

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Дисциплина «Администрирование сетевых подсистем»

Тема «Подготовка лабораторного стенда»

Студент: Щербак Маргарита Романовна

Ст. билет: 1032216537

Группа: НПИбд-02-21

МОСКВА

2023 г.

Цель работы

Приобрести практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

Задание

1. Сформировать box-файл с дистрибутивом Rocky Linux для VirtualBox.
2. Запустить виртуальные машины сервера и клиента, а также убедиться в их работоспособности.
3. Внести изменения в настройки загрузки образов виртуальных машин server и client, добавив пользователя с правами администратора и изменив названия хостов.
4. Скопировать необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель.

Выполнение работы

1. Формирование box-файла с дистрибутивом Rocky Linux для VirtualBox

В своей ОС установила последние версии Vagrant и VirtualBox. Для моей ОС Windows также понадобилось дополнительно установить Packer и FAR для удобства работы в терминале. Перед началом работы с Vagrant создала каталог \work\mrshcherbak для проекта, далее две папки: C:\work\ mrshcherbak\packer и C:\work\ mrshcherbak\vagrant (рис.1.1). В созданном рабочем каталоге разместила образ варианта операционной системы Rocky Linux: Rocky-9.2- x86_64-minimal.iso — минимальный дистрибутив Rocky Linux. Разместила подготовленные заранее для работы с Vagrant файлы (рис.1.2 – рис.1.3):

- vagrant-rocky.pkr.hcl — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину;
- 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 расположен в подкаталоге http;
- Vagrantfile — файл с конфигурацией запуска виртуальных машин — сервера и клиента.

Этот компьютер > Shcherbak (C:) > Work > mrshcherbak				
Имя	Дата изменения	Тип	Размер	
Far	05.11.2023 19:21	Папка с файлами		
packer	06.11.2023 16:55	Папка с файлами		
vagrant	06.11.2023 12:35	Папка с файлами		
packer.exe	05.11.2023 19:27	Приложение	81 566 КБ	

Рис.1.1. Создание каталогов

Этот компьютер > Shcherbak (C:) > Work > mrshcherbak > packer				
Имя	Дата изменения	Тип	Размер	
builds	06.11.2023 16:55	Папка с файлами		
http	05.11.2023 19:05	Папка с файлами		
packer_cache	06.11.2023 11:40	Папка с файлами		
provision	05.11.2023 19:05	Папка с файлами		
packer.exe	05.11.2023 19:27	Приложение	81 566 КБ	
Rocky-9.2-x86_64-minimal.iso	05.11.2023 18:47	Файл образа диска	1 573 184 КБ	
vagrant-rocky.pkr.hcl	21.10.2023 14:55	Файл "HCL"	4 КБ	

Рис.1.2. Файлы, необходимые для работы

Этот компьютер > Shcherbak (C:) > Work > mrshcherbak > vagrant				
Имя	Дата изменения	Тип	Размер	
.vagrant	05.11.2023 22:32	Папка с файлами		
provision	05.11.2023 19:05	Папка с файлами		
packer.exe	05.11.2023 19:27	Приложение	81 566 КБ	
Vagrantfile	21.10.2023 14:55	Файл	4 КБ	

Рис.1.3. Файлы, необходимые для работы

Создала 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 (рис.1.4 – рис.1.5).

Этот компьютер > Shcherbak (C:) > Work > mrshcherbak > vagrant > provision				
Имя	Дата изменения	Тип	Размер	
client	05.11.2023 19:05	Папка с файлами		
default	06.11.2023 12:33	Папка с файлами		
server	05.11.2023 19:05	Папка с файлами		

Рис.1.4. Создание каталогов

	Имя	Дата изменения	Тип	Размер
Имя	client	05.11.2023 19:05	Папка с файлами	
	default	06.11.2023 16:48	Папка с файлами	
	server	05.11.2023 19:05	Папка с файлами	

Рис.1.5. Создание каталогов

В каталогах default, server и client каждой папки разместила заранее подготовленный скрипт-заглушку 01-dummy.sh следующего содержания:

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

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

В каталоге default каждой папки разместила заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины (рис.1.6).

```

1  #!/bin/bash
2
3  echo "Provisioning script $0" .
4
5  username=mrshcherbak
6  userpassword=Rastamana035
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

```

Рис.1.6. Содержимое файла 01-user.sh

В каталоге default каждой папки разместила заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины (рис.1.7).

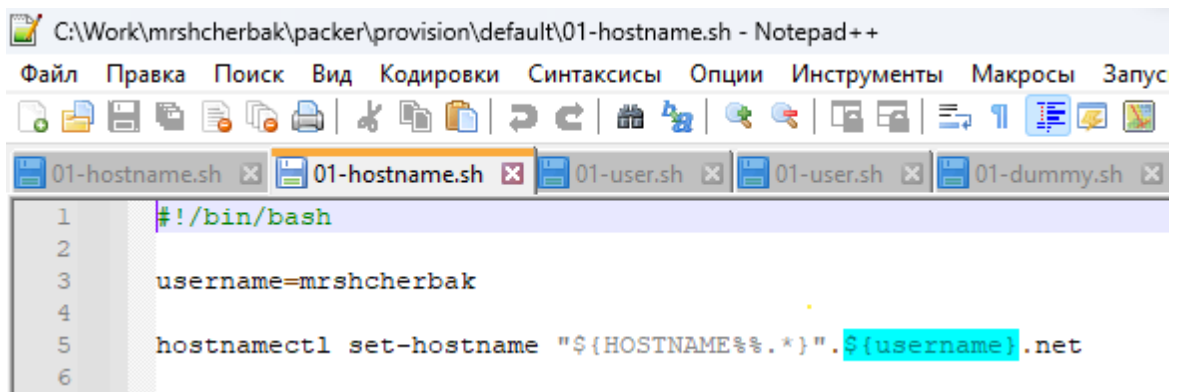


Рис.1.7. Содержимое файла 01-hostname.sh

2. Развёртывание лабораторного стенда на ОС Windows

Используя FAR, перешла в созданный рабочий каталог с проектом. В этом же каталоге размещён файл packer.exe. В командной строке ввела:

```
packer.exe init vagrant-rocky.pkr.hcl
```

```
packer.exe build vagrant-rocky.pkr.hcl
```

