

G Root Documentation

Task 1: Download VMWare Server Console Client and install on you PC or Laptop at home
<http://download3.vmware.com/software/vmserver/VMware-server-win32-client-1.0.6-91891.zip>

Task 2: Install Linux CentOS on your VM:

- Open VMWare Server Console
- Log onto Dingo server
- Select New Virtual Machine
- Select Typical > Select Linux version "Other Linux 64-bit"
- Name the VM "g_root" & set location: /opt/virtual-machines/students/
- Check: "Do not use a network connection"
- Make the virtual disk 15GB and uncheck allocate memory & split disk
- Click Finish.

Virtual Machine is now created. Now adjust the VM settings.

- Right Click VM > settings
- Adjust Memory to 512MB
- Select Processor has One core
- Click "Add" > Select ethernet Adapter
- Click "Custom" > Select from the dropdown menu "/dev/vmnet1 (host-only)"
- Click Finish

Continuing on the Hardware Tab:

- Select CD-ROM > check "Use ISO image"
- Select "Browse"
- Select "CentOS-5.11-x86_64-bin-DVD-1of2.iso" located in "/depot/mirrors/CentOS-5/isos/x86_64/"
- Click "Floppy"> Uncheck "Connect at power on"

Open the Options Tab:

- Click "Permissions"
- Uncheck: "Make this VM Private"
- Click "Finish"

Start VM with the click of the green button:

- Select "Skip"
- Click "Yes" to the warnings
- Select "Edit" > Uncheck "IPv6"
- On IPv4 menu select "Manual Configuration"
 - IP: 131.94.134.5
 - Subnet: 255.255.255.224
 - Host Name: cts4348-node-5.cs.fiu.edu
 - Gateway: 131.94.134.1
 - Primary & Secondary DNS: 131.94.134.129, 131.94.134.130
- Click "Next"> Select the Eastern Time Zone
- Create a strong root password
- Lastly Reboot

Task 3: Make your Machine an Operational Webserver

- Install the essential updates: yum update
- Install the apache web server: yum install httpd
- After updates start the apache web service httpd:

- `chkconfig --add httpd`
 - `chkconfig httpd on`
- Configure the Apache webserver by editing the `httpd.conf` file
 - `cd etc/ httpd/conf`
 - `gedit httpd.conf`
- Scroll down and set the `ServerAdmin` & `ServerName`
 - Set `ServerAdmin`: `gespi039@fiu.edu`
 - Set `ServerName`: `cts4348-node-5.cs.fiu.edu:80`
- Reset Apache services: `service httpd restart`

Create the `index.html` file

- Create the `index.html` file, head to the root folder : `cd /`
 - `gedit var/www/html/index.html`
- Inside the editor use basic html tags(`html`, `body`, etc) to create content
 - Member Names
 - Hardware elements of the workstation (virtualized hardware)
 - Which Linux version/distribution is installed
 - Hostname and network setup
 - All software packages installed, updated or changed
- Save the File

Task 4: Configure Firewall to open up port 80 to access your web site from the outside world

- Open port 80 via configuring the `iptables` file: `cd /`
 - `gedit etc/sysconfig/iptables`
- Once `iptables` file is open add the following:
 - `-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT`
- Restart the `iptables` service: `service iptables restart`

Task 5: Implement Web Server Directory-level Access Protection

- Create a new directory inside `var/www/html` & name it "journal". Type to begin: `cd /`
 - `mkdir var/www/html/journal`
- Create a `.htpasswd` file in `html` directory to store username/password pair
 - `htpasswd -c var/www/html .htpasswd "username"`
 - You will be prompted to create a password
 - To add more users type: `htpasswd .htpasswd "username"`
- Create an `.htaccess` file in the "journal" directory,
 - `gedit var/www/html/journal/.htaccess`
 - Inside `.htaccess` type:
 - `AuthUserFile /var/www/html/.htpasswd`
 - `AuthType Basic`
 - `AuthName "Please Login"`
 - `Require valid-user`(allows users that are in `.htpasswd` to access the directory)
- Create link & the journal webpage authenticated users will access
- The `HTML(journal.html)` file will be placed in `var/www/html/journal`
 - `cd /`
 - `gedit var/www/html/journal/journal.html`
- Edit the `index.html` file to contain a link to `journal.html` using ``
 - `gedit var/www/html/index.html`
 - `Login`

- Create the profe account
- Head to the root directory: `cd /`
- Edit the “passwd” file located inside “etc” directory
 - `gedit etc/passwd`
- Add the following line “profe:x:600:600:Jose F. Osorio:/home/profe:/bin/bash” at the bottom of the “passwd” file
- Change the permissions on the “shadow” file to allow read & write from the root
- Now edit the “shadow” file located inside the “etc” directory and type the following:
 - `profe:1ovjpj.Oh$OYY8Wmkdxl6cgdoHp8MiF1:13904:0:99999:7:::`
- Next open the “group” file located in “etc” directory and add the following
 - `profe:x:600:`
- Create the profe home directory inside /home. Type `cd /`
 - `mkdir home/profe`
 - `cd home`
- Change the permissions and ownership of the “profe” directory
 - `chown profe profe`
 - `chgrp profe profe`
 - `chmod u=rwx profe`
- Copy the following login files from root's home directory, change filecd ownership :
 - `cd /`
 - `cp ~root/.bashrc /home/profe`
 - `chown profe.profe /home/profe/.bashrc`
- Customize shell prompt for profe account
- Shell prompt should display [user id @ hostcd name current-system-time absolute-path-pwd # shell-command-history-number]
- Start by editing the .bashrc file in the “profe” directory located in “home”
 - `gedit home/profe/.bashrc`
 - add the following “`export PS1="[u@\H \T:$PWD #\!]"`” under “`alias mv= 'mv -1'`”
- Create a “.profile” file inside the “profe” directory
 - `cd /`
 - `vi home/profe/.profile`
- Inside the “.profile” type:
 - `if [-f ~/.bashrc]; then`
 - `./~/.bashrc`
 - `fi`
 - `export PS1="[u@\H \T:$PWD #\!]"`
 - `PATH=$PATH:$HOME/bin`
 - `export PATH`

Task 6: Modify your Apache httpd configuration to set up three name-based_ virtual hosts

- Create two directories inside “var/www”
 - `mkdir var/www/profeweb1`
 - `mkdir var/www/profeweb2`
- Create an “index.html” file inside each of the directories created
 - `vi var/www/profeweb1/index.html`
 - `vi var/www/profeweb2/index.html`
- Create HTML based content inside both “index.html” files

- vi var/www/profweb1/index.html
 - Welcome to PROFETEST1 Web Site
 - vi var/www/profweb2/index.html
 - Welcome to PROFETEST2 Web Site
- Edit the “httpd.conf” file in etc/httpd/conf
 - vi etc/httpd/conf/httpd.conf
- Look for “NameVirtualHost *:80” and uncomment that line
- Add the following to the end of the “httpd.conf” file:
 - <VirtualHost *:80>
 - ServerName cts4348-node.5.cs.fiu.edu
 - DocumentRoot var/www/html
 - </VirtualHost>
- Do the same for “profweb1”
 - <VirtualHost *:80>
 - ServerName www.profweb1.com
 - DocumentRoot var/www/profweb1
 - DirectoryIndex index.html
 - </VirtualHost>
- Do the same for “profweb2”
 - <VirtualHost *:80>
 - ServerName www.profweb2.com
 - DocumentRoot var/www/profweb2
 - DirectoryIndex index.html
 - </VirtualHost>
- Edit the “host” file located in the “etc” directory. Add the following
 - 127.0.0.1 www.profweb1.com
 - 127.0.0.1 www.profweb2.com
- restart the httpd service. Type:
 - service httpd restart

Task 7: Add the “profe” account to the list of sudoers and allow this account the same privileges as the superuser account.

- Logged in as the root head into the “sudoer” file in the “etc” directory. Type
 - cd /
 - nano etc/sudoers
- Add the following line under “root ALL=(ALL) ALL”:
 - profe ALL=(ALL) ALL
- Also promote the group members as root
 - joel ALL=(ALL) ALL
 - gespi039 ALL=(ALL) ALL
 - kbrow141 ALL=(ALL) ALL

TASK 8: Configure your Apache Web Server to enable requests to /~user/ to serve the user’s public_html directory.

- Open the “httpd.conf” file in etc/httpd
- Look for the “UserDir disable” and comment it out
 - #UserDir disable
- Remove the “#” symbol before “UserDir public_html”
- Head to home/profe directory and create a “public_html” directory

- mkdir home/profe/public_html
- Make an “index.html” file inside the public_html directory
 - cd home/profe/public_html
 - echo Welcome to Profe's Web Page!!! > index.html
- Change the security context in the profe directory. Type
 - cd /
 - chcon -R -h -t httpd_sys_content_t home/profe/public_html
 - chcon -t httpd_sys_content_t home/profe/public_html/index.html
- Restart the httpd service
 - service httpd restart
- Task Test out the user's webpage by typing in a browser's URL bar
 - cts4348-node-5.cd.fiu.edu/~profe

Task 9: Download and Install Webmin

- Download & Install RPM
 - wget <http://prdownloads.sourceforge.net/webadmin/webmin-1.750-1.noarch.rpm>
- Install Perl. Perl is needed for webmin-1.750. Type:
 - yum install perl perl-CPAN perl-Net-SSLeay perl-IO-Socket-SSL
- Once Perl is installed, install the RPM. Type:
 - rpm -U webmin-1.750-1.noarch.rpm
- To save space delete the RPM package located in “/root” after it has been installed
 - cd /
 - cd root
 - rm webmin-1.750-1.noarch.rpm
- Configure the “iptables” file located in “etc/sysconfig” directory in order to open port 10000 for the firewall
 - cd /
 - nano etc/sysconfig/iptables
- In the “iptables” file add the following before the REJECT rule:
 - -A R-Firewall-1-INPUT -m state NEW -m tcp -p tcp --dport 10000 -j ACCEPT
- Restart the iptables service
 - service iptables restart
- Test the webmin by typing the following on your browser:
 - 131.94.134.5:10000

Task 10: Download and Install Usermin

- To download Usermin type:
 - wget <http://prdownloads.sourceforge.net/webadmin/usermin-1.660-1.noarch.rpm>
- Now in Usermin
 - rpm -U usermin-1.660.1.noarch.rpm
- Head to the root and remove the “usermin-1.660-1.noarch.rpm” package
 - cd /
 - rm usermin-1.660.1.noarch.rpm
- Configure the Firewall to open port 20000
- Edit the “iptables” files in the “sysconfig” directory. Type
 - cd /
 - nano /etc/sysconfig/iptables
- Inside the “iptables” add the following:

- -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 20000 -j ACCEPT
- Now restart the iptables service. Type:
 - service iptables restart

Task 11: Add a Webmin user with the same webmin privileges as the root account.

- Open a browser and head into webmin: 131.94.134.5:10000
- Click on Webmin
- Click on Webmin Users
- Click on root
- Click Clone to copy the root account
- Create a password and fill in “real name”
- Click save
- Now you have created a privileged user

Task 12: Download and installed mkpasswd tool.

- To install type: *yum whatprovides */mkpassword*
- Then type: *yum install expect*
- The purpose of using the mkpasswd command is to generate true random passwords by using /dev/urandom.

