

Лабораторная работа

Номер 1

Мошаров Д. М.

01 января 1970

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Мошаров Денис Максимович
- 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

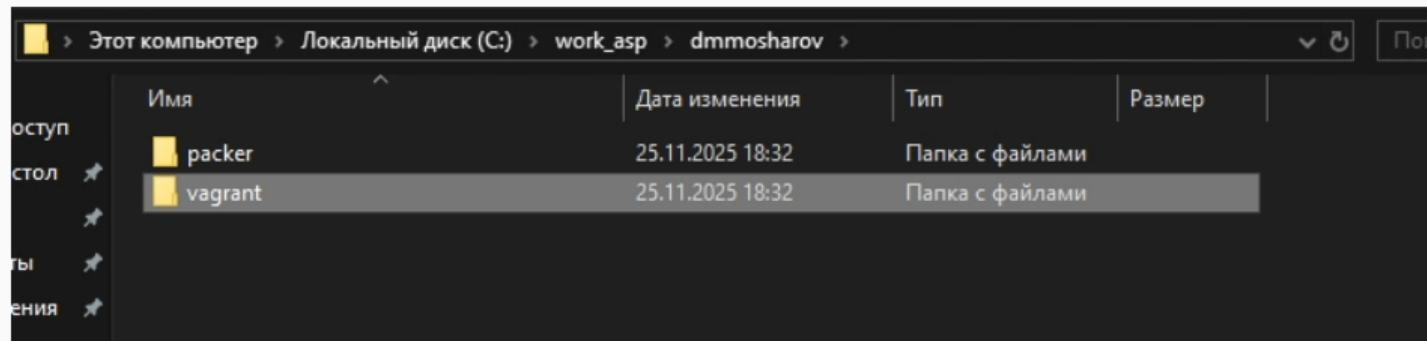


Рис. 1: Создание папок

Образ диска

Поместим заранее скачанный образ в первую папку

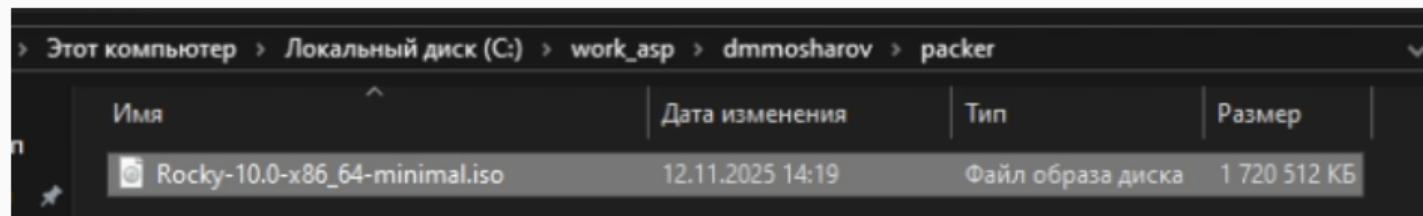


Рис. 2: Образ диска

vagrant-rocky.pkr.hd

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

```
C:\work\asp> dmidecharov> packer> vagrant-rocky.pkr.hd
74 source "qemu" "rockylinux" {
75   ...
76   ...
77   ...
78   ...
79   ...
80   ...
81   ...
82   ...
83   ...
84   ...
85   ...
86   ...
87   ...
88   ...
89   ...
90   ...
91   ...
92   ...
93   ...
94   ...
95   ...
96   ...
97   ...
98   ...
99   ...
100  ...
101  ...
102  ...
103  ...
104  ...
105  ...
106  ...
107  ...
108  ...
109  ...
110  ...
111  ...
112  ...
113  ...
114  ...
115  ...
116  ...
117  ...
118  ...
119  ...
120  ...
121  ...
122  ...
123  ...
124  ...
125  ...
126  ...
127  ...
128  ...
129  ...
130  ...
131  ...
132  ...
133  ...
134  ...
135  ...
136  ...
137  ...
138  ...
139  ...
140  ...}
```

