NCS 490 Lab 1

Tony Perez

Abstract

In out initial lessons of Linux Networking we learned how to work with the command line and how to get help when we did not know what a command meant. We also learned to use VIM because that is how we will be editing files on our CentOS box. This lab was to initially get us started with our CentOS Minimal VM. We walk through step on how to set up basic networking, partition resizing, and managing user and groups.

Introduction

Our initial set up of our CentOS Minimal VM we were given our IP addresses to use on the 10.103.67.0/16 network. We configured the VM to get it on the network and also be able to talk to the outside. CentOS uses a command line interface and so that is something many like myself have to get used to. VIM is our editor and that is something we have to practices using because it an editor that you must use constantly to be proficient in. According to our Professor VIM, once mastered, is powerful and efficient to use which I am starting to notice. We did run into some problems when setting up networking. We then went on to update and install software to our machine.

Network Configuration

Hostname:	pereztr-1
IP address:	10.103.67.80
Subnet:	255.255.0.0
Default Gateway:	10.103.0.1/16
DNS:	10.102.0.32

We first started off by editing some network files. We first disabled loopback on start up.
vim /etc/sysconfig/network-scripts/ifcfg-lo
ONBOOT=no

Then we edited our eth0 interface config and put in our given IP address subnet, and network in the file and also set it to boot on wake, turn on.
vim /etc/sysconfig/network-scripts/ifcfg-eth0
IPADDR=10.103.67.80
NETMASK=255.255.0.0
NETWORK=10.103.67.0
ONBOOT = yes

Next we edited the **f**ile which contained the DNS information.
vim /etc/resolv.conf
nameserver=10.102.0.32

Finally we changed our hostname and default gateway.
vim /etc/sysconfig/network
HOSTNAME=pereztr-1
GATEWAY=10.103.0.1

Restarted networking
/etc/init.d/network restart

Made sure the interfaces were up and could connect to the outside by pinging to google.
ping google.com
If we got a reply then we were set. We did face some problems when rebooting in our lab environment which said that our IP addresses were taken after restarting networking. We later found out that it was caused by two VM's and by taking them off we were able to use them.

After this we could use ssh and did not have to go through the web console. We ssh'ed into our VM, updated them and rebooted after all installing all updates.
yum update
reboot

Package Install

We installed vim, man, wget and links on our machine using the yum command and we were able to chain the different package name to do it in a single command.
yum install vim man links wget mlocate
updatedb
Vim is an improved Vi editor. Man is used to get manuals of command which gave information on what is was, how it worked and flags it used. Links is a web browser for our command line shell. Wget is used to download files from the web using HTTP, HTTPS, FTP protocols. The package mlocate lets us use the locate command to find files.

Hard Drive Resizing

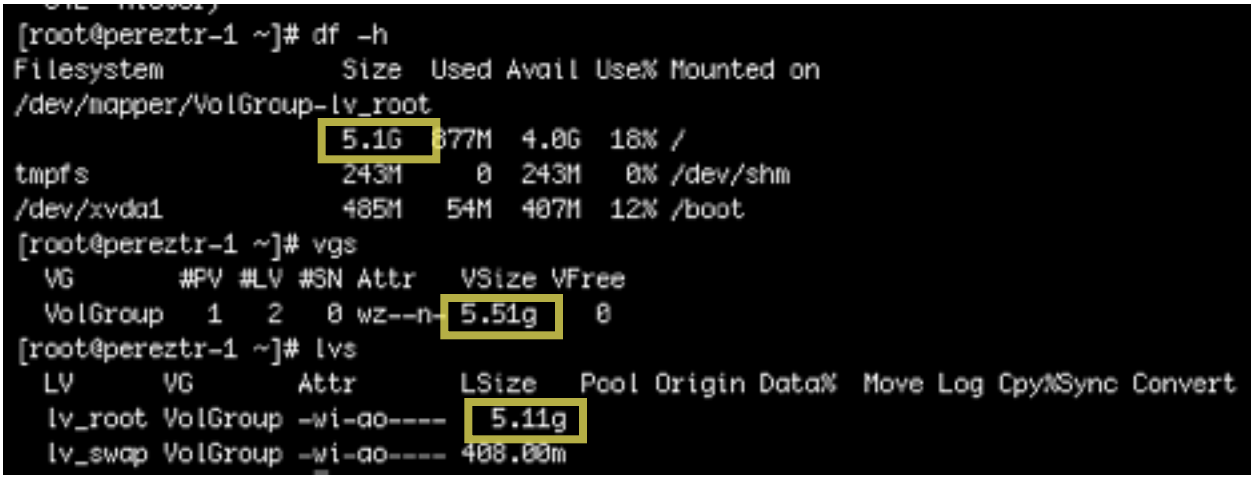
When resizing the hard drive we did it live. This meant we were using the hard drive we wanted resize. We went from 3.1GB to 5.1GB