Так начнется автоматическая установка образа операционной системы Rocky Linux в VirtualBox и последующее формирование box-файла с дистрибутивом Rocky Linux для VirtualBox (рис.2.1 – рис.2.2). По окончании процесса в рабочем каталоге сформируется box-файл с названием vagrant-virtualbox-rocky-9-x86_64.box (рис.2.3).

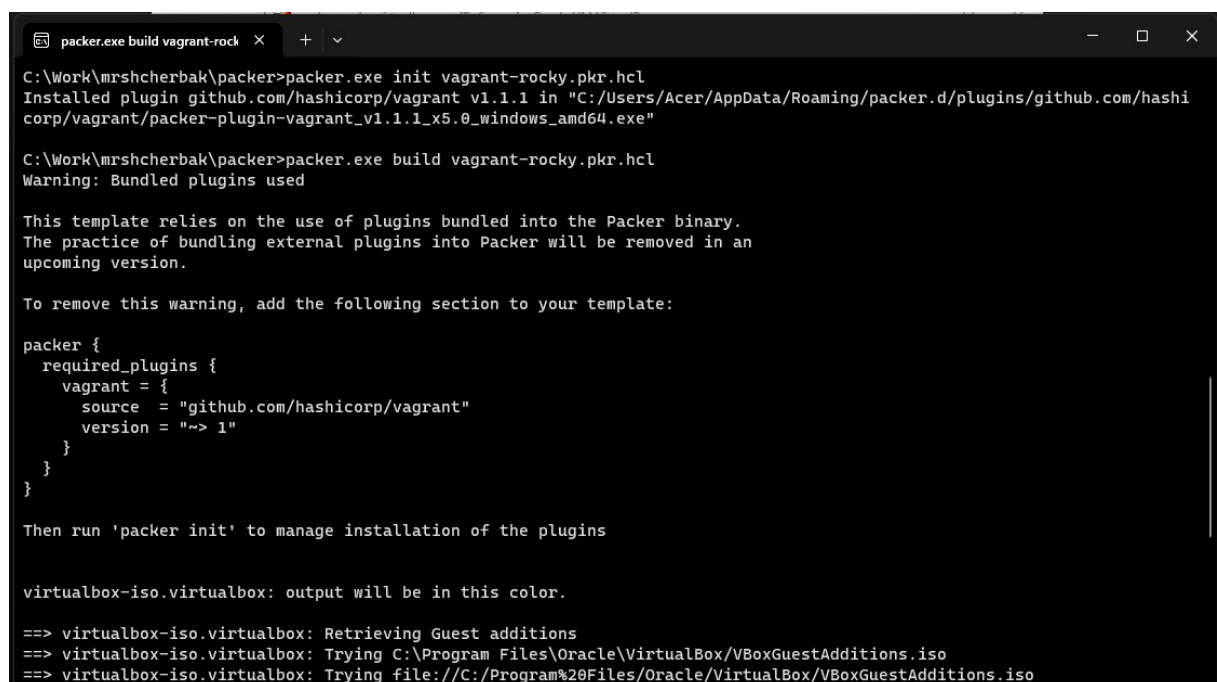


Рис.2.1. Выполнение команд

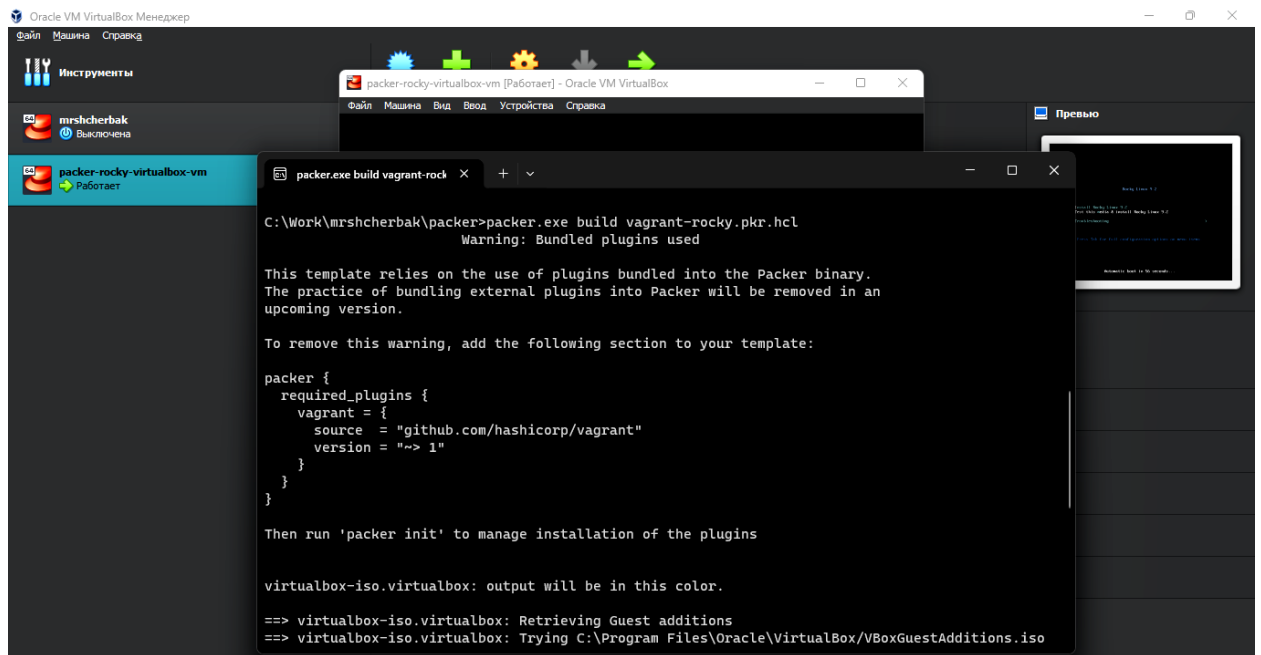


Рис.2.2. Автоматическая установка образа ОС Rocky Linux и последующее формирование box-файла

Этот компьютер > Shcherbak (C:) > Work > mrshcherbak > packer					
Имя	Дата изменения	Тип	Размер		
http	05.11.2023 19:05	Папка с файлами			
