

Администрирование сетевых подсистем

Лабораторная работа №1

Тойчубекова Асель Нурлановна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	22

Список иллюстраций

4.1	Создание рабочего каталога и подкаталогов	9
4.2	Добавление образа ОС	9
4.3	Редактирование подкаталога packer	10
4.4	Редактирование подкаталога packer и http	10
4.5	Редактирование подкаталога vagrant	11
4.6	Редактирование подкаталога vagrant	11
4.7	Редактирование подкаталога vagrant	11
4.8	Редактирование подкаталога vagrant	12
4.9	Редактирование подкаталога vagrant	12
4.10	Редактирование подкаталога vagrant	12
4.11	Редактирование подкаталога vagrant	13
4.12	Редактирование подкаталога vagrant	13
4.13	Начало формирования box-файла	14
4.14	Начало формирования box-файла	14
4.15	Процесс формирования box-файла	15
4.16	Процесс формирования box-файла	15
4.17	Процесс формирования box-файла	15
4.18	Регистрации образа виртуальной машины в vagrant	16
4.19	Запуска виртуальной машины Server	16
4.20	Запуска виртуальной машины Client	17
4.21	Логин в Server	17
4.22	Логин в Clientr	18
4.23	Подключение к серверу	18
4.24	Переход к пользователю antoychubekova	18
4.25	Выход из пользователя antoychubekova	19
4.26	Подключение к клиенту	19
4.27	Переход к пользователю antoychubekova	19
4.28	Выход из пользователя antoychubekova	19
4.29	Выключение ВМ	19
4.30	Выключение ВМ	20
4.31	Vagrant	21

Список таблиц

1 Цель работы

Целью данной работы является приобретение практических навыков 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.
4. Скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, вы можете попробовать раз- вернуть виртуальные машины на другом компьютере.

3 Теоретическое введение

Vagrant — это инструмент для создания и управления средами виртуальных машин в рамках единого рабочего процесса. Он позволяет автоматизировать процесс развертывания операционной системы и необходимого программного обеспечения, что значительно облегчает настройку и сопровождение виртуальных окружений. Благодаря этому Vagrant широко применяется при разработке, тестировании и обучении, так как позволяет быстро воспроизводить одинаковые среды на разных компьютерах.

Работа Vagrant основывается на нескольких ключевых понятиях. Во-первых, провайдер (provider) — это система виртуализации, с которой взаимодействует Vagrant (например, VirtualBox или VMware). Во-вторых, box-файл (или Vagrant Box) — это образ виртуальной машины с уже установленной операционной системой, который используется как основа для создания новых виртуальных машин. Третьим важным элементом является Vagrantfile — конфигурационный файл на языке Ruby, в котором задаются параметры виртуальной машины: используемый box-файл, сетевые настройки, ресурсы и другие параметры.

Дополнительно для управления версиями и типами провайдеров может использоваться HCL-файл (HashiCorp Configuration Language). Он описывает метаданные установки и облегчает работу с различными образами.

Для работы с Vagrant предусмотрен набор команд, среди которых:

- `vagrant init` — создание шаблонного файла конфигурации;
- `vagrant up` — запуск виртуальной машины;

- `vagrant halt` — остановка виртуальной машины;
- `vagrant reload` — перезагрузка с применением новых настроек;
- `vagrant destroy` — удаление виртуальной машины.

Пример конфигурации Vagrantfile показывает, как можно задать имя виртуальной машины, выбрать образ, настроить сетевое взаимодействие и выделить необходимые ресурсы. Таким образом, процесс развертывания становится стандартизированным и повторяемым.

В лабораторной работе также используется сетевая топология, где сервер выполняет функции маршрутизатора и DHCP-сервера. Это позволяет клиентской машине получать доступ в Интернет и автоматически получать IP-адреса во внутренней сети.

4 Выполнение лабораторной работы

Для начала лабораторной работы создаю в диске C рабочий каталог work1, в ней подкаталог antoychubekova, и в подкаталоге подкаталоги vagrant, packer. (рис. 4.1).

