

# **Лабораторная работа №1**

**Подготовка лабораторного стенда**

Беличева Дарья Михайловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>16</b>

## Список иллюстраций

3.1	Содержимое каталога packer . . . . .	6
3.2	Содержимое подкаталога http . . . . .	7
3.3	Содержимое каталога provision . . . . .	7
3.4	Содержимое файла 01-dummy.sh . . . . .	7
3.5	Содержимое файла 01-user.sh . . . . .	8
3.6	Содержимое файла 01-hostname.sh . . . . .	8
3.7	Появление box-файла . . . . .	9
3.8	Команда make addbox . . . . .	10
3.9	Команда make server-up . . . . .	10
3.10	Команда make client-up . . . . .	11
3.11	Окно server . . . . .	11
3.12	Окно client . . . . .	12
3.13	Команда make server-provision . . . . .	13
3.14	Команда make client-provision . . . . .	13
3.15	Терминал client . . . . .	14
3.16	Терминал server . . . . .	14

# 1 Цель работы

Приобрести практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

## 2 Задание

1. Сформируйте box-файл с дистрибутивом Rocky Linux для VirtualBox.
2. Запустите виртуальные машины сервера и клиента и убедитесь в их работоспособности.
3. Внесите изменения в настройки загрузки образов виртуальных машин server и client, добавив пользователя с правами администратора и изменив названия хостов.

### 3 Выполнение лабораторной работы

1. В ОС Windows создадим каталог для проекта.

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux (в этом практикуме будем использовать Rocky-9.2- x86\_64-minimal.iso — минимальный дистрибутив Rocky Linux).

В этом же каталоге разместим подготовленные заранее для работы с Vagrant файлы (рис. 3.1):

- vagrant-rocky.pkr.hcl
- ks.cfg (файл должен быть расположен в подкаталоге http)
- Vagrantfile
- Makefile

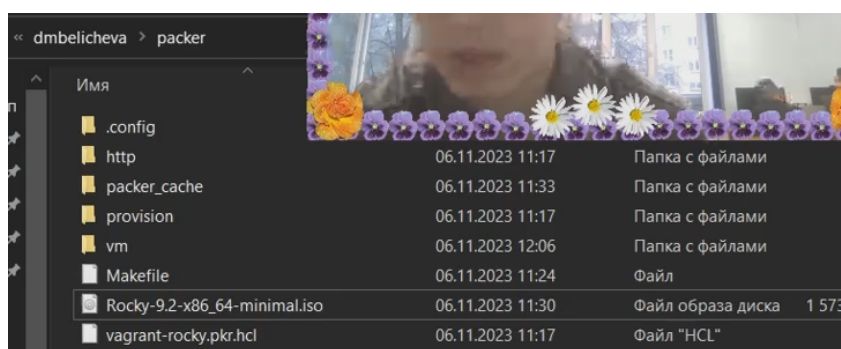


Рис. 3.1: Содержимое каталога packer

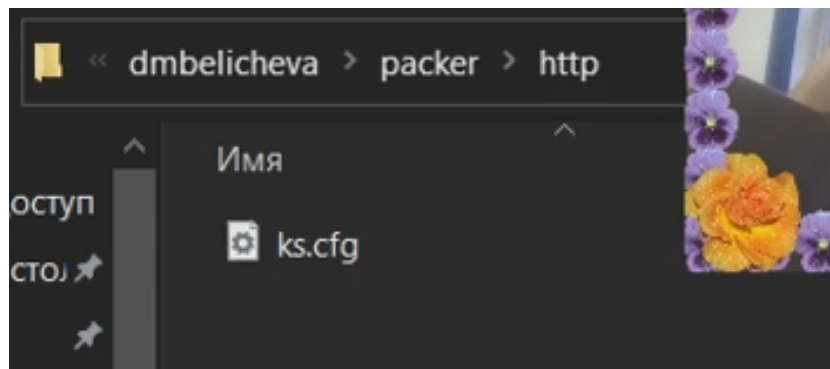


Рис. 3.2: Содержимое подкаталога http

1. В этом же каталоге 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.

