Warszawa, 21Z

Create 5G Core VM

Pracownia Dyplomowa Inżynierska 1

Andrzej Gawor

Contents

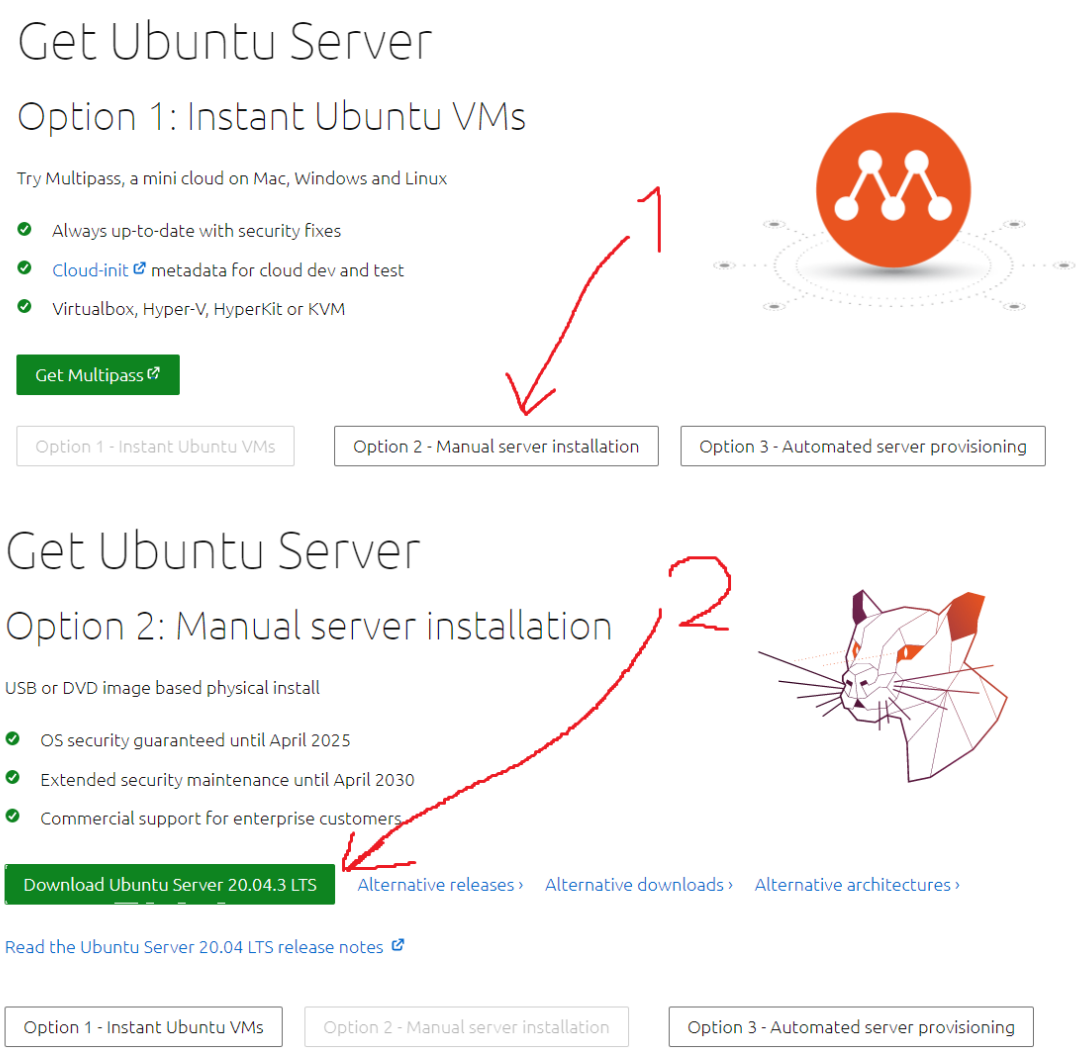
Step 1 - Create Ubuntu Server VM

1.1 Install Virtual Box

Many tutorials about this topic can be found online.

1.2 Download Ubuntu Server

Use this https://ubuntu.com/download/server



Get Ubuntu Server

Option 1: Instant Ubuntu VMs

Try Multipass, a mini cloud on Mac, Windows and Linux

- Always up-to-date with security fixes
- Cloud-init metadata for cloud dev and test
- Virtualbox, Hyper-V, HyperKit or KVM

Get Multipass
1
Option 1 - Instant Ubuntu VMs
Option 2 - Manual server installation
Option 3 - Automated server provisioning

Get Ubuntu Server

Option 2: Manual server installation

USB or DVD image based physical install

- OS security guaranteed until April 2025
- Extended security maintenance until April 2030
- Commercial support for enterprise customers

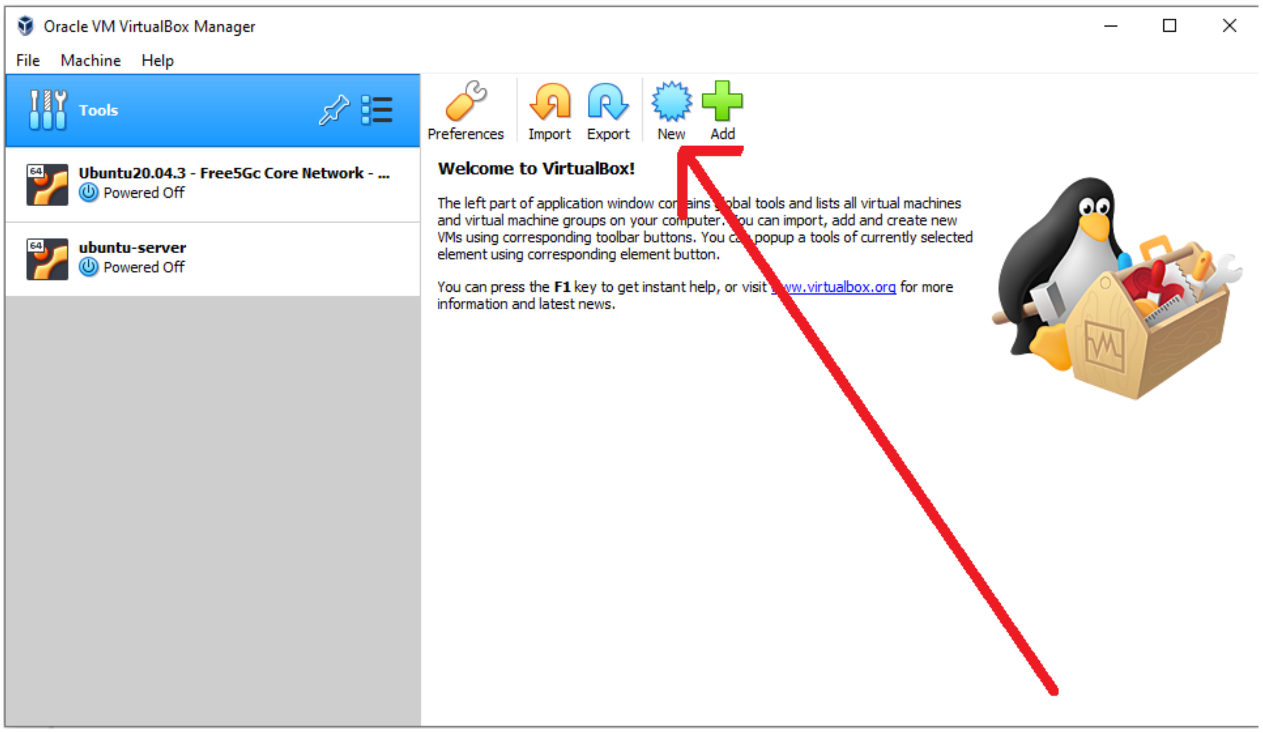
2
Download Ubuntu Server 20.04.3 LTS
Alternative releases ›
Alternative downloads ›
Alternative architectures ›
Read the Ubuntu Server 20.04 LTS release notes
Option 1 - Instant Ubuntu VMs
Option 2 - Manual server installation
Option 3 - Automated server provisioning

1.3 Create a Ubuntu Server VM

We use Ubuntu Server instead of Ubuntu Desktop because we only need a basic server machine without too many unnecessary functionalities. The resulting overhead to your host machine is smaller, and the VM starts up faster too.

1.3.1 New VM

Launch VirtualBox and click `new`.

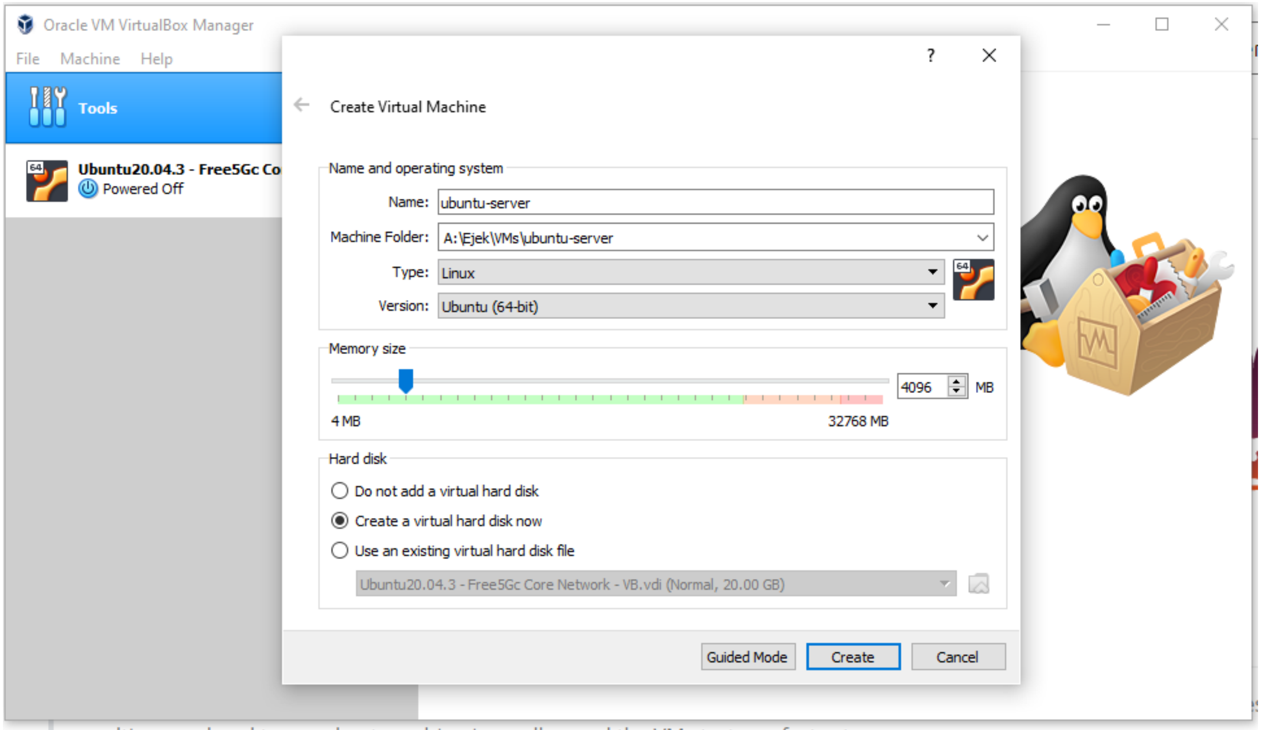


Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

- Name the first VM using a generic name as `ubuntu`, `ubuntu-server`, or `ubuntu-20.04`
- You can pick 1 or 2 (or more) CPUs, and about 2048M memory, although you can change them later



Oracle VM VirtualBox Manager

File Machine Help

Tools

Ubuntu20.04.3 - Free5Gc Co
Powered Off

Create Virtual Machine

Name and operating system

Name: ubuntu-server

Machine Folder: A:\Ejek\VMs\ubuntu-server

Type: Linux

Version: Ubuntu (64-bit)

