

# Serverless Microservice

## Como deployar um microserviço serverless



# Quem sou eu?



**Guilherme Rv Coelho**

grvcoelho

While I'm not billionaire I do things such as coding, juggling and magic for food. Working with for [@pagarme](#)

---

Pagar.me

Jundiaí, Brazil

<https://pagar.me>



**stone**<sup>co.</sup>



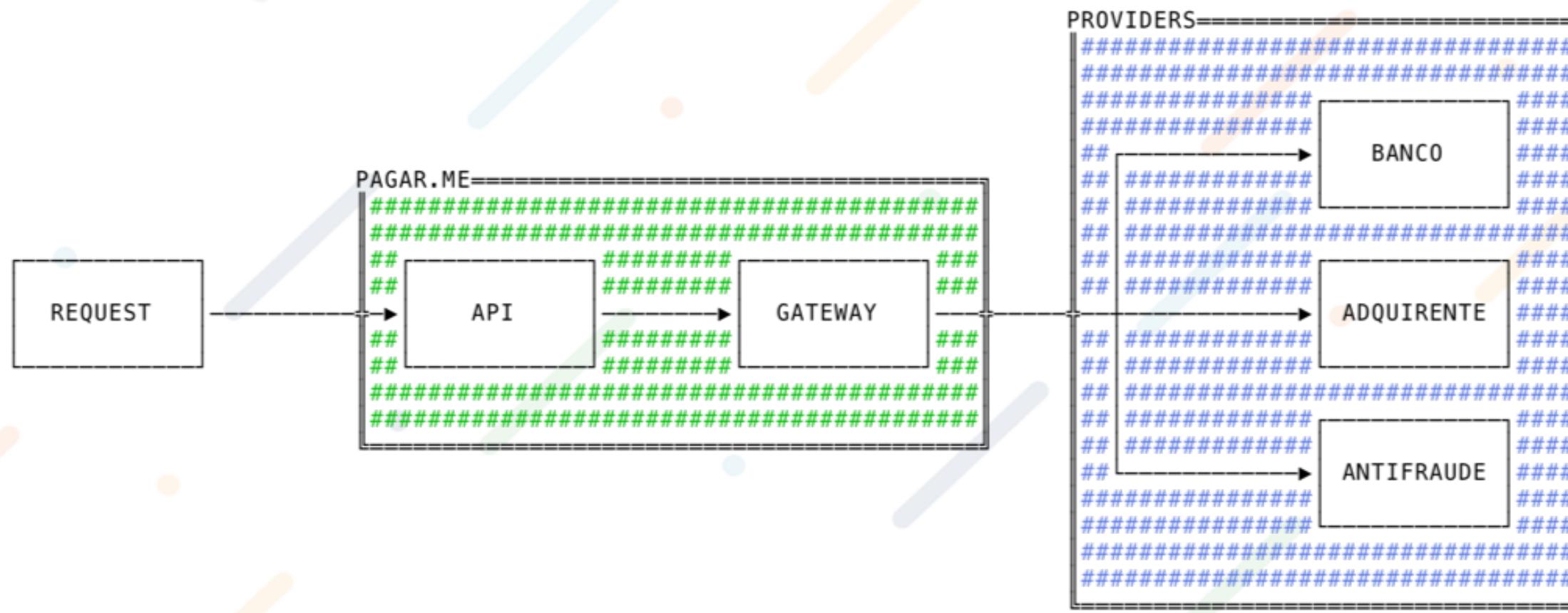
**pagar.me**



# Fluxo Transação - Pagar.me



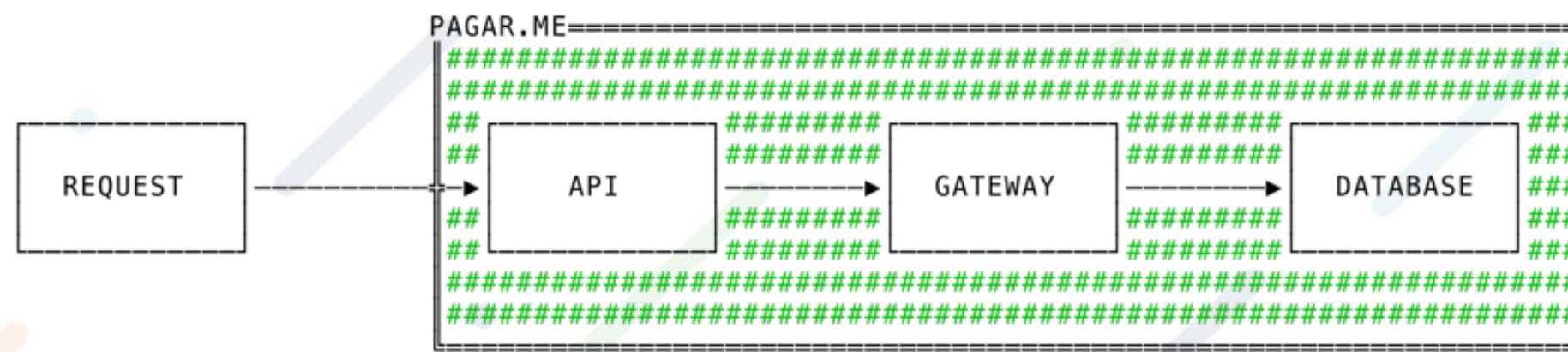
# Fluxo Transação - Pagar.me



# Fluxo Transação - Boleto



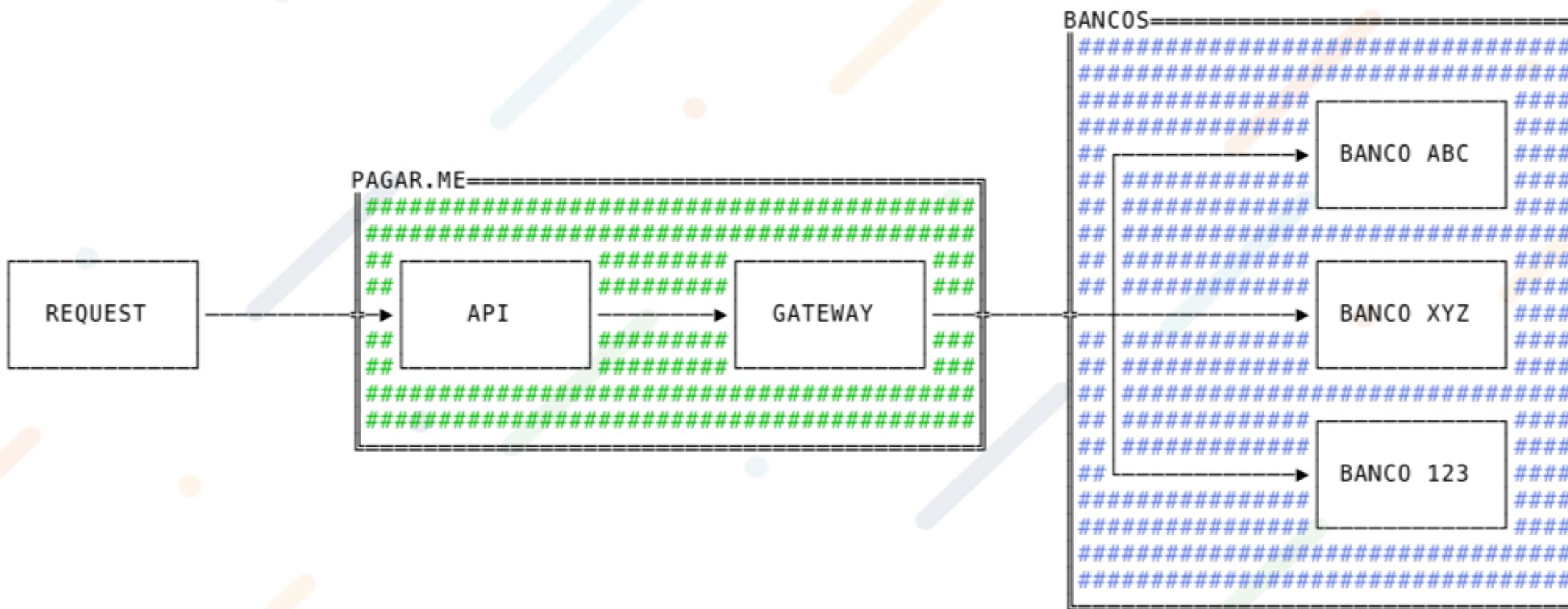