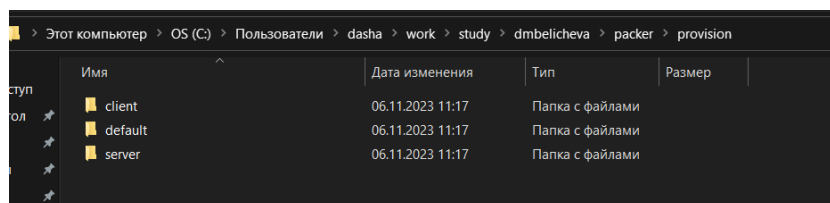


Рис. 3.3: Содержимое каталога provision

1. В каталогах default, server и client разместим заранее подготовленный скрипт заглушку 01-dummy.sh:

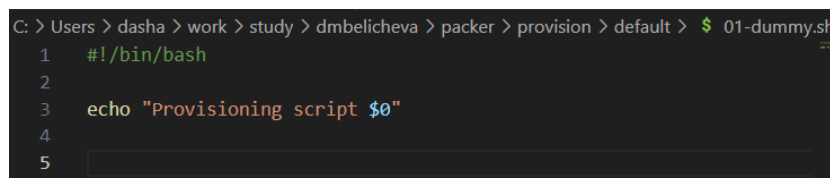


Рис. 3.4: Содержимое файла 01-dummy.sh

1. В каталоге default разместим заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины:

```
C: > Users > dasha > work > study > dmbelicheva > packer > provision > default > $ 01-user.sh
1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  username=dmbelicheva
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 ${username} | cut -d: -f6`
15     echo "export PS1='\u@\H \W]\\$ '" >> ${homedir}/.bashrc
16 fi
17
```

Рис. 3.5: Содержимое файла 01-user.sh

В этом скрипте в качестве значения переменной username вместо user укажем имя пользователя, совпадающее с моим логином, т.е. dmbelicheva.

1. В каталоге default разместите заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины:

```
C: > Users > dasha > work > study > dmbelicheva > packer > provision > default > $ 01-hostname.sh
1  #!/bin/bash
2
3  username=dmbelicheva
4
5  hostnamectl set-hostname "${HOSTNAME%%.*}.${username}.net"
6
7
8
```

Рис. 3.6: Содержимое файла 01-hostname.sh

В этом скрипте в качестве значения переменной username вместо user укажем имя пользователя, совпадающее с вашим логином, т.е. dmbelicheva.

## Развёртывание лабораторного стенда на ОС Linux



Я установила MSYS2 (сборка пакетов для Windows, которая позволяет использовать многие утилиты и приложения, которые обычно доступны только в Unix-подобных операционных системах), поэтому буду использовать команды для Linux.

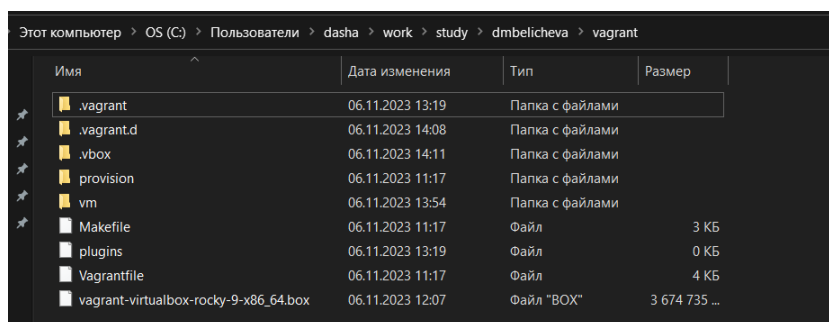
1. Перейдем в каталог с проектом:

```
cd C:\Users\dasha\work\study\dmbelicheva\packer\
```

2. Для формирования box-файла с дистрибутивом Rocky Linux для VirtualBox в терминале наберем:

```
make box
```

Начнётся процесс скачивания, распаковки и установки драйверов VirtualBox и дистрибутива ОС на виртуальную машину. После завершения процесса автоматического развёртывания образа виртуальной машины в каталоге C:\Users\dasha\work\study\dmbelicheva\vagrant\ временно появится каталог builds с промежуточными файлами .vdi, .vmdk и .ovf, которые затем автоматически будут преобразованы в box-файл сформированного образа: vagrant-virtualbox-rocky9-x86\_64.box.



