

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**дисциплина: Администрирование сетевых подсистем**

Студент: Астахова Марина Дмитриевна

Группа: НПИбд-02-23

МОСКВА

# **Тема: Подготовка Лабораторного стенда**

## **Цель работы:**

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

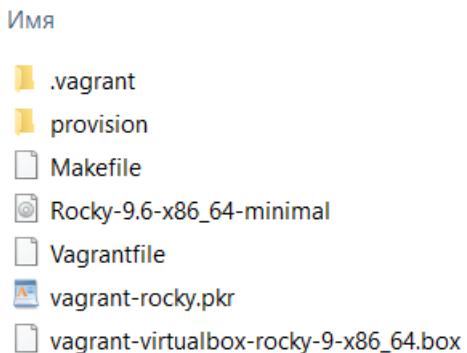
## **Выполнение работы:**

### **A. Подготовка каталога для работы**

1. Перед началом работы с Vagrant создали каталог для проекта.

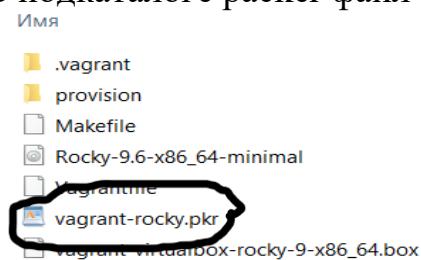
- `mkdir -p C:\Users\Marine\Admitnet\packer`
- `mkdir -p C:\Users\Marine\Admitnet\vagrant`

2. В созданном рабочем каталоге в подкаталоге packer разместили образ варианта операционной системы Rocky Linux



3. В этом же рабочем каталоге разместили подготовленные заранее для работы с Vagrant файлы:

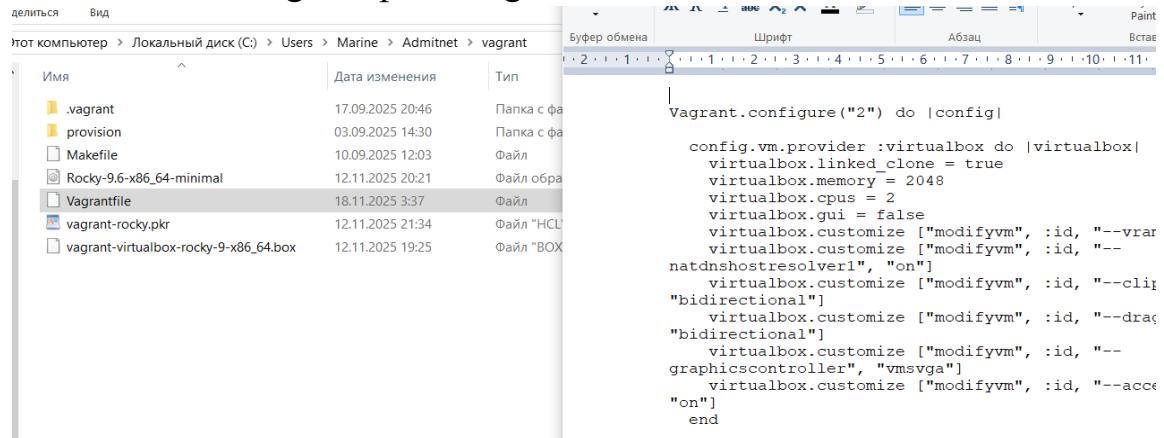
А. В подкаталоге packer файл `vagrant-rocky.pkr.hcl`



## B. В подкаталоге packer подкаталог http с файлом ks.cfg

Имя	Дата изменения	Тип
ks.cfg	03.09.2025 18:06	Текстовый докум...
rocky9-ks.cfg	12.11.2025 20:23	Файл "CFG"

## C. В подкаталоге vagrant файл Vagrantfile



```
Vagrant.configure("2") do |config|  
  config.vm.provider :virtualbox do |virtualbox|  
    virtualbox.linked_clone = true  
    virtualbox.memory = 2048  
    virtualbox.cpus = 2  
    virtualbox.gui = false  
    virtualbox.customize ["modifyvm", :id, "--vrar"]  
    virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]  
    virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]  
    virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]  
    virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]  
    virtualbox.customize ["modifyvm", :id, "--accelerate3d", "on"]  
  end
```

