

# Отчёт о лабораторной работе

Лабораторная работа 1

Мошаров Денис Максимович

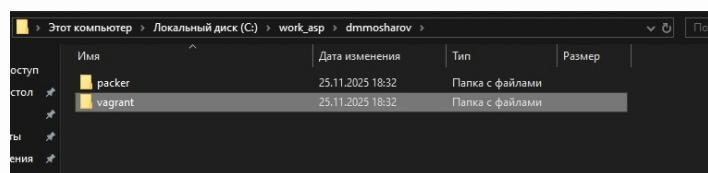
## Содержание

### Цель работы

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

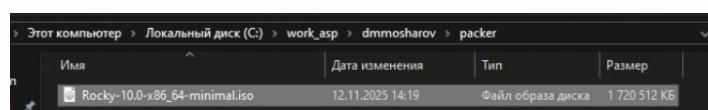
### Выполнение лабораторной работы

Для начала создадим папку с инициалами, в которой будет 2 папки, показанные на фото (рис. [-@fig:001]).



Создание папок

Поместим заранее скачанный образ в первую папку (рис. [-@fig:002]).



Образ диска

Создадим в той же папке файл, содержание которого перечислено на экране. Тут описано, как будет создаваться образ для vagrant (рис. [-@fig:003]).

```

C:\> work.asp > dmmosharov > packer > vagrant-rocky.pkr.hd
 74 source "vmlinuz" "rockylinux" {
 75   ...
 76   accelerator = "kvm"
 77   cpu_model = "host"          # Использовать характеристики хоста
 78   machine_type = "q35"        # Гиперсистемной платы
 79   firmware = "/usr/share/edk2-ovmf/OVMF_CODE.fd" # UEFI вместо BIOS. Проверьте местоположение
 80
 81   # Настройки видео
 82   vga = "virtio" # Для virtio-vga
 83
 84   ## Дополнительные флаги процессора
 85   qemuargs = [
 86     ["-device", "qemu-xhci"], # Виртуализированные USB контроллеры
 87     ["-device", "virtio-tablet"], # Устройства ввода
 88     ## GPU-passthrough
 89     #["-device", "virtio-gpu-pci"], # 3D-акселерация через VirGL
 90     # ["-vga", "none"]
 91   ]
 92 }
 93
 94 source "virtualbox-iso" "rockylinux" {
 95   boot_command = [
 96     "<op>",
 97     "+rt",
 98     "<down><down><end><wait>",
 99     "inst.ks=http://{{ .HTTPIP }}:{.HTTPPort }/rocky10-ks.cfg",
100     "biosdevname=0 net.ifnames=0",
101     "<enter><wait><leftCtrlOn><leftCtrlOff>"
102   ]
103   boot_wait = "10s"
104   disk_size = "${var.disk_size}"
105   export_opts = [
106     "-manifest",
107     "-vsyst", "0",
108     "-description", "${var.artifact_description}",
109     "-version", "${var.artifact_version}"
110   ]
111   guest_additions_path = "VBoxGuestAdditions.iso"
112   guest_os_type = "RedHat_64"
113   http_directory = "${var.http_directory}"
114   iso_checksum = "${var.iso_checksum_type}:${var.iso_checksum}"