Memory size

4096 MB

4 MB 32768 MB

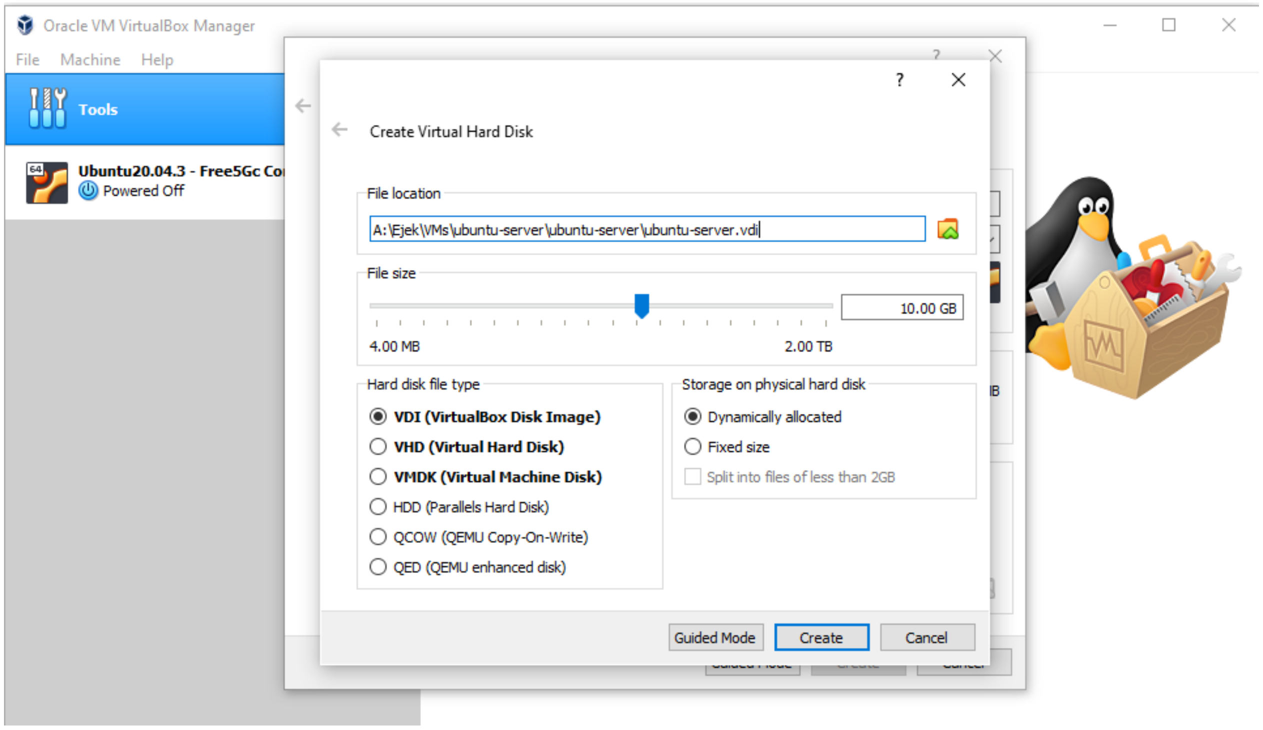
Hard disk

- Do not add a virtual hard disk
- [✓ Create a virtual hard disk now]
- Use an existing virtual hard disk file

Ubuntu20.04.3 - Free5Gc Core Network - VB.vdi (Normal, 20.00 GB)

Guided Mode | Create | Cancel

- Create a virtual hard disk VDI with about 10GB memory size.



Oracle VM VirtualBox Manager

File Machine Help

Tools

Ubuntu20.04.3 - Free5Gc Co
Powered Off

Create Virtual Hard Disk

File location

A:\Ejek\VMs\ubuntu-server\ubuntu-server\ubuntu-server.vdi

File size

10.00 GB

4.00 MB 2.00 TB

Hard disk file type

- [✓ VDI (VirtualBox Disk Image)]
- VHD (Virtual Hard Disk)
- VMDK (Virtual Machine Disk)
- HDD (Parallels Hard Disk)
- QCOW (QEMU Copy-On-Write)
- QED (QEMU enhanced disk)

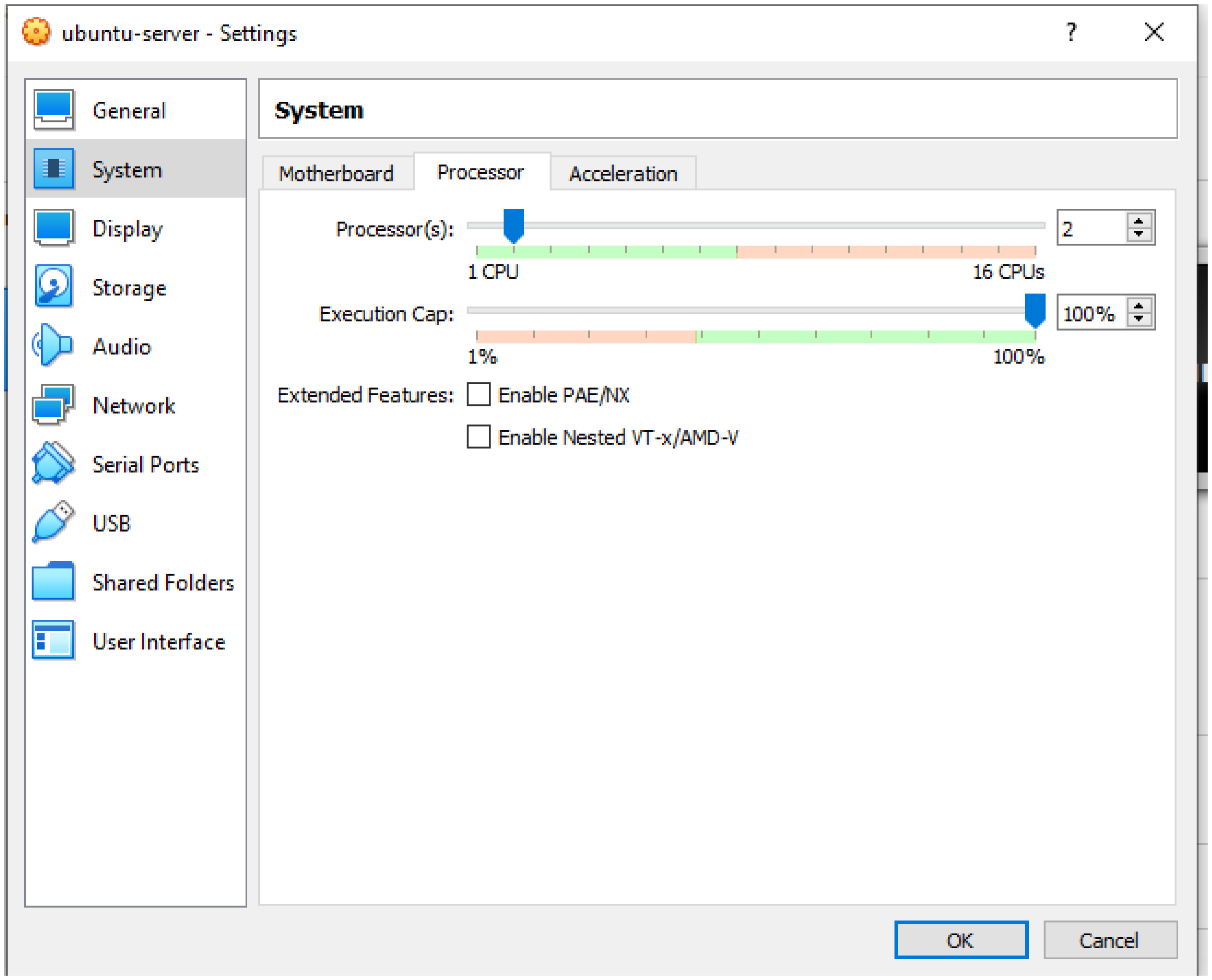
Storage on physical hard disk

- [✓ Dynamically allocated]
- Fixed size
- Split into files of less than 2GB

Guided Mode | Create | Cancel

1.3.2 Setup VM

- You can increase the CPU number from default 1 to 2.



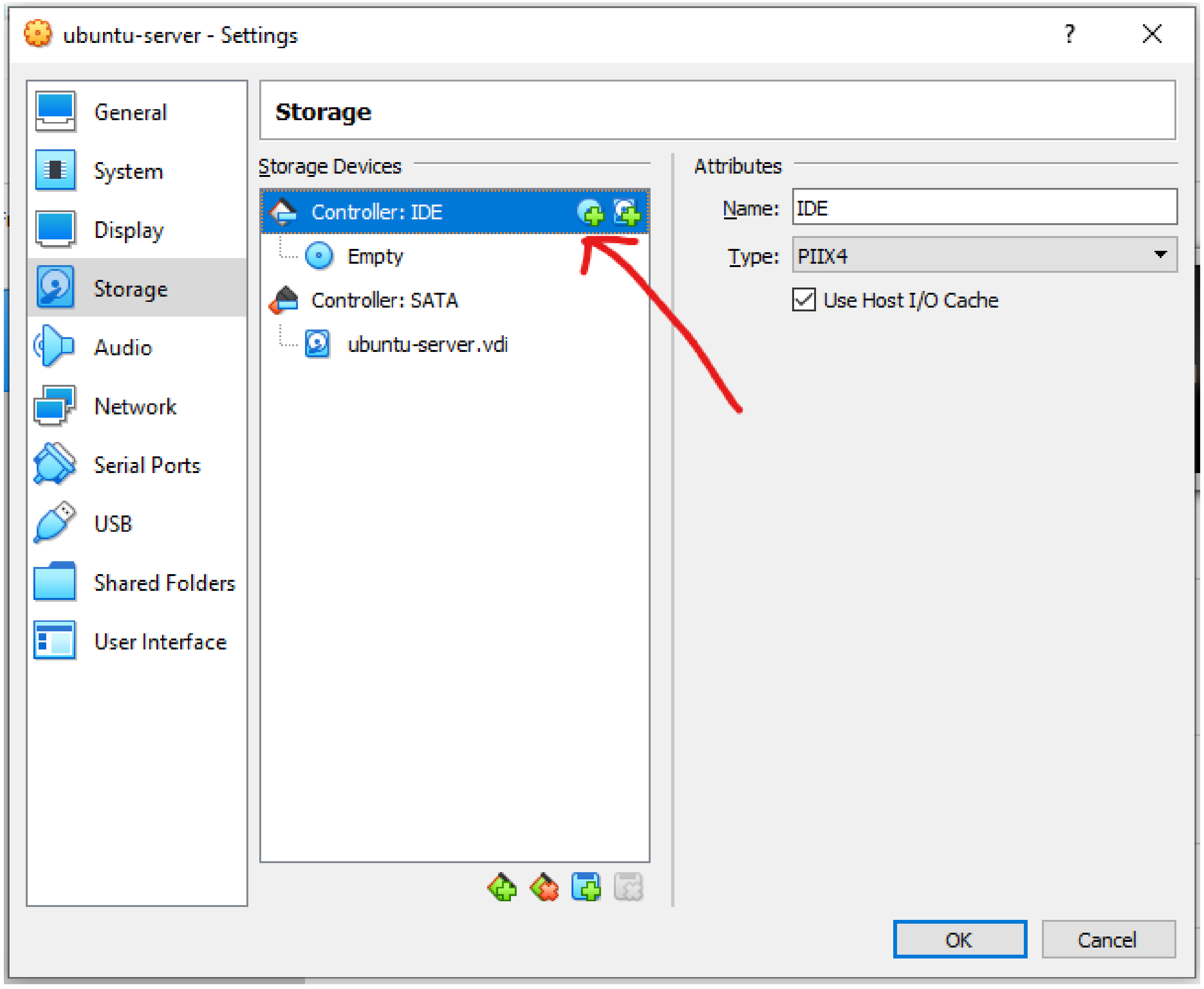
Processor(s): 2

Execution Cap: 100%

Extended Features: ☐ Enable PAE/NX

☐ Enable Nested VT-x/AMD-V

- Also, we need to plug in our "CD" with ubuntu (the .iso file)
- To do so move to "Storage" and click "+" under Controller: IDE option as shown below