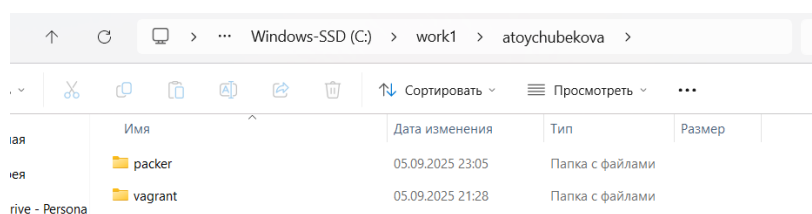


Рисунок 4.1: Создание рабочего каталога и подкаталогов

В созданном рабочем каталоге в подкаталоге packer размещаю образ варианта операционной системы Rocky Linux, Rocky-9.4-x86_64-minimal.iso. (рис. 4.2).

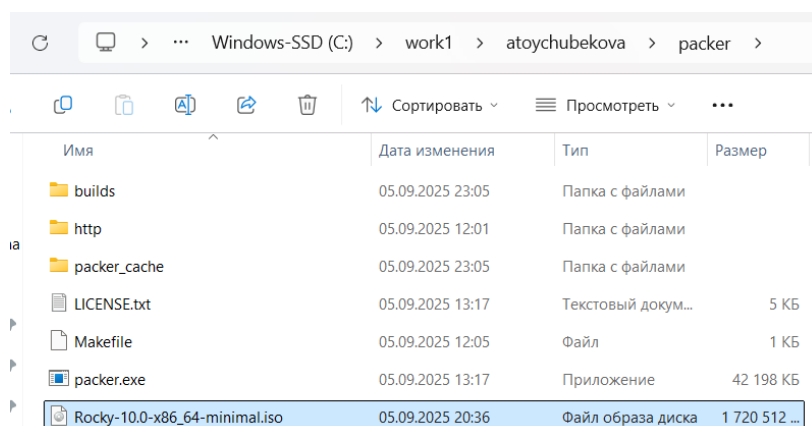


Рисунок 4.2: Добавление образа ОС

В подкаталоге packer помещаю файл vagrant-rocky.pkr.hcl — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину.

Реагирую этот файл как показано в туисе. (рис. 4.3).

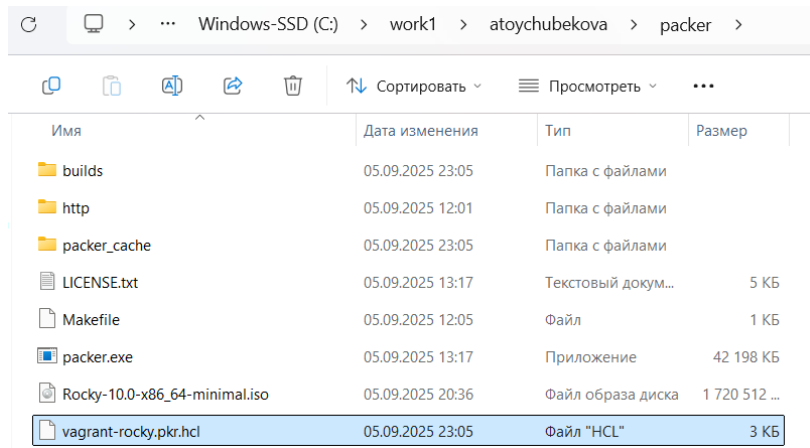


Рисунок 4.3: Редактирование подкаталога packer

В подкаталоге packer создаю подкаталог http с файлом ks.cfg — определяет настройки для установки дистрибутива, которые пользователь обычно вводит вручную, в частности настройки языка интерфейса, языковые настройки клавиатуры, тайм-зону, сетевые настройки и т.п. (рис. 4.4).

