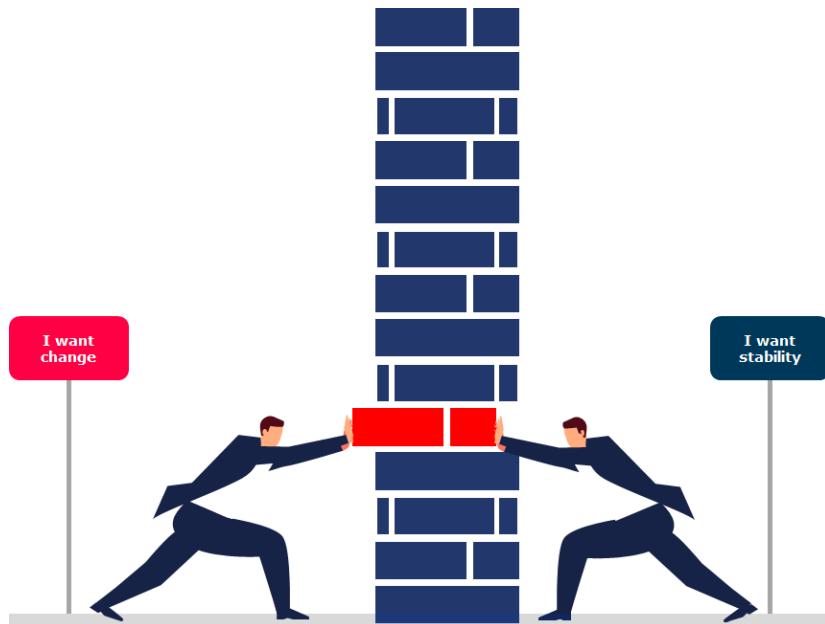


Engenheiro de Qualidade de Software

Development x Operations

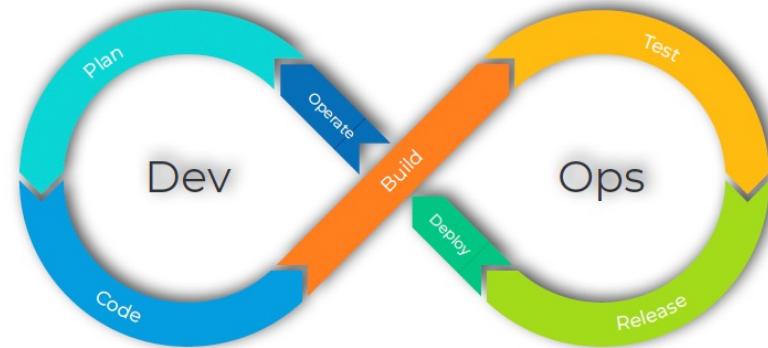


Development x Operations



DevOps

- DevOps é um processo de modificação cultural, baseado em pilares e focado em transformar como desenvolvemos e operamos softwares em empresas ágeis.
- Engloba qualquer coisa que ajude a comunicação e colaboração entre as áreas de desenvolvimento, operações e negócio.



Quando surgiu?

2008

Infraestrutura Ágil



Surgimento do termo infraestrutura ágil em listas de discussão, conversas sobre metodologia ágil para infraestrutura no Agile 2008.

2009

Termo DevOps



Criação do termo DevOps durante a conferência Velocity da O'Reilly,

2010

CAMS



Na edição Mountain View do DevOpsDays, John Willis e Damon Edwards criaram o acrônimo CAMS que define 4 eixos para o DevOps: Culture, Automation, Measurement and Sharing.

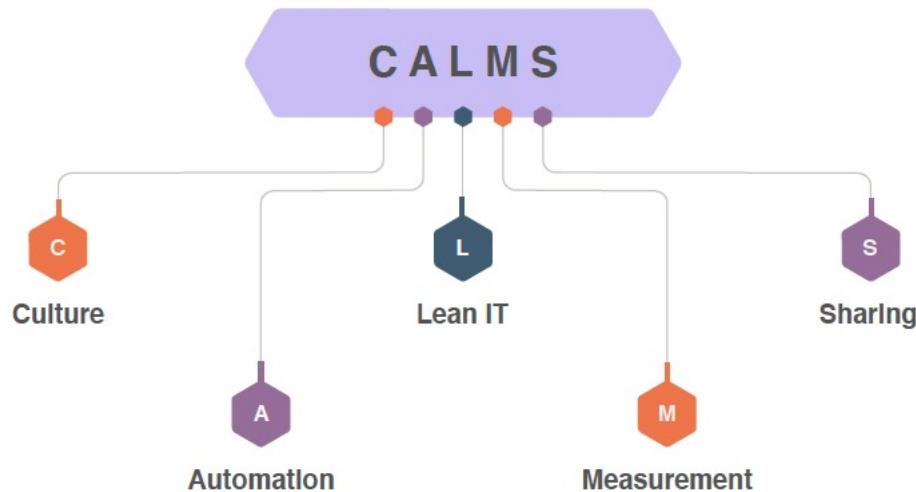
2012

CALMS



Jez Humble adicionou a letra L, referenciando o Lean, formando assim o CALMS

Pilares DevOps



Culture

- Relacionamento
- Organização
- Metodologia
- Ambiente
- Comportamento

***“You can’t directly change culture.
But you can change behavior, and
behavior becomes culture”***

Lloyd Taylor VP Infrastructure, Ngmoco



Automation

- Build
- Deploy
- Testes
- Monitoramento
- Configuração



TECHNOLOGY

Lean IT

- Mapear os processos
- Identificar gargalos
- Reduzir desperdícios
- Melhoria contínua



