



FaaS / AWS lambdas

Live de Python # 172



1. Cloud é de comer?

Uma introdução aos conceitos

2. FaaS / Serverless

Uma introdução aos conceitos

2. AWS lambda

No free tier, claro

3. Chalice

Um framework web python para lambdas

4. Integrações

S3, CloudWatch e etc...



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Ademar Peixoto, Alex Lima, Alex Lopes, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysson Oliveira, Amaziles Carvalho, André Rocha, Arnaldo Turque, Bruno Batista, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Dilenon Delfino, Elias Soares, Eugenio Mazzini, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Fabrícia Diniz, Fabrício Coelho, Flavkaze, Francisco Alencar, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Ostrock, Gustavo Chacon, Henrique Machado, Israel Fabiano, Italo Silva, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Prado, João Lugão, João Schiavon, Juan Gutierrez, Julio Silva, Jônatas Silva, Júlia Kastrup, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Maiquel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Marina Passos, Matheus Vian, Melissa Mendonça, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Ronaldo Silva, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Victor Geraldo, Vinícius Bastos, Vinícius Ferreira, Vítor Gomes, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Érico Andrei



Obrigado você



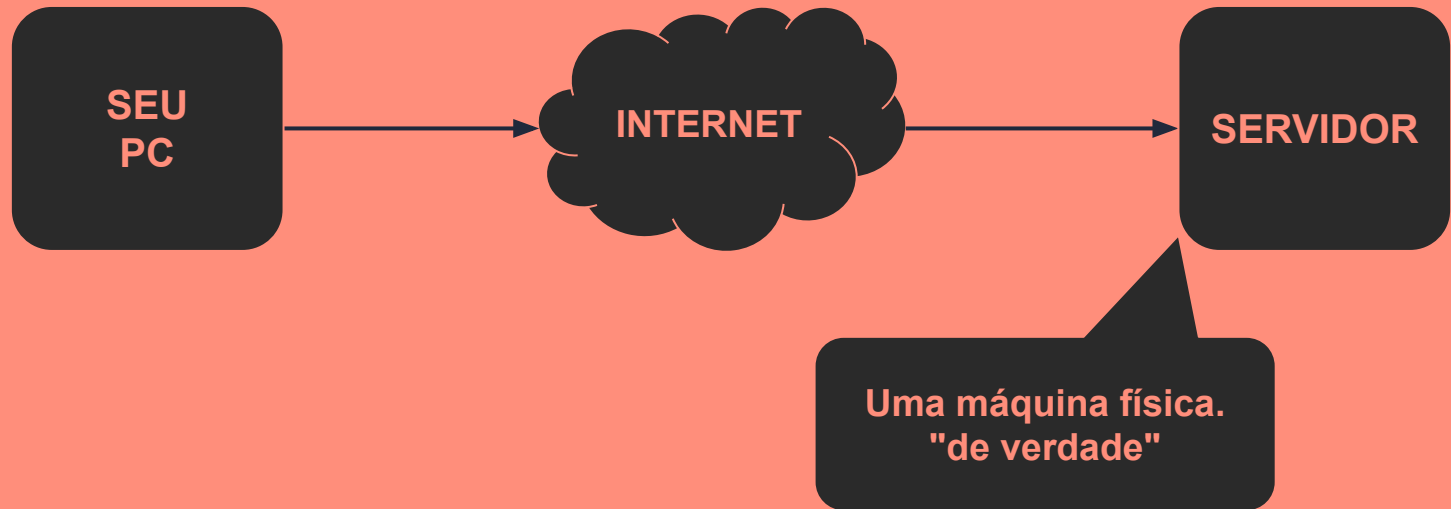
Uma introdução
conceitual

Cloud

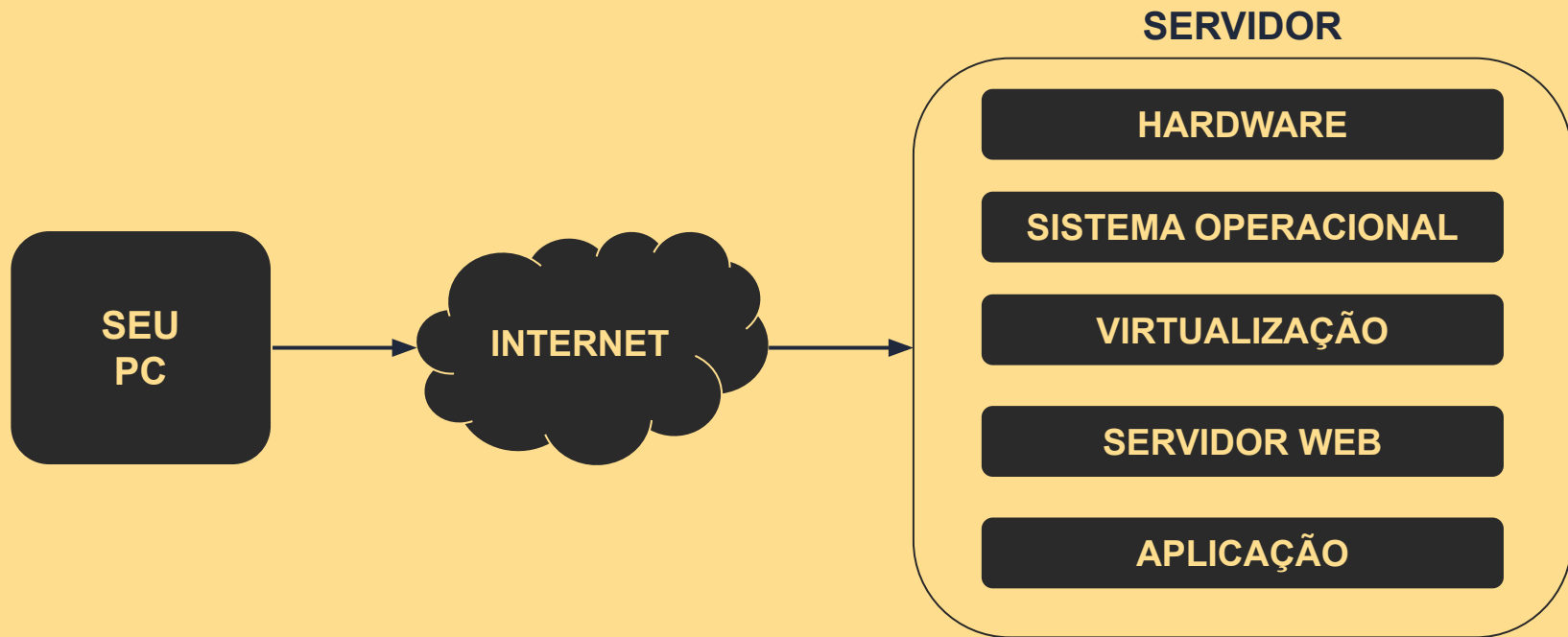
Como funcionam os servidores na WEB?



Como funcionam os servidores na WEB?



Como funcionam os servidores na WEB?



E como funciona a "nuvem"? (IaaS)

Um modelo tradicional de uma "máquina rodando".

Quando compramos isso de uma empresa, esse serviço é ofertado como **IaaS**, Infraestrutura como serviço.

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

E como funciona a "nuvem"? (IaaS)

Um modelo tradicional de uma "máquina rodando".

Quando compramos isso de uma empresa, esse serviço é ofertado como **IaaS**, Infraestrutura como serviço.

Um sistema operacional limpo, "você" deve configurar tudo

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

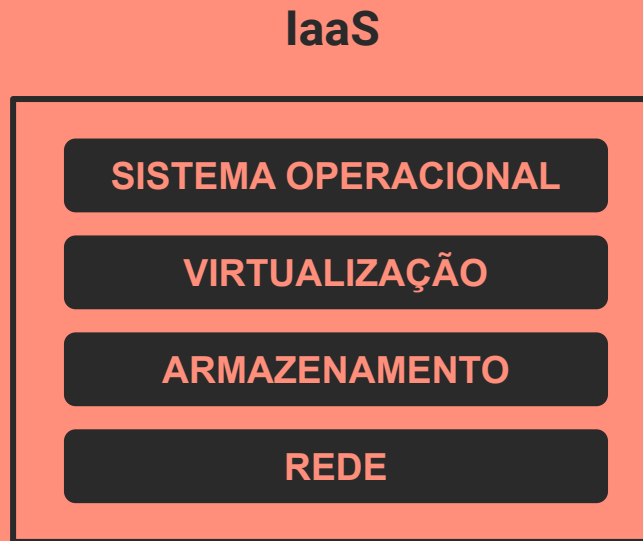
REDE

E como funciona a "nuvem"? (IaaS)

Um modelo tradicional de uma "máquina rodando".

Quando compramos isso de uma empresa, esse serviço é ofertado como **IaaS**, Infraestrutura como serviço.

