

# DEVOPS

1  
15

Prof /Profª Mauricio Antonio Ferste

**"Sem boas práticas de design, uma implementação DevOps não tem esperança de cumprir sua promessa de acelerar a inovação com alta qualidade e escala." Mark Hornbeek**

# Tempos Modernos



shutterstock

IMAGE ID: 328557545  
www.shutterstock.com

## CI/CD

- **CI/CD, que significa Integração Contínua e Entrega Contínua (ou Implantação Contínua), são práticas fundamentais no desenvolvimento de software que visam automatizar e agilizar o processo de entrega de software.**



# Como Implantar DevOps

## 6 Passos para o DevOps

DevOps não é um problema de ferramentas, e sim de encontrar maneiras de quebrar as barreiras culturais em prol de um novo modelo de trabalho.



1

Escolha um projeto de desenvolvimento de software no qual será possível aplicar alguma metodologia ágil. Procure buscar projetos com baixo risco para a organização, geralmente projetos para desenvolvimento de sistemas Web ou Mobile são os mais indicados.

2

Procure envolver pessoas com pré-disposição para aceitar as mudanças relacionadas a uma nova forma de trabalho e, que entendam o valor da iniciativa com base em práticas mais colaborativas e com foco na comunicação eficiente e confiança entre elas.

3

Será fundamental que todos compreendam o significado e o propósito do DevOps, desta maneira você fará com que todos estejam na mesma página em relação a definição e o propósito deste novo método de trabalho.

4

Entender a governança e os processos da TI ajudará a antecipar possíveis problemas nos quais poderão limitar suas iniciativas e, principalmente obter benefícios como a agilidade e a flexibilidade para o desenvolvimento e implantação do software em produção.

5

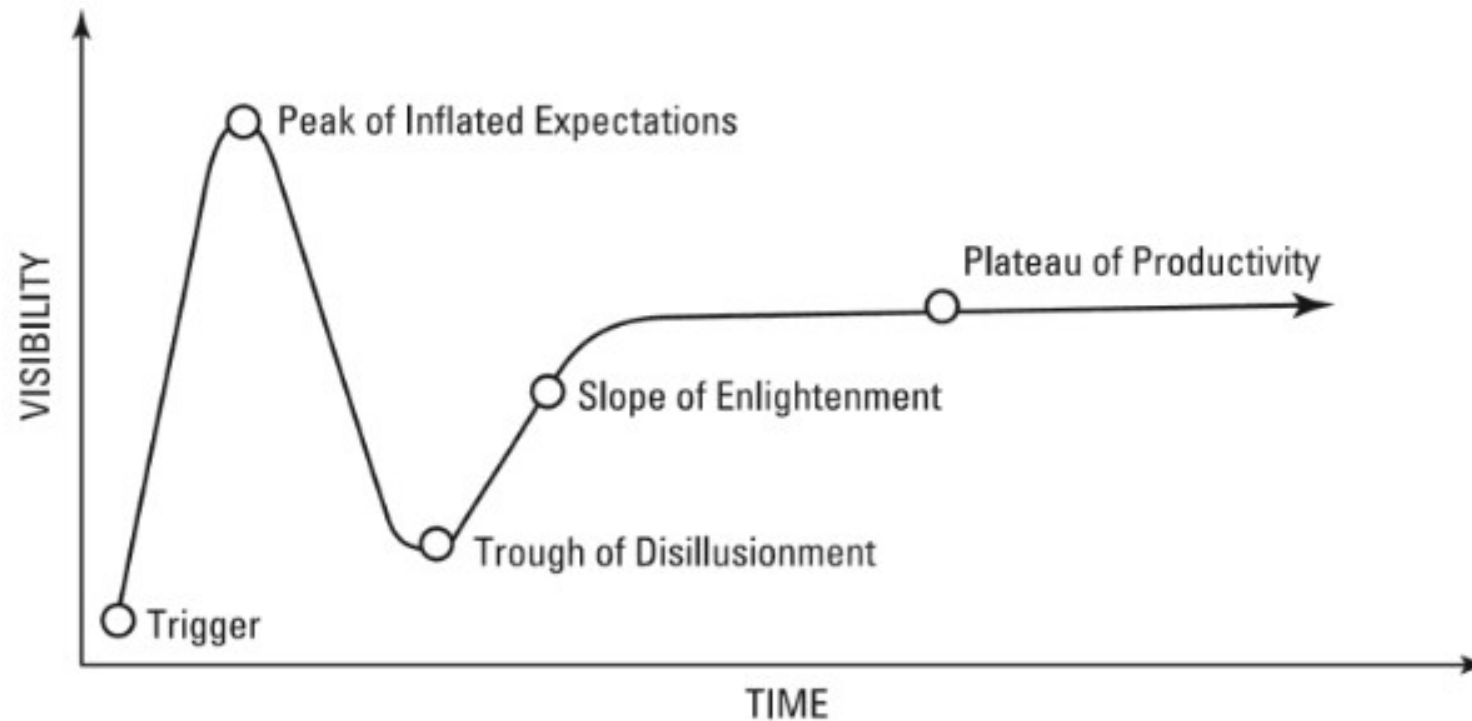
As práticas do DevOps pressupõem uma melhor colaboração e comunicação entre as pessoas, assim como remetem a necessidade de mudar o foco da TI para o produto final. Não crie metas individuais e sim metas e objetivos que possam ser atribuídos a todos da equipe.

