Steven Gordon

Creating a Virtual Network of Linux Guests using VirtualBox

Submitted by Steve on Wed, 11/09/2013 - 5:41pm

Introduction

> The following instructions were last updated around September 2013. They still should work, but I have sinced moved to a more "automated" approach for creating a virtual network - see the automated approach here (/sgordon/automatic-creation-of-virtual-network-with-vboxmanage). I recommend using the automated approach (which essentially is a set of scripts to do the steps on this page), as opposed to the manual setup described below. However I leave these instructions here for reference.

When teaching data communications and network security I like to allow students to see and use real networks and software that demonstrate basic concepts. Although I teach a lab on networking in which students get some hands on experience, I also like students to be able to try things in their own time. Creating a small network (e.g. a client, router and server) is not feasible for most students as they usually only have a single computer (most often a laptop, sometimes a PC at home). Hence the two options available is to ask students to use network simulation software or to setup a virtual network inside their own computer. This article gives instructions for the latter.

Virtual machine software, like VirtualBox, VMWare and Parallels, allow one or more *guest* operating systems to run as an application on another *host* operating system. The host OS runs on real hardware (your computer), while the guest OS run on virtual hardware, which is hardware emulated by the virtual machine software. The guest OS plus emulated hardware can be simply referred to as a *virtual machine* (VM).

One part of hardware emulated by the virtual machine software is a network interface card. Virtual machine software also emulates network devices, like switches, cables and routers. So a virtual network can be built by creating multiple virtual machines and connecting them together via virtual cables and switches.

I will use VirtualBox (https://www.virtualbox.org/), free virtual machine software that runs on Windows, Linux and Macintosh host operating systems, and supports most operating systems as guest. As most of my teaching uses Ubuntu Linux in demonstrations, and students need to gain experience with the Linux command-line for other courses, I will use a command-line only install of Ubuntu Linux as each of the guests. Specifically, I will use Ubuntu 12.04 LTS Server.

The virtual network topology (including number of nodes) needed will depend on the networking and security tasks to be demonstrated. Therefore my approach is to create a base virtual machine that can be easily copied (cloned) multiple times to create nodes in the virtual network. The following instructions focus on how to setup the base virtual machine, and then cloning the base virtual machine to create a node. To create further nodes, just repeat the relevant steps.

The procedure for setting up a virtual network is:

1. Create a base virtual machine in VirtualBox. This base VM will eventually be cloned to create the actual VMs that for the virtual network.
2. Install and configure Ubuntu in the base virtual machine. The configuration will be done so that once the base VM is cloned, it will be easy to use the cloned VM with minimal extra setup.
3. Clone the base virtual machine to create the nodes. The virtual network will be made up of multiple virtual machines, or *nodes*. Each node will be a clone of the base virtual machine.
4. Configure the nodes for the virtual network. Some minor configuration changes (such as host name, IP addresses) will be needed to setup each node in the virtual network.
5. Use your virtual network.

Terminology

The *host* computer or operating system is the one in which you run VirtualBox (i.e. your real computer). VirtualBox can run *virtual machines*. I will also refer to a virtual machine as the *guest* system (and later *nodes*). We will create a *base* virtual machine, and then clone it multiple times to make *nodes*. The nodes will form a *virtual network*, all running inside VirtualBox on the host. The (virtualised) hardware that connects a virtual machine to a network is referred to as a *(network) adapter*, *network interface card (NIC)* or *(network) interface*.

Prior Knowledge

I assume you have a computer networking background (e.g. understand IP addresses, interfaces, routing) and some basic Linux/Unix command line experience. If this is your first time using the command line interface (CLI) then you still should be able to follow the instructions, however you may not fully understand what is happening. The Free Software Foundation (http://shop.fsf.org/category/books/) offer a good, free book Introduction to the Command Line (http://en.flossmanuals.net/command-line/index/).

Add New Virtual Machine in VirtualBox

Install and start VirtualBox on your host computer.

Create a New Virtual Machine

I am currently using Ubuntu Server 12.04.3 LTS 32-bit, so I recommend using the same version. Download the .iso from Ubuntu (http://www.ubuntu.com/) or from a local mirror (https://launchpad.net/ubuntu/+cdmirrors) (e.g. Kasetsart Uni (http://mirror1.ku.ac.th/ubuntu-releases/12.04.3/ubuntu-12.04.3-server-i386.iso)). The file name will be something like `ubuntu-12.04.3-server-i386.iso` and about 700 MB in size. (i386 indicates the 32-bit version; AMD64 refers to the 64-bit version; I used Ubuntu version 12.04.3).

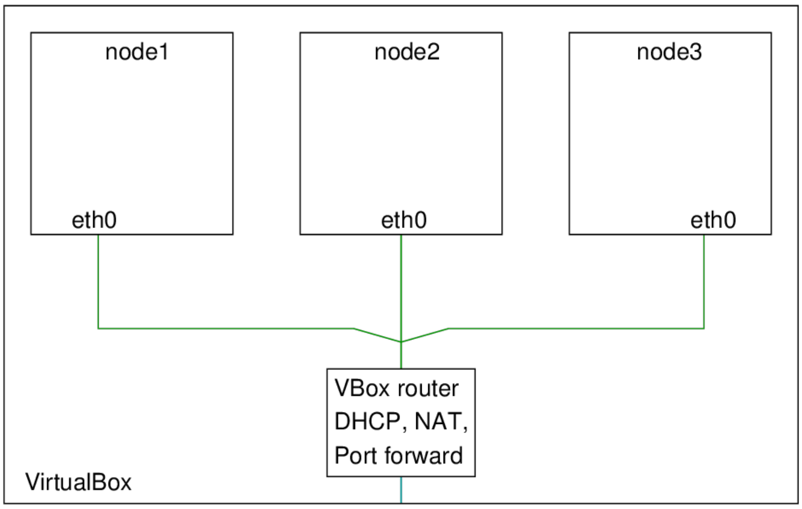
In VirtualBox, create a new virtual machine, called *base*. Two parameters that you must choose for your VM are:

- Memory (RAM): 128 MB
- Hard disk size: 2 GB

Configure Network Adapter Settings

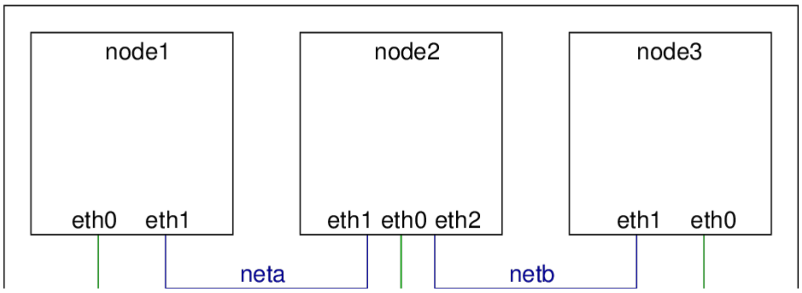
Before you start the virtual machine, edit the settings for the network adapters. By default, VirtualBox enables only the first adapter, but allows for four network adapters. We will configure all of them now (although may not use all four later).

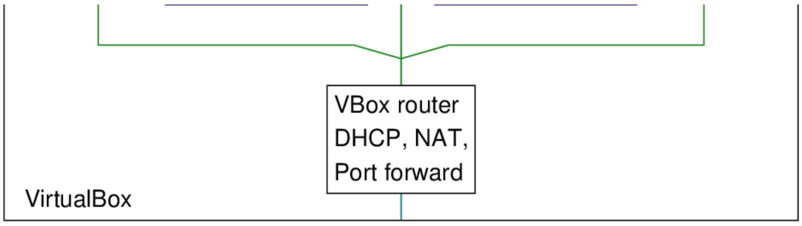
Adapter 1 is configured by default to use NAT. That allows the virtual machine to have Internet access, but not act as a server or contact other virtual machines. So for example if we had three virtual machines called *node1*, *node2*, and *node3*, the configuration using NAT only would look like the following:



Topology for virtual network with NAT only

For our virtual network we want our nodes to communicate with each other. However to simplify the management of the VMs, they should each also have Internet access (e.g. so software updates can be performed) and allow remote SSH connections (e.g. so you can connect to the guest via the host using SSH). The network configuration we are aiming for is in the following figure.

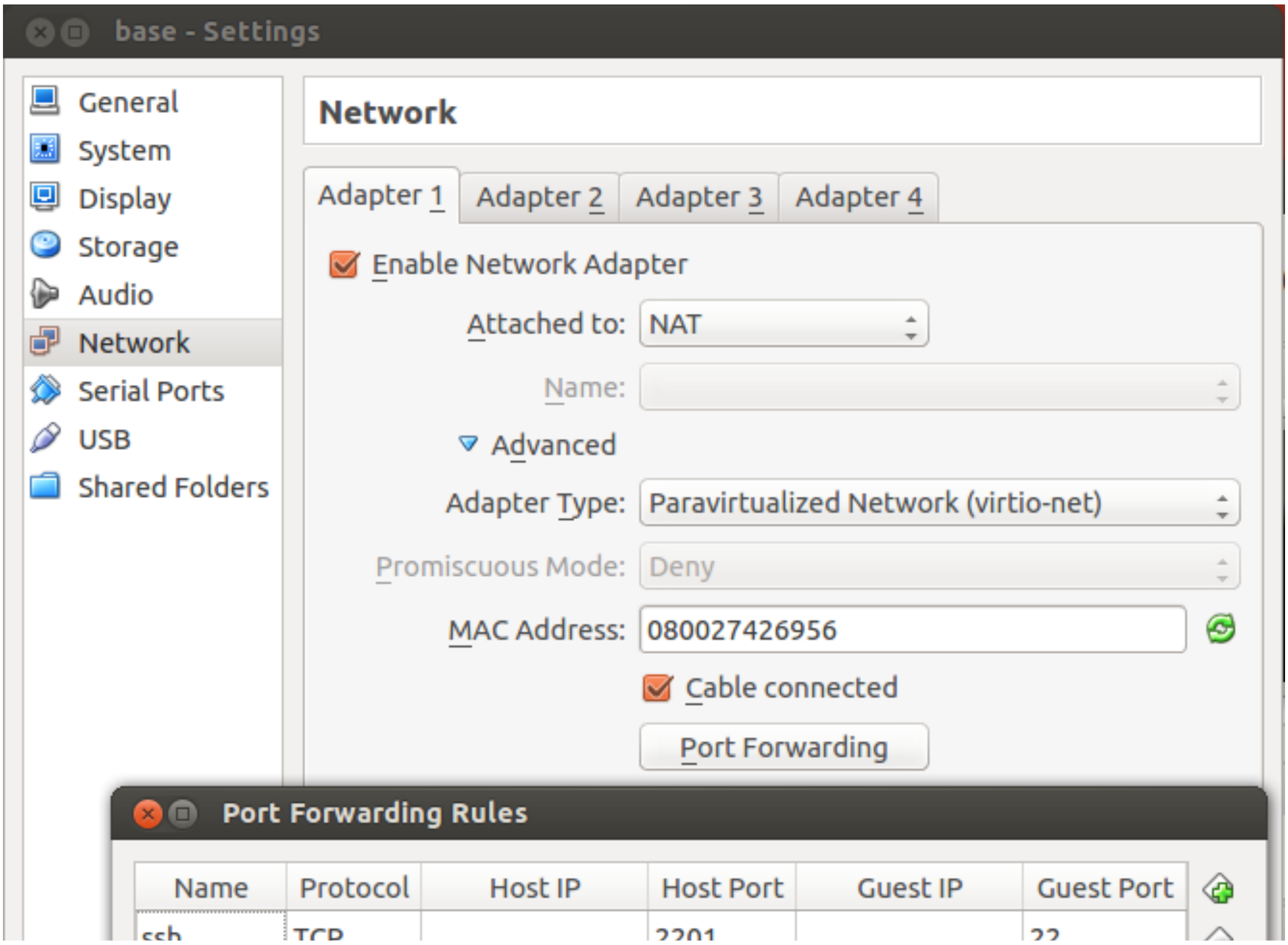




Topology for virtual network with NAT and two internal networks

The first step is to give the base virtual machine extra network adapters. In VirtualBox, find the Network settings to see a list of four potential network adapters. Adapter 1 is already enabled and configured to use NAT. Expand the *Advanced* options. There are two things to change:

1. Press the *Port Forwarding* button and add a new rule with the following parameters:
   - Name: ssh
   - Protocol: TCP
   - Host IP: *empty*
   - Host Port: 2201
   - Guest IP: *empty*
   - Guest Port: 22
2. The *Adapter Type* may optionally be changed to *Paravirtualized Network (virtio-net)*. This gives a small performance increase when using the network. (This step is optional - the default type should still work ok).



