



# Devops

Material de Apoio



E-mail: [contato@gama.academy](mailto:contato@gama.academy)

Website: [gama.academy](http://gama.academy)

# Devops

1ª Edição

**Autoria:** Alice Paixão e Douglas Moraes

**Editora:** Lauren da Silveira Francke

**Consultor de Acessibilidade:** Eduardo Zangrossi Lino

**Consultora de Design:** Juliana Moraes

**Especialista Técnica:** Hendy Almeida

**Equipe Multidisciplinar:**

Fabio Naito

Natália Gomes

Copyright © Gama Academy 2022

Proibida a reprodução e a distribuição sem autorização prévia.

# Sumário

Clique nos links para ir ao capítulo correspondente:

- **Tópico 01**  
[O que é DevOps](#)
- **Tópico 02**  
[Sistemas Operacionais \(Linux\)](#)
- **Tópico 03**  
[Orquestração e Containers](#)
- **Tópico 04**  
[Computação em nuvem](#)
- **Tópico 05**  
[Infraestrutura como código](#)
- **Tópico 06**  
[O que é CI/CD?](#)
- **Tópico 07**  
[O que são pipelines?](#)
- **Tópico 08**  
[Quais as principais ferramentas adotadas junto a metodologia DevOps](#)
- **Tópico 09**  
[Certificações](#)
- **Tópico 10**  
[Github actions](#)
- **Tópico 11**  
[Fechamento](#)



# Abertura



## **Acessibilidade é para todos!**

É com essa frase que começamos uma nova fase na Gama Academy, pensando em um formato mais acessível para nossas apostilas, tornando assim sua jornada conosco muito mais prazerosa e seu aprendizado mais dinâmico.

### **O que você verá de diferente:**

- A começar pelo novo visual dos nossos materiais;
- Trocamos o formato para o vertical, para facilitar a leitura e tornar muito mais fácil e dinâmico para quem usa o celular para estudar;
- O contraste de cores e tamanho das fontes seguem recomendações internacionais de acessibilidade da WCAG;
- Os textos passarão a ser melhor distribuídos para a redução da carga cognitiva necessária para entendê-los, assim passaremos a respeitar ainda mais o tempo de aprendizado de cada um, facilitando para que você possa retornar para o conteúdo que tenha gerado dúvidas com maior facilidade;
- Legendas em todas as imagens do conteúdo para facilitar a identificação;
- Todas as imagens que são necessárias para a devida contextualização sobre o tema, passarão a contar com audiodescrição. É um texto oculto que será lido apenas para pessoas que utilizam tecnologias assistivas, como por exemplo, leitores de tela.

Contamos muito com o feedback de vocês para tornarmos isso um processo de melhoria contínua e nossos materiais atingirem cada vez mais pessoas.



# Objetivos de Aprendizagem

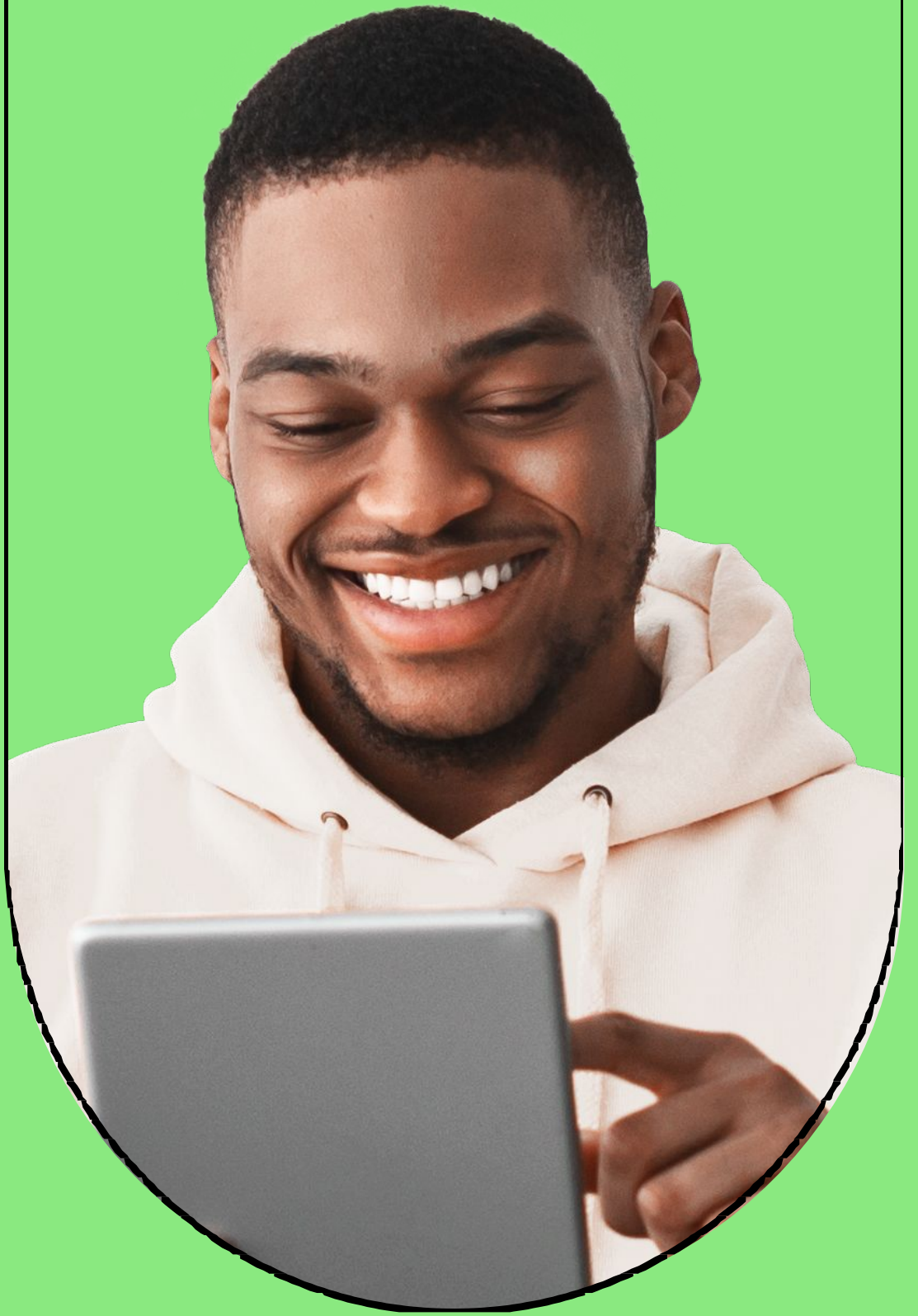
**A partir do estudo desta apostila, você deverá ser capaz de:**

- Compreender a cultura que está presente na maioria das organizações modernas;
- Desempenhar um papel como ator em times que adotam a cultura DevOps;
- Atuar tecnicamente na construção e manutenção de um ciclo de desenvolvimento DevOps;
- Analisar a importância e sugerir alterações em processos repetitivos que possam ser automatizados.

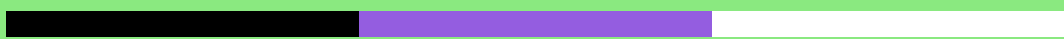
Esta apostila está dividida em tópicos e em cada um deles você irá encontrar uma pequena introdução sobre o tema e uma recapitulação do que você aprendeu.







# Tópico 01



# DevOps



# Introdução: Devops

---

Atualmente, cada vez mais as empresas entendem a importância de adotar os processos e métodos DevOps em seus projetos de desenvolvimento de software, acelerando os os ciclos de entrega e ajudando a aumentar a produtividade. Neste tópico vamos entender um pouco mais sobre Devops.