packer_cache	06.11.2023 11:40	Папка с файлами			
provision	05.11.2023 19:05	Папка с файлами			
packer.exe	05.11.2023 19:27	Приложение	81 566 КБ		
Rocky-9.2-x86_64-minimal.iso	05.11.2023 18:47	Файл образа диска	1 573 184 КБ		
vagrant-rocky.pkr.hcl	21.10.2023 14:55	Файл "HCL"	4 КБ		
vagrant-virtualbox-rocky-9-x86_64.box	06.11.2023 17:48	Файл "BOX"	3 729 129 КБ		

Рис.2.3. Появился сформированный box-файл

Для регистрации образа виртуальной машины в vagrant в командной строке ввела: `vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86_64.box` (рис.2.4).

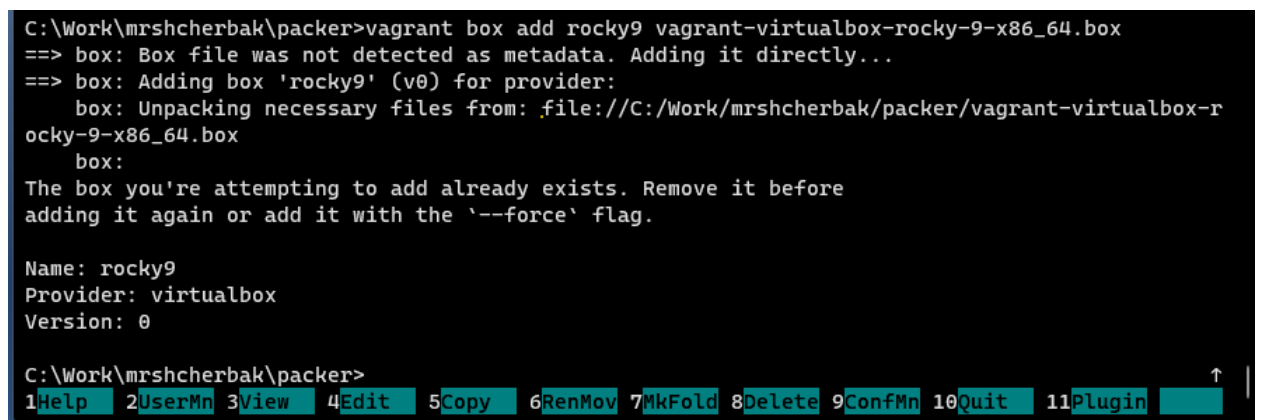
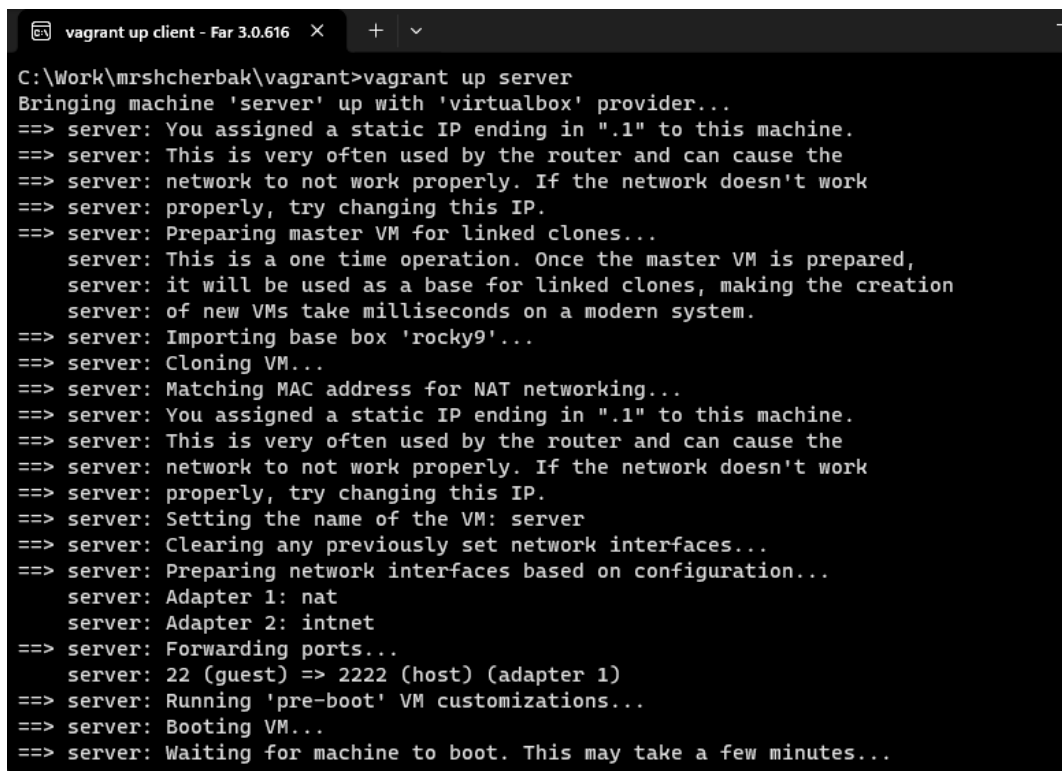


