Testing lab environment...

DBTech's Virtual Workshop on Distributed, Replicated, Embedded and Mobile Databases (Deliverable #28).

Antonio César Gómez Lora

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Introduction

Hi administrators! Here you can find instructions to determine if VirtualBox will work properly on your lab environment. You must use the same user (or at least a user with the same privileges) that will be finally used to work with this virtual machine in the virtual workshop, please, note that this is very important.

Our purpose is simple, to determine if:

- This user has enough privileges to create new virtual machines,
- This user has enough privileges execute VBoxManage to clone, and modify the properties of, virtual disks,
- The VirtualBox's internal network interface is working, and
- Optionally, VirtualBox machines can access the external network.

There exists two possible ways to test that VirtualBox is working properly. The first one is easier than the second, but it will require approximately 4 GB of your disk. The second one is more complex, but only requires 1.5 GB of your disk. These two alternative ways differs in how we are going to clone the virtual machines:

- Cloning the entire disks, or
- Working with differencing disks over an immutable shared disk.

The second choice consumes less disk space, but it can introduce unexpected results if we try to use snapshot feature (we have confirmed that on recent VirtualBox versions reverting to a snapshot state when using a differencing disk with autoreset disabled, it does not revert to the previous state, but it reset the differencing disk, reverting to the original state). So if you have enough disk space we highly recommend cloning the entire disk.

Cloning

Your starting point must be a machine with VirtualBox installed and the zip containing the virtual disk of our Debian model machine copied in one directory (this model machine is a minimal debian 5.0.6 + mysql + postgresql + pgpool-II). Remember that you must use the same user that finally you will use to follow the virtual workshop.

First, extract the zip file. This will extract a 1.06 GB file named "My new debian.vdi". This is the virtual disk of our original or model virtual machine. Now we are going to clone it.

It is not mandatory, but it is recommendable, that you copy the vdi file into the VirtualBox's HardDisk folder (usually in the user's home directory VirtualBox creates a directory called ".VirtualBox/HardDisks").

Run VirtualBox and open the Virtual Media Manager using Ctrl+D or menu entry File->Virtual Media Manager... (Fig. 1).

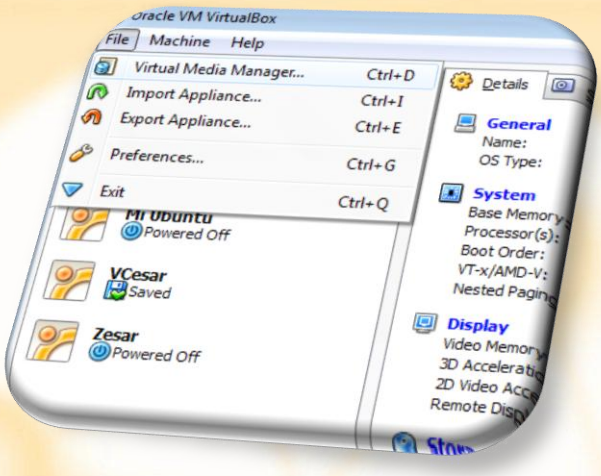

Fig. 1. Virtual Media Manager Menu Entry.

The Virtual Manager will appear and show your known virtual disks (Fig. 2).

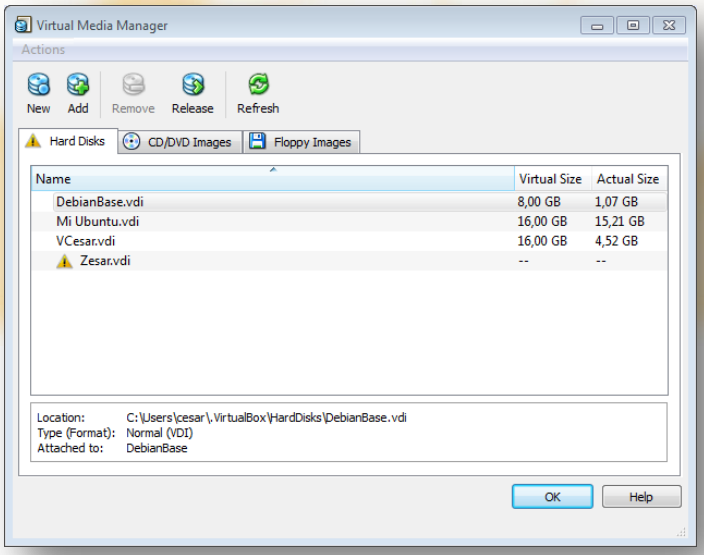

Fig. 2. Virtual Media Manager.

Click the Add button and add (Fig. 3) the new virtual disk to your virtual Hard Disks catalogue (Fig. 4).

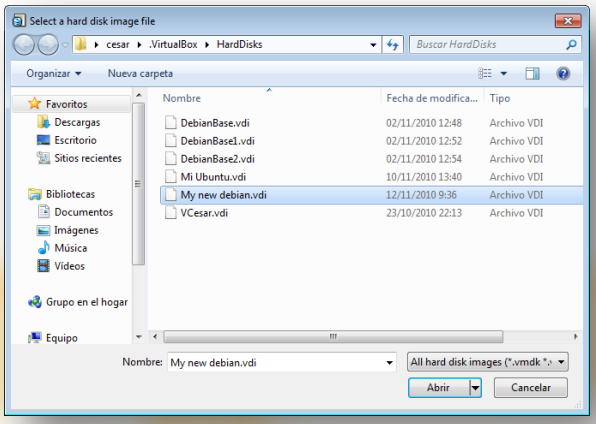

Fig. 3. Selecting Virtual Disk.

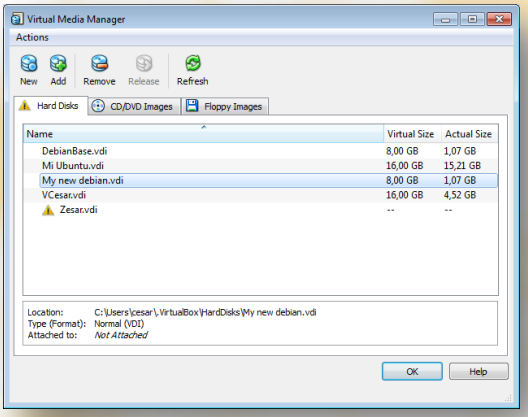

Fig. 4. Virtual Media Manager with the model disk.

Open a command window (or a shell) and go to VirtualBox's Hard Disks directory (Fig. 5).

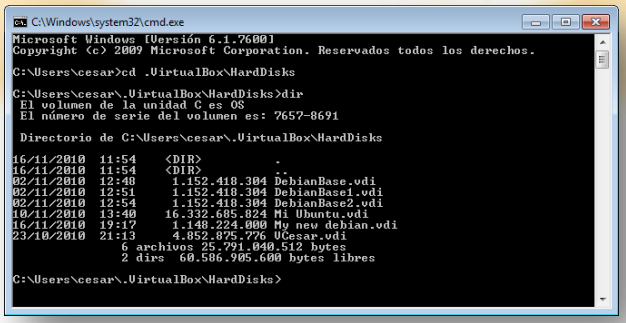

Fig. 5. Command window in VirtualBox's Hard Disks directory.

Now we must select a way…

Option 1: Cloning the entire disk

If you prefer the other way, please refer to Option 2: Working with differencing disks section.

Now we are going to execute the VBoxManage, a command line utility installed with VirtualBox. You must find where it is installed in your system. On windows it is usually installed on c:\Program Files\Oracle\VirtualBox directory. Let's assume that this is your VirtualBox's home directory.

Now execute these two sentences:

```
"c:\Program Files\Oracle\VirtualBox\VBoxManage.exe" clonehd "My new debian.vdi" "My new debian 2.vdi"
```

```
"c:\Program Files\Oracle\VirtualBox\VBoxManage.exe" clonehd "My new debian.vdi" "My new debian 3.vdi"
```

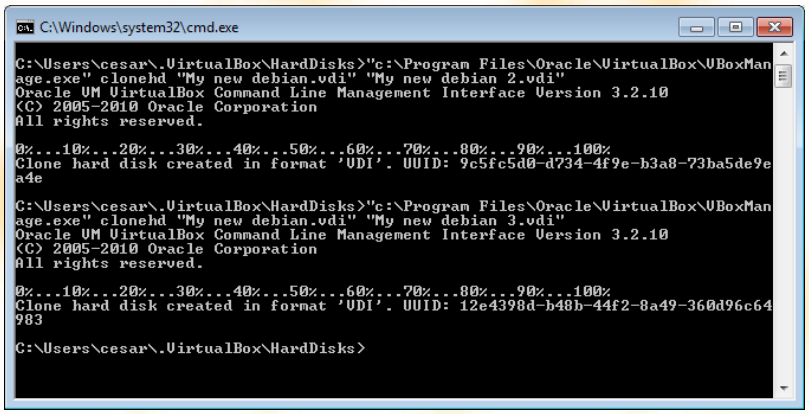

Fig. 6. Two clones.

If everything is ok, you will see something like Fig. 6, and you will have two new clones (Fig. 7).

