

# **Лабораторная работа №1**

**Дисциплина: Администрирование сетевых подсистем**

Жибицкая Евгения Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Контрольные вопросы</b>	<b>11</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

# Список иллюстраций

2.1	Установка необходимых программ . . . . .	6
2.2	Структура каталогов . . . . .	6
2.3	Структура packer . . . . .	7
2.4	Файлы vagrant . . . . .	7
2.5	Скрипты . . . . .	8
2.6	Установка ОС. Терминал . . . . .	9
2.7	Установка ОС . . . . .	9
2.8	Регистрация в vagrant . . . . .	9
2.9	Проверка скрипта . . . . .	10

## **Список таблиц**

# 1 Цель работы

Приобретение навыков установки Rocky Linux на виртуальную машину с помощью Vagrant.

## 2 Выполнение лабораторной работы

Перед непосредственным развертыванием машины необходимо подготовить пространство, установить различные программы и дополнения. Установим последнюю версию Vagrant, образ диска для Rocky, также нам понадобится Far и Packer(рис. 2.1).

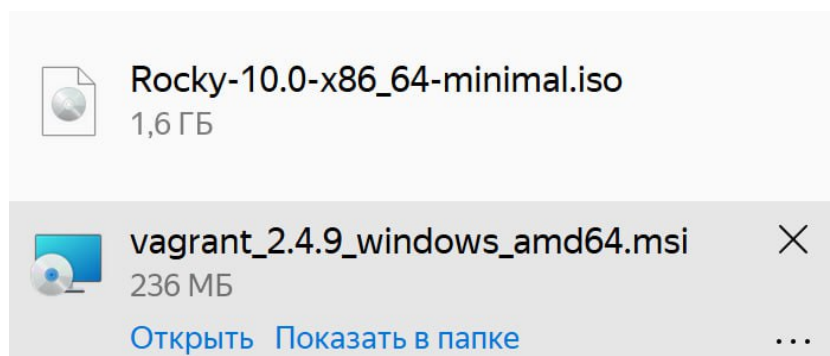


Рис. 2.1: Установка необходимых программ

Далее приступим к созданию рабочей структуры. Создадим каталоги для работы и разместим нужные подкаталоги: work - edzhbitskaya - packer и vagrant(рис. 2.2).

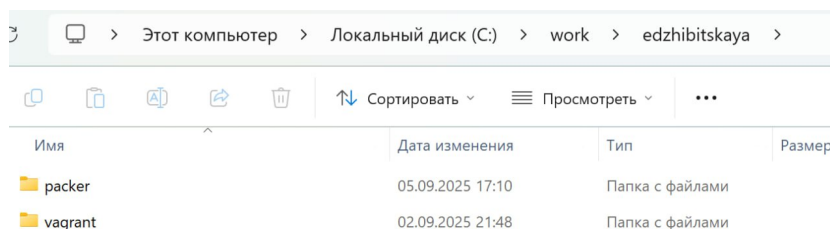
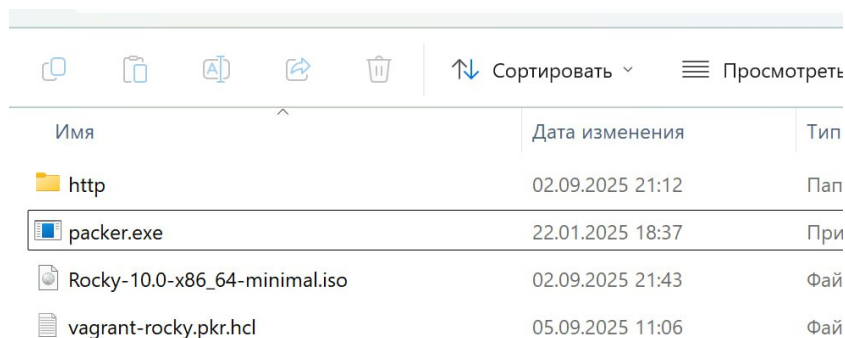


