

# **Отчёт о лабораторной работе**

**Лабораторная работа 1**

Андрюшин Никита Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

# Список иллюстраций

2.1	Создание папок . . . . .	6
2.2	Образ диска . . . . .	6
2.3	vagrant-rocky.pkr.hd . . . . .	7
2.4	Файл конфигурации . . . . .	8
2.5	Vagrantfile . . . . .	9
2.6	Каталог Vagrant . . . . .	9
2.7	Создание подпапок . . . . .	10
2.8	Файл заглушка . . . . .	10
2.9	01-user.sh . . . . .	10
2.10	01-hostname.sh . . . . .	11
2.11	первый файл . . . . .	11
2.12	Второй файл . . . . .	11
2.13	инициализируем packer . . . . .	11
2.14	билд образа . . . . .	12
2.15	добавление образа в vagrant . . . . .	12
2.16	запуск сервера . . . . .	12
2.17	запуск клиента . . . . .	13
2.18	Вход через GUI . . . . .	13
2.19	Логин на сервере . . . . .	14
2.20	Логин на клиенте . . . . .	14
2.21	Завершение работы . . . . .	14

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Для начала создадим папку с инициалами, в которой будет 2 папки, показанные на фото (рис. 2.1).

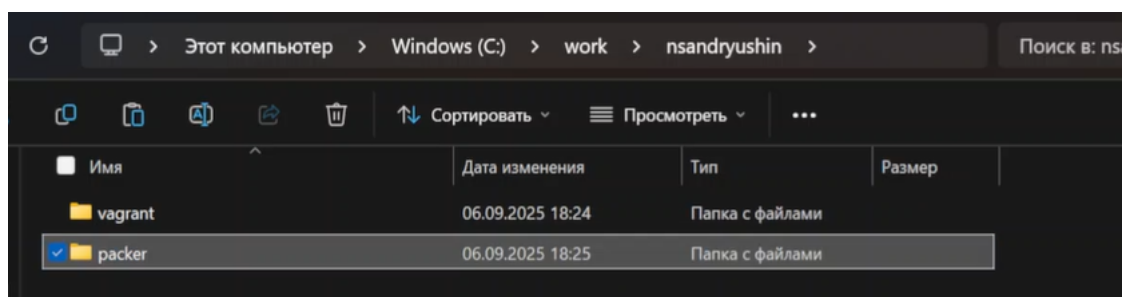


Рис. 2.1: Создание папок

Поместим заранее скачанный образ в первую папку (рис. 2.2).