6

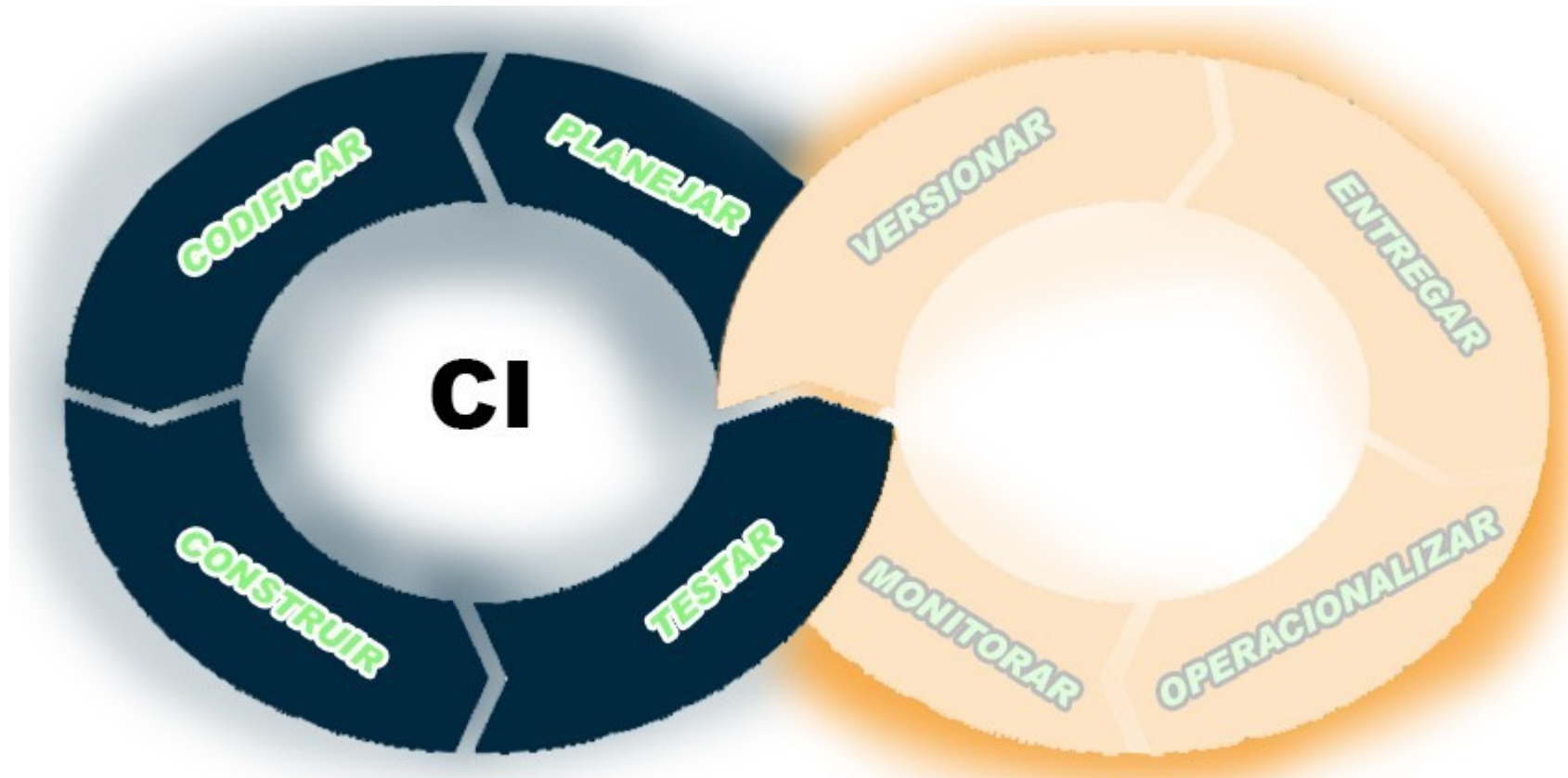
Sua agilidade e escalabilidade na nuvem dependerão diretamente da orquestração e automatização dos processos de testes e deploy. Procure evitar ao máximo a necessidade de interação humana.



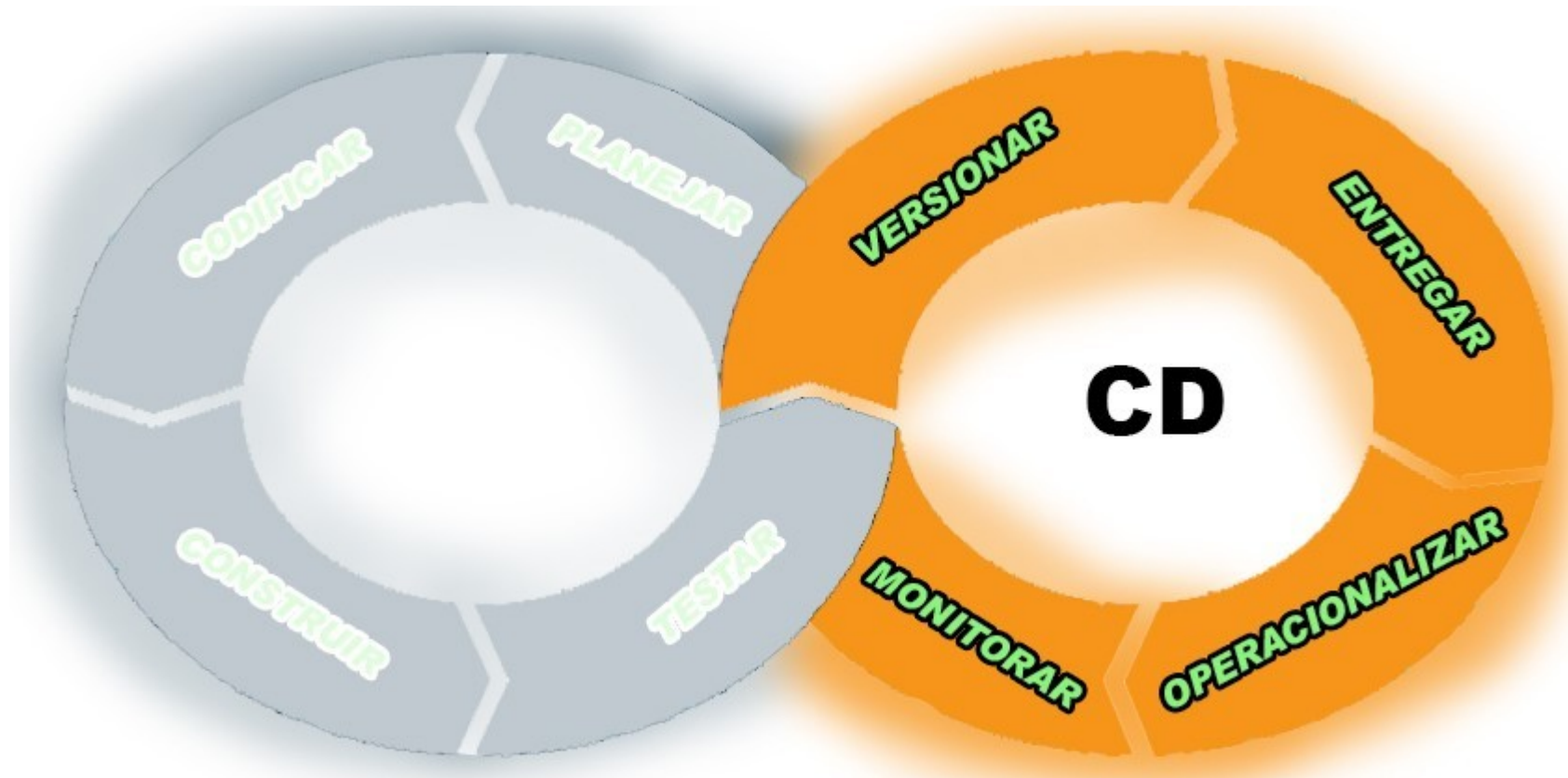
# Processo de Convencimento para uso de DevOps



