

# Chaos Engineering

Princípios e principais ferramentas



# Olá! Eu sou o Marcelo!

Caso você queira me encontrar nas redes sociais:



**[linkedin.com/in/mschirbel](https://linkedin.com/in/mschirbel)**



**[medium.com/@mschirbel](https://medium.com/@mschirbel)**

# Considerações iniciais

- Todo o material está disponível no GitHub  
[github.com/mschirbel/cursos](https://github.com/mschirbel/cursos);
- Podem fazer perguntas, sempre que quiserem!

---

# Sobre o que falaremos hoje

## **Definições**

Alguns conceitos que são importantes.

## **Pré requisitos**

Como dividir e aplicar o CE

## **Aspectos técnicos**

Fundamentos do Chaos Engineering

## **Principais ferramentas**

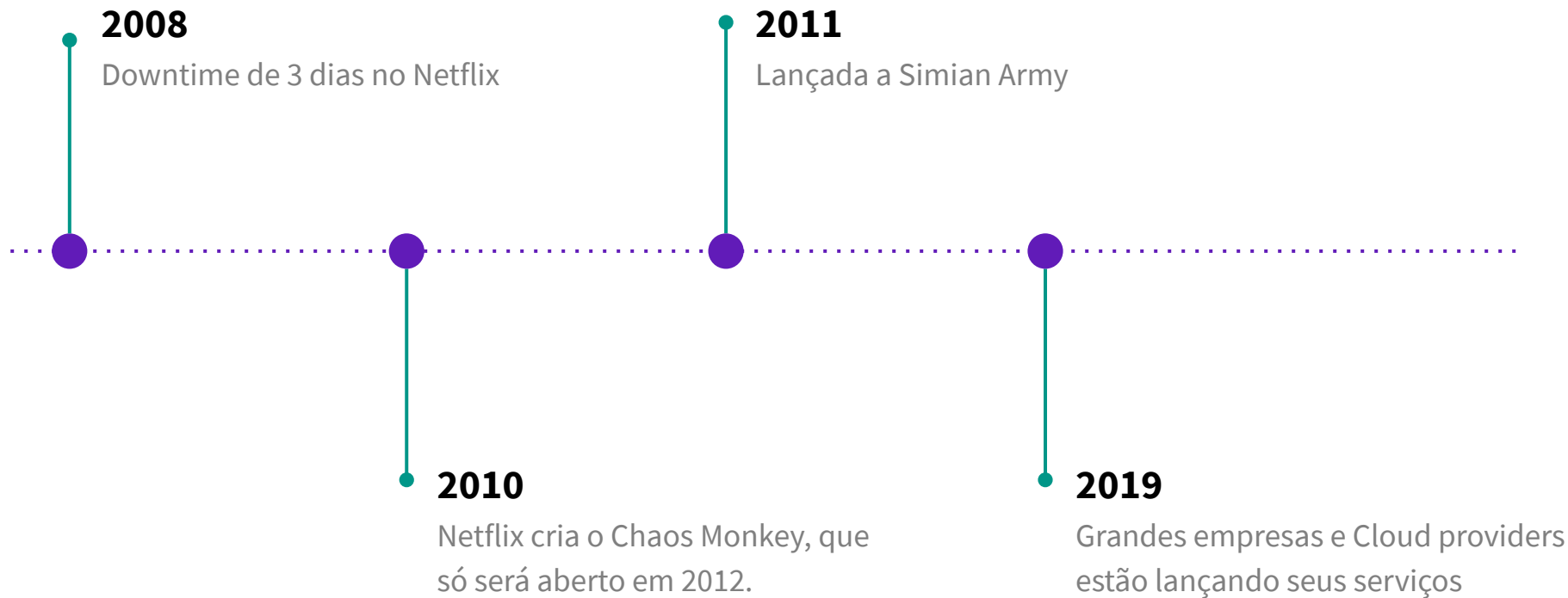
Algumas ferramentas que chamam atenção no mercado

# Definições

# O que é Engenharia do Caos?

- Caos vem do grego χάος;
- Afetar um sistema determinístico com comportamentos aleatórios;
- Experimentos que revelam fraquezas de sistemas.

# História do Chaos Engineering



# Conway's Law

(não é o do Jogo da Vida)

**Afirma que organizações que  
projetam sistemas são  
constrangidos a produzir projetos  
que são cópias das estruturas de  
comunicação dessas organizações**

---



# Princípios

# Tenha um estado padrão

- Saídas determinadas;
- Estabelecido por um tempo;
- Métricas apropriadas.

