

Integração Contínua com Jenkins

Mão na Massa



DevOps para Desenvolvedores
Java



Camilo Lopes

Integração Contínua com Jenkins

DevOps para desenvolvedores Java

Camilo Lopes

Esse livro está à venda em <http://leanpub.com/integracaocontinuacomjenkins>

Essa versão foi publicada em 2015-04-05

Todos os direitos são reservados.

©2013 - 2015 ITSLabs

Tweet Sobre Esse Livro!

Por favor ajude Camilo Lopes a divulgar esse livro no [Twitter](#)!

O tweet sugerido para esse livro é:

Acabei de comprar o ebook "Integração Contínua com Jenkins" do @camilope

A hashtag sugerida para esse livro é [#jenkins4java](#).

Descubra o que as outras pessoas estão falando sobre esse livro clicando nesse link para buscar a hashtag no Twitter:

<https://twitter.com/search?q=#jenkins4java>

Dedico esse livro ao que é mais importante na minha vida que é minha família, e em especial a minha mãe senhora Celidalva Medeiros, minha avó senhora Benigna Sales, ao meu pai sr. Camilo Lopes. Foram inúmeras pessoas que me ajudaram a chegar a conclusão desse trabalho, difícil poder listar todas aqui, mas gostaria de agradecer ao meu amigo e escritor Edson Gonçalves que é meu padrinho na minha carreira.

Conteúdo

Sobre o autor	1
Falando diretamente	2
ITSLabs	3
Sobre o livro	4
Objetivo	5
Quem pode ler?	6
O que veremos ?	7
Como esse livro foi escrito?	8
Atualizações	9
Capítulo 1 - Introdução sobre Integração Contínua	10
Introdução sobre Integração Contínua	10
Sobre o Jenkins	10
Instalando e configurando o Jenkins	11
Restart Jenkins	13
Resolvendo problemas de unicode	14

Sobre o autor



Camilo Lopes formado em Sistemas de Informação, trabalha com desenvolvimento de software desde 2003. Trabalhou em empresas como IBM, HP Labs, Ci&T, Infosys em projetos internacionais e nacionais. Em 2007 entrou no mundo Java e se especializou na tecnologia. Advoga uso de metodologias Ágeis desde 2010. Experiência em projetos Scrum presenciais e times remotos. Autor dos livros “Guia do Exame SCJP”, “Construindo aplicações JEE com frameworks” e “TDD na Prática”. Em 2013, largou o mundo corporativo das multinacionais e tem se dedicado à vida e desafio de start-up. Fundador da ITSLabs <http://www.itslabs.com.br> e UseCash <https://www.usecash.com.br>. A ITSLabs é uma aceleradora de start-up voltada para criação de produto MVP Web e Mobile. A UseCash é um novo meio de pagamento pré-pago que dispensa o uso do cartão físico permitindo ao consumidor receber e efetuar pagamentos usando apenas o CPF nos estabelecimentos credenciados. Nas horas livres quando não está brincando com novas tecnologias ou blogando <http://www.camilolopes.com.br>, prefere está com a família e amigos.

Falando diretamente

Envie um e-mail para camilo@camilolopes.com.br com sua crítica, elogio ou sugestão. Gostaria muito de receber dicas do que gostariam de ver em versões futuras do livro.

Twitter: [@camilolope](https://twitter.com/camilolope)

GitHub: github.com/camilolopes

ITSLabs

A ITSLabs <http://www.itslabs.com.br> é uma aceleradora de start-up voltada para criação de produto MVP Web e Mobile. Temos um compromisso com a comunidade em contribuir nossa experiência técnica do dia a dia através de blog, ebooks, vídeos etc.

Agradecemos sinceramente a você por ter adquirido esse e-book pagando uma valor simbólico. Pois, sabemos que ter uma cópia de algum outro colega não é algo difícil, mas você confia em nosso trabalho e está investindo em motivar que a equipe continue criando mais conteúdo.

