Lesson 1
Oracle Linux Boot Process and Common Linux Utilities

Contents

Gentle Reminder:
At this point, the IP addresses of your Client are dynamically allocated at each boot time via the DHCP protocol. The IP address of your Server is set to a static value. Please check the current IP addresses beforehand.

1. root Password recovery

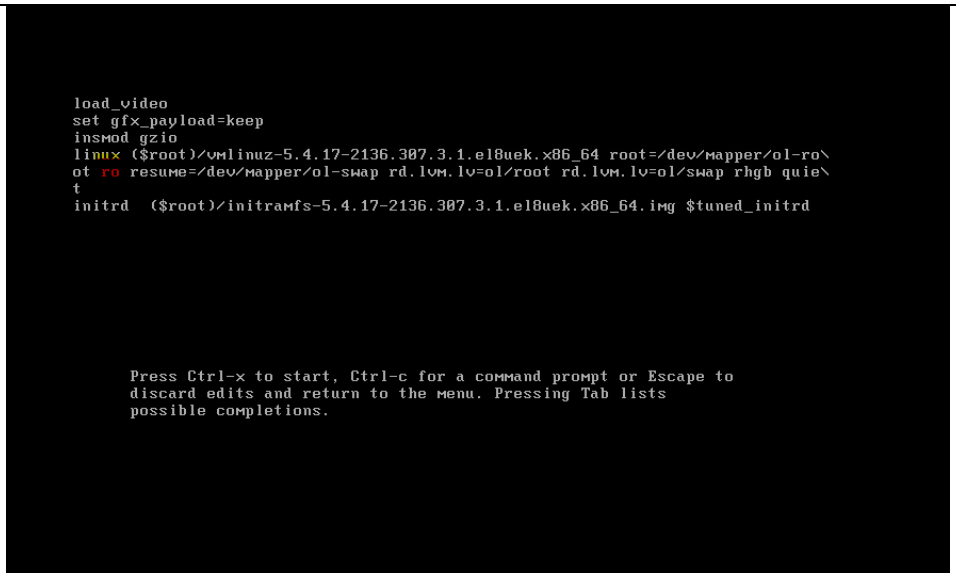
This exercise demonstrates how to reset the root password of an Oracle Linux system (Same approach may be applicable to RedHat, Centos and many other Linux OS). This technique is useful in case you have forgotten/lost the current root password.

Ref: https://youtu.be/eFKpbJSsObY
(The target system of the above demo video is a Centos 8 Linux.)

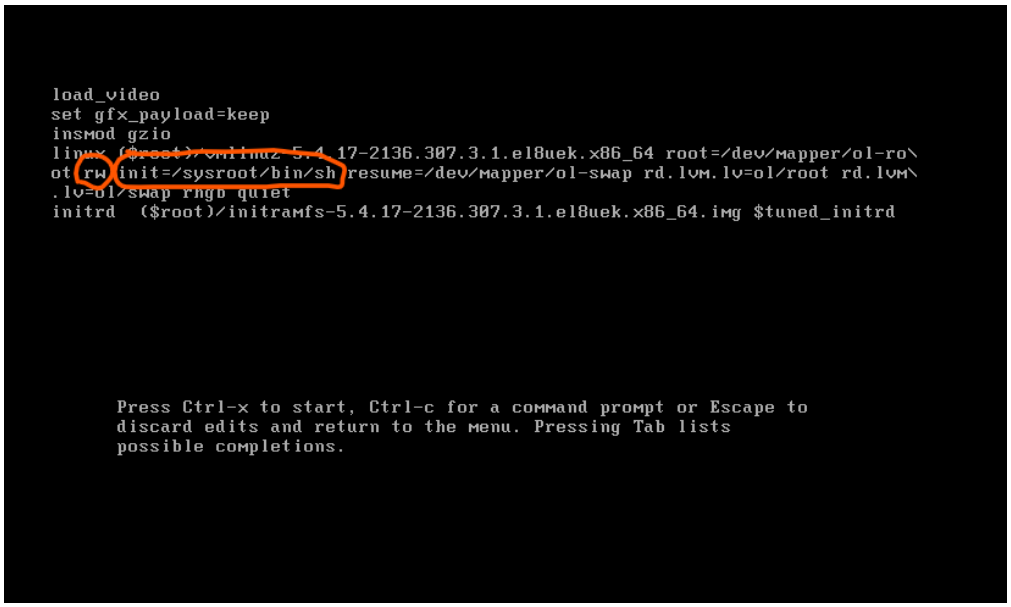
On client:

1. Power on the client VM and quickly press the 'e' while it is displaying the Grub boot menu. (You need to place the mouse pointer at the boot screen and click on it once before your keyboard event can be sent to the booting VM.)

2. Once you have successfully 'interrupted' the boot process at the Grub menu. You will be able to see the following bootloader set parameters screen:



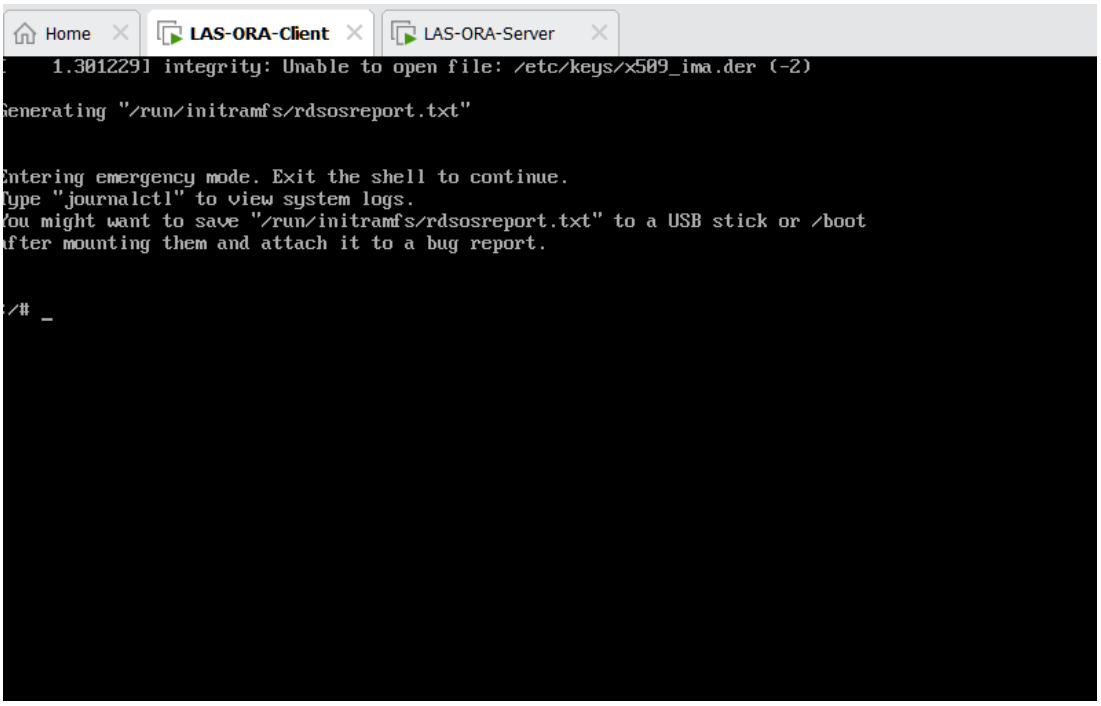
3. You may scroll down to search for the line starts with '**linux**'. This line defines which kernel boot image to be used with other required kernel parameters. By default, the initial boot will be run on a ram disk and the actual root file system will be mounted read only mode to /sysroot folder in the ram disk file system.
   At this line, you need to do two things:
   a) Change the read only mode to read write mode. Locate the ro parameter and replace it with rw.
   b) Append an additional kernel parameter right after the rw parameter.
      `init=/sysroot/bin/sh`
      This parameter will override the normal boot process to only run a simple sh (command prompt) instead of boot up the entire Linux system with the login interface.



