Лабораторная работа №1

Подготовка лабораторного стенда

Гурылев Артем Андреевич

2025-11-12

Содержание i

1. Информация

1.1 Докладчик

- Гурылев Артем Андреевич
- Студент
- Российский университет дружбы народов им. П. Лумумбы

1.2 Цель работы

- Целью данной лабораторной работы является приобретение практических навыков установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

2. Выполнение лабораторной работы

2.1 Конфигурация кода

В этой лабораторной работе устанавливается версия Rocky Linux 10, поэтому в ней был использован пример кода, подходящий для этой версии.

2.2 Конфигурация папок

Рисунок 1: Конфигурация папок Vagrant и Packer

2.3 Конфигурационные файлы

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

Vagrant.configure("2") do |config|

  config.vagrant.plugins = "vagrant-libvirt"
  config.vagrant.plugins = "vagrant-vbguest"

  config.vm.provider :virtualbox do |virtualbox|
    virtualbox.linked_clone = true
    # Customize the amount of memory on the VM
    virtualbox.memory = 2048
    virtualbox.cpus = 2
    ## Display the VirtualBox GUI when booting the machine
    virtualbox.gui = false
    ## Set the video memory to 12Mb
    virtualbox.customize ["modifyvm", :id, "--vram", "32"]
    virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
    virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
    virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
    virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
    virtualbox.customize ["modifyvm", :id, "--accelerate3d", "on"]
    virtualbox.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
  end

  config.vm.provider :libvirt do |libvirt|
    libvirt.driver = "kvm"
    libvirt.memory = 2048
    libvirt.cpus = 2
    libvirt.video_type = "virtio"
    libvirt.disk_bus = "virtio"
    libvirt.nic_model_type = "virtio"
    libvirt.management_network_name = "vagrant-libvirt"
    libvirt.management_network_address = "192.168.121.0/24"
    libvirt.storage_pool_name = "vagrant"
    # libvirt.storage_pool_name = "default"
  end

  ## Common configuration
  config.vm.provision "common dummy",
                      type: "shell",
                      preserve_order: true,
                      path: "provision/default/01-dummy.sh"
```

Рисунок 2: Конфигурационный файл Vagrantfile

2.4 Конфигурационные файлы

```
# System bootloader configuration
bootloader --append="no_timer_check console=tty0 console=ttyS0,115200n8 net.ifnames=0 biosdevname=0 elevator=noop" --location=mbr --timeout=1
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all
# Reboot after installation
reboot
# Use text mode install
text
# Keyboard layouts
keyboard --vckeymap=us,ru --xlayouts='us,ru'
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=dhcp --device=link --activate

# System authorization information
authselect select sssd with-sudo with-mkhomedir --force
authselect apply-changes
# Root password
rootpw vagrant
user --name=vagrant --password=vagrant
firstboot --disable
# Do not configure the X Window System
#skipx
# System services
services --enabled="NetworkManager,sshd,chronyd"
# System timezone
timezone UTC --utc
user --name=vagrant --password=vagrant
# Disk partitioning information
# part / --fstype="xfs" --size=10239
bootloader --location=mbr
clearpart --all --initlabel
autopart --type=lvm