# Fluxo Transação - Boleto



# Fluxo Transação - Boleto Registrado



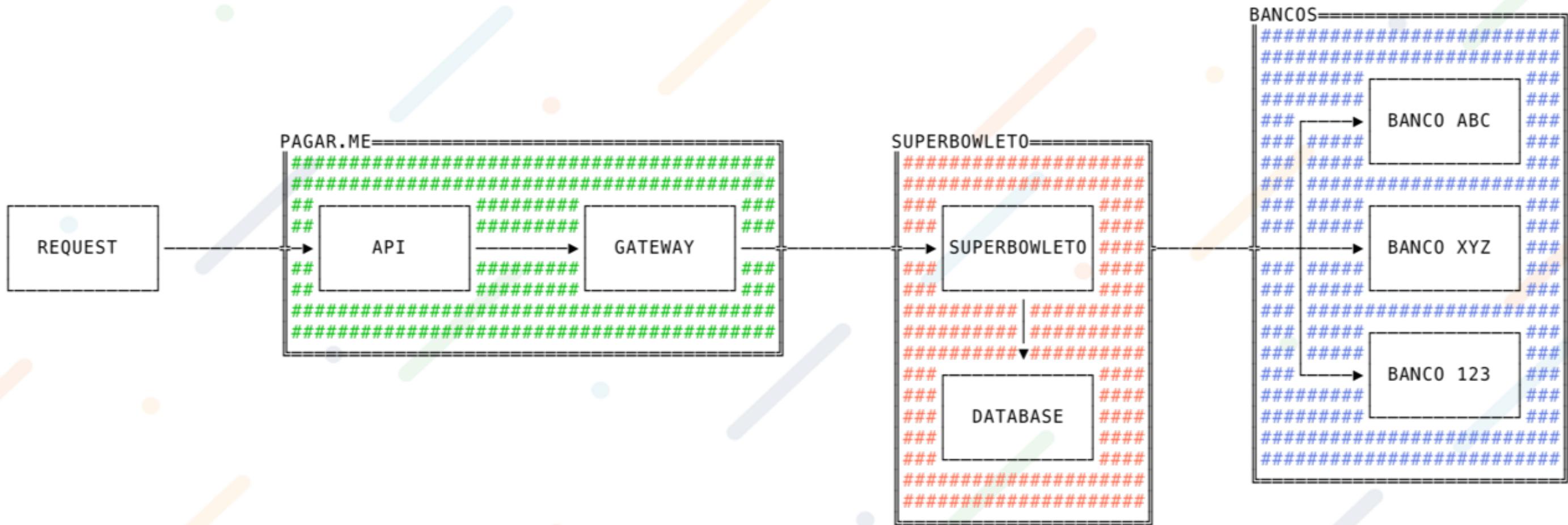
# Fluxo Transação - Boleto Registrado



# Superbowleto



# Superbowleto





# Requisitos Superbowleto



# Requisitos Superbowleto

- API Simples (CRUD)
- Salvar os dados em um banco de dados
- Fila para retentativa de registro de boleto
- Retorno via fila
- Ambientes idênticos live/sandbox





# Tecnologia





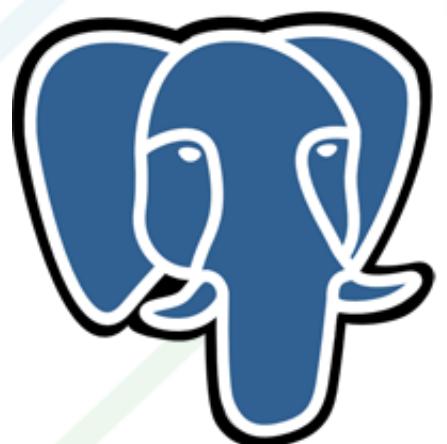
# TypeScript





# webpack





PostgreSQL





AWS SQS





# Deployando uma API Node.js



# Deployando uma API Node.js

- AutoScaling Group
- Load Balancing
- Clusters
- Service Discovery
- Monitoramento
- ...





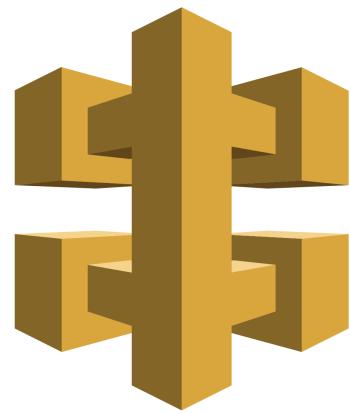
# Serverless Architecture





# AWS SQS - Filas





# API Gateway - Endpoints





# AWS Lambda - Functions



# Como criar funções Lambda



**SERVERLESS**



# serverless.yml

```
create-boleto:  
  handler: dist/boleto.create  
  events:  
    - http:  
      path: /boletos  
      method: post
```