Task 13: Download the rdesktop client and install

- To download & install rdesktop type:
 - yum -y install rdesktop
- Edit the “iptables” document to allow access to port 3389
 - type in: *sudo vi/etc/sysconfig/iptables*
- And add the following :
 - -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3389 -j ACCEPT
- To test rdesktop type:
 - rdesktop (JCCL Computer IP/Hostname)

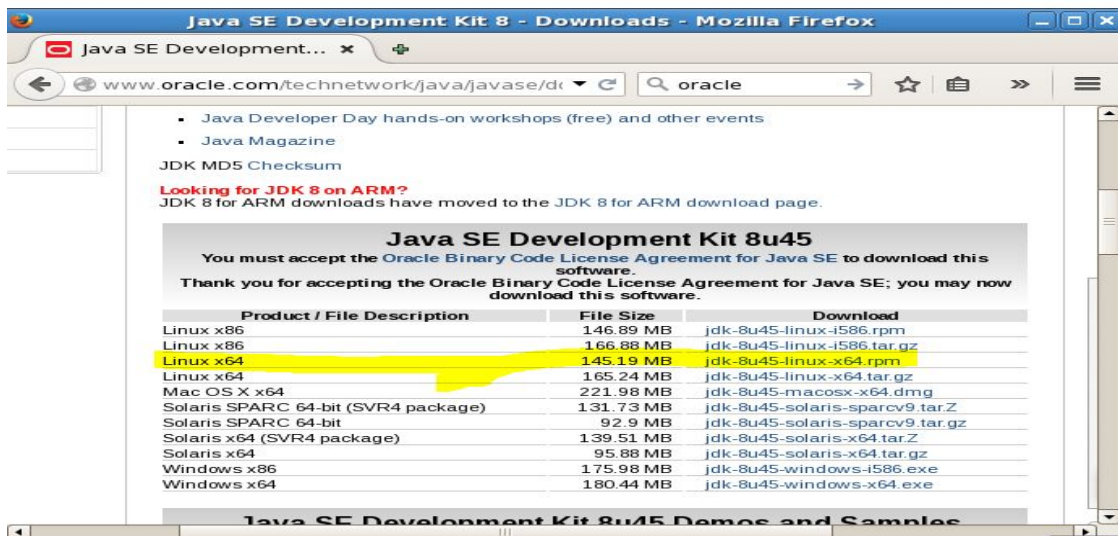
Task 14: Download most current version of xrdp server and install it on your VM.

- To Download & install
 - yum install gcc pam-devel openssl-devel
 - wget <http://ncu.dl.sourceforge.net/sourceforge/xrdp/xrdp-0.4.1.tar.gz>
- Extract the file
 - tar -zxvf xrdp-0.4.1.tar.gz
- Head into the “xrdp-0.4.1” file
 - cd /
 - cd root/xrdp-0.4.1
- Follow the series of commands
 - make
 - sudo make install
 - ln /usr/local/xrdp/xrdp_control.sh /etc/init.d/xrdesktop
 - gconftool-2 -type list -list-type=string -set /desktop/gnome/peripherals/keyboard/kbd/layouts [damnlayers] # map keyboard layout
 - sudo nano /etc/xrdp/sesman.ini

- change 127.0.0.1 to 0.0.0.0
- /sbin/chkconfig --add xrdesktop
- /sbin/service xrdesktop start
- Once that is down open “Remote Desktop Connection” on any windows computer at JCCL
- Type in your server domain and log in with your credentials of roots credentials

Task 15: Download most current version of the Java SE Software Development Kit(JDK) for Linux

- Start by uninstalling any previous versions of Java
- Check to see if you have any earlier versions of Java. Type:
 - rpm -qa | grep -E '^open[jre|jdk]|j[re|dk]'
- If there is a later version of java remove the older version. Type
 - yum remove java-1.6.0-openjdk
- Open your virtual console and log into your VM
- Download the latest version of Java from Oracle



Download and Install right from the Virtual Console

- After Installation
 - Select the correct Java Version folder
 - alternatives --config java
 - Then enter the number for version; In our case 4 was entered.

```
kbrow141@cts4348-node-5:/root
[kbrow141@cts4348-node-5 root]$ alternatives --config java
There are 4 programs which provide 'java'.

  Selection    Command
-----
  1             /usr/lib/jvm/jre-1.4.2-gcj/bin/java
  2             /root/Downloads/jdk1.8.0_45/bin/java
  3             /root/Downloads/jdk1.8.0_45/java
  *+ 4          /usr/java/jdk1.8.0_45/jre/bin/java

Enter to keep the current selection[+], or type selection number: █
```

Verify the Java version that is installed is the right one:

- a. java -version
- b. javac -version

Edit the environmental variables to access the Java directories (As root):

nano /etc/profile

Add the following lines:

```
GNU nano 1.3.12 File: /etc/profile
fi
for i in /etc/profile.d/*.sh ; do
  if [ -r "$i" ]; then
    if [ "${-#*i}" != "$-" ]; then
      . $i
    else
      . $i >/dev/null 2>&1
    fi
  fi
done
unset i
unset pathmunge

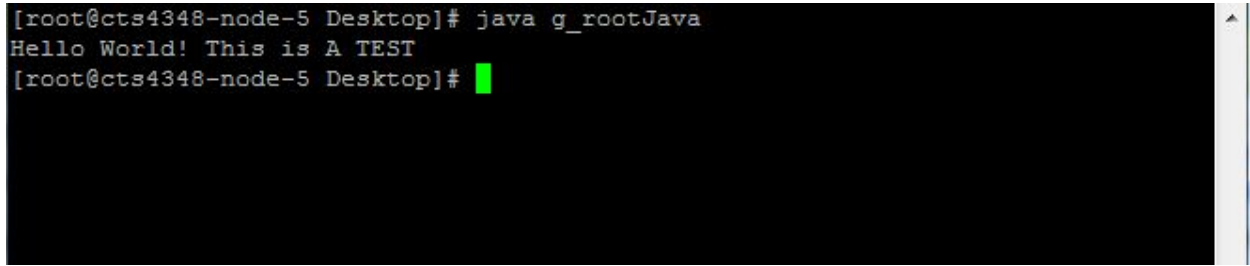
export JAVA_HOME=/usr/bin/java
export JRE_HOME=/usr/java/jre1.8.0_45
export PATH=$PATH:/usr/java/jdk1.8.0_45/bin
█
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^X Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Test that java is Installed for all Users:

```
[kbrow141@cts4348-node-5 root]$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
[kbrow141@cts4348-node-5 root]$ exit
exit
[root@cts4348-node-5 ~]# su joel
[joel@cts4348-node-5 root]$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
[joel@cts4348-node-5 root]$ exit
exit
[root@cts4348-node-5 ~]# su gespi039
[gespi039@cts4348-node-5 root]$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
```


Test of A Sample Java Hello World Code:

```
class g_rootJava{
    public static void main (String[] args){
        System.out.println("Hello World! THIS IS A TEST");
    }
}
```