☑ Enable Network Adapter

Attached to: NAT

Name:

Advanced

Adapter Type: Paravirtualized Network (virtio-net)

Promiscuous Mode: Deny

MAC Address: 080027426956

☑ Cable connected

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
ssh	TCP		2201		22

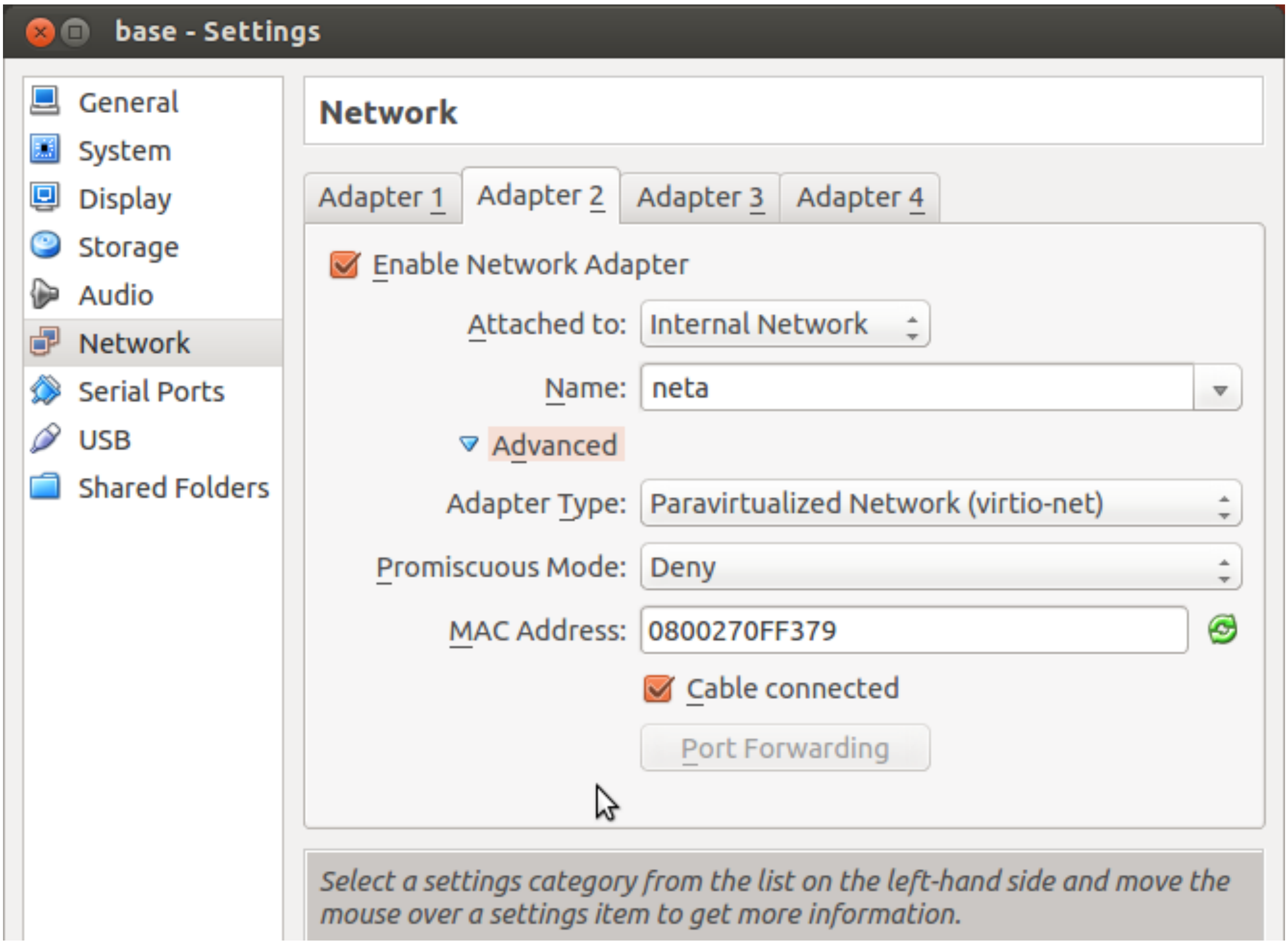
VirtualBox settings for network adapter 1 on base

The port forwarding was setup so that we can [access the virtual machine using SSH (/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling)](/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling). You don't have to use port 2201 - any port number unused by the host should be ok.

Now enable the other three adapaters (2, 3 and 4). For each adapter, use the following values:

- Attached to: Internal Network
- Name: `neta` (for Adapter 2), `netb` (Adapter 3), `netc` (Adapter 4)
- Adapter Type: Paravirtualized Network (virtio-net) [or optionally leave as the default value]
- Promiscuous Mode: Deny
- MAC Address: default value (it should be different across all adapters)
- Cable Connection: yes (checked) for Adapter 2; no (unchecked) for Adapters 3 and 4
- Port Forwarding: no rules

The figure below shows the settings for Adapter 2:



Network

☑ Enable Network Adapter

Attached to: Internal Network

Name: neta

Advanced

Adapter Type: Paravirtualized Network (virtio-net)

Promiscuous Mode: Deny

MAC Address: 0800270FF379

☑ Cable connected

Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.

VirtualBox settings for network adapter 2 on base

Configuring all four network adapters in the base virtual machine will make it easier in configuring nodes cloned from the base.

When you start the VM for the first time, VirtualBox prompts for a disc (iso) to boot from. Select the downloaded Ubuntu .iso image.

Install Ubuntu and Update Software

Install Ubuntu

The Ubuntu installation is rather straightforward. Most default options can be chosen or you can chose values that suit you (e.g. language, keyboard, time zone). There are however several important values to be chosen:

- On the first install menu (the black screen with the menu in the middle, which is displayed after you've chosen the language), press F4 and select *Install a minimal virtual machine*.
- Username and password: I chose *network* and *network*. The virtual machines are only intended to be used on your computer, so securing them with a strong password does not bring many benefits; choose a simple password and re-use it when necessary (e.g. MySQL, root user).
- How to manage upgrades? I chose *No automatic upgrades* as I'd like to keep the versions of software used in virtual networks the same for different users. However choosing automatic upgrades should also be acceptable.
- In the *Software Selection* menu, select: *Basic Ubuntu server*, *OpenSSH server* and *LAMP server*. (Note that if you forget to select software at this step, you can bring this menu up later once Ubuntu is installed by typed `sudo tasksel` on the command line.)

For those for which this is their first time install Ubuntu, I've taken a set of screenshots for my Ubuntu install (/sgordon/installing-ubuntu-as-a-virtual-machine-in-virtualbox) which you can follow.

Start and Login to Base Virtual Machine

At the end of the Ubuntu install procedure you are presented the option to *Continue* to reboot into the virtual machine. Do so, and login using the username and password you created, e.g. *network* and *network*. You should see a prompt like:

```
network@base:~$
```

meaning you are logged in as the user *network* on the host *base* and currently in your home (~) directory.

Install and Update Software

Once Ubuntu is installed and running, you should update the existing software packages and install some additional packages.

Choose a Different Repository Mirror (Optional)

Before you download packages, you may want to change the default Ubuntu repository mirror to another one (https://launchpad.net/ubuntu/+cdmirrors). For example, I will change my default mirror for Thailand (th.archive.ubuntu.com) to a mirror at Kasetsart University (mirror1.ku.ac.th). To do so, you need to edit the file /etc/apt/sources.list. The following will make the changes, as well as keep backups:

```
network@base:~$ sudo cp /etc/apt/sources.list /etc/apt
/sources.list.original
network@base:~$ sed
"s#th.archive.ubuntu.com#mirror1.ku.ac.th#g" /etc/apt
/sources.list > sources.list
network@base:~$ sudo mv sources.list /etc/apt/
```

Use apt-get to update and install packages as follows:

```
network@base:~$ sudo apt-get update
network@base:~$ sudo apt-get upgrade
network@base:~$ sudo apt-get install iperf wget lynx man-db manpages
manpages-dev iputils-tracepath traceroute tcpdump nano ethtool telnet
iptables iputils-arping ettercap-text-only dnsutils iptraf whois
subversion build-essential
```

Optionally, if you did not select *LAMP server* (or OpenSSH server) in the software selection menu during the install, then you can do so now by running tasksel.

```
network@base:~$ sudo tasksel
```

Add Root Password

By default, Ubuntu doesn't enable a login account for root user. Instead sudo should be used by the normal user. In some cases we may need root login and hence we will create a password. I will choose a password which is the same as the normal user (e.g. network):

```
network@base:~$ sudo su
root@base:/home/network# passwd
Enter new UNIX password: network
Retype new UNIX password: network
```

```
passwd: password updated successfully
root@base:/home/network# exit
exit
network@base:~$
```

Configure Networking on the Guest

The base system will have multiple network interfaces, however we may not use all of them:

1. `eth0` is used for normal Internet access. It is configured by the special DHCP server provided by VirtualBox. There is nothing to change with this interface; it should be set correctly already.
2. `eth1` is used for the internal network (to connect to other virtual machines). We will manually set the address for this interface.
3. `eth2` is used in the same way as `eth1`. However initially in VirtualBox we will set the cable to be disconnected. This interface will only be needed on some virtual machines (usually routers).
4. `eth3` is used in the same way as `eth2`.

Now configure addresses for the interfaces. The first interface, `eth0`, should be configured to use DHCP. VirtualBox includes a DHCP server that allocates IP addresses to guests. The other interfaces should be configured with static IP addresses.

