

# **Лабораторная работа №1**

**Подготовка лабораторного стенда**

Демидова Екатерина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Контрольные вопросы</b>	<b>14</b>
<b>5</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

3.1	Скрипт 01-user.sh . . . . .	6
3.2	Скрипт 01-hostname.sh . . . . .	7
3.3	Формирование box-файла . . . . .	8
3.4	Регистрация образа . . . . .	9
3.5	Команда make server-up . . . . .	9
3.6	Успешный запуск виртуальной машины Server . . . . .	10
3.7	Команда make client-up . . . . .	10
3.8	Успешный запуск виртуальной машины Client . . . . .	11
3.9	Проверка конфигурационного файла Vagrant . . . . .	11
3.10	Команда make server-provision . . . . .	12
3.11	Команда make client-provision . . . . .	12
3.12	Проверка работы сервера . . . . .	13
3.13	Проверка работы клиента . . . . .	13
4.1	Команды Vagrant . . . . .	15

# 1 Цель работы

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

## 2 Задание

1. Сформируйте box-файл с дистрибутивом Rocky Linux для VirtualBox
2. Запустите виртуальные машины сервера и клиента и убедитесь в их работоспособности.
3. Внесите изменения в настройки загрузки образов виртуальных машин server и client, добавив пользователя с правами администратора и изменив названия хостов
4. Скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, вы можете попробовать развернуть виртуальные машины на другом компьютере.

### 3 Выполнение лабораторной работы

Перед началом работы с Vagrant создадим каталог в /var/tmp с помощью команд:

```
mkdir -p /var/tmp/user_name/packer
```

```
mkdir -p /var/tmp/user_name/vagran
```

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux и в этом же каталоге разместим подготовленные заранее для работы с Vagrant файлы: vagrant-rocky.pkr.hc, ks.cfg, Vagrantfile, Makefile.

В этом же каталоге создадим каталог provision с подкаталогами default, server и client, в которых будут размещаться скрипты, изменяющие настройки внутреннего окружения базового (общего) образа виртуальной машины, сервера или клиента соответственно. В каталогах default, server и client разместим заранее подготовленный скриптзаглушку 01-dummy.sh следующего содержания:

```
#!/bin/bash
```

```
echo "Provisioning script $0"
```

В каталоге default разместим заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины следующего содержания(рис. 3.1):

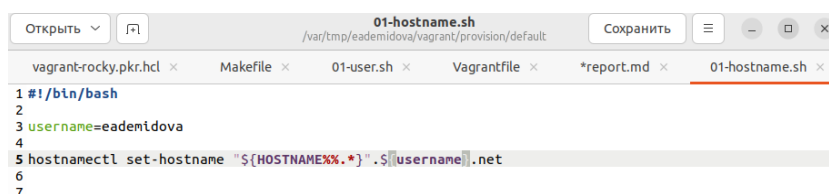


Рис. 3.1: Скрипт 01-user.sh

В каталоге default разместим заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины следующего содержания(рис. 3.2):



```
01-user.sh
/var/tmp/eademidova/vagrant/provision/default

vagrant-rocky.pkr.hcl x Makefile x 01-user.sh x Vagrantfile x report.md x

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

Рис. 3.2: Скрипт 01-hostname.sh

Перейдем в каталог с проектом:

```
cd /var/tmp/user_name/packer
```

В терминале наберем

```
make help
```

Посмотрим перечень указанных в Makefile целей и краткое описание их действий.

Для формирования box-файла с дистрибутивом Rocky Linux для VirtualBox в терминале наберем make(рис. 3.3):

```
edemidova@edemidova ThinkBook-14s Yoga (TL) /var/tmp/edemidova/packer$ make
Inspect      see components of a template
Validate     check that a template is valid
Version      Prints the Packer version

make: *** [pkcfile: /var/tmp/edemidova/packer$ make
Warning: Bundled Plugins used

This template relies on the use of plugins bundled into the Packer binary.
The practice of bundling external plugins into Packer will be removed in an
upcoming version.

