PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática

Lucas Sampaio Leite

Deploy

INSTITUTO FEDERAL
Baiano

- Deploy de um sistema web é o processo de colocar uma aplicação pronta para funcionar em um servidor, tornando-a acessível para os usuários pela internet.
- É a etapa em que o sistema sai do ambiente de desenvolvimento e passa a rodar em um ambiente de produção.

Deploy

INSTITUTO FEDERAL
Baiano

- Etapas envolvidas em um deploy:
  - Subir os arquivos para um servidor
  - Configurar o servidor
  - Configurar o banco de dados
  - Definir ambiente de produção
  - Executar a aplicação

Deploy

INSTITUTO FEDERAL
Baiano

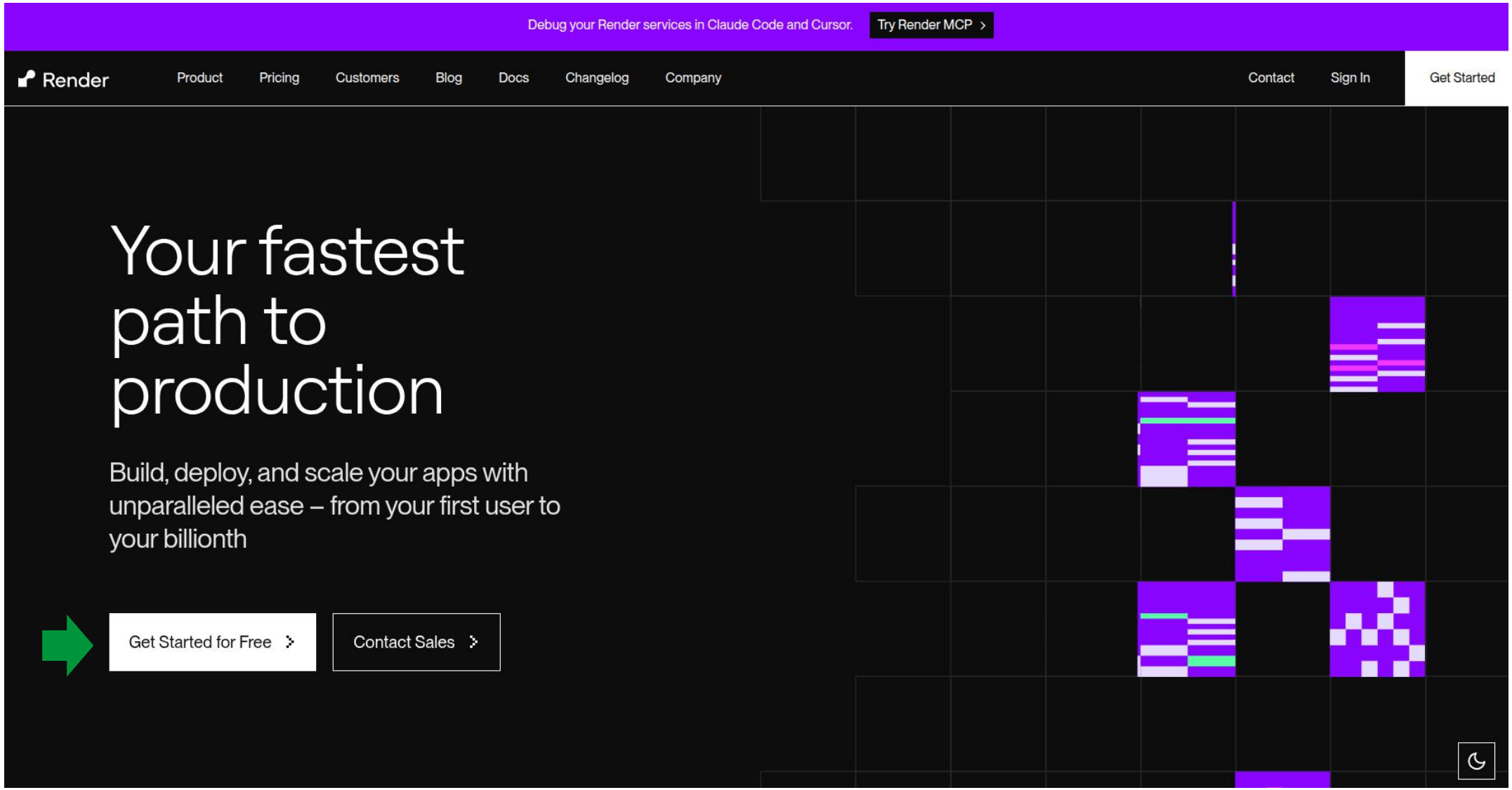
- Onde o deploy pode ser feito:
  - Servidores próprios (VPS: DigitalOcean, Linode, AWS EC2).
  - Hospedagens compartilhadas.
  - Plataformas de deploy automático, exemplo:
    - Heroku
    - Railway
    - Render
    - Vercel (mais comum para front-end)
    - Netlify
    - AWS / Azure / GCP

Plataforma Render

INSTITUTO FEDERAL
Baiano

- O Render é uma plataforma moderna de hospedagem em nuvem (cloud hosting) que permite fazer deploy de aplicações web, APIs, bancos de dados e serviços estáticos de forma simples.
- O Render aceita aplicações desenvolvidas com Flask, permitindo fazer deploy de APIs e sistemas web em poucos passos.

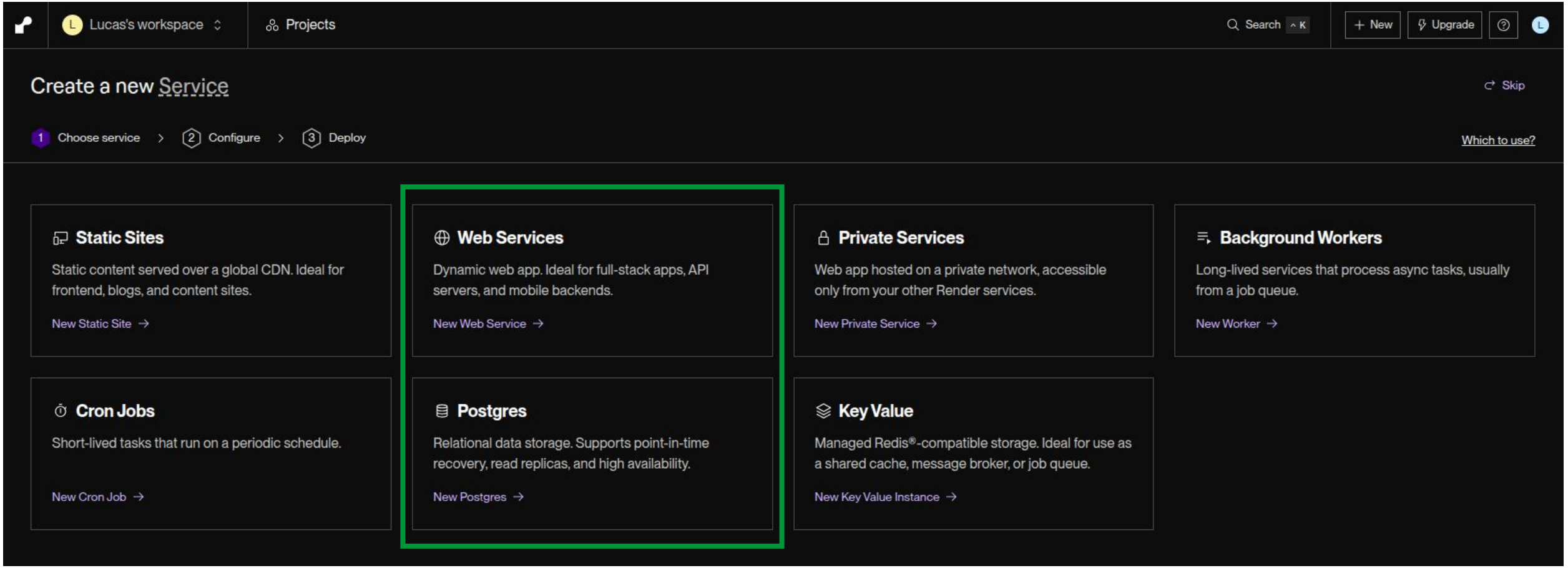
Deploy



https://render.com/

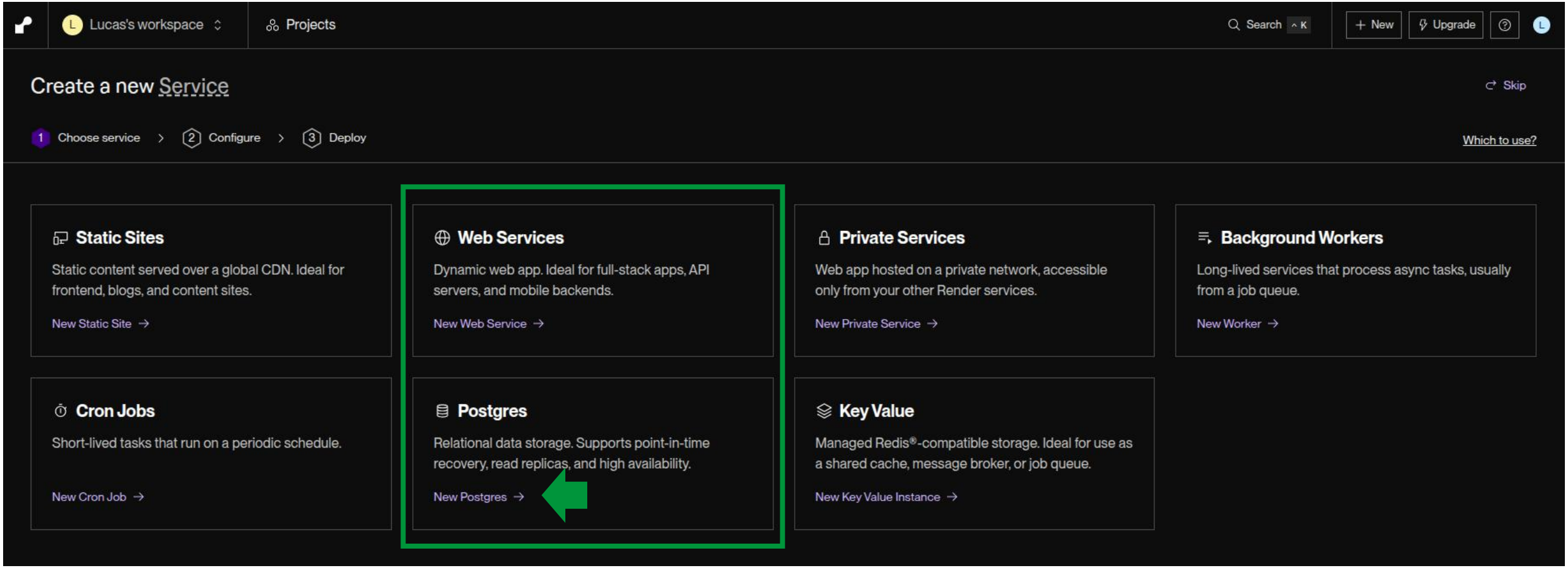
Deploy

INSTITUTO FEDERAL
Baiano



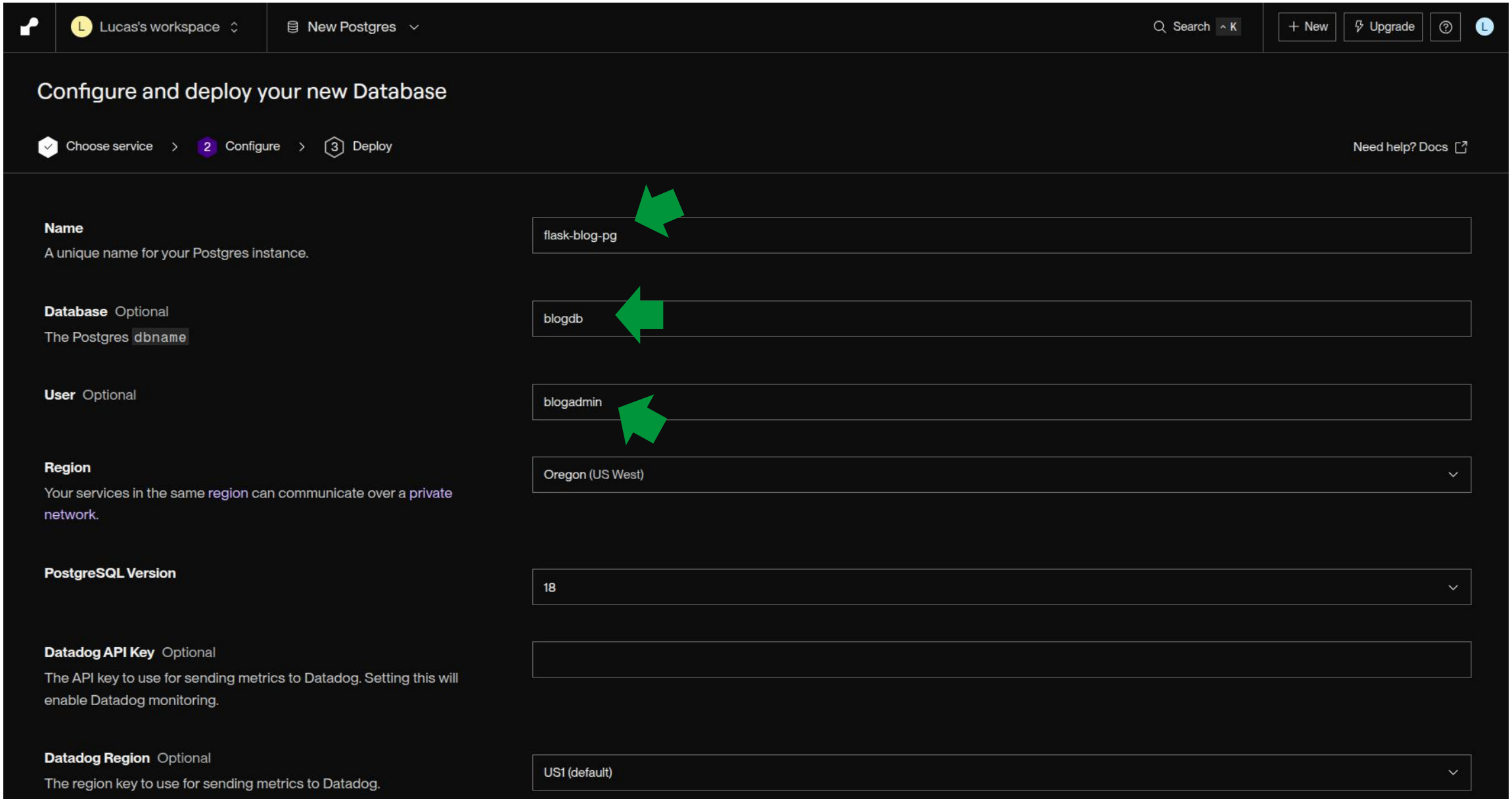
Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

Plan Options

Instance Type

Set your database's RAM and CPU. You can change your instance type later.

Free
For testing out Render Postgres

Free	256 MB (RAM)
$0 / month	0.1 CPU
	1 GB (Storage)

Basic
Reliability and performance for hobby projects. Starting at $6 / month plus storage.

Pro
Perfect for production use cases at any scale. Starting at $55 / month plus storage.

Accelerated
Memory-optimized plans offer significant performance improvements. Starting at $160 / month plus storage.

Need a custom instance type? We support up to 1024 GB RAM, 128 CPUs, and 16 TB storage.

Deploy

INSTITUTO FEDERAL
Baiano

Storage

Your database's capacity, in GB. You can increase storage later, but you can't decrease it.

Specify 1 GB or any multiple of 5 GB.

1 GB $0 / month

Storage Autoscaling

When your database is 90% full, automatically increase its storage by 50% (rounded up to the nearest 5 GB).

Max one increase per 12 hours. Learn more

Disabled $0 / month

High Availability

Run a standby instance of your database and automatically fail over to it if the primary encounters an issue. Learn more

Disabled

Only available for **Pro** instances and higher.