# serverless deploy

Deploy:

```
$ serverless deploy
```

Resultado:

service: superbboleto

stage: live

region: us-east-1

endpoints:

POST - <https://mc4evvg26b.execute-api.us-east-1.amazonaws.com/live/boletos>

functions:

create-boleto: live-superboleto-create-boleto





## Criar outros recursos da aplicação



# Criar outros recursos da aplicação

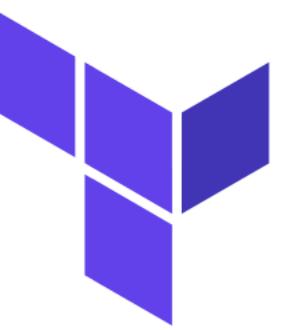
- IAM: Roles, policies, permissions
- VPC: subnets, security\_groups, route\_tables
- SQS: queues
- RDS: database



# Serverless + CloudFormation

```
resources:  
  Resources:  
    SuperbowletoQueue:  
      Type: AWS::SQS::Queue  
      Properties:  
        QueueName: boletos-to-register
```





HashiCorp  
**Terraform**



# Terraform

- Infrastructure as Code 
- Execution Plans + .tfstate 
- HCL, Variables, Interpolation, Modules, 
- Toda infra da Pagar.me hoje é Terraform 



# Terraform - Exemplo

```
variable "delay" {  
    type = "string"  
}  
  
resource "aws_queue" "superbowleto_queue" {  
    queue_name = "boletos-to-register"  
    delay_seconds = "${var.delay}"  
}  
  
output "queue_arn" {  
    value = "${aws_queue.superbowleto_queue.arn}"  
}
```



# Terraform - Apply

# Mostra tudo que vai mudar  
\$ terraform plan

# Aplica as mudanças  
\$ terraform apply



# Terraform - Módulos

```
module "sandbox" {  
  source  = "./application"  
  network = "10.0.0.0/16"  
  ...  
}  
  
module "live" {  
  source  = "./application"  
  network = "172.168.0.0/16"  
  ...  
}
```





## API Gateway - Endpoints públicos



# API Gateway - Opções

- ✗ Adicionar regras de autenticação
- ✗ Criar um "custom authorizer"
- ✓ Usar API Keys do API Gateway



**Serverless não suportava API Keys**

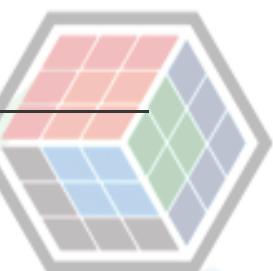


# Serverless não suportava API Keys

- CloudFormation não tinha suporte a Usage Plans<sup>1</sup>
- Criar Usage Plan no console da Amazon

---

<sup>1</sup> Problema corrigido 29 dias atrás: <https://github.com/serverless/serverless/pull/3819>



# Usar Terraform para criar endpoints



# Usar Terraform para criar endpoints

- rest\_api
- api\_resource
- api\_method
- api\_integration
- api\_deployment
- lambda\_permission
- ...



# Usar Terraform para criar endpoints

```
module "create_endpoint" {  
  source = "./api-endpoint"  
  method = "POST"  
  lambda = "${var.lambda}"  
  path   = "${var.resource_path}"  
}
```





# Senha do banco de dados



# Senha do banco de dados

-  **Premissa:** Todo mundo pode ter acesso **read** na Amazon
-  **Hard-coded no código:** senha plain-text no S3, no Cloudformation e na Lambda
-  **Env Var:** senha plain-text na Lambda





# Salvar senha no Credstash



# Credstash

- Banco de credenciais
- Key-Value store
- KMS + DynamoDB



# Salvar senha no Credstash

Deploy:

```
$ credstash put sandbox/database/password pass123
```



# Salvar senha no Credstash

Aplicação:

```
function init () {  
    return Promise.resolve()  
        .then(() => credstash.get('sandbox/database/password'))  
        .then(initDatabase)  
        .then(runFunction)  
}
```





# Senha do banco de dados no .tfstate



## .tfstate

- Representação da sua infraestrutura 
- Sem criptografia 😢
- Disponível no S3 (state\_lock) 



# Script para atualizar banco de dados

```
// Gera uma senha aleatória  
password = generatePassword()  
  
// Atualiza a instância do banco de dados  
aws.updateDatabasePassword(password)  
  
// Salva a senha no credstash  
credstash.put('database/password', password)
```