The configuration of the `eth1` interface is performed in the file `/etc/network/interfaces`. Edit the file with nano by typing:

```
network@base:~$ sudo nano /etc/network/interfaces
```

The lines for the `lo` and `eth0` interfaces do not need changing. You need to add new lines for the `eth1` interface. Add the following lines to the bottom of the file:

```
auto eth1
iface eth1 inet static
        address 192.168.1.11
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        post-up route add -net 192.168.0.0 netmask 255.255.0.0 gw
192.168.1.1 dev eth1
        pre-down route del -net 192.168.0.0 netmask 255.255.0.0 gw
192.168.1.1 dev eth1

#auto eth2
#iface eth2 inet static
#       address 192.168.2.2
#       netmask 255.255.255.0
#       network 192.168.2.0
#       broadcast 192.168.2.255
#       post-up route add -net 192.168.0.0 netmask 255.255.0.0 gw
192.168.2.2 dev eth2
#       pre-down route del -net 192.168.0.0 netmask 255.255.0.0 gw
```

```
192.168.2.2 dev eth2

#auto eth3
#iface eth3 inet static
#       address 192.168.3.3
#       netmask 255.255.255.0
#       network 192.168.3.0
#       broadcast 192.168.3.255
#       post-up route add -net 192.168.0.0 netmask 255.255.0.0 gw
192.168.3.3 dev eth3
#       pre-down route del -net 192.168.0.0 netmask 255.255.0.0 gw
192.168.3.3 dev eth3
```

What does this do? When the computer (or more precisely the network interface `eth1`) starts, it is given the IP address `192.168.1.11` with subnet mask `255.255.255.0`. The `post-up` line also adds a route to the routing table when the interface starts (the last line removes this route when the interface goes down). This route will act as the default route inside the virtual network (i.e. to all networks within `192.168.0.0/16`).

The set of lines for `eth2` and `eth3` are all currently commented out - they do nothing. They follow the same format as for `eth1`. I've included them for convenience when you create a virtual machine with two or three internal interfaces, e.g. router. You can just uncomment the lines and edit the addresses as needed.

Note that the all lines are not necessarily needed for all interfaces. For example, the `post-up` and `pre-down` lines may not be needed when configuring a router. Also the values I use may not be appropriate for your network. The lines I provide are just defaults - they should be changed appropriately depending on the virtual machine/network you are trying to configure. For a description of the `interfaces` file type `man interfaces` on the command line.

Configure Apache and MySQL Servers

Apache web server and MySQL database servers were installed when selecting the LAMP server package. By default both servers start upon boot. However in many cases they will not be needed, so we will configure the virtual machine so that both servers are off upon boot. Then you will need to manually start them if you need them.

To configure MySQL to be off upon boot:

```
network@base:~$ sudo sh -c 'echo "manual" > /etc/init/mysql.override'
```

To configure Apache to be off upon boot:

```
network@base:~$ sudo update-rc.d apache2 disable
```

Configure Other Software

There are several scripts and template files that I have developed to automate some tasks in the virtual network. To copy these files to each virtual machine and keep them up to date we will use Subversion (http://subversion.apache.org/) (SVN). To download the current files:

```
network@base:~$ svn checkout https://sandilands.info/svn/virtnet/
```

This downloads the latest version of the files and places them in the `virtnet` directory. From now on, to get the most recent versions (or any new files), just update this directory:

```
network@base:~$ svn update ~/virtnet
```

One file that is provided is a set of aliases and environment variables for the Bash shell used in the terminal. To use these aliases and environment variables:

```
network@base:~$ ln -s ~/virtnet/data/defaults/home/network
/DOTbash_aliases ~/.bash_aliases
```

This links the file `~/.bash_aliases` to the actual file `~/virtnet/data/defaults/home/network/DOTbash_aliases`. The aliases and environment variables will be available when you next login.

Password-less Logins

Once you have multiple virtual machines running in a network, you will often want to use SSH to connect to other machines. As a convenience, you can set each machine up so that you can SSH (login) without supplying a password (instead RSA public key cryptography is used). To set this up first generate a RSA key pair:

```
network@base:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/network/.ssh/id_rsa): ENTER
Enter passphrase (empty for no passphrase): ENTER
Enter same passphrase again: ENTER
Your identification has been saved in /home/network/.ssh/id_rsa.
Your public key has been saved in /home/network/.ssh/id_rsa.pub.
The key fingerprint is:
4d:5f:66:33:21:bf:85:a4:8b:70:b2:ea:5b:da:ed:32 network@node1
The key's randomart image is:
+--[ RSA 2048]----+
|            . o  |
|             = o |
|        o o . O .|
|         B o = = |
|        S o o .  |
|       .         |
|      . .        |
|     . +E.       |
|      +..+o      |
+-----------------+
```

Note that the default file location is ok (just press `ENTER`), and no passphrase is needed (press ENTER twice).

Now copy your public key (id_rsa.pub) into your authorized_keys file:

```
network@base:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

This allows you to login with SSH from the computer that has the corresponding private key (id_rsa) to the computer that has the public key in authorized_keys, without entering a password. That is, you login from *this* computer to *this same* computer without needed to enter your password. That doesn't seem much use (and its not yet). But once we clone this virtual machine to make other virtual machines it will allow you to login to those cloned virtual machines without entering a password. We'll see the cloning later.

Optionally, you may also copy your id_rsa.pub from the host (if you have one - otherwise generate a key pair as above) into the virtual machine authorized_keys files so you can login from host to virtual machine without entering a password.

Restart the Virtual Machine

Reboot the VM using the reboot command:

```
network@base:~$ sudo reboot
```

Prepare Interfaces for Cloning

The next step will be to clone this base virtual machine to create multiple other virtual machines. When we clone, VirtualBox can change the hardware (MAC) address of the network interfaces (so the new machine has a different hardware address to the original machine - this is desirable). However, Ubuntu is currently configured to give interface names (such as eth0 and eth1) based on hardware addresses. If we changed the hardware address, then Ubuntu will not be able to configure the interface upon booting, which is a problem. To avoid this we need to disable the current mapping from hardware address to interface name.

Edit the file /etc/udev/rules.d/70-persistent-net.rules and comment out the two lines that start with SUBSYSTEM by inserting the # character. (Alternatively, you can delete the lines).

```
network@base:~$ sudo nano /etc/udev/rules.d/70-persistent-net.rules
```

After adding the # character, it should like like the following (although the values, especially addresses may differ). Importantly, all lines are commented out.

```
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.
# PCI device 0x8086:/sys/devices/pci0000:00/0000:00:03.0 (e1000)
# SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="08:00:27:42:69:56", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x1af4:/sys/devices/pci0000:00/0000:00:08.0/virtio1
(virtio-pci)
# SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="08:00:27:0f:f3:79", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

Now shutdown the VM using the `poweroff` command:

```
network@base:~$ sudo poweroff
```

It's important that you comment out the lines in `/etc/udev/rules.d/70-persistent-net.rules` just before you poweroff and start the cloning with VirtualBox. If you start the virtual machine again *before* cloning, then Ubuntu will automatically add the lines to `/etc/udev/rules.d/70-persistent-net.rules` and you will have to remove them again. So now that the virtual machine is off, move to the next step of cloning it.

Clone the Base Virtual Machine

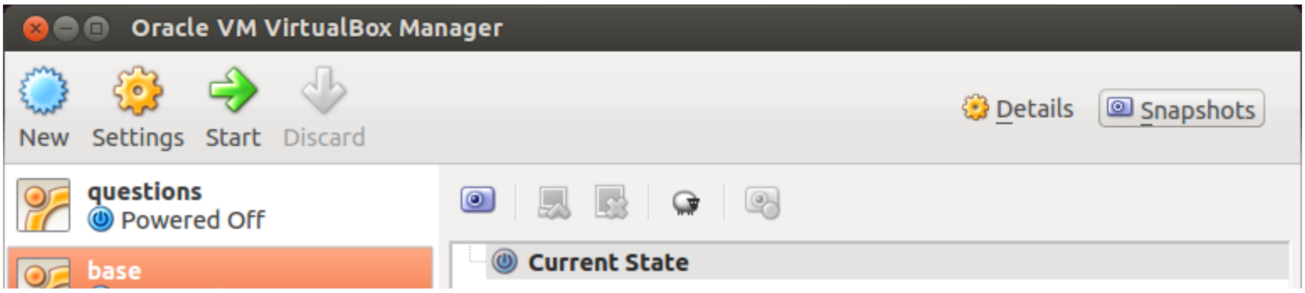
Our virtual network will consist of a set of virtual machines, or *nodes*. Each node will be created by cloning the base virtual machine. The previous step configured the base virtual machine so that once it is cloned and a node is created, it will be quicky and easy to configure that node correctly. This step describes how to clone in VirtualBox. You need to repeat this step as many times as the number of nodes you need in your virtual network. For example, if your network has three nodes, make three clones of the base virtual machine. The next step shows how to configure the node after it is cloned.

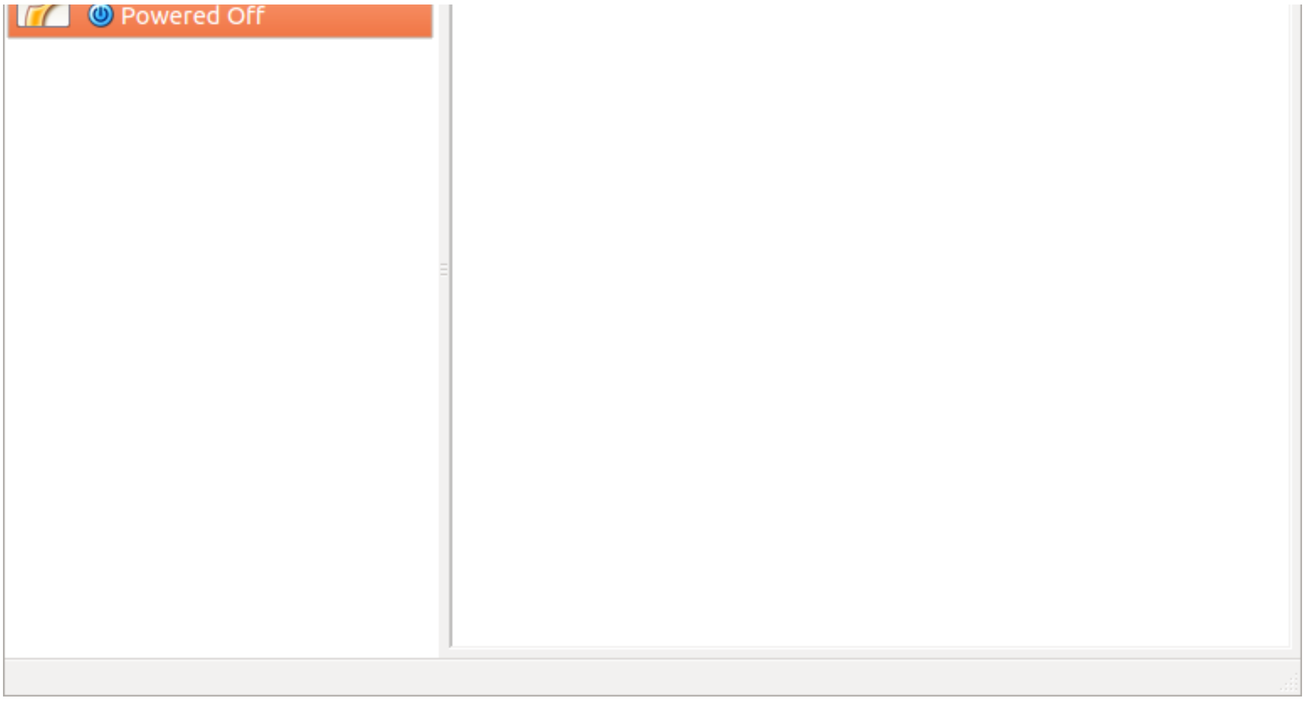
VirtualBox provides both a GUI and a command line interface (CLI) to manipulate virtual machines. If you are creating your first virtual network or just small networks, the GUI is a good choice. However once you understand the cloning process and you want to create large virtual networks, the CLI is very useful in automating the process. I provide instructions for both approaches below - choose whichever is appropriate for your situation.