# CI



# CD



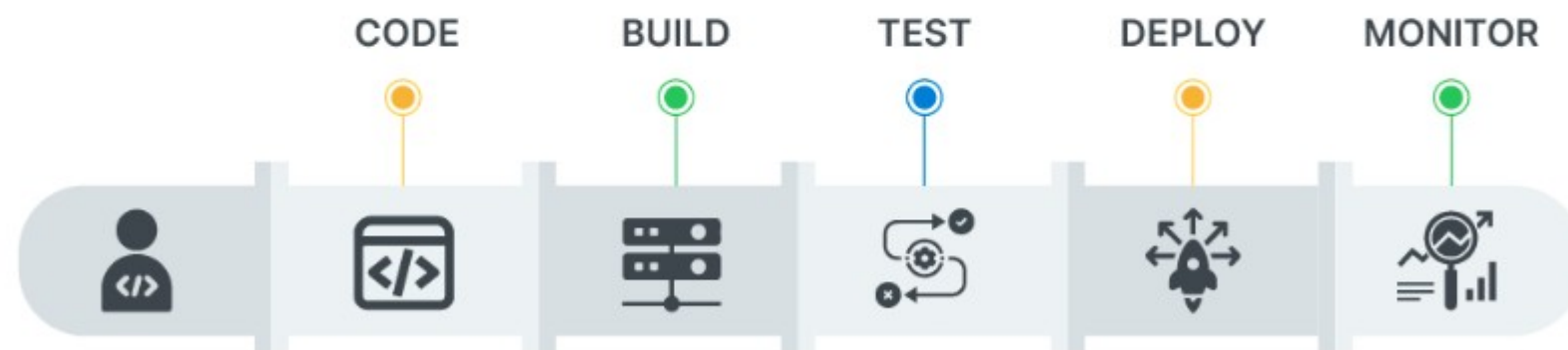


# TEMA 3 – FERRAMENTAS E FRAMEWORKS PARA CI/CD

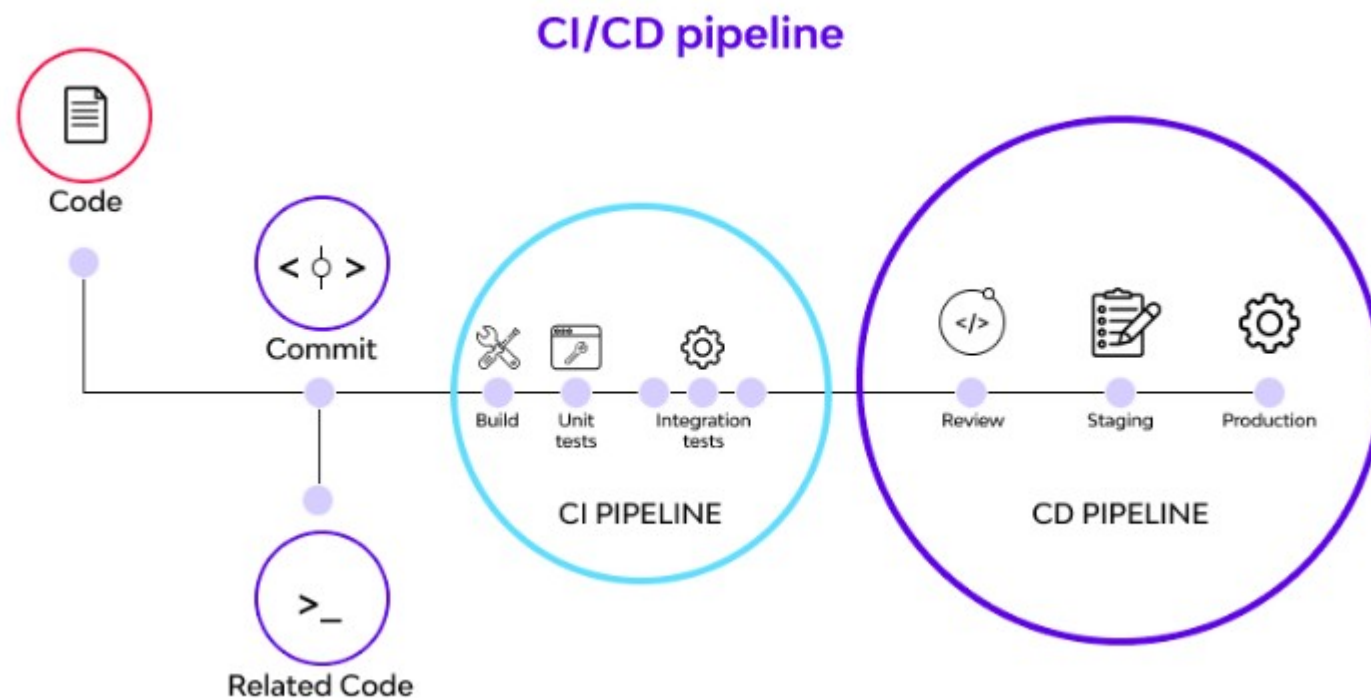
9  
15

# Pipeline CI/CD

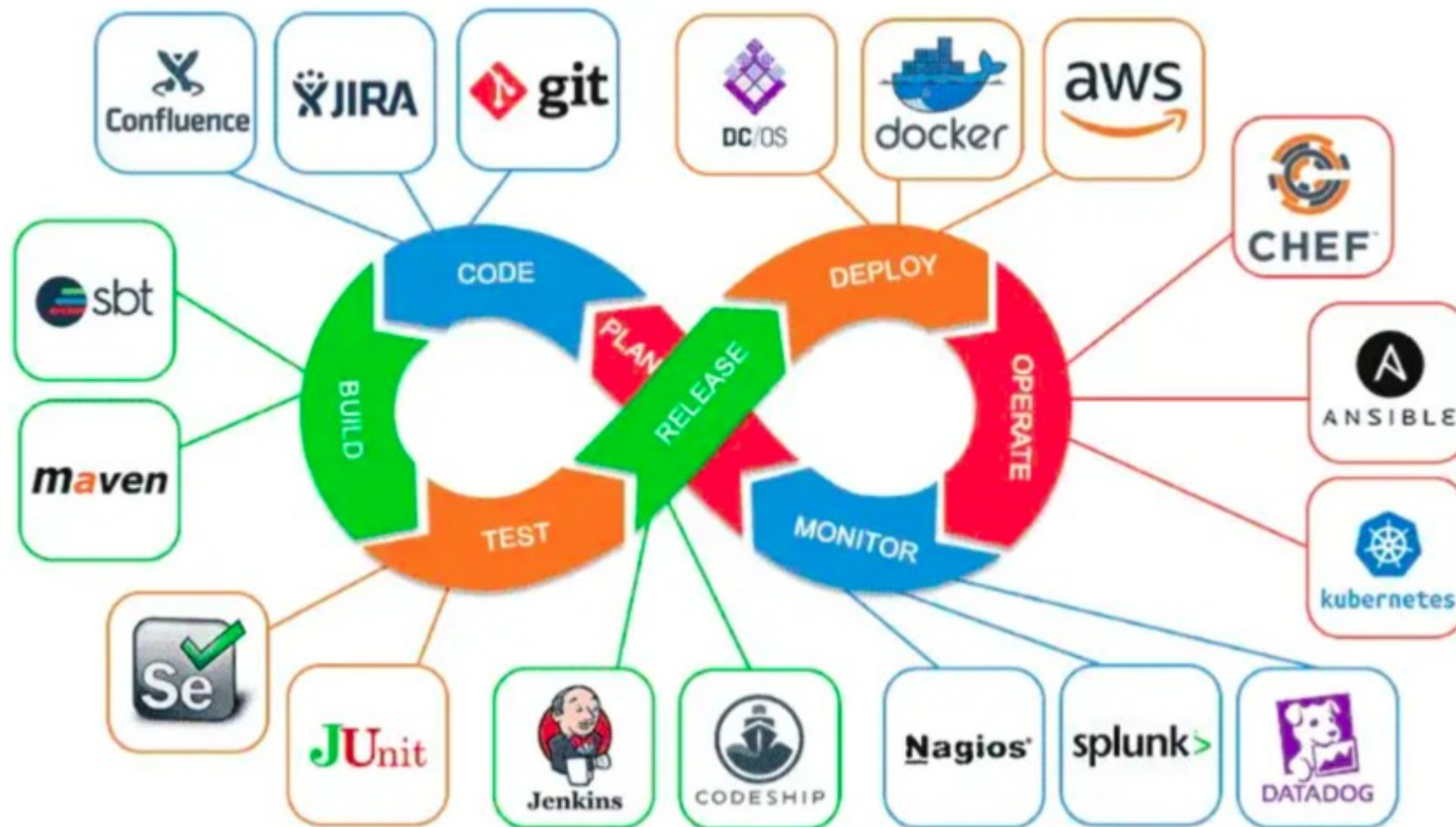
CI CD PIPELINE



# Pipeline CI/CD



# Ferramentas CI/CD



## Fluxo já existente no GitHub

```
git add file1.md file2.md
```

# criando um registro

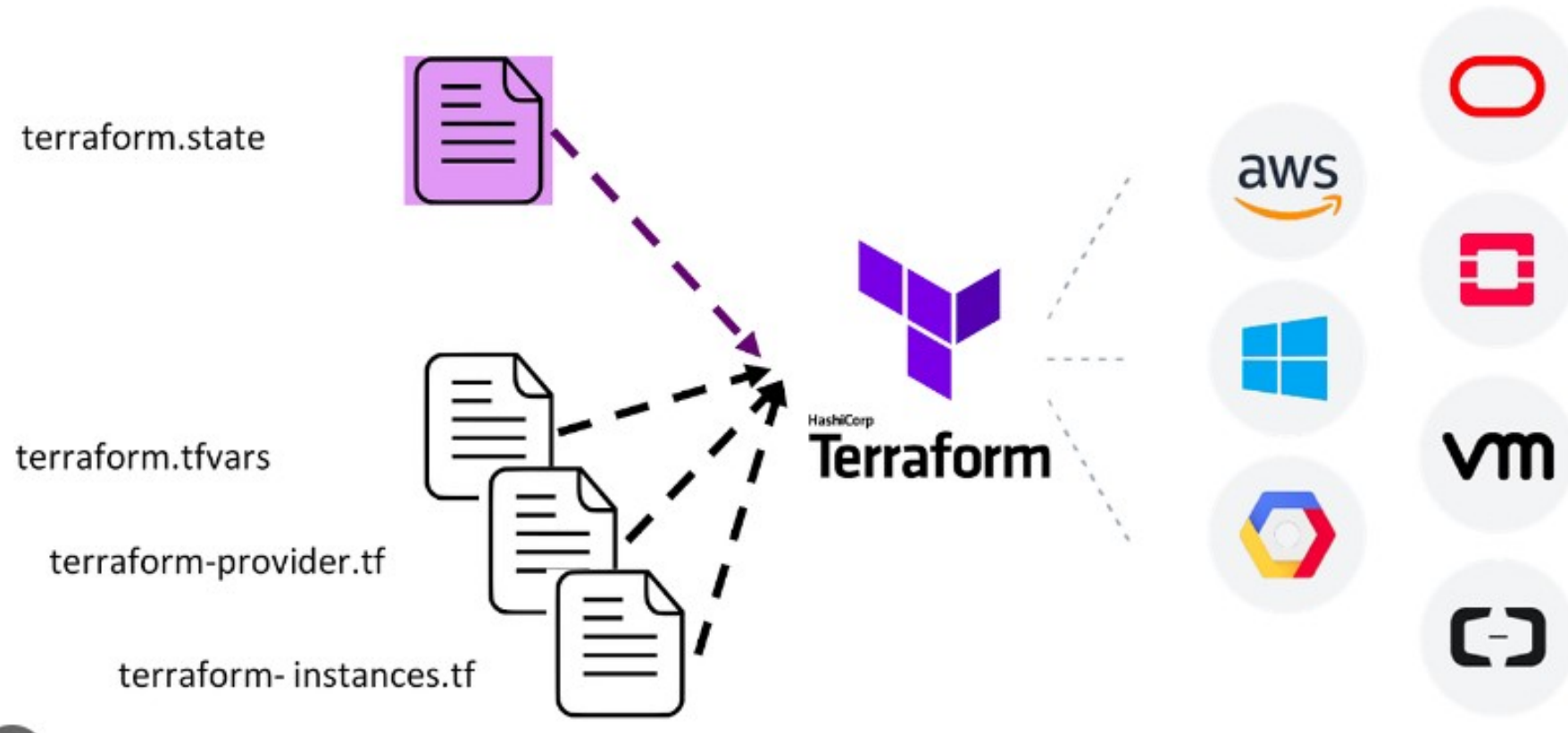
```
git commit -m "meu-branch"
```

# enviando para o GitHub

```
git push --set-upstream origin meu-branch