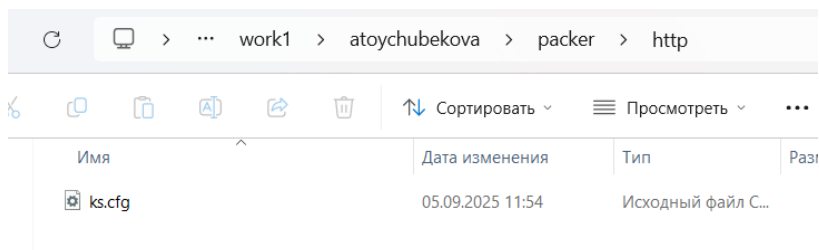


Рисунок 4.4: Редактирование подкаталога packer и http

В подкаталоге vagrant помещаю файл Vagrantfile — файл с конфигурацией запуска виртуальных машин — сервера и клиента, с текстом данные в туисе. (рис. 4.5).

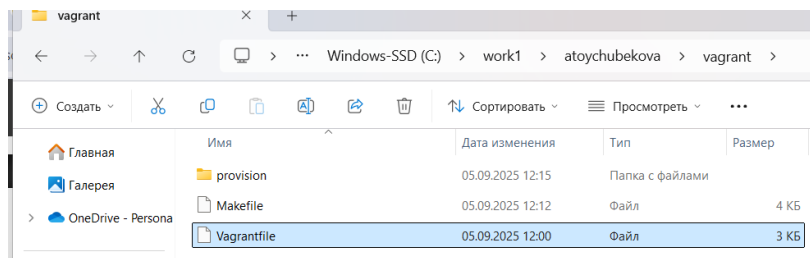


Рисунок 4.5: Редактирование подкаталога vagrant

В подкаталоге vagrant помещаю файл Makefile — набор инструкций для программы make по работе с Vagrant, с текстом данные в туисе. (рис. 4.6).

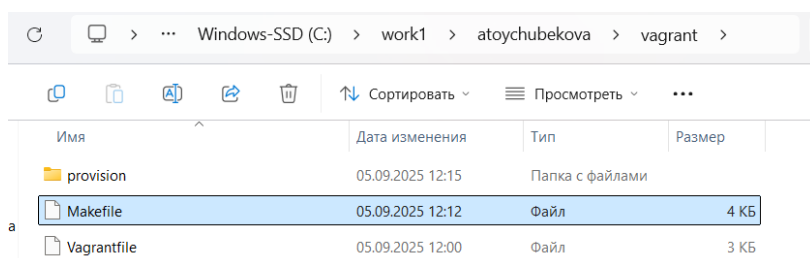


Рисунок 4.6: Редактирование подкаталога vagrant

В этом же рабочем каталоге в подкаталоге vagrant создаю 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. 4.7).

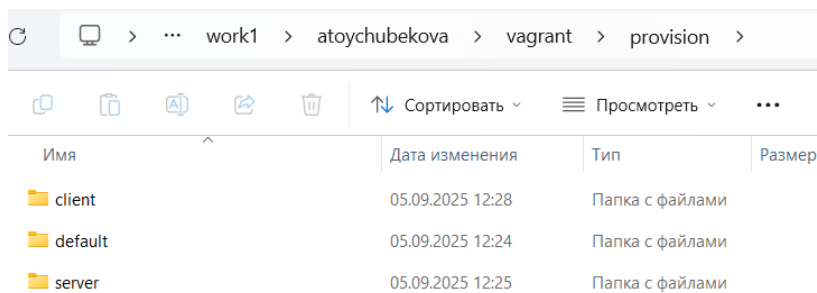


Рисунок 4.7: Редактирование подкаталога vagrant

В каталогах default, server и client разместите заранее подготовленный скрипт-заглушку 01-dummy.sh (скрипт дан в туисе). (рис. 4.8).

Имя	Дата изменения	Тип	Размер
01-dummy.sh	05.09.2025 12:17	sh_auto_file	1 КБ
01-hostname.sh	05.09.2025 12:24	sh_auto_file	1 КБ
01-user.sh	05.09.2025 12:23	sh_auto_file	1 КБ

Рисунок 4.8: Редактирование подкаталога vagrant

В каталоге default размещаю заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины. (рис. 4.9).