Рис.2.4. Выполнение команды

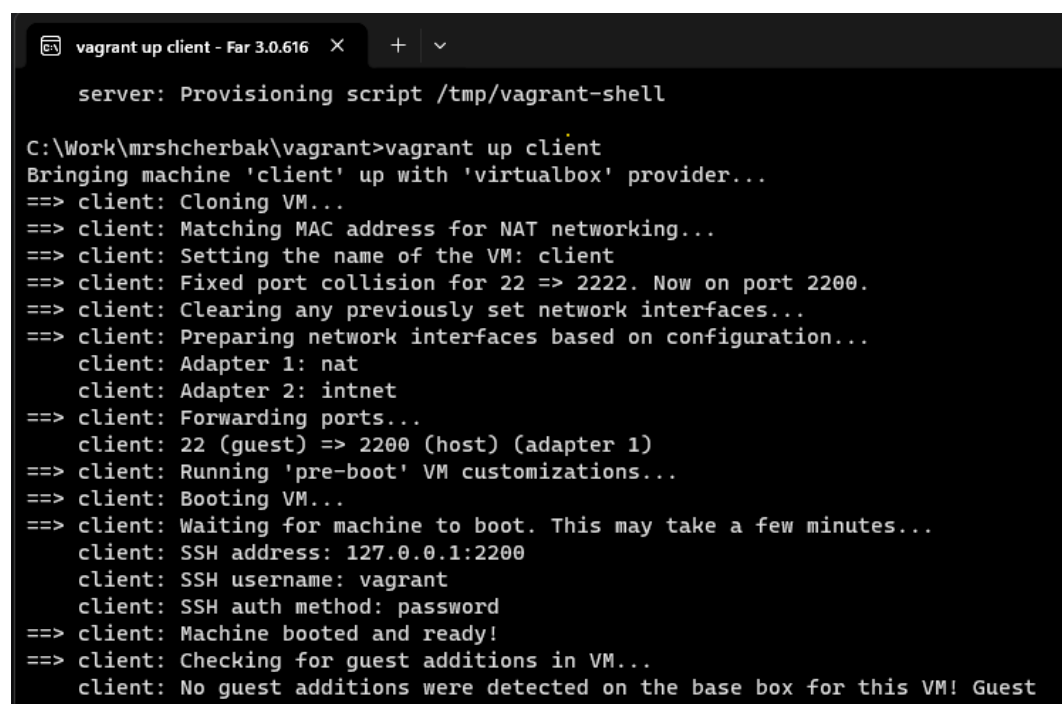
Для запуска виртуальной машины Server ввела в консоли: `vagrant up server` (рис.2.5).

A screenshot of a terminal window titled 'vagrant up client - Far 3.0.616'. The terminal shows the command 'vagrant up server' being executed. The output indicates that the machine 'server' is being brought up with the 'virtualbox' provider. It shows various steps like cloning the VM, matching MAC addresses, setting the name to 'server', and preparing network interfaces. The process concludes with 'Waiting for machine to boot. This may take a few minutes...'.

```
C:\Work\mrshcherbak\vagrant>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.
==> server: Setting the name of the VM: server
==> 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...
```

Рис.2.5. Выполнение команды

Для запуска виртуальной машины Client ввела в консоли: `vagrant up client` (рис.2.6).

