

Отчёт о лабораторной работе

Лабораторная работа 1

Мантуров Татархан Бесланович

Содержание

Цель работы	2
Выполнение лабораторной работы	2
Выводы	11
Список литературы	11

Список иллюстраций

Создание папок	2
Образ диска	3
vagrant-rocky.pkr.hd	4
Файл конфигурации	5
Vagrantfile	6
Каталог Vagrant	6
Создание подпапок	7
Файл заглушка	7
01-user.sh	7
01-hostname.sh	7
первый файл	8
Второй файл	8
инициализируем packer	8
билд образа	9
добавление образа в vagrant	9
запуск сервера	9
запуск клиента	10
Вход через GUI	10
Логин на сервере	10
Логин на клиенте	11
Завершение работы	11

Список таблиц

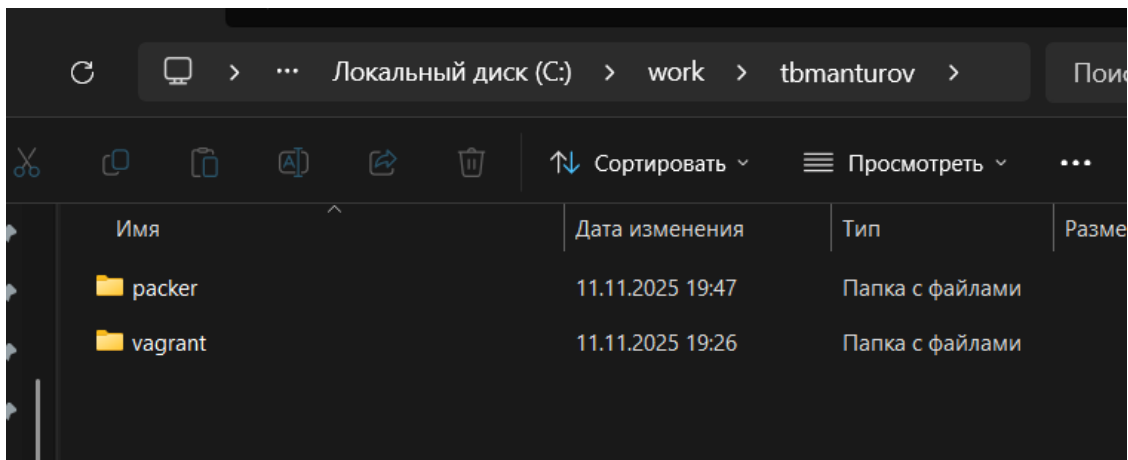
Элементы списка иллюстраций не найдены.

