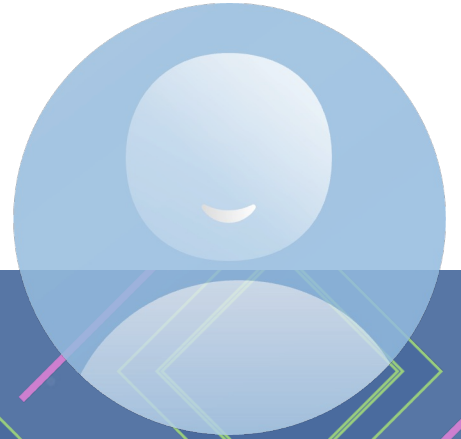
PUC Minas
Virtual

Práticas Técnicas Avançadas DevOps

Marco Mendes



PUC Minas
Virtual

Conteineirização de Ambientes

Lembrete - O primeiro caminho do DevOps

Para maximizar o fluxo, precisamos tornar o trabalho visível, reduzir o tamanho dos lotes e os intervalos de trabalho, aumentar a qualidade evitando que os defeitos sejam passados para os centros de trabalho mais à direita e otimizar constantemente as metas globais.

Ambientes de baixa maturidade

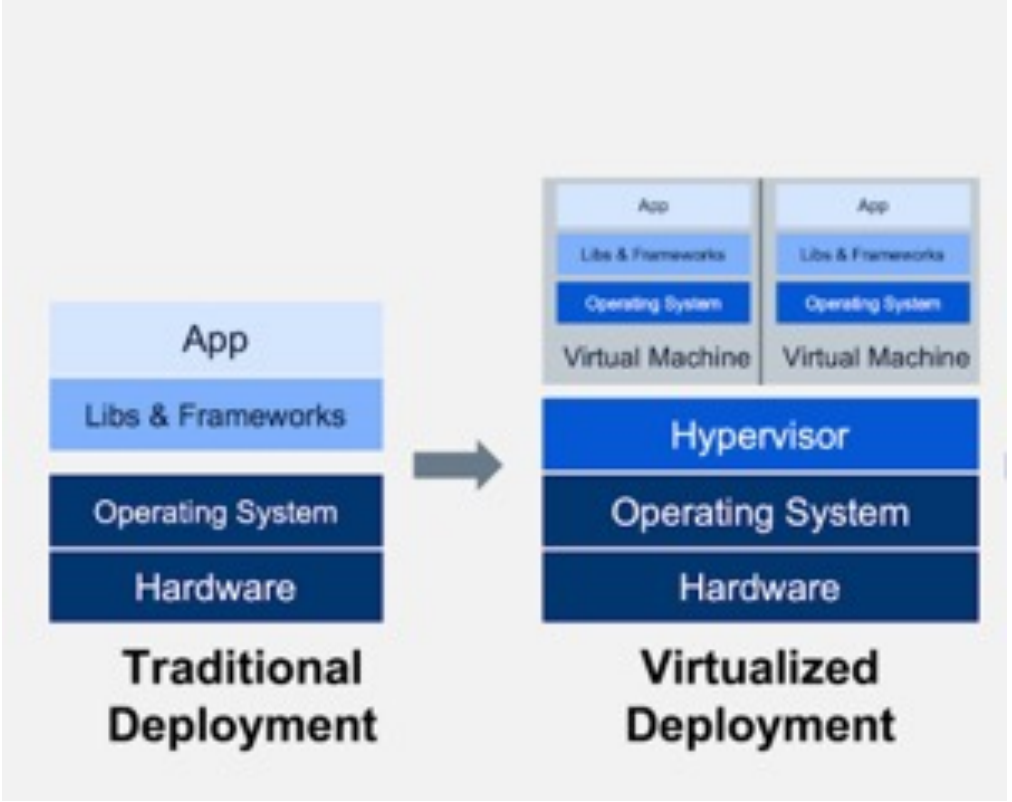
Mas.. o trabalho tradicional da infraestrutura física em muitas empresas é moroso e repetitivo.

Publicação em produção em ambientes de baixa maturidade

- Até pouco tempo atrás, a única forma de disponibilizar software em produção era através de acesso de trabalho manual de:
  - Provisionamento manual de hardware
  - Instalação manual de SOs e servidores
  - Configuração manual de aplicativos

O surgimento de tecnologias de virtualização como Hypervisor e programas como o VMWare e VirtualBox começou a facilitar a administração de servidores.