# DevOps

## O que de fato é Devops?

Basicamente uma metodologia de desenvolvimento de software, atualmente com forte adoção no mercado por conta de agilizar cadência de entregas e tem 100% de foco em desenvolvimento operacional.

Sua origem vem da palavra Dev de development, enquanto o Ops inclui a implantação dos serviços e estratégias em torno.

## Ciclo de desenvolvimento e DevOps



Ciclo de desenvolvimento e DevOps.

[Descrição da Imagem](#)



## Benefícios

Os maiores benefícios da cultura DevOps.

A adoção desta cultura garante vantagens estruturais ou seja a nível de utilização de serviços de Cloud Computing além dos listados abaixo.

- Melhoria no desenvolvimento e infra estrutura;
- Soluções mais estáveis e com otimização de custos;
- Entregas mais dinâmicas;
- Redução de paradas de desenvolvimento;
- Redução de incidentes;
- Redução de riscos operacionais;
- Maior valor agregado ao negócio;

## Qual a missão da metodologia DevOps?

Proporcionar velocidade para inovar. Entregas rápidas para responder a necessidade das pessoas desenvolvedoras e usuários finais.

Confiabilidade para manter uma experiência positiva. Escala para ter poder de crescimento conforme nosso tráfego. Colaboração e Segurança.

**DevOps = software development, IT infrastructure operations, and the intersection between them.**



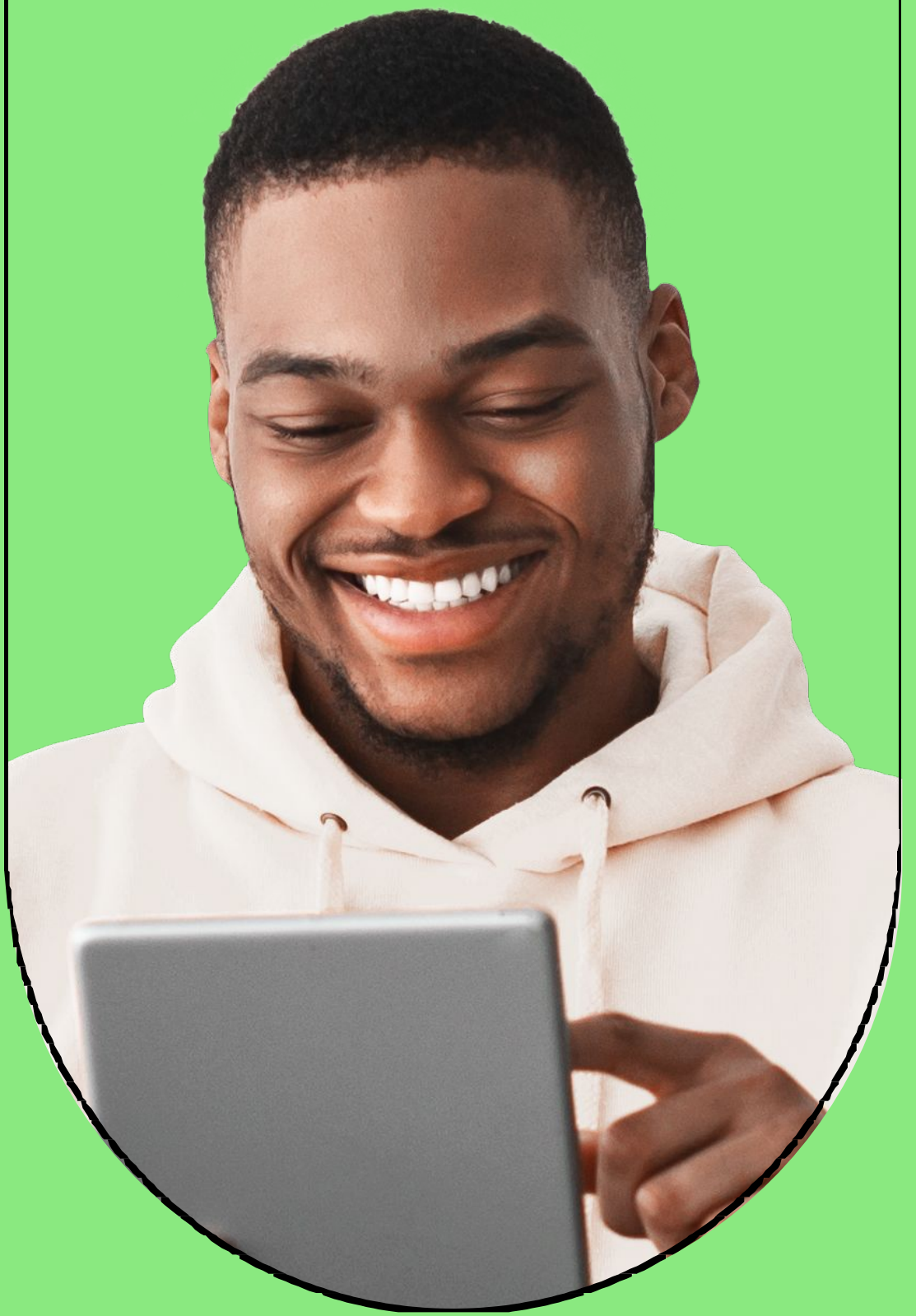


# Resumo do tópico

---

**Neste tópico você aprendeu que:**

1. Devops é uma metodologia de desenvolvimento de software e aprendeu sobre os maiores benefícios da cultura DevOps.



## Tópico 02



# Sistemas Operacionais



# Introdução: Sistemas Operacionais

---

Um sistema operacional é o software responsável pela intermediação entre o hardware e os programas de computador que são executados pelo usuário.

Neste tópico vamos saber um pouco mais sobre o Linux.



# Sistemas Operacionais (Linux)

## Porque falaremos de sistemas operacionais?

A base dos sistemas operacionais que são utilizados para hospedar softwares é [linux](#), nem sempre todas as pessoas desenvolvedoras são familiarizadas com esse sistema operacional, então vale a pena conhecer o site deles. DevOps trabalha muito com ambientes e infraestrutura, então é necessário estar familiarizado com a usabilidade e conceitos básicos de sistemas operacionais.

## Princípios básicos de Sistemas Operacionais

### Conceitos

- Hardware, sistema operacional e sistema computacional
- Máquina virtual, sistema hospedeiro (host) e sistema convidado (guest)
- Monitor de máquina virtual (virtual machine monitor ou hypervisor)
- Container/container e imagem

## Lidando com Scripts e comando no terminal

- [Bash](#)
- [PowerShell](#)





# Resumo do tópico

---

**Neste tópico você aprendeu que:**

1. É necessário conhecer um pouco sobre Linux;
2. DevOps trabalha muito com ambientes e infraestrutura.



## Tópico 03



# Orquestração e Containers



# Introdução: Orquestração e Containers

---

As ferramentas de orquestração de containers são aplicações em nuvem que permitem fazer o gerenciamento de múltiplos contêineres.

A popularidade deste tipo de solução cresceu nos últimos anos, acompanhando o crescimento da prática de containerização de aplicações, microsserviços e adoção de DevOps.



# Definindo orquestração e containers

Imagine um navio que transporta diversas cargas. Como eletrônicos, alimentos, têxteis. Neste navio existem divisórias para cada carga. Assim as cargas são isoladas e não tem contato entre si.