# Terraform - Local Provisioner

```
resource "aws_db_instance" "database" {  
    engine      = "postgres"  
    username   = "superbowleto"  
    password   = "touchdown1!"  
    # ...  
  
    provisioner "local-exec" {  
        command = "${path.module}/update-db-password.sh"  
    }  
}
```





# Cross-referencing entre ferramentas



# Cross-referencing entre ferramentas

- Role (terraform) -> Lambda (serverless)
- Lambda (serverless) -> API Gateway (terraform)
- Database (terraform) -> Lambda Var (serverless)





# Terraform everything!



# Terraform everything!

- Criar as Lambdas também com Terraform.
- Utilizar Serverless apenas para deployar as funções.



**✗ Serverless não atualiza Lambdas já criadas**



# Fazer o nosso próprio deployer de Lambdas



# Como deployar uma função Lambda

- Zip source files
- Upload para o S3
- Atualiza o código das funções



 pagarme / deployer

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights ▾

A tool for fetching, building, pushing and deploying applications. Edit

cli deploy rocker serverless Manage topics

16 commits 4 branches 1 release 2 contributors Apache-2.0

Tree: a03c7e9bab ▾ New pull request Create new file Upload files Find file Sourcegraph Clone or download ▾

grvcoelho committed on GitHub Merge pull request #8 from pagarme/lambda-functions ... Latest commit a03c7e9 10 days ago

build	deployer: add initial version	22 days ago
command	command: add a better help text	12 days ago
config	deployer: add initial version	22 days ago
deploy	lambda: get functions from environment	10 days ago
pipeline	deployer: add initial version	22 days ago
scm	deployer: add initial version	22 days ago
vendor	vendor: update vendors	17 days ago
.gitignore	Initial commit	22 days ago

[github.com/pagarme/deployer](https://github.com/pagarme/deployer)

## Deployer - Steps

- **SCM:** git
- **Build:** rocker
- **Deploy:** nomad, lambda



# deploy.yml

```
deploy:  
  type: lambda  
  s3_bucket: deploy-bucket  
  region: us-east-1  
  package:  
    - node_modules  
    - handler.js  
  
environment:  
  live:  
    functions:  
      - create-boleto  
      - update-boleto
```



# Deployer deploy

Deploy:

```
deployer deploy --ref "master" --env "live" superbowleto.yml
```

Chatbot:

```
sharon deploy superbowleto/master to production/live
```





**Response time > 10s**



## Response time > 10s

- Requests demorando constantemente 10, 11, 12 segundos
- Lambda espera o Event Loop do Node estar limpo
- Nós mantemos a conexão do banco de dados aberta (by design)





# Alterar o comportamento da Lambda



# Alterar o comportamento da Lambda

- `context.callbackWaitsForEmptyEventLoop`



# Alterar o comportamento da Lambda

```
export const handler = (event, context, callback) => {  
    context.callbackWaitsForEmptyEventLoop = false  
  
    // ...  
    callback(null, 'Success!')  
}
```



# Migração de banco de dados





# Lambda que roda migrations





**Futuro**



# Futuro

-  Monitoramento
-  Otimizações
-  Escalabilidade



**stone**<sup>co.</sup>

stone.co + Open Source = ❤



 pagarme / superbowleto

 Unwatch ▾ 23  Unstar 3  Fork 0

 Code

 Issues 0

 Pull requests 0

 Projects 0

 Wiki

 Settings

 Insights ▾

 A microservice to issue, register and manage boletos

 Edit

acceptance

boleto

microservice

Manage topics

 372 commits

 7 branches

 0 releases

 6 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

 Sourcegraph

 Clone or download ▾



grvcoelho committed on GitHub Merge pull request #198 from pagarme/better-credentials ... Latest commit 785390d 11 minutes ago



pr-template: add logs to checklist

2 months ago



queues: delete queue resource

2 months ago



database: use address instead of endpoint for database host connection

10 days ago



database: add migrations

a month ago



credentials: add credstash credentials getter

22 minutes ago



schema: accept token from create request

3 days ago



Add .editorconfig file

6 months ago



lint: support typescript

2 months ago

[github.com/pagarme/superbowleto](https://github.com/pagarme/superbowleto)



Obrigado ❤️

[github.com/grvcoelho](https://github.com/grvcoelho)

