



Heroku

Live de Python # 162

Roteiro



1. Heroku

Uma breve introdução

2. Heroku CLI

Criando nossa aplicação

3. Banco de dados

Deixando nossa aplicação mais bacana

4. Deploy contínuo

Fazer deploys automáticos

Uma breve
introdução

Heroku



Heroku é uma plataforma como serviço (PaaS) baseada em containers [dynos]. Para que a gente possa fazer deploy [colocar em produção] nossas aplicações.

Poliglota:

- Ruby
- GO
- Python
- Node.js
- Java
- PHP
- Scala
- Closure

História do Heroku



Heroku é uma empresa fundada em 2007, nos EUA. Em 2011 foi adquirida pela Sales Force.

O nome Heroku é uma junção das palavras "**Heroic**" e "**Haiku**" em homenagem a Yukihiro Matsumoto (Matz) criador da linguagem Ruby, a única suportada inicialmente pela empresa.

Explicação da estrutura



Explicação da estrutura

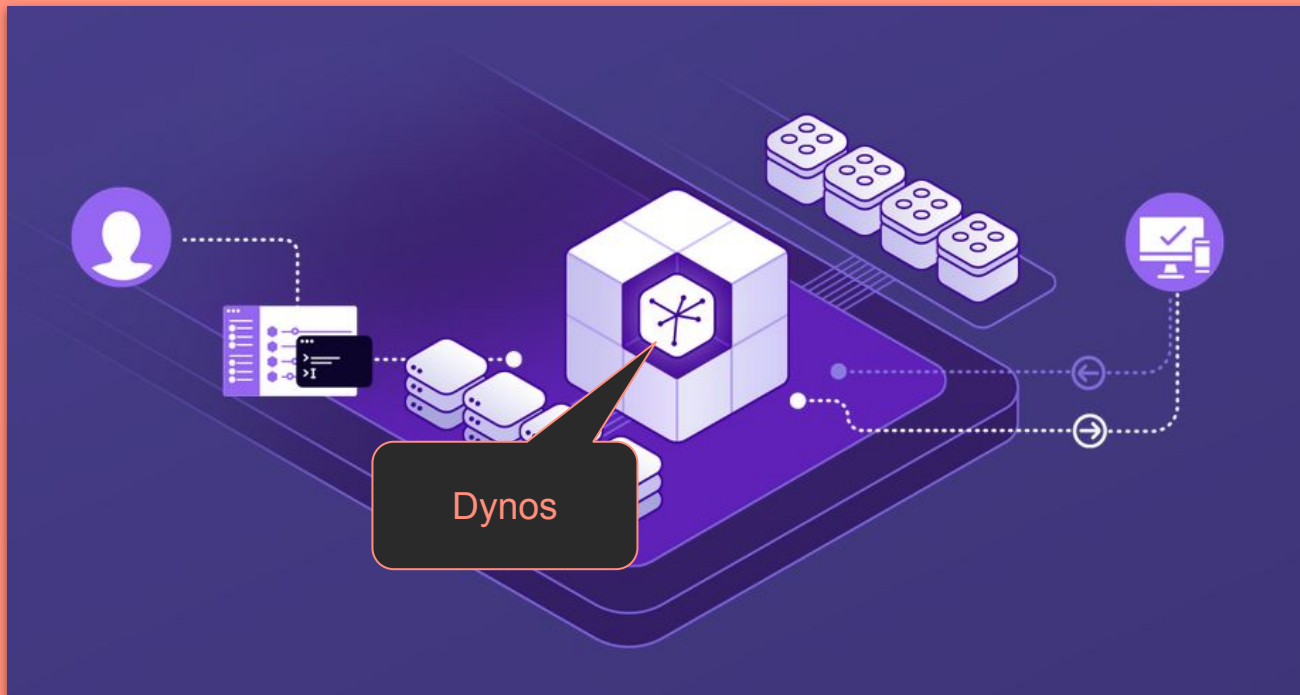


Explicação da estrutura

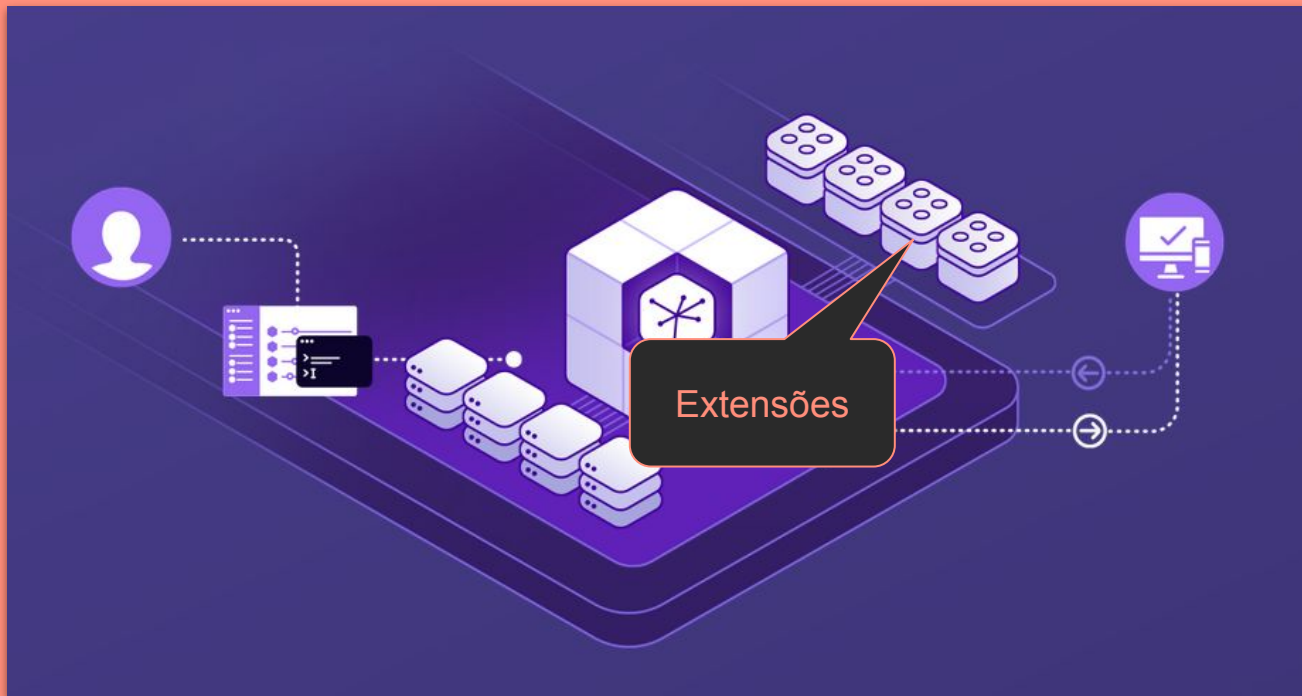


Bancos de
dados

Explicação da estrutura



Explicação da estrutura

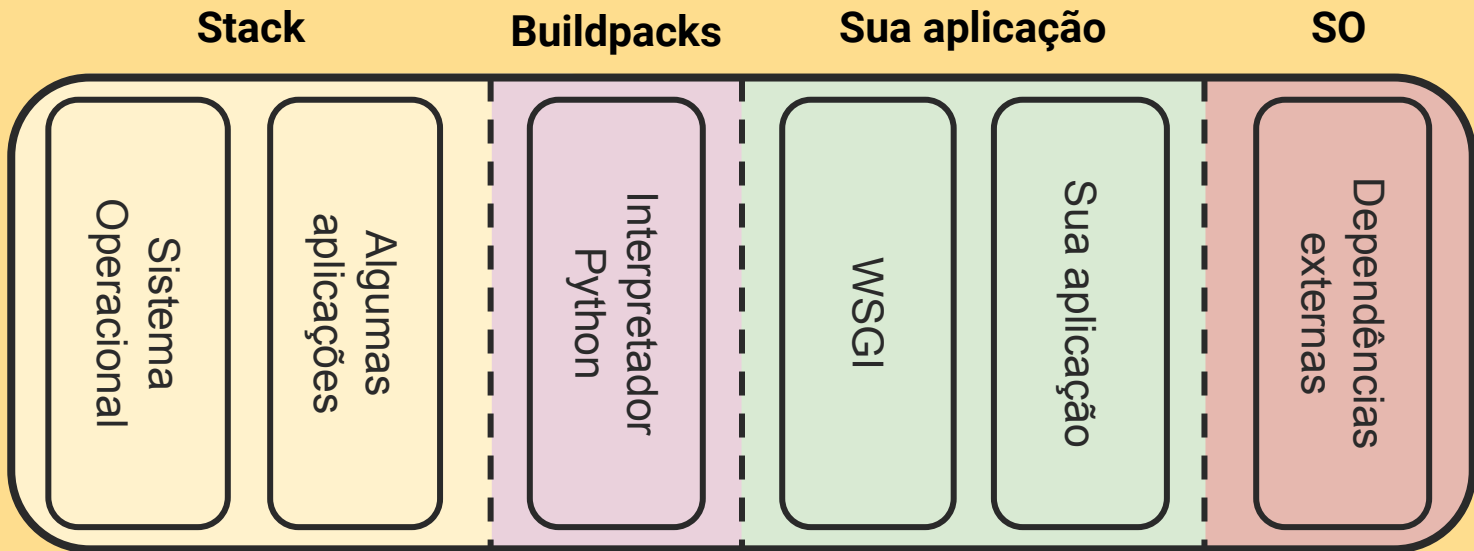


Tá, mas o que é PaaS?



"Máquina"

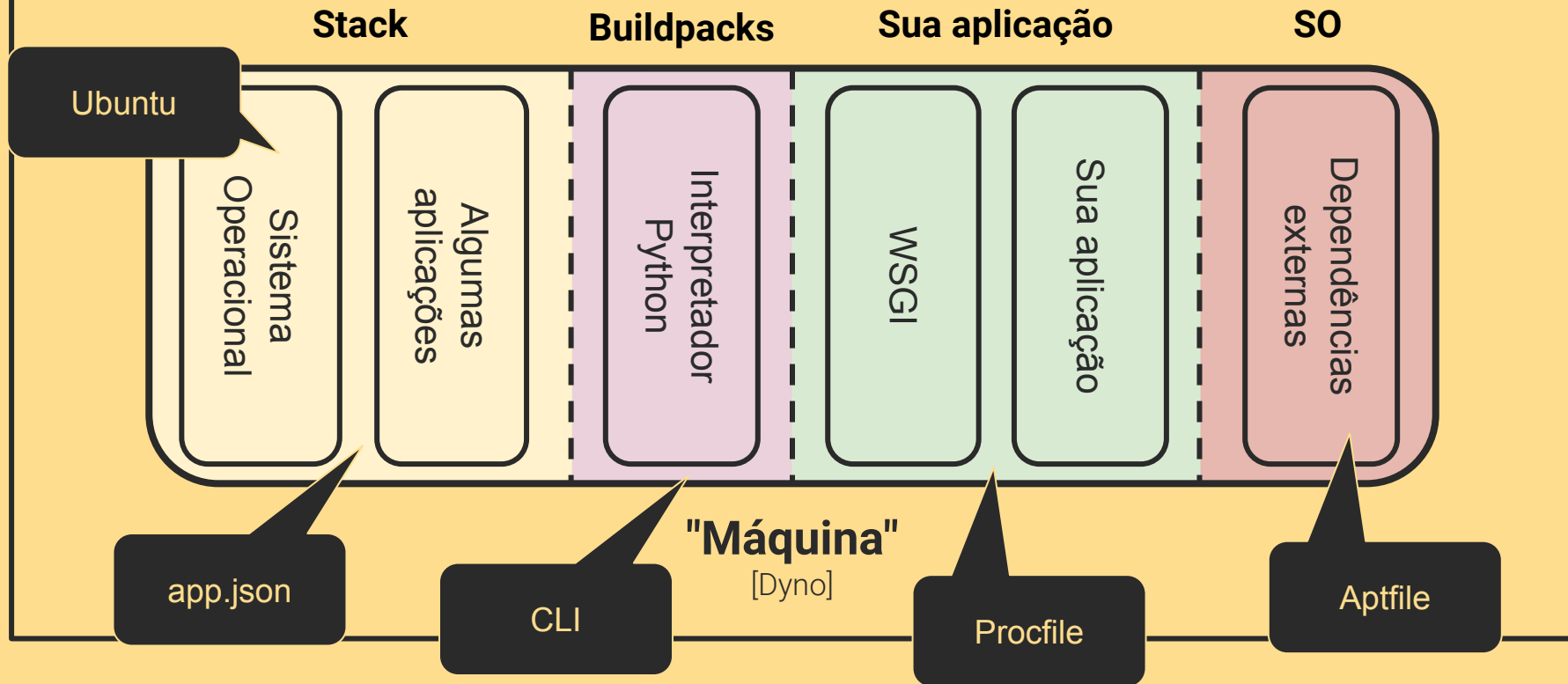
Tá, mas o que é PaaS?



