

O primeiro Playbook

Transcrição

Agora que já instalamos o Ansible, criamos a máquina virtual usando Virtualbox com Vagrant, além de gerarmos o primeiro arquivo de inventário para informarmos ao Ansible com quais servidores queremos trabalhar. Nós também testamos a instalação do Ansible usando a linha de comando para rodar o "Hello World".

Nosso próximo passo será montar o primeiro script. Como queremos chegar a uma instalação final de PHP, MySQL, Apache e rodar o Wordpress, executaremos uma série de comandos. Mas não faz sentido colocarmos tudo isso na linha de comando.

Um dos propósitos do Ansible é automatizar a infraestrutura e tornar simples a reprodução e a realização de testes. Desta forma, teremos a opção de colarmos em um controle de versão, trataremos o que fazemos na infraestrutura da mesma maneira como fazemos com o código de um serviço de uma aplicação normal.

Primeiramente vamos criar um script, que no Ansible é chamado de **playbook**. Clicaremos com o botão direito no menu, depois, selecionaremos "New File". O novo arquivo receberá o nome de `provisioning.yml`.

O formato `yml` é o padrão adotado pelo Ansible, para a criação dos arquivos de configuração. É um derivado de como criamos dicionários em Python e possui alguns detalhes que falaremos a seguir.

O arquivo sempre deve ser iniciado com **três hífen** (`---`) e o elemento de primeiro nível é sempre uma lista que começa com um filtro chamado `hosts`.

O filtro para você indicar quais hosts foram colocados no inventário, que será trabalhado pelo Ansible. Vamos rodar uma lista de comando a seguir, informando quais hosts serão usados.

```
- hosts: all
  tasks:
    - shell: 'echo hello > /vagrant/world.txt'
```

Observe que incluímos o campo `tasks`, com uma lista de comandos que serão aplicados nesses hosts filtrados.

Usaremos novamente o módulo `shell` para rodarmos um comando diferente. Nós iremos direcionar a palavra `hello` para dentro de um arquivo `word.txt`, o qual será criado dentro do diretório `/vagrant` na máquina virtual.

Nós usamos o comando Ansible para executar um módulo arbitrário, que foi o `shell` no exemplo. Mas para executar um `playbook`, digitaremos o comando diferente: `ansible-playbook`.

Lembrando que precisamos informar para o Ansible qual é o usuário, por isso, digitaremos na linha de comando `vagrant`.

Temos que informar o inventário, para relatar em quais máquinas ele vai executar o comando - no caso, é o arquivo `hosts` e informaremos a chave privada de autenticação `ssh`.

```
$ ansible-playbook provisioning.yml -u vagrant -i hosts --private-key .vagrant/machines/wordpress
```

No Ansible, se você ainda não estiver logado na máquina, o SSH vai pedir para você gravar a chave. Basta que você responda "sim" (digitando `yes`, em inglês). Em seguida, devemos esperar o Ansible terminar de executar o comando.

Veremos que o Ansible conseguiu executar um comando que alterou o estado da máquina com sucesso.

```
$ ansible-playbook provisioning.yml -u vagrant -i hosts --private-key .vagrant/machines/wordpress:
```

```
PLAY [all] *****
```

```
TASK [Gathering Facts] *****
```

```
The authenticity of hosts '172.17.177.40 (172.17.177.40)' cant't be established.  
ECDSA key fingerprint is SHA256:X+mM25xqAYN1JiBAoWJp//m1F6uBUvivGAPTT3CNbzw.  
Are you sure you want to continue connecting (yes/no)? yes  
ok: [172.17.177.40]
```

```
TASK [command] *****
```

```
changed: [172.17.177.40]
```

```
PLAY RECAP *****
```

```
172.17.177.40 : ok=2 changed=1 unreachable=0 failed=0
```

Em seguida, verificaremos se o script funcionou, logando na máquina virtual.

```
xxx-iMac: wordpress_com_ansible caelumrio$ vagrant ssh
```

E depois, vamos analisar o diretório `vagrant ssh` :

```
vagrant@vagrant-ubuntu-trusty-64:~$ ls /vagrant/  
hosts provisioning.yml Vagrantfile world.txt  
vagrant@vagrant-ubuntu-trusty-64:~$ cat /vagrant/world.txt  
hello  
vagrant@vagrant-ubuntu-trusty-64:~$ logout  
Connection to 127.0.0.1 closed.  
vagrant@vagrant_com_ansible caelumrio$ ls  
Vagrantfile      hosts              provisioning.yml    world.txt  
alurario$
```

Esse diretório Vagrant é especial e foi montado pelo Vagrant dentro da Virtualbox recém criada, que compartilhará todo conteúdo do projeto raiz (o arquivo Vagrantfile) com a máquina virtual. Isto ajudará você a carregar arquivo para dentro e fora da máquina virtual. Vimos que o arquivo `world.txt` foi criado, e o conteúdo dele é a palavra `hello`.

Com o comando `ls` conseguimos listar e ver que o arquivo `word` também existe na máquina de controle, porque este diretório é compartilhado entre a máquina hospedeira e sua máquina virtual.

Nós conseguimos provar que o script funcionou, criamos o próprio playbook, que criou um arquivo dentro da nossa máquina virtual - é possível vê-lo tanto dentro, quanto fora, devido ao tipo de funcionamento do Vagrant com Virtualbox.

Mas esse ainda não é o nosso objetivo, nós queremos instalar PHP, Apache e MySQL. Desta forma, rodaremos o `wordpress`. Faremos isto a seguir.