---

# Crie hipóteses plausíveis

- Queda de uma instância;
- Uma entrada incorreta;
- Latência entre serviços;
- Queda de uma região.

# Faça testes em produção

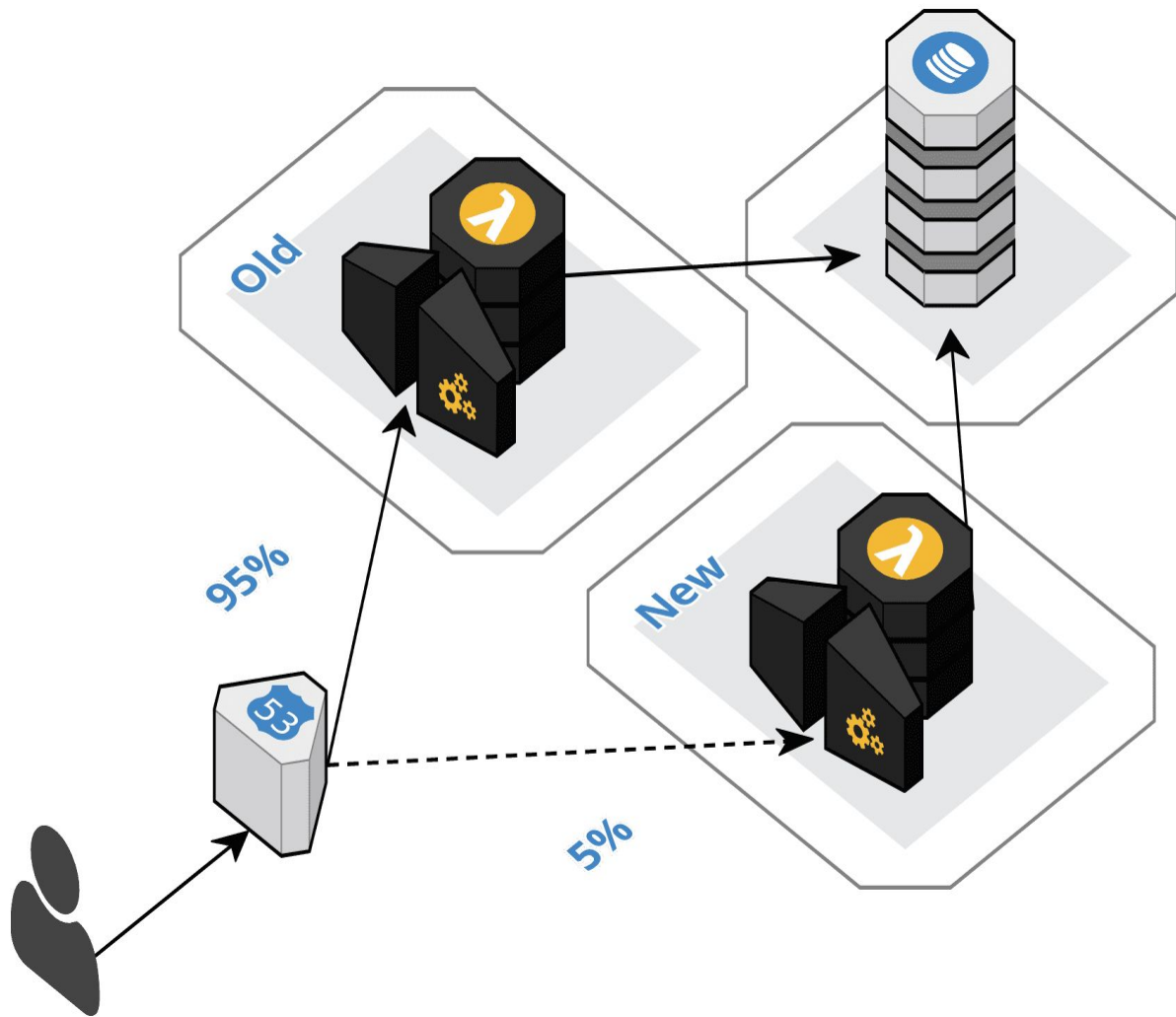
- É impossível determinar todas as entradas e saídas de um sistema;
- Quanto mais testes, mais erros.

# Canary Deployment

Exportamos uma nova versão ao ambiente de produção.

