



igti

—

Devops

Capítulo 3. Estrutura

Prof. Thiago Chierici



Aula 3.1. Virtualização x Containerização



- ☐ VMs.
- ☐ Containeres.
- ☐ Processos x Ferramentas.

- “This containers revolution is changing the basic act of software consumption. It’s redefining this much more lightweight, portable unit, or atom, that is much easier to manage... It’s a gateway to dynamic management and dynamic systems.”

Craig McLuckie, Google. Collaboration Summit 2015



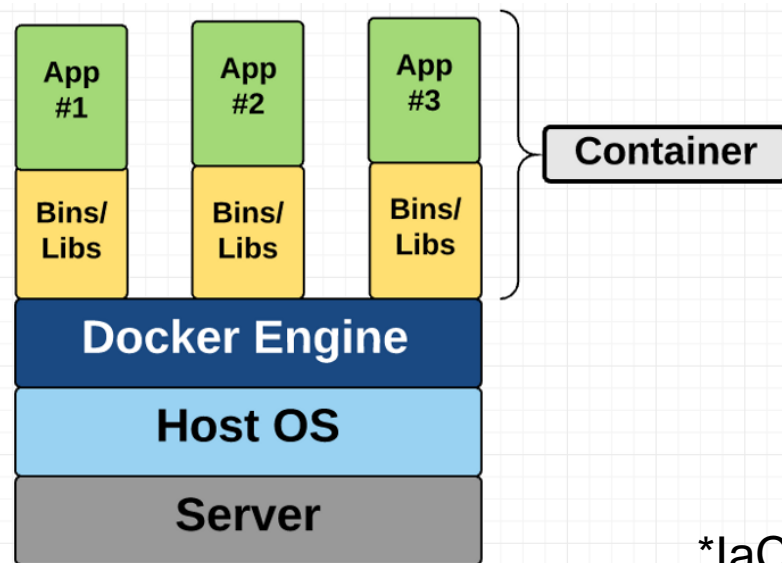
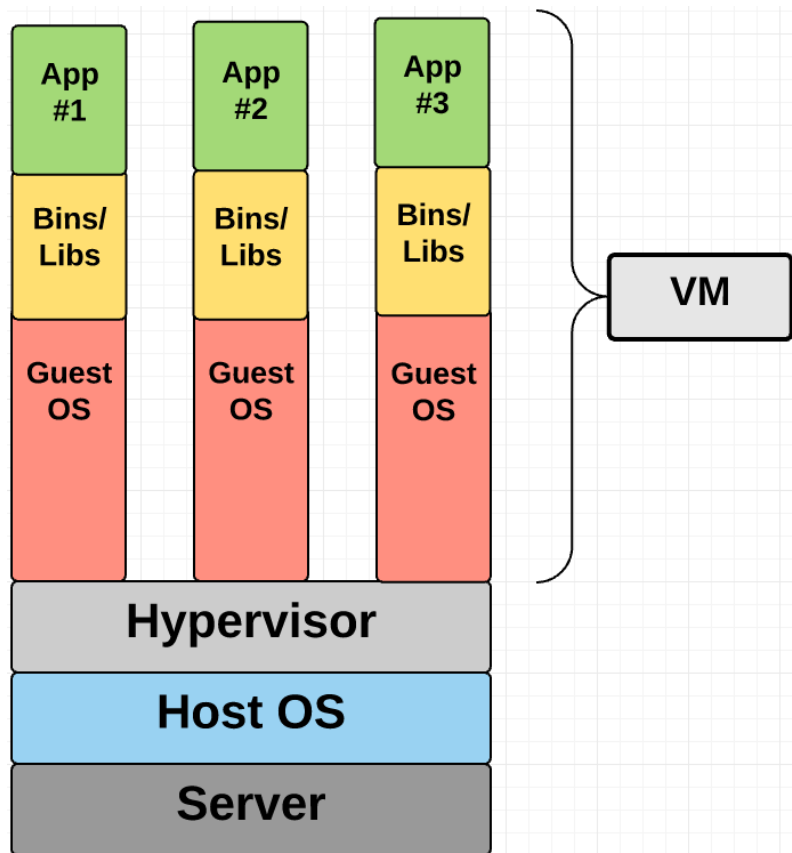
O que há em comum



- VMs e containeres buscam abstrair algumas camadas (especialmente de hardware) para facilitar a portabilidade.
- Também buscam criar contextos isolados e independentes.
- Uso de imagens para criação das instâncias.



