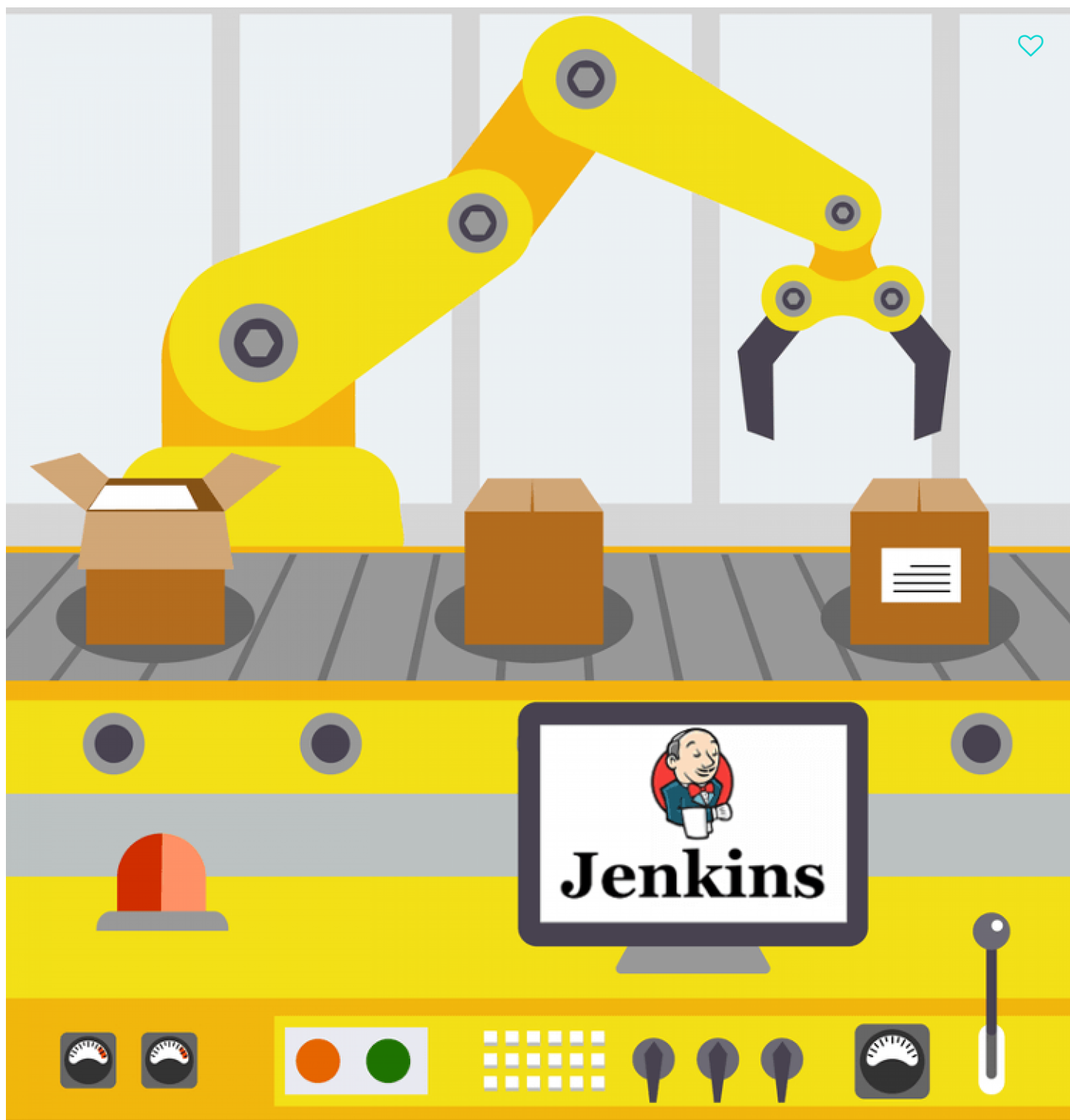


Home » DevOps » Guia definitivo: Como criar e implementar pipelines com Jenkins



14 de março de 2019



Julio Ballot



21660 Views

Guia definitivo: Como criar e implementar pipelines com Jenkins

O que é o Jenkins ?



