CMPE 283 Worksheet 01

Project 01

Deadline 0900 Tuesday, September 29

Classwork

In this class we will start setting up our droplets and Linux on our own computers.

Students who wish to set up a virtual server on a different provider from Digital Ocean or who want to use different operating system choices are absolutely free to do so. You should note that the instructions in the worksheet may not work exactly in other Linux distributions, though they will almost certainly work with slight modifications.

You should also note that your lecturer has not used Microsoft software at all for the past 14 years, so I will not have the knowledge to help you solve problems with Microsoft software, though you are, of course, free to use it if you have the necessary licences and are prepared to make the effort to support it yourself.

Setting up Linux as an alternative boot on your own computer will make it easier for you to experiment and learn offline and make certificated access to your droplet or other server easier. I strongly recommend you do this.

Setting up Linux on your own computer

1. Backup the data on your computer.

2. Boot Ubuntu from a USB stick. 64 bit verion is preferable on a modern computer, essential if RAM is more than 2 Gbyte.

3. Check that internet access and most common function are working in the Ubuntu booted from USB stick.

4. If all is ok click "Install" and follow the on screen instructions.

Setting up your droplet

I suggest you have Linux up and running on your computer while you are working on the server it will be handy, and also good practice.

If you do not yet have Linux on your computer, you will need to use a program like Putty to access your droplet.

Another useful thing to do. Open a text file in a text editor and keep it open on your desktop. The linux program is gedit. Write everything you do into the file. Copy/paste error messages you get there. Save the file frequently. This a record for you and also for anyone else (me in this case) who needs to help you. Believe me, having a record of what you did to "get into this mess" will save you hours of heartbreak later.

Now...

Log on to your droplet account through your web browser.

Go to your droplet by clicking on its entry on the main page.

Got to "settings" and change the droplet label to a sensible meaningful name to replace the number

Now we need to install Linux

Go to the "dashboard" for your droplet. Click "deploy a linux distribution"

Choose the latest 64 bit version of Debian GNU/Linux (7 at time of writing). We will use Debian because it is generally stable, does not change version very often and has a big community to support it. In terms of the politics of Free Software, Debian is one of the "purer" distributions and also one of the more democratic distributions. That is why change is slower but more reliable. Debian is also not tied to any one corporation. Ironically that makes it very important commercially.

You have some entries to make in the popup window.

You will need to give a root password for the new system.

Let's make a random one. Our server Linux is not yet up, so you can do this on your home Linux. Now you will see why I suggested you have Linux running on your machine.

Open up a terminal on your Ubuntu desktop. You can search for terminal in the Ubuntu dash that you get from clicking on the Ubuntu icon, then click on that. Or you can do it the right way, by pressing CTRL-ALT-T all three keys together. It's a lot faster than fiddling with the mouse. Now you have learnt how to do it, always

do it that way.

Type makepasswd to the terminal. You should get an error message that tells you how to fix the error (I hope). Follow the instructions. Type makepasswd again. You have a password. You can copy paste this from the screen (You really need to use the mouse to do this). You need to save it somewhere safe for yourself (off the computer, preferably) and you need to paste it into the browser screen where you are setting up your linux operating system.

While Digital Ocean is setting up your Linux, let's review what you just did. You tried to use a program that was not there in your home Linux. You were prompted to install it. You used two programs - sudo and apt-get to do that.

Sudo is program that will call another program with root privileges. It is a way of doing things that you are not allowed to do as normal user for just one command, without stepping up permanently to become root and living with the possibility of shooting yourself in the foot with any little mistake. Sudo is the default way of doing these things on Ubuntu. Su is the way of stepping up to be root permanently. We will set up our Debian system to also work with sudo.

