

Лабораторная работа №1

Подготовка лабораторного стенда

Спелов Андрей Николаевич

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	11
4 Ответы на контрольные вопросы:	12
Список литературы	14

Список иллюстраций

2.1	Создание каталога для проекта.	6
2.2	Размещение образа варианта операционной системы Rocky Linux в рабочем каталоге.	7
2.3	Проверка конфигурационного файла Vagrantfile.	7
2.4	Ввод необходимых команд.	8
2.5	Регистрации образа виртуальной машины в vagrant.	8
2.6	Запуск виртуальной машины Server.	8
2.7	Запуск виртуальной машины Client.	8
2.8	Вход под пользователем vagrant.	9
2.9	Подключение к серверу из консоли.	9
2.10	Переход к пользователю anspelov.	9
2.11	Проверка на сервере.	9
2.12	Выключение виртуальных машин.	10

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Перед началом работы с Vagrant создадим каталог для проекта. В ОС Windows packer и vagrant (рис. 2.1).

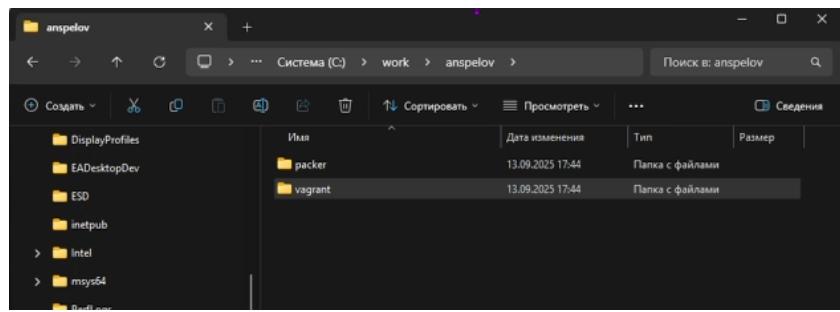


Рис. 2.1: Создание каталога для проекта.

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux, в этом практикуме используем Rocky-10-x86_64-minimal.iso (рис. 2.2). В этом же каталоге разместим подготовленные заранее для работы с Vagrant файлы и создадим 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.

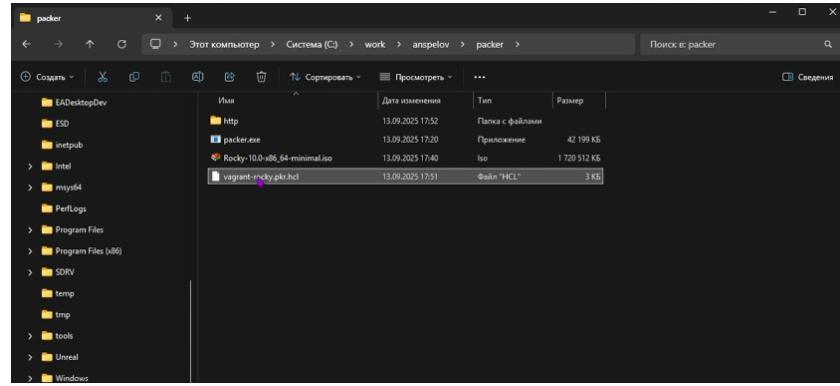


Рис. 2.2: Размещение образа варианта операционной системы Rocky Linux в рабочем каталоге.