%post
# configure swap to a file
# fallocate -l 2G /swapfile
# chmod 600 /swapfile
# mkswap /swapfile
# echo "/swapfile none swap defaults 0 0" >> /etc/fstab
```

Рисунок 3: Конфигурационный файл ks.cfg

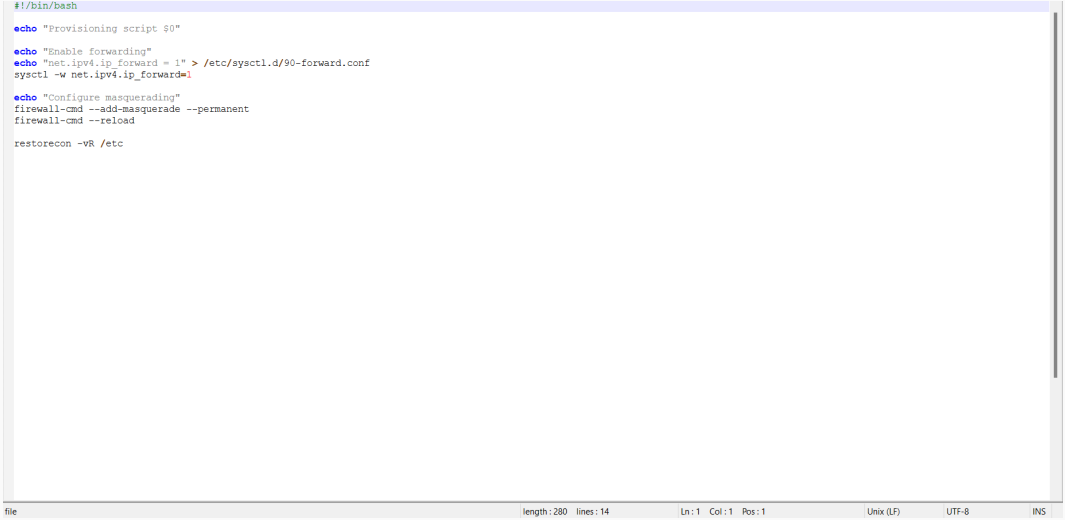
2.5 Конфигурационные файлы

```
packer {
  required_plugins {
    vagrant = {
      source  = "github.com/hashicorp/vagrant"
      version = "~> 1"
    }
    virtualbox = {
      version = "~> 1"
      source  = "github.com/hashicorp/virtualbox"
    }
    qemu = {
      version = "~> 1"
      source  = "github.com/hashicorp/qemu"
    }
  }
}

variable "artifact_description" {
  type    = string
  default = "Rocky 10.0"
}

variable "artifact_version" {
  type    = string
  default = "10.0"
}

variable "disk_size" {
  type    = string
  default = "61440"
}

variable "iso_checksum" {
  type    = string
  default = "de75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cadd12"
}

variable "iso_checksum_type" {
  type    = string
  default = "sha256"
}

variable "iso_url" {
  type    = string
```

d file | length : 5 519 | lines : 189 | Ln : 23 | Col : 8 | Pos : 402 | Unix (LF) | UTF-8 | INS

Рисунок 4: Конфигурационный факл vagrant-rocky.pkr.hcl

2.6 Конфигурационные файлы

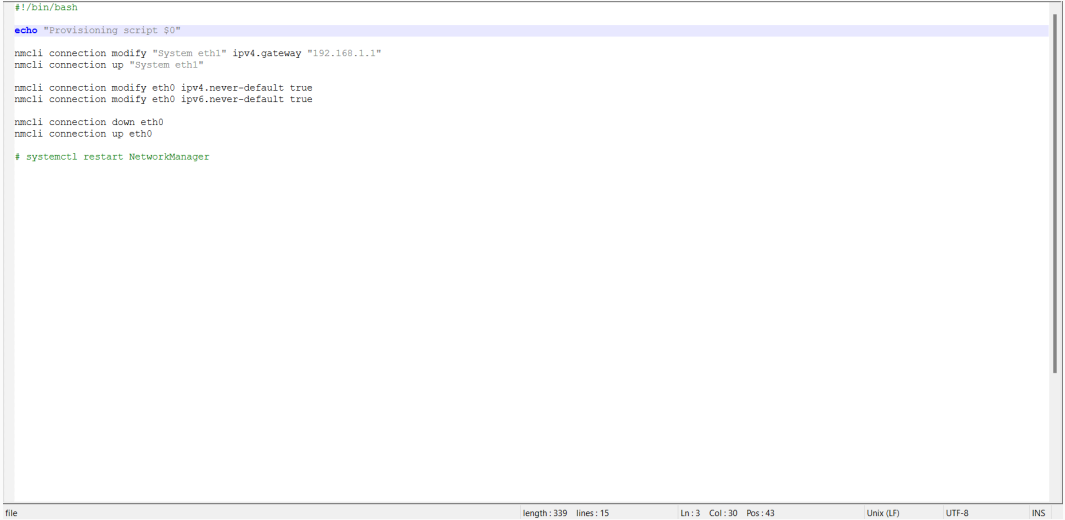