Рис. 3: vagrant-rocky.pkr.hd

Файл конфигурации

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

```
C:\> work_asp> dmeshtools> packit> http> ② rocky10-kvcfg
 39  Apost
 40  # configure swap to a file
 41  # fallcrite -l 2G /swapfile
 42  # mkswap -l /swapfile
 43  # swap /swapfile
 44  # echo "/swapfile none swap defaults 0 0" >> /etc/fstab
 45
 46  # sudo
 47  echo "Vagrant ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.d/vagrant
 48  chmod 0400 /etc/sudoers.d/vagrant
 49
 50  # Fix for https://github.com/centos/sig-cloud-instance-build/issues/38
 51  cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
 52  DEVICE="eth0"
 53  NM_CONTROLLED="no"
 54  BOOTPROTO="yes"
 55  TYPE="Ethernet"
 56  PERSISTENT_DHCLIENT="yes"
 57  EOF
 58
 59  # used: disable password authentication and DNS checks
 60  # See ~ /etc/ssh/sshd_config <EOF
 61  #:PasswordAuthentication no
 62  #:Substitute "/%{UseDNS} yes%/{UseDNS} no"
 63  #:Update
 64  #:X11Forwarding
 65  #:AllowAgentForwarding
 66  #:RealName >> /etc/sysconfig/sshd <EOF
 67
 68  # Decrease connection time by preventing reverse DNS lookups
 69  # (see https://lists.centos.org/pipermail/centos-devel/2016-July/016081.html)
 70  # and man sshd for more information)
 71  OPTIONS="-u"
 72  EOF
 73
 74  # Fix for issue #70, regular users can gain admin privileges via su
 75  # See ~ /etc/pam.d/su << EOF
 76  # allow vagrant to use su, but prevent others from becoming root or vagrant
 77  #/*account\:(sufficient\|)\(pam_succeed_if\|so\|wid<0\|use_uid\|quiet\|*/
 78  #append
 79  account      [success=1 default=ignore] pam_succeed_if.so user != vagrant use uid quiet
 80  account      required      pam_succeed_if.so user != vagrant use uid root:vagrant
 81
 82  update
 83  quit
 84
 85  EOF
```

Рис. 4: Файл конфигурации

Vagrantfile

Теперь добавим vagrantfile в папку vagrant

```
C:\> vagrantfile X
C:\> wakewebapp> denchanbox> vagrant > ● Vagrantfile
1   | # -*- mode: ruby -*-
2   | # $ vim set Pmenu!
3   |
4   | Vagrant.configure("2") do |config|
5   |
6   |   config.vagrant.plugins = "vagrant-libvirt"
7   |   config.vagrant.plugins = "vagrant-vguest"
8   |
9   |   config.vm.provider :virtualbox do |virtualbox|
10  |     virtualbox.linked_clone = true
11  |     # Customize the amount of memory on the VM
12  |     virtualbox.memory = 2048
13  |     virtualbox.cpus = 2
14  |     # Set the VirtualBox GUI when booting the machine
15  |     virtualbox.gui = false
16  |     # Set the video memory to 1024
17  |     virtualbox.customize [ "modifyvm", { id: "-vram", "1024" } ]
18  |     virtualbox.customize [ "modifyvm", { id: "-audiohwresolver1", "on" } ]
19  |     virtualbox.customize [ "modifyvm", { id: "-clipboard", "bidirectional" } ]
20  |     virtualbox.customize [ "modifyvm", { id: "-draganddrop", "on" } ]
21  |     virtualbox.customize [ "modifyvm", { id: "-graphicscontroller", "vga" } ]
22  |     virtualbox.customize [ "modifyvm", { id: "-accelerated3d", "on" } ]
23  |     virtualbox.customize [ "modifyvm", { id: "-nested-hw-virt", "on" } ]
24   end
25
26   config.vm.provider :libvirt do |libvirt|
27     libvirt.driver = "kvm"
28     libvirt.memory = 2048
29     libvirt.cpu = 2
30     libvirt.vcpu_type = "Intel"
31     libvirt.sandbox_type = "virtio"
32     libvirt.nic_model_type = "virtio"
33     libvirt.management_network_name = "vagrant-libvirt"
34     libvirt.management_network_address = "192.168.121.8/24"
35     libvirt.storage_pool_name = "vagrant"
36     libvirt.storage_pool_name = "default"
37   end
38
39   ## Comse configuration
40   config.vm.provision "comse_dammy",
41     type: "shell",
42     preserve_order: true,
43     path: "provision/default/01-dammy.sh"
44
45   config.vm.provision "comse_hostname",
46     type: "shell",
47     preserve_order: true,
48
```

