(CSC-05) Lab 2 - Parte A: Virtualbox & Vagrant

Neste relatório serão descritos os passos realizados ao seguir o roteiro disponível para o *Laboratório 2 - Parte A: Virtualbox e Vagrant*.

A execução do laboratório foi gravada no formato *asciinema* (gravação do terminal, sem intervalo entre comandos). Para cada subseção à seguir, serão disponibilizados os links referentes as gravações.

O Laboratório foi executado no seguinte ambiente:

- **Sistema Operacional**: Arch Linux x64 (5.8.8-arch1-1)
- **CPU**: Intel i7-6500U
- **RAM**: 8 GB
- **GPU**: Intel Skylake GT2 [HD Graphics 520]

Tabela de Conteúdos:

Parte I: Introdução

Gravação:

- Asciinema: https://asciinema.org/a/359506

Execução passo-a-passo:

A seguir, a execução dos passos do laboratório será seguida por *screenshots* mostrando o resultado obtido para cada passo.

3) Verificar qual a versão do vagrant está instalada (no powershell ou cmder)

```
# vagrant version
```

No caso executei o Laboratório em um ambiente Linux (Arch Linux x64) e portanto utilizei uma linha de comando para o ambiente Linux.

```
gitoso@LOVELACE ~/VirtualBox VMs vagrant version
Installed Version: 2.2.10
Latest Version: 2.2.10

You're running an up-to-date version of Vagrant!
```

4) Verificar as "boxes" instaladas

```
# vagrant box list
```

```
gitoso@LOVELACE ~/VirtualBox VMs vagrant box list
There are no installed boxes! Use `vagrant box add` to add some.
```

5) Criar um diretório com o nome "ubuntu18" e entrar nesse diretório

```
# cd Documentos
# mkdir ubuntu18
# cd ubuntu18
```

```
gitoso@LOVELACE ~/VirtualBox VMs mkdir ubuntu18
gitoso@LOVELACE ~/VirtualBox VMs cd ubuntu18
```