Monthly Total

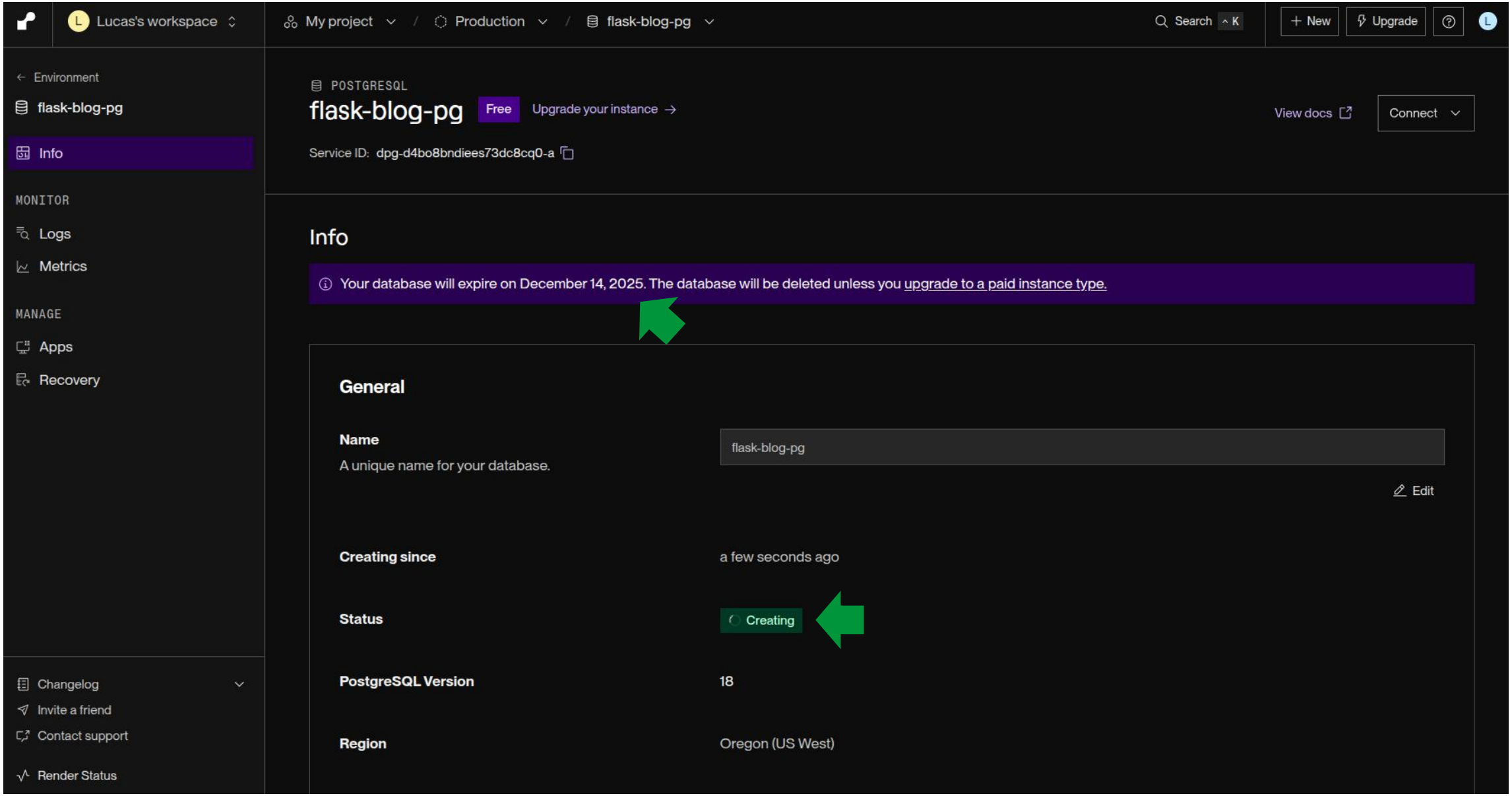
Database instances are billed and prorated by the second.

Instance: **Free**	**$0** / month
Total	**$0** / month

Create Database

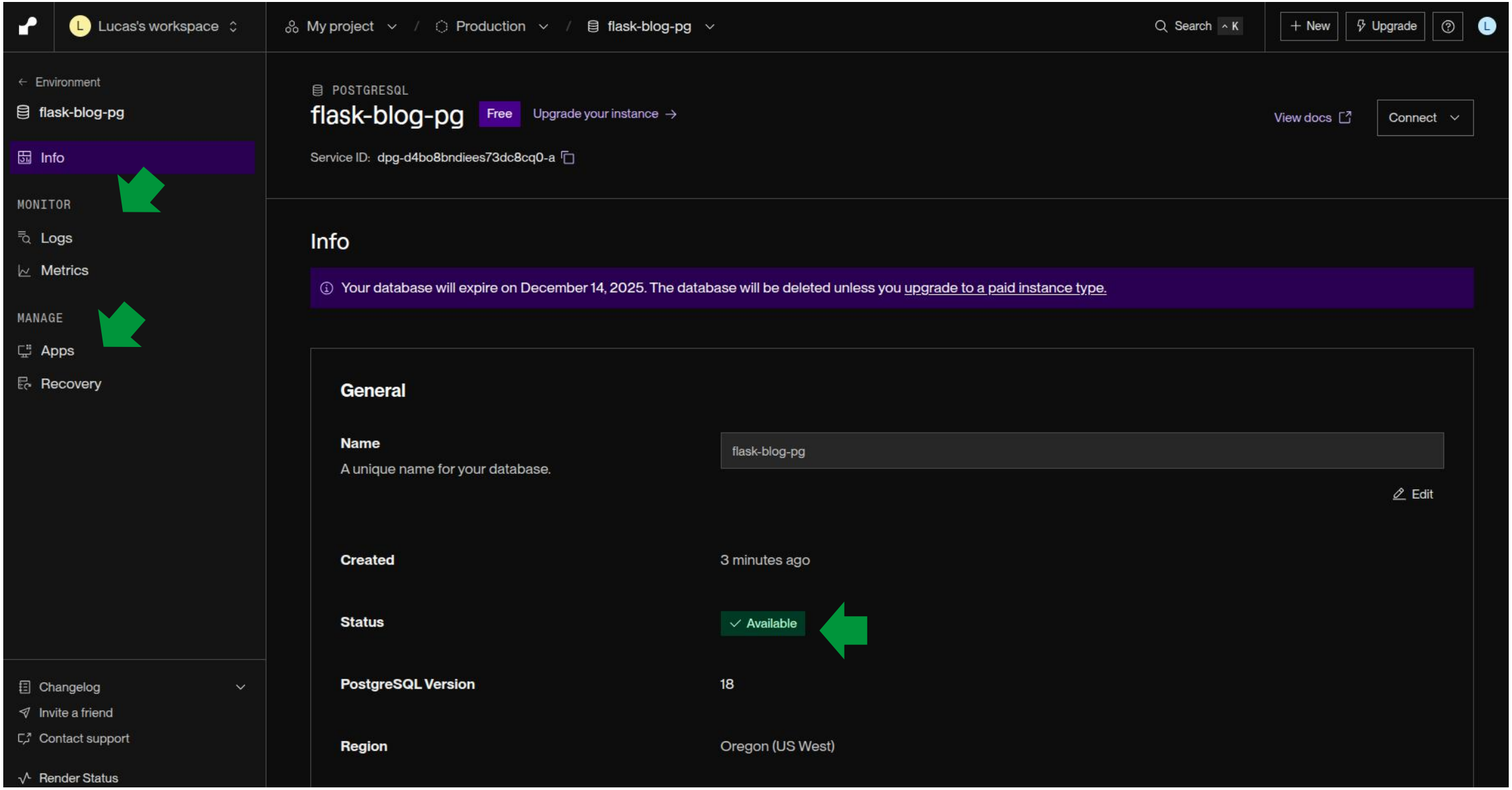
Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

Connections

Hostname
An internal hostname used by your Render services.

dpg-d4csdpggjchc73dk046g-a

Port

5432

Database

blogdb_vudo

Username

blogadmin

Password

Internal Database URL

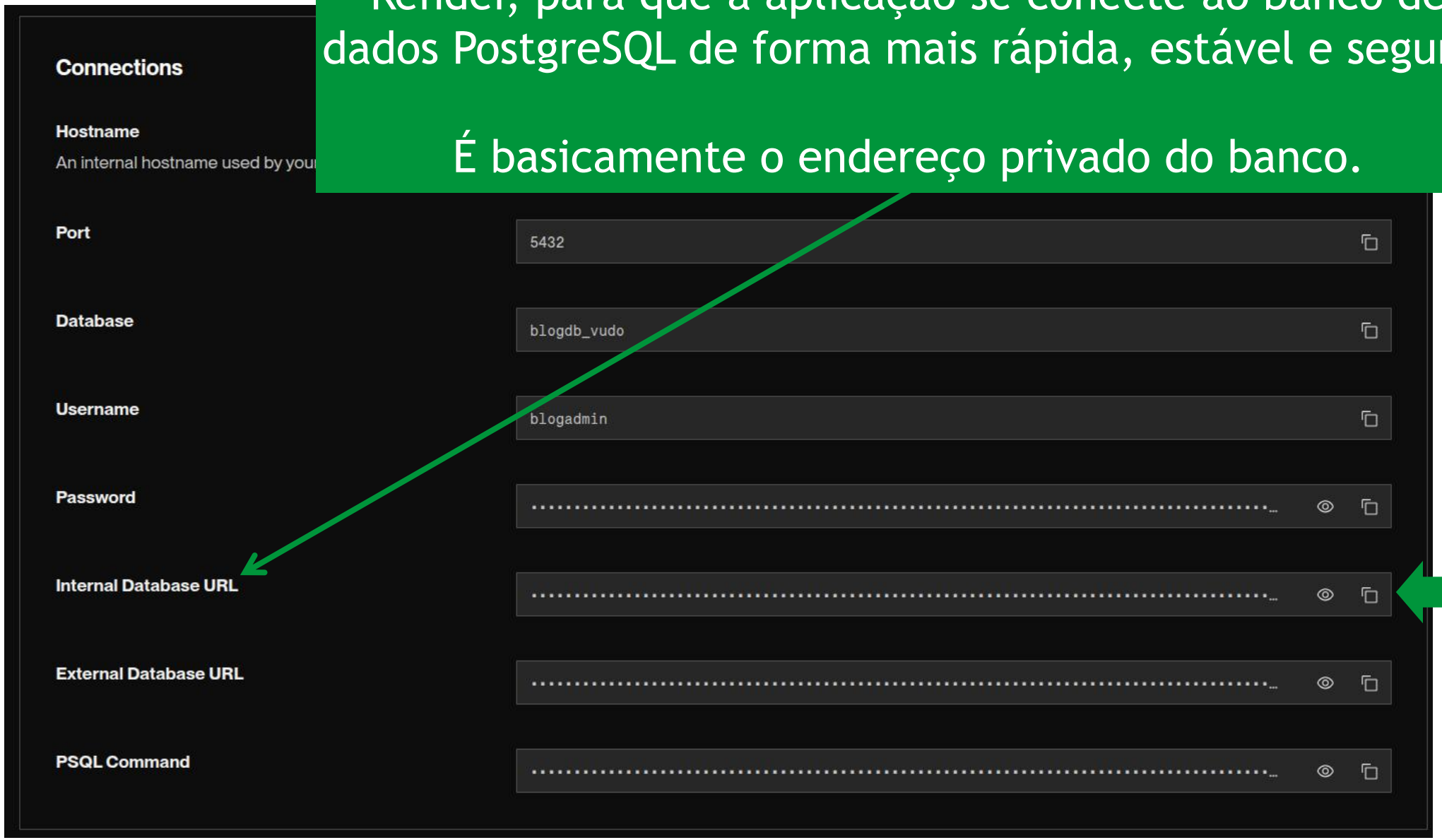
External Database URL

PSQL Command

Deploy

A Internal Database URL do Render é um endereço interno, usado somente dentro da infraestrutura do Render, para que a aplicação se conecte ao banco de dados PostgreSQL de forma mais rápida, estável e segura.

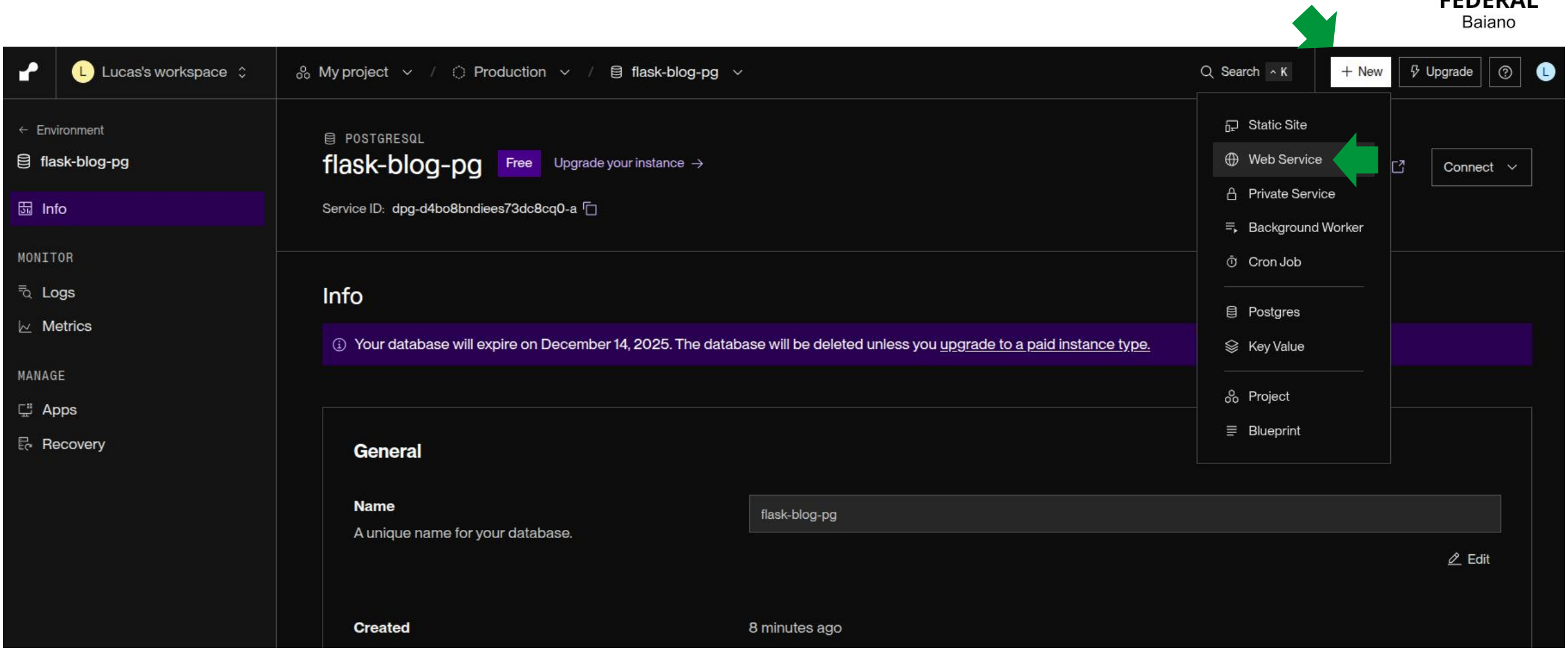
É basicamente o endereço privado do banco.