ubuntu-server - Settings

Storage

Storage Devices

- Controller: IDE
  - Empty
- Controller: SATA
  - ubuntu-server.vdi

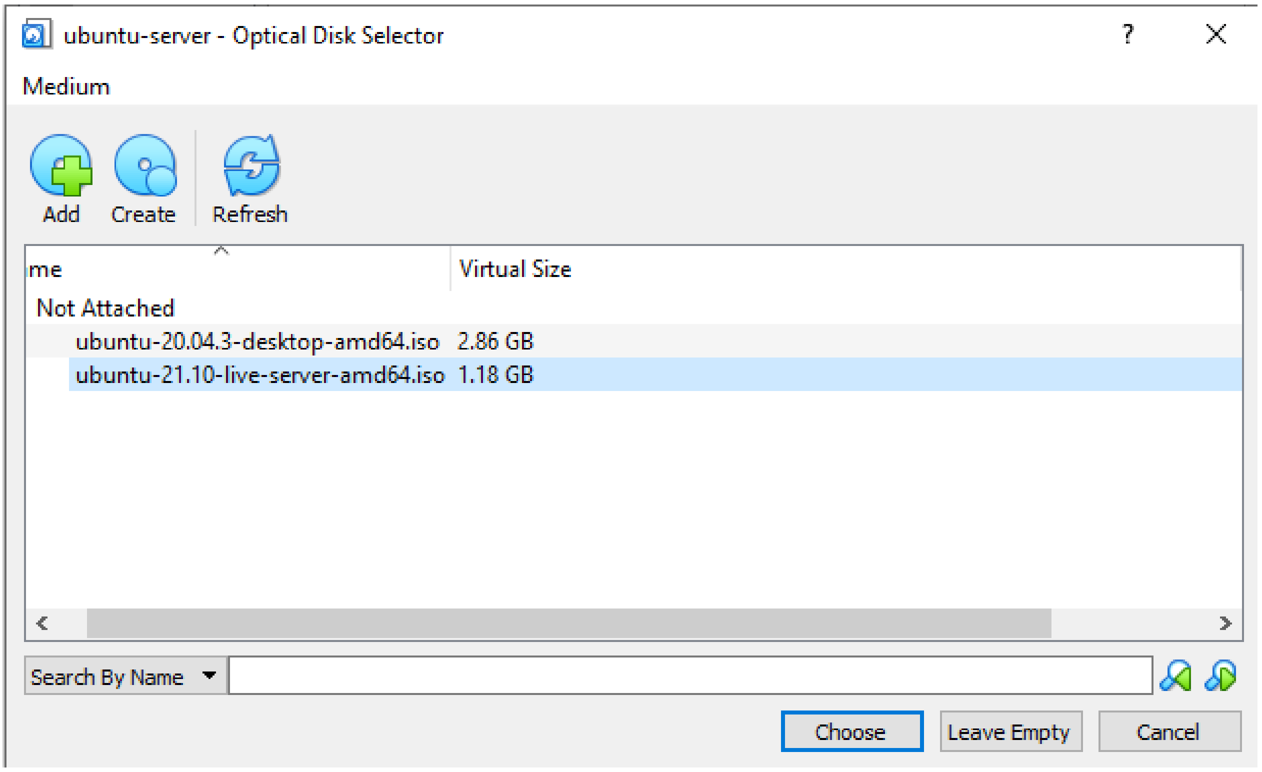
Attributes

Name: IDE

Type: PIIX4

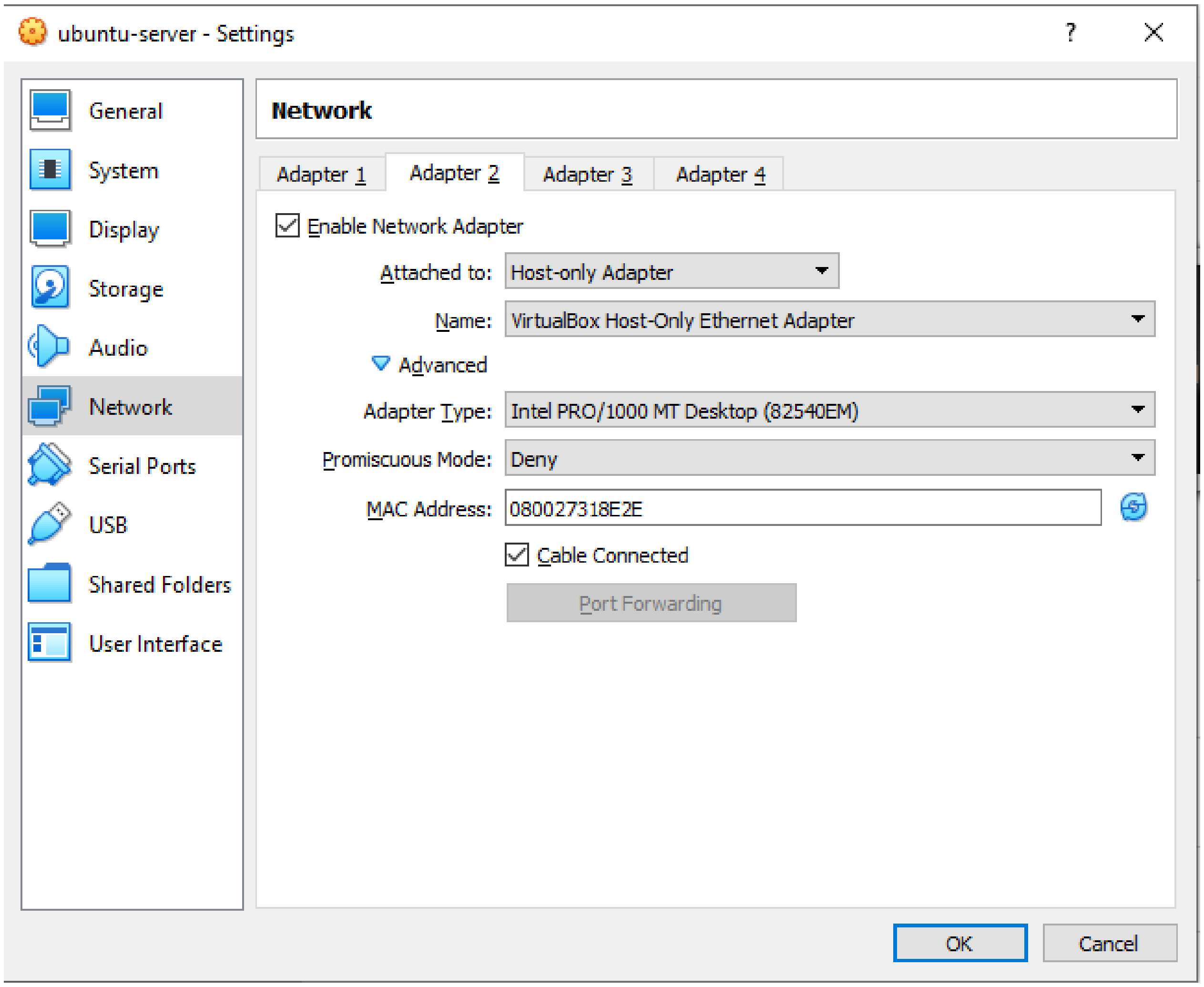
☑ Use Host I/O Cache

- And add your .iso file and the choose it



me	Virtual Size
Not Attached	
ubuntu-20.04.3-desktop-amd64.iso	2.86 GB
ubuntu-21.10-live-server-amd64.iso	1.18 GB

- Move to the "Network" and enable second adapter. Make it Host-only, rest of config leave default.



Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

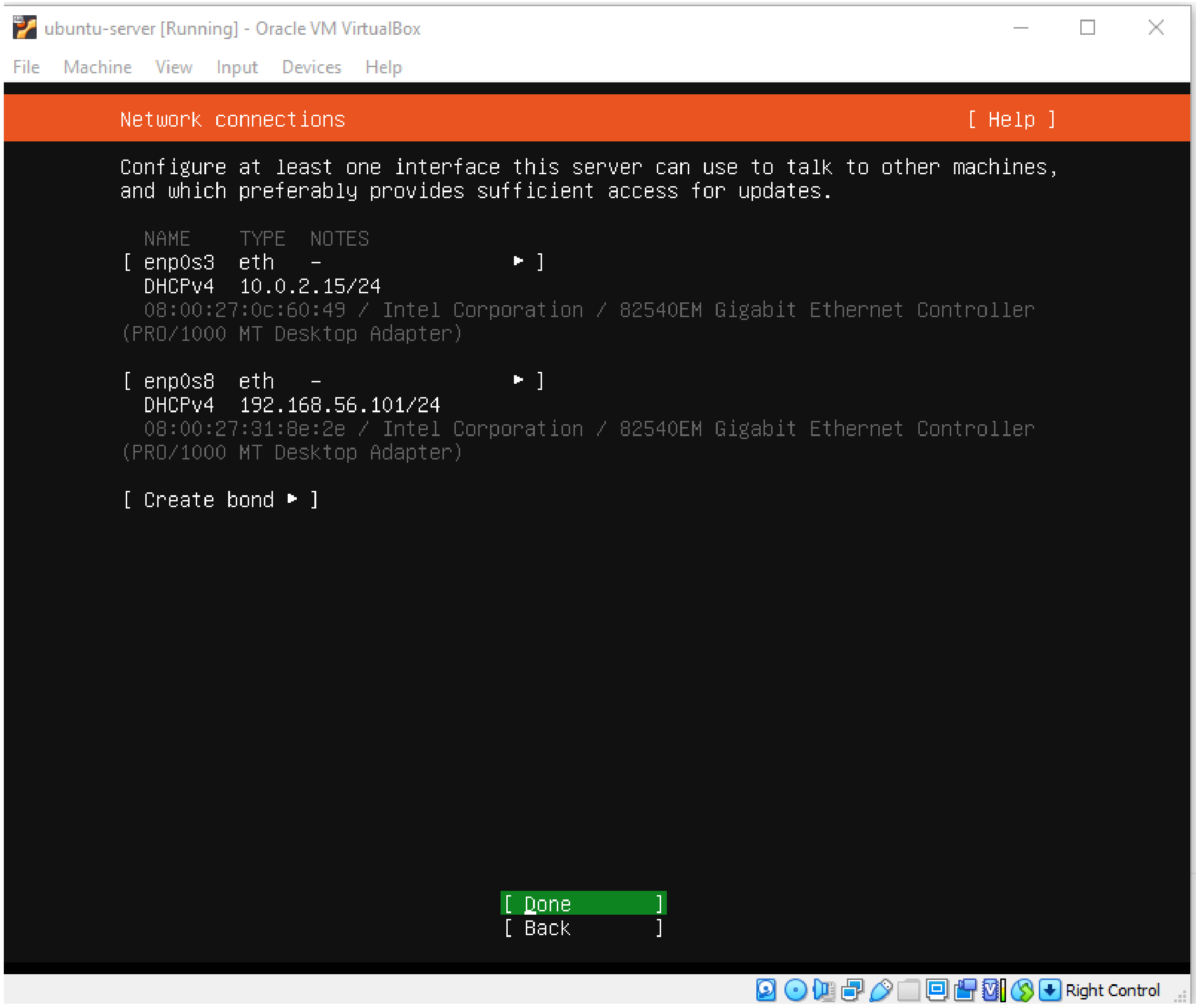
MAC Address: 080027318E2E

☑ Cable Connected

1.3.3 Start the VM

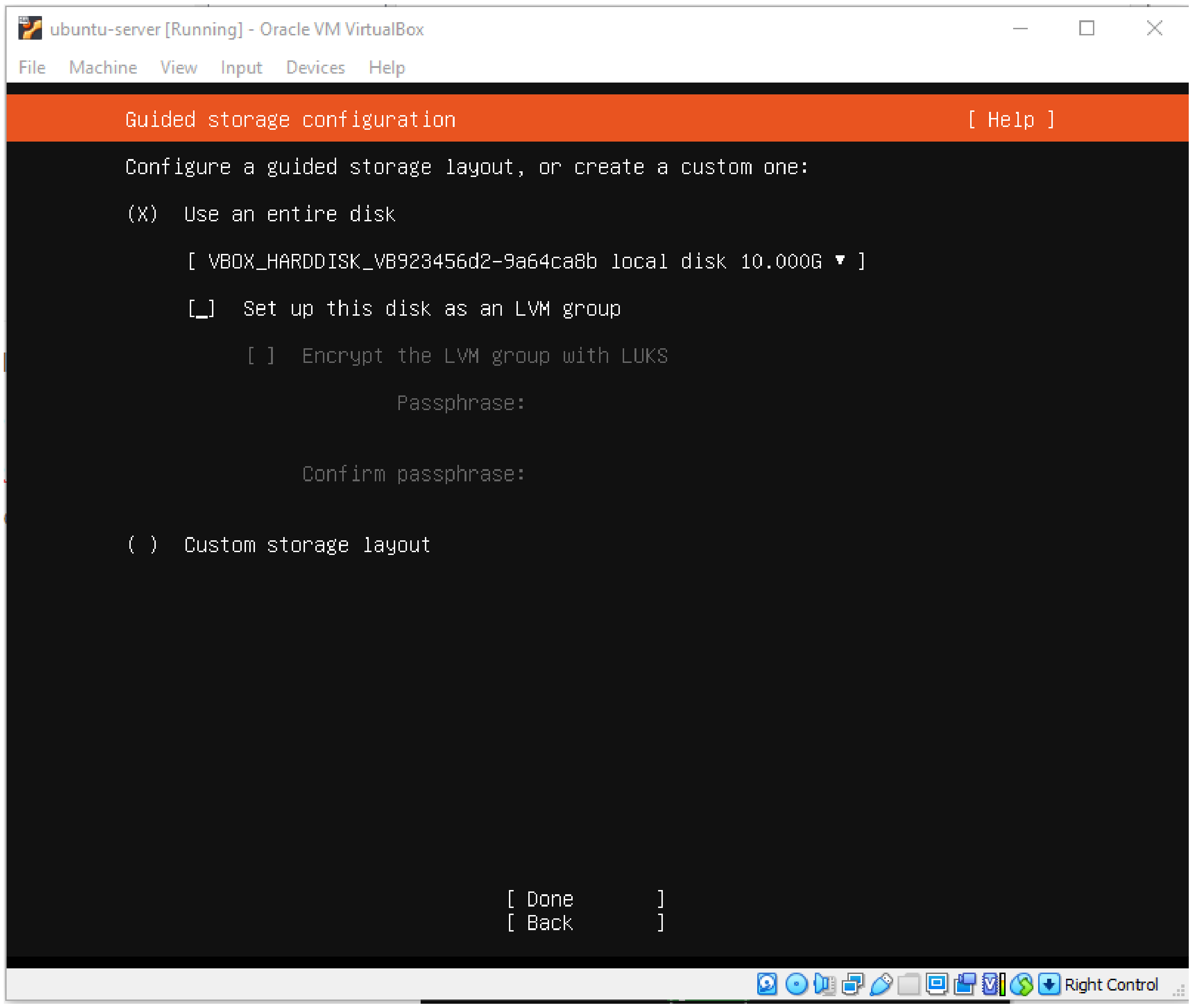
Installation is a long process I will list only the key-points. Unsaid things leave default and things like language, keyboard etc. are left for you.

