

Guia de Ferramentas Devops

Janeiro/2023

Objetivo do guia de ferramentas DevOps

E aí, blz?

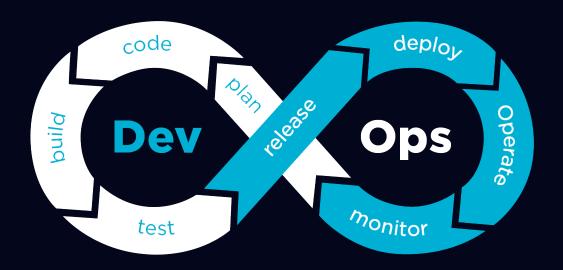
Eu espero que esse guia te ajude a entender quais opções de ferramentas você tem no mercado pra cada etapa.

Mas eu preciso te avisar que aqui eu não vou levantar nenhuma bandeira de qual é a melhor ferramenta e você já vai entender o porquê disso.

Minha intenção é que você fique à vontade para escolher as ferramentas que mais se adaptam ao seu cenário, mas não esqueça de sempre pesquisar o que o mercado mais está utilizando pra não se distanciar muito dele.

O que é DevOps?

Muita gente acredita que DevOps se resume a um cargo específico ou até mesmo a um conjunto de ferramentas. Mas isso é um grande erro. DevOps é muito mais do que isso. É uma cultura corporativa que, através de práticas e ferramentas específicas, além de uma filosofia importante, tem como objetivo automatizar e integrar os processos entre a produção de um software e as equipes de TI.



O fluxo DevOps consiste em oito fases, mas é importante lembrar que o símbolo do infinito representa a colaboração e melhoria contínua, repetindo as fases quantas vezes for necessário.

Ao longo de cada fase, as equipes colaboram e se comunicam para manter o alinhamento, a velocidade e a qualidade do software.

Como escolher uma ferramenta?

Hoje, com a quantidade de ferramentas que temos no mercado, é realmente muito difícil decidir qual adotar.

Por isso, antes de te dar essas opções, quero te apresentar alguns passos que vão te ajudar a eleger uma ferramenta candidata:

Passo 01: Levantar os requisitos da empresa, do negócio e das equipes

Não adianta você escolher uma ferramenta sem antes entender quais recursos você precisa.

Então, levante quais os recursos que você precisa; até quanto pode gastar com ferramentas; verifique se existe algum conhecimento preliminar da sua equipe em alguma ferramenta; se vai ser necessário um suporte corporativo; verifique eventuais parcerias que a sua empresa já possua... Tudo isso precisa estar alinhado. Melhor uma ferramenta mais simples que te atenda do que a mais completa que vai acabar gerando confusão.

Passo 02: Analisar adesão da ferramenta na comunidade e no mercado Esse ponto é muito importante!!

Você deve sempre evitar ferramentas que não tem muita adesão nas comunidades e no mercado.

Normalmente, as soluções não são bem documentadas, e conseguir pessoas capacitadas pode ser um desafio. E ainda pode acontecer da ferramenta ser descontinuada.

Uma boa dica é ver a frequência de pull requests que o projeto tem no GitHub; verificar a quantidade de issues abertas; analisar as empresas que adotam; e perguntar nos grupos de comunidade. Tudo isso vai te ajudar a levantar as melhores opções.

Passo 03: Eleger no mínimo 3 candidatos e montar prova de conceito

Agora que você tem mais conhecimento sobre as opções, chegou a hora de eleger, no mínimo, 3 opções e trabalhar em uma prova de conceito usando as opções.

Prova de conceito nada mais é do que a criação de um cenário pra teste de uma hipótese ou ferramenta. Então quanto mais próximo a sua realidade, melhor é a prova de conceito.

Conforme for utilizando a ferramenta na prova de conceito, analise a complexidade, a disponibilidade de documentação e do suporte, além da adesão das pessoas envolvidas no processo de cada ferramenta.

ATENÇÃO! Seguir esses passos pode ser a diferença entre garantir o sucesso ou fracasso da adoção de uma ferramenta no seu processo DevOps. Então, não ignore eles e nem pule etapas, pois isso PRECISA estar no seu planejamento.

Etapas e Ferramentas

Para cada fase do fluxo DevOps existe uma série de ferramentas fantásticas para atender as demandas do nosso dia a dia de trabalho.

Apenas para facilitar a compreensão do momento em que cada ferramenta é utilizada, catalogamos as ferramentas de acordo com as fases do fluxo DevOps.

Planejamento

Jira Software

O Jira é uma ferramenta criada pela Atlassian pra gestão de projetos de software. Ela dá suporte ao uso de metodologias ágeis com scrum e Kanban e possui recursos de automação de fluxo, além de ter integração com outras ferramentas DevOps do mercado como o Jenkins, GitHub e Slack. O Jira pode ser utilizado utilizando como serviço na núvem ou pode ser instalado no seu data center.

Trello

O Trello também é uma ferramenta criada pela Atlassian, mas diferente do Jira, o Trello é um gestor mais simples, focado em fluxo de execução de tarefas. Ele é recomendando pra equipes menores e que não precisam de muitas integrações.

G GitHub

O GitHub pertence a Microsoft e é mais conhecido pelo serviço de gerenciamento de código, porém ele também possui um gerenciador de tarefas e issues pra gerenciar o desenvolvimento de projeto. Ele pode ser usado de forma gratuita e é uma excelente opção se você já utiliza o ecossistema do GitHub.



