



Provisionamento com script – provisionamento na prática

Para fazer um provisionamento mais real podemos instalar um banco de dados e configura-lo no **Vagrantfile**. Acontece que **criar um banco de dados e adicionar configurações** é um processo que **precisa de mais linhas**, portanto, um **script**.

O **Vagrant** tem uma sintaxe para se trabalhar com **script**, é basicamente programar o **script** no **Vagrantfile** e chamar esse **script** no **provisionamento shell**:

```
$script_mysql = <<-SCRIPT
  apt-get update && \
  apt-get install -y mysql-server-5.7 && \
  mysql -e "create user 'devuser'@'%' identified by 'senha';"
SCRIPT

Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"
  config.vm.network "forwarded_port", guest: 80, host: 8090
  config.vm.network "public_network",
    ip: "192.168.100.41", bridge: "Realtek PCIe GbE Family Controller #2"

  config.vm.provision "shell",
    inline: "cat /configs/id_bionic.pub >> .ssh/authorized_keys"
  config.vm.provision "shell", inline: $script_mysql
```

- Os comandos shell no provisionamento **já elevam a permissão com "sudo"** então não tem necessidade de repetir.
- Toda vez que quiser usar quebra de linha tem que usar a `"\"`.
- O comando de **criação de usuário** é da **documentação do mysql-server** mas do jeito que está aí **ele permite a conexão com o banco a partir de qualquer host usando a senha," senha"**.

Para provar que deu certo eu vou entrar na **máquina virtual**, me conectar com o **mysql** e **fazer um select dos user**:

```
vagrant@vagrant:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select user from mysql.user;
-> ;
+-----+
| user                |
+-----+
| devuser              |
| debian-sys-maint     |
| mysql.session        |
| mysql.sys            |
| root                 |
+-----+
5 rows in set (0.00 sec)
```

Da maneira que está instalado, o **mysql** só poderia ser acessado pela **máquina virtual**. Isso em função do **bind address**, quando o servidor sobe o processo o **mysql** fica associado com esse **IP**, dá pra checar isso no arquivo **mysqld.cnf**:

```
vagrant@vagrant:~$ cat /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# * Basic Settings
#
user                = mysql
pid-file            = /var/run/mysqld/mysqld.pid
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir     = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 127.0.0.1
#
```

Da mesma maneira que podemos copiar arquivos da **pasta compartilhada** da **máquina local** para a **virtual** também podemos fazer o inverso, então basta **copiar esse arquivo** para a **máquina local**:

```
vagrant@vagrant:~$ cat /etc/mysql/mysql.conf.d/mysqld.cnf >> /configs/mysqld.cnf
vagrant@vagrant:~$
```



Com isso eu posso **editar o bind-address** do arquivo para algum outro IP, então configurar o **Vagrantfile** **provisionando a copia desse arquivo para a máquina virtual** e dessa maneira automatizar essa configuração:

```
40 #
41 # Instead of skip-networking the default is now to listen only on
42 # localhost which is more compatible and is not less secure.
43 bind-address         = 0.0.0.0
44 #
```

```
config.vm.provision "shell",
  inline: "cat /configs/mysqld.cnf > /etc/mysql/mysql.conf.d/mysqld.cnf"
config.vm.provision "shell",
  inline: "service mysql restart"

config.vm.synced_folder "./configs", "/configs"
config.vm.synced_folder ".", "/vagrant", disabled: true
end
```

- Usando apenas um **>** eu estou **excluindo o conteúdo que existe no destino** e copiando o conteúdo que estou passando, usando **>>** eu **mantenho o conteúdo original do destino** e adiciono o que estou copiando
- O comando de **restart** é necessário para atualizar as configurações

Dando uma conferida no **bind-address** da **máquina virtual**:

```
vagrant@vagrant:~$ cat /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
#  
# The MySQL database server configuration file.
```

```
#  
# You can copy this to one of:  
# - "/etc/mysql/my.cnf" to set global options,  
# - "~/.my.cnf" to set user-specific options.  
#  
#
```

```
# Instead of skip-networking the default is now to listen only on  
# localhost which is more compatible and is not less secure.
```

```
bind-address            = 0.0.0.0  
#
```