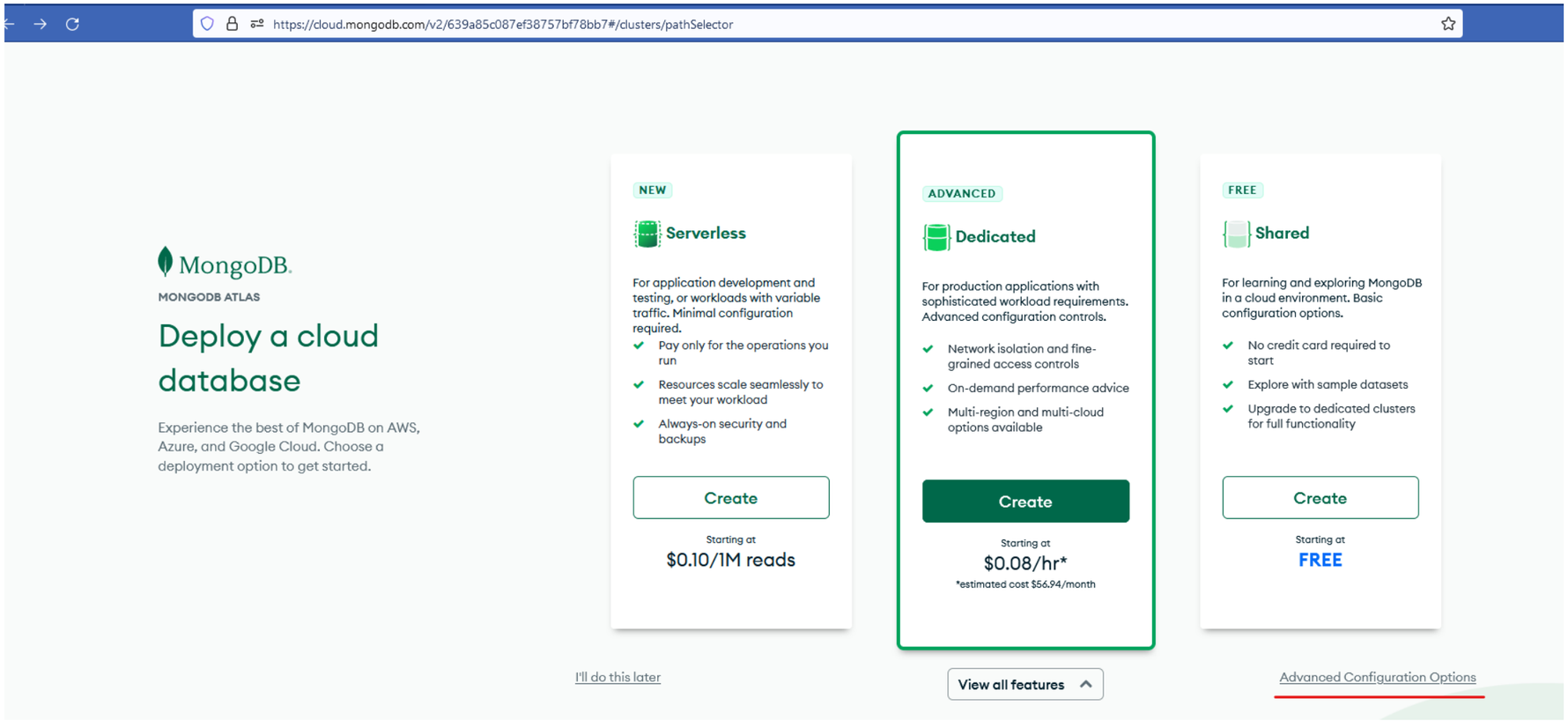
Passos após se cadastrar em https://www.mongodb.com/cloud/atlas/register (indicações em vermelho nas imagens)



https://cloud.mongodb.com/v2/639a85c087ef38757bf78bb7#/clusters/edit?filter=advanced&fromPathSelector=true

CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our documentation.

Serverless | Dedicated | FREE **Shared**

For learning and exploring MongoDB in a sandbox environment. Basic configuration controls.

No credit card required to start. Upgrade to dedicated clusters for full functionality.
Explore with sample datasets. Limit of one free cluster per project.

Cloud Provider & Region — AWS, Sao Paulo (sa-east-1)

Cluster Tier — M0 Sandbox (Shared RAM, 512 MB Storage) Encrypted

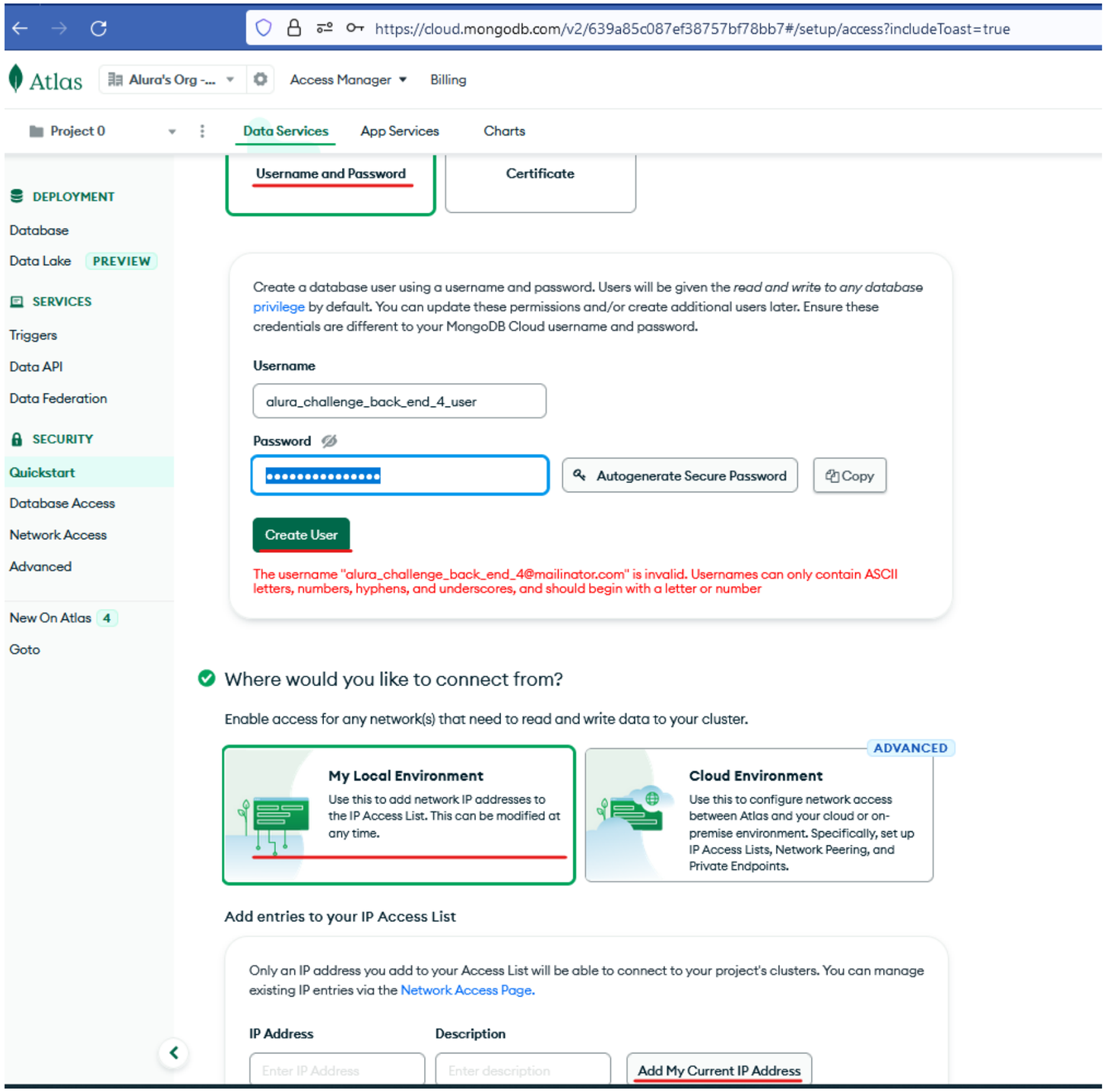
Additional Settings — MongoDB 5.0, No Backup

