

4. VirtualBox

4.1 Type 2 Hypervisor: VirtualBox

VirtualBox is a free virtualization software offered by Oracle for creating and running VMs. In this section, we are going to discuss with Type 2 hypervisor. Follow the steps given below for creating a VM and setting up network access for VM on windows platform. **OS installed in physical machine is called a host OS and OS installed on VM is called guest OS.** Make sure you have at least dual core, 4 GB memory, and 100 GB HDD in host machine to launch one VM.

1. Enable virtualization: restart the system → go to BIOS → security → enable VTx (hardware-assisted virtualization). Latest computers are shipped with virtualization enabled by default.
2. Download latest VirtualBox from <https://www.virtualbox.org/wiki/Downloads> and install on windows OS.

4.1.1 Creating a VM in VirtualBox

3. Launch VirtualBox and click new. Now, you are going to create virtual hardware and install guest OS


Create virtual hardware

4. Type the name of VM and select the OS (Linux → Ubuntu 64 bit) you want to install.
5. Select size of virtual memory you want (minimum 1 GB).

Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name:

Type: 

Version:

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024 MB**.



6. Create a virtual hard disk (a file will behave as a HDD).
7. Select hard disk file type: select VMDK
8. Dynamically allocated (occupies storage as it needs).
9. You can set the size of virtual HDD (minimum is 8 GB).

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

- ☐ Do not add a virtual hard disk
- ☒ Create a virtual hard disk now
- ☐ Use an existing virtual hard disk file

hadoop.vmdk (Normal, 8.00 GB)

Create

Cancel

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- ☐ VDI (VirtualBox Disk Image)
- ☐ VHD (Virtual Hard Disk)
- ☒ VMDK (Virtual Machine Disk)

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow (dynamically allocated) or if it should be created at its maximum (fixed size).

A **dynamically allocated** hard disk file will only use space on the hard disk as it fills up (up to a maximum **fixed size**), although it again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems, but is often faster to use.

You can also choose to **split** the hard disk file into several files of gigabytes each. This is mainly useful if you wish to store the virtual machine on removable USB devices or old systems, some of which cannot handle large files.

- ☒ Dynamically allocated
- ☐ Fixed size
- ☐ Split into files of less than 2GB

Next

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

hadoop

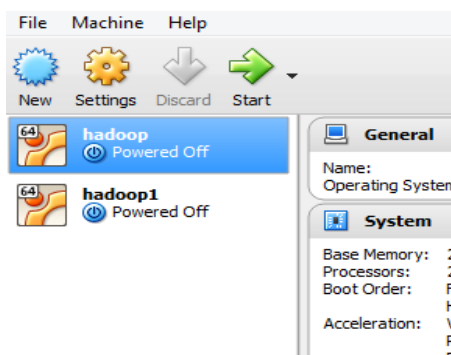
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4.00 MB 10.00 GB 2.00 TB

Create

Cancel

10. Now virtual hardware has been created and displayed in the left-hand side of VirtualBox as shown below.



Install guest OS on virtual hardware

11. Download **Ubuntu 16** server (headless/without GUI version) .iso image file to your local machine from <https://www.ubuntu.com/download/server>
12. Right click on your VM name → settings → storage → storage devices → controller: IDE → add optical drive → choose disk → choose the OS iso file.
13. Right click on the VM name → start → follow the Linux installation procedure given in Linux chapter.
14. The guest OS is successfully installed and launch your first VM.
15. Create 2 or more VMs in a similar way. Remember, you need 8 GB memory and dual-core (hyper-threaded processor) to create two or more VMs.

