

Лабораторная №1

Администрирование сетевых подсистем

Жибицкая Е.Д.

Российский университет дружбы народов, Москва, Россия

Цель

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

Ход 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

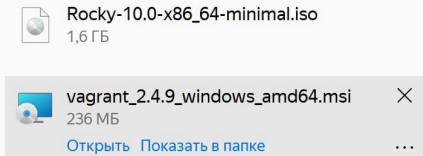


Рис. 1: Установка необходимых программ

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

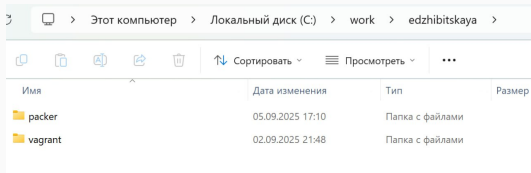


Рис. 2: Структура

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.

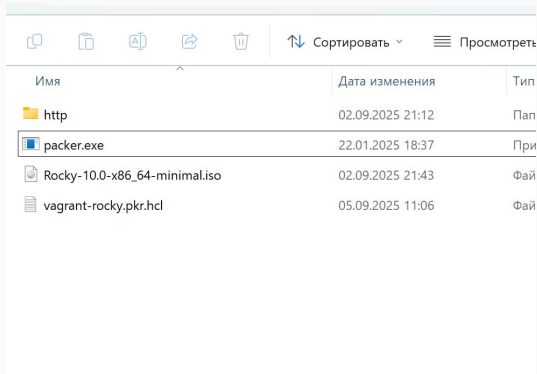


Рис. 3: Структура packer

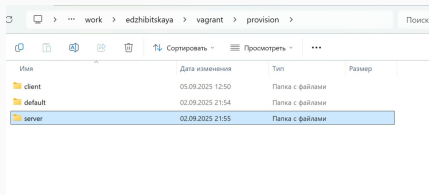
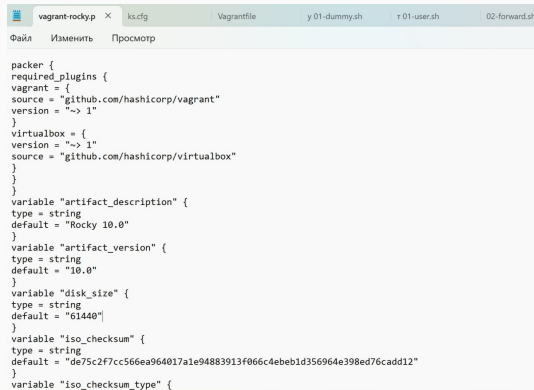


Рис. 4: Структура vagrant

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.



The image shows a code editor window with several tabs: 'vagrant-rocky.p', 'ks.cfg', 'Vagrantfile', 'y 01-dummy.sh', 'r 01-user.sh', and '02-forward.sh'. The 'vagrant-rocky.p' tab is active, displaying a Packer configuration script for Rocky Linux. The script defines required plugins (vagrant and virtualbox), sets their versions to '~> 1', and defines variables for artifact description, version, disk size, and ISO checksum.

```
packer {
  required_plugins {
    vagrant = {
      source = "github.com/hashicorp/vagrant"
      version = "~> 1"
    }
    virtualbox = {
      version = "~> 1"
      source = "github.com/hashicorp/virtualbox"
    }
  }
  variable "artifact_description" {
    type = string
    default = "Rocky 10.0"
  }
  variable "artifact_version" {
    type = string
    default = "10.0"
  }
  variable "disk_size" {
    type = string
    default = "61440"
  }
  variable "iso_checksum" {
    type = string
    default = "de75c2f7cc566ea964017a1e94883913f066c4ebeb1d356964e398ed76cadd12"
  }
  variable "iso_checksum_type" {
```

Рис. 5: Скрипт vagrant-rocky.pkr.hcl

Далее переходим к непосредственно развертыванию лабораторного стенда.

Открываем терминал и с помощью `far` или же просто командной строки переходим в каталог, создаем машину. Команды:

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

Создание образа ОС

```
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/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: Trying file://C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
==> virtualbox-iso.rockylinux: file://C:/Program%20Files/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. 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. 7: Установка ОС. Файлы

```
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. 8: Регистрация

Команда `vagrant box add rocky10`

`vagrant-virtualbox-rocky-10-x86_64.box`

Далее запускаем ВМ Сервер и Клиент, проверяем, что есть возможность перейти к нашему созданному ранее пользователю и выключаем машину.

```
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. 9: Запуск

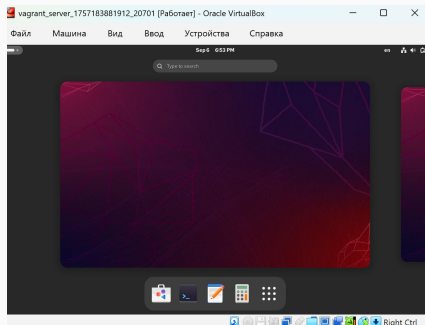


Рис. 10: Запуск

```
# 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. 11: Проверка скрипта

Для корректной отработки скриптов убеждаемся, что необходимый код есть в Vagrantfile. Еще раз логинимся и смотрим, что скрипты отрабатываются.

Предварительно фиксируем изменения командами
vagrant up server/client --provision.

Проверка и завершение

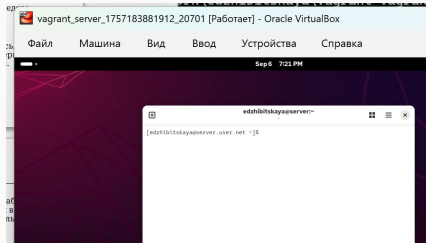


Рис. 12: Проверка

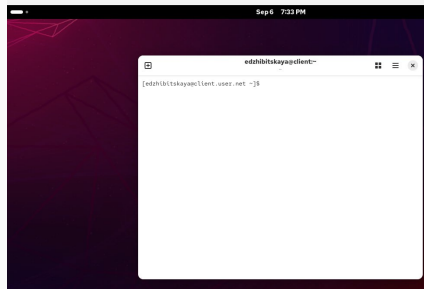


Рис. 13: Проверка

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 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 halt` — остановка и выключение виртуальной машины;
- `vagrant provision` — настройка внутреннего окружения имеющейся виртуальной машины;

Выводы

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