INSTITUTO FEDERAL
Baiano

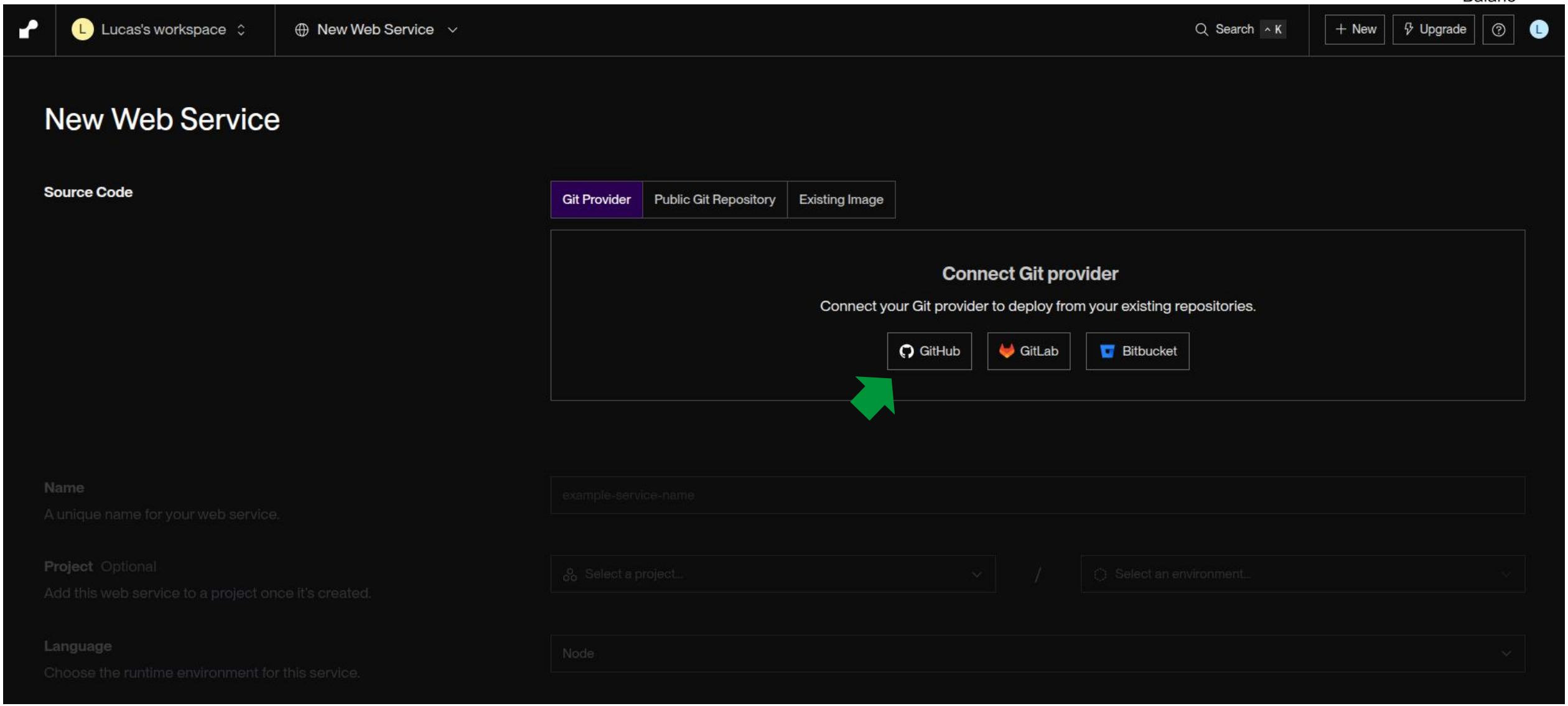
Deploy

INSTITUTO FEDERAL
Baiano



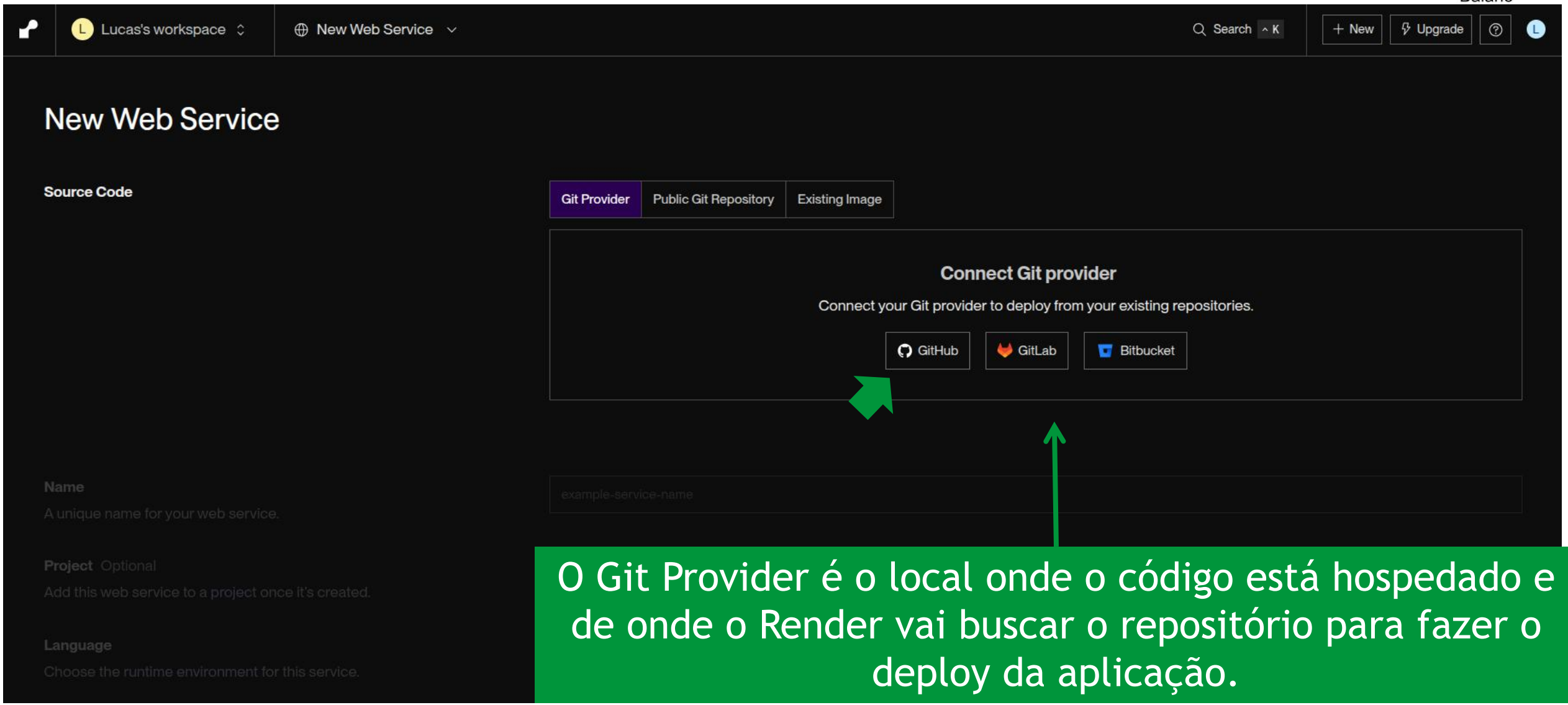
Deploy

INSTITUTO FEDERAL
Baiano



Deploy

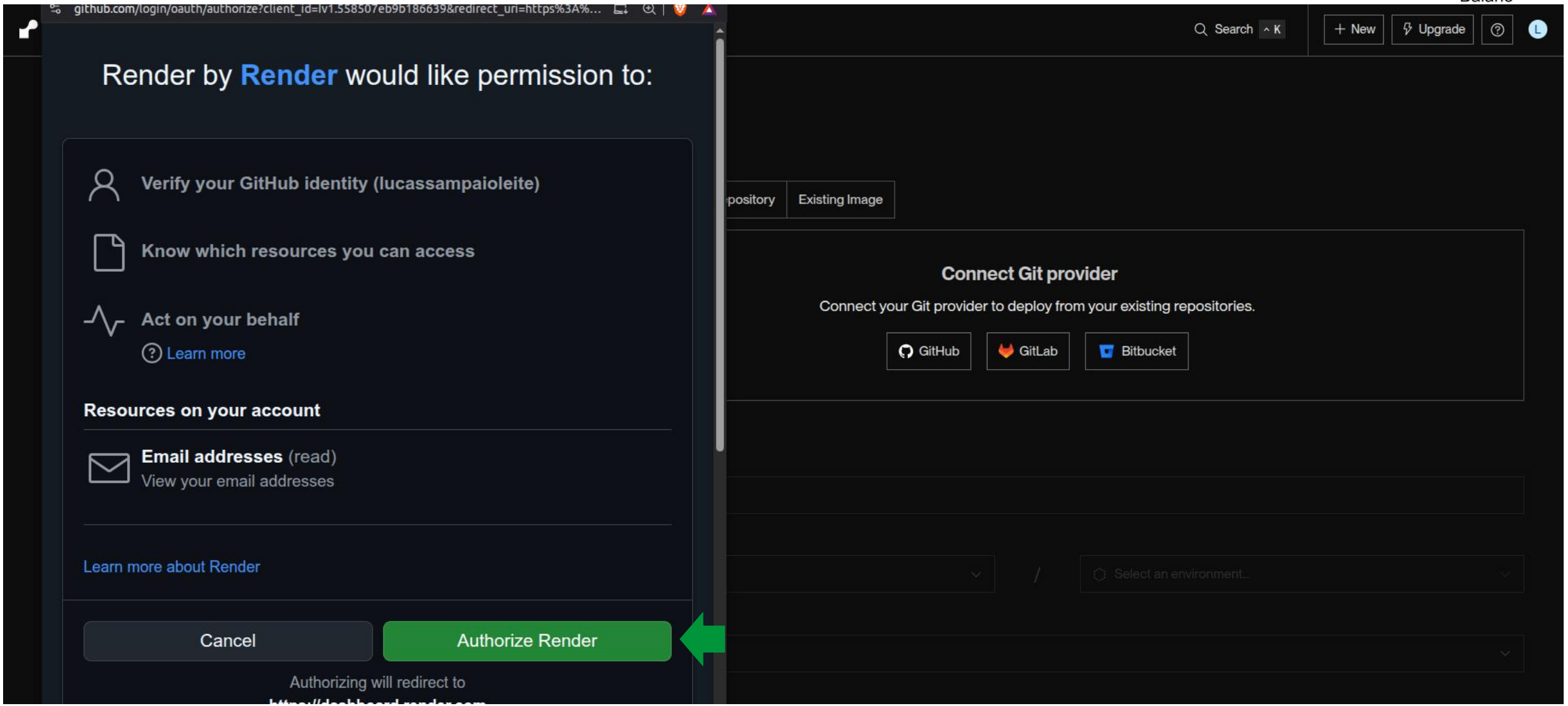
INSTITUTO FEDERAL
Baiano



O Git Provider é o local onde o código está hospedado e de onde o Render vai buscar o repositório para fazer o deploy da aplicação.