Para desenvolvimento de software utilizamos a mesma estrutura de hierarquia. Temos um ambiente em comum que roda sistemas que não se comunicam entre si. Para isso utilizamos duas tecnologias de **orquestração** com uma ferramenta chamada de **Kubernetes** (apelidada de k-eight's com a sigla K8S) e **containers** com uma ferramenta nomeada de Docker.

## Kubernetes

Kubernetes (K8S) é um produto Open Source utilizado para automatizar a implantação, o dimensionamento e o gerenciamento de aplicativos em contêiner. Ele agrupa contêineres que compõem uma aplicação em unidades lógicas para facilitar o gerenciamento e a descoberta de serviço. O Kubernetes se baseia em 15 anos de experiência na execução de containers em produção no Google, combinado com as melhores ideias e práticas da comunidade.

Entenda o básico e seus componentes através da documentação oficial:

<https://kubernetes.io/pt-br/docs/concepts/overview/>

## Docker

Docker é uma tecnologia de software que fornece contêineres, promovido pela empresa Docker, Inc. O Docker fornece uma camada adicional de abstração e automação de virtualização de nível de sistema operacional. Usa de recursos de um sistema hospedeiro para permitir sistemas convidados independentes executarem sem a necessidade do software do hardware convidado, evitando a sobrecarga de iniciar e criar sistemas convidados.

"Docker enables true independence between applications and infrastructure and developers and IT ops to unlock their potential and creates a model for better collaboration and innovation."



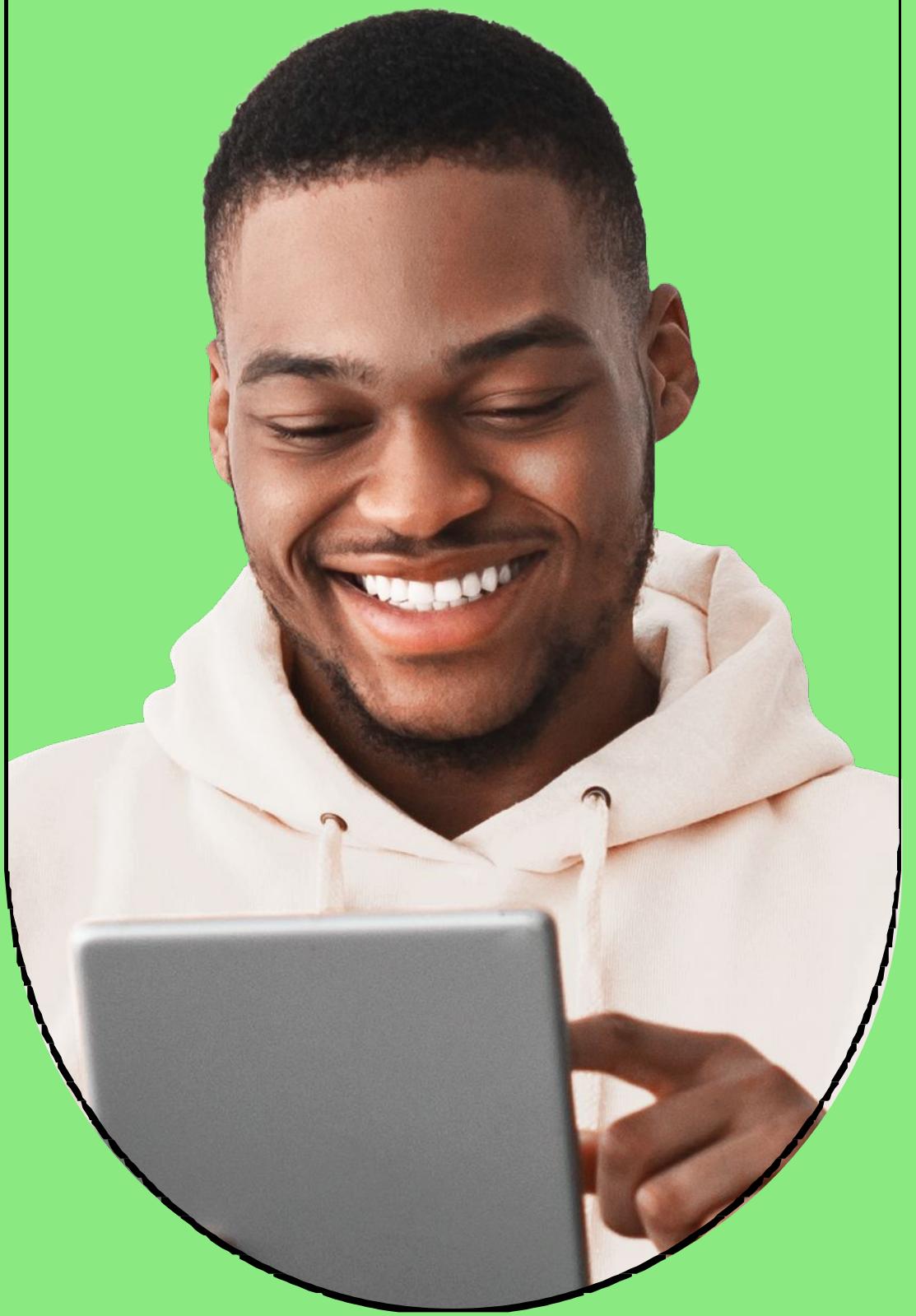




# Resumo do tópico

**Neste tópico você aprendeu sobre:**

1. Orquestração e containers, além de entender o que são Kubernetes e Docker.



## Tópico 04



# Computação em nuvem



# Introdução: Computação em nuvem

---

“Computação em Nuvem”, como o próprio nome sugere, engloba as chamadas nuvens, que são ambientes que possuem recursos (hardware, plataformas de desenvolvimento e/ou serviços) acessados virtualmente e de fácil utilização.

Esses recursos, devido à virtualização, podem ser reconfigurados dinamicamente de modo a se ajustar a uma determinada variável, permitindo, assim, um uso otimizado dos recursos. Esses ambientes são, em geral, explorados através de um modelo pay-per-use.” [VAQUERO et al. 2009]



# Computação em nuvem

A computação em nuvem é a entrega de recursos de tecnologia sob demanda com acesso via internet e com a definição de preço de pagamento conforme a utilização. Ao invés de comprar, ter e manter datacenters e servidores físicos, você pode acessar serviços de tecnologia, como capacidade computacional, armazenamento e bancos de dados, conforme a necessidade, usando um provedor de nuvem como a Amazon Web Services (AWS), Google (GCP), Microsoft (Azure)

Saiba mais:

Quer saber mais sobre a definição de Computação em nuvem para os gigantes da tecnologia? Acesse os links abaixo:

- [Definição AWS](#)
- [Definição Microsoft](#)
- [Definição Cisco](#)