Arquitetura VM x Container



*IaC

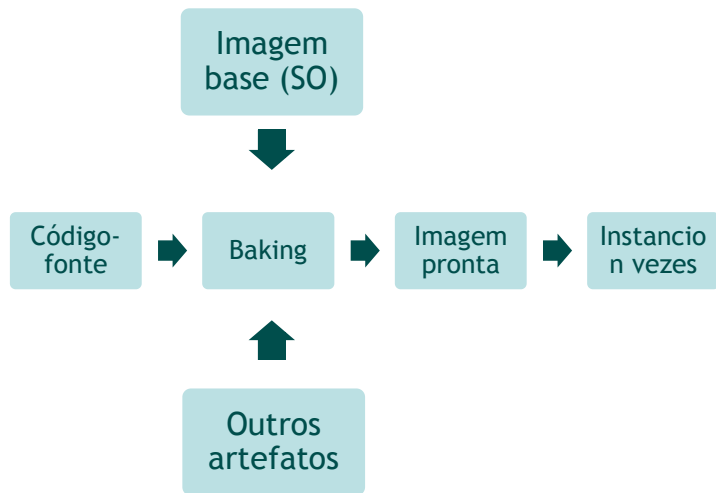
- Open source Linux container:
 - Uso como isolamento de contexto.
 - Mais eficiente que criar VMs (normais).
 - Nasceu na geração IaC.
 - Largamente adotado por toda cadeia do mercado.



docker



- Como escalar?
- To bake:
 - Integrar o projeto compilado, com a imagem base do SO e outros artefatos externos.
 - Geração de uma imagem (para container ou VM).
 - A partir dela podem ser criados quantos ambientes/máquinas for necessário.



- ☑ VMs e containers ajudam a utilizar recursos com maior eficiência.
- ☑ Containers são ainda mais eficientes pois conseguem garantir o isolamento no nível de aplicação, reaproveitando mecanismos do sistema operacional hospedeiro.



Próxima aula



- ❑ Infraestrutura como código.





Aula 3.2. Infraestrutura como código



Nesta aula



- ❑ Infraestrutura como código.
- ❑ Infra mutável x imutável.



- Tarefas de administração de sistemas:
 - Não automatizadas.
 - Scripts pontuais de baixo reuso.
 - Sem versionamento.
 - Sem idempotência.



Código x Configuração



```
1  ---
2  - name: Configure Raddit App Instance
3    hosts: all
4    become: true
5    tasks:
6      - name: Install Ruby
7        apt: "name={{ item }} state=present"
8        with_items:
9          - ruby-full
10         - build-essential
```

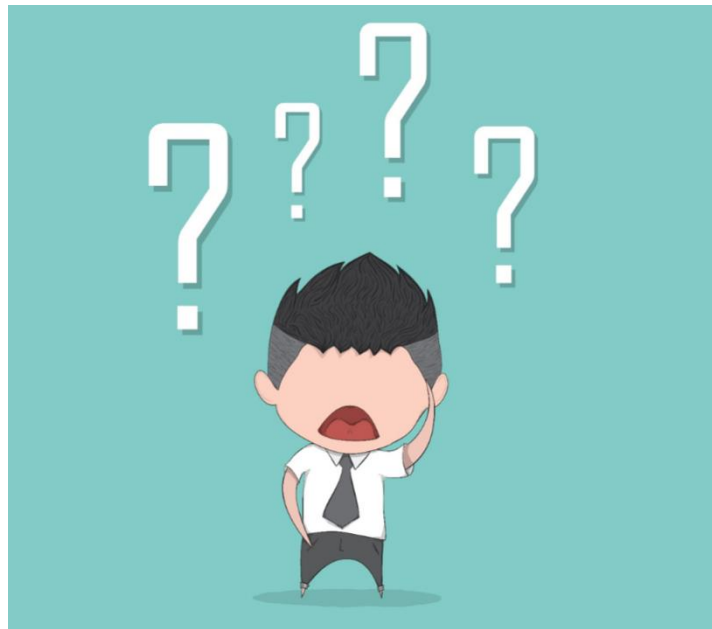
playbook1.yml hosted with ❤ by GitHub

[view raw](#)