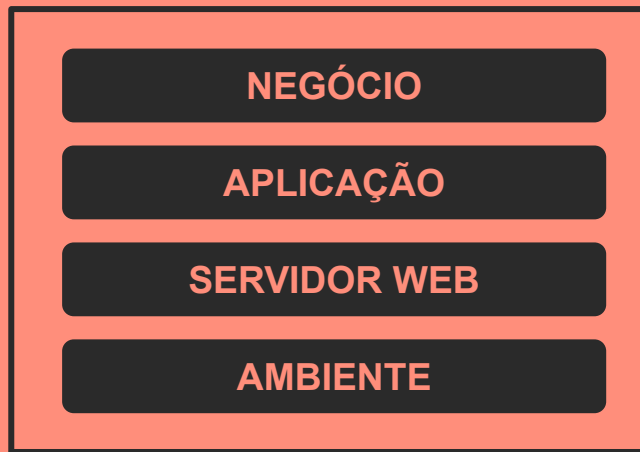
Exemplos: AWS EC2, Microsoft Azure, Google Compute Engine



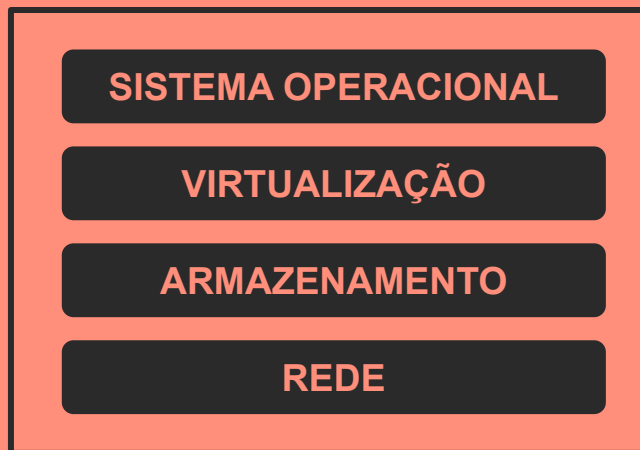
O que preciso manter?

Ainda é necessária a implementação:

- ambiente
 - Interpretador
 - Aplicações, ...
- Servidor web
 - nginx, apache, ...
- Sua aplicação
- Suas regras na sua aplicação



IaaS

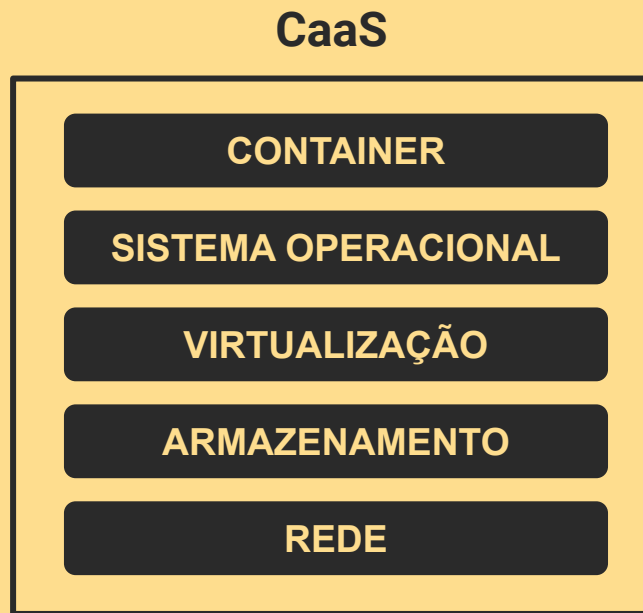


E como funciona a "nuvem"? (CaaS)

Com a grande adoção dos containers (docker por ex), temos um modelo de computação baseado em processamentos de aplicações containerizadas

Esse serviço é ofertado como **CaaS**, Container como serviço.