## 4. В этом же рабочем каталоге в подкаталоге vagrant создали каталог provision с подкаталогами default, server и client

Поделиться	Вид
Этот компьютер > Локальный диск (C:) > Users > Marine > Admitnet > vagrant > provision	
Имя	Дата изменения
client	28.10.2025 16:05
default	16.09.2025 21:11
server	29.10.2025 18:45

## 5. В каталогах default, server и client разместили заранее подготовленный скрипт-заглушку 01-dummy.sh следующего содержания:

Локальный диск (C:) > Users > Marine > Admitnet > vagrant > provision > client

Имя	Дата изменения	Тип	Размер
01-dummy	10.09.2025 12:03	Shell Script	1 КБ
01-routing	02.10.2025 15:31	Shell Script	1 КБ
mail.sh	28.10.2025 16:05	Shell Script	1 КБ

01-dummy - WordPad

```

Файл Главная Вид
Courier New 11 A A
Ж K Ч а ве X2 A - -
Буфер обмена Шрифт Абзац Изображение Рисунок
Ри
В
#! /bin/bash
echo "Provisioning script $0"

```

6. В каталоге default разместили заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины следующего содержания:

Локальный диск (C:) > Users > Marine > Admitnet > vagrant > provision > default

Имя	Дата изменения	Тип	Размер
01-dummy	10.09.2025 12:03	Shell Script	1 КБ
01-hostname	10.09.2025 12:03	Shell Script	1 КБ
01-user	12.11.2025 22:03	Shell Script	1 КБ