- Independe do estado atual do ambiente.
- Melhores práticas de desenvolvimento aplicadas na infra.
 - Versionamento.
 - Paridade com código.
 - Desenvolvedor evolui infra enquanto desenvolve.
 - Teste não precisa de uma série de pré-requisitos.



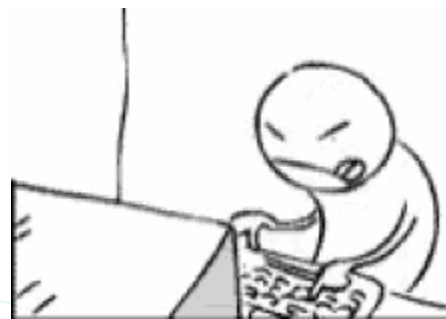
- Se sou um desenvolvedor, preciso preocupar com infra as code?



Como pratico?



- Pra quem quer colocar a mão na massa:
 - <https://github.com/Artemmkin/infrastructure-as-code-tutorial>
 - Além de IaC, um pouco sobre ferramentas que veremos na próxima aula.



- Trabalho manual nem sempre dá certo.
- Uma estrutura de servidores que permita alterações manuais é considerada mutável.
- Quando a única forma de alterar um servidor é através da recriação com novas configurações, a estrutura é considerada imutável.



- ☑ IaC gera ganhos de qualidade, produtividade, previsibilidade, entre outros.
- ☑ Uma estrutura imutável pode trazer ganhos adicionais.



- ❑ Automação da gestão da configuração.
- ❑ Ferramentas disponíveis e como aplicá-las.





Aula 3.3. Automação de gestão de configuração e infra

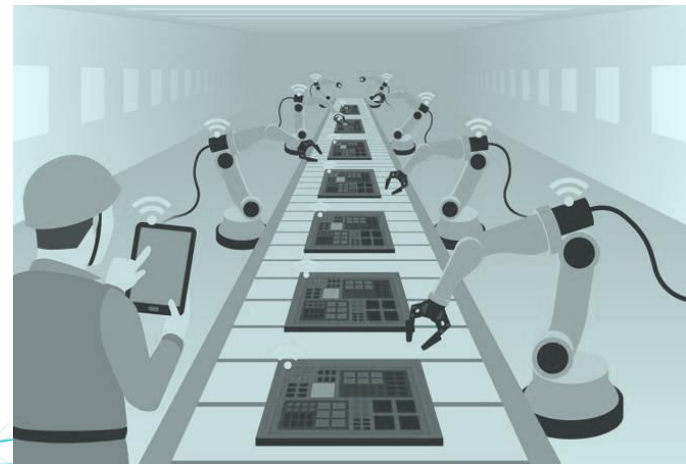


- ❑ Como utilizar a automação da gestão da configuração para tornar seu processo mais eficiente.

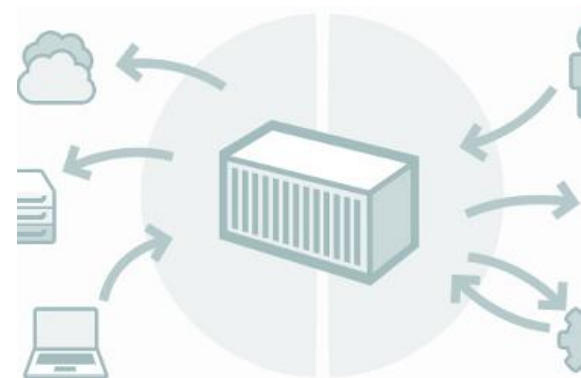


Por que automatizar?

