**Topics: Services Installation and Configuration**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID\_\_\_\_\_\_\_\_\_\_ Section:\_\_\_\_**

***Today we will finally learn how to update our system, upgrade our system; and lastly how to install additional software or services. After that, you will learn how to configure some installed services. Since your machine by now contains a number of users who can log on, we will today attach the virtual machine to the network. Until now, the virtual machine Ubuntu Server that we have installed, set up and configured is running within Virtualbox, with a network connection through Virtualbox. In order to make it visible on the network, we need to create a 'bridge' from the virtual machine to the real network.***

Start your machine in Virtualbox. Once it is up and running, find out its network configuration, using ifconfig.

**ifconfig eth0**

shows (note the first 2 lines):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is okay, but not very readable. At least, you could not use this information automatically in a script: It is not possible to assign the IP-address to a variable. Therefore, we could use some shell script commands to process the output:

**ifconfig | grep -n1 eth | grep inet | cut -d ':' -f2 \ | cut -d ' ' -f1**

This should give you the current IP:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Next, we assign this address to a variable:

**IPADDRESS="`ifconfig | grep -n1 eth | grep inet \ | cut -d ':' -f2 | cut -d ' ' -f1`"**

*Here are some explanations for the syntax and the quotes and ticks:*

***ifconfig*** *shows the configuration of all interfaces*

***grep -n1 eth*** *extracts one line ('****-n1****') above and below the occurrence of the term* ***eth****'* ***grep inet*** *selects only the line with '****inet****' in it*

***cut -d ':' -f2*** *cuts the line into parts, with '****:****' being the '****d****'elimiter and selects the second piece ('****-f2****')* ***cut -d ' ' -f1*** *cuts the input into parts delimited by space (' ') and selects the first piece ('****-f1****')*

*The* ***backtick*** *(****`****) is used to make the two lines being run as a command. The result (output) is then assigned to variable IPADDRESS.*

*The* ***double-quote*** *(****"****) is used to force this output into a string. It also makes the program more readable and removes eventual ambiguities.*

Check if it has worked:

**echo $IPADDRESS**

Back to our undertaking: getting the virtual machine on the network. Until now, it is only on the internal (virtual) network on your PC that VirtualBox offers. We want it to be 'visible' on the whole network of Uniten, though. Therefore we need to reconfigure Virtualbox to this behalf.

You need to halt your virtual machine (Ubuntu).

Once it is stopped ("Powered Off"), change its network configuration in VirtualBox:

Details- >Network->Adapter 1

"Enable Network Adapter" must be active

Set "Attached to: " to "Bridged Adapter" → OK

When this is done, restart your virtual machine.

When you retrieve the network address of your machine (as explained further up in this lab), it should now give you an IP-address starting with 172.20.16 … . Once this is the case, your machine is available on the Uniten network ('LAN'), and can be reached from any place within Uniten. Congratulations!

Tell your lecturer the IP-address of your machine. He'll note it on the white-board.

From here on, everyone can check that your virtual machine is up and running; irrespective if from a Windows- or Ubuntu-console, by ping-ing this address (or these addresses):

**ping AAA.BBB.CCC.DDD** [where **AAA.BBB.CCC.DDD** is the address of that server]

Note the reply of one virtual server of your liking: [stop the replies with **Ctrl-C**]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

It \*is\* getting better!: Remember, we installed a webserver ('**LAMP**' → **L**inux, → **A**pache,

→ **M**ySQL → **P**HP) during the installation in Lab 1. So, all your machines run a webserver already.

Try this out, from **Windows** (we haven't installed any GUI in our server yet!), with a browser of your choice: start any browser and type the IP- address of any of those machines of **another** student (refer to the list of IP-addresses on the whiteboard) into the address-bar of your browser:

**http://AAA.BBB.CCC.DDD** [where **AAA.BBB.CCC.DDD** is the address of that server]

It still is a boring web content. And it gets worse: all your machines look all alike, because they show the **default installation message** of Apache on Ubuntu. You could always modify this content on your own box, install a Content Management System (CMS) like Joomla, or anything of your liking.