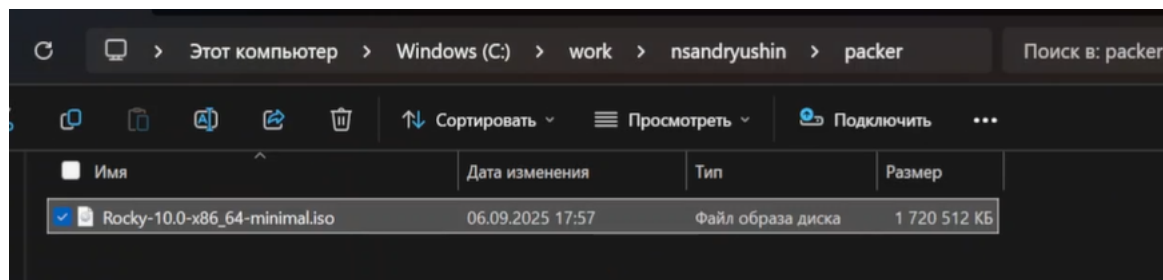
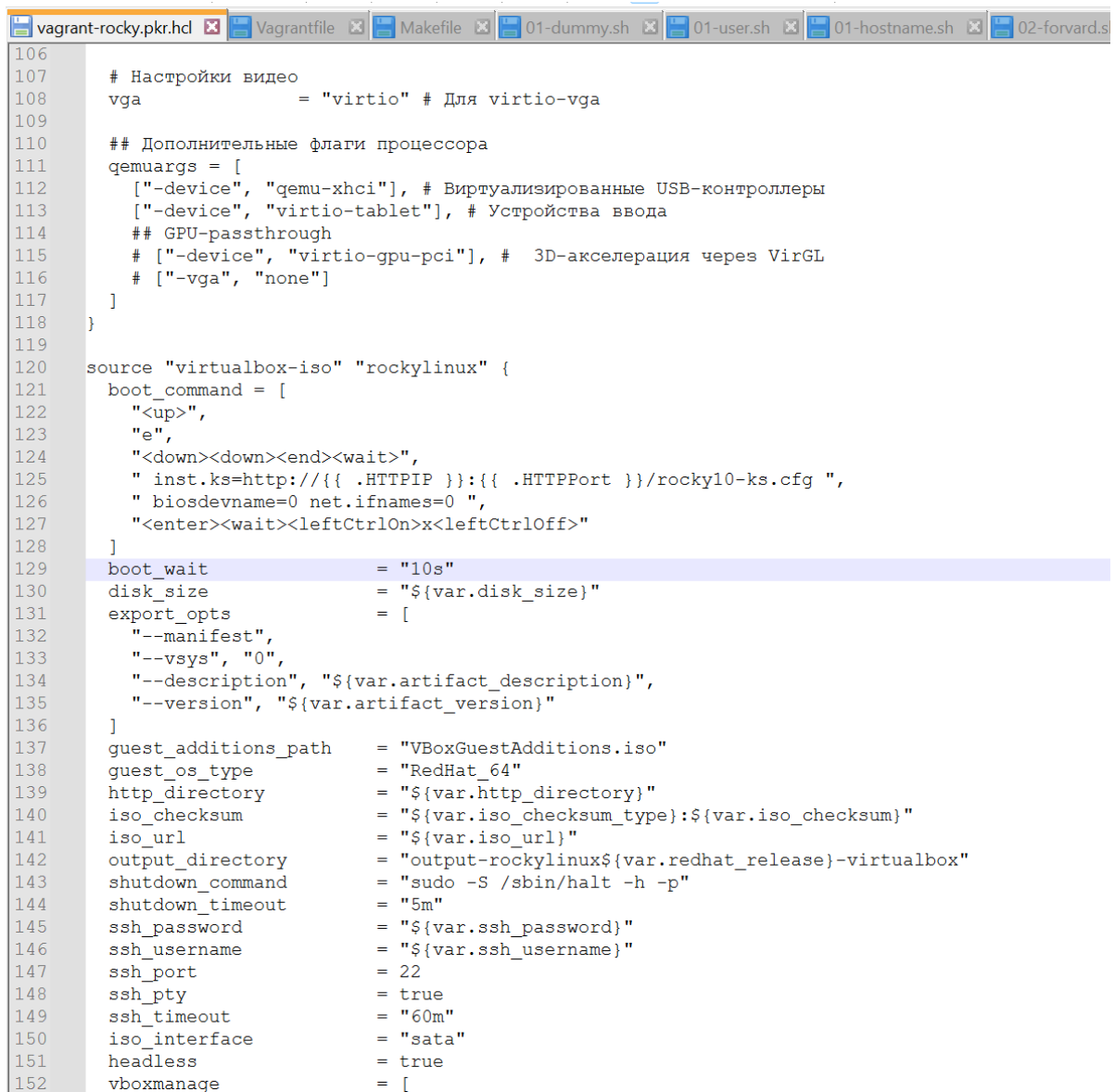