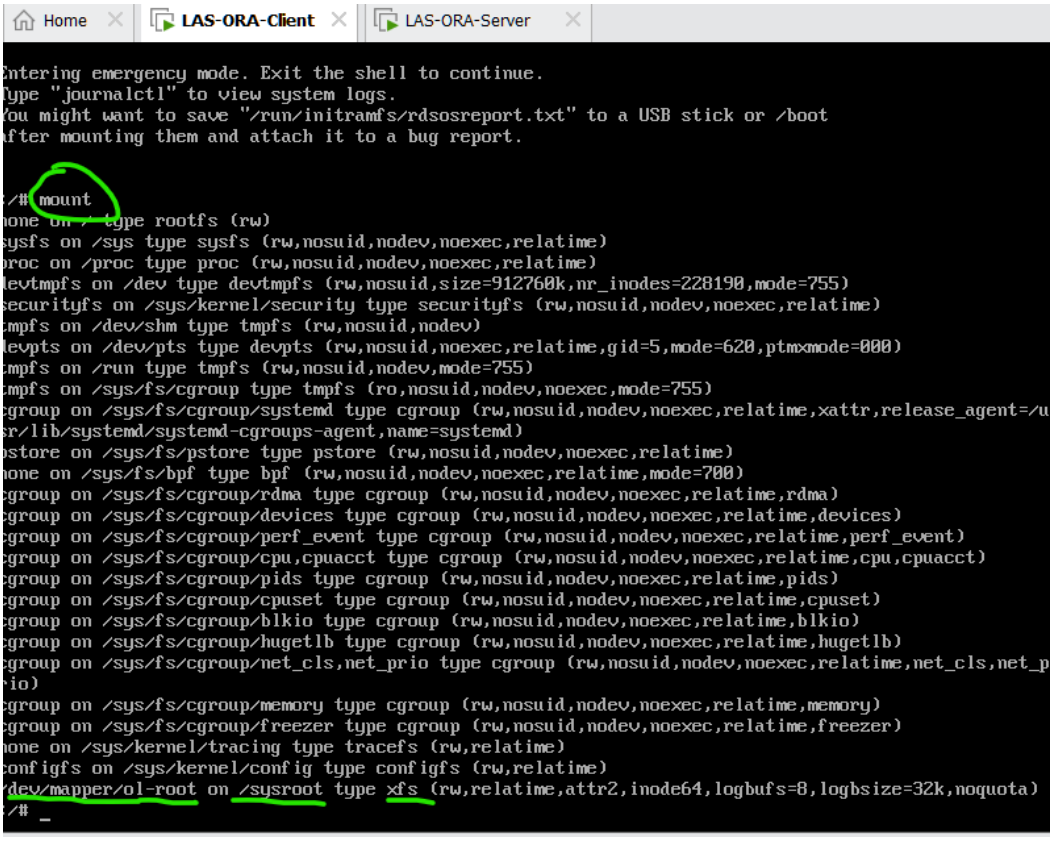
   As shown at the above, the 'ro' parameter has been replaced with 'rw' and the 'init=/sysroot/bin/sh' parameter has been added accordingly.

4. You may then press Ctrl-x to resume the boot process. The boot process will be

resumed. When the boot process is completed, it provides you a command prompt (with the root privilege).



5. At this point, your linux system is actually boot from a ramdisk with the bare minimum system files and tools. If you issue passwd command to change your root password, it will only save the updated the password to the temporary ramdisk file system. The real root file system at this point is mounted under /sysroot. You may type in the 'mount' command to verify it.

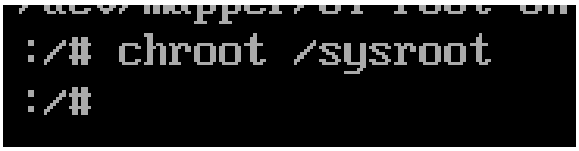


In the above screenshot, the first displayed mount entry, 'none on / type rootfs (rw)' ,is the current ramdisk based file system. The last entry is the real Linux root file system stored in the disk storage. Take note of the 'device name' of the real file system: /dev/mapper/ol-root shown in the output.