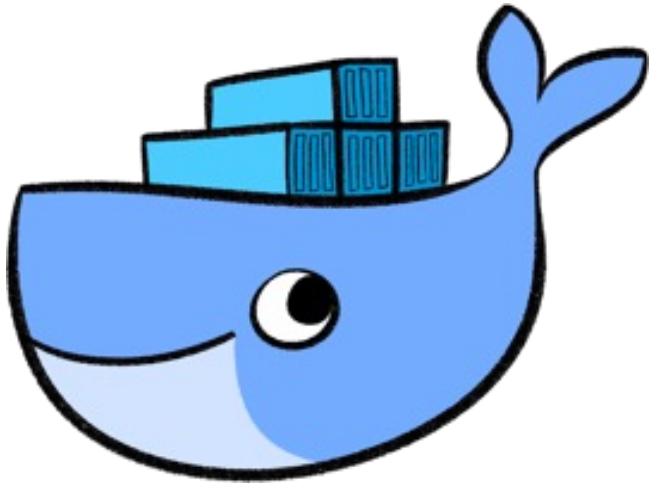
Virtualização de Ambientes



Fonte: https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/

Entram em cena os contêineres leves

- Produtos como o Docker foram criados a partir da evolução de conceitos existentes no Unix nos últimos 30 anos
  - 1979 - chroot system calls
  - 2000 - Free BSD Jails
  - 2001 - Linux Vservers
  - 2004 - Solaris Containers
  - 2006 - Cgroups (Control Groups)
  - 2013 - Docker



Conteineres

- São um método de virtualização em nível de sistema operacional que permite executar uma aplicação e suas dependências como processos e com recursos isolados que simulam uma máquina virtual.

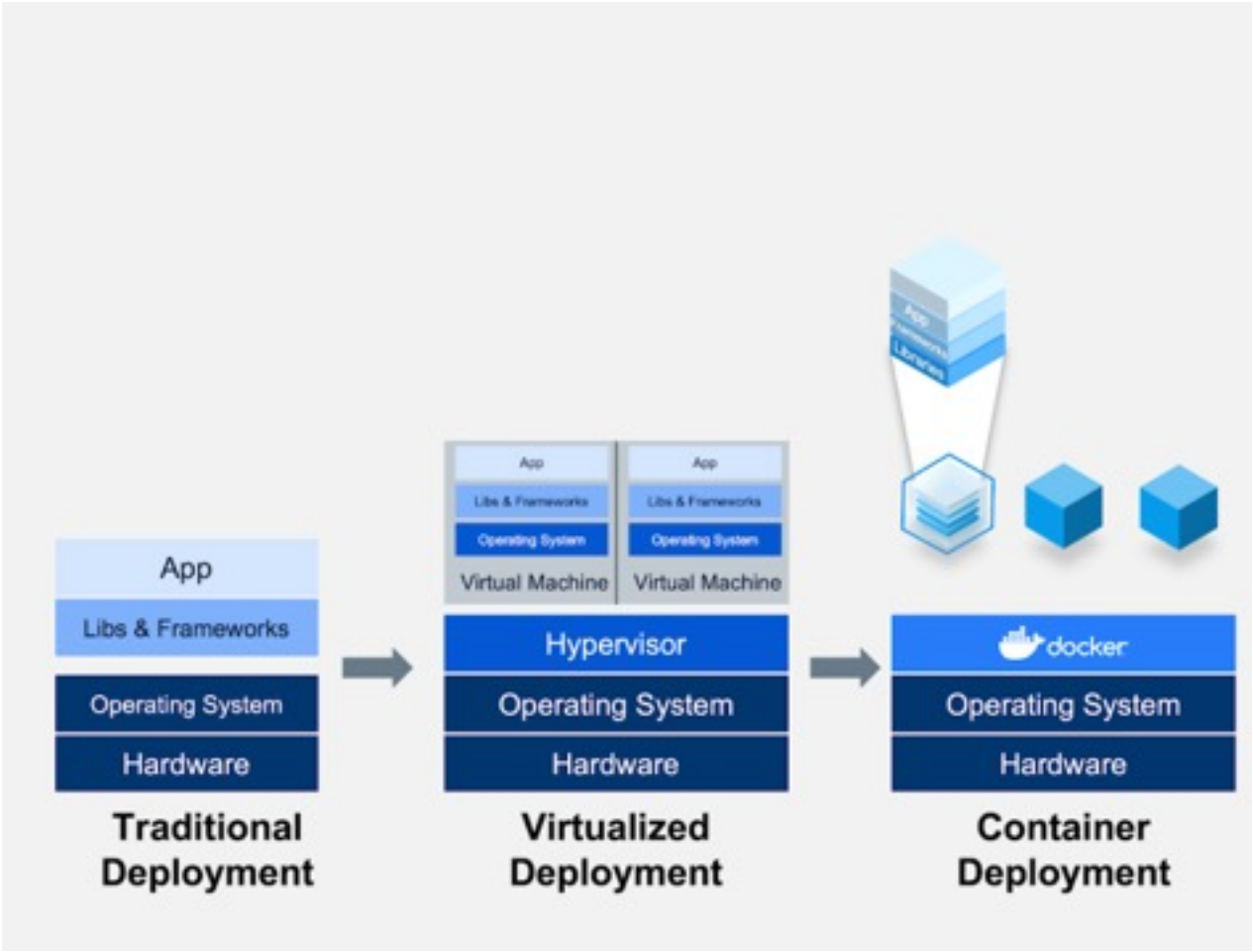
- Permitem empacotar facilmente o código, as configurações e as dependências de uma aplicação em elementos fundamentais que oferecem consistência ambiental, eficiência operacional, produtividade de desenvolvedores e controle de versões.