Measurement

- Métricas
- Feedback
- Conhecimento
- Previsibilidade
- Logs e Dashboards

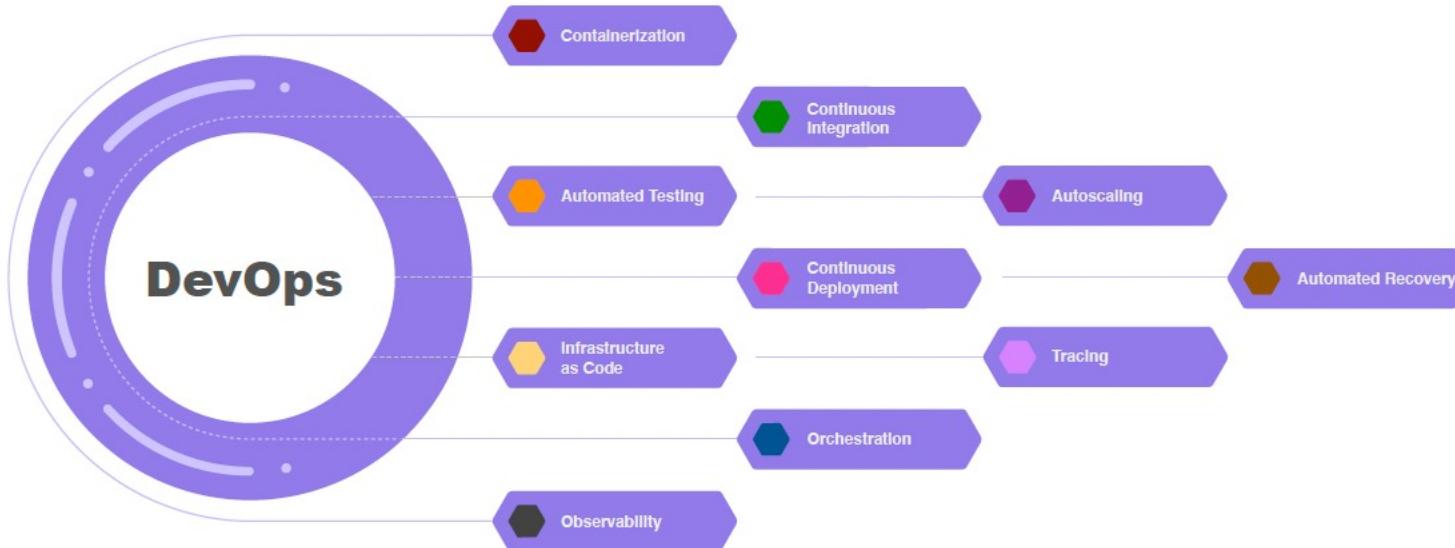


Sharing

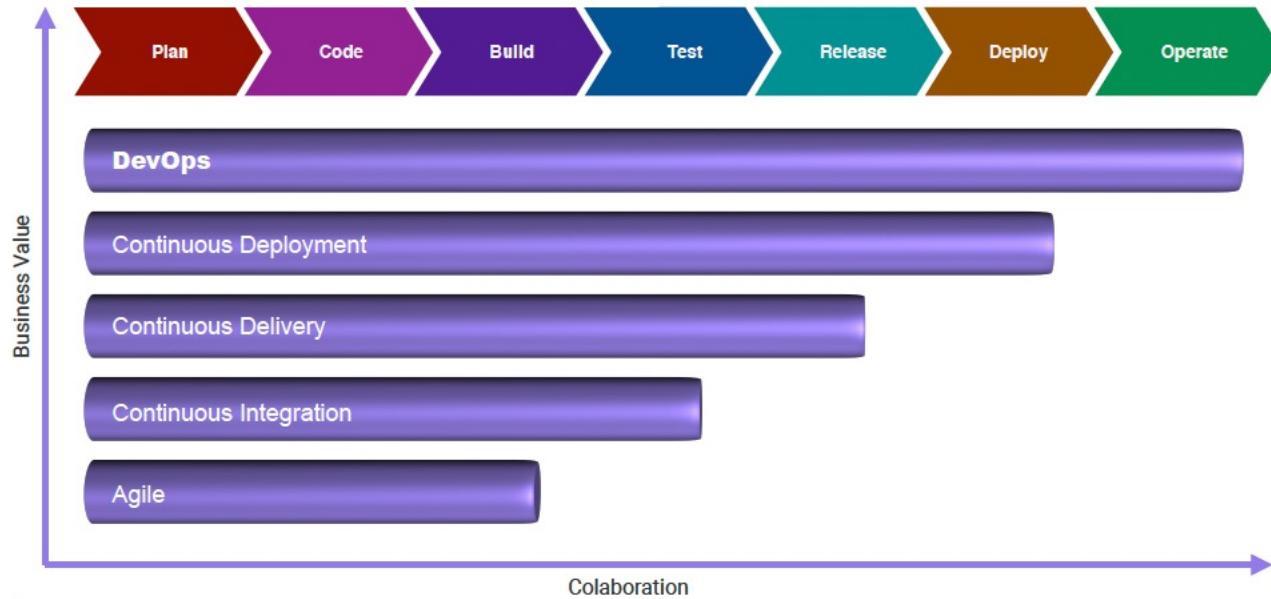
- Descentralização do conhecimento
- Nivelamento
- Autonomia
- Confiança



