



DevOps Jenkins

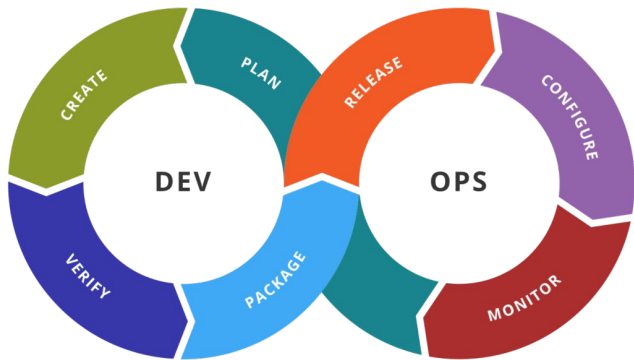
Luciano Kelvin
@Lucianokelvin

Sumário

- O que é integração contínua
- Por que usar integração contínua
- Ferramentas de integração contínua
- Jenkins
 - O que é um job
 - Plugins
 - Variáveis de ambiente
 - Emails com Jenkins
 - API do Jenkins
 - Execução de testes
 - Pipeline

O que é integração contínua

A integração contínua é uma prática de desenvolvimento de software de DevOps em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central. Depois disso, criações e testes são executados. Geralmente, a integração contínua se refere ao estágio de criação ou integração do processo de lançamento de software, além de originar um componente de automação (ex.: uma CI ou serviço de criação) e um componente cultural (ex.: aprender a integrar com frequência)

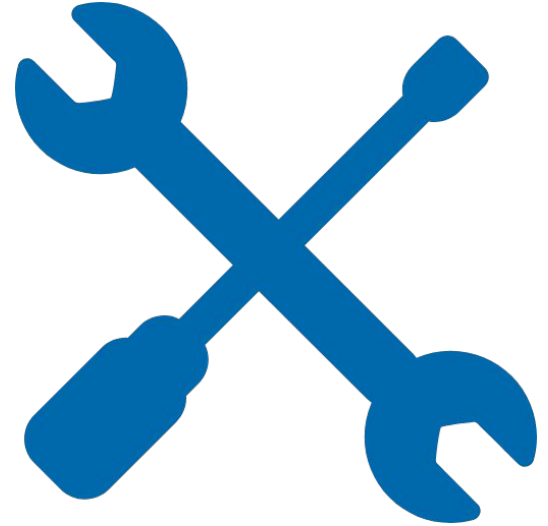


Por que usar integração contínua

Essa abordagem permite que times de desenvolvimento de software tenham **feedback rápido** sobre as mudanças que estão fazendo em uma aplicação específica, além de ser uma maneira mais barata para correção de problemas quando estes são encontrados.

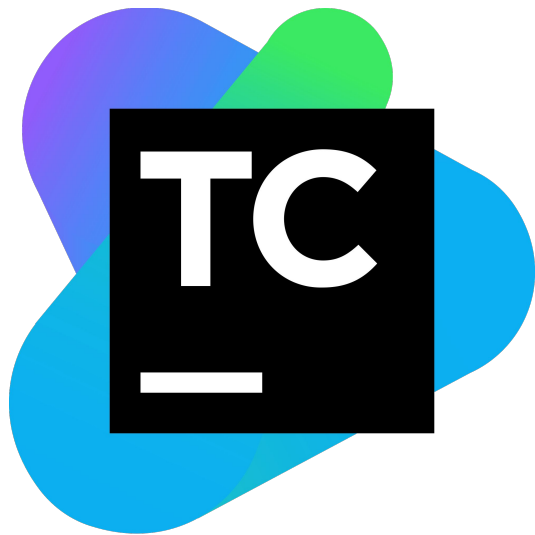
Ferramentas

- Teamcity
- Bamboo
- Travis CI
- Jenkins



TeamCity

- Desenvolvido pela JetBrains
- Fácil de configurar
- Fácil de utilizar
- Não é open source



Bamboo

- Integração com o JIRA
- Automação de testes
- Desenvolvido pela Atlassian



Travis CI

- Configuração é fácil
- Não precisa de um servidor dedicado
- Plano grátis para projetos open sources



Travis CI


Jenkins

- **É GRÁTIS**
- Customização
- Sistemas de plugins
- Controle total do sistema
- É open source



Como instalar ? (no Ubuntu)

- `wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -`
- `sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'`
- `sudo apt update`
- `sudo apt install jenkins`
- `systemctl status jenkins`
- `sudo ufw allow 8080`
- `sudo ufw status`

 [New Item](#)
 [People](#)
 [Build History](#)
 [Manage Jenkins](#)
 [My Views](#)
 [Credentials](#)
 [Lockable Resources](#)
 [New View](#)

Build Queue

No builds in the queue.