## Principais provedores de Cloud

- Aws
- GCP (Google Cloud Platafform)
- Azure



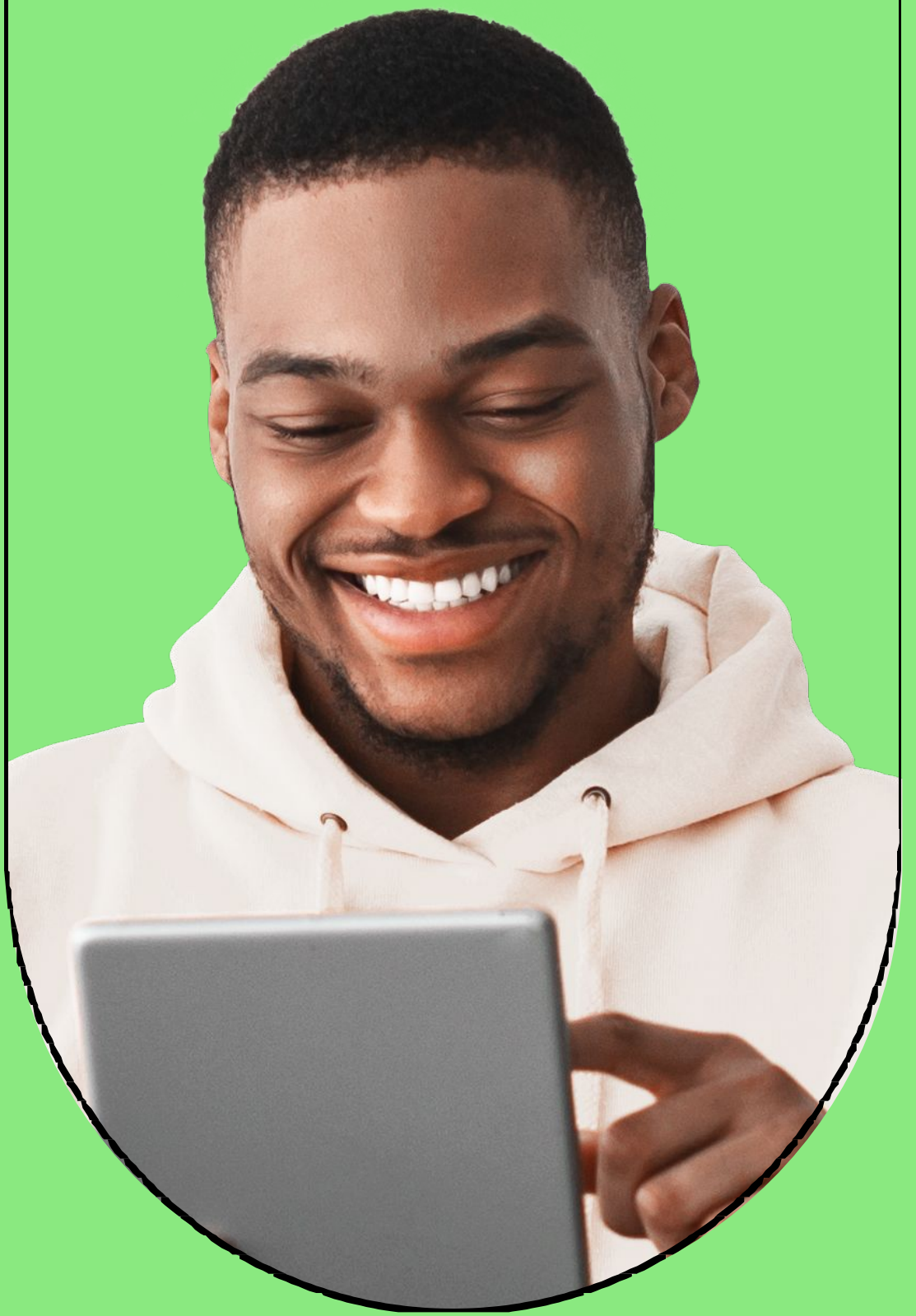
# Resumo do tópico

---

**Neste tópico você aprendeu sobre:**

1. O que é a computação em nuvem e aprendeu os diferentes tipos de provedores Amazon Web Services (AWS), Google (GCP), Microsoft (Azure).





Tópico 05



# Infraestrutura como código



# Introdução: Infraestrutura como código

---

A infraestrutura como código está diretamente ligada com a automação de processos e o tempo economizado.

Trabalhar com desenvolvimento é estar diretamente ligado com o cumprimento de prazos e a ausência de erros. Dessa forma, é algo bastante complicado e muitas vezes feito por uma grande equipe.



# Infraestrutura como código

Segundo a [Redhat](#), infraestrutura como código (IaC) é o gerenciamento e provisionamento da infraestrutura por meio de códigos, em vez de processos manuais. Com a IaC, são criados arquivos de configuração que incluem as especificações da sua infraestrutura, facilitando a edição e a distribuição de configurações. Ela também assegura o provisionamento do mesmo ambiente todas as vezes.

## **Automação em DevOps**

É a adoção de tecnologias que executam tarefas com o mínimo possível de interferência humana nos processos que viabilizam os loops de feedback entre as equipes de desenvolvimento e operações. Dessa forma,

é possível acelerar a implantação de atualizações iterativas em aplicações no ambiente de produção.

## **Gerenciamento de configuração**

O gerenciamento de configuração impede que sejam feitas alterações, tanto grandes quanto pequenas, que não sejam documentadas.

A configuração incorreta é a principal causa dos incidentes de segurança nos ambientes em containers ou orquestrados por Kubernetes.

As ferramentas de gerenciamento de configuração fazem alterações e implantações com mais agilidade e removem a possibilidade de erro humano ao mesmo tempo que tornam o gerenciamento do sistema mais previsível e escalável.

A principal ferramenta de gerenciamento de configuração é o [ansible](#). Veja como funciona através da [documentação oficial](#).

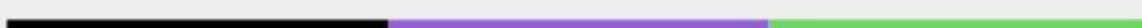
## Provisionamento

O provisionamento é o processo de definição da infraestrutura de TI através de código. Ele também se refere às etapas necessárias para gerenciar o acesso aos dados e recursos e para disponibilizá-los a usuários e sistemas. Acesse a documentação oficial da [RedHat](#) para compreender os tipos de provisionamento existentes em DevOps

## Terraform

O Terraform é um sistema Open Source destinado a provisionar infraestrutura como código ele fornece todo fluxo de implementação de ambientes robustos possui uma CLI própria e consistente que é capaz de gerenciar inúmeros serviços em nuvem. O Terraform codifica APIs de nuvem em arquivos de configuração declarativos. As principais ações do terraform são Write, Plan, Apply.

Compreenda melhor como funciona o Terraform que é o mecanismo mais utilizado mundialmente na comunidade de DevOps através da [documentação oficial](#).