Conteineres

- Podem ajudar a garantir rapidez, confiabilidade e consistência de implantação, independentemente do ambiente de implantação.

- Além disso, eles oferecem um controle mais granular dos recursos, aumentando a eficiência da infraestrutura.

Conteinerização de Ambientes



Fonte: https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/

Docker

- O Docker é uma tecnologia Open Source que permite criar, executar, testar e implantar aplicações distribuídas dentro de containers de software.
- Ele permite que você empacote um software de uma padronizada para o desenvolvimento de software, contendo tudo que é necessário para a execução: código, runtime, ferramentas, bibliotecas, etc.
- O Docker permite que você implante aplicações rapidamente, de modo confiável e estável, em muitos ambientes virtuais e físicos.

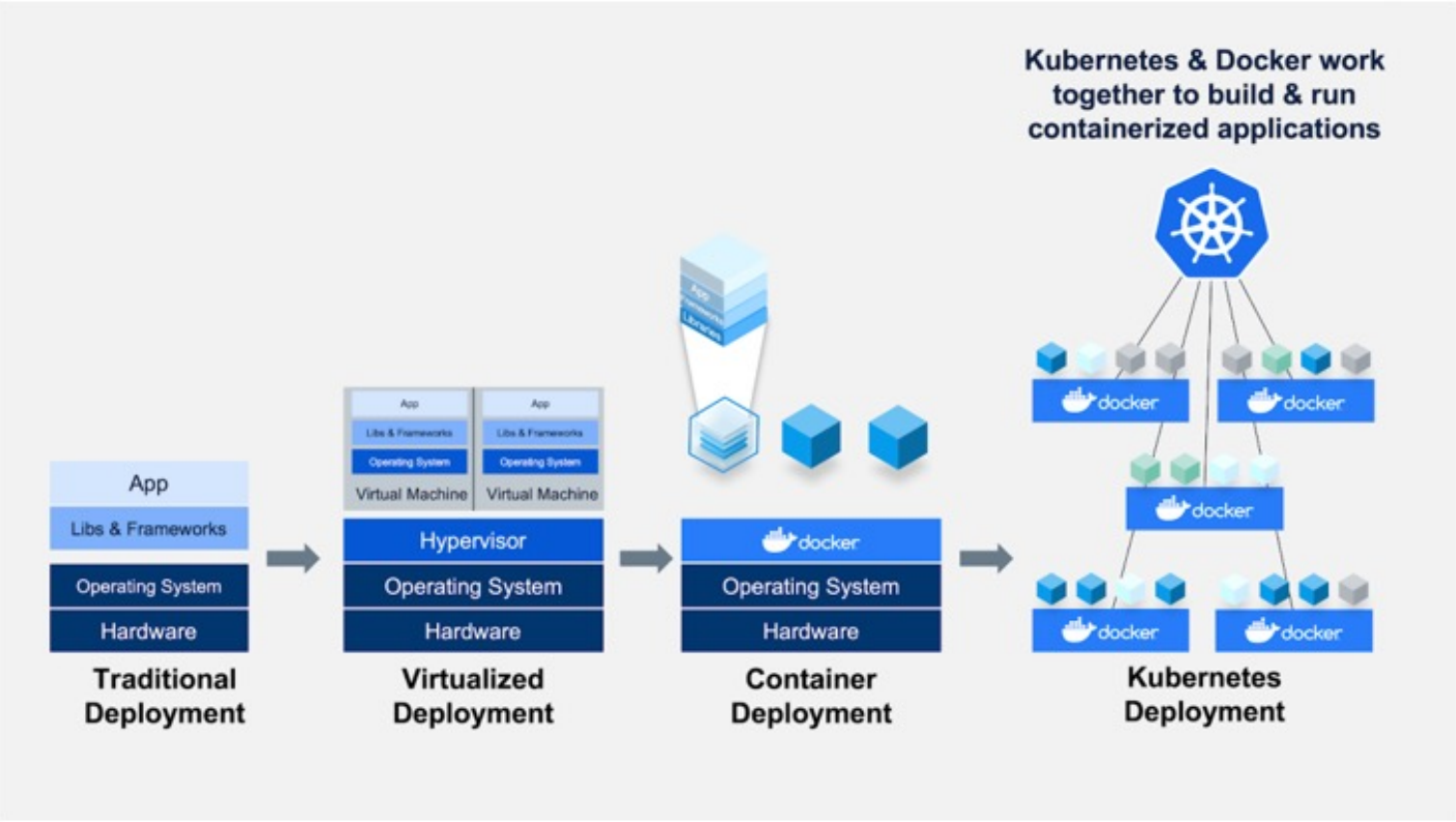
PUC Minas
Virtual

Orquestração de Contêineres

Docker

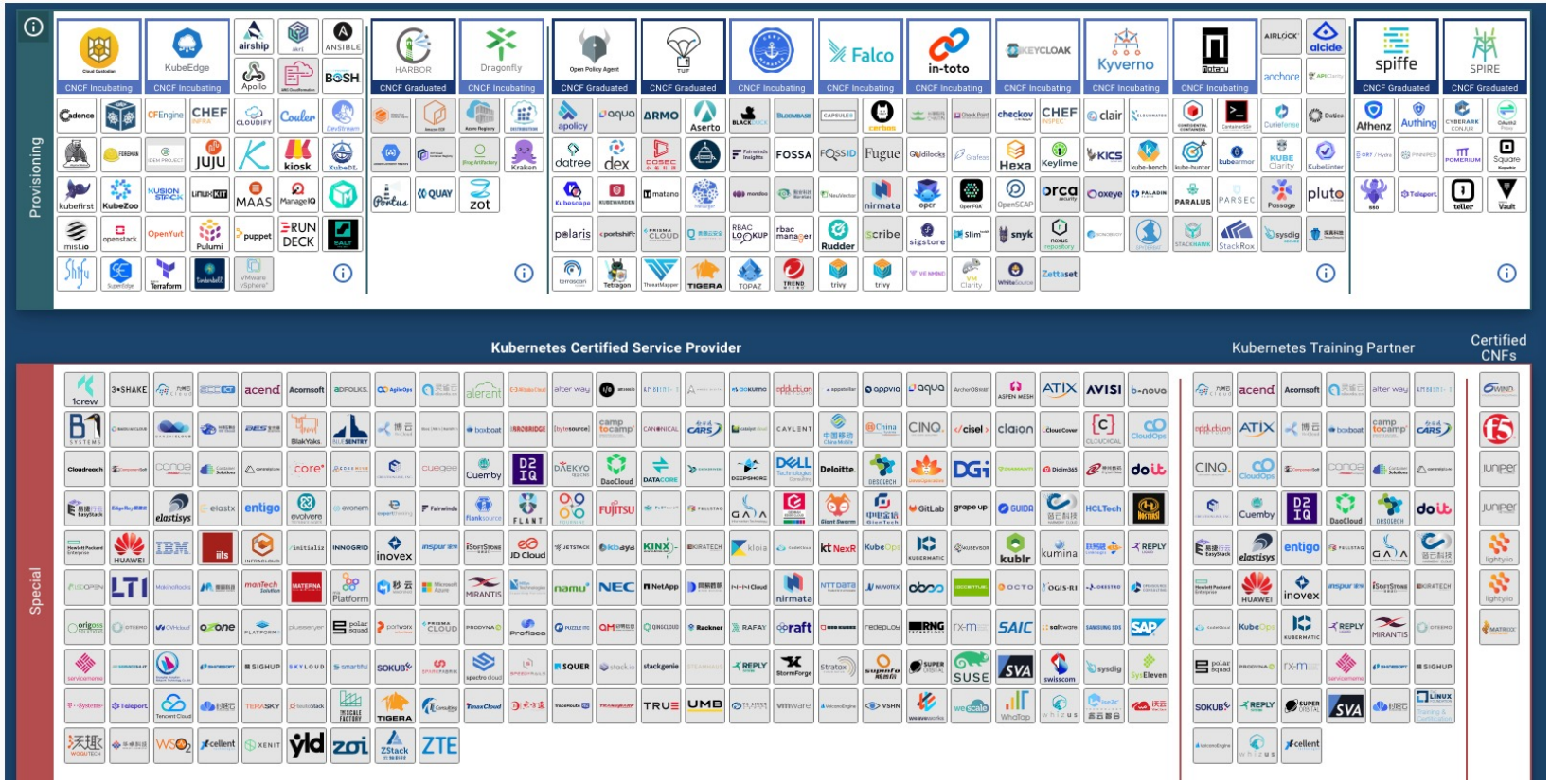
- A grande popularidade de contêineres começa a criar fazendas com muitos contêineres para administração pelos times de produção.
- Orquestradores de contêineres se tornam populares
  - Docker Swarm
  - Kubernetes
  - Mini-Kube/Open Shift