```
[root@cts4348-node-5 Desktop]# java g_rootJava
Hello World! This is A TEST
[root@cts4348-node-5 Desktop]#
```

Task 16: Download most current version of Apache's Tomcat (Catalina) Web Application Server and install in onto you VM

- Download Tomcat in /usr/share
 - wget
<http://mirror.symnds.com/software/Apache/tomcat/tomcat-8/v8.0.23/bin/apache-tomcat-8.0.23.tar.gz>
- Extract the package
 - tar -xzf apache-tomcat-8.0.23.tar.gz
- Delete the tomcat package that was leftover
 - rm apache-tomcat-8.0.23.tar.gz
- Now create a script for tomcat inside init.d called "tomcat"
 - cd /etc/init.d
 - nano tomcat
- Inside the tomcat file type:
 - !/bin/bash
 - # description: Tomcat Start Stop Restart
 - # processname: tomcat
 - # chkconfig: 234 20 80
 - JAVA_HOME=/usr/java/jdk1.8.0_45
 - export JAVA_HOME
 - PATH=\$JAVA_HOME/bin:\$PATH
 - export PATH
 - CATALINA_HOME=/usr/share/apache-tomcat-8.0.23
 -
 - case \$1 in
 - start)
 - sh \$CATALINA_HOME/bin/startup.sh
 - ;;

- stop)
- sh \$CATALINA_HOME/bin/shutdown.sh
- ;;
- restart)
- sh \$CATALINA_HOME/bin/shutdown.sh
- sh \$CATALINA_HOME/bin/startup.sh
- ;;
- esac
- exit 0
- Configure “tomcat” permissions to 755
 - chmod 755 tomcat
- Add tomcat to start up
 - chkconfig --add tomcat
 - chkconfig --level 2345 tomcat on
- Next test the script by starting up tomcat. Type
 - service tomcat start
- Add port 8080 to the iptables file located in etc/sysconfig
 - cd /etc/sysconfig
 - nano iptables
- Once the file is open type:
 - -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
- Save the file and restart iptables services
 - service iptables restart
- And restart tomcats services
 - service tomcat services
- To verify if tomcat is up and running go to your browser and type:
 - www.yourdomain.com:8080

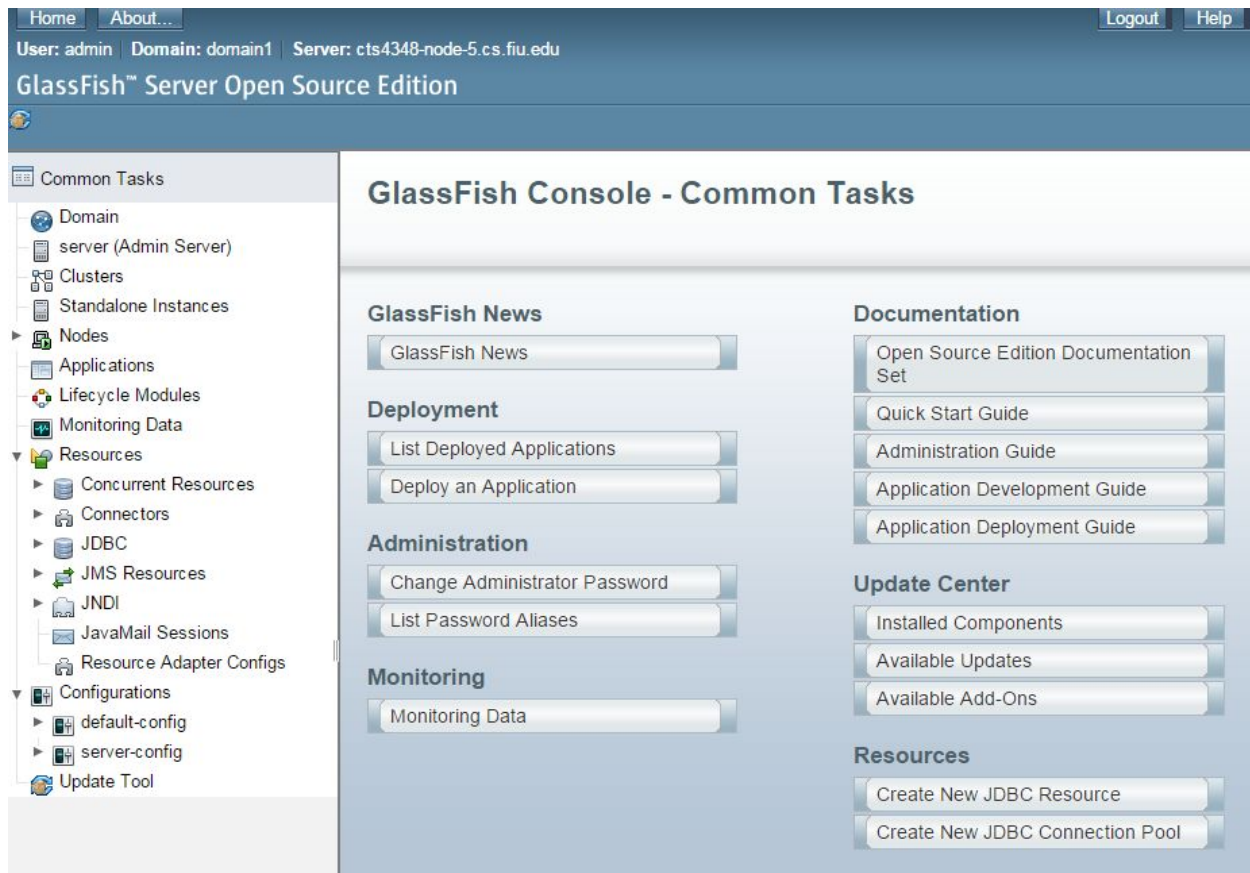
Task 17: Modify your Tomcat Web Application Server to enable the Manager, Server Status and Host Manager Applications

- To enable these applications modify the file “tomcat-users.xml”
 - nano /usr/share/apache-tomcat-8.0.23/conf/tomcat-users.xml
- Once the file is open add the following lines above “</tomcat-users>”:
 - <role rolename="manager-gui"/>
 - <user username="tomcat" password="secret" roles="manager-gui"/>
 - <role rolename="admin-gui"/>
 - <user username="tomcat" password="moarsecret" roles="admin-gui"/>
- Restart tomcat and login
 - service tomcat restart
 - on browser: www.yourdomain.com :8080

Task 18: Download, Install and Configure GlassFish 4 J2EE Server on your VM

- Download glassfish package inside /usr/share
 - wget <http://dlc.sun.com.edgesuite.net/glassfish/4.1/release/glassfish-4.1.zip>
- Extract glassfish
 - unzip glassfish-4.1.zip

- Open ports 4848 and 8888 on iptables in /etc/sysconfig
 - -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8888 -j ACCEPT
 - -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 4848 -j ACCEPT
- Restart iptables
 - service iptables restart
- Change the default port of glassfish from 8080 to 8888
 - nano /usr/share/glassfish4/glassfish/domains/domain1/config/domain.xml
- Start glassfish
 - /usr/share/glassfish4/bin/asadmin start-domain domain1
- Change the admin password in glassfish
 - /usr/share/glassfish4/bin/asadmin change-admin-password
 - Note: the default admin password is empty. Press enter to skip when prompted
- Enable secure admin for glassfish
 - /usr/share/glassfish4/bin/asadmin --host cts4348-node-5.cs.fiu.edu --port 4848 enable-secure-admin
 - Enter credentials when prompted
- Restart glassfish
 - /usr/share/glassfish4/bin/asadmin restart-domain
- Now access glassfish. In a browser URL bar type:
 - cts.4348-node-5.cs.fiu.edu:4848



Task 19: Download and install the sysstat package. Investigate and document in your journals purpose and use of each of the tools bundled in this package.

- Install sysstat
 - yum install sysstat

iostat - reports CPU statistics and input/output statistics for devices, partitions and network filesystems.

mpstat - reports individual or combined processor related statistics.

pidstat - reports statistics for Linux tasks (processes) : I/O, CPU, memory, etc.

sar - collects, reports and saves system activity information (CPU, memory, disks, interrupts, network interfaces, TTY, kernel tables, NFS, sockets etc.)

sadc - is the system activity data collector, used as a backend for sar.

sa1 - collects and stores binary data in the system activity daily data file. It is a front end to sadc designed to be run from cron.

sa2 - writes a summarized daily activity report. It is a front end to sar designed to be run from cron.

sadf - displays data collected by sar in multiple formats (CSV, XML, etc.) This is useful to load performance data into a database, or import them in a spreadsheet to make graphs.

cifsioostat - reports CIFS statistics.

Task 20: Download and install the zsh shell package. Use the yum package manager to install this shell. Make zsh be the default shell for profe.

- Install zsh shell
 - yum install zsh
- Change the profe accounts shell to zsh
 - chsh -s /bin/zsh profe

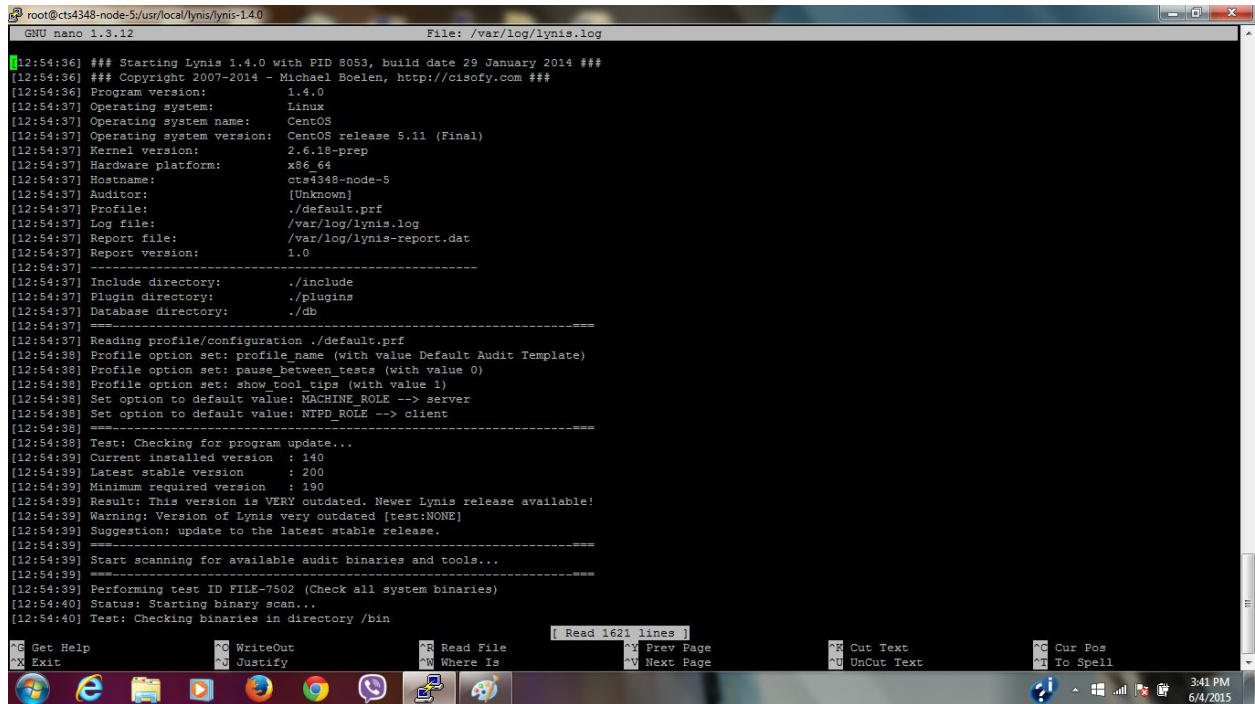
Task 21: Set up on your server a security policy to only allow up to two (2) concurrent login sessions for the profe account and no more than (5) for all other accounts.

- Edit the limits.conf file inside /etc/security
 - nano /etc/security/limits.conf
- Add the following to the file:
 - profe hard maxlogins 2
 - gespi039 hard maxlogins 5
 - joel hard maxlogins 5
 - kbrow141 hard maxlogins 5
- Save and done

Task 22: Install a security auditing tool on your server and produce an assessment audit report

- We will use Lynis as our security auditing tool. Before installation, create its directory with the following command:
- “mkdir /usr/local/lynis”
- Next, download the lynis package using the command:
- “wget <http://cisofy.com/files/lynis-1.4.0.tar.gz>”
- Extract it:
- “tar xzvf lynis-1.4.0.tar.gz”
- Create an assessment report:
- “./lynis -c -Q”
- Report logs can be found in the directory:

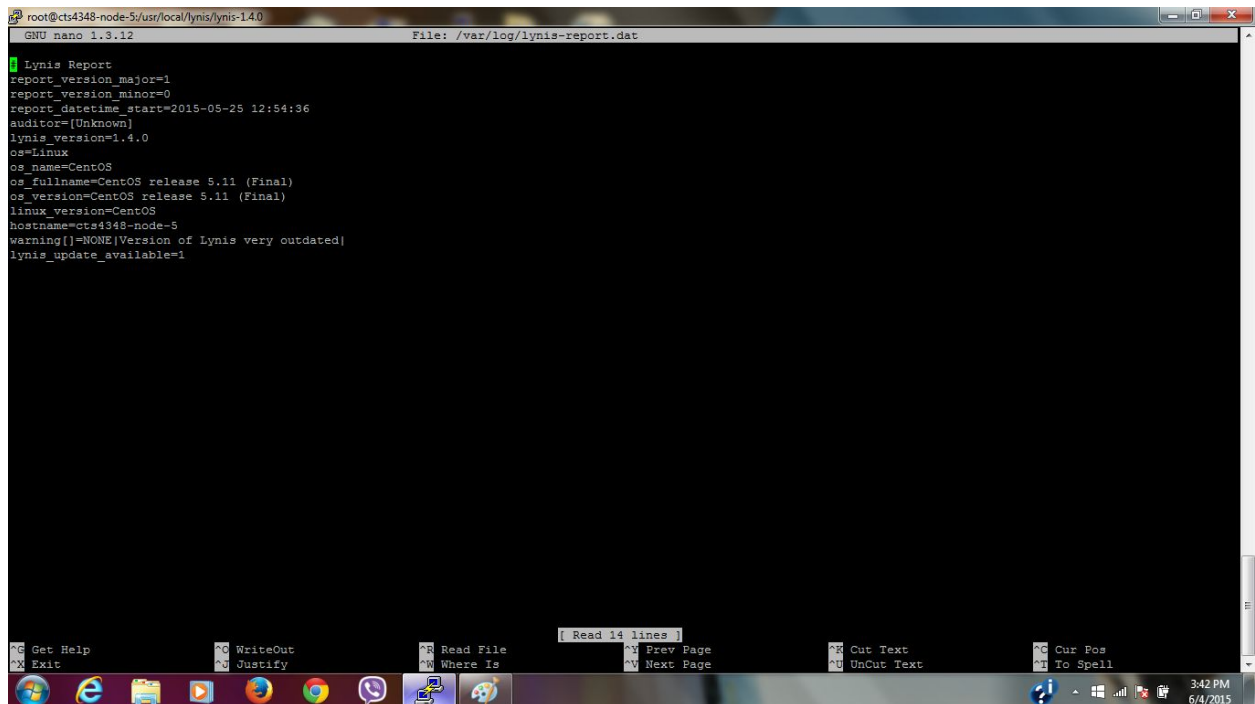
- “cd /var/log/lynis.log”



```
root@cts4348-node-5:/usr/local/lynis/lynis-1.4.0
GNU nano 1.3.12 File: /var/log/lynis.log