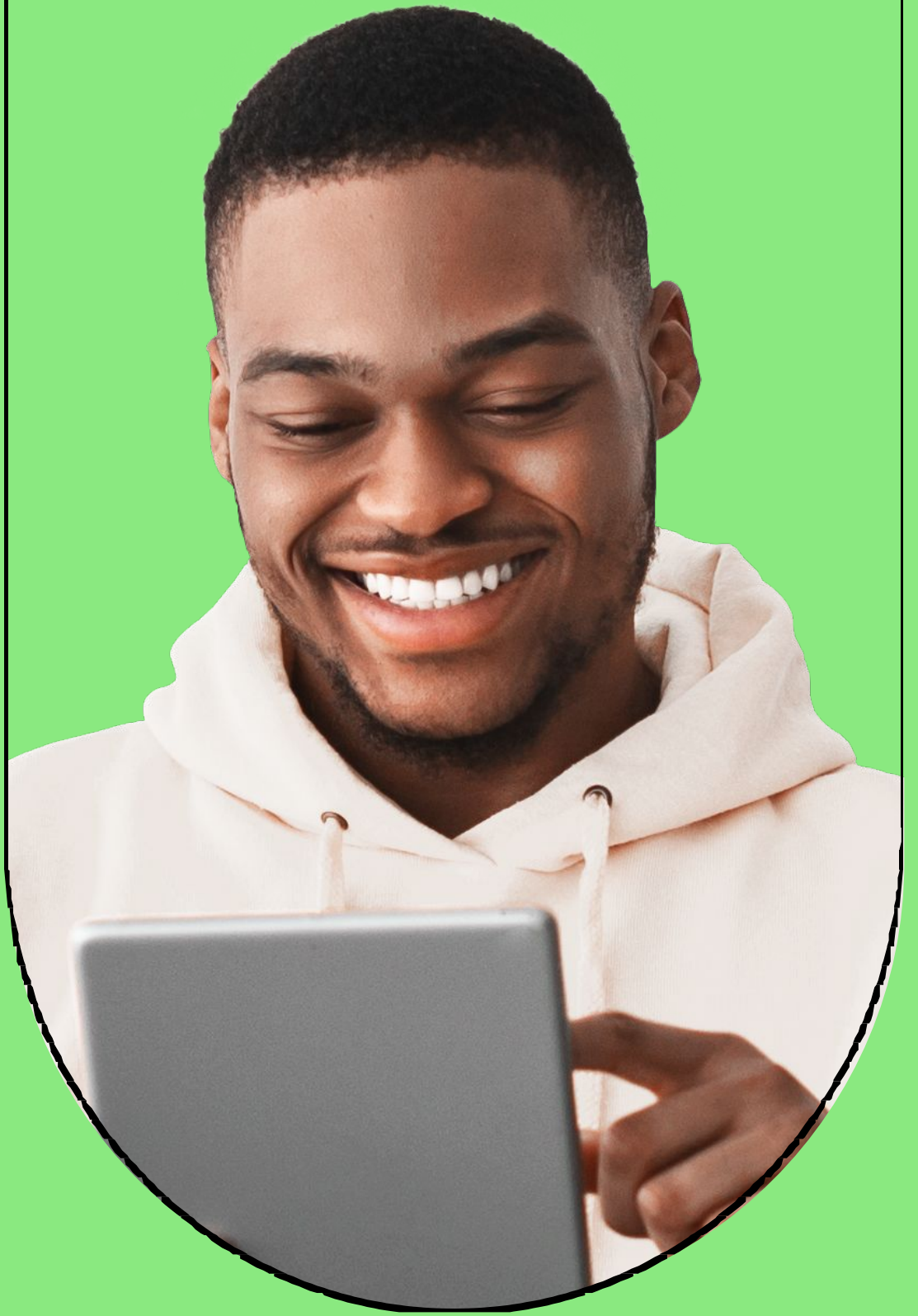
# Resumo do tópico

---

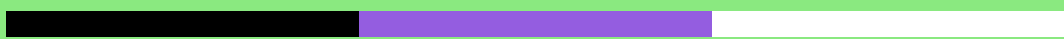
**Neste tópico você aprendeu sobre:**

1. O que é a infraestrutura como código;
2. O que é a Automação em DevOps;
3. O que é Provisionamento e Terraform.





## Tópico 06



# O que é CI/CD?



# Introdução: CID/CD

---

Neste capítulo vamos entender um pouco mais sobre CI (Continuous Integration) e sobre CD (Continuous Deployment) entendendo a nível mais profundo esta solução e compreendendo os ganhos em torno desta solução.

Como mencionado algumas vezes durante este módulo de DevOps a distribuição de código e publicação de projetos vem se tornando cada vez mais dinâmica e entendendo de maneira mais objetiva esta solução podemos ver com mais clareza a importância de deter conhecimentos da cultura DevOps.



A **Integração Contínua (CI)** é a prática onde a equipe de desenvolvimento inicia um ciclo de fusão ou integração de mudança de código. E nela inclui testes automáticos, de sendo eles unitários ou até mesmo integrados com regressão com o claro objetivo de manter a versão em seu melhor funcionamento na versão primária.

A **Entrega Contínua (CD)**, por sua vez trata de garantir a entrega de projetos em ambiente produtivo com escalabilidade e segurança, permitindo a implementação em qualquer momento do dia. Sem a necessidade de iniciar uma rotina de publicação fora do horário comercial.

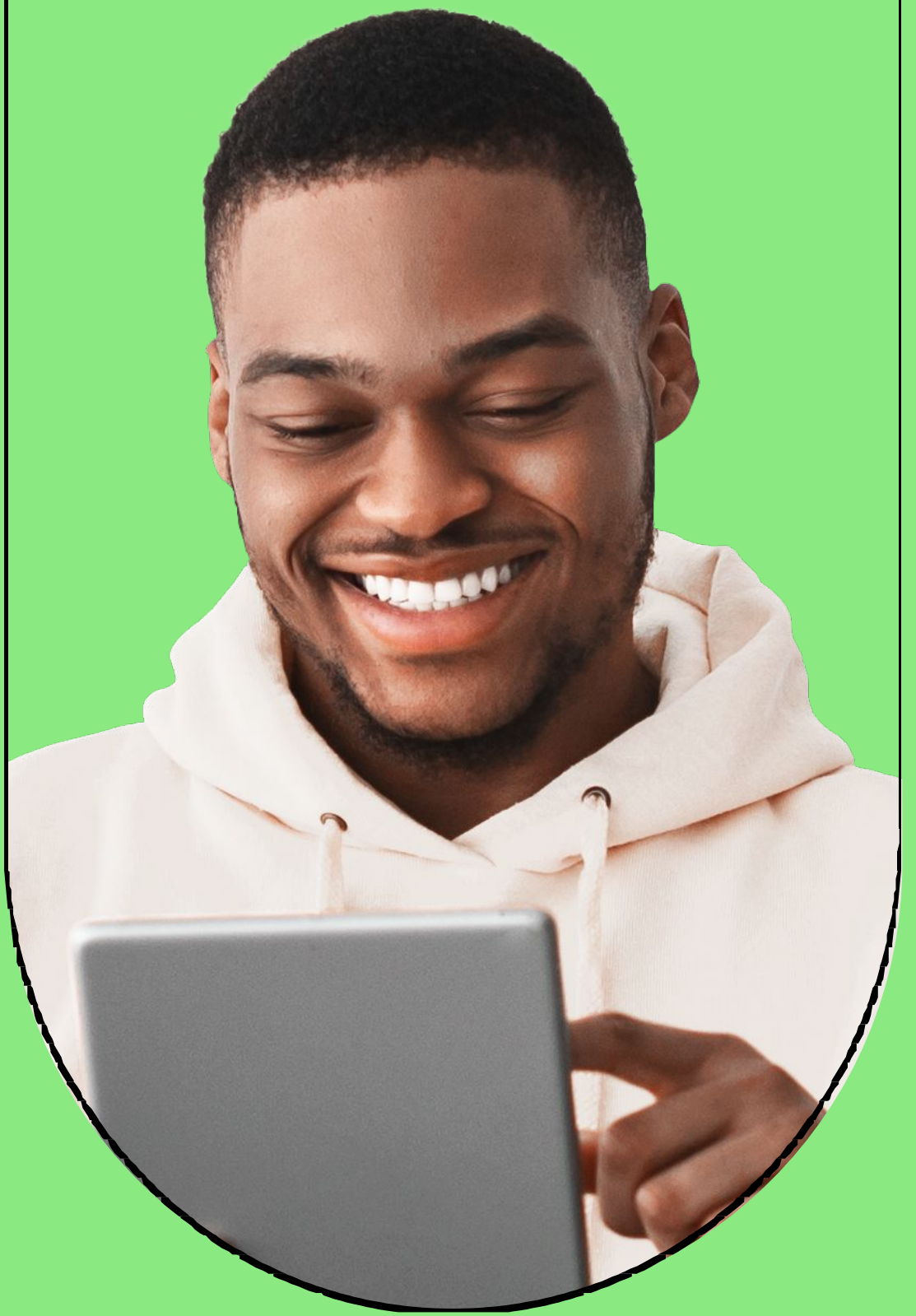


# Resumo do tópico

---

**Neste tópico você aprendeu sobre:**

1. O que é Integração Contínua (CI) e o que é Entrega Contínua (CD).



## Tópico 07



# O que são Pipelines?



# Introdução: O que são Pipelines?

---

O pipeline DevOps é uma forma automatizada de levar o software do controle de versão às mãos do usuário. Com esse recurso, todo o processo pode ser modelado como um mapa de fluxo de valor.

