

Отчёт по лабораторной работе 1

Подготовка лабораторного стенда

Элсаиед Адел

Содержание

1 Цель работы	5
2 Выполнение	6
2.1 Развёртывание виртуальных машин с использованием Vagrant и VirtualBox	6
3 Вывод	11
4 Контрольные вопросы	12

Список иллюстраций

2.1 Добавление box-образа Rocky Linux	6
2.2 Настройка Vagrantfile	7
2.3 Скрипт создания пользователя	7
2.4 Скрипт настройки hostname	8
2.5 Запуск виртуальной машины Server	8
2.6 Успешный вход в систему Server	9
2.7 Подключение к серверу по SSH	9
2.8 Запуск виртуальной машины Client	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

2.1 Развёртывание виртуальных машин с использованием Vagrant и VirtualBox

1. В операционной системе Windows в рабочем каталоге проекта выполнена регистрация box-образа Rocky Linux в Vagrant с помощью команды `vagrant box add rockylinux10 vagrant-virtualbox-rockylinux10-x86_64.box`. В процессе выполнения команда автоматически распаковала файл и успешно добавила box `rockylinux10` для архитектуры `amd64`.

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

Рис. 2.1: Добавление box-образа Rocky Linux

2. В файле `Vagrantfile` была выполнена настройка параметров виртуальной машины для провайдера `VirtualBox`. Заданы параметры использования `linked clone`, объём оперативной памяти 2048 МБ, количество процессоров – 2, включение графического интерфейса, настройка видеопамяти, буфера обмена, `drag-and-drop` и аппаратной виртуализации.

```

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", "64"]
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", "vmsvga"]
22    virtualbox.customize ["modifyvm", :id, "--accelerate3d", "off"]
23    virtualbox.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
24
25 end

```

Рис. 2.2: Настройка Vagrantfile

3. Для автоматической инициализации пользователя был подготовлен provisioning-скрипт 01-user.sh. Скрипт выполняет проверку наличия пользователя в системе, создаёт его при отсутствии, добавляет в группу wheel, задаёт пароль и настраивает переменную окружения PS1.

```

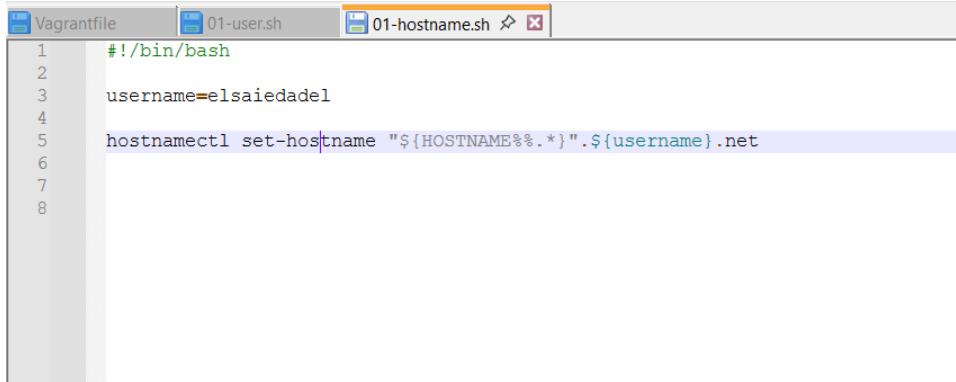
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 username=elsaiedadel
6 userpassword=123456
7
8 encpassword=`openssl passwd -1 ${userpassword}`
9
10 id -u $username
11 if [[ $? != 0 ]]
12 then
13   adduser -G wheel -p ${encpassword} ${username}
14   homedir=`getent passwd ${username} | cut -d: -f6`
15   echo "export PS1='[\u@\H \W]\$\'' >> ${homedir}/.bashrc
16 fi
17
18
19

