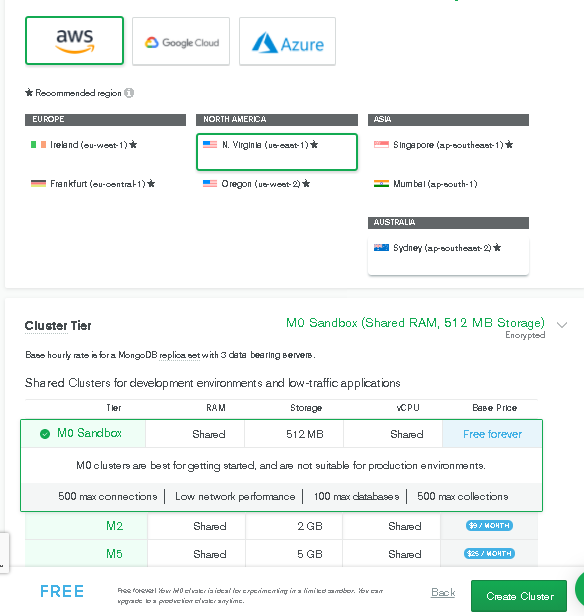
We need, now, render and work online!

Now you se… we will not use HEROKU, but ATLAS instead!

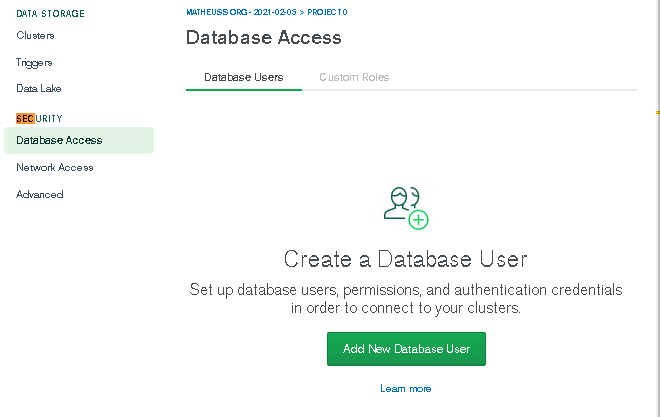
********

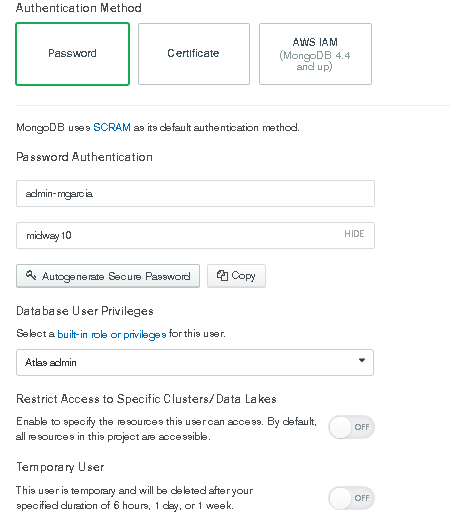
# ****Configuração inicial****

* Sign Up for a MONGODB Atlas
* Use the AWS cloud Cluster

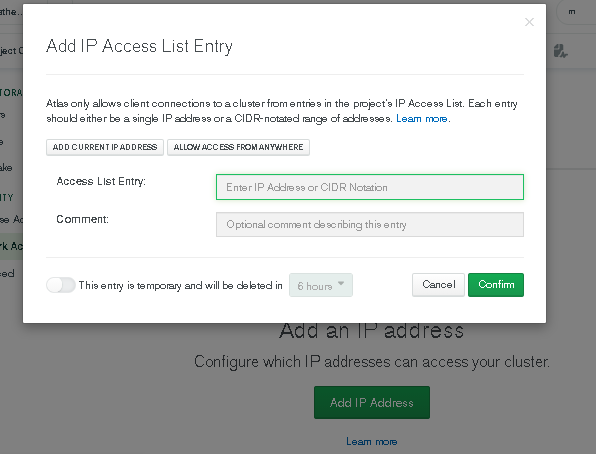


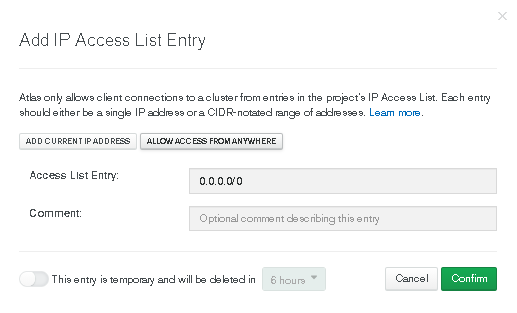
* Now, we wait the background to set up things…
* Good! Now, let’s create a new user!





* Now, let’s configure the IP :





* ****Now, we will connect to our Cluster, entering the following line to Command Line****

***mongo "mongodb+srv://c***luster0.0vq14.mongodb.net/myFirstDatabase" --username admin-mgarcia --password midway10

* When I run it, I am inside it! If I create a collections, it will be shown on Prompt
* Now, how we run it from an app ?? …
* We will go again to “CONNECT” and select “Connect from your application” , copy the path and paste it on the code!

mongodb+srv://admin-mgarcia:<password>@cluster0.0vq14.mongodb.net/myFirstDatabase?retryWrites=true&w=majority

//mongoose.connect("mongodb://localhost:27017/toDoListDB", { //we chage it for…

**mongoose.connect("mongodb+srv://admin-mgarcia:**midway10**@cluster0.0vq14.mongodb.net/ **toDoListDB**", {**

* ****Voìla! We can run it with nodemon app.js, even though we didin’t rund the “mongod … waiting for connections”. The connection now is out of my prompt… it is online instead!****
* ****If I run the “localhost” and add some item, it will be shown in the CLUSTER of Atlas !!!!****

**RESUMINDO!! NÓS SETAMOS O ATLAS PARA RECEBER APPS. CONFIGUREI O ADMIN , O ‘IP’ E O CLUSTER. CONFIGUREI O PROGRAMA PARA POSTAR DIRETO NO ATLAS .**

# ****CONNECTING TO HEROKU****

Let us go back to class “251” … We will, first, configure the Heroku. Always, go to “devcenter/heroku” for questions.