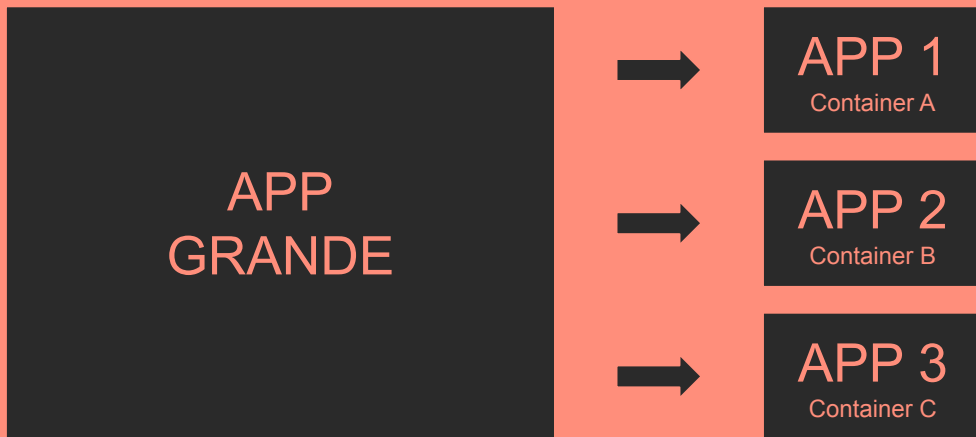
Exemplos: AWS ECS, Azure Container instances, Google Kubernetes engine, ...



Vantagens do CaaS - Microservices



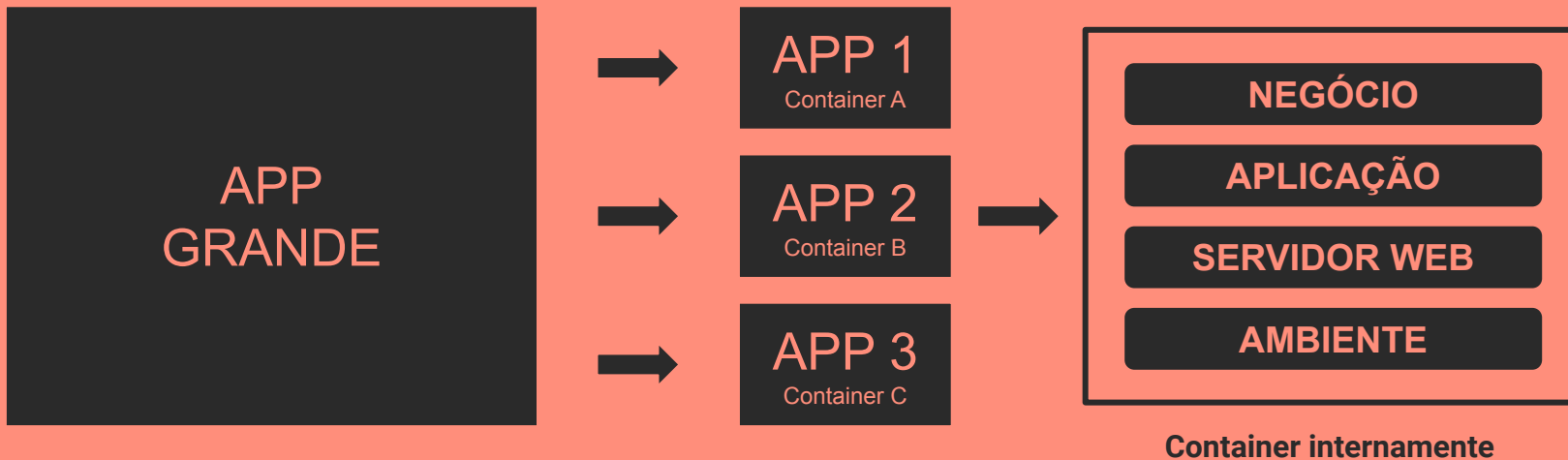
A facilidade de controle. No lugar de uma aplicação única, você pode ter várias que conversam entre si, pois existe "facilidade" de manutenção.



Vantagens do CaaS - Microservices



A facilidade de controle. No lugar de uma aplicação única, você pode ter várias que conversam entre si, pois existe "facilidade" de manutenção.

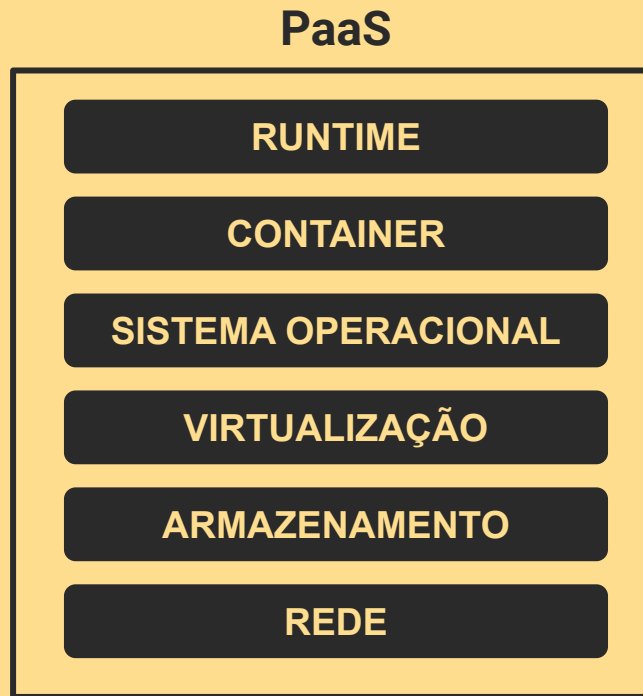


E como funciona a "nuvem"? (PaaS)