Sobre o livro

Nasceu a partir de uma necessidade de projeto e estudo que tive no início de 2012/2013 quando fui trabalhar em um projeto com integração contínua e simplesmente não encontrei nenhuma obra em português que tivesse pelo menos o ABC. O assunto era muito discutido em livros escritos em inglês. Apesar de saber o segundo o idioma, ainda assim tinha a necessidade algo mão na massa, focado e voltando para sair fazendo e poder colocar em produção nos projetos aqui na empresa. Eu precisava de um material que fosse bem nosso dia a dia, mão na massa. E nada de encontrar, e tive que criar um novo caminho, pesquisar na internet, ler documentação, consultar apenas uma parte de um livro que falava do assunto, e ir montando o quebra-cabeça. E depois de toda essa maratona, refletir e disse, será que não seria uma boa escrever um livro que gostaria de ler sobre integração contínua com o Jenkins, porém focado para desenvolvedores JAVA? E assim, nasceu Integração Contínua com Jenkins um livro para quem desenvolve em Java e quer montar um ambiente produtivo para build e deploy. Compartilhamos nesse ebook a nossa experiência aqui na ITSLabs, pois não vivemos sem ambiente Continuous Integration por aqui. Claro que não abordamos tudo, apenas a base para quem deseja sair do 0 x 0. Já estamos pensando no futuro ter um livro mais avançado.

Objetivo

Este livro tem como objetivo ensinar de maneira prática como montar um ambiente de integração contínua para projetos Java.

Quem pode ler?

Todo profissional Java ou DevOps que deseja aprender como montar um ambiente de integração contínua com Jenkins. O livro vai usar como exemplos projetos Java com Maven. É um livro para iniciantes com Integração Contínua. Não espere aprender tudo sobre assunto e se tornar um expert nesse livro.

O que veremos ?

Um pouquinho de cada coisa, vamos lá:

- Capítulo 1: vamos fazer uma introdução sobre o conceito de integração contínua e a importância de ter um ambiente CI, para desenvolvimento de Software;
- Capítulo 2: Iremos aprender como integrar Maven, Jenkins e Git, ou seja, já colocamos mão na massa o quanto antes;
- Capítulo 3: Aprenderemos como executar unit tests dentro do nosso ambiente CI. E se um teste falhar o que acontece? Vamos ver isso lá;
- Capítulo 4: Aqui vamos descobrir a potência dos plugins dentro do Jenkins. Quero conectar com o Github como faço? Tem plugin pra isso? Quem sabe, vamos descobrir;
- Capítulo 5: Conectando com o Github, ou seja, o Jenkins vai realizar o build do projeto direto do código fonte que está no repositório remoto;
- Capítulo 6: Agora chegou um momento, de deixar tudo automático, mudou o repositório, o próprio Jenkins começa o trabalho dele. Ou seja, para cada commit um build roda para validar se o código está chegando está ok.
- Capítulo 7: Temos a opção de não usar autenticação de usuário padrão do Jenkins, vamos ver aqui como usar autenticação vindo de outros serviços;
- Capítulo 8: Montaremos um ambiente com pipeline, que é um plugin no Jenkins que nos permite criar vários ambientes, aumentando mais ainda a nossa produtividade;
- Capítulo 9: E que tal, fazer um hotdeploy direto do Jenkins no seu servidor remoto, ou seja, o copy do .war ser feito por uma ferramenta? É isso que esse capítulo aguarda.
- Capítulo 10: Como integrar o Jenkins com a ferramenta mais famosa para análise de código estático no mundo Java? Estou falando dele mesmo, o sonar. Vamos ver como fazer Jenkins trabalhar com o Sonar.
- Capítulo 11: Vamos ver como clonar um workspace. Essa pratica é super importante para ganharmos velocidade nos builds e evitar redundância em processos.
- Capítulo 12: Certamente em algum momento no seu projeto você vai precisar criar uma Tag que significa uma versão estável da sua aplicação. Nessa aula vamos ver como automatizar o processo de criar uma tag, publicar no github.
- Capítulo 13 e 14: São parecidos, na verdade se completam, apenas separei para não gerar confusão. Mas, é fazer automatização da criação do seu banco de dados através do Jenkins e usando o plugin flyway.

