

Отчёт по лабораторной работе №1

Дисциплина: Администрирование сетевых подсистем

Мишина Анастасия Алексеевна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
2.1 Развёртывание лабораторного стенда на ОС Windows . . .	6
2.2 Внесение изменений в настройки внутреннего окружения виртуальной машины	11
3 Выводы	19

Список иллюстраций

2.1 Установка образа Rocky в Virtualbox и формирование box-файла	7
2.2 box-файл с дистрибутивом Rocky для Virtualbox	7
2.3 Регистрация образа виртуальной машины в vagrant	7
2.4 Запуск виртуальной машины Server	8
2.5 Запуск виртуальной машины Client	8
2.6 Логинимся в VM Server	9
2.7 Логинимся в VM Client	10
2.8 Подключение к серверу и клиенту, переход к пользователю, отключение ВМ	11
2.9 Фиксируем изменения в VM Server	12
2.10 Фиксируем изменения в VM Client	12
2.11 Логинимся в VM Server	13
2.12 Логинимся в VM Client	14
2.13 Приглашения в терминале	15
2.14 Выключение виртуальных машин	15
2.15 Копирование файлов	16

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 необходимые каталоги для работы и добавим в них файлы, описанные в мануале на ТУИСе. Так как я выполняю работу на Windows, расположение у меня следующее C:\work\aaamishina\packer и C:\work\aaamishina\vagrant. Помещаем в packer образ Rocky Linux, vagrant-rocky.pkr.hcl, каталог http с файлов настроек для установки дистрибутива - ks.cfg. В подкаталог vagrant помещаем Vagrantfile, а также каталог provision с подкаталогами default, server и client (их заполняем скриптами из мануала).

2.1 Развёртывание лабораторного стенда на ОС Windows

Для начала переходим в каталог с файлом packer.exe и выполняем следующие команды: packer.exe init vagrant-rocky.pkr.hcl и packer.exe build vagrant-rocky.pkr.hcl (рис. 2.1) для начала автоматической установки образа операционной системы Rocky Linux в VirtualBox и последующего формирования box-файла с дистрибутивом Rocky Linux для VirtualBox (рис. 2.2).

```

Administrator: C:\Windows\system32\cmd.exe - packer.exe build vagrant-rocky.pkr.hcl
C:\work\aaamishina\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/virtualbox v1.1.1 in "C:/Users/Настя/AppData/Roaming/packer.d/plugins/github.com/hashicorp/virtualbox/packer-plugin-virtualbox_v1.1.1_x5.0_windows_amd64.exe"
Installed plugin github.com/hashicorp/vagrant v1.1.4 in "C:/Users/Настя/AppData/Roaming/packer.d/plugins/github.com/hashicorp/vagrant/packer-plugin-vagrant_v1.1.4_x5.0_windows_amd64.exe"

C:\work\aaamishina\packer>packer.exe build vagrant-rocky.pkr.hcl
virtualbox-iso.virtualbox: output will be in this color.

==> virtualbox-iso.virtualbox: Retrieving Guest additions
==> virtualbox-iso.virtualbox: Trying C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso
==> virtualbox-iso.virtualbox: Trying file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.virtualbox: file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso => C:/Program Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.virtualbox: Retrieving ISO
==> virtualbox-iso.virtualbox: Trying Rocky-9.4-x86_64-minimal.iso
==> virtualbox-iso.virtualbox: Trying Rocky-9.4-x86_64-minimal.iso?checksum=sha256%3Aee3ac97fdfab58652421941599902012179c3753aece76824673105169c4a2