To remove this warning, add the following section to your template:

packer {
  required_plugins {
    vagrant = [
      {
        source = "github.com/hashicorp/vagrant"
        version = "~> 2"
      }
    ]
  }
}

Then run 'packer init' to manage installation of the plugins

virtualbox-iso:virtualbox: output will be in this color:
== virtualbox-iso.virtualbox: Retrieving Guest additions
== virtualbox-iso.virtualbox: Trying /usr/share/virtualbox/WindowsAdditions.iso
== virtualbox-iso.virtualbox: Trying /usr/share/virtualbox/WindowsAdditions.iso
== virtualbox-iso.virtualbox: Trying /usr/share/virtualbox/WindowsAdditions.iso
== virtualbox-iso.virtualbox: Retrieving ISO
== virtualbox-iso.virtualbox: Trying rocky-9-2-xx86_64-virtual.iso
== virtualbox-iso.virtualbox: Trying rocky-9-2-xx86_64-virtual.iso
== virtualbox-iso.virtualbox: Starting VMX server on port 8322
== virtualbox-iso.virtualbox: Creating virtual machine...
== virtualbox-iso.virtualbox: Creating hard drive build/packer-rocky-virtualbox-vm.vdi with size 40960 MB...
== virtualbox-iso.virtualbox: Mounting ISO
== virtualbox-iso.virtualbox: Creating Forwarded port mapping for communicator (500, 5000, etc) (host port 3000)
== virtualbox-iso.virtualbox: Executing custom VMChange Commands...
== virtualbox-iso.virtualbox: Executing: VBoxManage modifyvm packer-rocky-virtualbox-vm --memory 2048
== virtualbox-iso.virtualbox: Executing: VBoxManage modifyvm packer-rocky-virtualbox-vm --sata 1
== virtualbox-iso.virtualbox: Executing: VBoxManage modifyvm packer-rocky-virtualbox-vm --usb-lun0 hostproxscholar1 on
== virtualbox-iso.virtualbox: Starting the virtual machine...
== virtualbox-iso.virtualbox: Waiting for boot...
== virtualbox-iso.virtualbox: Typing the boot command...
== virtualbox-iso.virtualbox: Waiting for SSH communicator to connect: 127.0.0.1
== virtualbox-iso.virtualbox: Waiting for SSH to become available...
```

Рис. 3.3: Формирование box-файла

Начался процесс скачивания, распаковки и установки драйверов VirtualBox и дистрибутива ОС на виртуальную машину.

После завершения процесса автоматического развёртывания образа виртуальной машины в каталоге `/var/tmp/user_name/vagrant` временно появился каталог `builds` с промежуточными файлами `.vdi`, `.vmdk` и `.ovf`, которые затем автоматически будут преобразованы в box-файл сформированного образа: `vagrant-virtualbox-rocky-9-x86_64.box`.

Для регистрации образа виртуальной машины в Vagrant в терминале в каталоге `/var/tmp/user_name/vagrant` наберем `make addbox`(рис. 3.4):



```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ cd ..
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova$ cd vagrant/
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make help
Usage:
make <target>

Targets:
addbox          Add the built box to Vagrant
client-destroy  Destroy client
client-provision Up and provision client
client-up       Up client
plugins         Install plugins
server-destroy  Destroy server
server-provision Up and provision server
server-up       Up server
up              Up boxies

eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make addbox
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky9' (v0) for provider:
box: Unpacking necessary files from: file:///var/tmp/eademidova/vagrant/vagrant-virtualbox-rocky-9-x86_
64.box
==> box: Successfully added box 'rocky9' (v0) for ''!
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$
```

Рис. 3.4: Регистрация образа

Это позволит на основе конфигурации, прописанной в файле Vagrantfile, сформировать box-файлы образов двух виртуальных машин — сервера и клиента с возможностью их параллельной или индивидуальной работы.

Запустим виртуальную машину Server, введя `make server-up`(рис. 3.5, 3.6):