- Choose "Ubuntu Server" option from GRUB menu
- Choose "Ubntu Server" as the base for installation not "Ubuntu server (minimized)"
- Leave Network Connections default


ubuntu-server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Control

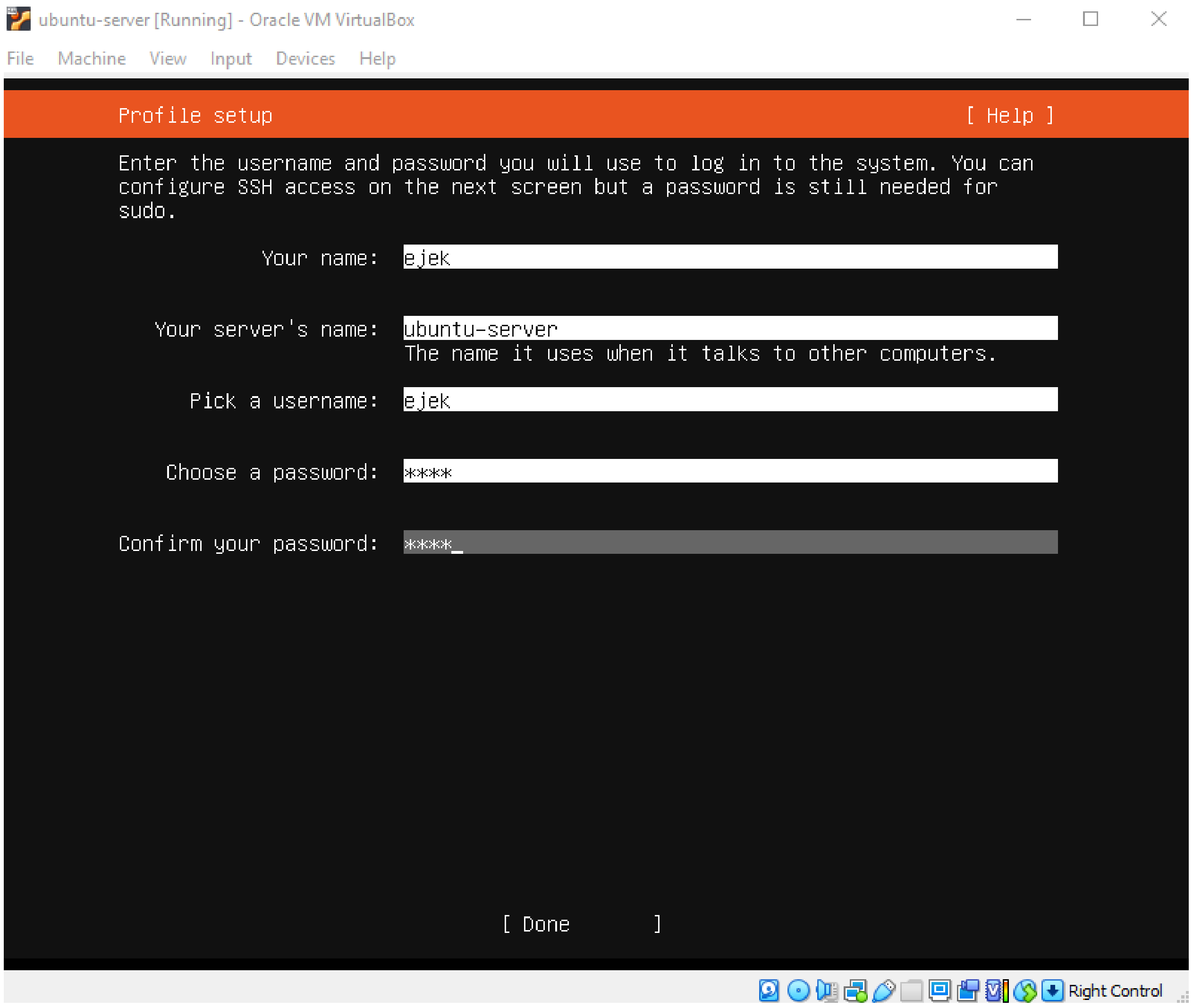
```
Network connections                                                  [ Help ]

Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

  NAME    TYPE  NOTES
[ enp0s3  eth   -                  ► ]
  DHCPv4  10.0.2.15/24
  08:00:27:0c:60:49 / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ enp0s8  eth   -                  ► ]
  DHCPv4  192.168.56.101/24
  08:00:27:31:8e:2e / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ Create bond ► ]



                                [ Done         ]
                                [ Back         ]
```

- Uncheck "the LVM option"
  - not choosing LVM will make it a little bit easier later if you want to extend your disk space


ubuntu-server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Control

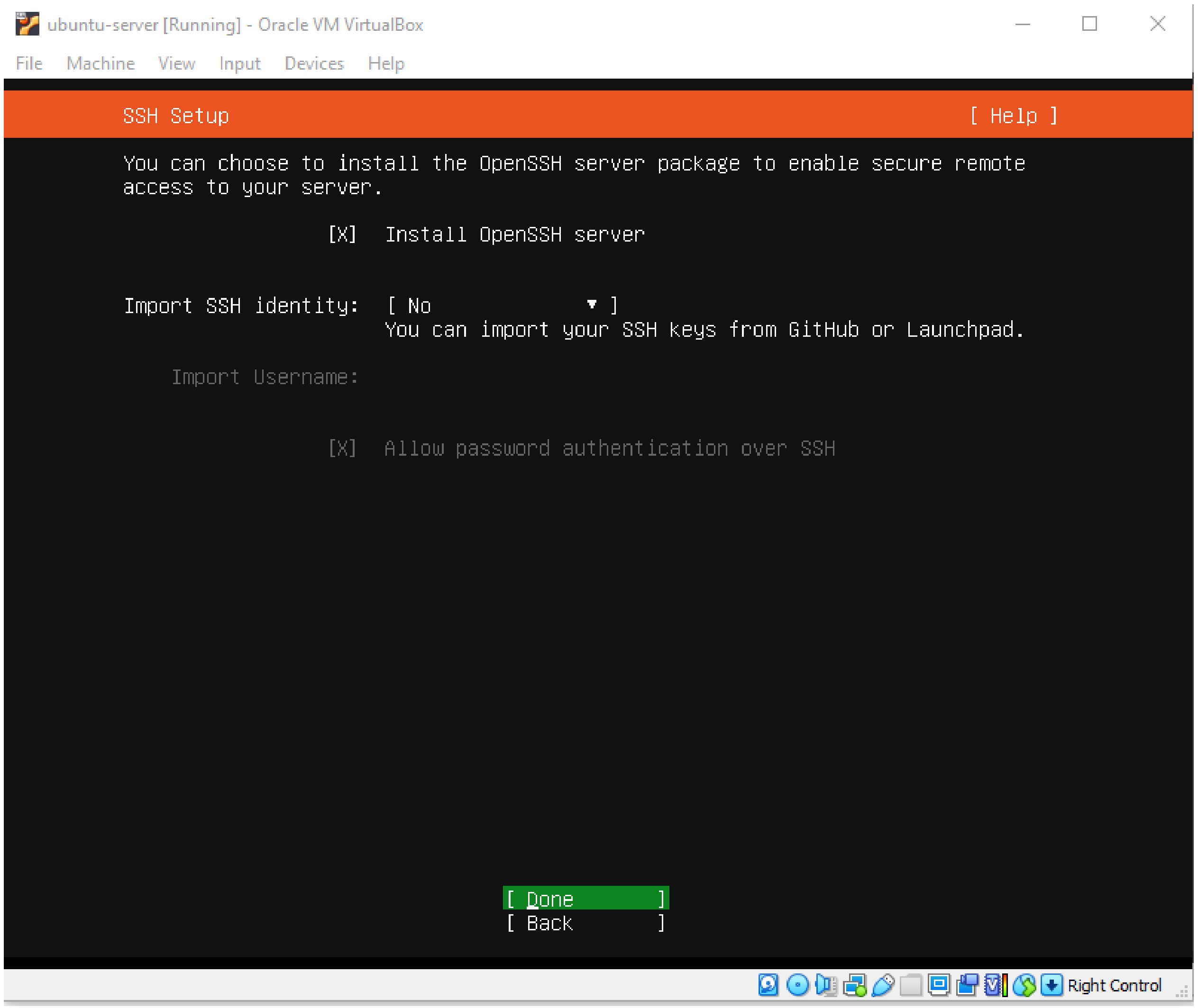
```
Guided storage configuration                                    [ Help ]

Configure a guided storage layout, or create a custom one:

(X)  Use an entire disk

     [ VBOX_HARDDISK_VB923456d2-9a64ca8b local disk 10.000G ▼ ]

     [_]  Set up this disk as an LVM group

          [ ]  Encrypt the LVM group with LUKS

                       Passphrase:


               Confirm passphrase:


( )  Custom storage layout











                            [ Done       ]
                            [ Back       ]
```

- it is recommended that you choose short username and password for ease of typing later


ubuntu-server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Control

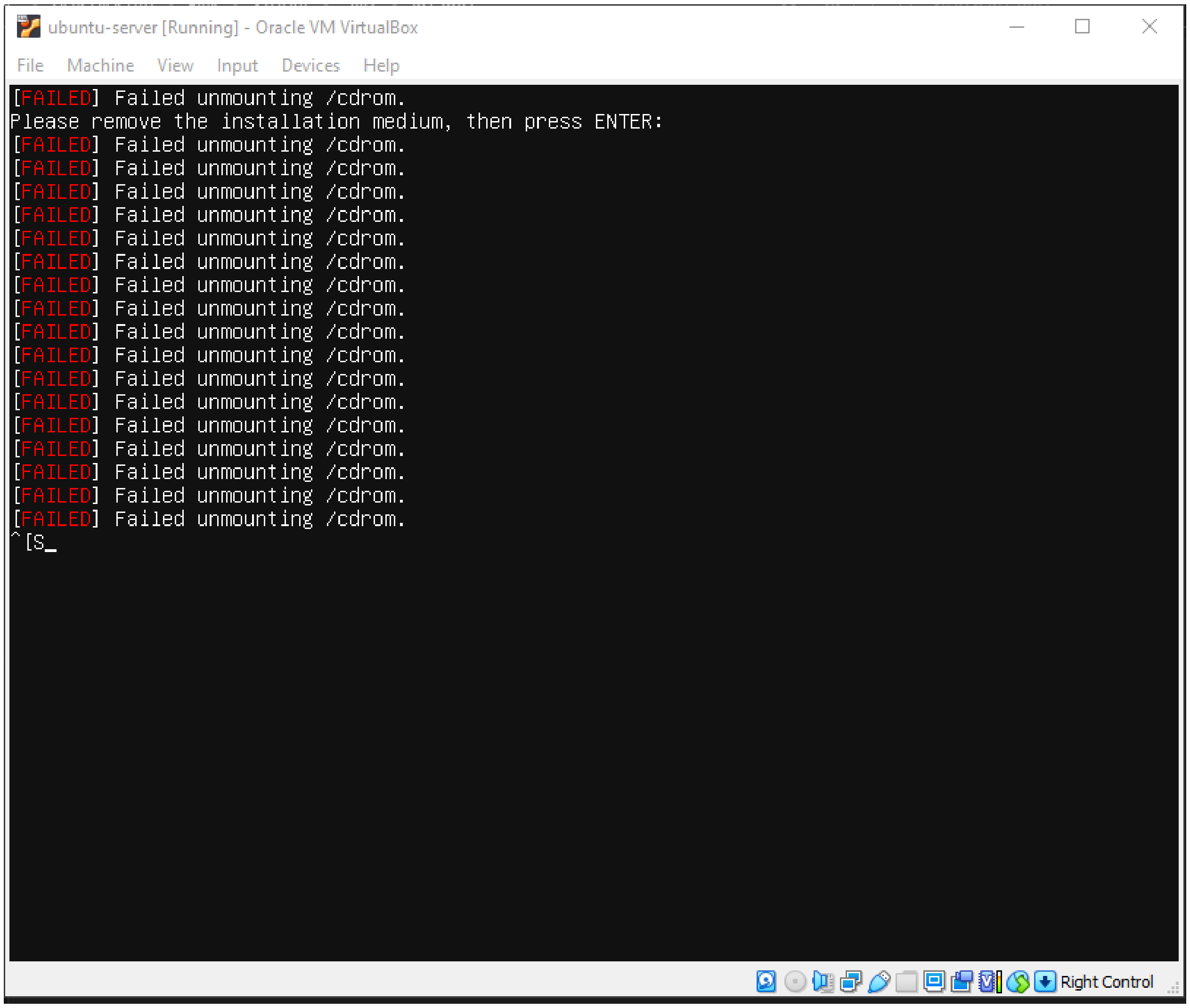
```
Profile setup                                                    [ Help ]

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

              Your name:  ejek

       Your server's name:  ubuntu-server
                            The name it uses when it talks to other computers.

        Pick a username:  ejek

      Choose a password:  ****

  Confirm your password:  ****_

                              [ Done       ]
```