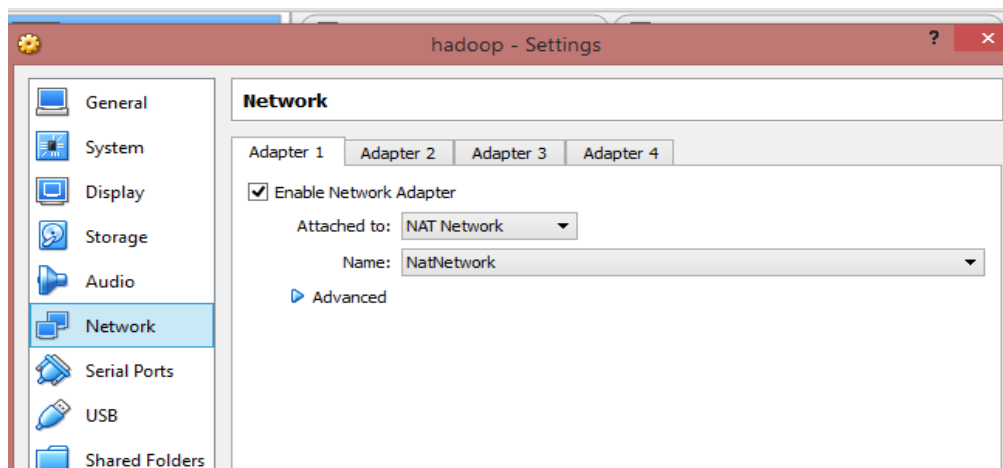
Note: physical server nodes in the real cluster do not have CD/DVD drives to install OS, mouse, keyboard, monitor... Then, how can we install the OS on virtual hardware? So, we use software like Poweriso, magiciso, magicdisc, daemon tools to create virtual CD/DVD drive to create VM. Therefore, you need not go near to any server in a cluster and connect CD/DVD drive to install OS on virtual hardware. However, using step 12, it can be accomplished.

4.1.2 Virtual network for VMs

You can't connect physical Ethernet cable, setup switch... for VMs. Virtual ethernet, virtual switch, virtual NIC... are provided by VirtualBox itself create a virtual network.

Virtual NIC: provides functionality of real NIC

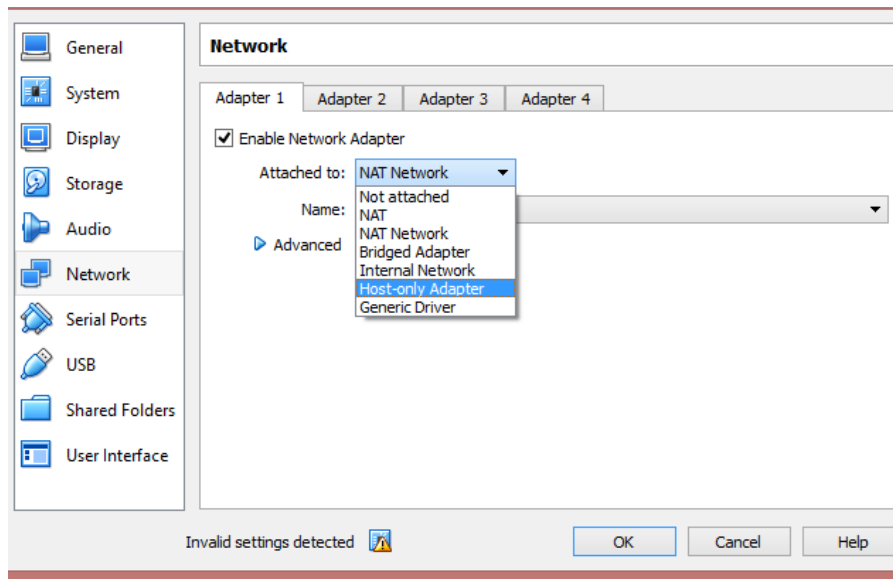
Right click a VM → settings → networks → 4 virtual NICs (adapters) are provided for each VM as shown below.



Virtual switch/router: functionality is done by VirtualBox.

Virtual IP: assigning IP for a VM is called virtual IP. Even if you set a static IP for VM, while coming out of the virtual router (VirtualBox), it is again swapped with host IP (NAT function) until it reaches gateway of the real network. It is called the **IP swapping**.

All these virtual components perform the same functionality of real components. Select a VM → right click → settings → network → Enable network adapter → there are different options to attach VM to network as shown below.



NAT – You need not set any network configurations after VM installation to access internet. By default, internet connection is provided by a VirtualBox for VM, which is sufficient to browse something on the web. It is like a private network under virtual router (VirtualBox). Real world can't communicate with this VM as there is no specific IP address set to VM.