Mas somente 5%(ou um valor menor) receberá essa atualização.

Por métricas definidas, sabemos se devemos ou não atualizar



# Experimente sempre

- Os experimentos de Chaos devem rodar continuamente;
- Em diversos horários e sob diversas condições.

---

# Minimize o impacto

- O cliente não deve sentir os experimentos;
- Saber quando fazer o fallback de algo errado.

---

# Pré-requisitos



# Uma cultura que não visa encontrar um culpado

- Todos erram. E todos devem aprender com os erros;
- Alguém com medo de errar nunca vai fazer o experimento necessário.

# Tenha hipóteses pequenas

- No caso do Netflix percebemos que sempre tratavam dessa forma;
- Apesar de pequenas, os objetivos eram grandes.

---

# Confiança nos times

- Confiança no seu time;
- Confiança nos outros times.

---

# Ferramentas

## Simian Army

Simian significa macaco.

Coleção de ferramentas de C.E.  
em código aberto.

A maioria delas foi  
descontinuada.

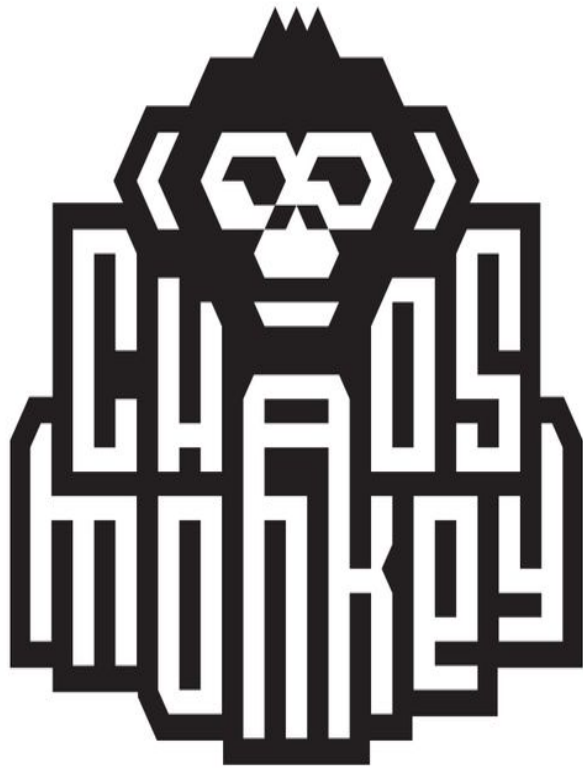


# Chaos Monkey

Primeira ferramenta a ser produzida.

Hoje está em código aberto e fora da Simian Army.

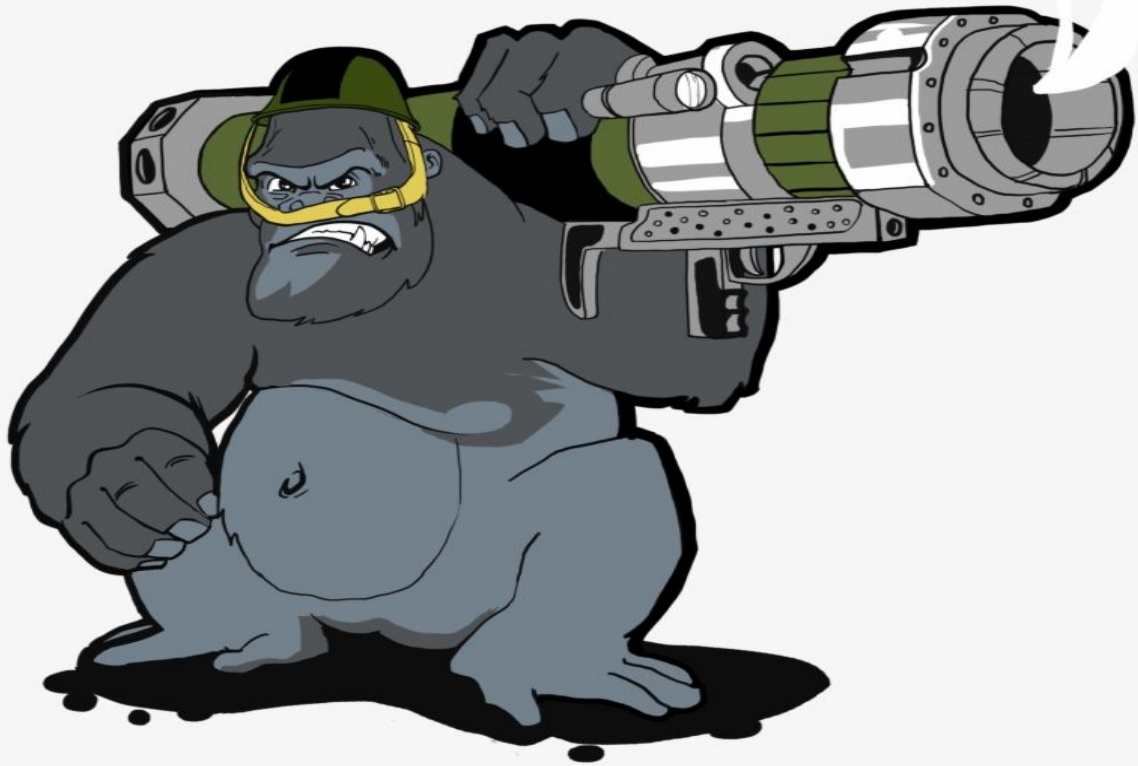
Derruba instâncias na AWS, desde que integrado com o Spinnaker.



