

## Criação do usuário no banco

### Transcrição

Vimos como é a criação do banco do MySQL, usando o módulo `mysql_db`, e começamos a mostrar como realizar a configuração do Ansible. O próximo passo será criar um usuário para o Wordpress se conectar no MySQL, veremos como fazer a seguir.

Mas existe um pequeno detalhe, não me lembro de memória, qual módulo é utilizado para administrar a parte de usuário do MySQL. Teremos que encontrar o meio para descobrir isso. Uma solução é **recorrermos a documentação do Ansible**.

Trata-se de um projeto bem documentado, que apresenta todas as descrições dos módulos de forma sucinta, com explicações das utilidades destes e exemplos.

Devemos descobrir como será feita essa parte de controle de usuário do MySQL. Primeiramente, verificaremos **qual é a versão do Ansible**, no caso, usamos `2.4`. É importante ficar atentos se estamos seguindo a versão adequada da documentação, porque alguns módulos mudam de uma versão para outra.

Inclusive, é possível encontrar **mudanças** na sintaxe de uma versão para outra. As mais atualizadas passam a dar suporte a características novas, por exemplo, adicionam algo mais no *template engine* ou alteram a forma de escrever o código.

Vale ressaltar que o nosso curso utiliza a versão `2.4` do Ansible. A seguir, procuraremos quais são os módulos que ajudam a administrar o MySQL. Na lista de módulos (seção [All Modules](http://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html) ([http://docs.ansible.com/ansible/latest/modules/list\\_of\\_all\\_modules.html](http://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html))), pesquisaremos por **"mysql"**. Encontraremos quatro:

- `mysql_db`
- `mysql_replication`
- `mysql_user`
- `mysql_variables`

O módulo `mysql_db` já foi usado, o `mysql_user` adiciona ou remove o usuário de um banco de dados MySQL - bem próximo da ação que desejamos realizar.

Se analisarmos mais profundamente a [documentação do mysql\\_user](http://docs.ansible.com/ansible/latest/modules/mysql_user_module.html#mysql-user-module) ([http://docs.ansible.com/ansible/latest/modules/mysql\\_user\\_module.html#mysql-user-module](http://docs.ansible.com/ansible/latest/modules/mysql_user_module.html#mysql-user-module)). Nela, encontraremos especificado a necessidade de termos instalado **MySQLdb** - fizemos essa instalação juntamente com MySQL. Depois, encontraremos o conteúdo de maior interesse: a lista com todos os parâmetros aceitos pelo módulo. Com isso, temos a oportunidade de entender qual é a sua utilidade.

No fim da documentação, encontraremos uma lista de exemplos de como se usa um determinado módulo. Vamos recapitular qual é o nosso objetivos, queremos criar um novo usuário, que tenha permissão para alterar um banco de dados criado por nós (`wordpress_db`). De acordo com o exemplo apresentado no material, veremos que ele cria um usuário de banco de dados e uma senha, com todos os privilégios do banco. Está próximo do que buscamos.

Usaremos o trecho como referência no nosso código em `provisioning.yml`, que é o uso de *task* Ansible normal. Abaixo de Cria o banco do MySQL :

```

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

- name: 'Cria o usuário do MySQL'
  mysql_user:
    name: bob
    password: 12345
    priv: ' *.*:ALL,GRANT'
    state: present

```

Lembre-se que o Ansible trabalha com `YML`, um formato muito **sensível a indentação**. Se você for fazer *copy/paste* da documentação, deve prestar atenção neste aspecto. Incluímos a tag `name` para informar a utilidade de criar o usuário do MySQL e o nome do módulo foi modificado para `wordpress_user`. Na linha referente à senha do usuário, deixaremos a senha simples `12345`, como foi usada no exemplo da documentação. Se optássemos por deixá-la vazia, o usuário ficaria sem senha.

Sobre as demais configurações, `priv` é referente aos privilégios do usuário, ou seja, o que ele pode fazer relacionado ao MySQL. O formato passado é `db.table.priv1.prive2`. Da forma como foi feita acima, definimos a permissão do usuário na qual ele poderá fazer todas as operações comuns, além de poder dar *grant* aos outros usuários - para todos os bancos de dados e tabelas, este não é nosso propósito. Este usuário tem o objetivo de fazer o Wordpress funcionar e não para administrar o banco de dados.

Então, definiremos `priv` com `ALL` para o banco `wordpress_db`, criado anteriormente. No `state`, manteremos `present`, utilizado na criação do usuário. Definiremos o usuário para autenticação como `root`, lembrando que este foi configurado sem senha, logo, será desnecessário informá-la aqui.

```

- 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

```

Agora estamos prontos para execução. De volta ao terminal, rodaremos o playbook informando o `hosts` e `provisioning.yml`.

```
$ ansible-playbook -i hosts provisioning.yml
```

Seremos informados que o Ansible conseguiu executar a alteração, conferiremos isso na nossa máquina virtual.

```
$ vagrant ssh
```

Depois, rodaremos o comando de autenticação do MySQL:

```
vagrant@vagrant-ubuntu-trusty-64:~$ mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.6.33-0ubuntu0.14.04.1 (Ubuntu)
```

Ele vai nos retornar a tabela com quatro elementos:

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| wordpress_db       |
+-----+
4 rows in set (0.00 sec)
```

Entraremos no banco `wordpress_db` que ainda não tem tabela criada.

```
mysql> use wordpress_db;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

Após executarmos `exit`, tentaremos autenticar com o usuário recém criado. Como ele tem senha, teremos que digitar `-p` e incluir em seguida a senha.

```
mysql> exit;
Bye
vagrant@vagrant-ubuntu-trusty-64:~$ mysql -u wordpress_user -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.6.33-0ubuntu0.14.04.1 (Ubuntu)
```

Veremos que a lista de acesso ao banco de dados será diferente.

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| wordpress_db       |
+-----+
```

```
+-----+  
2 rows in set (0.00 sec)
```

Vemos listados `information_schema` e `wordpress_db`, este será utilizado na administração e fará a instalação do `wordpress`.

Vimos como criar um novo usuário no MySQL, já temos a estrutura com o nosso banco de dados, usuário e as permissões corretas, vimos também como navegar pela documentação do Ansible para realizar tarefas com módulos que desconhecemos em profundidade, estamos prontos para o seguinte passo. Nós vamos instalar Wordpress mais adiante.