```
#!/bin/bash

echo "Provisioning script $0"

echo "Enable forwarding"
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
sysctl -w net.ipv4.ip_forward=1

echo "Configure masquerading"
firewall-cmd --add-masquerade --permanent
firewall-cmd --reload

restorecon -vR /etc
```

file length : 280 lines : 14 Ln : 1 Col : 1 Pos : 1 Unix (LF) UTF-8 INS

Рисунок 5: Скрипт виртуальной машины server 02-forward.sh

2.7 Конфигурационные файлы



```
#!/bin/bash

echo "Provisioning script $0"

nmcli connection modify "System eth1" ipv4.gateway "192.168.1.1"
nmcli connection up "System eth1"

nmcli connection modify eth0 ipv4.never-default true
nmcli connection modify eth0 ipv6.never-default true

nmcli connection down eth0
nmcli connection up eth0

# systemctl restart NetworkManager
```

file length : 339 lines : 15 Ln : 3 Col : 30 Pos : 43 Unix (LF) UTF-8 INS

Рисунок 6: Скрипт виртуальной машины client 01-routing.sh

2.8 Создание box-файла

```
C:\Work\aagurihlev\packer>packer.exe init vagrant-rocky.pkr.hcl

C:\Work\aagurihlev\packer>packer.exe build vagrant-rocky.pkr.hcl
virtualbox-iso.rockylinux: output will be in this color.
qemu.rockylinux: output will be in this color.

Build 'qemu.rockylinux' errored after 11 milliseconds 111 microseconds: Failed creating Qemu driver: exec: "qemu-system-x86_64": executable file not found in
%PATH%
==> virtualbox-iso.rockylinux: Retrieving Guest additions
==> virtualbox-iso.rockylinux: Trying C:\Program Files\Oracle\VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: Trying file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso => C:/Program Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: Retrieving ISO
==> virtualbox-iso.rockylinux: Trying Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.rockylinux: Trying Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cadd12
==> virtualbox-iso.rockylinux: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cadd12 => C:/Work/aag
urihlev/packer/Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.rockylinux: Starting HTTP server on port 8468
==> virtualbox-iso.rockylinux: Creating virtual machine...
==> virtualbox-iso.rockylinux: Creating hard drive output-rockylinux10-virtualbox\rockylinux10-virtualbox.vdi with size 61440 MiB...
==> virtualbox-iso.rockylinux: Mounting ISOs...
==> virtualbox-iso.rockylinux: Mounting boot ISO...
==> virtualbox-iso.rockylinux: Creating forwarded port mapping for communicator (SSH, WinRM, etc) (host port 2278)
==> virtualbox-iso.rockylinux: Executing custom VBoxManage commands...
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --memory 2048
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --cpus 2
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --nat-localhostreachable1 on
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --firmware EFI
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --vrde on
==> virtualbox-iso.rockylinux: Executing: modifyvm rockylinux10-virtualbox --vrdeport 3390
==> virtualbox-iso.rockylinux: Starting the virtual machine...
==> virtualbox-iso.rockylinux: The VM will be run headless, without a GUI. If you want to
==> virtualbox-iso.rockylinux: view the screen of the VM, connect via VRDP without a password to
==> virtualbox-iso.rockylinux: rdp://127.0.0.1:5952
==> virtualbox-iso.rockylinux: Waiting 10s for boot...
==> virtualbox-iso.rockylinux: Typing the boot command...
==> virtualbox-iso.rockylinux: Using SSH communicator to connect: 127.0.0.1
==> virtualbox-iso.rockylinux: Waiting for SSH to become available...
```

Рисунок 7: Результат работы packer

2.9 Создание box-файла