Indo um pouco mais além, temos um formato onde o container já está configura e só precisamos fazer upload do código

Esse serviço é ofertado como **PaaS**, Plataforma como serviço.

Exemplos: Heroku, Google App Image, AWS Beanstalk, Dokku



Vantagens do PaaS



Não é necessário o desenvolvimento de nada que não seja a aplicação. Com a aplicação desenvolvida, só é necessário fazer upload do código para ela.

Não envolvendo atividades relativas aos containers ou ao sistema operacional.

Uma visão geral

IaaS

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

CaaS

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

PaaS

RUNTIME

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

Uma visão geral

Ambiente, servidor web,
aplicação

IaaS

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

Ambiente, servidor web,
aplicação

CaaS

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

aplicação

PaaS

RUNTIME

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

FaaS

Uma introdução
conceitual

FaaS



O FaaS, Função como serviço, pode ser visto por duas frentes.

Uma das frentes é o modo de cloud, como uma evolução do PaaS. E outra frente como uma evolução no estilo de como as aplicações são feitas.

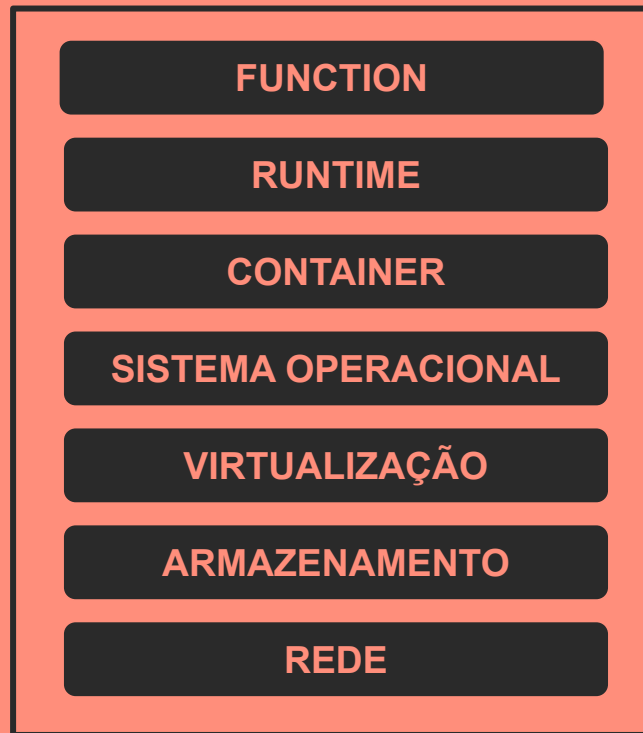
Vamos olhar primeiro para a frente cloud.

E como funciona a "nuvem"? (FaaS)

A ideia da Função é que não é mais necessário se preocupar com a "web" somente com a implementação da regra de negócio.

Exemplos: Open FaaS, OpenWhisk, AWS lambda, Azure Functions, ...

FaaS



Tá, mas calma aí.

O que você quer dizer com function?

FaaS

FUNCTION

RUNTIME

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

Tá, mas calma aí.

O que você quer dizer com function?

```
1 def lambda_handler(request, contexto):  
2     return {  
3         'mensagem': 'Olá mundo',  
4         'evento': request,  
5     }  
6
```

FaaS

FUNCTION

RUNTIME

CONTAINER

SISTEMA OPERACIONAL

VIRTUALIZAÇÃO

ARMAZENAMENTO

REDE

Vamos botar isso no ar?



Só de exemplo!