6) Criar um arquivo de inicialização do Vagrantfile

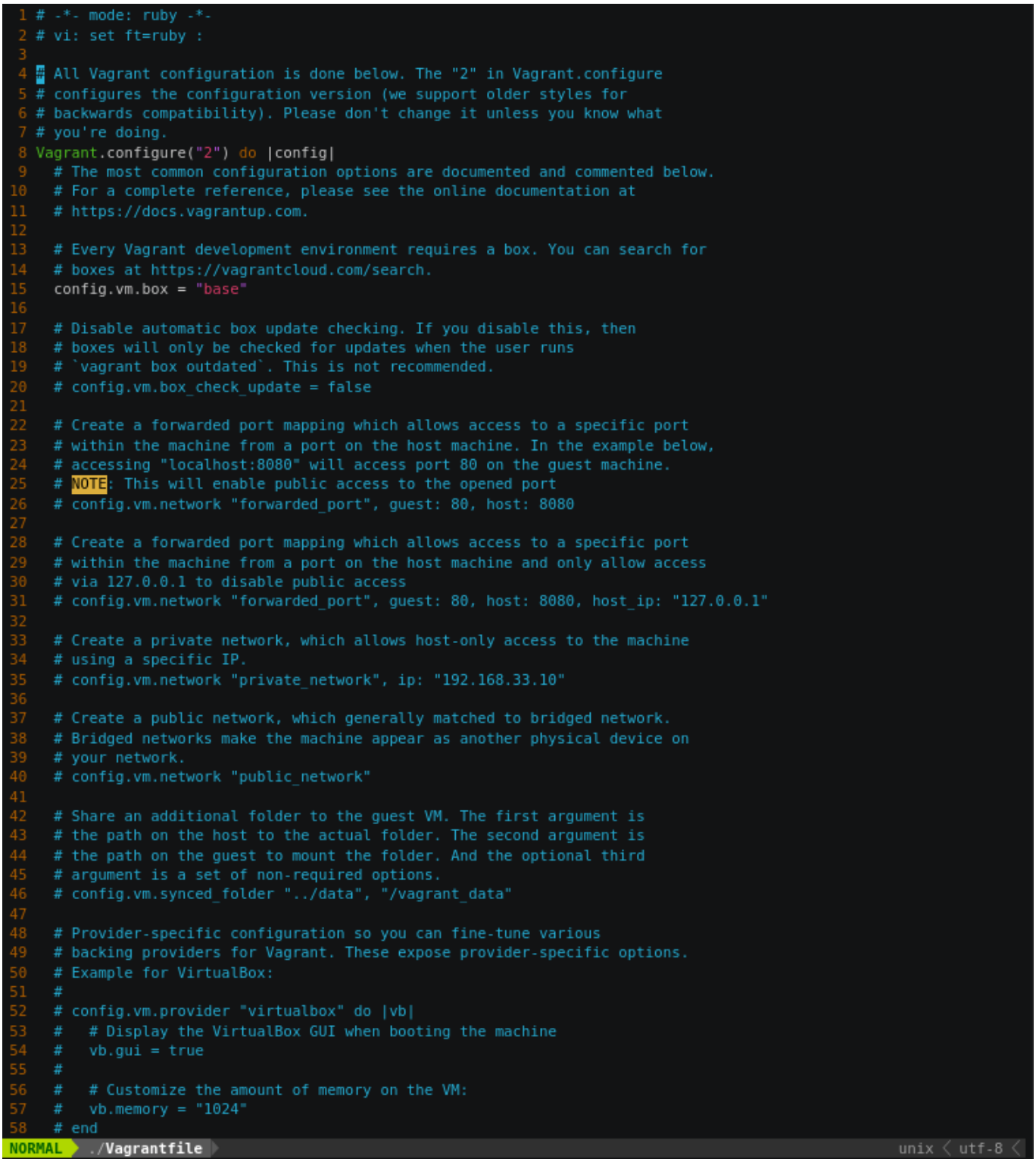
```
# vagrant init
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

7) Abrir em um editor de texto o arquivo Vagrantfile criado, entender

---

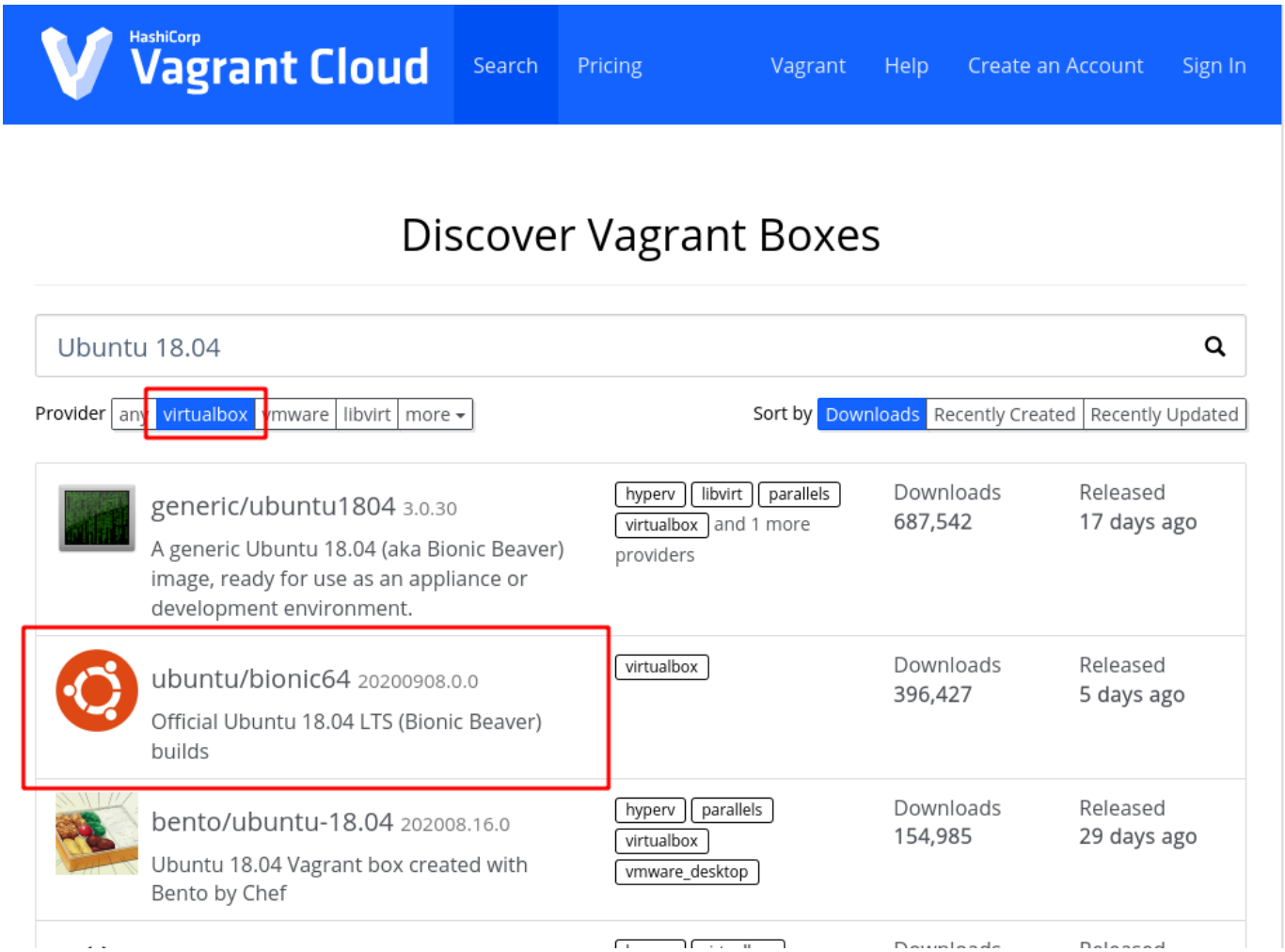
Segue abaixo uma imagem do conteúdo original do arquivo `Vagrantfile` estudado (aberto no editor `vim`)



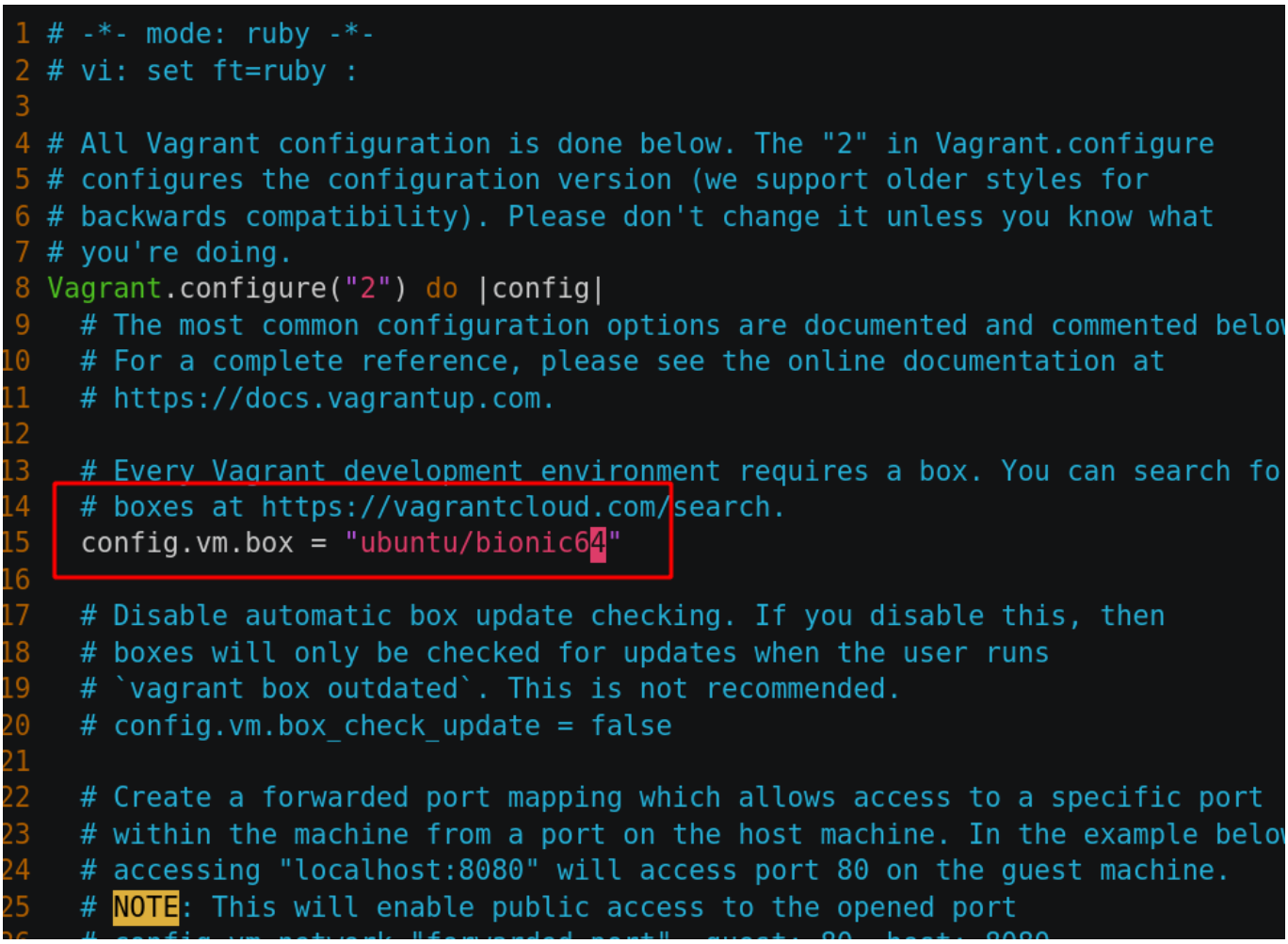
```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "base"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
  # config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  # config.vm.network "public_network"