A screenshot of a terminal window titled 'vagrant up client - Far 3.0.616'. The terminal shows the command 'vagrant up client' being executed. The output indicates that the machine 'client' is being brought up with the 'virtualbox' provider. It shows various steps like cloning the VM, matching MAC addresses, setting the name to 'client', fixing port collision for port 22, and preparing network interfaces. The process concludes with 'Waiting for machine to boot. This may take a few minutes...' and then provides SSH details: 'SSH address: 127.0.0.1:2200', 'SSH username: vagrant', and 'SSH auth method: password'.

```
server: Provisioning script /tmp/vagrant-shell

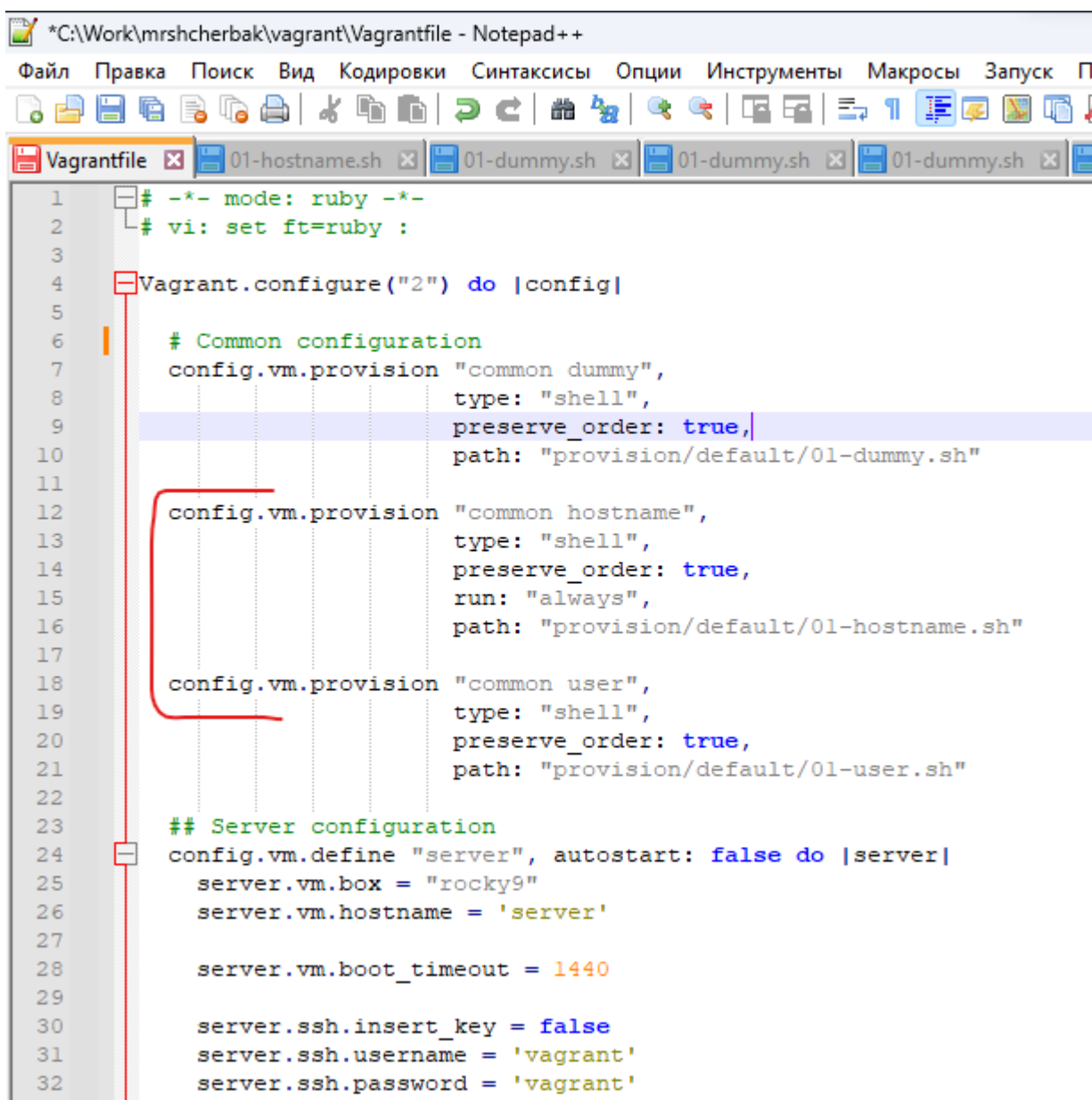
C:\Work\mrshcherbak\vagrant>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Cloning VM...
==> client: Matching MAC address for NAT networking...
==> client: Setting the name of the VM: client
==> client: Fixed port collision for 22 => 2222. Now on port 2200.
==> 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
==> client: Machine booted and ready!
==> client: Checking for guest additions in VM...
client: No guest additions were detected on the base box for this VM! Guest
```

Рис.2.6. Выполнение команды

Убедилась, что запуск обеих виртуальных машин прошёл успешно. Корректно выключила виртуальные машины.

3. Внесение изменений в настройки внутреннего окружения виртуальной машины

Для отработки созданных скриптов во время загрузки виртуальных машин убедилась, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется запись, выделенная красным цветом (рис.3.1).



```
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"
26          server.vm.hostname = 'server'
27
28          server.vm.boot_timeout = 1440
29
30          server.ssh.insert_key = false
31          server.ssh.username = 'vagrant'
32          server.ssh.password = 'vagrant'
```

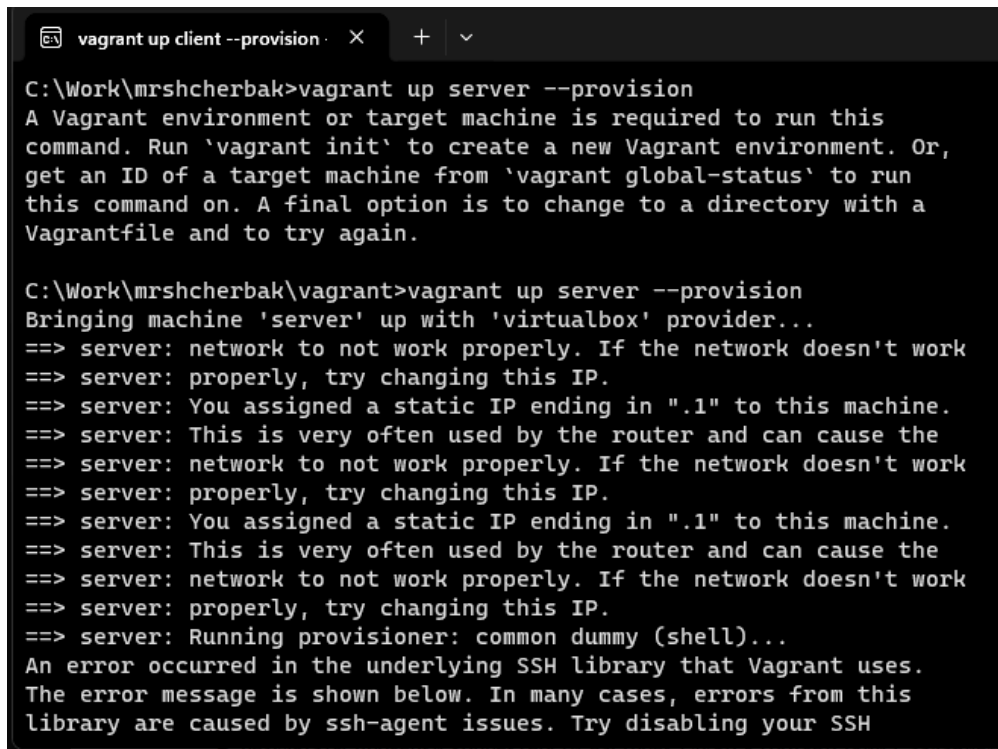
Рис.3.1. Содержимое файла Vagrantfile

Зафиксировала внесённые изменения для внутренних настроек виртуальных машин, введя в терминале:

vagrant up server --provision

vagrant up client --provision

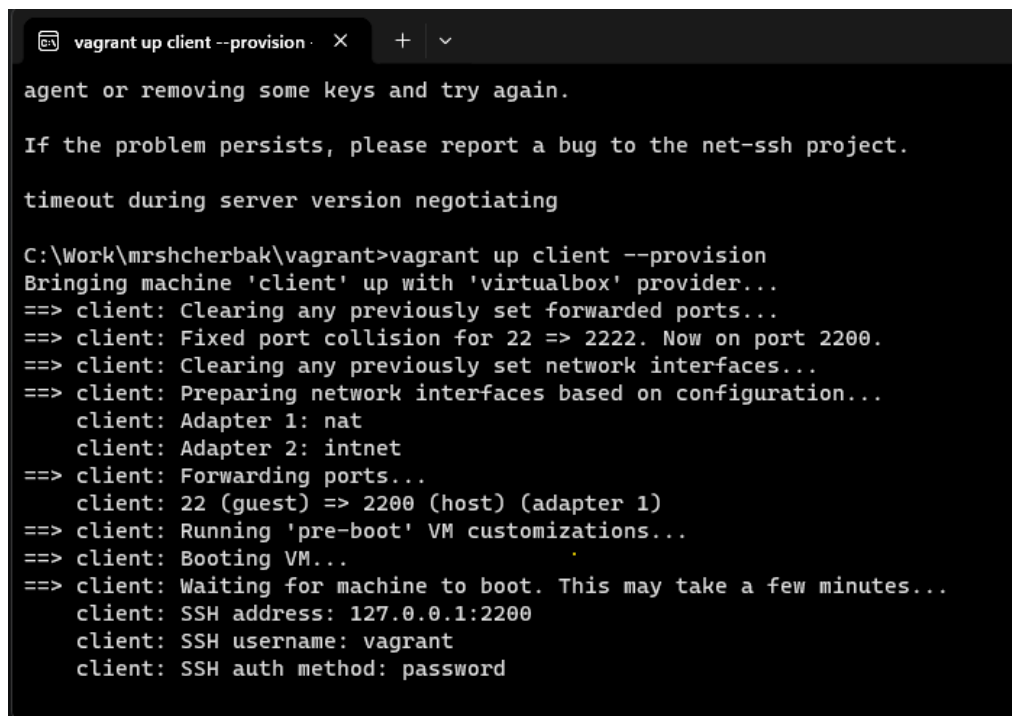
Выполнение команд представлено на рис.3.2 и рис.3.3.



```
C:\Work\mrshcherbak>vagrant up server --provision
A Vagrant environment or target machine is required to run this
command. Run 'vagrant init' to create a new Vagrant environment. Or,
get an ID of a target machine from 'vagrant global-status' to run
this command on. A final option is to change to a directory with a
Vagrantfile and to try again.

C:\Work\mrshcherbak\ vagrant>vagrant up server --provision
Bringing machine 'server' up with 'virtualbox' provider...
==> 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: 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: Running provisioner: common dummy (shell)...
An error occurred in the underlying SSH library that Vagrant uses.
The error message is shown below. In many cases, errors from this
library are caused by ssh-agent issues. Try disabling your SSH
```

