Break New Ground

Break New Ground

Como o chaos engineering garante a resiliência dos seus serviços

Elder Moraes | @elderjava
Developer Advocate

Julho, 2019

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Chaos Engineering é a disciplina de realizar **experimentos** em um sistema com o intuito de **construir confiança** com relação à sua capacidade de suportar **condições adversas** em **produção**.

http://principlesofchaos.org/

ORACLE

Resiliência é a capacidade de um sistema se **adaptar** diante de mudanças, falhas e anomalias

Em algum momento, **algo vai dar errado**. Mesmo nos sistemas mais bem construídos.

Crie falhas de propósito antes que elas aconteçam inesperadamente.

Encontre **fraquezas** e faça **correções**.

ORACLE

Onde injetar caos?

Aplicação

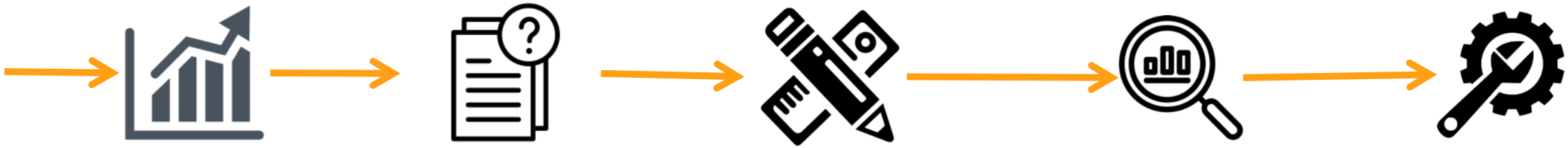
Cache

Banco de dados

Infra/Cloud

O importante é começar com “baby steps” e ganhar confiança aos poucos

Fases do Chaos Engineering



Estado estacionário

Comportamento regular de um serviço

Baseado na métrica de negócio

https://medium.com/netflix-techblog/sps-the-pulse-of-netflix-streaming-ae4db0e05f8a

Falando em métrica...

Métrica é uma medida para avaliar, controlar e/ou selecionar ***quantitativamente***: uma pessoa, um processo, um evento ou uma instituição

Métrica vs Health Check

RockyBalboaService

- Força do soco: 100%
- Sangramento: 0 ml/s
- Visão: 100%
- Nível suor: 1 ml/s
- Confiança: 10
- Pronúncia: Adrian!

- Força do soco: 0,5%
- Sangramento: 100 ml/s
- Visão: -10%
- Nível suor: 2000 ml/s (baldes)
- Confiança: 0
- Pronúncia: Ãnhãnãããnnnn

De volta ao caos

Hipótese

E se…?