```



# Terraform



# Ferramentas

 **dashboards online**

# Docker



Fig

# Máquinas Virtuais

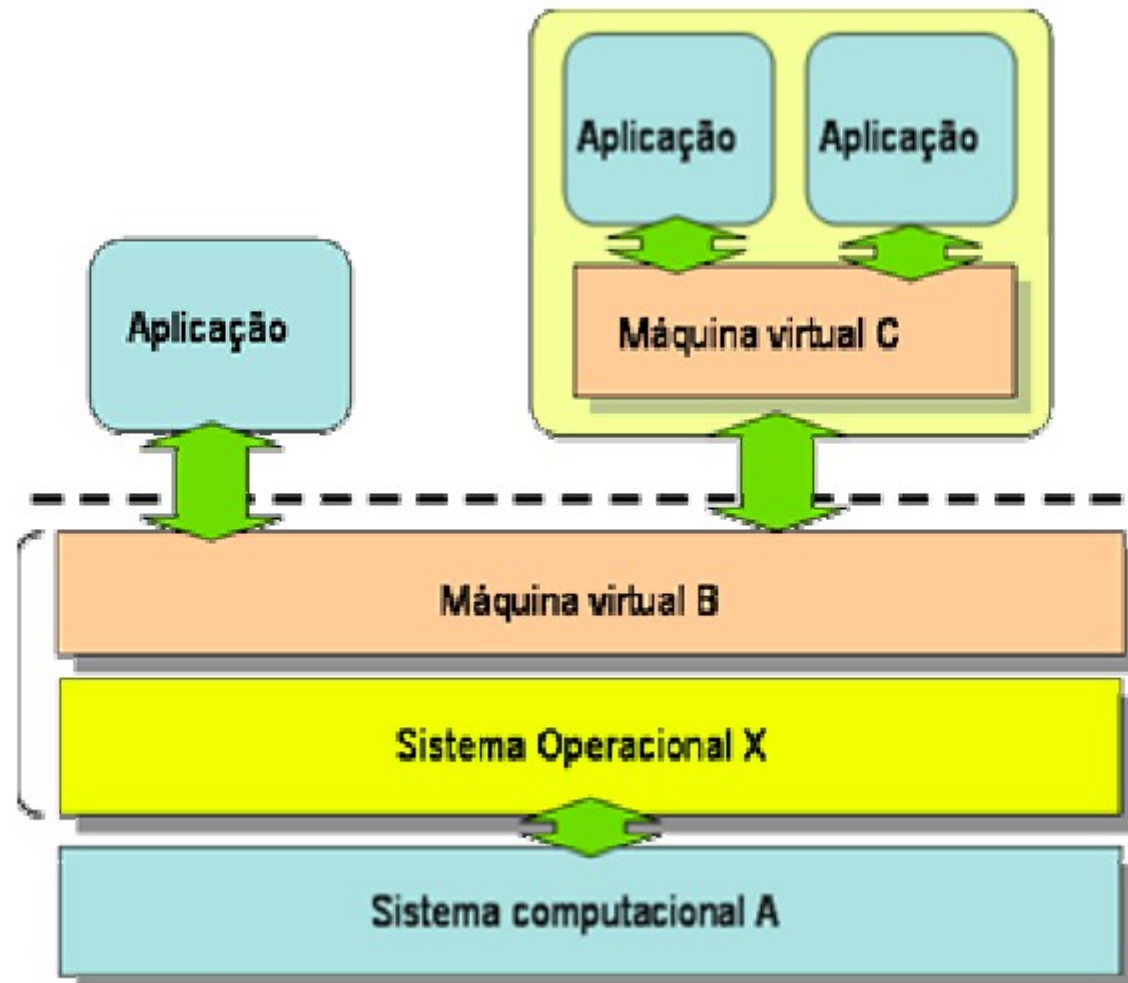
- **Máquinas Virtuais de Tipo 1 (Bare Metal):** Essas VMs são executadas diretamente sobre o hardware, sem a necessidade de um sistema operacional hospedeiro. O hipervisor age como um sistema operacional dedicado para gerenciar a execução das VMs. Isso resulta em melhor desempenho e eficiência, sendo comumente utilizado em ambientes de produção e servidores.

