

# Лабораторная работа №1

Дисциплина: Администрирование сетевых подсистем

---

Андреева С.В.

Группа НПИбд-01-23

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Андреева Софья Владимировна
- Группа НПИбд-01-23
- Российский университет дружбы народов
- Ссылка на репозиторий GitHub

# Вводная часть

---

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

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.

## Выполнение лабораторной работы

В ОС Windows создадим каталог для проекта.

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux (в этом практикуме будем использовать Rocky-10.2-x86\_64-minimal.iso — минимальный дистрибутив Rocky Linux).

В этом же каталоге разместим подготовленные заранее для работы с Vagrant файлы (рис. (**fig:001?**)):

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

В этом же каталоге создадим каталог provision с подкаталогами default

# Выполнение лабораторной работы

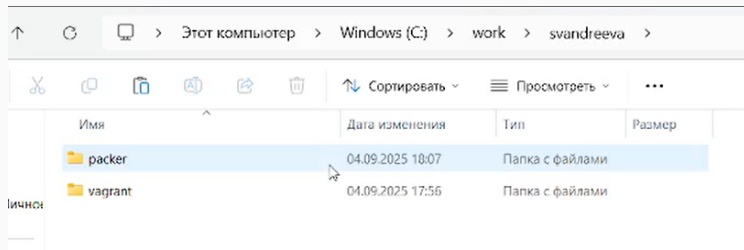


Рис. 1: Выполнение работы



# Выполнение лабораторной работы

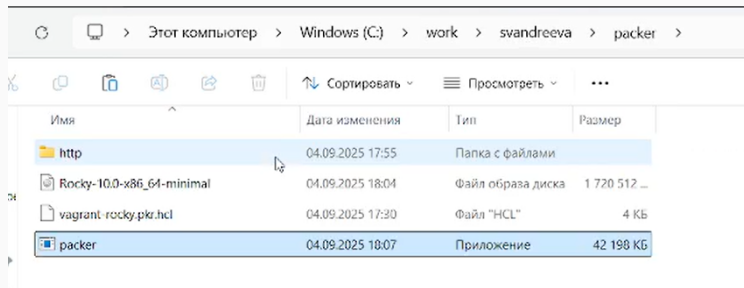
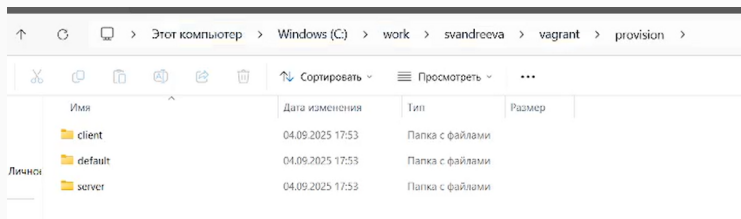