Configurações de tempo de execução [Info](#)

Editar

Tempo de execução
Python 3.8

Manipulador [Info](#)
function.lambda_handler



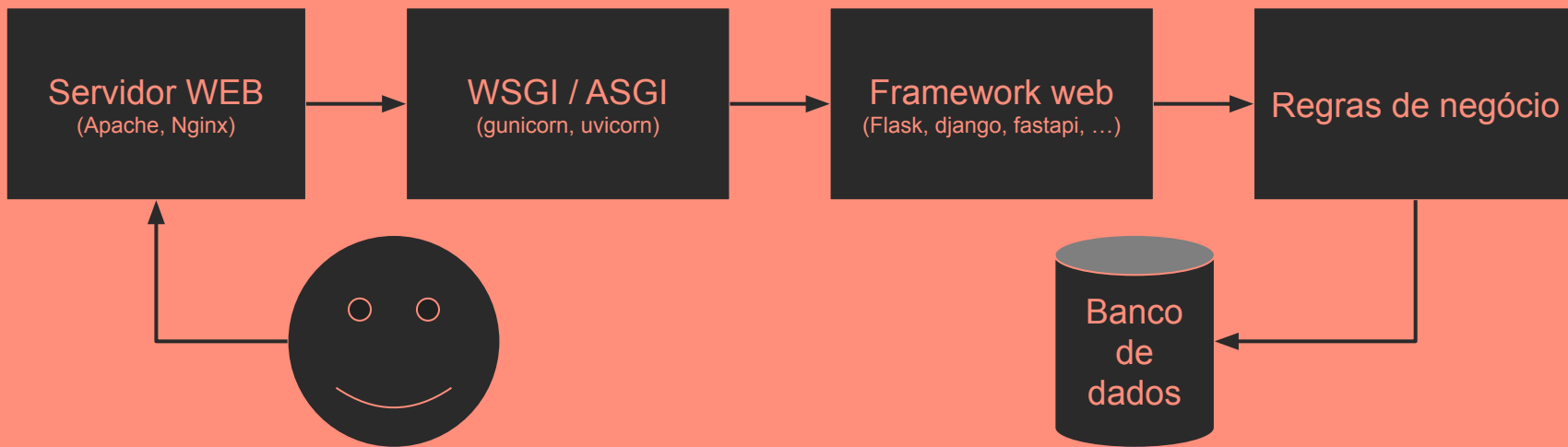
Só de exemplo!