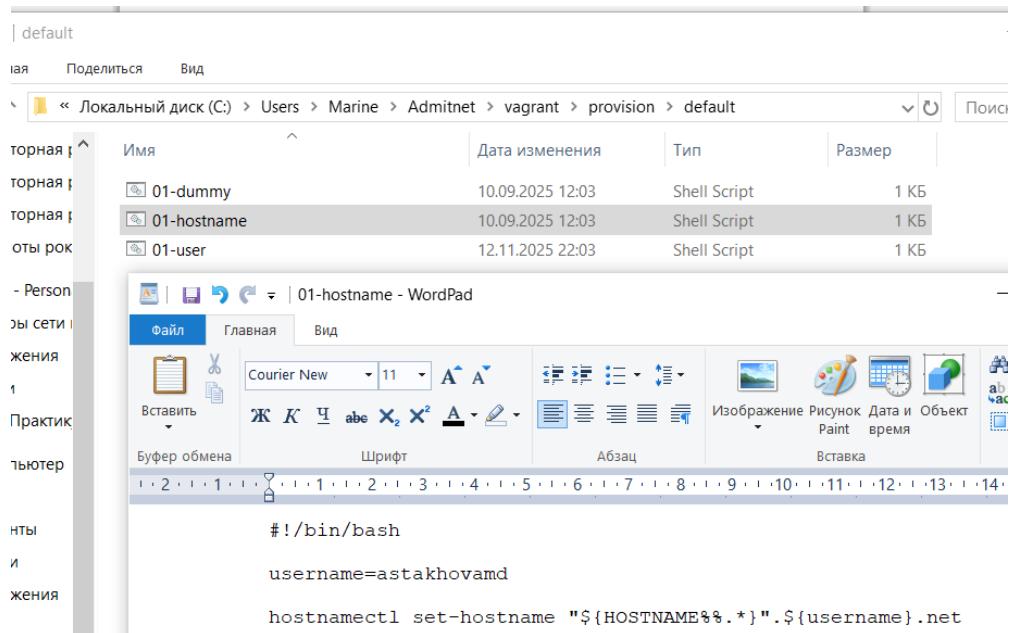
01-user - WordPad

```

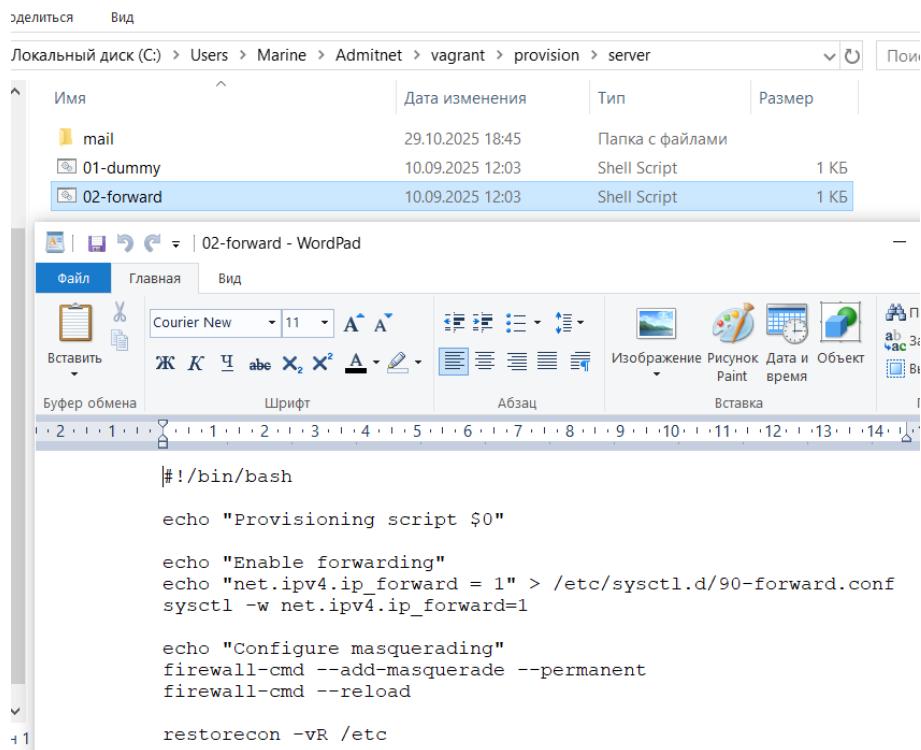
Файл Главная Вид
Courier New 11 A A
Ж K Ч а ве X2 A - -
Буфер обмена Шрифт Абзац Изображение Рисунок Дата и Объект
П
Вставка
#! /bin/bash
echo "Provisioning script $0"
username=astakhovamd
userpassword=12345
encpassword=`openssl passwd -1 ${userpassword}`
id -u $username
if [[ $? ]]
then
    adduser -G wheel -p ${encpassword} ${username}
    homedir=`getent passwd ${username} | cut -d: -f6`
    echo "export PS1='[\u@\H \W]\$\'" >> ${homedir}/.bashrc \

```

7. В каталоге default разместили заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины следующего содержания:



8. В каталоге server разместим заранее подготовленный скрипт 02-forward.sh следующего содержания:



9. В каталоге client разместите заранее подготовленный скрипт 01-routing.sh следующего содержания:

Поделиться Вид

Локальный диск (C:) > Users > Marine > Admitnet > vagrant > provision > client

Имя	Дата изменения	Тип	Размер
01-dummy	10.09.2025 12:03	Shell Script	1 КБ
01-routing	02.10.2025 15:31	Shell Script	1 КБ
mail.sh	28.10.2025 16:05	Shell Script	1 КБ

01-routing - WordPad

Файл Главная Вид

Вставить Буфер обмена Шрифт Абзац Вставка

Courier New 11 A A Ж K Ч а ве X<sub>2</sub> X<sup>2</sup> А А

Изображение Рисунок Дата и Объект  
Paint время

```
#!/bin/bash
echo "Provisioning script $0"
nmcli connection modify "eth1" ipv4.gateway "192.168.1.1"
nmcli connection up "eth1"

nmcli connection modify eth0 ipv4.never-default true
nmcli connection modify eth0 ipv6.never-default true

nmcli connection down eth0
nmcli connection up eth0

# systemctl restart NetworkManager
```