In this lab, we will only slightly modify the message, in order to prove that we are actually offering real-life web content.

All files and directories available as web-pages on your server are located in /var/www/. cd to this directory:

**cd /var/www**

For simplicity, just change the default welcome-message your server:

1. Save the existing file: sudo cp index.html index.html.save
2. Now you can edit the 'welcome'-file: sudo vi index.html
3. Be careful: Do not touch the HTML-tags (everything enclosed in < and >! Only modify the running text, put your name, instead, or another welcome message of your liking.
4. Better save frequently in between, and have another student try it out to see that modified message by reloading your web-page.
5. In case something goes wrong, you can always revert to the original version of the file: sudo cp index.html.save index.html
6. Have fun!
7. Close the browser and go back to your (virtual) Ubuntu server.

Next, we ween to install new services to the system. Before that, you need to update the system. Refer to the lecture notes: The update process starts with obtaining a suitable list of software repositories; the file sources.list in /etc/apt/.

**sudo apt-get update**

**sudo apt-get upgrade**

We have no GUI (yet), so let's check our local database of available packages for a database services on the command line. Which database services are available? Let's try:

**apt-cache search database | grep mysql**

There is one line that talks about some mysql database services: 'mysql-server'. Note this line:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Let us check out what the local repository knows about it:

**apt-cache show mysql-server**

Next, we will try to install the services.

**sudo apt-get install mysql-server**

To manage the database with GUI using web browser, install the phpmyadmin

**sudo apt-get install phpmyadmin apache2-utils**

* Select Apache2 for the server
* Choose YES when asked about whether to Configure the database for phpmyadmin with dbconfig-common
* Enter your MySQL password when prompted
* Enter the password that you want to use to log into phpmyadmin

After the installation has completed, add phpmyadmin to the apache configuration.

**sudo nano /etc/apache2/apache2.conf**

Add the phpmyadmin config to the file.

**include /etc/phpmyadmin/apache.conf**

Restart apache:

**sudo service apache2 restart**

Now, go to your web browser. Type his URL to see the pypmyadmin in interface.

<http://localhost/phpmyadmin>

You can see the webpage of your database. Now, you can create, remove or modify your database by using your mouse and keyboard. If you want to know how to use it, don’t miss the class for the Web Programming subject.

***During the installation of our server, we had as well installed OpenSSH. This is a utility that allows us to log on securely (encrypted) through the network.***

Create an account for at least one other student (with preference pretty far away from your PC). (Refer to earlier lab exercises if in doubt how to do this). Ask him / her for his /her preferred user name and create the account, including a home directory for your user. Then you either ask that user for the password of his / her choice, or you prescribe a password for him / her.

Once this is done, you may log on to your account on someone else's server, and others can log on to your server; all achieved **securely; through the network**!

The command to log on to an SSH-Server in \*nix (Ubuntu) is

ssh -l {username} {IP-address}, where the ' -l' is a dash-lowercase-L; **not** number one Example:

**ssh -l is055555 AAA.BBB.CCC.DDD**

[where **AAA.BBB.CCC.DDD** is the address of that server and **is055555** is the user name] Now you can move around on that other machine, over the network, and start to use it:

**whoami**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**hostname**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**who**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lastly, in order to simplify the installation of software collections, the administrator does not want to type all the package names. There is a tool for this: tasksel. It allows to select specific tasks for the server, and it will install all the necessary packages automatically.

We saw this in lab exercise 1, when we selected 'LAMP' and 'OpenSSH'. For those of you who forgot to do it then, it can still be done now.

**sudo tasksel**

Make sure that 'LAMP server' and 'OpenSSH server' are marked (you mark and un-mark with space). Once both are marked, use <tab>, the tab key, to advance to 'OK'. If the collections were installed during your initial install in lab exercise 1, you are done.

*This is an assignment. Please make sure that your email has been received by your lab tutor to get the marks.*