Para tanto, é essencial envolver os processos de:

- Integração Contínua
- Entrega Contínua
- Implantação Contínua
- CI-CD



# Pipelines

Entendendo as Pipelines e como funcionam de maneira estrutural. Nos capítulos anteriores falamos sobre diversas ferramentas que nos auxiliam na construção e gerenciamento de pipelines. E o mais importante como ela nos ajuda.

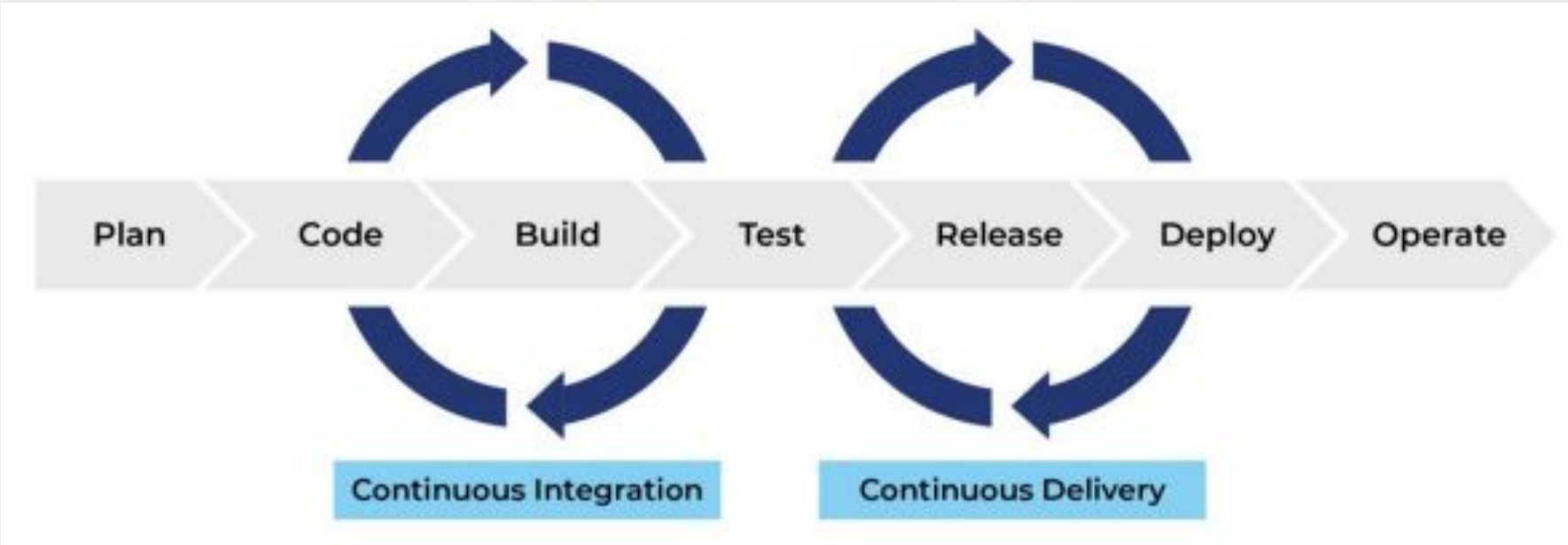
A missão de uma pipeline de uma maneira muito simples é garantir que nosso código está passando pelos devidos processos de validação, montando um fluxo de esteira de CI/CD de maneira manual, ou seja através de configurações implementadas pelo próprio desenvolvedor. Normalmente é um thread e ou um Hook associado a um branch específica de seu repositório a pipeline é responsável, por iniciar o fluxo completo de sua esteira de entrega.

O contexto de pipelines está presente em qualquer papel ligado a tecnologia no contexto atual.

[Saiba mais](#) sobre pipelines com a **documentação**  
**fornecida pela RedHat**



Ciclo de uma pipeline



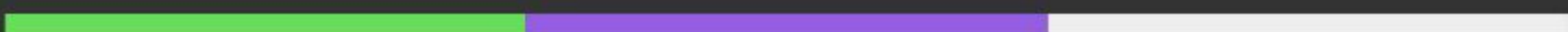
Ciclo de uma pipeline.

[Descrição da Imagem](#)