## ***В. Развёртывание лабораторного стенда на ОС Linux***

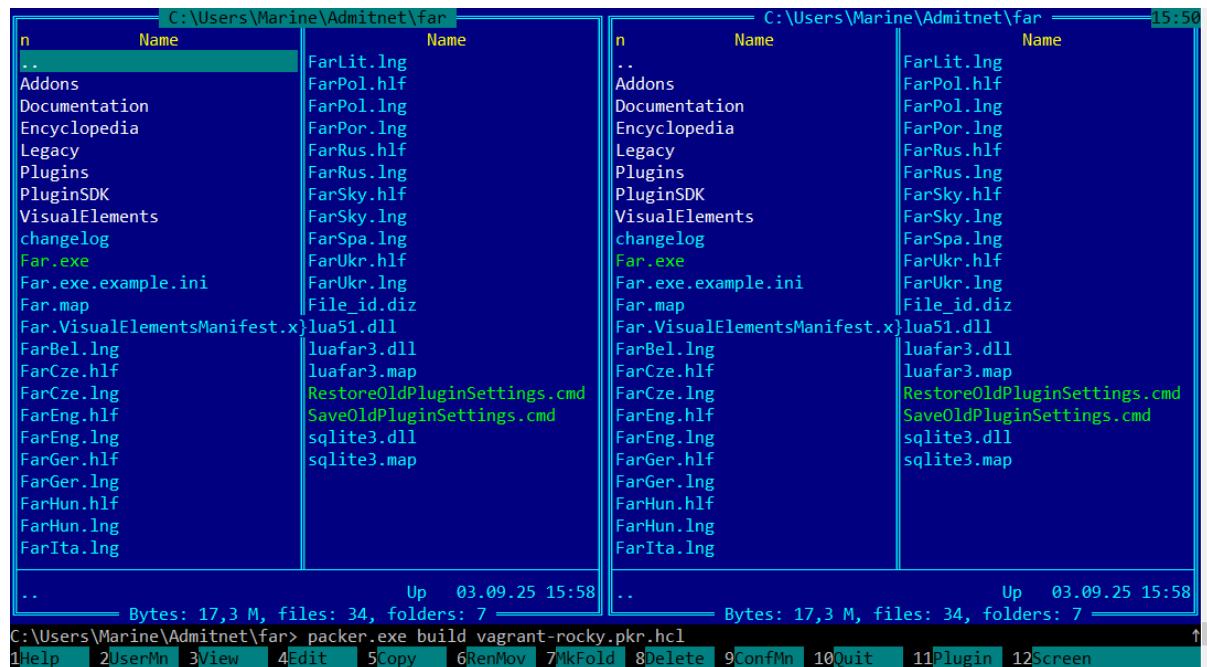
1. Перейдем в каталог с проектом:  
cd C:\Users\Marine\Admitnet\vagrant

grant

Папка .vagrant				
Имя				
.vagrant		17.09.2025 20:46	Папка с файлами	
provision		03.09.2025 14:30	Папка с файлами	
Makefile		10.09.2025 12:03	Файл	7 КБ
Rocky-9.6-x86_64-minimal		12.11.2025 20:21	Файл образа диска	2 199 232 ...
Vagrantfile		18.11.2025 3:37	Файл	4 КБ
vagrant-rocky.pkr		12.11.2025 21:34	Файл "HCL"	6 КБ
vagrant-virtualbox-rocky-9-x86_64.box		12.11.2025 19:25	Файл "BOX"	4 111 265 ...

