Отчёт по лабораторной работе №1

Сетевые технологии

Мурашов Иван Вячеславович

Содержание

Список иллюстраций

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 (см. раздел 1.5.4).
4. Скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, вы можете попробовать развернуть виртуальные машины на другом компьютере.

# 3. Выполнение лабораторной работы

## 3.1 Подготовка и запуск виртуальных машин

Перед началом работы с Vagrant создаю каталог для проекта и размещаю необходимые файлы в директории packer и vagrant ([рис. 1](#fig-001)).

|  |
| --- |
| Рисунок 1: Каталог packer |

Затем, поскольку я работаю на ОС Windows, используя FAR, перехожу в созданный мной рабочий каталог с проектом ([рис. 2](#fig-002)).

|  |
| --- |
| Рисунок 2: Менеджер FAR |

В командной строке ввожу

packer.exe init vagrant-rocky.pkr.hcl  
packer.exe build vagrant-rocky.pkr.hcl

для начала автоматической установки образа операционной системы Rocky Linux в VirtualBox и последующего формирования box-файла с дистрибутивом Rocky Linux для VirtualBox ([рис. 3](#fig-003)).

|  |
| --- |
| Рисунок 3: Установка образа Rocky Linux |

|  |
| --- |
| Рисунок 4: Установка образа Rocky Linux |

По окончании процесса в рабочем каталоге видим box-файл с названием vagrant-virtualbox-rocky-9-x86\_64.box ([рис. 5](#fig-006)).

|  |
| --- |
| Рисунок 5: Каталог packer |

Для регистрации образа виртуальной машины в vagrant в командной строке ввожу

vagrant box add rocky9 vagrant-virtualbox-rocky-9-x86\_64.box

|  |
| --- |
| Рисунок 6: Регистрация образа Rocky Linux |

Для запуска виртуальной машины Server ввожу в консоли

vagrant up server

|  |
| --- |
| Рисунок 7: Запуск vm-server |

Для запуска виртуальной машины Client ввожу в консоли

vagrant up client

Убеждаюсь, что запуск обеих виртуальных машин прошёл успешно и логинюсь под пользователем vagrant с паролем vagrant в графическом окружении ([рис. 8](#fig-009)), ([рис. 9](#fig-011)).

|  |
| --- |
| Рисунок 8: VM server и VM client |

|  |
| --- |
| Рисунок 9: VM server |

Подключитесь к серверу из консоли:

vagrant ssh server

|  |
| --- |
| Рисунок 10: Подключение к серверу из консоли |

Перехожу к пользователю ivmurashov ([рис. 11](#fig-012)).

|  |
| --- |
| Рисунок 11: Авторизация в терминале |

Далее выхожу и выключаю обе виртуальные машины:

vagrant halt server  
vagrant halt client

## 3.2 Внесение изменений в настройки внутреннего окружения виртуальной машины

Для отработки созданных скриптов во время загрузки виртуальных машин убеждаюсь, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется следующая запись ([рис. 12](#fig-013)).

|  |
| --- |
| Рисунок 12: Содержимое Vagrantfile |

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

vagrant up server --provision  
vagrant up client --provision

Логинюсь на сервере и клиенте под созданным пользователем. Убеждаюсь, что в терминале приглашение отображается в виде ivmurashov@server.ivmurashov.net на сервере ([рис. 13](#fig-014)).

|  |
| --- |
| Рисунок 13: Терминал vm-server |

Убеждаюсь, что в терминале клиента приглашение отображается в виде ivmurashov@client.ivmurashov.net ([рис. 14](#fig-015)).

|  |
| --- |
| Рисунок 14: Терминал vm-client |

Затем, выключаю виртуальные машины.

## 3.3 Ответы на контрольные вопросы

1. Для чего предназначен Vagrant?

Vagrant нужен для создания и управления средами виртуальных машин в одном рабочем процессе. Этот инструмент, по сути, позволяет автоматизировать процесс установки на виртуальную машину как основного дистрибутива операционной системы, так и настройки необходимого в дальнейшем программного обеспечения.

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

Box-файл (или Vagrant Box) — это сохранённый образ виртуальной машины с развёрнутой в ней ОС; по сути, box-файл используется как основа для клонирования виртуальных машин с теми или иными настройками.

Vagrantfile — это конфигурационный файл, написанный на языке Ruby, в котором указаны настройки запуска виртуальной машины.

1. Приведите описание и примеры вызова основных команд Vagrant.

С Vagrant можно работать, используя следующие основные команды: – 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.