Лабораторная работа

Номер 1

Андрюшин Н. С.

01 января 1970

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Андрюшин Никита Сергеевич
- Студент
- Российский университет дружбы народов

Цель работы

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

Создание папок

Для начала создадим папку с инициалами, в которой будет 2 папки, показанные на фото

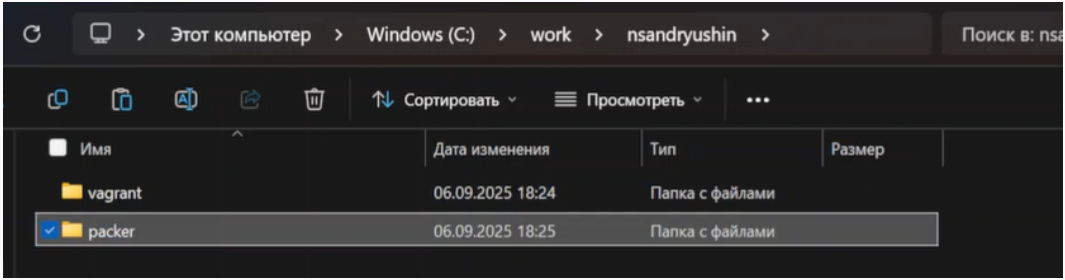


Рис. 1: Создание папок

Образ диска

Поместим заранее скачаный образ в первую папку

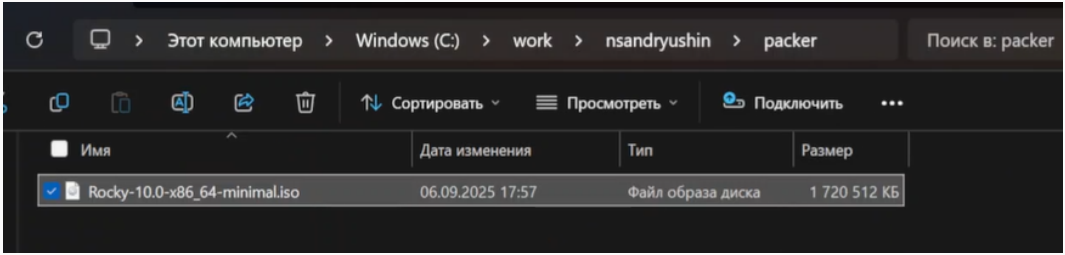


Рис. 2: Образ диска

Создадим в той же папке файл, содержание которого перечислено на экране.
Тут описано, как будет создаваться образ для vagrant

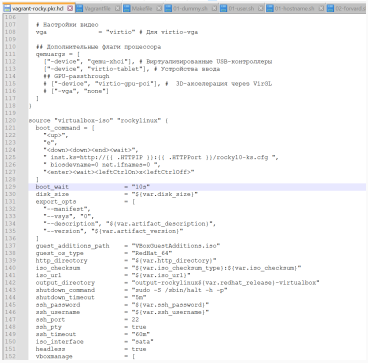


Рис. 3: vagrant-rocky.pkr.hd

Файл конфигурации

Создадим подпапку http, в которой поместим файл следующего содержания с конфигурацией

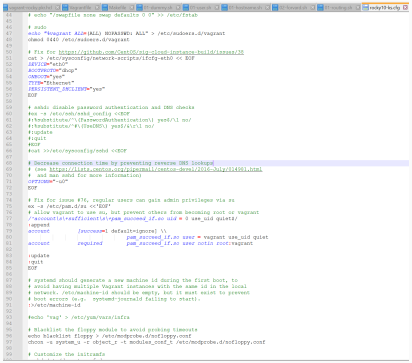
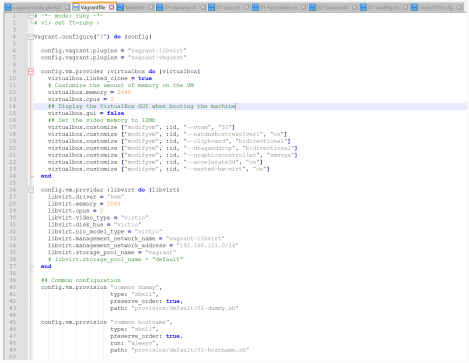


