


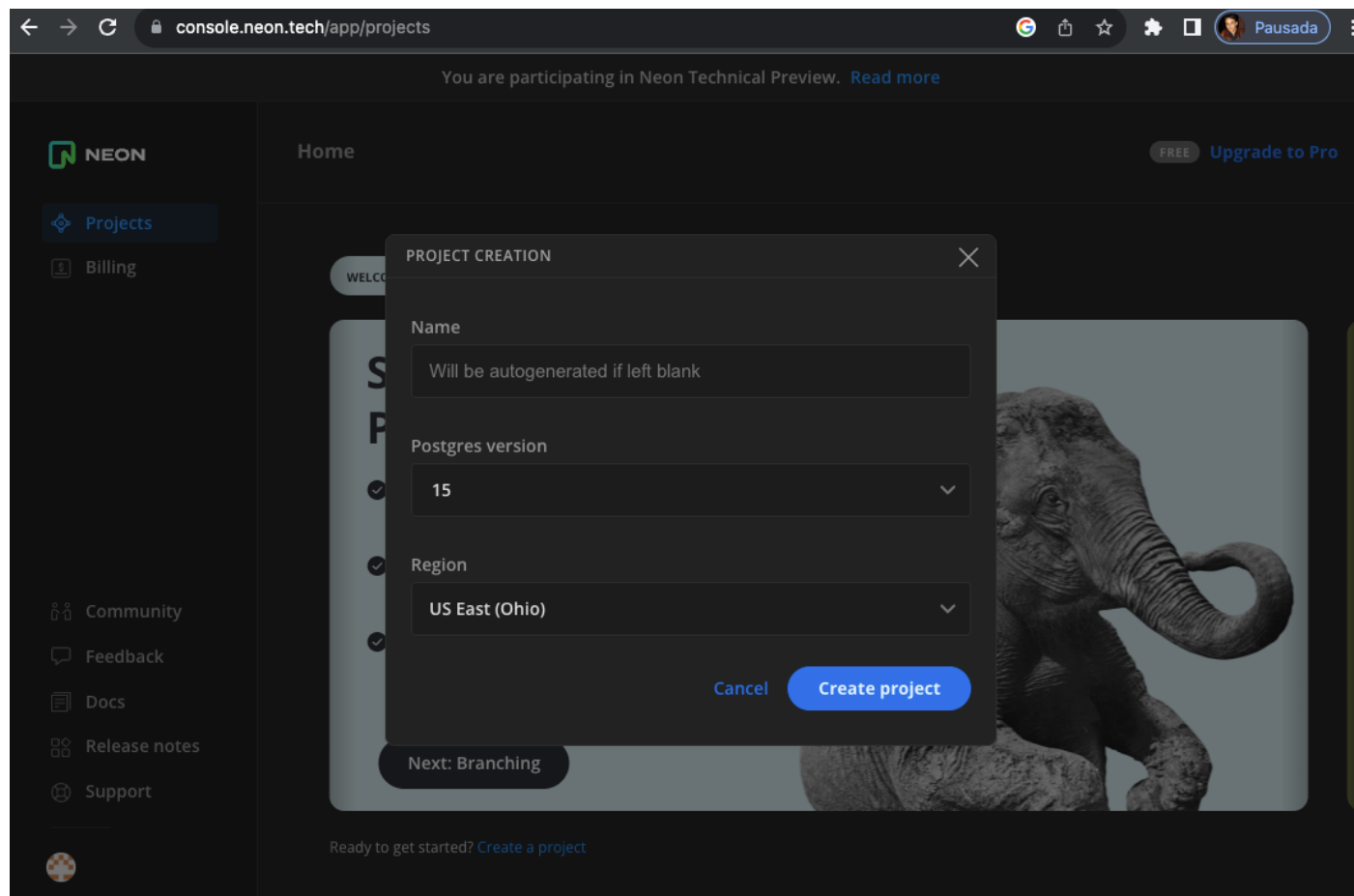
| | | |
|---|---|---|
|  | <p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p> | |
| <p><i>Disciplina</i> Lab. Aplicações Web I</p> | <p align="center">Realizando o deploy da API</p> | <p><i>Professor</i> Lázaro Eduardo da Silva</p> |

Nessa aula, vamos realizar o deploy da API online.

Para isso, vamos criar o banco de dados do projeto no servidor NEON, acesse-o no endereço abaixo e crie sua conta.

<https://neon.tech/>

Após fazer o login, você estará em uma tela como a abaixo, clique em **Create a Project** que uma janela modal será apresentada conforme imagem abaixo. Preencha os campos e clique em **Create a Project**.