# Chaos Gorilla

Foi descontinuada.

Simular a queda de uma Zona de Disponibilidade da AWS.



## Chaos Kong

Foi descontinuada na versão standalone.

Migrada para outra ferramenta.

Simular a queda de uma região inteira na AWS.





# Chaos Kube

Ferramenta muito simples.

Derruba algum Pod do  
Kubernetes, aleatoriamente.

Usamos aqui na Nexa!



# *chaoskube*

# Chaos Blade

Chaos Monkey da Alibaba.

Funciona em diversos sistemas.

Pode ser usado por linha de comando.

Parte das docs estavam todas em chinês :/



再去观察 CPU 情况，CPU 负载已回到正常状态：

```
CPU usage: 6.36% user, 4.74% sys, 88.88% idle
```

一次 CPU 满载演练完成。

## 你的第二个混沌实验

这次实验，我们演练 Dubbo 应用，我们的需求是 consumer 调用 com.alibaba.demo.HelloService 服务下的 hello 接口延迟 3 秒。接下来我们下载所需要的 Dubbo Demo：

dubbo-provider  
dubbo-consumer

下载完成后，执行以下命令启动应用，注意必须先启动 dubbo-provider，然后再启动 dubbo-consumer：

```
# 启动 dubbo-provider
nohup java -Djava.net.preferIPv4Stack=true -Dproject.name=dubbo-provider -jar dubbo-provider-1.0-SNAPSHOT.jar &

# 稍等 2 秒，然后启动 dubbo-consumer
nohup java -Dserver.port=8080 -Djava.net.preferIPv4Stack=true -Dproject.name=dubbo-consumer -jar dubbo-consumer-1.0-SNAPSHOT.jar &
```

访问 `http://localhost:8080/hello?msg=world`，返回以下信息，表示启动成功：

```
{
  msg: "Dubbo Service: Hello world"
}
```

接下来我们要使用 blade 工具进行混沌实验，在执行实验前，我们需要先执行 prepare 命令，挂载所需要的 java agent：

# Chaos Blade

Chaos Monkey da Alibaba

Funciona em diversos sistemas

Pode ser usado por linha de comando.

Parte das docs estavam em chinês :/



## ChAP: Chaos Automation Platform



Netflix Technology Blog

Follow

Jul 26, 2017 · 5 min read



We are excited to announce ChAP, the newest member of our chaos tooling family! [Chaos Monkey](#) and [Chaos Kong](#) ensure our resilience to instance and regional failures, but threats to availability can also come from disruptions at the microservice level. [FIT](#) was built to inject microservice-level failure in production, and ChAP was built to overcome the limitations of FIT so we can increase the safety, cadence, and breadth of experimentation.

At a high level, the platform interrogates the deployment pipeline for a user-specified service. It then launches experiment and control clusters of that service, and routes a small amount of traffic to each. A specified FIT scenario is applied to the experimental group, and the results of the

## Netflix ChAP

Chaos Automated Platform.

Código ainda não divulgado.

Pelo blog, parece ser muito completo.

Capaz de realizar múltiplos experimentos

## Gremlin

Dashboard para controle de experimentos de Chaos Engineering.

Suporta múltiplas plataformas.

Muito intuitivo.

Tem uma versão paga.



