Лабораторная работа №1

Подготовка лабораторного стенда

Беличева Дарья Михайловна

Содержание

# 1 Цель работы

Приобрести практические навыки 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.

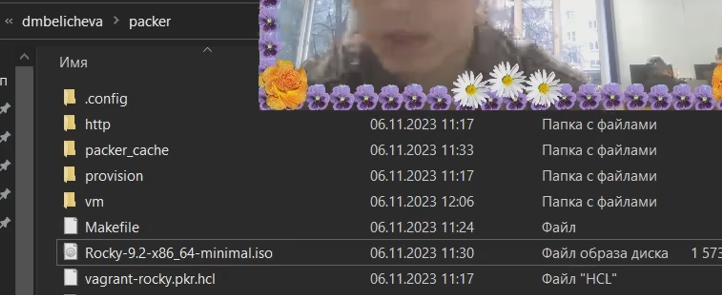
# 3 Выполнение лабораторной работы

1. В ОС Windows создалим каталог для проекта.

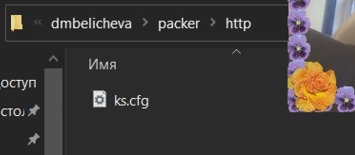
В созданном рабочем каталоге разместим образ варианта операционной системы Rocky Linux (в этом практикуме будем использовать Rocky-9.2- x86\_64-minimal.iso — минимальный дистрибутив Rocky Linux).

В этом же каталоге разместим подготовленные заранее для работы с Vagrant файлы (рис. ??):

* vagrant-rocky.pkr.hcl
* ks.cfg (файл должен быть расположен в подкаталоге http)
* Vagrantfile
* Makefile

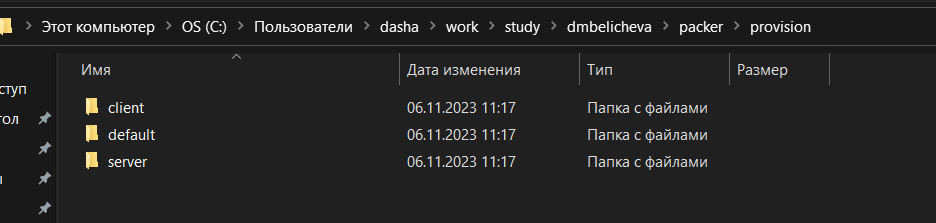


Содержимое каталога packer



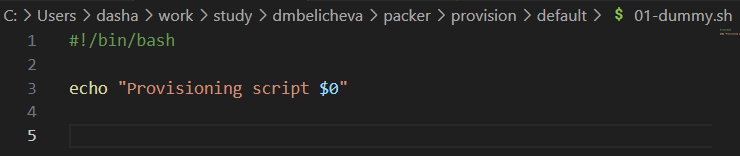
Содержимое подкаталога http

1. В этом же каталоге создадим 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.



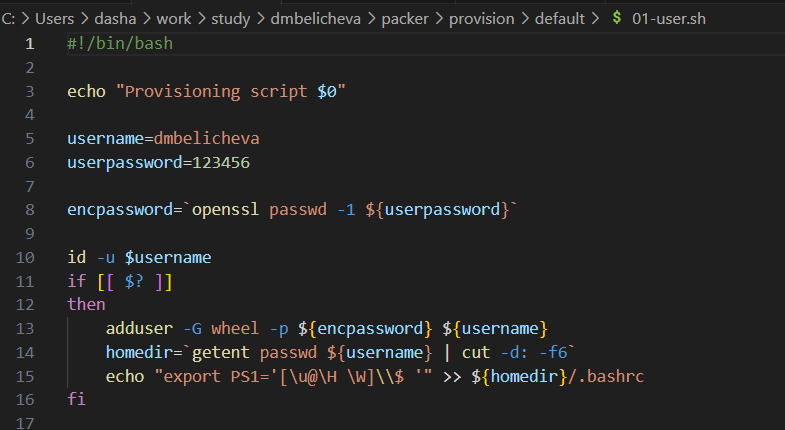
Содержимое каталога provision

1. В каталогах default, server и client разместим заранее подготовленный скриптзаглушку 01-dummy.sh:



Содержимое файла 01-dummy.sh

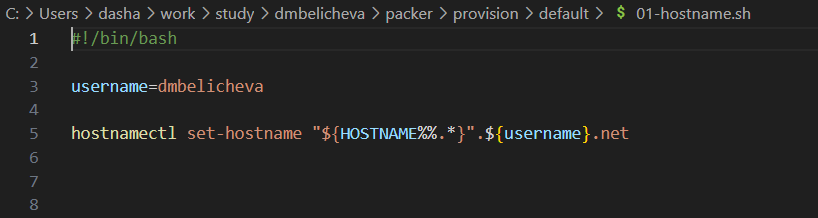
1. В каталоге default разместим заранее подготовленный скрипт 01-user.sh по изменению названия виртуальной машины:



Содержимое файла 01-user.sh

В этом скрипте в качестве значения переменной username вместо user укажем имя пользователя, совпадающее с моим логином, т.е. dmbelicheva.

1. В каталоге default разместите заранее подготовленный скрипт 01-hostname.sh по изменению названия виртуальной машины:



Содержимое файла 01-hostname.sh

В этом скрипте в качестве значения переменной username вместо user укажим имя пользователя, совпадающее с вашим логином, т.е. dmbelicheva.

**Развёртывание лабораторного стенда на ОС Linux**

Я установила MSYS2 (сборка пакетов для Windows, которая позволяет использовать многие утилиты и приложения, которые обычно доступны только в Unix-подобных операционных системах), поэтому буду использовать команды для Linux.

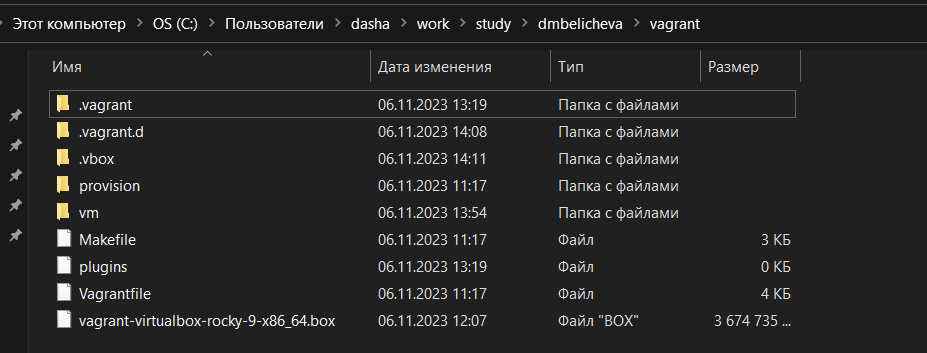
1. Перейдем в каталог с проектом:

cd C:\Users\dasha\work\study\dmbelicheva\packer\

1. Для формирования box-файла с дистрибутивом Rocky Linux для VirtualBox в терминале наберем:

make box

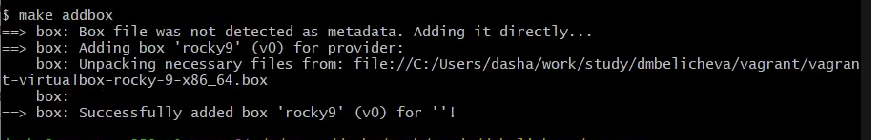
Начнётся процесс скачивания, распаковки и установки драйверов VirtualBox и дистрибутива ОС на виртуальную машину. После завершения процесса автоматического развёртывания образа виртуальной машины в каталоге C:\Users\dasha\work\study\dmbelicheva\vagrant\ временно появится каталог builds с промежуточными файлами .vdi, .vmdk и .ovf, которые затем автоматически будут преобразованы в box-файл сформированного образа: vagrant-virtualbox-rocky9-x86\_64.box.



Появление box-файла

1. Для регистрации образа виртуальной машины в Vagrant в терминале в каталоге C:\Users\dasha\work\study\dmbelicheva\vagrant\ наберем

make addbox

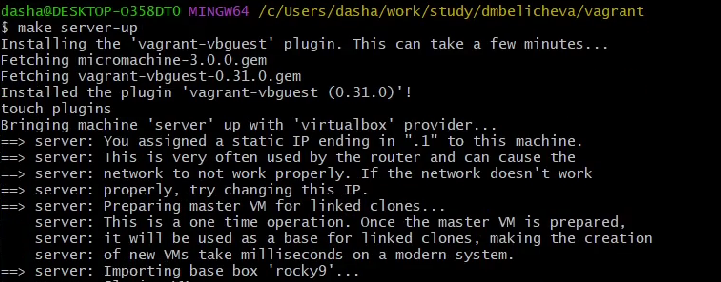


Команда make addbox

Это позволит на основе конфигурации, прописанной в файле Vagrantfile, сформировать box-файлы образов двух виртуальных машин - сервера и клиента с возможностью их параллельной или индивидуальной работы.