A aplicação irá apresentar uma string com a abaixo composta pelas partes


`postgres://USUÁRIO:SENHA@HOST/NOME_DO_BANCO_DE_DADOS`

Vamos utilizar estes dados nas variáveis de ambiente para que a nossa aplicação se conecte neste banco de dados. Guarde estes dados e vamos fazer o deploy da aplicação.

Antes de realizarmos o deploy da API, vamos fazer algumas configurações necessárias. A primeira é no arquivo que fica na pasta config arquivo cors.ts. Na linha 23 a configuração `enabled` deve estar `false`, troque para `true`.

```
23 | // enabled: true,
```

Ainda na pasta config, abra o arquivo `database.ts` e encontre a região onde o postgres é configurado e acrescente a configuração `ssl rejectUnauthorized: false`, para que nossa aplicação consiga conectar ao banco de dados com criptografia SSL.

| | | |
|---|---|---|
|  | <p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p> | |
| <p><i>Disciplina</i> Lab. Aplicações Web I</p> | <p align="center">Realizando o deploy da API</p> | <p><i>Professor</i> Lázaro Eduardo da Silva</p> |

```

36 pg: {
37   client: 'pg',
38   connection: {
39     host: Env.get('PG_HOST'),
40     port: Env.get('PG_PORT'),
41     user: Env.get('PG_USER'),
42     password: Env.get('PG_PASSWORD', ''),
43     database: Env.get('PG_DB_NAME'),
44     ssl: {
45       rejectUnauthorized: false
46     }
47   },
48   migrations: {
49     naturalSort: true,
50   },
51   healthCheck: false,
52   debug: false,
53 },

```

Acrescente o package-lock.json no seu arquivo .gitignore para que o servidor instale os pacotes e crie este arquivo online.

Envie suas alterações ao seu repositório no github e vamos criar a conta no servidor render.

Para isso, acesse o endereço abaixo e crie sua conta. Sugiro que crie sua conta conectando via GitHub

<https://render.com/>

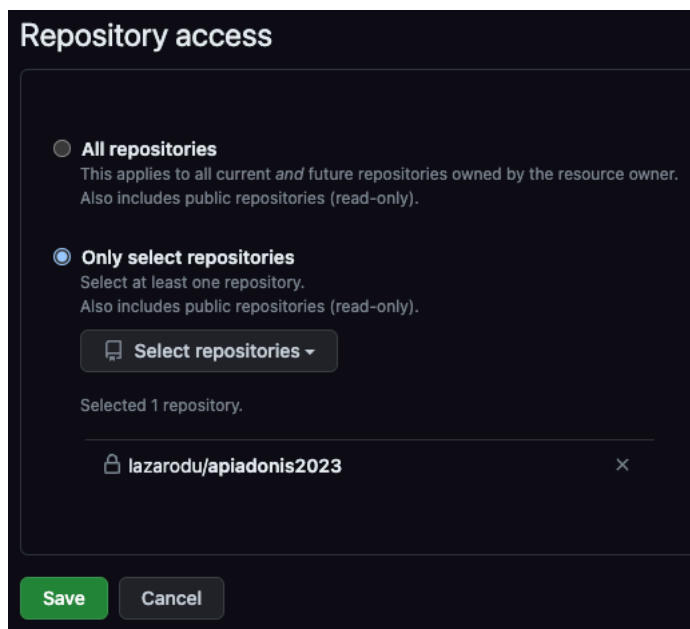
Ao fazer o login clique em New + e escolha a opção Web Service.


O próximo passo é configurar a permissão de acesso aos seus repositórios no github.

Para isso, clique em configure account.




Na tela que abrir você pode configurar somente o que deseja conectar, ou permitir o acesso a todos os repositórios.



| | | |
|---|---|---|
|  | <p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p> | |
| Disciplina Lab. Aplicações Web I | Realizando o deploy da API | Professor Lázaro Eduardo da Silva |

Ele deve aparecer na tela do render, clique em Connect.

Connect a repository


lazarodu / apiadonis2023
7 minutes ago

Connect

Preencha os campos Name, com o nome do seu projeto. Acrescente no Start Command o yarn migrate para que a api suba as tabelas no Neon.

Build Command
 This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.