```

Рис. 2.1: Установка образа Rocky в Virtualbox и формирование box-файла

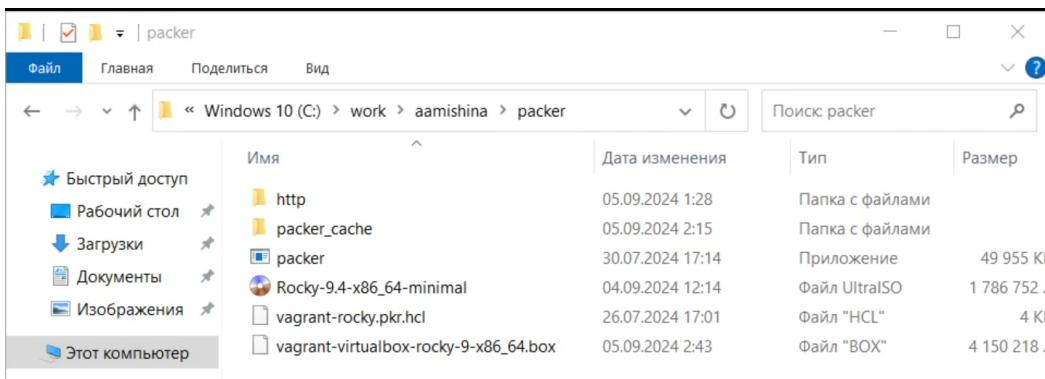


Рис. 2.2: box-файл с дистрибутивом Rocky для Virtualbox

Зарегистрируем образ виртуальной машины в vagrant: vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86_64.box (рис. 2.3).

```

C:\work\aaamishina\packer>vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86_64.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky9' (v0) for provider:
    box: Unpacking necessary files from: file:///C:/work/aaamishina/packer/vagrant-virtualbox-rocky-9-x86_64.box
    box:
The box you're attempting to add already exists. Remove it before
adding it again or add it with the '--force' flag.

Name: rocky9
Provider: virtualbox
Version: 0

C:\work\aaamishina\packer>

```

Рис. 2.3: Регистрация образа виртуальной машины в vagrant

Запустим виртуальные машины Server: vagrant up server (рис. 2.4) и Client: vagrant up client (рис. 2.5).

```
C:\work\aaamishina>vagrant up server
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'...
==> server: Cloning VM...
==> server: Matching MAC address for NAT networking...
==> 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.
```

Рис. 2.4: Запуск виртуальной машины Server

```
C:\work\aaamishina>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: 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...
```

Рис. 2.5: Запуск виртуальной машины Client

Запуск обеих виртуальных машин проходит успешно, залогинимся в графическом окружении.(рис. 2.4) и Client: vagrant up client (рис. 2.5).