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  # config.vm.synced_folder "../data", "/vagrant_data"

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
  # Example for VirtualBox:
  #
  # config.vm.provider "virtualbox" do |vb|
  #   # Display the VirtualBox GUI when booting the machine
  #   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
  #   vb.memory = "1024"
  # end
NORMAL  ./Vagrantfile                                                    unix  utf-8
```

8) Procurar por uma imagem útil para uso no VagrantCloud:

Link para o VagrantCloud: https://app.vagrantup.com/boxes/search



9) Usando editor de texto, modificar o "box" para ubuntu/bionic64



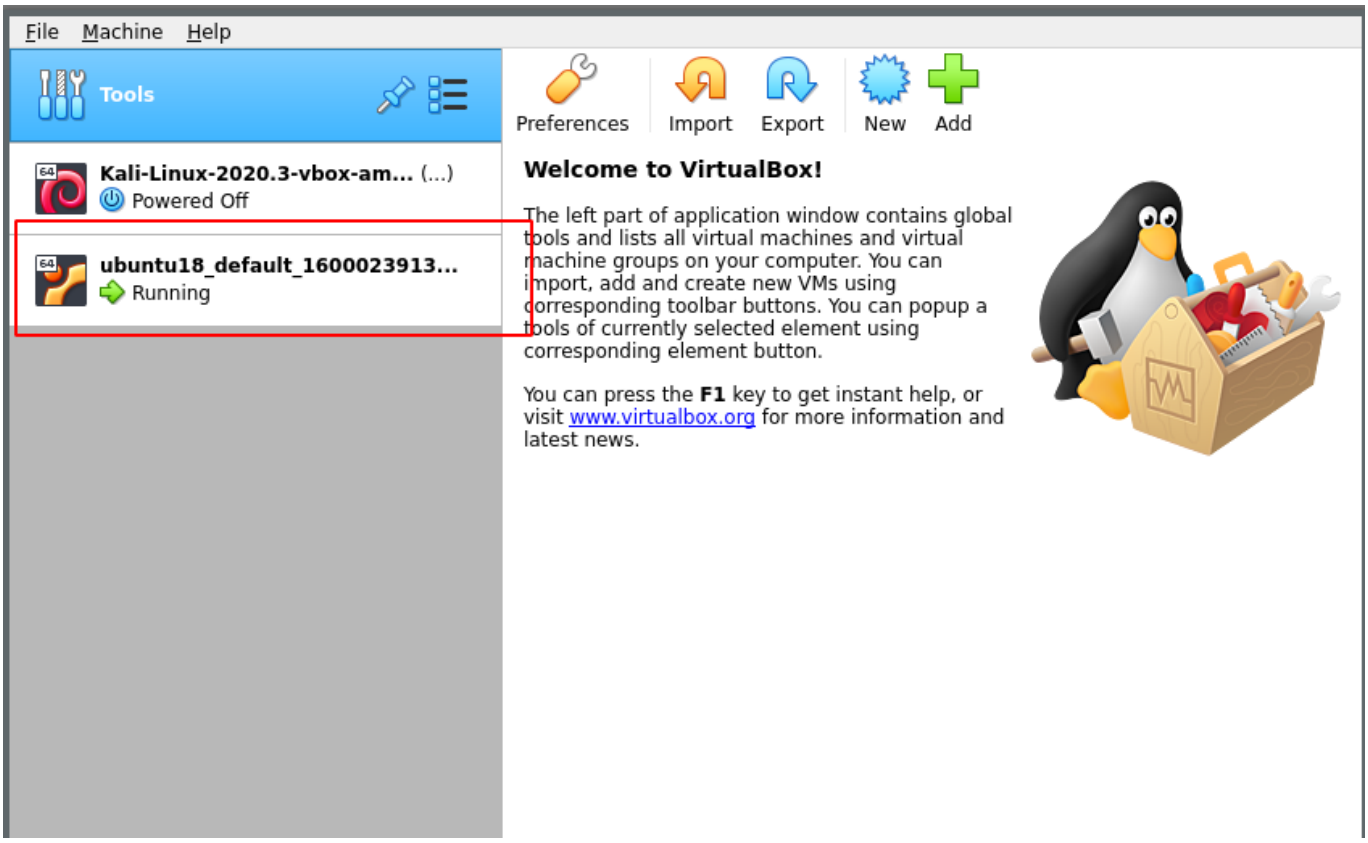
```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented belo
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search fo
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/bionic64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example belo
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
```

10) Após finalizar a edição, execute o comando, dentro da pasta ubuntu18

```
# vagrant up
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/bionic64' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'ubuntu/bionic64'
    default: URL: https://vagrantcloud.com/ubuntu/bionic64
==> default: Adding box 'ubuntu/bionic64' (v20200908.0.0) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/ubuntu/boxes/bionic64/versions/20200908.0.0/providers/virtualbox.
Download redirected to host: cloud-images.ubuntu.com
==> default: Successfully added box 'ubuntu/bionic64' (v20200908.0.0) for 'virtualbox'!
==> default: Importing base box 'ubuntu/bionic64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/bionic64' version '20200908.0.0' is up to date...
==> default: Setting the name of the VM: ubuntu18_default_1600025261388_59954
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> 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: Warning: Remote connection disconnect. Retrying...
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> 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:
    default: Guest Additions Version: 5.2.42
    default: VirtualBox Version: 6.1
==> default: Mounting shared folders...
    default: /vagrant => /home/gitoso/VirtualBox VMs/ubuntu18
```

11) Entre no virtualbox e verifique que sua maquina virtual esta rodando.



12) Vamos logar na maquina agora via SSH

```
# vagrant ssh
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant ssh
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-117-generic x86_64)

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

  System information as of Sun Sep 13 19:28:31 UTC 2020

  System load:  1.15              Processes:             99
  Usage of /:   10.5% of 9.63GB   Users logged in:       0
  Memory usage: 12%               IP address for enp0s3: 10.0.2.15
  Swap usage:   0%


