

# Лабораторная работа

Номер 1

---

Андрюшин Н. С.

01 января 1970

Российский университет дружбы народов, Москва, Россия

# Информация

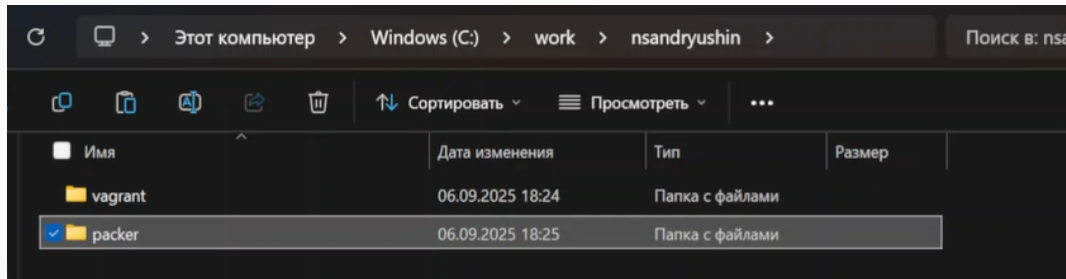
---

- Андрюшин Никита Сергеевич
- Студент
- Российский университет дружбы народов

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

## Создание папок

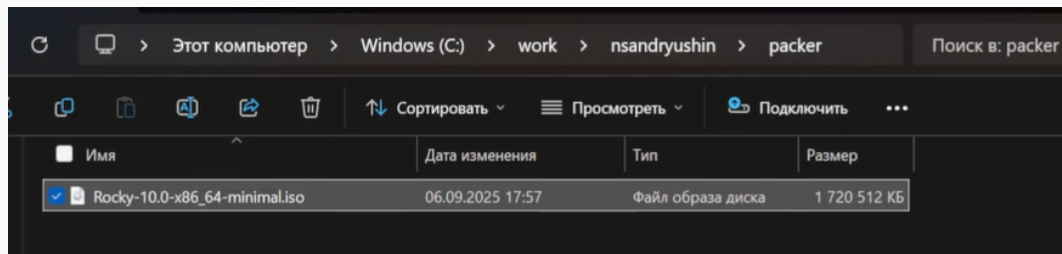
Для начала создадим папку с инициалами, в которой будет 2 папки, показанные на фото



**Рис. 1:** Создание папок

## Образ диска

Поместим заранее скачанный образ в первую папку



**Рис. 2:** Образ диска

```

107 # Застроены 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"], # Устройство ssoca
114      # GPU-passthrough
115      # ["-device", "virtio-gpu-pci", # 3D-ускорения через VlxGL
116       # ["-vga", "none"]
117     ]
118 ]
119
120 source "virtualbox-iso" "rockylinux" {
121     boot_command = [
122         "cnp",
123         "e",
124         "cdrom>cdrom>cds>wait",
125         " * inst.x86/ntp://[! .HTTPort ]::[! .HTTPort ]/rocky10-ks.cfg ",
126         " biosdevname=0 set.lfename=0 ",
127         "<enter><wait><leftCtrlOn><leftCtrlOf>"
128     ]
129     boot_wait = "10s"
130     disk_size = "${var.disk_size}"
131     export_opts = [
132         "--manifest",
133         "--sysa", "0",
134         "--description", "${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 -b -p"
144     shutdown_timeout = "5m"
145     ssh_password = "${var.ssh_password}"
146     ssh_username = "${var.ssh_username}"
147     ssh_port = 22
148     ssh_ptty = true
149     ssh_timeout = "60m"
150     iso_interface = "sata"
151     headless = true
152     vboxmanage = [