- Install SSH server


ubuntu-server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Control

```
SSH Setup                                                      [ Help ]

You can choose to install the OpenSSH server package to enable secure remote
access to your server.

                 [X]  Install OpenSSH server


Import SSH identity:  [ No               ▾ ]
                      You can import your SSH keys from GitHub or Launchpad.

    Import Username:

                 [X]  Allow password authentication over SSH

                                 [ Done       ]
                                 [ Back       ]
```

- Later follow the displayed instructions and leave default values.
- Reboot at the end.
- Here you have nothing to worry about. Just press enter

- 

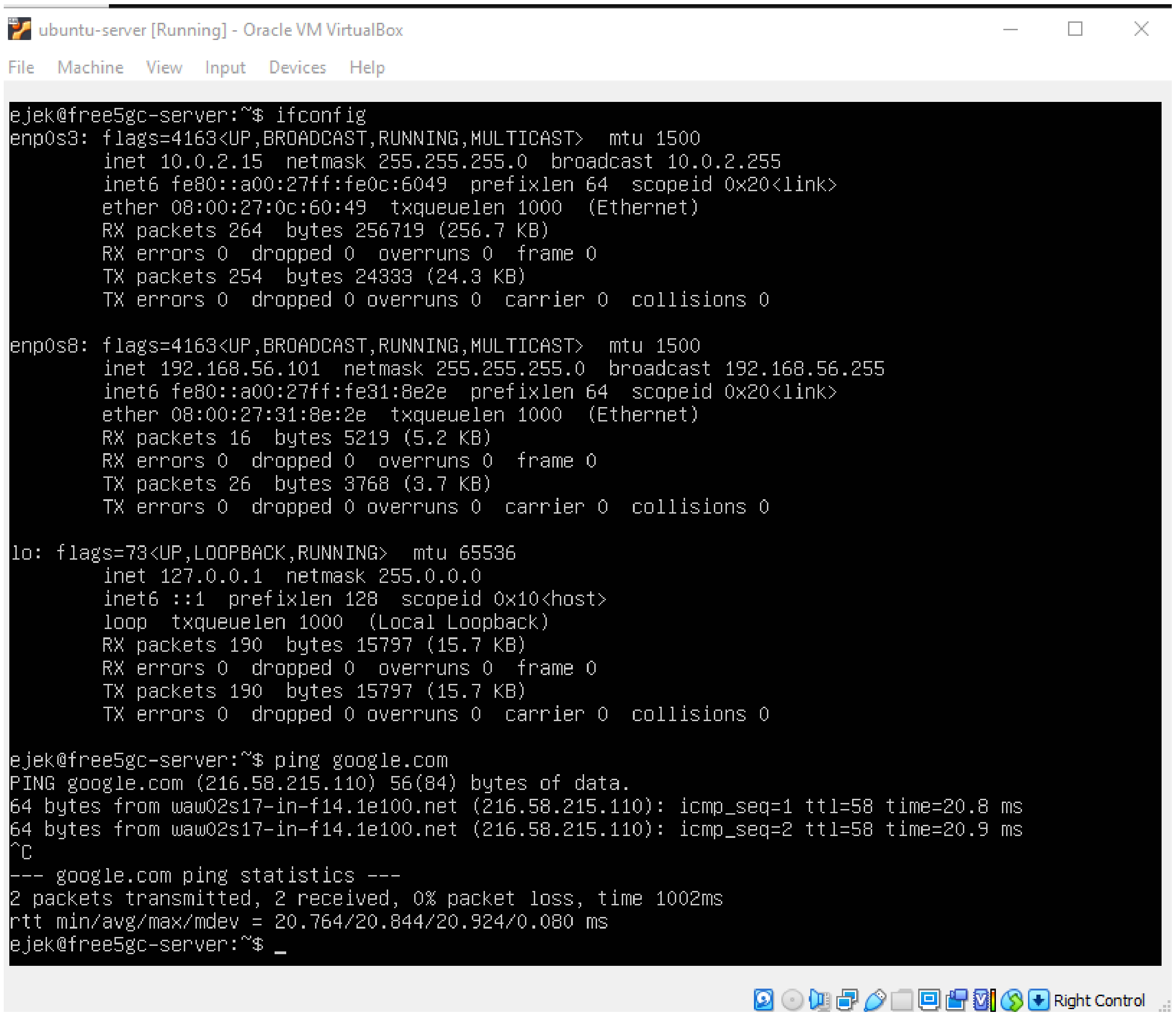
ubuntu-server [Running] - Oracle VM VirtualBox

```
[FAILED] Failed unmounting /cdrom.
Please remove the installation medium, then press ENTER:
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
[FAILED] Failed unmounting /cdrom.
^[S_
```

1.3.4 Test your ubuntu-server installation

After you've logged in type in some commands:

- `ping google.com`
- `ifconfig`
    - It may require installation of `net-tools`


ubuntu-server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Control

```
ejek@free5gc-server:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0c:6049  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0c:60:49  txqueuelen 1000  (Ethernet)
        RX packets 264  bytes 256719 (256.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 254  bytes 24333 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe31:8e2e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:31:8e:2e  txqueuelen 1000  (Ethernet)
        RX packets 16  bytes 5219 (5.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 3768 (3.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 190  bytes 15797 (15.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 190  bytes 15797 (15.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ejek@free5gc-server:~$ ping google.com
PING google.com (216.58.215.110) 56(84) bytes of data.
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=1 ttl=58 time=20.8 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=2 ttl=58 time=20.9 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 20.764/20.844/20.924/0.080 ms
ejek@free5gc-server:~$ _
```

Of course addresses on your machine may differ.

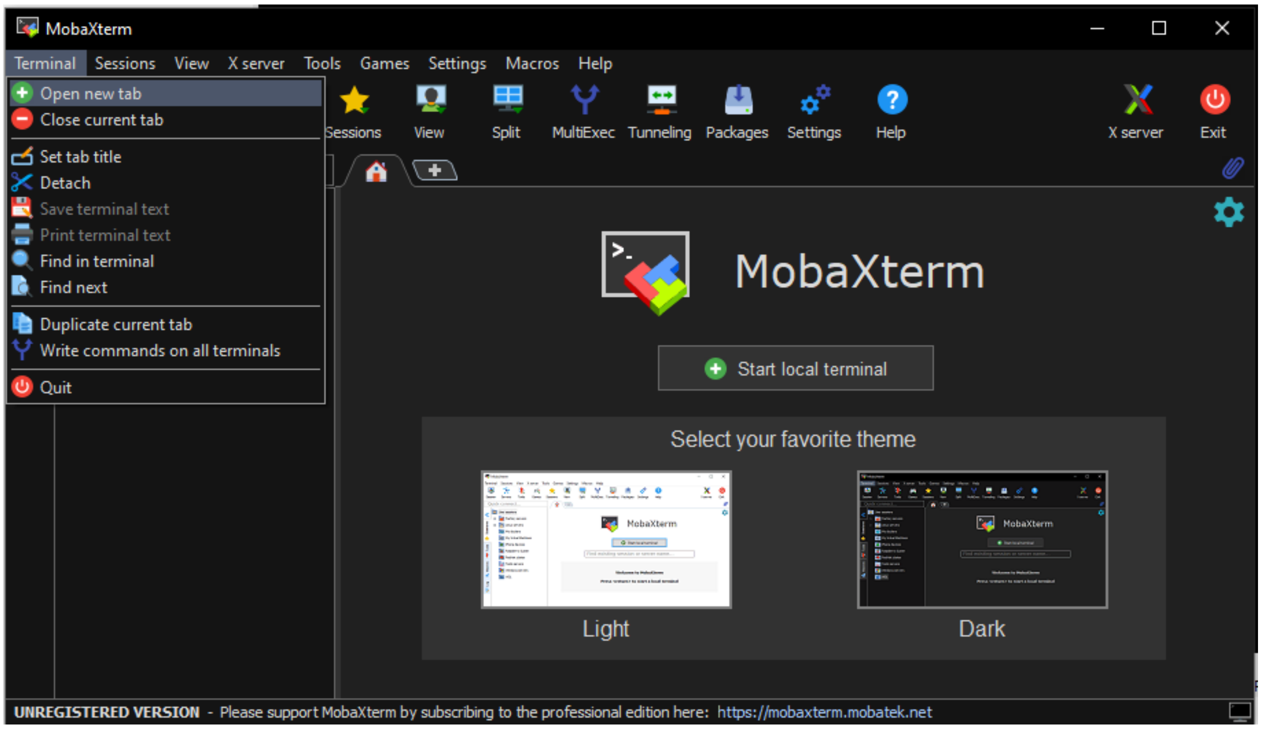
Your display may look different, but take notes about the IP address of the *Host-only* interface card. The example above shows `192.168.56.101`. You can SSH from your host machine into this Ubuntu VM using theis IP address later. (Another IP address, `10.0.2.15` is the IP address of the NAT interface card, the apps in your host machine cannot access it).

1.4 Connect to the Ubuntu-server VM via SSH

1.4.1 Launch MobaXterm

On your host system (in my case - Windows) launch MobaXterm, which is my favourite SSH client. You can download it from here.

- Open new terminal tab



UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

- Paste this command into the terminal

```
ssh <host_only_network_IP_adrress_of_ubuntu_server> -l <username_on_ubuntu_server>
```

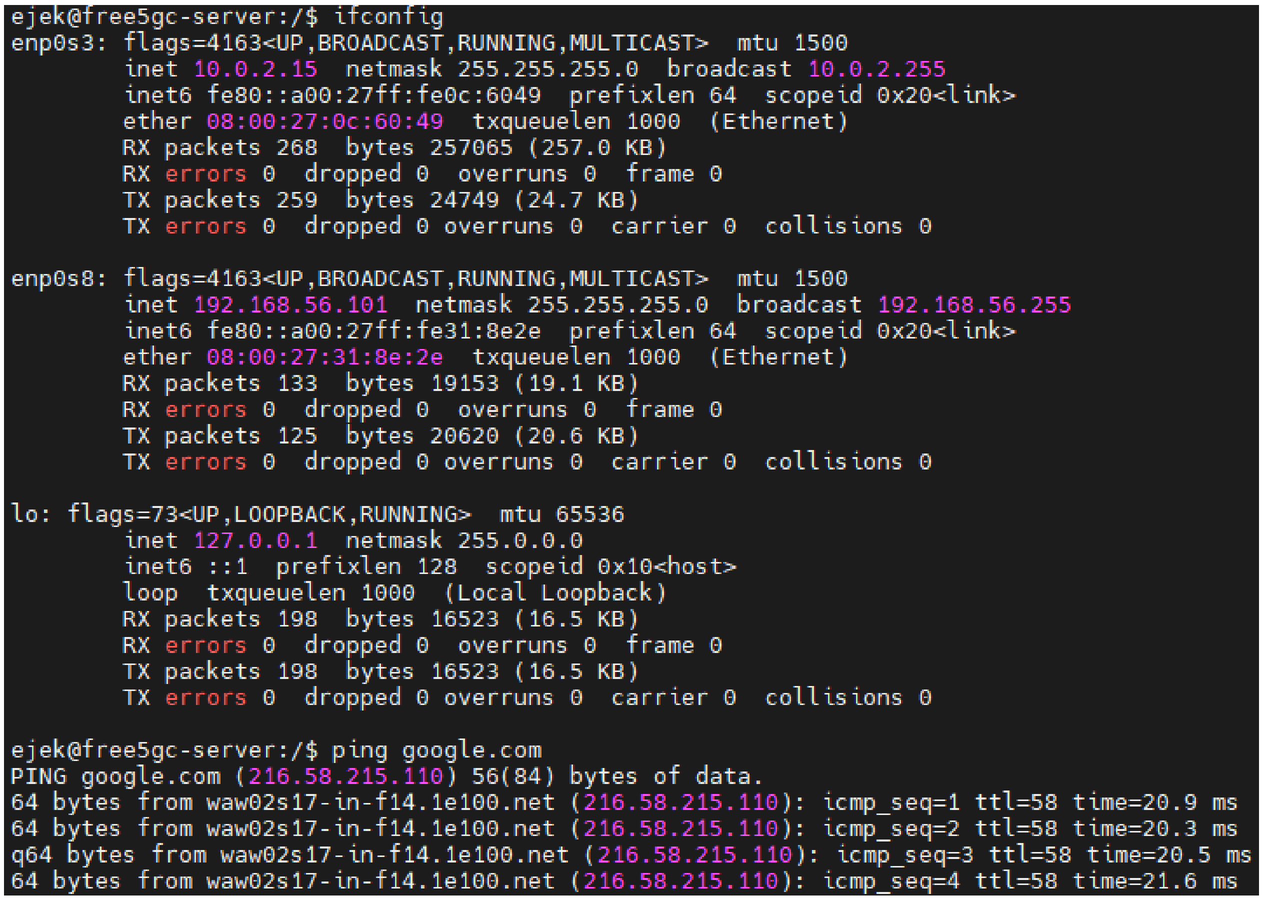
- In my case it is:

```
ssh 192.168.56.101 -l ejek
```

