РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе № 1

дисциплина: Администрирование сетевых подсистем

Студент:

Группа:

**МОСКВА**

2022 г.

# Цель работы:

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

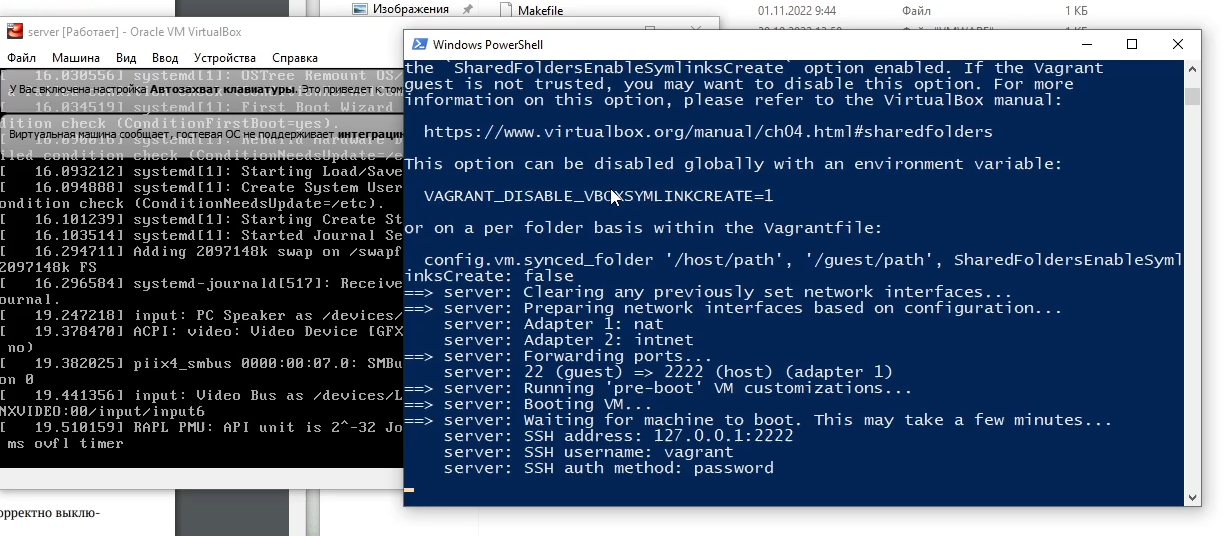
# Ход работы:

## Подготовка

1. Перед началом работы с Vagrant создаю каталог для проекта.
2. В созданном рабочем каталоге размещаю образ операционной системы
3. В этом же каталоге размещаю подготовленные заранее для работы с Vagrant файлы:

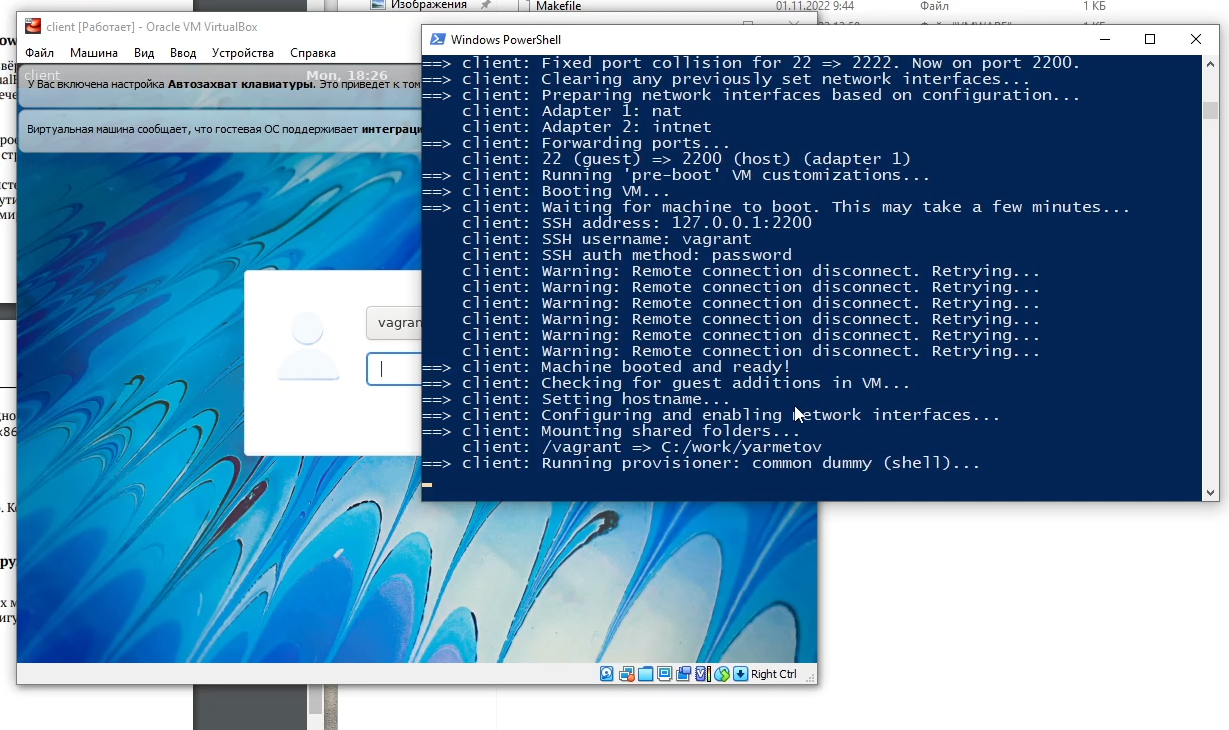
## Развёртывание лабораторного стенда на ОС Windows