Для отработки созданных скриптов во время загрузки виртуальных машин убедимся, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется определённая запись (дана в лабораторной работе) (рис. 2.3).

```

56 client.ssh.username = 'vagrant'
57 client.ssh.password = 'vagrant'
58 client.vm.network :private_network,
59   type: "dhcpc",
60   virtualbox_intnet: true
61 client.vm.provision "client dummy",
62   type: "shell",
63   preserve_order: true,
64   path: "provision/client/01-dummy.sh"
65 client.vm.provision "client routing",
66   type: "shell",
67   preserve_order: true,
68   run: "always",
69   path: "provision/client/01-routing.sh"
70 client.vm.provider :virtualbox do |v|
71   v.linked_clone = true
72   # Customize the amount of memory on the VM
73   v.memory = 1024
74   v.cpus = 1
75   v.name = "client"
76   # Display the VirtualBox GUI when booting the machine
77   v.gui = true
78   # Set the video memory to 12M
79   v.customize ["modifivm", :id, "--vram", "12"]
80   v.customize ["modifivm", :id, "--natdnshostresolver1", "on"]
81   v.customize ["modifivm", :id, "--clipboard", "bidirectional"]
82   v.customize ["modifivm", :id, "--draganddrop", "bidirectional"]
83   v.customize ["modifivm", :id, "--graphicscontroller", "vmsvga"]
84   v.customize ["modifivm", :id, "--accelerate3d", "on"]
85 end
86 end
87 end
88

```

Рис. 2.3: Проверка конфигурационного файла Vagrantfile.

Используя FAR, перейдем в созданный нами рабочий каталог с проектом. В этом же каталоге должен быть размещён файл packer.exe. В командной строке введем команды для начала автоматической установки образа операционной системы Rocky Linux в VirtualBox и последующего формирования box-файла с

дистрибутивом Rocky Linux для VirtualBox (рис. 2.4):

```
C:\work\anspelov\packer> packer.exe init vagrant-rocky.box
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/spelev/AppData/Roaming/packer/plugins/github.com/hashicorp/vagrant/plugin=vagrant_v1.1.5_x5_0_windows_amd64.exe"
```

Рис. 2.4: Ввод необходимых команд.

Для регистрации образа виртуальной машины в vagrant в командной строке введем (рис. 2.5):

```
C:\work\anspelov\packer>vagrant box add rocky10 vagrant-virtualbox-rockylinux10-x86_64.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky10' (v0) for provider: (amd64)
  box: Unpacking necessary files from: file:///C:/work/anspelov/packer/vagrant-virtualbox-rockylinux10-x86_64.box
  box:
==> box: Successfully added box 'rocky10' (v0) for '(amd64)'!
```

Рис. 2.5: Регистрации образа виртуальной машины в vagrant.

Для запуска виртуальной машины Server введем в консоли (рис. 2.6):

```
C:\work\anspelov\packer>vagrant up server
Vagrant has detected project local plugins configured for this
project which are not installed.

  vagrant-vbguest
Install local plugins (Y/N) [N]: Y
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Fetching micromachine-3.0.0.gem
Fetching vagrant-vbguest-0.32.0.gem
Installed the plugin 'vagrant-vbguest (0.32.0)'!
```

Рис. 2.6: Запуск виртуальной машины Server.

Для запуска виртуальной машины Client введите в консоли (рис. 2.7):

```
C:\work\anspelov\packer>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Box 'rockylinux10' could not be found. Attempting to find and insta
  client: Box Provider: virtualbox
  client: Box Version: >= 0
==> client: Box file was not detected as metadata. Adding it directly...
==> client: Adding box 'rockylinux10' (v0) for provider: virtualbox (amd64)
  client: Downloading: rockylinux10
  client:
An error occurred while downloading the remote file. The error
message, if any, is reproduced below. Please fix this error and try
again.
```

Рис. 2.7: Запуск виртуальной машины Client.

Убедимся, что запуск обеих виртуальных машин прошёл успешно, залогинившись под пользователем vagrant с паролем vagrant в графическом окружении (рис. 2.8).