Имя	Дата изменения	Тип	Размер
.vagrant	06.11.2023 13:19	Папка с файлами	
.vagrant.d	06.11.2023 14:08	Папка с файлами	
.vbox	06.11.2023 14:11	Папка с файлами	
provision	06.11.2023 11:17	Папка с файлами	
vm	06.11.2023 13:54	Папка с файлами	
Makefile	06.11.2023 11:17	Файл	3 КБ
plugins	06.11.2023 13:19	Файл	0 КБ
Vagrantfile	06.11.2023 11:17	Файл	4 КБ
vagrant-virtualbox-rocky-9-x86_64.box	06.11.2023 12:07	Файл "BOX"	3 674 735 ...

Рис. 3.7: Появление box-файла

1. Для регистрации образа виртуальной машины в Vagrant в терминале в каталоге C:\Users\dasha\work\study\dmbelicheva\vagrant\ наберем

make addbox

```
$ make addbox
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rocky9' (v0) for provider:
==> box: Unpacking necessary files from: file:///C:/Users/dasha/work/study/dmbelicheva/vagrant/vagrant-virtualbox-rocky-9-x86_64.box
box:
--> box: Successfully added box 'rocky9' (v0) for ''!
```

Рис. 3.8: Команда make addbox

Это позволит на основе конфигурации, прописанной в файле Vagrantfile, сформировать box-файлы образов двух виртуальных машин - сервера и клиента с возможностью их параллельной или индивидуальной работы.

1. Запустим виртуальную машину Server, введя

make server-up

```
dasha@DESKTOP-0358DT0 MINGW64 /c/users/dasha/work/study/dmbelicheva/vagrant
$ make server-up
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Fetching micromachine-3.0.0.gem
Fetching vagrant-vbguest-0.31.0.gem
Installed the plugin 'vagrant-vbguest (0.31.0)'!
touch plugins
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".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 'rocky9'...
```