* Now, I will check if I have everything up and running

heroku

heroku –version

node –version

git –version

npm –version

* **Inside the - devcenter/heroku – we will ‘get started’, choosing ‘app.js’, and then go to ‘ I am ready – I have installed Heroku CLI’**
* Now, go to :
  + - “However, if you have your own existing application that you want to deploy instead, see [*this article*](https://devcenter.heroku.com/articles/preparing-a-codebase-for-heroku-deployment) to learn how to prepare it for Heroku deployment.”
* **Now, lets prepare the git:**

cd onlineTDL

git init

git add .

git commit –m “Initial Comit

* Now, let’s log into Heroku

heroku login

//enter email and password

Now, we create a heroku remote repository

Heroku create

https://git.heroku.com/evening-retreat-09581.git

touch Procfile

start Procfile –a Atom

* **inside the atom/Procfile, we write**

**web: node app.js**

* **now, on the /app.js**

****let port = process.env.PORT;****

****if (port == null || port == "") {****

****port = 3000;****

****}****

****app.listen(port);****

****app.listen(port, function() {****

****console.log("Server started on port 3000");****

****});****

* **Now, inside the /package.json**

****.****

****.****

****.****

****"license": "ISC",****

****"engines": {****

****"node": "12.18.3"**** //it my version, to check, just type **node --version**

****},****

****.****

****.****

****.****

* Now, let’s ignore the non-important files.

touch .gitignore

start .gitignore -a Atom

* ****And inside the /gitignore :****

****/node\_modules****

****npm-debug.log****

****.DS\_Store****

****/\*.env****

git add .

git commit -m "Add gitignore, procfile and update ports "

git push heroku master