Рис. 5: Vagrantfile

Каталог Vagrant

Содержимое папки будет выглядеть так. Теперь создадим тут папку provision

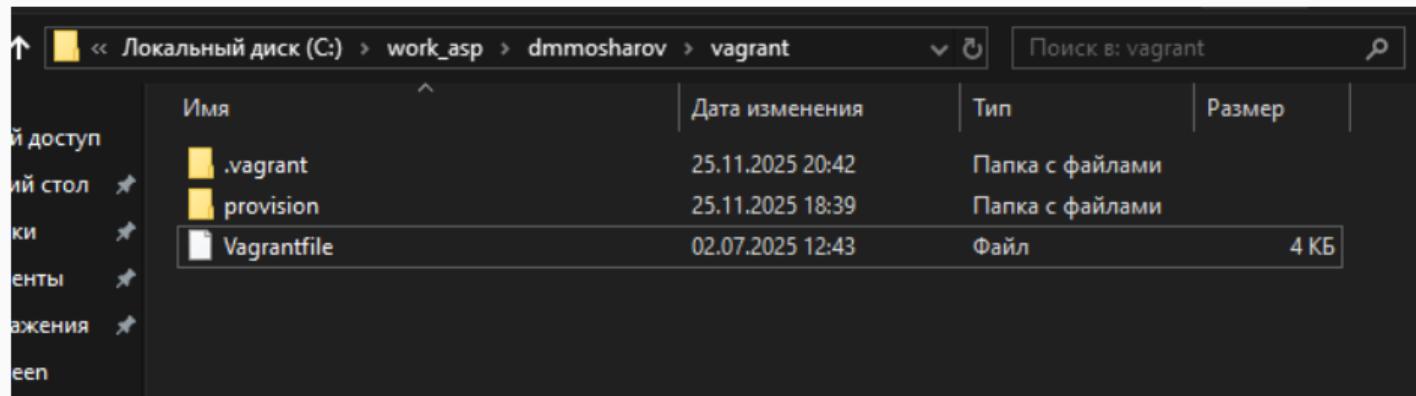


Рис. 6: Каталог Vagrant

Создание подпапок

В этой папке мы создадим 3 подпапки, показаные на фото

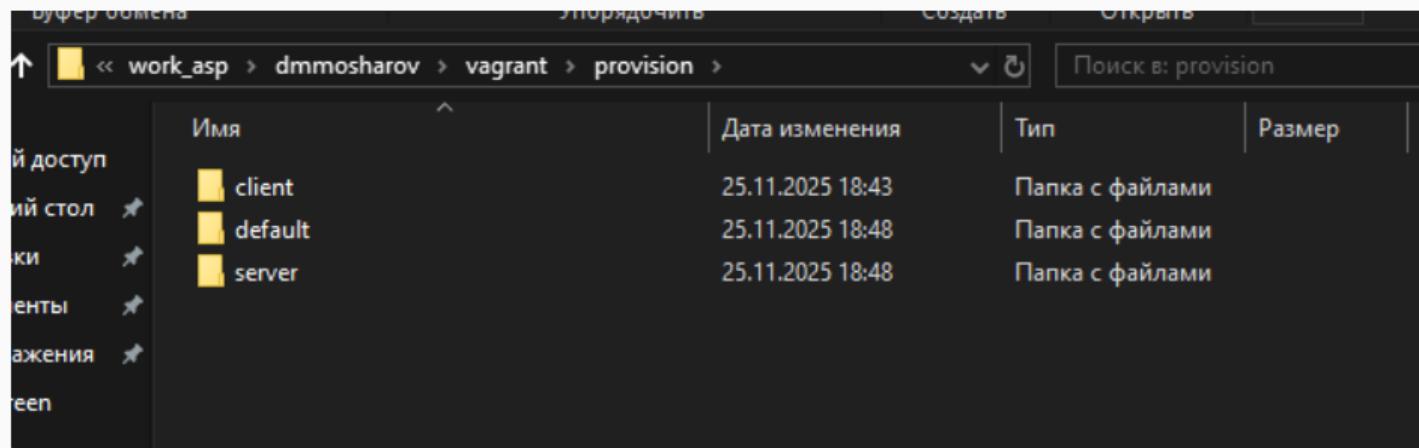


Рис. 7: Создание подпапок

Файл заглушка

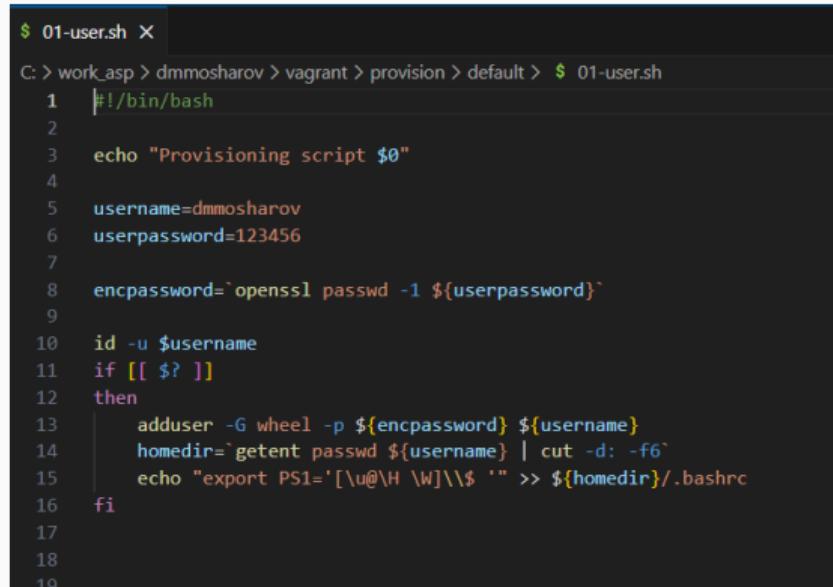
В каждую папку добавим файл заглушки со следующим содержанием

```
C: > work_asp > dmmosharov > vagrant > provision > client > $ 01-dummy.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5
```

Рис. 8: Файл заглушки

01-user.sh

В папку default добавим следующий файл



```
$ 01-user.sh X
C: > work_asp > dmmosharov > vagrant > provision > default > $ 01-user.sh
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 username=dmmosharov
6 userpassword=123456
7
8 encpassword=`openssl passwd -1 ${userpassword}`
9
10 id -u $username
11 if [[ $? ]]
12 then
13     adduser -G wheel -p ${encpassword} ${username}
14     homedir=`getent passwd ${username} | cut -d: -f6`
15     echo "export PS1='[\u@\H \W]\$\n'" >> ${homedir}/.bashrc
16 fi
17
18
19
```

Рис. 9: 01-user.sh

01-hostname.sh

И этот файл

```