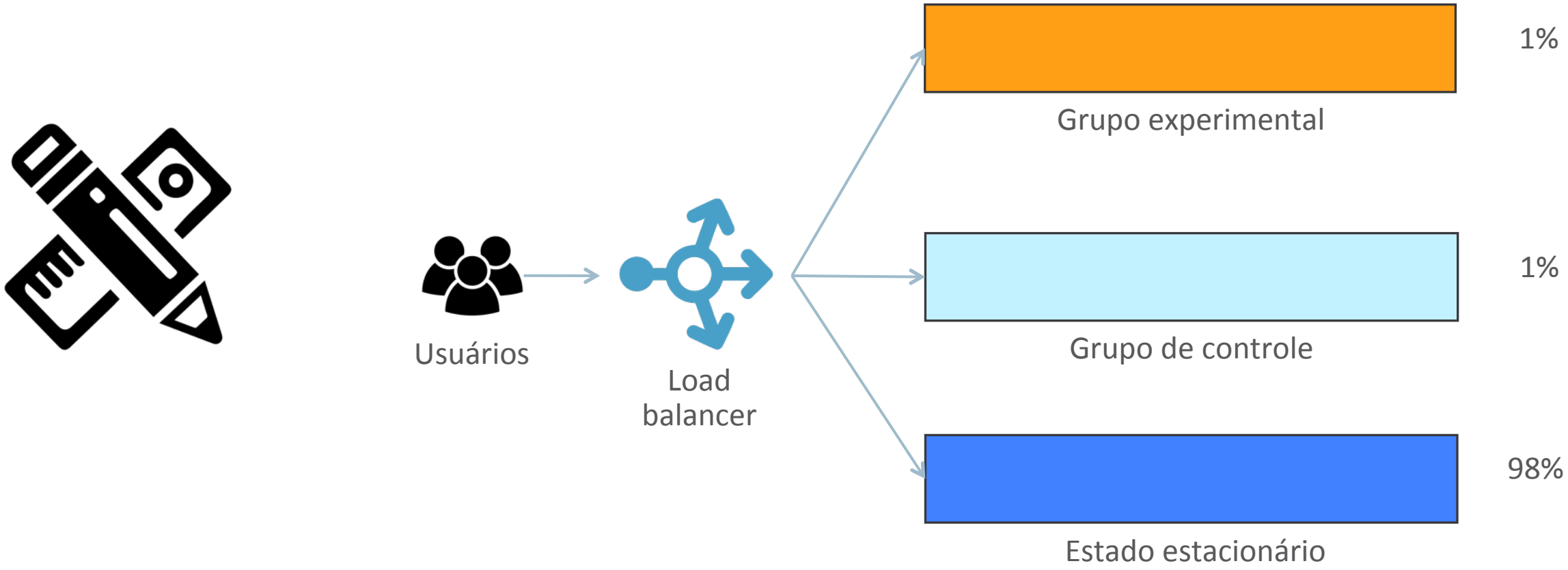
- Um serviço retornar 404
- O banco de dados parar
- O número de requests aumentar excessivamente
- A latência aumentar em 100%
- Um container parar de funcionar
- Uma porta tornar-se inacessível

Desenho do experimento

Melhores práticas

- Comece pequeno
- Mais próximo possível da produção
- Minimize o raio de ação
- Tenha um meio de parar emergencialmente
- Automatize

Desenho do experimento



Resultado e aprendizado

- Quanto tempo para detecção da falha?
- Alguém foi notificado? Quanto tempo depois?
- Houve graceful degradation? Quanto tempo para iniciar?
- Quanto tempo para auto restabelecimento (parcial e total)?
- Houve intervenção manual?
- Quanto tempo até retornar ao estado estacionário?

Importante: o objetivo não é achar culpados! ;-)

Correção

“We learn from failure, not from success”

– Dracula, Bram Stoker

Kubernetes & Chaos Engineering

Porque falar sobre Chaos & Kubernetes?

- Funcionalidades nativas para resiliência
- Restart de pods baseados em health-check/status
- Distribuição de pods em diferentes regiões