0 packages can be updated.
0 updates are security updates.


vagrant@ubuntu-bionic:~$ #
```

13) Dentro da máquina virtual verifique diversas características da VM

```
# free -h
# df -h
# sudo fdisk -l
```

```
vagrant@ubuntu-bionic:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           984M         77M        662M        596K        245M        768M
Swap:            0B          0B          0B
vagrant@ubuntu-bionic:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            481M     0  481M   0% /dev
tmpfs            99M  596K   98M   1% /run
/dev/sda1       9.7G  1.1G  8.7G  11% /
tmpfs           493M     0  493M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           493M     0  493M   0% /sys/fs/cgroup
vagrant         185G  177G  8.2G  96% /vagrant
tmpfs            99M     0   99M   0% /run/user/1000
vagrant@ubuntu-bionic:~$ sudo fdisk -l
Disk /dev/sda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x010d39dc

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 20971486 20969439  10G 83 Linux


Disk /dev/sdb: 10 MiB, 10485760 bytes, 20480 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

14) Dentro da maquina virtual atualize o sistema e instale o htop

```
# sudo apt update
# sudo apt upgrade
# sudo apt install htop
```

```
vagrant@ubuntu-bionic:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [849 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [701 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [232 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [8512 B]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2908 B]
Get:11 http://archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1072 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [356 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1107 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [344 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [19.6 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6768 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [7516 B]
Get:21 http://archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [7736 B]
Get:23 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B]
Fetched 18.7 MB in 9s (2119 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
vagrant@ubuntu-bionic:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vagrant@ubuntu-bionic:~$ sudo apt install htop
Reading package lists... Done
Building dependency tree
Reading state information... Done
htop is already the newest version (2.1.0-3).
htop set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
vagrant@ubuntu-bionic:~$
```

Executando o `htop` só para verificar:

```
  1  [                                                       0.0%]   Tasks: 27, 19 thr; 1 running
  2  [*                                                      0.7%]   Load average: 0.46 0.29 0.11
  Mem[|||||#************************                  79.8M/985M]   Uptime: 00:02:23
  Swp[                                                     0K/0K]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 1912 vagrant    20   0  105M  5364  4220 S  0.0  0.5  0:00.24 sshd: vagrant@pts/0
 2212 vagrant    20   0 26124  4324  3192 R  0.0  0.4  0:00.04 htop
    1 root       20   0 77832  8640  6436 S  0.0  0.9  0:01.38 /sbin/init
  401 root       19  -1 86628  8152  7456 S  0.0  0.8  0:00.23 /lib/systemd/systemd-journald
  425 root       20   0 43752  4668  3112 S  0.0  0.5  0:00.57 /lib/systemd/systemd-udevd
  427 root       20   0 97708  1848  1672 S  0.0  0.2  0:00.00 /sbin/lvmetad -f
  629 systemd-n  20   0 80080  5220  4620 S  0.0  0.5  0:00.04 /lib/systemd/systemd-networkd
  654 systemd-r  20   0 70660  5460  4896 S  0.0  0.5  0:00.04 /lib/systemd/systemd-resolved
  819 daemon     20   0 28332  2396  2192 S  0.0  0.2  0:00.00 /usr/sbin/atd -f
  836 root       20   0  281M  6752  5884 S  0.0  0.7  0:00.01 /usr/lib/accountsservice/accounts-daemon
  857 root       20   0  281M  6752  5884 S  0.0  0.7  0:00.00 /usr/lib/accountsservice/accounts-daemon
  832 root       20   0  281M  6752  5884 S  0.0  0.7  0:00.04 /usr/lib/accountsservice/accounts-daemon
  838 root       20   0 31748  3204  2908 S  0.0  0.3  0:00.00 /usr/sbin/cron -f
  859 root       20   0  157M  1680  1548 S  0.0  0.2  0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
  860 root       20   0  157M  1680  1548 S  0.0  0.2  0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
  841 root       20   0  157M  1680  1548 S  0.0  0.2  0:00.00 /usr/bin/lxcfs /var/lib/lxcfs/
  842 messagebu  20   0 50148  4480  3784 S  0.0  0.4  0:00.08 /usr/bin/dbus-daemon --system --address=systemd: --no
  858 root       20   0 70612  6028  5280 S  0.0  0.6  0:00.03 /lib/systemd/systemd-logind
  877 root       20   0  107M  2096  1876 S  0.0  0.2  0:00.00 /usr/sbin/irqbalance --foreground
  871 root       20   0  107M  2096  1876 S  0.0  0.2  0:00.01 /usr/sbin/irqbalance --foreground
 1058 root       20   0  166M 17212  9440 S  0.0  1.7  0:00.00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-s
  894 root       20   0  166M 17212  9440 S  0.0  1.7  0:00.13 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-s
  920 syslog     20   0  256M  4384  3612 S  0.0  0.4  0:00.00 /usr/sbin/rsyslogd -n
  921 syslog     20   0  256M  4384  3612 S  0.0  0.4  0:00.00 /usr/sbin/rsyslogd -n
  922 syslog     20   0  256M  4384  3612 S  0.0  0.4  0:00.00 /usr/sbin/rsyslogd -n
  897 syslog     20   0  256M  4384  3612 S  0.0  0.4  0:00.02 /usr/sbin/rsyslogd -n
 1059 root       20   0  183M 20072 12160 S  0.0  2.0  0:00.00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
  925 root       20   0  183M 20072 12160 S  0.0  2.0  0:00.18 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
  950 root       20   0  282M  6460  5676 S  0.0  0.6  0:00.00 /usr/lib/policykit-1/polkitd --no-debug
  952 root       20   0  282M  6460  5676 S  0.0  0.6  0:00.00 /usr/lib/policykit-1/polkitd --no-debug
  940 root       20   0  282M  6460  5676 S  0.0  0.6  0:00.02 /usr/lib/policykit-1/polkitd --no-debug
  943 root       20   0 16412  2332  2188 S  0.0  0.2  0:00.00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,960
  955 root       20   0 14888  1912  1780 S  0.0  0.2  0:00.01 /sbin/agetty -o -p -- \u --noclear tty1 linux
 1011 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.00 /usr/sbin/VBoxService
 1012 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.00 /usr/sbin/VBoxService
 1013 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.02 /usr/sbin/VBoxService
 1014 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.00 /usr/sbin/VBoxService
 1015 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.00 /usr/sbin/VBoxService
 1016 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.00 /usr/sbin/VBoxService
 1017 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.01 /usr/sbin/VBoxService
 1010 root       20   0  241M  2064  1636 S  0.0  0.2  0:00.03 /usr/sbin/VBoxService
 1120 root       20   0 72300  5724  4992 S  0.0  0.6  0:00.00 /usr/sbin/sshd -D
 1830 root       20   0  105M  7188  6180 S  0.0  0.7  0:00.01 sshd: vagrant [priv]
 1832 vagrant    20   0 76664  7584  6564 S  0.0  0.8  0:00.02 /lib/systemd/systemd --user
 1833 vagrant    20   0  109M  2372    12 S  0.0  0.2  0:00.00 (sd-pam)
 1913 vagrant    20   0 23072  4972  3348 S  0.0  0.5  0:00.20 -bash
```

15) Saia da maquina guest VM de volta para o host

```
# exit
```

```
vagrant@ubuntu-bionic:~$ exit
logout
Connection to 127.0.0.1 closed.
```

16) Verifique o estado da VM perante o vagrant

```
# vagrant status
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```

17) Verificar as boxes instaladas

```
# vagrant box list
```

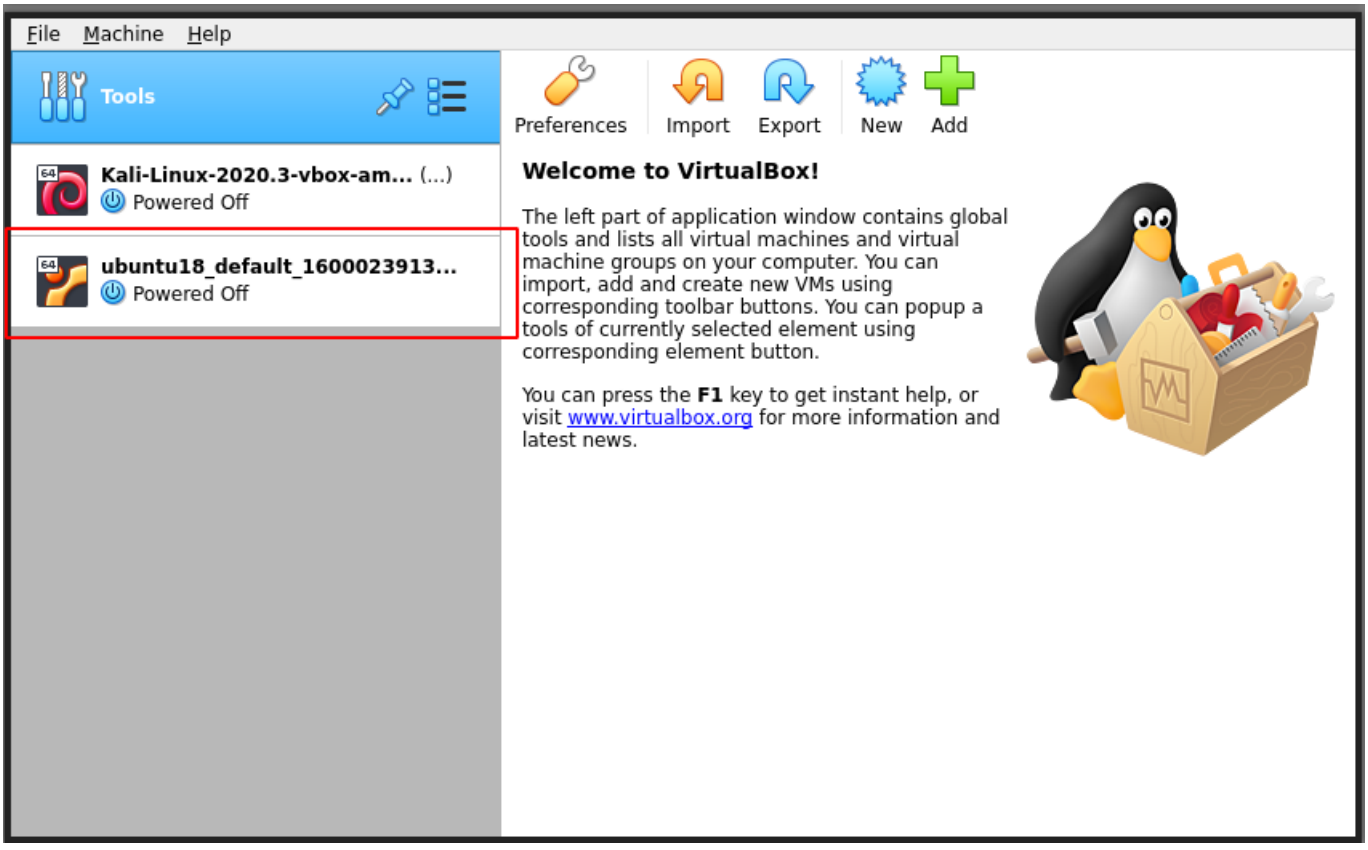
```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant box list
ubuntu/bionic64 (virtualbox, 20200908.0.0)
```

18) Desligue a maquina virtual de maneira "suave"

```
# vagrant halt
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant halt
==> default: Attempting graceful shutdown of VM...
```

Verificando o estado da VM no VirtualBox:



19) Destrua / Limpe a VM e a box dentro dela, liberando espaço e recursos

```
# vagrant destroy
# vagrant box list
```

```
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
gitoso@LOVELACE ~/VirtualBox VMs/ubuntu18 vagrant box list
ubuntu/bionic64 (virtualbox, 20200908.0.0)
```

Parte II: VM Windows

Gravação

- Asciinema: https://asciinema.org/a/359543

A seguir, a execução dos passos do laboratório será seguida por *screenshots* mostrando o resultado obtido para cada passo.

Execução passo-a-passo

20) Crie um diretório "windows2010" em Documentos

```
# cd Documentos
# mkdir windows2010
# cd windows2010
```

```
gitoso@LOVELACE ~/VirtualBox VMs mkdir windows2010
gitoso@LOVELACE ~/VirtualBox VMs cd windows2010
```

21) Crie um arquivo Vagrantfile do zero contendo o seguinte código

```
Vagrant.configure("2") do |config|
    config.vm.box = "tas50/windows_10"
    config.vm.hostname = "server"
    config.vm.provider "virtualbox" do |vb|
        vb.gui = false
        vb.memory = 2048
        vb.cpus = 2
    end
    config.vm.network :private_network, ip: "172.16.2.20"
    config.vm.provision "shell", path: "rdp.ps1"
end
```