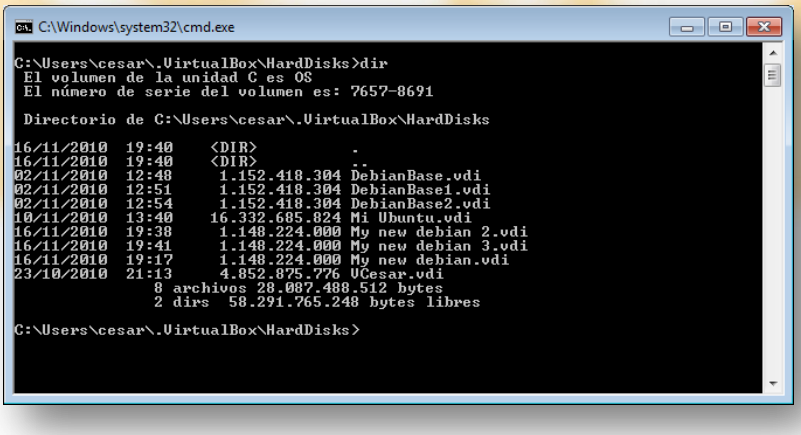

Fig. 7. Clone files.

Now we are going to add these two new disks to our catalogue using the Virtual Media Manager (Fig. 8), as we do at the beginning.

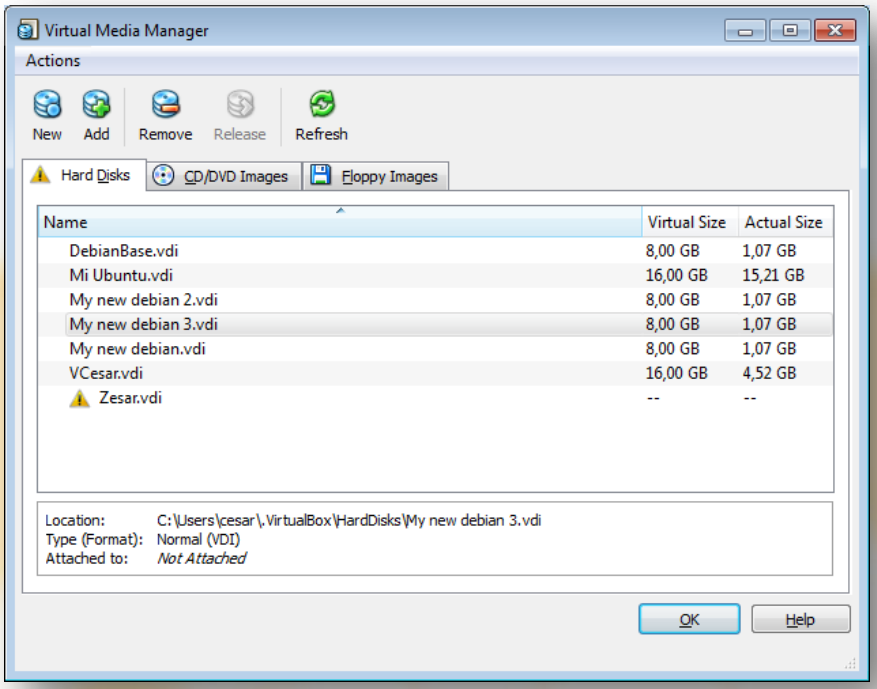

Fig. 8. Declared clones.

Now we create three new Virtual Machines, one for each "My new debian" disk. So we go to the main VirtualBox window (Fig. 9) and click the New button.

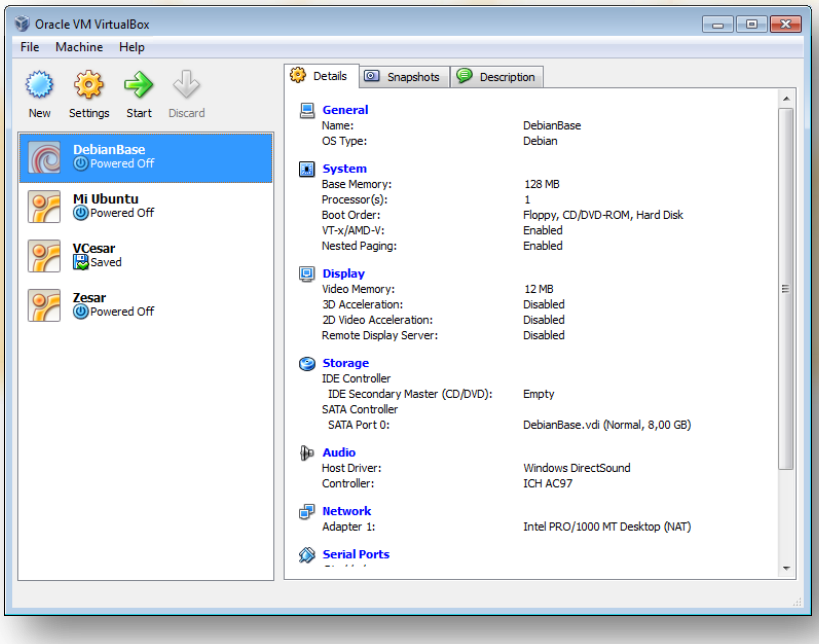

Fig. 9. VirtualBox main window.

The wizard starts (Fig. 10).

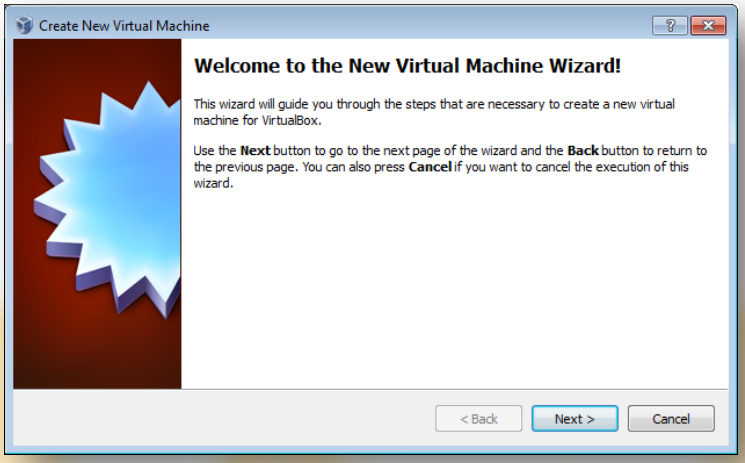

Fig. 10. New Virtual Machine Wizard.

Click on Next > and select your machine name (For example dbtech1) and type (Linux-Debian), Fig. 11.

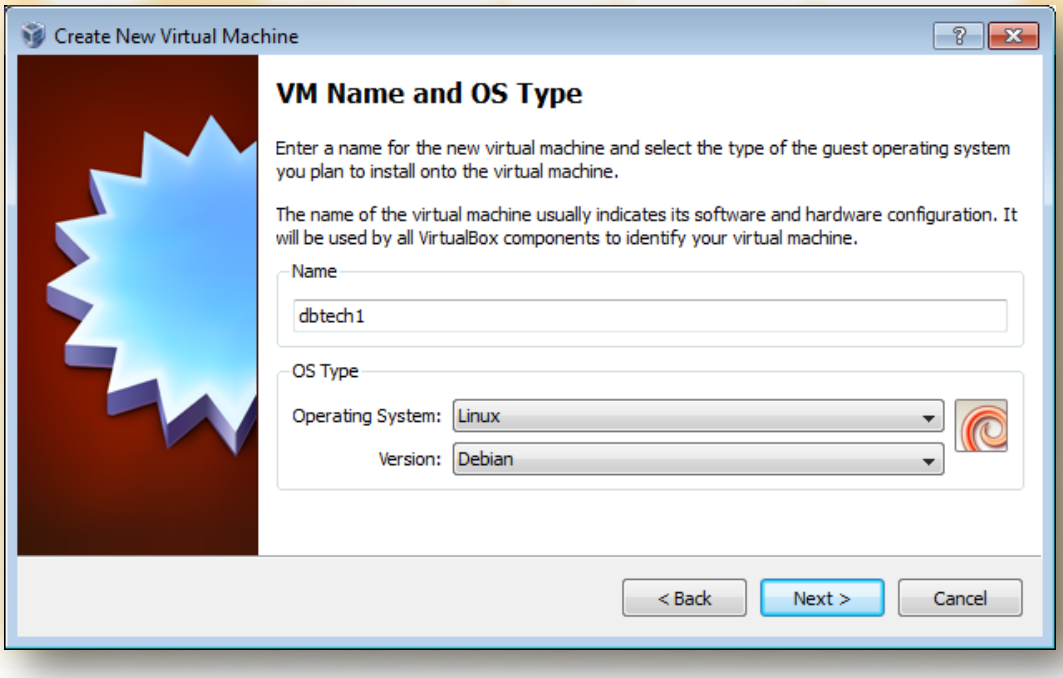

Fig. 11. VM Name and OS Type.

Click on Next > and select 128 MB RAM, Fig. 12.

