

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра теории  
вероятностей и кибербезопасности

## ОТЧЁТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

дисциплина: Администрирование сетевых подсистем

Студент: Пакавира Арсениу Висенте Луиш

Студ. билет № 1032225105

Группа: НФИбд-02-23

## МОСКВА

2024 г.

### Цель работы:

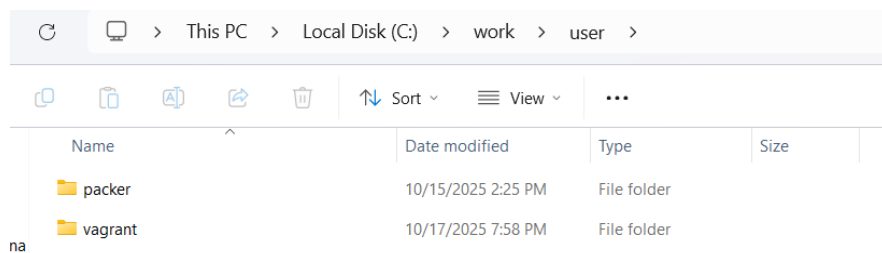
Целью данной работы является приобретение практических навыков установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

### Выполнение работы:

Перед началом работы с Vagrant создадим каталог для проекта (Рис. 1.1):

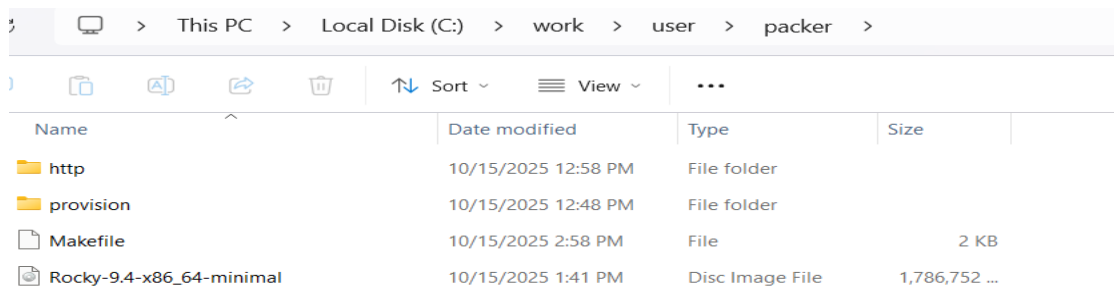
**C:\work\user\packer** и

**C:\work\user\vagrant**



**Рис. 1.1.** Создание каталога для проекта.

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux, в этом практикуме используем Rocky-9.2-x86\_64minimal.iso (Рис. 1.3). 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. 1.2.** 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. 2.1).

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

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

  ## 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"

  config.vm.provision "common user",
    type: "shell",
    preserve_order: true,
    path: "provision/default/01-user.sh"

  ## Server configuration
  config.vm.define "server", autostart: false do |server|
    server.vm.box = "rocky9"
    server.vm.hostname = 'server'