Como esse livro foi escrito?

Todo o livro foi escrito de uma forma direta e com uma escrita bem informal, o meu objetivo é compartilhar o conhecimento da maneira mais simples, possível sem rodeios e que você leitor tenha a sensação de um bate papo comigo ao invés de uma tradicional leitura. Eu espero que goste desse modelo, vou aguardar seu feedback.

Atualizações

Vamos buscar atualizar o livro periodicamente com as novidades sobre o assunto e também, com os feedbacks enviado por você leitor. Então, me mandar e-mail será extremamente importante para irmos evoluindo o e-book.

Capítulo 1 - Introdução sobre Integração Contínua

Nesse capítulo vamos conhecer um pouco sobre o Jenkins e aprender a instalar localmente. De fato será um capítulo introdutório, se você é iniciante acredito que vai gostar.

Introdução sobre Integração Contínua

Integração Contínua ou Continuous Integration (CI) é uma pratica muito importante no desenvolvimento de software que teve grande impacto causado pela metodologias ágeis. Equipe usam Agile(Scrum, XP etc) usar integração continua é praticamente como tomar café todos os dias de manhã, ou seja, é preciso ter.

Mas, por que?

A primeira vantagem é o feedback instantâneo. Ou seja, a equipe tem um feedback de imediato sobre qualquer mudança que acontece no repositório. Quando uma mudança é feita, ou seja, um commit é realizado por um desenvolvedor o build do projeto é iniciado automaticamente, roda todos os testes (caso tenha). Caso, o código não compile ou tenha falhas nos testes, o build vai falhar e a equipe não tem como fazer o deploy da aplicação. E assim, a equipe sabe disso de imediato, ou seja, a cada commit é preciso saber se o código novo está estável ou não.

Outro fator é a produtividade, ao invés de fazer todo o trabalho manual de deploy é possível transferir essa responsabilidade para a ferramenta de CI, onde podemos informar a máquina e onde será feito o deploy. O que pode levar manualmente horas, com CI pode-se levar segundos ou até minutos.

Sobre o Jenkins

O Jenkins nada mais é que uma ferramenta open source para Integração Contínua escrito em Java. Projeto que foi um fork do Hudson. O Jenkins oferece:

- Integração com ferramentas de controle de versão como SVN, Git, Mercurial;
- Build e deploy do projeto;
- Suporte a Maven e Ant;

O Jenkins pode ser baixado em .war e executável. Para executar .war basta fazer o deploy em um servidor como Tomcat, Jetty, Jboss etc.

Instalando e configurando o Jenkins

Há duas opções de termos o Jenkins rodando:

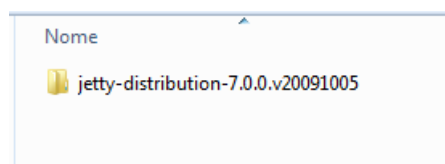
1. É instalar o Jenkins como serviço na máquina;
2. Fazer o deploy do Jenkins em um servidor de aplicação como Tomcat, Jetty, Jboss etc.

Particularmente prefiro ter um servidor e fazer o deploy do .war do Jenkins.

Requisitos

Faça o download de uma versão do jetty. Vou usar o jetty 7.x porque é o que tenho aqui na máquina, porém há versões mais recentes: <http://jetty.codehaus.org/jetty/>

Após o download do jetty, descompacte o arquivo:



Pasta Jenkins descompactado

Abra o prompt de comando do Windows e vá até o local onde está a pasta do jetty:

```
D:\CAMILLO\T.I\DEU\Servers\jetty-distribution-7.0.0.v20091005\jetty-distribution-7.0.0.v20091005>
```

Podemos iniciar o Jetty só para ver se está tudo ok com ele, então digite:

```
1 java -jar start.jar
```

```
\jetty-distribution-7.0.0.v20091005\jetty-distribution-7.0.0.v20091005>java -jar start.jar
```

