

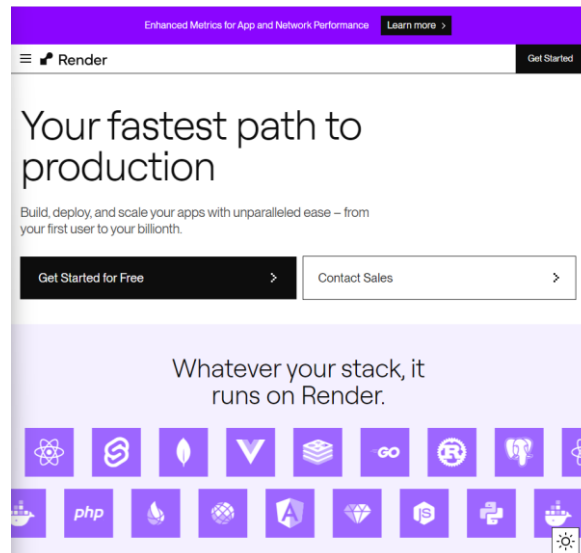
Publicando o projeto via Docker no Render

Requisitos básicos

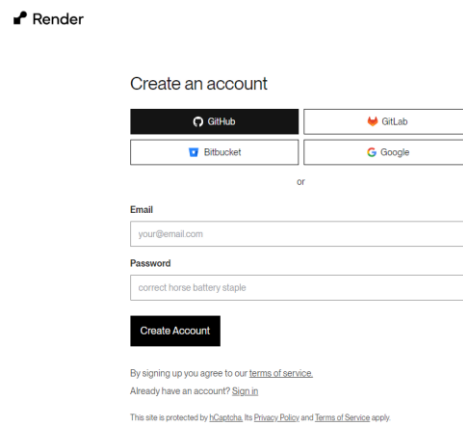
- Projeto publicado no Git.

Passos:

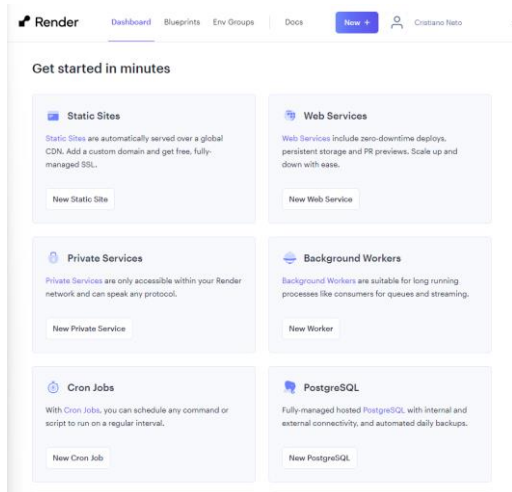
1. Acessar o <https://render.com> e selecionar o **Get Started**.



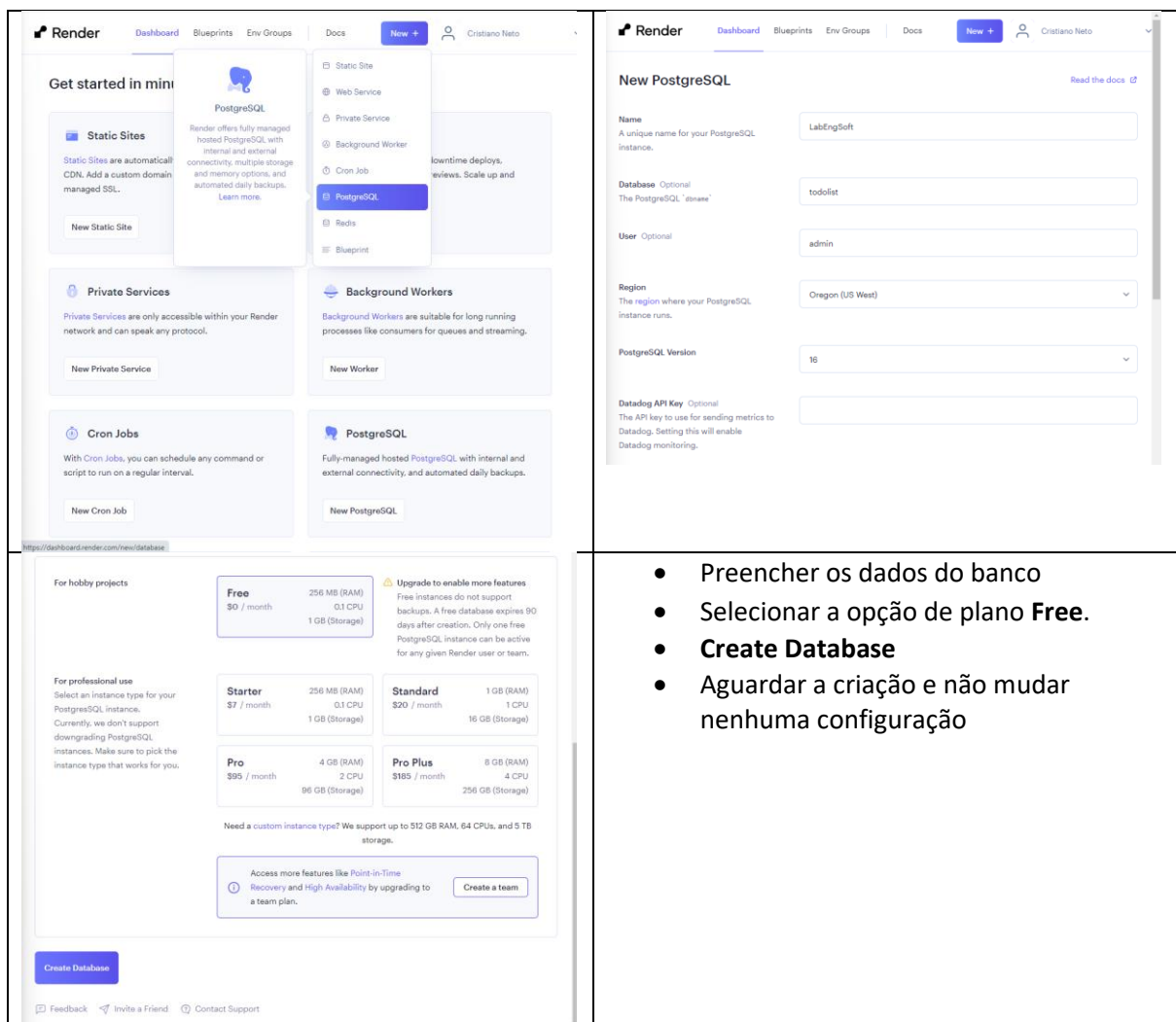
2. Escolher de preferência a opção **GitHub**.