Cloning with VirtualBox GUI

First create a snapshot of the base virtual machine. This snapshot, which is a version of the base virtual machine at specific time instance, will be cloned.

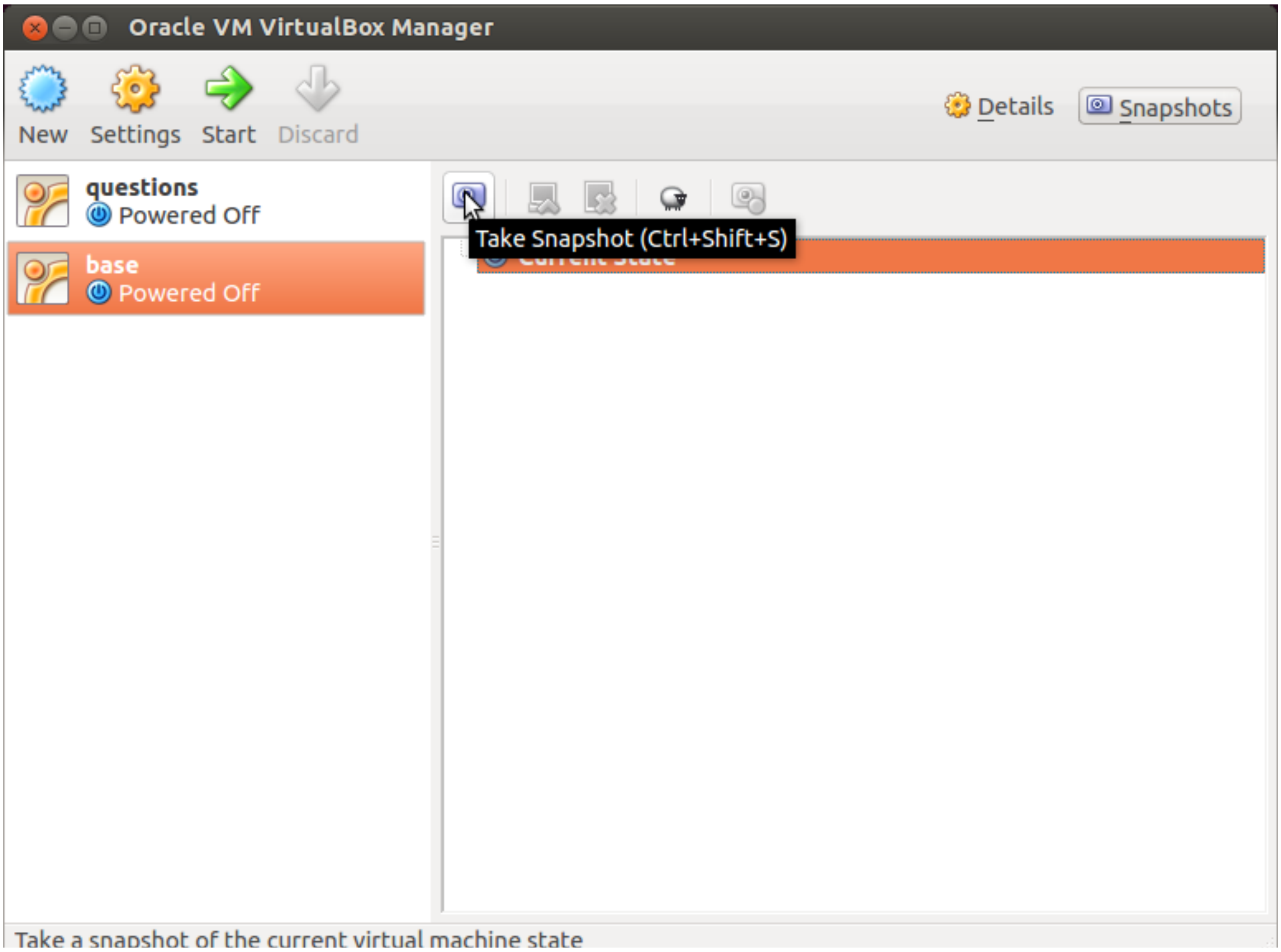
Select the base VM and then click the *Snapshots* button:





VirtualBox cloning of base VM: View snapshots of base VM

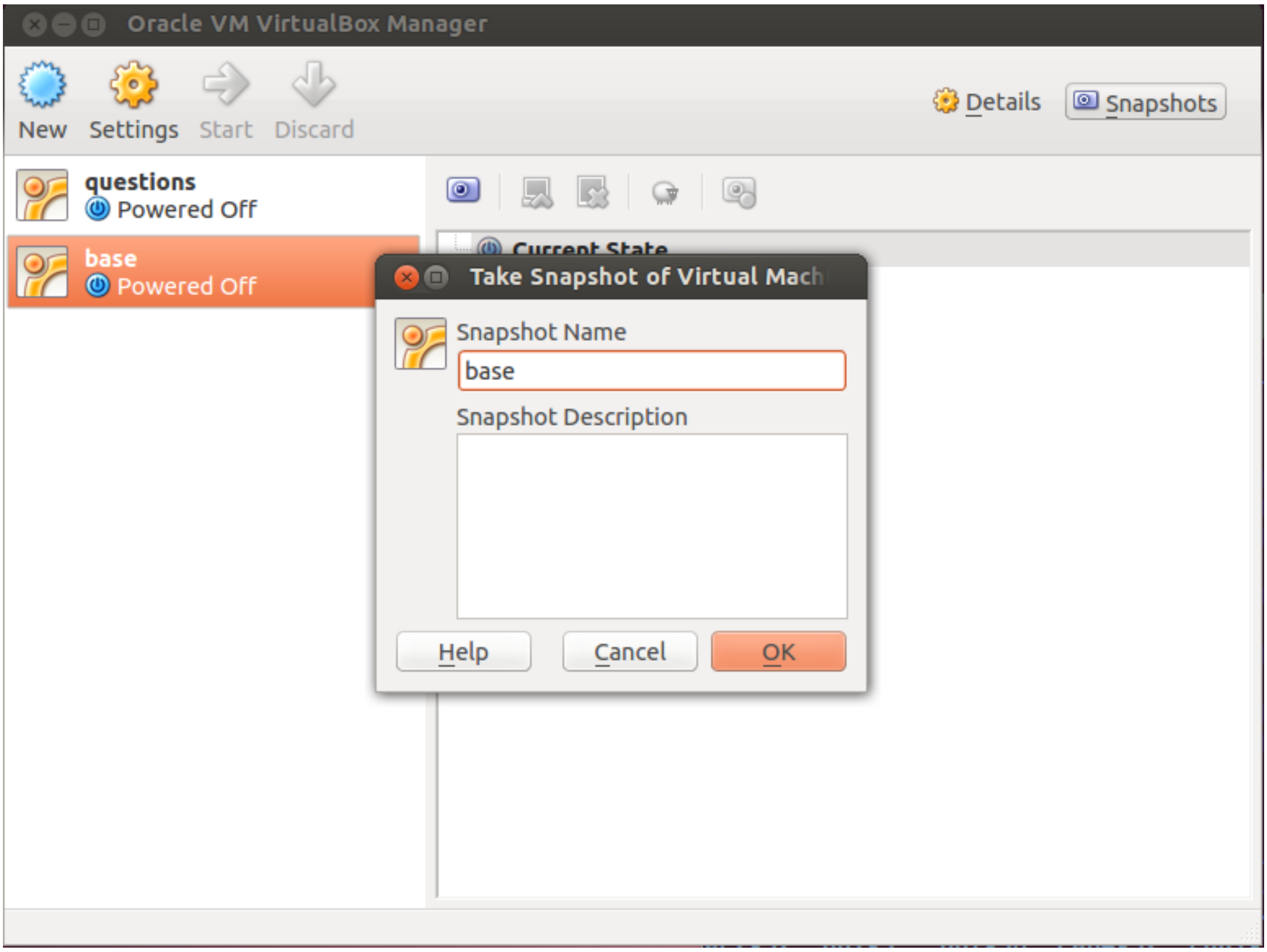
Click the *Take Snapshot* button:



Take a snapshot of the current virtual machine state

VirtualBox cloning of base VM: Take a snapshot of base VM

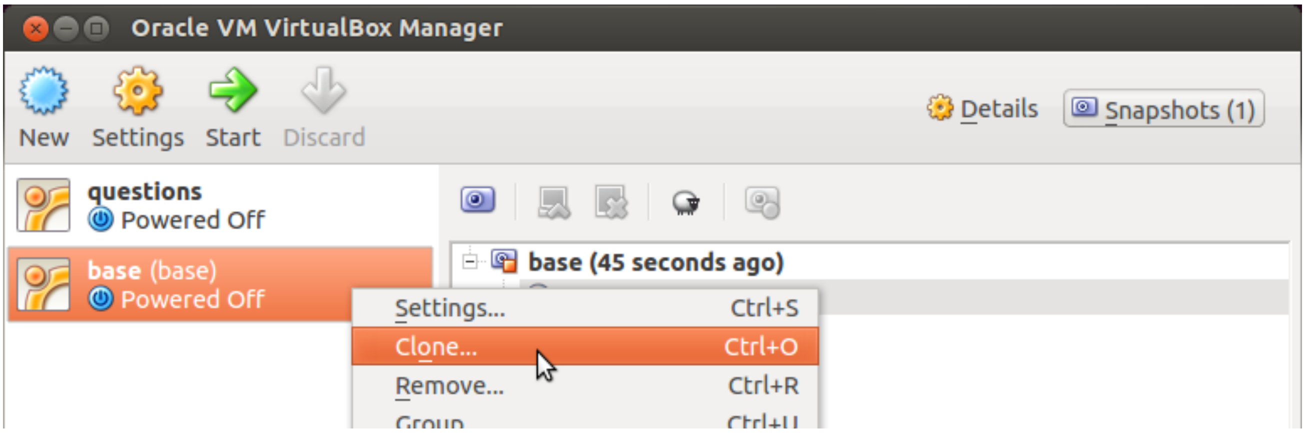
Name the snapshot *base* (i.e. same name as the VM):

