

TASK 2.2

ЧАСТИНА 1. РОБОТА З VIRTUALBOX

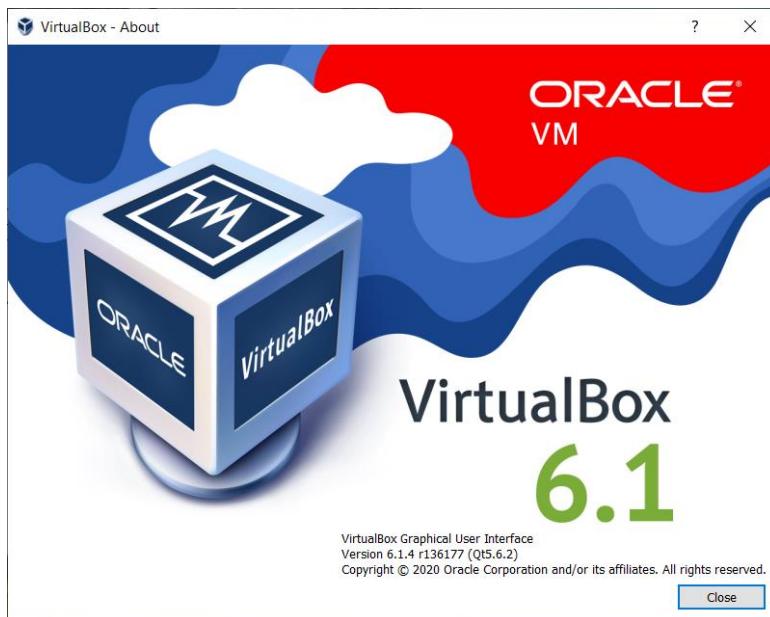
1. Перший запуск VirtualBox та віртуальної машини (VM).

1.1 Ознайомитись зі структурою керівництва користувача VirtualBox [1]

Done

1.2 З офіційного сайту VirtualBox [2] завантажити останню стабільну версію VirtualBox відповідно до хостової операційної системи (ОС), що встановлена на робочому місці студента. Для ОС Windows файл може називатися, наприклад, VirtualBox-6.0.12-133076-Win.exe. Провести інсталяцію VirtualBox.

I am going to use 6.1.4 r136177 because it is the last version



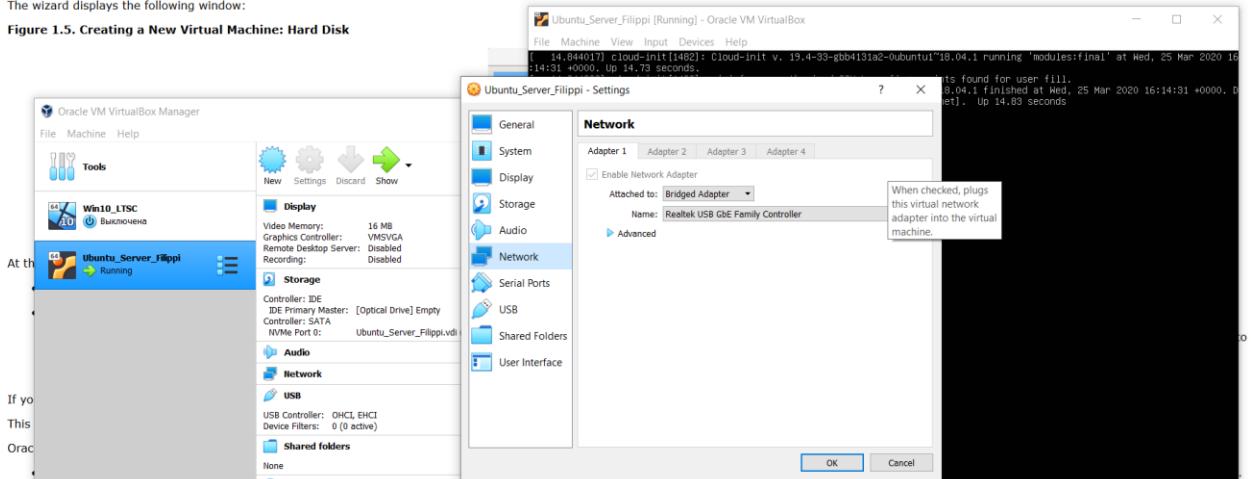
- 1.3 Завантажити з офіційного сайту останню стабільну версію образу ОС Ubuntu Desktop або Ubuntu Server [3].

File ubuntu-18.04.4-live-server-amd64.iso downloaded.

- 1.4 Створити VM1 та провести інсталяцію ОС Ubuntu користуючись інструкціями [1, п.1.7]. Ім'я машини задати як «ім'я хостової машини»_«прізвище студента»

The wizard displays the following window:

Figure 1.5. Creating a New Virtual Machine: Hard Disk



- A fixed-size file immediately occupies the file specified, even if only a fraction of the [file] is used.

For details about the differences, see [Section 5.2, "Disk Image Files \(VDI, VMDK, VHD, HDI\)](#).

To prevent your physical hard disk (host OS) from filling up, Oracle VM VirtualBox limits the size of the virtual hard disk to 2 TB. If you try to install, for a Windows or Linux guest, you will probably need several gigabytes for any swap space. `root@ubuntu:~# free -m`

Figure 1.6. Creating a New Virtual Machine: File Location and Size

1.5 Ознайомитись з можливостями керування VM1 – запуск, зупинка, перезавантаження, збереження стану, використання Host key та комбінацій клавіш, захват миші та ін. [1, п.1.8].

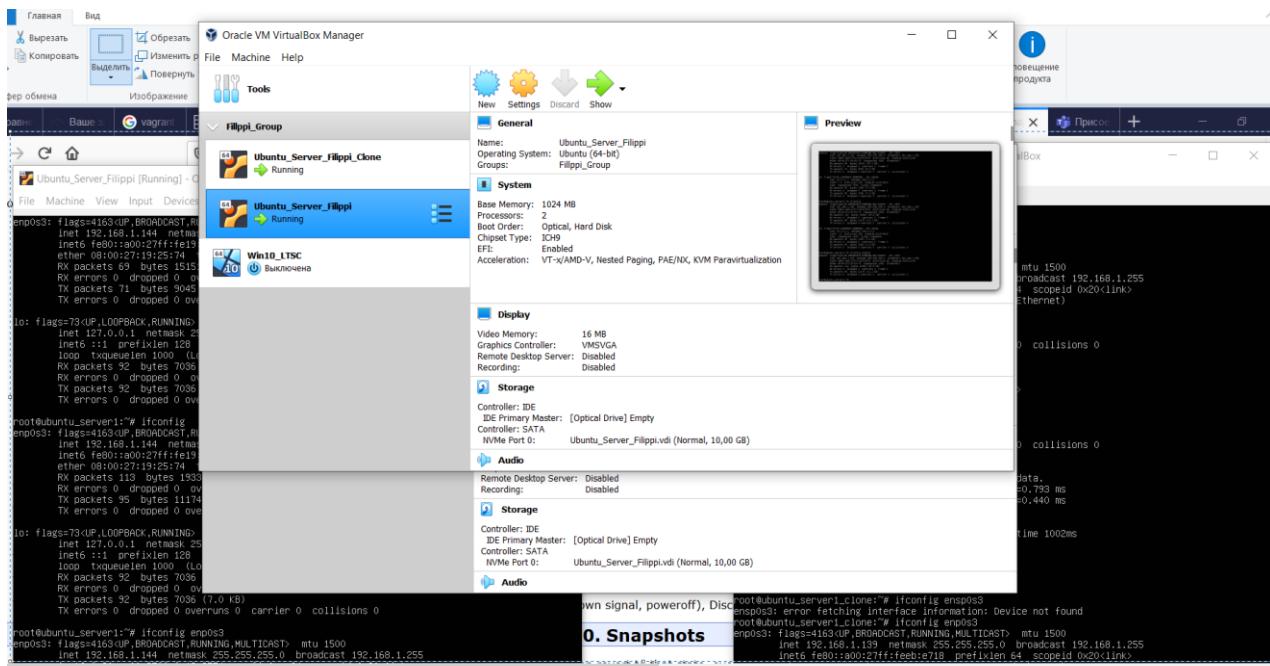
Done. Did Pause, save state, Host +Del, Host + F (full screen)

1.6 Клонувати існуючу VM1, створивши VM2 [1, п.1.13].