Цель работы

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

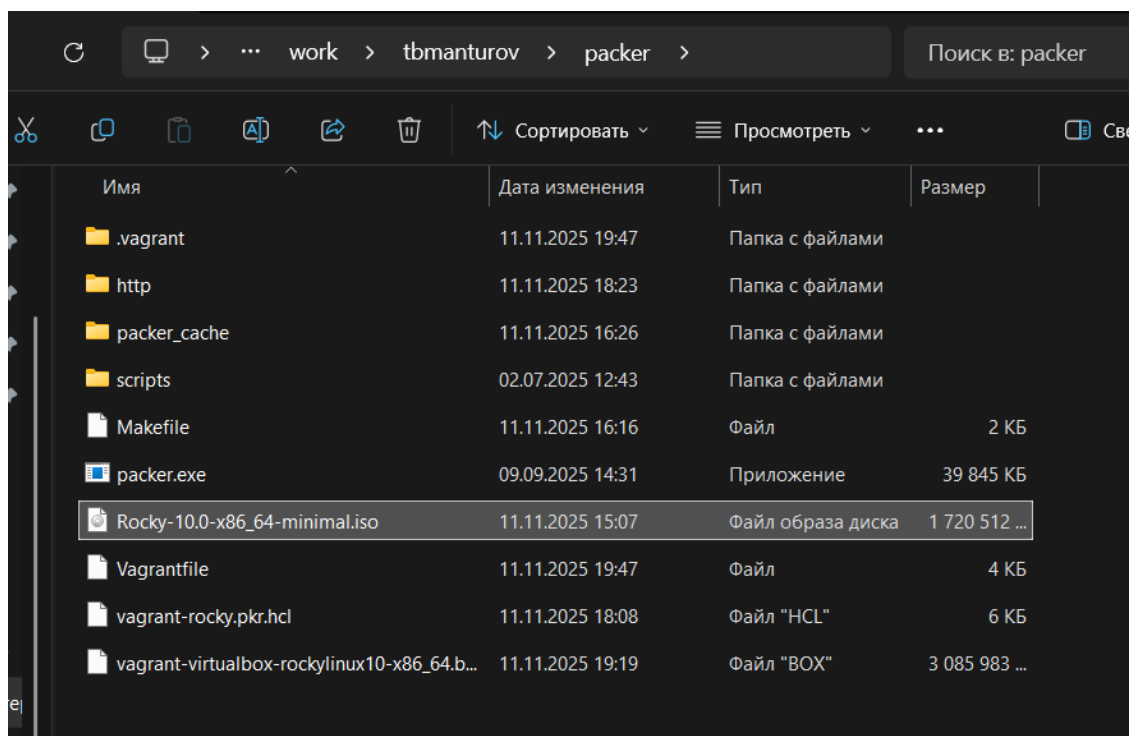
Выполнение лабораторной работы

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



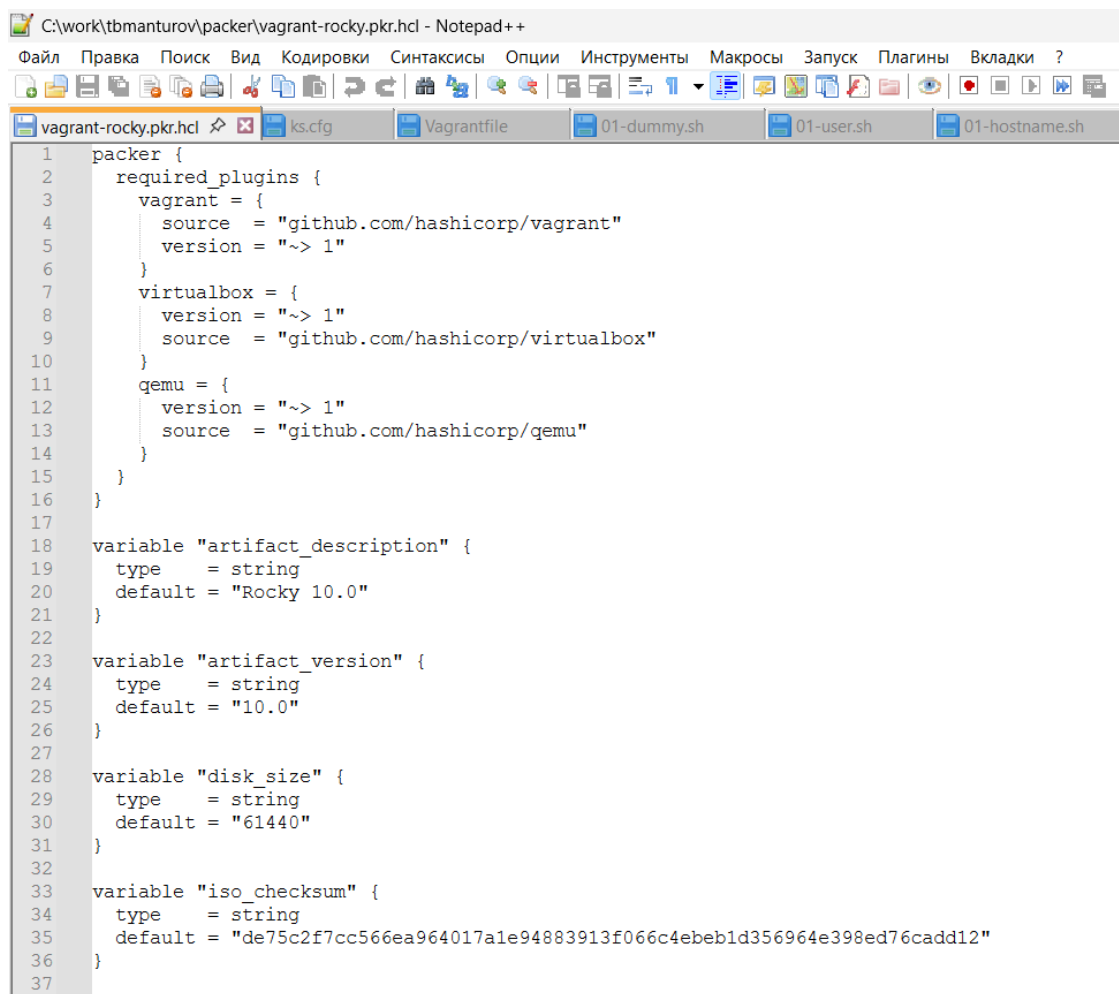
Создание папок

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]).



```
1 packer {
2   required_plugins {
3     vagrant = {
4       source = "github.com/hashicorp/vagrant"
5       version = "~> 1"
6     }
7     virtualbox = {
8       version = "~> 1"
9       source = "github.com/hashicorp/virtualbox"
10    }
11    qemu = {
12      version = "~> 1"
13      source = "github.com/hashicorp/qemu"
14    }
15  }
16 }
17
18 variable "artifact_description" {
19   type = string
20   default = "Rocky 10.0"
21 }
22
23 variable "artifact_version" {
24   type = string
25   default = "10.0"
26 }
27
28 variable "disk_size" {
29   type = string
30   default = "61440"
31 }
32
33 variable "iso_checksum" {
34   type = string
35   default = "de75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cadd12"
36 }
37
```

vagrant-rocky.pkr.hcl

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

```

# 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 --vkeymap=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