```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make addbox
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky9' (v0) for provider:
box: Unpacking necessary files from: file:///var/tmp/eademidova/vagrant/vagrant-virtualbox-rocky-9-x86_
64.box
==> box: Successfully added box 'rocky9' (v0) for ''!
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make server-up
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Fetching micromachine-3.0.0.gem
Fetching vagrant-vbguest-0.31.0.gem
Installed the plugin 'vagrant-vbguest (0.31.0)'!
touch plugins
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".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 'rocky9'...
```

Рис. 3.5: Команда `make server-up`

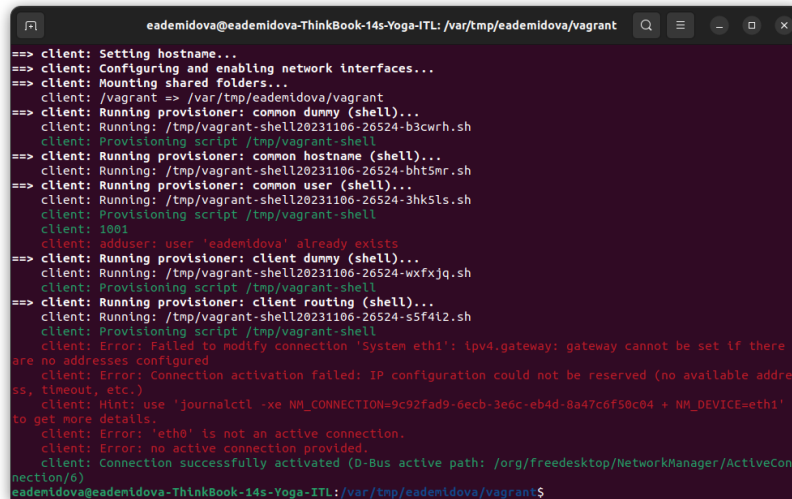
```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant
server: SSH auth method: password
==> server: Machine booted and ready!
Got different reports about installed GuestAdditions version:
Virtualbox on your host claims:
VBoxService inside the vm claims: 7.0.12
Going on, assuming VBoxService is correct...
[server] GuestAdditions seems to be installed (7.0.12) correctly, but not running.
Got different reports about installed GuestAdditions version:
Virtualbox on your host claims:
VBoxService inside the vm claims: 7.0.12
Going on, assuming VBoxService is correct...
Redirecting to /bin/systemctl start vboxadd.service
Redirecting to /bin/systemctl start vboxadd-service.service
==> server: Checking for guest additions in VM...
==> server: Setting hostname...
==> server: Configuring and enabling network interfaces...
==> server: Mounting shared folders...
server: /vagrant => /var/tmp/eademidova/vagrant
==> server: Running provisioner: common dummy (shell)...
server: Running: /tmp/vagrant-shell20231106-17060-5lx2w3.sh
server: Provisioning script /tmp/vagrant-shell
==> server: Running provisioner: common hostname (shell)...
server: Running: /tmp/vagrant-shell20231106-17060-1mwmac.sh
==> server: Running provisioner: common user (shell)...
server: Running: /tmp/vagrant-shell20231106-17060-ofej7n.sh
server: Provisioning script /tmp/vagrant-shell
server: id: 'eademidova': no such user
==> server: Running provisioner: server dummy (shell)...
server: Running: /tmp/vagrant-shell20231106-17060-115k42.sh
server: Provisioning script /tmp/vagrant-shell
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$
```

Рис. 3.6: Успешный запуск виртуальной машины Server

Запустим виртуальную машину Client, введя make client-up(рис. 3.7, 3.8):