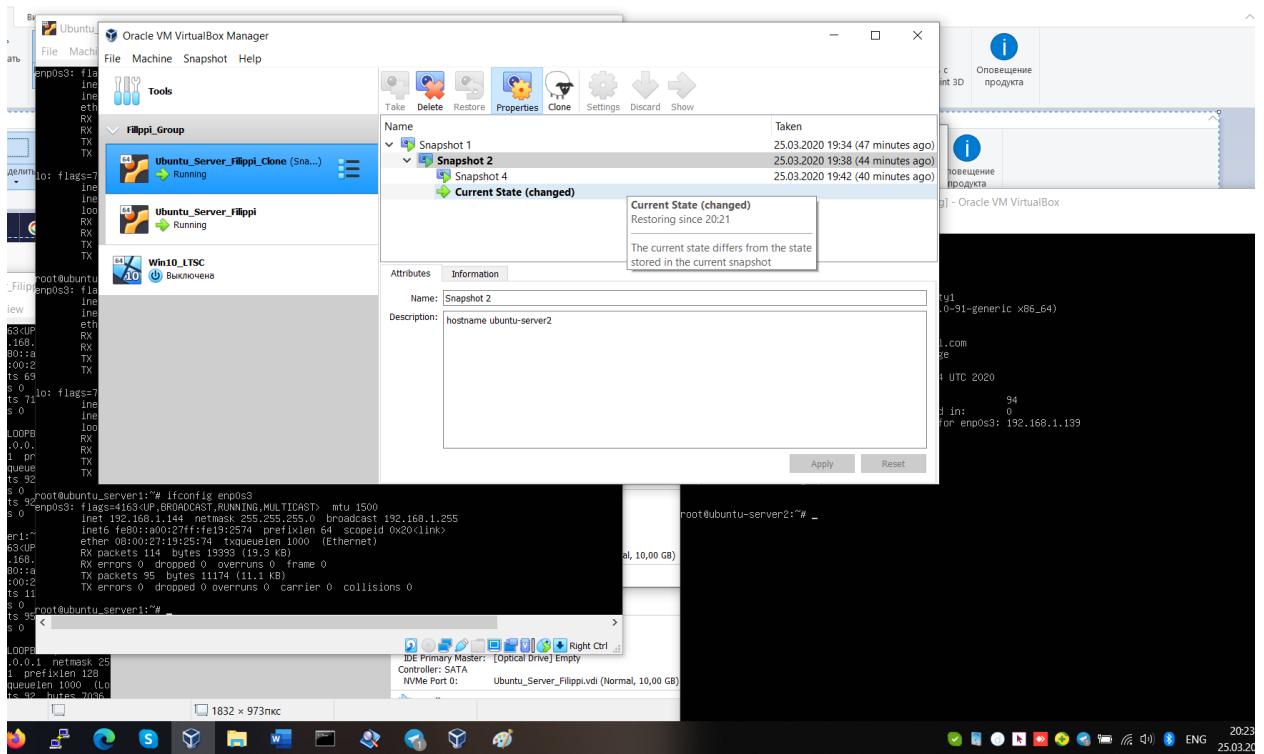
Done. Did with new mac-address

1.7 Створити групу з двох VM: VM1, VM2 та вивчити функції, що відносяться до груп [1, п.1.9].

Done. See screenshot bellow:

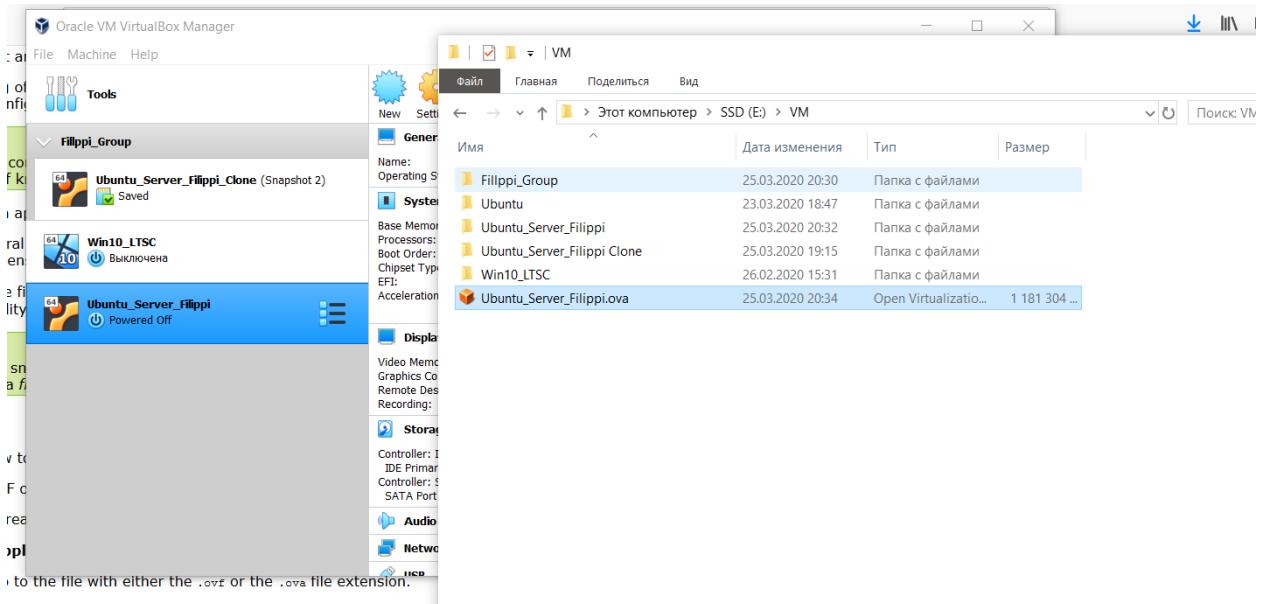


1.8 Для VM1 змінюючи її стан, зробити кілька різних знімків, утворивши розгалужене дерево знімків [1, п.1.10].