Orquestração de Ambientes



Fonte: https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/

CNCF do Kubernetes



PUC Minas
Virtual

Docker Swarm

PUC Minas
Virtual

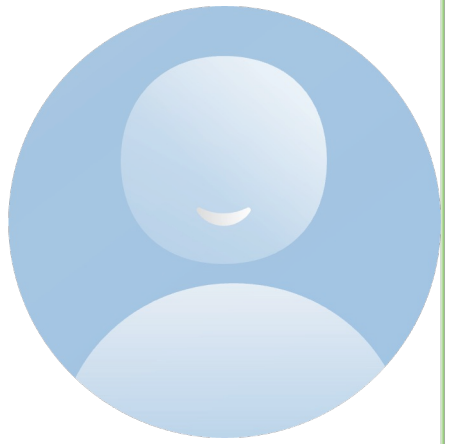
MiniKube

PUC Minas
Virtual

Infraestrutura como Código (IaC)

IaC – Infraestrutura Como Código

- Infraestrutura como Código (IaC) é uma abordagem na área de DevOps e gerenciamento de infraestrutura que trata a infraestrutura de computação como código de software



Transição da Infraestrutura Tradicional para Nuvens

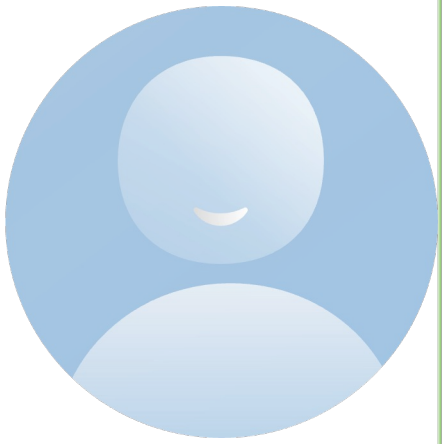
Era do Ferro	Era da Nuvem
Hardware Físico	Recursos Virtualizados
Provisionamento demora semanas	Provisionamento demora minutos
Processos manuais	Processos automatizados

Transição da Infraestrutura Tradicional para Nuvens

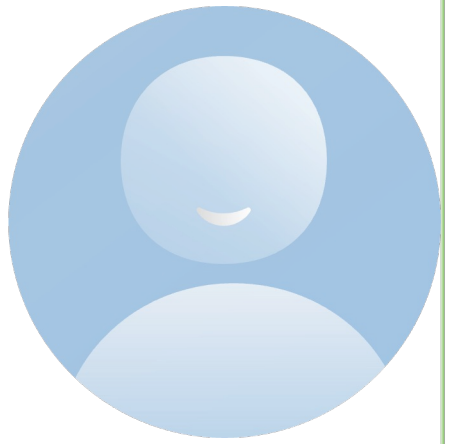
Era do Ferro	Era da Nuvem
Custo da mudança é alto	Custo da mudança é baixo
Mudanças representam falhas (tudo deve ser controlado e gerenciado)	Mudanças representam aprendizados e melhorias
Reduzir oportunidades de falhas	Maximizar o aprendizado
Entregas em grandes lotes, longos ciclos	Entregas em pequenas partes, teste continuo
Arquiteturas monolíticas	Arquiteturas de módulos, serviços, componentes, API e microsserviços

IaC – Infraestrutura Como Código

1. **Automação:** Automatizar a criação e gerenciamento da infraestrutura.
2. **Controle de Versão:** Tratar os arquivos de infraestrutura como código-fonte versionado.
3. **Declarativo:** Descrever o que se deseja na infraestrutura, não como alcançá-lo.
4. **Provisionamento e Configuração:** Incluir o provisionamento e a configuração dos recursos.
5. **Consistência e Conformidade:** Garantir consistência e conformidade entre ambientes.
6. **Testes e Validação:** Realizar testes automatizados para validar os arquivos de infraestrutura.



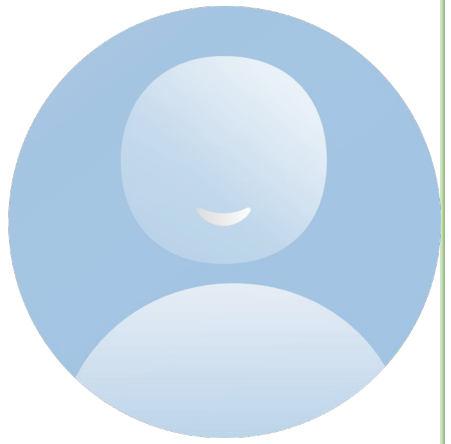
Ferramentas IaC



PUC Minas
Virtual

Engenharia do Caos

Os Macacos da Netflix

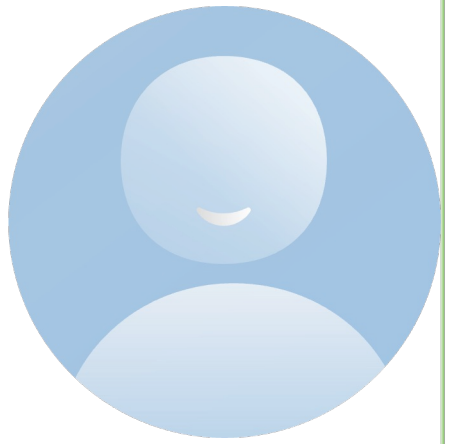


Fonte: https://github.com/Netflix/chaosmonkey

Engenharia do Caos

- *"Precisamos identificar fragilidades antes que elas se manifestem em comportamentos aberrantes em todo o sistema.*