1.4.2 Test SSH connection

Check if you can perfom following actions:

- `ping google.com`
- `ifconfig`



```
ejek@free5gc-server:/$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0c:6049  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0c:60:49  txqueuelen 1000  (Ethernet)
        RX packets 268  bytes 257065 (257.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 259  bytes 24749 (24.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe31:8e2e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:31:8e:2e  txqueuelen 1000  (Ethernet)
        RX packets 133  bytes 19153 (19.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 125  bytes 20620 (20.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 198  bytes 16523 (16.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 198  bytes 16523 (16.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ejek@free5gc-server:/$ ping google.com
PING google.com (216.58.215.110) 56(84) bytes of data.
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=1 ttl=58 time=20.9 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=2 ttl=58 time=20.3 ms
q64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=3 ttl=58 time=20.5 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=4 ttl=58 time=21.6 ms
```

From now on the communication with ubuntu-server can be done by SSH connection.

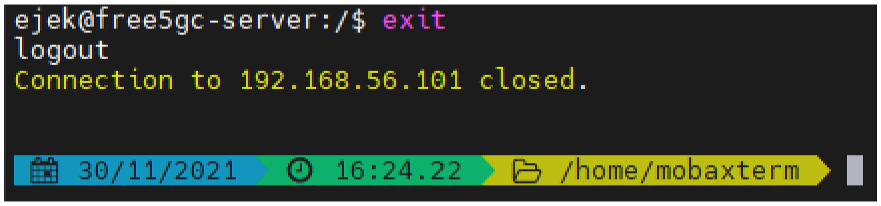
1.5 Update and Upgrade your Ubuntu

Let also update and upgrade the Ubuntu VM right now to make sure it is up-to-date with proper security updates.

```
sudo apt update
sudo apt upgrade
```

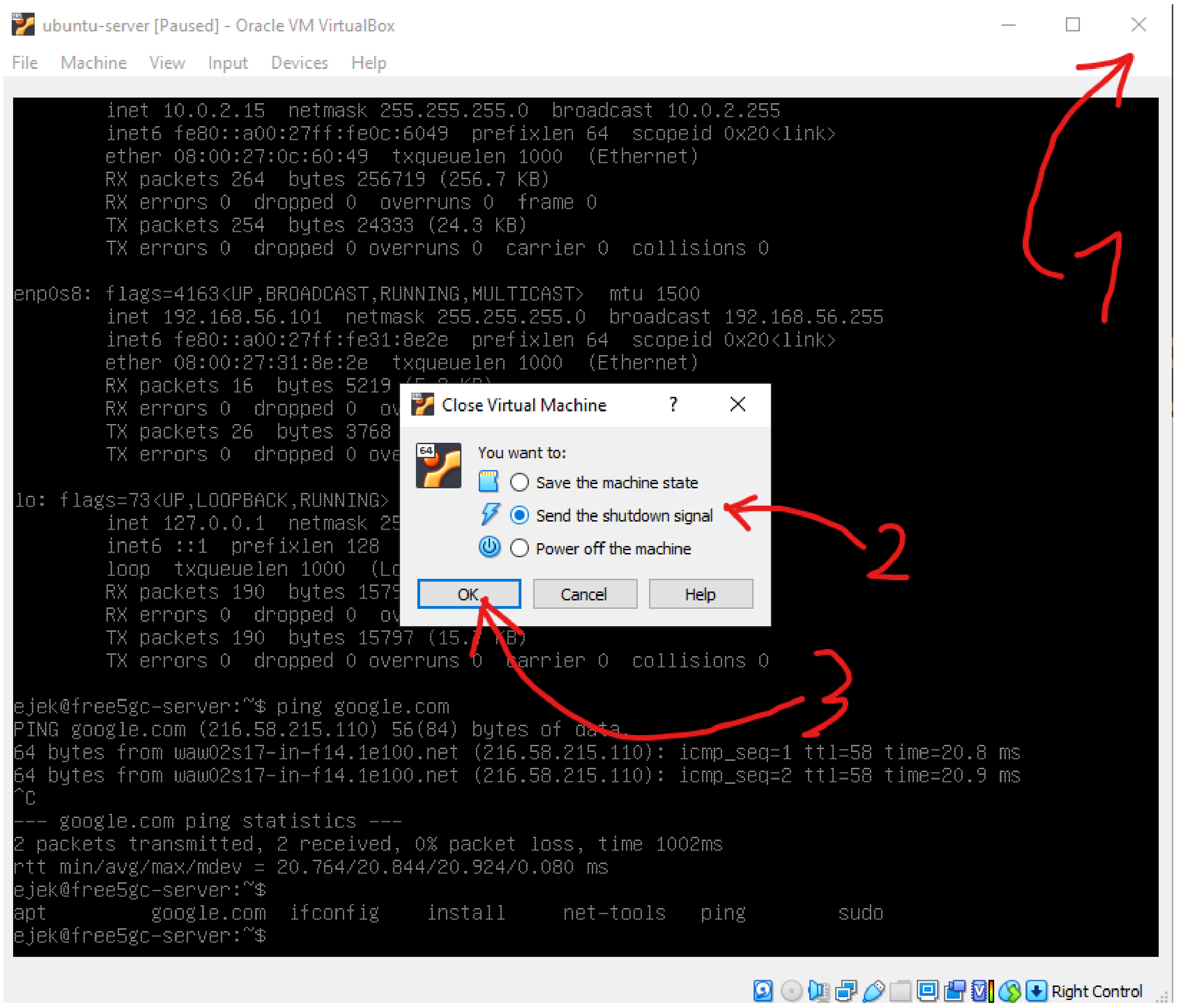
1.6 How to exit the configuration

- In SSH Client you can type `exit` to close the connection.



```
ejek@free5gc-server:/$ exit
logout
Connection to 192.168.56.101 closed.

 30/11/2021   16:24.22   /home/mobaxterm
```

- You can close the SSH Client now.
- On the Ubuntu-server VM just click the "X" to close the window. When the dialog pops-up, select "Send the shutdown signal"


ubuntu-server [Paused] - Oracle VM VirtualBox
File Machine View Input Devices Help
1
2
3
Right Control

```
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe0c:6049  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0c:60:49  txqueuelen 1000  (Ethernet)
        RX packets 264  bytes 256719 (256.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 254  bytes 24333 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe31:8e2e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:31:8e:2e  txqueuelen 1000  (Ethernet)
        RX packets 16  bytes 5219
        RX errors 0  dropped 0  ov
        TX packets 26  bytes 3768
        TX errors 0  dropped 0 ove

lo: flags=73<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1  netmask 25
        inet6 ::1  prefixlen 128
        loop  txqueuelen 1000  (Lo
        RX packets 190  bytes 1579
        RX errors 0  dropped 0  ov
        TX packets 190  bytes 15797 (15.  KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ejek@free5gc-server:~$ ping google.com
PING google.com (216.58.215.110) 56(84) bytes of data.
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=1 ttl=58 time=20.8 ms
64 bytes from waw02s17-in-f14.1e100.net (216.58.215.110): icmp_seq=2 ttl=58 time=20.9 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 20.764/20.844/20.924/0.080 ms
ejek@free5gc-server:~$
apt          google.com  ifconfig    install     net-tools   ping        sudo
ejek@free5gc-server:~$
```

Close Virtual Machine

You want to:

- ○ Save the machine state
- [✓ Send the shutdown signal]
- ○ Power off the machine

OK | Cancel | Help

You can close the Virtual Box window.

1.7 How to run it the next time

1. Launch Oracle VM VirtualBox Manager
2. Select "ubuntu-server" VM and start it.
3. Launch SSH Client (e.g. MobaXterm)
4. Type in `ssh <host_only_network_IP_adrress_of_ubuntu_server> -l <username_on_ubuntu_server>`
    1. e.g `ssh 192.168.56.101 -l ejek`

Step 2 - Clone VM and setup network

2.1 Check up an existing VM for Cloning

Shutdown the machine if it is running.

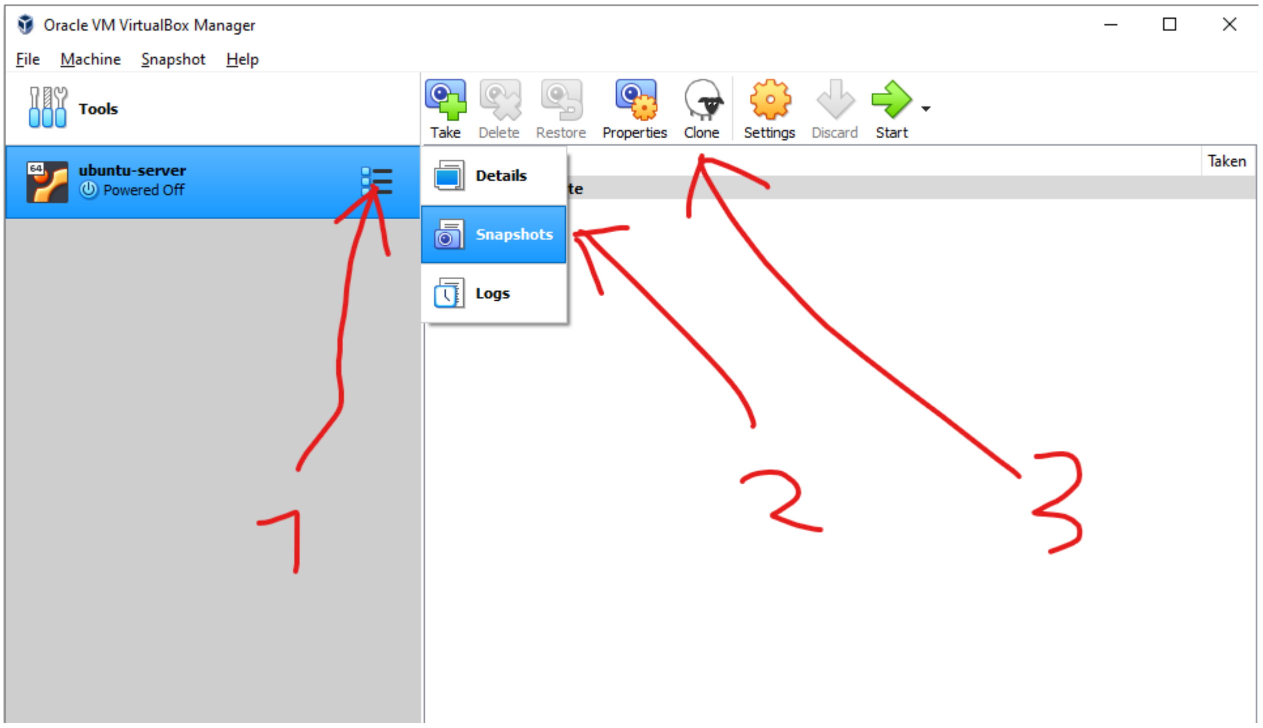
Launch VirtualBox, and make sure the Ubuntu VM (ubuntu) we created before can boot up, then:

- Log in into the VM using SSH from the host machine, and check if the VM has internet access
    - `ssh 192.168.56.101 -l ejek`
    - `ping google.com`
- Make sure you have done `sudo apt update` and `sudo apt upgrade` (or you can do it again)
- Shutdown the VM
    - `sudo shutdown -P now`

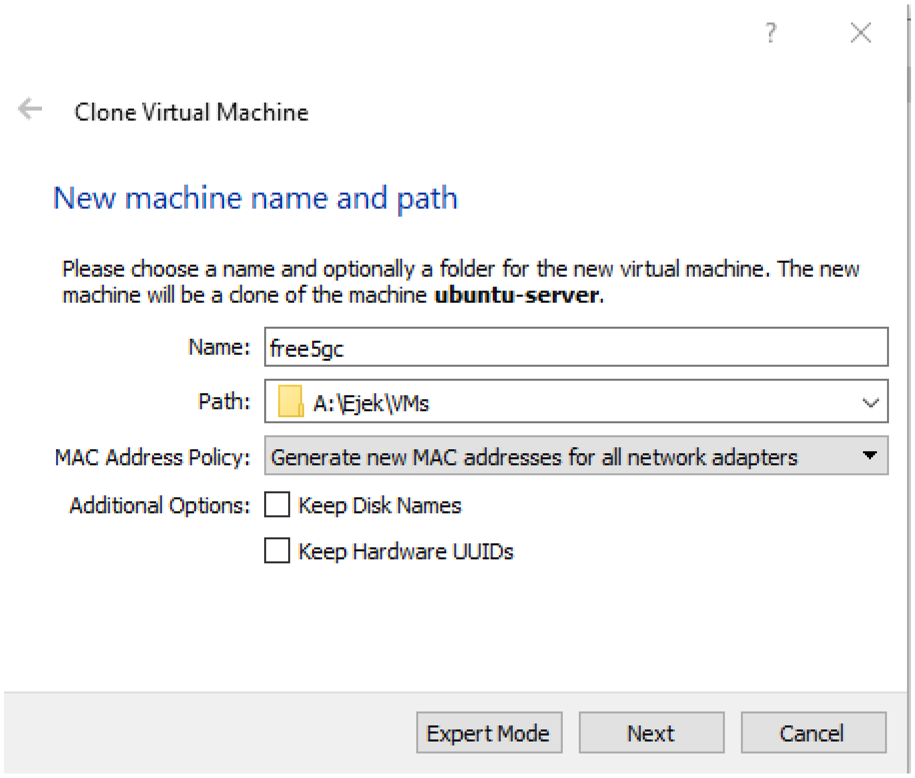
VM is ready to clone!