```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant
==> server: Running provisioner: common dummy (shell)...
server: Running: /tmp/vagrant-shell20231106-19608-98kg2b.sh
server: Provisioning script /tmp/vagrant-shell
==> server: Running provisioner: common hostname (shell)...
server: Running: /tmp/vagrant-shell20231106-19608-tkzgex.sh
==> server: Running provisioner: common user (shell)...
server: Running: /tmp/vagrant-shell20231106-19608-venfgo.sh
server: Provisioning script /tmp/vagrant-shell
server: id: 'eademidova': no such user
==> server: Running provisioner: server dummy (shell)...
server: Running: /tmp/vagrant-shell20231106-19608-o5jvfe.sh
server: Provisioning script /tmp/vagrant-shell
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make client-up
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Cloning VM...
==> client: Matching MAC address for NAT networking...
==> client: Setting the name of the VM: client
==> 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...
==> client: Waiting for machine to boot. This may take a few minutes...
client: SSH address: 127.0.0.1:2200
client: SSH username: vagrant
client: SSH auth method: password
```

Рис. 3.7: Команда make client-up

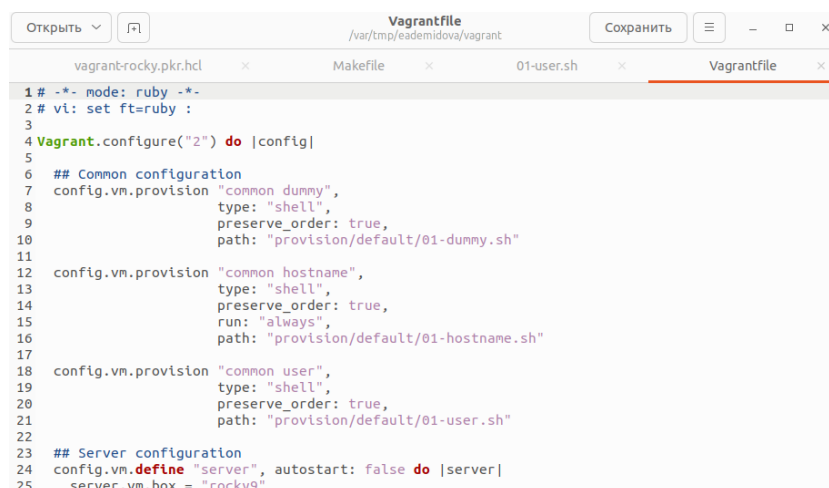


```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant
==> client: Setting hostname...
==> client: Configuring and enabling network interfaces...
==> client: Mounting shared folders...
client: /vagrant => /var/tmp/eademidova/vagrant
==> client: Running provisioner: common dummy (shell)...
client: Running: /tmp/vagrant-shell20231106-26524-b3cwrh.sh
client: Provisioning script /tmp/vagrant-shell
==> client: Running provisioner: common hostname (shell)...
client: Running: /tmp/vagrant-shell20231106-26524-bht5nr.sh
==> client: Running provisioner: common user (shell)...
client: Running: /tmp/vagrant-shell20231106-26524-3hk5ls.sh
client: Provisioning script /tmp/vagrant-shell
client: 1001
client: adduser: user 'eademidova' already exists
==> client: Running provisioner: client dummy (shell)...
client: Running: /tmp/vagrant-shell20231106-26524-wxfxjq.sh
client: Provisioning script /tmp/vagrant-shell
==> client: Running provisioner: client routing (shell)...
client: Running: /tmp/vagrant-shell20231106-26524-s5f4t2.sh
client: Provisioning script /tmp/vagrant-shell
client: Error: failed to modify connection 'system eth1': ipv4.gateway: gateway cannot be set if there
are no addresses configured
client: Error: Connection activation failed: IP configuration could not be reserved (no available addre
ss, timeout, etc.)
client: Hint: use 'journalctl -xe NM_CONNECTION=9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04 + NM_DEVICE=eth1'
to get more details.
client: Error: 'eth0' is not an active connection.
client: Error: no active connection provided.
client: Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveCon
nection/6)
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$
```

Рис. 3.8: Успешный запуск виртуальной машины Client

Затем выключим обе виртуальные машины и внесем изменения в настройки внутреннего окружения виртуальной машины.

Для отработки созданных скриптов во время загрузки виртуальных машин убедимся, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется следующая запись(рис. 3.9):

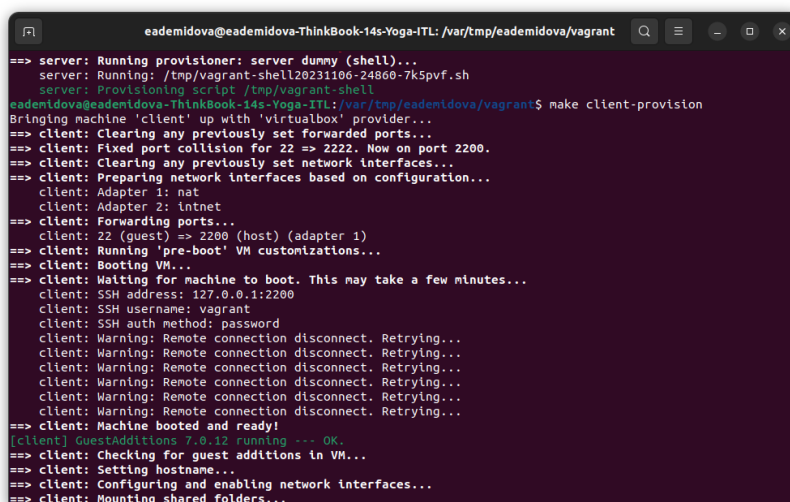