Имя	Дата изменения	Тип	Размер
01-dummy.sh	05.09.2025 12:17	sh_auto_file	1 КБ
01-hostname.sh	05.09.2025 12:24	sh_auto_file	1 КБ
01-user.sh	05.09.2025 12:23	sh_auto_file	1 КБ

Рисунок 4.9: Редактирование подкаталога vagrant

В каталоге default размещаю заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины. (рис. 4.10).

Имя	Дата изменения	Тип	Размер
01-dummy.sh	05.09.2025 12:17	sh_auto_file	1 КБ
01-hostname.sh	05.09.2025 12:24	sh_auto_file	1 КБ
01-user.sh	05.09.2025 12:23	sh_auto_file	1 КБ

Рисунок 4.10: Редактирование подкаталога vagrant

В каталоге server размещаю заранее подготовленный скрипт 02-forward.sh. (рис. 4.11).

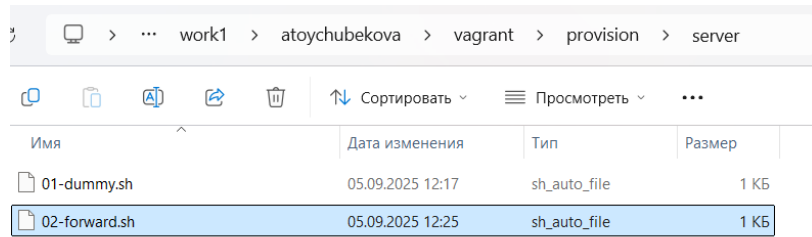


Рисунок 4.11: Редактирование подкаталога vagrant

В каталоге client размещаю заранее подготовленный скрипт 01-routing.sh. (рис. 4.12).

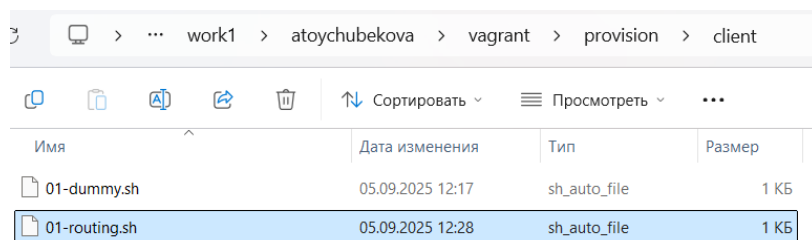


Рисунок 4.12: Редактирование подкаталога vagrant

Используя FAR, перехожу в созданный рабочий каталог с проектом. В этом же каталоге размещаю файл packer.exe. И ввожу команду для начала автоматической установки образа операционной системы Rocky Linux в VirtualBox и последующего формирования box-файла с дистрибутивом Rocky Linux для VirtualBox: packer.exe init vagrant-rocky.pkr.hcl. (рис. 4.13).

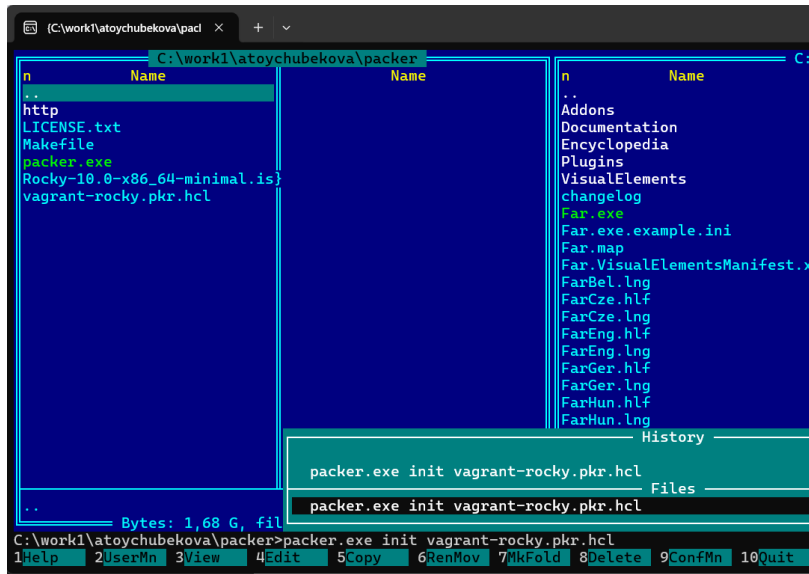


Рисунок 4.13: Начало формирования box-файла

Далее ввожу команду: `packer.exe build vagrant-rocky.pkr.hcl`. (рис. 4.14).