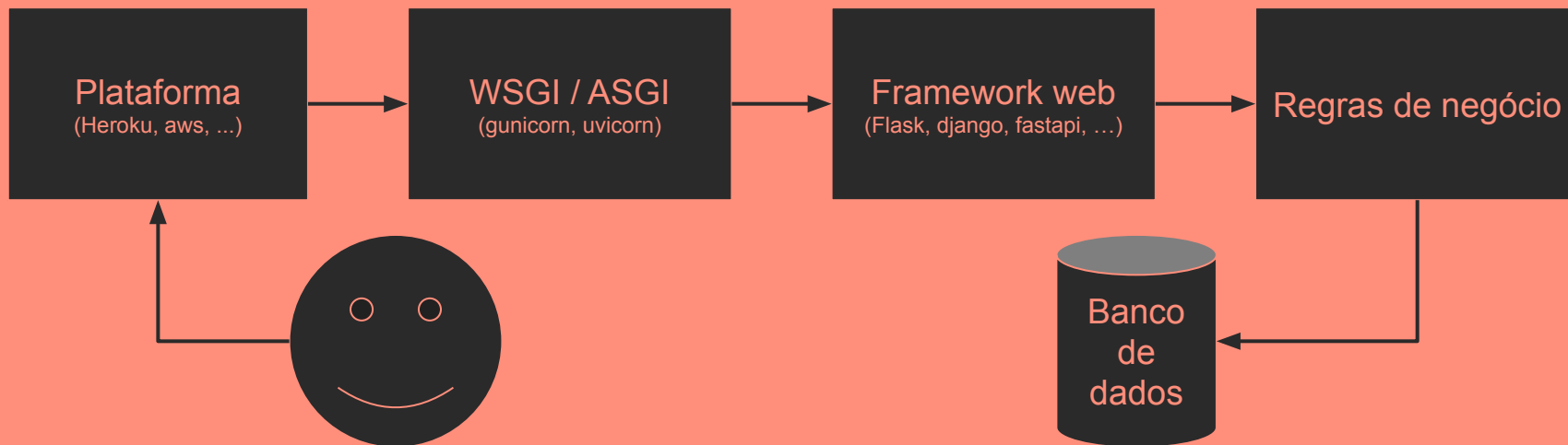
Vamos dar um passo pra trás



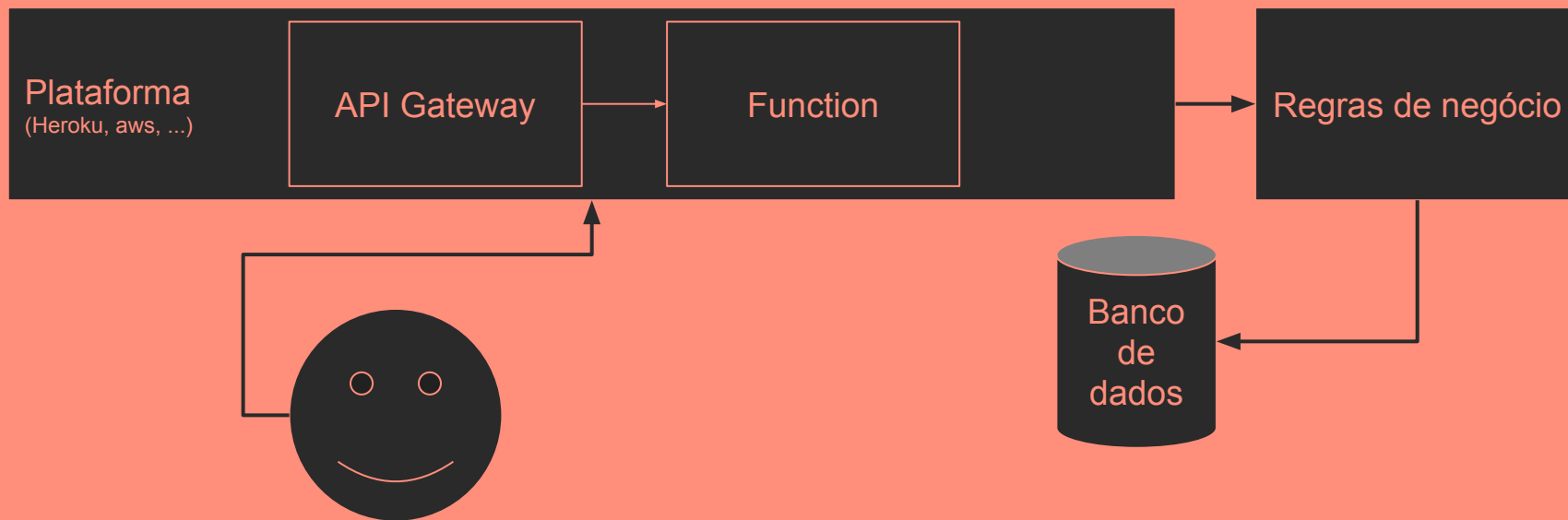
Como funciona uma aplicação "tradicional"



Modelo PaaS



Modelo FaaS



Serverless



Quanto usamos a plataforma de cloud como nosso "servidor" chamamos isso de serverless.

O que isso quer dizer? Que não temos que nos preocupar com os servidores, a cloud proveem isso.

AWS Lambda

No free tier :)

Lambda



Bom, vimos que existe uma função "perdida" na cloud. E para acionar essa função podemos contar com várias categorias de eventos diferentes.