E aguarde o jetty iniciar. Após a inicialização do jetty acesse <http://localhost:8080>

Você verá o seguinte:



Welcome to Jetty 7

This is the Test webapp for the Jetty 7 HTTP Server and Servlet Container. For more in

This is a test context that serves:

- static content ([tiny](#), [small](#), [medium](#), [large](#), [large gzipped](#))
- a [Hello World Servlet](#)
- a [Request Dump Servlet](#)
- a [Session Dump Servlet](#)
- a [Cookie Dump Servlet](#)
- a [Dispatcher Servlet](#)
- a [Transparent Proxy](#) (to [jetty.mortbay.org](#))
- a [CGI script](#) (unix only)
- a [Authentication](#)

This webapp is deployed in \$JETTY_HOME/webapp/test and configured by \$JETTY_

Pronto, temos o jetty rodando. Isso é bom.

Agora vamos parar o jetty para fazer o deploy do Jenkins. Pare a aplicação com ctrl + C.

Vá no site do Jenkins e baixe a versão .war. (estou usando a última versão)

Após ter baixado o .war coloque o mesmo na pasta webapps, dentro do diretório do jetty que você descompactou ainda pouco:

No meu caso:

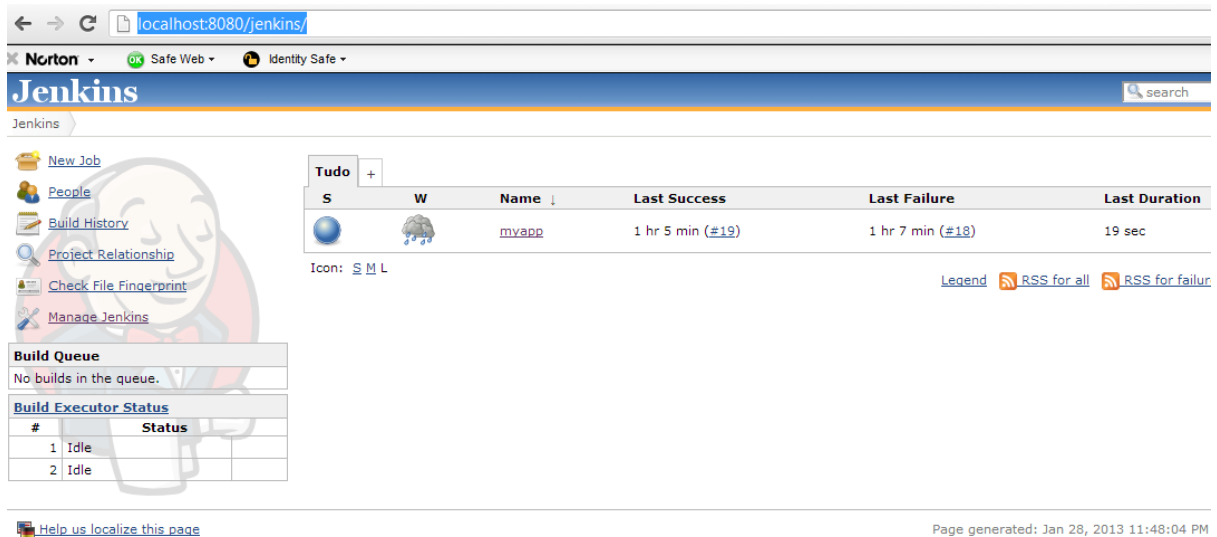
```
1 jetty-distribution-7.0.0.v20091005\jetty-distribution-7.0.0.v20091005\webapps
```

Agora, inicie o servidor Jetty novamente:

```
1 java -jar start.jar
```

E precisamos aguardar o jetty fazer o deploy do Jenkins. Na primeira vez será um pouco demorado. Após o jetty concluir o trabalho dele, vamos ver se o Jenkins realmente foi deployed.