[12:54:36] ### Starting Lynis 1.4.0 with PID 8053, build date 29 January 2014 ###
[12:54:36] ### Copyright 2007-2014 - Michael Boelen, http://cisofy.com ###
[12:54:36] Program version: 1.4.0
[12:54:37] Operating system: Linux
[12:54:37] Operating system name: CentOS
[12:54:37] Operating system version: CentOS release 5.11 (Final)
[12:54:37] Kernel version: 2.6.18-prep
[12:54:37] Hardware platform: x86_64
[12:54:37] Hostname: cts4348-node-5
[12:54:37] Auditor: [Unknown]
[12:54:37] Profile: ./default.prf
[12:54:37] Log file: /var/log/lynis.log
[12:54:37] Report file: /var/log/lynis-report.dat
[12:54:37] Report version: 1.0
[12:54:37] -----
[12:54:37] Include directory: ./include
[12:54:37] Plugin directory: ./plugins
[12:54:37] Database directory: ./db
[12:54:37] =====
[12:54:37] Reading profile/configuration ./default.prf
[12:54:38] Profile option set: profile_name (with value Default Audit Template)
[12:54:38] Profile option set: pause_between_tests (with value 0)
[12:54:38] Profile option set: show Tool tips (with value 1)
[12:54:38] Set option to default value: MACHINE ROLE --> server
[12:54:38] Set option to default value: NTPD ROLE --> client
[12:54:38] =====
[12:54:38] Test: Checking for program update...
[12:54:39] Current installed version : 140
[12:54:39] Latest stable version : 200
[12:54:39] Minimum required version : 190
[12:54:39] Result: This version is VERY outdated. Newer Lynis release available!
[12:54:39] Warning: Version of Lynis very outdated [test:NONE]
[12:54:39] Suggestion: update to the latest stable release.
[12:54:39] =====
[12:54:39] Start scanning for available audit binaries and tools...
[12:54:39] =====
[12:54:39] Performing test ID FILE-7502 (Check all system binaries)
[12:54:40] Status: Starting binary scan...
[12:54:40] Test: Checking binaries in directory /bin
```

- Report files can also be found in the /var/log/ directory:



```
root@cts4348-node-5:/usr/local/lynis/lynis-1.4.0
GNU nano 1.3.12 File: /var/log/lynis-report.dat

Lynis Report
report_version_major=1
report_version_minor=0
report_datetime_start=2015-05-25 12:54:36
auditor=[Unknown]
lynis_version=1.4.0
os=Linux
os_name=CentOS
os_fullname=CentOS release 5.11 (Final)
os_version=CentOS release 5.11 (Final)
linux_version=CentOS
hostname=cts4348-node-5
warning([NONE]Version of Lynis very outdated)
lynis_update_available=1
```

Task 23: Create a custom kernel from source code.

- install and configure the kernel source
 - compile and install you own kernel
1. Test whether gcc or g++ are already installed with the following commands:
 - a. gcc -v
 - b. g++ -v
 - c. if no information is displayed, proceed with installation
 2. Use the following commands for installation:
 - a. yum install gcc (Our server had gcc already installed)
 - b. yum install gcc-c++
 3. Install the latest kernel-src.rpm file from a Fedora update mirror site
Type:
 - a. In cd /root
`wget http://vault.centos.org/5.11/updates/SRPMS/kernel-2.6.18-406.el5.src.rpm`
 - b. yum install unifdef
 - c. yum install rpm-build
 - d. yum install redhat-rpm-config
 4. Install the kernel-2.6.*.src.rpm package into /usr/src/ directory
Make sure the directory path to /usr/src/redhat/SPECS exists. If it doesn't, create it manually.
Directory path already exist on our server)
 - a. mkdir /usr/src/redhat/
 - b. mkdir /usr/src/redhat/SPECS
 - c. rpm -Uvh kernel-2.6.18-406.el5.[src.rpm](#) (In /usr/src) directory
 5. Change directory to /usr/src/redhat/SPECS/ (verify this directory exists, if it does not, manually create), and issue the following command:
`rpmbuild -bp --target=x86_64 kernel.spec`
 6. Kernel tree will be located in /usr/src/redhat/BUILD/ (verify that this directory exists). Link to it from /usr/src with the following command:

`ln -s /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18-406.el5.x86_64 /usr/src/linux`
 7. Copy your currently running kernel's configurations to the new kernel source directory. In /boot, look for a file whose name starts with 'config-2.6.18....' and copy it to /usr/src/linux/.config

For example: `cp /boot/config-2.6.18-406.el5 /usr/src/linux/.config`
 8. In /usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18-406.el5.x86_64/linux-2.6.18-406.el5.x86_64
Run the following:
Compile new kernel and its modules:
 - make bzImage
 - make modules

make

Install new kernel and modules:

make modules_install
make install

9. Then double-check the bootloader configuration and add the new kernel if necessary.

nano /boot/grub/menu.lst

Set "default" and "timeout" parameters as follows:

default=1

timeout=60

Comment out the line attribute "hiddenmenu"

10. Reboot your system. At the shell prompt (must be root), type "reboot". When the system restarts, it should now display a menu (on the VMWare Server Console UI) with at least two kernel options to start from. Arrow down to the new custom kernel, and press Enter.

Task 24:

Download and compile lmbench benchmarking suite, compare default vs. self built kernel performance

1. Download the 'lmbench' benchmarking utility and extract it:

- a. wget <http://www.cs.fiu.edu/~osorioj/lmbench-3.0-a8.tgz>
- b. tar -zxvf lmbench-3.0-a8.tgz

2. Navigate to the following folder and compile the benchmark application:

- a. cd lmbench-3.0-a8
- b. make

3. Follow the directions in the README file:

cat README

4. Run the benchmark tool:

- a. cd src
- b. make results

5. Use the proposed defaults with exception of the 'Mail results' question to answer no instead of the default yes

6. Reboot the server and choose another kernel version and run the test again:

make rerun

7. Compare the output of the two results and enter it into the log file with a recommendation as to which the server should use:

make see

8. The comparison report is stored in the following directory:

/root/lmbench-3.0-a8/results/summary.out

```
root@cts4348-node-5:~/lmbench-3.0-a8/results
GNU nano 1.3.12

make[1]: Entering directory `/root/lmbench-3.0-a8/results'

      L M B E N C H  3 . 0  S U M M A R Y
      -----
      (Alpha software, do not distribute)

Basic system parameters
-----
Host          OS Description          Mhz  tlb  cache  mem  scal
           pages  line  bytes  par  load

-----
cts4348-n Linux 2.6.18-          x86_64-linux-gnu 1585          1
cts4348-n Linux 2.6.18-          x86_64-linux-gnu 1582          1
cts4348-n Linux 2.6.18-          x86_64-linux-gnu 1582          1

Processor, Processes - times in microseconds - smaller is better
-----
Host          OS  Mhz  null  null  open  slct  sig  sig  fork  exec  sh
           call I/O stat clos TCP  inst hndl proc proc proc
-----
cts4348-n Linux 2.6.18- 1585 0.39 0.59 2.68 3.87 6.63 0.52 1.97 14.K 22.K 37.K
cts4348-n Linux 2.6.18- 1582 0.39 0.58 2.60 3.85 6.66 0.51 2.03 14.K 21.K 38.K
cts4348-n Linux 2.6.18- 1582 0.39 0.58 2.58 3.93 6.32 0.52 1.77 4387 6807 12.K

Basic integer operations - times in nanoseconds - smaller is better
-----
Host          OS  intgr  intgr  intgr  intgr  intgr
           bit  add  mul  div  mod
-----
cts4348-n Linux 2.6.18- 0.6300 0.3200 0.2200 15.2 13.4
cts4348-n Linux 2.6.18- 0.6400 0.3200 0.2200 15.2 14.6
cts4348-n Linux 2.6.18- 0.6400 0.3200 0.2200 15.4 14.7

Basic uint64 operations - times in nanoseconds - smaller is better
-----
Host          OS  int64  int64  int64  int64  int64
           bit  add  mul  div  mod
-----
cts4348-n Linux 2.6.18-          0.2900
cts4348-n Linux 2.6.18-          0.3200
cts4348-n Linux 2.6.18-          0.3200

Basic float operations - times in nanoseconds - smaller is better
-----
Host          OS  float  float  float  float
           add  mul  div  bogo
-----
cts4348-n Linux 2.6.18- 1.9100 2.5500 9.4300 8.8800
cts4348-n Linux 2.6.18- 1.9000 2.5500 9.4200 8.8900
cts4348-n Linux 2.6.18- 1.9200 2.5900 9.5100 8.2200

Basic double operations - times in nanoseconds - smaller is better
-----
Host          OS  double  double  double  double
           add  mul  div  bogo
-----
cts4348-n Linux 2.6.18- 1.9000 3.1900 14.5 13.9
cts4348-n Linux 2.6.18- 1.9100 3.1800 14.5 13.9

^G Get Help          ^O WriteOut
^X Exit              ^J Justify
```

Task 25:

1. Download, install, setup and configure DhryStone Benchmark tool.
wget <http://users.cis.fiu.edu/~osorioj/dhrystone-2.1-1.src.rpm>


```
rpm -iv dhystone-2.1-1.src.rpm
```

2. Navigate to the following directory:

```
cd /usr/src/redhat/SPECS
```

3. Edit the following directory:

- a. nano dhystone-2.1.spec
- b. remove the line:
 - i. Copyright: distributable
- c. Add the line:
 - i. License: GPLv2

- 3) Create 2 rpm packages from dhystone-2.1.spec with commands:

```
rpmbuild -ba dhystone-2.1.spec
```

```
rpmbuild -bp dhystone-2.1.spec
```

4. Go to the Directory and execute dry2 and dry2reg:

- a. cd /usr/src/redhat/BUILD/dhry2.1
- b. make

5. Copy dry2 and dry2reg to /var/www/cgi-bin
cp dry2 dry2reg /var/www/cgi-bin

6. Go to the following Directory:

```
cd /var/www/cgi-bin
```

7. Create the following script:

```
nano dhry.cgi
```

8. Add the script to cgi file:

```
GNU nano 1.3.12      File: dhry.cgi      Modified
#!/usr/bin/perl
use CGI qw/:standard/;
$times = param('times');
print header, start_html("Result");
print(h1("Result"));
die("could not execute benchmark")
# unless (open(LS, "./dry2 $times | tail -3 |"));
unless (open(LS, "echo $times |./dry2 |"));
@lsout = <LS>;
print "<PRE>\n";
foreach $name (@lsout) {
chomp($name);
print $name, "<br>\n";
}
print "</PRE>\n";
print end_html;
```

^G Get Help **^O** WriteOut **^R** Read File **^Y** Prev Page **^K** Cut Text **^C** Cur Pos
^X Exit **^J** Justify **^W** Where Is **^V** Next Page **^U** UnCut Text **^T** To Spell

9. Press “Ctrl-X” to exit, then “Y” to save
10. Change the permissions:

chmod og+x dhry.cgi
11. Navigate to /var/www/html and create a file named dhry.html:
 - a. cd /var/www/html
 - b. nano dhry.html
 - c. Add the following html tags:

```
GNU nano 1.3.12      File: dhry.html      Modified
<h3>Benchmark example</h3>
<FORM action=/cgi-bin/dhry.cgi method=get>
Run benchmark
<input type=text name=times size=10> times
<br>
<input type=submit>
</form>

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

12. Edit the file httpd.conf:
 - a. nano /etc/httpd/conf/httpd.conf
 - b. Uncomment the following line
 - i. AddHandler cgi-script.cgi
13. Restart httpd:

service httpd restart
14. Test the script by going to the following link:

<http://131.94.134.5/dhry.html>

Task 26:

1. Download, install, test-run and document use of the following commands: atop, htop, iotop, glances
 - a. Install atop:
wget http://www.atoptool.nl/download/atop-2.1-1.x86_64.rpm
rpm -ivh atop-2.1-1.x86_64.rpm

or yum install atop

2. Run atop:
atop

Atop is an ASCII full-screen performance monitor which can log and report the activity of all server processes. One feature I really like is that atop will stay active in the background for long-term server analysis (up to 28 days by default). Other advantages include:

- Shows resource usage of ALL processes, even those that are closed/completed.
- Monitors threads within processes & ignores processes that are unused.
- Accumulates resource usage for all processes and users with the same name.
- Highlights critical resources using colors (red).
- Will add or remove columns as the size of the display window changes.
- Includes disk I/O and network utilization.