## Máquinas Virtuais

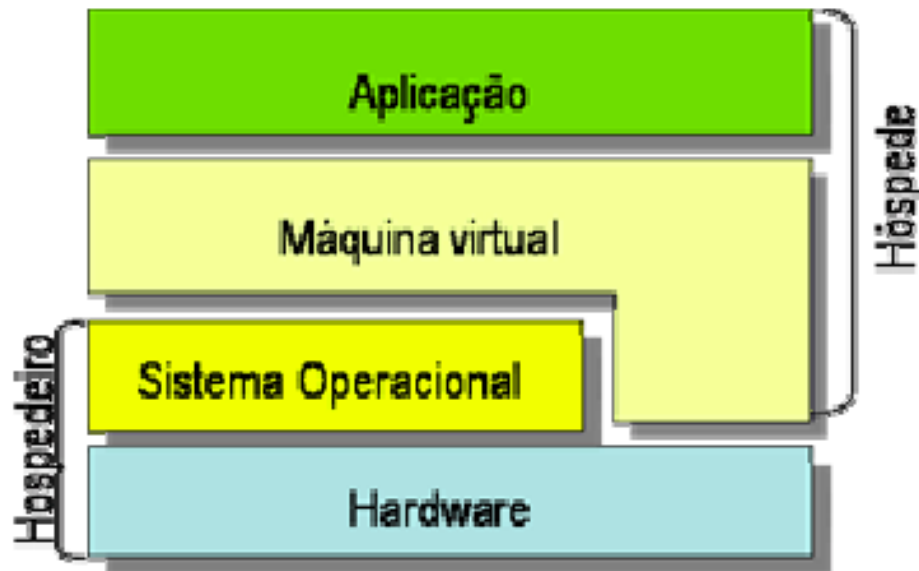
- **Máquinas Virtuais de Tipo 2 (Hospedadas):** Essas VMs são executadas sobre um sistema operacional hospedeiro convencional. Um aplicativo de virtualização é responsável por gerenciar as VMs, e o sistema operacional hospedeiro executa tarefas essenciais. Embora sejam mais fáceis de configurar e usar, as VMs de Tipo 2 geralmente possuem um desempenho inferior em comparação com as de Tipo 1 e são mais adequadas para ambientes de desenvolvimento e testes.



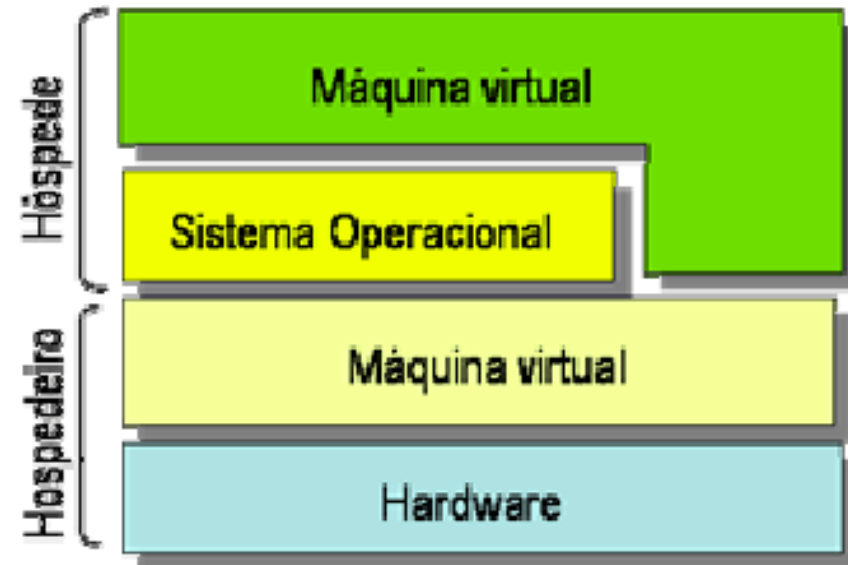
# Máquinas Virtuais



# Máquinas Virtuais



(a) Máquina virtual de processo



(b) Monitor de máquina virtual

# TEMA 2 – CONTAINERS: INTRODUÇÃO AO DOCKER

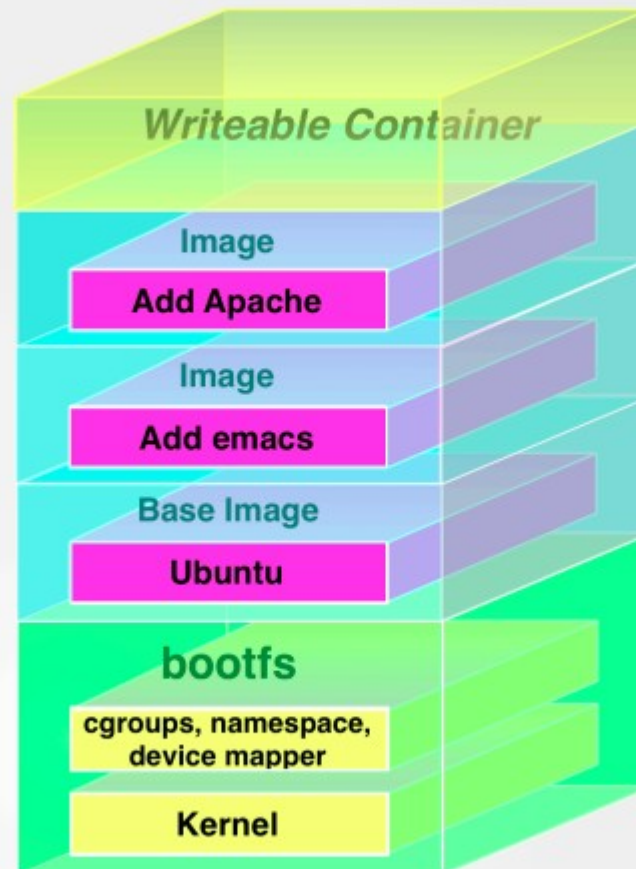
21  
15

# Contêiner

Contêineres são ambientes leves e isolados que encapsulam uma aplicação e suas dependências. Eles proporcionam consistência entre diferentes ambientes de desenvolvimento, teste e produção, eliminando as divergências que podem surgir devido a diferenças nos sistemas operacionais ou configurações.