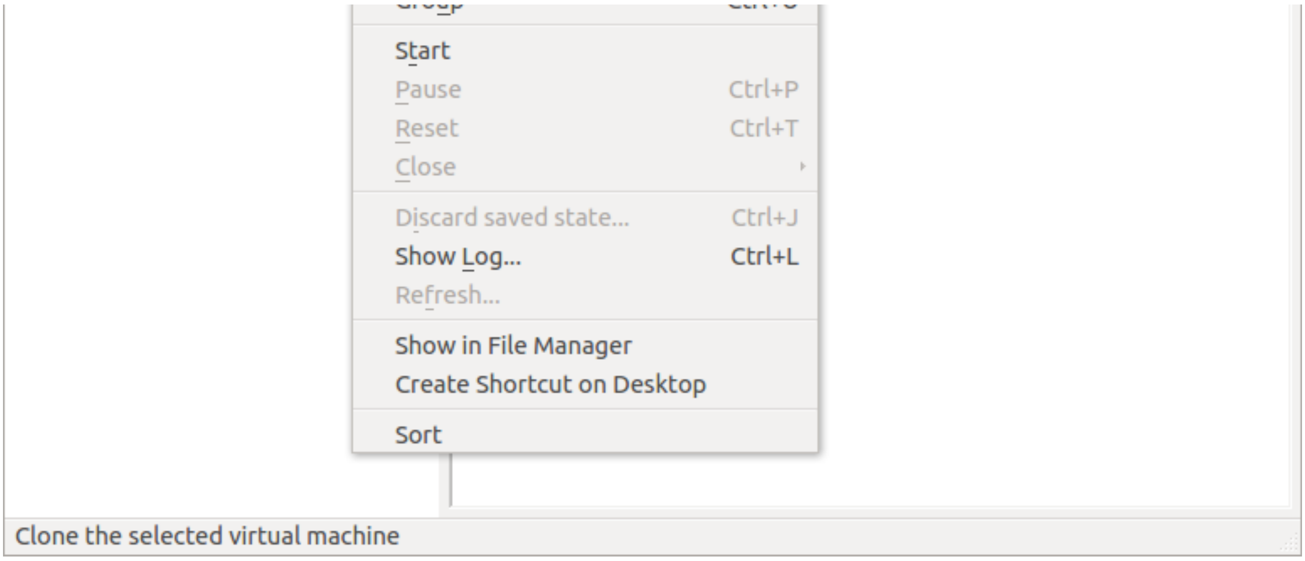


Snapshot Name: base

VirtualBox cloning of base VM: Name the snapshot base

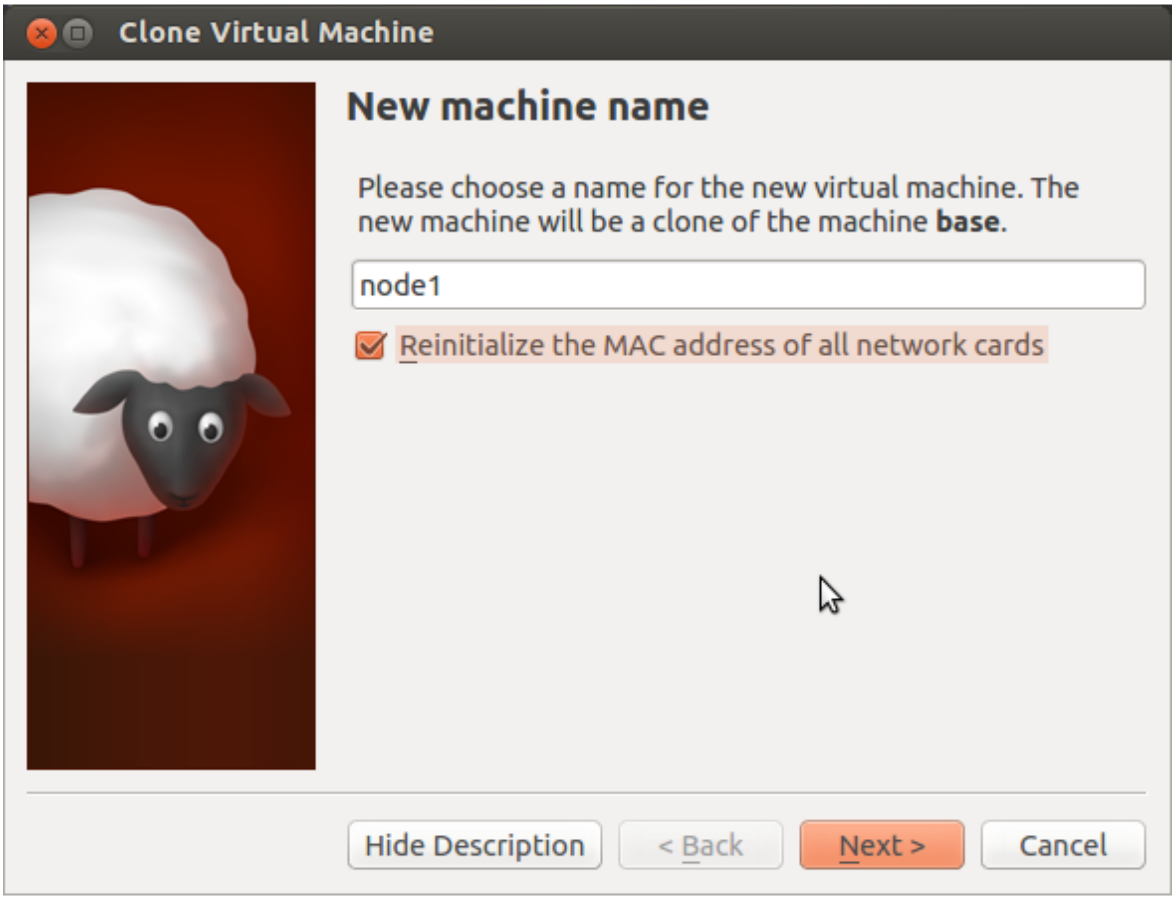
Now that the snapshot is created, right click on the base VM and select *Clone...*:




Start
Pause Ctrl+P
Reset Ctrl+T
Close
Discard saved state... Ctrl+J
Show Log... Ctrl+L
Refresh...
Show in File Manager
Create Shortcut on Desktop
Sort
Clone the selected virtual machine

VirtualBox cloning of base VM: Clone the base VM

Name the clone *node1* (and subsequent clones, *node2*, *node3* etc.) and check *Reinitialize the MAC address of all network cards*:



New machine name

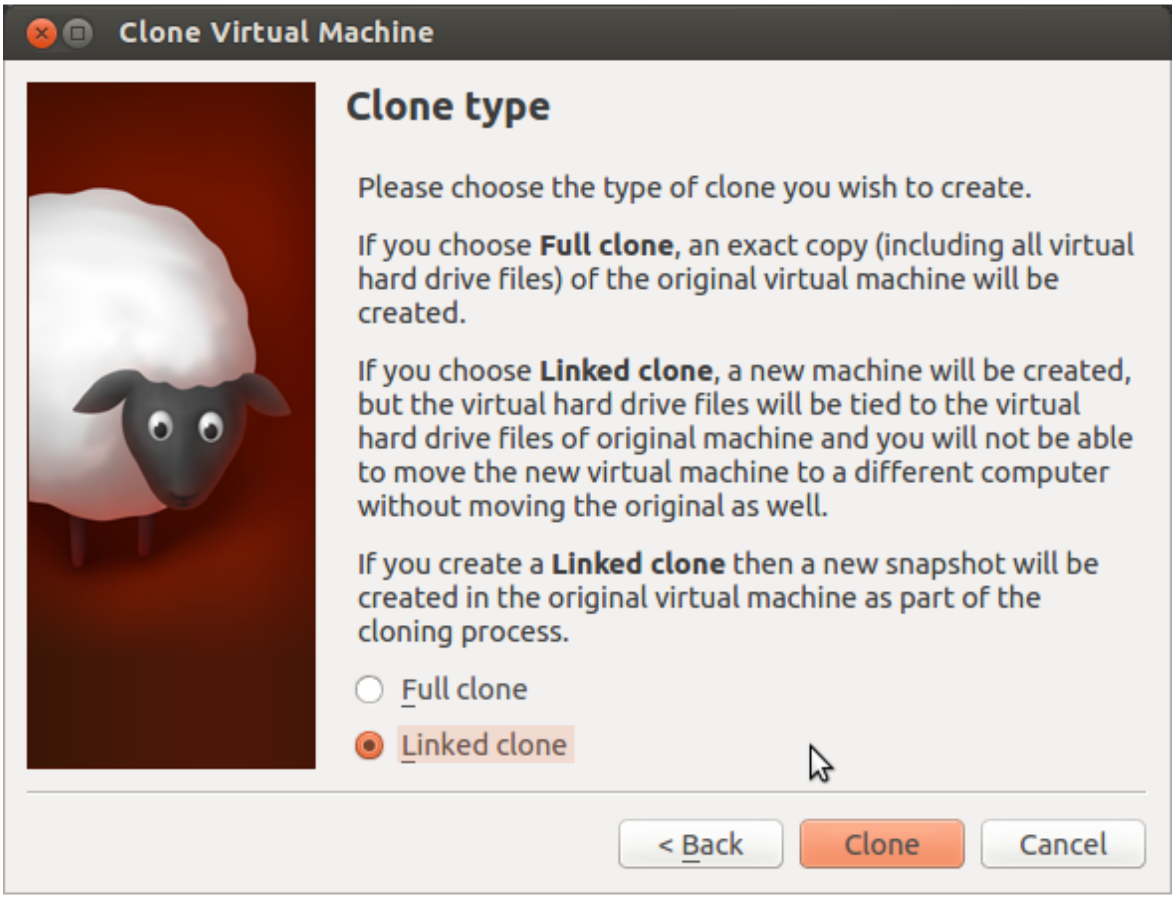
Please choose a name for the new virtual machine. The new machine will be a clone of the machine **base**.

node1

☑ Reinitialize the MAC address of all network cards

VirtualBox cloning of base VM: Name clone and reinitialize MAC addresses

Select *Linked clone* as the clone type. With a linked clone, only the *changes* between the cloned VM and the base VM hard disks are saved. With a full clone, the entire base VM hard disk is copied. Using a linked clone saves significant hard disk space on the host (at the expense of performance if the clone is used for a long time and many disk changes are made compared to the base).



Clone type

Please choose the type of clone you wish to create.

If you choose **Full clone**, an exact copy (including all virtual hard drive files) of the original virtual machine will be created.

If you choose **Linked clone**, a new machine will be created, but the virtual hard drive files will be tied to the virtual hard drive files of original machine and you will not be able to move the new virtual machine to a different computer without moving the original as well.

If you create a **Linked clone** then a new snapshot will be created in the original virtual machine as part of the cloning process.