Bridged networking – Attaching one network to another network is called bridging the network. We will attach virtual network which is in one physical machine with the physical LAN. Once done, your VM also will behave as if it is one of the systems in the LAN. We must assign a LAN IP for VM to communicate with the internet and host.

Internal network – Provides communication among VMs via virtual switch.

Host-only network – Communication is only between host and VMs and cannot connect to the internet.

The following table exhibits the connections possible with different network settings in VirtualBox. There are four virtual NICs possible for each VM. You can set any network type to any virtual NIC.

VM network type	VM ↔ Host	VM1 ↔ VM2	VM → Internet	Internet → VM
NAT	No	no	Yes	port forwarding
NAT Network	No	yes	Yes	port forwarding
Bridged	Yes	yes	Yes	Yes
Internal	No	yes	No	No
Host-only	Yes	yes	No	No

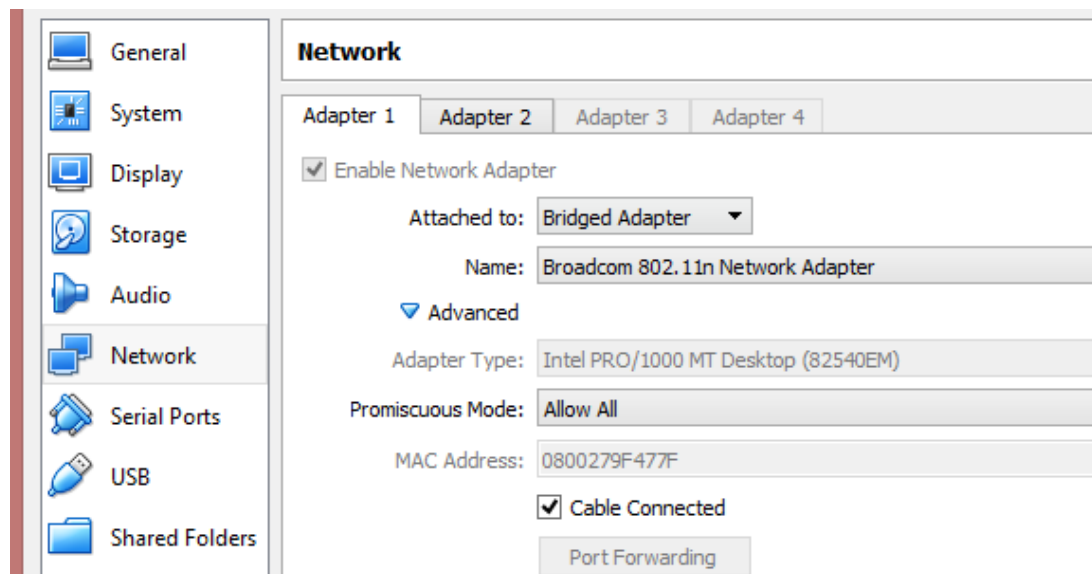
4.1.3 Setting up network for VM

This network connection will give access to internet, host, and neighbor VMs to form clusters. We are going to use two NICs to achieve

- connection between host to VMs
- connection from VM to internet
- connection among VMs

1. Install VM. Then, power off VM before you configure the virtual hardware.

Select a VM name → right click → settings → network. There are 4 NICs possible, but, we are going to use first two NICs. NIC1 is for bridged network and NIC2 is for internal network.



Select Adapter 1: NIC 1 (this NIC is used to bridge VM to the local network. So that, the VM will be seen by other computers in local network as a separate physical computer)

Select Enable network adapter

Attached to: Bridged Adapter // bridging VM to either wired or wireless of host

Name: Broadcom 802.11n network adapter // because my host is connected wi-fi.

If connected to ethernet, then select adapter accordingly.