```

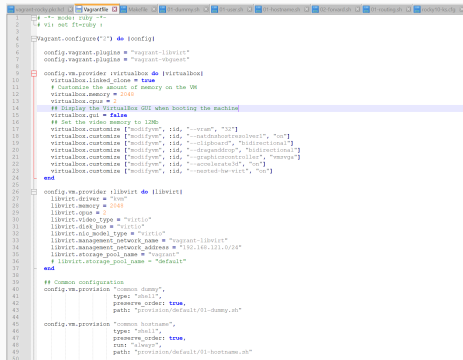
Создадим подпапку http, в которой поместим файл следующего содержания с конфигурацией

```
14 # echo "/usr/sbin:root:x:0:0:0:/usr/sbin" > /etc/passwd
15
16 # sudo
17 echo "Vagrant ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/vagrant
18 chmod 0440 /etc/sudoers.d/vagrant
19
20 # Fix for https://github.com/CloudCannon/vagrant-ssh-config/issues/10
21 cat > /etc/ssh/config <<EOF
22 Host *
23   AddKeysToAgent yes
24   UseKeychain yes
25   ForwardAgent yes
26   ForwardX11 yes
27 EOF
28
29 # sudo disable password authentication and SSH checks
30 sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
31 # sudo substitute /etc/ssh/sshd_config with /etc/ssh/sshd_config.d/ssh_config.d/ssh_config
32 # sudo substitute /etc/ssh/sshd_config with /etc/ssh/sshd_config.d/ssh_config
33 # sudo
34 # sudo
35 # sudo
36 # sudo
37 # sudo
38 # sudo
39 # sudo
40 # sudo
41 # sudo
42 # sudo
43 # sudo
44 # sudo
45 # sudo
46 # sudo
47 # sudo
48 # sudo
49 # sudo
50 # sudo
51 # sudo
52 # sudo
53 # sudo
54 # sudo
55 # sudo
56 # sudo
57 # sudo
58 # sudo
59 # sudo
60 # sudo
61 # sudo
62 # sudo
63 # sudo
64 # sudo
65 # sudo
66 # sudo
67 # sudo
68 # sudo
69 # sudo
70 # sudo
71 # sudo
72 # sudo
73 # sudo
74 # sudo
75 # sudo
76 # sudo
77 # sudo
78 # sudo
79 # sudo
80 # sudo
81 # sudo
82 # sudo
83 # sudo
84 # sudo
85 # sudo
86 # sudo
87 # sudo
88 # sudo
89 # sudo
90 # sudo
91 # sudo
92 # sudo
93 # sudo
94 # sudo
95 # sudo
96 # sudo
97 # sudo
98 # sudo
99 # sudo
100 # sudo
```