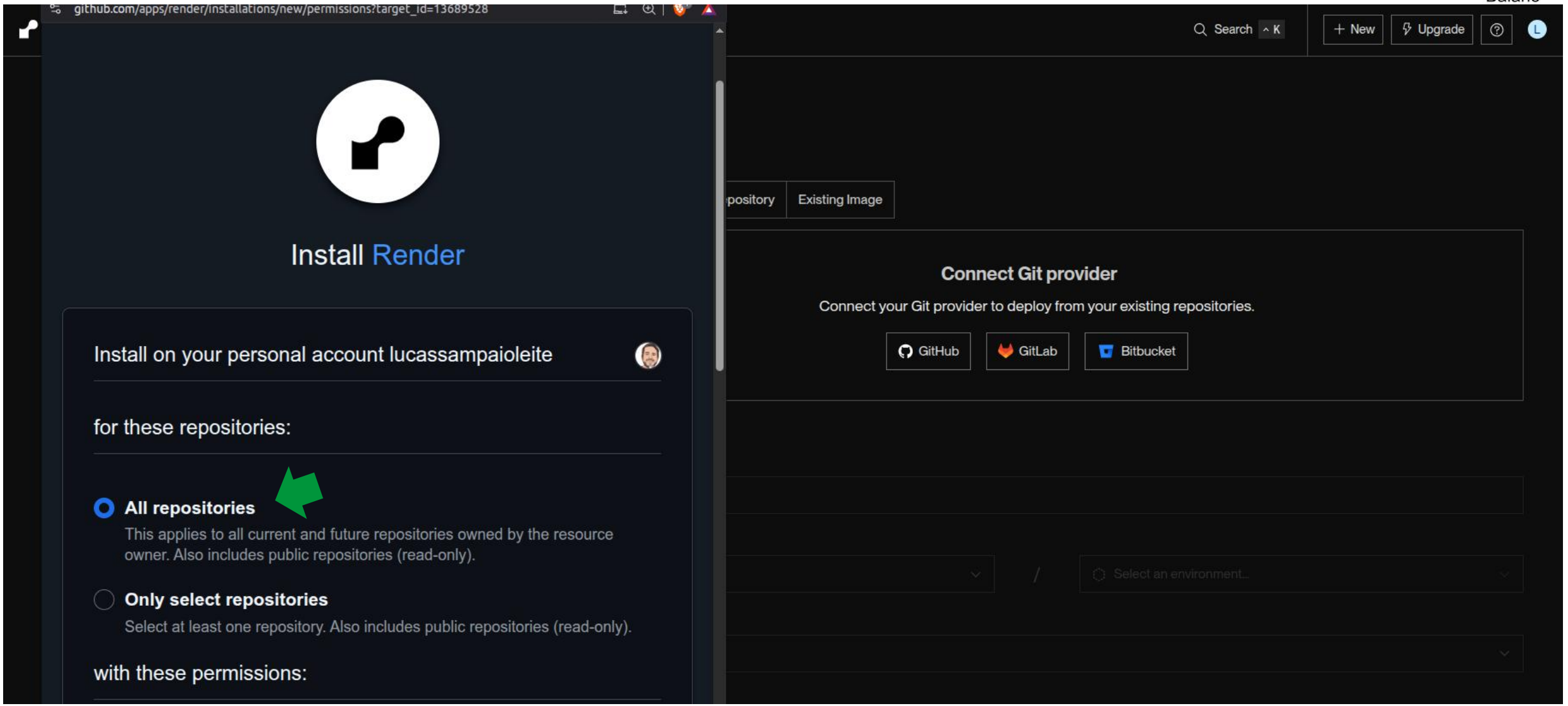
Deploy

INSTITUTO FEDERAL
Baiano



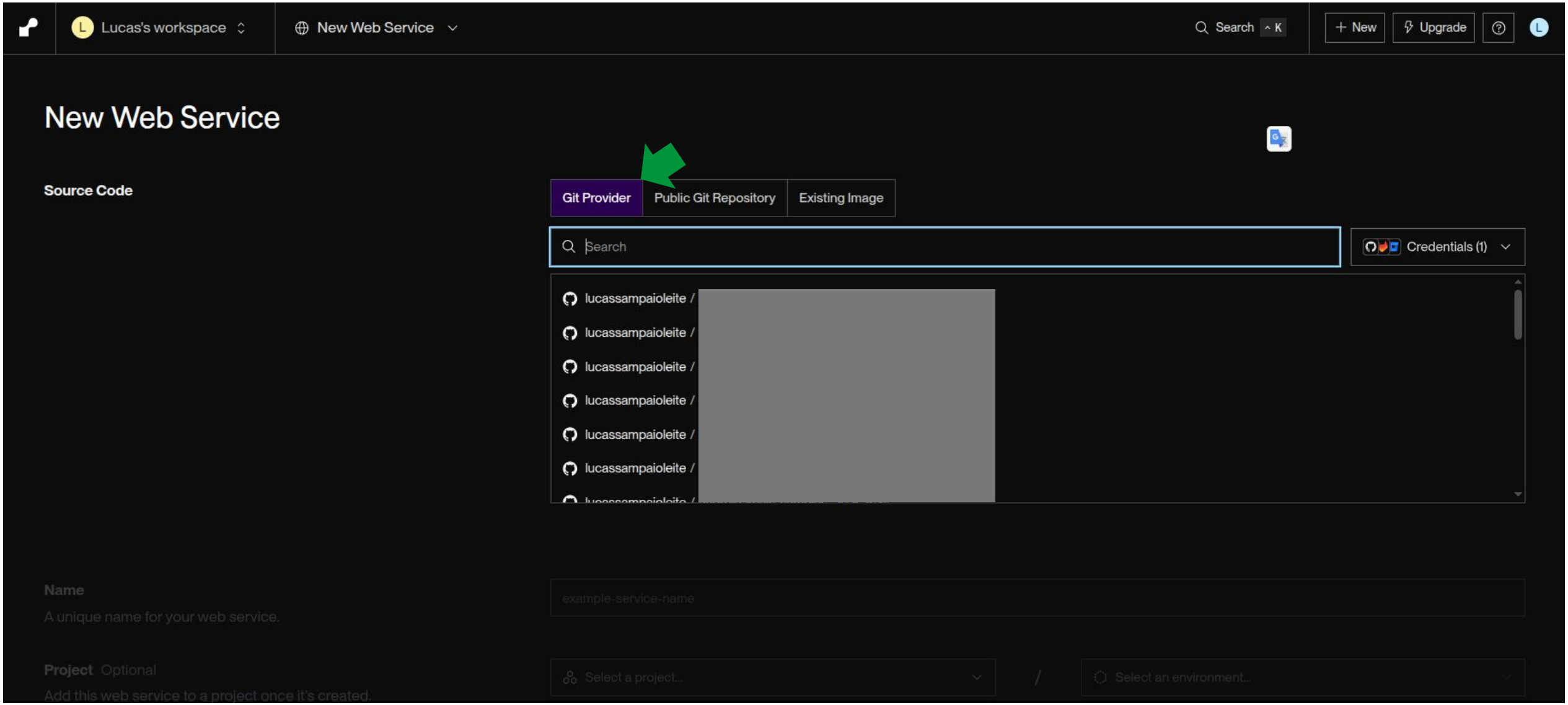
Deploy

INSTITUTO FEDERAL
Baiano

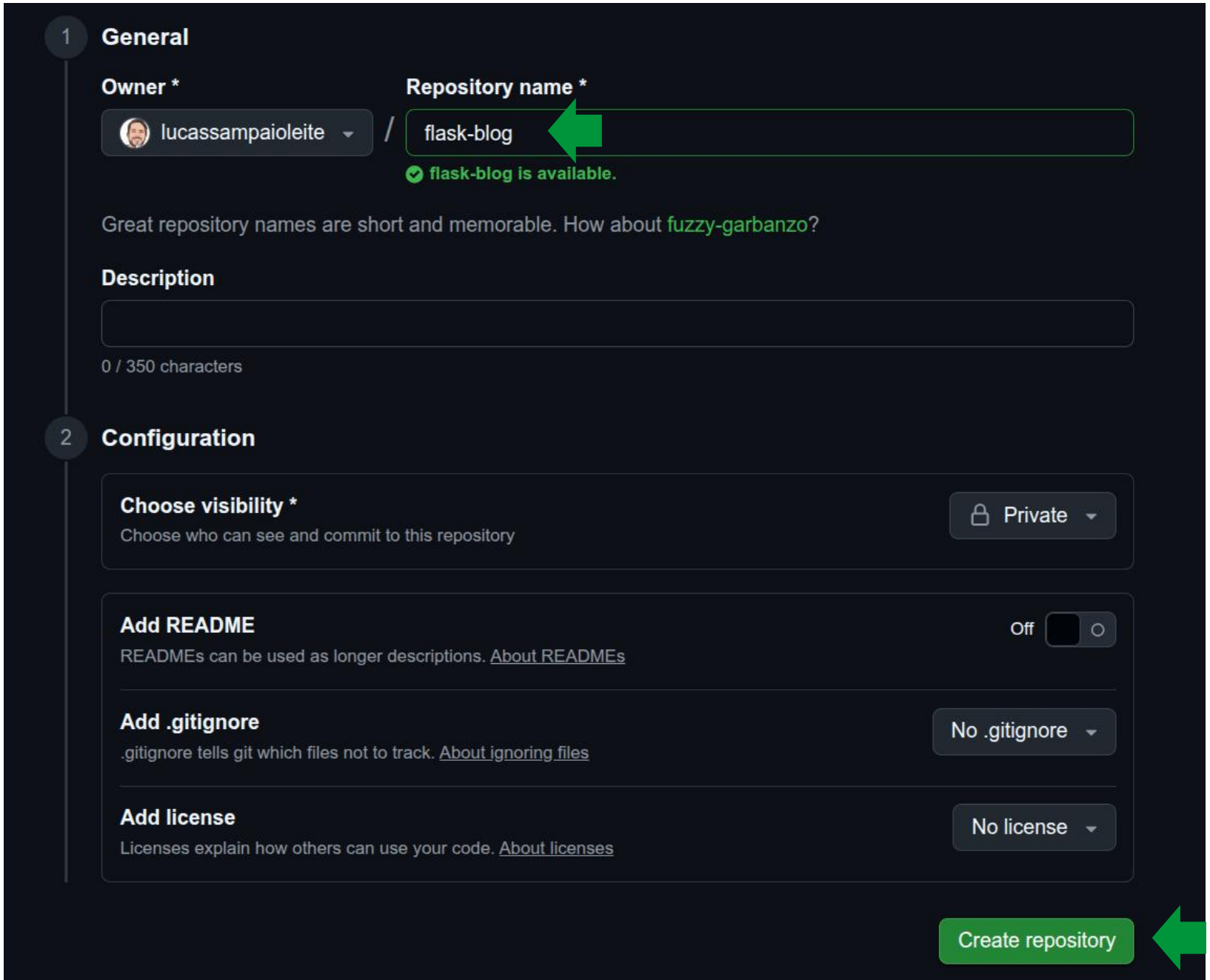


Deploy

INSTITUTO FEDERAL
Baiano

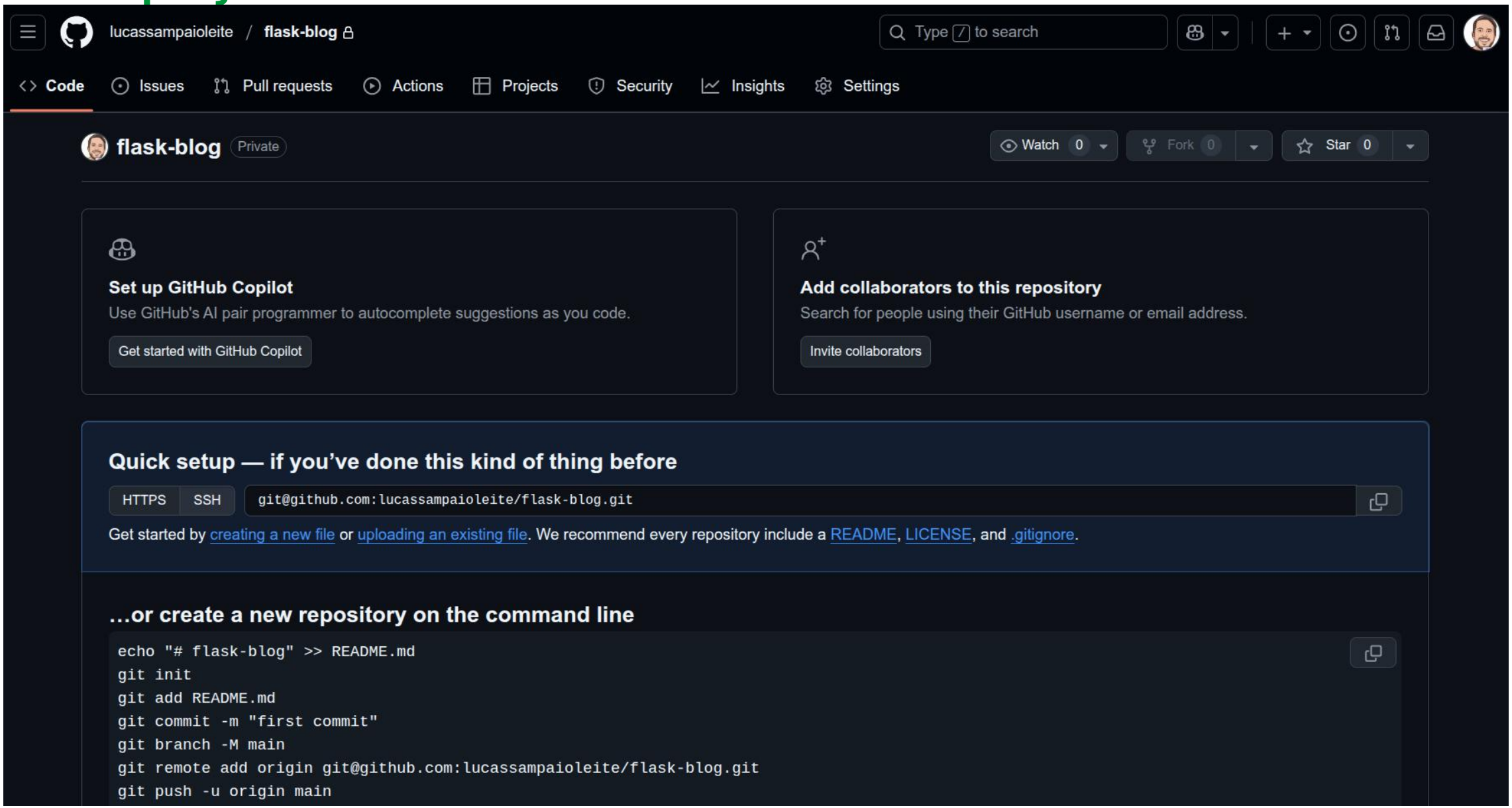


Deploy



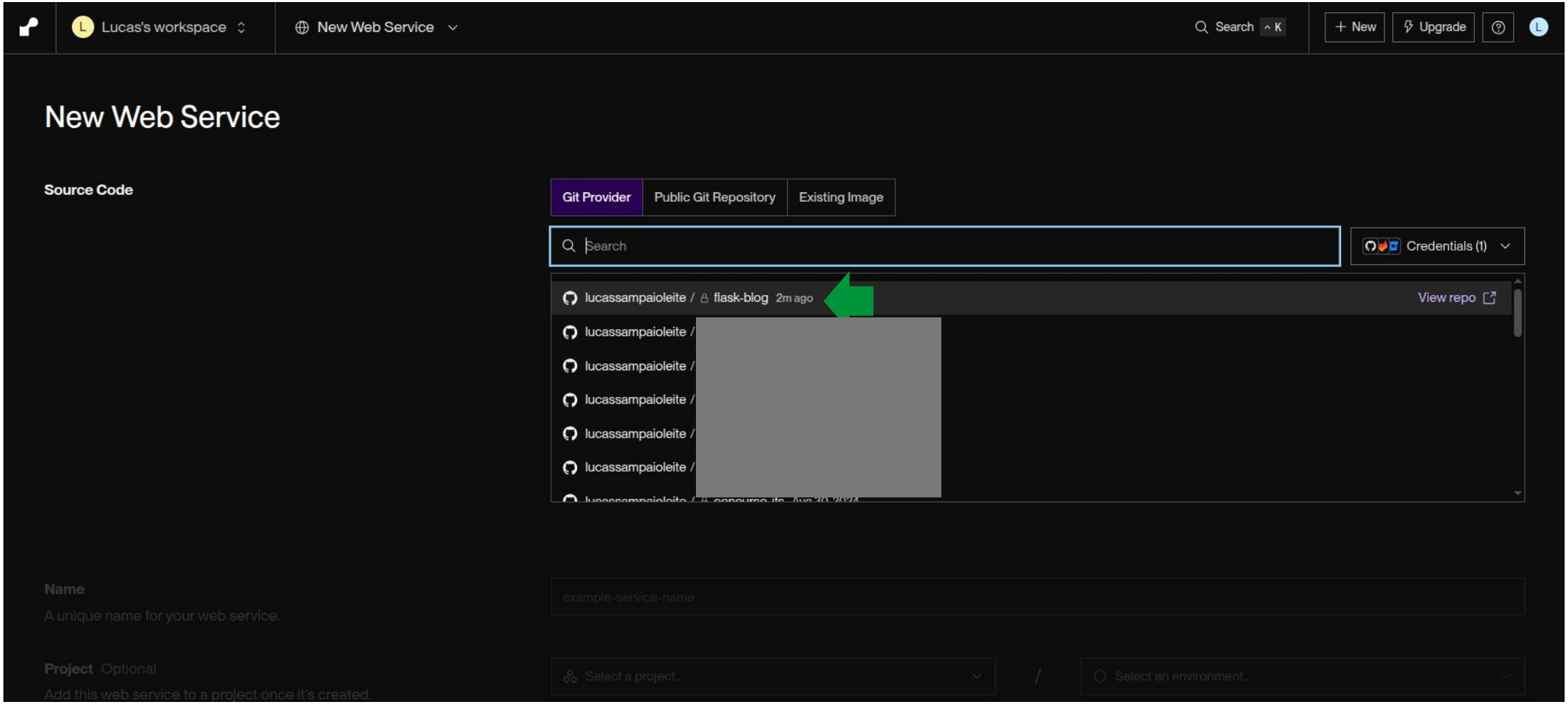
INSTITUTO FEDERAL
Baiano

Deploy


lucassampaioleite / flask-blog
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
flask-blog
Private
Watch 0
Fork 0
Star 0
Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators
Quick setup — if you've done this kind of thing before
HTTPS
SSH
git@github.com:lucassampaioleite/flask-blog.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# flask-blog" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:lucassampaioleite/flask-blog.git
git push -u origin main

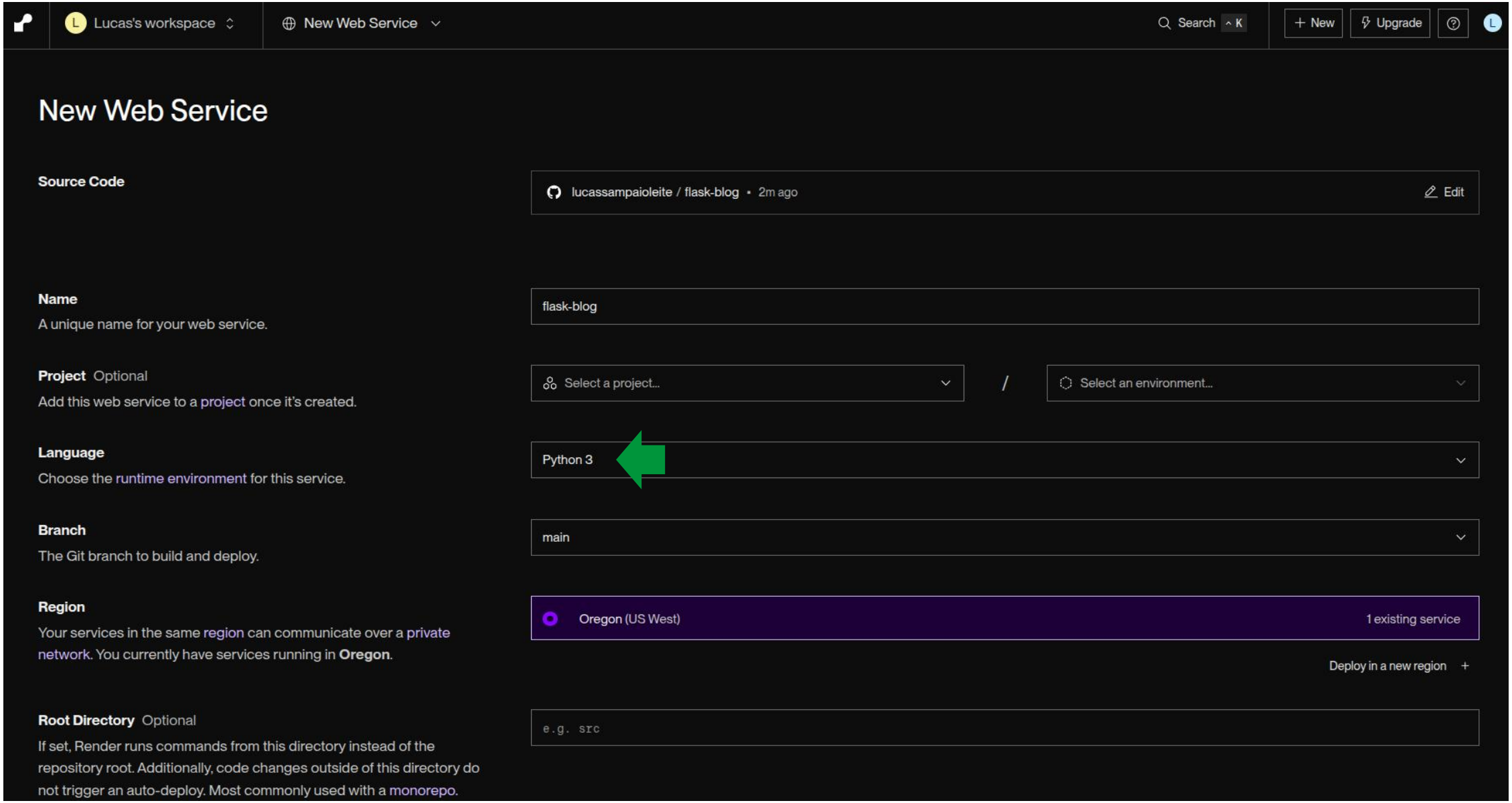
Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

Build Command

Render runs this command to build your app before each deploy.

```
$ pip install -r requirements.txt
```

Start Command

Render runs this command to start your app with each deploy.

```
$ gunicorn src.wsgi:app
```

Instance Type

For hobby projects

Free	512 MB (RAM)
$0 / month	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 features.

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

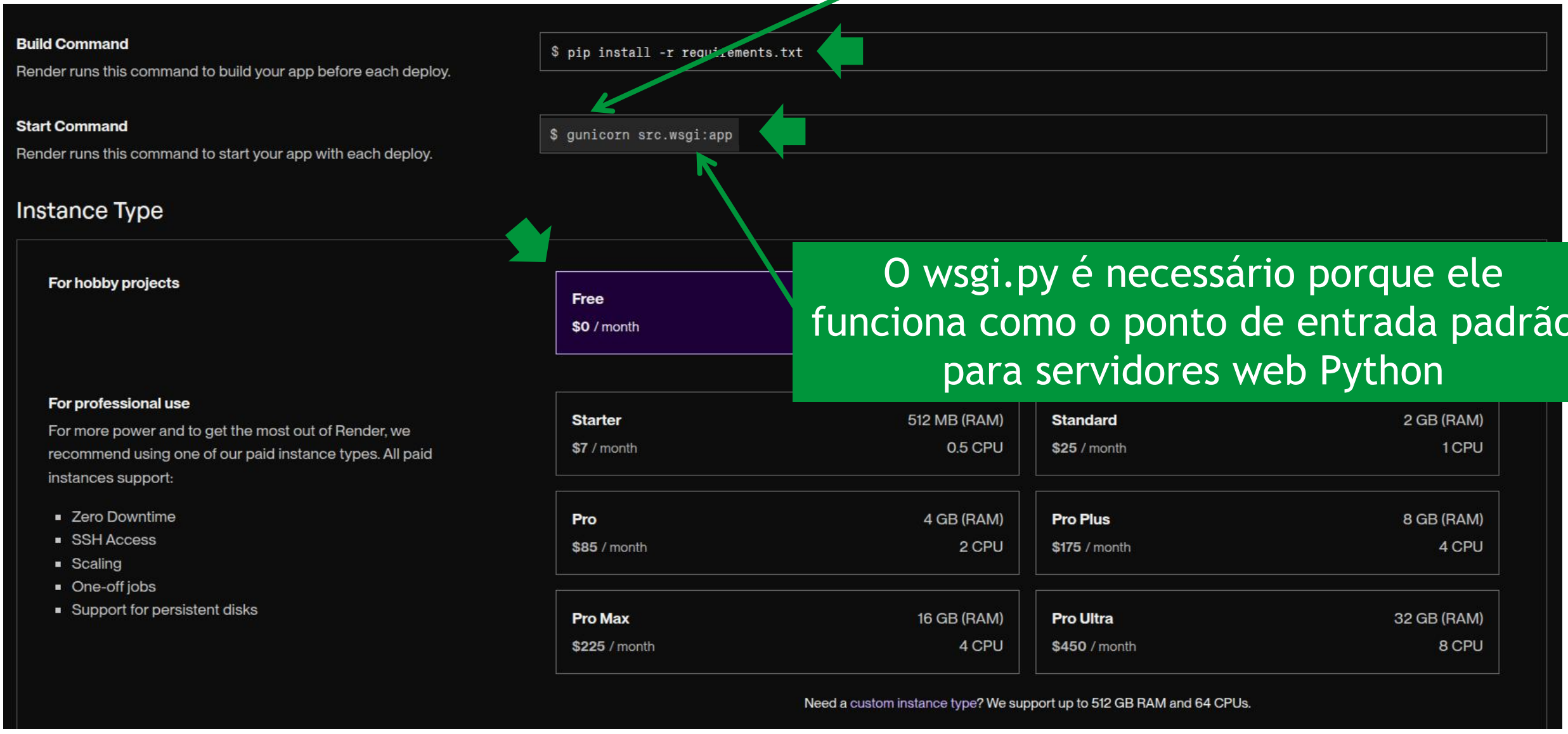
Starter	512 MB (RAM)	**Standard**	2 GB (RAM)
$7 / month	0.5 CPU	**$25** / month	1 CPU
Pro	4 GB (RAM)	**Pro Plus**	8 GB (RAM)
$85 / month	2 CPU	**$175** / month	4 CPU
Pro Max	16 GB (RAM)	**Pro Ultra**	32 GB (RAM)
$225 / month	4 CPU	**$450** / month	8 CPU

Need a custom instance type? We support up to 512 GB RAM and 64 CPUs.

