

# **Отчёт по лабораторной работе 1**

**Подготовка лабораторного стенда**

Чилеше Лупупа

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение работы</b>	<b>6</b>
2.1	Развёртывание лабораторного стенда в среде Windows . . . . .	6
2.2	Подготовка и настройка конфигурационных файлов . . . . .	6
2.3	Сборка образа виртуальной машины с помощью Packer . . . . .	8
2.4	Регистрация созданного образа в Vagrant . . . . .	8
2.5	Запуск виртуальной машины Server . . . . .	8
2.6	Проверка доступа к машине через SSH . . . . .	10
<b>3</b>	<b>Вывод</b>	<b>11</b>
<b>4</b>	<b>Контрольные вопросы</b>	<b>12</b>

# Список иллюстраций

2.1	Фрагмент Vagrantfile . . . . .	7
2.2	Provisioning-скрипт создания пользователя . . . . .	7
2.3	Provisioning-скрипт изменения hostname . . . . .	8
2.4	Добавление box-файла в Vagrant . . . . .	8
2.5	Запуск VM server через Vagrant . . . . .	9
2.6	Графический интерфейс виртуальной машины Server . . . . .	9
2.7	Подключение по SSH и проверка пользователя . . . . .	10

## **Список таблиц**

# 1 Цель работы

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

## **2 Выполнение работы**

### **2.1 Развёртывание лабораторного стенда в среде Windows**

Для развёртывания лабораторного стенда использовалась операционная система Windows с установленными VirtualBox и Vagrant. После установки необходимого ПО система была перезагружена.

### **2.2 Подготовка и настройка конфигурационных файлов**

В рабочем каталоге были размещены файл Vagrantfile и provisioning-скрипты для автоматической настройки виртуальных машин. На изображении показан фрагмент файла конфигурации Vagrant с настройками VirtualBox и libvirt:

```

1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  Vagrant.configure("2") do |config|
5
6      config.vagrant.plugins = ["vagrant-libvirt"]
7      config.vagrant.plugins = ["vagrant-vbguest"]
8
9      config.vm.provider :virtualbox do |virtualbox|
10         virtualbox.linked_clone = true
11         # Customize the amount of memory on the VM
12         virtualbox.memory = 2048
13         virtualbox.cpus = 2
14         ## Display the VirtualBox GUI when booting the machine
15         virtualbox.gui = true
16         ## Set the video memory to 12Mb
17         virtualbox.customize ["modifyvm", :id, "--vram", "32"]
18         virtualbox.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
19         virtualbox.customize ["modifyvm", :id, "--clipboard", "bidirectional"]
20         virtualbox.customize ["modifyvm", :id, "--draganddrop", "bidirectional"]
21         virtualbox.customize ["modifyvm", :id, "--graphicscontroller", "vmvga"]
22         virtualbox.customize ["modifyvm", :id, "--accelerate3d", "off"]
23         virtualbox.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
24     end
25
26     config.vm.provider :libvirt do |libvirt|
27         libvirt.driver = "kvm"
28         libvirt.memory = 2048
29         libvirt.cpus = 2
30         libvirt.video_type = "virtio"
31         libvirt.disk_bus = "virtio"
32         libvirt.nic_model_type = "virtio"

```

Рис. 2.1: Фрагмент Vagrantfile

Ниже приведён фрагмент скрипта создания пользователя, выполняемого при поднятии машины:

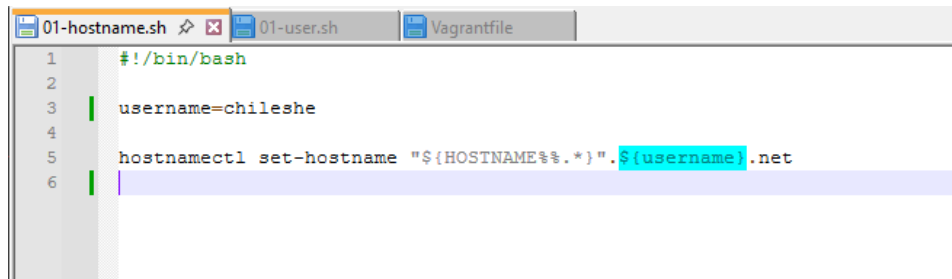
```

1  #!/bin/bash
2
3  echo "Provisioning script $0"
4
5  username=chileshe
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
18
19