Cluster Name — Cluster0

FREE **Free forever!** Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

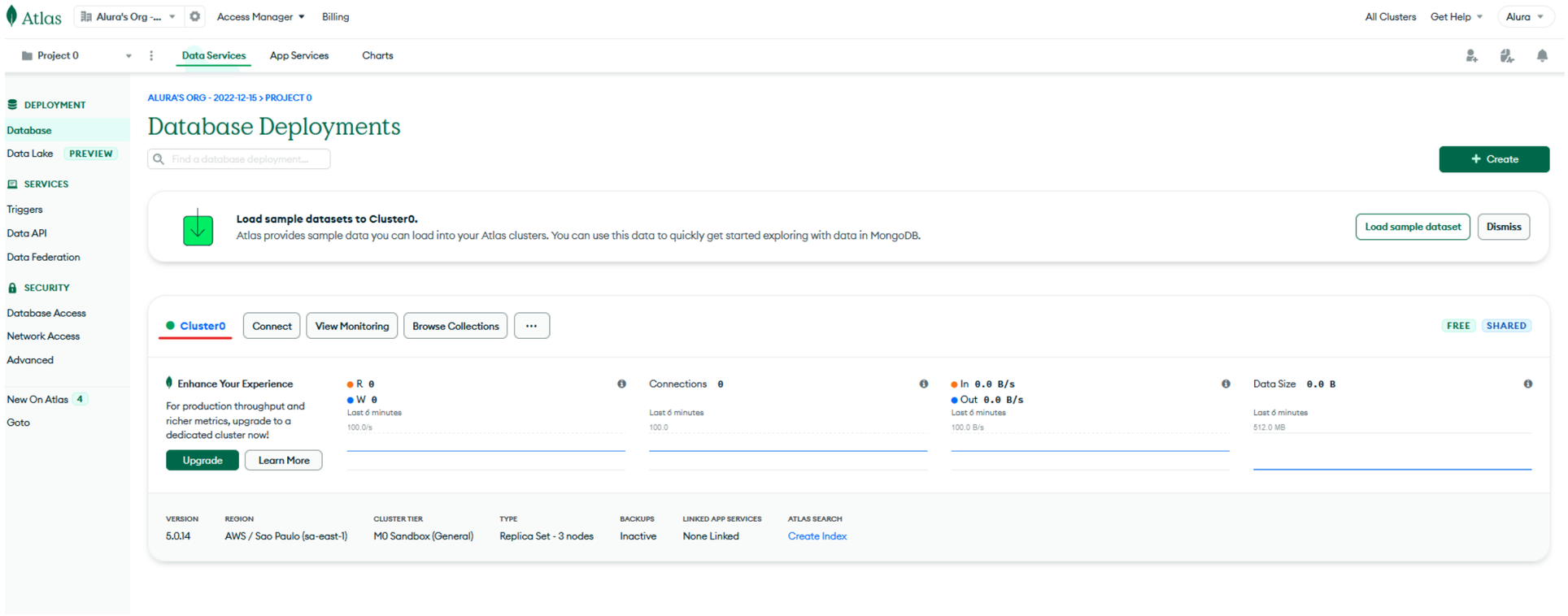
Back | Create Cluster

- Além de criar um usuário e senha, adicionar o IP atual para permitir sua conexão ao banco de dados.

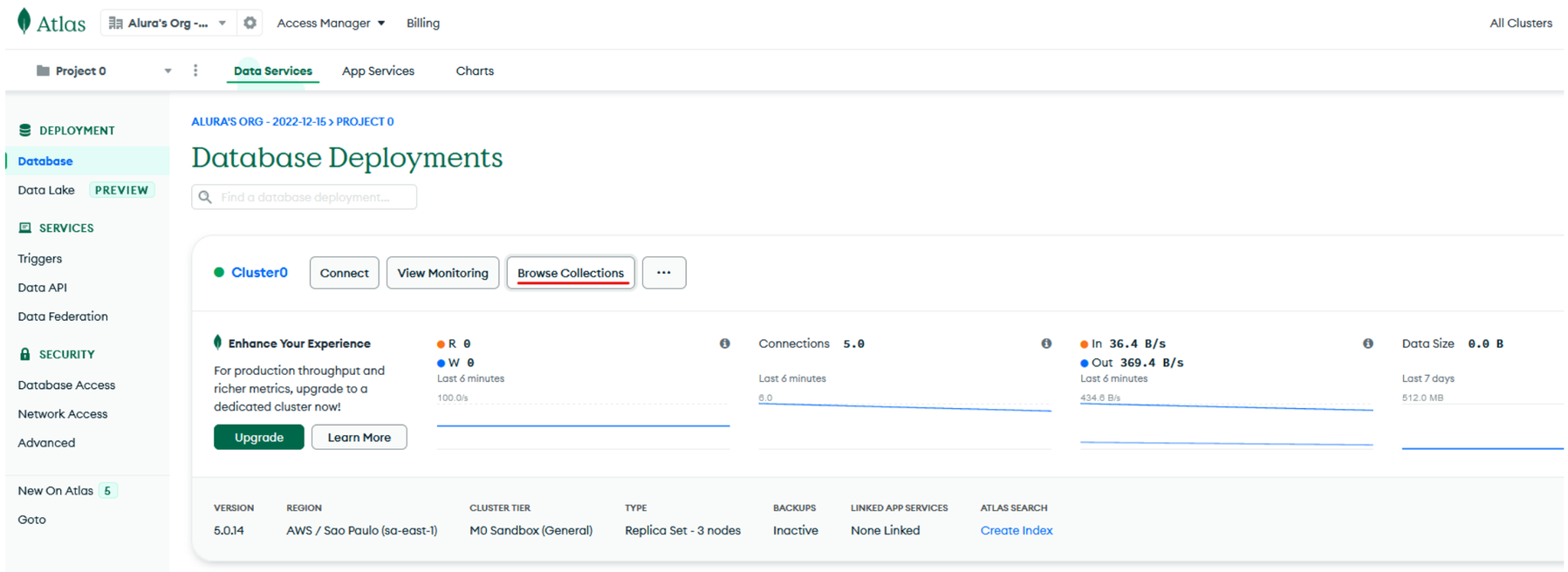


    - anotar username e password do banco de dados (será utilizado posteriormente)
    - clicar no botão Finish and Close

