

Лабораторная работа №1

Дисциплина: Администрирование сетевых подсистем

Жибицкая Евгения Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	13
4	Выводы	15
	Список литературы	16

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	10
2.9	Запуск	10
2.10	Запуск	10
2.11	Завершение работы	11
2.12	Проверка скрипта	11
2.13	Проверка	11
2.14	Проверка	12
2.15	vagrant up server/cclient –provision	12

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).

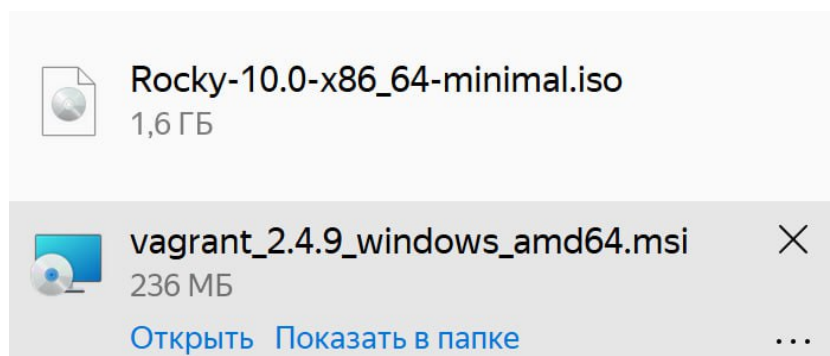


Рис. 2.1: Установка необходимых программ

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

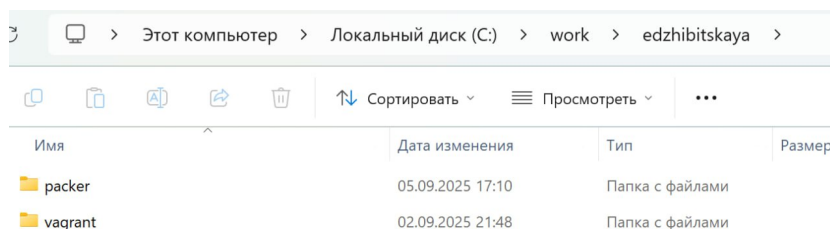
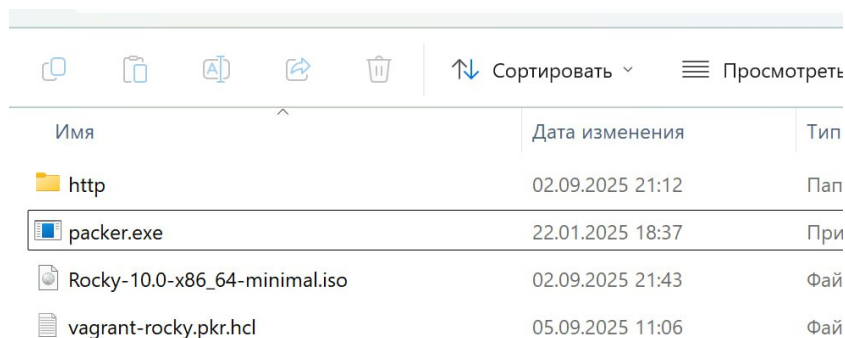


Рис. 2.2: Структура каталогов

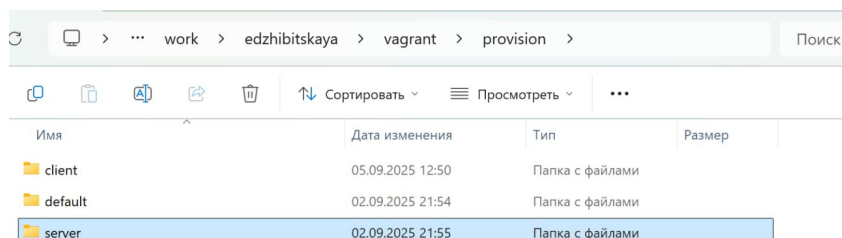
В каталог packer добавляем образ диска, файл vagrant-rocky.pkr.hcl и подкаталог http с файлом ks.cfg, который определяет настройки для установки дистрибутива, которые пользователь обычно вводит вручную. Также добавляем туда packer.exe (рис. 2.3).



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).