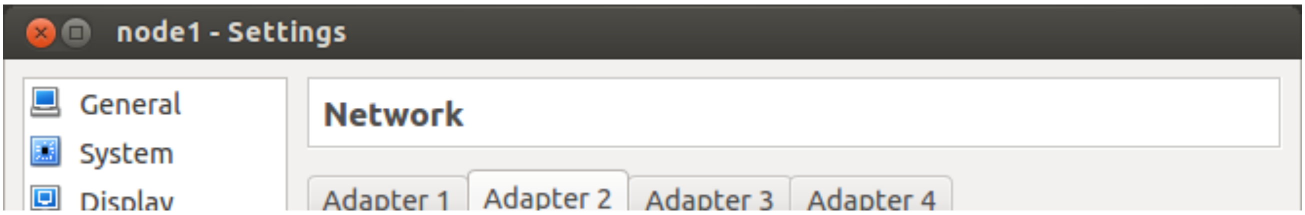
○ Full clone

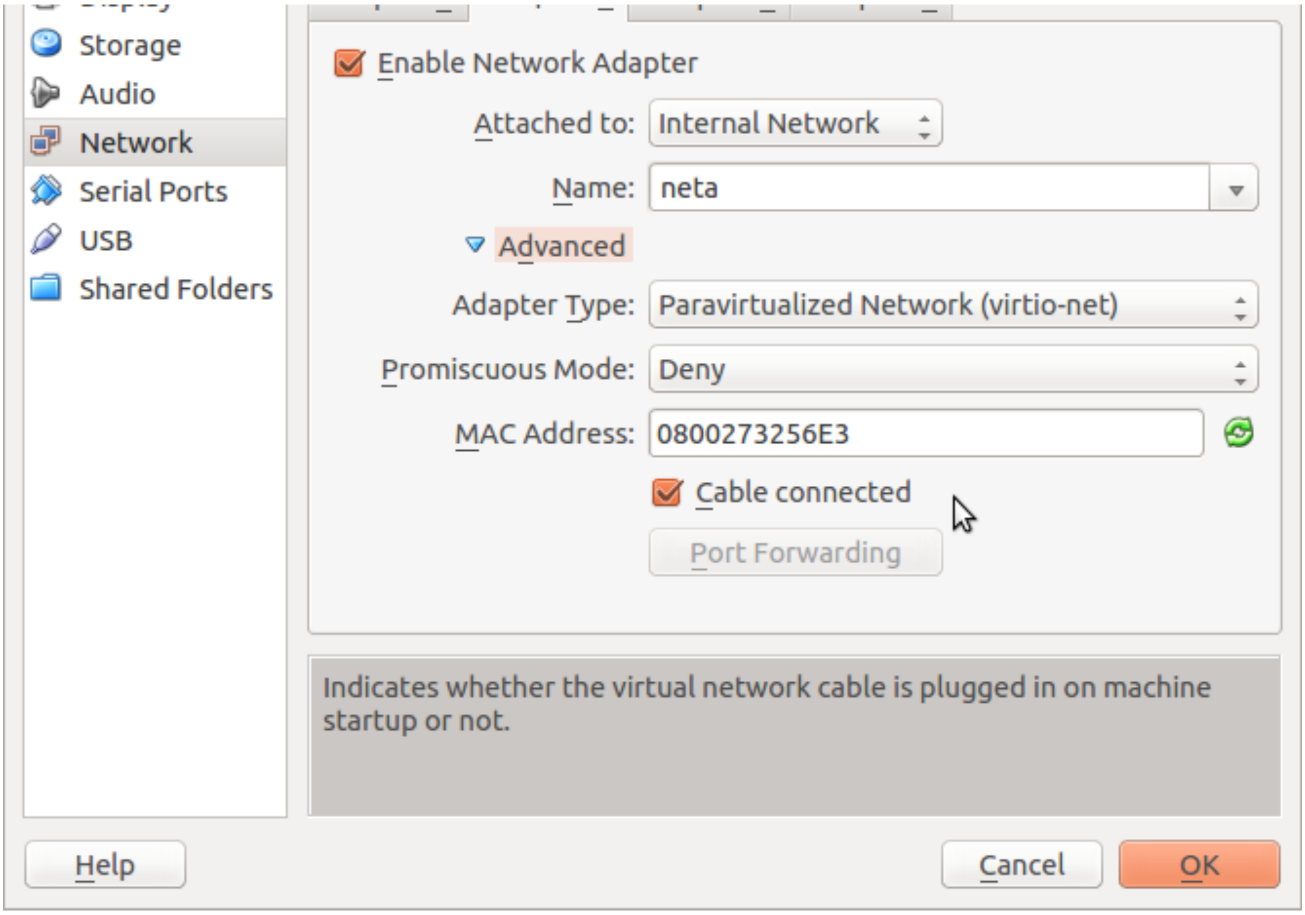
● Linked clone

VirtualBox cloning of base VM: Linked clone

Now the new virtual machine has been created, you may need to configure the network adapter settings. This depends on what role this node has in the virtual network. Recall that in our base virtual machine we configured four network adapters:

- Adapter 1: NAT, used for connection to host and internet.
- Adapter 2: internal, used for connecting to other nodes. Initially configured to use internal network *neta*.
- Adapter 3: internal, used for connecting to other nodes. Initially configured to use internal network *netb* but with the cable disconnected.
- Adapter 4: internal, used for connecting to other nodes. Initially configured to use internal network *netc* but with the cable disconnected.





☑ Enable Network Adapter

Attached to: Internal Network

Name: neta

Advanced

Adapter Type: Paravirtualized Network (virtio-net)

Promiscuous Mode: Deny

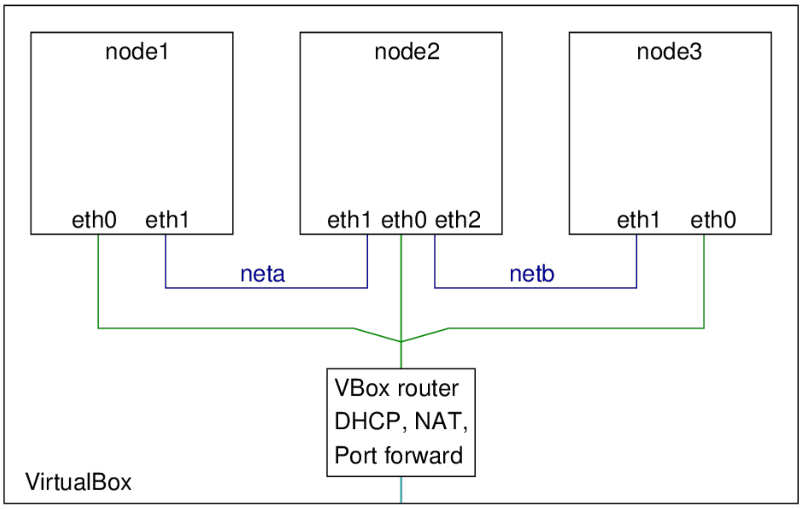
MAC Address: 0800273256E3

☑ Cable connected

Indicates whether the virtual network cable is plugged in on machine startup or not.

Network Adapter Settings of Cloned VM: set the network name and cable connected to appropriate values

You may need to change the settings for adpaters 2, 3 or 4, depending on how you intend this node to connect to other nodes in the virtual network. For example, if this node is `node1` in the network below, then no changes to the adapters are needed. However if this node is `node2`, then you need to connect the cable (by checking *Cable connected* for Adapter 3).



Topology for virtual network with NAT and two internal networks

In summary, for the new virtual machine, configure the settings for adapters 2, 3 and 4 by setting the appropriate internal network name (e.g. *neta*, *netb*, *netc*) and connecting/disconnecting the cable depending on whether the adapter is needed or not.

The final change is to set the port to be used for port forwarding on adapter 1 (that NAT interface). For the base virtual machine the port 2201 was set to forward to the SSH port 22. You should change the port so that it is unique for each node. For example, on `node1` use port 2201, on `node2` use port 2202, on `node3` use port 2203 and so on. (You can use other ports; they don't have to be 2201, 2202, etc.). If I was configuring `node5` then I would set the port to 2205.

Now you can start the new virtual machine and proceed to the next step to configure it.

Cloning with VirtualBox Command Line Interface

Anything you can do in the VirtualBox GUI can also be performed using the VirtualBox command line interface using the program `VBoxManage`. This is very useful if you want to automate steps, especially when creating large networks: put the `VBoxManage` commands in a script and then just run the script. The VirtualBox manual (https://www.virtualbox.org/manual/) gives the details of using `VBoxManage` (https://www.virtualbox.org/manual/ch08.html). Here I just present the commands that can be used to achieve the cloning performed with the GUI above.

To start, I assume that the base virtual machine exists *and* has a snapshot also called base. See the GUI steps to create a snapshot (in fact you can also do it on the command line).

List the current virtual machines:

```
$ VBoxManage list vms
"questions" {5d8bf8b2-40e9-47fa-b35c-dd1e54aa627c}
"base" {6fc0828a-1d62-4b9a-ba7f-6e334bc66b3d}
```

List the current running virtual machines:

```
$ VBoxManage list runningvms
"questions" {5d8bf8b2-40e9-47fa-b35c-dd1e54aa627c}
```

Now that we know the base virtual machine exists and is not currently running, we can clone it. Clone the base virtual machine, creating a new virtual machine called *node1*, using a *linked* clone and *register* with the VirtualBox manager:

```
$ VBoxManage clonevm base --snapshot base --name node1 --options link
--register
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Machine has been successfully cloned as "node1"
```

To check, list the VMs again:

```
$ VBoxManage list vms
```

```
"questions" {5d8bf8b2-40e9-47fa-b35c-dd1e54aa627c}
"base" {6fc0828a-1d62-4b9a-ba7f-6e334bc66b3d}
"node1" {cb4bd138-5cbe-4995-9250-a9f930bc8625}
```

Now change the adapter settings. To do this, you need to know the adapter that you want to change (e.g. adapter 2, 3 or 4). You can connect/disconnect the cable and name the internal network. For example, to set the network name on adapater 2 to be *netc*:

```
$ VBoxManage modifyvm node1 --nic2 intnet --intnet2 netc
```

Or to set the cable on adapter 3 to be disconnected:

```
$ VBoxManage modifyvm node1 --cableconnected3 off
```

To view information about a virtual machine (below I only show the output related to the network adapters):

```
$ VBoxManage showvminfo node1
...
NIC 1:           MAC: 080027929632, Attachment: NAT, Cable connected:
on, Trace: off (file: none), Type: virtio, Reported speed: 0 Mbps, Boot
priority: 0, Promisc Policy: deny, Bandwidth group: none
NIC 1 Settings:  MTU: 0, Socket (send: 64, receive: 64), TCP Window
(send:64, receive: 64)
NIC 1 Rule(0):   name = ssh, protocol = tcp, host ip = , host port =
2201, guest ip = , guest port = 22
NIC 2:           MAC: 080027FD87BE, Attachment: Internal Network
'netc', Cable connected: on, Trace: off (file: none), Type: virtio,
Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: deny,
Bandwidth group: none
NIC 3:           MAC: 080027876368, Attachment: Internal Network
'netb', Cable connected: off, Trace: off (file: none), Type: virtio,
Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: deny,
Bandwidth group: none
NIC 4:           MAC: 08002785F443, Attachment: Internal Network
'netc', Cable connected: off, Trace: off (file: none), Type: virtio,
Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: deny,
Bandwidth group: none
...
```

To set the port number for port forwarding you should delete the existing rule (if necessary) and add a new rule. For example, if I want to use port 2205 for a virtual machine and the old rule called *ssh* used a different port:

```
$ VBoxManage modifyvm node1 --natpf1 delete ssh
$ VBoxManage modifyvm node1 --natpf1 ssh,tcp,,2205,,22
```

To start a virtual machine without a GUI interface, i.e. in headless mode:

```
$ VBoxManage startvm --type headless node1
```

Configure Each Node

When you have created a node by cloning the base virtual machine, start and login to that node. There are several steps needed to finalize the node configuration. Also, depending on the role of the node in the virtual network, you may need to make further configuration changes.