2.2 Clone the VM

- From the virtual box select ubuntu-server VM, and from "snapshots" options select "clone".


Oracle VM VirtualBox Manager
File Machine Snapshot Help
Tools
ubuntu-server
Powered Off
Take Delete Restore Properties Clone Settings Discard Start
Details
Snapshots
Logs
Taken
1
2
3

- Name it "free5gc" and select correct MAC Address Policy



Clone Virtual Machine

New machine name and path

Please choose a name and optionally a folder for the new virtual machine. The new machine will be a clone of the machine **ubuntu-server**.

Name: free5gc

Path: A:\Ejek\VMs

MAC Address Policy: Generate new MAC addresses for all network adapters

Additional Options: ☐ Keep Disk Names

☐ Keep Hardware UUIDs

- In the next window its your choice, both (linked and full) options will do.
- After the new VM is created
    - Start up the new free5gc VM, and use the same username and password to log in
    - In the Ubuntu terminal run `ping google.com` to make sure it has internet access, and `ifconfig` to note IP address of the Host-only network interface
        - for example the IP could still be `192.168.56.101`, and interface name is `enp0s8`
    - Log in into free5gc VM using SSH, and run the same commands again, to check it SSH work properly

2.3 Change hostname

VM still has host name that you gave to the original one (or ubuntu if you left it default). Let's rename it to `free5gc`.

```
sudo nano /etc/hostname
```

In the file, change `<previous_name>` into `free5gc`。 If you are using `nano` ， you can press `Ctrl-O` to save the file, `Enter` to confirm name, and then `Ctrl-X` to exit.

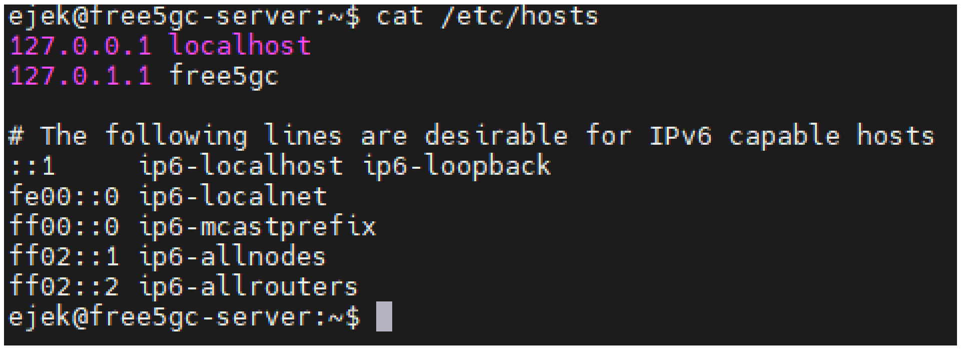
Check if changes are saved.

```
cat /etc/hostname
```

Let's also change the file `/etc/hosts/` by replacing `<previous_name>` with `free5gc`.

```
sudo nano /etc/hosts
```

Content of `/etc/hosts/` should look like this:



```
ejek@free5gc-server:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 free5gc

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ejek@free5gc-server:~$
```

The changes will take effect after next reboot.

```
sudo shutdown -r now
```

2.4 Set Static IP Address

Connect to the machine again.

The Host-only network interface, by default, gets its IP address through DHCP. The cloned free5gc VM seems to have trouble obtaining new IP address. We can change the host-only interface to use static IP address instead, which can save a lot of trouble later. So we will fix the static IP address as `192.168.56.101`.

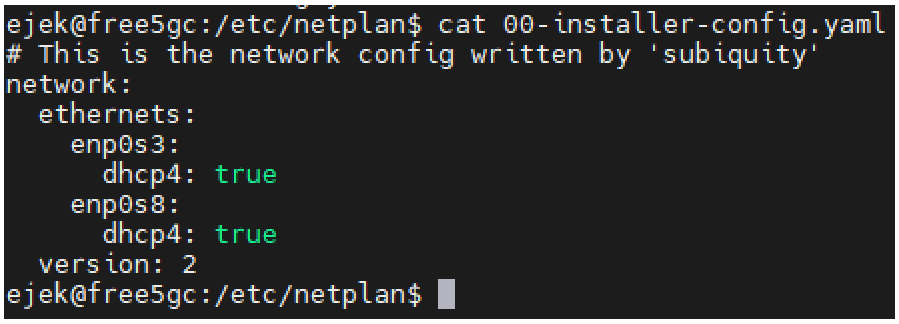
```
cd /etc/netplan
ls
```

The only file that will be shown is `00-installer-config.yaml`

Let's display its content:

```
cat 00-installer-config.yaml
```

Hint: You can use tab to autocomplete such long names as this one.



```
ejek@free5gc:/etc/netplan$ cat 00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: true
  version: 2
ejek@free5gc:/etc/netplan$
```

It means the VM has two interfaces:

- enp0s3
- enp0s8
  - which we know is the Host-only network Interface

To fix a static IP to the interface, we need to disable dhcp protocol and add `addresses` attribute with value `[<ip_address_1>, <ip_address_2>, ...]`

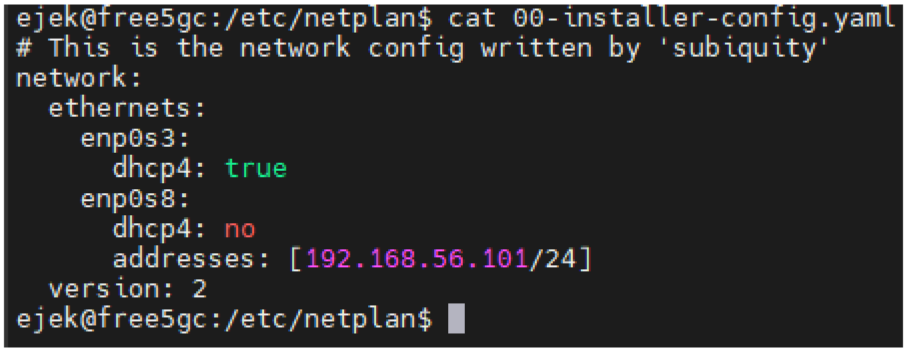
Use nano:

```
sudo nano 00-installer-config.yaml
```

and change it to:

```
# This is the network config written by 'subiquity'
network:
ethernets:
 enp0s3:
   dhcp4: true
 enp0s8:
   dhcp4: no
   addresses: [192.168.56.101/24]
version: 2
```

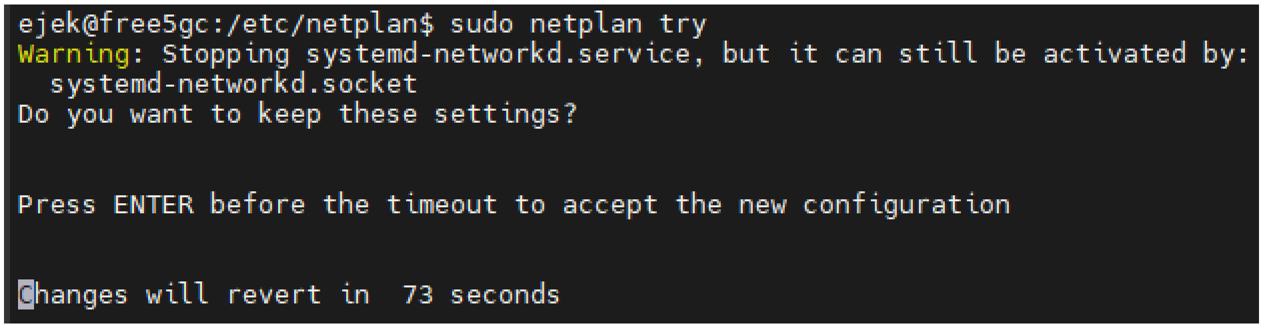
Verify:



```
ejek@free5gc:/etc/netplan$ cat 00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses: [192.168.56.101/24]
  version: 2
ejek@free5gc:/etc/netplan$
```

Now check if the new configuration is correct syntax:

```
sudo netplan try
```



```
ejek@free5gc:/etc/netplan$ sudo netplan try
Warning: Stopping systemd-networkd.service, but it can still be activated by:
  systemd-networkd.socket
Do you want to keep these settings?


Press ENTER before the timeout to accept the new configuration


Changes will revert in  73 seconds
```

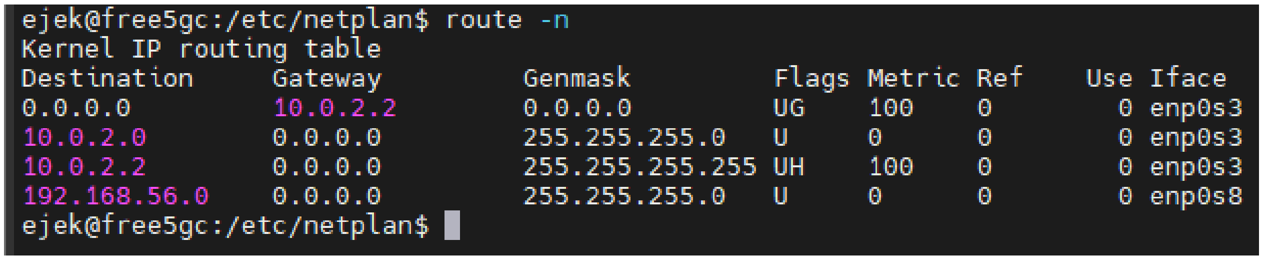
Press Enter to exit and then apply the new interface settings

```
sudo netplan apply
```

Run `ifconfig` to see if the network setting has been changed correctly:

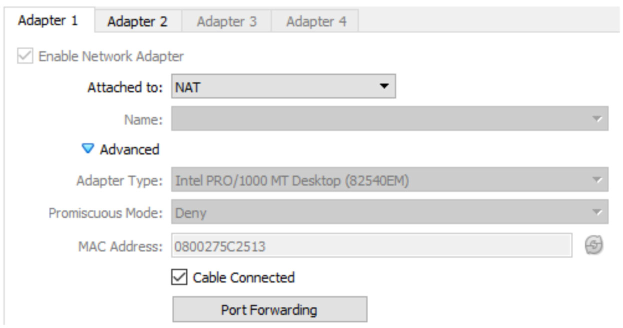
We can also check the routing table, just to have a grasp of what is going on regarding the network setting:

```
route -n
```

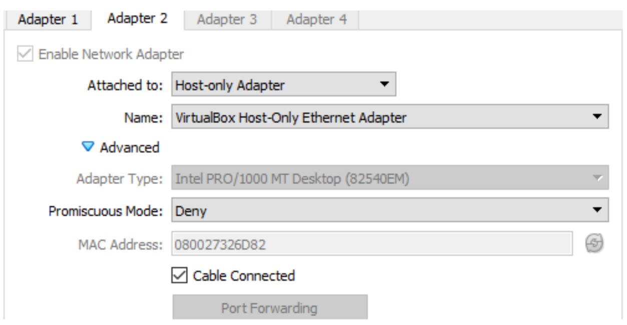


```
ejek@free5gc:/etc/netplan$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
10.0.2.2        0.0.0.0         255.255.255.255 UH    100    0        0 enp0s3
192.168.56.0    0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
ejek@free5gc:/etc/netplan$
```

As we remember the VM has to adapters which we can check in VirtualBox under VM network settings.