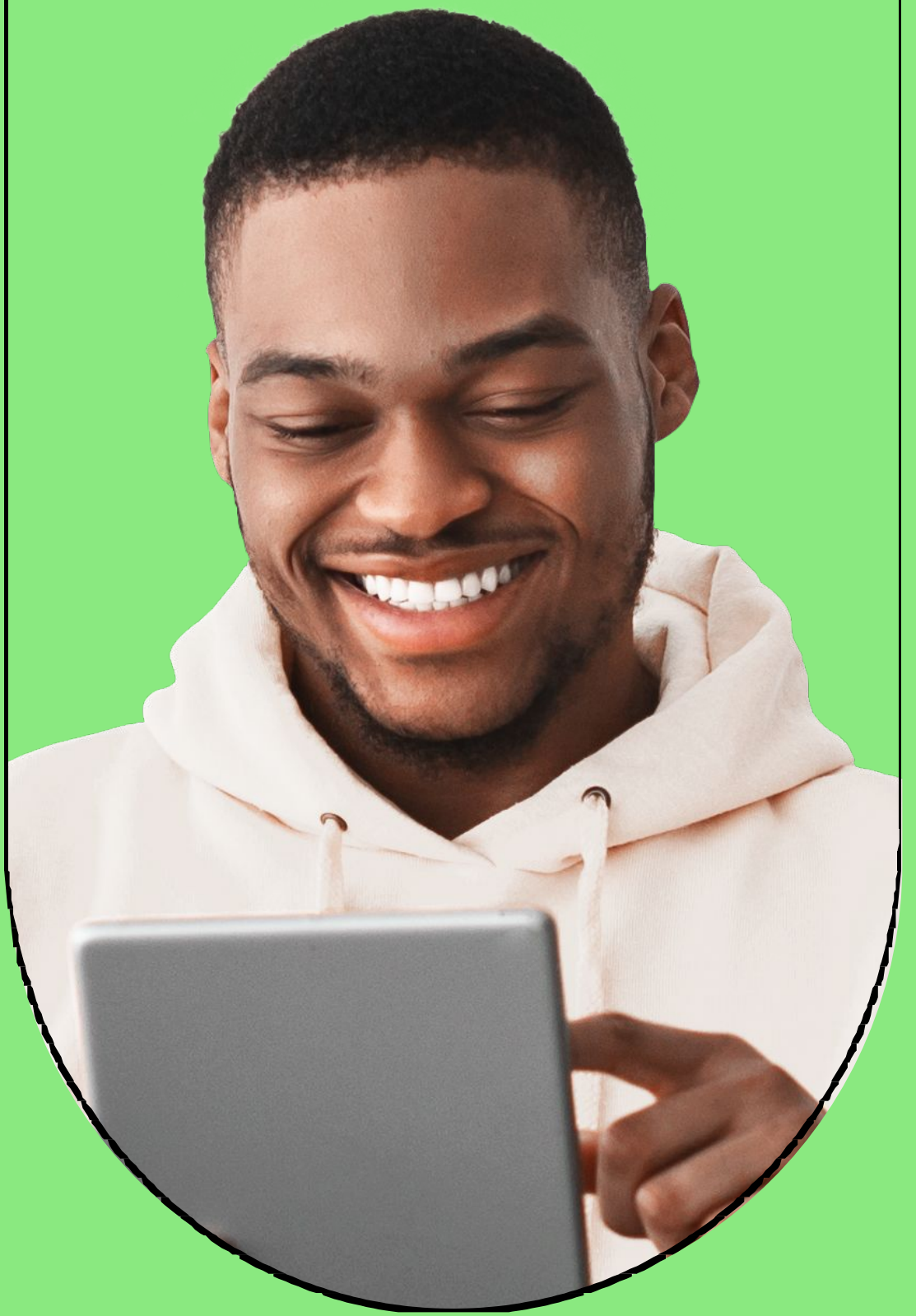


# Resumo do tópico



**Neste tópico você aprendeu sobre:**

1. O que são pipelines e seu ciclo;



## Tópico 08



**Quais as  
principais  
ferramentas?**



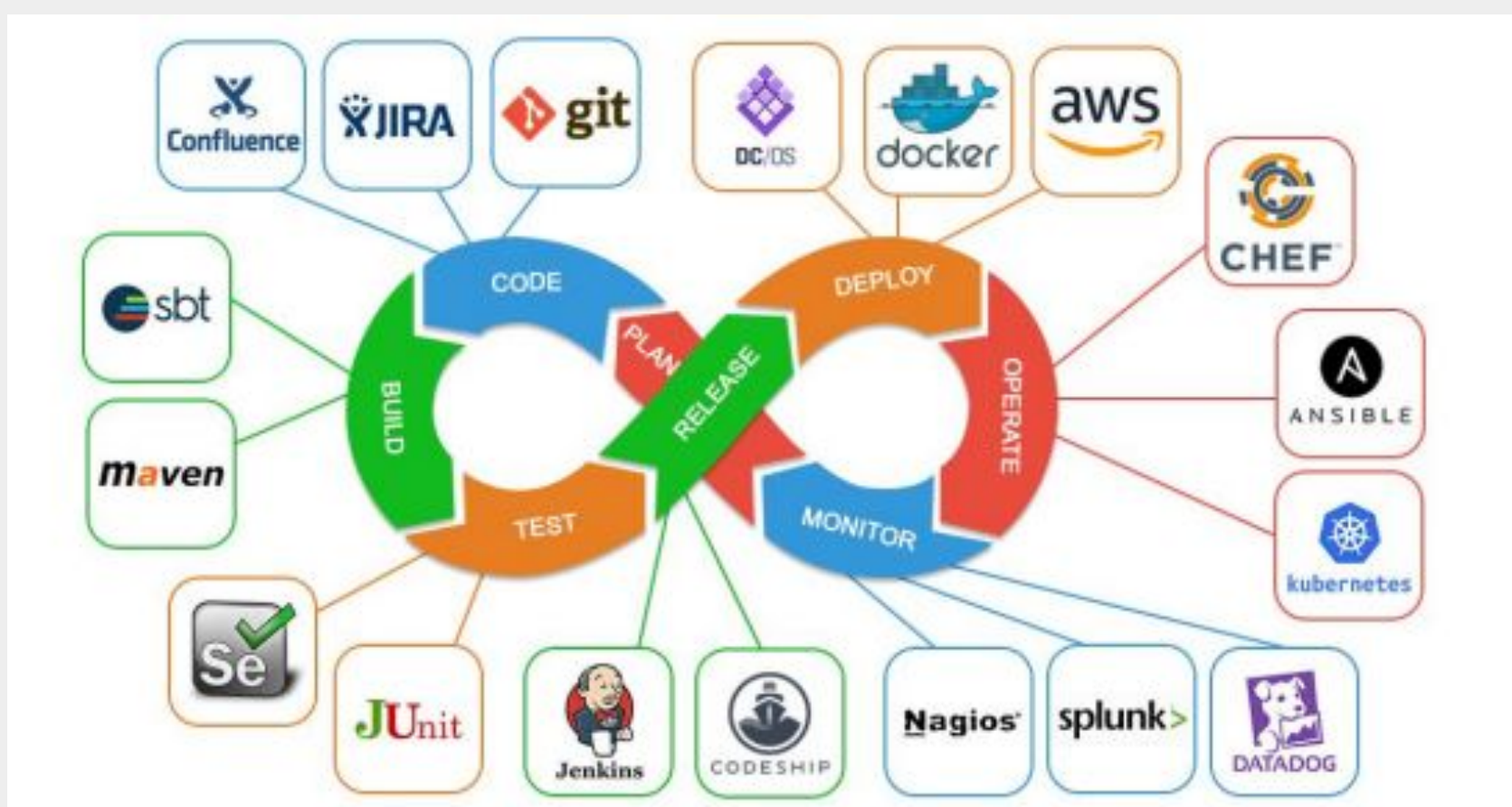
# Introdução: Ferramentas

---

Neste capítulo vamos, falar sobre as principais ferramentas em torno da Cultura DevOps, desde metodologias até mesmo controles e utilização de softwares para auxiliar no versionamento até a publicação e controle de versões em projetos.



# Ferramentas adotadas junto a metodologia DevOps



Ferramentas adotadas junto a metodologia DevOps.

[Descrição da Imagem](#)

## Ferramentas Devops

1. Jenkins
2. Docker
3. Puppet
4. Apache Maven
5. Gradle
6. CircleCI
7. Bamboo
8. Git
9. Github
10. Kubernetes

## Jenkins

Jenkins é um servidor de automação de fluxo, um open source altamente poderoso. Ele automatiza o fluxo em processo de desenvolvimento ou seja a construção de CI/CD e implantação de testes.

### Características e benefícios:

1. Suporta mais de 100 plugins para automação e integração de processos;
2. É um programa de código aberto que funciona de maneira independente (linguagem core é o Java);
3. Altamente permissivo e pode ser integrado facilmente a qualquer projeto.

Saiba mais na documentação oficial, [clcando aqui.](#)

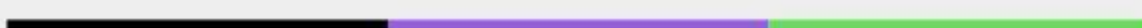
## Docker

Líder em contentores de softwares ou seja orquestra e disponibiliza containers com imagens de sistemas operacionais e runtimes para o desenvolvimento. Auxilia na construção, empacotamento e implantação de softwares.

### Características e benefícios:

1. Utiliza virtualização SO e entrega aplicações no formato de containers;
2. Suporte para diversos servidores de Cloud como AWS e GCP;
3. Facilita a implementação de recursos e reparos futuros.

Saiba mais na documentação oficial, [clcando aqui.](#)





## Puppet

Desenvolvido pela Puppet, este é um Open Source que auxilia no gerenciamento de configuração de softwares. Auxilia no gerenciamento do mais diversos estágios e ciclos do software como o provisionamento de infraestrutura de recursos em nuvem.

### Características e benefícios:

1. Escrito em C++, Ruby e Closure funciona em diversos sistemas operacionais;
2. Ferramenta é orientada por modelos e de uso limitado da linguagem de programação;
3. Ajuda a reduzir erros manuais e dá suporte a sua equipe para dimensionar sua infra estrutura.

Saiba mais na documentação oficial, [clikando aqui](#).