All Nodes

First change the host name in both the `/etc/hostname` and `/etc/hosts` files. You can edit them manually with the text editor nano (also using sudo) or change them directly with the following lines (replacing *node1* with your node name):

```
network@base:~$ sudo sed -i 's/base/node1/' /etc/hostname
network@base:~$ sudo sed -i 's/base/node1/' /etc/hosts
```

Next set the appropriate addresses for your node interfaces. Open `/etc/network/interfaces` with nano and change the values to suit your node. Once you have saved the file, restart the interface(s) that you changed by either rebooting or:

```
network@base:~$ sudo ifdown eth1
network@base:~$ sudo ifup eth1
```

Router Nodes

If your node is a router in the virtual network then you need to set forwarding to be on. You need to set the `ip_forward` parameter in the kernel to be `1`. There are several ways, one being:

```
network@base:~# sudo sysctl net.ipv4.ip_forward=1
```

Your node is now a router. However this setting will not be in effect after a reboot. To set it permanently you can edit the file `/etc/sysctl.conf` and set `net.ipv4.ip_forward` to `1`. Alternatively run:

```
network@base: ~# sudo sed -i 's/#net.ipv4.ip_forward=1
/net.ipv4.ip_forward=1/' /etc/sysctl.conf
```

Web Server Nodes

Although Apache web server is installed in all nodes, we configured the base to disable it upon boot. If you want a node to run a web server (as well as MySQL database) then you can perform the following:

```
network@base:~$ sudo service mysql start
```

```
mysql start/running, process 1036
network@base:~$ sudo apache2ctl start
```

If you see a warning from Apache about not being able to reliably determine the server's domain name, then don't worry. Later you can edit the ServerName variable in the Apache configuration to avoid this warning.

Note that Apache and MySQL will be off again when you reboot. To automatically start the server upon booting, effectively reverse the steps in disabling the servers on boot.

To configure MySQL to be on upon boot delete the override file:

```
network@base:~$ sudo rm /etc/init/mysql.override
```

To configure Apache to be on upon boot:

```
network@base:~$ sudo update-rc.d apache2 enable
```

Reboot and Test

Now reboot your node, login and test. Then repeat the steps of cloning and configuring for other nodes to create your virtual network.

Using Your Virtual Network

Once you have created several nodes and configured their network interfaces you can start using your virtual network. To finish I point to some further resources that may assist when using your network.

You can use the VirtualBox graphical interface to run and access the virtual guests. However often you may want to access the guests using applications on your host computer, e.g. web browser, file explorer, SSH client. I have described several ways to do this (/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling), that will work for a Linux, OSX and Windows host operating systems.

Content: Howto (/sgordon/taxonomy/term/212)
Interest: Linux (/sgordon/taxonomy/term/300)
Ubuntu Linux (/sgordon/taxonomy/term/302)
VirtualBox (/sgordon/taxonomy/term/328)

Comments

Sanchit Garg replied on Wed, 13/08/2014 - 1:21am PERMALINK (/SGORDON/COMMENT/3950#COMMENT-3950)

THANKS A LOT :). (/SGORDON/COMMENT/3950#COMMENT-3950)

Thanks a lot :).
Really helpful

Miguel replied on Wed, 20/08/2014 - 2:26pm PERMALINK (/SGORDON/COMMENT/3954#COMMENT-3954)

POSSIBLE ERROR IN THE TEXT (/SGORDON/COMMENT/3954#COMMENT-3954)

Hi,

I would assume that in the configuration of eth1, it should be: address 192.168.1.1 instead of 192.168.1.1. Is this the case?

admin replied on Wed, 20/08/2014 - 2:59pm PERMALINK (/SGORDON/COMMENT/3955#COMMENT-3955)

RE: POSSIBLE ERROR IN THE TEXT (/SGORDON/COMMENT/3955#COMMENT-3955)

I assume you mean where the IP of eth1 is set to 192.168.1.11 (.. dot eleven). That is deliberate (dot eleven, not dot one). In my setup I assigned addresses like 192.168.1.11, 192.168.1.12, 192.168.1.13 for those on network 192.168.1.0/24, and for those on network 192.168.2.0/24, I use 192.168.2.21, 192.168.2.22, etc. There are probably better ways and you don't have to use dot eleven, but as far as I can remember, 192.168.1.11 is correct.

BTW if you want to setup a virtual network in a slightly easier way try my VirtualBox image via: https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage (https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage)

Vinay replied on Fri, 05/09/2014 - 6:49am PERMALINK (/SGORDON/COMMENT/3971#COMMENT-3971)

THANKS (/SGORDON/COMMENT/3971#COMMENT-3971)

Thank you for these detailed & clear instructions.

Do you have any instructions for how to setup routing between guest nodes with Centos 6?

Mortner replied on Wed, 24/09/2014 - 3:50am PERMALINK (/SGORDON/COMMENT/3994#COMMENT-3994)

QUESTIONS (/SGORDON/COMMENT/3994#COMMENT-3994)

Many thanks, this saved me a lot of time.

I have 3 questions - but it may be related to the fact that I am running debians VM - and most probably to the fact that I have very little knowledge of how network works.

1) I cannot use eth0 and eth1 at the same time (ie I have to ifdow eth1 to connect to the internet. Is that normal ?)

2) I had to setup /etc/hosts manually with IP to hostname connection in order to be able to

ssh between nodes. Otherwise hostnames won't get resolved

3) I don't understand why the SSH port foward on eth0, since ssh is used between hosts, hence over eth1 ....

admin replied on Wed, 24/09/2014 - 7:53am PERMALINK (/SGORDON/COMMENT/3995#COMMENT-3995)

RE: QUESTIONS (/SGORDON/COMMENT/3995#COMMENT-3995)

Firstly, you may consider a slightly faster way to setup the virtual network (it is essentially some scripts to automate the above approach, including setting up interfaces and cloning VMs): https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage (https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage)

1) That is a routing issue. eth0 is used for Internet connection, eth1 and above are used for the virtual network. By default, the virtual network nodes are on the 192.168.0.0/16 subnet. While the virtualbox interface to the Internet is 10.0.2.2. So to be able to use both (without turning one interface down) then a special route to the virtual network is needed. I thought the /etc/network/interfaces file in the instructions above (post-up and pre-down) would do this correctly. If it is not working, then here is the routing table on one of my working nodes:

```
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 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0
0        0 eth0
192.168.0.0     192.168.1.11    255.255.0.0     UG    0
0        0 eth1
192.168.1.0     0.0.0.0         255.255.255.0   U     0
0        0 eth1
```

You should try setting the routes to be the same as this. Again, I suggest you try my newer, more automated steps (https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage) if you want to avoid some of this work.

2) Yes, you need to manually set /etc/hosts on all nodes. There is no internal DNS, so hostnames must be manually managed. Otherwise just use IP addresses.

3) Correct, ssh is used between hosts over eth1 (and above). The reason I setup the port forward on eth0 (which is the connection from the real host computer to the virtual guest) is so that I can SSH from my host computer to the virtual guest. That means I don't need to use the VirtualBox terminal interface, I just use my host computer to SSH into each of the nodes. It is described how at https://sandilands.info/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling (https://sandilands.info/sgordon

/accessing-virtualbox-guests-using-ssh-winscp-tunnelling).

Mortner replied on Thu, 25/09/2014 - 1:22am PERMALINK (/SGORDON/COMMENT/3998#COMMENT-3998)

MANY THANKS FOR THESE (/SGORDON/COMMENT/3998#COMMENT-3998)

Many thanks for these thorough answers. I will try your automated methods... and keep your pages in my bookmark folder !!

Anonymous replied on Sat, 04/10/2014 - 8:52pm PERMALINK (/SGORDON/COMMENT/4002#COMMENT-4002)

MISSING PICTURES ON HTTPS://SANDILANDS.INFO/VIRTNET/TOPOLOGIES (/SGORDON/COMMENT/4002#COMMENT-4002)

From: Dieter Wateka

This is a lot of valuable instructions.

There is one problem (perhaps only for me). I cannot see the pictures of the topologies. Are the given paths still correct?

Thank you.

admin replied on Sat, 04/10/2014 - 8:54pm PERMALINK (/SGORDON/COMMENT/4003#COMMENT-4003)

RE: MISSING PICTURES ON HTTPS://SANDILANDS.INFO/VIRTNET/TOPOLOGI (/SGORDON/COMMENT/4003#COMMENT-4003)

You are correct - the pictures were missing due to a change I made on the website last week. I've fixed it. You can see them now:
http://sandilands.info/virtnet/topologies

admin replied on Mon, 13/10/2014 - 8:55pm PERMALINK (/SGORDON/COMMENT/4004#COMMENT-4004)

THIS IS EXACTLY I WAS LOOKING (/SGORDON/COMMENT/4004#COMMENT-4004)

From: Sanju

This is exactly I was looking for. I have two queries - if you can help me with

1. How can I setup a default routing - instead of routing to specific network
2. In my case, I have internet access only through node1. How do I setup routing so that node2 is able to use internet from node1

admin replied on Mon, 13/10/2014 - 8:57pm PERMALINK (/SGORDON/COMMENT

/4005#COMMENT-4005)

RE: THIS IS EXACTLY I WAS LOOKING (/SGORDON/COMMENT/4005#COMMENT-4005)

First, if you haven't already, I suggest you look at my "improved" (more automated) way for creating the virtual network (/sgordon/automatic-creation-of-virtual-network-with-vboxmanage).

In the instructions above (when setting up the interfaces, see also my reply (/sgordon /comment/3995#comment-3995) to a related question) you setup a route to a specific *range* of networks, i.e. 192.168.0.0/16. This is because I assume all the virtual networks are in this range, e.g. 192.168.1.0/24, 192.168.2.0/24. So host 192.168.1.11 has a default gateway (router) of 192.168.1.1 to reach any network within 192.168.0.0/16. That routing entry is similar to the "default route". The reason is to differentiate from accessing the real Internet via the VirtualBox router 10.0.2.2. Here is my example routing table:

```
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 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0
0        0 eth0
192.168.0.0     192.168.1.1    255.255.0.0     UG    0
0        0 eth1
192.168.1.0     0.0.0.0         255.255.255.0   U     0
0        0 eth1
```

The first two routes are special cases for Internet access. The last two routes are for internal routing in the virtual network. Think of the 3rd route as the "default route" for your internal network. For more complex topologies you can add further routes using the `route` command or in the `interfaces` file with `post-up` and `pre-down`. Again, as a simpler solution, I suggest you look at my newer automated approach (/sgordon /automatic-creation-of-virtual-network-with-vboxmanage) which has many in-built topologies (/virtnet/topologies). If you want a different topology, then let me know and I will create it.