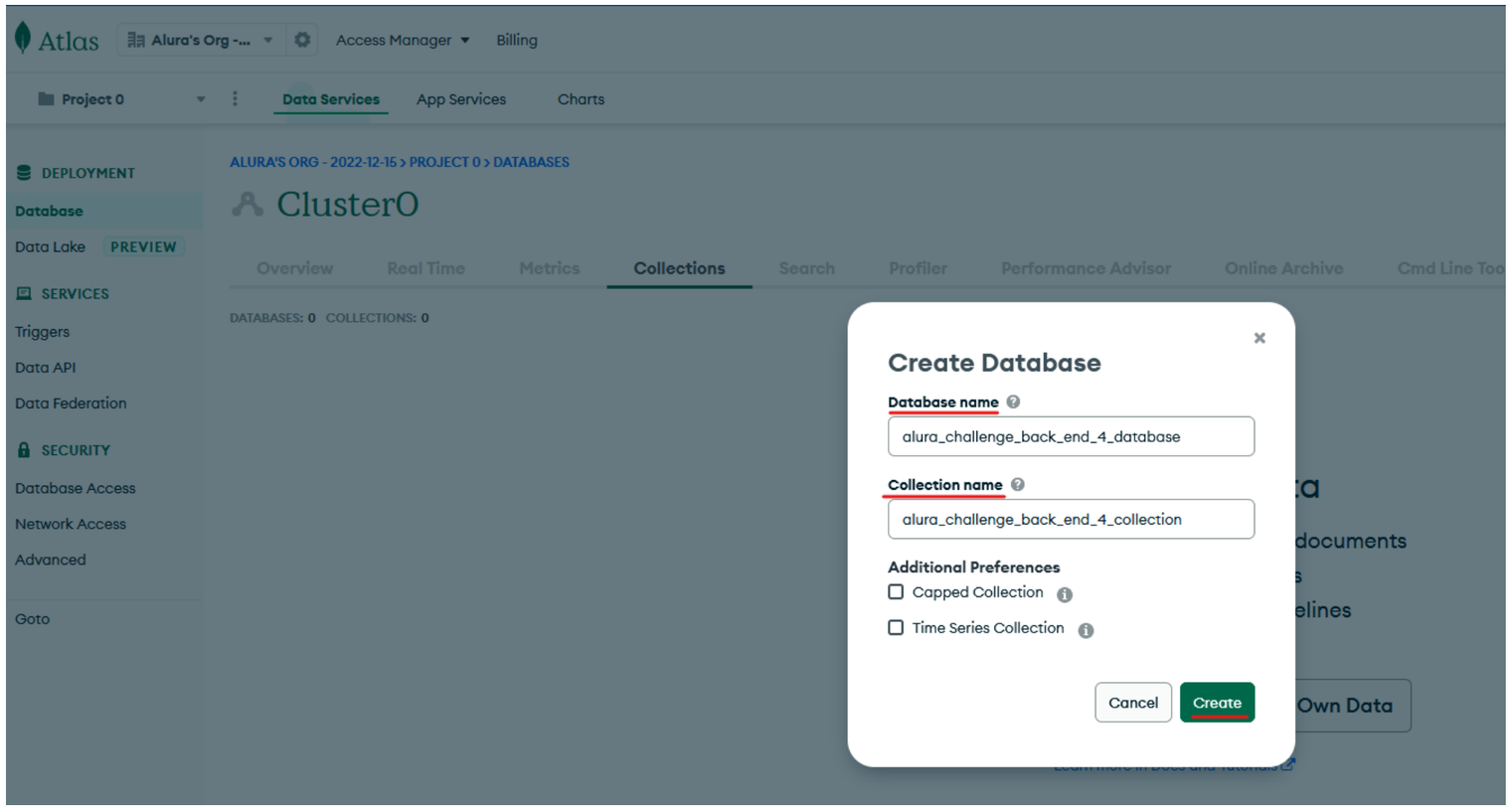
- anotar o nome do cluster (será utilizado posteriormente, ex: Cluster0)



- clicar nas opções “Browse Collections” + “Add my own data”

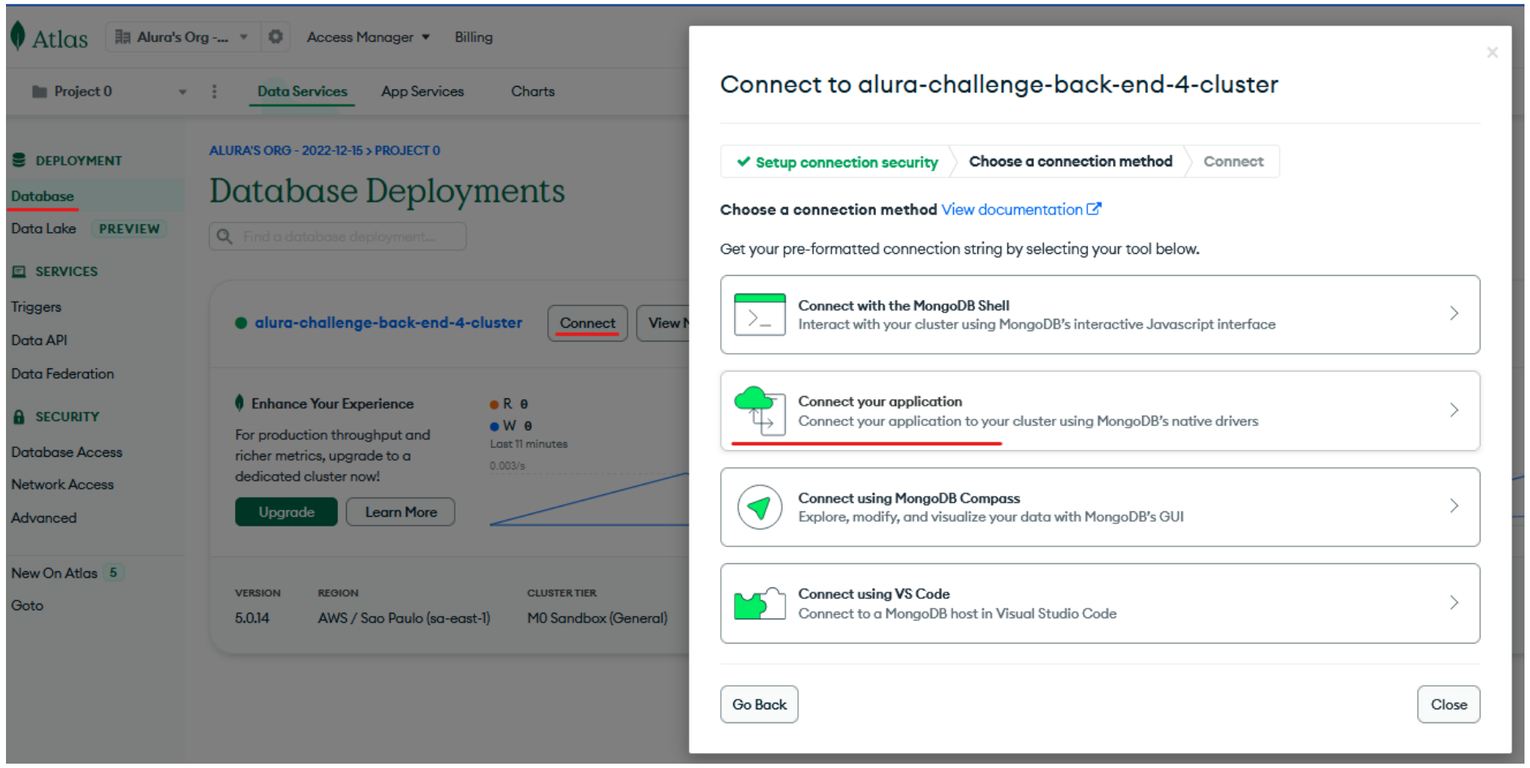


- informar um nome de banco de dados (database name) e um nome de coleção (collection name) + botão Create