```
==> virtualbox-iso.rockylinux: Gracefully halting virtual machine...                                    2:54
==> virtualbox-iso.rockylinux:
==> virtualbox-iso.rockylinux: Broadcast message from root@localhost on pts/2 (Sun 2025-11-09 23:46:02 UTC):
==> virtualbox-iso.rockylinux:
==> virtualbox-iso.rockylinux: The system will power off now!
==> virtualbox-iso.rockylinux:
==> virtualbox-iso.rockylinux: Preparing to export machine...
==> virtualbox-iso.rockylinux: Deleting forwarded port mapping for the communicator (SSH, WinRM, etc) (host port 2278)
==> virtualbox-iso.rockylinux: Exporting virtual machine...
==> virtualbox-iso.rockylinux: Executing: export rockylinux10-virtualbox --output output-rockylinux10-virtualbox\rockylinux10-virtualbox.ovf --manifest --vsys
 0 --description Rocky 10.0 --version 10.0
==> virtualbox-iso.rockylinux: Cleaning up floppy disk...
==> virtualbox-iso.rockylinux: Deregistering and deleting VM...
==> virtualbox-iso.rockylinux: Running post-processor:  (type vagrant)
==> virtualbox-iso.rockylinux (vagrant): Creating a dummy Vagrant box to ensure the host system can create one correctly
==> virtualbox-iso.rockylinux (vagrant): Creating Vagrant box for 'virtualbox' provider
==> virtualbox-iso.rockylinux (vagrant): Copying from artifact: output-rockylinux10-virtualbox\rockylinux10-virtualbox-disk001.vmdk
==> virtualbox-iso.rockylinux (vagrant): Copying from artifact: output-rockylinux10-virtualbox\rockylinux10-virtualbox.mf
==> virtualbox-iso.rockylinux (vagrant): Copying from artifact: output-rockylinux10-virtualbox\rockylinux10-virtualbox.nvram
==> virtualbox-iso.rockylinux (vagrant): Copying from artifact: output-rockylinux10-virtualbox\rockylinux10-virtualbox.ovf
==> virtualbox-iso.rockylinux (vagrant): Renaming the OVF to box.ovf...
==> virtualbox-iso.rockylinux (vagrant): Compressing: Vagrantfile
==> virtualbox-iso.rockylinux (vagrant): Compressing: box.ovf
==> virtualbox-iso.rockylinux (vagrant): Compressing: metadata.json
==> virtualbox-iso.rockylinux (vagrant): Compressing: rockylinux10-virtualbox-disk001.vmdk
==> virtualbox-iso.rockylinux (vagrant): Compressing: rockylinux10-virtualbox.mf
==> virtualbox-iso.rockylinux (vagrant): Compressing: rockylinux10-virtualbox.nvram
Build 'virtualbox-iso.rockylinux' finished after 42 minutes 22 seconds.

==> Wait completed after 42 minutes 22 seconds

==> Some builds didn't complete successfully and had errors:
--> qemu.rockylinux: Failed creating Qemu driver: exec: "qemu-system-x86_64": executable file not found in %PATH%

==> Builds finished. The artifacts of successful builds are:
--> virtualbox-iso.rockylinux: 'virtualbox' provider box: vagrant-virtualbox-rockylinux10-x86_64.box

C:\Work\aagurihlev\packer>                                                                              ↑
1Help  2UserMn  3View  4Edit  5Copy  6RenMov  7MkFold  8Delete  9ConfMn  10Quit  11Plugin  12Screen
```

Рисунок 8: Результат работы packer

2.10 Регистрация образа в Vagrant

```
C:\Work\aagurihlev\packer>vagrant box add rockylinux10 vagrant-virtualbox-rockylinux10-x86_64.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rockylinux10' (v0) for provider: (amd64)
    box: Unpacking necessary files from: file://C:/Work/aagurihlev/packer/vagrant-virtualbox-rockylinux10-x86_64.box
    box:
==> box: Successfully added box 'rockylinux10' (v0) for '(amd64)'!
```

Рисунок 9: Регистрация box-файла в Vagrant

2.11 Запуск виртуальной машины server