```

Рис. 2.3: Скрипт создания пользователя

4. Для задания имени хоста виртуальной машины был создан скрипт 01-

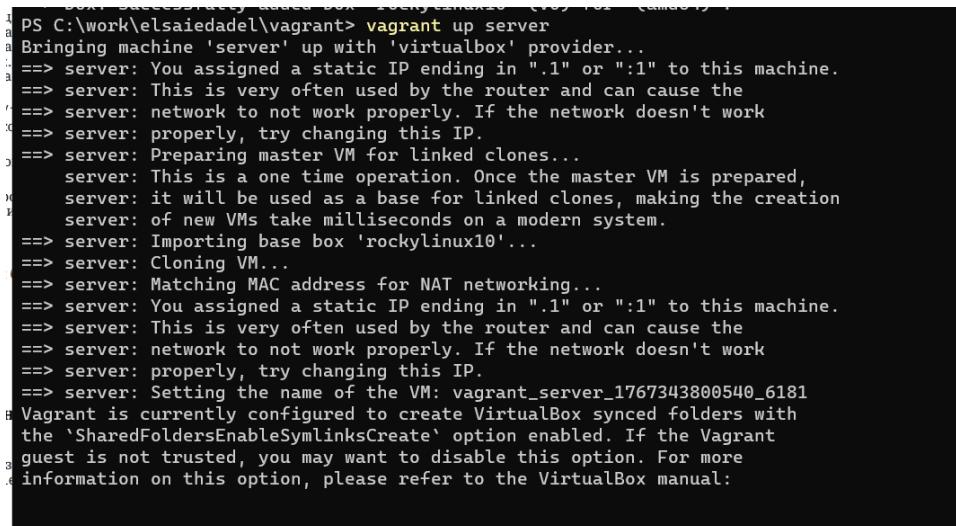
hostname.sh, который формирует имя хоста на основе текущего имени системы и логина пользователя с использованием утилиты hostnamectl.



```
1 #!/bin/bash
2
3 username=elsaiedadel
4
5 hostnamectl set-hostname "${HOSTNAME%.*}}.${username}.net
6
7
8
```

Рис. 2.4: Скрипт настройки hostname

5. Выполнен запуск виртуальной машины Server. В процессе запуска произведено клонирование виртуальной машины, настройка сетевых интерфейсов и автоматическая инициализация окружения. Виртуальная машина успешно запущена и готова к работе.



```
PS C:\work\elsaiedadel\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_1767343800540_6181
--> 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 виртуальной машины Server

6. После завершения загрузки выполнен вход в графическое окружение под созданным пользователем. Имя хоста было успешно изменено в соответствии с настройками provisioning-скрипта.

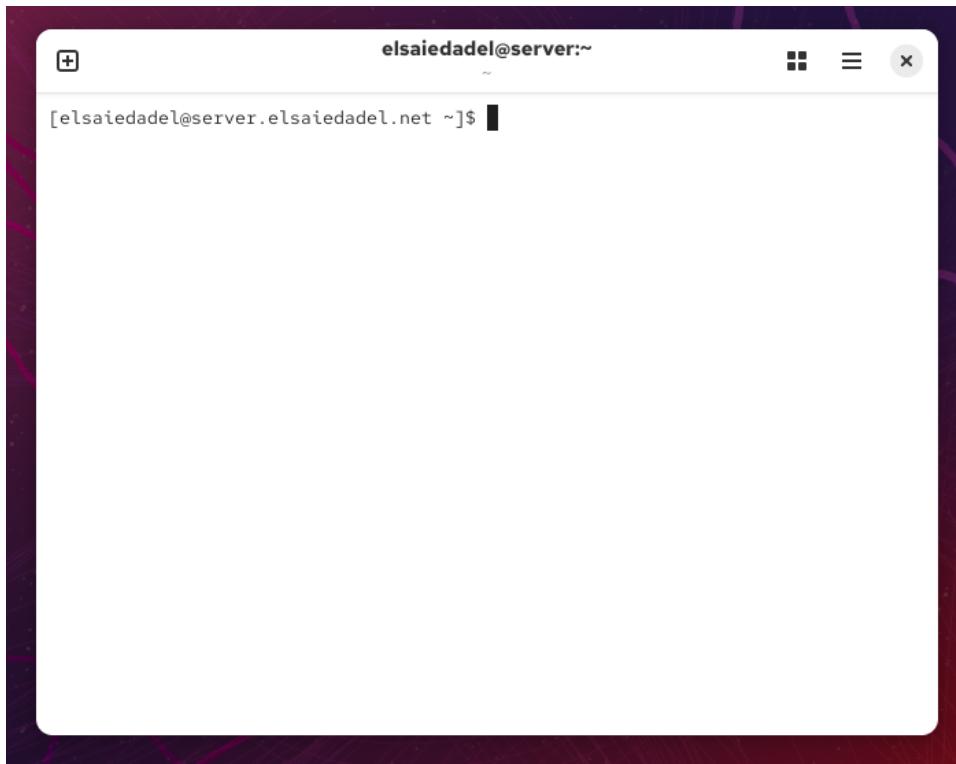


Рис. 2.6: Успешный вход в систему Server

7. Из консоли хоста выполнено подключение к серверу по SSH. После аутентификации выполнен вход под пользователем vagrant, затем произведён переход к созданному пользователю.