    server.vm.boot_timeout = 1440
  end
end
```

**Рис. 2.1.** Проверка конфигурационного файла Vagrantfile.

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

make server-provision (Рис. 2.2)

Затем

make client-provision (Рис. 2.3)

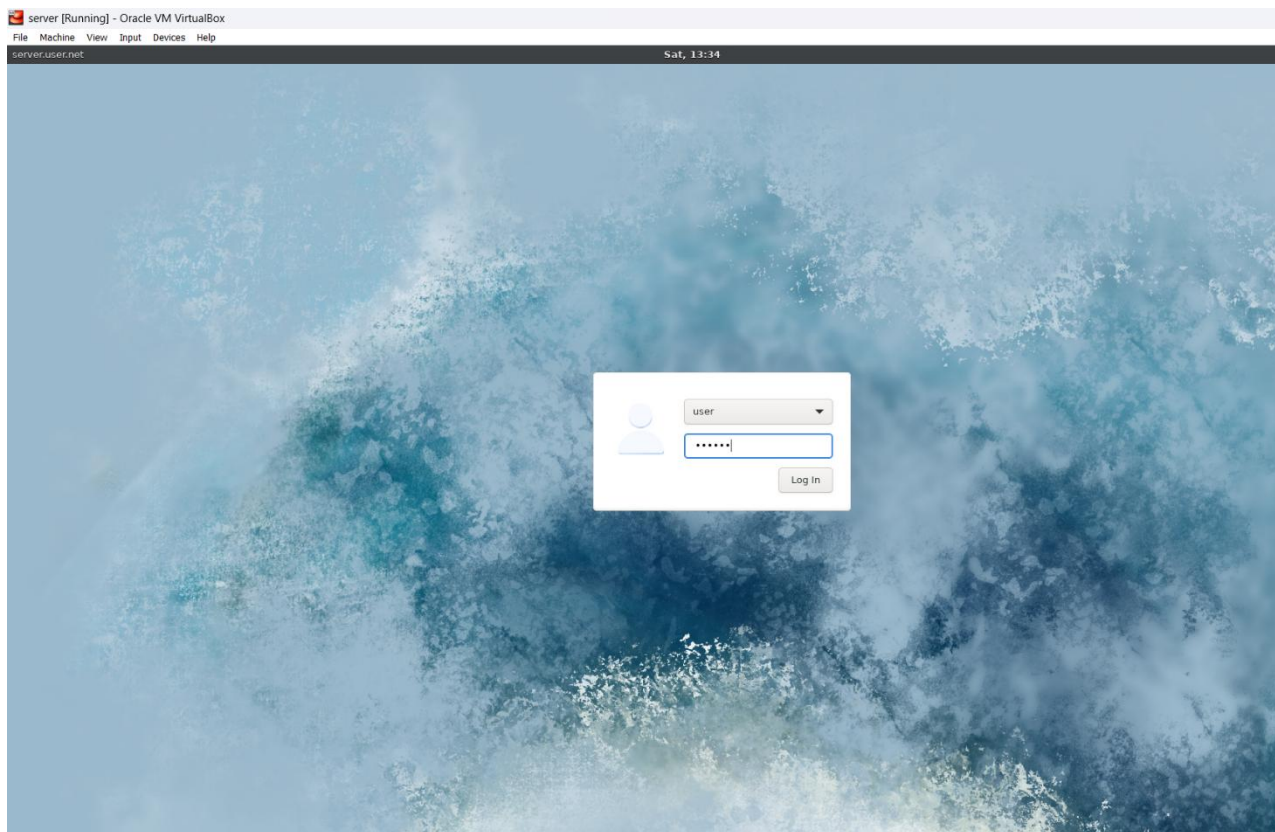
```
C:\work\user\vagrant>vagrant up server --provision
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
```

**Рис. 2.2.** Фиксирование внесённых изменений для внутренних настроек виртуальной машины (server).

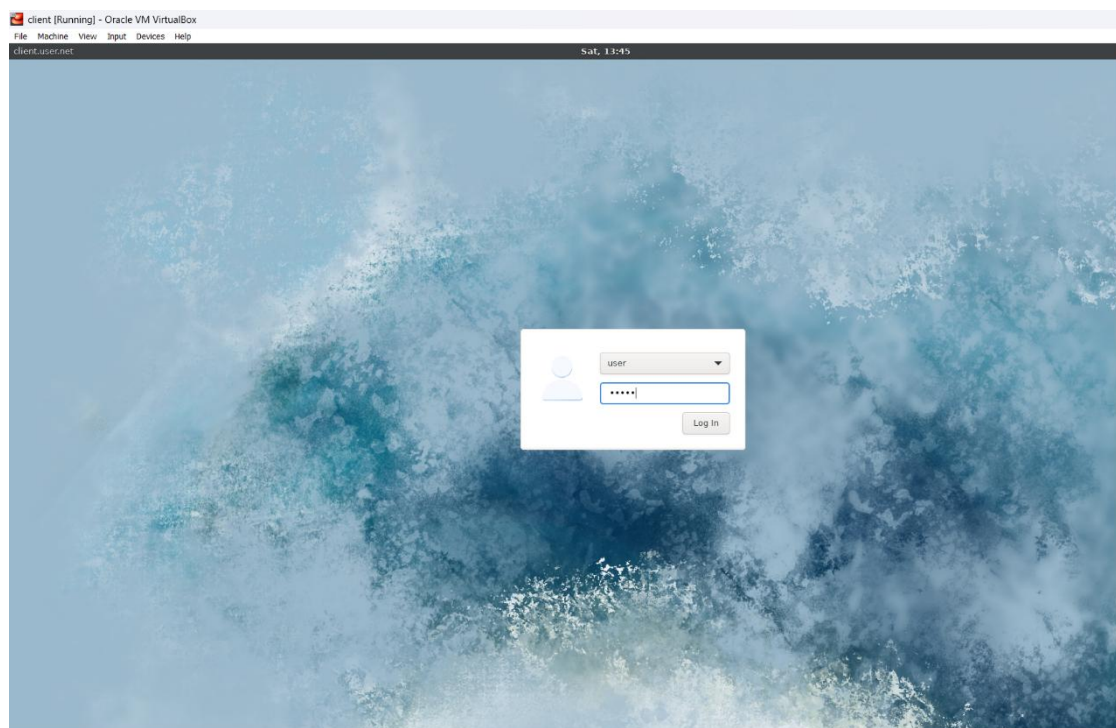
```