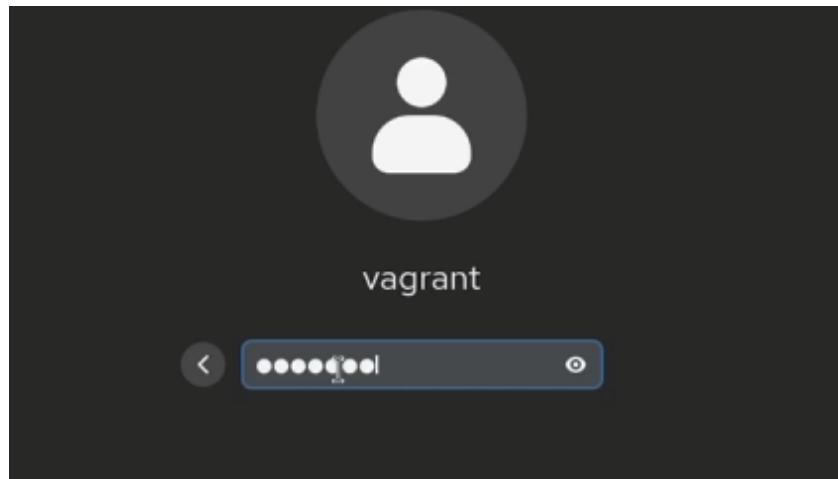


Рис. 2.8: Вход под пользователем vagrant.

Подключимся к серверу из консоли: `vagrant ssh server` Ввод пароль `vagrant` (рис. 2.9).

```
C:\work\anspelov\vagrant>vagrant ssh server
==> server: The machine you're attempting to SSH into is configured to use
==> server: password-based authentication. Vagrant can't script entering the
==> server: password for you. If you're prompted for a password, please enter
==> server: the same password you have configured in the Vagrantfile.
vagrant@127.0.0.1's password:
```

Рис. 2.9: Подключение к серверу из консоли.

Перейдем к пользователю `anspelov` (рис. 2.10): `su – anspelov`

```
vagrant@server:~$ su – anspelov
```

Рис. 2.10: Переход к пользователю `anspelov`.

Отлогинемся и выполним тоже самое для клиента (рис. 2.11).

```
vagrant@client:~$ su – anspelov
Password: |
```

Рис. 2.11: Проверка на сервере.

Выключите обе виртуальные машины (рис. 2.12): `vagrant halt server` `vagrant halt client`

```
C:\work\anspelov\vagrant>vagrant halt server
```

Рис. 2.12: Выключение виртуальных машин.

3 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

4 Ответы на контрольные вопросы:

1. Для чего предназначен Vagrant? – Это инструмент для создания и управления средами виртуальных машин в одном рабочем процессе. Он позволяет автоматизировать процесс установки на виртуальную машину как основного дистрибутива операционной системы, так и настройки необходимого в дальнейшем программного обеспечения.
2. Что такое box-файл? В чём назначение Vagrantfile? - box-файл (или Vagrant Box) — сохранённый образ виртуальной машины с развёрнутой в ней операционной системой, 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) в ранее созданную виртуальную машину); vagrant ssh – подключение к виртуальной машине через ssh.

4. Дайте построчные пояснения содержания файлов vagrant-rocky.pkr.hcl, ks.cfg, Vagrantfile, Makefile. Vagrantfile - Первые две строки указывают на режим работы с Vagrantfile и использование языка Ruby. Затем идёт цикл do, заменяющий конструкцию Vagrant.configure далее по тексту на config. Стока config.vm.box = “BOX_NAME” задаёт название образа (box-файла) виртуальной машины (обычно выбирается из официального репозитория). Стока config.vm.hostname = “HOST_NAME” задаёт имя виртуальной машины. Конструкция config.vm.network задаёт тип сетевого соединения и может иметь следующие назначения: – config.vm.network “private_network”, ip: “xxx.xxx.xxx.xxx” – адрес из внутренней сети; – config.vm.network “public_network”, ip: “xxx.xxx.xxx.xxx” – публичный адрес, по которому виртуальная машина будет доступна; – config.vm.network “private_network”, type: “dhcp” – адрес, назначаемый по протоколу DHCP. Стока config.vm.define “VM_NAME” задаёт название виртуальной машины, по которому можно обращаться к ней из Vagrant и VirtualBox. В конце идёт конструкция, определяющая параметры провайдера, а именно запуск виртуальной машины без графического интерфейса и с выделением 1 ГБ памяти.

Список литературы