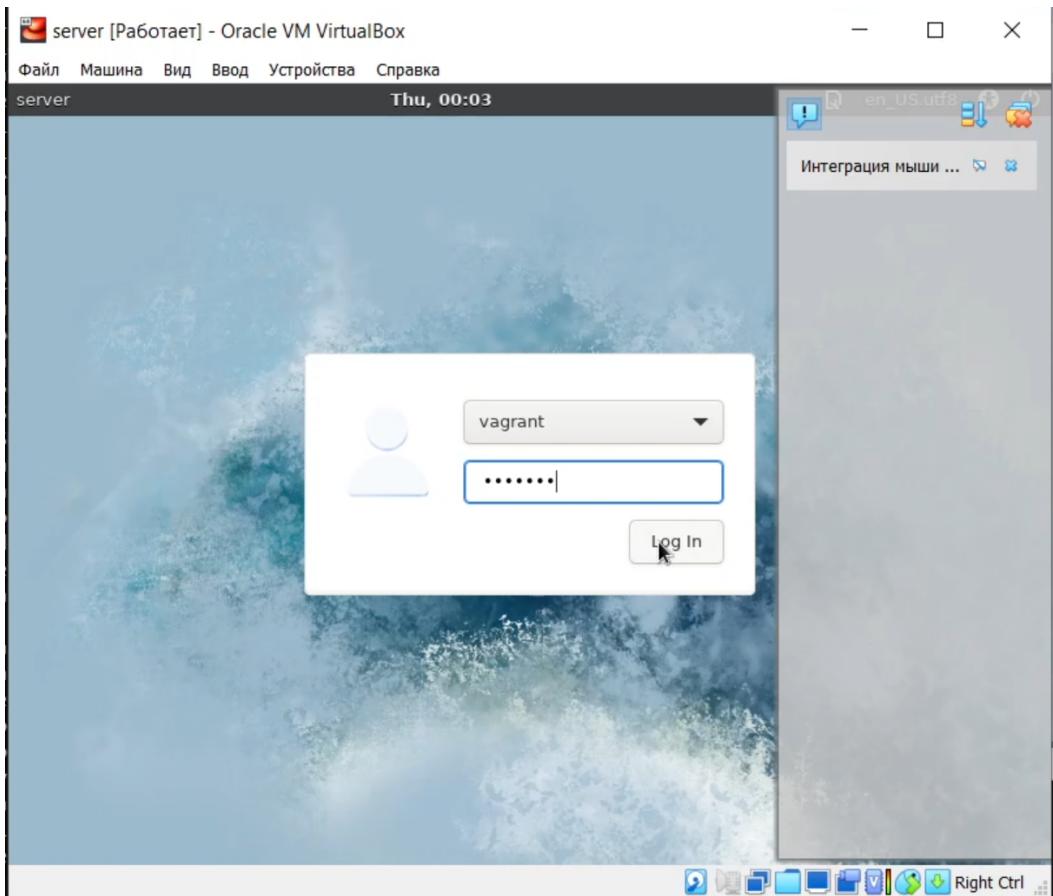


Рис. 2.6: Логинимся в ВМ Server

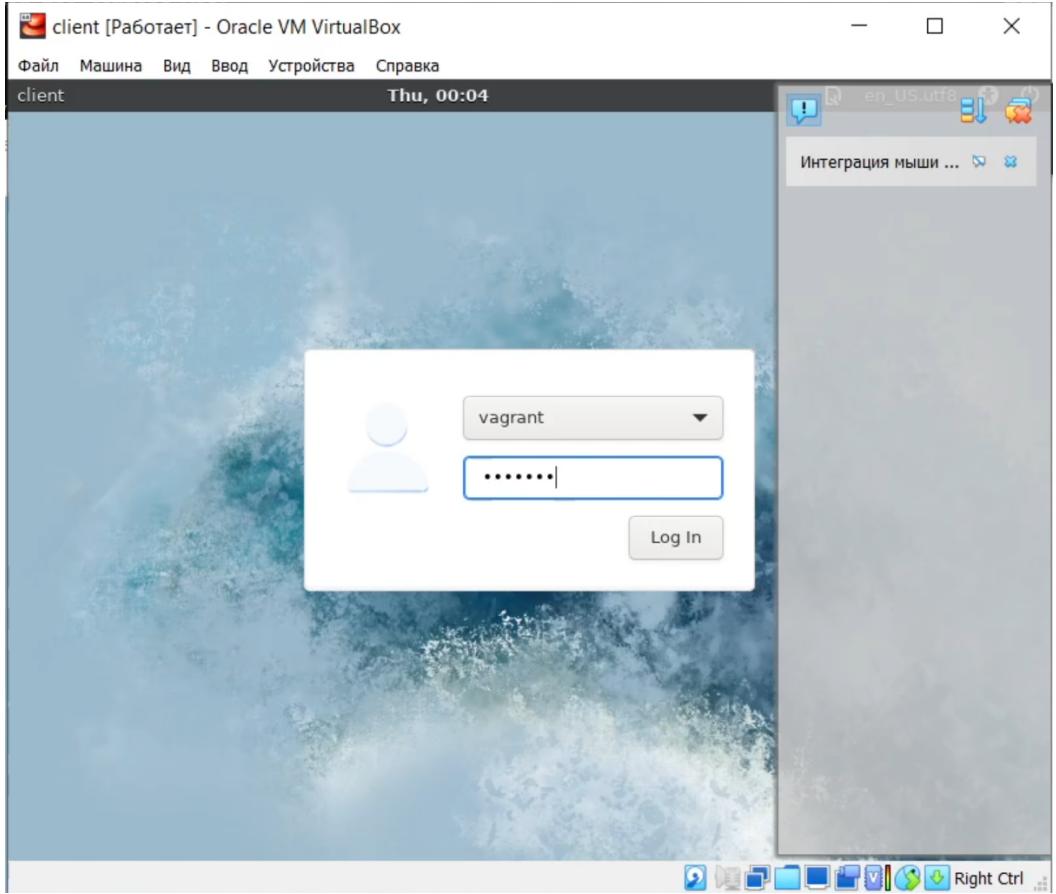


Рис. 2.7: Логинимся в BM Client

Подключаемся к серверу из консоли: `vagrant ssh server`, вводим пароль `vagrant`. Переходим к пользователю `aamishina`: `su - aamishina`. Отлогиниваемся и выполняем тоже самое для клиента. Выключаем обе виртуальные машины: `vagrant halt server` и `vagrant halt client` (рис. 2.8).

```
C:\work\aaamishina>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: Thu Sep  5 14:16:24 2024 from 10.0.0.2.2
[vagrant@server ~]$ su - aaamishina
su: user aaamishina does not exist or the user entry does not contain all the required fields
[vagrant@server ~]$ logout
Connection to 127.0.0.1 closed.

C:\work\aaamishina>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: Thu Sep  5 01:03:21 2024
[vagrant@client ~]$ su - aaamishina
su: user aaamishina does not exist or the user entry does not contain all the required fields
[vagrant@client ~]$ logout
Connection to 127.0.0.1 closed.

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

C:\work\aaamishina>vagrant halt client
==> client: Attempting graceful shutdown of VM...

C:\work\aaamishina>vagrant>
```