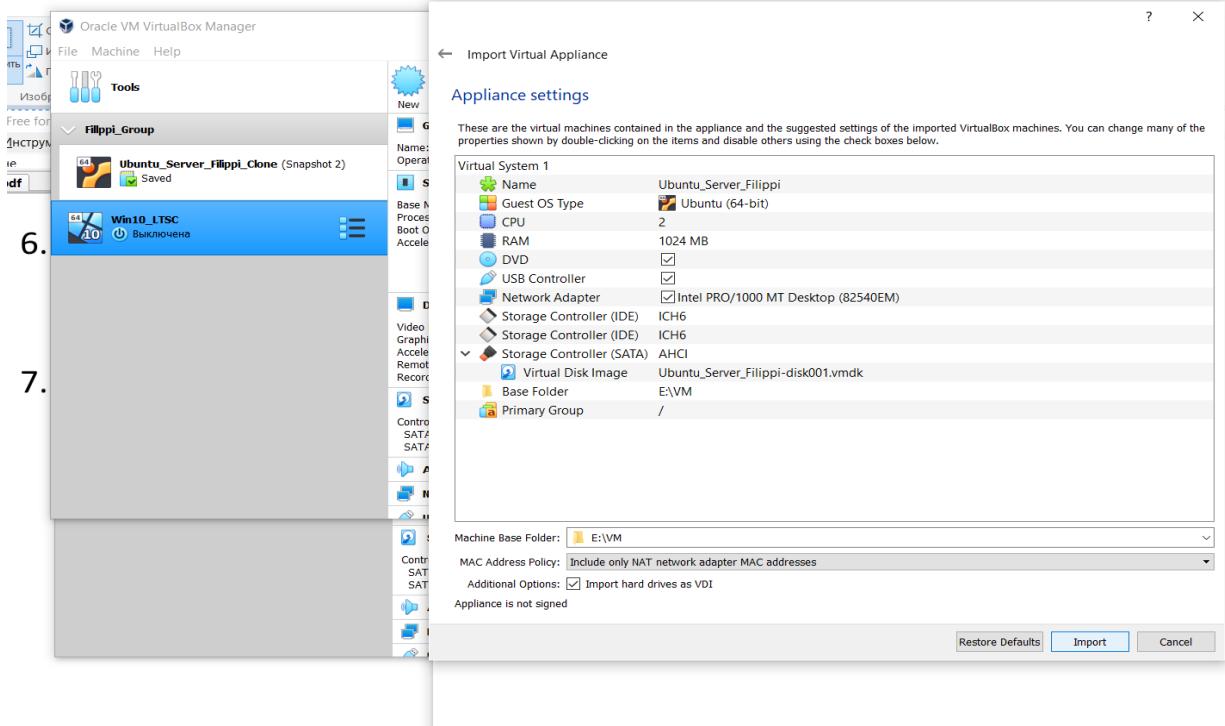
1.9 Зробити експорт VM1, файл *.ova зберегти на мережному диску зі спільним доступом. На цьому ж диску обрати файл *.ova, що створений іншим студентом та імпортувати його [1, п.1.14].

I exported and deleted Ubuntu_Server_Filippi VM



After that I restore it from *.ova file via import:

вій - Paint



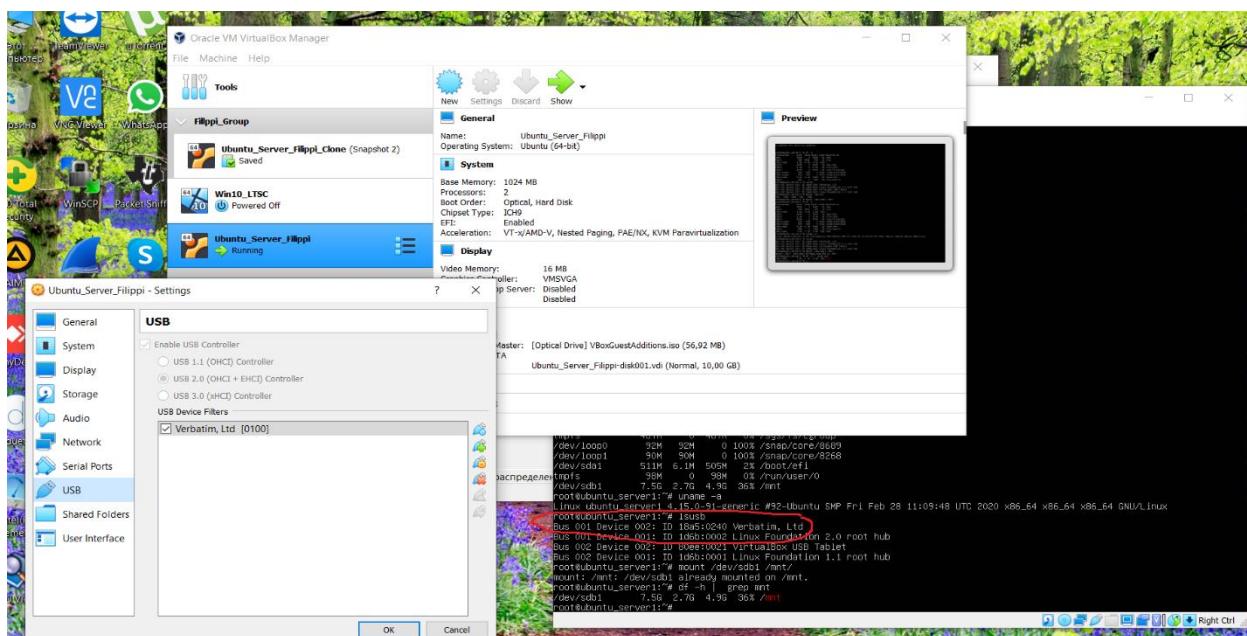
2. Конфігурація віртуальних машин

2.1 Вивчити можливості налаштування VM (загальні налаштування, системні параметри, дисплей, зберігання, аудіо, мережі тощо).

Done

2.2 Провести налаштування USB для підключення USB-портів хостової машини до VM [1, п.3.11].

I added USB flash Verbatim



2.3 Провести налаштування спільної папки для обміну даними між віртуальною машиною та хостом [1, п.4.3].

It is needed to install Guest Additions

<https://www.techrepublic.com/article/how-to-install-virtualbox-guest-additions-on-a-gui-less-ubuntu-server-host/>

image, and then following the steps. Unfortunately, installing this package on a GUI-less Linux server isn't as straightforward. Here are the steps to install this package on a Ubuntu server with no GUI.

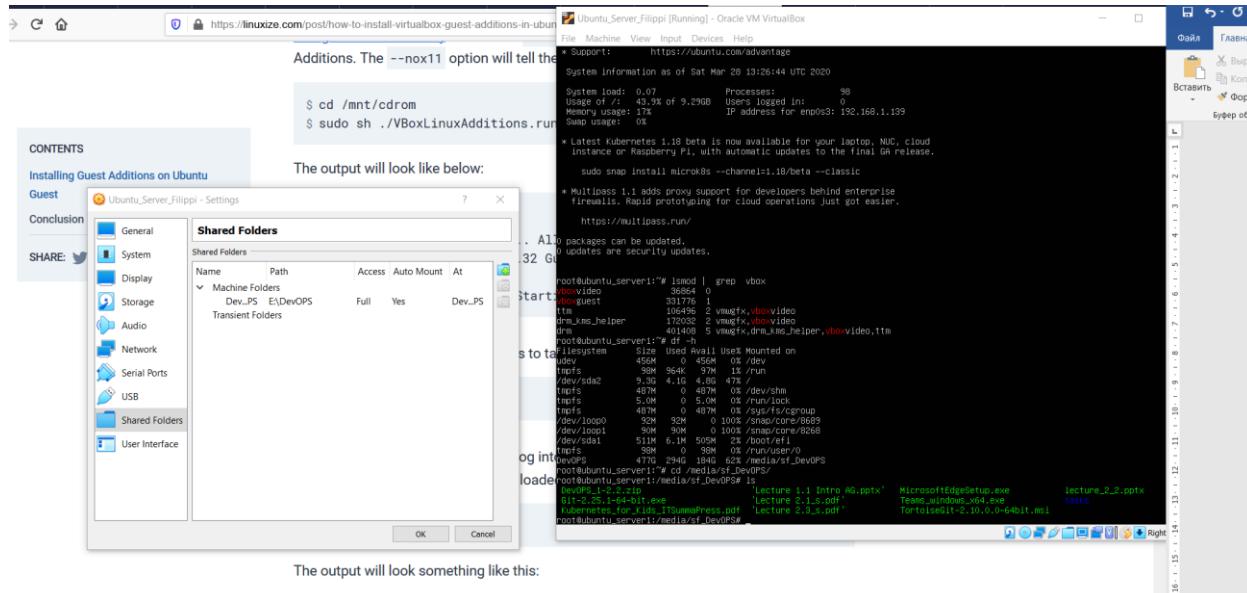
Installing the Guest Additions on a GUI-less server

1. Start VirtualBox.
2. Start the host in question.
3. Once the host has booted, click Devices | Insert Guest Additions CD Image.
4. Log in to your guest server.
5. Mount the CD-ROM with the command `sudo mount /dev/cdrom /media/`
6. Change into the mounted directory with the command `cd /media/cdrom/`
7. Install the necessary dependencies with the command `sudo apt-get install -y linux-headers-generic linux-headers-$(uname -r)`.
8. Change to the root user with the command `sudo su`.
9. Install the Guest Additions package with the command `./VBoxLinuxAdditions.run`
10. Allow the installation to complete.