Рис. 4: Файл конфигурации

Vagrantfile

Теперь добавим vagrantfile в папку vagrant



```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  config.vagrant.plugins = "vagrant-libvirt"
  config.vagrant.plugins = "vagrant-vbguest"

  config.vm.provider :virtualbox do |virtualbox|
    virtualbox.linked_clone = true
    # Customize the amount of memory on the VM
    virtualbox.memory = 1040
    virtualbox.cpus = 2
    ## Display the VirtualBox GUI when booting the machine
    virtualbox.gui = false
    ## Set the video memory to 12Mb
    virtualbox.customize ["modifyvm", :id, "--vram", "32"]
    virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
    virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
    virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
    virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
    virtualbox.customize ["modifyvm", :id, "--accelerate3d", "on"]
    virtualbox.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
  end

  config.vm.provider :libvirt do |libvirt|
    libvirt.driver = "kvm"
    libvirt.memory = 2048
    libvirt.cpus = 2
    libvirt.video_type = "virtio"
    libvirt.disk_bus = "virtio"
    libvirt.nic_model_type = "virtio"
    libvirt.management_network_name = "vagrant-libvirt"
    libvirt.management_network_address = "192.168.121.0/24"
    libvirt.storage_pool_name = "vagrant"
    # libvirt.storage_pool_name = "default"
  end

  ## 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"
```

Рис. 5: Vagrantfile

Каталог Vagrant

Содержимое папки будет выглядеть так. Теперь создадим тут папку provision

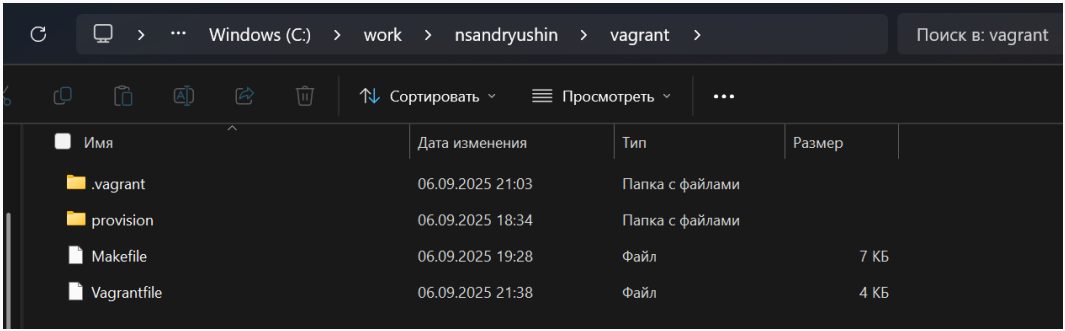


Рис. 6: Каталог Vagrant

Создание подпапок

В этой папке мы создадим 3 подпапки, показаные на фото

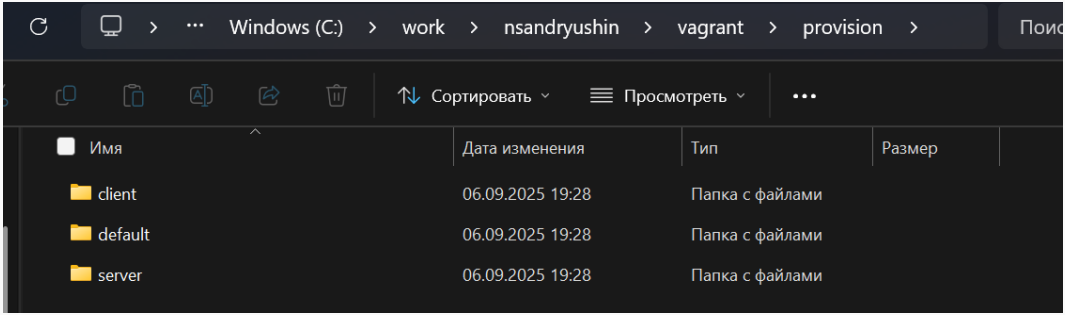


Рис. 7: Создание подпапок

Файл заглушка

В каждую папку добавим файл заглушку со следующим содержанием

vagrant-rocky.pkr.hcl | Vagrantfile | Makefile | 01-dummy.sh | 01-user

```
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5
```