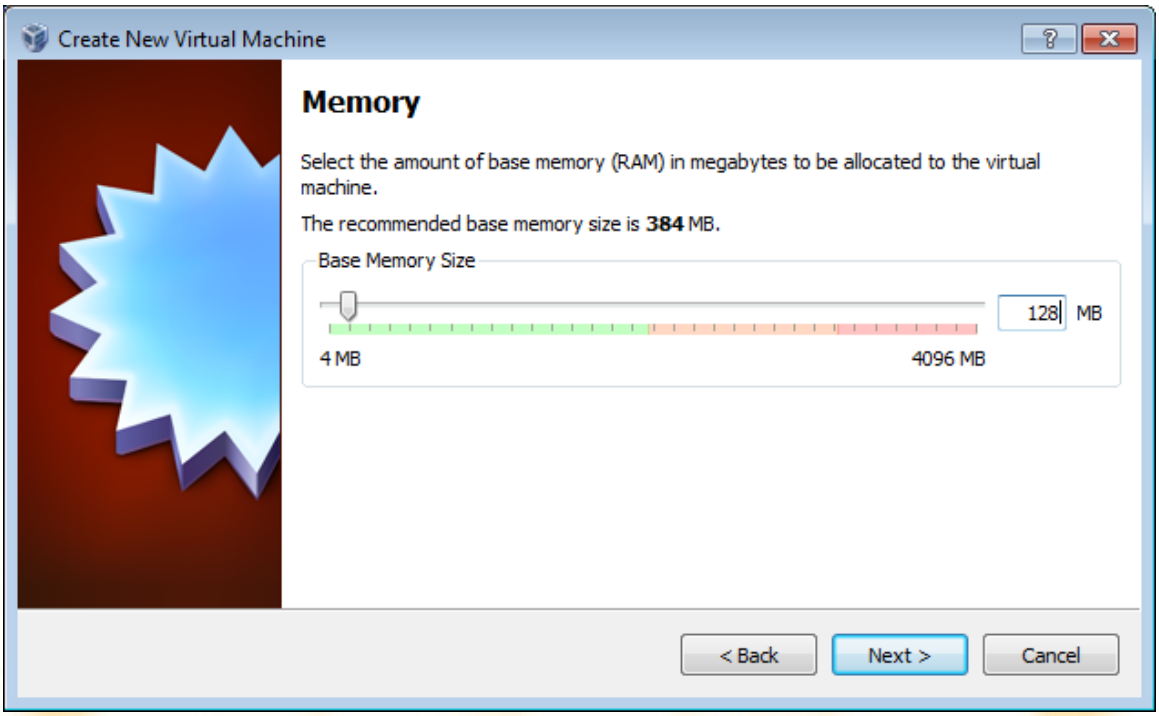

Fig. 12. Memory.

Click on Next > and as your Boot Hard Disk, select Use existing hard disk, and select My new debian.vdi in the drop-down list.

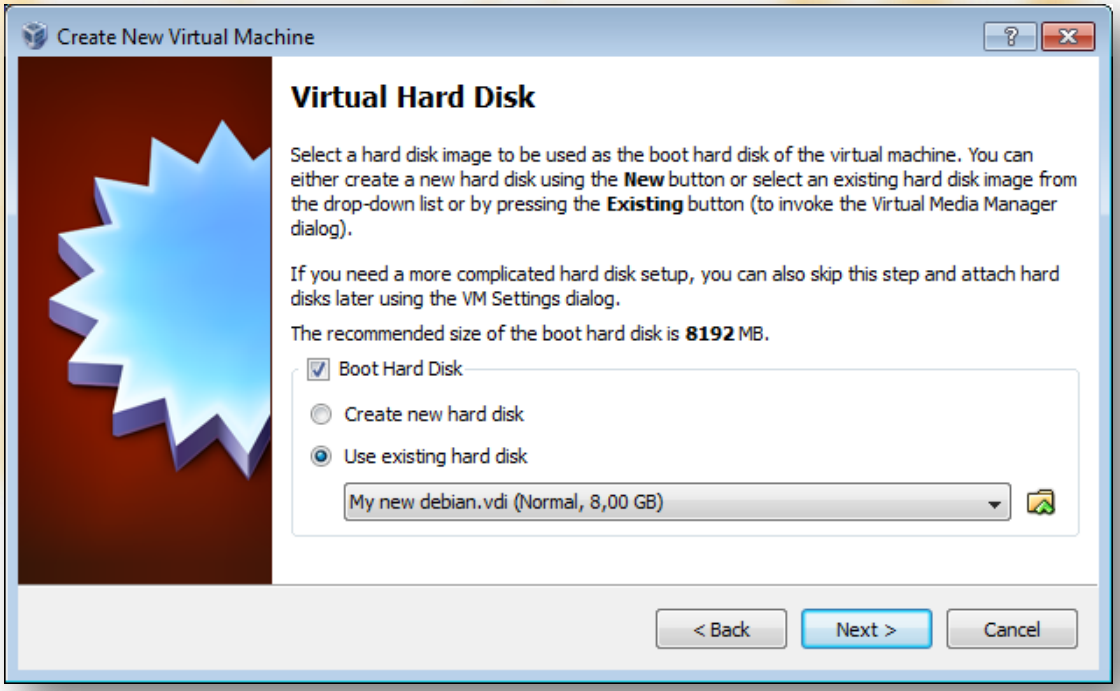

Fig. 13. Virtual Hard Disk window.

Click on Next > and Summary windows appears.

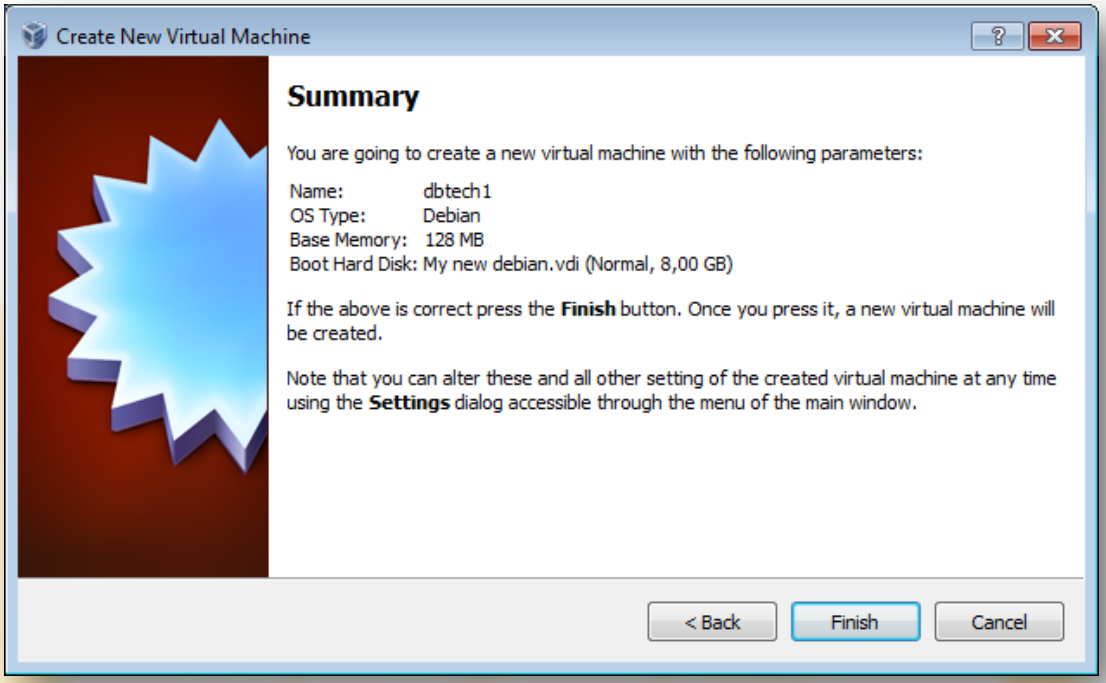

Fig. 14. Summary.

Click on Finish button and repeat the same process twice. Create dbtech2 machine using My new debian 2.vdi hard disk, and dbtech3 machine using My new debian 3.vdi, both of them with 128 MB RAM. The result is shown in Fig. 15.

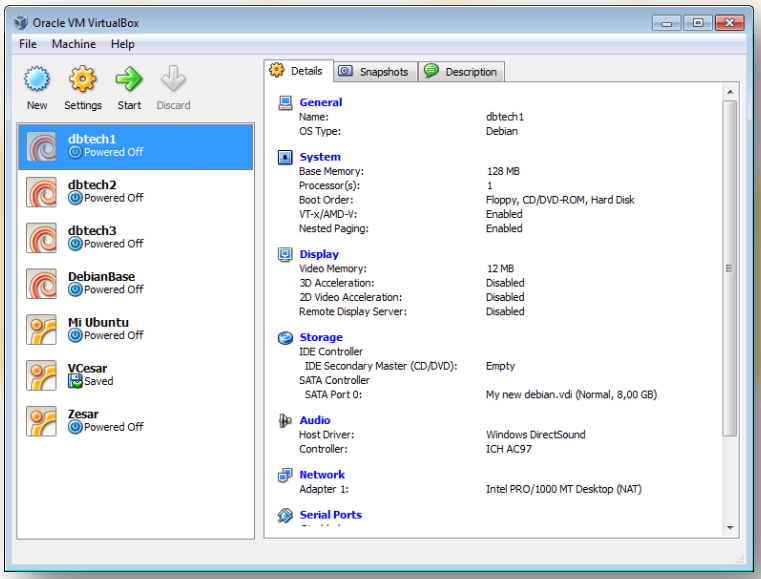

Fig. 15. Three machines.

If you have followed this option, please refer now to Configuring the network section.

Option 2: Working with differencing disks

Now assume that you prefer the second way. We start from Fig. 5.

As in option 1, we are going to execute the VBoxManage, a command line utility installed with VirtualBox. You must find where it is installed in your system. On windows it is usually installed on c:\Program Files\Oracle\VirtualBox directory. Let's assume that this is your VirtualBox's home directory.