You will see an error stating that the installer couldn't find the X.Org or XFree86 packages. That's okay, because it doesn't exist on this host.

Reboot your machine, and the Guest Additions will be working. You should see a performance increase, as well as the extra features (shared clipboard, shared folders) awarded by this installation.

Done. See Screenshot bellow:



2.4 Провести налаштування різних режимів роботи мережі для VM1, VM2.

Перевірити наявність зв'язку між VM1, VM2, Host, Internet для різних режимів роботи мережі. Для цього можна використати команду ping. Скласти відповідну таблицю можливих зв'язків.

I am going to configure 2 lans for each of VM. One is internal LAN between 2 VM. Another is the bridge between VM, Host and my local LAN with Internet(for easy and full access to internet and to services of VM).

<http://www.virtualbox.org/manual/ch06.html>

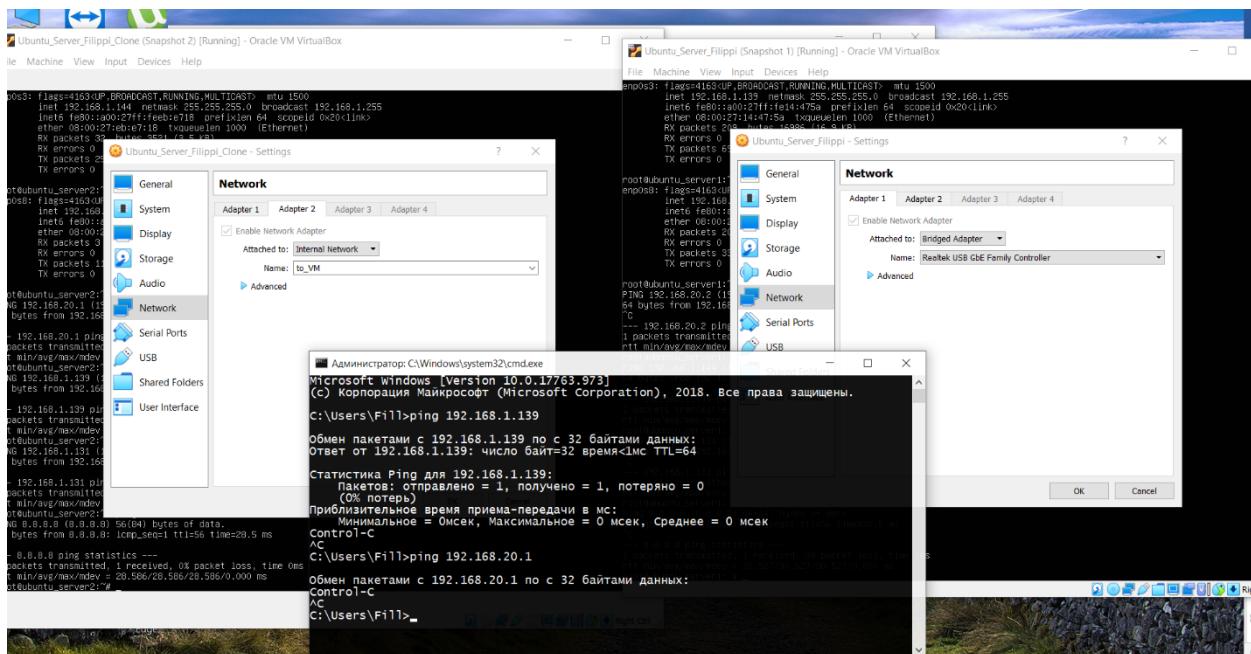
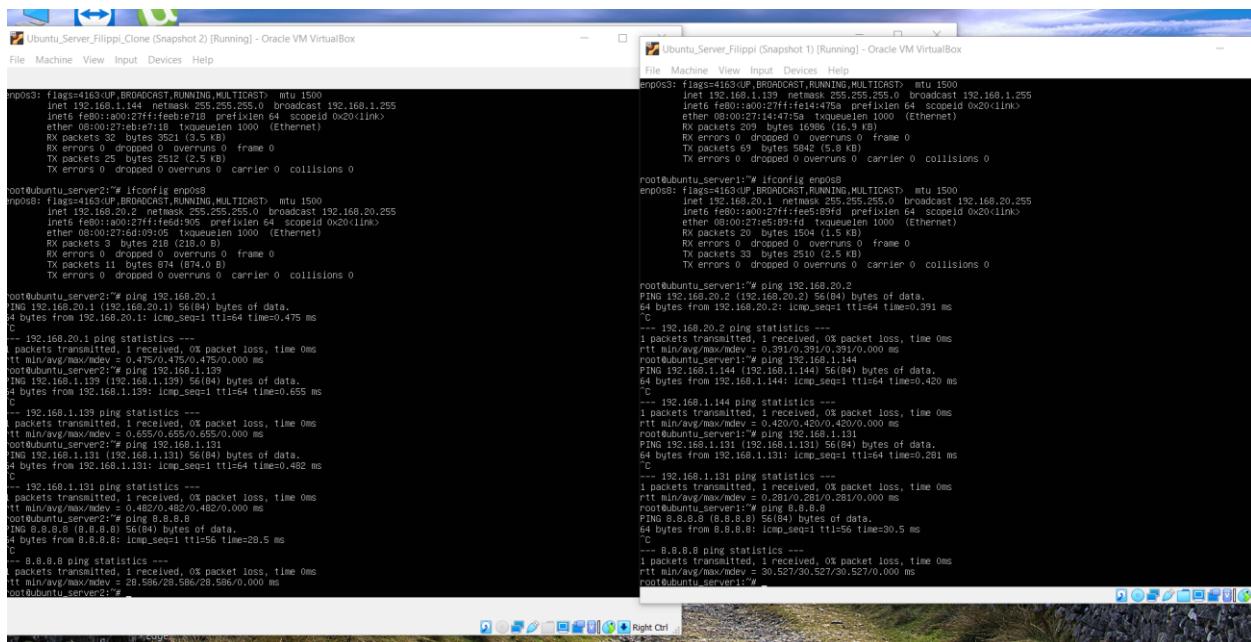
bridge to Host and Internet:

ubuntu_server1: 192.168.1.139 ubuntu_server2: 192.168.1.144

host: 192.168.1.131

internal between WM:

ubuntu_server1: 192.168.20.1 ubuntu_server2: 192.168.20.2



3. Робота з CLI через VBoxManage.

3.1 Запустити командний рядок cmd.exe.

