

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

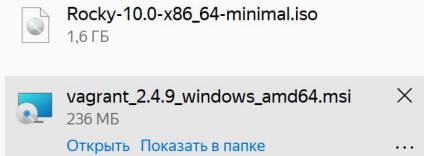


Рис. 1: Установка необходимых программ

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

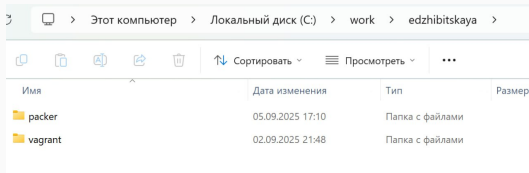


Рис. 2: Структура

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.

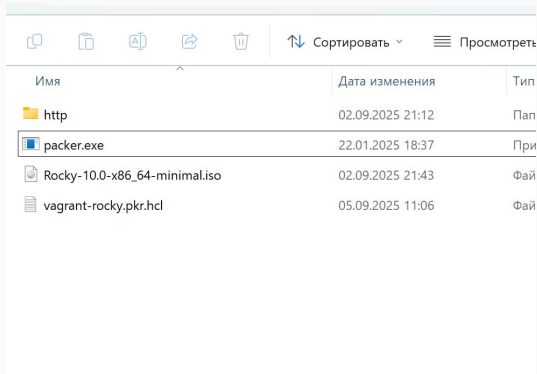


Рис. 3: Структура packer

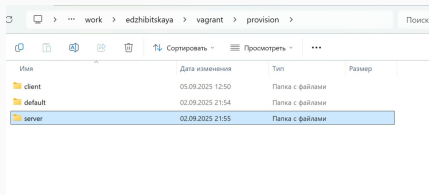
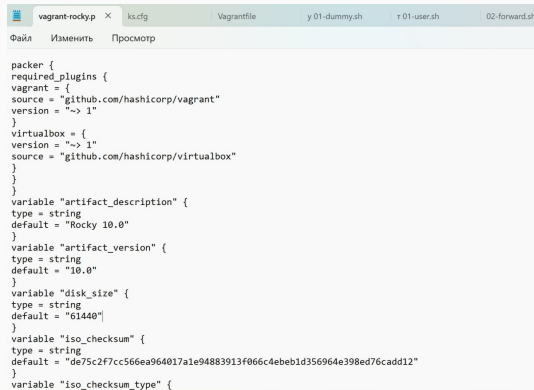


Рис. 4: Структура vagrant

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. The active tab is 'vagrant-rocky.p', and the file being edited is 'ks.cfg'. The code is a HashiCorp Configuration Language (HCL) script for a Vagrant virtual machine. It defines the required plugins (vagrant and virtualbox), sets their versions to '~> 1', and specifies the source for the Rocky Linux ISO. It also defines several 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\packer>packer.exe init vagrant-rocky.pkr.hcl
C:\work\edzhibitskaya\packer>packer.exe build vagrant-rocky.pkr.hcl
virtualbox-iso.virtualbox: 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
66c4eb1d356964e398ed76cadd12
=> virtualbox-iso.virtualbox: Rocky-10.0-x86_64-minimal.iso?checksum=sha256%3Ade75c2f
old356964e398ed76cadd12 => C:/work/edzhibitskaya/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-to
=> virtualbox-iso.virtualbox: Starting the virtual machine...
```

Рис. 6: Установка ОС. Терминал

Имя



builds



http



packer_cache



packer.exe



Rocky-10.0-x86_64-minimal.iso



vagrant-rocky.pkr.hcl

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.

```
regname=$(hostname -s)
## 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. 9: 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/cclient --provision.

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