Рис. 2.2: Образ диска

Создадим в той же папке файл, содержание которого перечислено на экране. Тут описано, как будет создаваться образ для vagrant (рис. 2.3).



```

106
107 # Настройки видео
108 vga = "virtio" # Для virtio-vga
109
110 ## Дополнительные флаги процессора
111 qemuargs = [
112   ["-device", "qemu-xhci"], # Виртуализированные USB-контроллеры
113   ["-device", "virtio-tablet"], # Устройства ввода
114   ## GPU-passthrough
115   # ["-device", "virtio-gpu-pci"], # 3D-акселерация через VirGL
116   # ["-vga", "none"]
117 ]
118 }
119
120 source "virtualbox-iso" "rockylinux" {
121   boot_command = [
122     "<up>",
123     "e",
124     "<down><down><end><wait>",
125     " inst.ks=http://{{ .HTTPIP }}:{{ .HTTPPort }}/rocky10-ks.cfg ",
126     " biosdevname=0 net.ifnames=0 ",
127     "<enter><wait><leftCtrlOn>x<leftCtrlOff>"
128   ]
129   boot_wait = "10s"
130   disk_size = "${var.disk_size}"
131   export_opts = [
132     "--manifest",
133     "--vsys", "0",
134     "--description", "${var.artifact_description}",
135     "--version", "${var.artifact_version}"
136   ]
137   guest_additions_path = "VBoxGuestAdditions.iso"
138   guest_os_type = "RedHat_64"
139   http_directory = "${var.http_directory}"
140   iso_checksum = "${var.iso_checksum_type}:${var.iso_checksum}"
141   iso_url = "${var.iso_url}"
142   output_directory = "output-rockylinux${var.redhat_release}-virtualbox"
143   shutdown_command = "sudo -S /sbin/halt -h -p"
144   shutdown_timeout = "5m"
145   ssh_password = "${var.ssh_password}"
146   ssh_username = "${var.ssh_username}"
147   ssh_port = 22
148   ssh_pty = true
149   ssh_timeout = "60m"
150   iso_interface = "sata"
151   headless = true
152   vboxmanage = [

```

Рис. 2.3: vagrant-rocky.pkr.hcl

Создадим подпапку http, в которой поместим файл следующего содержания с конфигурацией (рис. 2.4).

```
vagrant-rocky.pkr.hcl  Vagrantfile  Makefile  01-dummy.sh  01-user.sh  01-hostname.sh  02-forward.sh  01-routing.sh  rocky10-ks.cfg
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 #:%substitute/^(PasswordAuthentication) yes$/\1 no/
62 #:%substitute/^(UseDNS) yes$/&\r\1 no/
63 #:update
64 #:quit
65 #EOF
66 #cat >>/etc/sysconfig/ssh <<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
87 # systemd should generate a new machine id during the first boot, to
88 # avoid having multiple Vagrant instances with the same id in the local
89 # network. /etc/machine-id should be empty, but it must exist to prevent
90 # boot errors (e.g. systemd-journald failing to start).
91 :>/etc/machine-id
92
93 #echo 'vag' > /etc/yum/vars/infra
94
95 # Blacklist the floppy module to avoid probing timeouts
96 echo blacklist floppy > /etc/modprobe.d/nofloppy.conf
97 chcon -u system_u -r object_r -t modules_conf_t /etc/modprobe.d/nofloppy.conf
98
99 # Customize the initramfs
```

Рис. 2.4: Файл конфигурации

Теперь добавим vagrantfile в папку vagrant (рис. 2.5).



```
vagrant-rocky.pkr.hd | Vagrantfile | Makefile | 01-dummy.sh | 01-user.sh | 01-hostname.sh | 02-forward.sh | 01-routing.sh | rocky10-ks.cfg |
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 12Mb
17 |     virtualbox.customize ["modifyvm", :id, "--vram", "32"]
18 |     virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
19 |     virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
20 |     virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
21 |     virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmxvga"]
22 |     virtualbox.customize ["modifyvm", :id, "--accelerate3d", "on"]
23 |     virtualbox.customize ["modifyvm", :id, "--nested-hw-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.disk_bus = "virtio"
32 |     libvirt.nic_model_type = "virtio"
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",
49 |     path: "provision/default/01-hostname.sh"
50 | end
```

Рис. 2.5: Vagrantfile

Содержимое папки будет выглядеть так. Теперь создадим тут папку provision (рис. 2.6).