3. Depois de clicar no e-mail de confirmação, você terá acesso ao Dashboard, conforme a seguir.



4. Criar um banco de dados em PostgreSQL



- Preencher os dados do banco
- Selecionar a opção de plano **Free**.
- **Create Database**
- Aguardar a criação e não mudar nenhuma configuração

5. Como o **Render** não suporta Java de forma nativa, devemos criar um docker file para que nossa aplicação possa ser executada via **Docker**.

Dockerfile

```
1 FROM ubuntu:latest AS build
2
3 RUN apt-get update
4 RUN apt-get install openjdk-21-jdk -y
5 COPY . .
6
7 RUN apt-get install maven -y
8 RUN mvn clean install
9
10 FROM openjdk:21-jdk-slim
11
12 EXPOSE 8080
13
14 COPY --from=build target/todo-0.0.1-SNAPSHOT.jar app.jar
15
16 ENTRYPOINT ["java", "-jar", "app.jar"]
17
```

Linha 1: definição da imagem com ubuntu.

Linha 3: atualização do ambiente

Linha 4: instalação do openjdk

Linha 5: cópia do SRC para o container

Linha 7: instalação do maven

Linha 8: solicitação para gerar arquivo jar

Linha 10: escolha da imagem que vai executar nossa aplicação.

Linha 12: expor a porta 8080 para acesso externo.

Linha 14: cópia do .jar para execução.

Linha 16: execução do .jar pela máquina virtual java.

6. Criar agora um web Service com o nosso docker.

Render

Dashboard Blueprints Env Groups Docs New Cristiano Neto

POSTGRES LabEngSoft Free

Web Service

Info Metrics Recovery Logs

Name LabEngSoft Edit

Created 24 minutes ago

Status Available

PostgreSQL Version 16

Region Oregon (US West)

Read Replica Add Read Replica

Storage 4.81% used out of 1.0 GiB

Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

Build and deploy from a Git repository

Deploy an existing image from a registry

Host

Render

Dashboard Blueprints Env Groups Docs New Cristiano Neto

Create a new Web Service

Connect your Git repository or use an existing public repository URL.

GitHub

GitLab

Bitbucket

Connect a repository

Install Render

Where do you want to install Render?

crismacheto

ICEI-PUC-Minas-PMV-ADS

Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

© 2024 GitHub, Inc.

Install Render

Install on your personal account **crismacneto**

for these repositories:

☐ All repositories
This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

☒ Only select repositories
Select at least one repository. Also includes public repositories (read-only).

Select repositories

Selected 1 repository:

crismacneto/todoListSpringBoot

with these permissions:

- ✓ Read access to Dependabot alerts, administration, code, and metadata
- ✓ Read and write access to actions, checks, commit statuses, deployments, environments, issues, pull requests, repository hooks, and workflows

User permissions

Render can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

- ✓ Read access to email addresses

Connect a repository

Search...

crismacneto / todoListSpringBoot • 6 minutes ago

Connect

Render

Dashboard Blueprints Env Groups Docs New Cristiano Nieto

Region Oregon (US West)

Branch main

Root Directory Optional
Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

Runtime Docker

Instance Type

For hobby projects

Free \$0 / month 512 MB (RAM) 0.1 CPU

Upgrade to enable more features
Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these

- Preencher os dados do projeto
- Não esquecer de setar o plano como gratuito
- Preferencialmente escolher a mesma região do banco, que a config é mais simples.

7. O deploy já vai iniciar automaticamente e você pode acompanhar o log pelo dashboard.
8. Agora basta testar seus endpoints, apontando para o endereço apresentado no dashboard.
9. **Desafio:** alterar o banco de dados, de H2 para o PostgreSQL que foi criado no passo 4 deste roteiro.
 - a. Lembre-se que você está utilizando JPA
 - b. Lembre-se que agora vamos precisar criar variáveis de ambiente, para o Docker conseguir visualizar o banco.