1. Для регистрации образа виртуальной машины в Vagrant в терминале набираю vagrant box add
2. Запускаю виртуальную машину Server,



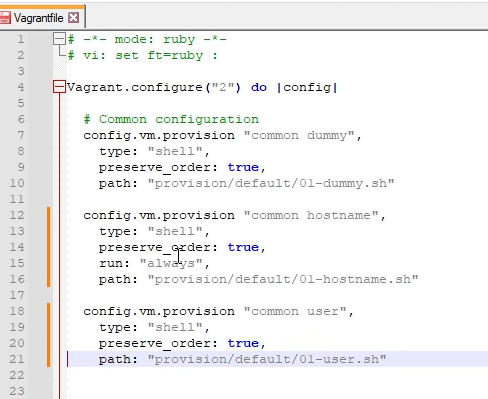
1. Запускаю виртуальную машину Client.
2. Запуск обеих виртуальных машин прошёл успешно.

Затем выключаю обе виртуальные машины.



## Внесение изменений в настройки внутреннего окружения виртуальной машины

* 1. Для отработки созданных скриптов во время загрузки виртуальных машин убедитесь, что в конфигурационном файле Vagrantfile до строк с конфигурацией сервера имеется следующая запись:

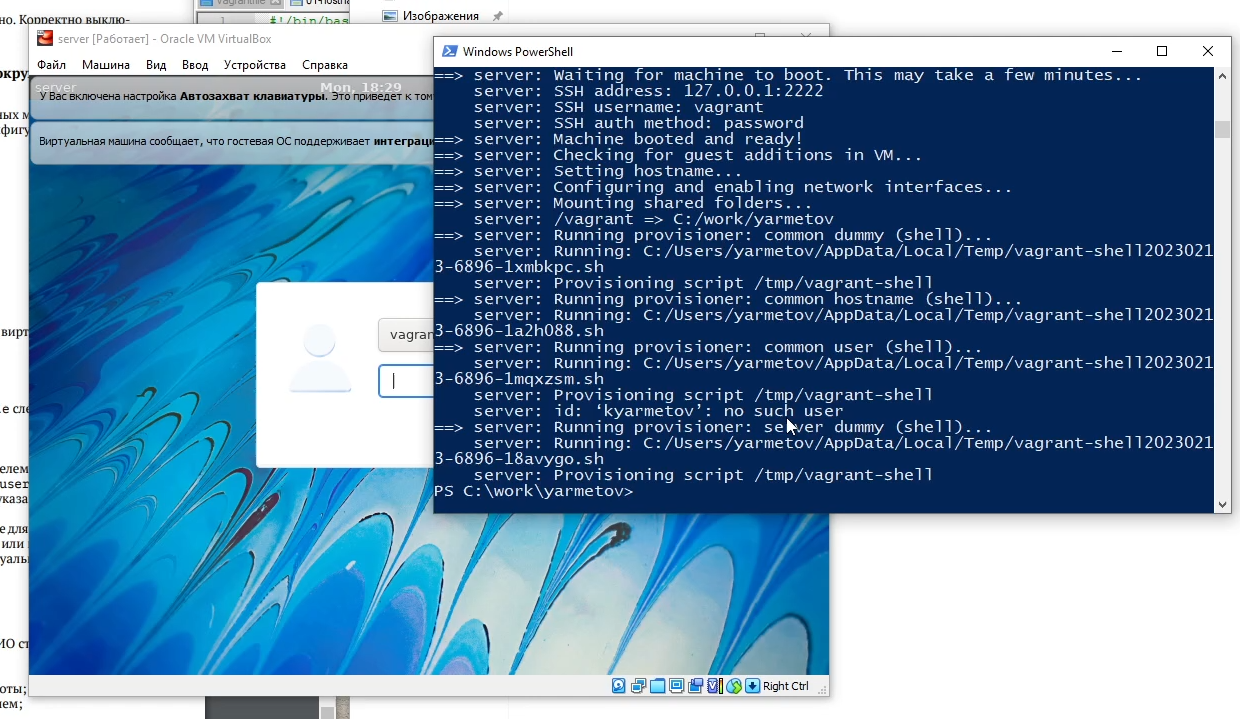


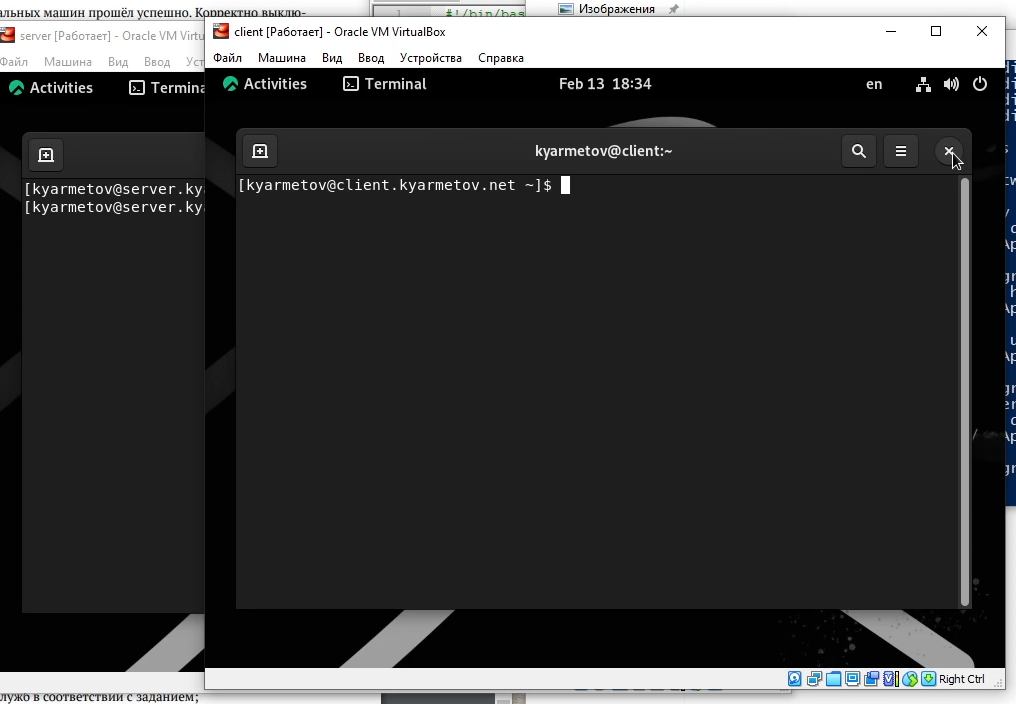
* 1. Зафиксирую внесённые изменения для внутренних настроек виртуальных машин, введя в терминале:

vagrant up server --provision

vagrant up client –provision

* 1. Логинюсь на машинах и убеждаюсь, что имя пользователя указано корректно





# Вывод:

Я приобрел практические навыки установки CentOS на виртуальную машину с помощью инструмента Vagrant.

# Ответы на контрольные вопросы:

1. Для чего предназначен Vagrant

Vagrant — представляет собой инструмент 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.

Vagrant является интерфейсом для базовой программы виртуализации. По умолчанию базовая программа - VirtualBox.  
Vagrant не привязан к конкретной системе. При переносе файлов на другой компьютер он просто будет использовать копию VirtualBox, которую вы установили на новом компьютере.

1. 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.