Advanced

Adapter type: Intel PRO/1000 MT Desktop (82540EM) by default

Promiscuous mode: allow all // allow your VM IP to be seen outside

MAC address is auto generated

Select cable connected

Select Adapter 2: NIC 2 (this is used to create virtual LAN in the host computer to connect more than one VMs on the same host. So that, all VMs on a host will be able to communicate each other)

Select Enable network adapter

Attached to: internal network

Name: any name

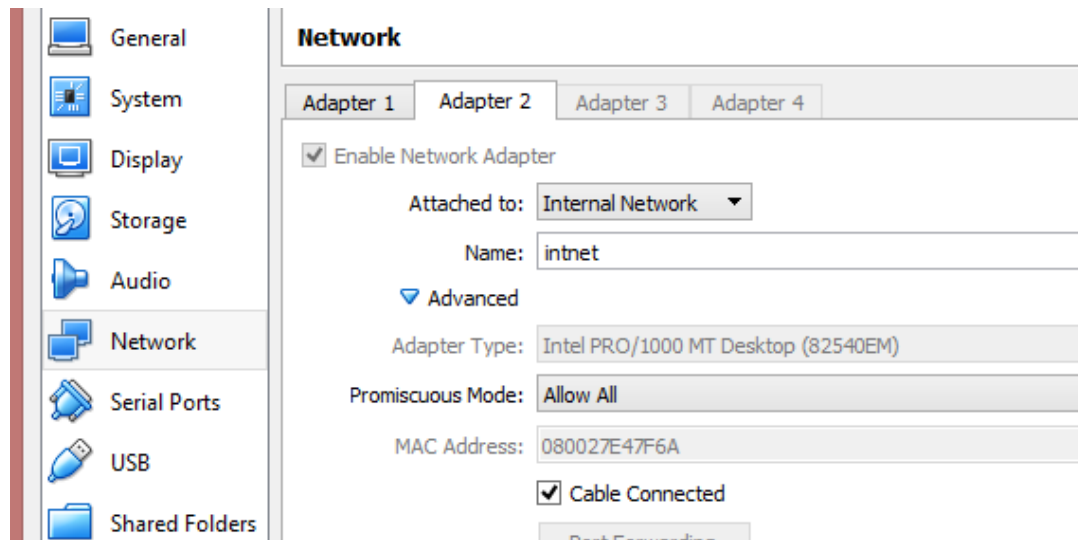
Advanced

Adapter type: Intel PRO/1000 MT Desktop (82540EM) by default

Promiscuous mode: allow all

MAC address is auto generated

Select cable connected



Note: If bridged networking used in adapter1, VMs can communicate each other without setting internal network in adapter2. However, if host only adapter is set in adapter 1 in all VMs, then you have to set internal network in adapter2 of all VMs to establish communication among VMs.

Adapter 3 and 4 are not used

Done

Close the VirtualBox and launch again.

1. Log into your VM.

By default, after installation, the VM will be able to connect to internet as VirtualBox provides DHCP. So, VM need not be set with any IP address. But, nobody will be able to identify your VM from outside of the host as it is not assigned with any local IP address. Therefore, we **need to set static IP**.

\$ ifconfig -a // to find available NIC interface (we have enlo

Ubuntu 16 or before

\$ sudo vi /etc/network/interfaces

auto lo

```

iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
address 10.100.54.31
netmask 255.255.0.0
gateway 10.100.54.1
dns-nameservers 10.20.1.22 10.20.1.21 10.3.0.101 4.2.2.2 8.8.8.8

```

enp0s3 denotes NIC 1

Available IP on my local network is 10.100.54.31

Assign netmask, gateway, and nameservers according to your local network

At this point you need not explicitly set anything for NIC 2. Because, it has been set with the internal network for virtual LAN. Restart the VM.

\$ sudo reboot

Ubuntu 17 or after

NetPlan is a new network configuration tool introduced in Ubuntu 17 to manage network settings. Therefore, you should not touch **/etc/network/interfaces** file to configure static IP as you did in Ubuntu 16 and previous versions. Instead, you have to edit **/etc/netplan/*.yaml** file. Data is entered as key:value pair in yaml file. To configure static IP,

\$ sudo vi /etc/netplan/01-netcfg.yaml // in my machine yaml file name is 01-netcfg.yaml

```

network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: no
      dhcp6: no
      addresses: [10.100.54.31/16]
      gateway4: 10.100.54.1
      nameservers:
        addresses: [10.20.1.22,10.20.1.21,10.3.0.101,4.2.2.2,8.8.8.8]