```

Рис. 2.2: Provisioning-скрипт создания пользователя

Дополнительно используется скрипт для автоматической смены имени хоста:



```
01-hostname.sh 01-user.sh Vagrantfile
1  #!/bin/bash
2
3  username=chileshe
4
5  hostnamectl set-hostname "${HOSTNAME%%.*}.${username}.net"
6
```

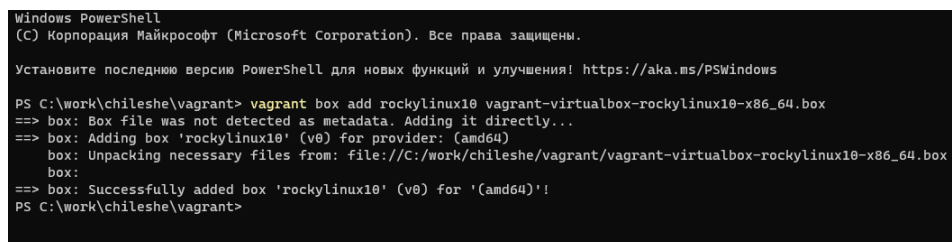
Рис. 2.3: Provisioning-скрипт изменения hostname

## 2.3 Сборка образа виртуальной машины с помощью Packer

Для автоматической установки Rocky Linux был использован Packer. После выполнения команд `packer init` и `packer build` в каталоге появился box-файл.

## 2.4 Регистрация созданного образа в Vagrant

Созданный образ был зарегистрирован в локальном репозитории Vagrant:



```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\work\chileshe\vagrant> 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:/work/chileshe/vagrant/vagrant-virtualbox-rockylinux10-x86_64.box
        box:
==> box: Successfully added box 'rockylinux10' (v0) for '(amd64)'!
PS C:\work\chileshe\vagrant>
```

Рис. 2.4: Добавление box-файла в Vagrant

## 2.5 Запуск виртуальной машины Server

После регистрации образа была запущена виртуальная машина Server:

```

PS C:\work\chileshe\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.
==> 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: vagrant_server_1763035612529_82365
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:

```

Рис. 2.5: Запуск VM server через Vagrant

Затем сервер успешно загрузился в графическом окружении:

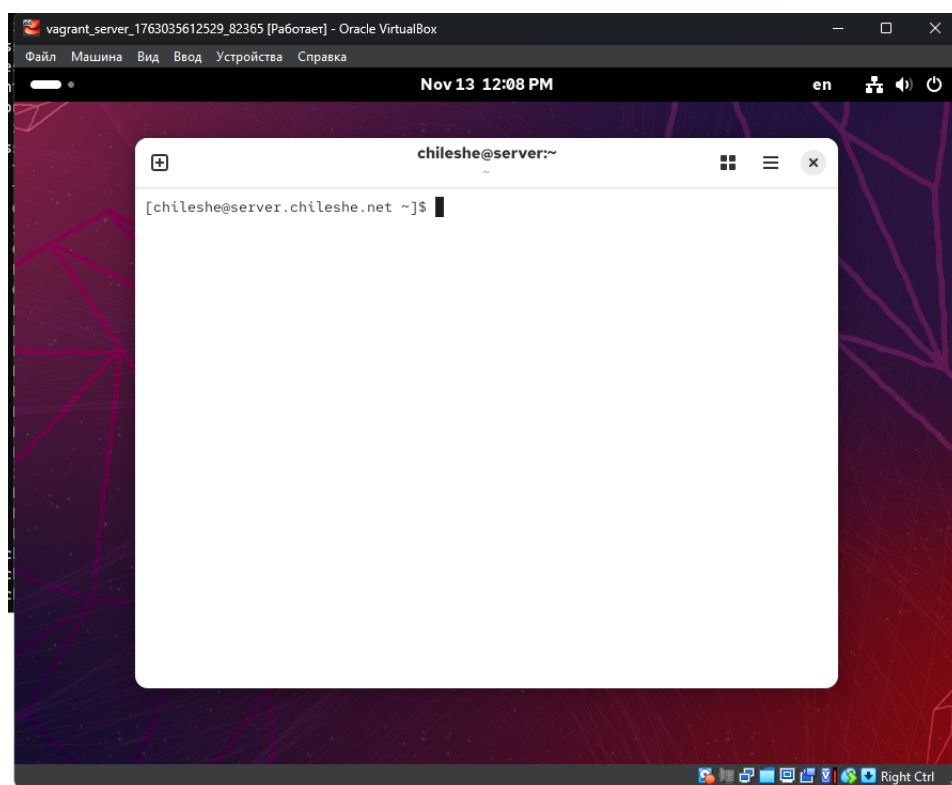


Рис. 2.6: Графический интерфейс виртуальной машины Server

## 2.6 Проверка доступа к машине через SSH

Для проверки корректности конфигурации выполнено подключение по SSH и смена пользователя:

```
PS C:\work\chileshe\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: Wed Sep  3 09:04:22 2025 from 10.0.2.2
vagrant@server:~$ su -
adm          dnsmasq          libstoragegmt    rtkit           tcpdump
avahi        flatpak          lp               setroubleshoot  tss
bin          ftp              mail             shutdown        vagrant
chileshe     games           nobody           sshd            vboxadd
chrony       gdm              operator         sssd            wsdd
clevis       geoclue          pesign           stapunpriv
colord       gnome-initial-setup pipewire         sync
daemon      gnome-remote-desktop polkitd          systemd-coredump
dbus        halt             root             systemd-oom
vagrant@server:~$ su chileshe
Password:
[chileshe@server.chileshe.net vagrant]$
[chileshe@server.chileshe.net vagrant]$
exit
vagrant@server:~$
logout
Connection to 127.0.0.1 closed.
PS C:\work\chileshe\vagrant> |
```

Рис. 2.7: Подключение по SSH и проверка пользователя

## 3 Вывод

В ходе выполнения работы был развёрнут лабораторный стенд на базе VirtualBox и Vagrant, автоматизирующий процесс создания и настройки виртуальных машин. Использование Packer позволило собрать собственный box-файл с Rocky Linux, а provisioning-скрипты обеспечили подготовку окружения без ручного вмешательства. Все этапы — от генерации образа до запуска и проверки доступа к виртуальным машинам — выполнены успешно, что подтвердило корректность конфигурации и работоспособность стенда.

## 4 Контрольные вопросы

### 1. Для чего предназначен Vagrant?

Vagrant используется для автоматизированного развертывания и управления виртуальными машинами. Он позволяет создавать повторяемые и предсказуемые среды разработки, устраняя необходимость ручной настройки каждой системы. С его помощью разработчики и администраторы могут быстро поднимать виртуальные стенды, используя единый конфигурационный файл.

### 2. Что такое box-файл? В чём назначение Vagrantfile?

Box-файл — это шаблон (образ) виртуальной машины, включающий предустановленную операционную систему и базовые настройки. Он служит основой для создания новых виртуальных машин в Vagrant.

Vagrantfile — главный конфигурационный файл проекта Vagrant. В нём указываются параметры виртуальных машин: используемый box-файл, объём памяти, количество процессоров, настройки сетевых интерфейсов, синхронизация папок и запуск provisioning-скриптов.

### 3. Приведите описание и примеры вызова основных команд Vagrant.

Основные команды Vagrant включают:

- `vagrant init` — создание нового Vagrantfile;
- `vagrant box add` — добавление box-файла в локальный репозиторий;
- `vagrant up` — запуск и создание виртуальной машины;
- `vagrant ssh` — подключение к машине по SSH;
- `vagrant halt` — остановка виртуальной машины;
- `vagrant destroy` — удаление созданной машины;

- `vagrant status` — просмотр состояния виртуальной машины.

Примеры использования:

- Запуск виртуальной машины: `vagrant up server`

- Подключение по SSH: `vagrant ssh client`

- Остановка машины: `vagrant halt server`

#### **4. Дайте построчные пояснения содержания файлов `vagrant-rocky.pkr.hcl`, `ks.cfg`, `Vagrantfile`, `Makefile`.**

- **`vagrant-rocky.pkr.hcl`** — конфигурация Packer. Описывает источник ISO-образа, параметры виртуальной машины для установки, путь к kickstart-файлу, а также процесс сборки box-файла.

- **`ks.cfg`** — kickstart-файл, используемый для автоматической установки Rocky Linux. Содержит параметры разметки диска, выбора пакетов, создания пользователей, сетевых настроек и других этапов инсталляции.

- **`Vagrantfile`** — конфигурационный файл Vagrant. Определяет используемый box-файл, настройки провайдера (VirtualBox или libvirt), параметры оперативной памяти, процессоров, сетевых интерфейсов, а также provisioning-скрипты, выполняющиеся при создании машины.

- **`Makefile`** — набор автоматизированных задач для управления проектом. Может включать цели для запуска, остановки и пересоздания виртуальных машин, упрощая выполнение сложных последовательностей команд.