- Tempo gasto na gestão de infra.
- Falhas manuais.
- Difícil garantir consistência entre servidores e ambientes.

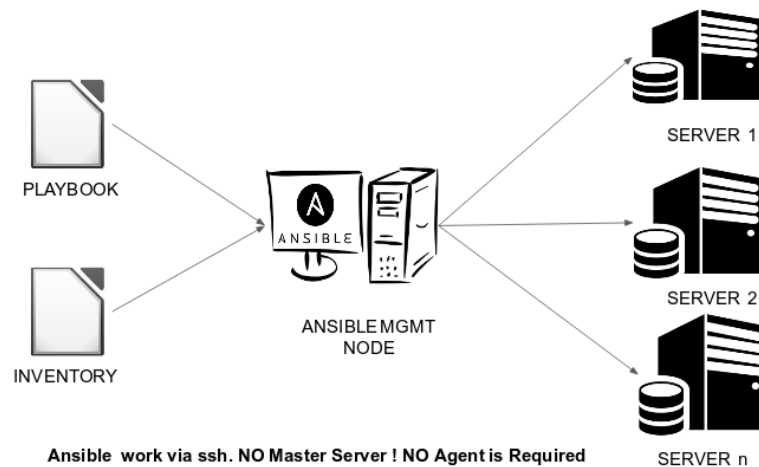
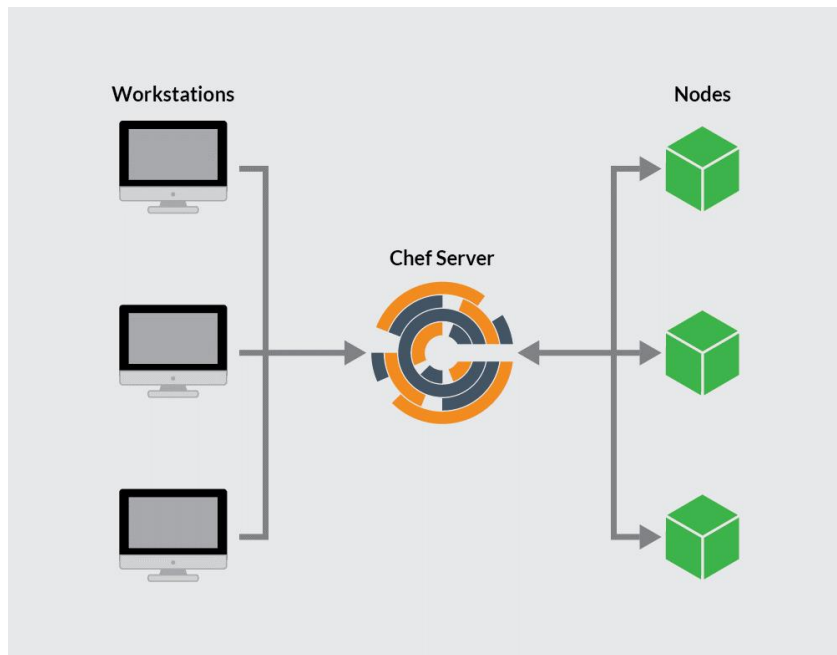


- Temos dois grupos de ferramentas relacionadas a IaC.
- Orquestração de configuração:
 - Foco em automatizar implantação de servidores / infra.
 - Ex.: Terraform, AWS CloudFormation, Azure Resource Manager and Google Cloud Deployment Manager.
- Gestão de configuração:
 - Foco em configurar uma infra já implantada.
 - Ex.: Chef, Puppet, Ansible.



<https://medium.com/formcept/configuration-management-and-continuous-deployment-cd0892dce998>

<https://medium.com/head-in-the-clouds/15-infrastructure-as-code-tools-you-can-use-to-automate-your-deployments-2cafc234b8ac>



- ☑ Automatizar a infraestrutura nos ajuda a implementar diversos processos importantes para o Devops.



- ❑ Um novo case com a Netflix, uma das empresas mais influentes em Devops.