1. Запустим виртуальную машину Server, введя

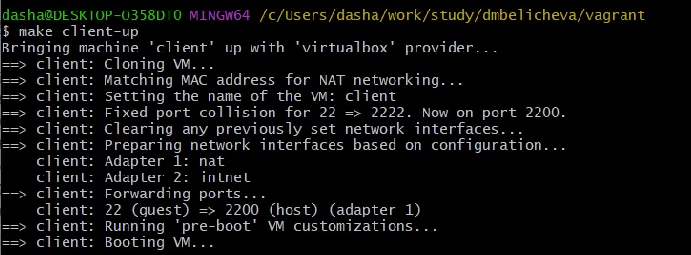
make server-up



Команда make server-up

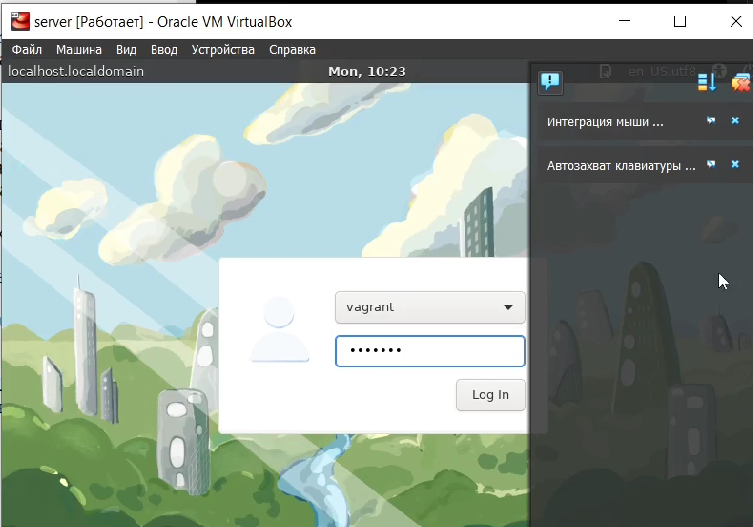
1. Запустим виртуальную машину Client, введя

make client-up

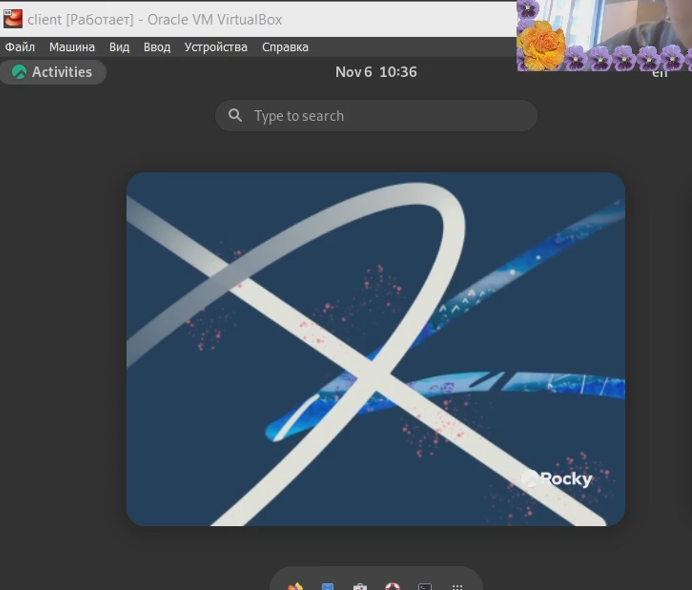


Команда make client-up

1. Убедимся, что запуск обеих виртуальных машин прошёл успешно, залогинемся под пользователем vagrant с паролем vagrant. Затем выключим обе виртуальные машины.