Algumas ferramentas para Chaos com Kubernetes

- Istio
- Chaos Toolkit
- Chaos Monkey
- Chaos Kong
- Kube Monkey

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
  ...
spec:
  hosts:
  - ratings
  http:
  - fault:
      delay:
        fixedDelay: 7s
        percent: 100
    match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
```

Injeção de falhas com Istio

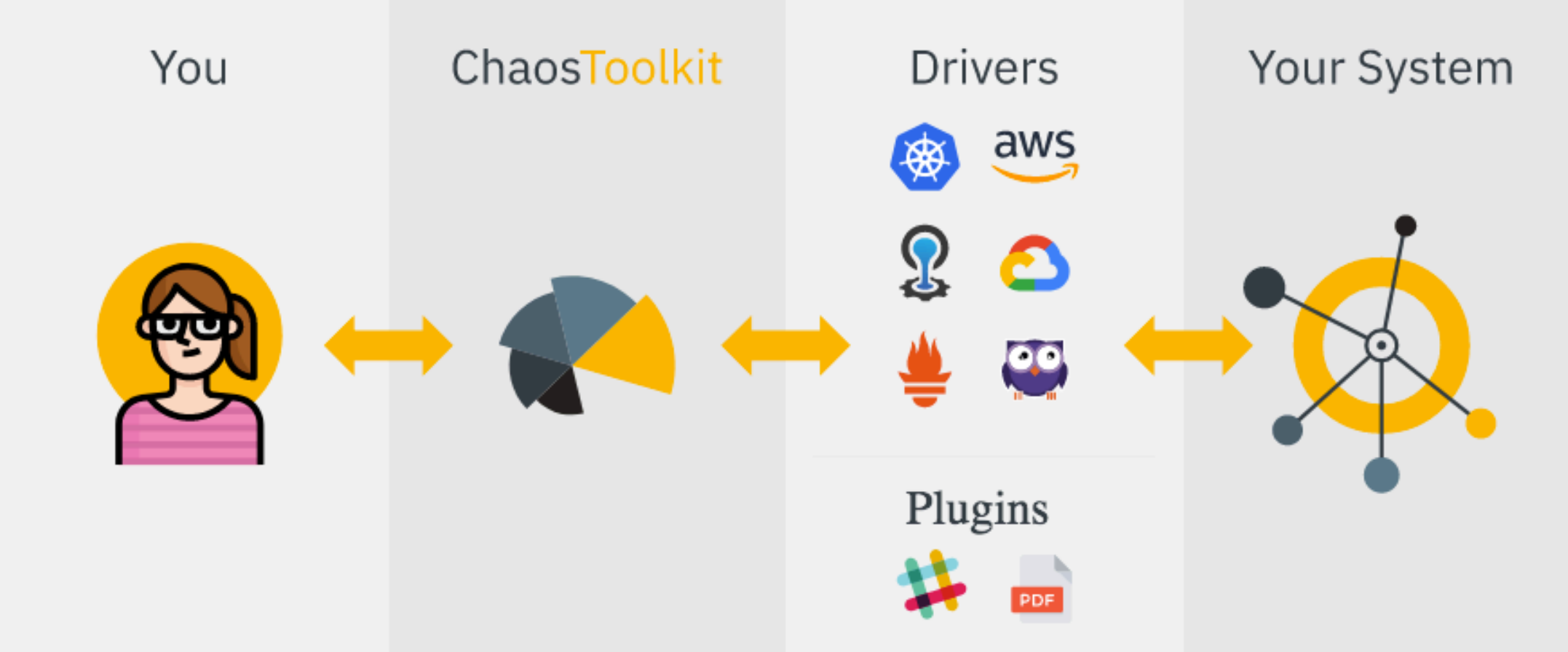
https://istio.io/docs/tasks/traffic-management/fault-injection/

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
  ...
spec:
  hosts:
  - ratings
  http:
  - fault:
      abort:
        httpStatus: 500
        percent: 100
    match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
```

Injeção de falhas com Istio

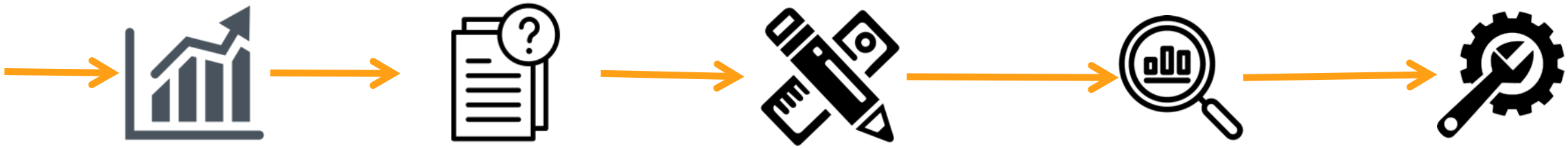
https://istio.io/docs/tasks/traffic-management/fault-injection/