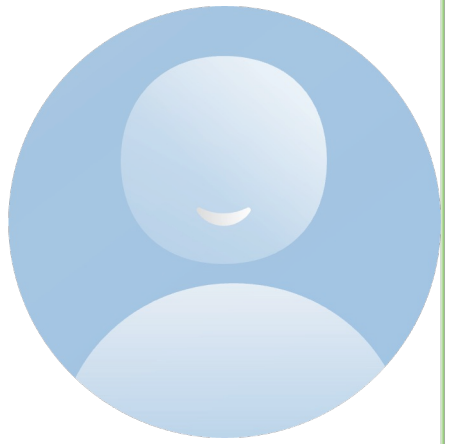
  Devemos lidar com as fraquezas mais significativas de forma proativa, antes que afetem nossos clientes na produção. Precisamos de uma maneira de gerenciar o caos inerente a esses sistemas, aproveitar o aumento da flexibilidade e velocidade e ter confiança em nossas implantações de produção, apesar da complexidade que eles representam."



- *http://principlesofchaos.org*

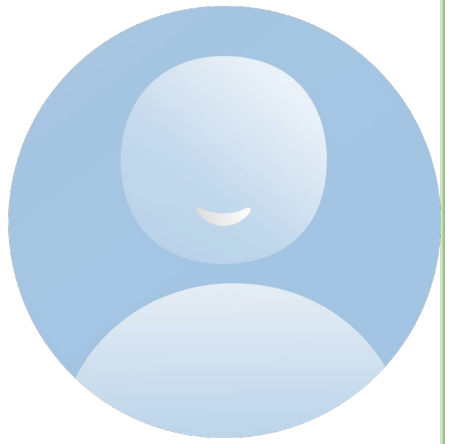
Exemplos de Estressores

- Injeção de latência na chamada de APIs
- Aumento da carga de um processador
- Interromper conexões de rede
- Terminar um contêiner.
- Remover um ambiente de produção inesperadamente.



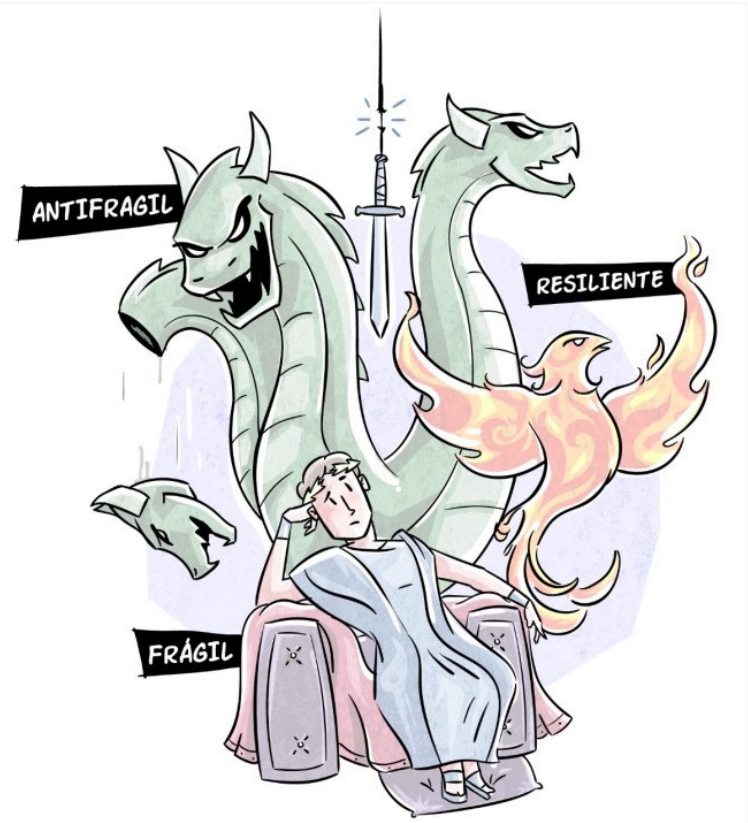
O processo de engenharia do caos

1. Comece definindo o "estado estável" como uma saída mensurável de um sistema que indica um comportamento normal.
2. A hipótese de que este estado estável continuará tanto no grupo controle quanto no grupo experimental.
3. Introduza variáveis que refletem eventos do mundo real, como servidores que falham, discos rígidos que funcionam mal, conexões de rede que são cortadas, etc.
4. Tente refutar a hipótese procurando uma diferença de estado estável entre o grupo controle e o grupo experimental.