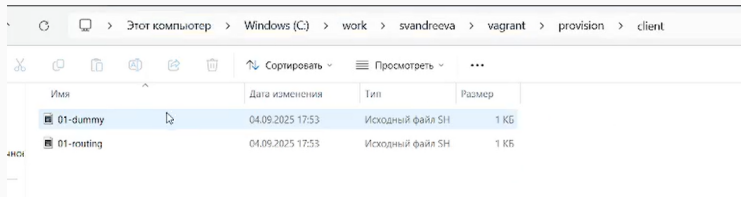
Рис. 2: Выполнение работы

# Выполнение лабораторной работы



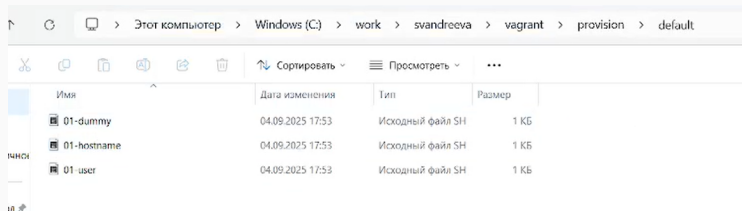
**Рис. 3:** Выполнение работы

# Выполнение лабораторной работы



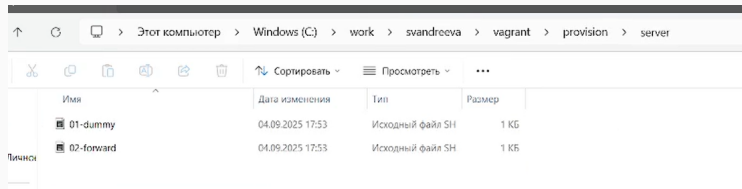
**Рис. 4:** Выполнение работы

# Выполнение лабораторной работы



**Рис. 5:** Выполнение работы

# Выполнение лабораторной работы



**Рис. 6:** Выполнение работы

## Развёртывание лабораторного стенда на ОС Linux

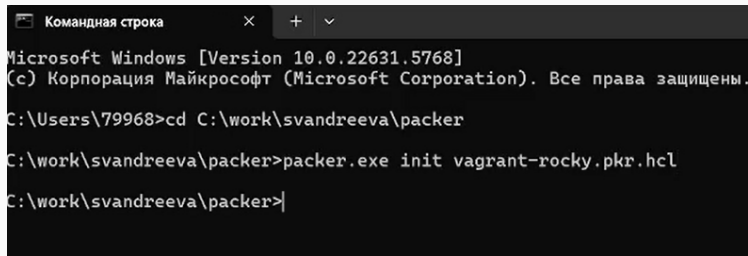
Я установила MSYS2 (сборка пакетов для Windows, которая позволяет использовать многие утилиты и приложения, которые обычно доступны только в Unix-подобных операционных системах), поэтому буду использовать команды для Linux.

Перейдем в каталог с проектом:

```
cd C:\work\study\svandreeva\packer\
```

В командной строке введем: `”” packer.exe init vagrant-rocky.pkr.hcl ””`

# Выполнение лабораторной работы



```
Командная строка
Microsoft Windows [Version 10.0.22631.5768]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\79968>cd C:\work\svandreeva\packer

C:\work\svandreeva\packer>packer.exe init vagrant-rocky.pkr.hcl

C:\work\svandreeva\packer>
```

Рис. 7: Выполнение работы

Для формирования box-файла с дистрибутивом Rocky Linux для VirtualBox в терминале наберем: `''' packer.exe build vagrant-rocky.pkr.hcl '''` 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:/work/svandreeva/vagrant'` временно появится каталог `builds` с промежуточными файлами `.vdi`, `.vmdk` и `.ovf`, которые затем автоматически будут преобразованы в box-файл сформированного образа: `vagrant-virtualbox-rocky10-x86_64.box`.



# Выполнение лабораторной работы

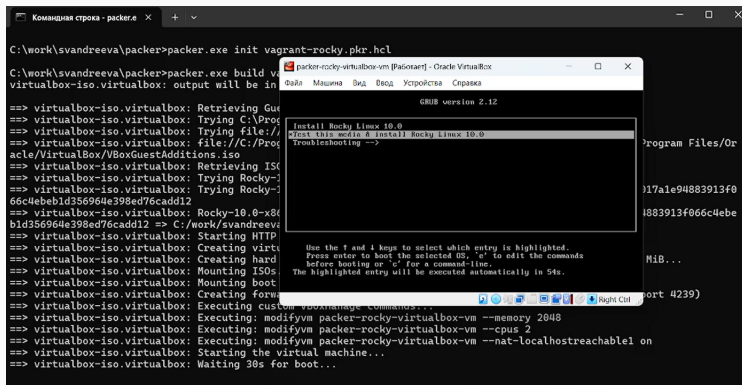


Рис. 8: Выполнение работы

# Выполнение лабораторной работы

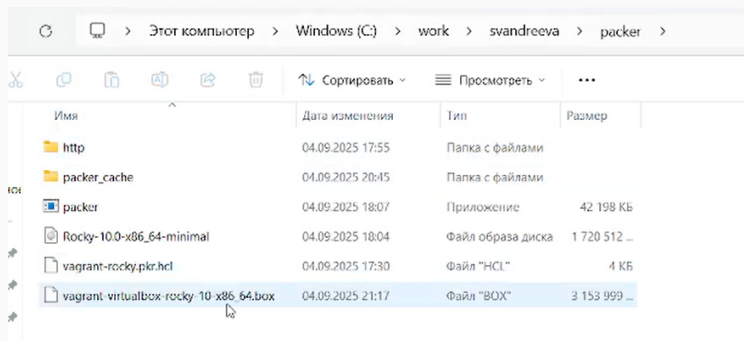


Рис. 9: Выполнение работы

Для регистрации образа виртуальной машины в Vagrant в терминале в каталоге 'C:/work/svandreeva/vagrant' наберем

```
vagrant box add rocky10 vagrant-virtualbox-rocky-10-x86_64.box
```

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.