Acesse <http://localhost:8080/Jenkins/> e verá:



S	W	Name	Last Success	Last Failure	Last Duration
S		mvapp	1 hr 5 min (#19)	1 hr 7 min (#18)	19 sec

#	Status
1	Idle
2	Idle

No meu caso, tenho um job criado no Jenkins, mas não se preocupe, pois faremos um mais na frente.

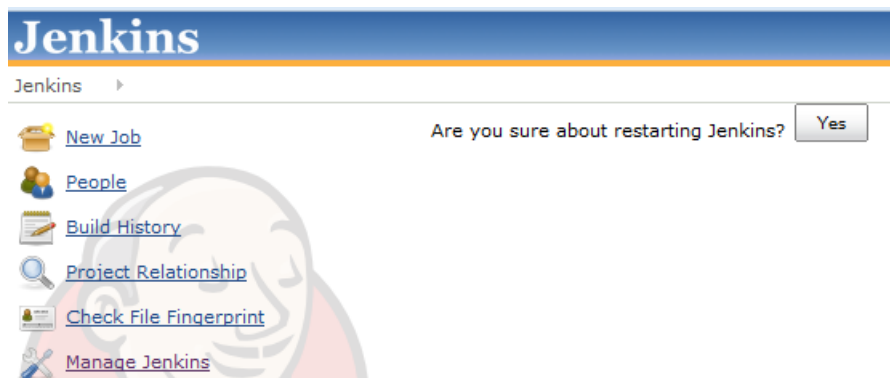
Restart Jenkins

Como reiniciar o Jenkins sem reiniciar o servidor pelo painel de controle?. Parece algo que deveria ser simples, ou seja, um simples botão resolveria a situação. Mas não temos esse cara no painel do Jenkins.



Então, como fazer?

<http://seuservidor:porta/Jenkins/restart>



Se tiver local:

<http://localhost:8080/Jenkins/restart>

Jenkins



Please wait while Jenkins is restarting.

Your browser will reload automatically when Jenkins is ready.

O Jenkins vai perguntar se realmente deseja reiniciar o servidor, caso deseje clique no botão “OK” e aguarde.

Resolvendo problemas de unicode

Muito recorrente quando estamos montando um ambiente contínuo. No meu caso tive problemas de Unicode quando o Jenkins executou mvn test. Vamos ver como resolver.

O problema

A questão é que os testes vão falhar. Se você tem algum teste que faz validação que envolve acento, por exemplo você espera “João” e devido ao problema de Unicode veio sem o acento e outro caractere. Como resolver?

```
root of factory hierarchy>
Results :
Failed tests:   testUpdateDataOfTheUser(com.camilolopes.readerweb.services.impl.UserServiceimImplTest): expected:<jo[Ã]o> but was:<jo[ã]o>
Tests run: 8, Failures: 1, Errors: 0, Skipped: 0
[ERROR] There are test failures.

Please refer to C:\Users\camilolopes\.jenkins\workspace\ReaderWeb\target\surefire-reports for the individual test results.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO]
```

A solução

É bem simples, precisamos apenas no nosso pom.xml informar qual Unicode vamos usar, nesse caso UTF-8. Daí precisamos configurar as propriedades assim:

```
1 <properties>
2 <maven.test.failure.ignore>false</maven.test.failure.ignore>
3 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
4   <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
5 </properties>
```

E agora podemos executar um mvn clean. E em seguida mandar o Jenkins compilar e o resultado será:

```
Results :  
  
Tests run: 8, Failures: 0, Errors: 0, Skipped: 0  
  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 10.978s  
[INFO] Finished at: Thu Apr 18 14:34:16 BRT 2013  
[INFO] Final Memory: 5M/77M  
[INFO] -----  
Finished: SUCCESS
```

Pronto, problema resolvido. Simples não?

Conclusão

Aprendemos um pouco sobre a importância de ter Integração Contínua durante o desenvolvimento de software, conhecemos rapidamente o Jenkins e vimos como instalá-lo localmente, reiniciá-lo e resolver problemas de unicode.