## Apache Maven

Desenvolvido pela Apache Software Foundation, a Maven é uma ferramenta de automação de construção muito eficiente. Como uma ferramenta de DevOps ele auxilia no gerenciamento da construção, documentação e documentação do seu projeto.

### Características e benefícios:

1. Metas pré definida com para realizar tarefas como compilação e empacotamento;
2. Suporte e grande variedade de plugins Maven e bibliotecas Java;
3. Possui licença Apache 2.0.

Saiba mais na documentação oficial, [clikando aqui](#).





## Gradle

Acelera a produtividade no desenvolvimento de software com o Gradle, um do Open Sources do segmento mais utilizado do mercado. Desenvolvido em Java, Kotlin e Groovy é usado para automatizar o desenvolvimento, testes, automatização e implantação com velocidade maior.

### Características e benefícios:

1. Utiliza API em conjunto com uma gama extensa de plugins;
2. Introduz uma DSL com base em Kotlin e Groovy;
3. Possui licença Apache 2.0.

Saiba mais na documentação oficial, [clcando aqui.](#)

## CircleCI

Independente do tamanho da equipe de vai atuar em DevOps, esta ferramenta oferece soluções muito bem elaboradas na entregas em CI/CD. Auxilia na implantação de softwares em alto desempenho em infra local ou na nuvem.

### Características e benefícios:

1. Opera em tubulação com entrega de CI/CD com virtualização;
2. Em caso de falhas ele notifica imediatamente a equipe envolvida;
3. Realiza distribuição automatizada de código;

Saiba mais na documentação oficial, [clcando aqui.](#)



## Bamboo

Desenvolvido pela Atlassian oferece solução em atualização contínua de softwares. Como ferramenta de DevOps a ferramenta oferece suporte para criar planos de construção em diversos estágios e aloca agentes para construções críticas.

### Características e benefícios:

1. Suporta inúmera construções;
2. Utiliza interface de usuário altamente intuitiva;
3. Permite testes automatizados rodando em paralelo com formato de regressão e validação.

Saiba mais na documentação oficial, [clcando aqui.](#)

## Git

Uma das ferramentas se não a mais conhecida no âmbito de desenvolvimento e versionamento de projetos. Esta ferramenta gerencia e distribui código (SCM) é utiliza para rastrear alterações em código fonte, assinar e integrar a repositórios.

### Características e benefícios:

1. Cross Platform, funciona nos principais sistemas operacionais do mercado;
2. Possui licença GPL v2;
3. Curva de aprendizado consideravelmente baixa.

Saiba mais na documentação oficial, [clcando aqui.](#)



## Github

Grande adesão no mercado deste versionador e gerenciador de código fonte e tudo distribuído pelo Git. Esta é a maior e mais avançada plataforma de desenvolvimento do mundo. Atualmente é utilizado por mais de 56 M de usuários, conta com diversos tipos de serviços integrados.

### Características e benefícios:

1. Utiliza virtualização SO e entrega aplicações no formato de containers;
2. Suporte para diversos servidores de Cloud como AWS e GCP;
3. Facilita a implementação de recursos e reparos futuros.

Saiba mais na documentação oficial, [clcando aqui.](#)

## Kubernetes

Líder em contentores de softwares ou seja orquestra e disponibiliza containers com imagens de sistemas operacionais e runtimes para o desenvolvimento. Auxilia na construção, empacotamento e implantação de softwares.

### Características e benefícios:

1. Demonstra melhor controle de acesso;
2. Oferece gerenciamento de tarefas;
3. Cross platform e oferece soluções como Github desktop.

Saiba mais na documentação oficial, [clcando aqui.](#)



## GitOps

GitOps é uma stack nova chamada de futuro da automação de infraestrutura. É um conjunto de práticas para gerenciar configurações de infraestrutura e aplicação usando o Git, que é um sistema de controle de versão open source. O GitOps funciona usando o Git como a única fonte de informações para infraestrutura e aplicações declarativas.

Saiba mais sobre essa tendência já adotada por grande parcela do mercado visitando o [website do manifesto](#).





# Resumo do tópico

---

**Neste tópico você aprendeu sobre:**

1. O que é Cultura DevOps,
2. O que são metodologias, controles e utilização de softwares para
3. auxiliar no versionamento;
4. O que é publicação e controle de versões em projetos.



## Tópico 9



# Certificações



# Introdução: Certificações

---

A busca por certificações DevOps cresce cada dia mais, e motivos para isso não faltam. A crescente adesão a esse tipo de cultura dentro das empresas faz com que se torne uma necessidade a procura por tais cursos e certificações.





# Certificações

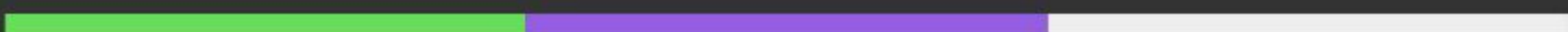
As certificações são as credenciais que reconhecem e validam o conhecimento e a experiência de uma pessoa. Uma certificação pode ajudar os profissionais a provar para os empregadores que desejam permanecer no emprego, e mostra sua dedicação em melhorar seu conjunto de habilidades e conhecimentos, o que implica em benefícios também para a empresa.

Abaixo está listado as certificações oficiais mais amadas pela comunidade de DevOps e mercado de tecnologia e o guia para emití-las:

- [AWS Certification](#)
- [HashiCorp Infrastructure Automation Certification](#)
- [Certificação do Google Cloud](#)
- [Certificações da Microsoft](#)



# Resumo do tópico



**Neste tópico você aprendeu sobre:**

1. O que é a certificação Devops e conheceu o guia para emití-las.



## Tópico 10



# Github actions



# Introdução: Tópico 10

---

GitHub Actions é uma plataforma de integração contínua e entrega contínua (CI/CD) que permite automatizar a sua compilação, testar e pipeline de implantação. É possível criar fluxos de trabalho que criam e testam cada pull request no seu repositório, ou implantar pull requests mesclados em produção.



# Github actions

Por fim, não poderíamos deixar de falar de uma das mais poderosas e mais utilizadas ferramentas de CI/CD.

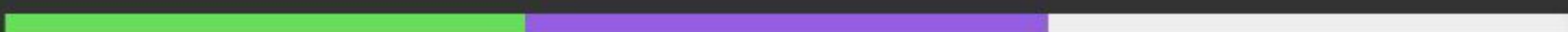
Sim, o GitHub Actions é um poderoso aliado para implementação de pipelines integrado diretamente ao seu repositório.

Com esta ferramenta é possível gerenciar diversos ambientes e suas entregas e tudo isso a partir de branches, as configurações desta solução ficam vinculadas ao seu projeto em formato de arquivo .YAML.

Saiba mais na documentação oficial, [clcando aqui.](#)



# Resumo do tópico



**Neste tópico você aprendeu sobre:**

1. O que é Github actions e como utilizá-lo.



## Tópico 11



# Fechamento





# Fechamento

## **Sobre o nosso aprendizado.**

Agora que entendemos conceitos e funcionalidades de extrema importância em Devops, agora é hora de avançar para o próximo nível.

Sem sombra de dúvidas, este nível de conhecimento vai te trazer maior visibilidade no mercado, por tanto, não deixe os estudos pararem por aqui.

## **Referência Bibliográfica**

- O projeto fênix, Engenharia de Confiabilidade do Google,
- Manual de DevOps, Microserviços Prontos Para a
- Produção, Código limpo, The Manager's Path,
- Descomplicando o Docker, Containers Com Docker - Do
- Desenvolvimento À Produção

