

Configurando processos contínuos

Transcrição

A primeira coisa que precisamos ter em mente é a periodicidade de execução do Job. Em um mundo ideal, ele deveria ser executado toda vez que o repositório do nosso projeto for alterado e com certeza isso é possível com Jenkins. Partindo dessa estratégia, podemos saber se o código do nosso repositório compila, se passa nos testes de unidade ou integração. Só se tudo estiver correto podemos atualizar nosso servidor com nosso projeto construído.

Um commit, um build

Para fazermos isso, a configuração deve ser no repositório `git` do projeto adicionando no arquivo `nomeDoSeuRepositorio/hooks/post-receive` o seguinte conteúdo:

```
curl http://endereco.do.jenkins/git/notifyCommit?url="https://github.com/alura-cursos/argentum-web.git  
(http://endereco.do.jenkins/git/notifyCommit?url="https://github.com/alura-cursos/argentum-web.git")"
```

Veja que esta configuração precisa do endereço remoto do nosso servidor de integração, do Jenkins. Contudo, como nosso servidor utilizado no curso roda localmente, o `hook` adicionando no repositório do `git` que chama nosso servidor de integração não funcionará. Não se preocupe, para efeitos de aprendizagem, podemos adotar outra abordagem, contudo vale lembrar que essa é a mais recomendada em ambiente de produção.

Build periódico

Uma solução é rodarmos nosso build de tempos em tempo e o Jenkins permite agendarmos a execução do nosso build em intervalos. Toda a configuração é feita através da opção "Configurar" do job. Nela, há a opção "Trigger de Builds". Nela, marcamos a check box "Construir periodicamente".

Em seguida, na área "Agenda", utilizamos a sintaxe do `cron`, um agendador de tarefas da plataforma Linux. Se você não domina a sintaxe do `cron`, não se preocupe, clicando no ícone interrogação ao lado do campo você terá instruções breves de como proceder. Vamos usar:

```
H/15 * * * *
```

O código acima fará com que nosso build seja realizado de quinze em quinze minutos. Veja que durante esse processo o Jenkins precisará varrer todo o seu repositório do `git`, o que seria impraticável se tivéssemos agendado para rodar nosso build de um em um minuto.