```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vim Vagrantfile
```

```
Vagrant.configure("2") do |config|
  config.vm.box = "tas50/windows_10"
  config.vm.hostname = "server"
  config.vm.provider "virtualbox" do |vb|
    vb.gui = false
    vb.memory = 2048
    vb.cpus = 2
  end
  config.vm.network :private_network, ip: "172.16.2.21" # 21 >> 20
  config.vm.provision "shell", path: "rdp.ps1"
end
~
~
~
```

(Alterei o IP para terminar com 21)

22) Crie um segundo arquivo de texto com o nome "rdp.ps1" e coloque no mesmo diretório

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal
Server' -name "fDenyTSConnections" -value 0
Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
Set-TimeZone 'E. South America Standard Time'
```

```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vim rdp.ps1
```

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -name "fDenyTSConnections" -valu
Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
Set-TimeZone 'E. South America Standard Time'
~
~
~
```

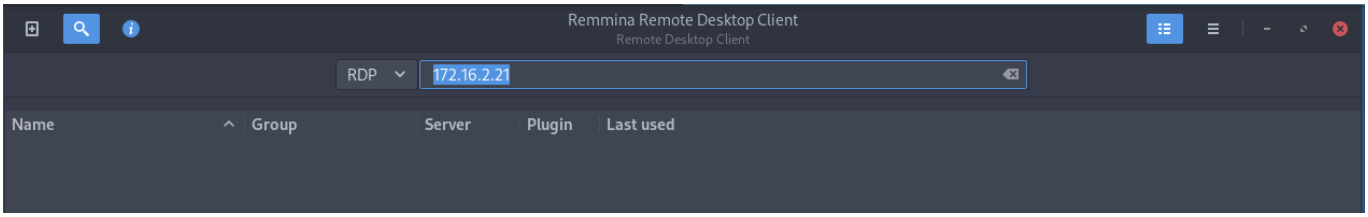
23) Verifique o virtualbox novamente, veja a VM Windows rodando. A opção "vb.gui=false" indica a inicialização "headless". O script "rdp.ps1" habilita a possibilidade de acessar a maquina virtual via Remote Desktop. Se você tiver instalar "Remote Desktop" tente acessar a VM. Se não for possível usar o Remote Desktop, acessar a VM diretamente via Virtualbox.

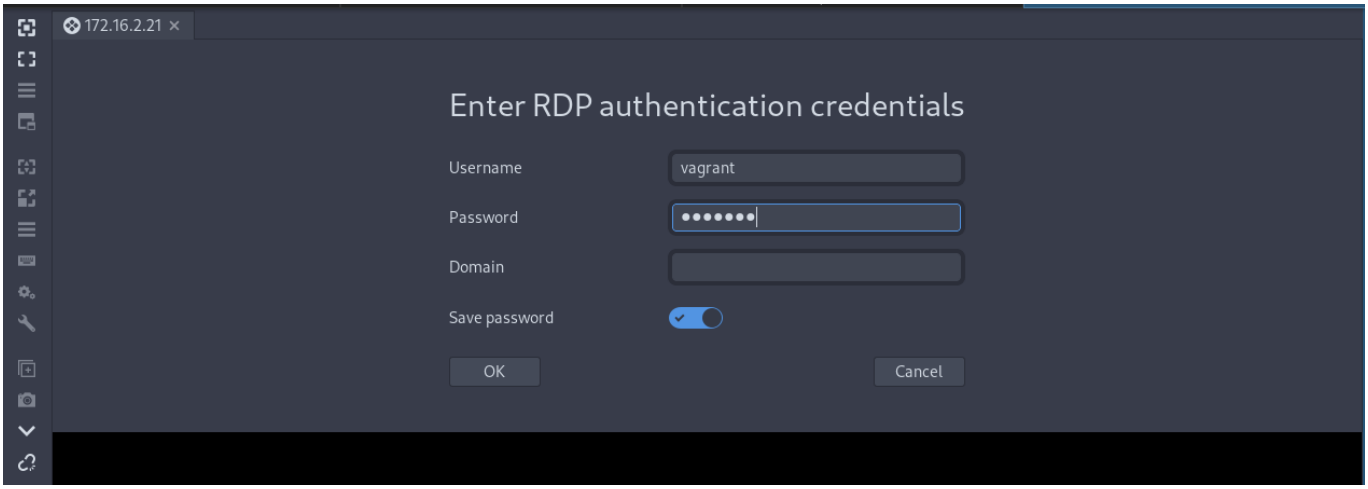
```
# endereço IP: 172.16.2.20
# porta default: 3389
# login/senha: vagrant / vagrant
```

Primeiramente inicializei a VM com o comando `vagrant up`

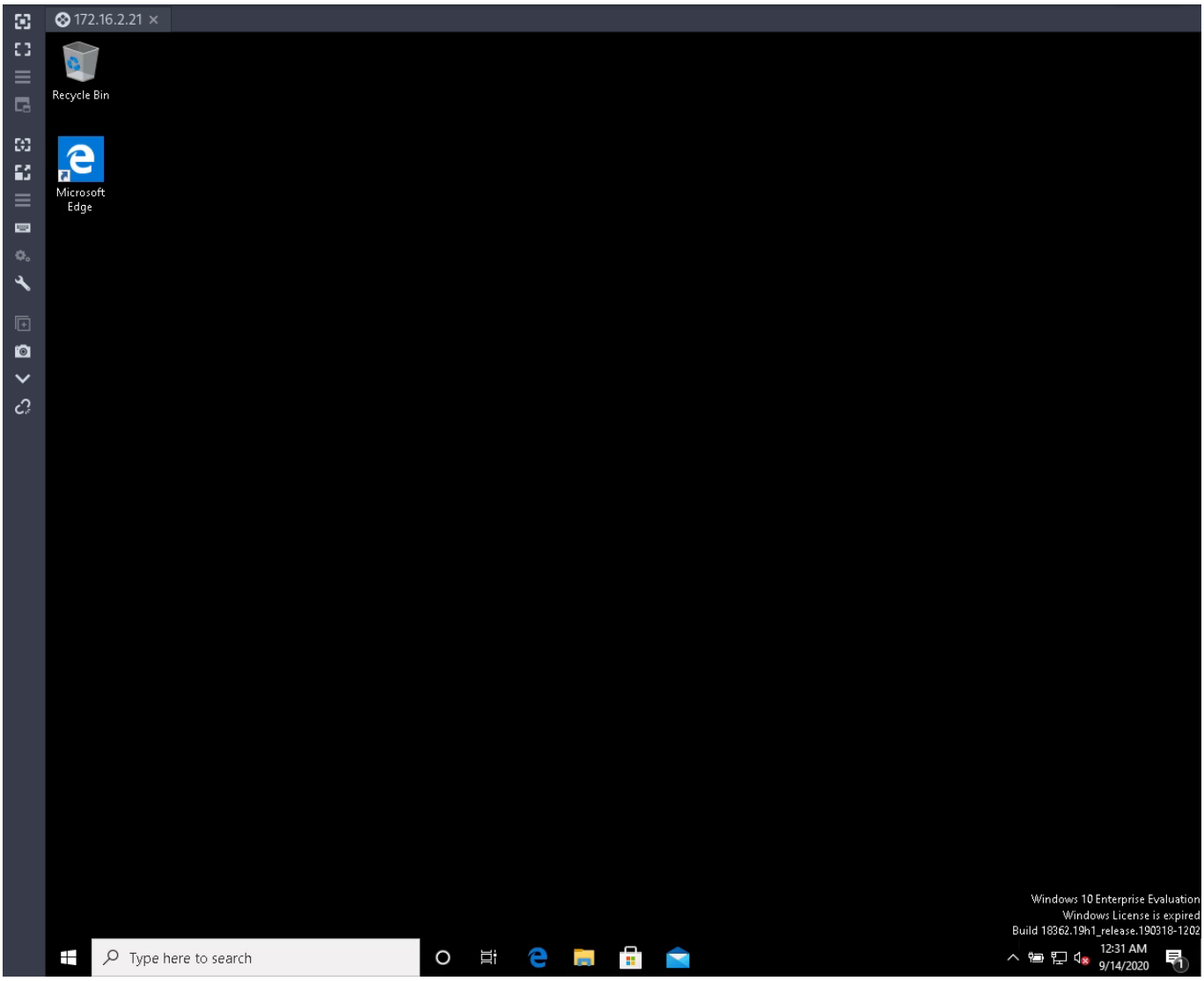