Рис.3.2. Выполнение команды vagrant up server --provision



```
agent or removing some keys and try again.

If the problem persists, please report a bug to the net-ssh project.

timeout during server version negotiating

C:\Work\mrshcherbak\ vagrant>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...
==> 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.3.3. Выполнение команды vagrant up client --provision

Залогинилась на сервере и клиенте под созданным пользователем. Убедилась, что в терминале приглашение отображается в виде `mrshcherbak@server.mrshcherbak.net` на сервере и в виде `mrshcherbak@client.mrshcherbak.net` на клиенте (рис.3.4 – рис.3.5). Выключила виртуальные машины.

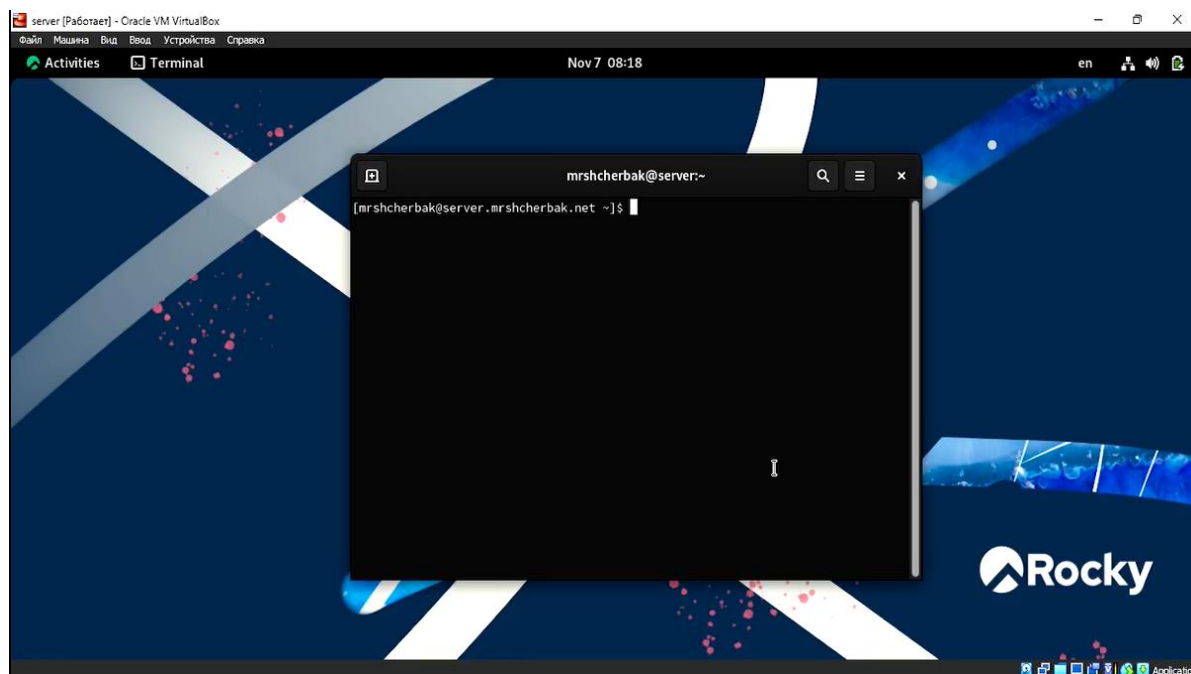


Рис.3.4. На сервере приглашение отображается в виде
`mrshcherbak@server.mrshcherbak.net`

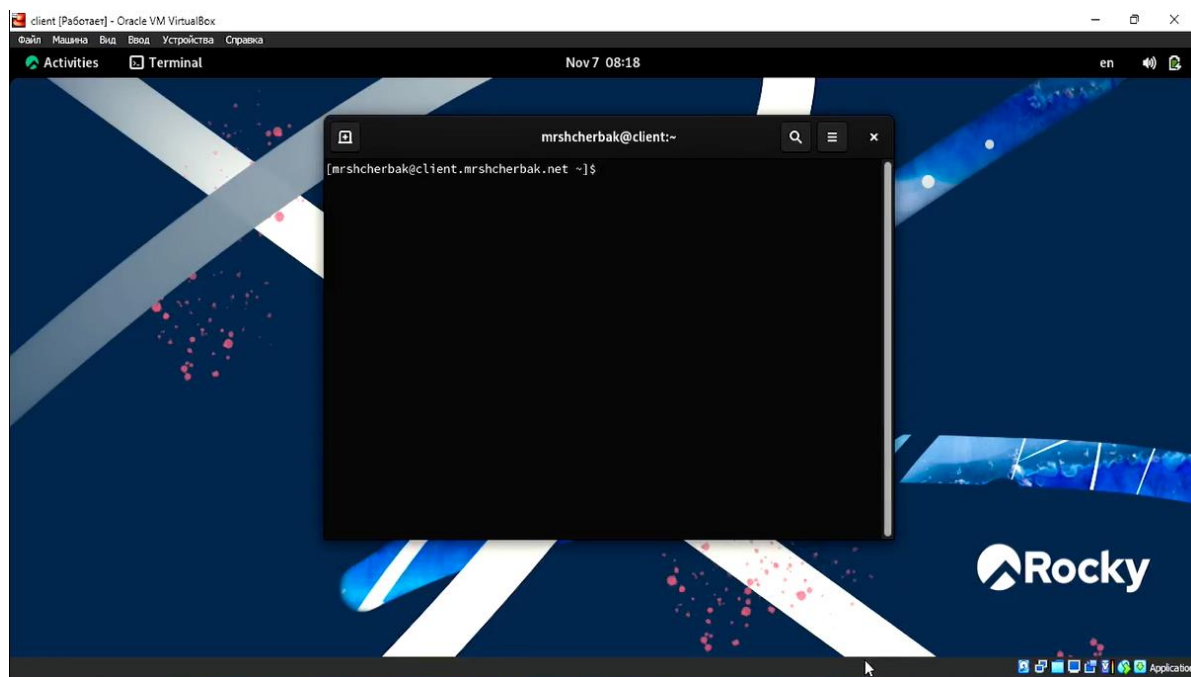


Рис.3.5. На клиенте приглашение отображается в виде
`mrshcherbak@client.mrshcherbak.net`

После выключения виртуальных машин скопировала необходимые для работы с Vagrant файлы и box-файлы виртуальных машин в другой каталог (рис.3.6).

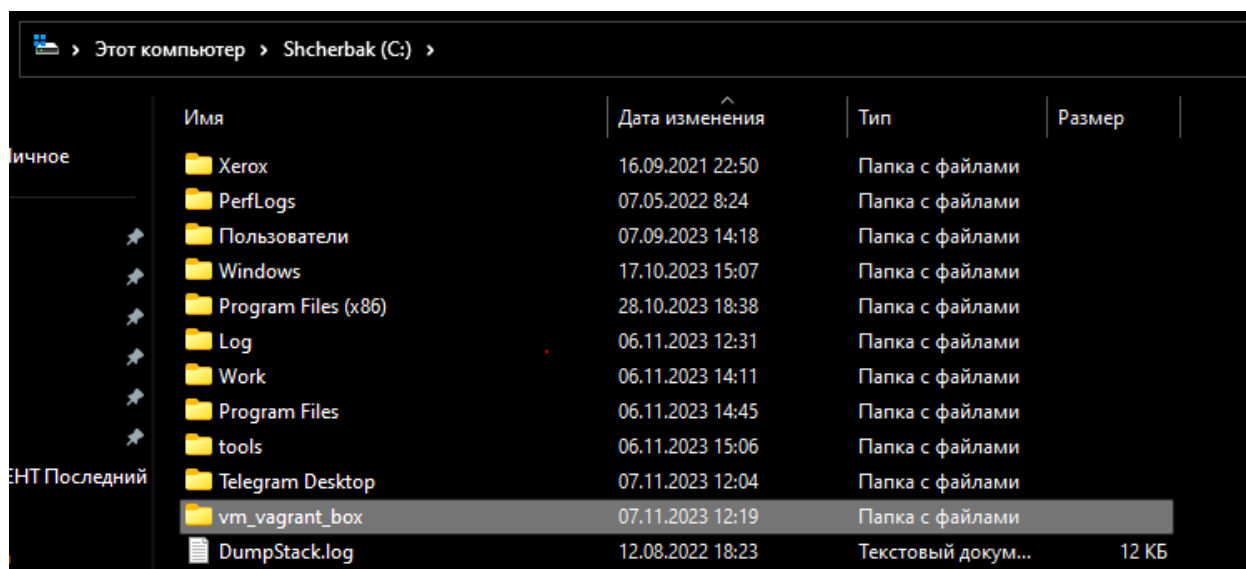


Рис.3.6. Копия необходимых файлов в другой каталог

Вывод: таким образом, в ходе л/р №1 я приобрела практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

Контрольные вопросы

1. Для чего предназначен Vagrant?

Vagrant — представляет собой инструмент для создания и управления средами виртуальных машин в одном рабочем процессе. Этот инструмент, по сути, 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; по сути, box-файл используется как основа для клонирования виртуальных машин с теми или иными настройками.

Vagrantfile - это файл конфигурации для инструмента Vagrant. Он используется для определения параметров и настроек виртуальной среды, в том числе выбора базового box-образа, настройки сети и других параметров

виртуальной машины. Vagrantfile позволяет описывать всю конфигурацию в одном месте, что упрощает управление виртуальными окружениями и делает процесс воспроизводимым.

3. Приведите описание и примеры вызова основных команд Vagrant.

С Vagrant можно работать, используя следующие основные команды:

- `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.

- `vagrant-rocky.pkr.hcl` — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину.

`packer` блок: определяет зависимость от плагина Vagrant и его версии.

`variable` блоки: определяют переменные, которые будут использоваться в конфигурации Packer. Например, `artifact_description`, `artifact_version`, и так далее.

source "virtualbox-iso" "virtualbox" блок: определяет источник для создания виртуальной машины с использованием образа VirtualBox.

build блок: определяет, какие источники использовать и какие провижинеры и пост-процессоры применять.

provisioner "shell" блок: используется для выполнения команд внутри виртуальной машины. В данном случае, устанавливает необходимое ПО и конфигурирует систему, также производится установка VirtualBox Guest Additions.