```
PS C:\work\elsaiedadel\vagrant>
PS C:\work\elsaiedadel\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 elsaiedadel
Password:
[elsaiedadel@server.elsaiedadel.net vagrant]$
exit
vagrant@server:~$
logout
Connection to 127.0.0.1 closed.
PS C:\work\elsaiedadel\vagrant> |
```

Рис. 2.7: Подключение к серверу по SSH

8. Аналогичным образом выполнен запуск виртуальной машины Client. В ходе запуска произведена настройка сетевых интерфейсов, проброс портов

и инициализация системы. Виртуальная машина успешно загружена и готова к эксплуатации.

```
PS C:\work\elsaiedadel\vagrant>
PS C:\work\elsaiedadel\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_1767344016057_93232
==> 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...
  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 a few minutes...
  client: SSH address: 127.0.0.1:2200
  client: SSH username: vagrant
  client: SSH auth method: password
==> client: Machine booted and ready!
Got different reports about installed GuestAdditions version:
Virtualbox on your host claims:
VBoxService inside the vm claims: 7.1.12
Going on, assuming VBoxService is correct...
```

Рис. 2.8: Запуск виртуальной машины Client

9. После проверки корректности работы обеих виртуальных машин выполнен выход из пользовательских сессий и завершение работы виртуальных машин.

3 Вывод

В ходе выполнения лабораторной работы было выполнено автоматизированное развёртывание виртуальных машин Server и Client в среде VirtualBox с использованием инструмента Vagrant. С помощью Packer был создан box-образ операционной системы Rocky Linux, который зарегистрирован в Vagrant и использован для последующего развёртывания виртуальной инфраструктуры. Реализована настройка параметров виртуальных машин, автоматическое создание пользователей и конфигурирование имени хоста средствами provisioning-скриптов. Проверка показала корректный запуск виртуальных машин, успешную аутентификацию пользователей и возможность подключения по SSH. Использование Vagrant позволило существенно упростить и стандартизировать процесс развёртывания и управления виртуальной средой.

4 Контрольные вопросы

1. Для чего предназначен Vagrant?

Vagrant предназначен для автоматизации создания, настройки и управления виртуальными машинами. Он позволяет описывать конфигурацию виртуальной среды в виде кода, обеспечивая воспроизводимость, удобство развертывания и единообразие окружений для разработки, тестирования и обучения.

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

Box-файл – это предварительно подготовленный образ виртуальной машины, содержащий операционную систему и базовые настройки, используемый Vagrant для быстрого развертывания виртуальных сред.

Vagrantfile – это конфигурационный файл проекта Vagrant, в котором описываются параметры виртуальных машин: используемый box, провайдер виртуализации, сетевые настройки, ресурсы, а также provisioning-скрипты для автоматической настройки системы.

3. Приведите описание и примеры вызова основных команд Vagrant.

vagrant box add – регистрирует box-образ в локальном репозитории Vagrant.

vagrant up – создаёт и запускает виртуальную машину в соответствии с настройками Vagrantfile.

vagrant ssh – выполняет подключение к виртуальной машине по SSH.

vagrant halt – корректно завершает работу виртуальной машины.

vagrant status – отображает текущее состояние виртуальных машин проекта.

vagrant destroy – удаляет виртуальную машину и все связанные с ней ресурсы.

4. Дайте построчные пояснения содержания файлов vagrant-rocky.pkr.hcl,

ks.cfg, Vagrantfile, Makefile.

vagrant-rocky.pkr.hcl

Файл конфигурации Packer, в котором описывается процесс автоматической установки операционной системы Rocky Linux. В нём указывается тип билдера (VirtualBox), путь к ISO-образу, параметры виртуальной машины, а также сценарии provisioning, выполняемые в процессе сборки образа.

ks.cfg

Файл автоматической установки (Kickstart), содержащий ответы на вопросы установщика: разметку диска, выбор пакетов, настройки сети, пользователя и пароля. Используется для полностью автоматизированной установки операционной системы без участия пользователя.

Vagrantfile

Файл конфигурации Vagrant, в котором задаётся версия конфигурации, используемый box-образ, параметры провайдера VirtualBox, настройки аппаратных ресурсов, сетевые параметры и provisioning-скрипты, выполняемые при запуске виртуальных машин.

Makefile

Файл сценариев сборки, предназначенный для автоматизации типовых операций проекта. Содержит цели для запуска Packer, сборки box-файла, его регистрации в Vagrant и выполнения вспомогательных команд, что упрощает управление процессом сборки и развёртывания.