**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: Бансимба Клодели Дьегра

Студ. билет № 1032215651

Группа: НПИбд-02-22

**МОСКВА**

2024 г.

# Цель работы:

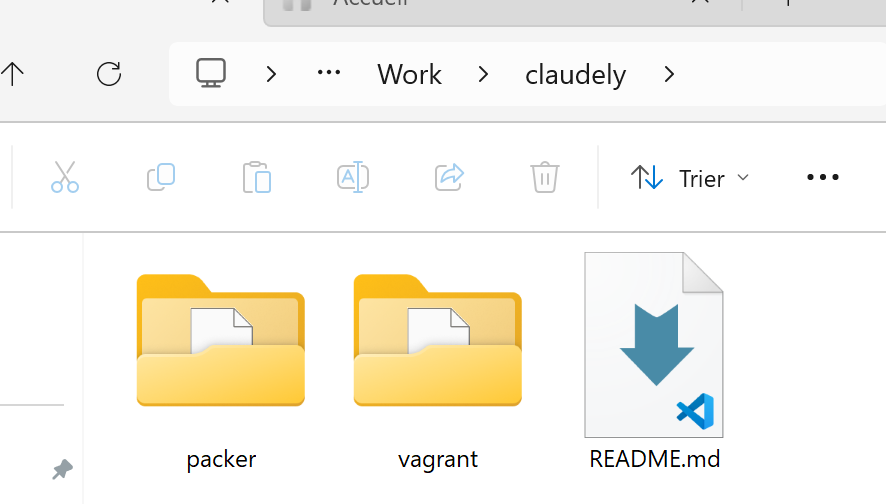
# Целью данной работы является приобретение практических навыков установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