Now execute this sentence (Fig. 16):

```
"c:\Program Files\Oracle\VirtualBox\VBoxManage.exe" modifyhd "My new
debian.vdi" --type immutable
```

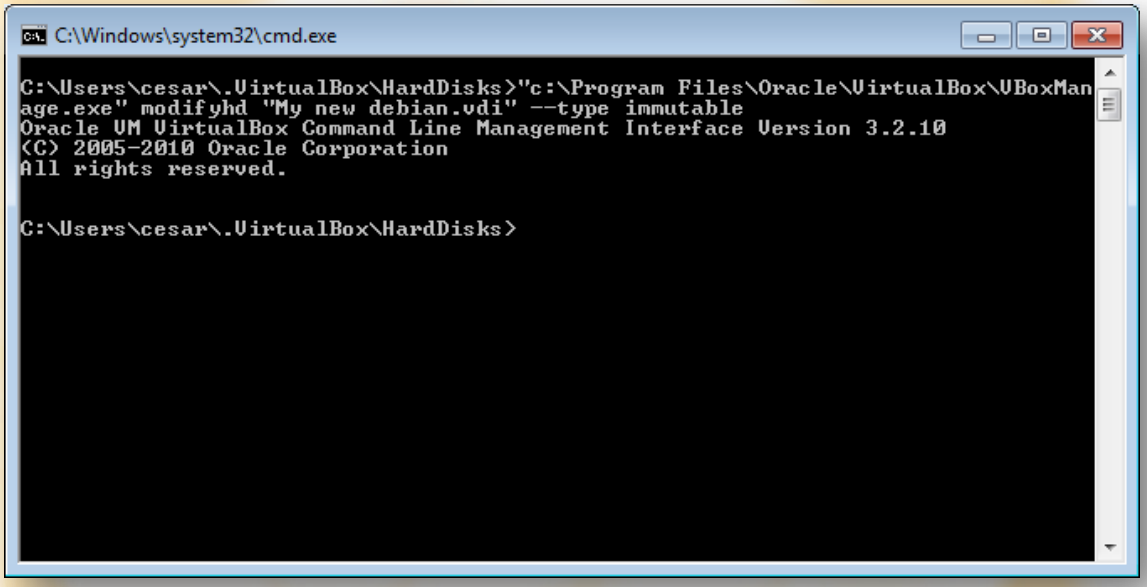

Fig. 16. Declaring immutable.

Now My new debian.vdi virtual disk will be treated by VirtualBox as a read only disk. Open the Virtual Media Manager, if you have the manager open click on Refresh button, and click on My new debian disk. On the bottom part of the window (Fig. 17) you will see the Location of the file and the Type(Format) and if it is Attached to some machine. Observe that now the type must be *Immutable (VDI)*, not *Normal (VDI)*.

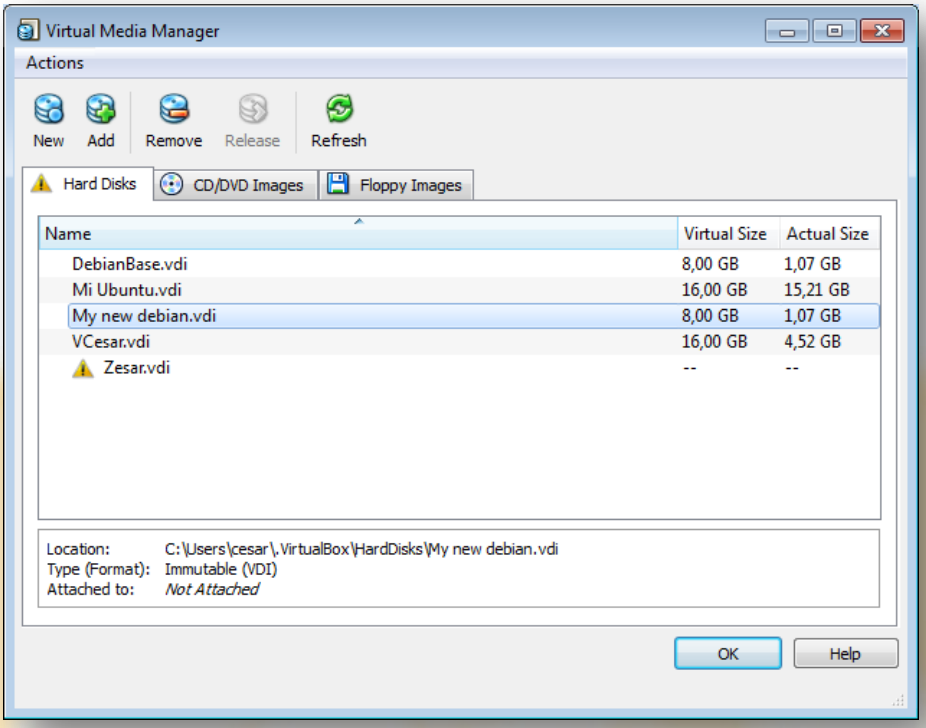

Fig. 17. Immutable type.

Now we create three new Virtual Machines, one for each “My new debian” disk. So we go to the main VirtualBox window (Fig. 18) and click the New button.

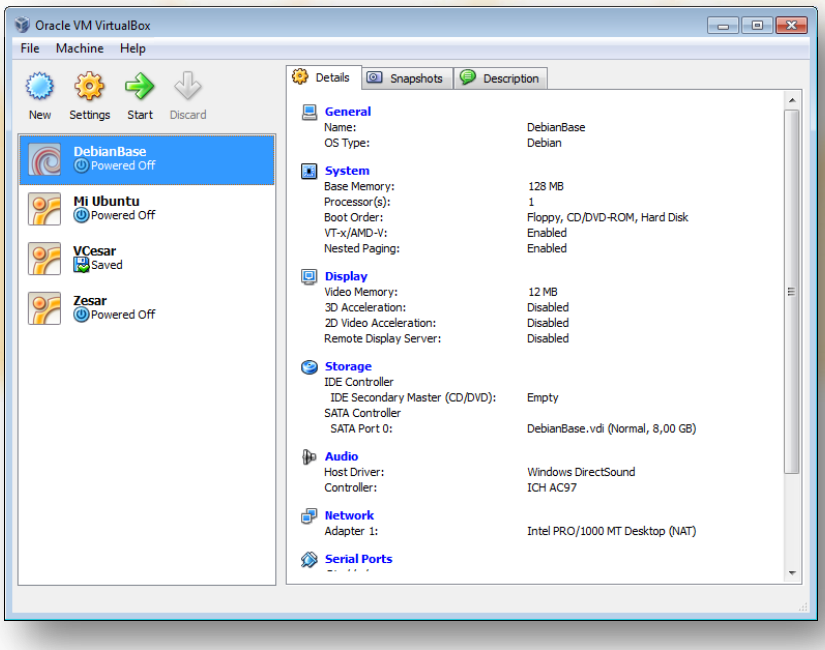

Fig. 18. VirtualBox main window (op. 2).

The wizard starts (Fig. 19).

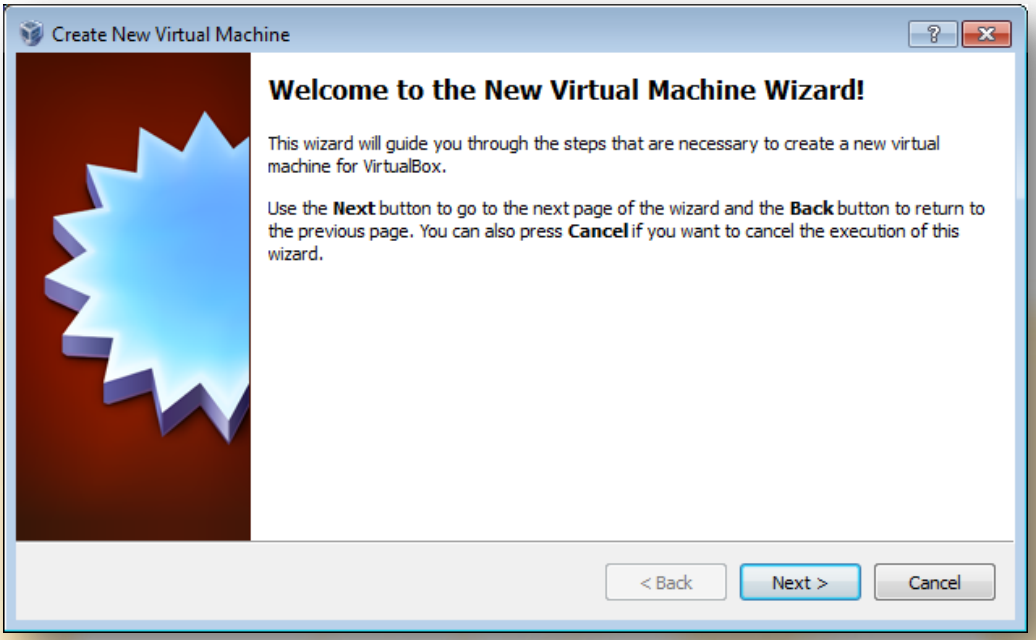

Fig. 19. New Virtual Machine Wizard (op. 2).

Click on Next > and select your machine name (For example dbtech1) and type (Linux-Debian), Fig. 20.