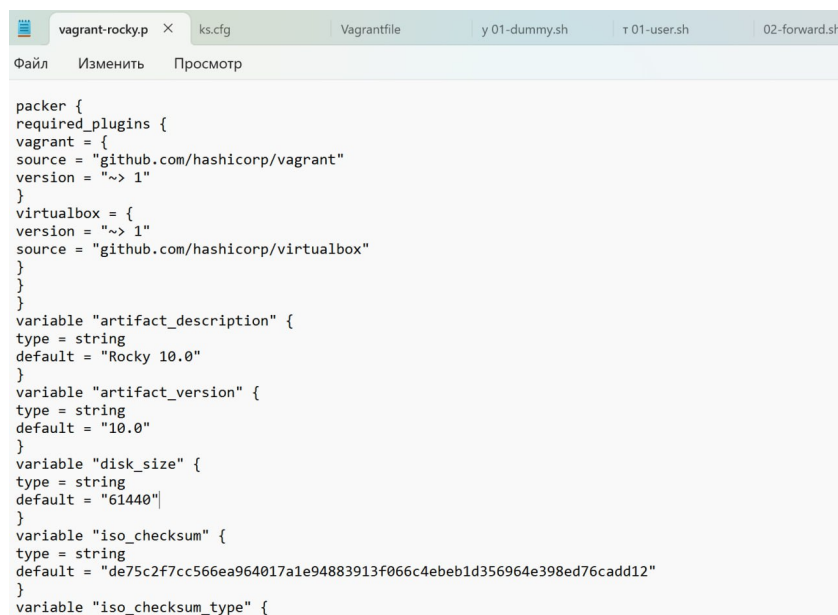


Рис. 2.5: Скрипты

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\edzhibitskaya>cd packer

C:\work\edzhibitskaya\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/qemu v1.1.3 in "C:/Users/janes/AppData/Roaming/packer.d/plugins/github.com/hashicorp/qemu/packer-plugin-qemu_v1.1.3_x5.0_windows_amd64.exe"

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

Build 'qemu.rockylinux' errored after 3 milliseconds 160 microseconds: Failed creating Qemu driver: exec: "qemu-system-x86_64": executable file not found in %PATH%
==> virtualbox-iso.rockylinux: Retrieving Guest additions
==> virtualbox-iso.rockylinux: Trying C:\Program Files\Oracle\VirtualBox\VirtualBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: Trying file://C:/Program Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: file://C:/Program Files/Oracle/VirtualBox/VBoxGuestAdditions.iso => C:/Program Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: Retrieving ISO
==> virtualbox-iso.rockylinux: Trying Rocky-10.0-x86_64-minimal.iso
==> virtualbox-iso.rockylinux: Trying Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4eb1d356964e398ed76cadd12
==> virtualbox-iso.rockylinux: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f7cc566ea964017a1e94883913f066c4eb1d356964e398ed76cadd12 => C:/work/edzhibitskaya/packer/Rocky-10.0-x86_64-minimal.iso

```

Рис. 2.6: Установка ОС. Терминал








 http	06.09.2025 19:38
 packer_cache	06.09.2025 19:41
 scripts	06.09.2025 19:34
 packer.exe	22.01.2025 18:37
 Rocky-10.0-x86_64-minimal.iso	02.09.2025 21:43
 vagrant-rocky.pkr.hcl	06.09.2025 19:32
 vagrant-virtualbox-rockylinux10-x86_64.box	06.09.2025 20:23

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\edzhibitskaya\packer>
C:\work\edzhibitskaya\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-rocky10-x86_64.box

```

Рис. 2.8: Регистрация в vagrant

Далее запускаем VM Сервер и Клиент(рис. 2.9, рис. 2.10), проверяем, что есть возможность перейти к нашему созданному ранее пользователю и выключаем машину(рис. 2.11).Для корректной отработки скриптов убеждаемся, что необходимый код есть в Vagrantfile(рис. 2.12). Еще раз логинимся и смотрим, что скрипты отрабатываются(рис. 2.13, рис. 2.14),. Предварительно фиксируем изменения командами `vagrant up server/ccclient --provision`(рис. 2.15).

```

C:\work\edzhibitskaya\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to this machi
==> 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 creat
server: of new VMs take milliseconds on a modern system.
==> server: Importing base box 'rockylinux10'...
==> server: Cloning VM