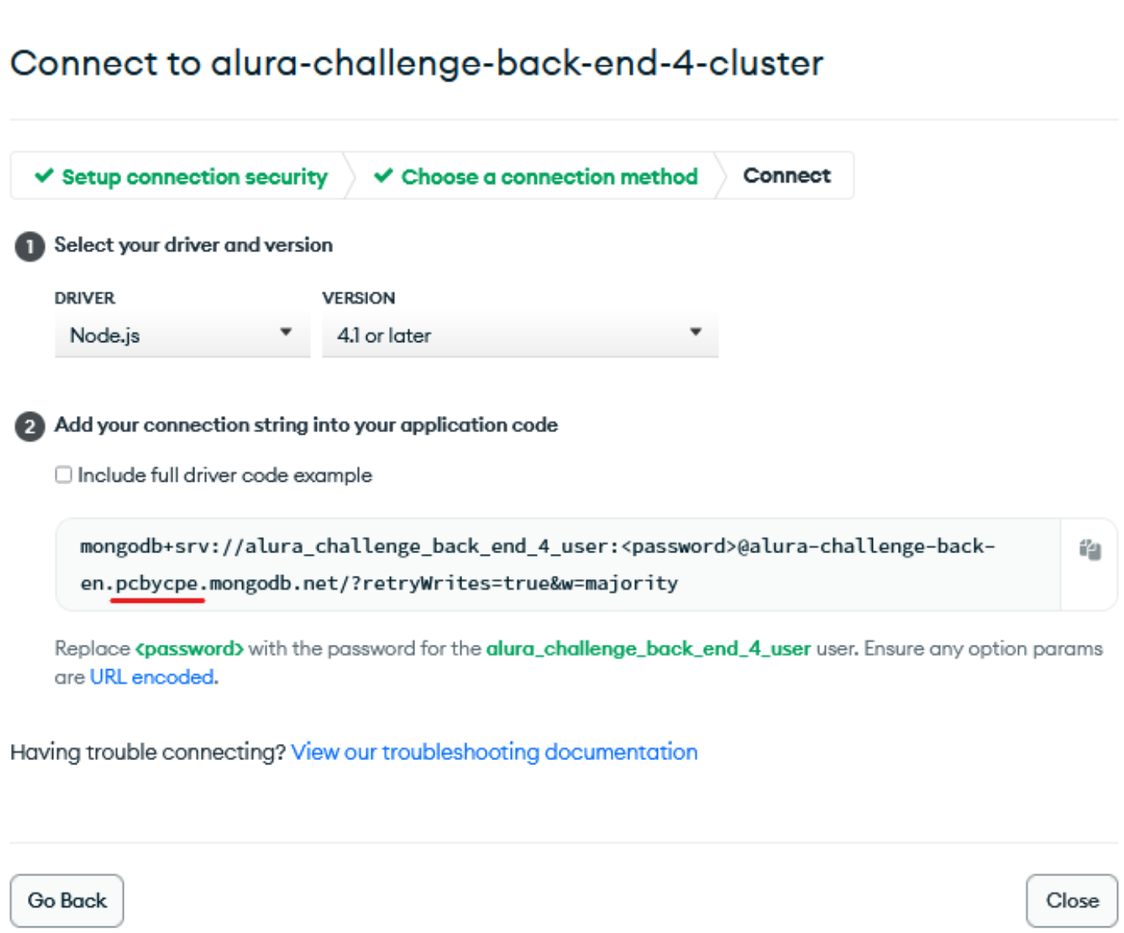


- anotar o database name (será utilizado posteriormente, por ex: alura_challenge_back_end_4_database)

- acessar as opções Database + Connect + “Connect your application”



- na janela exibida, anotar o subdomínio (será utilizado posteriormente, ex: pcbycpe)

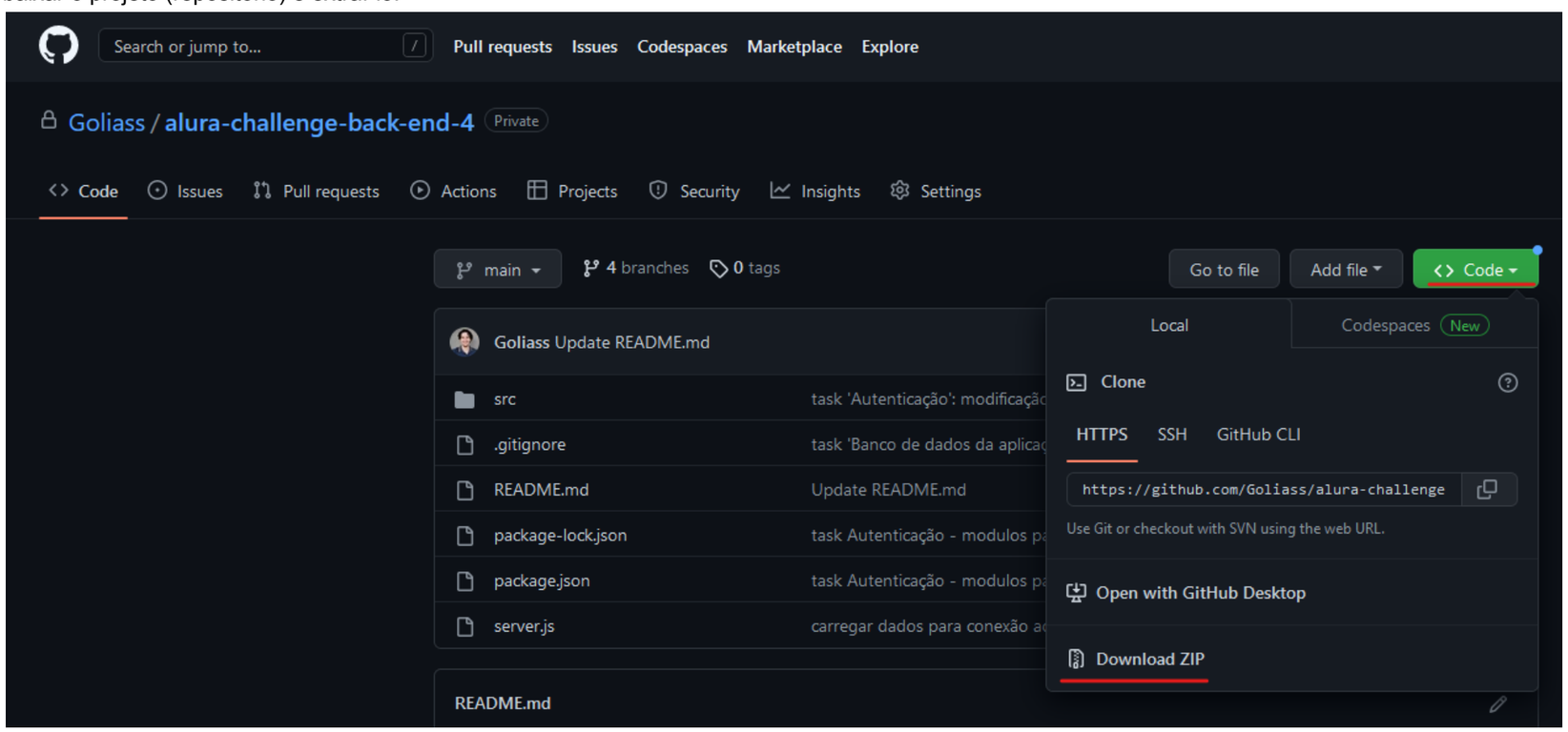


nodejs / npm

- nodejs e npm
  instalar a versão 6.14.11 do **npm** ou outra compatível
  instalar a versão v14.16.0 do **nodejs** ou outra compatível

projeto

- baixar o projeto (repositório) e extraí-lo:



- abrir o terminal (cmd / bash), acessar a pasta raiz do repositório/projeto (ex: alura-challenge-back-end-4) e executar (comando a seguir + enter) **npm install**

- criar arquivo .env na raiz do [diretório do] projeto com as seguintes variáveis de ambiente, fazendo algumas substituições conforme indicado:

```
# database
DBUSER="alura_challenge_back_end_4_user" # substituir pelo NOME/USERNAME do usuário do banco de dados (anotado anteriormente)
DBPASS="password" # substituir pela SENHA/PASSWORD do usuário do banco de dados (anotado anteriormente)
DBCLUSTER="Cluster0" # substituir pelo NOME do cluster (anotado anteriormente)
DBNAME="alura_challenge_back_end_4_database" # substituir pelo NOME do banco de dados (anotado anteriormente)
SUBDOMAIN="pcbycpe" # substituir pelo subdomínio (anotado anteriormente)

# authentication
saltRounds=12
jwtKey="str1ngAleatori@" # substituir por uma string aleatória
jwtExpirationTimeSpan="15m" # exs 1s 15m
```

- pelo terminal (cmd / bash), na pasta raiz do repositório/projeto (ex: alura-challenge-back-end-4), executar **npm run dev**
    - se execução com sucesso, será exibida informação parecida com a seguinte:

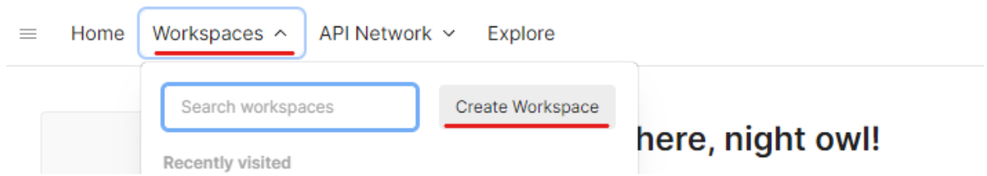
```
> nodemon server.js

[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Servidor escutando em http://localhost:3001
Database connection succeed
```

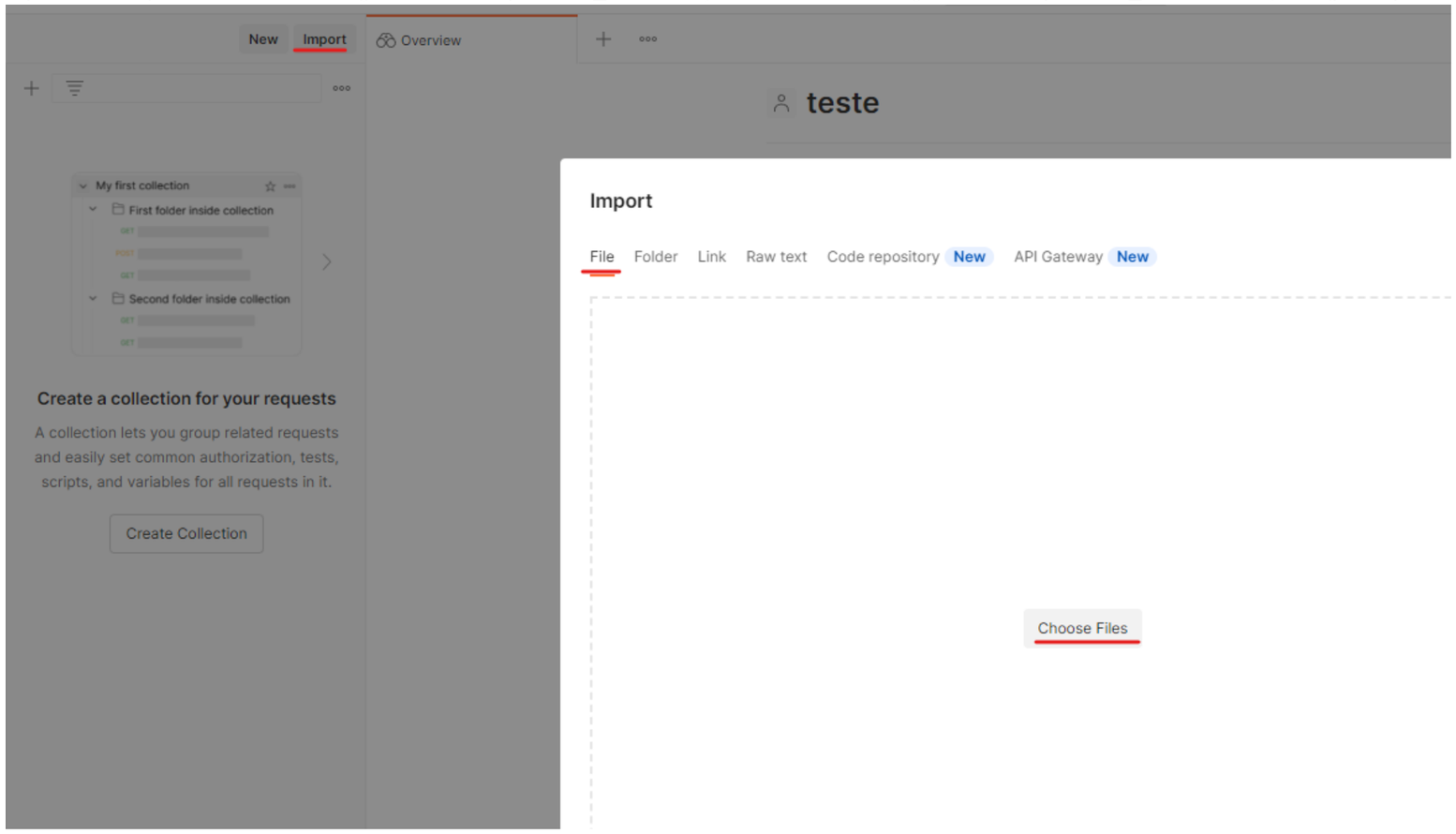
    - Se erro, atualizar o IP na interface (pode ter expirado), para permitir/considerar o IP atual do computador (ver mais acima o tópico "*Além de criar um usuário e senha, adicionar o IP atual para permitir sua conexão ao banco de dados*") e seguir o passo anterior novamente.

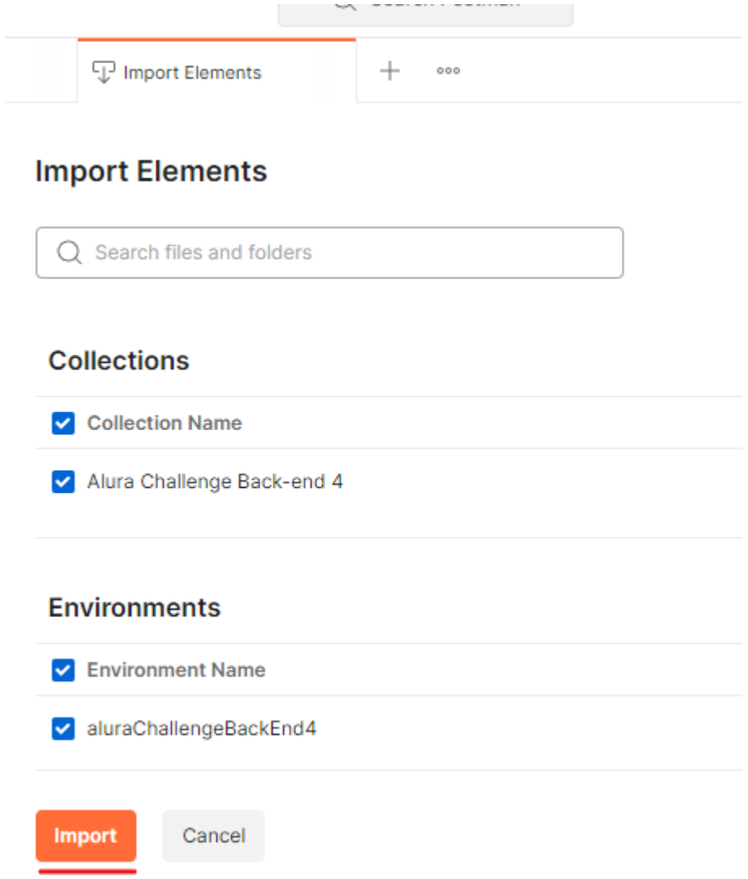
Postman (requisições)

- instalar/abrir o Postman
- se não existir nenhum workspace, criar uma

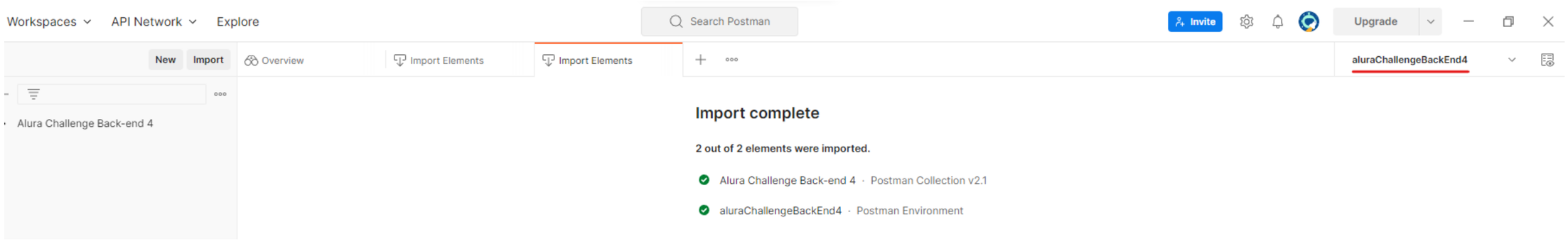


- importar os arquivos “Alura Challenge Back-end 4.postman_collection.json” e “aluraChallengeBackEnd4.postman_environment.json” (pasta postman do projeto)