**Выполнение работы:**

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

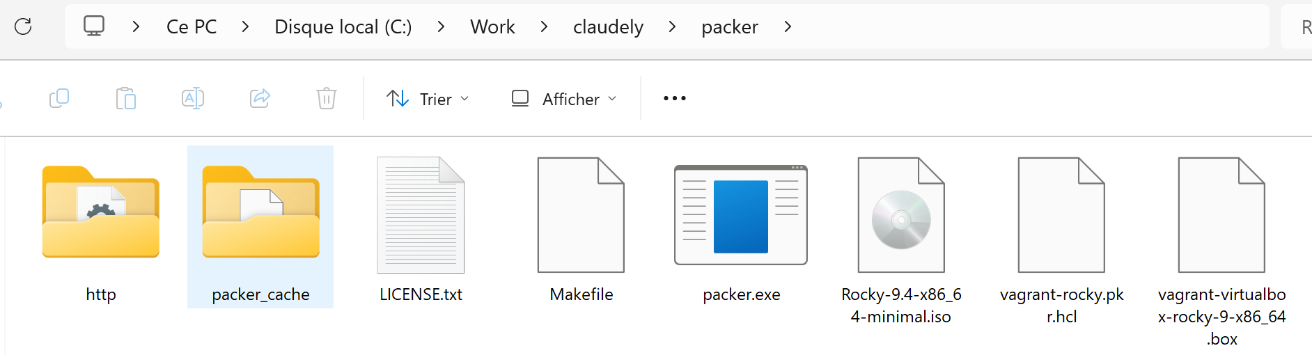
**C:\work\claudely\packer и**

**C:\work\claudely\vagrant**

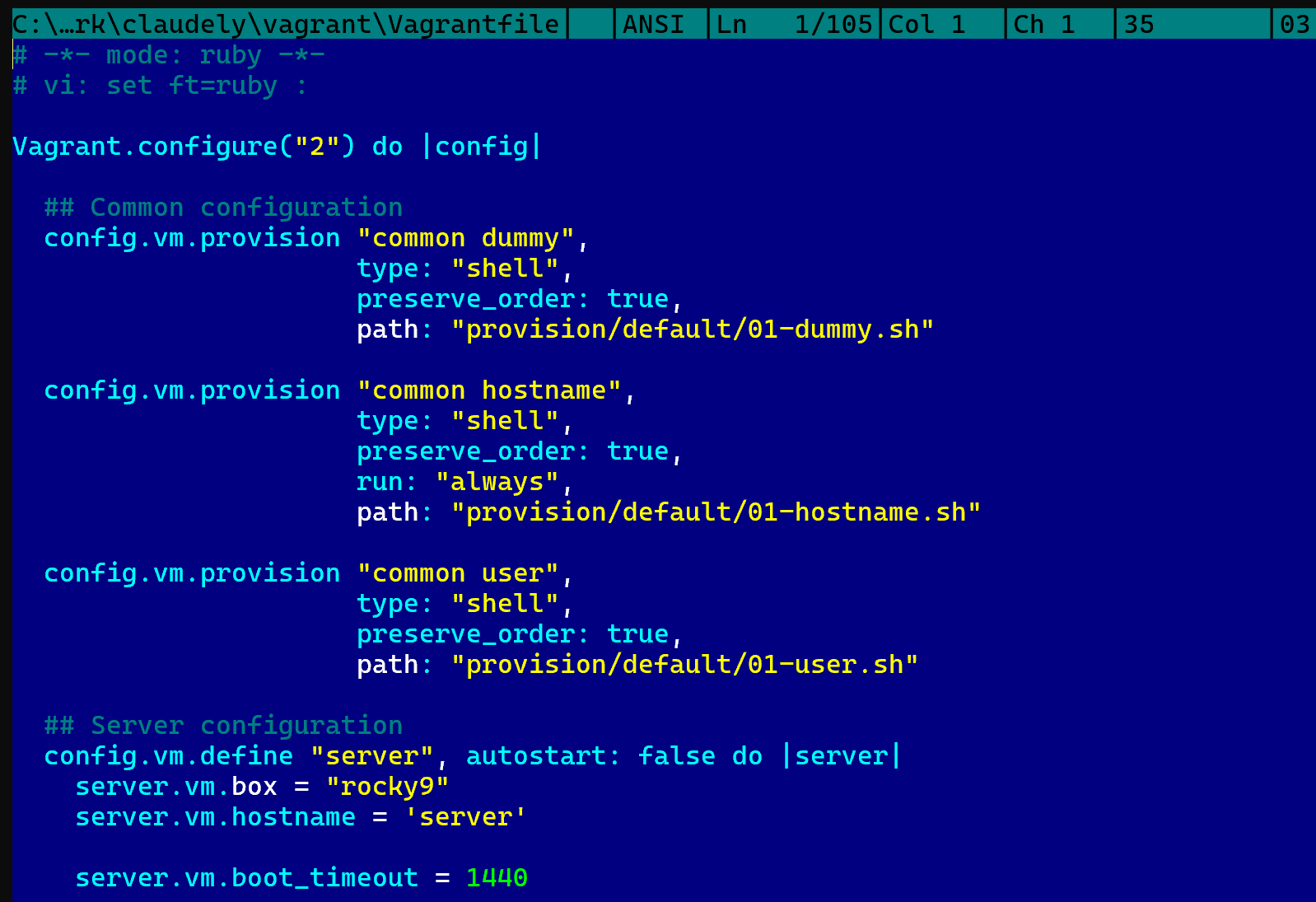


**Рис. 1.1.** Создание каталога для проекта.

В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux, в этом практикуме используем Rocky-9.2-x86\_64-minimal.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.

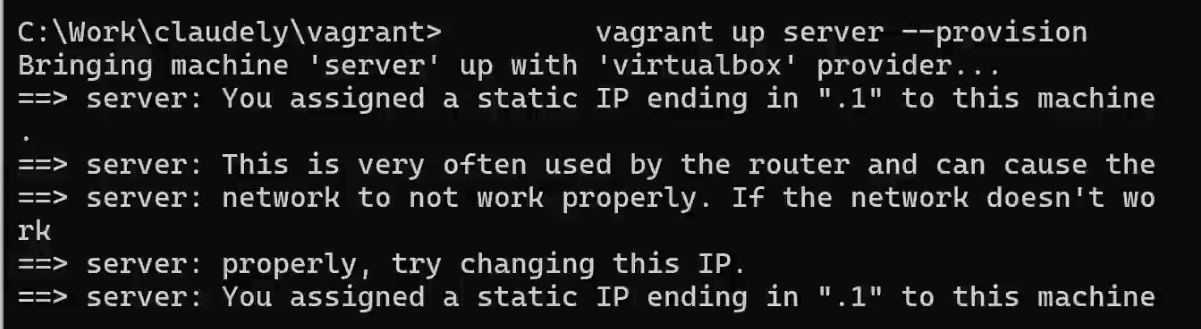
Для отработки созданных скриптов во время загрузки виртуальных машин убедимся, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется определённая запись (дана в лабораторной работе) (Рис. 2.1).