```
Открыть  Vagrantfile  /var/tmp/eademidova/vagrant  Сохранить
vagrant-rocky.pkr.hcl  Makefile  01-user.sh  Vagrantfile
1 # -*- mode: ruby -*-
2 # vi: set ft=ruby :
3
4 Vagrant.configure("2") do |config|
5
6   ## Common configuration
7   config.vm.provision "common dummy",
8     type: "shell",
9     preserve_order: true,
10    path: "provision/default/01-dummy.sh"
11
12   config.vm.provision "common hostname",
13     type: "shell",
14     preserve_order: true,
15     run: "always",
16     path: "provision/default/01-hostname.sh"
17
18   config.vm.provision "common user",
19     type: "shell",
20     preserve_order: true,
21     path: "provision/default/01-user.sh"
22
23   ## Server configuration
24   config.vm.define "server", autostart: false do |server|
25     server.vm.box = "rocky9"
```

Рис. 3.9: Проверка конфигурационного файла Vagrant

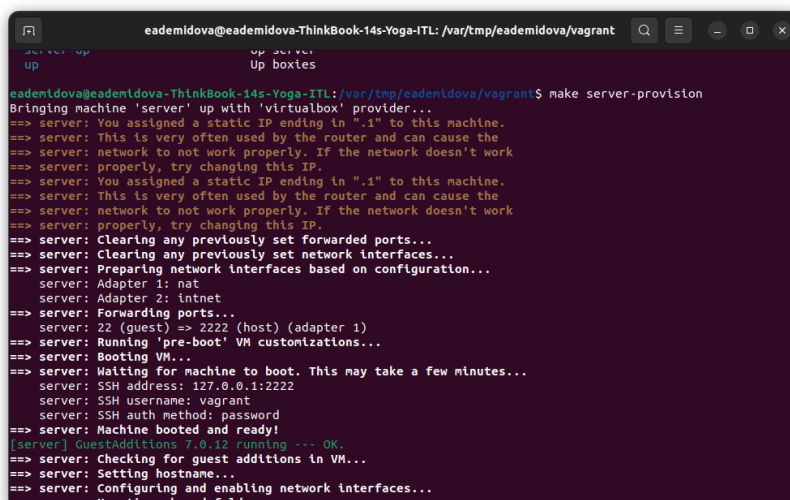
Зафиксируем внесённые изменения для внутренних настроек виртуальных

машин, введя в терминале `make server-provision`, а затем `make client-provision`(рис. 3.10, 3.11):

A terminal window titled 'eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant'. The prompt is 'eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant\$'. The user enters 'make client-provision'. The output shows the server running a dummy shell, provisioning a script, and then the client machine being brought up with the 'virtualbox' provider. The client performs various setup steps: clearing forwarded ports, fixing port collisions (22 to 2222), clearing network interfaces, preparing network interfaces (nat and intnet adapters), forwarding ports, running pre-boot VM customizations, booting the VM, and waiting for it to boot. The SSH address is 127.0.0.1:2200, username is 'vagrant', and the auth method is 'password'. There are several warnings about remote connection disconnections. The client is then checked for guest additions (7.0.12 running), checked for guest additions in the VM, setting the hostname, configuring and enabling network interfaces, and mounting shared folders.