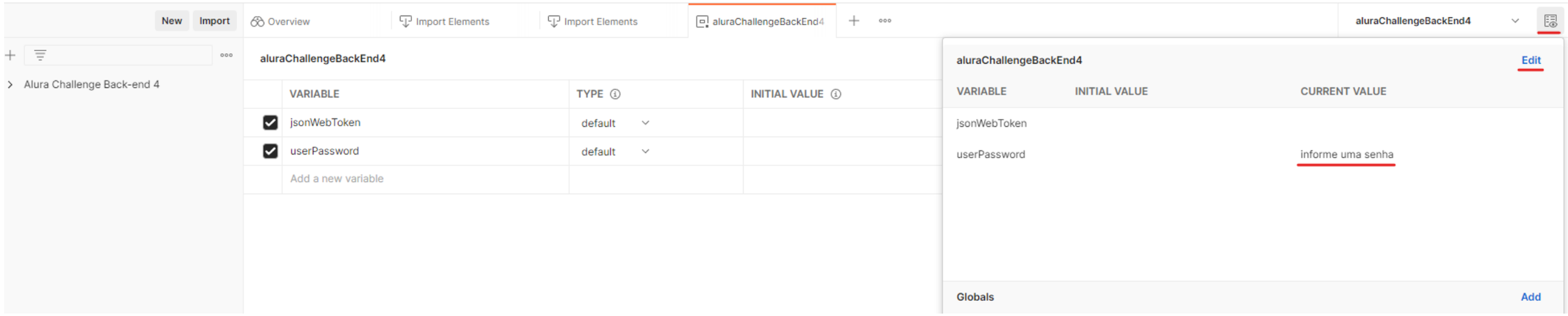




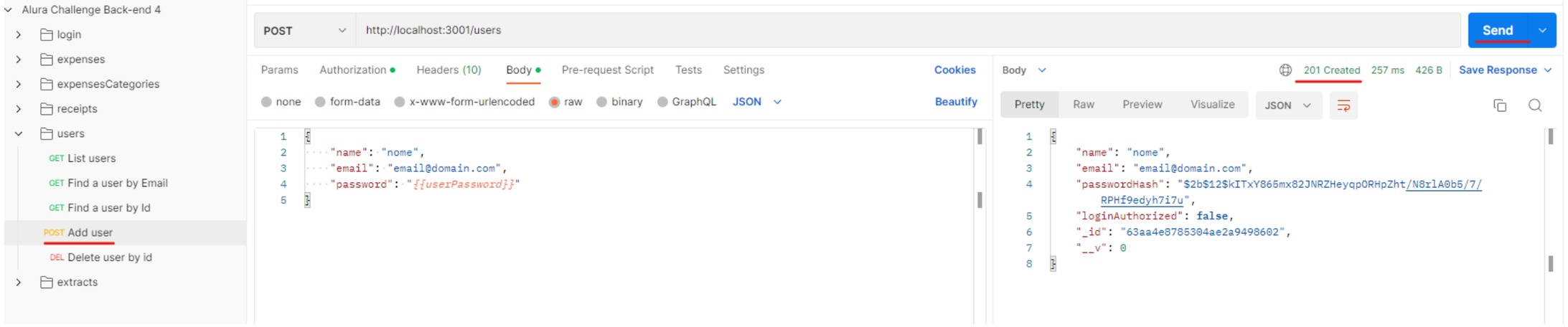
- Selecionar o environment (variáveis de ambiente do programa)



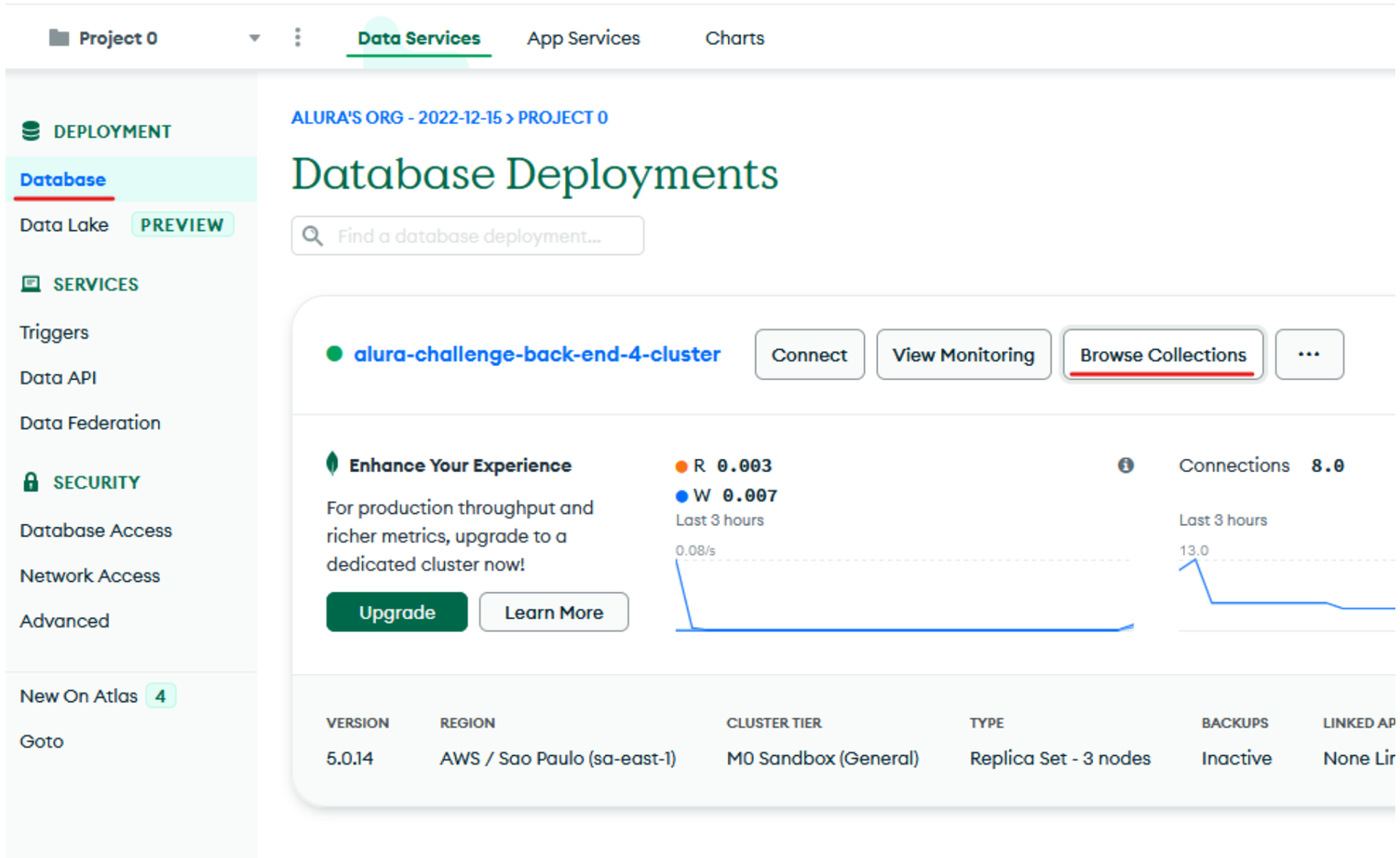
- Editar o environment, informando uma senha no respectivo campo + botão Salvar