```

Файл конфигурации

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

```
vagrant-rocky.pkr.hcl  ks.cfg  Vagrantfile  01-dummy.sh  01-user.sh  01-hosts
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", "vmsvga"]
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
end
```

Vagrantfile

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

Имя	Дата изменения	Тип	Размер
.vagrant	11.11.2025 19:43	Папка с файлами	
provision	11.11.2025 14:10	Папка с файлами	
Makefile	11.11.2025 16:16	Файл	
Vagrantfile	11.11.2025 16:17	Файл	

Каталог Vagrant

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

Имя	Дата изменения	Тип	Р
server	11.11.2025 14:14	Папка с файлами	
default	11.11.2025 14:12	Папка с файлами	
client	11.11.2025 14:14	Папка с файлами	

Создание подпапок

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

```

1  #!/bin/bash
2  echo "Provisioning script $0"
3

```

Файл заглушка

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

```

1  #!/bin/bash
2  echo "Provisioning script $0"
3  username=tbmanturov
4  userpassword=123456
5  encpassword=`openssl passwd -1 ${userpassword}`
6  id -u $username
7  if [[ $? ]]
8  then
9      adduser -G wheel -p ${encpassword} ${username}
10     homedir=`getent passwd ${username} | cut -d: -f6`
11     echo "export PS1='[\u@\H \W]\\$ ' >> ${homedir}/.bashrc
12 fi

```

01-user.sh

И этот файл (рис. [-@fig:010]).

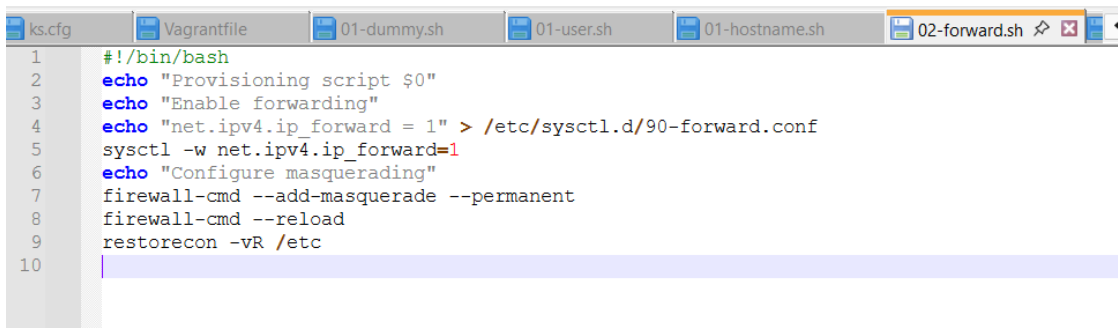
```

1  #!/bin/bash
2  username=tbmanturov
3  hostnamectl set-hostname "${HOSTNAME%%.*}.${username}.net"

```

01-hostname.sh

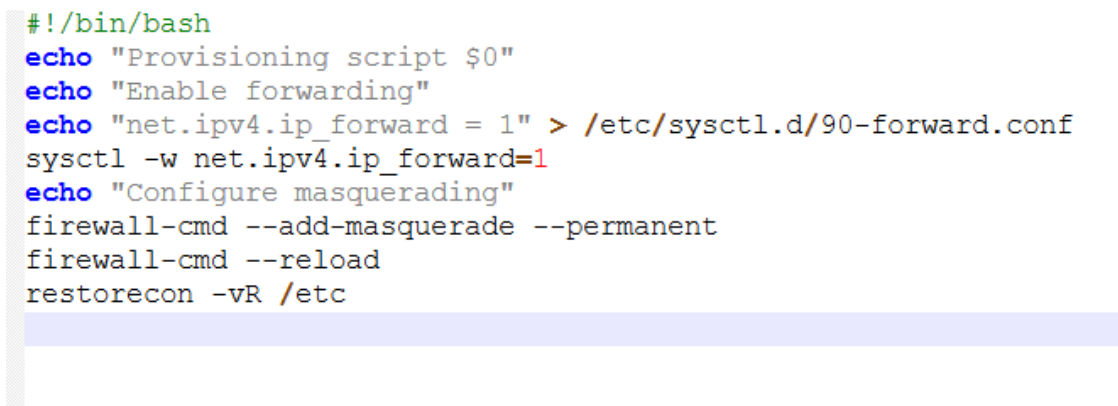
Добавим также 2 файла в папку сервера. Первый файл (рис. [-@fig:011]).



```
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
10
```