Práticas DevOps

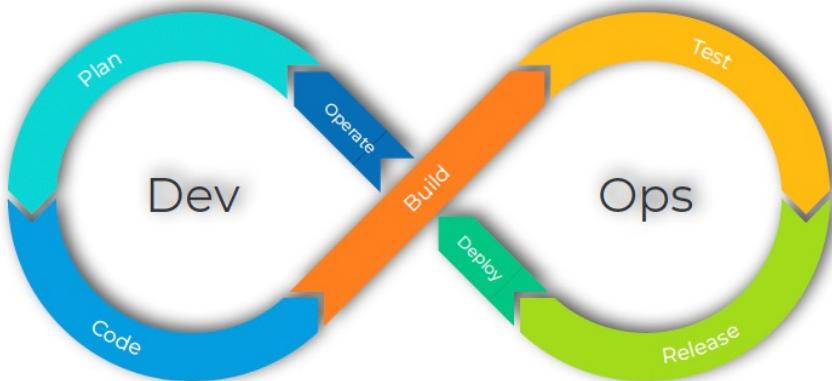


Ciclo DevOps



CI & CD

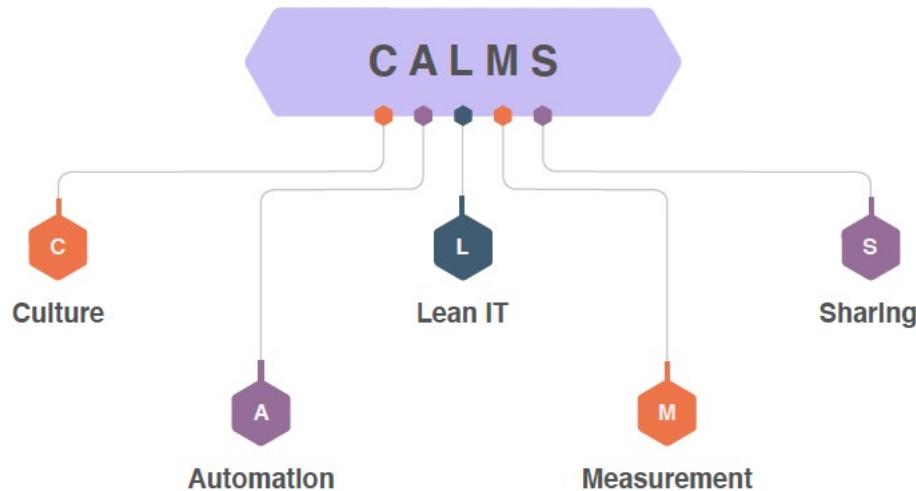
Review DevOps



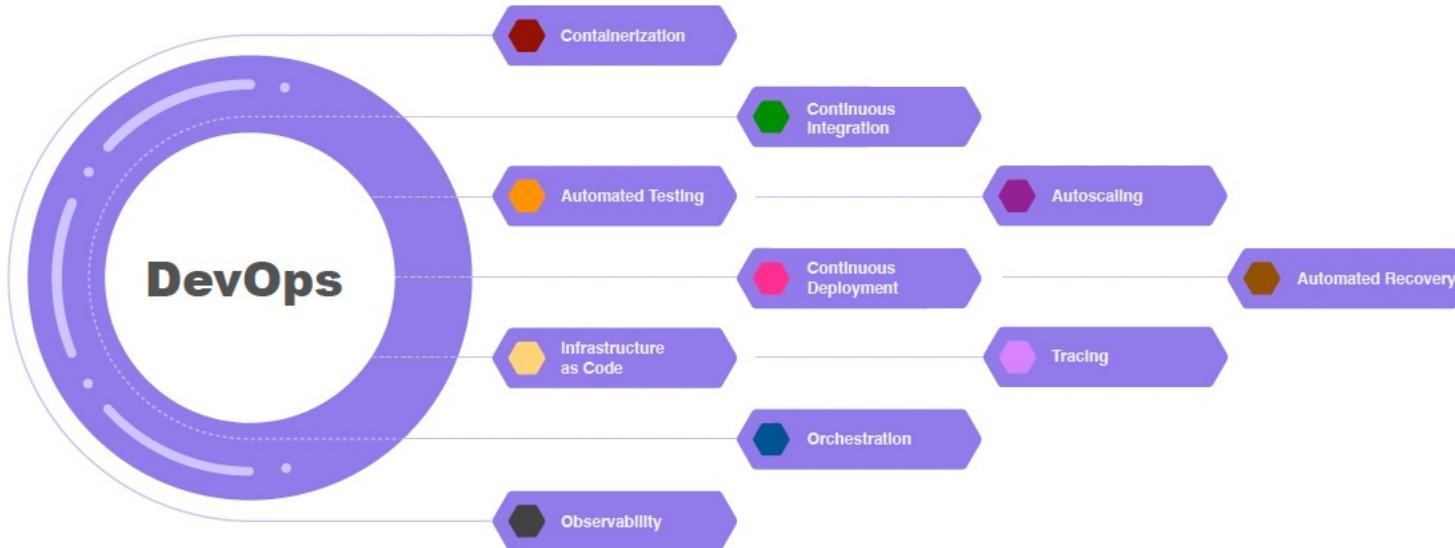
Review DevOps

- Cultura

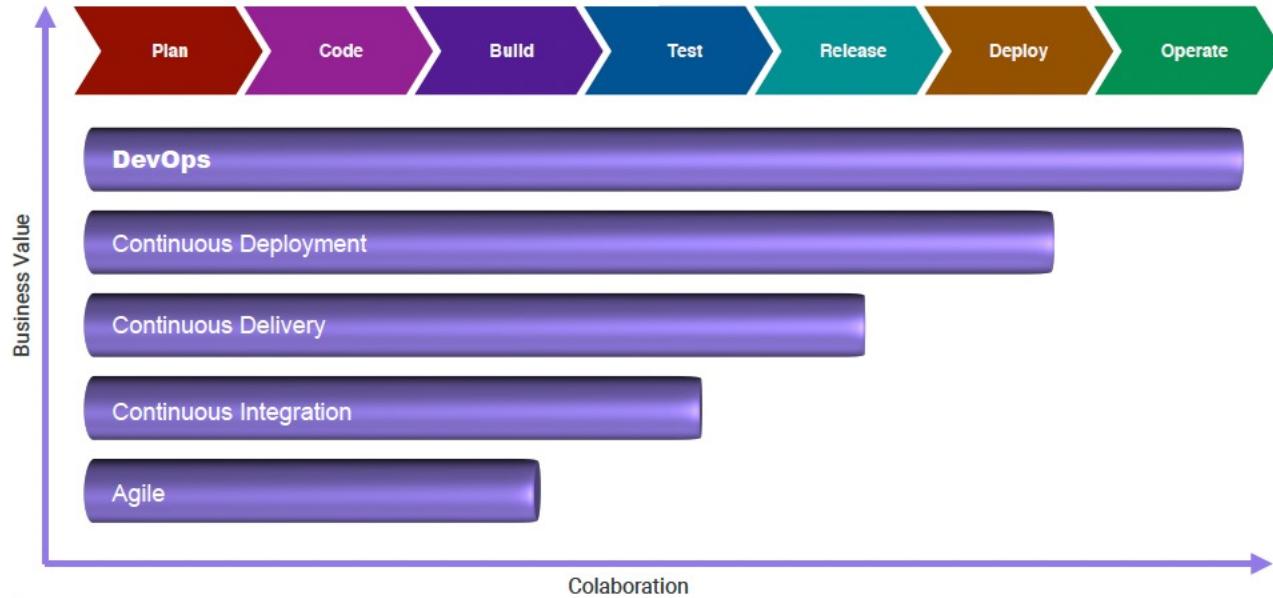
- Processos



Review DevOps



Review DevOps



Continuous Integration

Equipe de desenvolvimento de software

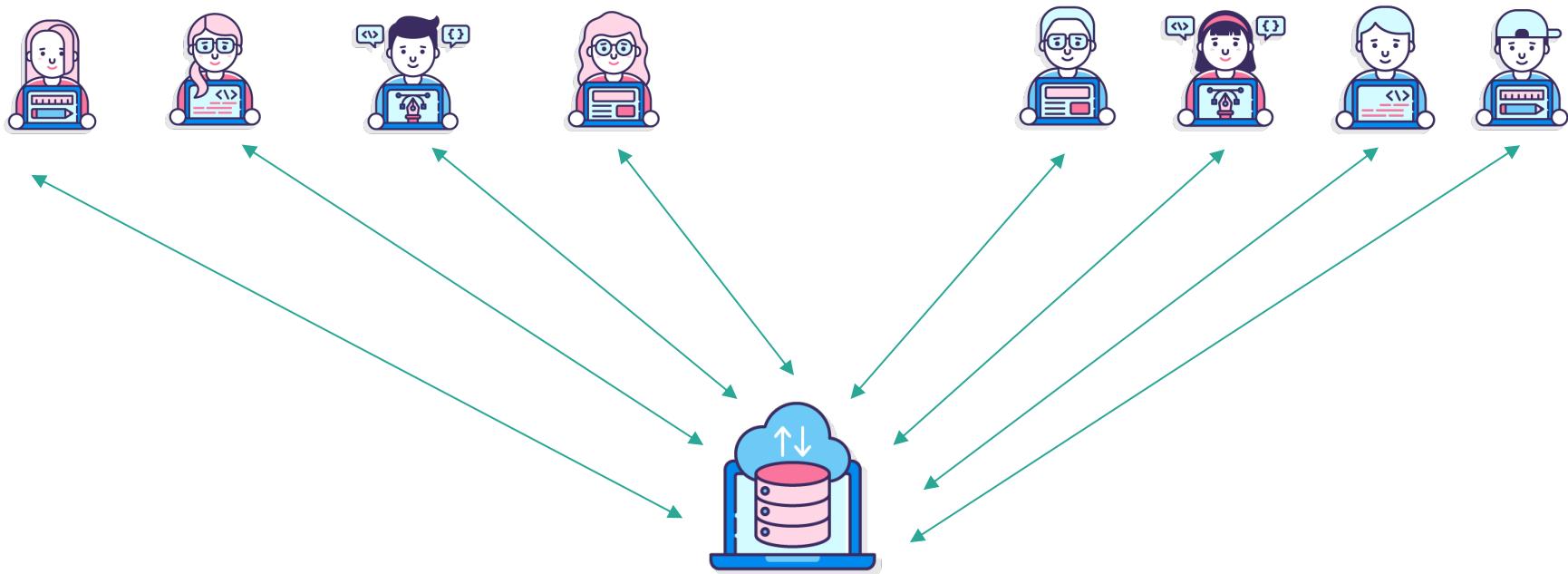
- Projeto grande
- Múltiplas equipes
- Trabalhando simultaneamente
- Alterando trechos do mesmo código