```

vagrant-rocky.pkr.hd

Создадим подпапку http, в которой поместим файл следующего содержания с конфигурацией (рис. [-@fig:004]).

```

C:\> work.asp > dmmosharov > packer > http > rocky10-ks.cfg
 39 %post
 40 # configure swap to a file
 41 # fallocate -l 2G /swapfile
 42 # chmod 600 /swapfile
 43 # mkswap /swapfile
 44 # echo "/swapfile none swap defaults 0 0" >> /etc/fstab
 45
 46 # sudo
 47 echo "%Vagrant ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/vagrant
 48 chmod 0440 /etc/sudoers.d/vagrant
 49
 50 # Fix for https://github.com/centOS/sig-cloud-instance-build/issues/38
 51 cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
 52 DEVICE="eth0"
 53 BOOTPROTO="dhcp"
 54 ONBOOT="yes"
 55 TYPE="Ethernet"
 56 PERSISTENT_DHCLIENT="yes"
 57 EOF
 58
 59 # sshd: disable password authentication and DNS checks
 60 #ex -s /etc/ssh/sshd_config <<EOF
 61 #:x$subst((^)(PasswordAuthentication) yes$/\1 no/
 62 #:$subst(/#(\UseDNS) yes#/&\1 no/
 63 #:update
 64 #:quit
 65 #EOF
 66 #cat >>/etc/sysconfig/sshd <<EOF
 67
 68 # Decrease connection time by preventing reverse DNS lookups
 69 # (see https://lists.centos.org/pipermail/centos-devel/2016-July/014981.html)
 70 # and man sshd for more information)
 71 OPTIONS="-u0"
 72 EOF
 73
 74 # Fix for issue #76, regular users can gain admin privileges via su
 75 ex -s /etc/pam.d/su <<'EOF'
 76 # allow vagrant to use su, but prevent others from becoming root or vagrant
 77 /account(s)+sufficient(s+pam_succeed_if.so uid = 0 use_uid quiet/
 78 :append
 79 account [success=1 default=ignore] \\
 80   | | | | | pam_succeed_if.so user = vagrant use_uid quiet
 81 account required pam_succeed_if.so user notin root:vagrant
 82
 83 :update
 84 :quit
 85 EOF
 86

```

Файл конфигурации

Теперь добавим vagrantfile в папку vagrant (рис. [-@fig:005]).

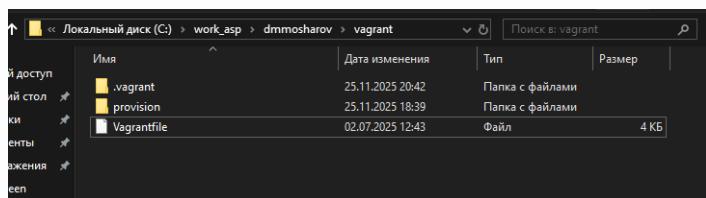
```

# Vagrantfile
C > work_asp > dmmosharov > vagrant > Vagrantfile
1   | # -*- mode: ruby -*-
2   | # vi: set ft=ruby :
3
4   Vagrant.configure("2") do |config|
5
6     config.vagrant.plugins = "vagrant-libvirt"
7     config.vagrant.plugins = "vagrant-vbguest"
8
9     config.vm.provider :virtualbox do |virtualbox|
10    virtualbox.linked_clone = true
11    # Customize the amount of memory on the VM
12    virtualbox.memory = 2048
13    virtualbox.cpus = 2
14    ## Display the VirtualBox GUI when booting the machine
15    virtualbox.gui = false
16    ## Set the video memory to 128MB
17    virtualbox.customize ["modifyvm", :id, "-vram", "32"]
18    virtualbox.customize ["modifyvm", :id, "-natdnshostresolver", "on"]
19    virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
20    virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
21    virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
22    virtualbox.customize ["modifyvm", :id, "--accelerated3d", "on"]
23    virtualbox.customize ["modifyvm", :id, "--nested-hv-virt", "on"]
24  end
25
26  config.vm.provider :libvirt do |libvirt|
27    libvirt.driver = "kvm"
28    libvirt.memory = 2048
29    libvirt.cpus = 2
30    libvirt.video_type = "virtio"
31    libvirt.use_sandbox = "virtnet"
32    libvirt.vnc_model_type = "vnc"
33    libvirt.management_network_name = "vagrant-libvirt"
34    libvirt.management.network.address = "192.168.121.0/24"
35    libvirt.storage_pool_name = "vagrant"
36    # libvirt.storage_pool_name = "default"
37  end
38
39  ## Common configuration
40  config.vm.provision "common_dummy",
41    type: "shell",
42    preserve_order: true,
43    path: "provision/default/01-dummy.sh"
44
45  config.vm.provision "common_hostname",
46    type: "shell",
47    preserve_order: true,
48    run: "always"