```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'tas50/windows_10'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'tas50/windows_10' version '202002.04.0' is up to date...
==> default: Setting the name of the VM: windows2010_default_1600052647209_36957
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
==> default: Forwarding ports...
    default: 3389 (guest) => 3389 (host) (adapter 1)
    default: 5985 (guest) => 55985 (host) (adapter 1)
    default: 5986 (guest) => 55986 (host) (adapter 1)
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: WinRM address: 127.0.0.1:55985
    default: WinRM username: vagrant
    default: WinRM execution_time_limit: PT2H
    default: WinRM transport: negotiate
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Waiting for machine to reboot...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
    default: /vagrant => /home/gitoso/VirtualBox VMs/windows2010
==> default: Running provisioner: shell...
    default: Running: rdp.ps1 as C:\tmp\vagrant-shell.ps1
```

Em seguida realizei o acesso via RDP utilizando um software chamado Remmina





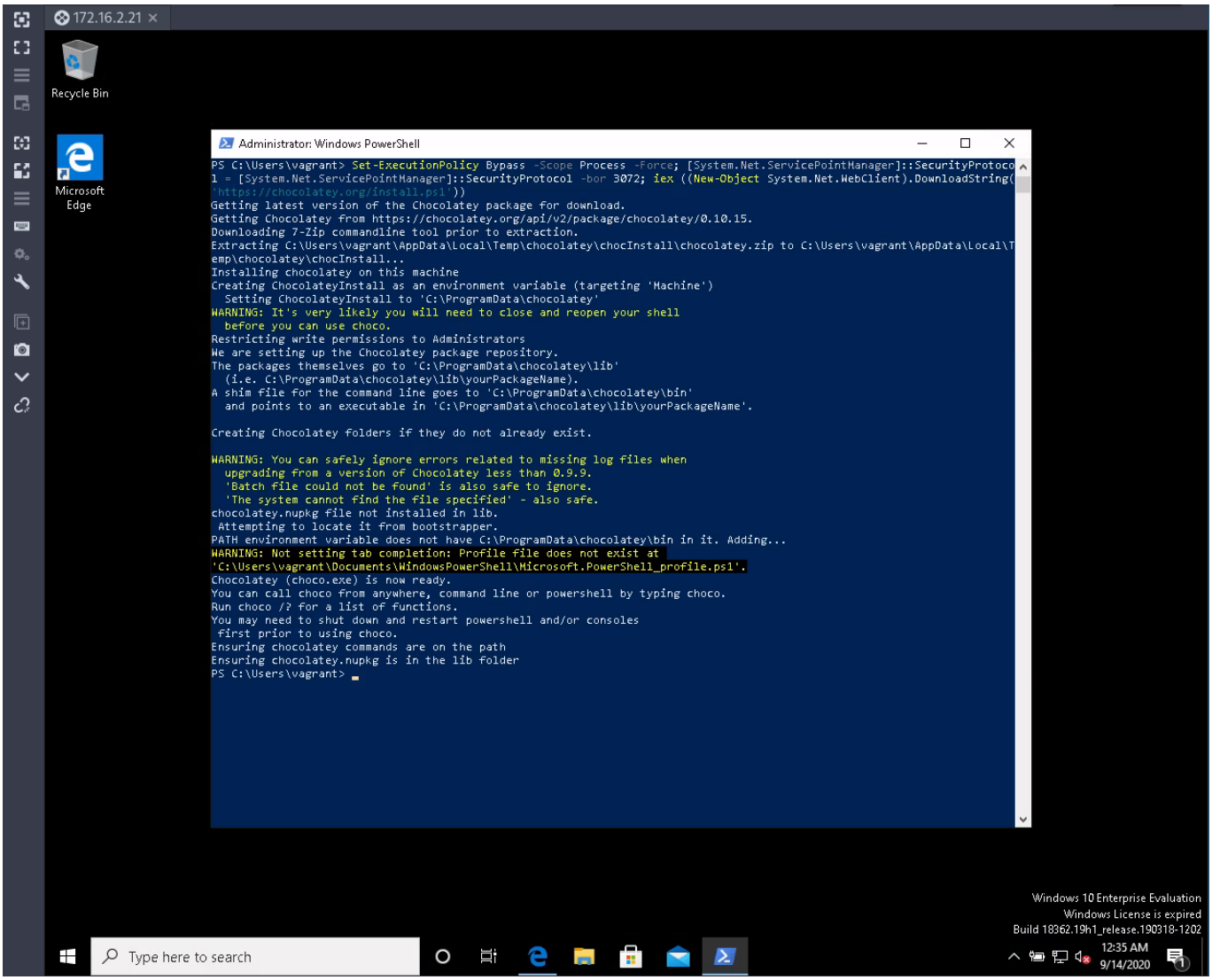
E foi possível acessar o desktop e verificar que o acesso foi realizado com sucesso



24) Acionar o browser e instalar o "chocolatey"

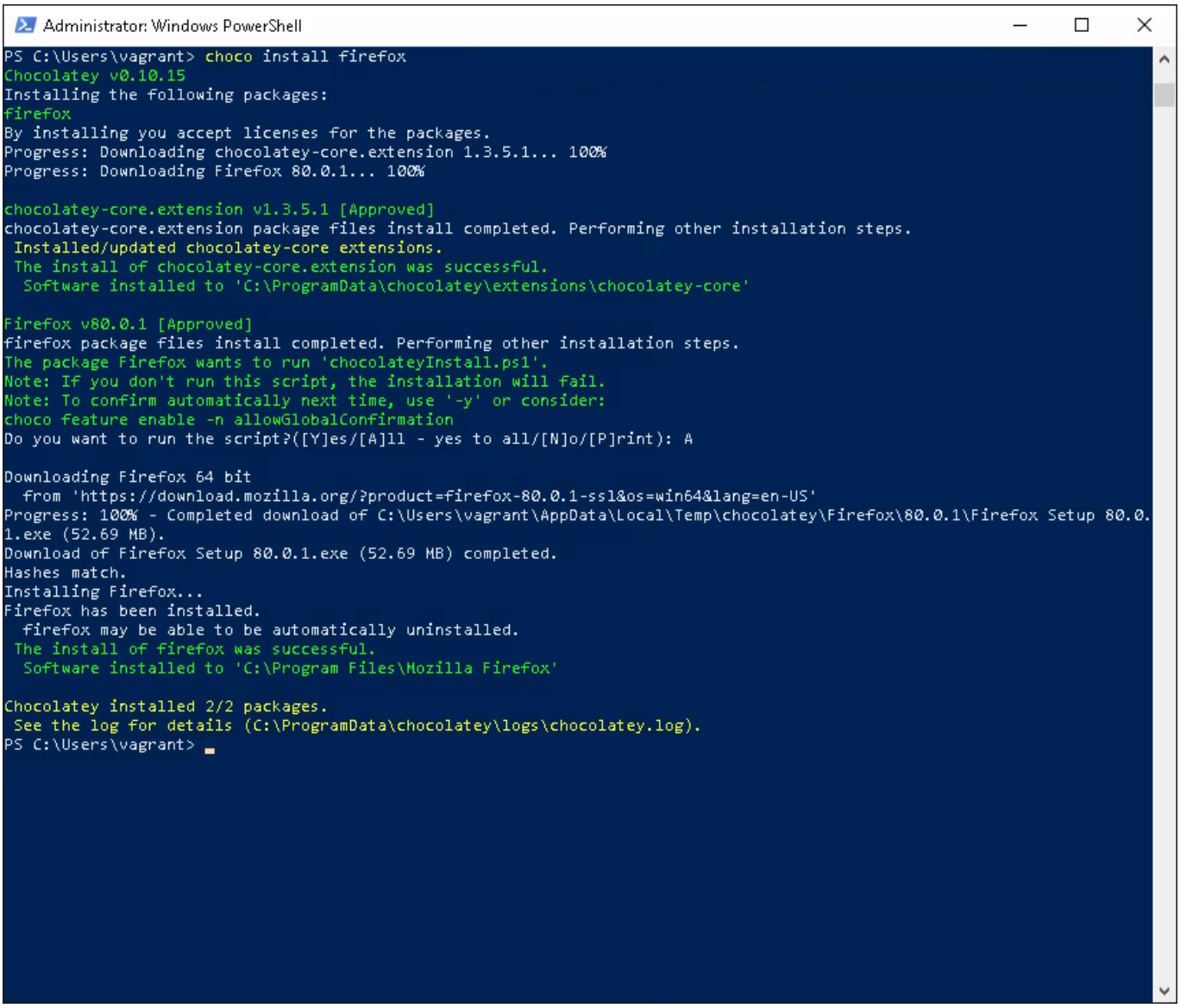
```
# vá no site do chocolatey / Install
# acione o programa “powershell” com modo administrador
# copie e cole o comando para instalar o chocolatey a partir do powershell
```

A instalação do `chocolatey` foi realizada executando o comando mencionado no site oficial



25) Instale o browser "Firefox"