1. Приведите описание и примеры вызова основных команд Vagrant.
2. vagrant help — вызов справки по командам Vagrant;
3. vagrant box list — список подключённых к Vagrant box-файлов;
4. vagrant box add — подключение box-файла к Vagrant;
5. vagrant destroy — отключение box-файла отVagrant и удаление его из виртуального окружения;
6. vagrant init — создание «шаблонного» конфигурационного файла Vagrantfile для его последующего изменения;
7. vagrant up — запуск виртуальной машины с использование инструкций по запуску из конфигурационного файла Vagrantfile;
8. vagrant reload — перезагрузка виртуальной машины;
9. vagrant halt — остановка и выключение виртуальной машины;
10. vagrant provision — настройка внутреннего окружения имеющейся виртуальной машины (например, добавление новых инструкций (скриптов) в ранее созданную виртуальную машину);
11. vagrant ssh— подключение к виртуальной машине через ssh.
12. Дайте построчные пояснения содержания файлов

vagrant-centos.json ks.cfg Vagrantfile Makefile

1. Содержание файла vagrant-centos.json:

{#Задаются основные переменные

**"variables":** {

#задаётся образ оптического диска

**"iso\_url":** "CentOS-7-x86\_64-Minimal-1810.iso",

#чек-сумма для проверки целостности данных

**"iso\_checksum":** "714acc0aefb32b7d51b515e25546835e55a90da

↪ 9fb00417fbee2d03a62801efd",

#тип чек-суммы, т.е. каким алгоритмом было рассчитано вышеприведённое значение

**"iso\_checksum\_type":** "sha256",

#версия системы

**"redhat\_release":** "7",

#версия платформы

**"redhat\_platform":** "x86\_64",

# Описание дистрибутива

**"artifact\_description":** "CentOS 7.5 (build 1810)",

# Версия дистрибутива

**"artifact\_version":** "7.5.1804",

#Поскольку Vagrant при оперировании образом диска

#подсоединяется как пользователь, ему необходимы логин и

#пароль. Две нижеприведённые строки задают их.

**"ssh\_username":** "vagrant",

**"ssh\_password":** "vagrant",

# размер в мегабайтах жесткого диска для создания

# виртуальной машины

**"disk\_size":** "40960"

},

#настройки сборки виртуальной машины

**"builders":** [

{ #имя машины

**"name":** "centos-{{user `redhat\_release`}}",

#тип iso - iso для виртуальной машина

**"type":** "virtualbox-iso",

# имя файла OVF для новой виртуальной машины

**"vm\_name":** "packer-centos-vm",

#время ожидания начала загрузки

**"boot\_wait":** "10s",

#размер виртуального диска

**"disk\_size":** "{{user `disk\_size`}}",

#тип гостевой ОС

**"guest\_os\_type":** "RedHat\_64",

#каталог для http

**"http\_directory":** "http",

# URL-адрес ISO, содержащий установочный образ

**"iso\_url":** "{{user `iso\_url`}}",

#чек-сумма для файла ISO ОС

**"iso\_checksum":** "{{user `iso\_checksum`}}",

# тип контрольной суммы, указанной в iso\_checksum.

**"iso\_checksum\_type":** "{{user `iso\_checksum\_type`}}",

# Путь на гостевой виртуальной машине, в который будет

# загружен гостевой вариант VirtualBox ISO.

**"guest\_additions\_path":** "VBoxGuestAdditions.iso",

#Команда загрузки/установки ОС

**"boot\_command":** [

# Имитация нажатия клавиши escape.

"<esc>",

#Добавляет паузу 1 c

"<wait><esc><esc>",

# Путь к файлу кикстарта и изменение имени сетевого

# интерфейса

"linux inst.ks=http://{{.HTTPIP}}:{{.HTTPPort}}/

↪ ks.cfg biosdevname=0

↪ net.ifnames=0",

# Имитация нажатия клавиши enter.

"<enter>"