```
C:\Work\aagurihlev\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Preparing master VM for linked clones...
    server: This is a one time operation. Once the master VM is prepared,
    server: it will be used as a base for linked clones, making the creation
    server: of new VMs take milliseconds on a modern system.
==> server: Importing base box 'rockylinux10'...
==> server: Cloning VM...
==> server: Matching MAC address for NAT networking...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Setting the name of the VM: vagrant_server_1762732829550_39604
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
    server: Adapter 1: nat
    server: Adapter 2: intnet
==> server: Forwarding ports...
    server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations...
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
    server: SSH address: 127.0.0.1:2222
    server: SSH username: vagrant
    server: SSH auth method: password
==> server: Machine booted and ready!
```

Рисунок 10: Запуск виртуальной машины server

2.12 Запуск виртуальной машины server

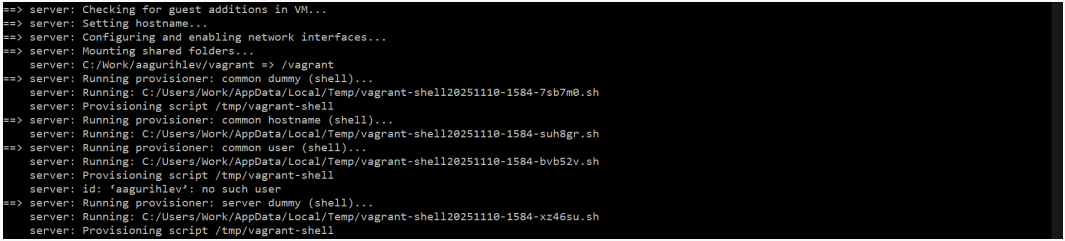


Рисунок 11: Запуск виртуальной машины server

2.13 Запуск виртуальной машины client

```
==> client: Booting VM...
==> client: Waiting for machine to boot. This may take a few minutes...
==> client: Machine booted and ready!
==> client: Checking for guest additions in VM...
==> client: Setting hostname...
==> client: Configuring and enabling network interfaces...
==> client: Mounting shared folders...
    client: C:/Work/aagurihlev/vagrant => /vagrant
==> client: Running provisioner: common dummy (shell)...
    client: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-20820-2pju08.sh
    client: Provisioning script /tmp/vagrant-shell
==> client: Running provisioner: common hostname (shell)...
    client: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-20820-oq76yr.sh
==> client: Running provisioner: common user (shell)...
    client: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-20820-n54dk8.sh
    client: Provisioning script /tmp/vagrant-shell
    client: id: ‘aagurihlev’: no such user
==> client: Running provisioner: client dummy (shell)...
    client: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-20820-17t70j.sh
    client: Provisioning script /tmp/vagrant-shell
==> client: Running provisioner: client routing (shell)...
    client: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-20820-l4krvm.sh
    client: Provisioning script /tmp/vagrant-shell
    client: Error: unknown connection 'System eth1'.
    client: Error: unknown connection 'System eth1'.
    client: Connection 'eth0' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/2)
    client: Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)
```

Рисунок 12: Запуск виртуальной машины client

2.14 Логин с использованием SSH

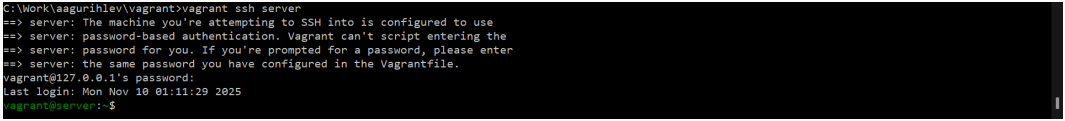


Рисунок 13: Логин под пользователем vagrant виртуальной машины server

2.15 Логин с использованием SSH

Рисунок 14: Логин под собственным пользователем виртуальной машины server

2.16 Логин с использованием SSH

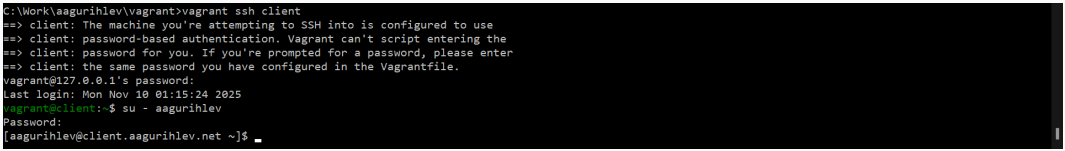


Рисунок 15: Логин под пользователем vagrant и под собственным пользователем виртуальной машины client

2.17 Выводы

- В результате выполнения лабораторной работы были получены навыки работы с программой Vagrant, а также настроена система виртуальных машин Linux server и client.