Рис. 2.8: Подключение к серверу и клиенту, переход к пользователю, отключение ВМ

2.2 Внесение изменений в настройки внутреннего окружения виртуальной машины

Убеждаемся, что в файле Vagrantfile с конфигурацией файла есть отрывок из мануала в ТУИСе.

Фиксируем внесённые изменения для внутренних настроек виртуальных машин, введя в терминале: vagrant up server –provision (рис. 2.9) и vagrant up client –provision (рис. 2.10).

```
C:\work\aaamishina>vagrant up 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...

```

Рис. 2.9: Фиксируем изменения в VM Server

```
C:\work\aaamishina>vagrant up 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...
```

Рис. 2.10: Фиксируем изменения в VM Client

Залогинимся на сервере (рис. 2.11) и клиенте (рис. 2.12) под созданным пользователем.

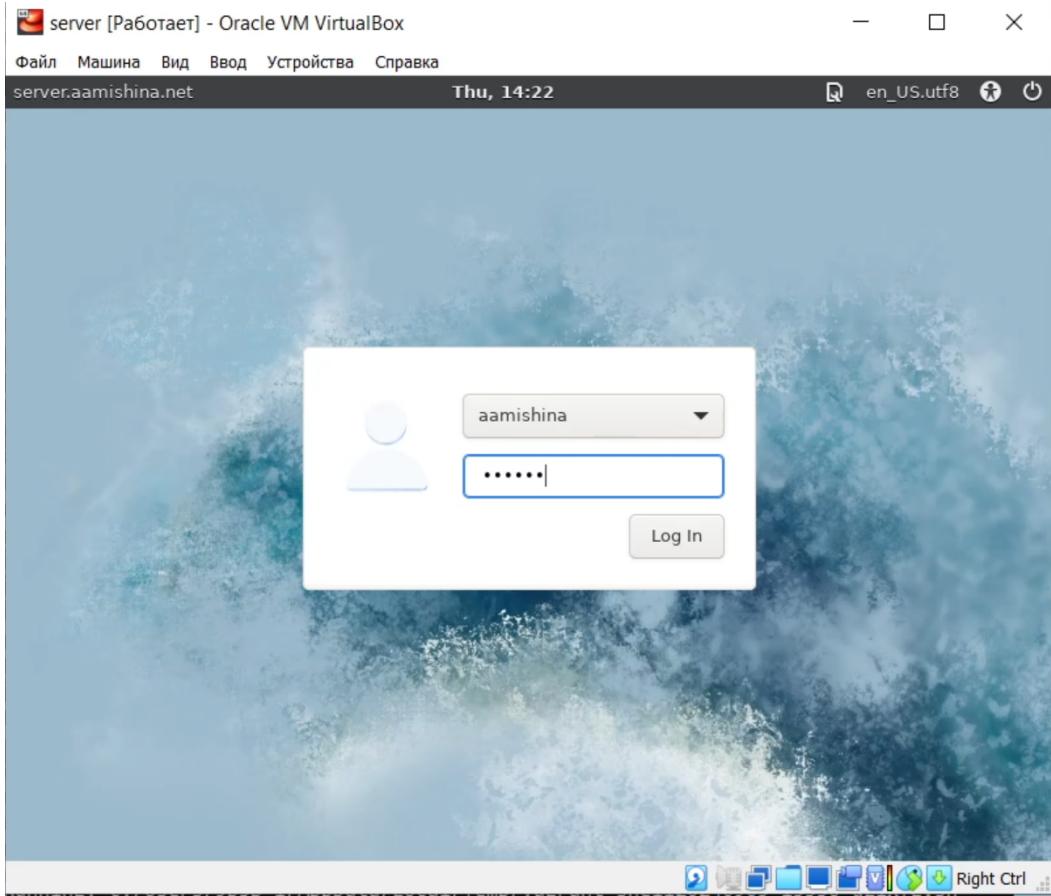


Рис. 2.11: Логинимся в ВМ Server

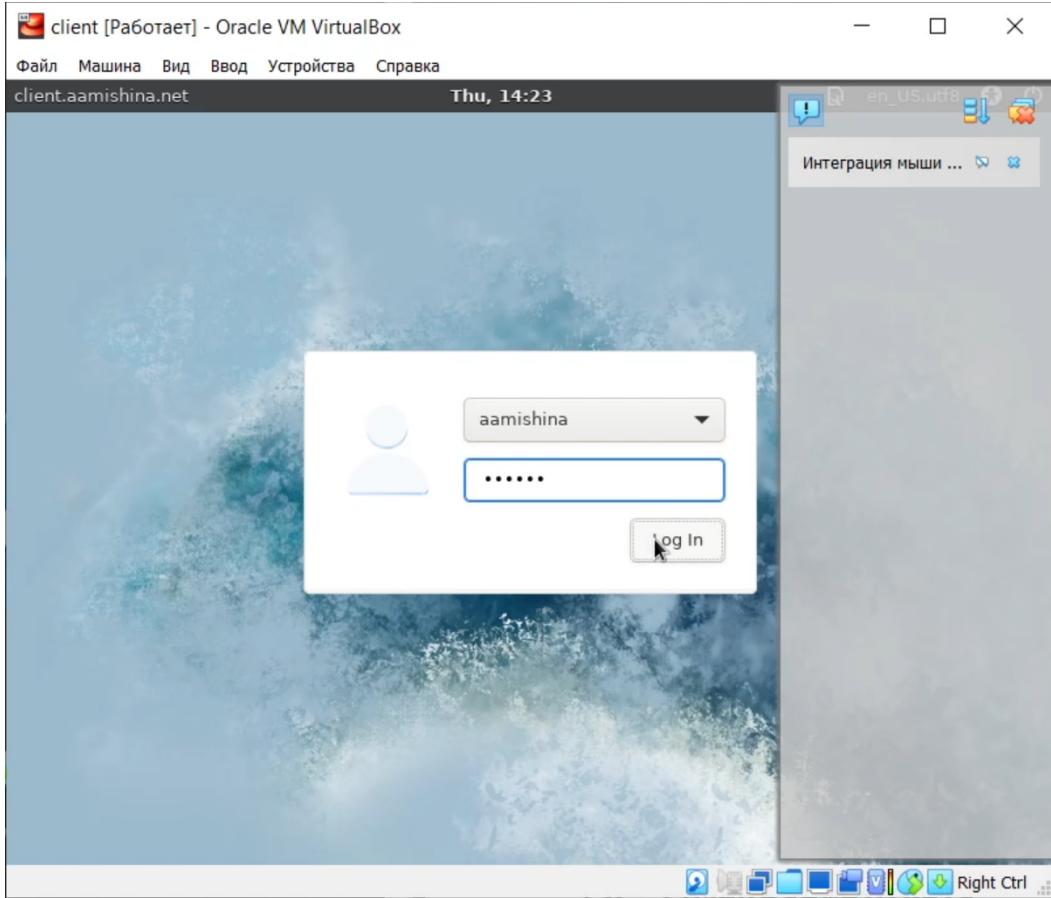


Рис. 2.12: Логинимся в ВМ Client

Убедимся, что в терминале приглашение отображается в виде `aamishina@server.aamishina.net` на сервере и в виде `aamishina@client.aamishina.net` на клиенте (рис. 2.13).