Bringing machine 'client' up with 'virtualbox' provider...
==> client: Clearing any previously set forwarded ports...
==> client: Fixed port collision for 22 => 2222. Now on port 2200.
==> client: Clearing any previously set network interfaces...
==> client: Preparing network interfaces based on configuration...
```

**Рис. 2.3.** Фиксирование внесённых изменений для внутренних настроек виртуальной машины (client).

Залогинимся на сервере (рис. 2.4) и клиенте (рис. 2.5) под созданным пользователем.

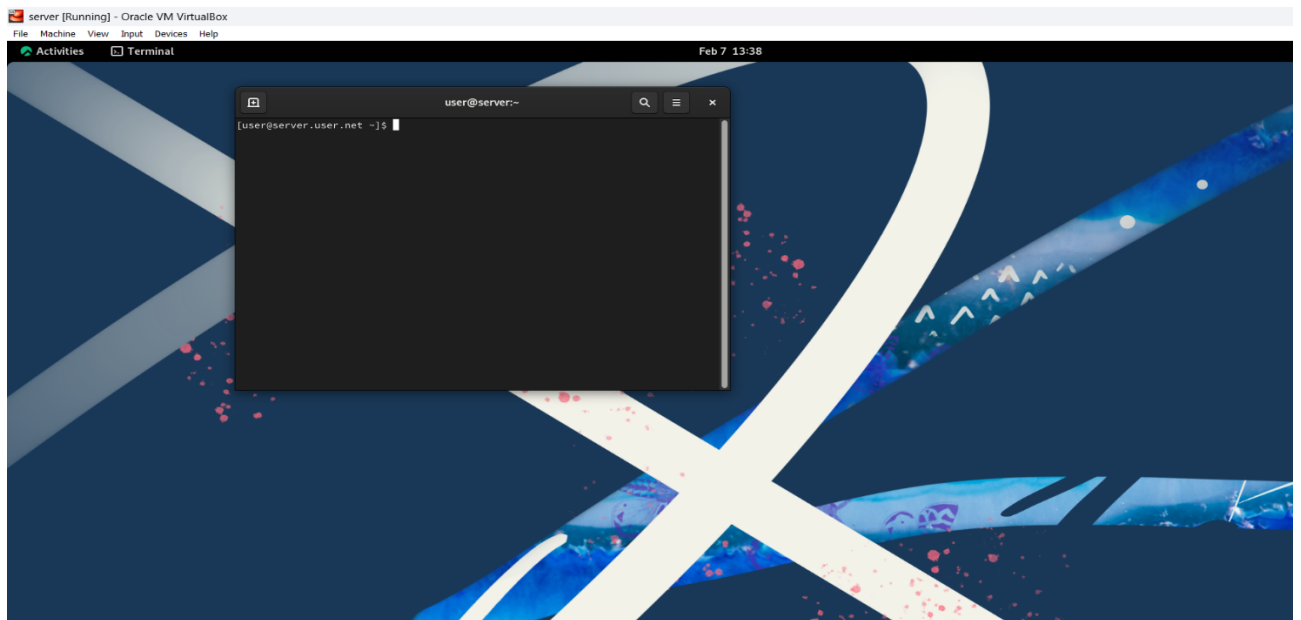


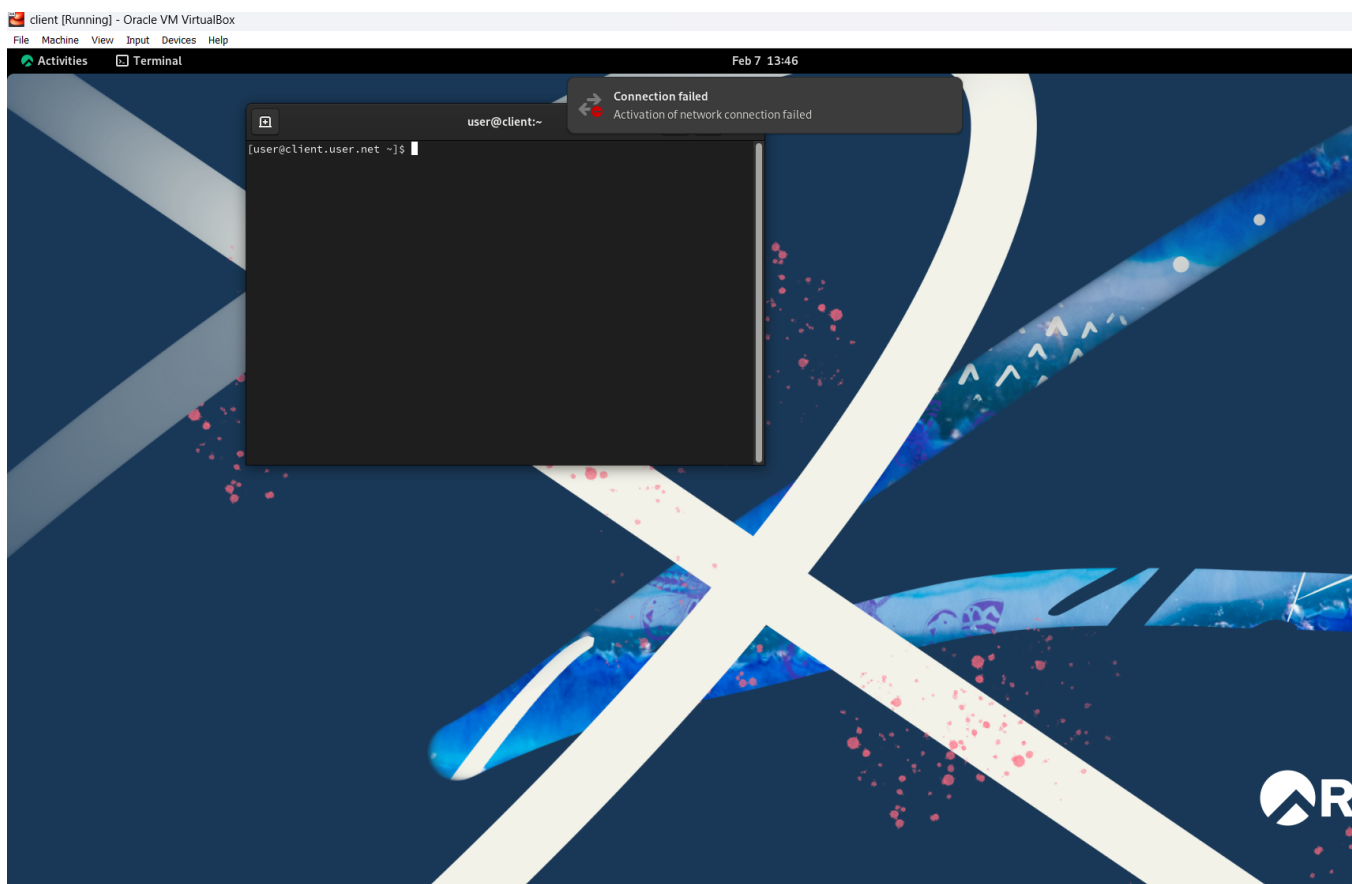
**Рис. 2.4.** Вход в учётную запись user на сервере.



**Рис. 2.5.** Вход в учётную запись user на клиенте.

Убедимся, что в терминале приглашение отображается в виде `user@server.user.net` на сервере и [user@client.user.net](#) на клиенте





**Рис. 2.6.** Проверка на сервере на и клиенте

### **Вывод:**

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

### **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`. 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 значения: – `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.