

## Separando banco e aplicação

### Transcrição

Colocamos o Wordpress para funcionar, revimos os conceitos relativos à instalação de uma aplicação de três camadas. Agora, você tem o embasamento necessário para instalar e configurar aplicações semelhantes, que use um web server, um banco de dados e a sua aplicação.

O próximo passo será preparar o ambiente para que ele se assemelhe a um ambiente de produção. O primeiro passo será separar o banco de dados e a aplicação, ou seja, eles usarão servidores diferentes. Primeiramente, precisaremos ter duas máquinas virtuais - até o momento só temos uma. Por isso, teremos que alterar o `Vagrantfile`, que atualmente está assim:

```
Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/trusty64"

  config.vm.provider "virtualbox" do |v|
    v.memory = 1024
  end

  config.vm.define "wordpress" do |m|
    m.vm.network "private_network", ip: "172.17.177.40"
  end
end
```

Teremos que incluir uma máquina nova, que se diferenciará pelo `ip` e receberá o nome `mysql`:

```
Vagrant.configure("2") do |config|

  config.vm.box = "ubuntu/trusty64"

  config.vm.provider "virtualbox" do |v|
    v.memory = 1024
  end

  config.vm.define "wordpress" do |m|
    m.vm.network "private_network", ip: "172.17.177.40"
  end

  config.vm.define "mysql" do |m|
    m.vm.network "private_network", ip: "172.17.177.42"
  end
end
```

O próximo passo será modificar o arquivo `hosts`, onde colocamos todas as máquinas que desejamos configurar.

```
[wordpress]
172.17.177.40 ansible_user=vagrant ansible_ssh_private_key_file="/Users/marcoscropalato/wordpre:
```

```
[database]
172.17.177.42 ansible_user=vagrant ansible_ssh_private_key_file="/Users/marcoscropalato/wordpress
```

Como essa é uma máquina para instalar só banco de dados, não sendo uma máquina onde rodaremos Wordpress e o que colocamos, nós iremos separá-la em um grupo chamado `database`. A escolha será arbitrária, poderia ser qualquer outro. Enquanto isso, a outra máquina permanecerá a mesma. Ou seja, teremos no servidor que este no grupo `wordpress`, nós vamos aplicar a Web application, onde está o Apache, PHP e Wordpress.

No servidor onde encontraremos o MySQL, estará no grupo de `database`. Com isso, nós conseguimos criar a nossa máquina virtual, rodaremos o comando `vagrant up` e passaremos o nome da máquina `mysql` que criaremos:

```
xxx-iMac:wordpress_com_ansible caelumrio$ vagrant up mysql
Bringing machine mysql' up with 'virtual' provider...
==> mysql: Checking if box 'ubuntu/trusty64' is up to date...
==> mysql: A newer version of the box 'ubuntu?trusty64' is available! You currently
==> mysql: have version '20171129.0.0'. The latest is version '20171205.0.1'. Run
==> mysql: `vagrant box update` to update.
==> mysql: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> mysql: flag to force provisioning. Provisioners marked to run always will still run.
```

Após a criação, teremos que separar os playbooks. Ele está instalando tudo o que precisamos para ter uma aplicação de três camadas, nos servidores que forem encontrados. Agora, queremos que um servidor seja especializado no banco de dados, e outro foque na Web Application. Queremos criar um grupo de hosts novo, com o nome do grupo que colocamos no arquivo de hosts, no caso `database`.

E o que vamos instalar nesta máquina? Faremos exatamente o fizemos com o MySQL na outra máquina e configurar o banco de dados. Nós copiaremos os módulos necessários, para o MySQL no `Instala pacotes de dependência do sistema operacional`. No entanto, não precisaremos das dependências do PHP, mas `python-mysqldb` que é o módulo usado pelo Ansible para conseguir falar com MySQL. Precisaremos disso na máquina configurada.