Deploy

O gunicorn é um servidor web de produção usado para executar aplicações Python, como Flask ou Django, em ambientes reais de produção.

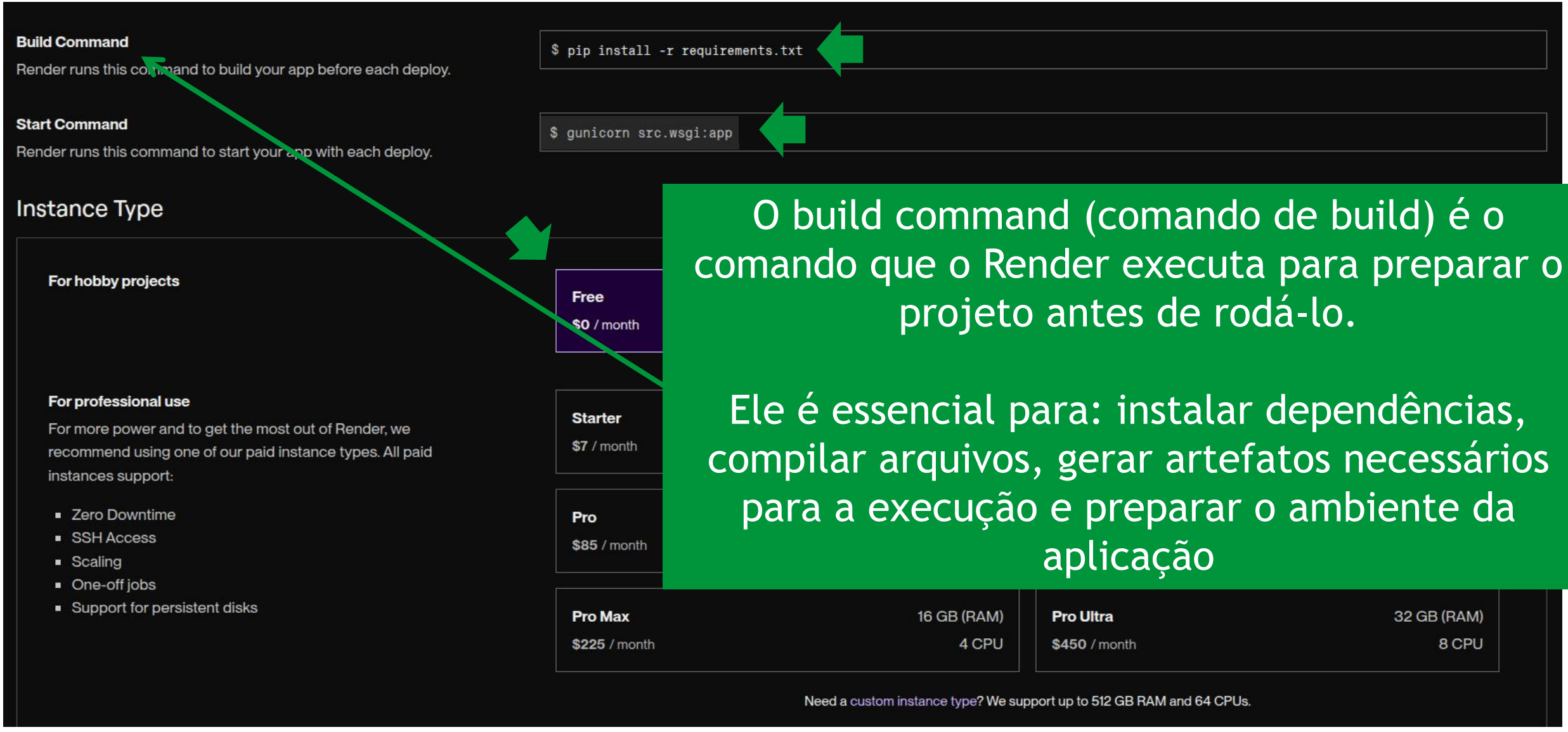
INSTITUTO FEDERAL
Baiano



O wsgi.py é necessário porque ele funciona como o ponto de entrada padrão para servidores web Python

Deploy

INSTITUTO FEDERAL
Baiano

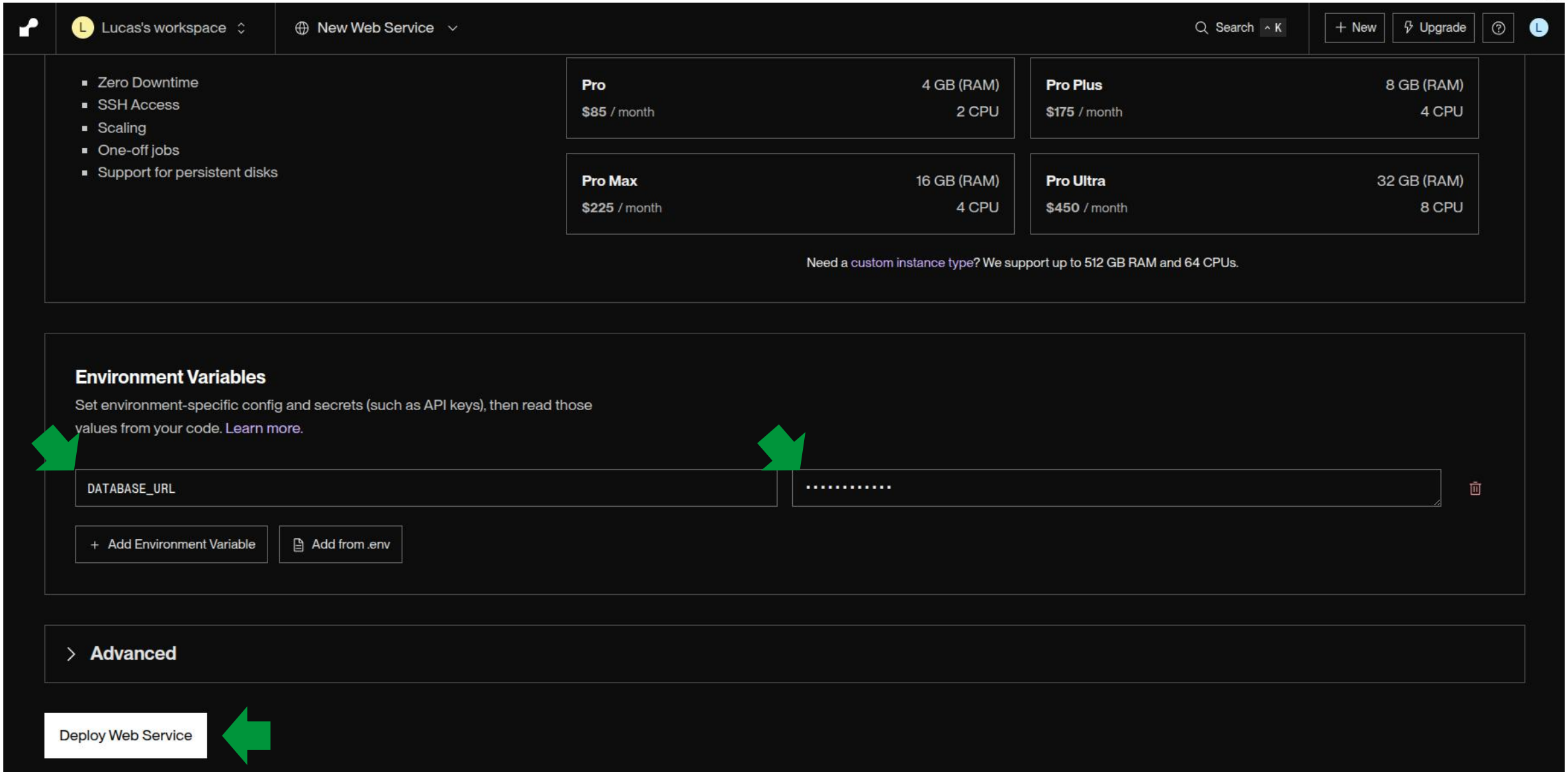


O build command (comando de build) é o comando que o Render executa para preparar o projeto antes de rodá-lo.

Ele é essencial para: instalar dependências, compilar arquivos, gerar artefatos necessários para a execução e preparar o ambiente da aplicação

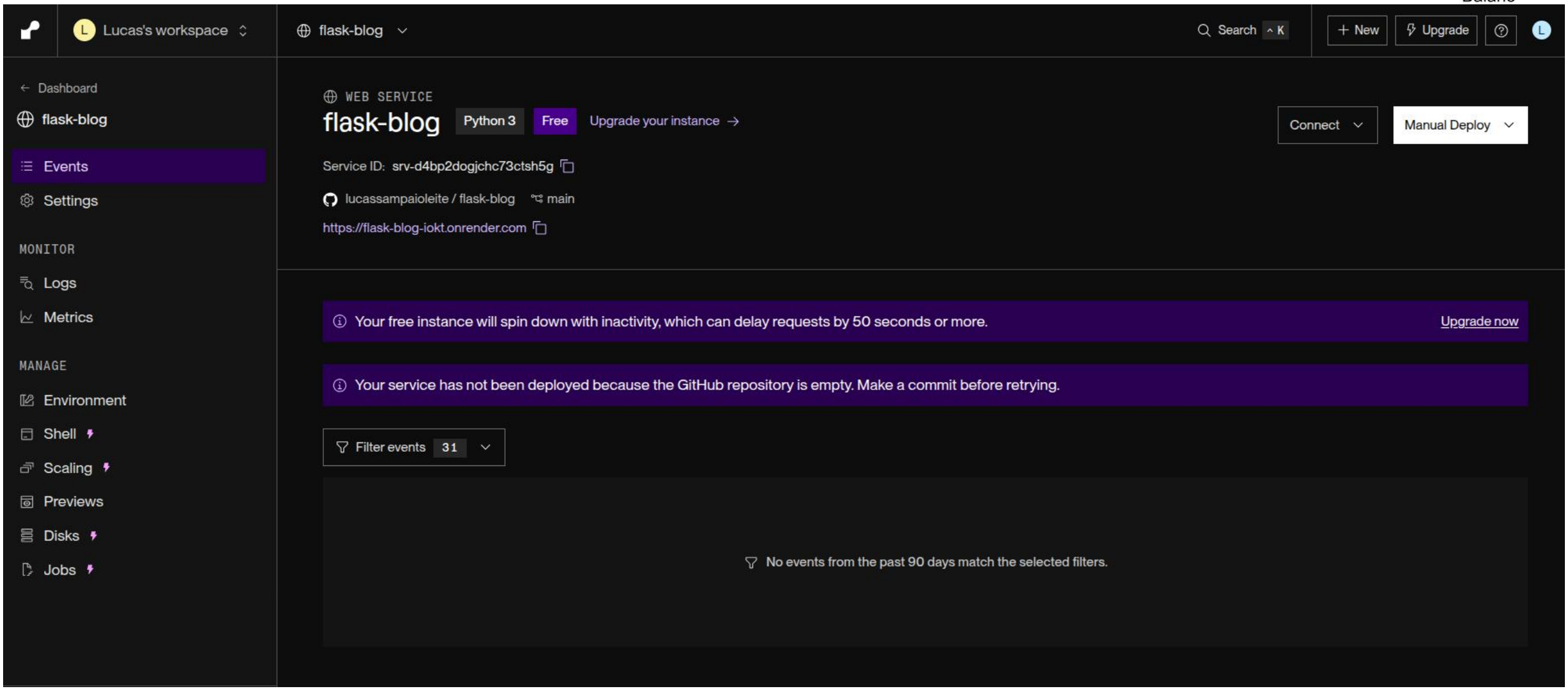
Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano



Deploy

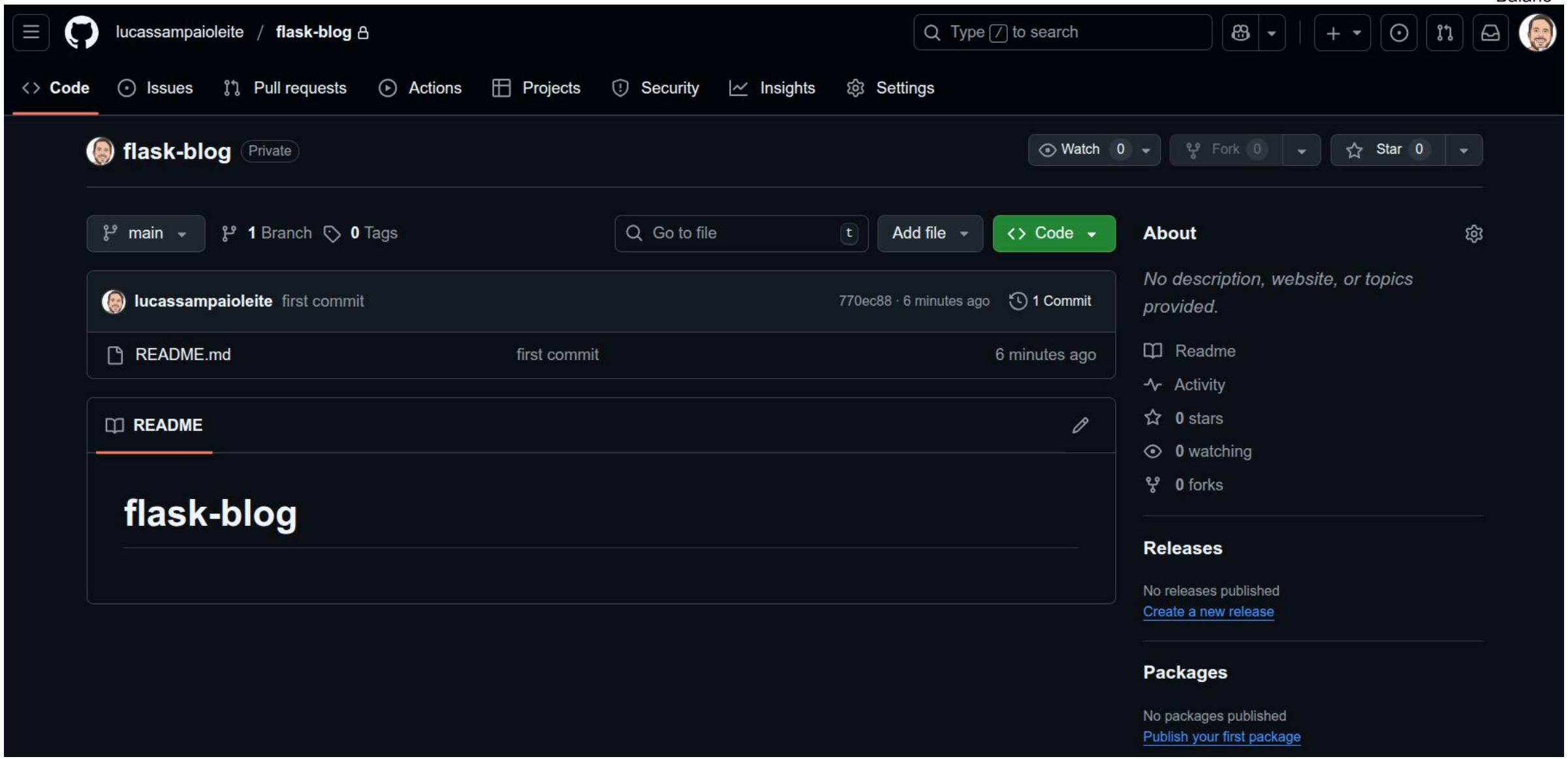
...or create a new repository on the command line

```
echo "# flask-blog" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:lucassampaioleite/flask-blog.git
git push -u origin main
```

```
Initialized empty Git repository in /home/lucas/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog/.git/
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add README.md
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "first commit"
[master (root-commit) 770ec88] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git branch -M main
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git remote add origin git@githu
b.com:lucassampaioleite/flask-blog.git
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:lucassampaioleite/flask-blog.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        Pipfile
        Pipfile.lock
        __pycache__/
        application.py
        controllers/
        extensions/
        instance/
        models/
        views/

nothing added to commit but untracked files present (use "git add" to track)
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

Deploy

INSTITUTO FEDERAL
Baiano

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        Pipfile
        Pipfile.lock
        __pycache__/
        application.py
        controllers/
        extensions/
        instance/
        models/
        views/

nothing added to commit but untracked files present (use "git add" to track)
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

O __pycache__ é uma pasta criada automaticamente pelo Python para armazenar arquivos compilados do se código, com extensão .pyc.

Deploy

INSTITUTO FEDERAL
Baiano

- O .gitignore é um arquivo especial usado pelo Git para dizer quais arquivos ou pastas devem ser ignorados — ou seja, não serão adicionados, versionados ou enviados para o repositório.
- gitignore.io é uma ferramenta online gratuita que cria automaticamente arquivos .gitignore completos e atualizados para qualquer linguagem, framework ou IDE.

Deploy

INSTITUTO FEDERAL
Baiano

gitignore.io

Deploy

INSTITUTO FEDERAL
Baiano

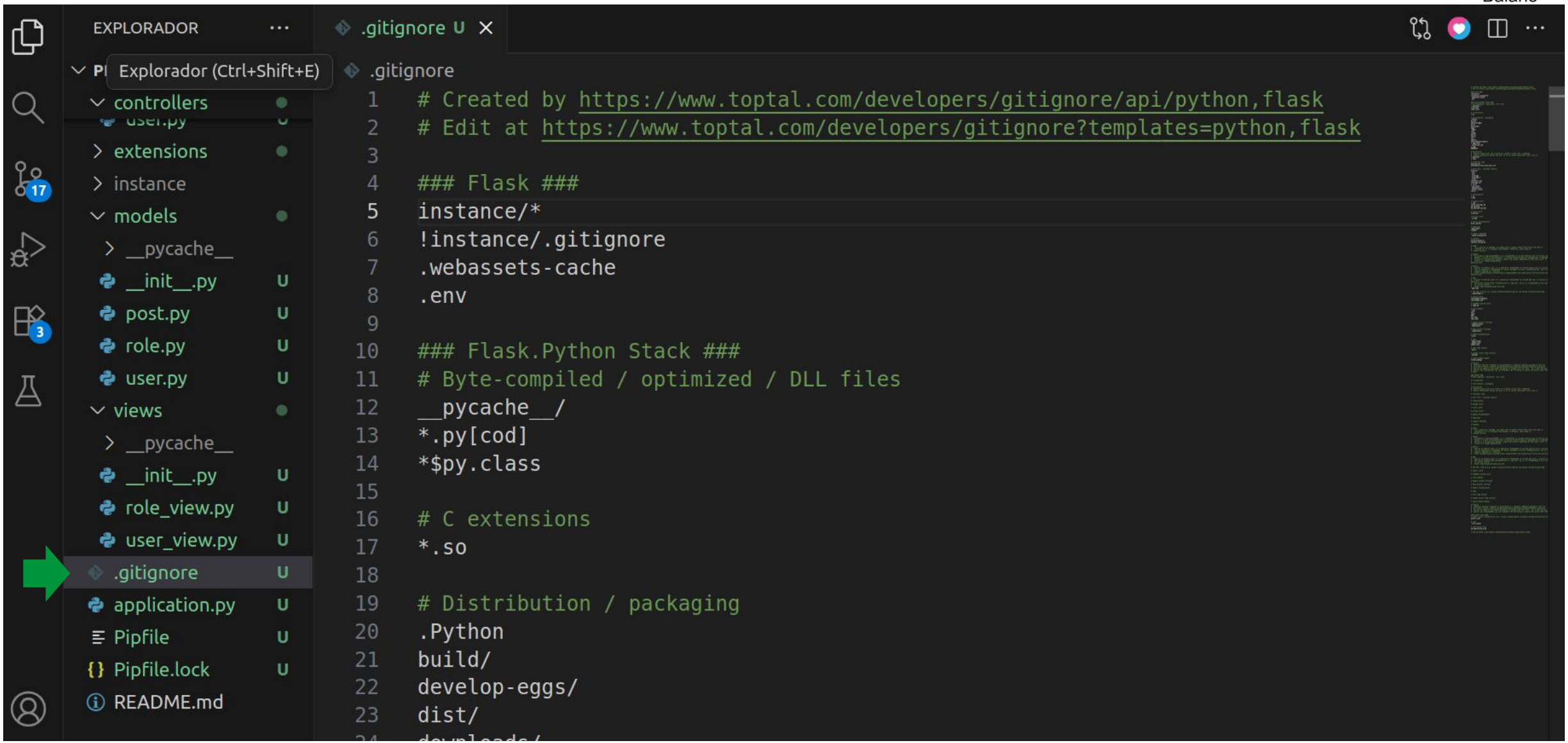
Deploy