Um contêiner representa um ambiente isolado para o seu código, caracterizado pela ausência de conhecimento sobre o sistema operacional ou os arquivos do host.

# Contêiner





## Docker - Definição

Docker consiste em sistemas de arquivos sobrepostos, sendo a base um sistema de arquivos de inicialização, chamado bootfs, semelhante ao típico sistema de arquivos de inicialização do Linux/Unix. Normalmente, os usuários do Docker não interagem diretamente com o sistema de arquivos de inicialização. Após a inicialização de um contêiner, ele é movido para a memória e o sistema de arquivos de inicialização é desmontado para liberar a RAM usada pela imagem do disco (DOCKER, 2019).

# Docker

← → ↻ 🔒 docker.com



🌟 **New** Announcing Docker Build Cloud: Develop up to 39X faster →

[Docs](#) [Get support](#) [Contact sales](#)



[Products](#) ▾ [Developers](#) ▾ [Pricing](#) [Blog](#) [About Us](#) ▾ [Partners](#)



[Sign In](#)

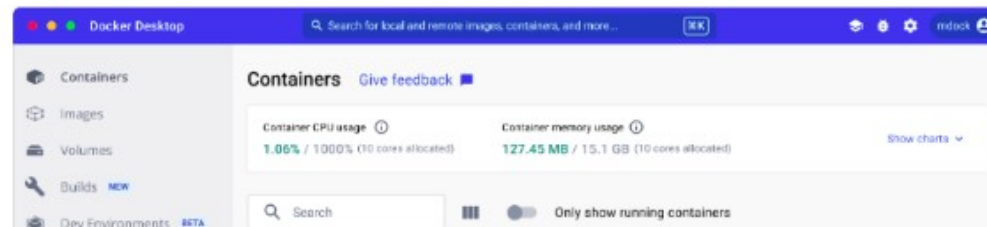
[Get Started](#)

# Develop faster. Run anywhere.

Build with the **#1 most-used developer tool**

Download for Linux ▾

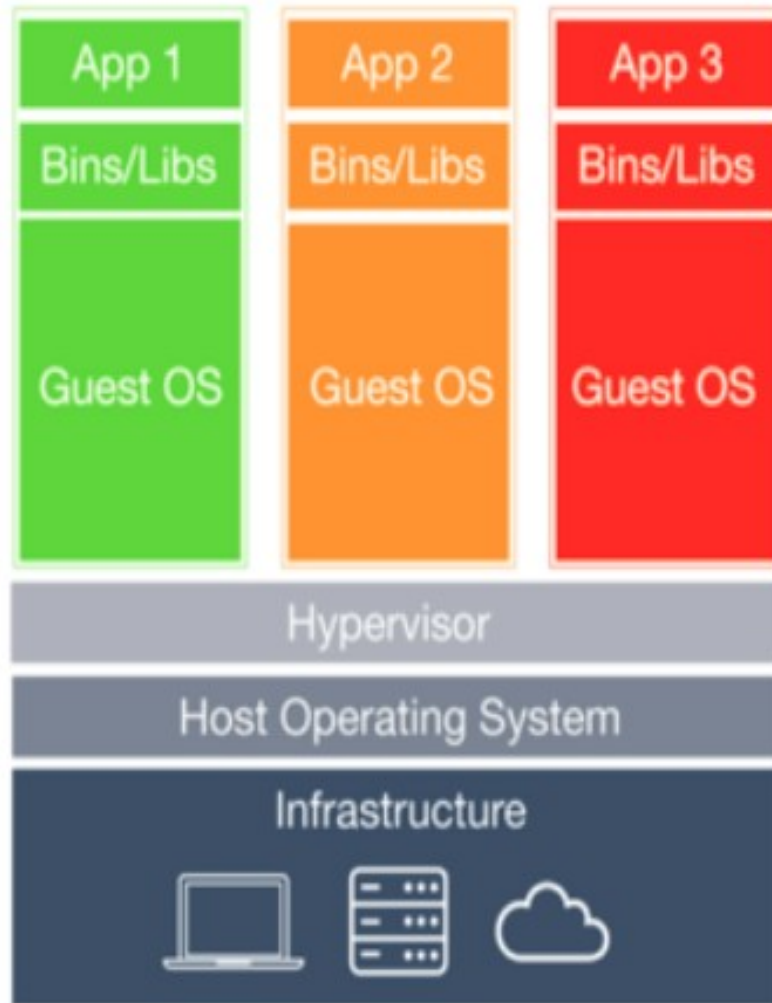
[Learn more about Docker](#)



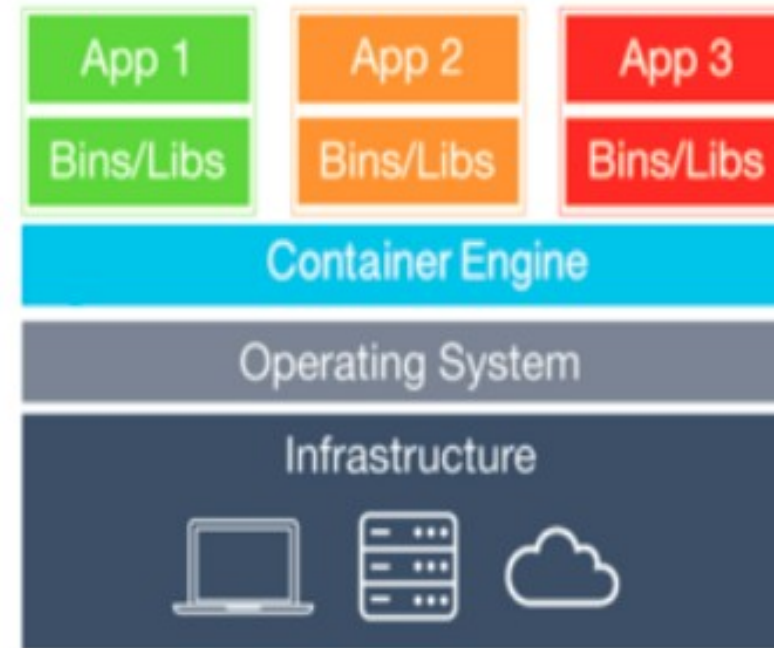
Sesc Senac IFPD

SENAC

# Máquina Virtual x Container



Hypervisor-based Virtualization



Container virtualization

# Vantagens e Desvantagens

- containers proporcionam mais agilidade
- Os containers economizam no licenciamento de VM

# Vantagens e Desvantagens

containers permitem adoção híbrida e multicloud.