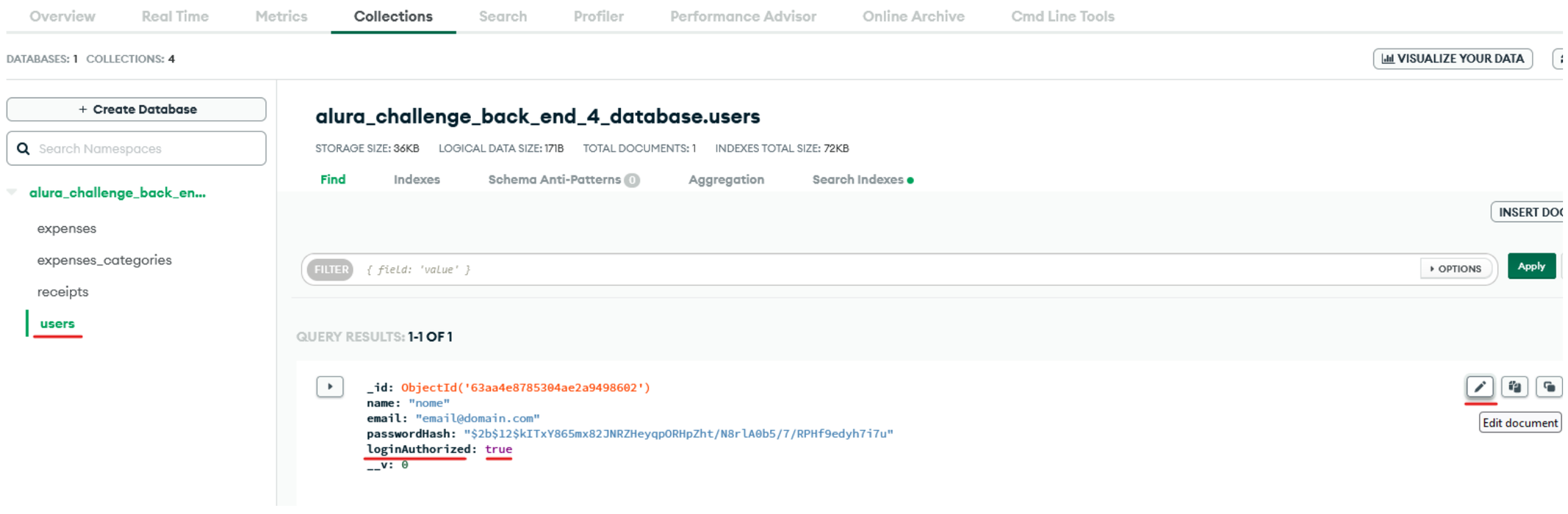
- Executar a rota de cadastro de usuário (se execução com sucesso, o status/resposta será “201 Created”)



- Após a criação do usuário com sucesso, na página (site) do MongoDB, acessar as opções Database + “Browse Collections”:

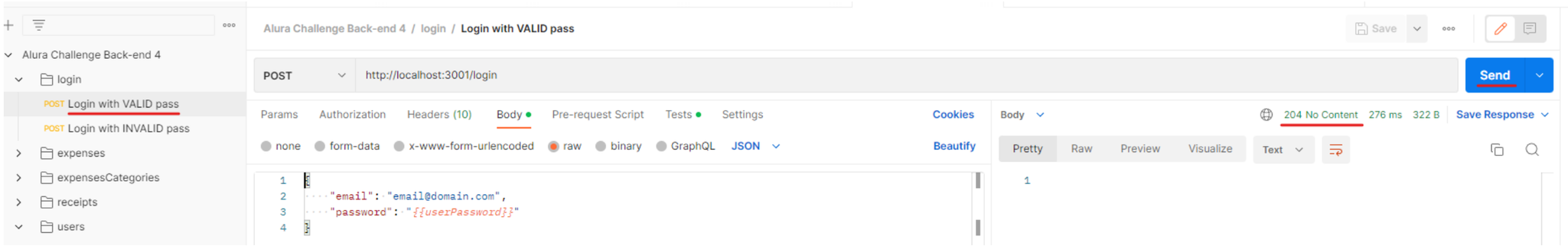


- Clicar na coleção users + botão editar + alterar o valor do campo loginAuthorized para “true” + botão UPDATE

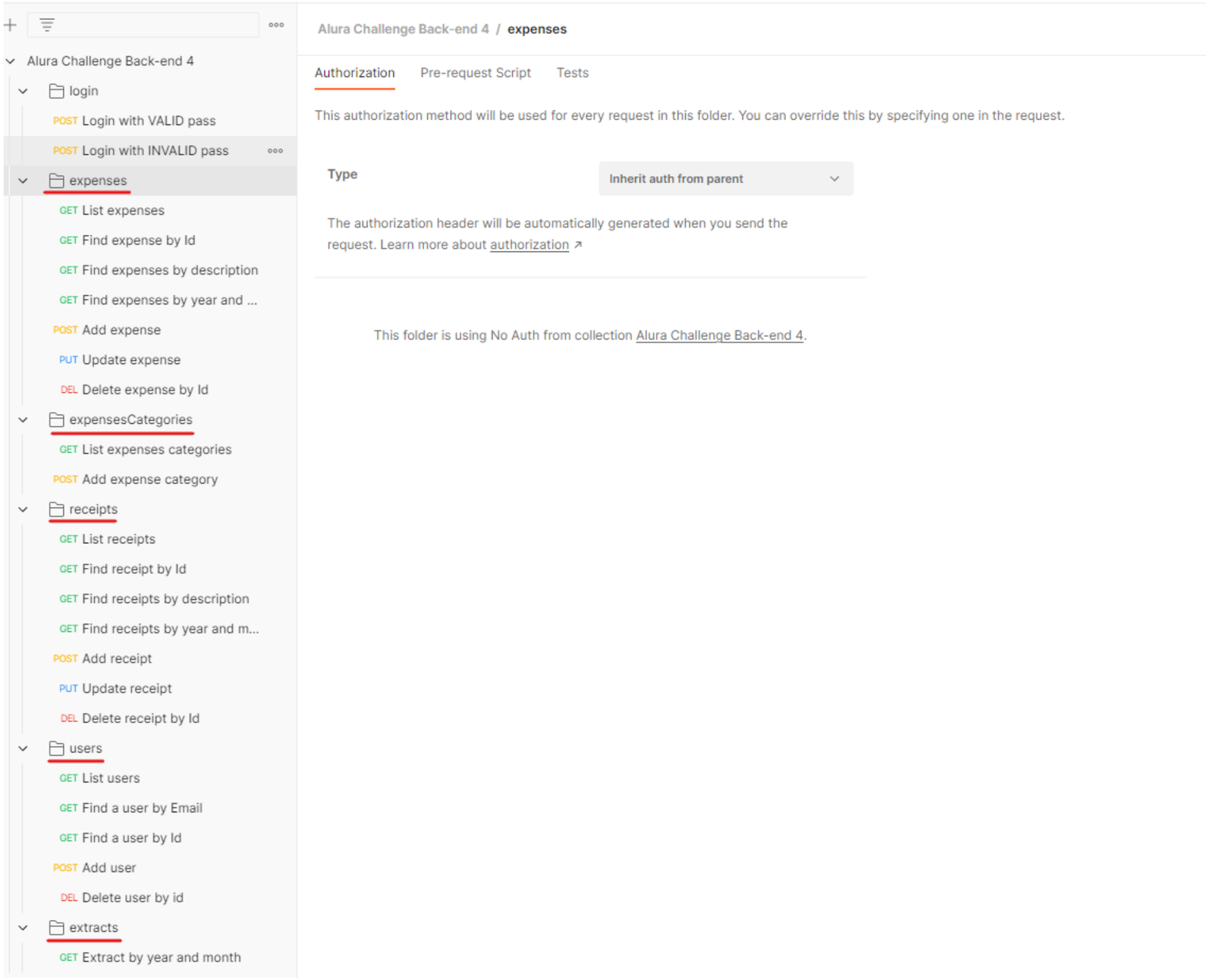


  - sem essa configuração não será possível executar com sucesso as demais rotas/requisições do Postman.