post-processor "vagrant" блок: определяет пост-процессор для создания Vagrant box файла. Указывает уровень сжатия и имя выходного файла.

- 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.;

bootloader блок: указывает параметры загрузчика, такие как аргументы ядра и местоположение загрузчика.

zerombr и clearpart блоки: очищают информацию о разделах для установки системы на "чистый" диск.

reboot блок: задает перезагрузку после завершения установки.

text блок: устанавливает режим установки в текстовом формате.

keyboard, lang блоки: устанавливают раскладку клавиатуры и язык системы.

network блок: конфигурирует сетевые параметры для автоматического получения IP-адреса по DHCP.

authselect и rootpw блоки: выбирают метод аутентификации и устанавливают пароль для пользователя root.

user блок: создает пользователя vagrant с паролем vagrant.

services блок: включает необходимые службы, такие как NetworkManager, sshd, chronyd.

timezone блок: устанавливает временную зону системы.

part блок: определяет раздел / с файловой системой XFS и размером 10 ГБ.

%post блок: содержит команды, которые будут выполнены после завершения установки.

Настроены swap-файл, sudo, сетевой интерфейс, sshd, SELinux, и другие параметры.

%packages блок: содержит список пакетов, которые будут установлены. Например, bash-completion, bzip2, chrony и т.д.

%addon com_redhat_kdump блок: отключает Kdump (причина "disable").

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

Vagrant.configure("2") блок: задает версию конфигурации Vagrant.

Общая конфигурация:

config.vm.provision блоки определяют три шага для общей конфигурации (01-dummy.sh, 01-hostname.sh, 01-user.sh).

Конфигурация сервера (server):

config.vm.define определяет виртуальную машину с именем "server".

Указаны параметры виртуальной машины, такие как имя box, имя хоста, таймаут загрузки, настройки SSH и сетевые параметры.

server.vm.provision блок определяет шаг для конфигурации сервера (01-dummy.sh).

server.vm.provider блок определяет параметры провайдера (VirtualBox) для сервера.

Конфигурация клиента (client):

config.vm.define определяет виртуальную машину с именем "client".

Указаны параметры виртуальной машины, такие как имя box, имя хоста, таймаут загрузки, настройки SSH и сетевые параметры.

client.vm.provision блок определяет два шага для конфигурации клиента (01-dummy.sh, 01-routing.sh).

`client.vm.provider` блок определяет параметры провайдера (VirtualBox) для клиента.

- `Makefile` — набор инструкций для программы `make` по работе с Vagrant.

`help`: выводит справку о доступных командах и их использовании.

`plugins`: устанавливает Vagrant-плагин (`vagrant-vbguest`).

`addbox`: добавляет построенный Vagrant box (`vagrant-virtualbox-rocky-9-x86_64.box`) в Vagrant.

`up`: поднимает все виртуальные машины (сервер и клиент).

`server-up` и `client-up`: поднимают только сервер или только клиента.

`server-provision` и `client-provision`: поднимают и (выполняют дополнительные настройки) сервер или клиента.

`server-destroy` и `client-destroy`: уничтожают (останавливают и удаляют) сервер или клиента.