Рис. 3.9: Команда make server-up

1. Запустим виртуальную машину Client, введя

make client-up

```
dasha@DESKTOP-0358D10 MINGW64 /c/Users/dasha/work/study/dmbelicheva/vagrant
$ make client-up
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Cloning VM...
==> client: Matching MAC address for NAT networking...
==> client: Setting the name of the VM: client
==> 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. 3.10: Команда make client-up

1. Убедимся, что запуск обеих виртуальных машин прошёл успешно, залогинимся под пользователем vagrant с паролем vagrant. Затем выключим обе виртуальные машины.

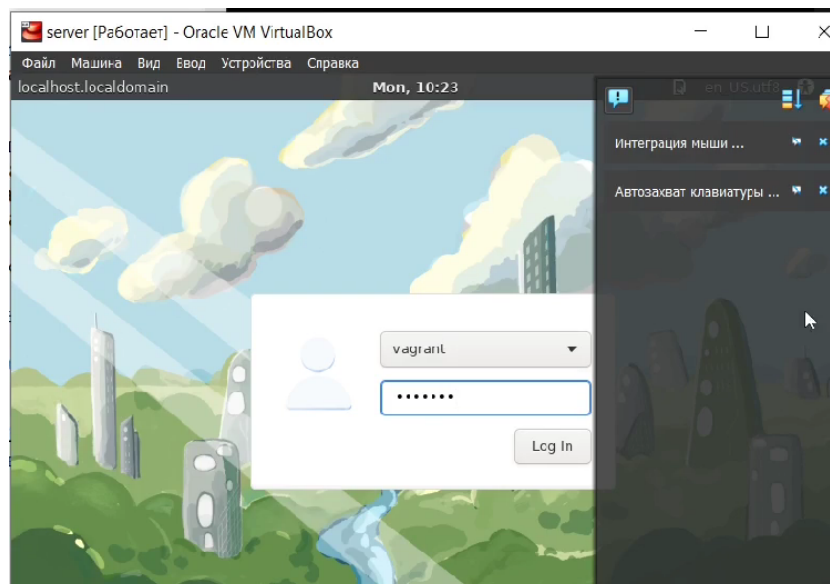


Рис. 3.11: Окно server

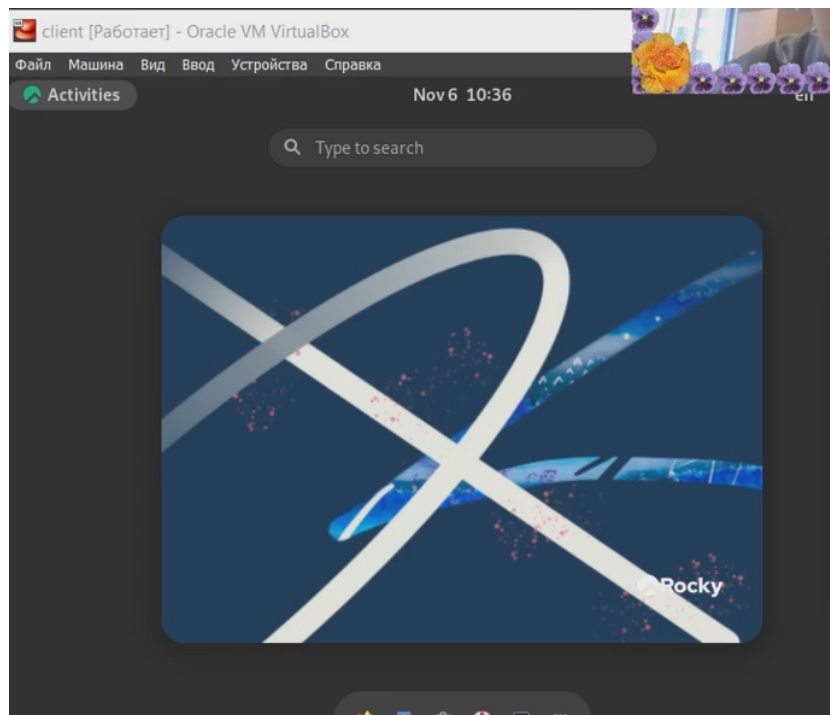


Рис. 3.12: Окно client

### **Внесение изменений в настройки внутреннего окружения виртуальной машины**

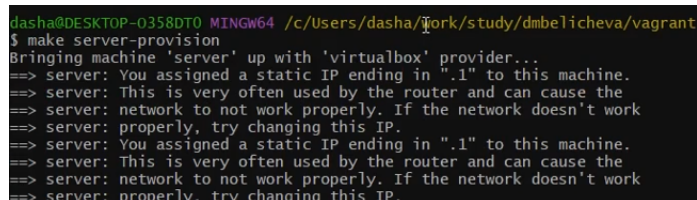
1. Для отработки созданных скриптов во время загрузки виртуальных машин убедимся, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется следующая запись:

```
# 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"
```

1. Зафиксируем внесённые изменения для внутренних настроек виртуальных машин, введя в терминале:

```
make server-provision
```

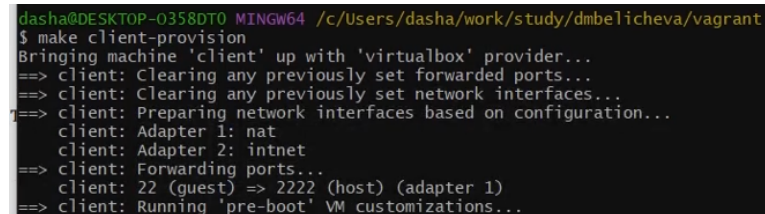


```
dasha@DESKTOP-0358DT0 MINGW64 /c/Users/dasha/work/study/dmbelicheva/vagrant  
$ make server-provision  
Bringing machine 'server' up with 'virtualbox' provider...  
==> server: You assigned a static IP ending in ".1" to this machine.  
==> server: This is very often used by the router and can cause the  
==> server: network to not work properly. If the network doesn't work  
==> server: properly, try changing this IP.  
==> server: You assigned a static IP ending in ".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.
```

Рис. 3.13: Команда make server-provision

Затем