первый файл

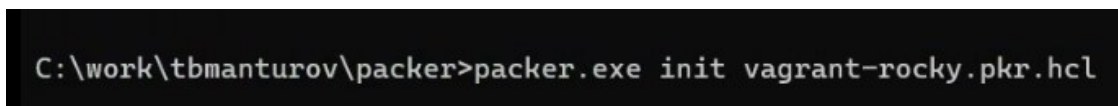
И второй файл (рис. [-@fig:012]).



```
#!/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
```

Второй файл

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



```
C:\work\tbmanturov\packer>packer.exe init vagrant-rocky.pkr.hcl
```

инициализируем packer

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


```

==> 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
so
==> 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%3Ade75c2f7cc566ea964017a1e9
==> virtualbox-iso.rockylinux: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913
bmanturov/packer/Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.rockylinux: Starting HTTP server on port 8022
==> virtualbox-iso.rockylinux: Creating virtual machine...
==> virtualbox-iso.rockylinux: Creating hard drive output-rockylinux10-virtualbox\rockylinux10-virtualbox.vdi w
==> 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 39
==> 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://127.0.0.1:5935
==> virtualbox-iso.rockylinux: Waiting 10s for boot...
==> virtualbox-iso.rockylinux: Typing the boot command...

```

билд образа

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

```

C:\work\tbmanturov\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' (v0) for provider: (amd64)
box: Unpacking necessary files from: file:///C:/work/tbmanturov/packer/vagrant-virtualbox-rockylinux10-x86_64.box
box:
==> box: Successfully added box 'rocky10' (v0) for '(amd64)'!

```

добавление образа в vagrant

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

```

C:\work\tbmanturov\vagrant>vagrant up server
Vagrant has detected project local plugins configured for this
project which are not installed.

vagrant-vbguest
Install local plugins (Y/N) [N]: Y
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...

```

запуск сервера

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

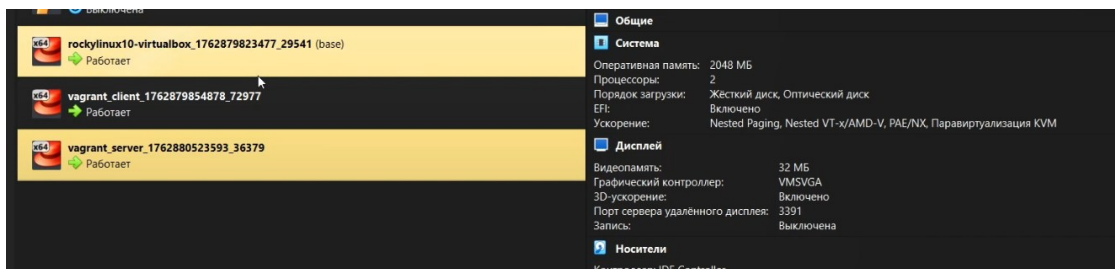
```

==> 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!
[server] GuestAdditions seems to be installed (7.2.0) correctly, but not running.
Redirecting to /bin/systemctl start vboxadd.service
Redirecting to /bin/systemctl start vboxadd-service.service
grep: warning: stray \ before /
grep: warning: stray \ before /
grep: warning: stray \ before /
grep: warning: stray \ before /
grep: warning: stray \ before /
grep: warning: stray \ before /

```

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 авторизируемся от имени собственного пользователя, и отключимся (рис. [-@fig:019]).

```

C:\work\tbmanturov\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: Tue Nov 11 17:15:34 2025
vagrant@server:~$ su -tbmanturov
su: invalid option -- 't'
Try 'su --help' for more information.
vagrant@server:~$ su - tbmanturov
Password:
su: Authentication failure
vagrant@server:~$ su - tbmanturov
Password:
Last failed login: Tue Nov 11 17:19:16 UTC 2025 on pts/0
There were 6 failed login attempts since the last successful login.
[tbmanturov@server.tbmanturov.net ~]$ exit
logout

```

Логин на сервере

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

```
C:\work\tbmanturov\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: Tue Nov 11 17:18:39 2025 from 10.0.2.2
vagrant@server:~$ su - tbmanturov
Password:
Last login: Tue Nov 11 17:19:32 UTC 2025 on pts/0
[tbmanturov@server.tbmanturov.net ~]$ exit
logout
vagrant@server:~$ exit
```

Логин на клиенте

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

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

C:\work\tbmanturov\vagrant>vagrant halt server
|
```

Завершение работы

Выводы

В результате выполнения лабораторной работы были получены навыки работы с vagrant

Список литературы