Рис. 8: Файл заглушка

01-user.sh

В папку default добавим следующий файл

vagrant-rocky.pkr.hcl | Vagrantfile | Makefile | 01-dummy.sh | 01-user.sh | 01-hostname.sh | 02-forvard.sh | 01-ro

```
#!/bin/bash

echo "Provisioning script $0"

username=nsandryushin
userpassword=123456

encpassword=`openssl passwd -1 ${userpassword}`

id -u $username
if [[ $? ]]
then
    adduser -G wheel -p ${encpassword} ${username}
    homedir=`getent passwd ${username} | cut -d: -f6`
    echo "export PS1='[\u@\H \W]\\$ '" >> ${homedir}/.bashrc
fi
```

Рис. 9: 01-user.sh

01-hostname.sh

И этот файл

vagrant-rocky.pkr.hcl | Vagrantfile | Makefile | 01-dummy.sh | 01-user.sh | 01-hostname.sh | 02-forvard.sh

```
1  #!/bin/bash
2
3  username=nsandryushin
4
5  hostnamectl set-hostname "${HOSTNAME%%.*}".${username}.net
6
7
8
```

Рис. 10: 01-hostname.sh

первый файл

Добавим также 2 файла в папку сервера. Первый файл (рис.

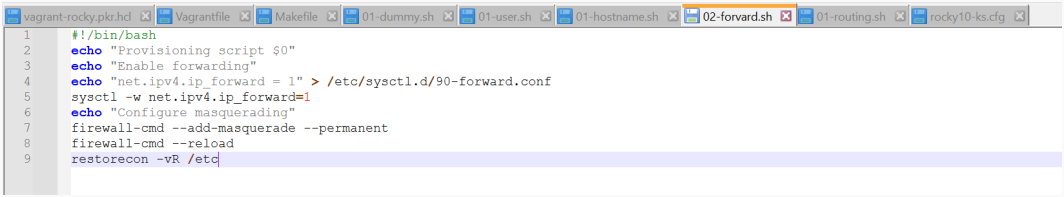


Рис. 11: первый файл

Второй файл

И второй файл

vagrant-rocky.pkr.hcl | Vagrantfile | Makefile | 01-dummy.sh | 01-user.sh | 01-hostname.sh | 02-forvard.sh | 01-routing.sh | rocky10-ks.cfg | 02-forward.sh

```
#!/bin/bash

echo "Provisioning script $0"

echo "Enable forwarding"
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
sysctl -w net.ipv4.ip_forward=1

echo "Configure masquerading"
firewall-cmd --add-masquerade --permanent
firewall-cmd --reload

restorecon -vR /etc
```

Рис. 12: Второй файл

инициализируем packer

Теперь инициализируем packer

```
C:\work\nsandryushin\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/virtualbox v1.1.2 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashic
ndows_amd64.exe"
Installed plugin github.com/hashicorp/qemu v1.1.3 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp/qe
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/mega_/AppData/Roaming/packer.d/plugins/github.com/hashicorp
64.exe"
```

