

Build e deploy manual

Transcrição

Agora que fizemos as configurações básicas do Jenkins, podemos usar o Jenkins pra fazer o que realmente importa que é automatizar tarefas pra gente, compilar e empacotar os projetos, colocar os nossos projetos no servidor entre outras. E qual o projeto que iremos trabalhar? Antes de mais nada, antes de configurar, antes de iniciar a criação do nosso *Job* (que é onde fazemos toda a automação) precisamos ver qual será o projeto que iremos trabalhar. Precisamos saber quais serão passos que queremos automatizar, que é então o que faremos agora, vamos entender primeiro por completo onde o Jenkins vai nos ajudar. Pra isso a primeira coisa que eu preciso saber é onde está o nosso projeto? Em qual repositório de código o nosso projeto se encontra? No SVN, no GIT ou no CVS? No caso, o nosso projeto, tá no github do **alura-cursos** e o repositório chama-se **argentum-web** (<https://github.com/alura-cursos/argentum-web>). O que é importante saber desse projeto? É importante saber que ele usa *Maven*. No arquivo `pom.xml` eu tenho configurações mínimas de *Maven* para poder fazer tudo o que eu preciso no nosso projeto, ou seja: compilar, empacotar, rodar os meus testes, rodar meus testes integrados, etc. Esse é o mínimo que precisamos saber para avançar.

Então podemos voltar na raiz do repositório para buscar a URL que precisamos para baixar o projeto e tentar compilá-lo. Para isso vamos em **Clone or Download** e copiar a URL que é mostrada pra nossa área de transferência. Vamos voltar ao terminal e navegar até o diretório onde estamos colocando os nossos arquivos (`cd $AMBIENTE_IC`) para podermos criar uma pasta para colocar o projeto. Para isso, faremos `mkdir projeto` e acessaremos a pasta que criamos fazendo `cd projeto`.

Após o `clone` podemos acessar a pasta do projeto, fazendo `cd argentum-web` e começar a trabalhar com o Maven. Nosso primeiro passo será compilar o projeto e para isso vamos utilizar o comando `mvn compile`.

Podemos ver que o projeto compilou com sucesso e tudo aparentemente está em ordem. O que faremos agora é empacotar o projeto, ou seja, gerar o `.war` e deixá-lo pronto pra fazer o deploy no servidor java. Para isso, vamos utilizar mais um comando do Maven que é o comando `mvn package`. O comando `mvn package` ou o **goal** `package` é mais avançado no ciclo de vida. O que significa que quando eu peço um `package` ele irá também executar o `compile` que fizemos anteriormente. Após executarmos esse comando e navegarmos até a pasta `target` podemos ver lá o meu `argentum.war`. Agora o que queremos é pegar esse `argentum.war` e colocar em algum servidor Java (qual você preferir). No curso vamos utilizar o Tomcat, mas o projeto não existem restrições.

Instalando o Tomcat

Primeiramente visite a página do Tomcat (<http://tomcat.apache.org/>) (<http://tomcat.apache.org/>), e entre na página do **Tomcat 7**. Baixe o **Core** adequado para o seu sistema operacional e descompacte em uma pasta de sua escolha.

Navegue para a pasta no qual você descompactou o seu tomcat, normalmente com o nome de `apache-tomcat-7` e entre na pasta `/bin` que fica dentro dela. Aqui, precisamos executar o arquivo `startup.sh`, que inicializará nosso Tomcat, porém este arquivo não tem permissão de execução ainda. Para resolver isto, vamos utilizar o comando abaixo:

```
chmod +x *.sh
```

Que dará a permissão de execução a todos os arquivos `.sh` da pasta `bin`. Agora com as permissões corretas, vamos utilizar o comando dentro da pasta `bin`:

```
./startup.sh
```

Que inicializará o nosso tomcat na porta 8080 do localhost. Para verificar se tudo ocorreu corretamente basta visitar a página <http://localhost:8080/> e verificar se uma página de bem vindo do Tomcat 7 será aberta.

Adicionando o projeto no Tomcat

Agora precisamos adicionar o nosso projeto ao Tomcat, para conseguirmos executá-lo em nosso servidor. Para isto volte para a pasta raiz do tomcat, e entre na pasta `webapps/`. Copie o `argentum.war`, que estava na pasta `/target`, que foi gerando pelo Maven para a pasta `webapps/`. Para atestar se o projeto está funcionando corretamente, faça o acesso dele através do `localhost:8080/argentum-web/`.

A página inicial do projeto deve abrir corretamente e você verá a imagem abaixo:

