



OSLO AND AKERSHUS  
UNIVERSITY COLLEGE  
OF APPLIED SCIENCES

## ASSIGNMENT 1

### PHYSICAL LAYER AND INFRASTRUCTURE

EXPLORING VIRTUALBOX, BUSYBOX, SHELL COMMANDS & PHYSICAL  
LAB

---

# Enterprise Networking: Practices and Technologies (MS018A)

---

*Submitted By:*

Simon Kiwanuka Takite  
s307043

Adam Buju  
Number

August 18, 2015

# 1 Installing Ubuntu 12.04 Server (x86) on a virtual machine

Virtualbox allows to run an operating system (Guest machine/OS) on top of another machine (Host machine/OS). Virtualbox runs as a hypervisor, controlling the host resources and processor, allocating resources as needed to each operating system and ensures that the guest machines do not disrupt the operations of the host machine. The virtual machine instances share hardware resource of the host machine hence the hardware resources of the host machine dictate the number of virtual instances that can run on the host machine.

In this lab, we will install ubuntu 12.04 server - x86 onto the host machine. The figure below shows my host system resources.



Figure 1: Host machine resources.

## 1.1 Creating a new virtual machine in VirtualBox (GUI)

1. Start virtualbox and follow the instructions on the new virtual machine creation wizard.



Figure 2: Virtualbox virtual machine wizard.

The wizard finishes by displaying the newly created virtual machine on virtualbox box machine list. Virtualbox will require an image to boot during the installation process. This can be specified by clicking on the virtual machine storage settings and selecting the ubuntu iso image stored on the hard drive and attaching it to the virtual machine CD/DVD attributes.

With all the necessary attributes specified, virtualbox will start the process of installing of the virtual machine. The new virtual machine will assume and act like its connected to physical hardware. The installation

follows a normal OS installation process by specifying language, hard disk format, location and other attributes required during the installation.

## 1.2 Creating virtual machines with VboxManage

Alternatively, virtualbox comes with a command line interface VboxManage to create , control and manage the virtual machines on linux hosts.

### Listing 1: VBoxManage CLI tool

```
1 #!/bin/bash
2 VBoxManage createvm --name "ubuntu-assignment-vm" --register
3 VBoxManage modifyvm "ubuntu-assignment-vm" --memory 512 --acpi
   on --boot1 dvd --nic1 bridged --bridgeadapter1 eth0 --
   ostype Ubuntu
4 VBoxManage createvdi --filename ~/VirtualBox\ VMs/ubuntu-
   assignment-vm/ubuntu-assignment-vm-disk01.vdi --size 8192
5 VBoxManage storagectl "ubuntu-assignment-vm" --name "IDE
   Controller" --add ide
6 VBoxManage storageattach "ubuntu-assignment-vm" --storagectl "
   IDE Controller" --port 0 --device 0 --type hdd --medium ~/
   VirtualBox\ VMs/ubuntu-assignment-vm/ubuntu-assignment-vm-
   disk01.vdi
7 VBoxManage storageattach "ubuntu-assignment-vm" --storagectl "
   IDE Controller" --port 1 --device 0 --type dvddrive --
   medium /iso/ubuntu-12.04.1-server-i386.iso
```

The script code listing 1 is fairly easy to understand, on line #2, we specify the name of the vm instance, assign memory, ostype and network options on line #3. On line #4, we create the hard disk, allocate its size and attach it to the vm instance on line #6. Line #7 attaches the installation media and the view will be added to our host machine.

## 2 Creating and cloning virtual machines from existing VM instances

### 2.1 VDI image installation

In this lab, a virtual machine was downloaded from <http://www.gns3.net/appliances> and installed following the steps described in the virtual machine installation wizard above 2. However, this time we had to specify a different linux flavor

and assign the vdi source to the VDI image `linux-microcore-4.0.2-clean.vdi` downloaded from GNS3 website.

#### Listing 2: Host\_Clean VM settings

```
1 absolootly@ubuntuhost:~/Dropbox/assignments/networking$  
  vboxmanage showvminfo "Host_Clean"  
2 Name: Host_Clean  
3 Groups: /  
4 Guest OS: Other Linux (32 bit)  
5 UUID: 07a5f8c3-9694-41e2-b170-7986399e5acc  
6 Config file: /home/absolootly/VirtualBox VMs/Host_Clean/  
  Host_Clean.vbox  
7 Snapshot folder: /home/absolootly/VirtualBox VMs/Host_Clean/  
  Snapshots  
8 Log folder: /home/absolootly/VirtualBox VMs/Host_Clean/  
  Logs  
9 Hardware UUID: 07a5f8c3-9694-41e2-b170-7986399e5acc  
10 Memory size: 256MB  
11 ....  
12 IDE (0, 0): /home/absolootly/Dropbox/assignments/networking/  
  linux-microcore-4.0.2-clean.vdi (UUID: 95a9df80-4431-49ba-  
  b777-764918b92c3b)  
13 IDE (1, 0): Empty  
14 NIC 1: MAC: 0800276D274F, Attachment: NAT, Cable  
  connected: on, Trace: off (file: none), Type: Am79C973,  
  Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy:  
  deny, Bandwidth group: none  
15 NIC 1 Settings: MTU: 0, Socket (send: 64, receive: 64), TCP  
  Window (send:64, receive: 64)  
16 ....
```

The listing 4 shows the excerpt from VBoxManage showinfo on host.Clean machine. On line #12, it shows that the virtual machine is running the linux-microcore image instead of the automatically generated VDI from the virtual machine creation wizard and assignend with our installation iso image.

## 2.2 Cloning a VM instance

Cloning a virtual machine creates a nearly identical copy of the original vm instance. Full clone creates an exact copy of the original machine and Linked clone creates a new virtual machine where the hard drives files are tired to the virtual hard files of the original machine. Hence, making it impossible to

port the new virtual machine to another host without moving the original as well.

In this lab, we used the virtualbox GUI to clone the Host\_Clean virtual machine into Host1 machine. During this process, we reinitialized the MAC address of all network cards on Host1 to prevent conflict in connectivity. Listing 3 and 4 line #2 show the differences between the two host machines.

---

**Listing 3: Host1 MAC address**

```
1 absolootly@ubuntuhost:~$ vboxmanage showvminfo "Host1" | grep  
  'NIC 1:' | awk '{ print $4}' | sed s/,//  
2 0800270B4B1A
```

---

---

**Listing 4: Host\_Clean MAC address**

```
1 absolootly@ubuntuhost:~$ vboxmanage showvminfo "Host_Clean" |  
  grep 'NIC 1:' | awk '{ print $4}' | sed s/,//  
2 0800276D274F
```

---

Both Host1 and Host\_Clean instances are connected to the internet via NAT. The virtualbox networking engine maps traffic from to the virtual machine transparently. A virtual router is placed between each virtual machine and host hence maximizing security between vm instances and the host.