



Criar várias máquinas a partir de um único file – Multi machine

Se eu quiser subir várias **máquinas virtuais** com um único **Vagrantfile** eu posso. Essa parte se chama multi machine e faz necessário que criemos um **“bloco”** dentro da **configuração principal do vagrant (vagrant.configure)**.

Esses blocos representam **máquinas virtuais** únicas que tem como propriedade **“cabeçalho”** a **config.vm.define**, repare que tudo **que é configuração da máquina criada** agora é **referenciada pelo nome a ela atribuído (mysql)**:

```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"

  config.vm.define "mysqldb" do |mysql|
    mysql.vm.network "forwarded_port", guest: 80, host: 8090
    mysql.vm.network "public_network",
      ip: "192.168.100.41", bridge: "Realtek PCIe GbE Family Controller #2"

    mysql.vm.provision "shell",
      inline: "cat /configs/id_bionic.pub >> .ssh/authorized_keys"
    mysql.vm.provision "shell", inline: $script_mysql
    mysql.vm.provision "shell",
      inline: "cat /configs/mysqld.cnf > /etc/mysql/mysql.conf.d/mysqld.cnf"
    mysql.vm.provision "shell", inline: "service mysql restart"

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

Pra fixar mais ainda o conceito eu vou meter outra **máquina virtual** nesse meio, uma máquina de **“devweb”**. Repare que agora eu tenho dois **“blocos”** de máquinas virtuais dentro da **config principal do vagrant**, essa mesma config que define o box sendo como um **ubuntu (hashicorp/bionic64)**.

Então ambas as máquinas **vão ser do mesmo box** devido ao **escopo**, porém suas próprias configurações serão especializadas:

```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"

  config.vm.define "mysqldb" do |mysql|
    mysql.vm.network "public_network",
      ip: "192.168.100.41", bridge: "Realtek PCIe GbE Family Controller #2"

    mysql.vm.provision "shell",
      inline: "cat /configs/id_bionic.pub >> .ssh/authorized_keys"
    mysql.vm.provision "shell", inline: $script_mysql
    mysql.vm.provision "shell",
      inline: "cat /configs/mysqld.cnf > /etc/mysql/mysql.conf.d/mysqld.cnf"
    mysql.vm.provision "shell", inline: "service mysql restart"

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

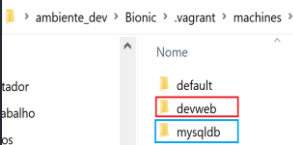
  config.vm.define "devweb" do |devweb|
    devweb.vm.network "forwarded_port", guest: 80, host: 8090
    devweb.vm.network "public_network",
      ip: "192.168.100.41", bridge: "Realtek PCIe GbE Family Controller #2"
  end
end
```

Rodando um “**vagrant status**” já é possível ver as duas máquinas configuradas, ambas em estado de “**não criada**”:

```
lucas@DESKTOP-A5T3N3N MINGW64 ~/Desktop/ambiente_dev/Bionic
$ vagrant status
Current machine states:

mysqlb                          not created (virtualbox)
devweb                          not created (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
```



A partir de agora todos os comandos que não forem especificados para uma das máquinas irá executar em todas.