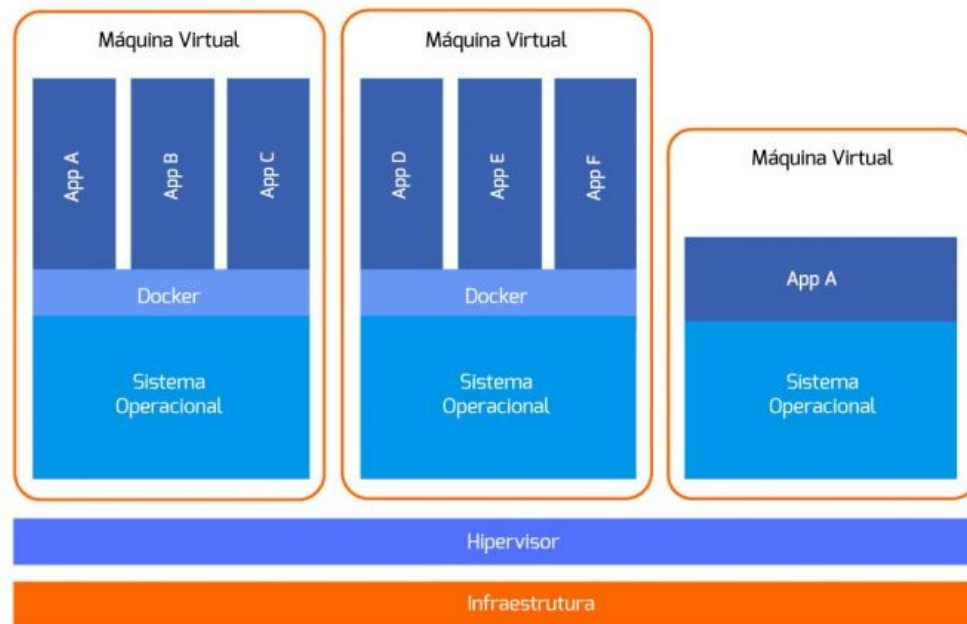


Fonte: <https://vertigo.com.br/containers-vs-maquinas-virtuais/>



# Vantagens e Desvantagens

Os containers se integram a sistemas legados



Fonte: <https://vertigo.com.br/containers-vs-maquinas-virtuais/>

# YAML

← → ↻ 🔒 yaml.org

%YAML 1.2

---

YAML: YAML Ain't Markup Language™

What It Is:

YAML is a human-friendly data serialization language for all programming languages.

YAML Resources:

YAML Specifications:

- YAML 1.2:
  - Revision 1.2.2 # Oct 1, 2021 *\*New\**
  - Revision 1.2.1 # Oct 1, 2009
  - Revision 1.2.0 # Jul 21, 2009
- YAML 1.1
- YAML 1.0

YAML Matrix Chat: '#chat:yaml.io' # Our New Group Chat Room!