Рис. 2.2: Структура каталогов

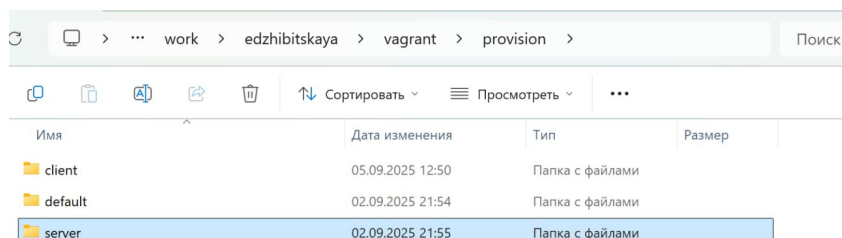
В каталог packer добавляем образ диска, файл vagrant-rocky.pkr.hcl и подкаталог http с файлом ks.cfg, который определяет настройки для установки дистрибутива, которые пользователь обычно вводит вручную. Также добавляем туда packer.exe (рис. 2.3).



Имя	Дата изменения	Тип
http	02.09.2025 21:12	Папка
packer.exe	22.01.2025 18:37	Приложение
Rocky-10.0-x86_64-minimal.iso	02.09.2025 21:43	Файл
vagrant-rocky.pkr.hcl	05.09.2025 11:06	Файл

Рис. 2.3: Структура packer

В каталоге vagrant размещаем vagrantfile, создаем 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. Размещаем там заранее подготовленный скрипт заглушку 01-dummy.sh (рис. 2.4).



Имя	Дата изменения	Тип	Размер
client	05.09.2025 12:50	Папка с файлами	
default	02.09.2025 21:54	Папка с файлами	
server	02.09.2025 21:55	Папка с файлами	

Рис. 2.4: Файлы vagrant

В созданных файлах размещаем данные нам скрипты, меняем в файле `vagrant-rocky.pkr.hcl` значение чек-суммы, версию Rocky(рис. 2.5).

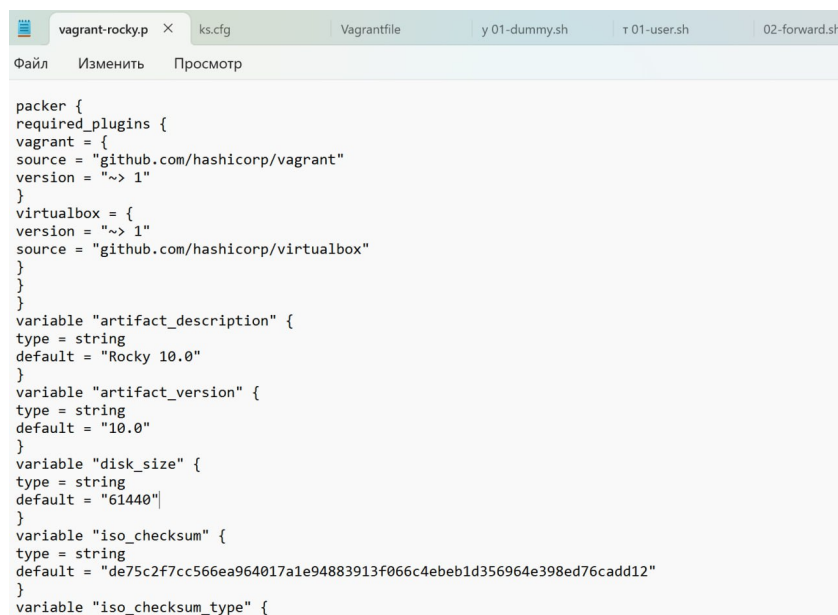


Рис. 2.5: Скрипты

Далее переходим к непосредственно разворачиванию лабораторного стенда. Открываем терминал и с помощью `far` или же просто командной строки переходим в каталог, создаем машину используя команды

