

# Infraestrutura como código em 2019

Guilherme Oki

# Sobre mim

- Carioca morando em SP
- Engineering Manager na Wildlife
- Apaixonado por software livre
- Entusiasta do desenvolvimento ágil



---

# **Antes de mais nada...**

Não estou aqui para defender nenhum produto/serviço.

---

## Com isso...

O modelo de gerência de configuração usando puppet, chef, salt e ansible\* tem que acabar!

---

# Onde queremos chegar?

Nossa infraestrutura precisa ser  
imutável!!!

# Why???

"The primary advantage of using immutable infrastructure is to avoid [configuration drift](#)" - Martin Fowler

# Como atingimos?

Primeiro vamos recapitular o que surgiu na área de infraestrutura ao longo dos anos

# Voltando ao passado...

## Scripts

- Shell, Perl and PHP scripts
  - Difícil de implementar idempotência
  - Código Read-Only

# Primeiros passos...

## Orquestradores de configuração

- Libs (Capistrano e Fabric)
  - Bom para orquestrar tarefas (deploy)
  - Garantem idempotência
  - Não escala para muitas máquinas (>100)



Fabric

---

# Junto a isso surgiram...

## Gerenciadores de configuração

- Puppet, Chef, Ansible\* e Salt
  - Garantem idempotência por padrão
  - Garantem o estado desejado
  - Escopo limitado!
  - Orquestrar é difícil!

---

# Um tempo depois surgiu...

## Containers

- Docker
  - Garante imutabilidade da aplicação
  - Fácil de levar para outro ambiente
  - Rápido rollout e deploy

---

# Logo depois...

## Provisionadores de infraestrutura

- Terraform + Packer
  - Garante imutabilidade da Infra!
  - Provisiona a infraestrutura e garante o estado dela!
  - Deploy de Infra, não de Apps! (imagens muito pesadas e rollout lento)
  - Orquestrar é difícil! (Nomad pode ajudar)

---

# E agora só se fala em...

## Orquestradores de containers

- Kubernetes/Swarm
  - Resolve problemas complexos de orquestração
  - Mais complexo para Ops
  - Bem escalável

---

# **Infra imutável é fácil de alcançar?**

Com certeza não!!! Ainda existem diversos problemas  
nessa jornada!!

---

# Receita de bolo

Provisionador de Infra + Containers\*Orquestrador

Immutable + Immutable \* Scheduler = Immutable

---

# Baby steps...

- Escrever IaC para 20% dos recursos que correspondem a 80% da operação.
- Firewall, Network, IAM, DNS, ...
- Criar um fluxo de CI/CD para seu IaC (GitOps).
  - User-friendly (manter módulos e/ou boilerplate) + Docs
  - Tflint, tfenv ...

---

# Baby steps...

- Use containers para suas apps
- Não tente ser um service provider, optimize o uso de SaaS.
- Priorize ter um bom monitoramento.
- Use OpenMetrics e OpenTracing (Prometheus e Jaeger)

---

# Por fim...

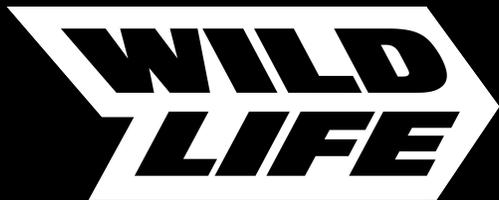
- Priorize IaC para a aplicação
  - CI/CD, Repos, ...
- Priorize IaC na camada de serviços
  - Pagerduty, Github, ...
- Priorize IaC no Kubernetes para Ops
  - ArgoCD or Flex

Dúvidas?



#WeAreHiring

<https://boards.greenhouse.io/wildlifestudios>



Obrigado!