Aula 3.4. Inspiração – Netflix

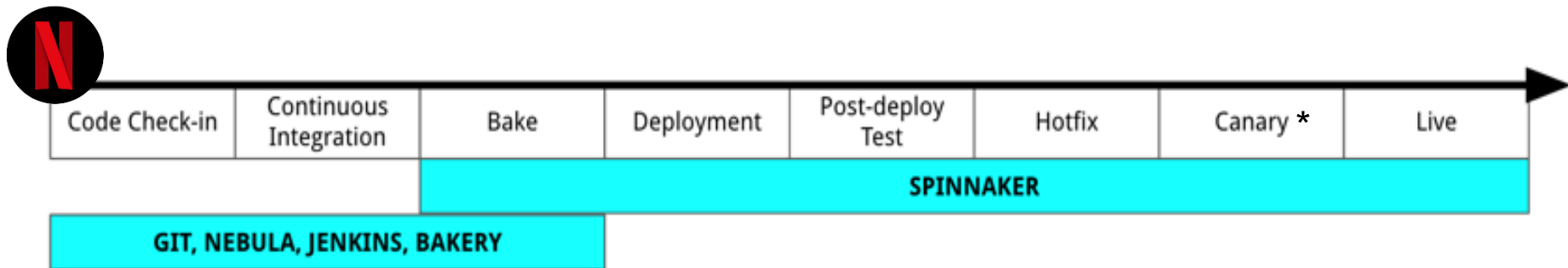


- ❑ Primeiro momento de inspiração baseado na Netflix, uma das maiores referências na área de Devops.
- ❑ Veremos como a Netflix compila e publica seu código, e como a infra viabiliza um modelo de alta agilidade e robustez.





- Código é compilado e testado localmente com o **Nebula**:
 - Suite open-source baseada em Gradle para testes, gestão de dependências, versões, empacotamento, entre outros.

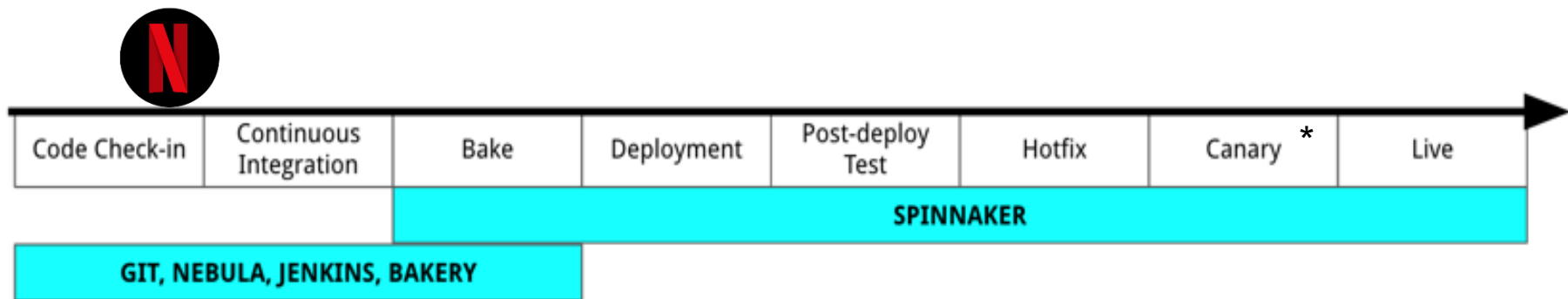


Fonte: <https://medium.com/netflix-techblog/how-we-build-code-at-netflix-c5d9bd727f15>