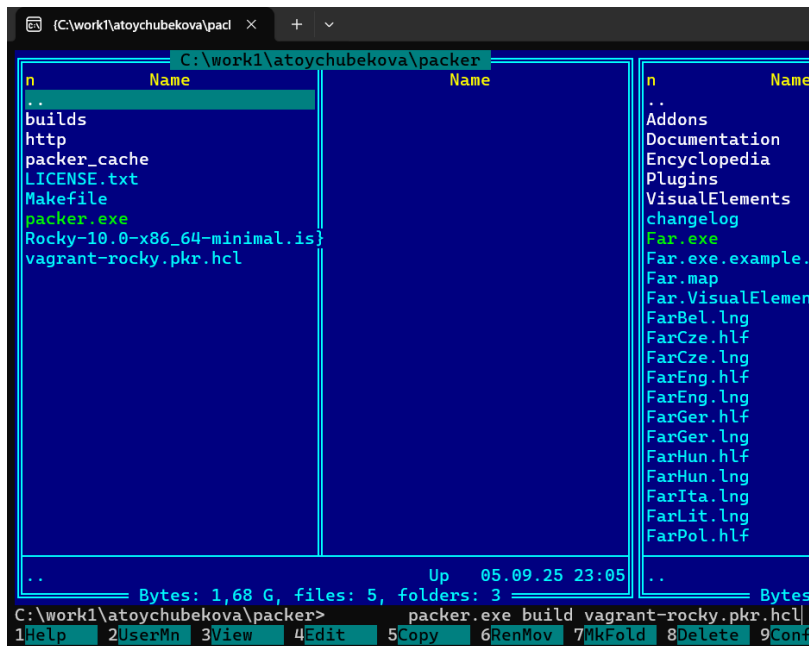


Рисунок 4.14: Начало формирования box-файла

Процесс установки образа ОС в VirtualBox и последующего формирования box-файла. По окончании процесса в рабочем каталоге сформировался box-файл с на-

званием `vagrant-virtualbox-rockylinux10-x86_64.box`. (рис. 4.15, рис. 4.16, рис. 4.17).