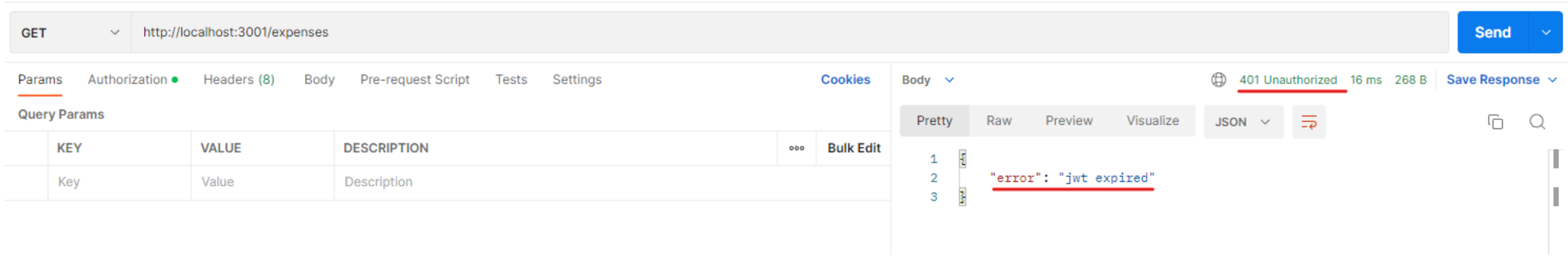
- De volta ao programa Postman, executar a rota “Login with VALID pass” (se execução com sucesso, o status/resposta será “204 No Content”



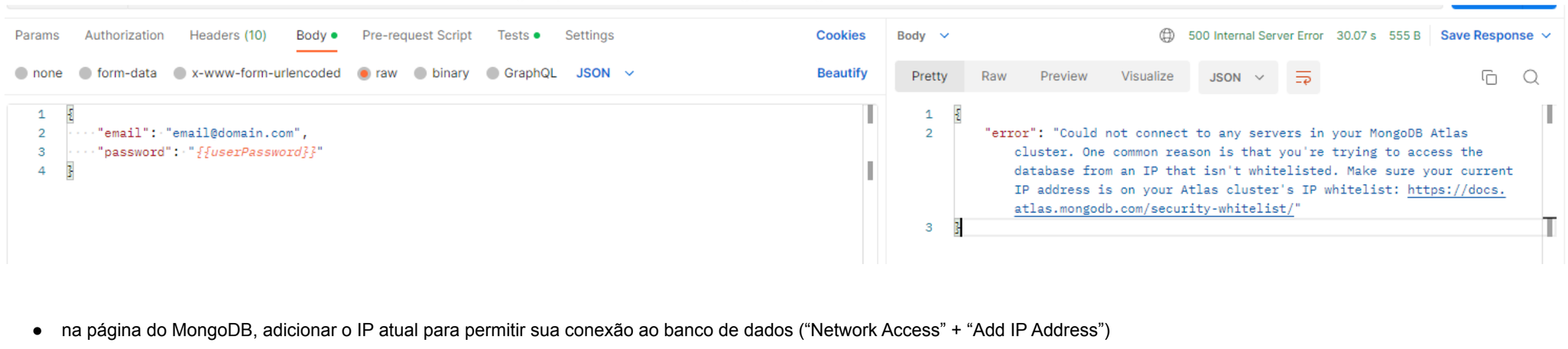
- Executar as demais requisições (essas não tem ordem / obrigatoriedade de execução):



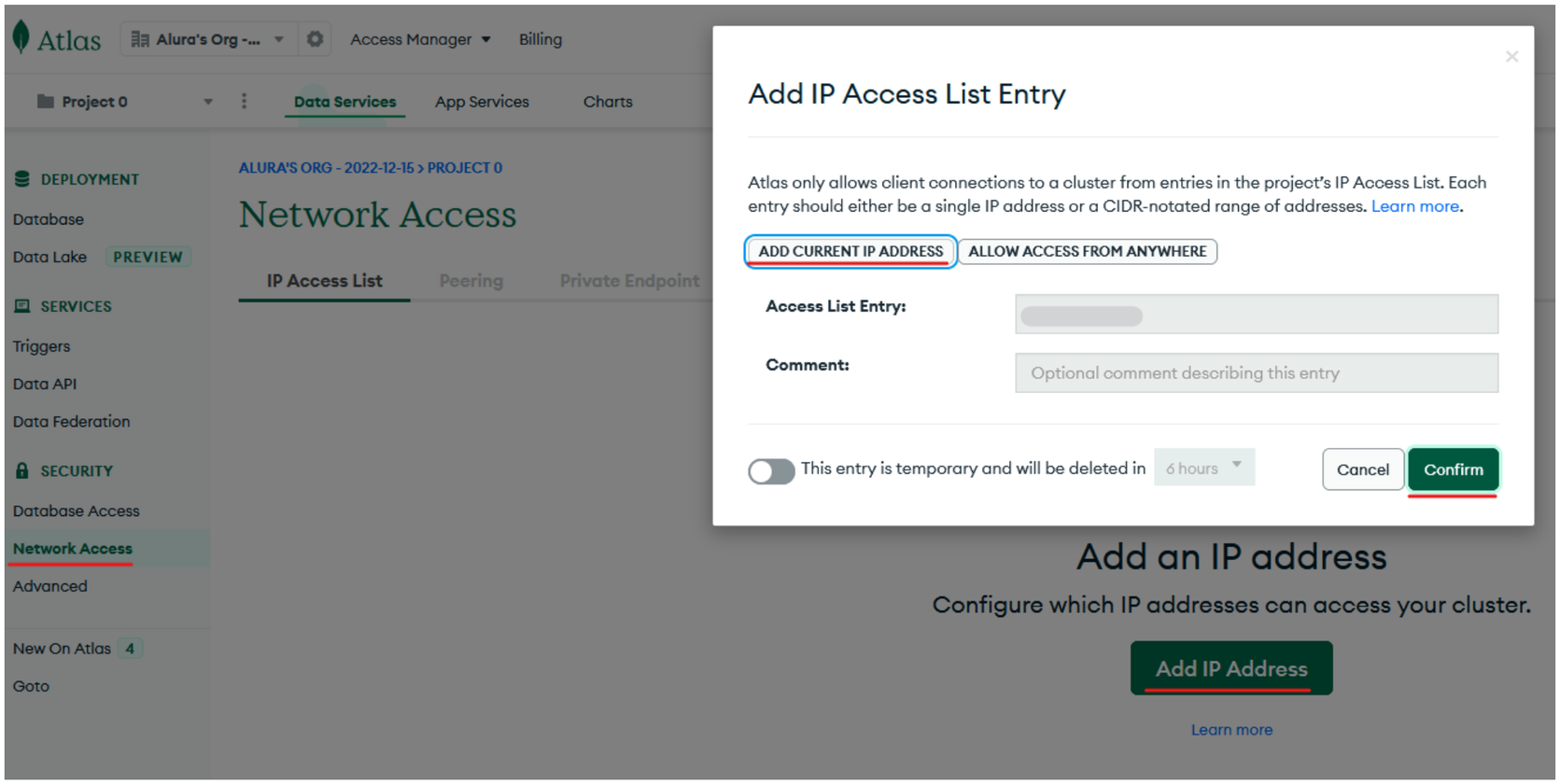
- ***Observações***:
  - Após 15 minutos o usuário deixa de ter permissão de execução das requisições (o status/resposta da requisição será 401 Unauthorized). Para ser possível executá-las novamente, executar a rota “Login with VALID pass” (ver acima)



- Caso ocorra o erro “500 Internal Server Error”, com uma mensagem parecida com a da imagem a seguir,



- na página do MongoDB, adicionar o IP atual para permitir sua conexão ao banco de dados (“Network Access” + “Add IP Address”)



- pelo terminal (cmd / bash), na pasta raiz do repositório/projeto (ex: alura-challenge-back-end-4):
    - desativar o servidor (teclas Ctrl + C)
    - ativar o servidor novamente (**npm run dev)**