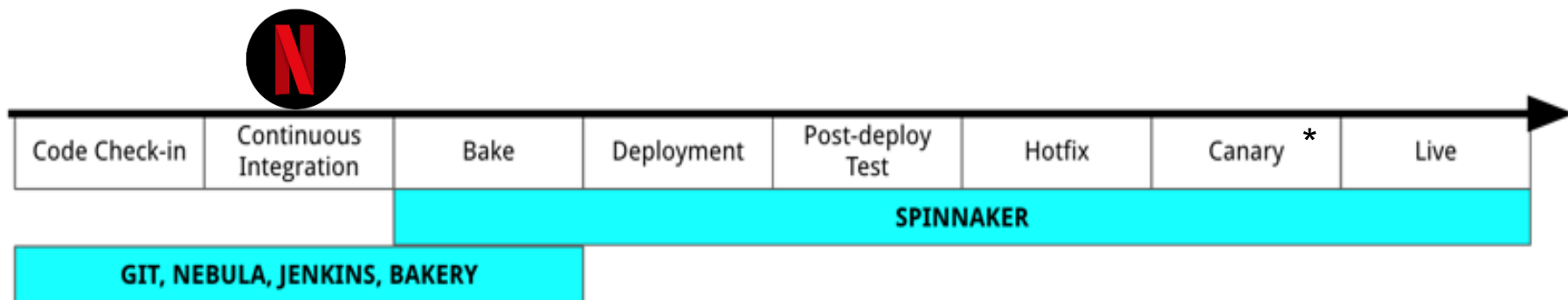
Do código aos 75 milhões de usuários



- Código é integrado ao **GIT**:
 - Sempre após execução local de testes.



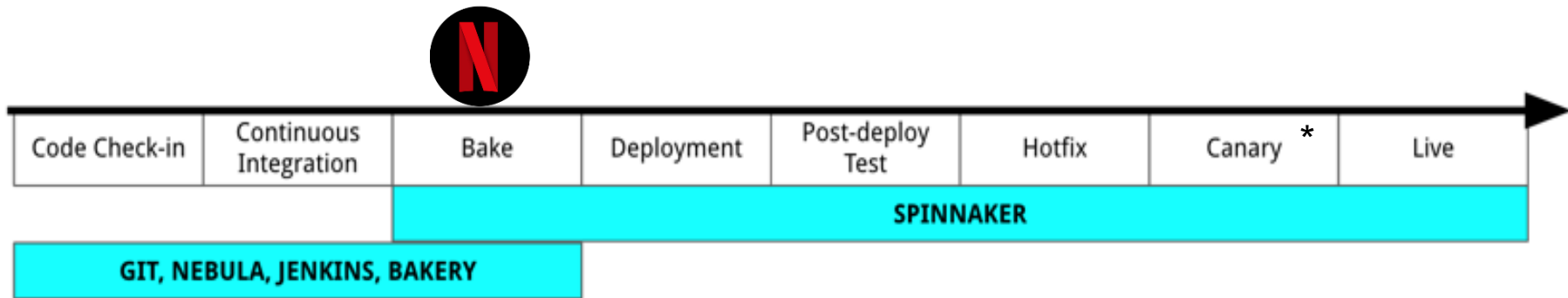
- **Jenkins** executa um script **Nebula**, que compila, testa e integra o pacote para implantação:
 - Dezenas de servidores Jenkins-master cuidam de diferentes tarefas, incluindo CI.



Do código aos 75 milhões de usuários

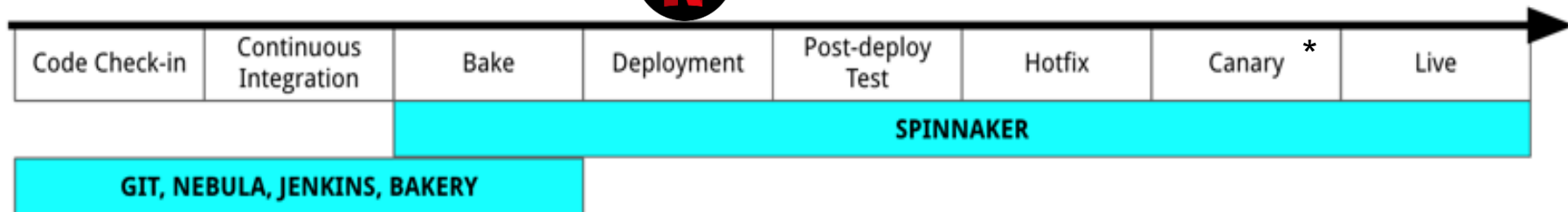


- Pacote é integrado (Baked) a uma imagem Amazon, com o **Aminator**:
 - Servidores são imutáveis.
 - Ferramenta open-source da Netflix, que usa imagens base e pacotes para gerar uma imagem imutável pronta pra qualquer ambiente.