Lambda



API gateway



LAMBDA



CloudWatch

Lambda



API gateway



LAMBDA



CloudWatch

CRON

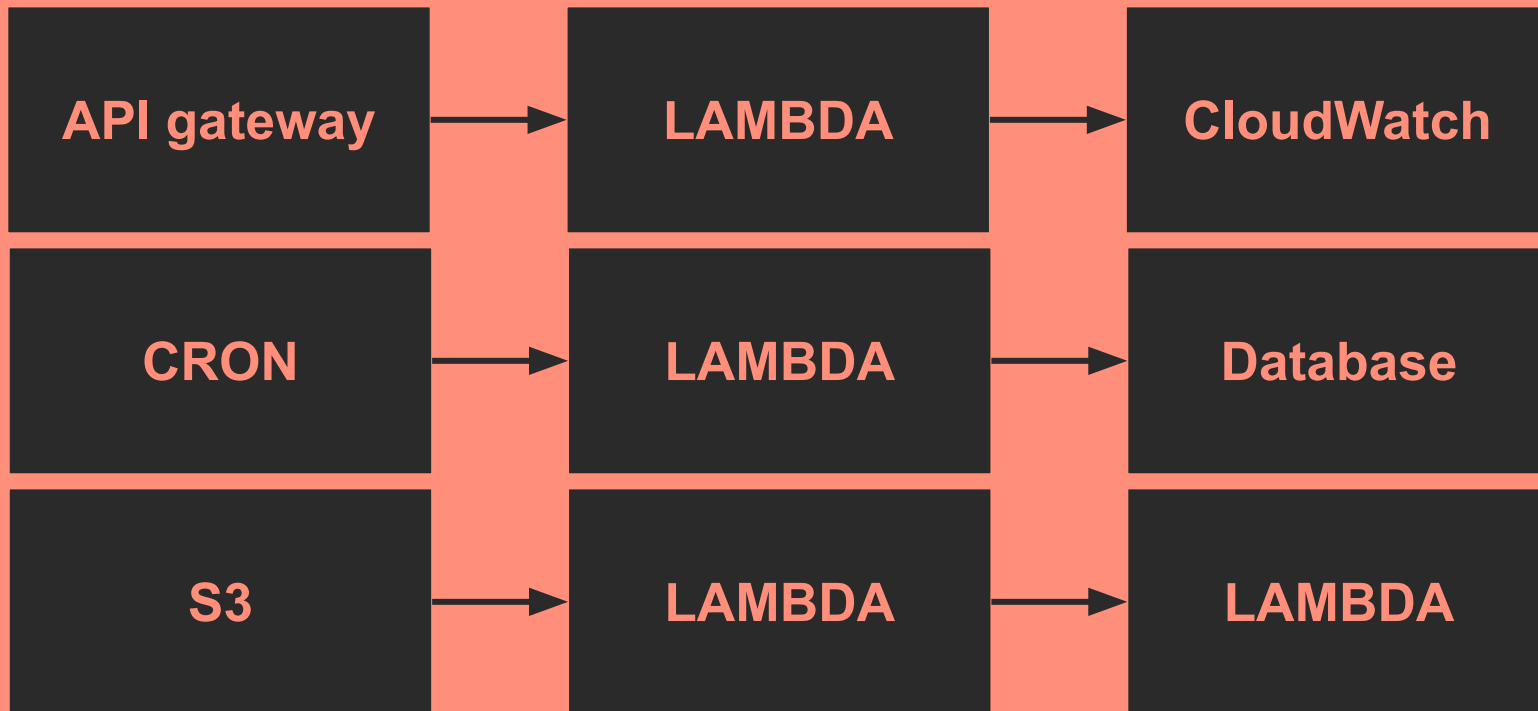


LAMBDA



Database

Lambda



Vamos criar nossa função de fato



Tudo que precisamos fazer é criar um arquivo com uma função

```
1 def lambda_handler(request, contexto):  
2     return {  
3         'mensagem': 'Olá mundo',  
4     }  
5
```

function.py

Passos para fazer deploy



- Criar uma role
- Associar essa role a uma police
- Gerar um zip do nosso arquivo
- Criar nosso lambda

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Principal": {  
7         "Service": "lambda.amazonaws.com"  
8       },  
9       "Action": "sts:AssumeRole"  
10    }  
11  ]  
12 }
```

policy.json

```
pip install awscli
```



Vamos instalar o cliente



Você deve configurar suas credenciais
da AWS no arquivo `~/.aws/credentials`

(link na descrição)



Observação



Como fazer isso no terminal?



```
1 # Cria a role
2 aws iam create-role --role-name lambda-ex --assume-role-policy-document file://policy.json
3
4 # Cria ativa a policy
5 aws iam attach-role-policy \
6     --role-name lambda-ex \
7     --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
8
9 # Cria o lambda
10 aws lambda create-function \
11     --function-name my-function \
12     --zip-file fileb://function.zip \
13     --handler lambda_handler \
14     --runtime python3.8 \
15     --role arn:aws:iam::123456789012:role/lambda-ex
```

https://docs.aws.amazon.com/pt_br/lambda/latest/dg/gettingstarted-awscli.html

SAM

Serveless Application Model



Agora as coisas ficam mais legais



Um framework
apra lambdas

chalice

Chalice



Chalice é um projeto em python feito especificamente para aplicações que rodam no lambda. Ele é quase um `flask` serverless.

```
1  from chalice import Chalice
2
3  app = Chalice(app_name='live de python')
4
5  @app.route('/')
6  def index():
7      return {
8          'message': 'Boas vindas a live de python!'
9      }
```

```
pip install chalice
```



Vamos instalar



Chalice



Vamos criar um projetinho?

```
$ chalice new-project
```

```
$ chalice local
```

```
$ chalice deploy
```

```
$ chalice logs
```

Integrando com
outros serviços

Integr
ações

Integrando



- Scheduler
- S3



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