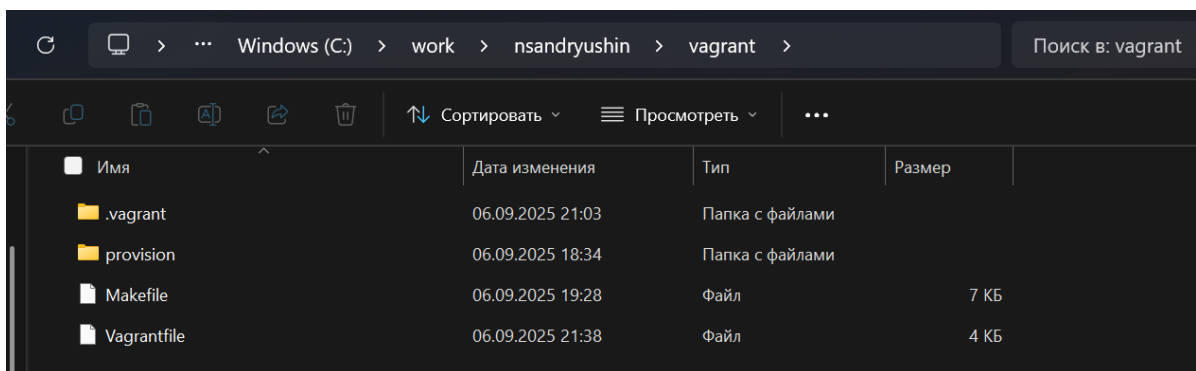


Рис. 2.6: Каталог Vagrant

В этой папке мы создадим 3 подпапки, показанные на фото (рис. 2.7).

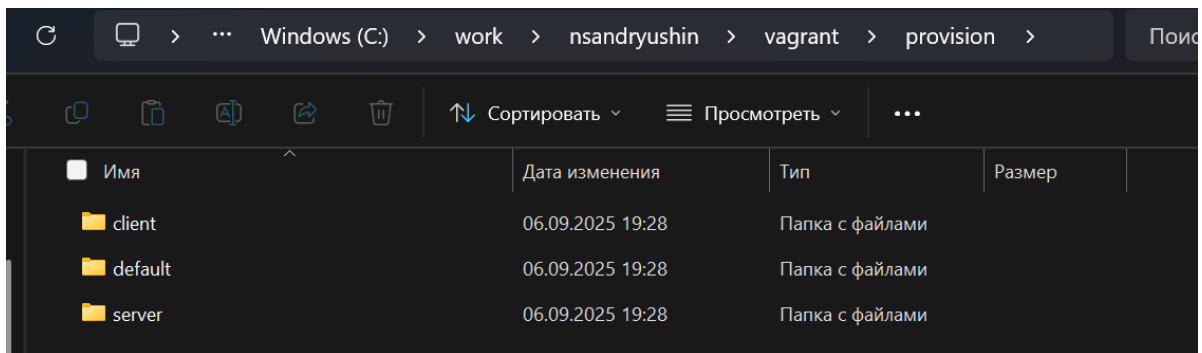


Рис. 2.7: Создание подпапок

В каждую папку добавим файл заглушку со следующим содержанием (рис. 2.8).

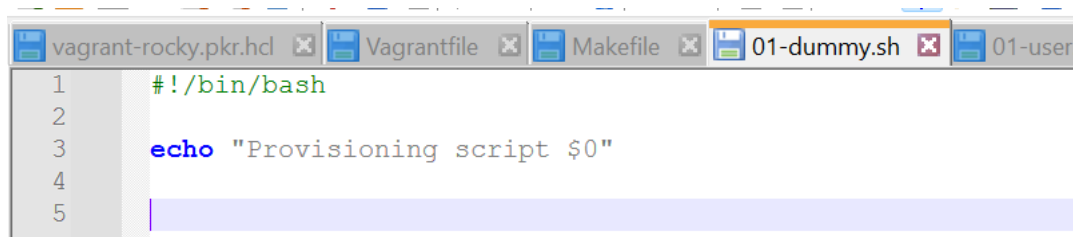


Рис. 2.8: Файл заглушка

В папку default добавим следующий файл (рис. 2.9).