```

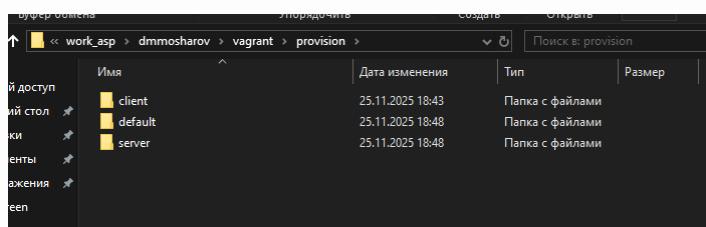
Vagrantfile

Содержимое папки будет выглядеть так. Теперь создадим тут папку provision (рис. [-@fig:006]).



Каталог Vagrant

В этой папке мы создадим 3 подпапки, показанные на фото (рис. [-@fig:007]).



Создание подпапок

В каждую папку добавим файл заглушки со следующим содержанием (рис. [-@fig:008]).

```

C: > work_asp > dmmosharov > vagrant > provision > client > $ 01-dummy.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5

```

Файл заглушки

В папку default добавим следующий файл (рис. [-@fig:009]).

```
$ 01-user.sh >
C: > work_asp > dmmosharov > vagrant > provision > default > $ 01-user.sh
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 username=dmmosharov
6 userpassword=123456
7
8 encpassword=`openssl passwd -1 ${userpassword}`
9
10 id -u $username
11 if [[ $? ]]
12 then
13     adduser -G wheel -p ${encpassword} ${username}
14     homedir=`getent passwd ${username} | cut -d: -f6`
15     echo "export PS1='[\u@\H \W]\$\'' >> ${homedir}/.bashrc
16 fi
17
18
19
```

01-user.sh

И этот файл (рис. [-@fig:010]).

```
C: > work_asp > dmmosharov > vagrant > provision > default > $ 01-hostname.sh
1 #!/bin/bash
2
3 username=dmmosharov
4
5 hostnamectl set-hostname "${HOSTNAME%%.*}}.${username}.net
6
7
8
```

01-hostname.sh

Добавим также 2 файла в папку сервера. Первый файл (рис. [-@fig:011]).

```
C: > work_asp > dmmosharov > vagrant > provision > server > $ 02-forward.sh
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 echo "Enable forwarding"
6 echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
7 sysctl -w net.ipv4.ip_forward=1
8
9 echo "Configure masquerading"
10 firewall-cmd --add-masquerade --permanent
11 firewall-cmd --reload
12
13 restorecon -vR /etc
14
```

первый файл

И второй файл (рис. [-@fig:012]).

```
C: > work_asp > dmmosharov > vagrant > provision > server > $ 01-dummy.sh
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5
```

Второй файл

Теперь инициализируем packer (рис. [-@fig:013]).

```
C:\work_asp\dmmosharov\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/denis/AppData/Roaming/packer.d/plugins/github.com/hashicorp/vagrant\tmp\hashicorp-vagrant_v1.1.5_x5.0_windows_amd64.exe"
```

инициализируем packer

И сделаем билд образа (рис. [-@fig:014]).

```
C:\work_asp\dmmosharov\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 6 milliseconds 784 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\VirtualBoxGuestAdditions.iso
--> virtualbox-iso.rockylinux: Trying file:///C:/ProgramFiles/Oracle/VirtualBox/VBoxGuestAdditions.iso => C:/Program Files/Oracle/VirtualBox/GuestAdditions.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%3Ade75c2f7cc566ea964017a1e94883913f066c4ebbe1d356964e398ed76cadd2
```