☑ Enable Network Adapter
Attached to: NAT
Name:
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 0800275C2513
☑ Cable Connected



☑ Enable Network Adapter
Attached to: Host-only Adapter
Name: VirtualBox Host-Only Ethernet Adapter
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 080027326D82
☑ Cable Connected

- Now the NAT network adapter enp0s3 has IP `10.0.2.2` and it belongs to NAT network `10.0.2.0/24`.
- The host-only adapter enp0s8 has IP `192.168.56.0`

From the display above, we learn that the Host-only network `192.168.56.0/24` does not have internet access by itself (even though we can access it using SSH from the host machine). Internet access is through the NAT network `10.0.2.0/24`, with the gateway being `10.0.2.2` (provided by VirtualBox).

2.5 How to run it the next time

1. Launch Oracle VM VirtualBox Manager
2. Select "free5gc" VM and start it.
3. Launch SSH Client (e.g. MobaXterm)
4. Type in:

```
ssh 192.168.56.101 -l ejek
```

This is how we interact with free5gc VM from now on.

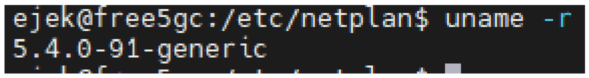
Step 3 Free5GC installation

3.1 Check Linux Kernel version

In order to use the UPF element, you must use the `5.0.0-23-generic` or `5.4.x` version of the Linux kernel. free5gc uses the gtp5g kernel module, which has been tested and compiled against that kernel versions only.

To determine the version of the Linux kernel you are using:

```
uname -r
```



```
ejek@free5gc:/etc/netplan$ uname -r
5.4.0-91-generic
```

Kernel version shown above is ok.

3.2 Install golang

Go is a programming language that was used to develop free5gc. Go was originally written in C language.

- First make sure Golang (go) is not installed:

```
go version
```

If go is installed remove it.

Assuming it is installed at `/usr/local/go`

```
sudo rm -rf /usr/local/go
```

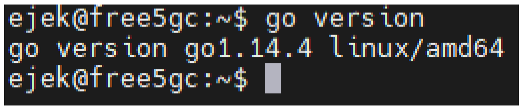
- Install go in version `1.14.4`

```
sudo wget https://dl.google.com/go/go1.14.4.linux-amd64.tar.gz
sudo tar -C /usr/local -zxvf go1.14.4.linux-amd64.tar.gz
mkdir -p ~/go/{bin,pkg,src}
# The following assume that your shell is bash
echo 'export GOPATH=$HOME/go' >> ~/.bashrc
echo 'export GOROOT=/usr/local/go' >> ~/.bashrc
echo 'export PATH=$PATH:$GOPATH/bin:$GOROOT/bin' >> ~/.bashrc
echo 'export GO111MODULE=auto' >> ~/.bashrc
source ~/.bashrc
```

Commands above download a tar package from web, extract it and copy its content to installation folder. Then export some path variables.

- Check if Go is installed (the desired version is `1.14.4`)

```
go version
```



```
ejek@free5gc:~$ go version
go version go1.14.4 linux/amd64
ejek@free5gc:~$
```

3.3 Install tools

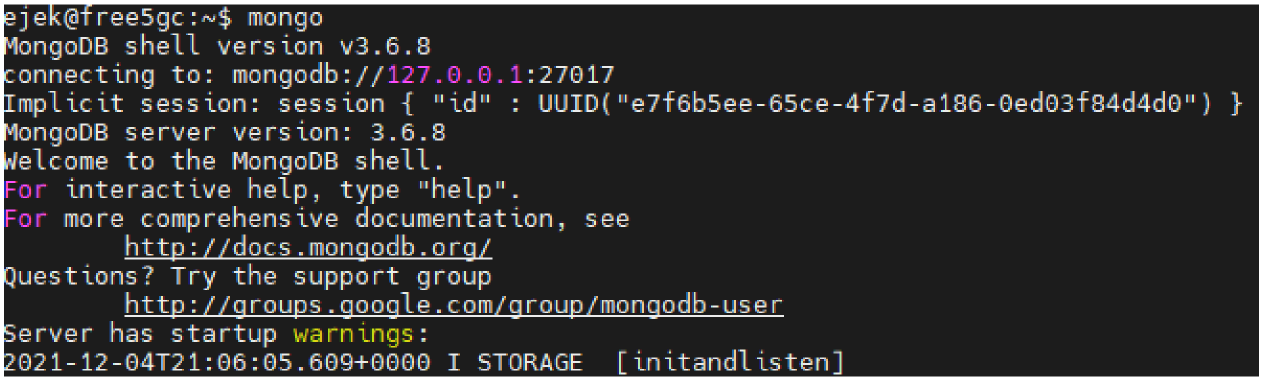
- Install Control-Plane supporting Packages
  - Which is mongodb database

```
sudo apt -y update
sudo apt -y install mongodb
sudo systemctl start mongodb
```

We also started mongodb service

You can check if mongodb is installed, by running its shell:

```
mongo
```



```
ejek@free5gc:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("e7f6b5ee-65ce-4f7d-a186-0ed03f84d4d0") }
MongoDB server version: 3.6.8
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2021-12-04T21:06:05.609+0000 I STORAGE  [initandlisten]
```

You can exit it by typing `exit` or clicking `ctrl+d`

- Install User-plane Supporting Packages
  - Which are some development tools for `go`

```
sudo apt -y update
sudo apt -y install git gcc g++ cmake autoconf libtool pkg-config libmnl-dev libyaml-dev
go get -u github.com/sirupsen/logrus
```

3.4 Setup Networking

3.4.1 Enable IP Forwarding

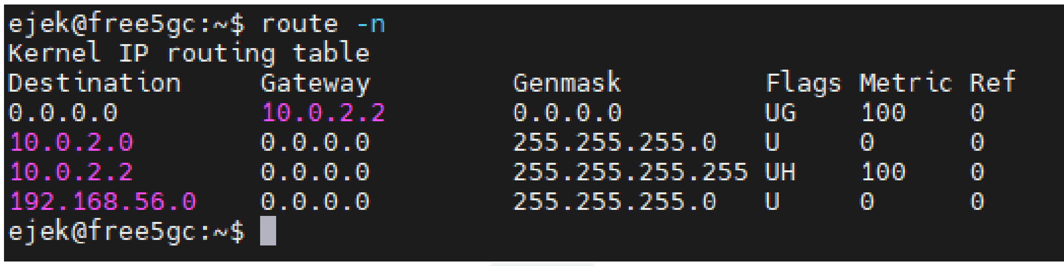
- The term **IP Forwarding** describes sending a network package from one network interface to another one on the same device. It should be enabled when you want your system to act as a router that transfers IP packets from one network to another.

```
sudo sysctl -w net.ipv4.ip_forward=1
```

3.4.2 Set data network interface

Data Network is a term from 5G. For example it can be Internet. We want to know how to access it from our 5G Core.

```
sudo iptables -t nat -A POSTROUTING -o <dn_interface> -j MASQUERADE
```

- Here as `<dn_interface>` you should put the interface, which has access to the internet.
    - 

```
ejek@free5gc:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0
10.0.2.2        0.0.0.0         255.255.255.255 UH    100    0
192.168.56.0    0.0.0.0         255.255.255.0   U     0      0
ejek@free5gc:~$
```

        - You can check that with running `route -n` command and see which Interface is used to reach `0.0.0.0` destination

    - In my case it is

```
sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

3.4.3 Stop firewall

UFW (uncomplicated firewall) is a firewall configuration tool that runs on top of iptables , included by default within Ubuntu distributions. It provides **a streamlined interface for configuring common firewall use cases via the command line**.

We need to stop `ufw` service:

```
sudo systemctl stop ufw
```

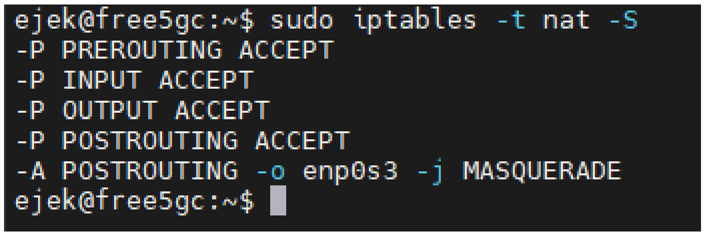
3.4.4 Check setup

- After you've stopped the `ufw` wait a few seconds and check if `ufw` status is `inactive`

```
sudo ufw status
```

- Check `iptables` configuration

```
sudo iptables -t nat -S
```



```
ejek@free5gc:~$ sudo iptables -t nat -S
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -o enp0s3 -j MASQUERADE
ejek@free5gc:~$
```

3.4.5 End notes

Note that these network settings will disappear after reboot. So make sure you run the above commands after each reboot. (You can search the web and find ways to make the settings persistent).

Make sure you run this commands after each reboot:

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
sudo systemctl stop ufw
```

3.5 Install Control Plane Elements

3.5.1 Clone the repository

```
cd $HOME
git clone --recursive -b v3.0.6 -j `nproc` https://github.com/free5gc/free5gc.git
```

With this command we will clone the latest stable build (v3.0.6)

- If You want, You can look around the repository with `ls` and `cd` commands.

3.5.2 Compile Network Function Services

Repository has a makefile, which can be used to build all network function services.

```
cd ~/free5gc
make
```

It may take a few minutes.

3.6 Install User Plan Function

As noted before, the GTP kernel module used by the UPF requires that you use Linux kernel version `5.0.0-23-generic` or `5.4.x`. To verify your version:

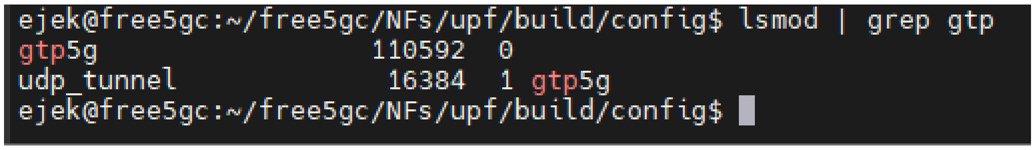
```
uname -r
```

3.6.1 Retrieve the 5G GTP-U kernel module and build it

```
git clone -b v0.4.0 https://github.com/free5gc/gtp5g.git
cd gtp5g
make
sudo make install
```

- To check if gtp5g is installed successfully, see if the following command shows some information:

```
lsmod | grep gtp
```



```
ejek@free5gc:~/free5gc/NFs/upf/build/config$ lsmod | grep gtp
gtp5g                 110592  0
udp_tunnel             16384  1 gtp5g
ejek@free5gc:~/free5gc/NFs/upf/build/config$
```

3.6.2 Build UPF

It was done in step 5.2.

3.6.3 Customize UPF

Customize the UPF as desired. The UPF configuration file is `free5gc/NFs/upf/build/config/upfcfg.yaml`.

As for now we won't change anything there.

3.7 Install web console

3.7.1 Install nodejs and yarn packages

```
sudo apt remove cmdtest
sudo apt remove yarn
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
sudo apt-get update
sudo apt-get install -y nodejs yarn
```

3.7.2 Build WebConsole

```
cd ~/free5gc
make webconsole
```

3.7.3 Check if WebConsole is installed

- Run WebConsole server

```
cd ~/free5gc/webconsole
./bin/webconsole
```

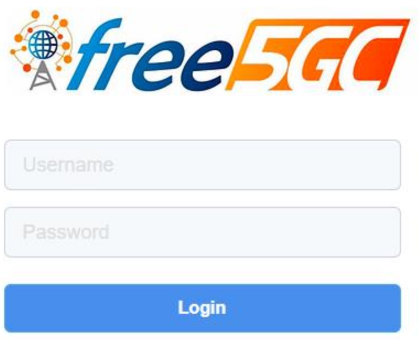
- On your Host-System (in my case Windows 10) open your favorite browser and go to url:

```
http://<free5gc_VM_ip_address>:5000/#/
```

In my case it is:

```
http://192.168.56.102:5000/#/
```

- If everything is ok, you should see a login screen



```
Username: admin
Password: free5gc
```

Step 4 Test installation

free5GC provides some testing procedures to make sure it works properly. First let's just test the basic registration procedure:

```
cd ~/free5gc
./test.sh TestRegistration
```

If everything runs properly without "red" error messages, and the word "PASS" appears near the end of the screen output, then free5GC is running properly.

We can further check other free5GC procedures:

```
./test.sh TestGUTIRegistration

./test.sh TestServiceRequest

./test.sh TestXnHandover

./test.sh TestDeregistration

./test.sh TestPDUSessionReleaseRequest

./test.sh TestPaging

./test.sh TestN2Handover
```

```
./test.sh TestNon3GPP

./test.sh TestReSynchronisation

./test_ulcl.sh -om 3 TestRegistration
```