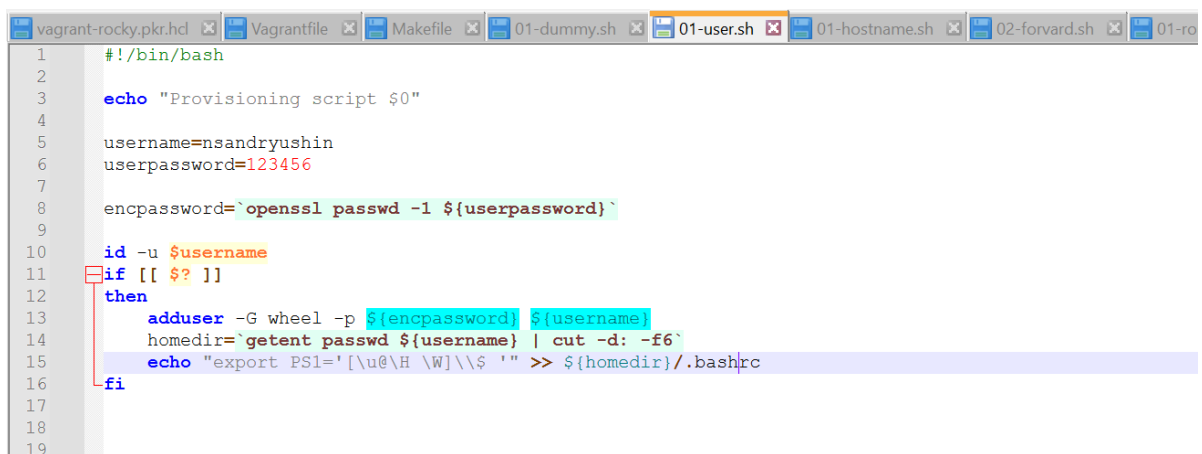


Рис. 2.9: 01-user.sh

И этот файл (рис. 2.10).

```
1 #!/bin/bash
2
3 username=nsandryushin
4
5 hostnamectl set-hostname "${HOSTNAME%%.*}".${username}.net
6
7
8
```

Рис. 2.10: 01-hostname.sh

Добавим также 2 файла в папку сервера. Первый файл (рис. 2.11).

```
1 #!/bin/bash
2 echo "Provisioning script $0"
3 echo "Enable forwarding"
4 echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
5 sysctl -w net.ipv4.ip_forward=1
6 echo "Configure masquerading"
7 firewall-cmd --add-masquerade --permanent
8 firewall-cmd --reload
9 restorecon -vR /etc
```

Рис. 2.11: первый файл

И второй файл (рис. 2.12).

```
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
```

Рис. 2.12: Второй файл

Теперь инициализируем packer (рис. 2.13).

```
C:\work\nsandryushin\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/virtualbox v1.1.2 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/virtualbox_amd64.exe"
Installed plugin github.com/hashicorp/qemu v1.1.3 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/qemu_amd64.exe"
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/vagrant_amd64.exe"
```

Рис. 2.13: инициализируем packer

И сделаем билд образа (рис. 2.14).

```

C:\work\nsandryushin\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 5 milliseconds 826 microseconds: Failed creating Qemu driver: exec: "qemu-system-x86_64": executable f
==> 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/VBox
==> 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%3Ade75c2f7cc566ea964017a1e94883913f066c4eb1d356964e398
==> virtualbox-iso.rockylinux: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4eb1d356964e398ed76cad
y-10.0-x86_64-minimal.iso
==> virtualbox-iso.rockylinux: Starting HTTP server on port 8504
==> 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 3591)
==> 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-localhost reachable 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://127.0.0.1:5975
==> virtualbox-iso.rockylinux: Waiting 10s for boot...

```

Рис. 2.14: билд образа

После этого добавим его в vagrant (рис. 2.15).

```

C:\work\nsandryushin\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/nsandryushin/packer/vagrant-virtualbox-rockylinux10-x86_64.box

```

Рис. 2.15: добавление образа в vagrant

Запустим через вагрант VM сервера (рис. 2.16).

```

C:\work\nsandryushin\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'...
Progress: 60%

```

Рис. 2.16: запуск сервера

И запустим еще клиент (рис. 2.17).

```

C:\work\nsandryushin\vagrant>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Cloning VM...
==> client: Matching MAC address for NAT networking...
==> client: Setting the name of the VM: vagrant_client_1757182730334_3384
==> 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...
client: 22 (guest) => 2200 (host) (adapter 1)
==> client: Running 'pre-boot' VM customizations...
==> client: Booting VM...