# Выполнение лабораторной работы

```
C:\work\svandreeva\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: Unpacking necessary files from: file:///C:/work/svandreeva/packer/vagrant-virtualbox-rocky-10-x86_64
      .box
      box:
==> box: Successfully added box 'rocky10' (v0) for '(amd64)'!
C:\work\svandreeva\packer>
```

Рис. 10: Выполнение работы

Запустим виртуальную машину Server, введя

```
vagrant up server
```

Запустим виртуальную машину Client, введя

```
vagrant up client
```

# Выполнение лабораторной работы

```
C:\work\svandreeva\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 'rocky10'...
==> server: Cloning VM...
==> server: Matching MAC address for NAT networking...
==> 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: Setting the name of the VM: server
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:

VAGRANT_DISABLE_VBOXSYMLINKCREATE=1

or on a per folder basis within the Vagrantfile:

config.vm.synced_folder '/host/path', '/guest/path', SharedFoldersEnableSymlinksCreate: false
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
server: Adapter 1: nat
server: Adapter 2: intnet
```

Рис. 11: Выполнение работы

## Выполнение лабораторной работы

```

server: the guest additions and repackage the box to continue.
server:
server: This is not an error message; everything
server: in which case you may ignore this message.
==> server: Setting hostname...
==> server: Configuring and enabling network interface.
==> server: Mounting shared folders...
server: C:\work\svandreeva\vagrant> vagrant
server: Running provisioner: common dummy (sh)
server: Running: C:\Users\79968\AppData\Local\
server: Provisioning script /tmp/vagrant-shell
server: Running provisioner: common hostname
server: Running: C:\Users\79968\AppData\Local\
server: Running provisioner: common user (sh)
server: Running: C:\Users\79968\AppData\Local\
server: Provisioning script /tmp/vagrant-shell
server: id: 'svivanov': no such user
server: Running provisioner: server dummy (sh)
server: Running: C:\Users\79968\AppData\Local\
server: Provisioning script /tmp/vagrant-shell

C:\work\svandreeva\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: client
==> client: Fixed port collision for 22 => 2222.
==> 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...
==> client: Waiting for machine to boot. This may take

```

**Рис. 12: Выполнение работы**

Убедимся, что запуск обеих виртуальных машин прошёл успешно, залогинемся под пользователем `vagrant` с паролем `vagrant`. Подключимся к серверу из консоли: `''' vagrant ssh server '''` Затем выключим обе виртуальные машины.



# Выполнение лабораторной работы

```
C:\work\svandreeva\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:
vagrant@127.0.0.1's password:
vagrant@127.0.0.1's password:
Last failed login: Thu Sep  4 19:33:52 UTC 2025 from 10.0.2.2 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Thu Sep  4 19:28:16 2025
vagrant@server:~$ su - svivanov
Password:
[svivanov@server.svivanov.net ~]$ logout
vagrant@server:~$ su - client
su: user client does not exist or the user entry does not contain all the required fields
vagrant@server:~$ cd C:\work\svandreeva\vagrant
-bash: cd: C:\worksvandreevavagrant: Нет такого файла или каталога
vagrant@server:~$ logout

C:\work\svandreeva\vagrant>client ssh server
"client" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\work\svandreeva\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: Thu Sep  4 19:40:21 UTC 2025 from 10.0.2.2 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu Sep  4 18:13:10 2025 from 10.0.2.2
vagrant@client:~$ su - svivanov
Password:
[svivanov@client.svivanov.net ~]$ logout
vagrant@client:~$ logout

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

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

---

## Внесение изменений в настройки внутреннего окружения виртуальной машины

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"
```

## Выполнение лабораторной работы

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

```
vagrant up server --provision
```

Затем

```
vagrant up client --provision
```

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. Выключим виртуальные машины.

# Выполнение лабораторной работы

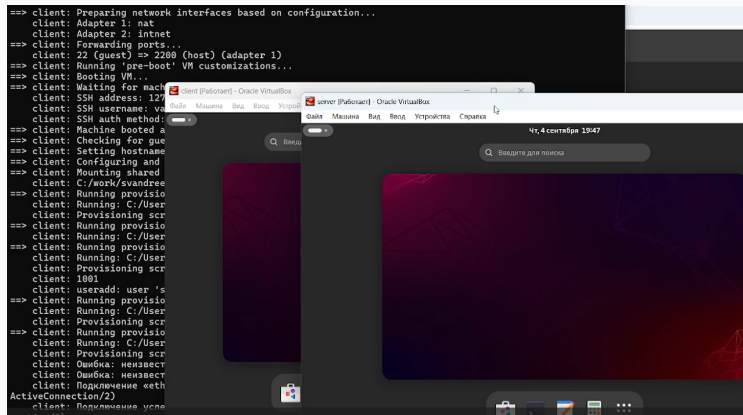


Рис. 14: Выполнение работы

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