Рис. 4: Файл конфигурации



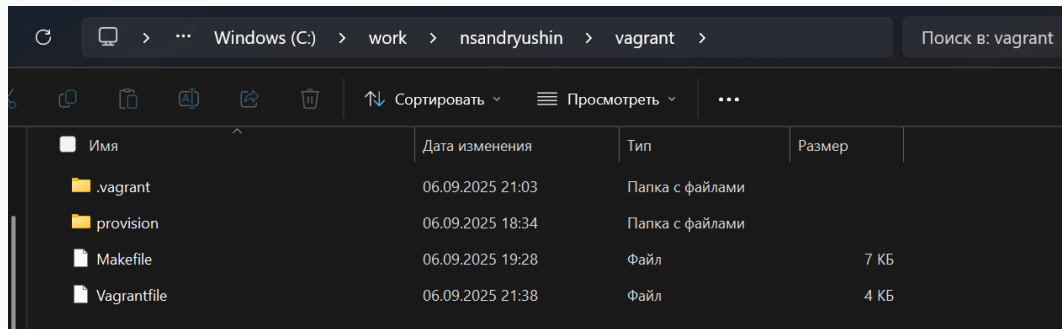
Теперь добавим vagrantfile в папку vagrant



```
1 # -*- mode: ruby -*-
2 # vi: set filetype=ruby :
3
4
5 Vagrant.configure("2") do |config|
6   config.vagrant.plugins = ["vagrant-libvirt",
7                             "vagrant-vboxmanage"]
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
15     ## Display the VirtualBox GUI when booting the machine
16     virtualbox.gui = false
17     ## Set the video memory to 128M
18     virtualbox.customize ["modifyvm", :id, "--vram", "128"]
19     virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
20     virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
21     virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
22     virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vboxvga"]
23     virtualbox.customize ["modifyvm", :id, "--accelerate3d", "on"]
24     virtualbox.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
25   end
26
27   config.vm.provider :libvirt do |libvirt|
28     libvirt.driver = "kvm"
29     libvirt.memory = 2048
30     libvirt.cpus = 2
31     libvirt.virtio_type = "virtio"
32     libvirt.virtio_model = "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. 5: Vagrantfile

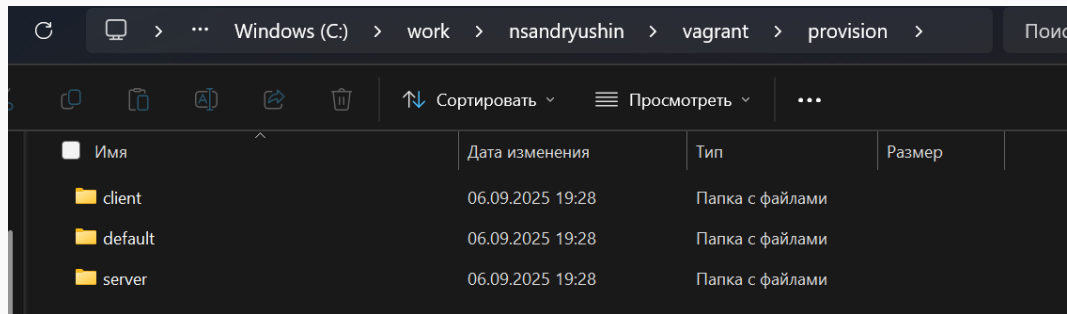
Содержимое папки будет выглядеть так. Теперь создадим тут папку provision



**Рис. 6:** Каталог Vagrant

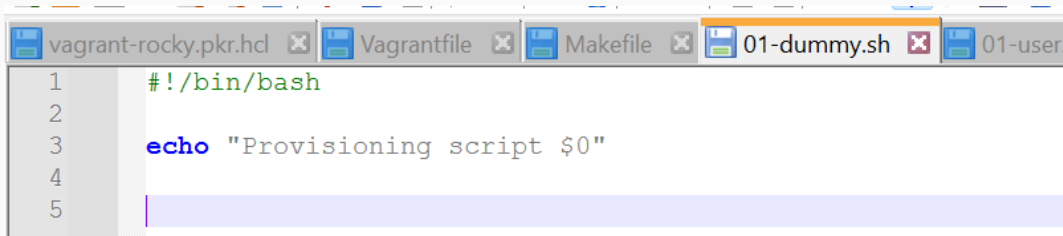
## Создание подпапок

В этой папке мы создадим 3 подпапки, показанные на фото



**Рис. 7:** Создание подпапок

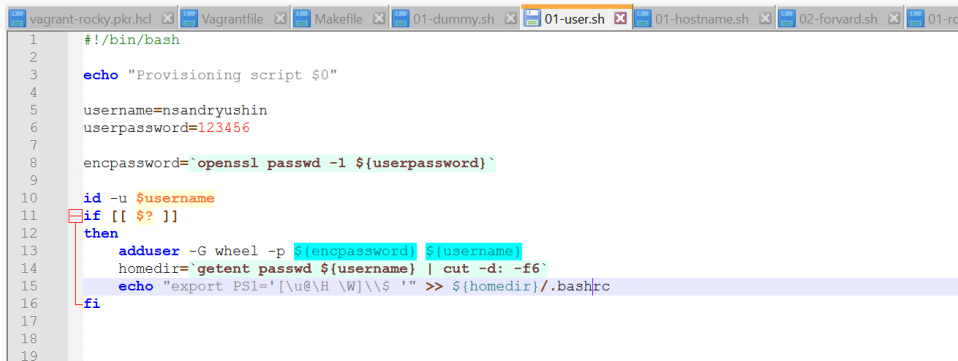
В каждую папку добавим файл заглушку со следующим содержанием



```
vagrant-rocky.pkr.hcl Vagrantfile Makefile 01-dummy.sh 01-user
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5
```

**Рис. 8:** Файл заглушка

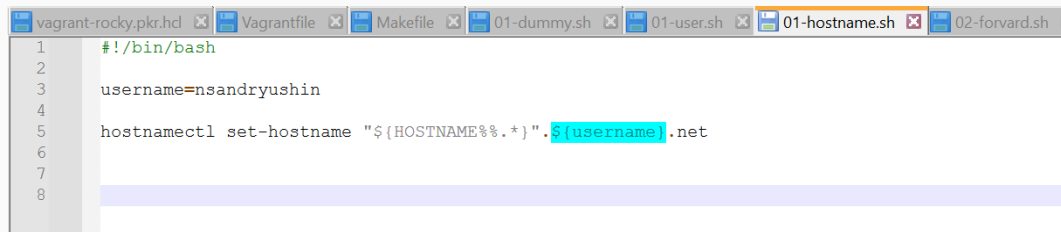
В папку default добавим следующий файл