Regarding Internet access from node2, can you not configure it to access the Internet via VirtualBox? I have normal Internet access via VirtualBox on all nodes I create. Check the routes on node 2, it should be similar to the routing table above for the first 2 lines at least. If you want to get node2 to access the Internet via node1, then that is a bit harder. I think you will need to run masquarading (NAT) using iptables on node1, but I haven't tried it.

router (http://mepha31.blogspot.com/) replied on Fri, 16/01/2015 - 8:10pm PERMALINK (/SGORDON/COMMENT/4047#COMMENT-4047)

THANKS (/SGORDON/COMMENT/4047#COMMENT-4047)

thank you for valuable information

Vince replied on Mon, 20/04/2015 - 1:33pm PERMALINK (/SGORDON/COMMENT/4093#COMMENT-4093)

I THINK THE THIRD LINE OF THE (/SGORDON/COMMENT/4093#COMMENT-4093)

I think the third line of the gateway should be 192.168.1.1
I couldn't get the results above if I follow your instruction.
And I am also have the issue with internet connection. I cannot let both eth0 and eth1 up(I have to ifdown eth1 to get internet connection).
Any idea why?

admin replied on Mon, 20/04/2015 - 3:05pm PERMALINK (/SGORDON/COMMENT/4094#COMMENT-4094)

RE: I THINK THE THIRD LINE OF THE (/SGORDON/COMMENT/4094#COMMENT-4094)

You are correct, it should be 192.168.1.1. I have just fixed it in my comment above to reflect this.

Regarding the internet access, is it a DNS problem? The VirtualBox router is 10.0.2.2 and the DNS server is 10.0.2.3. You should at least be able to ping both of them. Then check /etc/resolv.conf. Mine is:

```
$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3)
generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE
OVERWRITTEN
nameserver 10.0.2.3
search siit.tu.ac.th
```

where siit.tu.ac.th is for my current network (you could delete that line). I recall some problems with internet access recently, but not the cause/solution. Maybe check there is nothing strange in the DNS server on node1 and on your host. I've seen some comments/problems about VirtualBox and DNS and references to the Advanced NAT DNS settings https://www.virtualbox.org/manual/ch09.html#nat-adv-dns but haven't tried them (Internet access is currently working for me).

admin replied on Tue, 28/10/2014 - 8:58pm PERMALINK (/SGORDON/COMMENT/4006#COMMENT-4006)

GREAT INFO (/SGORDON/COMMENT/4006#COMMENT-4006)

From: Joe Rivera

Thanks for the info and detailed instructions. Is there any reason the base VM can't be configure with more memory and disk space assuming the host system can support it?

admin replied on Wed, 29/10/2014 - 8:59pm PERMALINK (/SGORDON/COMMENT/4007#COMMENT-4007)

RE: GREAT INFO (/SGORDON/COMMENT/4007#COMMENT-4007)

No reason. In some cases I have increased the RAM (e.g. to 512MB) in the VM and it works fine. Same should apply for the disk space, however I don't recall if you need to do anything regarding partitions in the guest after changing the disk space (I think not?). I suggest make sure you have a copy of the base.tar.gz file (or base directory) and then just try it. Worst case, just delete and revert back to the original.

Note that there are some easier, more automated instructions (http://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage) that you may want to try instead of the above. Just download the base image and get started. I have updated in the last week. Currently at release 41.

Pushpita Chaterjee replied on Thu, 18/12/2014 - 5:29pm PERMALINK (/SGORDON/COMMENT/4034#COMMENT-4034)

VM WITH OVS CONNECTED TO TWO OTHER GUEST VMS (/SGORDON/COMMENT/4034#COMMENT-4034)

Nice and Informative post. I need to know how to set up the virtual box vms to create a small topology. One guest VM (S) is running OVS, having two internal network interfaces. Each interface is connected to another VMs. All VMs are ubuntu 14.04. The configuration is : S-eth1-->H1-eth0 and S-eth2-->H2-eth0. H1 and H2 should be able to ping.

admin replied on Mon, 05/01/2015 - 4:39pm PERMALINK (/SGORDON/COMMENT/4040#COMMENT-4040)

RE: VM WITH OVS CONNECTED TO TWO OTHER GUEST VMS (/SGORDON/COMMENT/4040#COMMENT-4040)

The above instructions create the topology you want: H1 - S - H2, where S is a router. However the interface names are different (I use eth1 and eth2, as eth0 is used for connecting the VM to the host computer).

You may want to look at the newer instructions (https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage), which include an easy way to choose and create one of many common topologies.

Okta Satria replied on Mon, 05/01/2015 - 9:40am PERMALINK (/SGORDON/COMMENT/4037#COMMENT-4037)

(FIRST, SORRY FOR MY ENGLISH) (/SGORDON/COMMENT/4037#COMMENT-4037)

(First, sorry for my english)
Thanks very much for your article.

I am new in learning network security. I'he followed your instructions and it works. But, i failed when i try to build router. I use kali linux as a host n ubuntu as guest. node1 as router and node2 as server to be hacked. I want, to access node2 (server) from kali via node1 (router). I 'he try severals configuration but with several network adapter from VBox but always failed.

Could you tell me what i have to do? So, my kali can access node2 (server) via node1 (router). I use node1 (router) to use snort and my own network capturing.

Thank in advance!

Okta Satria replied on Mon, 05/01/2015 - 11:09am PERMALINK (/SGORDON/COMMENT/4038#COMMENT-4038)

THANKS VERY MUCH. I USE YOUR (/SGORDON/COMMENT/4038#COMMENT-4038)

Thanks very much. I use your article to setup my own lab. Now, i learn network security and network monitoring. I use Kali as host and ubuntu as guest. I have 2 Nodes: Node1 as router (routing host and server) and Node2 as server. I use router (Node1) to monitor network traffic between host (Kali) and server (Node2). I try severals way with severals network adapter from VBox but it failed. Could you tell me what i have to do?
Thanks in advance !

admin replied on Mon, 05/01/2015 - 4:31pm PERMALINK (/SGORDON/COMMENT/4039#COMMENT-4039)

RE: THANKS VERY MUCH. I USE YOUR (/SGORDON/COMMENT/4039#COMMENT-4039)

As I understand your network topology is:

```
Host --- Node1 --- Node2
```

but Host is in fact your real computer running Kali - it is not a virtual node in VirtualBox. I don't think if this will work (if it does, it may get confusing/inconvenient). Although you can login from Host direct to Node1 or from Host direct to Node2, you cannot easily connect from Host to Node2 via Node1.

I suggest you create topology 5 (http://sandilands.info/virtnet/topologies) and use node1 as your client, node2 as the router and node3 as the server. Although node1 will be running Ubuntu (not Kali), you can install additional security software on it if you need. (If you want GUI applications on node1 then you need to expand the disk size and install a desktop environment.)

Also if you take this approach, I suggest try the newer instructions (https://sandilands.info/sgordon/automatic-creation-of-virtual-network-with-vboxmanage) which hopefully make creating a virtual topology easier.

Okta Satria replied on Tue, 06/01/2015 - 9:07am PERMALINK (/SGORDON/COMMENT/4043#COMMENT-4043)

I WILL TRY IT BUT I HAVE 1 (/SGORDON/COMMENT/4043#COMMENT-4043)

I will try it but i have 1 problem, i just have 4GB memory in host.
Ok, thank you for your reply.

ta2tyrants replied on Mon, 09/03/2015 - 8:01am PERMALINK (/SGORDON/COMMENT/4067#COMMENT-4067)

I HAVE A QUESTION IN WHICH (/SGORDON/COMMENT/4067#COMMENT-4067)

I have a question in which you have the user do the following: "network@base:~$ sudo nano /etc/udev/rules.d/70-persistent-net.rules". I am using ubuntu server 14.04 and I do not have the text file in the rules.d directory. I have to actually create the file. All i have in the rules.d directory is a "README" file. And it's just a file that explains how the rules.d directory works when you add files to it.

Where do I find the bolded information:
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:/sys/devices/pci0000:00/0000:00:03.0 (e1000)
# SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:42:69:56", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x1af4:/sys/devices/pci0000:00/0000:00:08.0/virtio1 (virtio-pci)
# SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:0f:f3:79", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"

Looking for some guidance. By the way I find your site to be very educational. Cheers mate!

admin replied on Mon, 09/03/2015 - 8:10am PERMALINK (/SGORDON/COMMENT/4068#COMMENT-4068)

RE: I HAVE A QUESTION IN WHICH (/SGORDON/COMMENT/4068#COMMENT-4068)

If the file "70-persistent-net.rules" does not already exist, then I suggest try to skip this step. That is, don't create the file.

What this file does is assigns the interface "eth0" to the LAN card with MAC address 08:00:27:42:69:56. But when you clone the virtual machine the MAC address changes, so the new MAC address will be assigned eth2 (while eth0 will be unused). This is a problem, because my instructions/skips refer explicitly to eth0. To overcome this problem, its important to make sure the lines that refer to the specific MAC address are not in the file before the clone. Hence I gave the instructions of commenting them out.

The easiest solution however is to delete the file. It is automatically generated (if needed) anyway, so there is no harm in deleting it.

In summary, before cloning: if you don't have the file "70-persistent-net.rules" then do not create it; if you do have the file then delete it.

ta2tyrants replied on Tue, 10/03/2015 - 2:19am PERMALINK (/SGORDON/COMMENT/4069#COMMENT-4069)

THANK YOU VERY MUCH! I WASN (/SGORDON/COMMENT/4069#COMMENT-4069)

thank you very much! I wasn't expecting a reply so quickly!

cbellows replied on Mon, 27/04/2015 - 9:15pm PERMALINK (/SGORDON/COMMENT/4099#COMMENT-4099)

CONFIGURE OTHER SOFTWARE (/SGORDON/COMMENT/4099#COMMENT-4099)

Thanks Steve for the nice and easy to follow tutorial. Of course, there is a but coming...

I think I missed the point here. I am not quite sure what I should be configuring other software for.

Do I need it and what does it do exactly? If I do not download it, will my installation still work?

Thanks
Charles

Ulan replied on Tue, 06/10/2015 - 7:40pm PERMALINK (/SGORDON/COMMENT/4279#COMMENT-4279)

THANKS A LOT PROFESSOR. VERY (/SGORDON/COMMENT/4279#COMMENT-4279)

Thanks a lot Professor. Very helpful!

Ulan replied on Tue, 06/10/2015 - 7:47pm PERMALINK (/SGORDON/COMMENT/4280#COMMENT-4280)

IT IS NOT ENOUGH TO SAY JUST (/SGORDON/COMMENT/4280#COMMENT-4280)

It is not enough to say just 'thank you'. Really great explanation with every detail! Not only this article but all pages in this website are useful. I would like to participate your classes :).