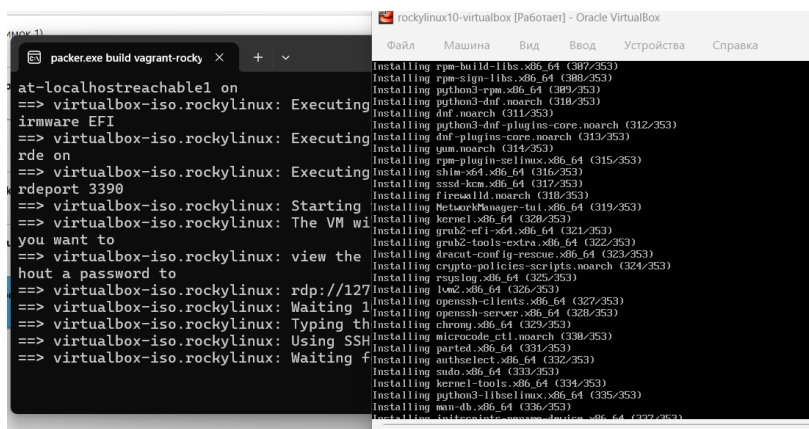


Рисунок 4.15: Процесс формирования box-файла

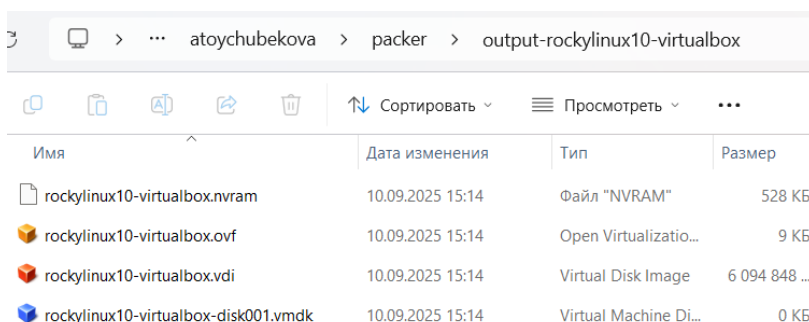


Рисунок 4.16: Процесс формирования box-файла

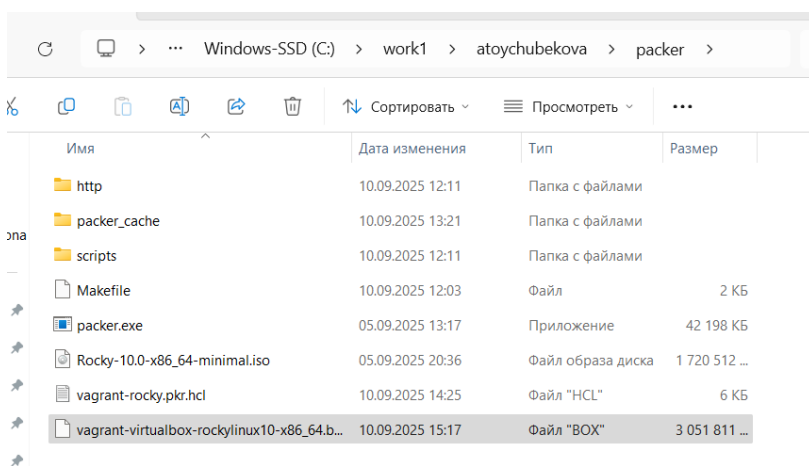


Рисунок 4.17: Процесс формирования box-файла

Для регистрации образа виртуальной машины в vagrant в командной строке ввожу: `vagrant box add rockylinux10 vagrant-virtualbox-rocky-10-x86_64.box`. (рис. 4.18).

```
C:\work1\atoychubekova\packer>vagrant box add rockylinux10 vagrant-virtualbox-rockylinux10-x86_64.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'rockylinux10' (v0) for provider: (amd64)
    box: Unpacking necessary files from: file:///C:/work1/atoychubekova/packer/vagrant-virtualbox-rockylinux10-x86_64.box
    box:
==> box: Successfully added box 'rockylinux10' (v0) for '(amd64)'!
```

Рисунок 4.18: Регистрации образа виртуальной машины в vagrant

Для запуска виртуальной машины Server ввожу в консоли `vagrant up server`. (рис. 4.19).

```
C:\work1\atoychubekova\vagrant>vagrant up server
Vagrant has detected project local plugins configured for this
project which are not installed.

    vagrant-vbguest
Install local plugins (Y/N) [N]: y
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Fetching micromachine-3.0.0.gem
Fetching vagrant-vbguest-0.32.0.gem
Installed the plugin 'vagrant-vbguest (0.32.0)'!

Vagrant has completed installing local plugins for the current Vagrant
project directory. Please run the requested command again.

C:\work1\atoychubekova\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 'rockylinux10'...
==> server: Cloning VM...
==> server: Matching MAC address for NAT networking...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
```

Рисунок 4.19: Запуска виртуальной машины Server

Для запуска виртуальной машины Client ввожу в консоли `vagrant up client`. (рис. 4.20).


```

C:\work1\atoychubekova\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: vagrant_client_1757514537783_16619
==> 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...
==> client: Booting VM...
==> client: Waiting for machine to boot. This may take a few minutes...
      client: SSH address: 127.0.0.1:2222

```

Рисунок 4.20: Запуска виртуальной машины Client

Мы видим, что запуск обеих виртуальных машин прошёл успешно, логируюсь под пользователем vagrant с паролем vagrant в графическом окружении. (рис. 4.21, рис. 4.22).