],

# Команда, используемая для «вежливого» завершения

# работы машины, как только она будет настроена

**"shutdown\_command":** "sudo -S /sbin/halt -h -p",

# количество времени ожидания после выполнения

# shutdown\_command

**"shutdown\_timeout":** "5m",

# Установка таймаута для соединений SSH

**"ssh\_wait\_timeout":** "15m",

# Настройка аутентификации

**"ssh\_username":** "{{user `ssh\_username`}}",

**"ssh\_password":** "{{user `ssh\_password`}}",

# TCP-порт для ssh

**"ssh\_port":** 22,

# Подключение интерактивной оболочки

**"ssh\_pty":** true,

# Это путь к каталогу, в котором будет создана

# результирующая виртуальная машина

**"output\_directory":** "builds",

# пользовательские VBoxManage команды для выполнения для

# дальнейшей настройки создаваемой виртуальной машины

# Объём оперативной памяти – 1024 Мб

# Кол-во виртуальных ядер процессора(CPU) - 1

**"vboxmanage":** [

[ "modifyvm", "{{.Name}}", "--memory", "1024"],

[ "modifyvm", "{{.Name}}", "--cpus", "1" ]

],

# тип контроллера, к которому прикреплен основной

# жесткий диск

**"hard\_drive\_interface":** "sata",

# Путь в виртуальной машине для загрузки файла, который # содержит версию VirtualBox, которая была использована # для создания машины

**"virtualbox\_version\_file":** ".vbox\_version",

# дополнительные параметры для перехода к экспорту

# VBoxManage

**"export\_opts":**

[ # файл, содержащий метаданные для группы сопутствующих

# файлов

"—manifest",

# Параметр экспорта c номером виртуальной системы

# для изменения настроек VM

"--vsys", "0",

# Описание дистрибутива

"--description", "{{user`artifact\_description`}}",

# Версия

"--version", "{{user `artifact\_version`}}"

]

}

],

**"post-processors":** [

{#генерирование box-файла

"output": "vagrant-centos-{{user`redhat\_release`}}-

↪ {{user`redhat\_platform`}}.box",

#степень сжатия

**"compression\_level":** "6",

#тип box-файла

**"type":** "vagrant"

}

],

# установка программного обеспечения с помощью системы

# развёртывания shall. Будет сгенерирован shall файл, который

# будет передан внутрь образа

**"provisioners":** [{

**"type":** "shell",

**"inline":** [

"sleep 30",

"sudo yum -y install deltarpm",

"sudo yum -y install epel-release",

"sudo yum -y groups mark convert",

"sudo yum -y groupinstall 'Development Tools'",

"sudo yum -y install kernel-devel",

"sudo yum -y install dkms",

"sudo mkdir /tmp/vboxguest",

"sudo mount -t iso9660 -o

↪ loop/home/vagrant/VboxGuestAdditions.iso/tmp

↪ /vboxguest",

"cd /tmp/vboxguest",

"sudo ./VBoxLinuxAdditions.run",

"cd /tmp",

"sudo umount /tmp/vboxguest",

"sudo rmdir /tmp/vboxguest",

"rm /home/vagrant/VboxGuestAdditions.iso",

"sudo yum -y groupinstall 'Server with GUI'",

"sudo yum install -y mc htop tmux",

"sudo systemctl set-default graphical.target",

"echo Image Provisioned!"

]

}]

}

1. Cодержание файла ks.cfg

install

# Использовать установочный носитель CDROM

сdrom

# Задаётся язык системы

lang ru\_RU.UTF-8

# Задаются раскладки клавиатуры

keyboard --xlayouts='us,ru'

# Системный часовой пояс

timezone --utc Etc/UTC

# Сетевая информация. –onboot yes - интерфейс включается автоматически при включении сервера.

#--bootproto=dhcp – работает по dhcp.

