

## Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

1) Configure o seu ambiente inicial. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```
# Configurando o ambiente e conectando máquina virtual:
# Máquina virtual com o vagrant
# Requisitos: http://vagrantup.com e https://www.virtualbox.org/
# Baixar e descompactar o arquivo 1110-aula-inicial.zip
# Entendendo o Vagrantfile
# Subindo o ambiente virtualizado
  vagrant plugin install vagrant-disksize
  vagrant up
  vagrant ssh
    ps -ef | grep -i mysql # Verificando se o MySQL esta rodando
    mysql -u devops -p # Senha mestre; show databases
    mysql -u devops_dev -p # Senha mestre; show databases
    # Instalando o Jenkins
    cd /vagrant/scripts
    # Visualizar o conteudo do arquivo de instalacao do jenkins
    sudo ./jenkins.sh

    # Acessar: 192.168.33.10:8080
    sudo cat /var/lib/jenkins/secrets/initialAdminPassword

# Credenciais
  Nome de usuário: alura
  Senha: mestre123
  Nome completo: Jenkins Alura
  Email: aluno@alura.com.br

# Reload nas permissoes do docker
  sudo usermod -aG docker $USER
  sudo usermod -aG docker jenkins
  exit
vagrant reload
```

2) Versione o seu código, que você baixou no início da aula. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```
# 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
```

3) Crie o seu primeiro *job*, que vai monitorar o seu repositório. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```
# Configurando a chave privada criada no ambiente da VM no Jenkins
cat ~/.ssh/id_rsa
Credentials -> Jenkins -> Global Credentials -> Add Credentials -> SSH Username with private key
# Criando o primeiro job que vai monitorar o repositório
Novo job -> jenkins-todo-list-principal -> Freestyle project:
Esse job vai fazer o build do projeto e registrar a imagem no repositório.
# Gerenciamento de código fonte:
  Git: git@github.com:rafaelvzago/treinamento-devops-alura.git [SSH]
  Credentials: git (github-ssh)
  Branch: master
# Trigger de builds
  Pool SCM: * * * * *
# Ambiente de build
  Delete workspace before build starts
# Salvar
# Validar o log em: Git Log de consulta periódica
```