

05

Ambiente inicial

Transcrição

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

```
# 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 Vagranfile  
# 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 permissões do docker  
    sudo usermod -aG docker $USER  
    sudo usermod -aG docker jenkins  
    exit  
vagrant reload
```

[00:00] Olá aluno, tudo bem? Bem-vindo de volta, vamos continuar nosso curso de Jenkins. Na aula agora a gente vai entender como vai funcionar o nosso ambiente pra esse treinamento.

[0:12] Como cada um de vocês possui uma característica diferente de sistema operacional, distribuição pra quem utiliza Linux, a gente vai padronizar a nossa configuração da seguinte maneira: nós vamos utilizar o VirtualBox pra máquinas virtuais, acessando virtualbox.org a gente consegue fazer o download, cada um faz pra sua própria distribuição e o Vagrant pra fazer a orquestração dessas máquinas virtuais.

[0:40] Vocês tem como pré-requisito pra esse treinamento o curso de Vagrant, se vocês seguirem lá vocês vão entender exatamente como funciona cada um dos blocos que a gente vai acabar utilizando.

[0:50] Tem mais uma ferramenta que a gente vai utilizar também que é o Cmder que é uma ferramenta pra quem usa Windows, é um terminal um pouquinho mais rico em recursos, diferente do prompt padrão. Já quem usa Linux ou Mac pode usar o próprio terminal mesmo, os comandos funcionam pros dois.

[01:08] Então o primeiro arquivo que vocês fizeram download pra aula foi o aula-inicial.zip, a gente vai extrair esse arquivo aqui pra entender como vai funcionar. Então a gente vai ver agora o primeiro arquivo do nosso treinamento que é o Vagrantfile, vamos editar aqui esse arquivo pra entender mais ou menos o que ele faz.

[01:31] Bem rapidamente, ele basicamente determina qual vai ser o sistema operacional que a gente vai usar, que vai ser o Ubuntu; o quanto de disco que a gente vai utilizar, por padrão o Vagrant usa 10GB, a gente vai precisar de um pouquinho mais pra fazer algumas coisas; o mapeamento das portas que a gente vai utilizar tá dentro dessa seção aqui; ele vai criar um IP, esse IP é muito importante, a gente vai usar ele pra quase todos os serviços que a gente vai acessar; e, aqui pra baixo, o quanto de memória e algumas configurações que são meio que necessárias pra gente poder trabalhar nesse início de treinamento.

[02:10] Então aqui, feito o download, os requisitos básicos a gente já cobriu, a gente descompactou esse arquivo e entendeu como o Vagrantfile funciona. Feito isso, depois da sua instalação, você acabou reiniciando a máquina, vamos só validar pra ver que realmente o Vagrant tá instalado e tá apto a funcionar.

[02:31] Então pra fazer essa validação a gente vai digitar vagrant -h, lembrando que eu tô já com o Cmder aberto e eu naveguei até o diretório do meu projeto, se eu digitar aqui um dir ele me mostra que, basicamente, ele tem o zip e os arquivos que a gente extraiu.

[02:50] Só o último overview em relação a esses arquivos, eu tenho alguns scripts aqui nesse diretório, esses scripts vão pré-instalar algumas coisas pra gente. Basicamente a pré-instalação vai ser a instalação do Docker que é a instalação padrão dele. Se a gente editar o arquivo aqui do Docker a gente vai ver como é que ele é composto, é do próprio site do Docker mesmo o arquivo de instalação dele.

[03:17] E tem um arquivinho, esse aqui é um complemento da nossa instalação só pra mostrar pra vocês, a gente chama o arquivo de instalação, puxa uma imagem aqui que a gente vai entender mais pra frente pra que ela serve, e dá algumas permissões pros usuários.

[03:34] E tem um arquivinho aqui, que é o arquivo principal do nosso curso, entre aspas, que é como que o Jenkins é instalado. Então se a gente abrir aqui o arquivo de instalação a gente vai entender que é um arquivo bem simples, basicamente ele baixa a chave dos repositórios, a gente tá usando Ubuntu novamente, e faz um apt-get update e instala o Jenkins.

[03:56] Da onde veio esse script? Só por curiosidade, se a gente vir aqui no Google e digitar Jenkins Ubuntu, na própria wiki do Jenkins é o mesmo script que a gente tá trabalhando pra fazer a instalação. Então instalação feita aqui, agora a gente vai começar a colocar essa máquina pra rodar.

[04:22] Então pra fazer o Vagrant funcionar, a gente precisa primeiro fazer a instalação de um plugin pra que permita um tamanho de disco maior dentro do Vagrant. Esse plugin a gente instala com o seguinte comando: vagrant plugin install vagrant-disksize.

[04:41] Então a gente vem aqui no terminal e faz a instalação desse plugin. Esse plugin, enquanto ele instala, ele basicamente vai permitir que a gente coloque um parâmetro que aumente o tamanho do disco dentro do Vagrantfile, por default são 10GB, a gente vai colocar 30GB pra que caiba todos os nossos sistemas. Assim que terminar a instalação, a gente volta.

[05:02] Feita a instalação do plugin, a gente vai começar a provisionar a máquina com o seguinte comando: vagrant up. A partir de agora ele vai ler o Vagrantfile e vai fazer o provisionamento da máquina. Assim que a instalação terminar, a gente

volta.

