



## Provisionamento de máquina virtual - Network public

O **vagrant** pode configurar a máquina virtual em um nível digamos mais “**cru**”. Ele por si só não faz configurações mais detalhadas do ambiente, como instalar softwares na máquina virtual como **nginx** ou **algum banco de dados**, pra isso é necessário a utilização de **scripts** de **provisionamento**.

**Provisionamento** nesse contexto é justamente isso, preparar configurações mais detalhadas da **máquina virtual** para deixá-la pronta para seu determinado uso.

Quando falei sobre “**scripts**” eu quis dizer sobre o código que irá ser executado na máquina, por exemplo, na configuração de provisionamento do **VagrantFile** eu posso dizer qual código vai ser utilizado e onde, aqui no caso vai ser no **shell** (*shell é onde se executa comandos no linux*):

```
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 PC
  config.vm.provision "shell", inline: "echo hello world"
end
```

Quando subimos a **máquina virtual** ele **informa que ela já foi provisionada** com as instruções (*nesse caso é só um echo, então ele só mostra no terminal DA MÁQUINA VIRTUAL o hello world*);

```
lucas@DESKTOP-A5T3N3N MINGW64 ~/Desktop/ambiente_dev/Bionic
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'hashicorp/bionic64' version '1.0.282' is up to date...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: bridged
==> default: Forwarding ports...
    default: 80 (guest) => 8090 (host) (adapter 1)
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection reset. Retrying...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default: Guest Additions Version: 6.0.10
    default: VirtualBox Version: 6.1
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
    default: /vagrant => C:/Users/lucas/Desktop/ambiente_dev/Bionic
==> default: Machine already provisioned. Run 'vagrant provision' or use the '--provision
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```

Existe também o comando “**vagrant provision**” que vai executar somente o provisionamento da **máquina virtual**, ou seja, tudo que foi provisionado.

Pra deixar isso mais evidente dá pra alterar o script de provisionamento para que ele desvie a saída para um arquivo txt:

```
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 PC
  config.vm.provision "shell", inline: "echo hello world >> hello.txt"
end

lucas@DESKTOP-A5T3N3N MINGW64 ~/Desktop/ambiente_dev/Bionic
$ vagrant provision
==> default: Running provisioner: shell...
    default: Running: inline script
```

Se eu entrar na **máquina virtual** e pegar o conteúdo do arquivo, vai ta escrito o que foi provisionado:

```
lucas@DESKTOP-A5T3N3N MINGW64 ~/Desktop/ambiente_dev/Bionic
$ vagrant ssh
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Feb  9 19:46:15 UTC 2022

System load:  0.09          Processes:            94
Usage of /:   2.6% of 61.80GB Users logged in:          0
Memory usage: 11%          IP address for eth0: 10.0.2.15
Swap usage:   0%           IP address for eth1: 192.168.100.41

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.
   https://ubuntu.com/blog/microk8s-memory-optimisation

255 packages can be updated.
198 updates are security updates.

Last login: Tue Feb  8 17:41:34 2022 from 192.168.100.23
vagrant@vagrant:~$ ls
hello.txt  id_bionic.pub
vagrant@vagrant:~$ cat hello.txt
hello world
vagrant@vagrant:~$
```

A ideia com isso é **automatizar qualquer instalação e também comandos que sejam necessários serem feitos dentro da máquina virtual** para que eles executem na hora que ela for levantada. Então pode ser instalar qualquer software, atualizar qualquer pacote necessário, popular base, abrir portas de conexão e etc...