Jenkins é uma ferramenta de auxílio a construção de [pipelines](#) de **construção de software**. Com ele podemos [automatizar](#) toda as etapas de compilação e [teste de software](#) base da **Integração Contínua (CI)**.

O que é Integração Contínua ?

Trata-se de implementações e alterações contínuas no código utilizando metodologias Ágeis. São empregados elementos como: **User-Stories, Unit Tests** e **QA** de forma automatizada, preparando uma nova versão do software para realizar a Entrega ou Implementação Contínua (CD).

Entrega ou Implementação Contínua ?

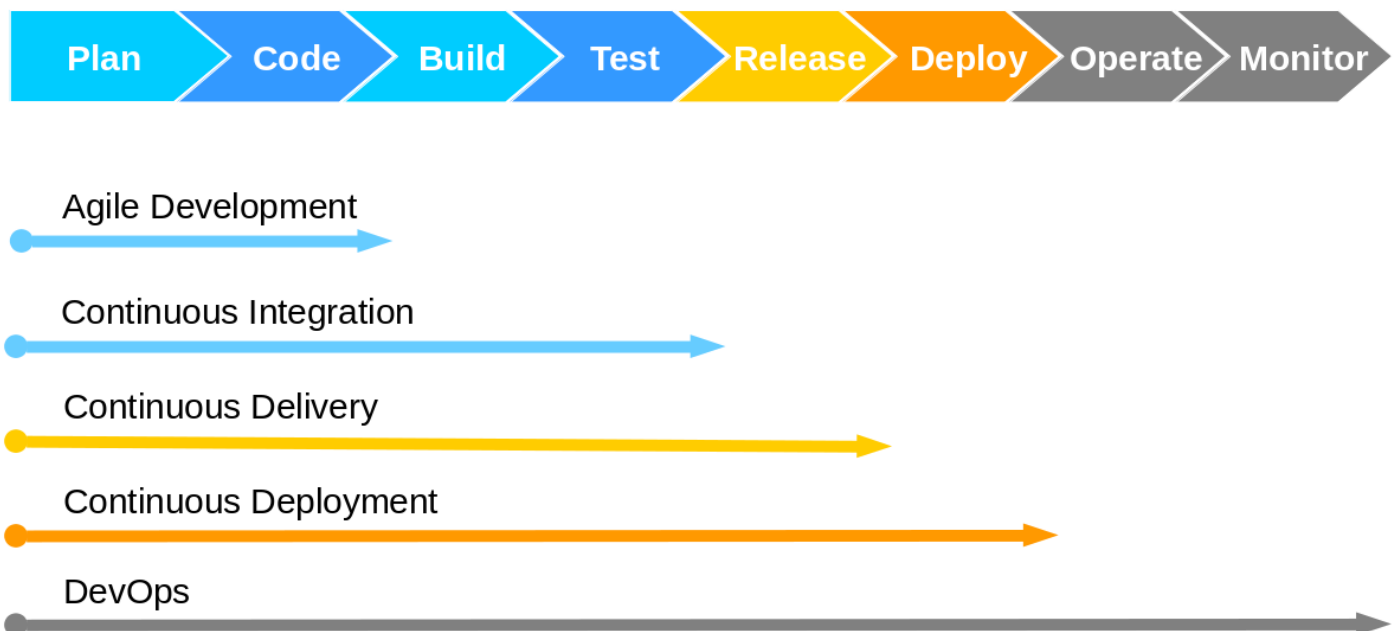
Conjunto de práticas cujo objetivo é garantir que alterações ou novas versões de software, sejam colocadas no ambiente de produção a qualquer momento.

Entrega – Aguarda confirmação manual.

Implementação – Deploy direto para produção.

Definição de um Pipeline

Para que possamos utilizar os recursos do Jenkins dentro de um [pipeline de entrega automatizada de software](#), primeiramente devemos definir as etapas de construção. É muito importante que o processo de desenvolvimento e entrega de software esteja definida antes da construção do Pipeline, uma técnica de grande auxílio nesse momento é a aplicação de um **Mapa de Fluxo de Valor**, que tem como objetivo identificar todas as etapas e processos pelo qual seu produto passa desde o requerimento feito pelo cliente até a entrega final.