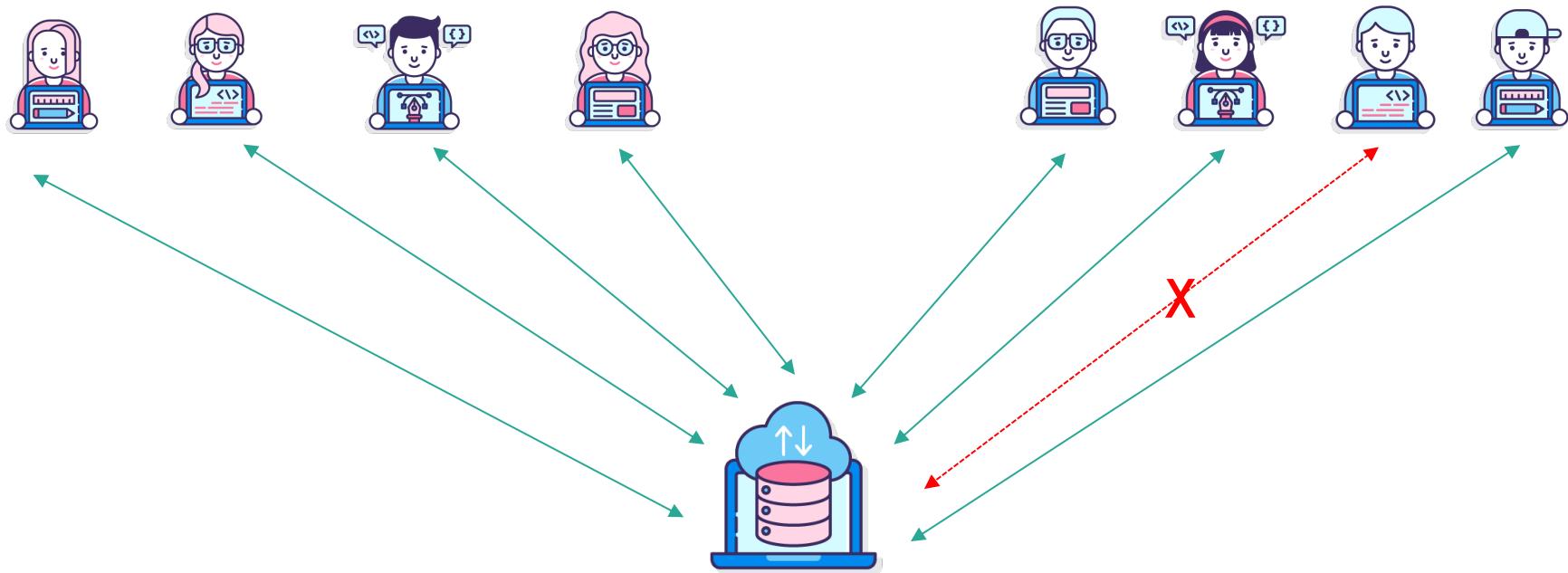
Equipe de desenvolvimento de software



Equipe de desenvolvimento de software



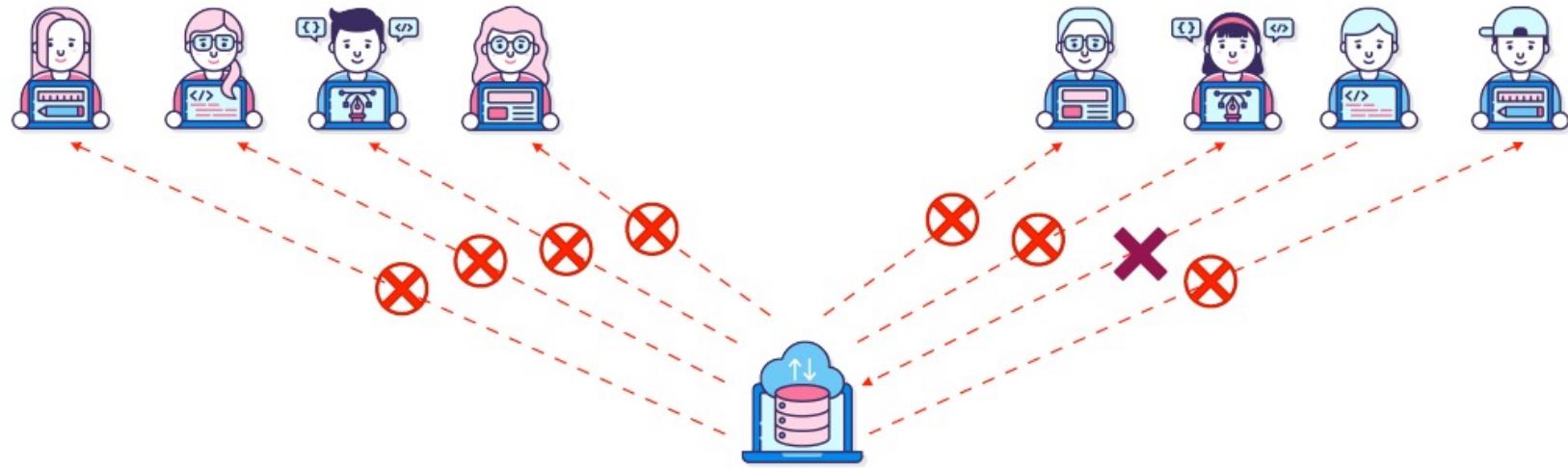
Equipe de desenvolvimento de software



Resultado



Equipe de desenvolvimento de software

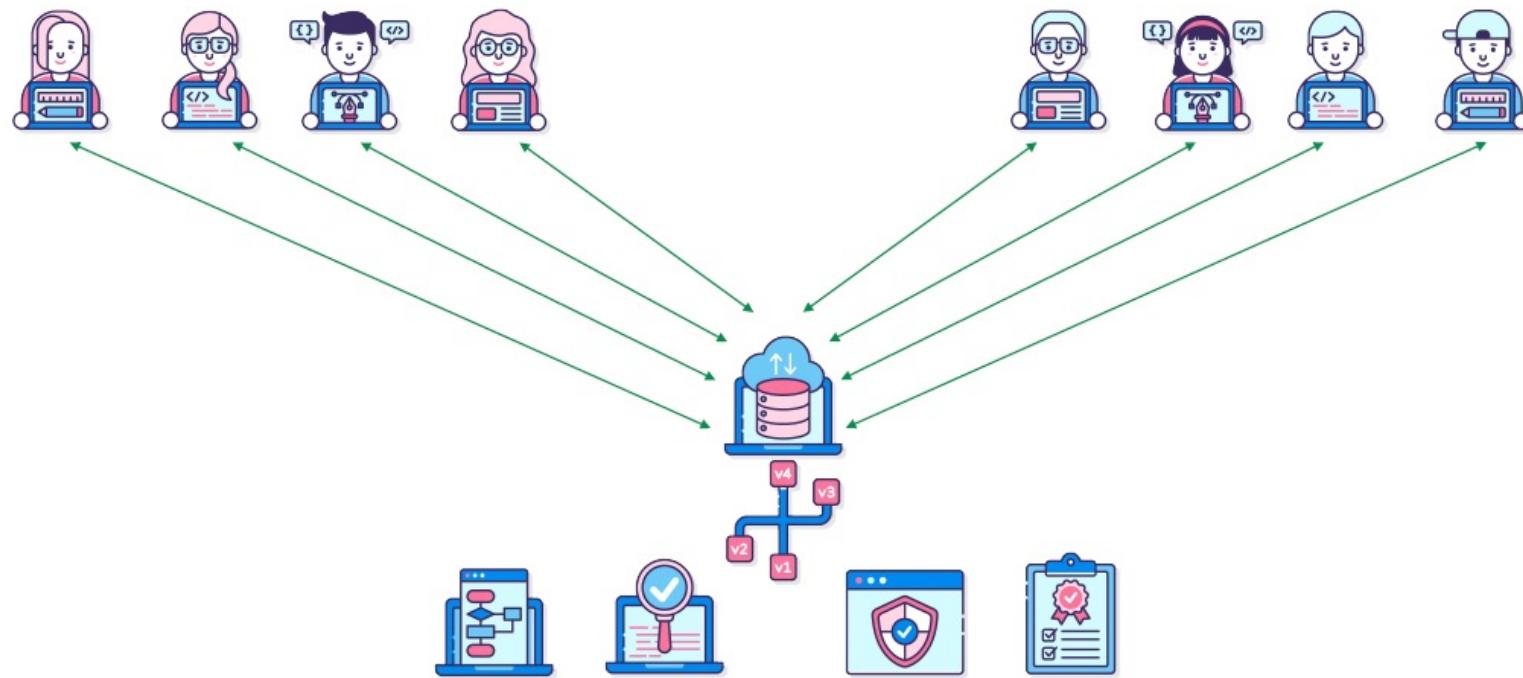


Integração contínua

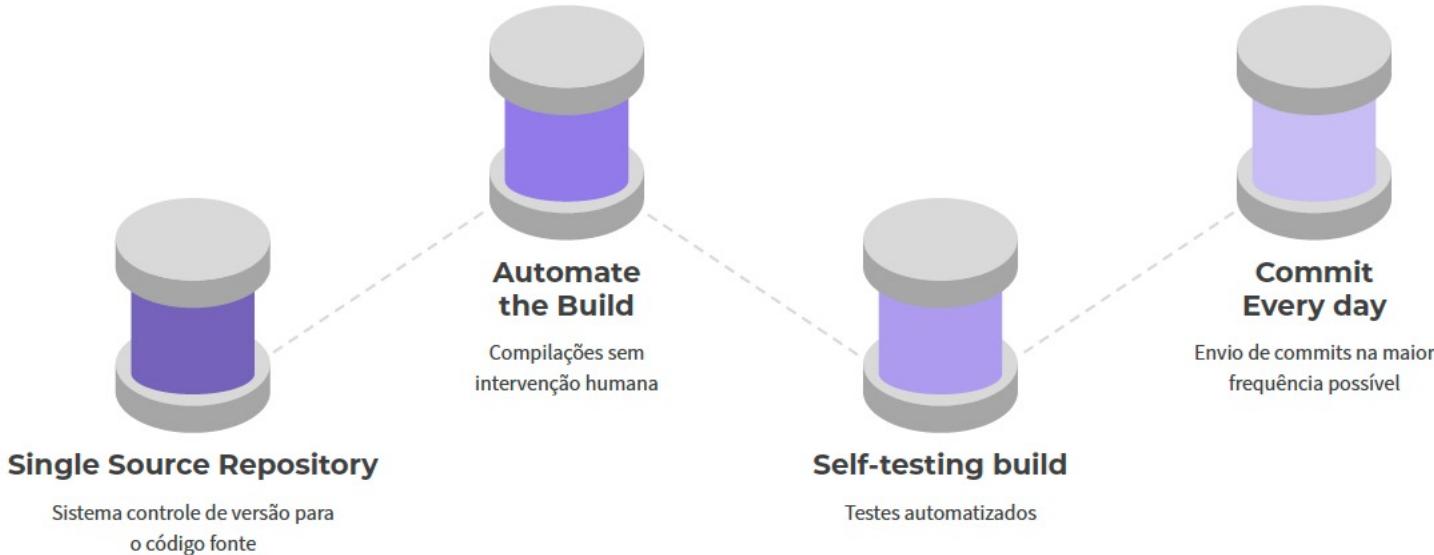
"IC é uma prática na qual o código que está sendo desenvolvido é integrado, versionado, construído e verificado diversas vezes ao dia em um ambiente dedicado..."