ATOP - cts4348-node-5 2015/05/27 22:20:31 ----- 1h16m22s elapsed														^	
PRC		sys	26.65s		user	25.35s		#proc	112		#zombie	0		#exit	0
CPU		sys	3%		user	3%		irq	0%		idle	94%		wait	0%
CPL		avg1	0.10		avg5	0.06		avg15	0.01		csw	486740		intr	4634629
MEM		tot	497.8M		free	29.0M		cache	161.7M		buff	20.9M		slab	29.2M
SWP		tot	1.0G		free	1.0G					vmcom	370.7M		vmlim	1.2G
PAG		scan	72765		steal	67577		stall	1		swin	0		swout	0
LVM		p00-LogVol100	busy		0%			read	31272		write	24311		avio	0.38 ms
LVM		p00-LogVol101	busy		0%			read	156		write	0		avio	0.07 ms
DSK		sda	busy		0%			read	28665		write	7612		avio	0.64 ms
DSK		hdc	busy		0%			read	10		write	0		avio	3.80 ms
NET		transport			tcpi	4486		tcpo	4118		udpi	146		udpo	146
NET		network			ipi	4671		ipo	4306		ipfrw	0		deliv	4642
NET		eth0	0%		pcki	5002		pcko	2028		si	1 Kbps		so	0 Kbps
NET		lo	----		pcki	2359		pcko	2359		si	6 Kbps		so	6 Kbps
*** system and process activity since boot ***															
PID	SYSCPU	USRCPU	VGROW	RGROW	RDDSK	WRDSK	ST	EXC	S	CPU	CMD	1/16			
1889	5.28s	14.50s	1.6G	91216K	41456K	1724K	N-	-	S	0%	java				
2513	0.22s	6.36s	99984K	5600K	2260K	48K	N-	-	S	0%	Xorg				
2015	4.60s	0.01s	10256K	700K	OK	OK	N-	-	S	0%	hald-addon-sto				
1991	2.03s	0.87s	31348K	4288K	144K	4K	N-	-	S	0%	hald				
5	2.39s	0.00s	OK	OK	OK	OK	N-	-	S	0%	events/0				
421	1.92s	0.24s	14100K	2336K	12115K	OK	N-	-	S	0%	udev				
3234	0.91s	0.35s	102.3M	16304K	OK	OK	N-	-	S	0%	sshd				

3. Install htop:
yum install htop

4. Run htop:
htop

```

root@cts4348-node-5:~
CPU[|||||100.0%] Tasks: 113, 30 thr; 2 running
Mem[|||||310/497M] Load average: 0.27 0.08
Swp[|||||] Uptime: 01:19:24

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3778 root 19 0 112M 43260 1700 S 37.6 8.5 0:05.08 /usr/libexec/webm
3806 root 15 0 68520 1616 1156 R 0.4 0.3 0:00.08 htop
2323 root 18 0 1620M 91856 13460 S 0.4 18.0 0:02.04 /usr/java/jdk1.8.
2325 root 15 0 1620M 91856 13460 S 0.4 18.0 0:00.03 /usr/java/jdk1.8.
1889 root 25 0 1620M 91856 13460 S 0.4 18.0 0:20.35 /usr/java/jdk1.8.
3484 root 15 0 102M 16312 3072 S 0.4 3.2 0:01.09 sshd: root@pts/2
1 root 15 0 10372 688 572 S 0.0 0.1 0:00.92 init [5]
2 root RT -5 0 0 0 S 0.0 0.0 0:00.00
3 root 34 19 0 0 S 0.0 0.0 0:00.00
4 root RT -5 0 0 0 S 0.0 0.0 0:00.00
5 root 10 -5 0 0 S 0.0 0.0 0:00.00
6 root 11 -5 0 0 S 0.0 0.0 0:00.01
11 root 11 -5 0 0 S 0.0 0.0 0:00.00
15 root 10 -5 0 0 S 0.0 0.0 0:00.13
16 root 20 -5 0 0 S 0.0 0.0 0:00.00
72 root 20 -5 0 0 S 0.0 0.0 0:00.00
75 root 20 -5 0 0 S 0.0 0.0 0:00.00
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice +F9Kill F10Quit

```

5. Install iotop:

yum install iotop

6. Run iotop:

iotop

```

root@cts4348-node-5:~
Total DISK READ: 0.00 B/s | Total DISK WRITE: 0.00 B/s
TID PRIO USER DISK READ DISK WRITE SWAPIN IO> COMMAND
2110 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.19 % java -Dja-trap start
2 rt/3 root 0.00 B/s 0.00 B/s 0.00 % 0.10 % [migration/0]
2186 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
3 be/7 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % [ksoftirqd/0]
1695 be/3 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % auditd
4 rt/3 root 0.00 B/s 0.00 B/s 0.10 % 0.00 % [watchdog/0]
11 be/3 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % [kthread]
15 be/3 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % [kblockd/0]
2187 be/4 root 0.00 B/s 0.00 B/s 0.19 % 0.00 % java -Dja-trap start
1694 be/3 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % auditd
2784 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
3458 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % httpd
2325 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
2338 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
2340 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
329 be/3 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % [ata/0]
3461 be/4 apache 0.00 B/s 0.00 B/s 0.00 % 0.00 % httpd
1992 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % hald-runner
3662 be/0 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % atop -a --150527 600
2022 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % java -Dja-trap start
1536 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % mcstransd
1 init [5]

```

The iotop command is top like utility for disk I/O. It watches I/O usage information output by the Linux kernel (requires v2.6.20 or later) and displays a table of current I/O usage by processes or threads on the system. This post explains how to install and use iotop to find out what's stressing (or program names) on your hard drives under Linux operating systems.

Task 27: Install mySQL, php and WordPress on your server

- Install mySQL onto the server
 - yum install mysql
 - yum install mysql-server
- Install latest php version onto the server


```
# rpm -Uvh http://mirror.webtatic.com/yum/el5/latest.rpm
```

 - yum install php54w.x86_64 php54w-cli.x86_64 php54w-common.x86_64 php54w-gd.x86_64 php54w-ldap.x86_64 php54w-mbstring.x86_64 php54w-mcrypt.x86_64 php54w-mysql.x86_64 php54w-pdo.x86_64 php54w-stall php
 - yum install php-mysql
- Edit the httpd.conf and restart service


```
edit nano /etc/httpd/conf/httpd.conf
```

Add lines:

```
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

- service httpd restart
- Start the mySQL service
 - service mysqld start
- Head into the html directory
 - cd /var/www/html
- Download WordPress package
- wget --no-check-certificate <http://wordpress.org/latest.tar.gz>
- Extract the tar WordPress tar file
 - tar zxvf wordpress-4.0.tar.gz
- Enter mySQL to create a database and when prompted for password press enter and leave it blank.
 - mysql -u root -p
- Once logged in type:
 - CREATE DATABASE wordpress; GRANT ALL PRIVILEGES ON wordpress.* TO "localhost" IDENTIFIED BY "password"; FLUSH PRIVILEGES; EXIT;
- Head into the html directory
 - cd /var/www/html/wordpress
- Make a backup of the wp-config.php file and rename it to wp-config-sample.php
 - cat wp-config-sample.php > wp-config.php
- Now edit the original wp-config.php file
 - nano wp-config.php
- Type the following inside the file
 - define('DB_NAME', 'wordpress'); // The name of the database
 - define('DB_USER', 'root'); // Your MySQL username
 - define('DB_PASSWORD', 'yourpasswordhere'); // ...and password
 - define('DB_HOST', '127.0.0.1'); // 99% chance you won't need to change this value
 - define('DB_CHARSET', 'utf8');
 - define('DB_COLLATE', '');
 - define('SECRET_KEY', 'put your unique phrase here');

<http://131.94.134.5/wordpress/wp-admin/install.php>

Enter the following:

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	g_root
Username	root
Password, twice	root's Password
Your E-mail	gespi039@fiu.edu .
Privacy	Uncheck box: Allow search engines to index this site. http://131.94.134.5/wordpress/wp-login.php

Task 28: Download, Extract, Install and Configure Mantis Bug Tracker Web Application on your VM.

1. Download MantisBT tarball to your server:
 - a. `wget http://sourceforge.net/projects/mantisbt/files/mantis-stable/1.2.19/mantisbt-1.2.19.tar.gz/download`
 - b. `tar -xzf mantisbt-1.2.19.tar.gz`
2. Copy mantisBT to the html folder and change the permissions:
 - a. `cp -r mantisbt-1.2.19 /var/www/html`
 - b. `cd /var/www/html`
 - c. `chown -R apache:apache mantisbt-1.2.19`
3. Install Mantis BugTracker in a web browser:
 - a. go to <http://131.94.134.5/mantisbt-1.2.19/admin/install.php>
 - b. use the following options:
 - i. Type of Database: MySQL (default)
 - ii. Hostname (for Database Server): localhost
 - iii. Username (for Database): root
 - iv. Password (for Database): root's password
 - v. Database name (for Database): bugtracker
 - vi. Admin Username (to create Database if required): root
 - vii. Admin Password (to create Database if required): root's password
 - viii. Print SQL Queries instead of Writing to the Database: Uncheck box
 - ix. hit the Install/Upgrade Database button
4. IN command:
 1. Set the root user password for all local domains
 2. **SET PASSWORD FOR 'root'@'localhost' = PASSWORD('root's password');**
 3. **SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('root's password');**
 4. Exit MySQL
 5. **exit**
 6. **service mysqld restart**



Login	
Username	root
Password
Remember my login in this browser	<input type="checkbox"/>
Secure Session	<input checked="" type="checkbox"/> Only allow your session to be used from this IP address.
<input type="button" value="Login"/>	

- 5) Scroll down and in the sentence click the create word:
Please log in as the administrator and [create](#) your first project.
and log in using:
Username: administrator
Password: root
- 6) Back in the terminal, move admin folder to admin-backup:
`cd /var/www/html/mantisbt-1.2.19`
`mv admin/ admin-security`
- 7) Back in mantis page, create a new project, usernames and roles:
- Enter project name: New Project
 - leave the rest as default and click on “Add Project”
 - Then click on the tab manage
 - then on manage users
 - then on the button create a new account and create(give different access level to each user):
 - root
 - Giancarlo
 - Kevaghn
 - Joel

Goto: http://131.94.134.5/mantisbt-1.2.19/login_page.php

Username: administrator

Password: root

Task 29: Add a second (virtual) Hard Disk to your VM of size 4 Gig. Then expand the size of your primary logical volume with the newly available disk space. Please follow the instructions provided in class.

- Start by shutting down your machine via the virtual console
- Continuing on the virtual console, add a 4GB hard disk with the following:
 - Right click your virtual machine and click settings
 - Click add
 - Click next, select hard disk and click next again
 - Select Create a new virtual disk, click next and select SCSI
 - Set the storage space to 4GB and uncheck allocate all disk space now and split disk into 2GB files
 - Click next and then finish
- Start the Virtual Machine and head into the Terminal. Type the following:
 - `pvccreate /dev/sdb`
 - `vgextend VolGroup00 /dev/sdb`
 - `lvextend -L +4060M /dev/VolGroup00/LogVol00`
 - `resize2fs /dev/VolGroup00/LogVol00`

TASK 30: Add a third (virtual) Hard Disk to your VM of size 4 Gig. Partition this disk in two equal size partitions.

- Start by shutting down your machine via the virtual console
- Continuing on the virtual console, add a 4GB hard disk with the following:
 - Right click your virtual machine and click settings
 - Click add
 - Click next, select hard disk and click next again
 - Select Create a new virtual disk, click next and select SCSI
 - Set the storage space to 4GB and uncheck allocate all disk space now and split disk into 2GB files

- Click next and then finish
- Create the partitions of 2GB
 - fdisk /dev/sdc
 - type: n
 - type:p
 - type:l
 - type: +2012M
 - type:p
 - type:n
 - type:p
 - leave as default and press enter
 - leave as default and press enter
 - type:p
 - type:w
- Edit the yum Repo in order to download CentOSplus
 - nano /etc/yum.repos.d/CentOs-Base.repo
- In the Centosplus section change the value for enabled to “enabled=1”
- Add “includepngs=kernel*” after the previous step
- Save the file
- Update Yum
 - yum update
- Reboot and use the centosplus kernel
- Install kmod-xfs, xfsprogs, xfsdump
 - yum install kmod-xfs
 - yum install xfsprogs
 - yum install xfsdump
- Make a xfs file system for the first partition
 - mkfs -t xfs /dev/sdc1
- Make a ext3 file system for the second partition
 - mkfs.ext3 -L /ext3part /dev/sdc2
- Create mount directories for both partitions
 - mkdir /mnt/sdc1
 - mkdir /mnt/sdc2
- Now mount the partitions
 - mount /dev/sdc1 /mnt/sdc1
 - mount /dev/sdc2 /mnt/sdc2
- Edit the fstab file
 - nano /etc/fstab
- Enter the following lines
 - /dev/sdc1 /mnt/sdc1 xfs defaults 0 0
 - /dev/sdc2 /mnt/sdc2 ext3 defaults 0 0

TASK 31: Enable user quotas for the file system that you created in the second partition of your third disk.

- 1) Edit the /etc/fstab and include the line:

```
/dev/sdc2                      /mnt/sdc2              ext3   defaults,usrquota 1 1
```

ensure it is close to the top in the file. You could also use webmin as a gui to monitor and change quota restrictions.

2) Create the file to hold quota info with:

```
touch /mnt/sdc2/aquota.user
```

3) Set permissions with command:

```
chmod 600 /mnt/sdc2/aquota.user
```

4) Mount and remount file system with command:

```
mount -o remount /dev/sdc2
```

5) Check partition with command:

```
quotacheck -vgu /dev/sdc2
```

6) Run restore with command:

```
restorecon /mnt/sdc2
```

7) Run quotaon with command

```
quotaon -av
```

8) Set quota limits for user with command:

```
edquota username
```

- Run the command for each user

9) To check quota run command:

```
quota -u username
```

10) Navigate to sdc2 directory and create directories for each user:

```
cd /mnt/sdc2/
```

```
mkdir home
```

```
cd home
```

```
mkdir gespi039 kbrow141 joel profe
```

11) Give permissions to each user with:

```
chown gespi039/mnt/sdc2/home/gespi039
```

```
chgrp gespi039/mnt/sdc2/home/gespi039
```

```
chmod u=rwx /mnt/sdc2/home/gespi039
```

```
chown kbrow141/mnt/sdc2/home/kbrow141
```

```
chgrp kbrow141/mnt/sdc2/home/kbrow141
```

```
chmod u=rwx /mnt/sdc2/home/kbrow141
```

```
chown joel/mnt/sdc2/home/joel
```

```
chgrp joel/mnt/sdc2/home/joel
```

```
chmod u=rwx /mnt/sdc2/home/joel
```

```
chown profe/mnt/sdc2/home/profe
```

```
chgrp profe/mnt/sdc2/home/profe
```

```
chmod u=rwx /mnt/sdc2/home/profe
```

12) Log in as each user and copy files to users directories with command:

```
cp -r /usr/share/doc /mnt/sdc2/home/user
```

13) Get quote report with command:

```
repquota -a
```

```
[root@cts4348-node-5 ~]# repquota -a
*** Report for user quotas on device /dev/sdc2
Block grace time: 7days; Inode grace time: 7days
```

User		used	Block limits			used	File limits		
			soft	hard	grace		soft	hard	grace
root	--	68884	0	0		5	0	0	
gespi039	--	205424	0	0		9155	0	0	
kbrow141	--	205424	0	0		9155	0	0	
profe	--	205424	0	0		9155	0	0	
joel	--	205424	0	0		9155	0	0	

```
[root@cts4348-node-5 ~]#
```

Step 32: Relocate MySQL Data files to your XFS or JFS Filesystem

1) Create mysql directory inside sdc1:

```
mkdir /mnt/sdc1/mysql
```

2) Stop MySQL and copy over the data using rsync and assign correct ownership to the directory:

```
service mysqld stop
```

```
rsync -avz /var/lib/mysql/ /mnt/sdc1/mysql
```

```
chown mysql:mysql /mnt/sdc1/mysql
```

3) Open /etc/my.cnf and change or add following line under [mysqld] directive:

```
datadir = /mnt/sdc1/mysql
```

4) Set SELINUX permissions on new location:

```
o semanage fcontext -a -t mysqld_db_t "/mnt/(.*)?"
```

```
o chcon -R system_u:object_r:mysqld_db_t /mnt
```

```
o restorecon -Rv /mnt
```

```
o semanage fcontext -a -t mysqld_db_t "/mnt/sdc1/(.*)?"
```

```
o restorecon -Rv /mnt/sdc1
```

```
o chcon -R system_u:object_r:mysqld_db_t /mnt/sdc1
```

```
o semanage fcontext -a -t mysqld_db_t "/mnt/sdc1/mysql/(.*)?"
```

```
o restorecon -Rv /mnt/sdc1/mysql
```

```
o chcon -R system_u:object_r:mysqld_db_t /mnt/sdc1/mysql
```

5) Start MySQL service:

service mysqld start

6) Verify datadir variable in MySQL is correctly set by executing following command in mysql shell:

mysql -u root -p

- press enter for password then:

mysql> SHOW VARIABLES LIKE 'datadir';

You should get:

```
root@cts4348-node-5:~  
Starting mysqld: [ OK ]  
[root@cts4348-node-5 ~]# mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 5.0.95 Source distribution  
  
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> SHOW VARIABLES LIKE 'datadir';  
+-----+  
| Variable_name | Value                |  
+-----+  
| datadir       | /mnt/sdc1/mysql/    |  
+-----+  
1 row in set (0.01 sec)  
  