билд образа

После этого добавим его в vagrant (рис. [-@fig:015]).

```
C:\work_asp\dmmosharov\packer>vagrant box add rocky10 vagrant-virtualbox-rockylinux10-x86_64.box
--> box: Box file was not detected as metadata. Adding it directly...
--> box: Adding box 'rocky10' for provider: (amda)
--> box: Unpacking necessary files from: file:///c:/work_asp\dmmosharov\packer\vagrant-virtualbox-rockylinux10-x86_64.box
```

добавление образа в vagrant

Запустим через vagrant BM сервера (рис. [-@fig:016]).

```
C:\work_asp\dmmosharov\packer>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
--> server: Box 'rockylinux10' could not be found. Attempting to find and install...
   server: Box Provider: virtualbox
   server: Box Version: >= 0
--> server: Box file was not detected as metadata. Adding it directly...
--> server: Adding box 'rockylinux10' (<v0>) for provider: virtualbox (amd64)
   server: Downloading rockylinux10...
```

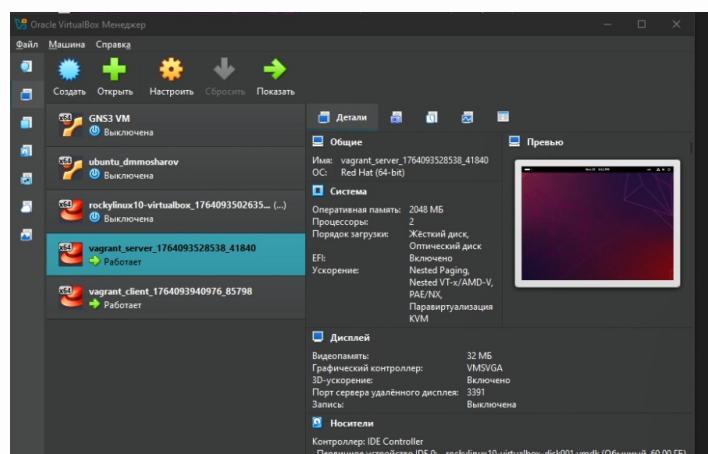
запуск сервера

И запустим еще клиента (рис. [-@fig:017]).

```
C:\work_asp\dmmosharov\packer>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
--> client: Clearing any previously set forwarded ports...
--> client: Fixed port collision for 22 => 2222. Now on port 2200.
--> client: Clearing any previously set network interfaces...
--> client: Preparing network interfaces based on configuration...
   client: Adapter 1: nat
   client: Adapter 2: intnet
--> client: Forwarding ports
```

запуск клиента

Убедимся, что они оба работают, через графический интерфейс. Войдём туда под пользователем vagrant (рис. [-@fig:018]).



Вход через GUI

Теперь попробуем зайти на сервер через ssh, после чего авторизуемся от имени собственного пользователя, и отключимся (рис. [-@fig:019]).

```
Last login: Tue Nov 25 18:05:10 2025  
vagrant@server:~$ su - dmmosharov  
Password:  
[dmmosharov@server.dmmosharov.net ~]$ logout  
vagrant@server:~$ logout
```

Логин на сервере

Сделаем то же самое для клиента (рис. [-@fig:020]).

```
Last failed login: Tue Nov 25 19:14:32 U...  
There was 1 failed login attempt since t...  
/0dmmosharov@client:~$ sudo -i
```

Логин на клиенте

Выключим обе машины (рис. [-@fig:021]).

```
C:\work_asp\dmmosharov\vagrant>vagrant halt server  
==> server: Attempting graceful shutdown of VM...  
  
C:\work_asp\dmmosharov\vagrant>vagrant halt client  
==> client: Attempting graceful shutdown of VM...
```

Завершение работы

## Выводы

В результате выполнения лабораторной работы были получены навыки работы с vagrant

## Список литературы