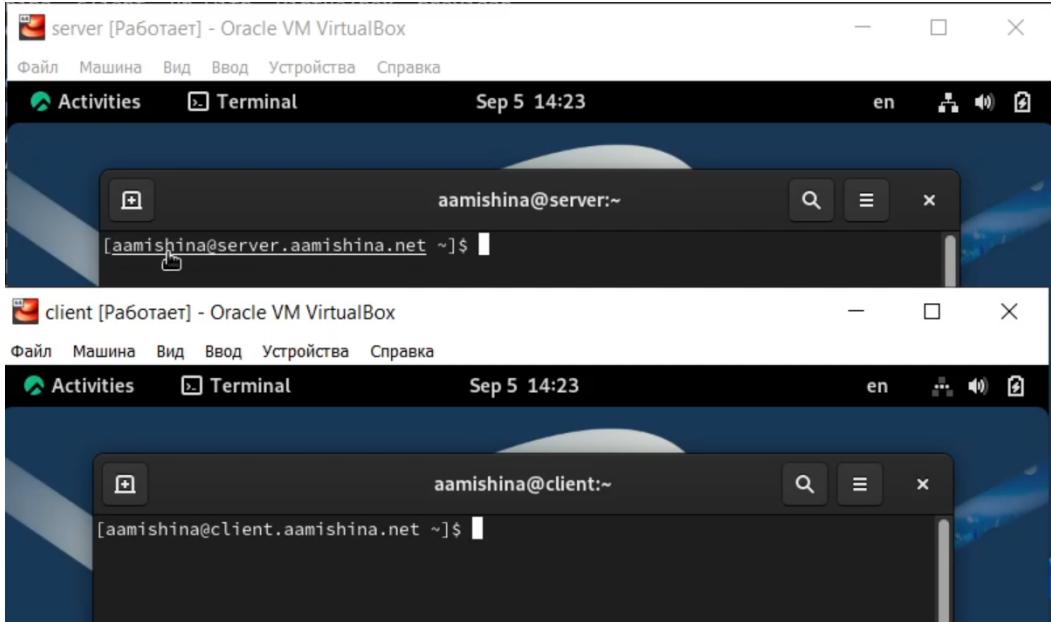


Рис. 2.13: Приглашения в терминале

Выключаем обе виртуальные машины (рис. 2.14) и копируем необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель или в другой каталог вашей ОС (рис. 2.15).

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

Рис. 2.14: Выключение виртуальных машин

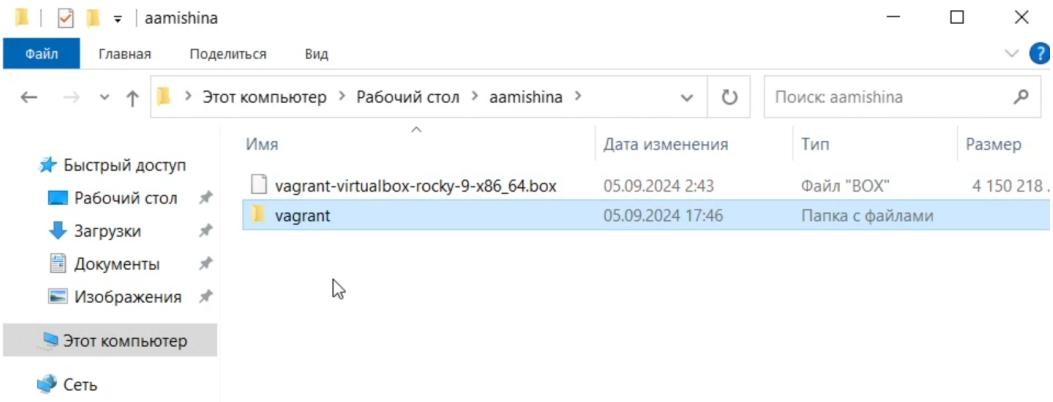


Рис. 2.15: Копирование файлов

Контрольные вопросы:

1. Для чего предназначен Vagrant? – Это инструмент для создания и управления средами виртуальных машин в одном рабочем процессе. Он позволяет автоматизировать процесс 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, 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. `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.

4. Дайте построчные пояснения содержания файлов vagrant-rocky.pkr.hcl, ks.cfg, Vagrantfile, Makefile. Vagrantfile - Первые две строки указывают на режим работы с Vagrantfile и использование языка Ruby. Затем идёт цикл do, заменяющий конструкцию Vagrant.configure далее по тексту на config. Стока config.vm.box = “BOX_NAME” задаёт название образа (бокс-файла) виртуальной машины (обычно выбирается из официального репозитория). Стока config.vm.hostname = “HOST_NAME” задаёт имя виртуальной машины. Конструкция config.vm.network задаёт тип сетевого соединения и может иметь следующие назначения: – config.vm.network “private_network”, ip: “xxx.xxx.xxx.xxx” – адрес из внутренней сети; – config.vm.network “public_network”, ip: “xxx.xxx.xxx.xxx” – публичный адрес, по которому виртуальная машина будет доступна; – config.vm.network “private_network”, type: “dhcp” – адрес, назначаемый по протоколу DHCP. Стока config.vm.define “VM_NAME” задаёт название виртуальной машины, по которому можно обращаться к ней из Vagrant и VirtualBox. В конце идёт конструкция, определяющая параметры провайдера, а именно

запуск виртуальной машины без графического интерфейса и с выделением 1 ГБ памяти.

3 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.