Окно server



Окно client

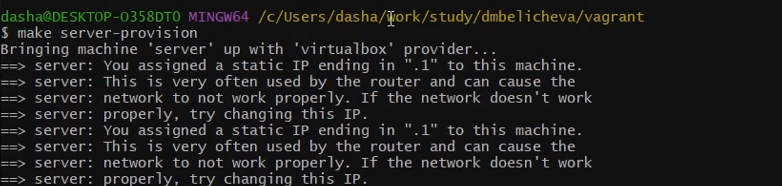
**Внесение изменений в настройки внутреннего окружения виртуальной машины**

1. 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"

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

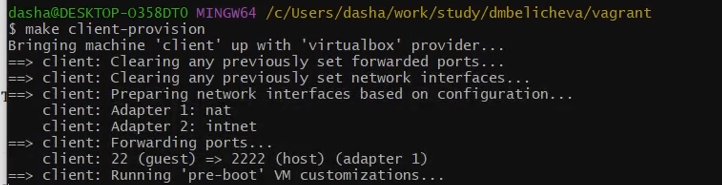
make server-provision



Команда make server-provision

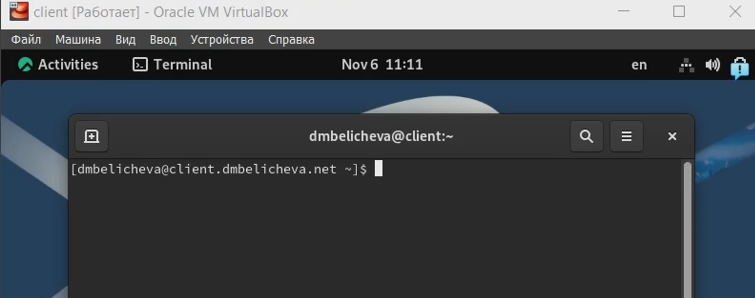
Затем

make client-provision

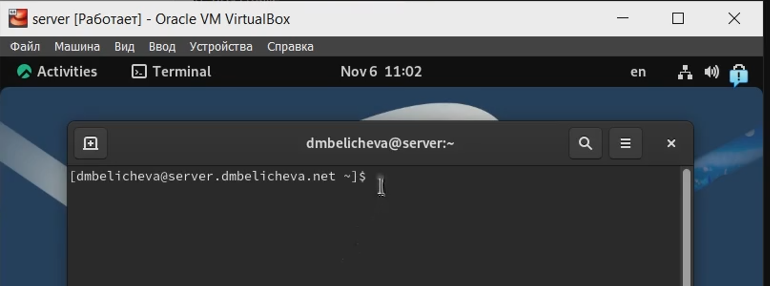


Команда make client-provision

1. Залогинемся на сервере и клиенте под созданным пользователем. Убедимся, что в терминале приглашение отображается в виде user@server.user.net на сервере и в виде user@client.user.net на клиенте, где вместо user указан мой логин - dmbelicheva.



Терминал client



Терминал server

1. Выключим виртуальные машины.

# 4 Выводы

В процессе выполнения данной лабораторной я приобрела практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

# 5 Контрольные вопросы

1. Для чего предназначен Vagrant? Инструмент для создания и управления средами виртуальных машин в одном рабочем процессе.
2. Что такое box-файл? В чём назначение Vagrantfile? box-файл (или Vagrant Box) — сохранённый образ виртуальной машины с развёрнутой в ней операционной системой; по сути, 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 отVagrant 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.

1. Дайте построчные пояснения содержания файлов vagrant-rocky.pkr.hcl, ks.cfg, Vagrantfile, Makefile.

Пример содержимого файла Vagrantfile:

# -\*- mode: ruby -\*-  
# vi: set ft=ruby :  
Vagrant.configure(2) do |config|  
config.vm.box = "BOX\_NAME"  
config.vm.hostname = "HOST\_NAME"  
config.vm.network "private\_network", ip: "192.168.1.1"  
config.vm.define "VM\_NAME"  
config.vm.provider "virtualbox" do |vb|  
vb.gui = false  
vb.memory = "1024"  
end  
end

Первые две строки указывают на режим работы с Vagrantfile и использование языка Ruby. Затем идёт цикл do, заменяющий конструкцию Vagrant.configure далее по текстуна 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.