```

Once, static IP is set, restart the network interfaces

\$ sudo netplan apply

\$ sudo netplan --debug apply // gives you the error details if any

Caution:

1. Be careful with spacing between key: value and in other places.
2. 10.100.54.31/16 is a CIDR notation. I have used 16 in 10.100.54.31/16 as my local network netmask is 255.255.0.0. You have to include respective number of appropriate netmasks of your local network. Ex:

CIDR notation for 255.255.255.0 is 24

CIDR notation for 255.255.0.0 is 16

You can refer <https://doc.m0n0.ch/quickstartpc/intro-CIDR.html> for more.

If you have more than one VM on a same host, then follow the same procedure. I have assigned IP of VM2 is 10.100.54.32 and VM3 IP is 10.100.54.33. Host machine IP is 10.100.54.30.

4. Assign the following domain name in all three VMs.

```
$ sudo vi /etc/hosts
```

```
host      10.100.54.30
node1     10.100.54.31
node2     10.100.54.32
node3     10.100.54.33
```

5. Checking connectivity: Launch three VMs and try the following commands in all VMs.

```
$ ping google.com
```

```
$ ping host
```

```
$ ping node1
```

```
$ ping node2
```

```
$ ping node3
```

All these VMs are in one physical machine. Ping these VMs from other systems in the LAN. VM will behave as if one of the machines in the LAN. If we get the response back, then we are done with network connections of VMs.

Download putty software in your host and launch it → enter node1 IP → enter username and password. Now, you are connected with VM1, similarly you can connect with other VMs too. Similar way, you can connect to VMs from any physical machine using putty.

Note: If a firewall is enabled or anti-virus is installed in the host, then the connection between VM and host may not be successful via putty. Therefore, disable firewall or uninstall anti-virus in the host → restart the host → launch VirtualBox → launch VMs.

6. Update and upgrade VM

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

7. You can import/export/clone VMs after shutting down it. You can copy VM from C:\Users\Rathinaraja\VirtualBox VMs location, carry anywhere you want, and install the VM by importing into VirtualBox in another machine. Cloning VMs might lead to network issue for cloned VM. Therefore, export VM to get VM as a file. Then, import that VM file to create more than one VMs quickly rather than installing guest OS from scratch for each VM.

File → export appliance → select VM name → select location → export

File → import appliance → select the VM file (previously exported) → open → import

Once you imported, you have to change the static IP as imported VM's IP will clash with other VMs.

Note: if you want to scale up/down (increase/decrease) VM's memory, HDD... you need to power off VM first.

4.1.4 To share files from host to VM

Execute the following commands in the VM to attach host's local storage.

```
$ sudo apt-get install virtualbox-guest-additions-iso  
$ sudo apt-get install virtualbox-guest-utils  
$ sudo apt-get install build-essential  
$ sudo apt-get install virtualbox-guest-x11
```

Shut down VM → go to VM settings → select shared folders → select add

folder path: local drive path

folder name: name of the folder

auto mount --> ok

start VM

```
$ sudo ls /media/sf_users/
```

4.2 Virtual disk format

There are many virtual disk formats and each has its own benefits. The important thing is to find out which disk image format your hypervisor supports. We shortly discuss some virtual disk formats.

ISO: it is an archive format for the data contents of an optical disc such as CD-ROM.

QCOW2: Qemu Copy On Write is supported by the QEMU emulator that can expand disk space dynamically and supports copy on write.

RAW (.img, .raw): it is an unstructured disk image format. If you have a file without an extension it is possibly a raw format.

VDI: virtual disk image is supported by VirtualBox and the QEMU emulator.

VHD: virtual hard disk is a common disk format used by VMware, Xen, Microsoft, VirtualBox...

VHDX: an enhanced version of the VHD format, which supports larger disk size.

VMDK: common disk format supported by many common hypervisors.

OVF: Open Virtualization Format specification provides a means of describing the properties of a virtual system. It is XML based and has generous allowances for extensibility.

OVA: Open Virtual Appliance is an OVF file packaged (kind of tar file) with all of its supporting files (disk images...).