```

Рис. 2.9: Запуск

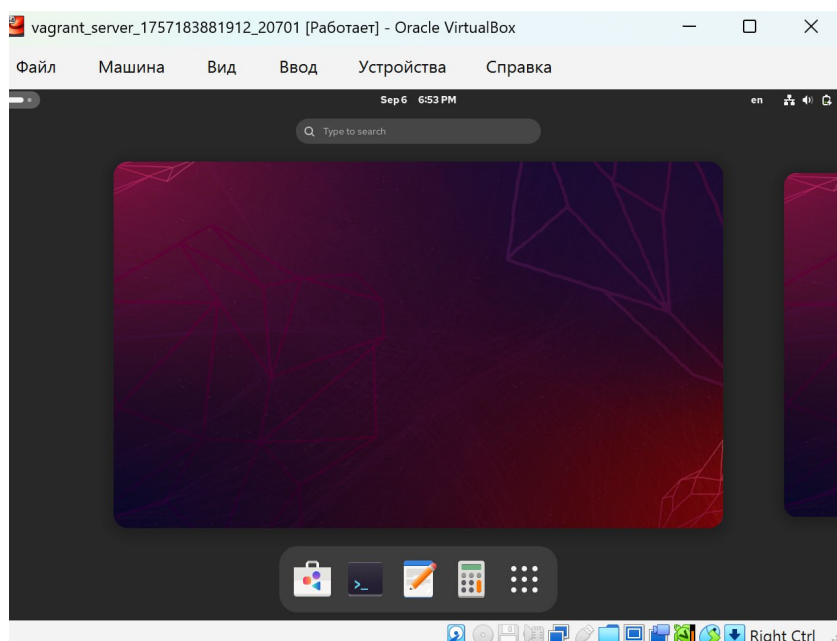


Рис. 2.10: Запуск

```

C:\work\edzhibitskaya\vagrant>vagrant halt server

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

C:\work\edzhibitskaya\vagrant>

```

Рис. 2.11: Завершение работы

```

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

```

Рис. 2.12: Проверка скрипта

```

==> server: Running provisioner: server dummy (shell)...
server: Running: C:/Users/janes/AppData/Local/Temp/vagrant-shell2
906-5692-6wln2o.sh
server: Provisioning script /tmp/vagrant-shell

C:\work\edzhibitskaya\vagrant>vagrant up client --provision

```

Рис. 2.13: Проверка

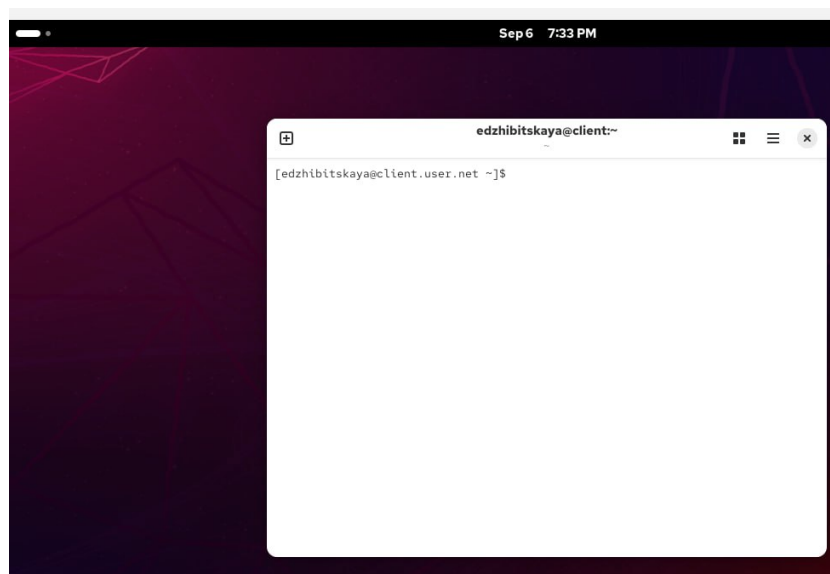


Рис. 2.14: Проверка

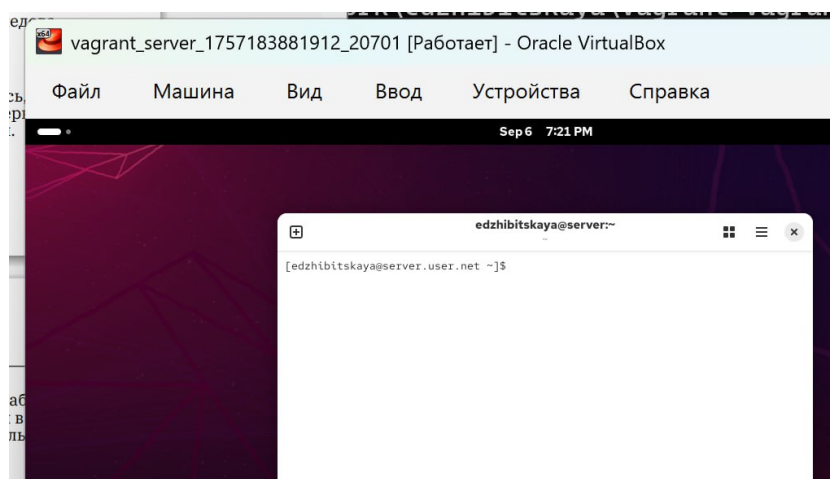


Рис. 2.15: vagrant up server/cclient –provision

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