#--device=eth0 - название интерфейса, который будет получать

# адрес по dhcp

#--activate--noipv6 ipv6 не используется

network --onboot yes --bootproto=dhcp --device=eth0

↪ --activate--noipv6

# Интерфейс для настройки ресурсов аутентификации системы.

# Параметр --enableshadow используются для настройки функций

# аутентификации через /etc/pam.d/system-auth

# Алгоритм, используемый для хранения новых хэшей паролей,

# задан параметром --passalgo

authconfig --enableshadow --passalgo=sha512

# Пароль суперпользователя - vagrant

rootpw vagrant

# Создание пользователя Vagrant

# Он имеет пароль vagrant и принадлежит группе vagrant

user --name=vagrant --groups=vagrant —password=vagrant

# Добавление сетевых сервисов

services —enabled=NetworkManager,sshd,chronyd

# Отключить запуск установочного интерфейса при первой

# загрузке

firstboot —disabled

# Конфигурация загрузчика системы.

bootloader --location=mbr

text

skipx

# Настройка генерируемого после входа сообщения на info

logging --level=info

#Удаление всех разделов на всех дисках, известных установщику

zerombr

# Информация для очистки раздела

clearpart --all –initlabel

# автоматическая разметка

autopart

# пересобрать

reboot

# не устанавливать группу @Base

%packages --nobase

@Core

# Установка OpenSSH для клиента и сервера

openssh-clients

openssh-server

%end

%post –erroronfail

# Обновление софта

yum -y update

# Добавление пользователя vagrant к суперпользователям

cat > /etc/sudoers.d/vagrant<< EOF\_sudoers\_vagrant

vagrant ALL=(ALL) NOPASSWD: ALL

EOF\_sudoers\_vagrant

/bin/chmod 0440 /etc/sudoers.d/vagrant

/bin/sed -i "s/^.\*requiretty/#Defaults requiretty/"

↪ /etc/sudoers

# Исправиление конфигурации sshd для CentOS 7 и работы с

# Vagrant

cat >> /etc/ssh/sshd\_config << EOF\_sshd\_config

TCPKeepAlive yes # поддержка соединения в активном состоянии

ClientAliveInterval 0 # время простоя клиента в секундах

ClientAliveCountMax 3 # количество проверок доступности

# клиента

EOF\_sshd\_config

%end

1. Cодержание файла Vagrantfile

*# -\*- mode: ruby -\*-*

*# vi: set ft=ruby :*

*# код представлен на языке ruby*

Vagrant.configure("2") **do** |config|

# Общая конфигурация проекта  
 # Подключение дополнительных скриптов к обеспечению.

# Каждый provision имеет оригинальное имя.

config.vm.provision "common user",

# Тип провизора- shell.

type: "shell",

preserve\_order: **true**,

# Путь к файлу пост-конфигурации

path: "provision/default/01-user.sh"

# Имя provision - common hostname.

config.vm.provision "common hostname",

# Тип провизора- shell.

type: "shell",

preserve\_order: **true**,

# Запуск производится всегда вне зависимости от запуска

# машины с опцией переконфигурирования

run: "always",

# Путь к файлу пост-конфигурации

path: "provision/default/01-hostname.sh"

# Конфигурация машины server

config.vm.define "server", autostart: **false do** |server|

# VM server базируется на VM centos7

server.vm.box = "centos7"

# Имя хоста VM - server

server.vm.hostname = 'server'

# Vagrant автоматически вставляет ключ для

# использования в SSH, заменяя ключ безопасности Vagrant

# по умолчанию в машине, если он обнаружен.

# В данном случае, из-за false этого не происходит

server.ssh.insert\_key = **false**

# Логин, который Vagrant будет использовать для

# аутентификации пользователя SSH

server.ssh.username = 'vagrant'

# Пароль, который Vagrant будет использовать для

# аутентификации пользователя SSH

server.ssh.password = 'vagrant'

# Второй интерфейс как внутренняя сеть для связи server и

# client. У server изначально установлен ip-адрес.

server.vm.network :private\_network, ip: "192.168.1.1",

↪ virtualbox\_\_intnet: **true**

# Настройка конфигурации, зависящей от провайдера,

# которая используется для изменения его индивидуальных

# параметров.

server.vm.provider :virtualbox **do** |v|

# возможность запуска сразу несколько VM из одного файл-

# диска.

v.linked\_clone = **true**

# Использование разрешения хоста в качестве прокси-сервера # DNS в режиме NAT.

v.customize ["modifyvm", :id,

↪ "--natdnshostresolver1","on"]

# Настройка объема памяти на виртуальной машине

v.memory = 1024

v.cpus = 1 # 1 виртуальное ядро

v.name = "server" # имя провайдера

# Отображение графического интерфейса VirtualBox при

# загрузке машины

v.gui = **true**

# Изменение объёма видеопамяти 12Mb

v.customize ["modifyvm", :id, "--vram", "12"]

**end**

**end**

# Конфигурация client

config.vm.define "client", autostart: **false do** |client|

# VM client базируется на VM centos7

client.vm.box= "centos7"

# Имя хоста VM - client

client.vm.hostname = 'client'

# Замена ключа безопасность для использования в SSH

# не производится

client.ssh.insert\_key = **false**

# Логин, который Vagrant будет использовать для

# аутентификации пользователя SSH

client.ssh.username = 'vagrant'

# Пароль, который Vagrant будет использовать для

# аутентификации пользователя SSH

client.ssh.password = 'vagrant'