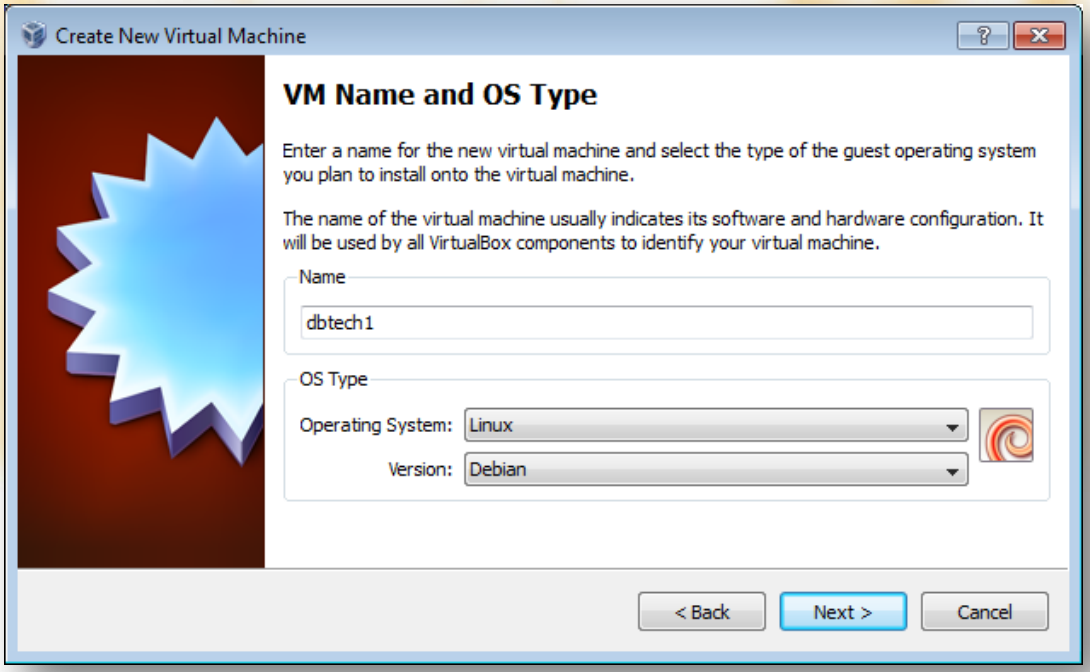

Fig. 20. VM Name and OS Type (op. 2).

Click on Next > and select 128 MB RAM, Fig. 21.

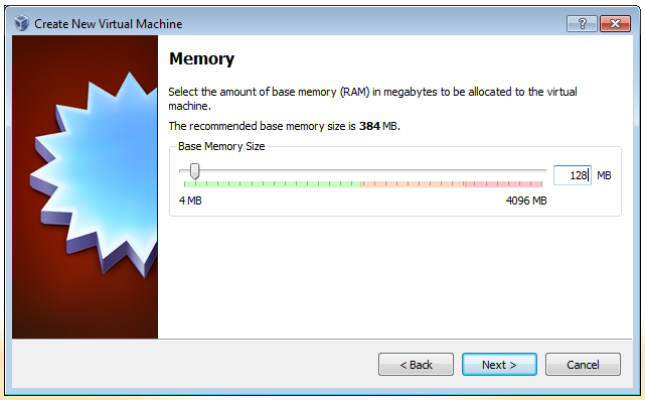

Fig. 21. Memory (op. 2).

Click on Next > and as your Boot Hard Disk, select Use existing hard disk, and select My new debian.vdi in the drop-down list, Fig. 22. Note the blue start before the name and the type after the name indicating both that it is an immutable disk.

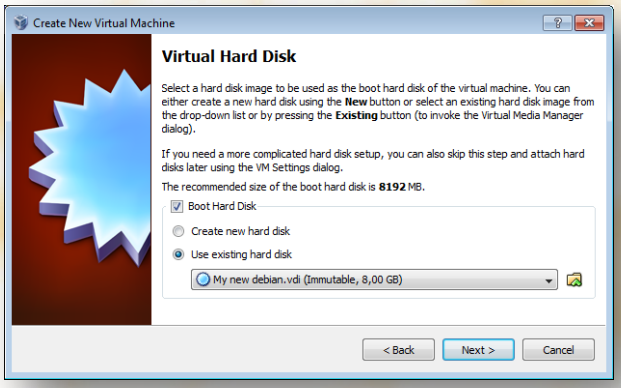

Fig. 22. Virtual Hard Disk window (op. 2).

Now the summary window appears (Fig. 23). Click on Finish button.

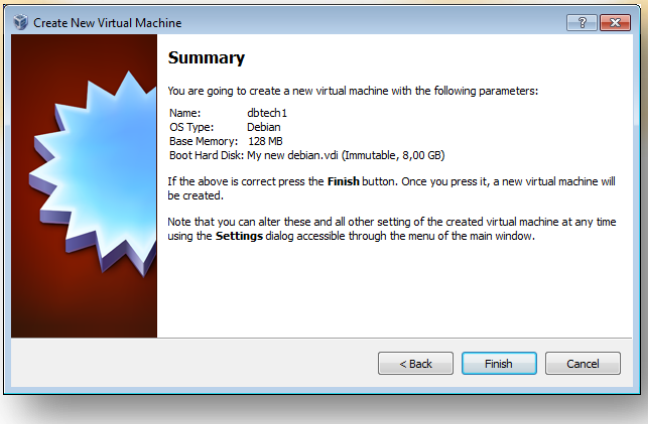

Fig. 23. Sumary (op. 2).

Immutable disks are not directly attached to virtual machines. VirtualBox automatically creates a differencing disk over the immutable disk, and attach the machine to this differencing disk. The differencing disk will store all subsequent disk writes, allowing the virtual machine to write on the disk, but maintaining immutable disk unaltered. Multiple differencing disks can be attached to a single immutable disk. VirtualBox will do a reset of the differencing disk every time the machine starts or restarts if it has the autoreset flag on; the content will be preserved if autoreset flag is off. Latest versions, declare differencing disk with autoreset off by default, but originally the default was autoreset on.

Repeat the same process twice. Create dbtech2 and dbtech3 machine using My new debian.vdi, both of them with 128 MB RAM. The result is shown in Fig. 24.

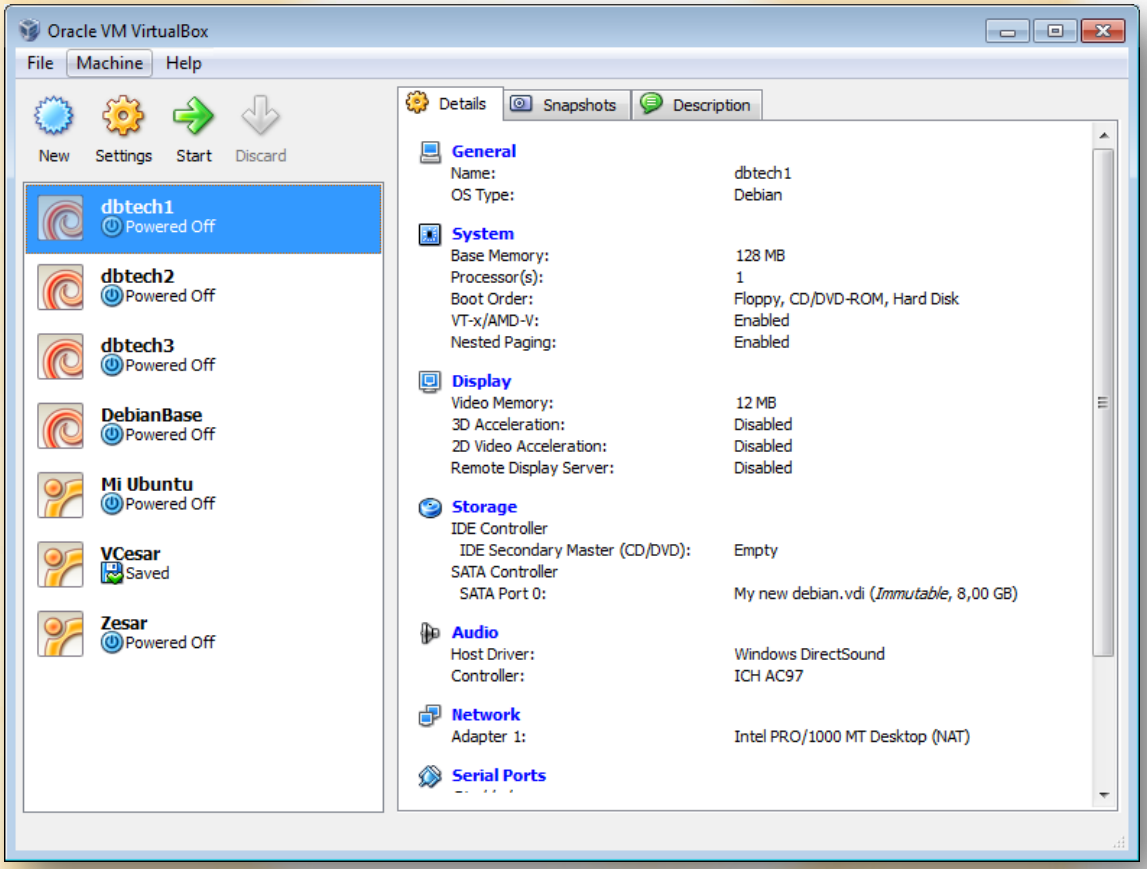

Fig. 24. Three machines (op. 2).

Now, for security, let's turn off the autoreset flag for the three differencing disk. Open the Virtual Media Manager and double click on My new debian.vdi, now three child disks appears (Fig. 25), our three differencing disks. The name will differ, because VirtualBox uses random UUIDs and use them as file names.

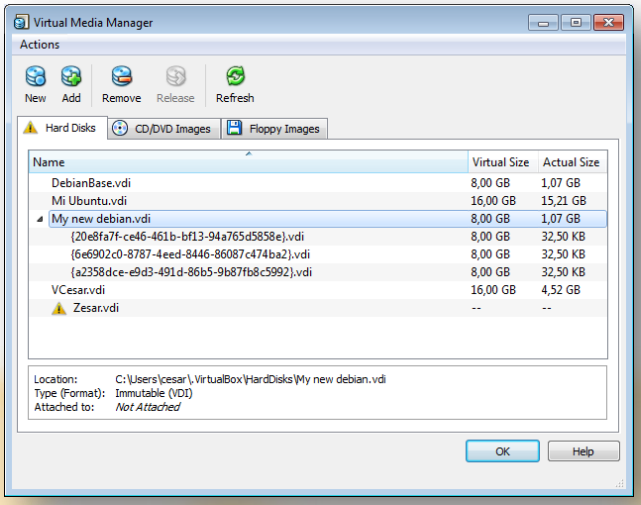

Fig. 25. Child disks.

