Отчёт по лабораторной работе №1

Дисциплина: Администрирование сетевых подсистем

Ибрахим Мохсейн Алькамаль

Содержание

# 1. Цель работы

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

# 2. Выполнение лабораторной работы

В рабочем каталоге ОС Windows созданы подкаталоги C:\work\alkamal\packer и C:\work\alkamal\vagrant, предназначенные для подготовки box-файла Vagrant ([рис. 1](#fig-1)). В каталоге packer размещены образ Rocky-9.4-x86\_64-minimal.iso, исполняемый файл packer.exe, файл конфигурации vagrant-rocky.pkr.hcl и подкаталог http, что соответствует требованиям к структуре проекта ([рис. 2](#fig-2)).

|  |
| --- |
| Рисунок 1: Структура рабочего каталога с подкаталогами packer и vagrant |

|  |
| --- |
| Рисунок 2: Содержимое каталога packer с ISO-образом и HCL-файлом |

Файл vagrant-rocky.pkr.hcl содержит блок packer с разделом required\_plugins (vagrant, virtualbox), а также переменные, определяющие версию дистрибутива, размер диска, контрольную сумму ISO-образа, архитектуру и параметры SSH ([рис. 3](#fig-3)). В подкаталоге http размещён файл ks.cfg, задающий автоматизированные параметры установки системы: загрузчик, очистку разделов, язык, сетевые настройки (DHCP), создание пользователя и post-настройки ([рис. 4](#fig-4)).

|  |
| --- |
| Рисунок 3: Файл vagrant-rocky.pkr.hcl с описанием переменных и плагинов |

|  |
| --- |
| Рисунок 4: Файл ks.cfg с параметрами автоматической установки Rocky Linux |

В каталоге vagrant размещён файл Vagrantfile, определяющий конфигурацию виртуальной машины server: имя box rocky9, сетевой интерфейс с IP 192.168.1.1, параметры VirtualBox (RAM, CPU, VRAM, графический контроллер) и подключение provision-скриптов ([рис. 5](#fig-5)). Также создан каталог provision с подкаталогами default, server и client, предназначенными для разделения сценариев настройки окружения ([рис. 6](#fig-6)).

|  |
| --- |
| Рисунок 5: Файл Vagrantfile с конфигурацией виртуальной машины server |

|  |
| --- |
| Рисунок 6: Структура каталога provision с подкаталогами default, server и client |

В подкаталогах default, server и client размещены скрипты-заглушки 01-dummy.sh, выводящие сообщение о выполнении provisioning ([рис. 9](#fig-7)). В каталоге default дополнительно размещены скрипты 01-user.sh (создание пользователя alkamal, добавление в группу wheel, настройка PS1) и 01-hostname.sh (установка FQDN вида \*.alkamal.net) ([рис. 10](#fig-8), [рис. 11](#fig-9)). В каталоге server размещён скрипт 02-forward.sh, активирующий IP-маршрутизацию и masquerading через firewalld, а в каталоге client — скрипт 01-routing.sh, настраивающий шлюз 192.168.1.1 и параметры интерфейсов через nmcli ([рис. 12](#fig-10), [рис. 13](#fig-11)).

|  |
| --- |
| Рисунок 7: Скрипты-заглушки 01-dummy.sh в каталогах default, server и client |

|  |
| --- |
| Рисунок 8: Скрипты-заглушки 01-dummy.sh в каталогах default, server и client |

|  |
| --- |
| Рисунок 9: Скрипты-заглушки 01-dummy.sh в каталогах default, server и client |

|  |
| --- |
| Рисунок 10: Скрипт 01-user.sh для создания пользователя и настройки окружения |

|  |
| --- |
| Рисунок 11: Скрипт 01-hostname.sh для установки доменного имени хоста |

|  |
| --- |
| Рисунок 12: Скрипт 02-forward.sh для включения IP-forwarding и masquerading |

|  |
| --- |
| Рисунок 13: Скрипт 01-routing.sh для настройки маршрутизации клиента |

# 3. Развёртывание лабораторного стенда на ОС Windows

В рабочем каталоге C:\work\alkamal\packer выполнены команды packer.exe init vagrant-rocky.pkr.hcl и packer.exe build vagrant-rocky.pkr.hcl, в результате чего произведена автоматическая установка Rocky Linux в VirtualBox и сформирован box-файл ([рис. 14](#fig-12)). По завершении сборки в каталоге создан файл vagrant-virtualbox-rocky-9-x86\_64.box, подтверждающий корректное формирование образа для Vagrant ([рис. 15](#fig-13)).

|  |
| --- |
| Рисунок 14: Выполнение команд packer.exe init и packer.exe build |

|  |
| --- |
| Рисунок 15: Сформированный box-файл vagrant-virtualbox-rocky-9-x86\_64.box |

Далее выполнена регистрация образа в Vagrant командой vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86\_64.box, после чего box добавлен под именем rocky9 ([рис. 16](#fig-14)). Запуск виртуальной машины Server осуществлён командой vagrant up server, выполнена инициализация сетевых интерфейсов, проброс портов и запуск provisioning-сценариев ([рис. 17](#fig-15)).