```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make client-provision
==> server: Running provisioner: server dummy (shell)...
server: Running: /tmp/vagrant-shell20231106-24860-7k5pvf.sh
server: Provisioning script /tmp/vagrant-shell
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make client-provision
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...
client: 22 (guest) => 2200 (host) (adapter 1)
==> client: Running 'pre-boot' VM customizations...
==> client: Booting VM...
==> client: Waiting for machine to boot. This may take a few minutes...
client: SSH address: 127.0.0.1:2200
client: SSH username: vagrant
client: SSH auth method: password
client: Warning: Remote connection disconnect. Retrying...
client: Warning: Remote connection disconnect. Retrying...
client: Warning: Remote connection disconnect. Retrying...
client: Warning: Remote connection disconnect. Retrying...
client: Warning: Remote connection disconnect. Retrying...
==> client: Machine booted and ready!
[client] GuestAdditions 7.0.12 running --- OK.
==> client: Checking for guest additions in VM...
==> client: Setting hostname...
==> client: Configuring and enabling network interfaces...
==> client: Mounting shared folders...
```

Рис. 3.10: Команда `make server-provision`

A terminal window titled 'eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant'. The prompt is 'eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant\$'. The user enters 'make server-provision'. The output shows the server machine being brought up with the 'virtualbox' provider. The server performs various setup steps: assigning a static IP ending in '.1', clearing forwarded ports, clearing network interfaces, preparing network interfaces (nat and intnet adapters), forwarding ports, running pre-boot VM customizations, booting the VM, and waiting for it to boot. The SSH address is 127.0.0.1:2222, username is 'vagrant', and the auth method is 'password'. The server is then checked for guest additions (7.0.12 running), checked for guest additions in the VM, setting the hostname, configuring and enabling network interfaces, and mounting shared folders.