Selecting each differencing disk you will see, on the bottom part of the window, its location (Fig. 26). Note that differencing disks are stores as snapshots in the corresponding machine directory, not in the HardDisks directory.

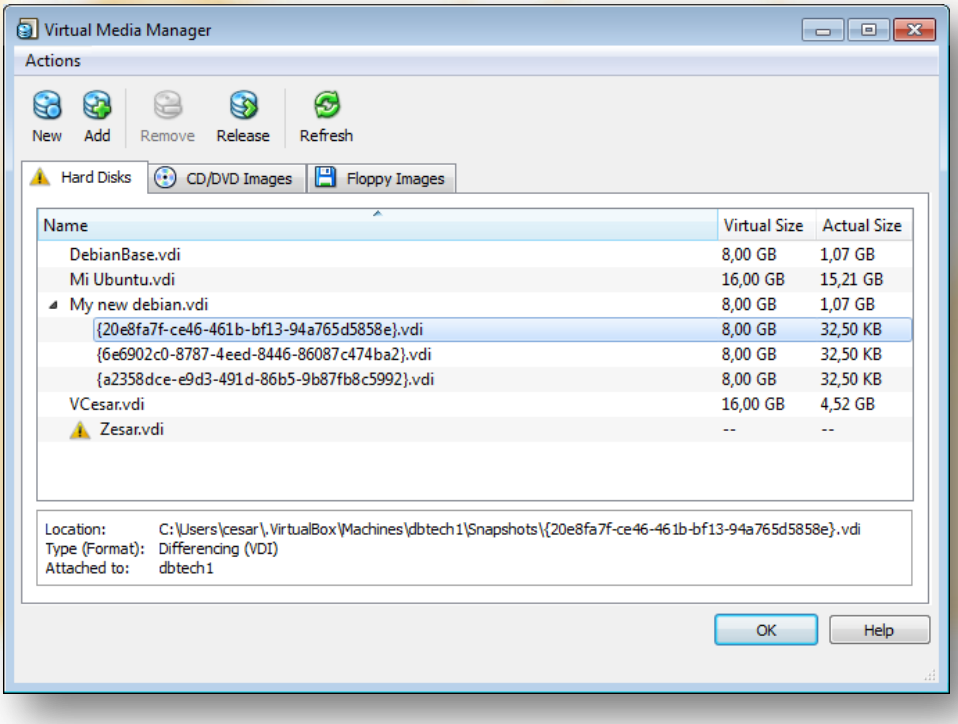

Fig. 26. Differencig disk location.

Open a command window (or terminal shell), go to each directory and execute this sentence:

```
"c:\Program Files\Oracle\VirtualBox\VBoxManage.exe" modifyhd uuid --
autoreset off
```

Use uuid instead of the file name, or you will get an error (Fig. 27). The correct sentence for the example would be:

```
"c:\Program Files\Oracle\VirtualBox\VBoxManage.exe" modifyhd 20e8fa7f-ce46-461b-bf13-94a765d5858e --autoreset off
```

A tip: you can use the auto-completion OS feature to extract the uuid or you can copy it from the location section on the Virtual Media Manager.

Repeat the process with the other differencing disks.

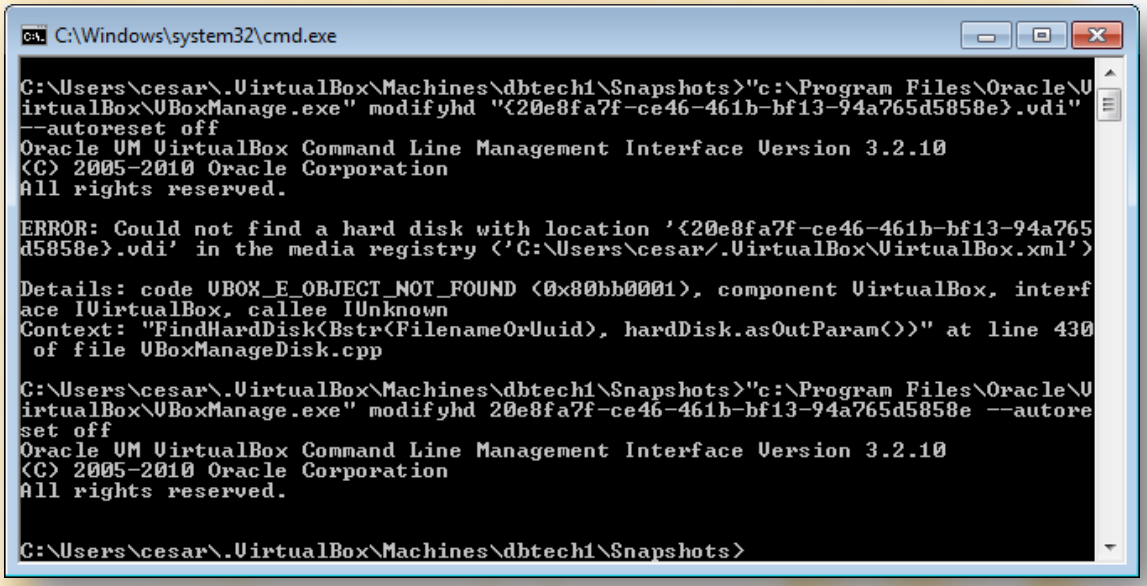

Fig. 27. Autoreset off.

Configuring the network

Here both ways (option 1 and 2) converge. Now we are going to configure the network infrastructure for each clone.

Go to main VirtualBox's main window, select the dbtech1 clone and click on Settings, go to Network tab, you will see that Adapter 1 is enabled an attached to NAT (Fig. 28). This means you're your host will act as NAT router for this virtual interface.

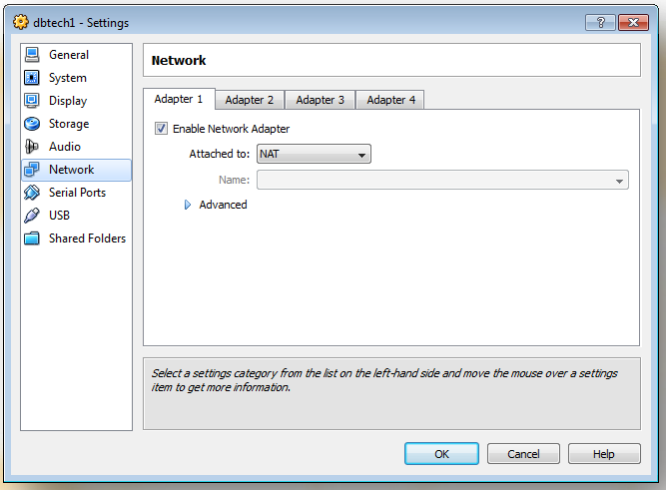

Fig. 28. NAT.

Go to Adapter 2 tab, enable it and attach to Internal Network, you can use different internal networks (identified by its name), let's leave the default intnet (Fig. 29).

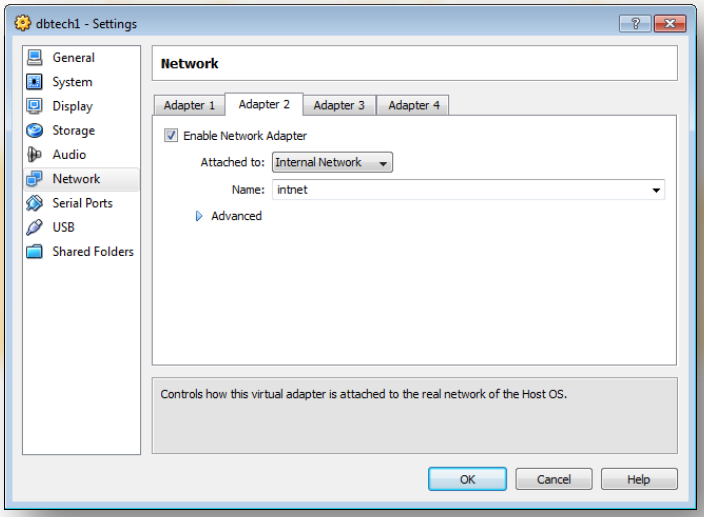

Fig. 29. Internal Network.

Click on Ok button and repeat the process with dbtech2 and dbtech3.

The virtual network hardware is configured, now is the turn to configure the network interfaces in the clones' OS.

On VirtualBox's main window select dbtech1 and click on Start button. Debian will start and if everythink is ok you will see something like Fig. 30.

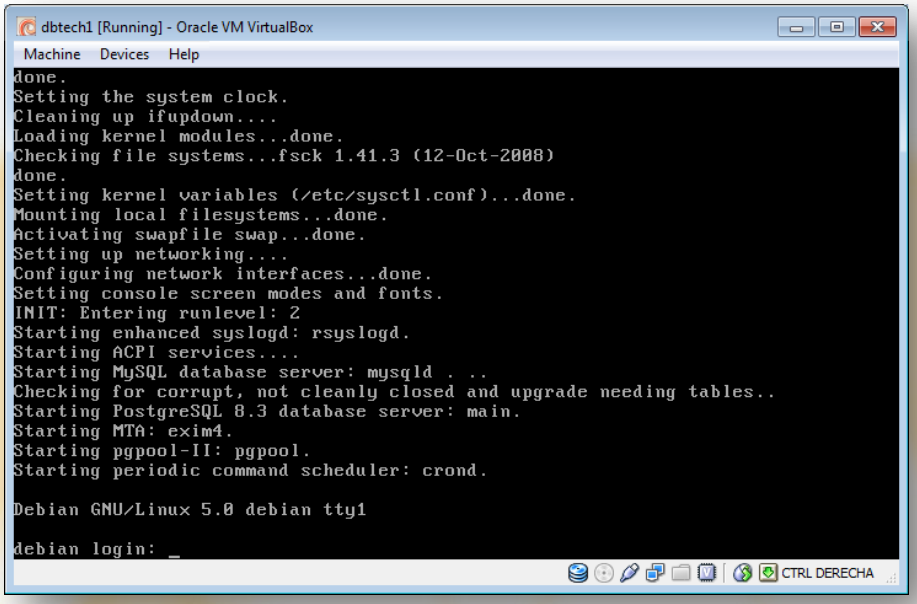

Fig. 30. Debian's login.