Start Command
 This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

Clique em Advanced e configure as variáveis de ambiente.

render

Dashboard
Blueprints
Env Groups
Docs
Community
Help

New +


lazarodu@gmail.com

Events

Logs

Disks

Environment

Shell

PRs

Jobs

Metrics

Scaling

Settings

Key

Value

APP_KEY

DB_CONNECTION

DRIVE_DISK

HOST

NODE_ENV

PG_DB_NAME

PG_HOST

PG_PASSWORD

PG_PORT

PG_USER

PORT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....


Create Environment Group

Add Environment Variable

Save Changes

As variáveis do banco de dados, configure com os dados do NEON de acordo com a string de conexão que o servidor apresentou para você. A variável PORT deve receber o valor 443 que é a porta do HTTPS.

As demais variáveis preencha conforme o arquivo .env da sua aplicação.

| | | |
|---|---|---|
|  | <p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p> | |
| <p><i>Disciplina</i> Lab. Aplicações Web I</p> | <p align="center">Realizando o deploy da API</p> | <p><i>Professor</i> Lázaro Eduardo da Silva</p> |

E clique em Create Web Service.

Acompanhe o deploy e verifique se tudo ocorreu conforme o esperado.

August 13, 2023 at 9:40 PM

Live

6e6bc18 Configurações para Deploy

Search logs

Search

Maximize

Scroll to top

```

Aug 13 09:41:40 PM $ node ace migration:run
Aug 13 09:42:00 PM Your service is live 🎉
Aug 13 09:41:49 PM [ info ] Upgrading migrations version from "1" to "2"
> migrated database/migrations/1690573768918_users
> migrated database/migrations/1690573768920_api_tokens
Aug 13 09:41:51 PM
Aug 13 09:41:51 PM Migrated in 3.33 s
Aug 13 09:41:51 PM Done in 10.59s.
Aug 13 09:41:52 PM yarn run v1.22.5
Aug 13 09:41:52 PM $ node build/server.js
Aug 13 09:41:56 PM [00:41:56.283] INFO (apiadonisjs/107): started server on 0.0.0.0:443

```

Caso ocorra tudo bem, sua API está pronta para ser consumida pelo seu aplicativo no endereço:
https://NOME_DO_SE_PROJETO.onrender.com

Altere o endereço no seu arquivo de testes do projeto e teste a criação do usuário e a autenticação.

api.rest

POST /register

```

1 ##### Registro do usuário
2 Send Request
3 POST https://apiadonis2023.onrender.com/register HTTP/1.1
4 Content-Type: application/json
5 {
6   "name": "Lázaro",
7   "email": "lazarodu@gmail.com",
8   "password": "1234"
9 }
10
11 ##### Autenticação do usuário
12 Send Request
13 POST https://apiadonis2023.onrender.com/login HTTP/1.1
14 Content-Type: application/json
15 {
16   "email": "lazarodu@gmail.com",
17   "password": "1234"
18 }

```


Response(2633ms)

```

1 HTTP/1.1 200 OK
2 Date: Mon, 14 Aug 2023 00:43:23 GMT
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 124
5 Connection: close
6 CF-Ray: 7f6529c178950170-GRU
7 CF-Cache-Status: DYNAMIC
8 Content-Encoding: gzip
9 Vary: Accept-Encoding
10 x-render-origin-server: Render
11 Server: cloudflare
12 alt-svc: h3=":443"; ma=86400
13
14 {
15   "name": "Lázaro",
16   "email": "lazarodu@gmail.com",
17   "created_at": "2023-08-14T00:43:23.277+00:00",
18   "updated_at": "2023-08-14T00:43:23.277+00:00",
19   "id": 1
20 }

```

Você ainda consegue verificar se tudo ocorreu como esperado observando as tabelas criadas no neon e os registros inseridos na tabela de usuários.



Centro Federal de Educação Tecnológica de Minas Gerais

Campus VIII – Varginha

Curso Técnico em Informática

Disciplina

Lab. Aplicações Web I


Realizando o deploy da API

Professor

Lázaro Eduardo da Silva

← → ↻ console.neon.tech/app/projects/soft-paper-01837703/query

You are participating in Neon Technical Preview. [Read more](#)



Home / mobile / SQL Editor

FREE [Upgrade to Pro](#)

Dashboard

Branches

Tables

SQL Editor

Operations

Settings

Integrations

Roles

Databases

Saved History

Untitled Save

main... Read-write Active neondb

1 select * from users

Run Explain Analyze 1ms 1 row

| # | id | name | email | password |
|---|----|--------|--------------------|--|
| 1 | 1 | Lázaro | lazarodu@gmail.com | \$crypt\$n=16384,r=8,p=1\$9c4KAdD8lUXtzp |

Community

Bom trabalho!