```
eademidova@eademidova-ThinkBook-14s-Yoga-ITL: /var/tmp/eademidova/vagrant$ make server-provision
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".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: You assigned a static IP ending in ".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: Clearing any previously set forwarded ports...
==> 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!
[server] GuestAdditions 7.0.12 running --- OK.
==> server: Checking for guest additions in VM...
==> server: Setting hostname...
==> server: Configuring and enabling network interfaces...
==> server: Mounting shared folders...
```

Рис. 3.11: Команда `make client-provision`

Залогинемся на сервере и клиенте под созданным пользователем. Убедимся, что в терминале приглашение отображается в виде `eademidova@server.eademidova.net`

на сервере и в виде eademidova@client.eademidova.net на клиенте(рис. 3.12, 3.13):

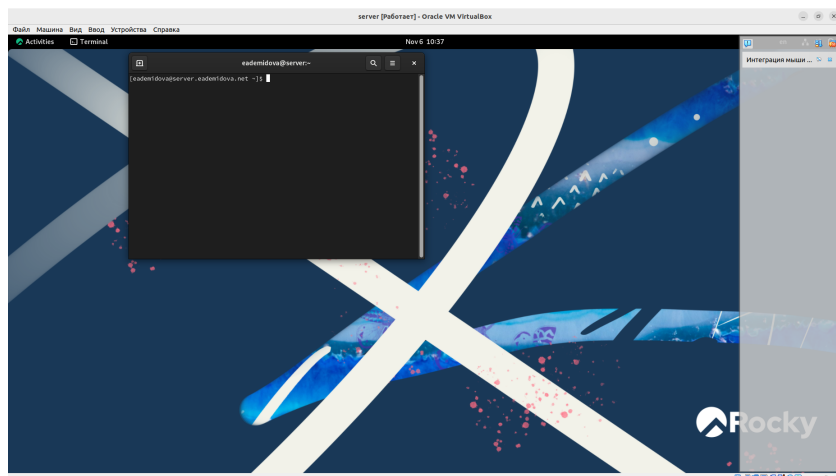


Рис. 3.12: Проверка работы сервера

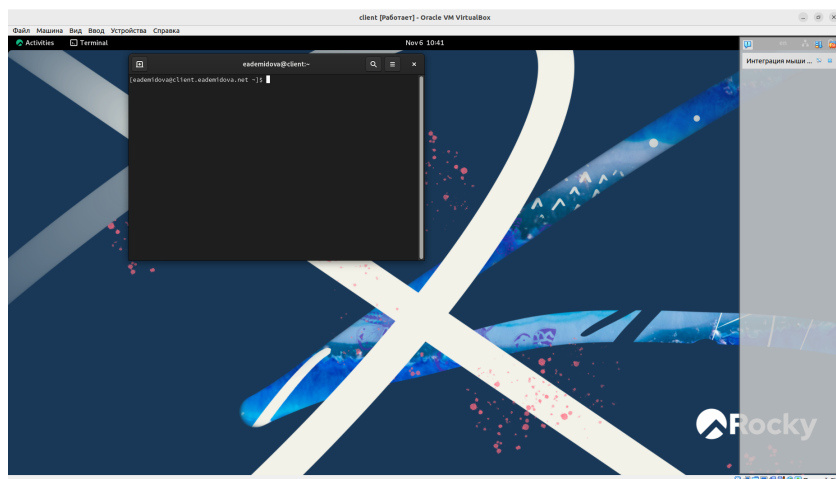


Рис. 3.13: Проверка работы клиента

## 4 Контрольные вопросы

1. Для чего предназначен Vagrant? Vagrant — представляет собой инструмент для создания и управления средами виртуальных машин в одном рабочем процессе. Этот инструмент предназначен для автоматизации процесса установки на виртуальную машину как основного дистрибутива операционной системы, так и настройки необходимого в дальнейшем программного обеспечения.

2. Что такое box-файл? В чём назначение Vagrantfile?

box-файл (или Vagrant Box) — сохранённый образ виртуальной машины с развёрнутой в ней операционной системой; по сути, box-файл используется как основа для клонирования виртуальных машин с теми или иными настройками.

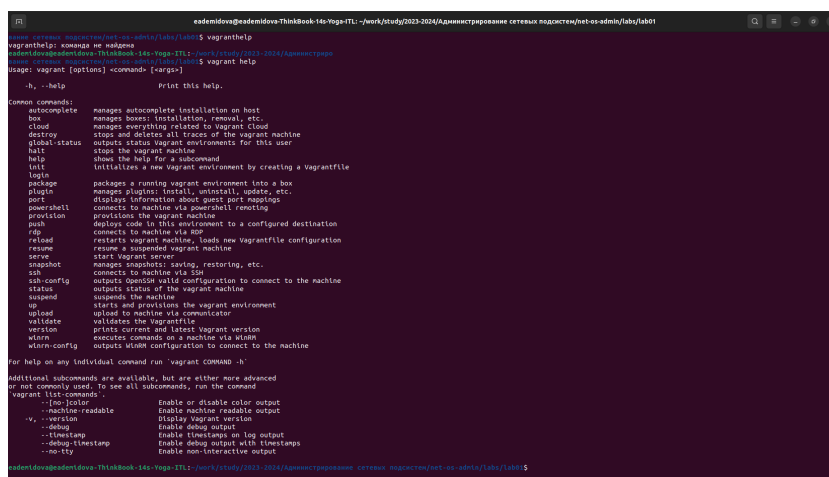
Vagrantfile — конфигурационный файл, написанный на языке Ruby, в котором указаны настройки запуска виртуальной машины.

3. Приведите описание и примеры вызова основных команд Vagrant.

С Vagrant можно работать, используя следующие основные команды (рис. 4.1):

- `vagrant help` — вызов справки по командам Vagrant;
- `vagrant box list` — список подключённых к Vagrant box-файлов;
- `vagrant box add` — подключение box-файла к Vagrant;
- `vagrant destroy` — отключение box-файла от Vagrant и удаление его из виртуального окружения;
- `vagrant init` — создание «шаблонного» конфигурационного файла Vagrantfile для его последующего изменения;
- `vagrant up` — запуск виртуальной машины с использованием инструкций по запуску из конфигурационного файла Vagrantfile;
- `vagrant reload` — перезагрузка виртуальной машины;

`vagrant halt` — остановка и выключение виртуальной машины; – `vagrant provision` — настройка внутреннего окружения имеющейся виртуальной машины (например, добавление новых инструкций (скриптов) в ранее созданную виртуальную машину); – `vagrant ssh` — подключение к виртуальной машине через `ssh`