O GitLab pertence a própria GitLab e é um concorrente direto do GitHub, pois ele possui serviços de gerenciamento de código, pipelines e gerenciamento de projetos. O GitLab pode ser utilizado como serviço ou também instalado no seu data center.



O Slack não é exatamente uma ferramenta de gestão de projetos e sim uma ferramenta de comunicação. Ele é utilizado pra comunicação interna e muitas vezes integrado com os serviços de gerenciamento de projeto e monitoramento pra alertar e comunicar eventos para as equipes.

Codificação



O Git é a ferramenta de gerenciamento de código mais utilizada no mercado e tem como foco o desenvolvimento colaborativo.

G GitHub

O GitHub é a plataforma de hospedagem de código mais conhecida e utilizada no mercado, com ele você consegue não só gerenciar código, mas também utilizar o GitHub Actions pra criar automações e pipelines CI/CD.

Bitbucket

O Bitbucket é uma solução concorrente ao GitHub que pertence a Atlassian e também possui criação de pipelines.



GitLab

O GitLab é o concorrente direto do GitHub e tem crescido cada vez mais no mercado. Ele também possui recursos de automação e pipelines de CI/CD e pode ser usado como serviço de nuvem ou instalado no seu ambiente.

♥ VS Code

O VSCode é um editor de texto com possibilidades de uso de plugins. Hoje você pode utilizar ele pra programar em diversas tecnologias diferentes, além de usar pra Kubernetes, Terraform, Docker e outros.



No processo de build, uma das ferramentas de extrema importância são os gerenciadores de pacotes. Com eles, é possível gerenciar as dependências do projeto relacionadas a bibliotecas e frameworks. E você também pode utilizar os gerenciadores pra criar e distribuir suas próprias bibliotecas e frameworks.

> Cada linguagem → Maven => Java tem o seu gerenciador de pacote, aqui vão alguns exemplos:

NPM => NodeJS Nuget => .NET Pip => Python



O Docker é hoje o principal container runtime utilizado no mercado. Ele pertence a Docker e possui todo um ferramental pra criação de imagens e execução de containers.



O Podman é o container engine concorrente ao Docker e pertence a Red Hat. Mas ele possui o conceito de Pod que é inexistente no Docker.

container d

O ContainerD é o container runtime originalmente criado pela própria Docker e depois foi doado para a Cloud Native Computing Foundation. Ele é uma solução extremamente leve pra executar containers e é o principal container runtime utilizado pra executar containers no Kubernetes.

Teste

Um dos pontos mais impotantes na área de teste é o teste unitário, ou teste de unidade. Esse teste visa testar pequenas unidades do código, geralmente uma função ou tarefas do negócio. Existem diversas opções de ferramentas e frameworks de testes pra cada linguagem,

> abaixo segue algumas → Junit => Java opções e as linguagens que fazem parte:

NUnit => .NET Mocha => NodeJS



O Selenium é um conjunto de ferramentas criado pra criar automações para web browsers. Com ele você consegue criar testes funcionais e testes de regressão. Ele possui suporte pra diversas linguagens e é extremamente utilizado no mercado.



O Jenkins é uma ferramenta de pipeline open source escrito em Java. Ele é uma das ferramentas mais maduras e utilizadas no mercado.



O GitHub Actions é a ferramenta de pipeline do GitHub. Pertence a Microsoft e é bastante utilizado no mercado.



O GitLab também possui sua ferramenta de pipeline e concorre fortemente com o GitHub.

Deploy



O Terraform é uma das ferramentas de infraestrutura como código mais utilizada no mercado. Ela pertence a Hashicorp e é utilizada pra provisionamento de recursos em cloud providers, ou seja, ele tem suporte a multiplas clouds.



O Ansible é uma ferramenta de infraestrutura como código criada pela Red Hat e diferente do Terraform, ele é focado em gerenciamento de configuração. Você vai utilizar ele pra configurar a sua infraestrutura criada pelo Terraform.

Operação

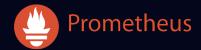


O Kubernetes é o principal orquestrador de containers do mercado. Com ele você consegue gerenciar os seus processos em containers e garantir escalabilidade, resiliência e uma gestão unificada das suas aplicações.



Nomad - um orquestrador de tarefas focado em simplicidade e flexibilidade, capaz de rodar diversos tipos de tarefas e serviços que estão sempre ativos a trabalhos em batch, mas não tão completo quanto o Kubernetes.

Monitoramento



O Prometheus é uma das ferramentas de monitoramento e coleta de métricas mais utilizada no mercado. Foi criado pela SoundCloud e é OpenSource. Você pode utilizar ele em ambiente onpremisse ou dentro do seu cluster Kubernetes.



O Grafana é a ferramenta de criação de Dashboards criada pela Grafana Labs. Ela dá suporte a diversas fontes de dados e pode inclusive, utilizar várias fontes ao mesmo tempo.

Como a Jornada DevOps de Elite pode te ajudar

Como você pode ver, a cultura DevOps abrange um universo de conceitos e ferramentas que dificultam o aprendizado autodidata. Mas com uma orientação correta, fica muito mais simples e rápido.

Por isso, não deixe de **assistir TODAS as 5 aulas da Jornada DevOps de Elite** (principalmente a útlima).

Faça os desafios que serão propostos, poste as suas badges no LinkedIn e tire todas as suas dúvidas na comunidade do Discord para aproveitar ao máximo toda a experiência que preparamos pra você!

Bons estudos! Nos vemos em breve!

Fabrício Veronez