```
make client-provision
```



```
dasha@DESKTOP-0358DT0 MINGW64 /c/Users/dasha/work/study/dmbelicheva/vagrant  
$ make client-provision  
Bringing machine 'client' up with 'virtualbox' provider...  
==> client: Clearing any previously set forwarded ports...  
==> 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) => 2222 (host) (adapter 1)  
==> client: Running 'pre-boot' VM customizations...
```

Рис. 3.14: Команда make client-provision

1. Залогинемся на сервере и клиенте под созданным пользователем. Убедимся, что в терминале приглашение отображается в виде user@server.user.net на сервере и в виде user@client.user.net на клиенте, где вместо user указан мой логин - dmbelicheva.

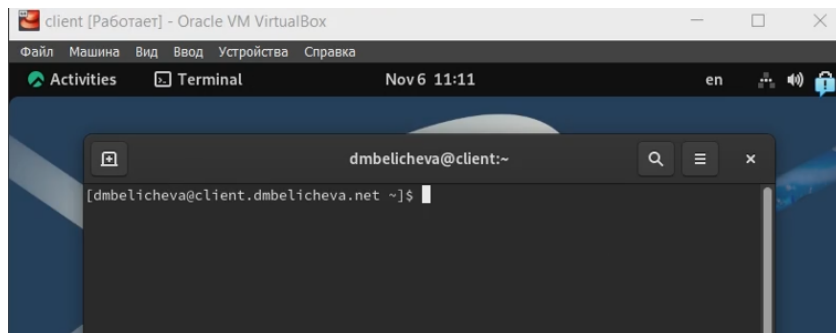


Рис. 3.15: Терминал client

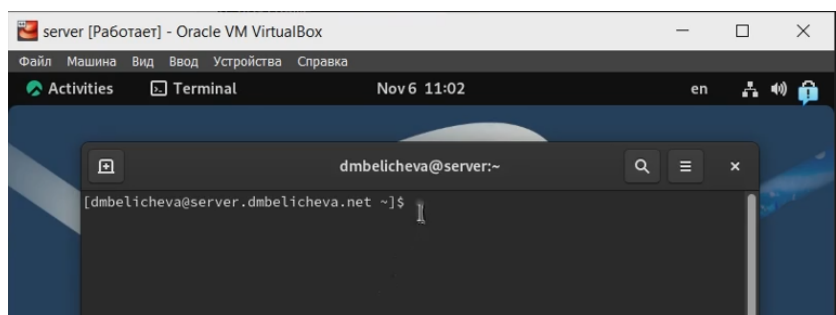


Рис. 3.16: Терминал server

4. Выключим виртуальные машины.

## 4 Выводы

В процессе выполнения данной лабораторной я приобрела практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

## 5 Контрольные вопросы

1. Для чего предназначен Vagrant? Инструмент для создания и управления средами виртуальных машин в одном рабочем 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 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`.

Пример содержимого файла `Vagrantfile`:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure(2) do |config|
  config.vm.box = "BOX_NAME"
  config.vm.hostname = "HOST_NAME"
  config.vm.network "private_network", ip: "192.168.1.1"
  config.vm.define "VM_NAME"
  config.vm.provider "virtualbox" do |vb|
    vb.gui = false
    vb.memory = "1024"
  end
end
```

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`. Строка `config.vm.box = "BOX_NAME"` задаёт название образа (`box`-файла) виртуальной машины (обычно выбирается из официального репозитория). Строка `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.

Строка `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.