3.2 Вивчити призначення та виконати основні команди VBoxManage list, showvminfo, createvm, startvm, modifyvm, clonevm, snapshot, controlvm [1, п.8].

```
Administrator: C:\Windows\system32\cmd.exe
<none>
Currently Attached USB Devices:
<none>
Bandwidth groups: <none> (0.0000000000000000)
Shared folders:<none>
VRDE Connection: not active
Clients so far: 0
Capture: active
Capture audio: not active
Capture screens: 0
Capture files: E:\VAA\Filippi_Group\ubuntu_Server_Filippi_Clone\ubuntu_Server_Filippi_Clone.webm
Capture rate: 1024x768
capture: 512kbps
capture FPS: 25kbps
Capture options: vc_enabled=true,ac_enabled=false,ac_profile=med
Guest:
Configured memory balloon size: 0MB
OS type: Linux26_64
Additions run level: 1
Additions version: 5.2.8_KernelUbuntu r120774
Guest Facilities:
Facility "VirtualBox Base Driver": active/running (last update: 2020/03/28 14:16:47 UTC)
Facility "Seamless Mode": not active (last update: 2020/03/28 14:16:47 UTC)
Facility "Graphics Mode": not active (last update: 2020/03/28 14:16:47 UTC)
Snapshots:
  Name: Snapshot 1 (UUID: 550bfaf4-b639-4660-a84a-1b6d4ecf1b79)
  Description:
before changing Hostname
  Name: Snapshot 2 (UUID: b2cccad3a-fb11-44ab-9fac-b800e7668a94) *
  Description:
hostname ubuntu-server2
  Name: Snapshot 4 (UUID: cae9e64a-566c-47ba-a715-427a1af61821)
  Description:
st2
c:\Program Files\Oracle\VirtualBox>VBoxManage.exe list vms
"Win10_LTSC" {8707d386-eab7-4360-9a14-c5669285d7b2}
"Ubuntu_Server_Filippi_Clone" {b86e57f5-ccao-4294-a453-36f2ea8bef04}
"Ubuntu_Server_Filippi" {ff0ff367-5507-4230-955d-2c4542162937}
c:\Program Files\Oracle\VirtualBox>VBoxManage.exe showvminfo Ubuntu_Server_Filippi_Clone
```

List of VM, control VM (poweroff), start VM in headless mode

```
Administrator: C:\Windows\system32\cmd.exe
Syntax error: Not enough parameters
::\Program Files\Oracle\VirtualBox>VBoxManage.exe list vms
"Win10_LTSC" {8707d386-eab7-4360-9a14-c5669285d7b2}
"Ubuntu_Server_Filippi_Clone" {b86e57f5-ccao-4294-a453-36f2ea8bef04}
"Ubuntu_Server_Filippi" {ff0ff367-5507-4230-955d-2c4542162937}

::\Program Files\Oracle\VirtualBox>VBoxManage.exe controlvm poweroff
VBoxManage.exe: error: Could not find a registered machine named 'poweroff'
VBoxManage.exe: error: Details: code VBOX_E_OBJECT_NOT_FOUND (0x80bb0)
VBoxManage.exe: error: Context: "FindMachine(Bstr(a->arg[0]).raw(),

::\Program Files\Oracle\VirtualBox>VBoxManage.exe list vms
"Win10_LTSC" {8707d386-eab7-4360-9a14-c5669285d7b2}
"Ubuntu_Server_Filippi_Clone" {b86e57f5-ccao-4294-a453-36f2ea8bef04}
"Ubuntu_Server_Filippi" {ff0ff367-5507-4230-955d-2c4542162937}

::\Program Files\Oracle\VirtualBox>VBoxManage.exe controlvm startvm Ubuntu_Serv
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

::\Program Files\Oracle\VirtualBox>VBoxManage.exe controlvm startvm Ubuntu_Serv
BoxManage.exe: error: Could not find a registered machine named 'startvm'
VBoxManage.exe: error: Details: code VBOX_E_OBJECT_NOT_FOUND (0x80bb0)
VBoxManage.exe: error: Context: "FindMachine(Bstr(pszVm).raw(), machi

::\Program Files\Oracle\VirtualBox>VBoxManage.exe startvm
Usage:
  Note: BoxManage startvm <uuid>|<vmname>[ --runas user ] [ --runasgroup group ]
        [ --type guids|sd|headless|separate ]
        [ -E|-putenv <NAME>[=<VALUE>] ]
```

8.13. VBoxManage controlvm

```
::\Program Files\Oracle\VirtualBox>VBoxManage.exe controlvm Ubuntu_Server_Filippi poweroff
Waiting for VM "Ubuntu_Server_Filippi" to power off...
VM "Ubuntu_Server_Filippi" has been successfully started.

::\Program Files\Oracle\VirtualBox>VBoxManage.exe list vms
active on host, without permanently changing its state. The VM window is gray, to indicate that the VM is currently paused. This is equivalent to
"Win10_LTSC" {8707d386-eab7-4360-9a14-c5669285d7b2}
"Ubuntu_Server_Filippi_Clone" {b86e57f5-ccao-4294-a453-36f2ea8bef04}
"Ubuntu_Server_Filippi" {ff0ff367-5507-4230-955d-2c4542162937}

::\Program Files\Oracle\VirtualBox>VBoxManage.exe startvm --type headless Ubuntu_Server_Filippi
Computes a cold reboot of the virtual machine is done, which immediately restarts and reboots the
waiting for VM "Ubuntu_Server_Filippi" to power on. This is equivalent to selecting the Resume item in the Machine menu of the GUI.
VM "Ubuntu_Server_Filippi" has been successfully started.

::\Program Files\Oracle\VirtualBox>
```

Take and delete snapshot:

ЧАСТИНА 2. РОБОТА З VAGRANT

1. Завантажити необхідну версію Vagrant відповідно інструкціям [5] та відповідно до хостової операційної системи (ОС), що встановлена на робочому місці студента. Для ОС Windows файл може називатися, наприклад, `vagrant_2.2.0_x86_64.msi`. Провести інсталяцію Vagrant. Перевірити наявність шляху до Vagrant bin у змінній Path (My computer -> Properties -> Advanced system settings-> Advanced -> Environment Variables).

Installed version: vagrant_2.2.7_x86_64.msi

```
Администратор: C:\Windows\system32\cmd.exe
provision provisions the vagrant machine
push deploys code in this environment to a configured destination
rdp connects to machine via RDP
reload restarts vagrant machine, loads new Vagrantfile configuration
resume resume a suspended vagrant machine
snapshot manages snapshots: saving, restoring, etc.
ssh connects to machine via SSH
ssh-config outputs OpenSSH valid configuration to connect to the machine
status outputs status of the vagrant machine
suspend suspends the machine
up starts and provisions the vagrant environment
upload upload to machine via communicator
validate validates the Vagrantfile
version prints current and latest Vagrant version
winrmL1 Packet Sniffer
winrm-config executes commands on a machine via WinRM
For help on any individual command run `vagrant COMMAND -h`
Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
`vagrant list-commands`.

C:\Users\Fill>vagrant -v
Vagrant 2.2.7

C:\Users\Fill>
```

2. Запустіть powershell. Створіть папку «прізвище студента» (англійською мовою). В цьому прикладі створимо папку `vagrant_test`. Далі заходимо в папку.

```
Administrator: C:\Windows\system32\cmd.exe - powershell
PS E:\DevOPS\vagrant\Filippi>
PS E:\DevOPS\vagrant\Filippi>
PS E:\DevOPS\vagrant\Filippi> ls
PS E:\DevOPS\vagrant\Filippi> pwd
Path
-----
E:\DevOPS\vagrant\Filippi

PS E:\DevOPS\vagrant\Filippi>
```

3. Проведемо ініціалізацію оточення з вказівкою боксу Vagrant за замовчуванням:

`init hashicorp/precise64`

```
Administrator: C:\Windows\system32\cmd.exe - powershell
PS E:\DevOPS\vagrant\Filippi>
PS E:\DevOPS\vagrant\Filippi>
PS E:\DevOPS\vagrant\Filippi> ls
PS E:\DevOPS\vagrant\Filippi> pwd
Path
-----
E:\DevOPS\vagrant\Filippi

PS E:\DevOPS\vagrant\Filippi> vagrant init hashicorp/precise64
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
PS E:\DevOPS\vagrant\Filippi>
```

4. Запускаємо `vagrant up` та спостерігаємо за повідомленнями під час завантаження та запуску VM.

```
Administrator: C:\Windows\system32\cmd.exe - powershell
up           starts and provisions the vagrant environment
upload        upload to machine via communicator
validate      validates the Vagrantfile
version       prints current and latest Vagrant version
winrm         executes commands on a machine via WinRM
winrm-config  outputs WinRM configuration to connect to the machine
For help on any individual command run 'vagrant COMMAND -h'
Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
  Vagrant subcommands
PS E:\DevOPS\vagrant\Filippi> vagrant halt
=> default: Attempting graceful shutdown of VM...
PS E:\DevOPS\vagrant\Filippi> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
=> default: Checking if box 'hashicorp/precise64' version '1.1.0' is up to date...
=> default: Clearing any previously set forwarded ports...
=> default: Clearing any previously set network interfaces...
=> default: Preparing network interfaces based on configuration...
=> default: Adapter 1: nat
=> default: Forwarding ports...
=> default: 22 (guest) => 2222 (host) (adapter 1)
=> default: Booting VM...
=> default: Waiting for machine to boot. This may take a few minutes...
=> default: SSH address: 127.0.0.1:2222
=> default: SSH username: vagrant
=> default: SSH auth method: private key
=> default: Machine booted and ready!
=> default: Checking for guest additions in VM...
=> default: The guest additions on this VM do not match the installed version of
=> default: VirtualBox! In most cases this is fine, but in rare cases it can
=> default: prevent things such as shared folders from working properly. If you see
=> default: shared folder errors, please make sure the guest additions within the
=> default: virtual machine match the version of VirtualBox you have installed on
=> default: your host and reload your VM.
=> default: Guest Additions Version: 4.2.0 (101222). Za
=> default: VirtualBox Version: 6.1
=> default: Mounting shared folders...
=> default: /vagrant => E:/DevOPS/vagrant/Filippi
=> default: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
=> default: flag to force provisioning. Provisioners marked to run always will still run.
PS E:\DevOPS\vagrant\Filippi>
```

5. Підключаємося до VM за допомогою програми PuTTY (завантажити можна з [6]), використовуючи SSH, IP-адресу та порт що вказані вище (127.0.0.1:2222). За замовчуванням login – vagrant та password також vagrant.

6. Зафіксуйте дату та час, виконавши команду date

```
root@precise64: ~
RX packets:580 errors:0 dropped:0 overruns:0 frame:0
TX packets:394 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:68850 (68.8 KB) TX bytes:63913 (63.9 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@precise64:~# uname -a
Linux precise64 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
root@precise64:~# date
Sun Mar 29 09:01:21 UTC 2020
root@precise64:~# w
 09:01:28 up 2 min, 1 user, load average: 0.11, 0.05, 0.02
USER     TTY     FROM           LOGIN@   IDLE   JCPU   PCPU WHAT
vagrant  pts/0    10.0.2.2       09:00    0.00s  0.24s  0.00s sshd: vagrant [
root@precise64:~#
```

7. Зупиніть та видаліть створену VM.

```
root@precise64: ~
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@precise64:~# uname -a
Linux precise64 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
root@precise64:~# date
Sun Mar 29 09:01:21 UTC 2020
root@precise64:~# w
 09:01:28 up 2 min, 1 user, load average: 0.11, 0.05, 0.02
USER     TTY     FROM           LOGIN@   IDLE   JCPU   PCPU WHAT
vagrant  pts/0    10.0.2.2       09:00    0.00s  0.24s  0.00s sshd: vagrant [
root@precise64:~# Broadcast message from root@precise64
          (unknown) at 9:04 ...
The system is going down for halt NOW!
[1] 11886 Putty Fatal Error
  * Remote side unexpectedly closed network connection
    OK
```

Припинення VM веде до відображення повідомлення про закриття зв'язку з гостем.

```
--> default: Machine booted and ready!
default: Checking for guest additions in VM...
default: The guest additions on this VM do not match the installed version of
default: VirtualBox! In most cases this is fine, but in rare cases it can
default: prevent things such as shared folders from working properly. If you see
default: shared folder errors, please make sure the guest additions within the
default: virtual machine match the version of VirtualBox you have installed on
default: your host and reload your VM.
default: Guest Additions Version: 4.2.0
default: VirtualBox Version: 6.1
default: Mounting shared folders...
default: >>> vagrant => E:/DevOPS/vagrant/Filippi
--> default: Machine already provisioned. Run `vagrant provision` or use the `--provision` flag to force provisioning. Provisioners marked to run always will still run.
PS E:\DevOPS\vagrant\Filippi> vagrant halt
--> default: Attempting graceful shutdown of VM...
PS E:\DevOPS\vagrant\Filippi>
```

8. Створити тестову середу з двох серверів, використовуючи інструкції [7].
Параметри серверів задаються викладачем або обираються самостійно студентом.

I created 2 VM startup in Vagrantfile and successfully run that configuration

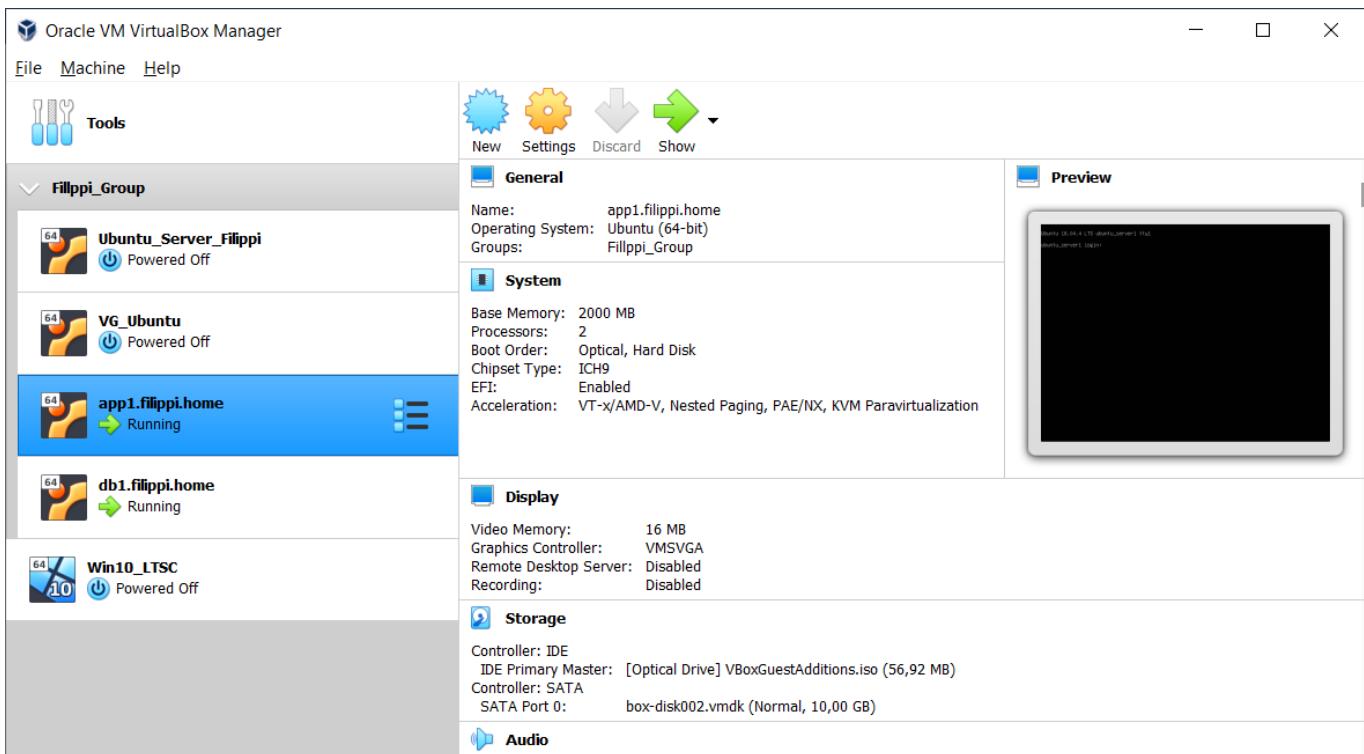
In Vagrantfile example of [7] was the mistake, but I fixed it)))