```
# Created by https://www.toptal.com/developers/gitignore/api/python,flask
# Edit at https://www.toptal.com/developers/gitignore?templates=python,flask

### Flask ###
instance/*
!instance/.gitignore
.webassets-cache
.env

### Flask.Python Stack ###
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

INSTITUTO FEDERAL
Baiano

Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        Pipfile
        Pipfile.lock
        application.py
        controllers/
        extensions/
        models/
        views/
```

Deploy

INSTITUTO FEDERAL
Baiano

- O psycopg2-binary é o driver que permite que sua aplicação Python (Flask) se conecte a um banco PostgreSQL, que é o banco padrão oferecido pelo Render.
  - Em outras palavras, é o "conector" que permite ao Flask conversar com o PostgreSQL.
- O gunicorn é o servidor web recomendado para rodar aplicações Flask em produção.

Deploy

INSTITUTO FEDERAL
Baiano

- pipenv install psycopg2-binary

```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ pipenv install p
sycopg2-binary
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing psycopg2-binary...
✔ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (659eb1)...
All dependencies are now up-to-date!
Upgrading psycopg2-binary in  dependencies.
Building requirements...
Resolving dependencies...
✔ Success!
```

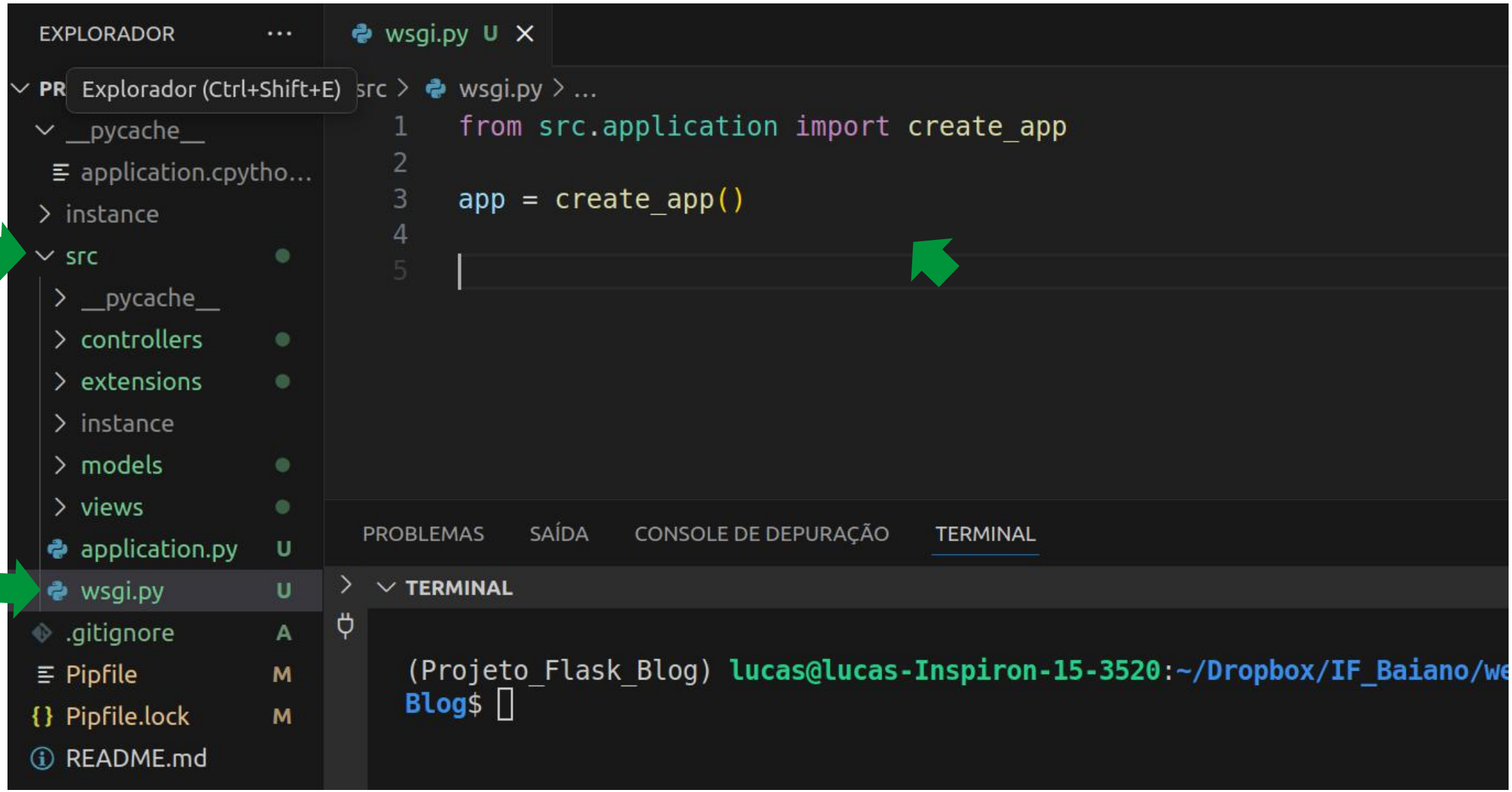
Deploy

INSTITUTO FEDERAL
Baiano

- pipenv install gunicorn

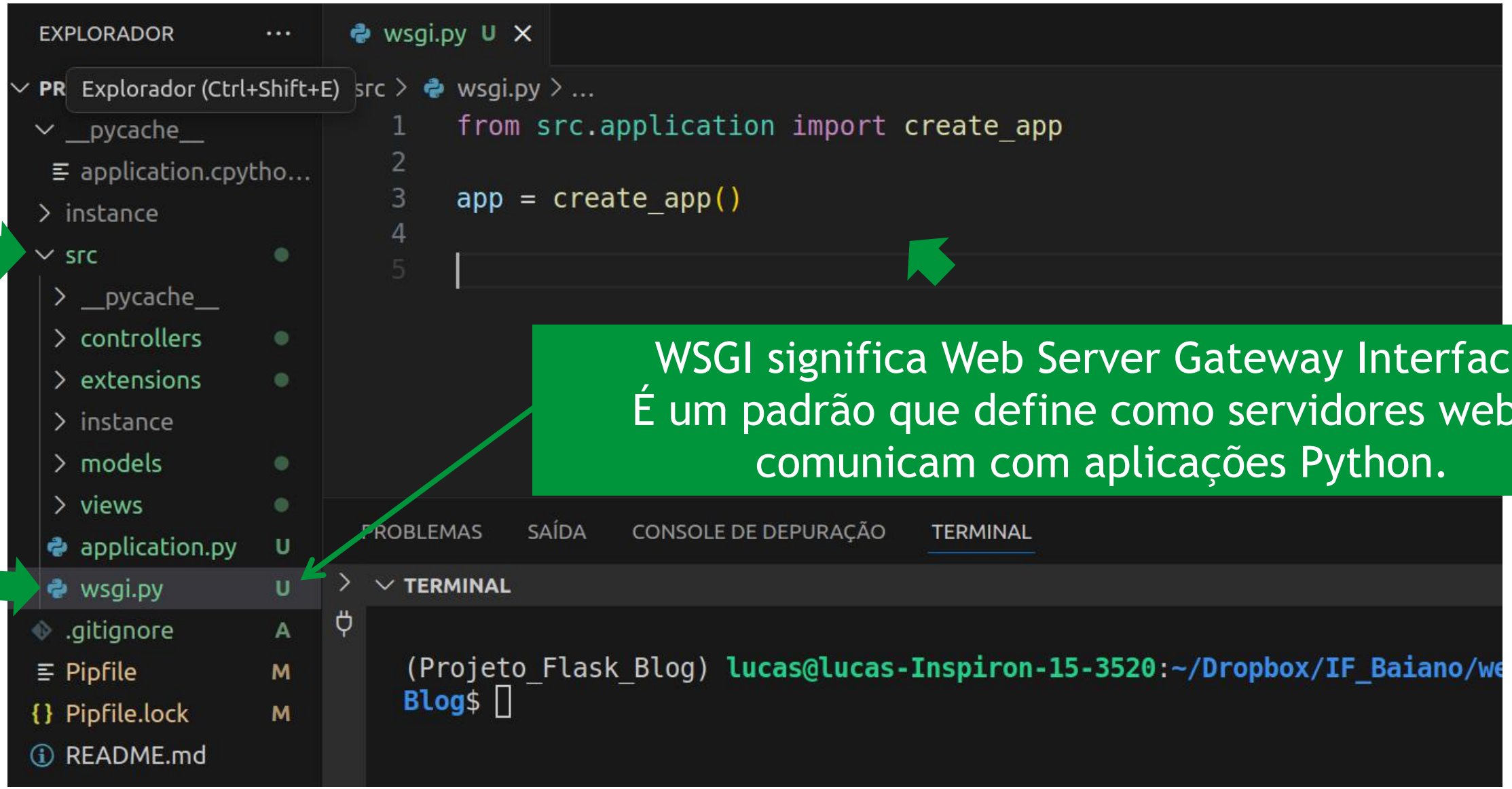
```
lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ pipenv install g
unicorn
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing gunicorn...
✔ Installation Succeeded
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
Installing dependencies from Pipfile.lock (3c5272)...
All dependencies are now up-to-date!
Upgrading gunicorn in  dependencies.
Building requirements...
Resolving dependencies...
✔ Success!
```

Deploy


EXPLORADOR
wsgi.py U
Explorador (Ctrl+Shift+E)
src > wsgi.py > ...
1 from src.application import create_app
2
3 app = create_app()
4
5
__pycache__
application.cpytho...
instance
src
__pycache__
controllers
extensions
instance
models
views
application.py U
wsgi.py U
.gitignore A
Pipfile M
Pipfile.lock M
README.md
PROBLEMAS
SAÍDA
CONSOLE DE DEPURAÇÃO
TERMINAL
TERMINAL
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/we
Blog$

Deploy

INSTITUTO FEDERAL
Baiano



WSGI significa Web Server Gateway Interface.
É um padrão que define como servidores web se comunicam com aplicações Python.

Deploy

INSTITUTO FEDERAL
Baiano

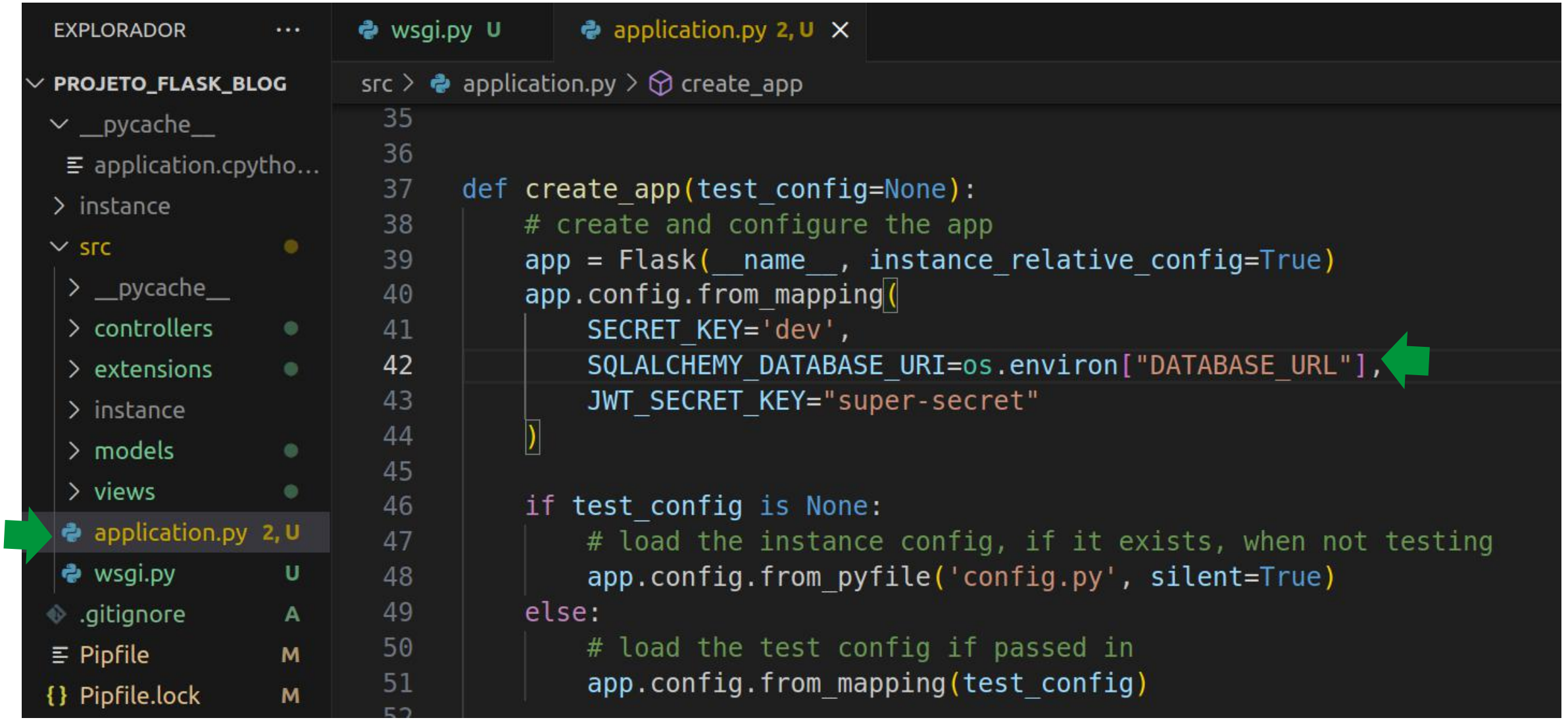
```
from src.extensions import db
from src.extensions import jwt
```

Corrigir erros de import:
src.models
src.controllers
src.views
src.extensions

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_B
log$ flask --app src.application run --debug
 * Serving Flask app 'src.application'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 110-182-683
```