Build Executor Status



1 Idle

2 Idle

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		Freestyle project	29 sec - #1	N/A	41 ms	
		Github Org project	N/A	20 sec - log	2.9 sec	
		Multibranch	N/A	N/A	N/A	
		Other	N/A	N/A	N/A	

Icon: [S](#) [M](#) [L](#)

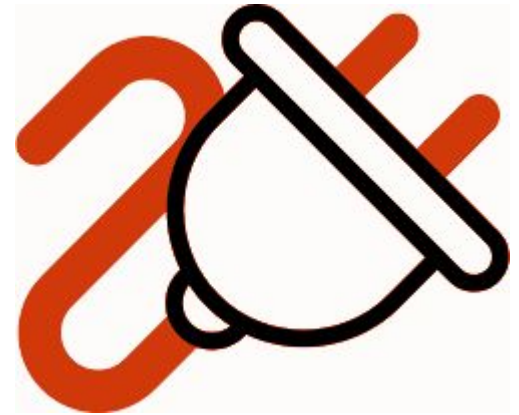
[Legend](#)
 [RSS for all](#)
 [RSS for failures](#)
 [RSS for just latest builds](#)

O que é um job ?

Um job pode ser visto como uma parte do seu processo de build, ele pode envolver compilar o código, rodar baterias de teste, analisar o código e etc.

Plugins

Os plug-ins são o principal meio de melhorar as funcionalidades de um ambiente Jenkins para atender às necessidades específicas da organização ou do usuário. Existem mais de mil plugins diferentes que podem ser instalados no Jenkins e integrar várias ferramentas de criação, servidores na nuvem, ferramentas de análise e muito mais.



Variáveis de ambientes

O Jenkins disponibiliza variáveis de ambiente para que facilitar a personalização dos nossos jobs.

Enviando email com o jenkins

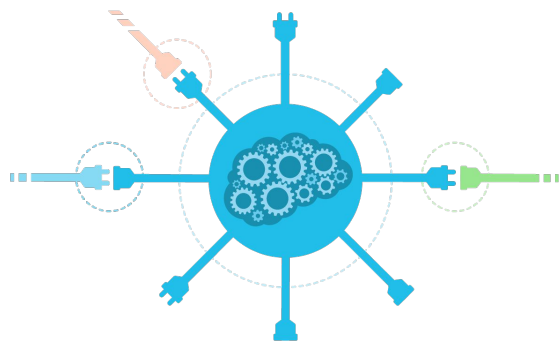
Caso haja a necessidade de enviar um email, o jenkins oferece opções de envio de email, basta só configurar no Job.



API do Jenkins

Jenkins oferece uma API para facilitar a execução de algumas tarefas e consumir alguns dados do servidor jenkins. Algumas das opções que a API do Jenkins oferece são:

- Obter o resultado dos testes executados no build
- Pegar o resultado dos últimos builds
- Executar um build remotamente
- etc.



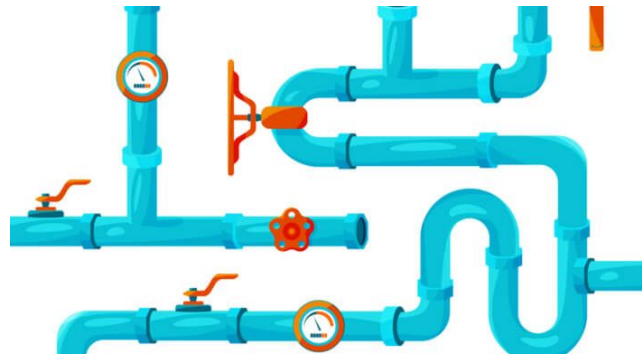
Executar testes no Jenkins

O Jenkins fornece uma funcionalidade pronta para uso para o JUnit e fornece vários plug-ins para teste de unidade para outras tecnologias.



Pipeline

- É o conjunto de instruções que podem ser executadas para fazer a integração da sua aplicação.
- O Jenkinsfile é a maneira mais recomendada para criar Pipelines
- A sintaxe utilizada é o groovy, com ele podemos definir o pipeline de forma declarativa.





That's all Folks!