Login as root (password is dbtech). This machine has been configured with Spanish keyboard layout. If you want to change to your own keymap (the available keymaps are installed in /usr/share/keymaps directory). Most qwerty type keyboards are in /usr/share/keymaps/i386/qwerty/

If you have a Finnish keyboard, try:

```
install-keymap /usr/share/keymaps/i386/querty/fi.kmap.gz
```

or

```
install-keymap /usr/share/keymaps/i386/querty/fi-latin1.kmap.gz
```

Remember to reconfigure your keyboard on each clone.

If we execute ifconfig command (Fig. 31) we will see that we do not have any network interface but local (ip loopback).

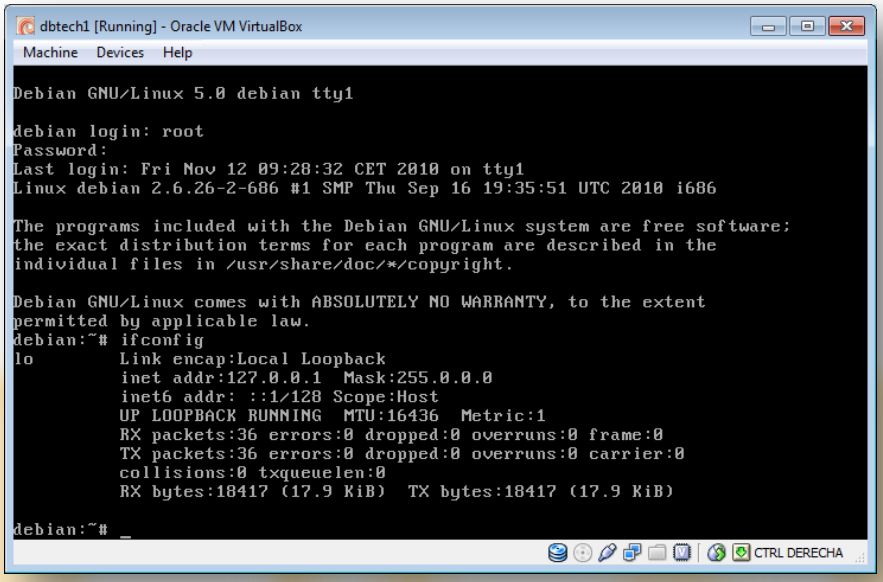

Fig. 31. No network devices.

The reason or this behavior is that we create the model machine using a virtual NAT network interface (with a random MAC address assigned by VirtualBox). All clones remember this original network interface and its MAC address. Each clone has added two new adapters (with their corresponding new random MAC address). So each clone has three adapters: eth0 the old one and eth1 and eth2 the new ones. By default Debian only configures eth0, but eth0 device is not present (remember that it is the older virtual interface used in the model generation). We need to help Debian to "forget" the old device, reassign NAT as eth0, internal as eth1 and properly configure them.

On the button status of the virtual machine, right-click on the network icon, and click on the contextual menu entry Network Adapters… (Fig. 32)

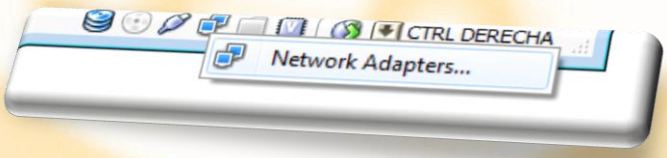

Fig. 32. Network entry.

You will see the Network Adapters window (Fig. 33). Here you can see the Mac Address assigned to each adapter. Leave this window open and return to dbtech1 window.

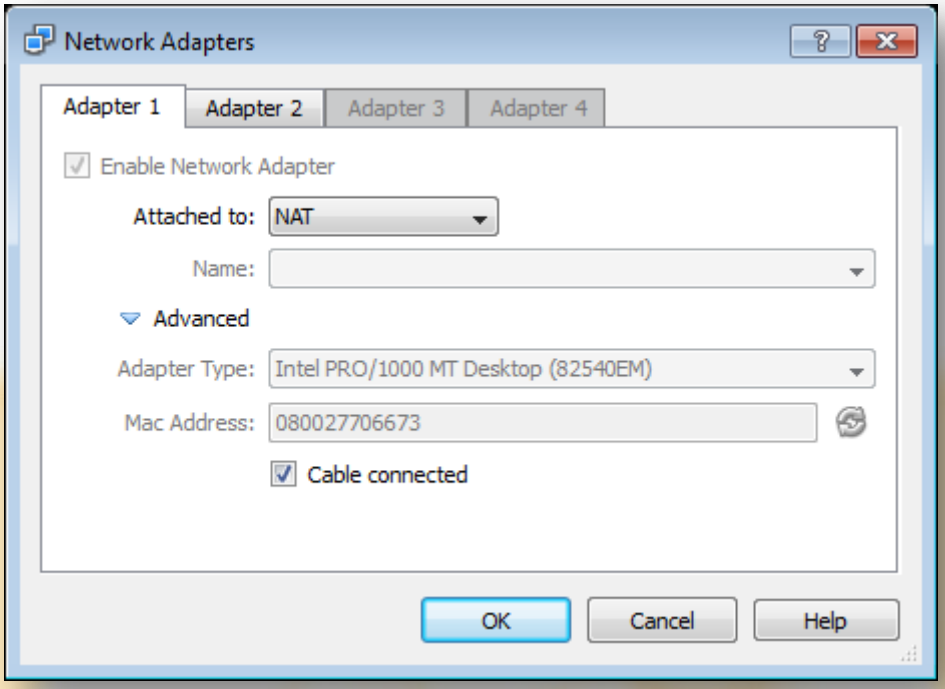

Fig. 33. Network Adapters window.

The MAC to ethX assignment in Debian is controlled in the file /etc/udev/rules.d/70-persistent-net.rules. We must edit it (use vi or nano). Execute the command:

```
nano /etc/udev/rules.d/70-persistent-net.rules
```

In this file there exists one line for each network interface detected (present or past). Lines starting with # are comments. Each network line declares some variables we are interested in ATTR and NAME. ATTR indicates the MAC Address and NAME the name that linux will assign to this interface. Assign eth0 as the NAME for NAT adapter (look for the corresponding Mac Address in the Network Adapters window). Assign eth1 as the NAME for internal adapter. And finally delete, or simple add a # character before (Fig. 34), the one that does not correspond with any of your adapters (this refer to the original model adapter).

Push CTRL+X and nano ask you if you want to save the changes; answer pushing Y to save the file. Nano will ask now the name of the file; by default it uses the original file name. Simply push ENTER key to confirm the name.

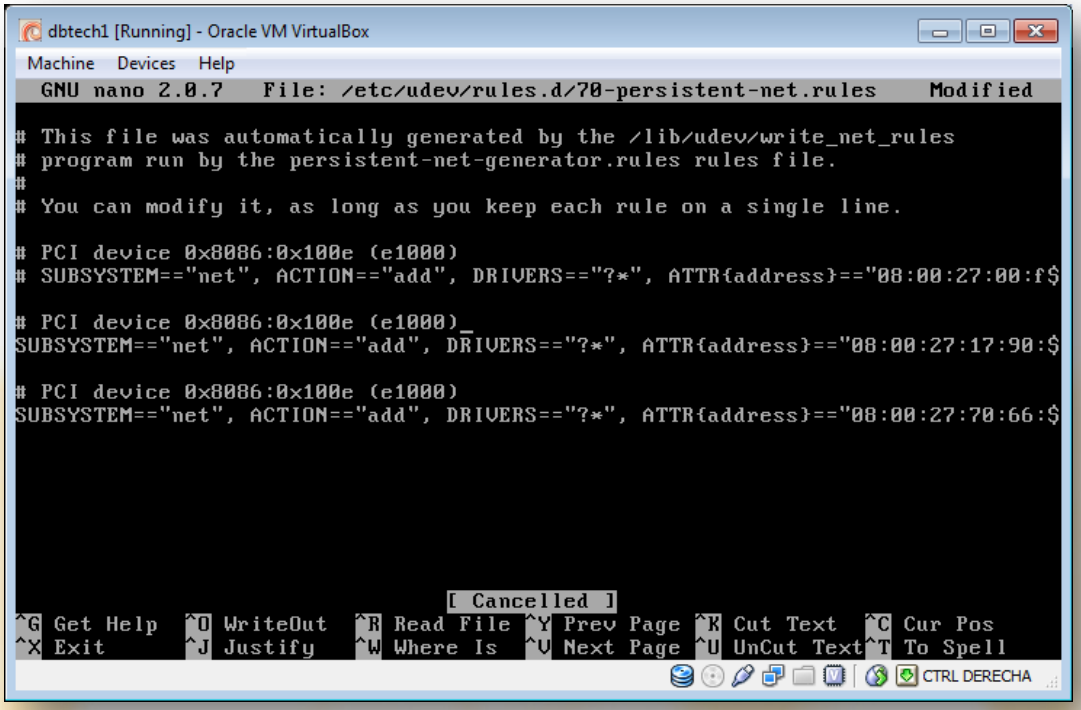

Fig. 34. 70-persistent-net.rules

Now let's edit the adapter configuration file. Let's execute:

```
nano /etc/network/interfaces
```

The adapter eth0 is, by default, configured to use dynamic DHCP (perfect for the NAT adapter). And let's configure eth1. We are going to use the net 192.168.192.0, using 192.168.192.1 for dbtech1, 192.168.192.2 for dbtech2 and 192.168.192.3 for dbtech3. We use this domain because it is a free subnet in our LAN. You can use any other free subnet if you have a conflict.