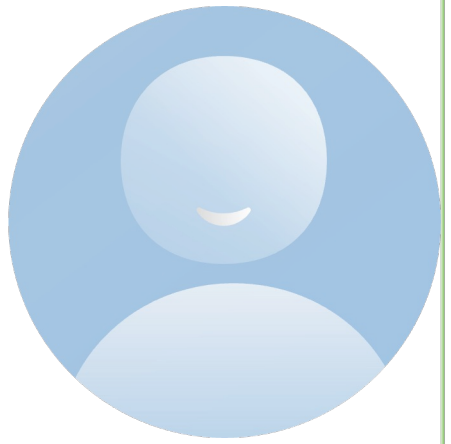
Fonte: https://github.com/Netflix/chaosmonkey

"Sistemas antifrágeis se tornam cada vez melhores e mais fortes sob ataques e erros contínuos", Nicholas Taleb

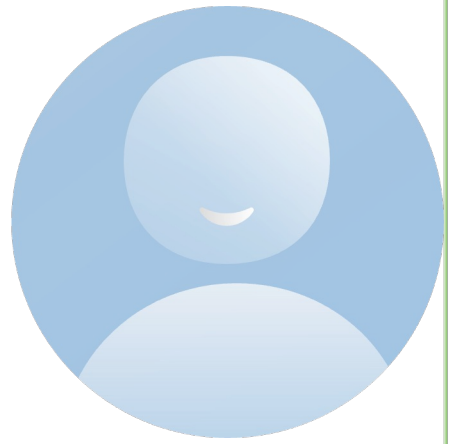


Fonte: Autor

- Engenharia do caos é uma prática avançada para *experimentar um sistema a fim de construir confiança na capacidade do sistema de suportar condições turbulentas na produção.*