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

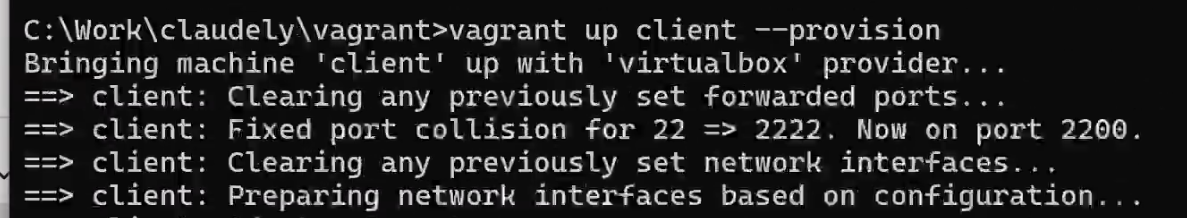
Зафиксируем внесённые изменения для внутренних настроек виртуальных машин, введя в терминале:

make server-provision (Рис. 2.2)

Затем

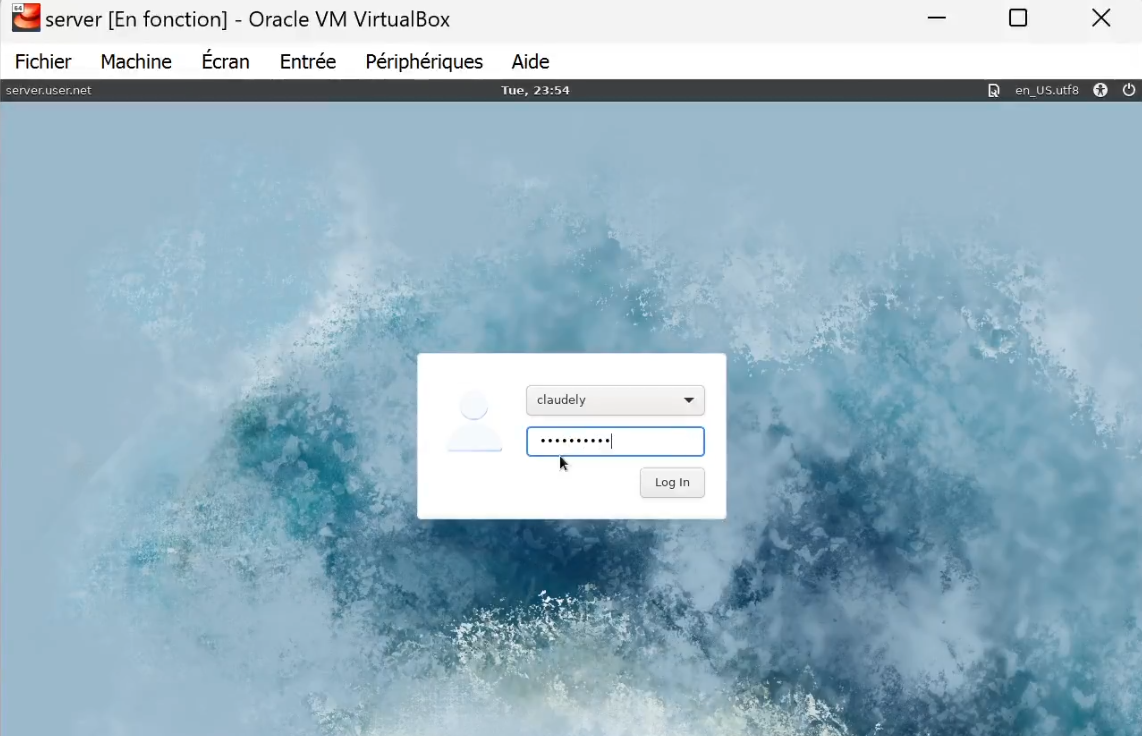
make client-provision (Рис. 2.3)

