

Versionamento do código

Transcrição

O script que o instrutor segue durante a aula é o seguinte:

```
# Passos para a configuracao do git e versionamento do codigo
# Criar uma conta em github.com
ssh-keygen -t rsa -b 4096 -C "<seu-usuario>@gmail.com"
cat ~/.ssh/id_rsa.pub
# Configurar a chave no github
git config --global user.name "<seu-usuario>"
git config --global user.email <seu-usuario>@<seu-providor>
ssh -T git@github.com
# Fazer download do código nos anexos e copiar para o diretório compartilhado app
cd /vagrant/jenkins-todo-list
git init
git add .
git commit -m "Meu primeiro commit"
git log
# Criar um repositório no github: jenkins-todo-list
git remote add origin git@github.com:<seu-usuario>/jenkins-todo-list.git
git push -u origin master
```

[00:00] Ok pessoal, então agora é a hora da gente começar a montar o nosso ambiente. A gente já tem o Vagrant funcional, já tem o MySQL e o Jenkins também. Pra exemplo nós fornecemos pra vocês uma aplicação escrita em Django que é um todo list, é muito simples, grava no MySQL e a gente vai utilizar ele daqui pra frente.

[00:20] Pra começar, a primeira coisa que a gente tem que fazer é extrair o arquivo dentro do nosso diretório de trabalho. Então é o arquivo jenkins-todo-list.zip, a gente vai extrair ele aqui.

[00:35] Bom, se a gente olhar agora no nosso diretório do Vagrant eu tenho o código fonte da minha aplicação. Porque que eu preciso do código fonte? Por que agora eu versionar o código fonte, eu vou colocar esse código dentro do GitHub. Pra que isso aconteça eu preciso ter, primeira coisa, um par de chaves SSH pra fazer a conexão.

[00:57] Então pra criar esse par de chaves que que a gente faz? A gente passa o tipo da chave com o e-mail. Ele vai perguntar pra gente se vai ser a chave padrão, vai ser a chave padrão, a gente vai dar um enter. Não vai ter passphrase a gente vai conectar direto utilizando um par de chaves. E aí foi criado.

[01:20] Agora que que a gente precisa fazer? A gente precisa, localmente, configurar tanto o nosso git com e-mail, quanto com o nosso usuário. Então a primeira coisa vai ser configurar com usuário e depois a gente vai configurar com o e-mail. Desse jeito, a hora que a gente fizer o commit do nosso código push pro repositório, ele vai saber quem é que fez e vai ter os registros direitinho lá.

[01:51] Agora é a etapa que a gente vai acessar o GitHub, porque? Eu tenho já o par de chaves, só que agora eu preciso colocar a minha chave pública no GitHub pra permitir acesso aos meus repositórios. Então o que a gente vai fazer? A gente vai abrir aqui o nosso github.com, aqui no canto alto a gente vai clicar em Settings aí aqui embaixo eu tenho as chaves de SSH que eu posso instalar, remover e manusear elas de qualquer maneira.

[02:27] Então eu vou criar aqui chamado alura-jenkins e agora eu preciso saber qual é a minha chave pública, é só eu pegar o conteúdo desse arquivo aqui, /home/vagrant/.ssh/id_rsa.pub que é a minha chave pública. Copio e colo aqui no campo da chave.

[02:58] Ele vai pedir pra eu confirmar minha senha. Senha confirmada, instalado, agora eu vou fazer o teste pra saber se a minha chave pública foi corretamente instalada e se tá tudo certinho. Dou um ssh -T git@github.com. Ele vai perguntar se eu aceito fingerprint, eu dou um yes e pronto, ele me reconheceu e me autenticou.

[03:29] Agora que que a gente vai fazer? A gente vai versionar o nosso código pela primeira vez. Então vamos acessar o diretório dele aqui, um ls aqui, é esse carinha aqui. Primeira coisa que eu tenho que fazer é iniciar o meu repositório local, o git é distribuído, ele é diferente de outros versionadores, então eu preciso iniciar o repositório localmente. Como é que eu faço isso? git init, ele inicializou um repositório aqui pra mim, tá local ainda, então a gente vai chegar lá.

[04:04] Agora que que a gente vai fazer? Vai adicionar todos os arquivos dentro desse versionamento, da seguinte maneira: "git add .". Isso a gente fez agora, mas normalmente no dia a dia a gente não versiona todos os arquivos todas as vezes que a gente altera alguma coisa, a gente versiona o arquivo que é necessário naquele momento. Nesse caso a gente tá versionando de uma vez só.

[04:29] E agora a gente vai criar o primeiro commit, só vamos trocar a mensagem aqui e colocar assim: "Meu primeiro commit". E ele adicionou todos os arquivos que estavam dentro desse diretório, com exceção dos arquivos que eu listei, ou diretórios, dentro desse arquivo aqui que é o gitignore. Então aqui a gente só exclui exatamente o que nós não queremos que entre dentro do nosso versionamento.

[05:05] Vocês, também como pré-requisito, têm o curso de git da Alura, vocês já devem ter assistido. É muito simples de criar esse arquivo, vamos dar um head nele só pra ver o conteúdo dele, são só a cada linha uma entrada, na hora que eu versionei tudo que ele colocou aqui em cima não estava, ele até avisa que tem um arquivo gitignore aqui.

[05:29] Bom, será que ele realmente versionou? Isso aqui tá tudo local. A gente vai digitar git log e aí a gente tem nosso primeiro versionamento local. Então é o meu primeiro commit. Qual foi o autor, qual foi o e-mail, ele trouxe daquelas configurações que a gente fez. Qual que é a data que foi versionado e qual que é o head, qual que é o código do commit, a gente vai ver mais para frente como que a gente lida com isso aí.

[05:56] Bom, agora que a gente tá dentro do GitHub a gente precisa criar um repositório. Como que eu crio um repositório no GitHub? Eu já criei um aqui mas eu vou mostrar pra vocês. A gente vem aqui nesse sinal de mais, Novo Repositório, coloquem o nome do repositório, por exemplo jenkins-todo-list, nesse nosso caso, defina se ele é público ou privado.

[06:24] Vale ressaltar que, se a gente colocar ele como privado, a única maneira de fazer clone ou qualquer interação com ele é usando chave SSH. E aí vocês vão criar o repositório. Como eu já tenho esse repositório ele tá falando que já existe na minha conta, que é esse repositório aqui.

[06:42] Agora tá quase no final, a gente já versionou localmente, a gente já fez o nosso commit, eu só preciso avisar agora onde fica o repositório central dele, que é o caso do GitHub. A gente faz usando esse comando aqui: git remote add, vamos colocar ele aqui, git remote add origin e o usuário git@github.com: o repositório.

[07:10] Acabei adicionando, e agora eu tenho que fazer o que? Enviar meu código pro repositório, ele tá versionado só localmente agora. E é muito simples fazer isso, git push origin master. Pronto, meu código tá versionado, se eu chegar aqui no repositório e atualizar a página tá aqui. Tá versionado, todos os meus arquivos estão aqui. E agora, a próxima aula a gente vai aprender como criar o nosso primeiro job no Jenkins pra olhar esse repositório.

[07:58] Bom, a gente já tá com o ambiente configurado, com nosso código versionado, com o Jenkins instalado e o GitHub já tem o último código nosso. Tá faltando o quê agora? Criar o nosso primeiro job pra construir a aplicação. Mas isso aí eu vou

te mostrar no próximo vídeo.