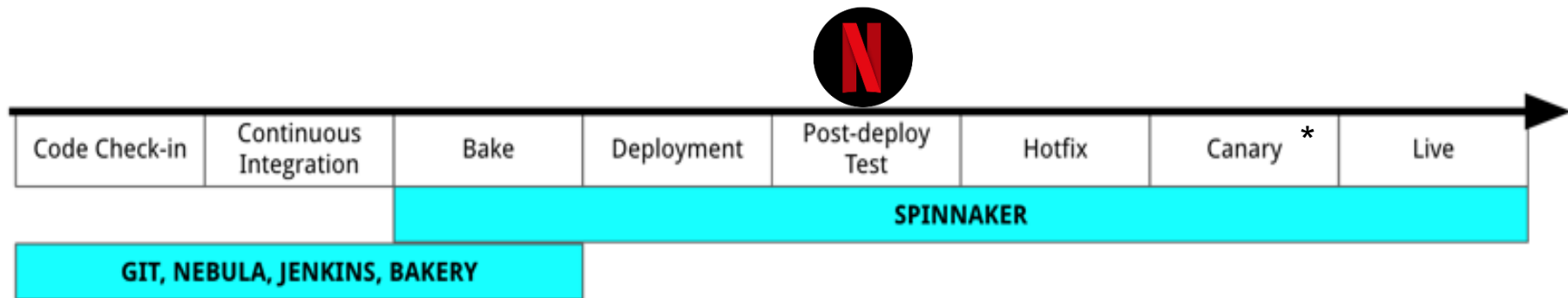


Mais detalhes: <https://medium.com/netflix-techblog/ami-creation-with-aminator-98d627ca37b0>

- O **Spinnaker** disponibiliza a imagem para servir de base para milhares de instâncias:
 - Já em formato binário e de rápida inicialização nos clusters AWS.



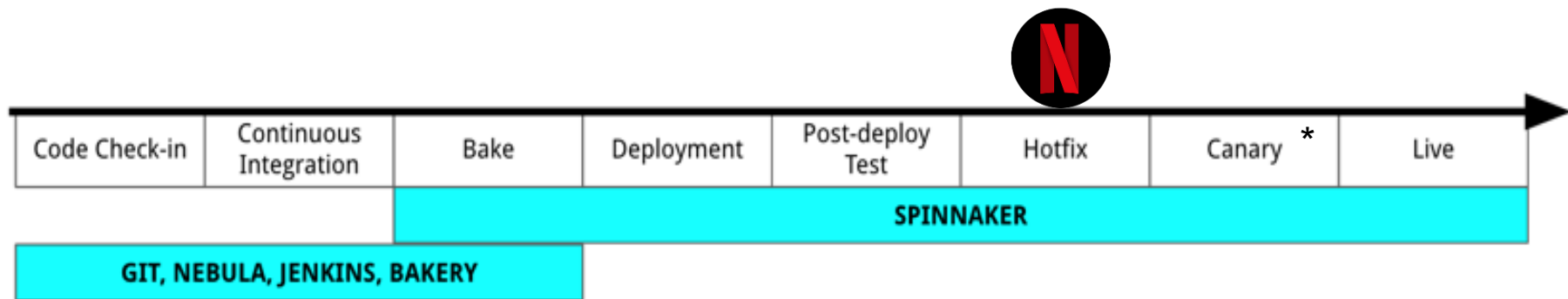
- Bateria de testes automatizados de diversos tipos são executados:
 - Maior foco em testes integrados.