WILDT, MOURA, LACERDA e HELM. eXtreme Programming, 2018

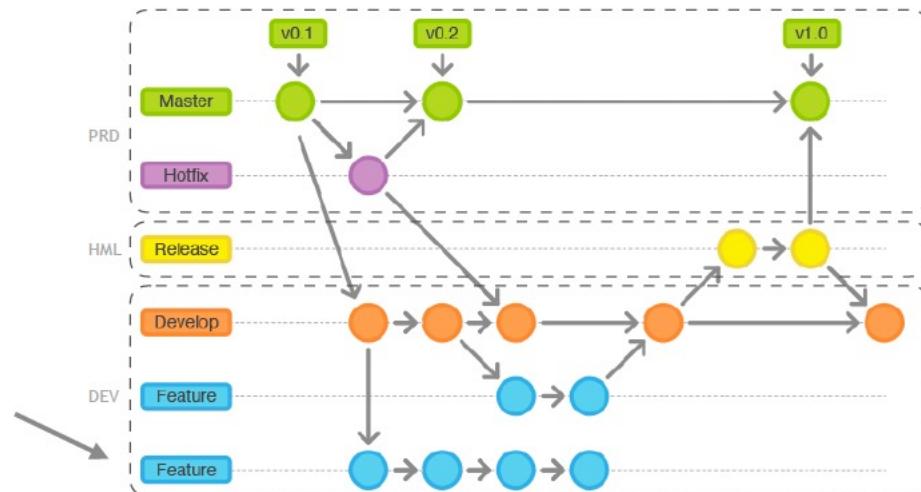
Integração contínua



Integração contínua



GitFlow



Integração contínua



git



GitHub



GitLab



Azure DevOps



Bamboo



Jenkins



Travis CI

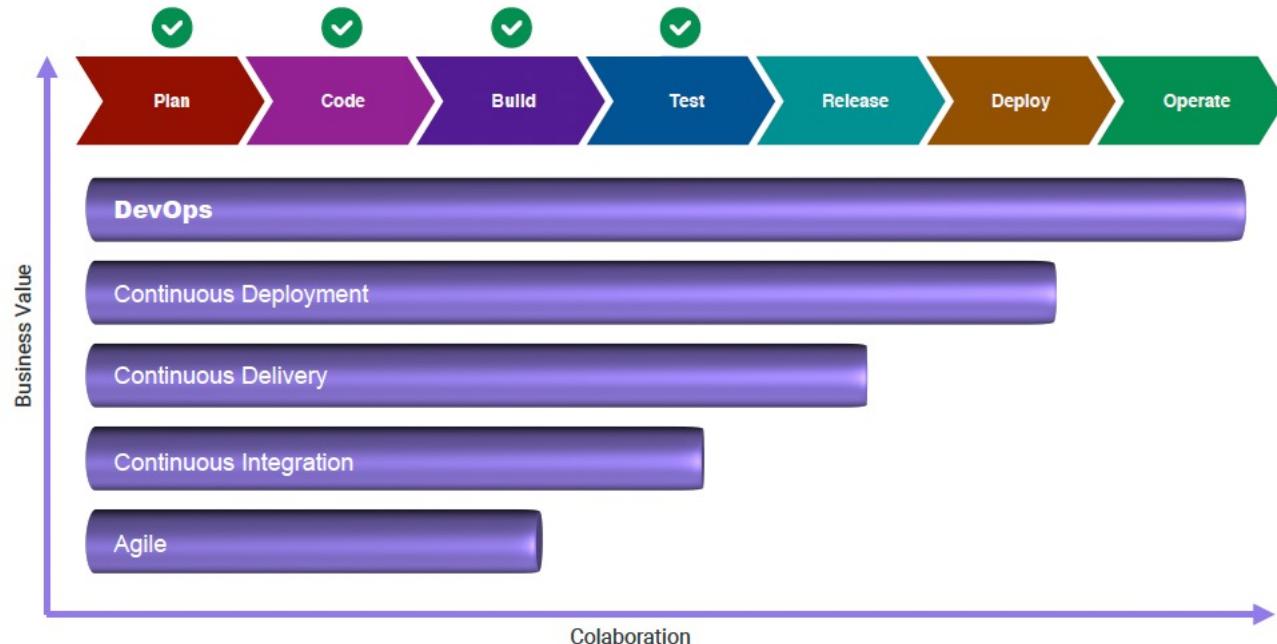
> go

Integração contínua

**"Integração Contínua não livra
os bugs, mas os tornam
dramaticamente mais fáceis
de encontrar e remover."**

Martin Fowler

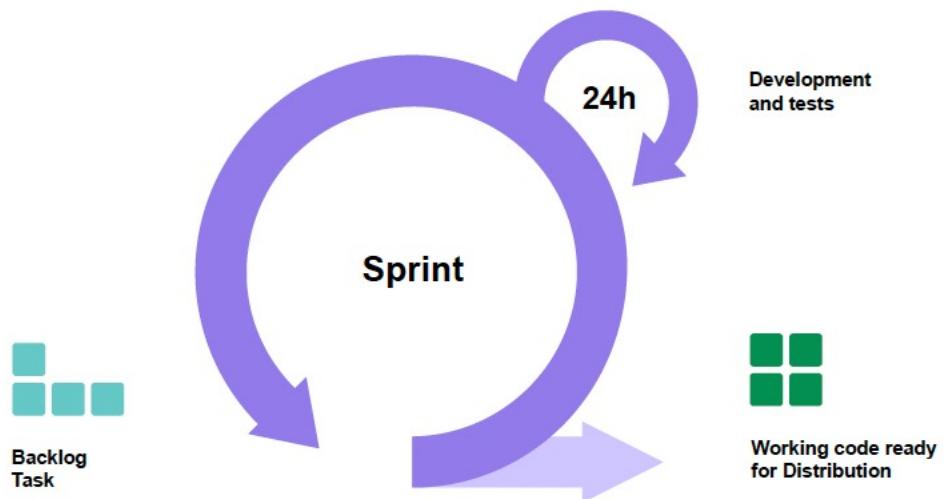
Review DevOps



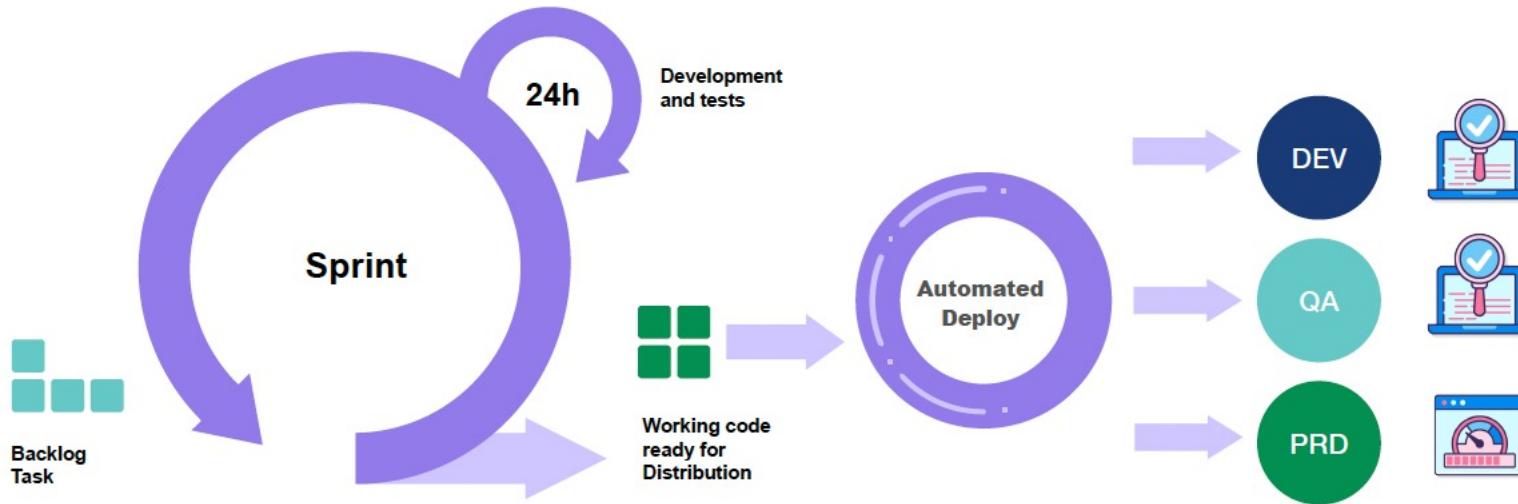
Continuous Delivery

Continuous Deployment

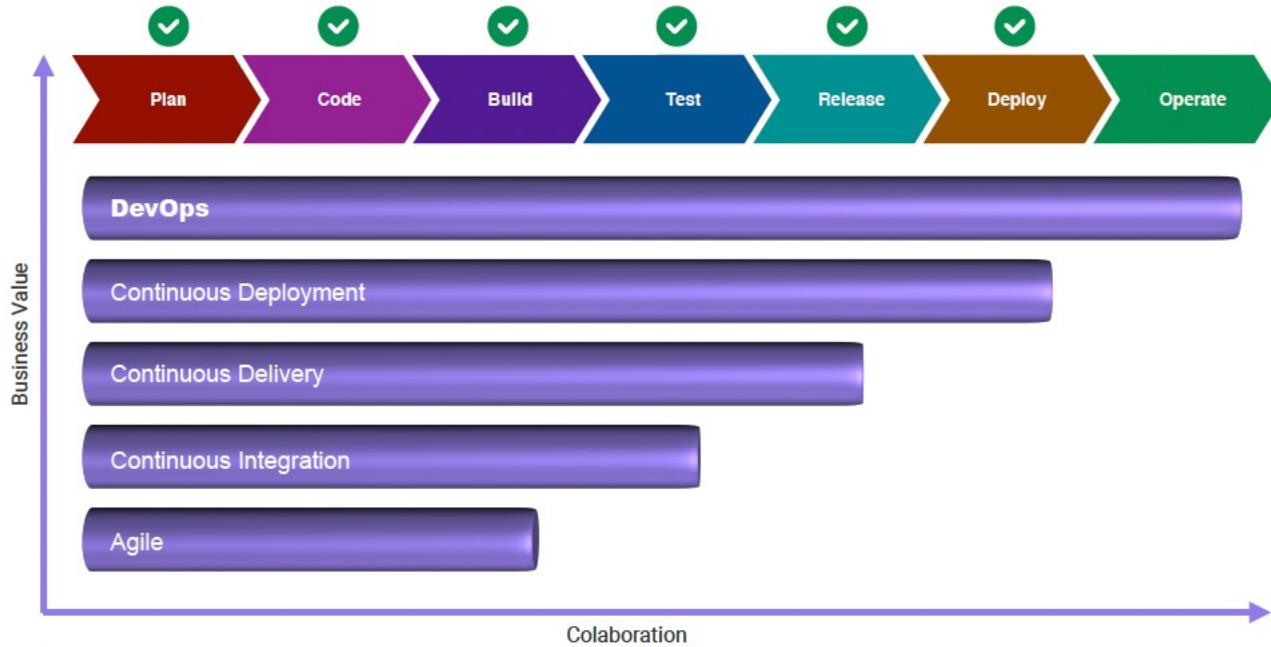
Continuous Delivery



Continuous Deployment



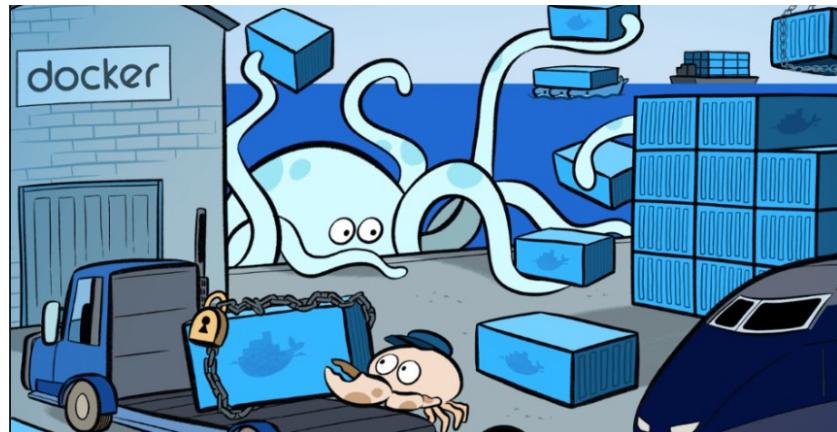
Continuous DevOps



Docker

O que é Docker?

- Estivador?
- Não é máquina virtual;
- Plataforma, open-source, que permite desenvolver, enviar e executar aplicativos como containers;
- Desenvolvido em GO LANG;
- Utiliza o LXC (Linux containers), um subsistema do Kernel do Linux



Por que Docker?

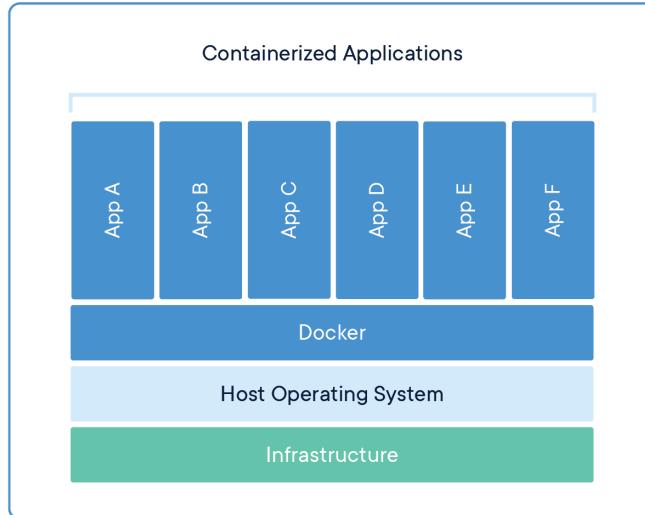
Padronização e replicação de ambientes;

- Ambientes de testes integrados;
- Simples implementação e fácil manutenção;
- Economia de recursos;
- Comunidade Docker ativa.