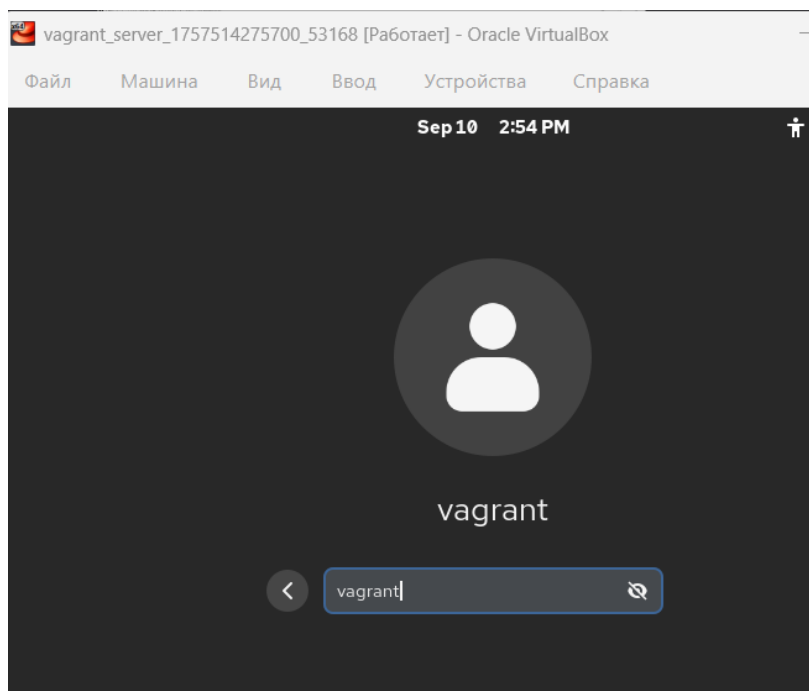


Рисунок 4.21: Логин в Server

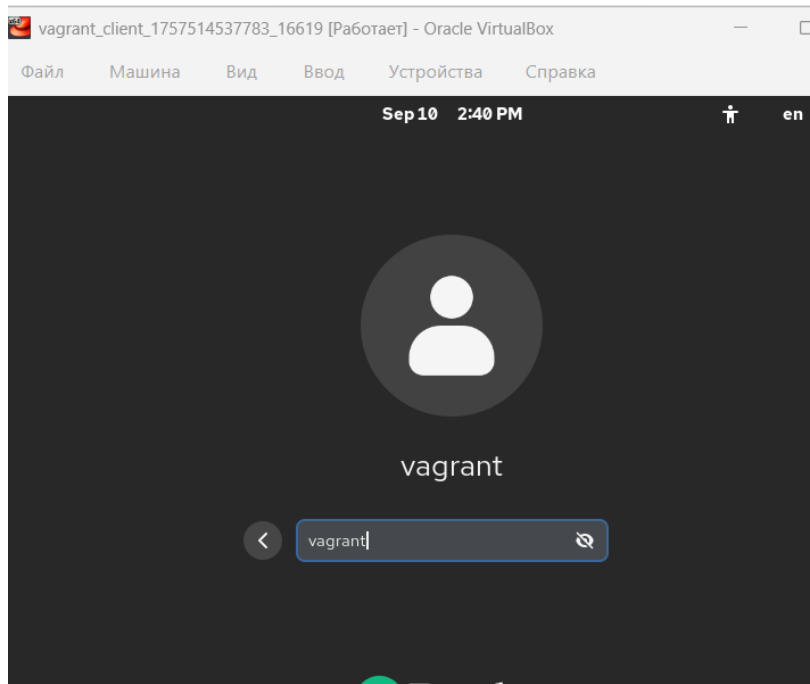


Рисунок 4.22: Логин в Clientr

Подключаюсь к серверу из консоли, введя: `vagrant ssh server`, при запросе ввожу пароль `vagrant`. (рис. 4.23).

```
C:\work1\atoychubekova\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:
Last login: Thu Sep 11 07:44:08 2025
vagrant@server:~$
```