2. Используя FAR, перейдем в созданный рабочий каталог с проектом. В этом же каталоге должен быть размещён файл packer.exe. В командной строке введем

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



3. Для регистрации образа виртуальной машины в vagrant в командной строке введите vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86\_64.box

```
Couldn't open file C:/Users/Marine/Admitnet/packer/rockylinux-virtualbox-rocky-9-x86_64.box
PS C:\Users\Marine\Admitnet\packer>
PS C:\Users\Marine\Admitnet\packer> vagrant box add rockylinux/9 vagrant-virtualbox-rocky-9-x86_64.box
--> box: Box file was not detected as metadata. Adding it directly...
--> box: Adding box 'rockylinux/9' (v0) for provider: (amd64)
  box: Unpacking necessary files from: file:///C:/Users/Marine/Admitnet/packer/vagrant-virtualbox-rocky-9-x86_64
  box:
--> box: Successfully added box 'rockylinux/9' (v0) for '(amd64))!
PS C:\Users\Marine\Admitnet\packer>
```

#### 4. Для запуска виртуальной машины Server введите в консоли vagrant up server\

```
PS C:\Users\Marine\Admitnet\vagrant> vagrant up
Bringing machine 'server' up with 'virtualbox' provider...
Bringing machine 'client' 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: 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: 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: Running provisioner: common dummy (shell)...
  server: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-uxg3ed.sh
  server: Provisioning script /tmp/vagrant-shell
--> server: Running provisioner: common hostname (shell)...
  server: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-3ju5sq.sh
--> server: Running provisioner: common user (shell)...
  server: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-cbmbmx.sh
  server: Provisioning script /tmp/vagrant-shell
  server: id: 'astakhovamd': no such user
--> server: Running provisioner: server dummy (shell)...
  server: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-pwotn8.sh
  server: Provisioning script /tmp/vagrant-shell
--> client: Running provisioner: common dummy (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-vse5cg.sh
  client: Provisioning script /tmp/vagrant-shell
--> client: Running provisioner: common hostname (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-js0q1g.sh
--> client: Running provisioner: common user (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-x8ilfs.sh
  client: Provisioning script /tmp/vagrant-shell
  client: id: 'astakhovamd': no such user
--> client: Running provisioner: client dummy (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-3o08d8.sh
  client: Provisioning script /tmp/vagrant-shell
--> client: Running provisioner: client routing (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-14804-eb63s0.sh
  client: Provisioning script /tmp/vagrant-shell
  client: Error: unknown connection 'eth1'.
  client: Error: unknown connection 'eth1'.
  client: Connection 'eth0' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/2)
    client: Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
```

#### 5. Для запуска виртуальной машины Client введите в консоли vagrant up client

```
client: Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
PS C:\Users\Marine\Admitnet\vagrant> vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Machine already provisioned. Run `vagrant provision` or use the `--provision` flag to force provisioning. Provisioners marked to run always will still run.
==> client: Running provisioner: common hostname (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-10968-p249i9.sh
==> client: Running provisioner: client routing (shell)...
  client: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-10968-77ivlr.sh
  client: Provisioning script /tmp/vagrant-shell
  client: Error: unknown connection 'eth1'.
  client: Error: unknown connection 'eth1'.
  client: Connection 'eth0' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
  client: Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
PS C:\Users\Marine\Admitnet\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 network to not work properly. If the network doesn't work properly, try changing this IP.
==> 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 network to not work properly. If the network doesn't work properly, try changing this IP.
==> 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 network to not work properly. If the network doesn't work properly, try changing this IP.
==> server: Machine already provisioned. Run `vagrant provision` or use the `--provision` flag to force provisioning. Provisioners marked to run always will still run.
==> server: Running provisioner: common hostname (shell)...
  server: Running: C:/Users/Marine/AppData/Local/Temp/vagrant-shell20251112-9684-xq00xf.sh
PS C:\Users\Marine\Admitnet\vagrant>
```

Лицес. что в ко

## 6. Выключим Машины

```
[vagrant@localhost ~]$ exit
logout
PS C:\Users\Marine\Admitnet\vagrant>
```

## *Выход*

В ходе выполнения лабораторной работы были получены навыки развертывания лабораторного стенда и запуска виртуальной машины на vagrant.

## ***Контрольные вопросы***

1. *Для чего предназначен Vagrant?*

Vagrant предназначен для создания и управления виртуальными средами разработки. Его основная цель - обеспечить воспроизводимые, портативные и изолированные среды для разработчиков.