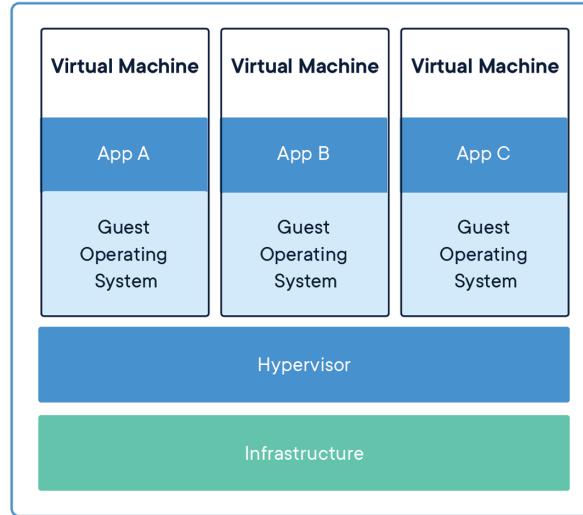
“na minha máquina funciona”



Containers e Máquinas Virtuais



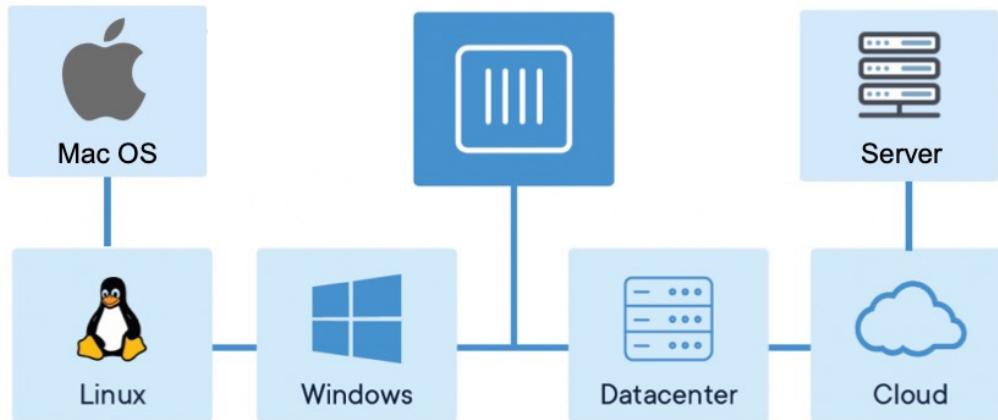
Os contêineres são uma abstração na camada do aplicativo que agrupa o código e as dependências.



As máquinas virtuais (VMs) são uma abstração do hardware físico que transforma um servidor em vários servidores.

Distribuição de Containers

Os contêineres docker estão em toda parte: **linux, mac os, windows, data center, nuvem, servidor, etc.**



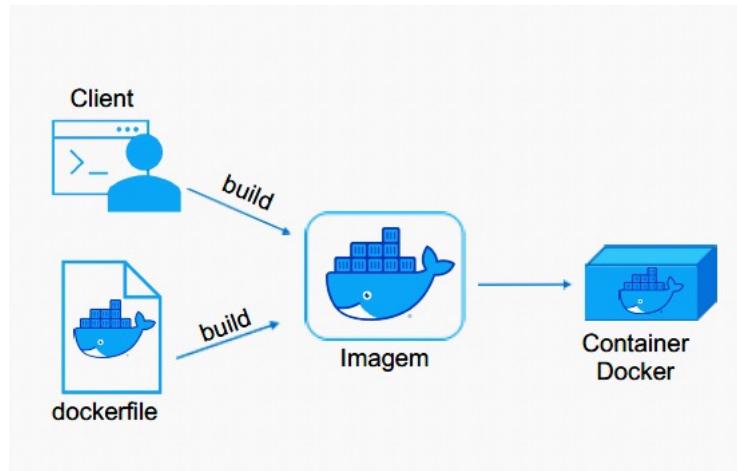
Imagens e Containers

Imagen contém informações sobre o que é necessário para criar container. Normalmente contém uma união de sistemas de arquivos (AUFS) em camadas empilhados (layers) uns sobre os outros.

Container é uma instância de tempo de execução de uma imagem docker.

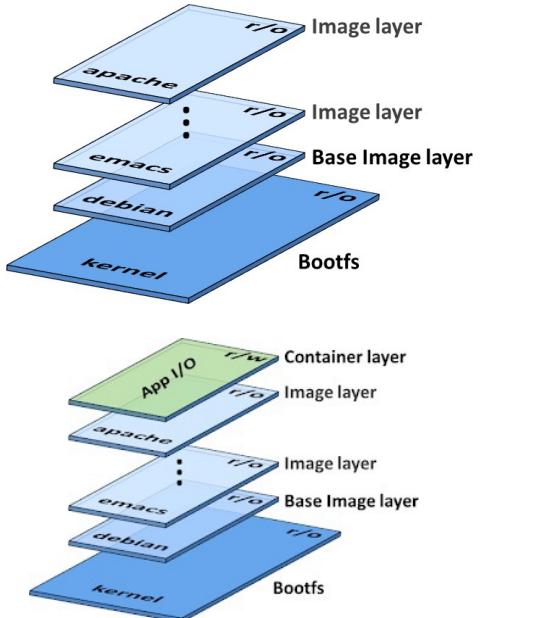
Você pode criar uma imagem docker de duas maneiras:

- Método interativo
- Método dockerfile



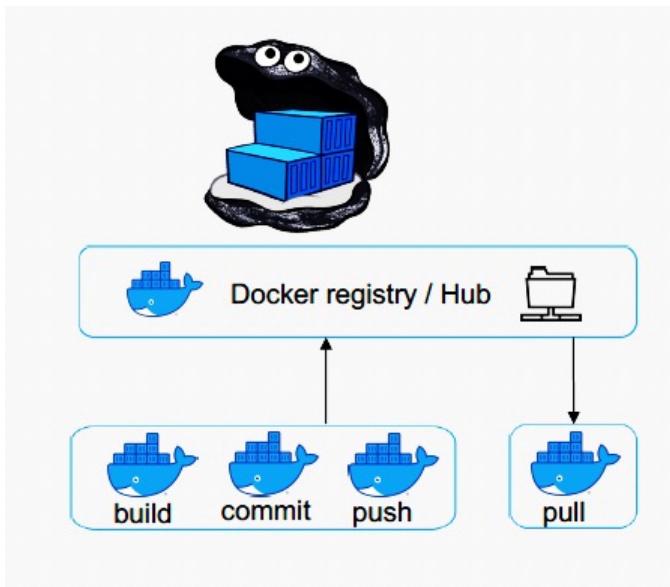
Dockerfile e Layers (Camadas)

```
ex-build-go > 📄 Dockerfile > ...
1  FROM golang:1.11-alpine AS build
2
3  RUN apk add --no-cache git
4  RUN go get github.com/golang/dep/cmd/dep
5
6  COPY Gopkg.lock Gopkg.toml /go/src/project/
7  WORKDIR /go/src/project/
8
9  RUN dep ensure -vendor-only
10
11 COPY . /go/src/project/
12 RUN go build -o /bin/project
13
14 FROM scratch
15 COPY --from=build /bin/project /bin/project
16 ENTRYPOINT ["/bin/project"]
17 CMD ["--help"]
```

