

Лабораторная работа №1

Подготовка лабораторного стенда

Гурылев Артем Андреевич

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	12

Список иллюстраций

2.1	Конфигурация папок Vagrant и Packer	5
2.2	Конфигурационный файл Vagrantfile	6
2.3	Конфигурационный файл ks.cfg	6
2.4	Конфигурационный файл vagrant-rocky.pkr.hcl	7
2.5	Скрипт виртуальной машины server 02-forward.sh	7
2.6	Скрипт виртуальной машины client 01-routing.sh	8
2.7	Результат работы packer	8
2.8	Результат работы packer	9
2.9	Регистрация box-файла в Vagrant	9
2.10	Запуск виртуальной машины server	10
2.11	Запуск виртуальной машины server	10
2.12	Запуск виртуальной машины client	10
2.13	Логин под пользователем vagrant виртуальной машины server	11
2.14	Логин под собственным пользователем виртуальной машины server	11
2.15	Логин под пользователем vagrant и под собственным пользователем виртуальной машины client	11

1 Цель работы

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

2 Выполнение лабораторной работы

В данной лабораторной работе устанавливается версия Rocky Linux 10, поэтому в ходе её выполнения был использован пример кода.



Рисунок 2.1: Конфигурация папок Vagrant и Packer

```

# -*- 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"
end

```

Рисунок 2.2: Конфигурационный файл Vagrantfile

```

# 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
# falloccate -l 2g /swapfile
# chmod 600 /swapfile
# mkswap /swapfile
+ echo "swaponfile none swan.defaults 0 0" >> /etc/fstab

```

Рисунок 2.3: Конфигурационный файл ks.cfg

```

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
  default = "https://rockylinux.org/releases/10.0/iso/rocky-10.0-01.iso"
}

```

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

Рисунок 2.4: Конфигурационный файл vagrant-rocky.pkr.hcl

```

#!/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 2.5: Скрипт виртуальной машины server 02-forward.sh

```
#!/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
```

Рисунок 2.6: Скрипт виртуальной машины client 01-routing.sh

Теперь приступим к основной части работы. Для начала нам нужно сформировать box-файл виртуальной машины - для этого используем программу packer. Он работает с конфигурационным файлом vagrant-rocky.pkr.hcl, в котором содержатся настройки и команды, которые выполняться во время создания базовой виртуальной машины. (рис. 2.7, рис. 2.8)

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

C:\Work\agurihlev\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\VBBoxGuestAdditions.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/agurihlev/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-localhostreachble1 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 2.7: Результат работы packer


```
==> virtualbox-iso.rockylinux: Gracefully halting virtual machine...
==> 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 --sys
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\aaagurihlev\packer>
```

Рисунок 2.8: Результат работы packer

Далее необходимо зарегистрировать созданный box-файл в программе Vagrant.
(рис. 2.9)

```
C:\Work\aaagurihlev\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/aaagurihlev\packer\vagrant-virtualbox-rockylinux10-x86_64.box
box:
==> box: Successfully added box 'rockylinux10' (v0) for '(amd64)'!
```

Рисунок 2.9: Регистрация box-файла в Vagrant

Теперь запустим виртуальные машины server и клиент - они основаны на box-файле, который мы создали с помощью packer. Начнем с сервера, выполнив команду `vagrant up server`. (рис. 2.10, рис. 2.11)

```

C:\Work\agurihlev\ 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 2.10: Запуск виртуальной машины server

```

==> server: Checking for guest additions in VM...
==> server: Setting hostname...
==> server: Configuring and enabling network interfaces...
==> server: Mounting shared folders...
server: C:/Work/agurihlev/vagrant => /vagrant
==> server: Running provisioner: common dummy (shell)...
server: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-1584-7sb7m0.sh
server: Provisioning script /tmp/vagrant-shell
==> server: Running provisioner: common hostname (shell)...
server: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-1584-suh8gr.sh
==> server: Running provisioner: common user (shell)...
server: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-1584-bvb52v.sh
server: Provisioning script /tmp/vagrant-shell
server: id: 'agurihlev': no such user
==> server: Running provisioner: server dummy (shell)...
server: Running: C:/Users/Work/AppData/Local/Temp/vagrant-shell20251110-1584-xz46su.sh
server: Provisioning script /tmp/vagrant-shell

```

Рисунок 2.11: Запуск виртуальной машины server

После запуска и проверки server запустим виртуальную машину client командой `vagrant up client`. (рис. 2.12)

```

==> 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/agurihlev/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: 'agurihlev': 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-shell20251110-20820-14krvm.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 2.12: Запуск виртуальной машины client

Запустив обе виртуальные машины, зайдем в пользователя vagrant в обеих системах, а затем проверим ssh-соединение с ними. Вписав в консоли команду `vagrant ssh server`, подключимся к серверу и залогинимся в собственного пользователя, затем выйдем. (рис. 2.13, рис. 2.14)

```
C:\Work\aaagurihlev\vagrant>vagrant ssh server
==> server: The machine you're attempting to SSH into is configured to use
==> server: password-based authentication. Vagrant can't script entering the
==> server: password for you. If you're prompted for a password, please enter
==> server: the same password you have configured in the Vagrantfile.
vagrant@127.0.0.1's password:
Last login: Mon Nov 10 01:11:29 2025
vagrant@server:~$
```

Рисунок 2.13: Логин под пользователем vagrant виртуальной машины server

```
vagrant@server:~$ su - aaagurihlev
Password:
Last failed login: Mon Nov 10 01:13:44 UTC 2025 on pts/0
There were 4 failed login attempts since the last successful login.
[aaagurihlev@server.aaagurihlev.net ~]$
```

Рисунок 2.14: Логин под собственным пользователем виртуальной машины server

Тоже самое сделаем и с клиентом - посредством команды `vagrant ssh client`. (рис. 2.15)

```
C:\Work\aaagurihlev\vagrant>vagrant ssh client
==> client: The machine you're attempting to SSH into is configured to use
==> client: password-based authentication. Vagrant can't script entering the
==> client: password for you. If you're prompted for a password, please enter
==> client: the same password you have configured in the Vagrantfile.
vagrant@127.0.0.1's password:
Last login: Mon Nov 10 01:15:24 2025
vagrant@client:~$ su - aaagurihlev
Password:
[aaagurihlev@client.aaagurihlev.net ~]$
```

Рисунок 2.15: Логин под пользователем vagrant и под собственным пользователем виртуальной машины client

Выключим обе виртуальные машины командами `vagrant halt server` и `vagrant halt client`.

3 Выводы

В результате выполнения лабораторной работы была настроена система виртуальных машин Linux server и client, в которых будут выполняться следующие лабораторные работы, а также получены навыки работы с программой vagrant.