I used my box "ubuntu server 18.04"

My variant of Vargantfile is in screenshot:

```
# Префикс для LAN сети
BRIDGE_NET="192.168.1."
# Префикс для Internal сети
INTERNAL_NET="192.168.20."
# Домен который будем использовать для всей площадки
DOMAIN="filippi.home"
# Массив из хешей, в котором заданы настройки для каждой виртуальной машины
servers=[
{
  :hostname => "app1." + DOMAIN,
  :ip => BRIDGE_NET + "150",
  :ip_int => INTERNAL_NET + "1",
  :ram => 2000
},
{
  :hostname => "db1." + DOMAIN,
  :ip => BRIDGE_NET + "151",
  :ip_int => INTERNAL_NET + "2",
  :ram => 1000
}
]

# Входим в Главную конфигурацию vagrant версии 2
Vagrant.configure(2) do |config|
  # Добавить шару между хостовой и гостевой машиной
  #config.vm.synced_folder "E://DevOPS/vargant/Filippi/", "/src/shara"
  # Отключить дефолтную шару
  config.vm.synced_folder ".", "/vagrant", disabled: true
  # Проходим по элементах массива "servers"
  servers.each do |machine|
    # Применяем конфигурации для каждой машины. Имя машины(как ее будет видно в Vbox GUI) находится в переменной "machine[:hostname]"
    config.vm.define machine[:hostname] do |node|
      # Поднять машину из образа "ubuntu 14.04", который мы создали в предыдущей статье
      node.vm.box = "ubuntu server 18.04"
      # Также, можно указать URL откуда стянуть нужный box если такой есть
      #node.vm.box_url = "http://files.vagrantup.com/precise64.box"
      # Пул портов, который будет использоваться для подключения к каждый VM через 127.0.0.1
      node.vm.usable_port_range = (2200..2250)
      # Hostname который будет присвоен VM (самой ОС)
      node.vm.hostname = machine[:hostname]
      #VBoxManage.exe list bridgedifs overwrite NAT adapter :adapter=>1
      # Добавление и настройка Bridge сетевого адаптера(моста). Чтобы узнать точное название bridge адаптера, нужно использовать VBoxManage.exe (смотрите ниже)
      node.vm.network "public_network", ip: machine[:ip], bridge: 'Realtek USB GbE Family Controller'
      # Добавление и настройка внутреннего сетевого адаптера (Intnet)
      node.vm.network "private_network", ip: machine[:ip_int], virtualbox_intnet: "intnet"
      # Настройка SSH доступа
      # Домен/IP для подключения
      #node.ssh.host = machine[:ip]
      # Для доступа по ранее добавленному ключу
      #node.ssh.private_key_path = "private_key"
      # SSH логин пользователя
      # node.ssh.username = "alex"
      # SSH пароль
    end
  end
end
```



```
vagrant@app1:~
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65566
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 4 bytes 336 (336.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4 bytes 336 (336.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vagrant@app1:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=64 time=0.710 ms
^C
--- 192.168.20.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.710/0.710/0.710/0.000 ms
vagrant@app1:~$ ping 192.168.1.131
PING 192.168.1.131 (192.168.1.131) 56(84) bytes of data.
64 bytes from 192.168.1.131: icmp_seq=1 ttl=64 time=0.465 ms
^C
--- 192.168.1.131 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.465/0.465/0.465/0.000 ms
vagrant@app1:~
```

```
fil@db1:~
```

```
System load: 0.12      Users logged in: 0
Usage of /: 44.5% of 9.29GB  IP address for enp0s3: 10.0.2.15
Memory usage: 18%        IP address for enp0s8: 192.168.1.151
Swap usage: 0%          IP address for enp0s9: 192.168.20.2
Processes: 103

* Kubernetes 1.18 GA is now available! See https://microk8s.io for docs or
install it with:
  sudo snap install microk8s --channel=1.18 --classic

* Multipass 1.1 adds proxy support for developers behind enterprise
firewalls. Rapid prototyping for cloud operations just got easier.
  https://multipass.run/

0 packages can be updated.
0 updates are security updates.
```

```
MINGW64/e/DevOps/vagrant/Filippi
```

```
$ vagrant up
Bringing machine 'app1.filippi.home' up with 'virtualbox' provider...
Bringing machine 'db1.filippi.home' up with 'virtualbox' provider...
=> app1.filippi.home: You assigned a static IP ending in ".1" to this machine.
=> app1.filippi.home: This is very often used by the router and can cause the
=> app1.filippi.home: network to not work properly. If the network doesn't work
=> app1.filippi.home: properly, try changing this IP.
=> app1.filippi.home: You assigned a static IP ending in ".1" to this machine.
=> app1.filippi.home: This is very often used by the router and can cause the
=> app1.filippi.home: network to not work properly. If the network doesn't work
=> app1.filippi.home: properly, try changing this IP.
=> app1.filippi.home: Clearing any previously set network interfaces...
=> app1.filippi.home: Preparing network interfaces based on configuration...
=> app1.filippi.home: Adapter 1: nat
=> app1.filippi.home: Adapter 2: bridged
=> app1.filippi.home: Adapter 3: intnet
=> app1.filippi.home: Forwarding ports...
=> app1.filippi.home: 22 (guest) => 2222 (host) (adapter 1)
=> app1.filippi.home: Running 'pre-boot' VM customizations...
=> app1.filippi.home: Booting VM...
=> app1.filippi.home: Waiting for machine to boot. This may take a few minutes...
=> app1.filippi.home: SSH address: 127.0.0.1:2222
=> app1.filippi.home: SSH username: vagrant
=> app1.filippi.home: SSH auth method: password
=> app1.filippi.home: Inserting generated public key within guest...
=> app1.filippi.home: Removing insecure key from the guest if it's present...
=> app1.filippi.home: Key inserted! Disconnecting and reconnecting using new SSH key...
=> app1.filippi.home: Machine booted and ready!
=> app1.filippi.home: Checking for guest additions in VM...
=> app1.filippi.home: Setting hostname...
=> app1.filippi.home: Configuring and enabling network interfaces...
=> db1.filippi.home: Importing base box 'ubuntu server 18.04'...
=> db1.filippi.home: Mounting shared folder 'F:\Filippi\home'
=> db1.filippi.home: Setting the name of the host to 'db1.filippi.home'
=> db1.filippi.home: Fixed port collision for 22 => 2222. Now on port 2200.
=> db1.filippi.home: Clearing any previously set network interfaces...
=> db1.filippi.home: Preparing network interfaces based on configuration...
=> db1.filippi.home: Adapter 1: nat
=> db1.filippi.home: Adapter 2: bridged
=> db1.filippi.home: Adapter 3: intnet
=> db1.filippi.home: Forwarding ports...
=> db1.filippi.home: 22 (guest) => 2200 (host) (adapter 1)
=> db1.filippi.home: Running 'pre-boot' VM customizations...
=> db1.filippi.home: Booting VM...
=> db1.filippi.home: Waiting for machine to boot. This may take a few minutes...
=> db1.filippi.home: SSH address: 127.0.0.1:2200
=> db1.filippi.home: SSH username: vagrant
=> db1.filippi.home: SSH auth method: password
=> db1.filippi.home: Inserting generated public key within guest...
=> db1.filippi.home: Removing insecure key from the guest if it's present...
=> db1.filippi.home: Key inserted! Disconnecting and reconnecting using new SSH key...
=> db1.filippi.home: Machine booted and ready!
```

9. Створити власний Vagrant box, використовуючи інструкції [8] та вимоги, що визначає викладач або обирає студент.