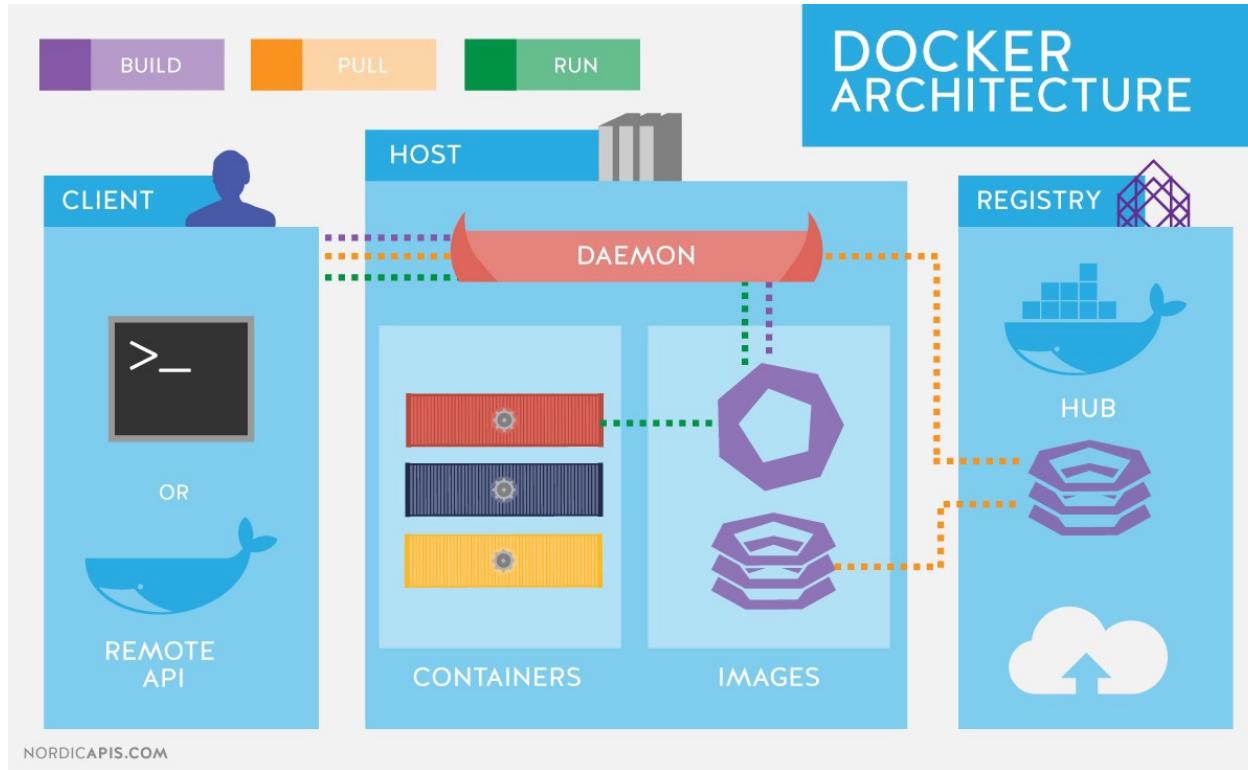


Registry & Docker Hub

- O registry é um aplicativo do lado do servidor sem estado e altamente escalonável que armazena e permite distribuir imagens docker;
- Docker hub é o maior repositório do mundo de imagens e containers para armazenamento e compartilhamento de imagens;
- Empresas podem escolher um plano de assinatura para repositórios privados.

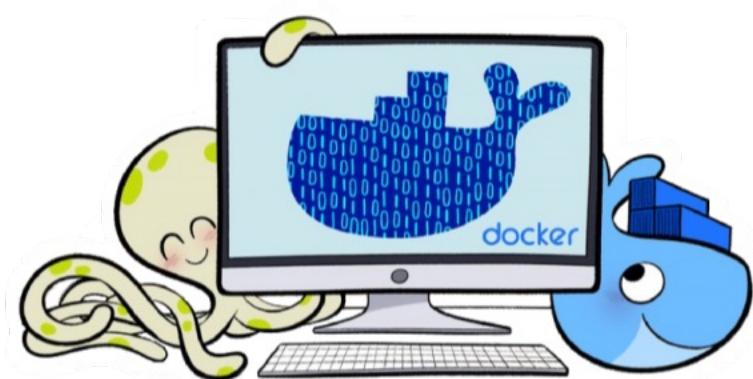


Registry & Docker Hub



Onde e como aprender?

- Pwd – (play with docker) é um playground do docker que permite aos usuários executar comandos do docker no navegador: <https://www.docker.com/playwith-Docker>
- Docker 101 tutorial é um tutorial em um container onde aprende o essencial de docker: <https://www.docker.com/101-tutorial>
- Docker docs é uma documentação rica de conteúdos mantida pelo time do docker: <https://docs.docker.com/getstarted/resources/>



Comandos básicos

- `docker help` – busca comandos docker para te auxiliar
- `docker container ps` – lista os containers (somente os executados)
- `docker image ls` – lista as imagens criadas
- `docker container ps -a` – lista container (parados e executando)
- `docker pull <image>` - baixa uma imagem
- `docker run <image>` - executa uma imagem
- `docker stop <id_container>` - para a execução de um container

Comandos básicos

- `docker search <nome_imagem>` - busca uma imagem no Dockerhub
- `docker build` – constrói uma imagem no Dockerfile
- `docker stats <container>` - informações de uso do hardware
- `docker inspect <container>` - inspeciona o conteúdo estático do container
- `docker image prune` – limpa as imagens pendentes ou não referenciadas
- `docker rmi <id_imagem>` - remove uma imagem do seu host
- `docker rm <id_container>` - remove um container de seu host

Parâmetros úteis

- `-a` - lista todas as imagens ou containers (mesmo parados)
- `-it` - entra em modo iterativo com o container
- `--name` - cria um nome para o container
- `-rm` - cria um container e remove após o uso
- `-p` – define a porta a ser usada
- `-d` - executa em modo background
- `-c` - executa um comando de console específico
- `-t` – cria um nome (tag) para o container

Referências:

- <https://www.docker.com/>
- <https://www.docker.com/play-with-docker>
- <https://docs.docker.com/samples/>
- <https://twitter.com/docker>
- <https://github.com/gomex/docker-para-desenvolvedores>
- <https://nordicapis.com/docker-containers-and-apis-a-briefoverview/>
- <https://docs.docker.com/get-started/overview/>
- <http://neokobo.blogspot.com/2017/03/docker-image.html>
- <https://mirandacontainer.com.br/historia-completa-containers/>