# GREMLIN

[Attacks](#) > New

## Choose The Targets

Specify the coverage and details for impact.



0



## Hosts

## Containers

## No active hosts.

Install the Gremlin client on any host you wish to attack.

## Random

Designate a random amount of targets to be impacted.

Off





## Clients

### Infrastructure

---

#### Install Gremlin

Activate on any host you wish to attack

[AWS](#)[Azure](#)[CentOS 7](#)[Docker](#)[Ubuntu 16.04](#)[View Installation Guide](#)

# Conteúdo sobre Chaos Engineering



# Netflix Tech Blog

Experimentos realizados no Netflix.

Resultados e previsões.

Área somente sobre Chaos Engineering.

TAGGED IN

## Chaos Engineering



Netflix TechBlog

Learn about Netflix's world class engineering efforts, company culture, product developments and more.

More information

FOLLOWERS  
77K

MORE, ON MEDIUM

Chaos Engineering



Netflix Technology Blog in Netflix TechBlog

Jul 26, 2017 · 5 min read

### ChAP: Chaos Automation Platform

We are excited to announce ChAP, the newest member of our chaos tooling family! [Chaos Monkey](#) and [Chaos Kong](#) ensure our resilience to instance and regional failures, but threats to availability can also come from disruptions at the microservice level. [FIT](#) was built to inject...

Read more...



523

4 responses



Netflix Technology Blog in Netflix TechBlog

Oct 19, 2016 · 4 min read



### Netflix Chaos Monkey Upgraded



...are pleased to announce a significant upgrade to one of our more popular

# Gremlin Tutorials

Diversos tutoriais sobre  
diversas ferramentas.

Comunidade muito presente.



Canal do Slack bem ativo.

**Gremlin** | [Community](#) [Tutorials](#) [Events](#) [Talks](#) [Slack](#) [Product](#) [USE FOR FREE](#)

## Explore the tutorials.


Learn how to build resilient systems through these hands-on how-to's.



### CHAOS ENGINEERING



#### Chaos Engineering Monitoring & Metrics Guide


IntroductionPrior to your first Chaos Engineering experiments it is important to collect a specific set of metrics...



**Tammy Butow**  
Principal Site Reliability Engineer



#### Email Template: Internal Chaos Engineering Report - 10x Reduction In Incidents


Creating Your Own Chaos Engineering ReportsWhen you start to practice Chaos Engineering it is important to ensure you...

**Tammy Butow**  
Principal Site Reliability Engineer





#### Chaos Engineering: the history, principles, and practice

With the rise of microservices and distributed cloud architectures, the web has grown increasingly complex. We all...


**Gremlin**  
Chaos Engineer



### GAMEDAYS



#### How to Run a GameDay


GameDays were coined by Jesse Robbins when he worked at Amazon and was responsible for availability. Jesse created...



**Eugene Wu**



#### What I Learned Running the Chaos Lab: Kafka Breaks


At Chaos Conf, a few weeks ago, we helped dozens of folks plan GameDays around the critical dependencies of their apps...

**Kyla Hultman**



#### Planning Your Own Chaos Day

What is a Chaos Day?A Chaos Day is a dedicated team day focused on using Chaos Engineering to reveal weaknesses in your...

**Tammy Butow**

# Pavlos Ratis

Lista de conteúdo sobre Chaos Engineering.

Livros, E-books, Podcasts, Apresentações, Artigos e Pessoas.

dastergon / awesome-chaos-engineering

Sponsor

Watch 209

Star 2,443

Fork 229

Code

Issues 1

Pull requests 1

Security

Insights

A curated list of Chaos Engineering resources.

netflix-chaos-monkey

chaos-engineering

chaos-monkey

simian-army

site-reliability-engineering

resilience

chaos

chaos-community

chaos-testing

awesome

awesome-list

182 commits

1 branch

0 releases

30 contributors

CC0-1.0

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

dastergon

minor cosmetic fix

Latest commit c72dc32 2 days ago

.github

add buymeacoffee in funding page2 months ago

CONTRIBUTING.md

add contributing file2 years ago

LICENSE

add license2 years ago

README.md

minor cosmetic fix2 days ago

README.md

Awesome Chaos Engineering

awesome

A curated list of awesome [Chaos Engineering](#) resources.

What is Chaos Engineering?

Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production. - [Principles Of Chaos Engineering](#) website.

Contents

Culture

Books

**Alguma dúvida?**