```
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  username=nsandryushin
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
```

Рис. 9: 01-user.sh

И этот файл




The image shows a terminal window with a tabbed interface. The tabs include 'vagrant-rocky.pkr.hcl', 'Vagrantfile', 'Makefile', '01-dummy.sh', '01-user.sh', '01-hostname.sh' (which is the active tab), and '02-forward.sh'. The active tab displays the following script content:

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

**Рис. 10:** 01-hostname.sh

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



The image shows a terminal window with a title bar containing several open files: 'vagrant-rocky.pkr.hd', 'Vagrantfile', 'Makefile', '01-dummy.sh', '01-user.sh', '01-hostname.sh', '02-forward.sh' (which is highlighted), '01-routing.sh', and 'rocky10-ks.cfg'. The terminal content is a shell script with the following lines:

```
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. 11:** первый файл

## И второй файл



The screenshot shows a terminal window with a tab bar at the top containing several files: vagrant-rocky.pkr.hd, Vagrantfile, Makefile, 01-dummy.sh, 01-user.sh, 01-hostname.sh, 02-forward.sh, 01-routing.sh, rocky10-ks.cfg, and 02-forward.sh (which is the active tab). The terminal content is as follows:

```
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. 12:** Второй файл



## инициализируем packer

Теперь инициализируем packer

```
C:\work\nsandyushin\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/v1.1.2/virtualbox_windows_amd64.exe"
Installed plugin github.com/hashicorp/qemu v1.1.3 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/qemu/v1.1.3/qemu-windows-amd64.exe"
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/vagrant/v1.1.5/vagrant64.exe"
```

**Рис. 13:** инициализируем packer

## И сделаем билд образа

```
C:\work\insandryushin\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%3Ade75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398
==> virtualbox-iso.rockylinux: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cad
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-localhostreachabl 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. 14: билд образа

После этого добавим его в vagrant

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

**Рис. 15:** добавление образа в vagrant

## запуск сервера

Запустим через вагрант ВМ сервера

```
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. 16: запуск сервера

## запуск клиента

И запустим еще клиент

```
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. 17: запуск клиента

# Вход через GUI

Убедимся, что они оба работают, через графический интерфейс. Войдём туда под пользователем vagrant

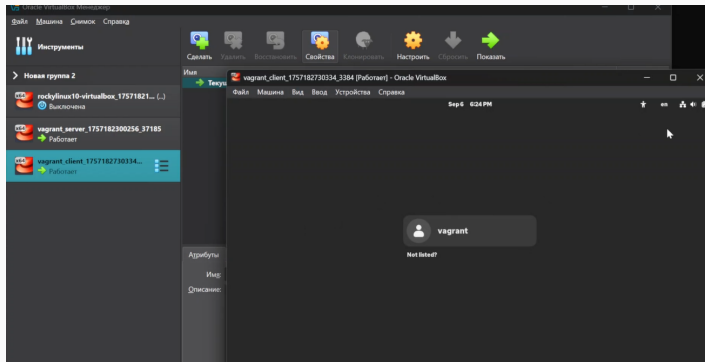


Рис. 18: Вход через GUI

## Логин на сервере

Теперь попробуем зайти на сервер через ssh, после чего авторизируемся от имени собственного пользователя, и отключимся

```
C:\work\nsandryushin\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 - nsandryushin
Password:
[nsandryushin@server.nsandryushin.net ~]$ ^C
[nsandryushin@server.nsandryushin.net ~]$ exit
logout
vagrant@server:~$
```

**Рис. 19:** Логин на сервере

Сделаем то же самое для клиента

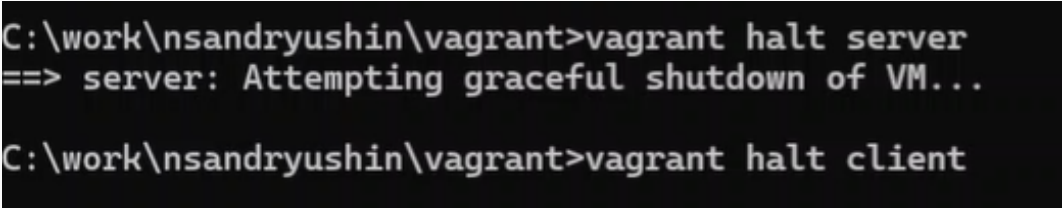
```
C:\work\nsandyushin\vagrant>vagrant ssh client
==> client: The machine you're attempting to SSH into is configured to use
==> client: password-based authentication. Vagrant can't script entering the
==> client: password for you. If you're prompted for a password, please enter
==> client: the same 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. 20:** Логин на клиенте



## Завершение работы

Выключим обе машины



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

**Рис. 21:** Завершение работы

В результате выполнения лабораторной работы были получены навыки работы с vagrant