The apt-get program is part of the system for installing packages on Debian. Because Ubuntu is Debian underneath, Ubuntu and Debian use the same package manager. Every Linux distribution has a (large) set of programs that have been compiled, set up, configured and documented to work correctly with that version of the operating system. These programs are stored in repositories for that distribution on the internet. You install and update them using a package manager program. The package manager programs are different for different distributions. Red Hat, Suse each have their own, different from Debian. The packages themselves can also be different. So a program like Apache may work or be configured slightly differently for different distributions. So you should always read the documentation for the distribution package as well as for the program. If you are looking for hints to solve problems, be sure to Google "debian apache" along with the error message, as your problem may be distribution specific.

Logging in to our Linux

Our Linux should now be set up on our droplet, so we can log in to it using ssh. From your terminal, type

```
ssh root@<ip no>
```

where ip no is the ip number for your droplet (you can see that on your droplet dashboard) You will be asked to accept the security key information for the server - just say yes enter the password.

You are now on the system and you are "root", so you can break anything you want.

Firstly we must do some elementary security work. The moment your system is up and running ill intentioned robots will start trying to log in to it.

We are now using the command line. The BASH command line, actually.

The BASH command line

Some useful beginning things to know about the command line.

Generally we will write something like

```
<program name> -<option1> -<maybe another option(s)> <some file name>  <maybe anoth
```

If we make a mistake on the command line, or want to repeat a command we typed before, we can use the up arrow key (as many times as we like) to get back to any command we typed before, then use the left and right arrow keys, delete and backspace keys to edit the line, then type return to execute it. This saves a lot of time. But be careful!

If our command references a file or directory, we can type the first one or more characters of the file or directory name, then use the tab key. If the file name is unique, the name is filled in for you. This drives my daughter crazy when I do it because she thinks it is "spooky". But it makes command typing super fast.

If we want to find out how a program or command works and what we should write after the program name, type

```
man <program name>
```

or for even more information, sometimes

```
info <program name>
```

To get out of the information pages, type

```
q
```

Try

```
man ls
```

and see what you get.

```
q
```

to get out. When we are using the command line, there is a "current directory". The assumption is that the files we are referencing are in this directory, unless we specify otherwise. The current directory will be shown before the prompt on your terminal. Be careful when using the command history, because a command you did before may do something different if the current directory has changed. Linux directories are quite like Windows directories, except the slash is a forward slash. Windows came after Linux and I guess Microsoft made everything slightly different for intellectual property reasons.

To change the current directory, use the command

```
cd
```

(this would be chdir in DOS)

In linux directories "." means the current directory, ".." means the directory above in the hierarchy, so we can sometimes write

```
cd ../../..
```

to get out of some deep place we have got to.

First let us make sure our system is completely up to date

```
apt-get update
```

checks the repository

```
apt-get upgrade
```

does the actual update, if necessary.

We need some user accounts.

We are already root, so we can do this directly

We will need to make passwords. Do the same as we did on Ubuntu to get the makepasswd program installed (no need for sudo, because we are root) If you try to use makepasswd, wthout installing it, you will not get the friendly prompt you got on Ubuntu to install it. This is because you are a Debian user now. You are going to get less help from the system. However, the same procedure as you used for Ubuntu will get makepasswd installed for you and you can get it installed in the same way.

Make a password.

Do

```
adduser yourname
```

follow the prompts and accept the default values for everything by pressing return. If you are paying attention you will realise that you need to enter "y" for the very last question.

Check what you have done. Open a second tab in the terminal program on your own computer. CTRL-SHIFT-T in the terminal window will do that.

```
ssh yourname@<ip no>
```

enter your password to log in

then

```
ls
```

will list your home directory Except, of course that you haven't made any files yet, so your home directory is empty. Type

```
touch yournametestfile
```

touch just makes an empty file with the given name for you. and try

```
ls
```

again

```
ls -l
```

will list your directory in more detail. We will come back to what that detail means in a session or two.

```
ls -la
```

even more detail. Now

```
logout
```

then go back to your original terminal tab.

Do

```
ls -la
```

Just to see what you have there.

Assignment

1. Finish up what you started in class. Be prepared to show what you have done in the next lab class.

©Chris Stephenson 2014