Chaos Toolkit



https://github.com/chaostoolkit/chaostoolkit

Chaos Toolkit – Segue as fases de Chaos Engineering



https://github.com/chaostoolkit/chaostoolkit

Chaos Toolkit – Suporte ao Kubernetes

https://github.com/chaostoolkit/chaostoolkit-kubernetes

```
{
    "name": "all-our-microservices-should-be-healthy",
    "type": "probe",
    "tolerance": "true",
    "provider": {
        "type": "python",
        "module": "chaosk8s.probes",
        "func": "microservice_available_and_healthy",
        "arguments": {
            "name": "myapp",
            "ns": "myns"
        }
    }
},
{
    "type": "action",
    "name": "terminate-db-pod",
    "provider": {
        "type": "python",
        "module": "chaosk8s.pod.actions",
        "func": "terminate_pods",
        "arguments": {
            "label_selector": "app=my-app",
            "name_pattern": "my-app-[0-9]$",
            "rand": true,
            "ns": "default"
        }
    },
    "pauses": {
        "after": 5
    }
}
```

Chaos Monkey

https://github.com/netflix/chaosmonkey

- “Mata” aleatoriamente máquinas virtuais e/ou containers em produção
- Funciona integrado ao Spinnaker (https://www.spinnaker.io/)
- Suporte a AWS, GCE, Azure, Cloud Foundry e Kubernetes

https://medium.com/netflix-techblog/tagged/chaos-kong

- “Mata” uma região inteira do provedor de cloud
- Esse tipo de incidente é raro, mas acontece

Kube Monkey

https://github.com/asobti/kube-monkey

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: monkey-victim
  namespace: app-namespace
  labels:
    kube-monkey/enabled: enabled
    kube-monkey/identifier: monkey-victim
    kube-monkey/mtbf: '2'
    kube-monkey/kill-mode: "fixed"
    kube-monkey/kill-value: '1'
spec:
  template:
    metadata:
      labels:
        kube-monkey/enabled: enabled
        kube-monkey/identifier: monkey-victim
```

Kube Monkey

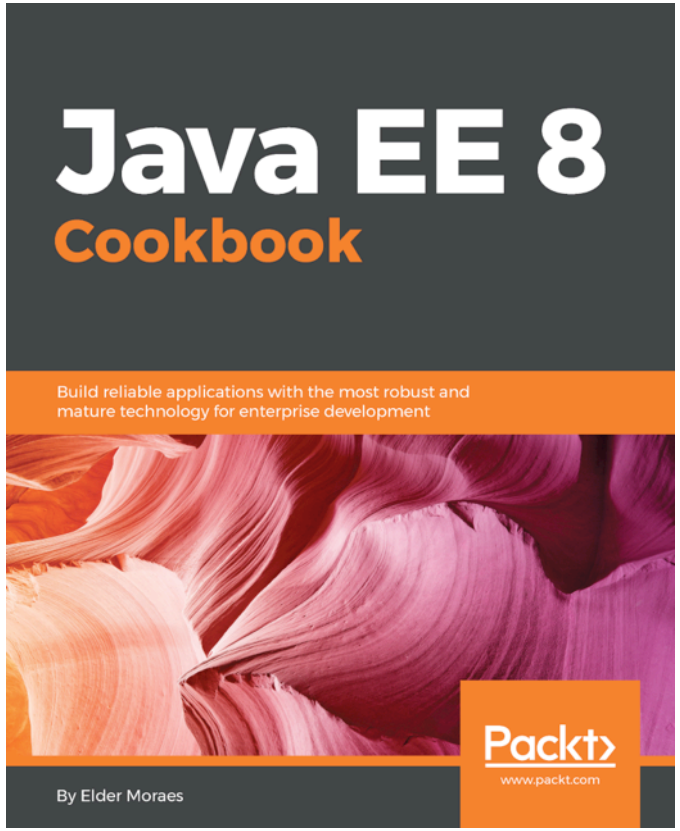
https://github.com/asobti/kube-monkey