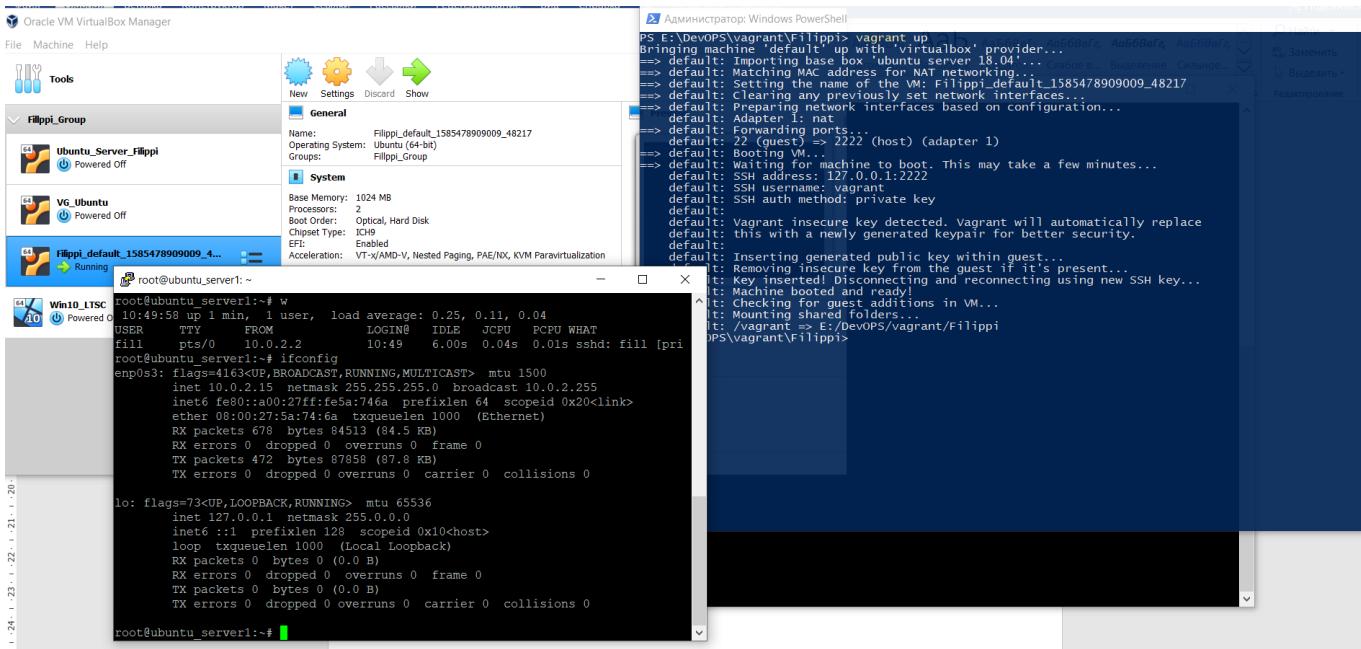
```
PS E:\DevOPS\vagrant\Filippi> vagrant box remove hashicorp/precise64
Box 'hashicorp/precise64' (v1.1.0) with provider 'virtualbox' appears
to still be in use by at least one Vagrant environment. Removing
the box could corrupt the environment. We recommend destroying
these environments first:

default (ID: c53d5999002f48369f8e96ff4bf21736)

Are you sure you want to remove this box? [y/N] y
Removing box 'hashicorp/precise64' (v1.1.0) with provider 'virtualbox'...
PS E:\DevOPS\vagrant\Filippi> vagrant box list
There are no installed boxes! Use `vagrant box add` to add some.
PS E:\DevOPS\vagrant\Filippi> vagrant package
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Waiting for cleanup before exiting...
Vagrant exited after cleanup due to external interrupt.
PS E:\DevOPS\vagrant\Filippi> vagrant box list
There are no installed boxes! Use `vagrant box add` to add some.
PS E:\DevOPS\vagrant\Filippi> vagrant package --base 'VG_Ubuntu' --output VG_Ubuntu_template
==> VG_Ubuntu: Attempting graceful shutdown of VM...
VG_Ubuntu: Guest communication could not be established! This is usually because
VG_Ubuntu: SSH is not running, the authentication information was changed,
VG_Ubuntu: or some other networking issue. Vagrant will force halt, if
VG_Ubuntu: capable.
==> VG_Ubuntu: Forcing shutdown of VM...
==> VG_Ubuntu: Exporting VM...
==> VG_Ubuntu: Compressing package to: E:/DevOPS/vagrant/Filippi/VG_Ubuntu_template
PS E:\DevOPS\vagrant\Filippi>
```

```
Vagrant exited after cleanup due to external interrupt.
PS E:\DevOPS\vagrant\Filippi> vagrant box list
There are no installed boxes! Use `vagrant box add` to add some.
PS E:\DevOPS\vagrant\Filippi> vagrant package --base 'VG_Ubuntu' --output VG_Ubuntu_template
==> VG_Ubuntu: Attempting graceful shutdown of VM...
VG_Ubuntu: Guest communication could not be established! This is usually because
VG_Ubuntu: SSH is not running, the authentication information was changed,
VG_Ubuntu: or some other networking issue. Vagrant will force halt, if
VG_Ubuntu: capable.
==> VG_Ubuntu: Forcing shutdown of VM...
==> VG_Ubuntu: Exporting VM...
==> VG_Ubuntu: Compressing package to: E:/DevOPS/vagrant/Filippi/VG_Ubuntu_template
PS E:\DevOPS\vagrant\Filippi> vagrant box add VG_Ubuntu_template --name 'ubuntu server 18.04'
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'ubuntu server 18.04' (v0) for provider:
box: Unpacking necessary files from: file:///E:/DevOPS/vagrant/Filippi/VG_Ubuntu_template
box:
==> box: Successfully added box 'ubuntu server 18.04' (v0) for 'virtualbox'!
PS E:\DevOPS\vagrant\Filippi>
```

My Box was created and started successfully



ПЕРЕЛІК ПОСИЛАНЬ

1. Oracle VM VirtualBox User Manual <https://www.virtualbox.org/manual/>
2. Офіційна сторінка VirtualBox [https://www.virtualbox.org/](https://www.virtualbox.org)
3. Сторінка завантаження Ubuntu <https://ubuntu.com/download>
4. Сторінка документації Vagrant <https://www.vagrantup.com/docs/index.html>
5. Сторінка з інструкціями щодо інсталяції Vagrant <https://www.vagrantup.com/docs/installation/index.html>
6. Сторінка завантаження PuTTY <https://www.putty.org/>
7. Робота з vagrantfile <http://sysadm.pp.ua/linux/sistemyvirtualizacii/vagrantfile.html>
8. Створення власного Vagrant box <http://sysadm.pp.ua/linux/sistemyvirtualizacii/vagrant-box-creation.html>