[05:24] Bom, nossa instalação acabou, então agora a gente vai configurar a nossa máquina pra fazer a instalação do Jenkins. Então a primeira coisa que a gente vai fazer é logar na máquina utilizando o comando vagrant ssh, com esse comando ele vai fazer a autenticação no Linux utilizando uma chave que o próprio sistema criou pra gente.

[05:50] Então conectamos, a gente tá vendo aí que tem o IP da máquina, tem o quanto que ela tá utilizando de recurso. Vamos fazer uma validação rápida que é se o banco de dados foi instalado, então cabe um parênteses aí.

[06:04] O banco de dados a gente deixou pré-configurado no Vagrant pra que vocês não se preocupem com isso e que é uma especificidade da nossa aplicação. Nesse caso nós criamos a seguinte configuração: nós criamos dois usuários que é usuário devops e o usuário devops dev.

[06:25] Qual que é a diferença entre eles? Um vai acessar o banco de produção e o outro o banco de desenvolvimento. Os dois têm a mesma senha, que é a senha mestre, é uma senha que a gente vai utilizar no decorrer do curso pro banco de dados, e, pra esses usuários, eu também criei os bancos de dados respectivos, tanto de dev quanto de prod. Então vamos validar isso aqui. Primeira coisa, vamos validar se o MySQL foi instalado, ps -ef | grep -i mysql. Então nosso MySQL tá rodando, tá legalzinho.

[07:02] Agora vamos conectar no banco e testar se as permissões estão ok. Primeira coisa que a gente vai fazer: mysql -u devops -p, a senha mestre, conectamos o banco. Vamos, esse bios, as databases que a gente tem, show databases. O banco de dados todo tá criado. Então esse vai ser o nosso de produção.

[07:32] Vamos fazer o teste agora com o usuário devops_dev, com a mesma senha mestre. Vamos dar uma olhadinha nos bancos de dados. Eu tenho o todo dev e que é o banco de dados que vai receber os dados em desenvolvimento, e eu tenho um outro que é o test_todo_dev, por quê? Porque quando a gente for criar o nosso Pipeline o teste vai ser executado somente com o usuário de desenvolvimento, o nosso Pipeline, quando ele passar pelo teste em produção, não faz sentido eu testar novamente.

[08:09] Então agora que a gente criou e já fez a validação dos nossos dados, vamos instalar o Jenkins que é o ator principal do nosso treinamento, pra isso a gente vai no diretório vagrant/scripts\$. Esse /vagrant é automático, é uma configuração que vocês vão ter aí também. E dentro de scripts eu tenho o script jenkins.sh, então o que que eu vou fazer agora? Com o sudo eu vou rodar esse cara.

[08:44] Ele vai perguntar pra mim se eu quero fazer a instalação, ele atualizou os meus repositórios e aí a gente vai deixar ele instalar. A gente dá um ok e deixa ele instalar, assim que a instalação acabar a gente volta.

[09:00] Bom, a instalação acabou, agora tá na hora da gente acessar o Jenkins pela primeira vez. Pra isso a gente precisa primeiro descobrir qual que é o IP que a nossa máquina tem, com o seguinte comando: ip address. Esse aqui é o IP que a gente vai usar.

[09:16] O Jenkins sobe na porta 8080 como padrão, então pra acessar o nosso Jenkins a gente abre o nosso navegador e digita o IP, dois pontos e a porta de conexão. Legal, apareceu a tela do Jenkins, tá na hora da gente fazer a instalação dele, e é muito simples fazer. Ele pede uma chave de instalação e ele ainda fala onde essa chave tá pra gente, /var/lib/jenkins. A gente vai dar um cat nessa chave aqui. Essa aqui é a chave de instalação.

[09:58] Porque que eu usei o sudo? Por que esse arquivo tá protegido pra edição e leitura. Copio a chave, colo a chave aqui na senha de administrador e dou um continuar. Feito isso ele vai perguntar pra gente o seguinte: você quer instalar algum plugin ou quer instalar os plugins que são sugeridos?

[10:18] Vamos instalar os sugeridos porque alguns básicos já vem e a gente já consegue criar o nosso primeiro job. Então eu clico pra instalar os plugins sugeridos e agora a gente vai esperar a instalação terminar, demora uns minutinhos, e aí a gente

volta.

[10:33] Legal, instalação concluída dos plugins, ele pede pra criar um usuário administrador. Então o usuário vai ser o seguinte: o nome do usuário vai ser alura, a senha vai ser mestre123, eu tô criando mestre123, e não mestre, porque ele pede números e dígitos. A gente confirma a senha aqui também, mestre123, nome completo Aluno Alura.

[11:07] O e-mail é o e-mail pra notificações, a gente vai usar um e-mail de exemplo, você vai utilizar o seu e-mail ou o e-mail da sua companhia pra receber as notificações de acordo com job que você configurar. Aqui vai ser aluno@alura.com.br. Só uma ressalva, os plugins que nós vamos utilizar não enviaram e-mails, então fiquem sossegados se vocês não tiverem um e-mail pra colocar aqui.

[11:33] E agora a gente salva e continua, ele confirma pra gente que ele vai subir nesse IP e nessa porta, a gente salva e termina. Pronto, nosso Jenkins tá instalado e tá pronto pra utilizar.

[11:48] Nas próximas aulas a gente vai começar a configurar o nosso Jenkins e configurar os primeiros jobs.