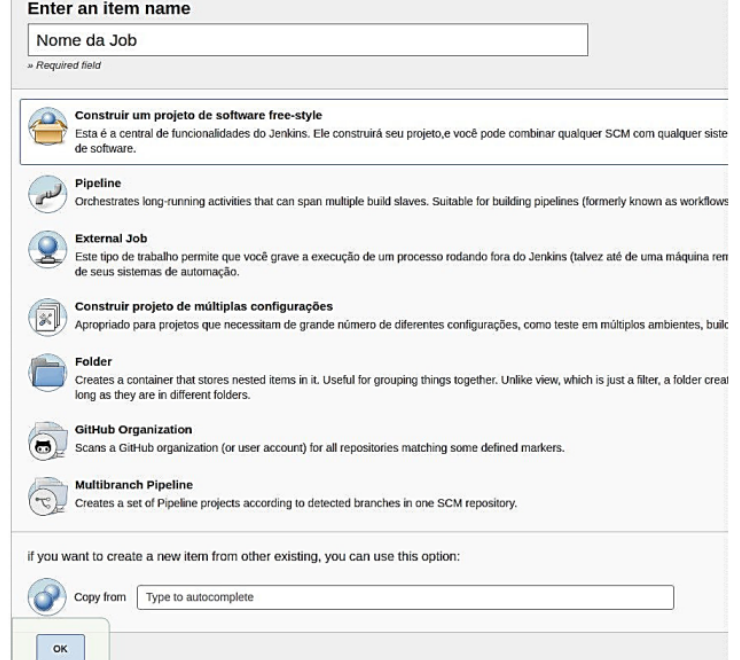
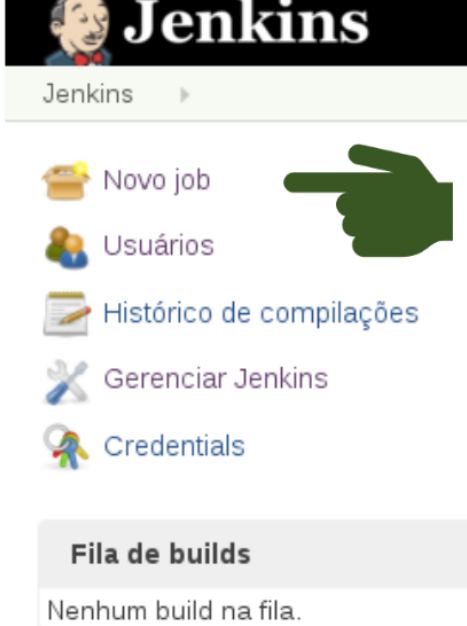


O Pipeline apresentado acima é um bom começo para [implementação do CI/CD](#). Ele contém as etapas de desenvolvimento, teste, release e deploy. Operação e [Monitoramento](#) fazem parte mas vão além da proposta deste post.

Criando Pipeline no Jenkins

Para criarmos um **pipeline** no **Jenkins** precisamos adicionar um novo job e defini-lo como Pipeline.





Ao adicionar o novo Job precisamos definir de onde o Jenkins irá buscar nosso arquivo **Jenkinsfile**. Como uma boa prática, deixamos o Jenkinsfile na raiz do repositório **SCM** da nossa aplicação. Dessa forma o Pipeline será definido por Projeto. Abaixo um exemplo com um projeto básico de HelloWorld **Java** utilizando o framework MVC SpringBoot.

Name	Last commit	Last update
📁 .settings	Primeiro commit	4 months ago
📁 src/main/java/com/boraji/tutorial/springboot	Update HelloWorldController.java	2 days ago
📁 target/classes/META-INF	Primeiro commit	4 months ago
📄 .classpath	Primeiro commit	4 months ago
📄 .project	Primeiro commit	4 months ago
📄 Jenkinsfile	Update Jenkinsfile	2 days ago
📄 pom.xml	Update pom.xml	2 days ago

Esse é um exemplo de repositório **Git** no **Gitlab**, note a presença do Jenkinsfile como parte do projeto.

Agora com o Jenkinsfile em nosso repositório devemos configurar o Job no Jenkins para buscar esse arquivo em nosso SCM. Dentro do Job, na etapa de Pipeline defina como **Pipeline Script from SCM**, após isso informe o endereço de seu servidor SCM e a credencial de login. Isso será o suficiente para que o Jenkins se autentique no repositório e busque o Jenkinsfile.