Implementando Integração Contínua

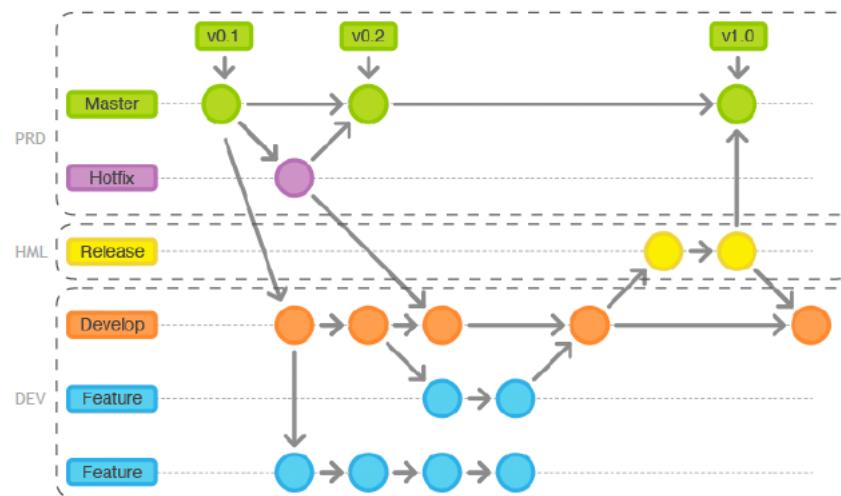
Review ferramentas CI/CD

 Bamboo

Azure DevOps



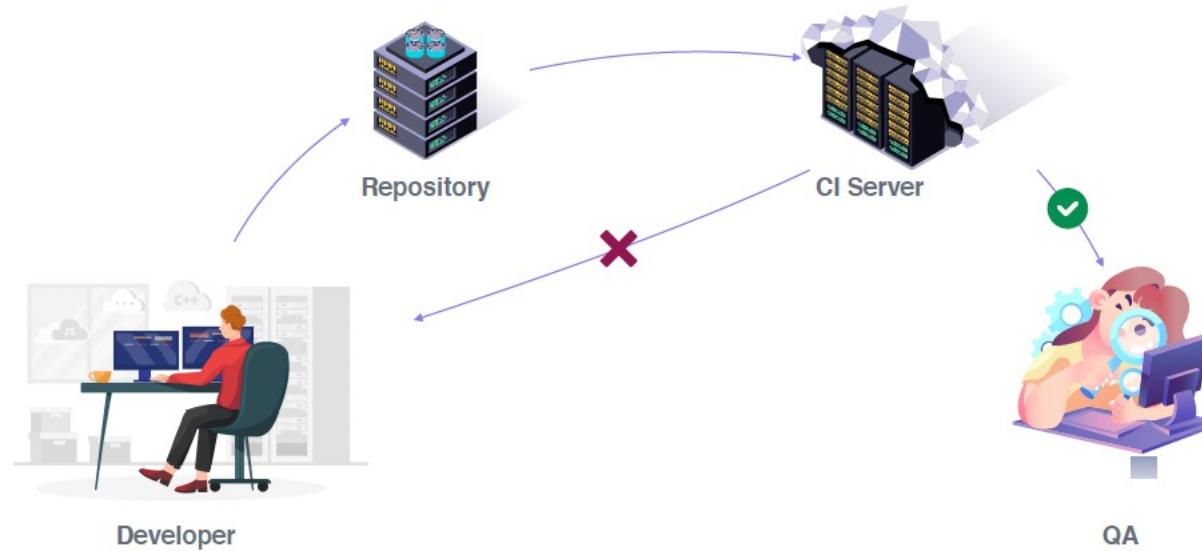
Review ferramentas CI/CD



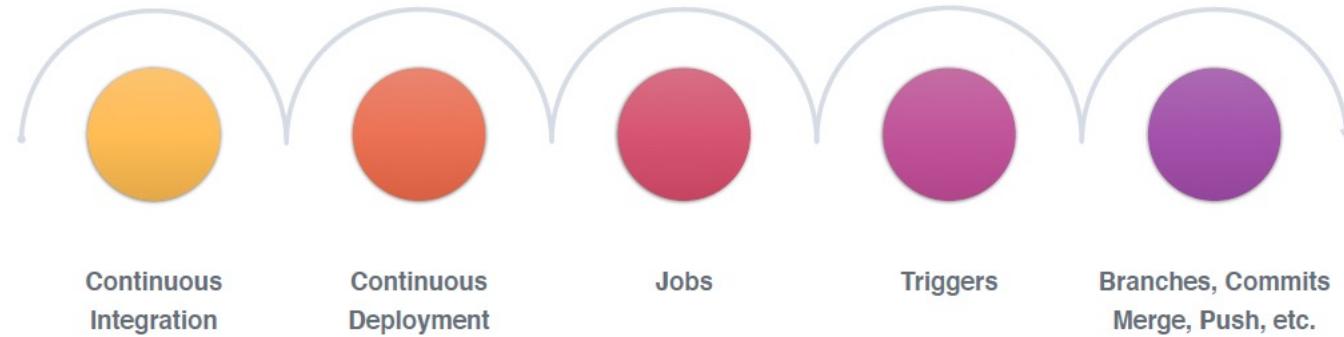
Review ferramentas CI/CD

 Bamboo

Integração contínua com Jenkins



Integração contínua com Jenkins

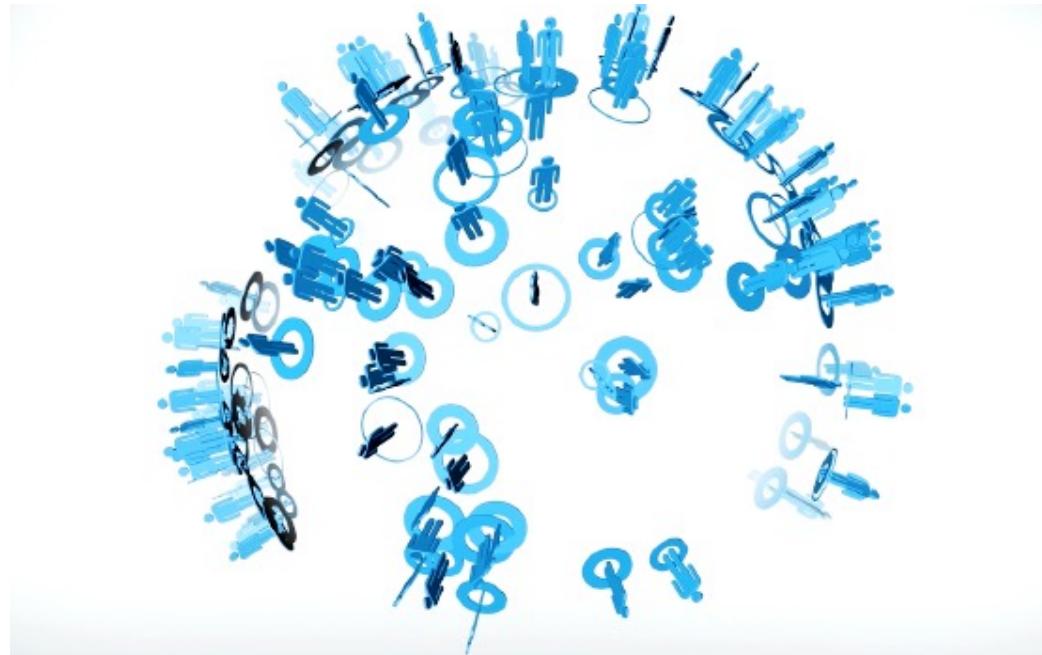


Hands on Jenkins



Plataformas Cloud

Equipes distribuídas



Múltiplas aplicações



Tudo na Nuvem



Google Cloud

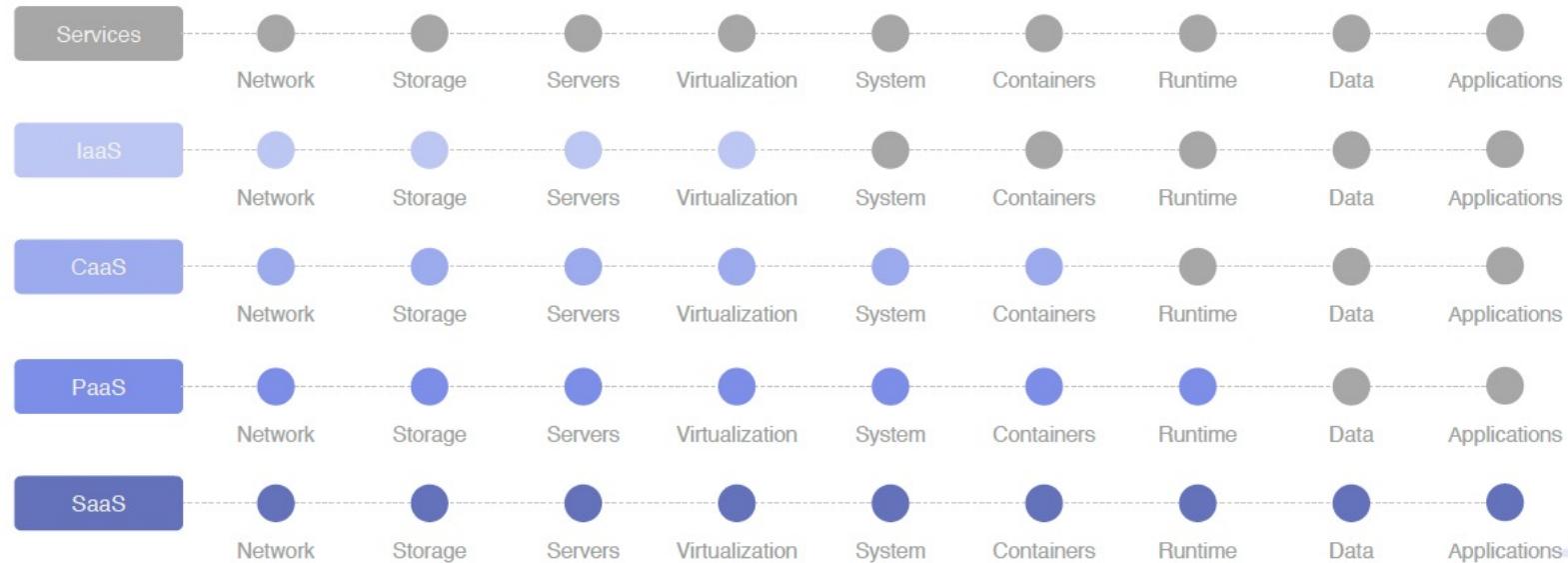


Microsoft
Azure



ORACLE
CLOUD

Serviços na Nuvem



Infrastructure as a Service



- ✓ Rede
- ✓ Armazenamento
- ✓ Servidores
- ✓ Virtualização



Containers as a Service

IaaS

CaaS

PaaS

SaaS

- ✓ IaaS
- ✓ Sistema Operacional
- ✓ Imagens customizadas



Platform as a Service

IaaS

CaaS

PaaS

SaaS

- ✓ IaaS
- ✓ CaaS (pode não utilizar)
- ✓ Runtime de linguagens
- ✓ Foco na lógica de desenvolvimento



Software as a Service

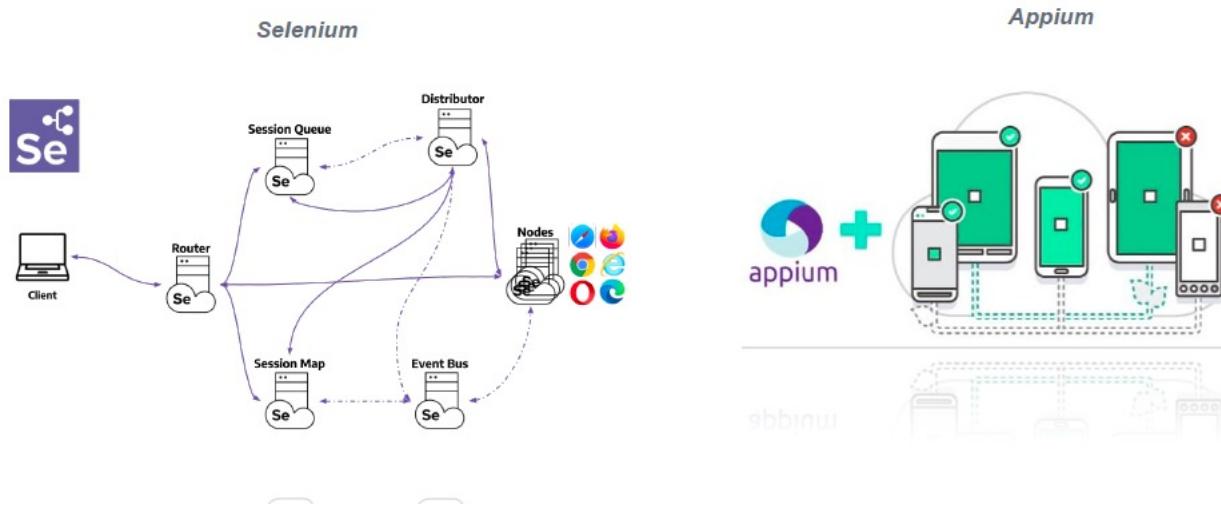


- ✓ PaaS
- ✓ Hospedagem
- ✓ Licenciamento



Testes na Nuvem

Testes na Nuvem



Testes na Nuvem



BrowserStack



SAUCE LABS

Testing at the speed of awesome.

AWS Device Farm



AppCenter