Рисунок 4.23: Подключение к серверу

Перехожу к пользователю `antoychubekova`. (рис. 4.24).

```
vagrant@server:~$ su - antoychubekova
Password:
[antoychubekova@server.antoychubekova.net ~]$
```

Рисунок 4.24: Переход к пользователю `antoychubekova`

Отлогиниваюсь. (рис. 4.25).

```

[antoychubekova@server.antoychubekova.net ~]$ exit
logout
vagrant@server:~$ exit
logout

C:\work1\atoychubekova\vagrant>
1Help 2UserMn 3View 4Edit 5Copy 6RenMov

```

Рисунок 4.25: Выход из пользователя antoychubekova

Выполняю тоже самое для клиента. (рис. 4.26, рис. 4.27,рис. 4.28).

```

C:\work1\atoychubekova\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:
Last login: Thu Sep 11 09:04:45 2025
vagrant@client:~$

```

Рисунок 4.26: Подключение к клиенту

```

vagrant@client:~$ su - antoychubekova
Password:
[antoychubekova@client.antoychubekova.net ~]$ |

```

Рисунок 4.27: Переход к пользователю antoychubekova

```

[antoychubekova@client.antoychubekova.net ~]$ exit
logout
vagrant@client:~$ exit
logout

C:\work1\atoychubekova\vagrant>
1Help 2UserMn 3View 4Edit 5Copy 6RenMov

```

Рисунок 4.28: Выход из пользователя antoychubekova

Выключаю обе виртуальные машины. (рис. 4.29).

```

C:\work1\atoychubekova\vagrant>vagrant halt server
==> server: Attempting graceful shutdown of VM...

C:\work1\atoychubekova\vagrant>vagrant halt client
==> client: Attempting graceful shutdown of VM...

C:\work1\atoychubekova\vagrant>
1Help 2UserMn 3View 4Edit 5Copy 6RenMov 7M

```

Рисунок 4.29: Выключение VM

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»

У меня в Vagrant файле все прописано. (рис. 4.30).

```
## 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"
```

Рисунок 4.30: Выключение VM

Убедимся, что в терминале приглашение отображается в виде `antoychubekova@server.antoychubekova.net` на сервере и в виде `antoychubekova@client.antoychubekova.net` на клиенте. Это видно на (рис. 4.24, рис. 4.27)

После выключения виртуальных машин скопирую необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на в другой каталог ОС. Используя эти файлы, я могу развернуть виртуальные машины на другом компьютере. (рис. 4.31).

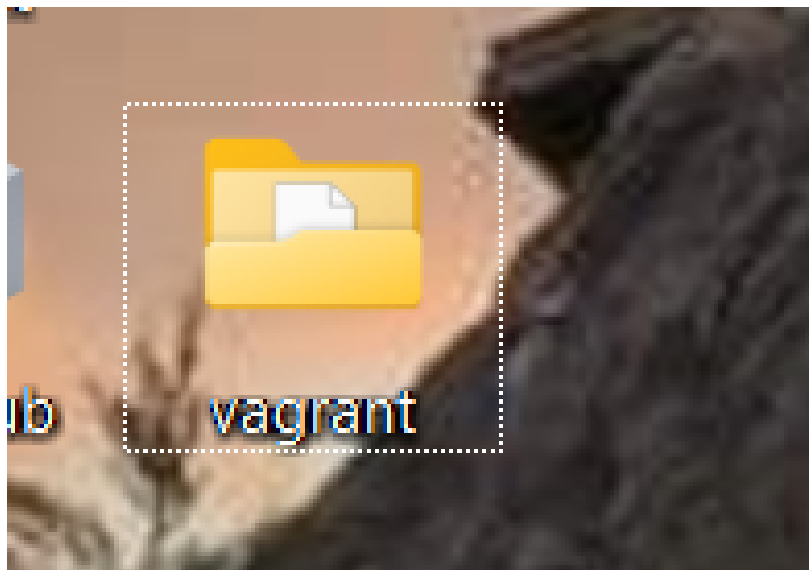


Рисунок 4.31: Vagrant

5 Выводы

В ходе выполнения лабораторной работы №1 я приобрела практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.