YAML IRC Channel: libera.chat#yaml # The old chat

YAML News: twitter.com/yamlnews

YAML Mailing List: yaml-core # Obsolete, but historical

YAML on GitHub: # github.com/yaml/

YAML Specs: yaml-spec/

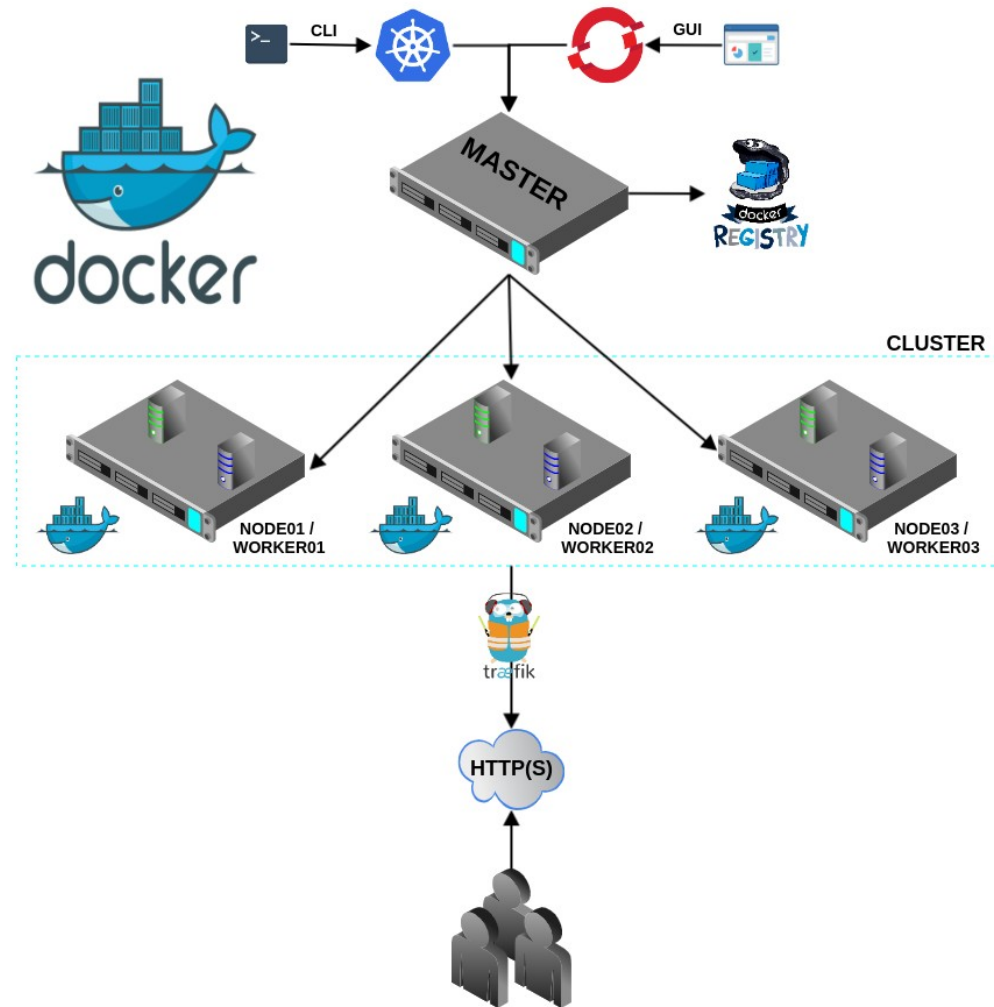
YAML 1.2 Grammar: yaml-grammar/

YAML Test Suite: yaml-test-suite/

YAML Issues: issues/

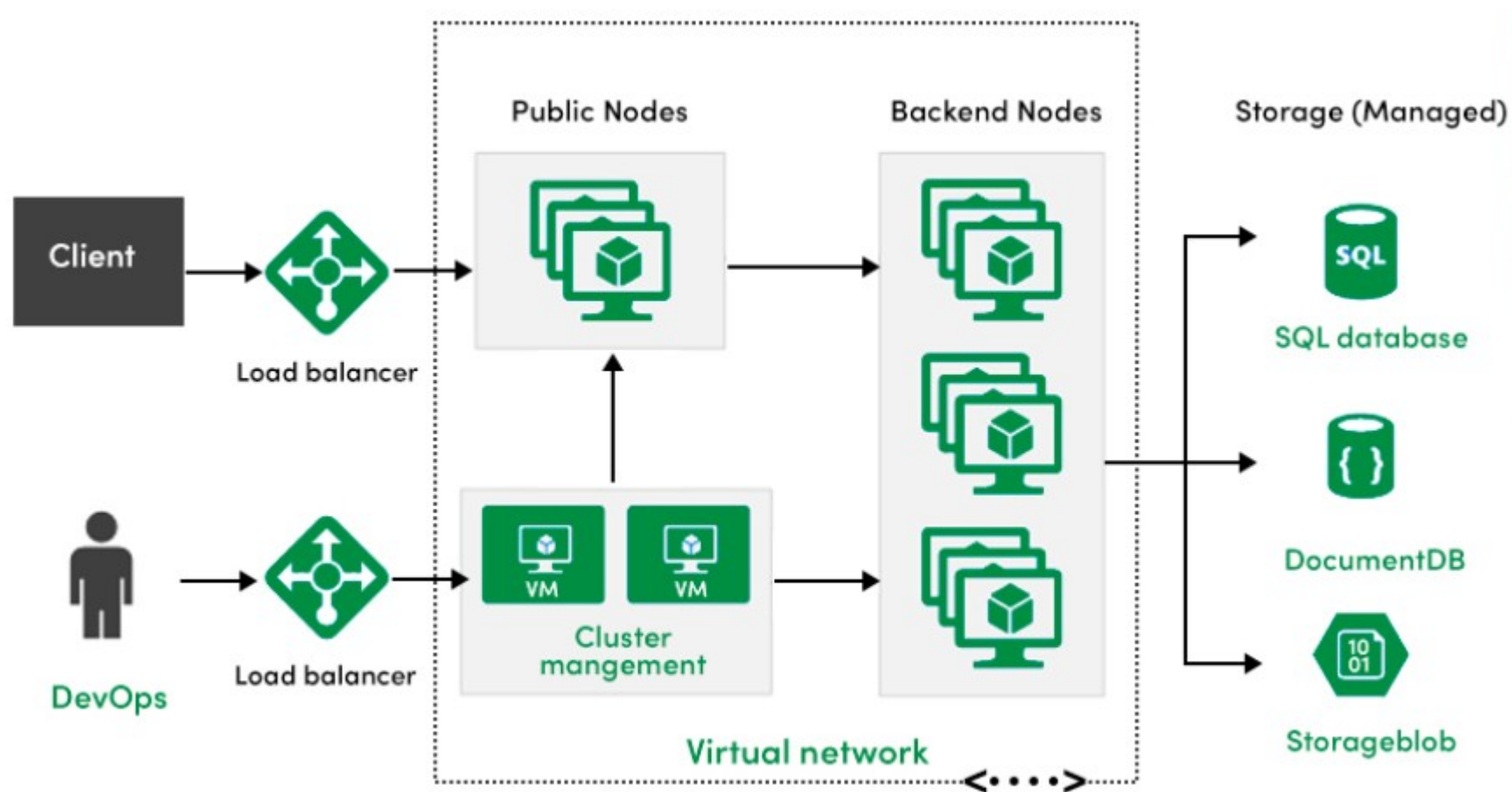
<https://yaml.org/>

# KUBERNETES

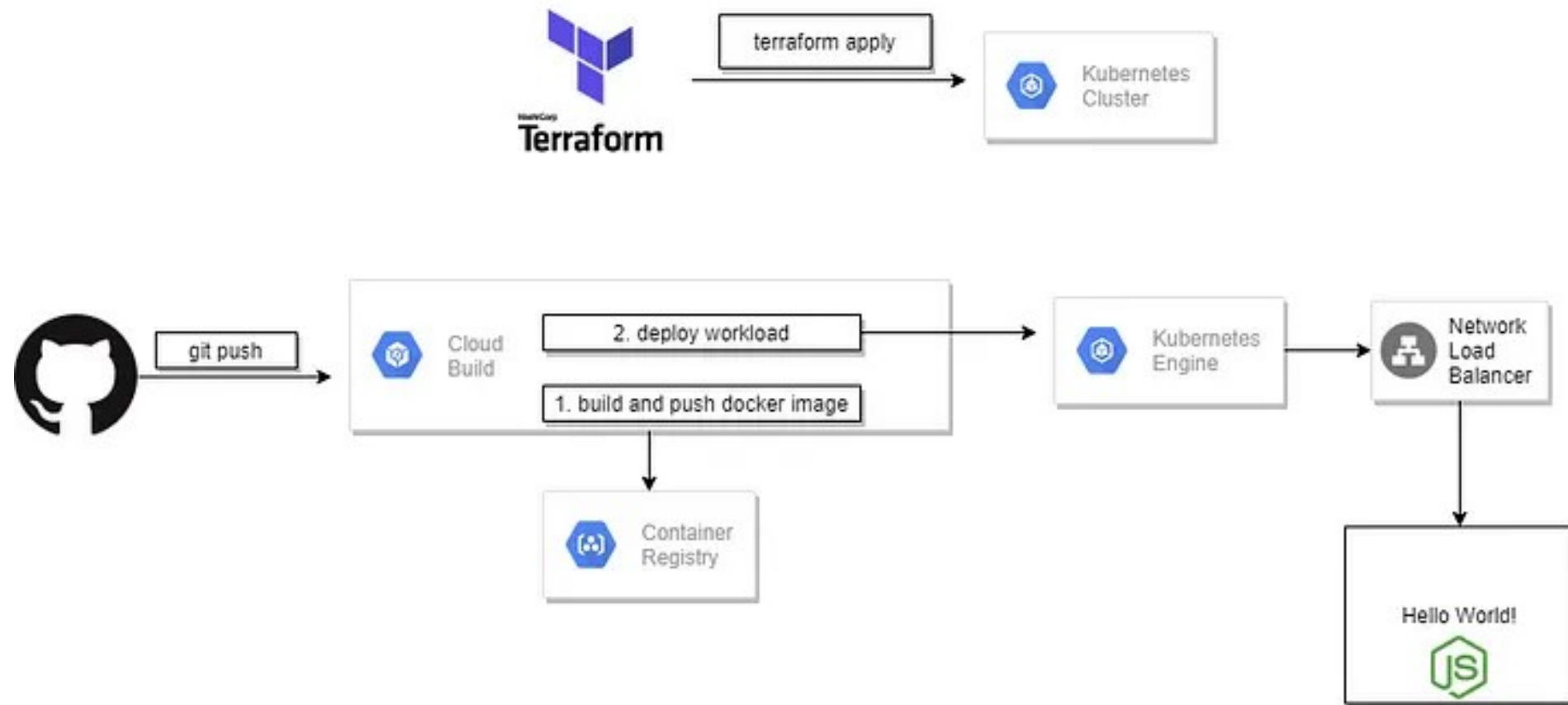


<https://yaml.org/>

# Orquestração com Kubernetes



# Contextualização IAC



# Vantagens e Desvantagens

- Com a automatização do provisionamento e da configuração de recursos, a infraestrutura pode ser criada, modificada ou removida rapidamente, reduzindo o tempo de lançamento de novos produtos e serviços.
- Redução de erros: A padronização e a consistência do código, aliadas aos testes automatizados, diminuem significativamente a quantidade de erros humanos e a probabilidade de falhas na infraestrutura.

# Vantagens e Desvantagens

- Versionamento e rastreabilidade: O uso de sistemas de controle de versão, como Git, permite rastrear mudanças, identificar problemas e reverter para versões anteriores, facilitando a manutenção e a auditoria.
- Escalabilidade e flexibilidade: A automação e a padronização proporcionadas pelo Terraform e IaC facilitam a escalabilidade de recursos e a adaptação a diferentes ambientes, diminuindo a necessidade de intervenção manual e possibilitando o uso de infraestrutura imutável.



## Vantagens e Desvantagens

- Redução de custos: A implementação de Terraform e IaC permite otimizar o uso de recursos, identificar e eliminar desperdícios, e ajustar a infraestrutura de acordo com as necessidades específicas do negócio. Isso resulta em economia de custos e maior eficiência operacional.
- Segurança e conformidade: A padronização do código e a integração com políticas de segurança garantem a conformidade com as normas e regulamentações. Além disso, o uso de IaC permite a criação de ambientes isolados, que facilitam a detecção e a correção de vulnerabilidades.