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

Box-файл - это пакетный формат для Vagrant, содержащий в себе готовую виртуальную машину (VM). Он включает в себя образ диска (обычно в формате .vmdk, .vdi или .img) и метаданные, описывающие VM. По сути, это шаблон, на основе которого Vagrant создает экземпляры виртуальных машин.

Box-файлы обеспечивают удобный способ распространения и повторного использования предварительно настроенных VM. Это избавляет от необходимости каждый раз с нуля устанавливать операционную систему, необходимое ПО и выполнять базовую настройку. Box-файлы позволяют быстро и легко запускать виртуальные машины с определенной конфигурацией, необходимой для разработки, тестирования или развертывания приложений.

## **3. Приведите описание и примеры вызова основных команд Vagrant**

Vagrantfile - это файл конфигурации, написанный на языке Ruby, который описывает, как Vagrant должен запускать и настраивать виртуальную машину. Он определяет, какой box-файл использовать, какие ресурсы (CPU, RAM) выделить, какие сетевые настройки применить, какие общие папки создать и какие provision-скрипты выполнить.

Vagrantfile служит центральным местом для определения конфигурации VM. Он позволяет воспроизводимо создавать одинаковые VM на разных машинах, обеспечивая консистентность окружения разработки. Vagrantfile можно хранить в системе контроля версий (например, Git), что позволяет отслеживать изменения в конфигурации VM и делиться ею с другими разработчиками.

## **4. Дайте построчные пояснения содержания файлов vagrant: rocky.pkr.hcl, ks.cfg, Vagrantfile, Makefile.**

### **rocky.pkr.hcl (Packer Configuration)**

- **Заголовок:** Определение источника виртуальной машины, в данном случае, VirtualBox. Задаёт ему имя "rocky".
- **Имя виртуальной машины:** Указывает имя, которое будет у виртуальной машины в VirtualBox.

- **Тип гостевой ОС:** Определяет, какой тип операционной системы (RedHat 64-bit) установлен на виртуальной машине.
- **Каталог вывода:** Указывает папку, куда будет сохранён созданный образ виртуальной машины.
- **Режим без графического интерфейса:** Определяет, будет ли виртуальная машина запускаться без отображения графического интерфейса во время сборки.
- **Размер диска:** Устанавливает размер виртуального жёсткого диска.
- **Количество процессоров:** Определяет количество процессоров, выделяемых для виртуальной машины.
- **Объём оперативной памяти:** Устанавливает количество оперативной памяти, выделяемой для виртуальной машины.
- **Формат вывода:** Задаёт формат файла образа виртуальной машины (OVA - более компактный).
- **Каталог для HTTP-сервера:** Определяет папку, откуда Packer будет раздавать файлы по HTTP во время сборки. (Например, для доступа к файлу автоматической установки ks.cfg).
- **Диапазон портов HTTP-сервера:** Указывает, какие порты может использовать HTTP-сервер Packer.
- **Команда выключения:** Определяет команду, которая будет выполнена для выключения виртуальной машины после завершения сборки.
- **Блок настроек VirtualBox:**

Переименование виртуальной машины: Переименовывает виртуальную машину.

Отключение аудио: Отключает звук.

Отключение USB: Отключает USB-устройства.

Отключение UART1: Отключает последовательный порт UART1.

- **Включение аппаратной виртуализации:** Включает поддержку вложенной виртуализации.

- **Блок сборки:**

Источники: Указывает источник, используемый для сборки виртуальной машины (наш источник "rocky" для VirtualBox).

Provisioner (скрипт обновления): Выполняет скрипт updates.sh для обновления системы.

\* Команда выполнения указывает способ запуска скрипта через sudo  
Provisioner (скрипт окружения рабочего стола):

Выполняет скрипт desktop.sh для установки окружения рабочего стола.

\* Команда выполнения указывает способ запуска скрипта через sudo

- Provisioner (скрипт очистки):

Выполняет скрипт cleanup.sh для удаления временных файлов.

\* Команда выполнения указывает способ запуска скрипта через sudo

Provisioner (удаление артефактов):

Выполняет команды для удаления файла конфигурации установки, machine ID и перегенерацию machine ID.