mysql>
```

TASK 33: Download, install and properly set up rsyslog in your server to store syslog messages in MySQL Database.

- Start by installing rsyslog
 - yum install rsyslog rsyslog-mysql
- Stop syslog and turn off the auto start
 - service syslog stop
 - chkconfig syslog off
- Start up rsyslog and turn on auto start
 - service rsyslog start
 - chkconfig rsyslog on
- Create the user and rsyslog database
- Head into the mySQL interface
 - mysql -u root -p
- Enter the password
- Create the database with in the syslog-mysql directory
 - mysql -u root -p < /usr/share/doc/rsyslog-mysql-3.22.1/createDB.sql
 - mysql -u root -p
 - GRANT ALL ON Syslog.* TO 'root'@'localhost' IDENTIFIED BY 'PASSWORD';
 - FLUSH PRIVILEGES;
 - EXIT;

- Now edit the rsyslog.conf file
 - nano /etc/rsyslog.conf
- Add the following:
 - \$ModLoad ommysql
- Replace /var/log/messages with the following
 - :ommysql:localhost,Syslog,root,PASSWORD
- Restart rsyslog service
 - service rsyslog restart

NOTE: To check your if your syslog DB is working, manually log an entry and check the DB via mysql

- logger -p local0.info "hello world!"
- Enter mysql and use "SELECT Facility, Priority, Message from SystemEvents;" to check the log
- Download loganalyzer
 - wget http://download.adiscon.com/loganalyzer/loganalyzer-3.6.6.tar.gz
- Extract the package
 - tar xzf loganalyzer-3.6.6.tar.gz
- Make a directory and store the contents of the extracted file inside the created directory
 - mkdir /var/www/html/loganalyzer
 - cp -R loganalyzer-3.6.6/src/ /var/www/html/loganalyzer/
 - cp loganalyzer-3.6.6/contrib/* /var/www/html/loganalyzer
- Change the permissions in the loganalyzer directory
 - cd /var/www/html/loganalyzer
 - chmod 700 *sh
 - ./configure.sh
 - semanage fcontext -a -t httpd_sys_content_t "/var/www/html(/.*)?"
 - restorecon -Rv /var/www/html
 - chcon -R system_u:object_r:httpd_sys_content_t /var/www/html
- Open a browser and type cts4348-node-5.cs.fiu.edu/loganalyzer/install.php
 - Click Next
 - Click Next
 - Enable user database
 - Enter database name: Syslog
 - Enter database user: root
 - Enter password: PASSWORD
 - And enable credentials
 - Click Next until you hit create user page
 - Enter a username and password (LAuser: PASSWORD)
 - Click Next
 - Change source type to MySQL native
 - Change the DB name to Syslog
 - Enter username and password
 - Enable row counting
 - Click Next
 - Click Finish
- Login LAuser:PASSWORD

Task 34:

1. Add a crontab entry (system level) to run yum update every other day at 1 am to update your server.
2. **#cd /etc**
3. **#crontab -e**
4. Input the line: **0 1 */2 * * root run-parts /usr/bin/yum -y update**
5. Save and quit from the text editor
6. Schedule a job (as root) to run every six hours to trace how long the system has been running. Investigate a Linux built-in command suited for this task. Redirect output to a log file to be inspected during the server eval (for example: command >> /var/log/running.log
7. Enter: **#crontab -e**
8. Enter the line: **0 */6 * * * root uptime >> /var/log/running.log**
9. Save and quit from text editor.
10. Schedule a job for the “profe” account to run every 2 1/2 hours, on weekends and on Thursdays, to email a list of the currently logon users and list of actively running processes.
11. Enter: **#su profe**
12. Enter: **#crontab -e**
13. Enter: **0 0,5,10,15,20 * * 0,4,6 who | mail -s “Logged on user” profe; ps aux | mail -s “Running Processes” profe**
14. On the next line enter: **30 2,7,12,17,22 * * 0,4,6 who | mail -s “Logged on user” profe; ps aux | mail -s “Running Processes” profe**
15. Save and quit from the text editor.
16. Verify correct execution of these schedules and output produced by the commands/programs being executed. Disable the above cron jobs after a week or two.
17. Enter: **#crontab -l**
18. Enter: **#cat /var/log/yum.log**
19. Enter: **#crontab -u profe -r**

Task 35:

- URI of JCCL printer is: “lpd://papermill.cis.fiu.edu/venom”
 - Printer Model: HP LaserJet 8000 Postscript
 - Print a 2-page document on this printer in duplex mode. Enable option to print a banner on the front of the job and bring this output the day of the server evaluations for credit.
 - Allow public access to the CUPS administration portal. Enable access to administrative functions in this portal only after root account has been authenticated.
1. Install CUPS with the command: **#yum install cups**
 2. Within the VM desktop open the browser and place within the address bar: <http://127.0.0.1:631>
 3. Within the page find and click the **add printer button**.
 4. Input the following information:

Name: Venom

Location: JCCL-241

Description: HP Laserjet 8000 Postscript

press continue

Device: LPD/LPR Host or Printer

press continue

Device URI: lpd://papermill.cis.fiu.edu/venom

press continue

Make: HP

press continue

Model: HP Laserjet 8000 series Postscript

press Add Printer

5. Click on the Printers Tab and search for the Venom printer. In the Set Printers Option change the following to:

Duplex Unit: Installed

Duplex: Flip on Long Edge (Standard)

Starting Banner: Standard

Click on Set Printers Option to save the changes. Click on Print Test Page to test the configuration.

Task 36:

1. Install and configure Bacula Backup System.
2. To test your installation, perform a backup of some files in your server to a file system or directory target on the same machine.

Also know how to obtain status information from the bacula director through the command line. Consider integrating with webmin to administer the backup tool from there. Please start the installation ASAP as it takes some time to correctly configure this tool. Do not wait until the last minute to do so as you won't get it done in time and right.

Here are some useful links. Review the tutorial and manual before installing it so that you familiarize with Bacula components:

- o <http://www.bacula.org/en/>
 - o <http://www.bacula.org/en/?page=downloads>
 - o <http://www.bacula.org/5.0.x-manuals/en/main/main>
 - o http://www.bacula.org/5.0.x-manuals/en/main/main/Brief_Tutorial.html
 - o <http://www.unixmen.com/install-and-configure-bacula-server-in-centos-6-4-rhel-6-4/>
3. `cd /tmp`
 - o `wget http://sourceforge.net/projects/bacula/files/latest/download?source=files`
 - o `tar xzvf bacula-7.0.5.tar.gz`
 - o Enable atomic repo
 4. Compile the package, run configure, make and make install
 - o `cd bacula-7.0.5`
 - o `yum install mysql-devel`
 - o `CFLAGS="-g -Wall" ./configure --enable-smartalloc --with-mysql --enable-conio`
 - o `make`
 - o `make install`
 - o `make install-autostart`

5. Create user database and tables for bacula, enter root password when prompted.

- `cd /etc/bacula`
- `./grant_mysql_privileges -p`
- `./create_mysql_database -p`
- `./make_mysql_tables -p`

6. Change bacula password

- `mysql -u root`
- `UPDATE mysql.user SET password=PASSWORD("root Password") WHERE user='bacula';`
- `FLUSH PRIVILEGES;`
- `exit`

7. Update Bacula Director

- `vi /etc/bacula/bacula-dir.conf`
- Director:

- Password:
 - # Client (File Services) to backup
 - Client {
- Address = myIP
- Password: "root password"
 - # Definition of file storage device
 - Storage {
 - Name = File
- Address = myIP
- Password: "root password"
 - # Generic catalog service
 - Catalog {
- `dbpassword = "root password"`
 - Console
- Password = "root password"

2. Update Bacula Console

- `vi /etc/bacula/bconsole.conf`
- Director {

- address = myIP
- Password = "root password"

3. Update Storage Daemon

- `vi /etc/bacula/bacula-sd.conf`
- Director

- Password = "root password"
 - Comment the section under
- #Restricted Director, used by tray-monitor to get the
- # status of the storage daemon
 - Device
- Archive Device = /backup

4. Update file daemon

- `vi /etc/bacula/bacula-fd.conf`
- Director

- Password = "root password"

- Comment the section under

- #Restricted Director, used by tray-monitor to get the
- # status of the storage daemon
- 2. Create backup folder
 - mkdir /backup
 - change permissions to bacula (didn't need to)
- 3. Start services
 - service bacula-dir start
 - service bacula-fd start
 - service bacula-sd start
 - chkconfig bacula-dir on
 - chkconfig bacula-fd on
 - chkconfig bacula-sd on
- 4. Add bacula ports to firewall
 - vi /etc/sysconfig/iptables
 - -A INPUT -m state --state NEW -m tcp -p tcp --dport 9101 -j ACCEPT
 - -A INPUT -m state --state NEW -m tcp -p tcp --dport 9102 -j ACCEPT
 - -A INPUT -m state --state NEW -m tcp -p tcp --dport 9103 -j ACCEPT
 - service iptables restart
- 5. Manage Bacula With Webmin
 - Log on to webmin
 - http://cts4348-node-5.cs.fiu.edu:10000
 - Click on refresh modules on the bottom left
 - Click on System
 - click on Bacula Backup System
 - Click on Module Configuration
 - Set the database settings
- Database type: MySQL
- User: bacula
- Password to login with: root password
 - save

Task 37:

1. Install Wire Shark (former Ethereal) Network Protocol Analyzer (<http://www.wireshark.org/>)

Documentation: (<http://www.wireshark.org/docs/>)

- yum install ethereal-gnome
- To execute the tool, type the following command at the shell prompt: tethereal
- Here's a link to the old man pages:
<http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man1/tetherea1.1.html>
- 2. Exercise 1 start up tethereal. Redirect the output to a text file.
 - Logon to your server using ssh via Putty.
- 3. Inspect and understand the resulting output file and trace all inbound and outbound packets emitted by the ssh interactions.
 - tethereal >> /tmp/shark1.pcap
 - Open putty and log on to cts4348-node-5 as root
 - stop capture by pressing ctrl+c
 - view ex1.pcap file and look for ssh traffic

4. Exercise 2: start up tethereal. Redirect the output to a text file.
 - o Startup Internet Explorer or Firefox, and go to your Drupal/WordPress portal and perform some tasks or browsing
5. Inspect and understand the resulting output file and trace all inbound and outbound packets generated by the http interactions between browser and your portal.
 - o tethereal >> /tmp/shark2.pcap
 - o open Chrome on your computer and go to: <http://131.94.134.5/wordpress/wp-login.php?>
 - o create a new comment on the page
 - o stop capture by pressing ctrl+c
 - o view shark2.pcap file and look for http traffic
6. Exercise 3: do a ping to www.google.com and trace the in/out packets resulting from the pinging. Make sure to stop the pinging soon after or specify number of response packets on the ping command line.
 - o log into the vmware console and log on to our machine
 - o log on to putty
 - o start a ping from the console
 - o start capture: tethereal >> /tmp/shark3.pcap
 - o stop capture
 - o stop ping
 - o view shark3.pcap file and look for the icmp traffic
7. Exercise 4: from your Linux virtual machine, perform a file transfer using ftp. Trace inbound and outbound packets resulting from the ftp interaction.
 - o log into 2 putty sessions
 - o session 1: start capture: tethereal >> /tmp/shark4.pcap
 - o session2: start ftp download: wget <ftp://ftp.pbone.net/mirror/ftp.sourceforge.net/pub/sourceforge/b/ba/bastille-linux/bastille-linux/3.0.9/Bastille-3.0.9-1.0.noarch.rpm>
 - o Session 1: stop capture
 - o view shark4.pcap file and look for the ftp traffic.
8. Write a small summary explaining your observations from the trail of in/out bound packets traced from the various interactions between your server and the opposite end resulting from the above exercises. Publish the trace outputs and summary on your group portal.

Shark 1:

The establishment of an encrypted communication was made between the server the the ssh client putty. or this connection to be established, putty required an encrypted key from the server which was provided. The, encrypted packets and data were exchanged between the two ssh client and server.

Shark 2:

A 3 way handshake TCP mechanism was used to make connection using a HTTP 1.1 get request. The reply was received and accepted thus allowing access to wordpress. As a result of adding a comment to the page we see "HTTP Continuation or non-HTTP traffic" packets in our HTTP trace. This is because the packet was too large to be received in one packet. Continuous packets were received until the request was complete.

Shark 3:

Several ICMP packets that identified both the request and reply of the ping message between putty and the local cache server from google.

Shark 4:

FTP connection is established between the server and ssh client upon the wget command download of the file. An FTP response of 220- segments is received from the site that host the file to be downloaded. The request is received with the user's login name which can be anonymous if the server allows it. The response is received asking for the password, the password is received via the following request. At this point a response is received saying the login was successful. A FTP SYST request is received indicating the host operating system was found. A FTP Response is received indicating the host name was found. The password is requested again, a path was created for the file, the file type is set to image, and FTP sets up for binary mode. The directory is changed to the location where the file will go. The size of the file is looked up, and the client is set to passive mode where it waits for a connection. The file transfer begins, when the transfer is complete the data connection is stopped.

TASK 38:

1. Enable FTP services. Allow anonymous FTP access to your Virtual Machine. Allow anonymous users to upload and download files. Set up Upload and Download directories in which to jail anonymous access. ALS, set up your (non-root) accounts (also consider the profe account) with FTP access to allow them to upload and download files from their respective home directories. You may choose to install VSFTPD, ProFTPD, or any other implementation available for the Linux platform.

1. Install vsftpd

o yum install vsftpd

2. Remove the current config file:

o rm /etc/vsftpd/vsftpd.conf

3. Create a new config file with the following contents:

o write_enable=YES

o anonymous_enable=YES

o anon_root=/var/ftp

o local_umask=022

o chroot_local_user=YES

o chroot_list_file=/etc/vsftpd/chroot_list

o local_enable=YES

o anon_upload_enable=YES

o anon_mkdir_write_enable=YES

o dirmessage_enable=YES

o xferlog_enable=YES

o connect_from_port_20=YES

o xferlog_std_format=YES

o listen=YES

o pam_service_name=vsftpd

o userlist_enable=YES

o tcp_wrappers=YES

o use_localtime=YES

4. Create directories for anonymous upload/download

- o mkdir -p /var/ftp/upload
 - o mkdir -p /var/ftp/download
 - o chown ftp:ftp /var/ftp/upload
 - o chown ftp:ftp /var/ftp/download
 - o chmod 722 /var/ftp/upload
 - o chmod 755 /var/ftp/download
5. Open port on firewall
- o vi /etc/sysconfig/iptables
 - o Add: -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
6. Add IPTABLES module
- o vi /etc/sysconfig/iptables-config
 - o Add the line:
 - o IPTABLES_MODULES="ip_conntrack_ftp"
7. Restart IPTABLES
- o service iptables restart
8. Edit tcp wrappers /etc/hosts.allow:
- o vi /etc/hosts.allow
 - o Add:
 - o vsftpd: ALL
9. Allow in SELINUX
- o setsebool -P allow_ftpd_full_access on
 - o setsebool -P ftp_home_dir access on
10. Start services
- o service vsftpd restart
 - o chkconfig vsftpd on

TASK 39. Configure emailing services on your server:

- o configure sendmail to send and accept
2. install and enable dovecot for pop3 and imap
3. install spamassassin
4. Install sendmail
- o yum install sendmail
 - o chkconfig sendmail on
 - o service sendmail start
5. Open port on firewall
- o -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
6. restart firewall
- o service iptables restart
7. Configure sendmail
- o vi /etc/mail/sendmail.mc
8. Edit/Add the following lines:
- o DAEMON_OPTIONS('Port=smtp,Addr=127.0.0.1, Name=MTA')
 - o LOCAL_DOMAIN('cts4348-node-5.cs.fiu.edu')
 - o MASQUERADE_AS('cts4348-node-5.cs.fiu.edu')
 - o MAILER(smtp)dnl

- o MAILER(local)dnl

9. Apply changes to sendmail

- o yum install sendmail-cf
- o make -C /etc/mail
- o service sendmail restart

10. Install spamassassin

- o yum install spamassassin
- o chkconfig spamassassin on
- o service spamassassin start

11. Install procmail

- o yum install procmail

12. Set up procmail recipes

- o vi /etc/procmailrc

13. Add the following:

```
:0 fw
```

```
| /usr/bin/spamc
```

```
:0
```

```
* ^subject: .*(virus|VIRUS)
```

```
| /dev/null
```

```
:0
```

```
* ^from: .*Osorio.*
```

```
! gespi039@fiu.edu
```

14. Install dovecot

- o yum install dovecot
- o chkconfig dovecot on
- o service dovecot start

15. Open ports on firewall

- o -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 110 -j ACCEPT
- o -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 143 -j ACCEPT

16. restart firewall

- o service iptables restart

17. Allow IMAP and POP3 in dovecot

- o vi /etc/dovecot.conf

18. Add the following lines in the beginning

```
protocols = imap pop3
```

```
protocol imap {
```

```
    listen = *:143
```

```

    }
    protocol pop3 {
        listen = *:110
    }
    listen = [*]

```

19. restart dovecot to apply changes
 - o service dovecot restart

test:

```
echo "This is g_root's Server" | mail -s "g_root" kbrow141@fiu.edu
```

TASK 40: Install and properly configure CVS – Concurrent Version Control System on your server

- 1) Install the following:


```

yum install cvs
yum install xinetd

```
- 2) Create group and user:


```

groupadd cvs
useradd -G cvs cvsuser
passwd cvsuser

```

 - enter root's password
- 3) Create cvs directory and change permission:


```

mkdir /cvs
chmod -R 777 /cvs

```
- 4) Create a new file and add the following info:


```

nano /etc/xinetd.d/cvspserver

```

 - add this lines:


```

service cvspserver
{
    disable = no
    port = 2401
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    passenv = PATH
    server = /usr/bin/cvs
    env = '$HOME=/cvs'
    server_args = -f --allow-root=/cvs pserver
    bind = 131.94.134.5
}

```

- now restart xinetd
service xinetd restart
- 5) Open port 2401 to allow connections to the cvspserver:
nano /etc/sysconfig/iptables
- add line:
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2401 -j ACCEPT
- restart iptables
service iptables restart
- 6) Create a repository:
cvs -d /cvs init
- 7) Create a test project folder under the cvs user:
su - cvsuser
mkdir /home/cvsuser/project
- 8) Assign and give full privileges to cvs group:
chgrp cvs/home/cvsuser/project/
chmod g+srwx /home/cvsuser/project
- 9) Create cvs repository with:
cvs -d /home/cvsuser/project init
- 10) Create a test file and write some text:
cd /home/cvsuser/project/
nano test
- 11) Import our directory into the repository:
cvs -d /cvs import project INITIAL start
- and check out project folder
cvs -d /cvs checkout project
- 12) Commit the change made to test file:
cvs -d /cvs commit project
- 13) View CVS history to check test file:
cvs -d /cvs log project/project/test
cvs -d /cvs annotate project/project/test

TASK 41: Install and properly configure GIT:

1. Install and configure GIT:

- a. `yum install git`
- b. `git config --global user.name "root"`
- c. `git config --global user.email kbrow141@fiu.edu`

2. Make a new directory for GIT:

- a. `mkdir /home/GitProject`
- b. `cd /home/GitProject`
- c. `git init`

3. Make a file in the directory:

- a. `nano gitproject.txt`
- b. add some random text:
 - i. "This is a GIT project"
- c. Press "Ctrl-X" to exit, then "Y" to save

4. Commit the file:

`git commit -m 'Initial upload of the project'`

5. Make changes to the gitproject file:

- a. `nano gitproject.txt`
- b. Change text:
 - i. "This is a GIT project, Goodbye"
- c. save and quit

6. view changed made to the file:

`git diff`

7. Add and commit the new file:

- a. `git add gitproject.txt`
- b. `git commit`

8. To view status and commit logs:

`git status` or `git log gitproject.txt`

TASK 42. Export a Samba share for work group CTS4348 (Click here for step-by-step instructions). Earn 5 extra points by successfully mapping a Network Drive to your Samba Share on a Windows Desktop in the JCCL Lab.

Install Samba:

`# yum install samba`


```
# yum install samba-common
# yum install samba-client
```

Check if Samba (or smb) is on, enabled it, and then start the service:

```
# chkconfig --list | grep smb
# chkconfig smb on
# service smb status
# service smb start
```

Create Samba user accounts from the Unix accounts, set password for a real user on the system (e.g. kbrow141):

```
# cd /etc/samba
useradd kbrow141
```

```
# smbpasswd -a kbrow141
Police393!
```

```
User: profe
Profe393
```

```
User: joel
Joel393
```

```
User: gespi039
Gespi393
```

Install SWAT to configure and manage Samba; SWAT uses xinetd to run:

```
# yum install samba-swat
# chkconfig --list | grep xinetd
# chkconfig xinetd on
```

Edit xinetd SWAT configuration to enable SWAT:

```
# vi /etc/xinetd.d/swat set 'disable=no' (was disable=yes)
# service xinetd start
```

Open firewall ports for Samba, SWAT (port 901):

```
# edit /etc/sysconfig/iptables by hand
service iptables restart
```

Create a directory /home/shared, then make it guest available as SHARED

```
# mkdir /home/shared
```

Set up Samba share:

Open browser, navigate to `http://localhost:901` (SWAT Tool).
Click on 'Global', change workgroup to CTS4348.
Click Commit Change.
Click on 'Shares'.
Type in new share 'SHARED' and then click 'Create new Share'.
Click on 'Shares' again.
Select 'SHARED' from the selection box, then 'Change Share'.
Set the 'path' option to '/home/shared'.
Set 'guest ok' to 'yes'.
Set 'available' to 'yes'.
Click on 'Commit Changes'.

(Rather than using the web interface, you could alternatively edit `/etc/samba/smb.conf` manually to add a new share. Make sure to run `testparm` on the modified config.)

Verify your work: See what shares are available on your Samba server using user `kbrow141`:

```
# smbclient -L //localhost -U kbrow141
```

Connect to the public SHARED folder as user guest:

```
# smbclient //localhost/SHARED -U guest
```

Connect to joe's home directory on the linux system (homes share) over Samba:

```
# smbclient //localhost/kbrow141 -U kbrow141
```

(Since SELinux default policy does not allow this, override with:
`/usr/sbin/setsebool -P samba_enable_home_dirs=1`)

TASK 43:

1. Install and configure Instant Messaging Services on your server. Make sure to open the necessary ports your selected package requires. Test your IM Service with a proper Windows Client tool and make sure you are able to

communicate with other IM members registered on your Linux Server. I will try to test your installation with Pidgin IM Client on Windows. Therefore, I encourage you to test your server with the same tool to ensure highest points for this exercise.

2. Download and install openfire

- o Open firefox and go to

<http://www.igniterealtime.org/downloadServlet?filename=openfire/openfire-3.10.2-1.i386.rpm>

- o `rpm -ivh openfire-3.10.2-1.i386.rpm`

3) Edit openfire.xml file:

`nano /opt/openfire/conf/openfire.xml`

- add 131.94.134.5 to the network:

```
<network>
```

```
    <interface>131.94.134.5</interface>
```

```
</network>
```

- restart Openfire

`service openfire restart`

4) Open the the following ports editing the iptable:

`nano /etc/sysconfig/iptables`

- add lines:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9090 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5222 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 7070 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 7777 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 7443 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3478 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3479 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5229 -j ACCEPT
```

- restart iptables

`service iptables restart`

5) Go to <http://131.94.134.5:9090> in your browser to seup openfir:

Domain: cts4348-node-5.cs.fiu.edu

Admin Console Port: 9090

Secure Admin Console Port: 9091

continue

check: Standard Database Connection

continue

Database Driver Presets: MySQL

JDBC Driver Class: com.mysql.jdbc.Driver

Database URL: jdbc:mysql://localhost:3306/kevaughndb (Create Database manually in mysql)

Username: root
Password: root's password
Minimum Connections: 5
Maximum Connections: 25
Connection Timeout: 1.5 Days
 continue
check: Default
 continue

Admin Email Address: kbrow141@fiu.edu
New Password: root's mysql password
Confirm Password: root's mysql password
finish

6. In the terminal, reset openfire:

 service openfire restart

7. Log in to OpenFire Admin Console

Username: admin
Password: roots' mysql password

8. Click on User/Groups and create a new user as follows:

- a. User: kbrow141
- b. Password: root's mysql password
- c. User: gespi039
- d. Password: root's mysql password
- e. User: joel
- f. Password: root's mysql password

9. Download and install Pidgin on two Windows computers and configure with the following settings to test the server:

BASIC TAB:

- a) Protocol: XMPP
- b) username: gespi039, kbrow141, joel
- c) Domain: cts4348-node-5.cs.fiu.edu
- d) Password: root's mysql password

ADVANCE TAB:

- a) Connection security: Use encryption if available
- b) Connect port: 5222
- c) Connect server: 131.94.134.5
- d) File transfer proxies: proxy.eu.jabber.org

10. Add Buddy to Buddy list

 kbrow141@cts4348-node-5.cs.fiu.edu

joel@cts4348-node-5.cs.fiu.edu
gespi039@cts4348-node-5.cs.fiu.edu

TASK 44: Install a VoIP PBX Package of your choice on your server. Make sure to set up a client Soft IP Phone to test your setup and document all installation, configuration and testing steps.

1) Download Ventrilo from website using Centos on the VMWare Console
<http://www.ventrilo.com/dlprod.php?id=102>

2) Open a terminal window (putty SSH) to the host computer that will be running the Ventrilo server.

3) Set working directory to where ever you want to create the ventrilo directory.

4) Type "mkdir ventrilo"

5) Type "cd ventrilo"

6) Copy the tar.gz file into this new directory.

7) Type "gunzip " followed by the name of the tar.gz file

8) Type "tar xf " followed by the name of the tar file. (gunzip removed the gz extension).

9) Note: Some platforms allow for combining steps 7 and 8 into a single command by typing "tar zxf " followed by the name of the tar.gz file.

10) Edit nano ventrilo_srv.ini file
nano /ventrilo/ventsrv/ventrilo_srv.ini

11) Under [Server] Change the following:

Name:cts4348-node-5.cs.fiu.edu

Auth=1

AdminPassword=root's password

Password=Kevaughn393!

12) Edit IPtables to add default ventrilo port number :3784

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3784 -j ACCEPT

-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 3784 -j ACCEPT

13) restart iptables

service iptables restart

14) Start Ventrilo server: "./ventrilo_srv"

a. cd /opt/ventrilo/ventsrv/.

b. ./ventrilo_srv

15) Download windows/Mac Client from Ventrilo website: <http://www.ventrilo.com/download.php>

An android app called **Ventriloid** can be used as mobile connection.

- Install Client
- Set Username/server name of your choice
- For server setup, use server IP address (131.94.134.5) or Hostname (cts4348-node-5.cs.fiu.edu)
- Set port to 3784 or leave as default
- Input Ventrilo server password: Kevaghn393!
- Click ok

Ventrilo offers a push to talk service that is full-duplex in communication and also offers a chat service for users

To run at Boot:

- `nano /etc/init.d/ventrilo`

Add script:

```
#!/bin/sh
cd /opt/ventrilo/ventsrv
./ventrilo_srv -d
```

Change permission so script is executable

- `chmod +x /etc/init.d/ventrilo`
- `/etc/init.d/ventrilo start`