C: > work_asp > dmmosharov > vagrant > provision > default > $ 01-hostname.sh
1  #!/bin/bash
2
3  username=dmmosharov
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.

```
C: > work_asp > dmmosharov > vagrant > provision > server > $ 02-forward.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  echo "Enable forwarding"
6  echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
7  sysctl -w net.ipv4.ip_forward=1
8
9  echo "Configure masquerading"
10 firewall-cmd --add-masquerade --permanent
11 firewall-cmd --reload
12
13 restorecon -vR /etc
14
```

Рис. 11: первый файл

Второй файл

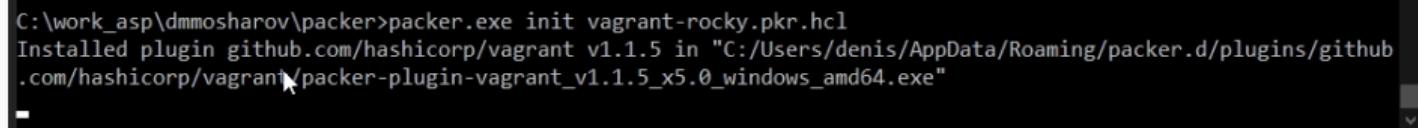
И второй файл

```
C: > work_asp > dmmosharov > vagrant > provision > server > $ 01-dummy.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5
```

Рис. 12: Второй файл

инициализируем packer

Теперь инициализируем packer



```
C:\work_asp\dmmosharov\packer>packer.exe init vagrant-rocky.pkr.hcl
Installed plugin github.com/hashicorp/vagrant v1.1.5 in "C:/Users/denis/AppData/Roaming/packer.d/plugins/github.com/hashicorp/vagrant\packer-plugin-vagrant_v1.1.5_x5.0_windows_amd64.exe"
```

Рис. 13: инициализируем packer

билд образа

И сделаем билд образа