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

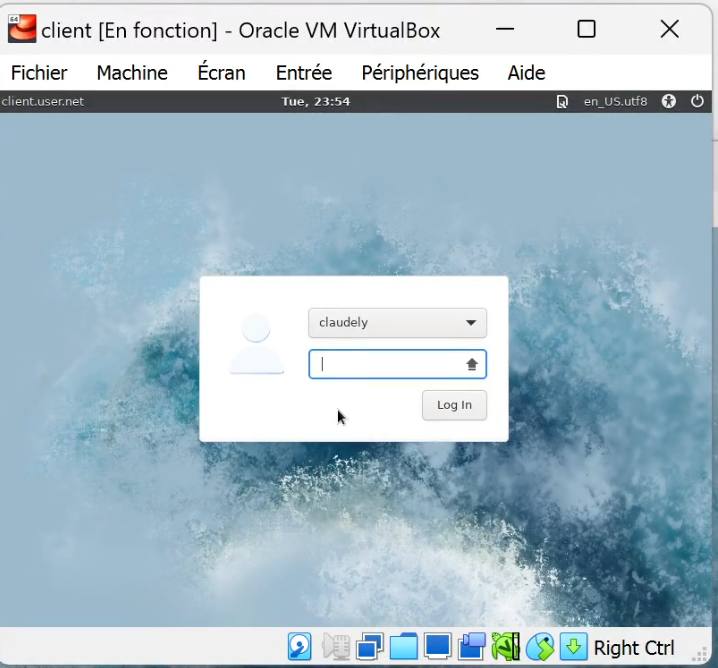


**Рис. 2.3.** Фиксирование внесённых изменений для внутренних настроек виртуальной машины (client).

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

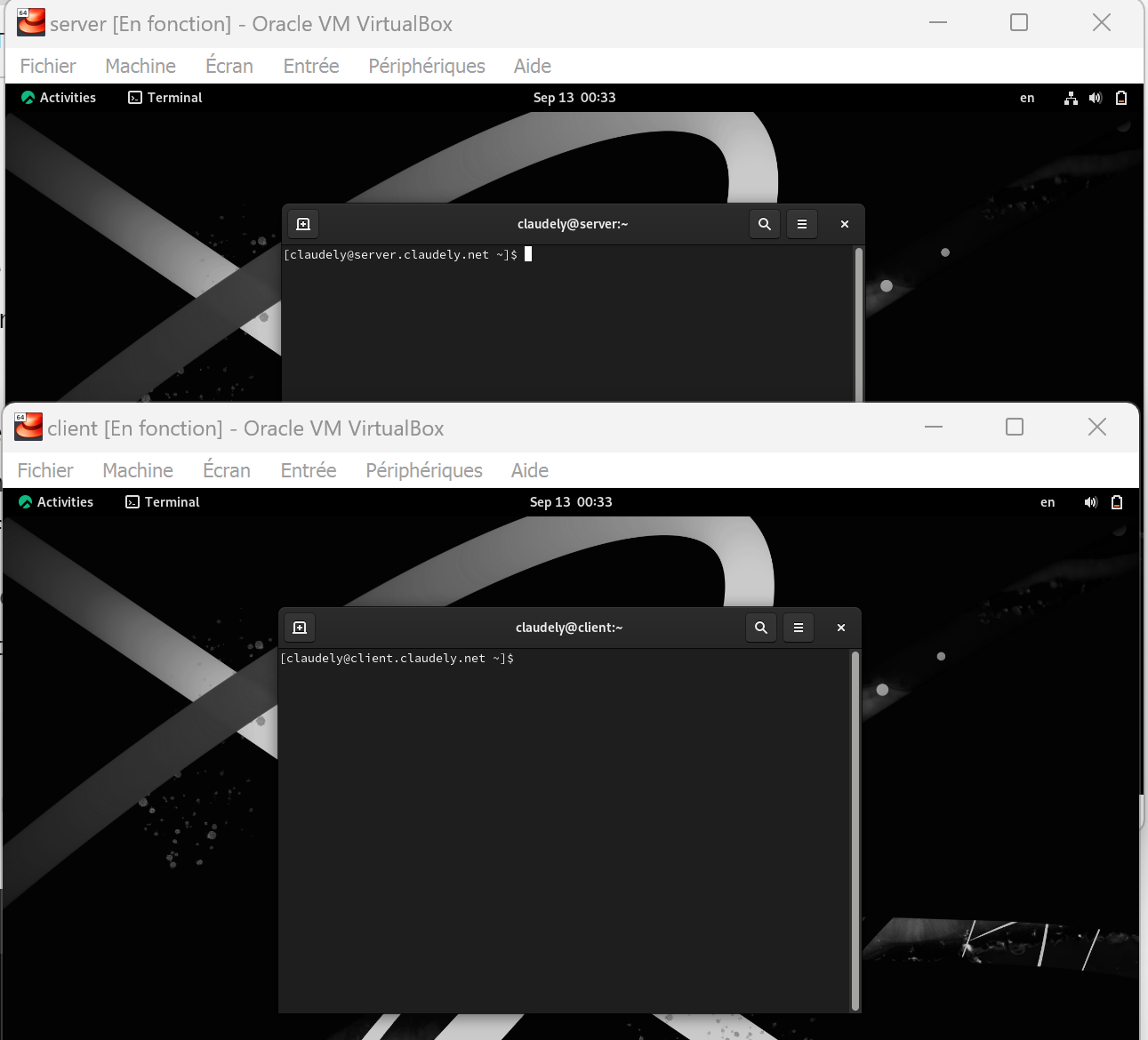


**Рис. 2.4.** Вход в учётную запись claudely на сервере.



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

Убедимся, что в терминале приглашение отображается в виде claudely@server.claudely.net на сервере и [claudely@client.claudely.net](mailto:claudely@client.claudely.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.

1. 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 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.