Команда выполнения указывает способ запуска скрипта через sudo

## **2. *ks.cfg (Kickstart Configuration)***

- **Версия RHEL:** Указывает, что файл предназначен для Red Hat Enterprise Linux (или совместимой, как Rocky Linux) версии 9.
- **Диск для установки:** Указывает использовать только указанный диск (sda) для установки.
- **Источник установки:** Определяет адрес репозитория для установки пакетов операционной системы.
- **Графический режим установки:** Включает графический интерфейс для процесса установки.
- **Язык:** Устанавливает язык системы.
- **Раскладка клавиатуры:** Устанавливает раскладку клавиатуры.
- **Сетевые настройки:** Настраивает сетевой интерфейс (eth0) для получения IP-адреса по DHCP. IPv6 отключен.
- **Имя хоста:** Устанавливает имя компьютера.
- **Пароль root:** Задаёт пароль для учетной записи root (небезопасно в plaintext).
- **Создание пользователя:** Создаёт учетную запись пользователя с именем "vagrant" и паролем "vagrant", добавляет его в группу "wheel".
- **Часовой пояс:** Устанавливает часовой пояс.
- **Загрузчик:** Настраивает загрузчик операционной системы на указанный диск.
- **Инициализация разделов:** Очищает диск и создаёт новую таблицу разделов.
- **Разделы диска:** Определяет размеры и типы файловых систем для разделов /boot, swap и корневого раздела /.
- **Список пакетов:** Указывает группы пакетов, которые будут установлены (окружение рабочего стола, базовая система, утилиты для работы с контейнерами).
- **Скрипт после установки:**

Разрешение sudo без пароля для пользователя vagrant: Добавляет правило, позволяющее пользователю "vagrant" выполнять команды sudo без ввода пароля.

Отключение TTY для sudo: Отключает требование TTY для sudo.

- **Установка Guest Additions:** Устанавливает необходимые пакеты для VirtualBox Guest Additions (для лучшей интеграции виртуальной машины с хост-системой). Монтирование и запуск установки Guest Additions: Монтирует CD-ROM, запускает установщик Guest Additions и размонтирует CD-ROM.

## **3. *Vagrantfile***

- **Заголовок:** Указывает версию API Vagrant, используемую в этом файле конфигурации.
- **Имя Box:** Указывает имя box-файла, который будет использоваться для создания виртуальной машины.
- **Версия Box:** Указывает версию box-файла.
- **URL Box:** Определяет, где Vagrant может найти box-файл (в данном случае, локальный файл).
- **Проброс портов:** Настраивает перенаправление трафика с порта 8080 на хост-машине на порт 80 на виртуальной машине.
- **Приватная сеть:** Создаёт приватную сеть между хост-машиной и виртуальной машиной, присваивая виртуальной машине статический IP-адрес.
- **Конфигурация VirtualBox:**
  - Имя виртуальной машины:** Устанавливает имя виртуальной машины в VirtualBox.
  - Объем памяти:** Определяет, сколько памяти будет выделено виртуальной машине.
  - Количество CPU:** Определяет, сколько CPU будет выделено виртуальной машине.
  - Общая папка:** Создаёт общую папку между текущей папкой на хост-машине и папкой /vagrant на виртуальной машине.
- **Provisioner (установка ПО):** Выполняет shell-скрипт для установки Python 3 и Ansible.

#### **4. Makefile**

- **Тип цели:** Определяет фиктивные цели.
- **Цель по умолчанию:** Указывает, что цель box будет выполнена по умолчанию при вызове make.
- **Цель box (создание box-файла):**

Сборка с Packer: Запускает Packer для сборки образа виртуальной машины на основе конфигурации rocky.pkr.hcl.

Упаковка с Vagrant: Упаковывает готовую виртуальную машину в box-файл. Важно помнить, что перед этим шагом VM должна быть создана Vagrant на основе образа Packer.
- **Цель clean (очистка):**

Удаление выходной папки: Удаляет папку, содержащую сгенерированные файлы, чтобы начать с чистого листа.