```
osendmova@osendmova-ThinkBook 14: vagrant - /work/study/2023-2024/Администрирование/сетевых/подсистем/net-os-admin/lab01
vagrant help: команда не найдена
osendmova@osendmova-ThinkBook 14: vagrant - /work/study/2023-2024/Администрирование/сетевых/подсистем/net-os-admin/lab01$ vagrant help
Usage: vagrant [options] [-command] [-args]

-h, --help          print this help.

Common commands:
autocompile         manages autocompile installation on host
box                 manages boxes: installation, removal, etc.
cloud               manages everything related to Vagrant cloud
destroy             stops and destroys all traces of the vagrant machine
global-status       outputs status Vagrant environments for this user
halt                stops the vagrant machine
help                shows the help for a subcommand
init                initializes a new vagrant environment by creating a Vagrantfile
login               packages a running vagrant environment into a box
package             manages plugins: install, uninstall, update, etc.
plugin              displays information about guest and host mappings
port                connects to machine via powershell remoting
powershell          provisions the vagrant machine
provision            provisions code in this environment to a configured destination
push                connects to machine via scp
reload              restarts vagrant machine, loads new Vagrantfile configuration
resume              resume a suspended vagrant machine
serve               start vagrant server
snapshot            manages snapshots: saving, restoring, etc.
ssh                 connects to machine via SSH
ssh-config           outputs option valid configuration to connect to the machine
status              outputs status of the vagrant machine
suspend            suspends the machine
upload              uploads to machine via communicator
validate            validates the Vagrantfile
version             prints current and latest Vagrant version
winrm               executes commands on a machine via winrm
winrm-config        outputs WinRM configuration to connect to the machine

For help on any individual command run 'vagrant COMMAND -h'

Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
'vagrant list-commands'.

  -f, --force          Enable or disable color output
  -m, --machine-readable Enable machine readable output
  -v, --version        Display Vagrant version
  -d, --debug          Enable debug output
  -t, --timestamp      Enable timestamps on log output
  -D, --debug-timestamp Enable debug output with timestamps
  -n, --no-tty         Enable non-interactive output
```

Рис. 4.1: Команды Vagrant

4. Дайте построчные пояснения содержания файлов `vagrant-rocky.pkr.hcl`, `ks.cfg`, `Vagrantfile`, `Makefile`.

`vagrant-rocky.pkr.hcl` — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину. в частности, в разделе переменных этот файл содержит указание на версию дистрибутива, его хэш-функцию, имя и пароль пользователя по умолчанию; в разделе `builders` указаны специальные синтаксические конструкции для автоматизации работы VirtualBox; в разделе `provisioners` прописаны действия (по сути `shell`-скрипт) по установке дополнительных пакетов дистрибутива

`ks.cfg` — определяет настройки для установки дистрибутива, которые пользователь обычно вводит вручную, в частности настройки языка интерфейса, языковые настройки клавиатуры, тайм-зону, сетевые настройки и т.п.; файл должен быть расположен в подкаталоге `http/`

Vagrantfile — файл с конфигурацией запуска виртуальных машин — сервера и клиента.

Makefile — набор инструкций для программы make по работе с Vagrant



## 5 Выводы

В результате выполнения данной работы были приобретены практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.