"Máquina"

[Dyna]

Tá, mas o que é PaaS?



CLI

Criando nossa
aplicação



O CLI é a maneira de interagir com o heroku sem necessaria ter que abrir a página da plataforma. Fazer tudo via linha de comando

Nossos primeiro comandos:

- **heroku login**

<https://devcenter.heroku.com/articles/heroku-cli#getting-started>

Desse ponto em diante, você precisa ter o GIT instalado
no seu pc

<https://git-scm.com/downloads>



Nota importante



Bora criar um novo projeto



```
1  # Criar o projeto
2  mkdir nome_do_projeto
3  cd nome_do_projeto
4
5  # Configurações iniciais
6  poetry init -n # OU python -m venv .venv
7  poetry shell # source .venv/bin/activate
8
9  # Iniciar o git
10 git init .
```

Aqui tudo depende



Se você quer usar um framework síncrono [django, flask, bottle, ...], você pode instalar o **gunicorn**.

Caso esteja usando um framework assíncrono [Starlette, Fast-API], você pode instalar o **uvicorn**.

```
1  poetry add gunicorn # pip install gunicorn  
2  
3  poetry add uvicorn # pip install uvicorn
```

Criando um projeto no heroku



```
1  # heroku login
2  heroku create nome_do_projeto
```

Criando um projeto no heroku



```
1 # heroku login
2 heroku create nome_do_projeto
```

```
1 $ cat .git/config
2
3 [core]
4     repositoryformatversion = 0
5     filemode = true
6     bare = false
7     logallrefupdates = true
8 [remote "heroku"]
9     url = https://git.heroku.com/nome_do_projeto.git
10     fetch = +refs/heads/*:refs/remotes/heroku/*
11
```

Ninguém é de ferro,
né?



Codar um pouquinho



Publicando nossa aplicação pela primeira vez



Agora vaiiiiiiiiiiiii

```
1  git add .  
2  git commit -m "primeiro commit"  
3  git push heroku main # ou master
```

Procfile



Procfile é o arquivo que o Heroku usa para dizer para o dyno o que deve ser executado

```
1  # Procfile
2  <process type>: <command>
3
4  # No nosso caso
5  web: gunicorn "app:create_app()"
6
```

Publicando nossa aplicação pela segunda vez



Antes não foi, mas agora VAI

```
1  git add .  
2  git commit -m "adicionando Procfile"  
3  git push heroku main # ou master
```


Buildpacks



Os buildpacks dizem ao heroku que tipo de "ecossistema" estamos tentando rodar. Como nosso sistema é Python, devemos dizer a ele

```
1 heroku buildpacks:add heroku/python
2
3 # Caso esteja usando Poetry
4
5 heroku buildpacks:add https://github.com/moneymeets/python-poetry-buildpack.git
```

Publicando nossa aplicação pela terceira vez



Tá tirando né?

```
1  git add .  
2  git commit -m "Não é possível, funciona!"  
3  git push heroku main # ou master
```

Possíveis problemas



- Versão incompatível do python
- Incompatibilidade com alguma versão de biblioteca
- Não conseguiu acessar algo

Possíveis problemas



- Versão incompatível do python
- Incompatibilidade com alguma versão de biblioteca
- Não conseguiu acessar algo

LEIA OS LOGS

Fazendo nossa
aplicação
persistir

Bancos
de
dados

Quais bancos o Heroku suporta?



- Postegres
- Redis
- Apache Kafka

```
1  # Adicionar o Banco a nossa aplicação
2  heroku addons:create heroku-postgresql:hobby-dev
3
4  # Pegar informações do banco
5  heroku pg:info
6
7  # Pega nossas credenciais
8  heroku pg:credentials:url
```

Codar mais um cadin?



Pra gente ver o banco funcionando



Deploy Contínuo

Fazer uma vez,
para executar
sempre