

02

Rodando o Puppet automaticamente

Transcrição

Já tenho o meu Java, já tenho o meu Tomcat. Já tenho o MySQL, tudo instalado. Copiei meu arquivo `.war` e, inclusive, falei pra minha aplicação usar um ambiente – de desenvolvimento ou de produção, seja qual for – isto é, mudei uma variável de ambiente, editei um arquivo de configuração, usando aquele `file` do Puppet.

Claro, na documentação do Puppet eu consigo ver tudo o que tem lá: `exec`, `file`, criar diretório, o `user`, colocar chave de autenticação de ssh etc. Tem diversas coisas que eu consigo fazer dentro do meu `puppet file`.

Agora, o que eu queria fazer, que eu fiz mais ou menos, foi restartar o Tomcat. Quando eu restartei o Tomcat, eu fui lá e falei: `sudo service tomcat7 restart`. Mas calma aí... Quando eu faço essa alteração no arquivo de configuração do Tomcat, eu já podia restartar automaticamente. E a gente já viu isso.

Basta a gente falar: quando executar isso, `notify` o service do Tomcat 7. Com isso, quando eu fizer a alteração desse arquivo, ele deve restartar o serviço, e aí ele relê o arquivo de configuração, fica feliz e contente. Todo mundo feliz e contente.

Dou um `vagrant destroy`, agora que eu já fiz esse `notify` e, de novo, não preciso ser mão de vaca com o `vagrant destroy` e `vagrant up`. Posso dar à vontade.

Agora tem uma sacada ainda importante: até agora, a gente ficou nessa história de “dá um `vagrant up` e roda o puppet, dá um `vagrant up` e roda o puppet.” Só que eu queria só dar o `vagrant up`! E automaticamente ele já rodasse o puppet pra mim, de graça. Afinal, se eu estou criando a máquina, eu quero provisionar o software. Eu quero esse software na minha máquina!

Se eu quero esse software na minha máquina toda vez que eu crio uma máquina, então é bom que quando eu levante a máquina com `vagrant up` ele já rode o puppet. Ele já rode o `provisioning` do software.

Como é que eu vou fazer isso? Eu vou falar pra ele, quando eu configuro a minha máquina chamada “web”, que o meu `provision`, isso é, o `provision` é para usar um chamado – adivinha – `puppet`. E dentro dele, eu vou falar qual é o arquivo de manifest. O arquivo de manifest que eu vou usar é o `web.pp`.

Então, eu falo para ele:

- Olha, por favor, use o puppet e leia o arquivo `web.pp`. Toda vez que eu der um `vagrant up`, automaticamente ele vai rodar o passo de `provision`. E o passo de `provision` é rodar todos os `provision`s que estiverem aqui. Nesse caso só tem um `provision` escrito, que é o `puppet`. Maravilha! Ele roda o puppet pra gente e o arquivo `web.pp`.

```
Vagrant.configure("2") do |config|  
  config.vm.box = "hashicorp/precise32"  
  
  config.vm.define :web do |web_config|  
    web_config.vm.network "private_network", ip: "192.168.50.10"  
    web_config.vm.provision "puppet" do |puppet|  
      puppet.manifest_file = "web.pp"
```

```
end  
end
```

```
end
```

Toda vez que eu der `vagrant up`, a máquina está perfeita, linda, da maneira que eu queria usar.

Eu quero testar isso aí, é óbvio. Dou um `vagrant up`, ele levanta a máquina. Quando ele levanta a máquina, ele já roda todo o Puppet. E olha quanta coisa a gente já fez! A gente já deu um `apt update`, a gente já deu um `install jdk`, a gente já deu um `install` do Tomcat, a gente deu `install` do MySQL, a gente criou o banco no MySQL, alterou o arquivo de configuração do Tomcat, colocou o nosso war, restartou o serviço! Tudo isso a gente fez, e o `vagrant up` fez tudo isso pra gente. Quando eu dei o `vagrant up`, eu tenho uma máquina pronta pra eu usar.

E eu estou lá com o VRaptor Music Jungle pronto pra eu loggar. É claro, se eu quero testar tudo isso, eu vou lá no meu MySQL, depois de criar um usuário, e está lá no meu banco o usuário que eu acabei de criar. Eu fico feliz e contente.

Só o `vagrant up` levanta uma máquina completamente no estado que eu gostaria.

Não seja mão de vaca. Você pode dar o `vagrant destroy` e dar um `vagrant up` de novo.

Aqui, o que eu vou fazer? Vou dar um `require => Package[“tomcat7”]` e dar um `notify => Service[“tomcat7”]`. Dou um `destroy`, e falo: “Toda vez que você der um `vagrant up`, por favor, rode o `provision` do Puppet, no `web.pp`.

Agora é dar `vagrant up`. E esperar, esperar ele baixar tudo e esperar ele instalar tudo. Mas ele vai instalar tudo! Vai instalar tudo aquilo que a gente precisa e, quando ele chegar lá no final, e tiver instalado tudo, tudo, tudo, ele fala: terminei, e tua máquina está pronta!

`vagrant ssh`, eu estou com o MySQL todo configurado, posso ir no navegador, acessar o meu navegador lá no `192.168.50.10:8080`, Music Jungle, e cadastrar meu usuário que está lá no MySQL.

`vagrant up`: uma linha de código e eu tenho um ambiente igual ao de produção, igual ao de homologação, de desenvolvimento etc. Sempre querendo mudar uma coisinha ou outra, a gente pode através das configurações que a gente vai passando pro Puppet.

Legal? Vamos lá pros exercícios?