# Внутренняя сеть для связи server и client.

client.vm.network :private\_network, type: "dhcp",

↪ virtualbox\_\_intnet: **true**

# Настройка конфигурации, зависящей от провайдера,

# которая используется для изменения его индивидуальных

# параметров.

client.vm.provider :virtualbox **do** |v|

# возможность запуска сразу несколько VM из одного файл-

# диска.

v.linked\_clone = **true**

# Использование разрешения хоста в качестве прокси-сервера # DNS в режиме NAT.

v.customize ["modifyvm", :id,

↪ "--natdnshostresolver1","on"]

# Настройка объема памяти на виртуальной машине

v.memory = 1024

v.cpus= 1 # 1 виртуальное ядро

v.name= "client" # имя провайдера

# Отображение графического интерфейса VirtualBox при

# загрузке машины

v.gui=true

# Объём видеопамяти 12Mb

v.customize ["modifyvm", :id, "--vram", "12"]

**end**

**end**

**end**

1. Cодержание файла Makefile

.PHONY: version

# Описание справки о Makefile

help:

@echo 'Usage:' ##вывод строки

@echo ' make <target>' ##вывод строки

@echo ##вывод пустой строки

@echo 'Targets:' ##вывод строки

## Поиск целей в Makefile, сортировка и вывод

@grep -E '^[a-zA-Z\_0-9.-]+:.\*?## .\*$$' $(MAKEFILE\_LIST)

↪ | sort | awk 'BEGIN {FS = ":.\*?## "};

↪ {printf "\033[36m%-30s\033[0m %s\n", $$1, $$2}'

@echo ##вывод пустой строки

all: box add2vagrant

box: ## Генереция box-файла из CentOS

## Создание и экспорт файла в текущий каталог

@export TMPDIR=`pwd`; packer build vagrant-centos.json

add2vagrant: ## Регистрация box в Vagrant

@export VAGRANT\_HOME =`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH =`pwd`/.vagrant; vagrant

↪ box add centos7 vagrant-centos-7-x86\_64.box

##ссылка на описание команды в контрольном вопросе №3(3)

## **VAGRANT\_HOME** - каталог, в котором Vagrant хранит

## глобальное состояние.

## **VAGRANT\_DOTFILE\_PATH** - каталог, в котором Vagrant хранит

## состояние, специфичное для VM, такое как Virtual UBID.

## **VBOX\_USER\_HOME** - директория пользователя, в которой

## VirtualBox сохраняет данные настроек пользователя.

## **VBOX\_INSTALL\_PATH** – каталог,в который будет установлена VM

## В **Makefile** во всех командах прописано изменение путей этих

## каталогов таким образом, чтобы они появились не в домашнем

## каталоге пользователя, а в текущем рабочем каталоге

## лабораторной работы.

up: ## Запустить server и client

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME =`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH =`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME =`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH =`pwd`/vm; vagrant up

##ссылка на описание команды в контрольном вопросе №3(6)

@VBoxManage setproperty machinefolder default

server: ## Запуск server

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant up server

##ссылка на описание команды в контрольном вопросе №3(6)

@VBoxManage setproperty machinefolder default

client: ## Запуск client

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant up client

##ссылка на описание команды в контрольном вопросе №3(6)

@VBoxManage setproperty machinefolder default

server-provision:

## Запуски и настройка внутреннего окружения server

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant up server

↪ --provision

##ссылка на описание команды в контрольном вопросе №3(9)

@VBoxManage setproperty machinefolder default

client-provision:

## Запуск и настройка внутреннего окружения client

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant up client

↪ --provision

##ссылка на описание команды в контрольном вопросе №3(9)

@VBoxManage setproperty machinefolder default

server-destroy: ## Отключение и удаление server

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant destroy server

##ссылка на описание команды в контрольном вопросе №3(4)

@VBoxManage setproperty machinefolder default

client-destroy: ## Отключение и удаление client

@VBoxManage setproperty machinefolder `pwd`/vm

-@export VAGRANT\_HOME=`pwd`/.vagrant.d; export

↪ VAGRANT\_DOTFILE\_PATH=`pwd`/.vagrant; export

↪ VBOX\_USER\_HOME=`pwd`/.vbox; export

↪ VBOX\_INSTALL\_PATH=`pwd`/vm; vagrant destroy client

##ссылка на описание команды в контрольном вопросе №3(4)

@VBoxManage setproperty machinefolder default