Começaremos com as fazer os ajustes:

```
- hosts: database
  tasks:
    - name: 'Instala pacotes de dependencia do sistema operacional'
      apt:
        name: "{{ item }}"
        state: latest
      become: yes
      with_items:
        - mysql-server-5.6
        - python-mysqldb
```

Em seguida, precisaremos criar o banco de dados lógico e o usuário com as permissões corretas. Aproveitaremos o trecho do `Cria o banco do MySQL` e `Cria o usuário do MySQL`.

```
---
- hosts: database
  tasks:
```

```
-name: 'Instala pacotes de dependencia do sistema operacional'
apt:
  name: "{{ item }}"
  state: latest
became: yes
with_items:
  - mysql-server-5.6
  - python-mysqldb

- name: 'Cria o banco do MySQL'
mysql_db:
  name: wordpress_db
  login_user: root
  state: present

- name: 'Cria o usuário do MySQL'
mysql_user:
  login_user: root
  name: wordpress_user
  password:12345
  priv: 'wordpress_db.*:ALL'
  state: present

- hosts: all
handlers:
  -name: restart apache
  service:
    name: apache2
    state: restarted
  become: yes
```

E o que não queremos que aconteça na máquina, queremos evitar que os comandos de Wordpress e PHP Apache sejam executados nela. Para isto, precisaremos modificar o filtro `hosts` que no momento está configurado como `all` para `wordpress` - o nome do grupo criado no arquivo `hosts` com a finalidade de configurar o servidor original.

Se deixássemos com `all`, após a configuração do banco de dados, quando o trecho do arquivo for encontrado, ele configurará tudo novamente. Com isso, teríamos duas máquinas iguais.

```
- name: 'Cria o usuário do MySQL'
mysql_user:
  login_user: root
  name: wordpress_user
  password:12345
  priv: 'wordpress_db.*:ALL'
  state: present

- hosts: wordpress
handlers:
  -name: restart apache
  service:
    name: apache2
    state: restarted
  become: yes
```

Agora, quando ele encontrar os servidores database , vai executar os comandos relacionados, e os mesmos acontecerão ao encontrarem o wordpress . Então, os comandos relacionados também serão executados.

Revisão:

- Criamos a máquina virtual `mysql` no `Vagrantfile` ;
- Colocamos o IP, usuário e a chave privada da máquina criada no grupo `database` , no arquivo de inventário para o Ansible conseguir enxergá-la.
- Incluímos um grupo novo com **tarefas específicas para a instalação** do banco de dados no playbook. Neste processo, será feita a configuração da máquina.

O próximo passo será rodar o playbook e ver qual será o resultado. Como a máquina virtual é nova, ele solicitará que a chave SSH seja gravada. No Terminal, teremos o seguinte:

```
xxx-iMac:wordpress_com_ansible caelumrio$ ansible-playbook -i hosts provisioning.yml
```

```
PLAY [database] *****
```

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

```
The authenticity of hosts '172.17.177.42 (172.17.177.42)' cant be established.
```

```
ECDSA key fingerprint is SHA256:7jp5Cn61cGCGuG0f0wxCeqb6h+CA89IPooFYQQf5qY.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
ok: [172.17.177.42]
```

```
TASK [Instala pacotes de dependencia do sistema operacional]*****
```

```
◀────────────────────────────────────────────────────────────────────────────────▶
```

Depois disso, a instalação será iniciada, o download do MySQL será realizado e, em seguida, será a vez da configuração do banco de dados.

Com a conclusão da execução do playbook, nós veremos que as duas máquinas foram configuradas.

```
TASK [Configura Apache para servir o Wordpress] *****
```

```
ok: [172.17.177.40]
```

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

```
172.17.177.40      : ok=9    changed=2    unreachable=0    failed=0
```

```
172.17.177.42      : ok=4    changed=3    unreachable=0    failed=0
```

```
◀────────────────────────────────────────────────────────────────────────────────▶
```

Como a máquina virtual é nova, ele solicitará que a chave SSH seja gravada. Feito isso, vamos testar se o banco de dados foi instalado na máquina nova. O próximo passo será logar com o usuário e senha que estamos criando.

```
vagrant@vagrant-ubuntu-trusty-64:~$ mysql -u wordpress_user -p12345
```

Conseguimos conectar. Com o comando `database` conseguiremos listar os itens da lista.

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
+-----+
| information_schema |
| wordpress_db |
+-----+
2 rows in set (0.00 sec)
```

A nossa instalação de MySQL foi exitosa, no entanto, a máquina do Wordpress ficou com a instalação original.

```
tasks:
- name: 'Instala pacotes de dependencia do sistema operacional'
  apt:
    name: "{{ item }}"
    state: latest
  become: yes
  with_items:
    - php5
    - apache2
    - libapache2-mod-php5
    - php5-gd
    - libssh2-php
    - php5-mcrypt
    - mysql-server-5.6
    - python-mysqldb
    - php5-mysql
```

Removeremos o MySQL que restou na máquina, ela deve ficar limpa, tendo apenas o necessário para rodar o Wordpress. Veremos como isso será feito a seguir.