So we include at the end of the /etc/network/interface the configuration for eth1 (Fig. 35):

```
auto eth1

iface eth1 inet static

address 192.168.192.1

netmask 255.255.255.0
```

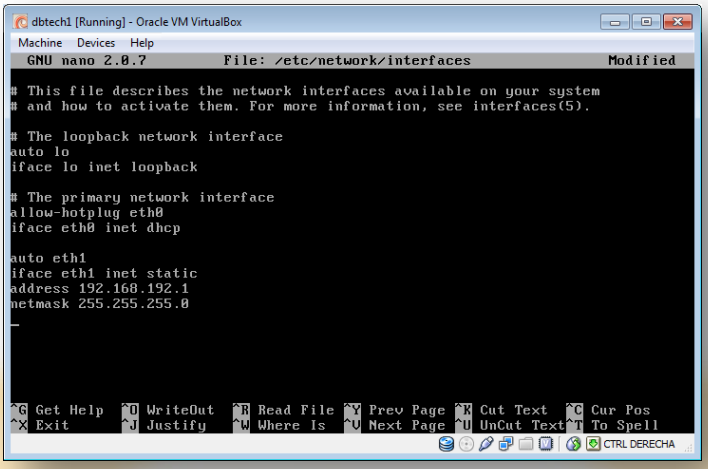

Fig. 35. interfaces.

Push CTRL-X, Y and finally ENTER.

Execute the command reboot (the machine will restart). After login as root execute the commands:

```
ifconfig eth0

ifconfig eth1
```

And verify that they are up, eth0 has an ip address, and eth1 has the 192.168.192.1 address (Fig. 36).

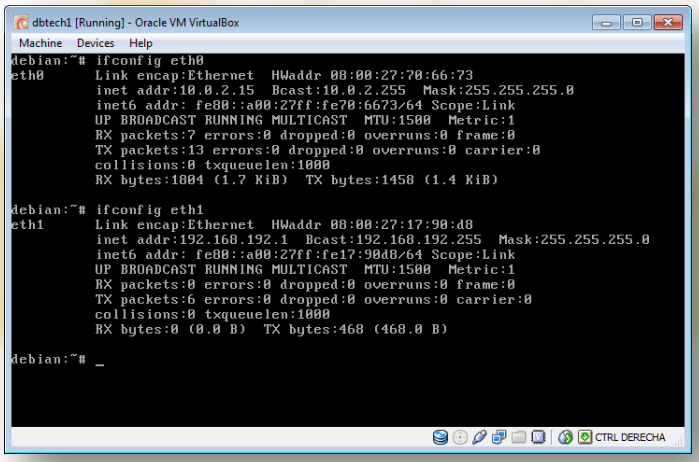

Fig. 36. ifconfig.

Now repeat the Debian configuration process with dbtech2 and dbtech3. Remember edit /etc/udev/rules.d/70-persistent-net.rules, edit /etc/network/interfaces and reboot. Don't forget to use 192.168.192.2 and 192.168.192.3 for dbtech2 and dbtech3 respectively.

After that, test if dbtech1 can access dbtech2 and dbtech3. Execute:

```
ping 192.168.192.2
```

Push CTRL-C is it dbtech2 respond. And repeat with 192.168.192.3.

If everything is ok, you will see something like Fig. 37.

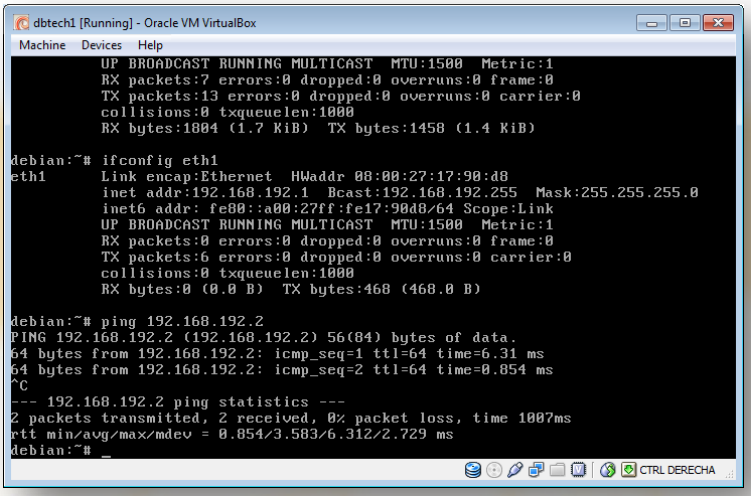

Fig. 37. Ping.

Now you have three clones correctly configured and interconnected. You can proceed with the virtual workshop.

But as an additional task, let's see if the NAT interface is working properly. If the host has direct internet access you only have to do something like:

```
ping www.google.com
```

and it must works (Fig. 38). But If your host is configured to use a proxy, then you must declare it on Debian. Simply declare the variables with your particular values.

HTTP_PROXY=http://username:password@proxyserver.net:port/
FTP_PROXY=http://username:password@proxyserver.net:port/

If your organization use anonymous proxy the use the form:

HTTP_PROXY=http://proxyserver.net:port/

FTP_PROXY=http://proxyserver.net:port/

You can declare them in your session or you can introduce these lines at the end of the /etc/profile file.

```
export HTTP_PROXY=http://username:password@proxyserver.net:port/

export FTP_PROXY=http://username:password@proxyserver.net:port/
```

exit the session and login again.

And not try to ping and external machine

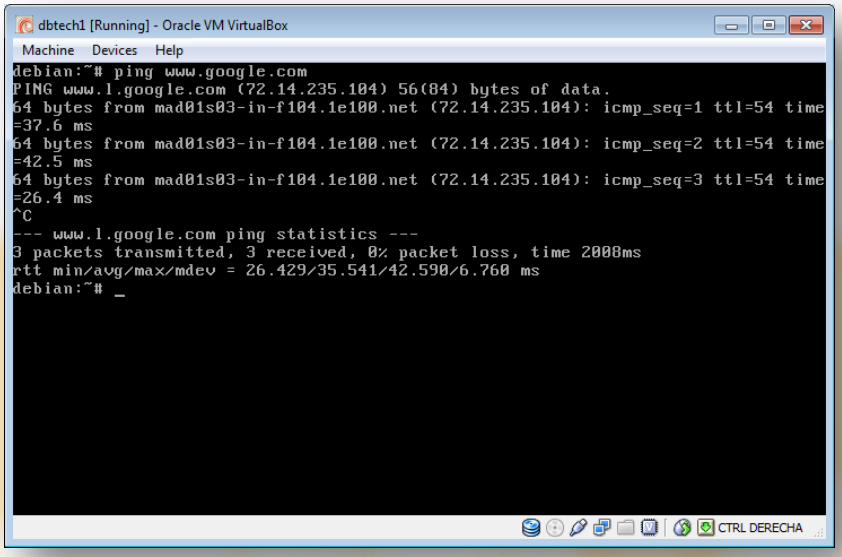

Fig. 38. External ping.

If you want to shutdown the machine, execute as root:

```
shutdown -h now
```

Figure Index

Fig. 1. Virtual Media Manager Menu Entry. 2
Fig. 2. Virtual Media Manager. 2
Fig. 3. Selecting Virtual Disk. 3
Fig. 4. Virtual Media Manager with the model disk. 3
Fig. 5. Command window in VirtualBox's Hard Disks directory. 3
Fig. 6. Two clones. 4
Fig. 7. Clone files. 4
Fig. 8. Declared clones. 5
Fig. 9. VirtualBox main window. 5
Fig. 10. New Virtual Machine Wizard. 6
Fig. 11. VM Name and OS Type. 6
Fig. 12. Memory. 7
Fig. 13. Virtual Hard Disk window. 7
Fig. 14. Summary. 8
Fig. 15. Three machines. 8
Fig. 16. Declaring immutable. 9
Fig. 17. Immutable type. 10
Fig. 18. VirtualBox main window (op. 2). 10
Fig. 19. New Virtual Machine Wizard (op. 2). 11
Fig. 20. VM Name and OS Type (op. 2). 11
Fig. 21. Memory (op. 2). 12
Fig. 22. Virtual Hard Disk window (op. 2). 12
Fig. 23. Sumary (op. 2). 12
Fig. 24. Three machines (op. 2). 13
Fig. 25. Child disks. 14
Fig. 26. Differencig disk location. 14
Fig. 27. Autoreset off. 15
Fig. 28. NAT. 16
Fig. 29. Internal Network. 16
Fig. 30. Debian's login. 17
Fig. 31. No network devices. 18
Fig. 32. Network entry. 18
Fig. 33. Network Adapters window. 19
Fig. 34. 70-persistent-net.rules 20
Fig. 35. interfaces. 21
Fig. 36. ifconfig. 21
Fig. 37. Ping. 22
Fig. 38. External ping. 23