Рис. 13: инициализируем packer

билд образа

И сделаем билд образа

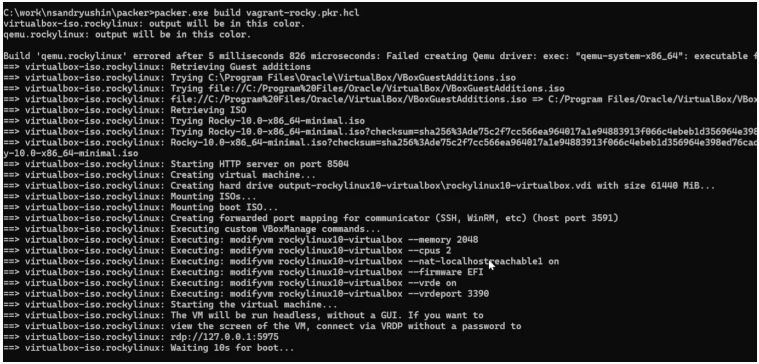


Рис. 14: билд образа

добавление образа в vagrant

После этого добавим его в vagrant

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

Рис. 15: добавление образа в vagrant

запуск сервера

Запустим через вагрант ВМ сервера

```
C:\work\nsandryushin\vagrant>     vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> 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 creation
    server: of new VMs take milliseconds on a modern system.
==> server: Importing base box 'rockylinux10'...
Progress: 60%
```

Рис. 16: запуск сервера

запуск клиента

И запустим еще клиент

```
C:\work\nsandryushin\vagrant>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Cloning VM...
==> client: Matching MAC address for NAT networking...
==> client: Setting the name of the VM: vagrant_client_1757182730334_3384
==> client: Fixed port collision for 22 => 2222. Now on port 2200.
==> client: Clearing any previously set network interfaces...
==> client: Preparing network interfaces based on configuration...
    client: Adapter 1: nat
    client: Adapter 2: intnet
==> client: Forwarding ports...
    client: 22 (guest) => 2200 (host) (adapter 1)
==> client: Running 'pre-boot' VM customizations...
==> client: Booting VM...
```

Рис. 17: запуск клиента

Вход через GUI

Убедимся, что они оба работают, через графический интерфейс. Войдём туда под пользователем vagrant

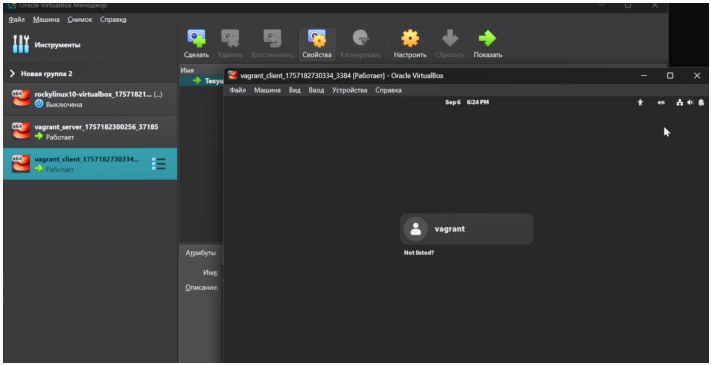


Рис. 18: Вход через GUI

Логин на сервере

Теперь попробуем зайти на сервер через ssh, после чего авторизируемся от имени собственного пользователя, и отключимся

```
C:\work\nsandryushin\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:
Last login: Sat Sep  6 18:26:57 2025
vagrant@server:~$ su - nsandryushin
Password:
[nsandryushin@server.nsandryushin.net ~]$ ^C
[nsandryushin@server.nsandryushin.net ~]$ exit
logout
vagrant@server:~$
```

Рис. 19: Логин на сервере

Логин на клиенте

Сделаем то же самое для клиента

```
C:\work\nsandryushin\vagrant>vagrant ssh client
==> client: The machine you're attempting to SSH into is configured to use
==> client: password-based authentication. Vagrant can't script entering the
==> client: password for you. If you're prompted for a password, please enter
==> client: the same password you have configured in the Vagrantfile.
vagrant@127.0.0.1's password:
vagrant@127.0.0.1's password:
Last failed login: Sat Sep  6 18:51:20 UTC 2025 from 10.0.2.2 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Sat Sep  6 18:30:46 2025 from 10.0.2.2
vagrant@client:~$ su - nsandryushin
Password:
[nsandryushin@client.nsandryushin.net ~]$ exit
logout
vagrant@client:~$
```

Рис. 20: Логин на клиенте

Завершение работы

Выключим обе машины

```
C:\work\nsandryushin\vagrant>vagrant halt server
==> server: Attempting graceful shutdown of VM...

C:\work\nsandryushin\vagrant>vagrant halt client
```

Рис. 21: Завершение работы

Выводы

В результате выполнения лабораторной работы были получены навыки работы с vagrant