Ferramentas Engenharia do Caos

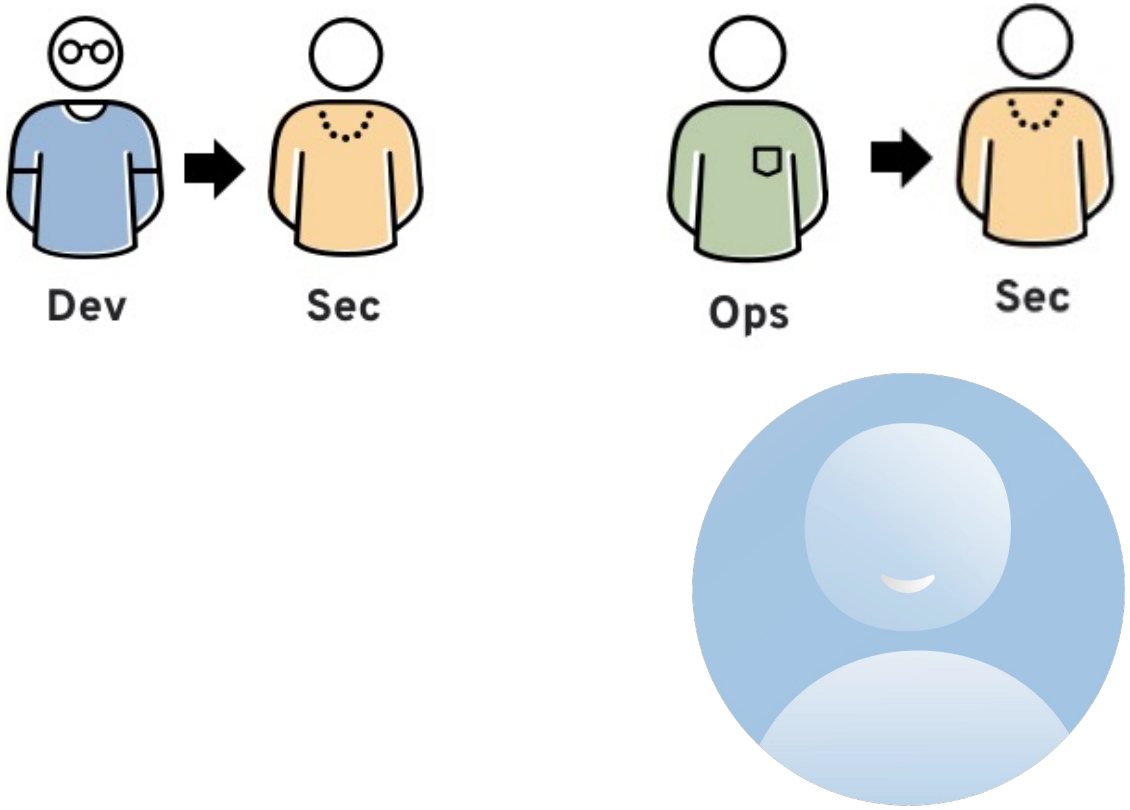


PUC Minas
Virtual

DevSecOps

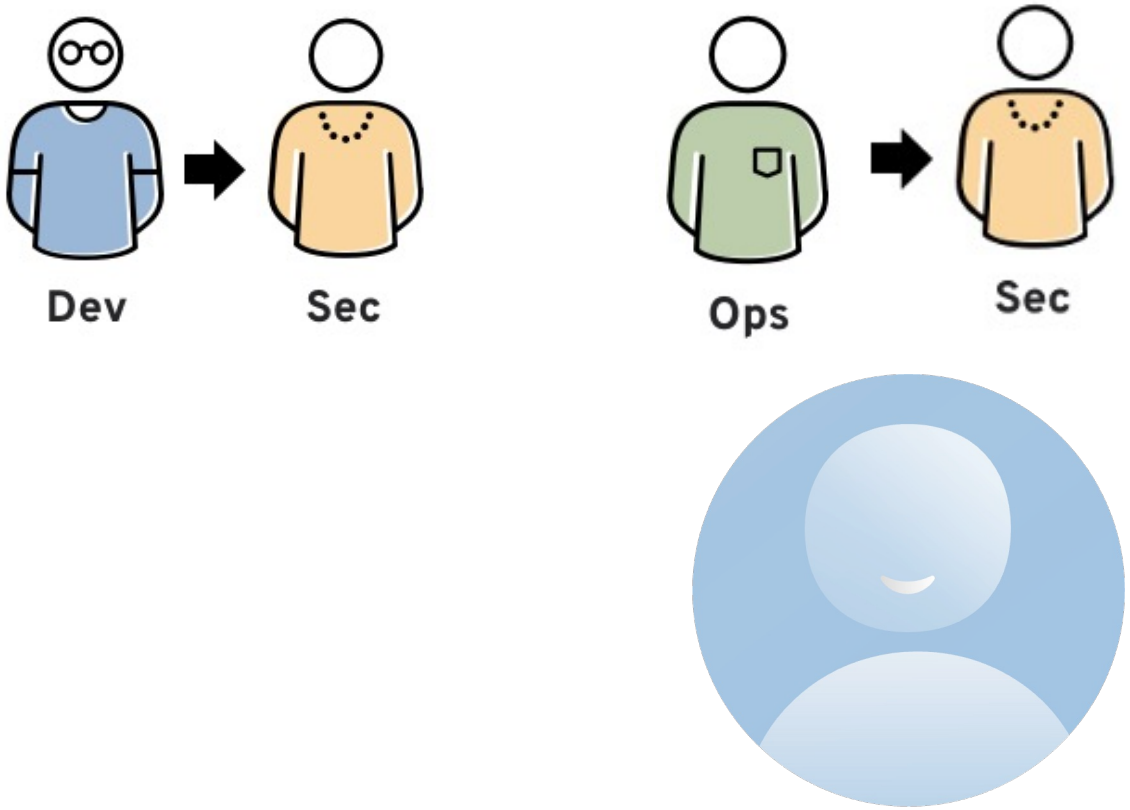
Por que?

- DevOps não envolve apenas as equipes de desenvolvimento e de operações.
- O envolvimento das áreas de segurança no final da construção de produtos é um desastre para a estabilidade da organização.
- É necessário que a equipe de segurança da TI desempenhe um papel integrado em todo o ciclo de vida das aplicações da sua empresa.



O que é o DevSecOps

- No DevSecOps, segurança é uma responsabilidade compartilhada e integrada do início ao fim.
- DevSecOps significa pensar na segurança da aplicação e da infraestrutura desde o início.
- Isso implica também automatizar algumas barreiras de segurança para evitar que o fluxo de trabalho de fique lento.



Exemplos

- Verificação automatizada da segurança do código após commits.
- Verificação de segurança automatizada de contêineres Dockers.
- Verificação de segurança automatizada de ambientes.

Exemplos

- Testes de segurança
- Dias de jogos
- Times vermelhos e times azuis

Exemplos

- Gerenciamento integrado de senhas e segredos
- Engenharia do caos que removem instâncias inseguras de produção (ex. *Netflix Security Monkey*)
- Automação de gerenciamento das configurações de serviços e sistemas

Exemplos

- Coleta de dados de log e eventos
- Monitoramento de integridade de chaves de arquivo e registro
- Inventário de processos em execução e aplicativos instalados
- Monitoramento de portas abertas e configuração de rede
- Detecção de kits de acesso de administrador (*rootkit*) ou artefatos de malware

Exemplos

- Segurança automatizada de ambientes de nuvens
- Conformidade a regulações como GDPR/LGPD ou ISO 27001
- Avaliação de configuração e monitoramento de políticas
- Execução de respostas ativas

Ferramentas Comuns

- SonarQube
- Veracode
- Wazuh
- GitHub Actions
- Trivy
- Starboard
- OWASP Zed Attack Proxy
- Hashicorp Vault
- Aqua Security

PUC Minas
Virtual