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

Запускается установка образа операционной системы(рис. 2.6) и (рис. 2.7).



```

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

==> virtualbox-iso.virtualbox: Retrieving Guest additions
==> virtualbox-iso.virtualbox: Trying C:\Program Files\Oracle\VirtualBox\VBGuestAddi
==> virtualbox-iso.virtualbox: Trying file://C:/Program%20Files/Oracle/VirtualBox/VBox
==> virtualbox-iso.virtualbox: file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAd
acle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.virtualbox: Retrieving ISO
==> virtualbox-iso.virtualbox: Trying Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.virtualbox: Trying Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3A
36c4ebeb1d356964e398ed76cadd12
==> virtualbox-iso.virtualbox: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f
b1d356964e398ed76cadd12 => C:/work/edzhbitskaya/packer/Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.virtualbox: Starting HTTP server on port 8404
==> virtualbox-iso.virtualbox: Creating virtual machine...
==> virtualbox-iso.virtualbox: Creating hard drive builds\packer-rocky-virtualbox-vm.v
==> virtualbox-iso.virtualbox: Mounting ISOs...
==> virtualbox-iso.virtualbox: Mounting boot ISO...
==> virtualbox-iso.virtualbox: Creating forwarded port mapping for communicator (SSH,
==> virtualbox-iso.virtualbox: Executing custom VBoxManage commands...
==> virtualbox-iso.virtualbox: Executing: modifyvm packer-rocky-virtualbox-vm --memory
virtualbox-iso.virtualbox: Executing: modifyvm packer-rocky-virtualbox-vm --cpus 2
virtualbox-iso.virtualbox: Executing: modifyvm packer-rocky-virtualbox-vm --nat-lo
==> virtualbox-iso.virtualbox: Starting the virtual machine...

```

Рис. 2.6: Установка ОС. Терминал

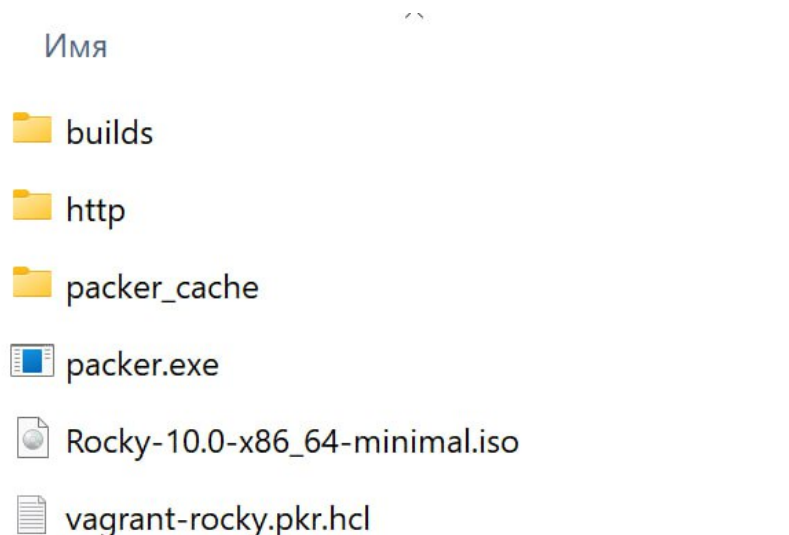


Рис. 2.7: Установка ОС

Для регистрации в Vagrant вводим команду `vagrant box add rocky10 vagrant-virtualbox-rocky-10-x86_64.box`(рис. 2.8).

```

C:\work\edzhbitskaya\packer>
C:\work\edzhbitskaya\packer>vagrant box add rocky10 vagrant-virtualbox-rocky-10-x86_64.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky10' (v0) for provider: (amd64)
box: Downloading: vagrant-virtualbox-rocky-10-x86_64.box

```

Рис. 2.8: Регистрация в vagrant

Далее запускаем ВМ Сервер и Клиент, проверяем, что есть возможность перейти к нашему созданному ранее пользователю и выключаем машину. Для корректной отработки скриптов убеждаемся, что необходимый код есть в Vagrantfile (рис. 2.9). Еще раз логинимся и смотрим, что скрипты отрабатываются. Предварительно фиксируем изменения командами `vagrant up server/cclient --provision`

```
## Common configuration
config.vm.provision "common dummy",
  type: "shell",
  preserve_order: true,
  path: "provision/default/01-dummy.sh"
config.vm.provision "common hostname",
  type: "shell",
  preserve_order: true,
  run: "always",
  path: "provision/default/01-hostname.sh"
config.vm.provision "common user",
  type: "shell",
  preserve_order: true,
  path: "provision/default/01-user.sh"
```

Рис. 2.9: Проверка скрипта

## 3 Контрольные вопросы

1. Для чего предназначен Vagrant?

Он предназначен для создания и управления средами виртуальных машин в одном процессе.

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.

- `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) ранее созданную виртуальную машину);
- `vagrant ssh` — подключение к виртуальной машине через `ssh`.

## 4 Выводы

В ходе работы были получены навыки по работе с Vagrant и установке с его помощью Rocky Linux на виртуальную машину.

# **Список литературы**

ТУИС