|  |
| --- |
| Рисунок 16: Регистрация box-файла в Vagrant |

|  |
| --- |
| Рисунок 17: Запуск виртуальной машины server через vagrant up |

Запуск виртуальной машины Client выполнен командой vagrant up client, произведена настройка сетевых адаптеров и подключение через SSH ([рис. 18](#fig-16)). В графическом интерфейсе VirtualBox обе виртуальные машины успешно загружены, выполнен вход под пользователем vagrant с паролем vagrant ([рис. 19](#fig-17), [рис. 20](#fig-17-1)).

|  |
| --- |
| Рисунок 18: Запуск виртуальной машины client через vagrant up |

|  |
| --- |
| Рисунок 19: Графический вход в систему на виртуальной машине server |

|  |
| --- |
| Рисунок 20: Графический вход в систему на виртуальной машине client |

Подключение к серверу выполнено командой vagrant ssh server, после ввода пароля осуществлён переход к пользователю alkamal командой su - alkamal и последующий выход из сеанса ([рис. 21](#fig-18)). Аналогичные действия выполнены для клиента с использованием команды vagrant ssh client ([рис. 22](#fig-18-1)).

|  |
| --- |
| Рисунок 21: Подключение по SSH к server и переход к пользователю alkamal |

|  |
| --- |
| Рисунок 22: Подключение по SSH к client и переход к пользователю alkamal |

Завершение работы лабораторного стенда выполнено командами vagrant halt server и vagrant halt client, что обеспечило корректное выключение обеих виртуальных машин ([рис. 23](#fig-19)).

|  |
| --- |
| Рисунок 23: Остановка виртуальных машин server и client |

# 4. Внесение изменений в настройки внутреннего окружения виртуальной машины

В конфигурационном файле Vagrantfile перед разделом с описанием виртуальной машины server добавлены provisioning-блоки common user и common hostname, обеспечивающие выполнение скриптов 01-user.sh и 01-hostname.sh при загрузке ВМ ([рис. 24](#fig-20)). Это гарантирует создание пользователя alkamal и установку доменного имени вида \*.alkamal.net.

|  |
| --- |
| Рисунок 24: Фрагмент Vagrantfile с блоками common user и common hostname |

Для применения изменений выполнены команды vagrant up server --provision и vagrant up client --provision, в результате чего повторно запущены provisioning-сценарии и зафиксированы внутренние настройки виртуальных машин ([рис. 25](#fig-21), [рис. 26](#fig-22)). В процессе выполнения на сервере зафиксировано существование пользователя alkamal, что подтверждает корректную работу скрипта создания пользователя.

|  |
| --- |
| Рисунок 25: Повторный запуск server с ключом –provision |

|  |
| --- |
| Рисунок 26: Повторный запуск client с ключом –provision |

После применения настроек выполнен вход в графическом интерфейсе под пользователем alkamal на сервере и клиенте ([рис. 27](#fig-23), [рис. 28](#fig-23-1)). При подключении по SSH командой vagrant ssh выполнен переход к пользователю alkamal через su - alkamal, при этом приглашение терминала отображается в формате alkamal@server.alkamal.net и alkamal@client.alkamal.net, что подтверждает корректную настройку hostname и пользовательского окружения ([рис. 29](#fig-24)).

|  |
| --- |
| Рисунок 27: Графический вход под пользователем alkamal на сервере |

|  |
| --- |
| Рисунок 28: Графический вход под пользователем alkamal на клиенте |

|  |
| --- |
| Рисунок 29: Проверка SSH-подключения и отображения приглашения пользователя |

# 5. Выводы

В ходе работы выполнена автоматическая сборка box-файла Rocky Linux с использованием Packer и его регистрация в Vagrant. Развёрнуты виртуальные машины server и client, применены provisioning-скрипты для создания пользователя и настройки hostname. Подтверждена корректная работа SSH-доступа и сетевых параметров. Лабораторный стенд в ОС Windows успешно подготовлен к дальнейшей работе.

# 6. 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 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.

# 7. Список литературы

1. GNU Bash Manual. — 2019. — URL: https://www.gnu.org/software/bash/manual/ (дата обр. 13.09.2021).
2. GNU Make Manual. — 2016. — URL: http://www.gnu.org/software/make/manual/ (visited on 09/13/2021).
3. Powers S. *Vagrant Simplified [Просто о Vagrant]* / Пер.: А. Панин // Библиотека сайта ruslinux.net. — 2015. — URL: http://rus-linux.net/MyLDP/vm/vagrant-simplified.html (visited on 09/13/2021).
4. Vagrant Documentation. — URL: https://www.vagrantup.com/docs (visited on 09/13/2021).
5. Купер М. *Искусство программирования на языке сценариев командной оболочки*. — 2004. — URL: https://www.opennet.ru/docs/RUS/bash\_scripting\_guide/ (дата обр. 13.09.2021).