```
KUBEMONKEY_DRY_RUN=true
KUBEMONKEY_RUN_HOUR=8
KUBEMONKEY_START_HOUR=10
KUBEMONKEY_END_HOUR=16
KUBEMONKEY_BLACKLISTED_NAMESPACES=kube-system
KUBEMONKEY_TIME_ZONE=America/New_York
```


LET'S PRAY
TO THE DEMO GODS
memegenerator.net

"Chaos doesn't cause problems. It reveals them."

– Nora Jones, Senior Chaos Engineer, Netflix



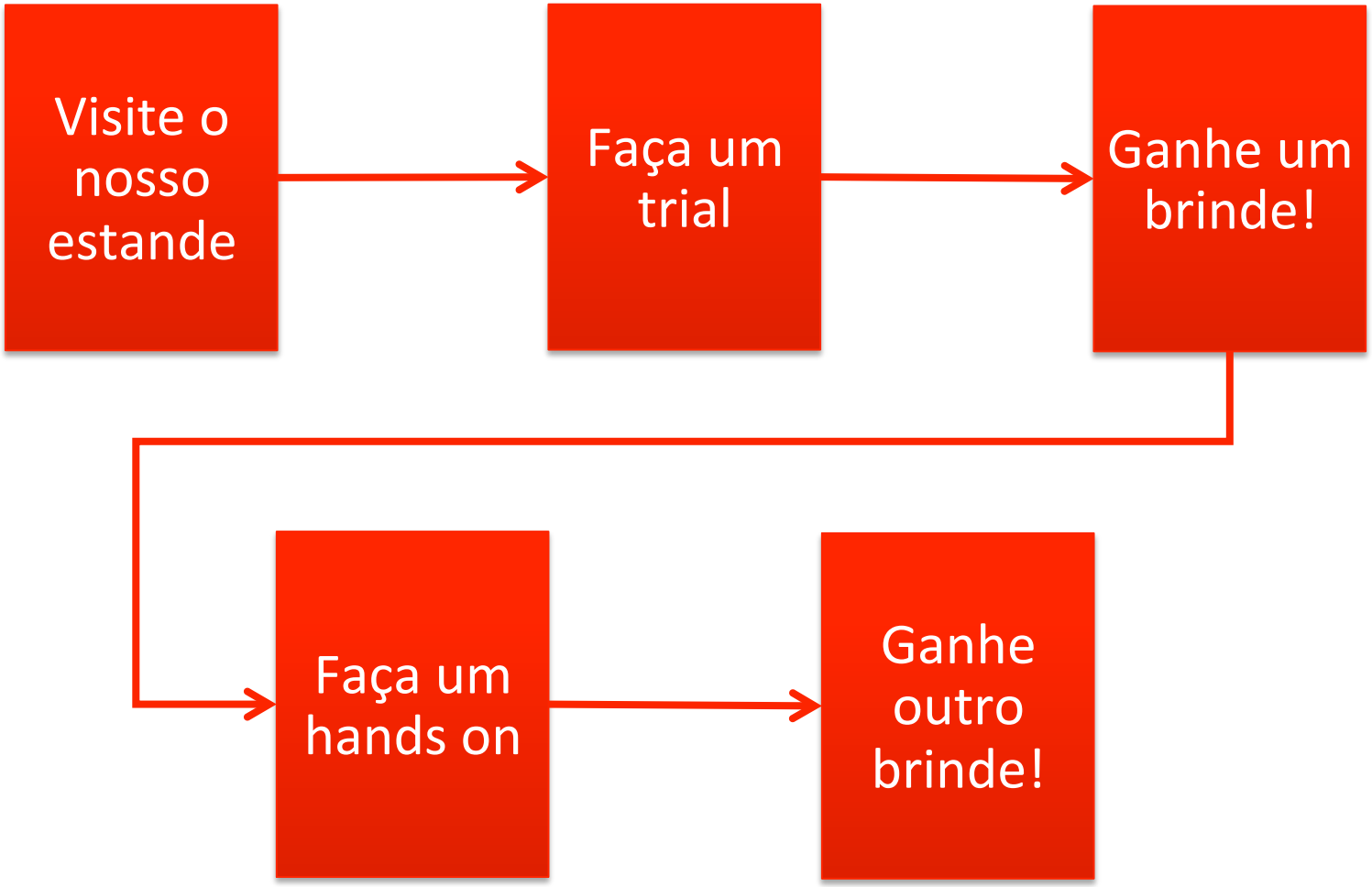
@elderjava

book.eldermoraes.com

ORACLE

Java@Cloud Age

bit.ly/javacloudage



Break New Ground