```
# choco install Firefox
```



26) Reconfigurar a memoria para 1G RAM, alterando o arquivo Vagrantfile e testar recarregando essa configuração

```
# vagrant reload
```

```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vim Vagrantfile
```

```
Vagrant.configure("2") do |config|
  config.vm.box = "tas50/windows_10"
  config.vm.hostname = "server"
  config.vm.provider "virtualbox" do |vb|
    vb.gui = false
    vb.memory = 1024
    vb.cpus = 2
  end
  config.vm.network :private_network, ip: "172.16.2.21" # 21 >> 20
  config.vm.provision "shell", path: "rdp.ps1"
end
~
~
```

```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vagrant reload
==> default: Attempting graceful shutdown of VM...
==> default: Checking if box 'tas50/windows_10' version '202002.04.0' 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: hostonly
==> default: Forwarding ports...
    default: 3389 (guest) => 3389 (host) (adapter 1)
    default: 5985 (guest) => 55985 (host) (adapter 1)
    default: 5986 (guest) => 55986 (host) (adapter 1)
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: WinRM address: 127.0.0.1:55985
    default: WinRM username: vagrant
    default: WinRM execution_time_limit: PT2H
    default: WinRM transport: negotiate
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
    default: /vagrant => /home/gitoso/VirtualBox VMs/windows2010
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
```

Ao tentar rodar o "provision" novamente, houve um erro pois a opção

```
Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
```

do arquivo `rdp.ps1` exige mais memória RAM que o disponível na nova configuração da VM.

Dessa forma, a única maneira possível de acessar a VM neste caso foi remover esta opção opção do arquivo `rdp.ps1`. Como essa opção habilita o acesso via RDP, a única alternativa de acesso com 1GB de RAM foi o acesso via a própria interface do Virtualbox

27) Destruir a VM

```
gitoso@LOVELACE ~/VirtualBox VMs/windows2010 vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
```

Parte III: "Mini-Cluster"

Gravação

- Asciinema: ...

A seguir, a execução dos passos do laboratório será seguida por *screenshots* mostrando o resultado obtido para cada passo.

Execução passo-a-passo

28) Criar um novo diretório "doisubuntu"

29) Criar uma arquivo Vagrantfile novo contendo 2 maquinas virtuais

30) Acionar os ambientes no vagrant

31) Verificar o status e acessar cada uma das maquinas de maneira independente, e atualizar a maquina vm01, fazer teste de ping entre as maquinas

32) Atualizar o Vagrantfile e trocar o box da vm02 por bionic64. E Instalar apache2 na vm02

33) Testar acessar o servidor web com o browser a partir do Host e a partir do vm01 guest usando "wget" no Red Hat

34) Explorando um pouco dos "plug-ins" do Vagrant