```

Рис. 2.17: запуск клиента

Убедимся, что они оба работают, через графический интерфейс. Войдём туда под пользователем vagrant (рис. 2.18).

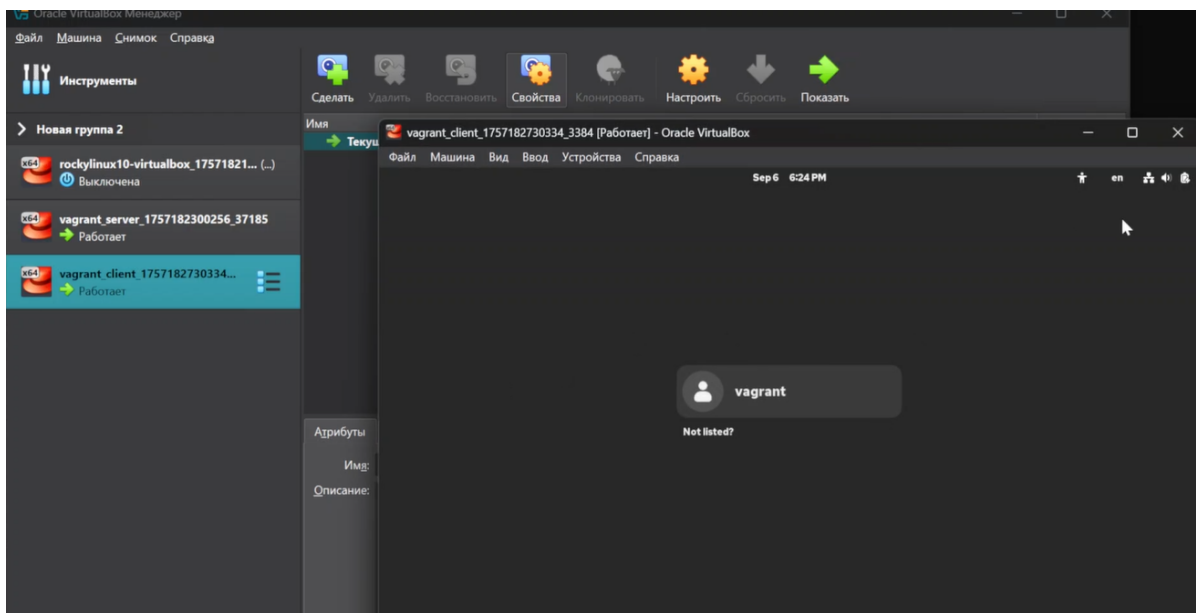


Рис. 2.18: Вход через GUI

Теперь попробуем зайти на сервер через ssh, после чего авторизируемся от имени собственного пользователя, и отключимся (рис. 2.19).

```

C:\work\nsandyushin\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: Sat Sep  6 18:26:57 2025
vagrant@server:~$ su - nsandyushin
Password:
[nsandyushin@server.nsandyushin.net ~]$ ^C
[nsandyushin@server.nsandyushin.net ~]$ exit
logout
vagrant@server:~$

```

Рис. 2.19: Логин на сервере

Сделаем то же самое для клиента (рис. 2.20).

```

C:\work\nsandyushin\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:
vagrant@127.0.0.1's password:
Last failed login: Sat Sep  6 18:51:20 UTC 2025 from 10.0.2.2 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Sat Sep  6 18:30:46 2025 from 10.0.2.2
vagrant@client:~$ su - nsandyushin
Password:
[nsandyushin@client.nsandyushin.net ~]$ exit
logout
vagrant@client:~$ |

```

Рис. 2.20: Логин на клиенте

Выключим обе машины (рис. 2.21).

```

C:\work\nsandyushin\vagrant>vagrant halt server
==> server: Attempting graceful shutdown of VM...

C:\work\nsandyushin\vagrant>vagrant halt client

```

Рис. 2.21: Завершение работы

## **3 Выводы**

В результате выполнения лабораторной работы были получены навыки работы с vagrant

## **Список литературы**