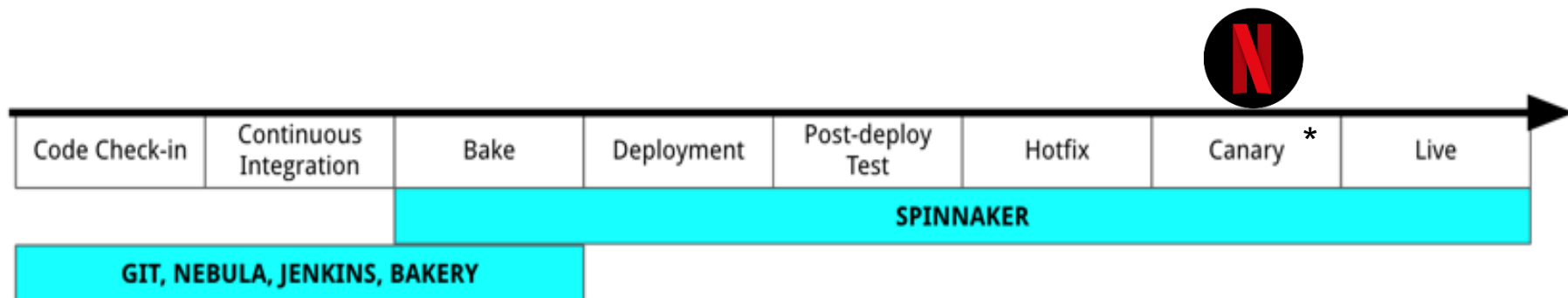
Do código aos 75 milhões de usuários



- Ajustes podem ser necessários.
 - Impactos não serão percebidos pelos usuários finais.



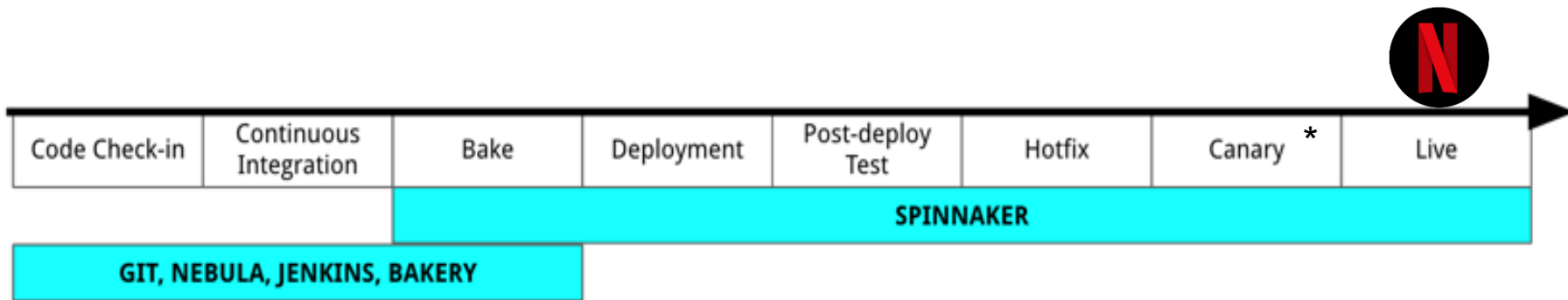
- Canary, entre outras técnicas de deployment, pode ser aplicada pela ferramenta.
 - Em um próximo capítulo, apresentaremos os detalhes de várias técnicas de deploy, incluindo as utilizadas aqui.



Do código aos 75 milhões de usuários



- Em alguns casos, entre o commit/check-in e o live, são gastos menos de 16 minutos:
 - Métricas são utilizadas para tentar otimizar passos que estejam menos eficientes.



- ☑ Netflix é uma das maiores referências de mercado para Devops e em especial para IaC.
- ☑ Depois que começou a assistir esse vídeo, alguém pode ter feito um commit em algum código do Netflix, que já foi integrado, testado e distribuído em servidores de todo o mundo.
- ☑ Infra como código trouxe grandes ganhos na estratégia do Netflix:
 - Velocidade do commit ao go-live.
 - Testabilidade.
 - SCM da infra.



- ❑ Próximo capítulo o foco é a qualidade.
- ❑ O que muda nos processos e ferramentas da qualidade quando trabalhamos com Devops?