Deploy

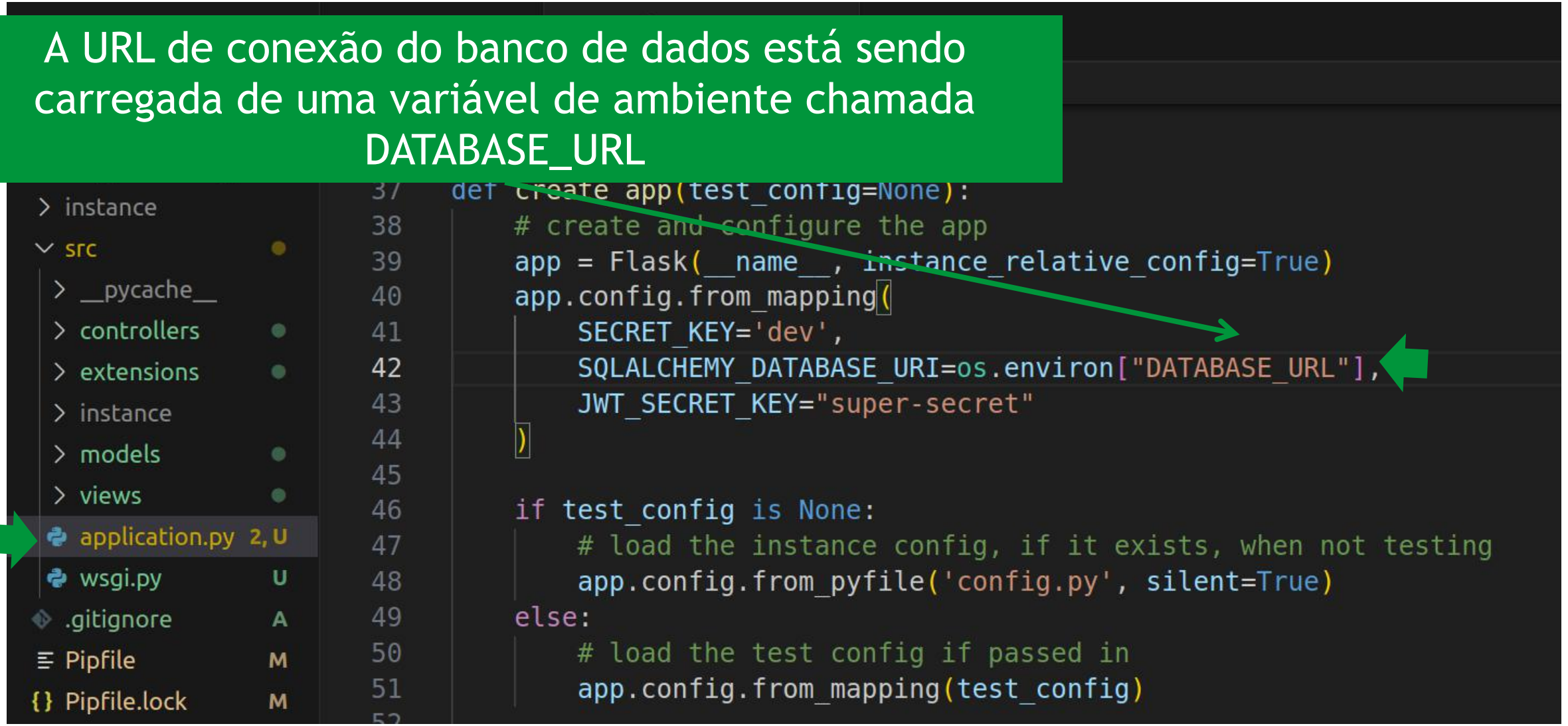
INSTITUTO FEDERAL
Baiano



Deploy

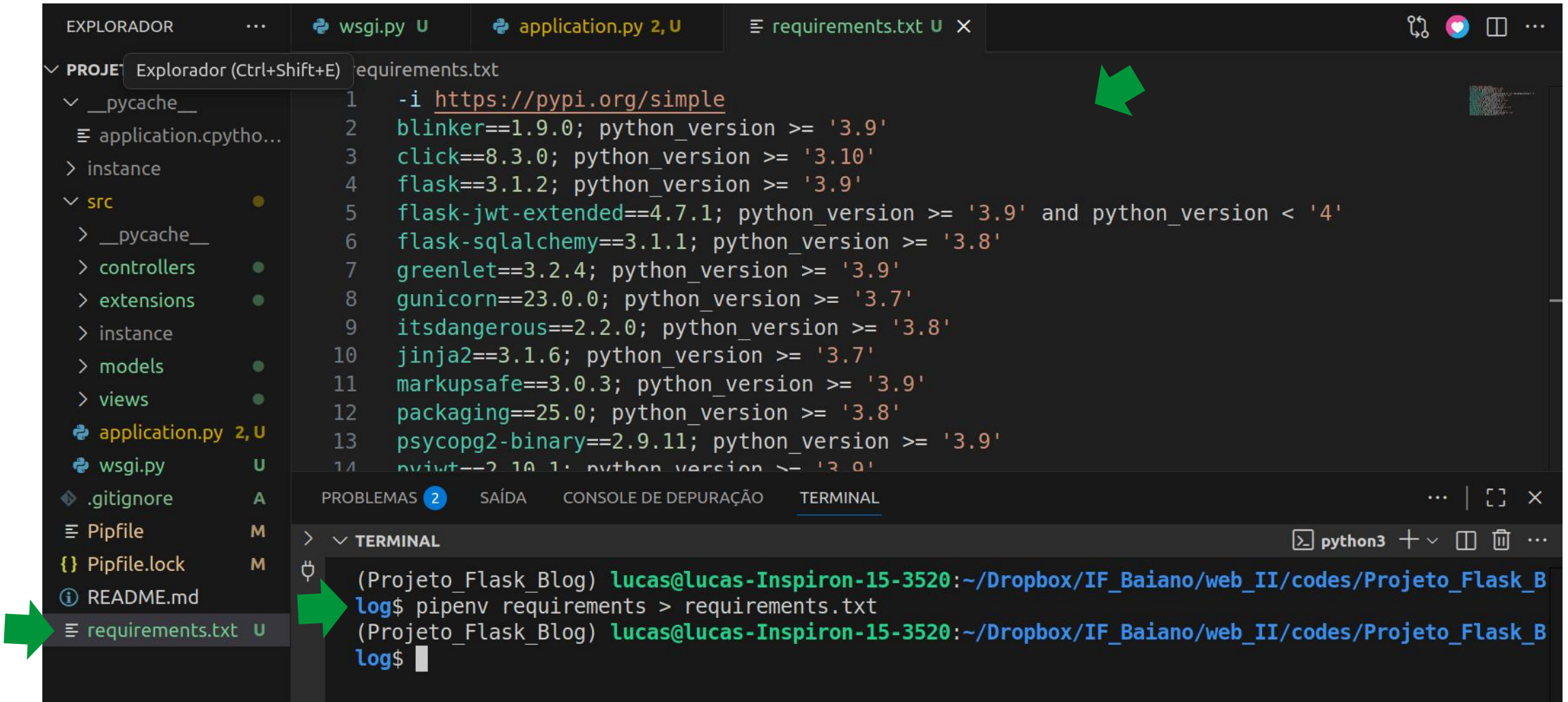
INSTITUTO FEDERAL
Baiano

A URL de conexão do banco de dados está sendo carregada de uma variável de ambiente chamada DATABASE_URL



Deploy

- pipenv requirements > requirements.txt



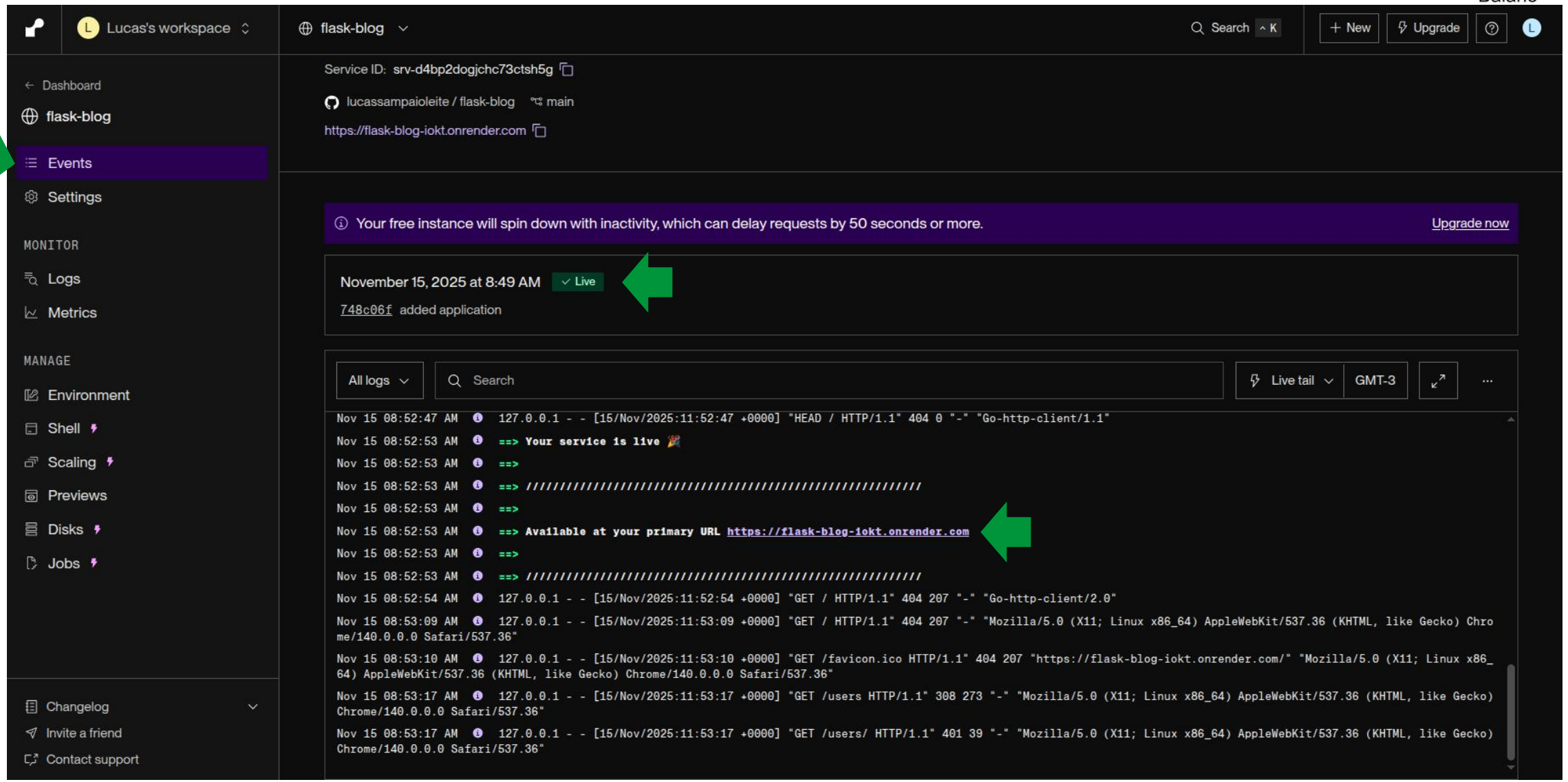
Deploy

INSTITUTO FEDERAL
Baiano

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add .
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "added init_db"
[main e1ab334] added init_db
 1 file changed, 8 insertions(+)
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 398 bytes | 398.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:lucassampaioleite/flask-blog.git
   748c06f..e1ab334  main -> main
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