```
C:\work_asp\dmmosharov\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 8 milliseconds 784 microseconds: Failed creating Qemu driver: exec: "qemu-system-x
86_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/Or
acle/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%3Ade75c2f7cc566ea964017a1e94883913f0
66c4ebbeb1d356964e398ed76cadd12
```

Рис. 14: билд образа

добавление образа в vagrant

После этого добавим его в vagrant

```
c:\work_asp\dmmosharov\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_asp/dmmosharov/packer/vagrant-virtualbox-rockylinux10-x86_64.box
  box:
```

Рис. 15: добавление образа в vagrant

запуск сервера

Запустим через вагрант ВМ сервера

```
C:\work_asp\dmmosharov\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: Box 'rockylinux10' could not be found. Attempting to find and install...
    server: Box Provider: virtualbox
    server: Box Version: >= 0
==> server: Box file was not detected as metadata. Adding it directly...
==> server: Adding box 'rockylinux10' (v0) for provider: virtualbox (amd64)
    server: Downloading rockylinu...  
→
```

Рис. 16: запуск сервера

запуск клиента

И запустим еще клиент

```
C:\work_asp\dmmosharov\vagrant\provision>vagrant up client
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Clearing any previously set forwarded ports...
==> 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
```

Рис. 17: запуск клиента

Вход через GUI

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

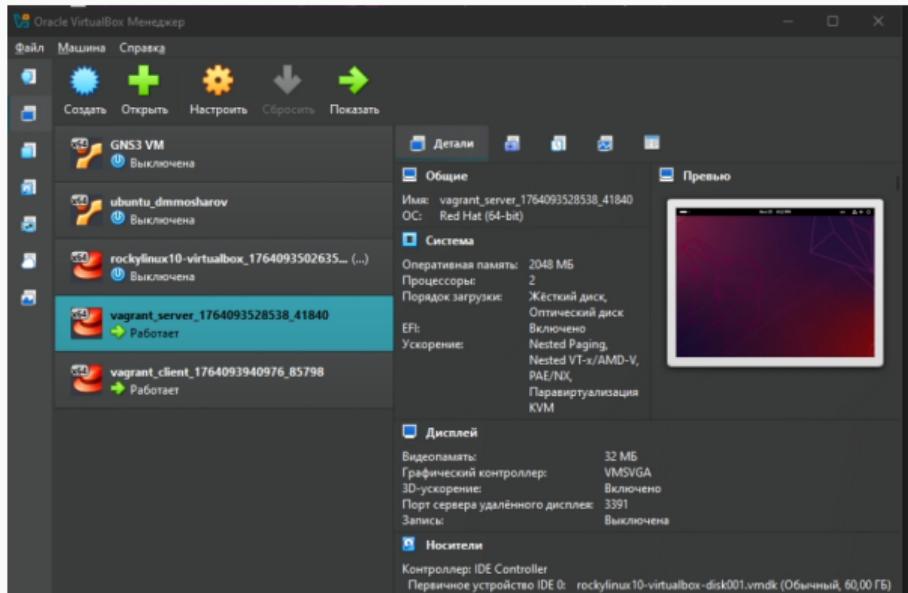


Рис. 18: Вход через GUI

Логин на сервере

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

```
Last login: Tue Nov 25 18:05:10 2025
vagrant@server:~$ su - dmmosharov
Password:
[dmmosharov@server.dmmosharov.net ~]$ logout
vagrant@server:~$ logout
[1]+ 15 SIGHUP                          su
```

Рис. 19: Логин на сервере

Логин на клиенте

Сделаем то же самое для клиента

```
vagrant@client:~$ su - dmmosharov
2. Password:
Last failed login: Tue Nov 25 19:14:32 U
There was 1 failed login attempt since t
/0dmmosharov@client:~$ sudo -i
```

Рис. 20: Логин на клиенте

Завершение работы

Выключим обе машины

```
C:\work_asp\dmmosharov\vagrant>vagrant halt server  
==> server: Attempting graceful shutdown of VM...  
  
C:\work_asp\dmmosharov\vagrant>vagrant halt client  
==> client: Attempting graceful shutdown of VM...
```

Рис. 21: Завершение работы

Выводы

В результате выполнения лабораторной работы были получены навыки работы с vagrant