fdisk /dev/xvda

```
[tony@pereztr-1 ~]$ sudo fdisk /dev/xvda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

:p
:d
:2
:n
:p
:2
:p
:t
:2
:l
:8e
:w
pvresize /dev/xvda2
lvextend -L+2G /dev/mapper/VolGroup-lv_root
df -h
resize2fs /dev/mapper/VolGroup-lv_root
reboot

We then checked to see if everything was fine and the resize was successful.
df -h
vgs
lvs



User & Group Management

We then went on to making a user for ourselves.
vim /etc/passwd
This was used to see users on the machine and describing them.
vim /etc/group
This displays the groups and which users are in them.
vim /etc/shadow
This file has the hashed password data to users on the machine.

We started by adding a user for ourselves so we do not have to use the root account.
useradd tony
passwd tony
type secure password

Logout and then log in using the account we just created. We can either add sudo to the beginning of our commands that require permission or move into the root user. I preferred being root.
su -
type password

We then added our regular account to the wheel group which allows that user to execute more privileged commands.
gpasswd -a tony wheel
Next we have to edit the sudoer file to let us give the wheel group privilege and if you want not require a password.
visudo

```
## Allows people in group wheel to run all commands
# %wheel        ALL=(ALL)       ALL

## Same thing without a password
 %wheel ALL=(ALL)       NOPASSWD: ALL
```

Logout completely for these setting to take effect.

We then had a task to create five users and groups: HR, Engineering, Accounting, Development, and Private group. We also and to assign the users to a group. This was done with commands:
adduser *username*
passwd *username*
enter password
gpasswd -s *username group*
My five users included: Jhuan, Kevin, Henry, Bob, Nicole and I gave them a password and a comment with their name. You can either add a comment by editing the /etc/passwd file or when creating the user using the flag -c.
useradd -c "*comment" username*

We then were assigned pages 100-103 of our text which also made us add users and group except they provided the information.
We added users Ying Yang and Mel Mellow and gave them passwords.
useradd -c "Ying Yang" yyang
useradd -c "Mel Mellow" -n mmellow
The flag -n will not make a group for mmellow but instead put it in the general users group.
We then added a bogususer and gave it a tcsh shell and put it in the group games.
useradd -s /bin/tcsh -g games bogususer
We then made and research, sales and bogus group
groupadd research
groupadd sales
For the bogus group we used the flag -r to make it a system group (its GID will be lower than 999)
groupadd -r bogus
We can verify
grep bogus /etc/group

```
[root@pereztr-1 ~]# grep bogus /etc/group
bogus:x:498:
```

We then modified user attributes with usermod. First we change the UID of bogususer
usermod -u 1600 bogususer
Verify
id bogususer

```
[root@pereztr-1 ~]# usermod -u 1600 bogususer
[root@pereztr-1 ~]# id bogususer
uid=1600(bogususer) gid=20(games) groups=20(games)
```

Next we change the GID of bogususer to that of bogus group and give it and expiration date.
usermod -g 498 -e 2017-12-12 bogususer

```
[root@pereztr-1 ~]# usermod -g 498 -e 2017-12-12 bogususer
[root@pereztr-1 ~]# id bogususer
uid=1600(bogususer) gid=498(bogus) groups=498(bogus)
```

View expiration information
chage -l bogususer

```
[root@pereztr-1 ~]# chage -l bogususer
Last password change                                    : Sep 17, 2014
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Dec 12, 2017
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

Now we play around with modifying groups with groupmod. We first change the name of name of group bogus to bogusgroup.
groupmod -n bogusgroup bogus
Then change the GID of the new bogus group to 1600
groupmod -g 1600 bogusgroup
Verify
grep bogusgroup /etc/group

```
[root@pereztr-1 ~]# grep bogusgroup /etc/group
bogusgroup:x:1600:
```

We then finish off by deleting user bogususer and group bogusgroup.
userdel -r bogususer
groupdel bogusgroup
The -r flag removes all files in the user's home directory.

Conclusion

In this first lab we ran through some basic step to configuring our CentOS VM. We got networking working, gave ourselves more storage, and made users. This is something that will be done when setting up fresh linux boxes or images.