Deploy

INSTITUTO FEDERAL
Baiano



Deploy

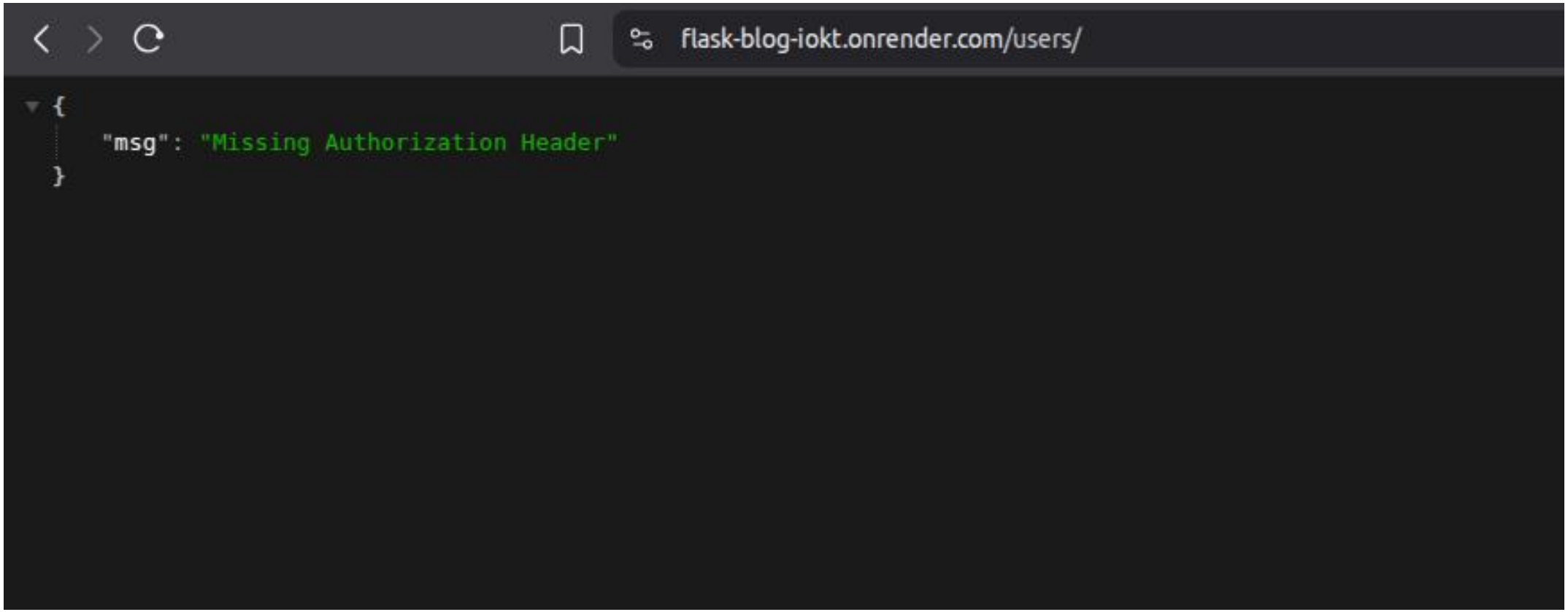
INSTITUTO FEDERAL
Baiano

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

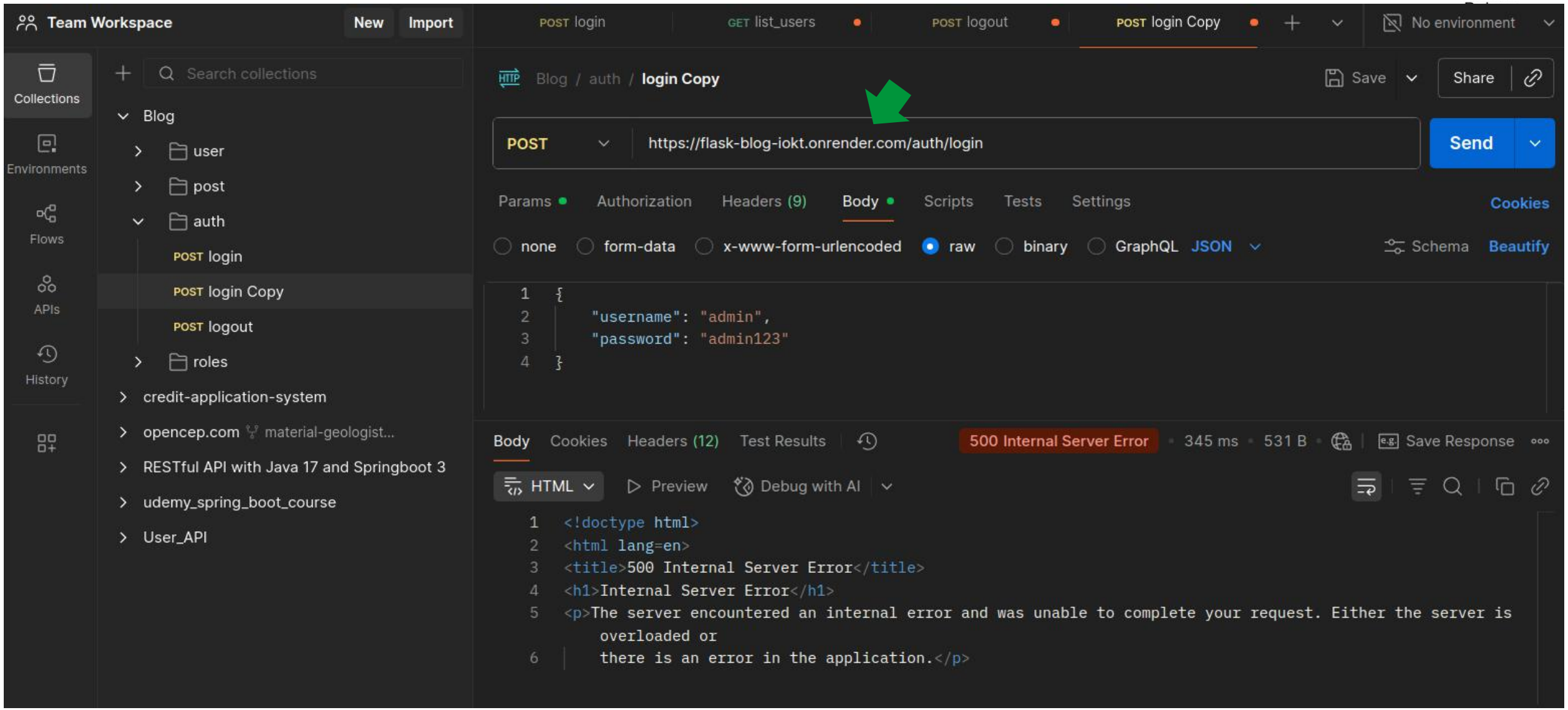
Deploy

INSTITUTO FEDERAL
Baiano



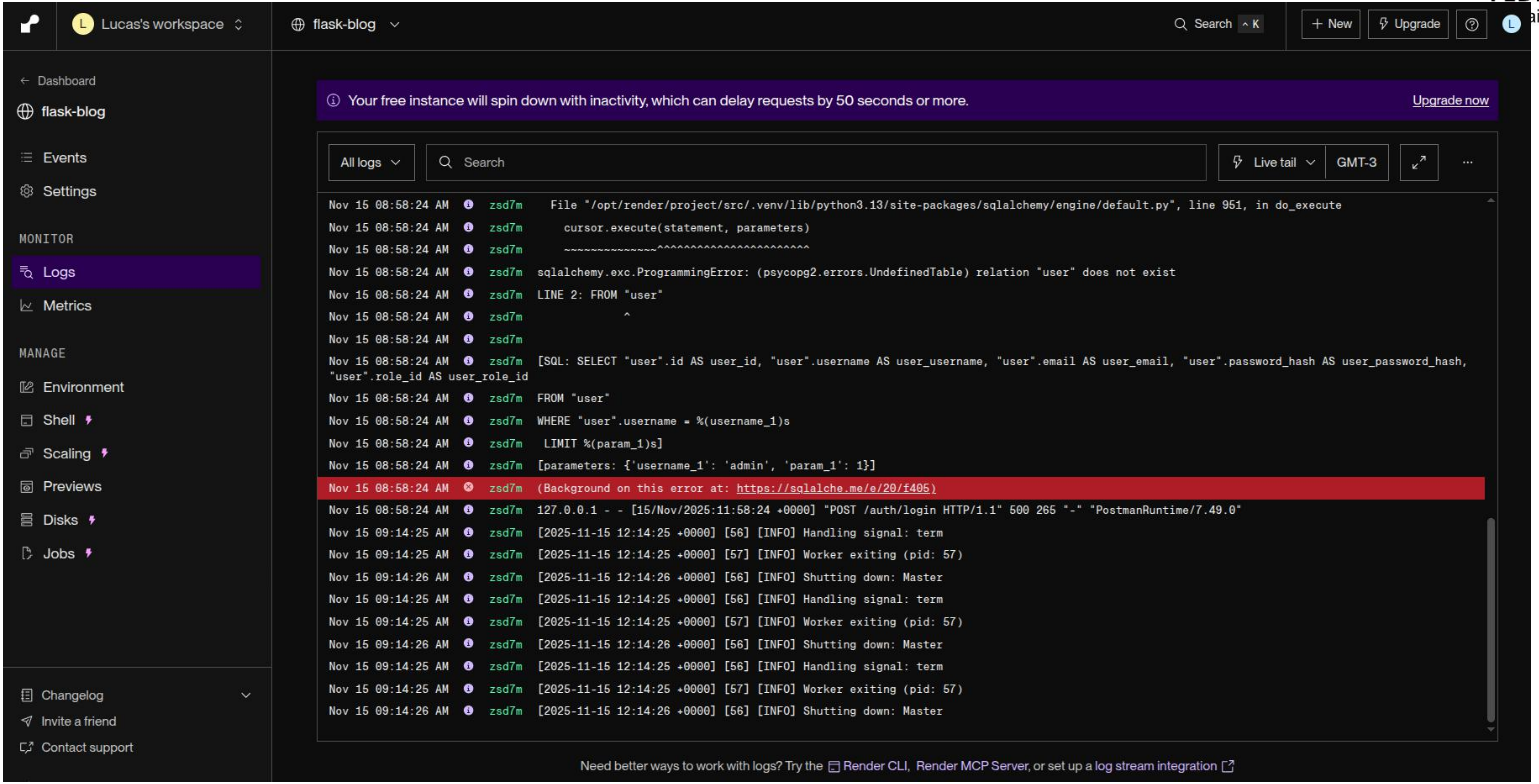
Deploy

INSTITUTO FEDERAL



Deploy

INSTITUTO FEDERAL
Baiano



Deploy

INSTITUTO FEDERAL
Baiano

```python
class User(db.Model):
    id: Mapped[int] = mapped_column(primary_key=True)
    username: Mapped[str] = mapped_column(
        db.String(80), unique=True, nullable=False)
    email: Mapped[str] = mapped_column(db.String(120), nullable=True)
    password_hash: Mapped[str] = mapped_column(db.String(300), nullable=False)
```

Aumentar o tamanho do campo da hash

Deploy

```python
@click.command("init-db")
def init_db_command():
    with current_app.app_context():
        init_db()

def init_db():
    with current_app.app_context():
        db.create_all()

        # cria a role "admin" se não existir
        role_admin = Role.query.filter_by(name="admin").first()
        if not role_admin:
            role_admin = Role(name="admin")
            db.session.add(role_admin)
            db.session.commit()
            click.echo("Role 'admin' criada!")

        # cria um usuário "admin" se não existir
        if not User.query.filter_by(username="admin").first():
            user = User(username="admin",
```

Em application.py, função create_app()

```python
with app.app_context():
    init_db()
```

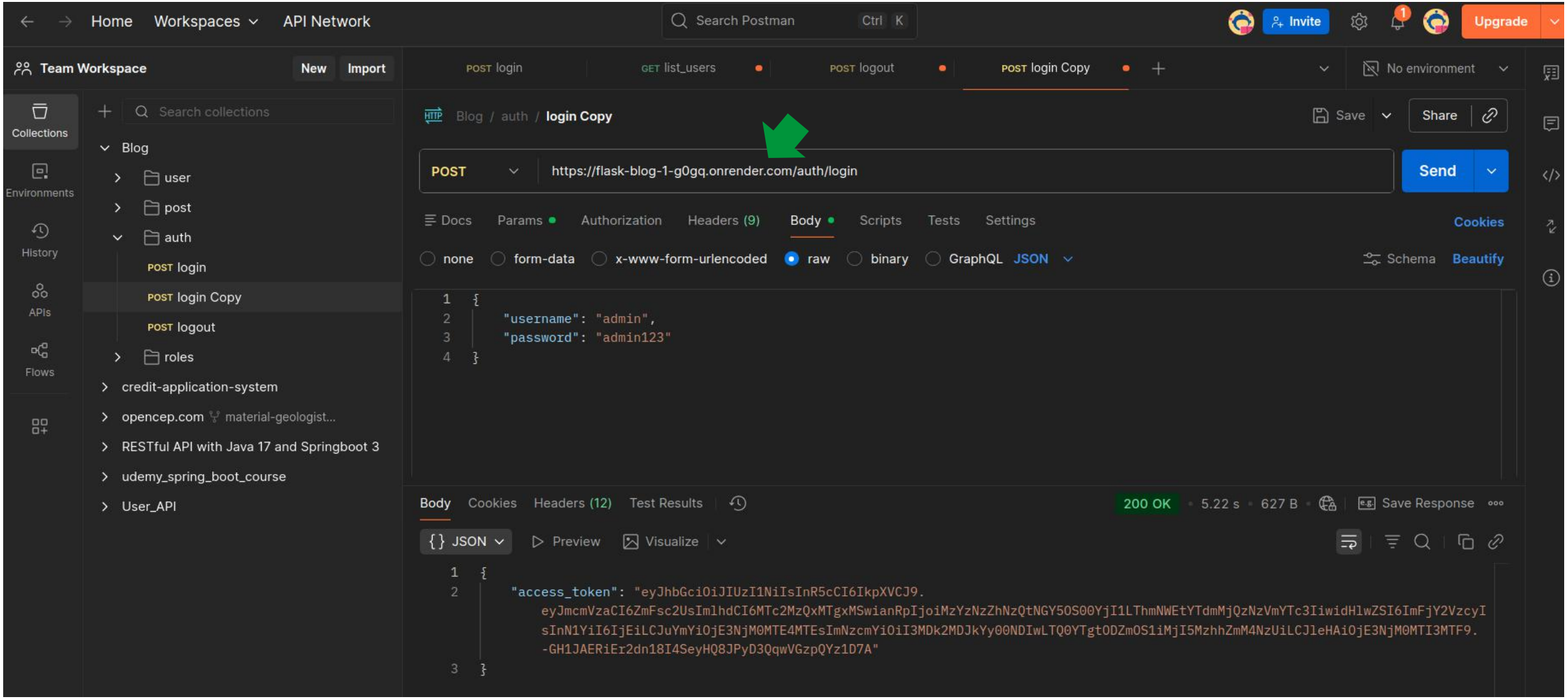
Deploy

INSTITUTO FEDERAL
Baiano

```
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git add .
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git commit -m "added init_db"
[main e1ab334] added init_db
 1 file changed, 8 insertions(+)
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 398 bytes | 398.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:lucassampaioleite/flask-blog.git
   748c06f..e1ab334  main -> main
(Projeto_Flask_Blog) lucas@lucas-Inspiron-15-3520:~/Dropbox/IF_Baiano/web_II/codes/Projeto_Flask_Blog$
```

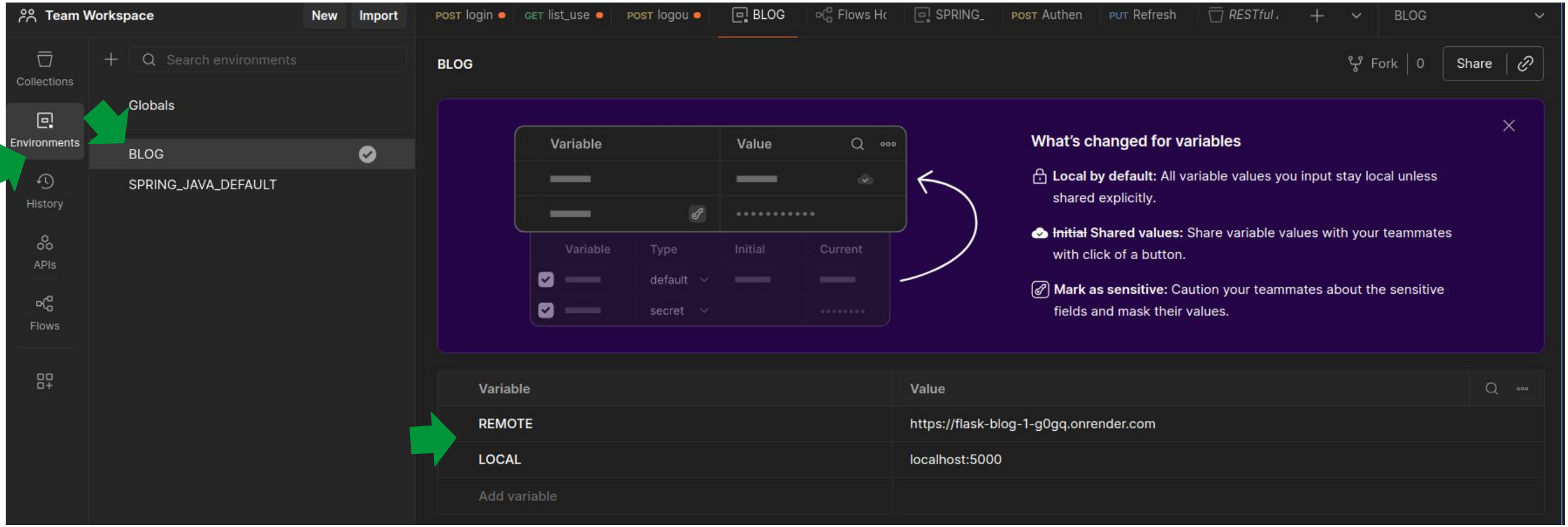
Deploy

INSTITUTO FEDERAL
Baiano



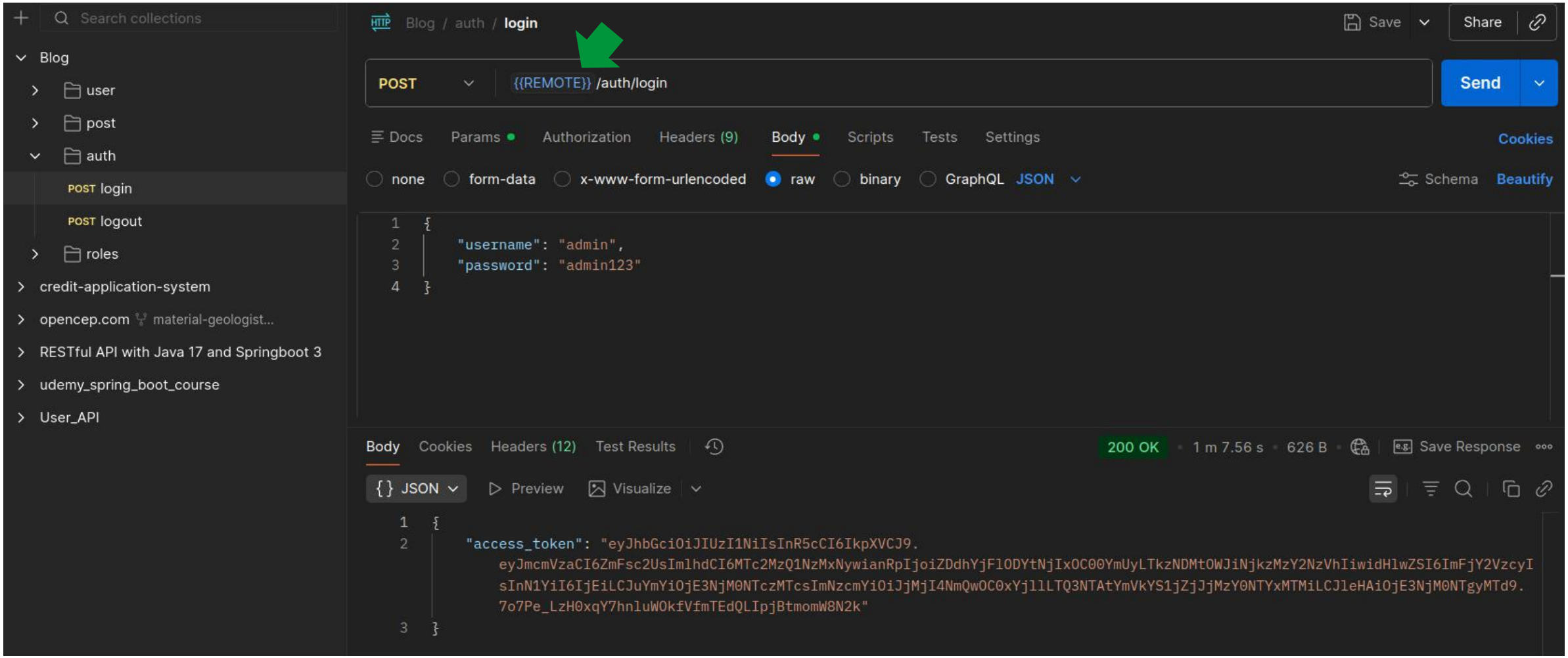
Deploy

INSTITUTO FEDERAL
Baiano



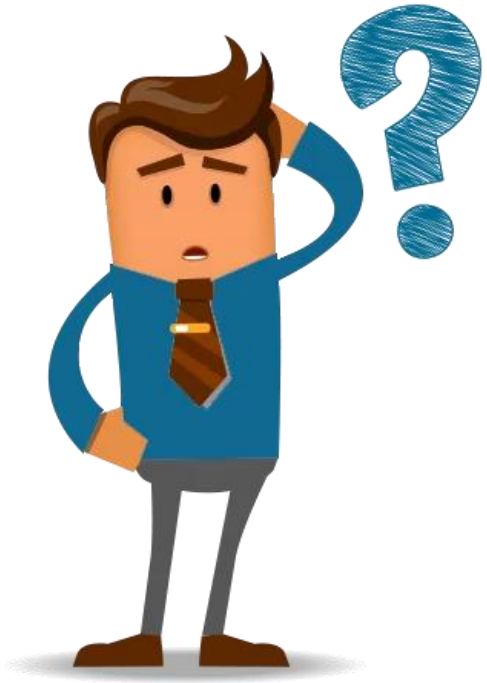
Deploy

INSTITUTO FEDERAL
Baiano



Dúvidas

INSTITUTO FEDERAL
Baiano



PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática

Lucas Sampaio Leite