6. To proceed to reset the root password which is stored in our 'real' root file system, we need change our current root mount from / to /sysroot.
   The **chroot** command can change the current root mount point to the targeted filesystem.

   Type :

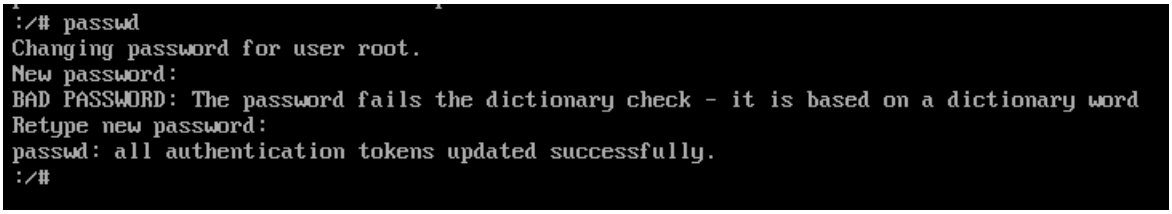
   ```
   chroot /sysroot
   ```

   Now, you can use the passwd command to reset your root password and it will be stored in your 'real' root file system.

7. Type the passwd command to change the root password to 'student'.
   You may use
   passwd
   Or
   passwd root

   to reset your root password:

   The system may give you BAD PASSWORD warning, but the reset will be successful. (For testing purpose, you may reset the root password of your client system to 'P@ssw0rd!' for now.)

8. To follow the Oracle Linux / Redhat (Linux that support SELINUX) convention . We need to ensure the '.autorelabel' file is created at the / folder. This is to ensure the next system will relabel all the SELINUX related context in the next boot time. This relabeling may take a while but it is a good practice to ensure the SELINUX subsystem will function accordingly. Type

   ```
   touch /.autorelabel
   ```

to create or update the timestamp of the file (if it is already existed).
Take note that the timestamp file name is **'/.autorelabel'**.

```
:/# touch /.autorelabel
```

After the file is updated type in

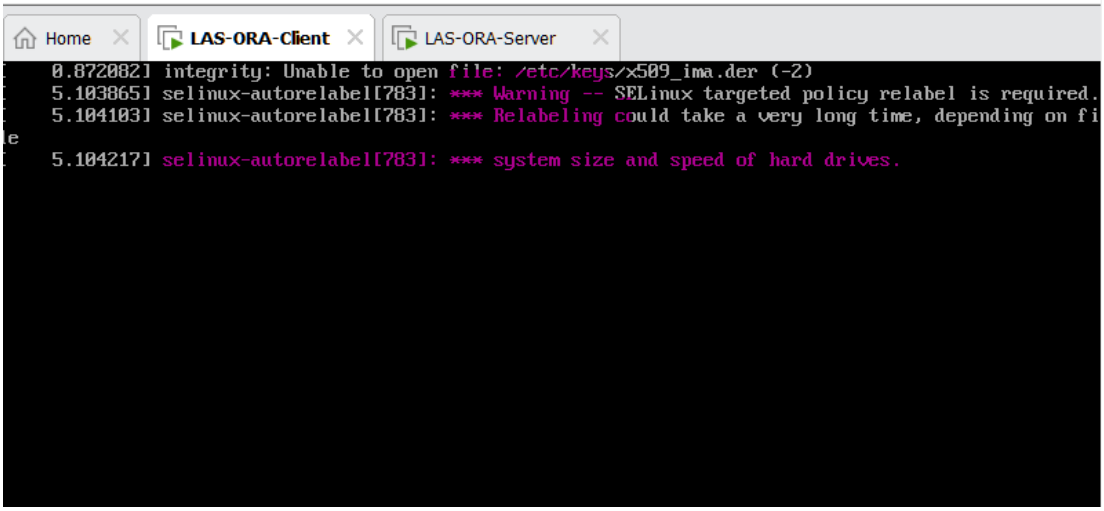
```
exit
```

to exit the shell.
At the system prompt type

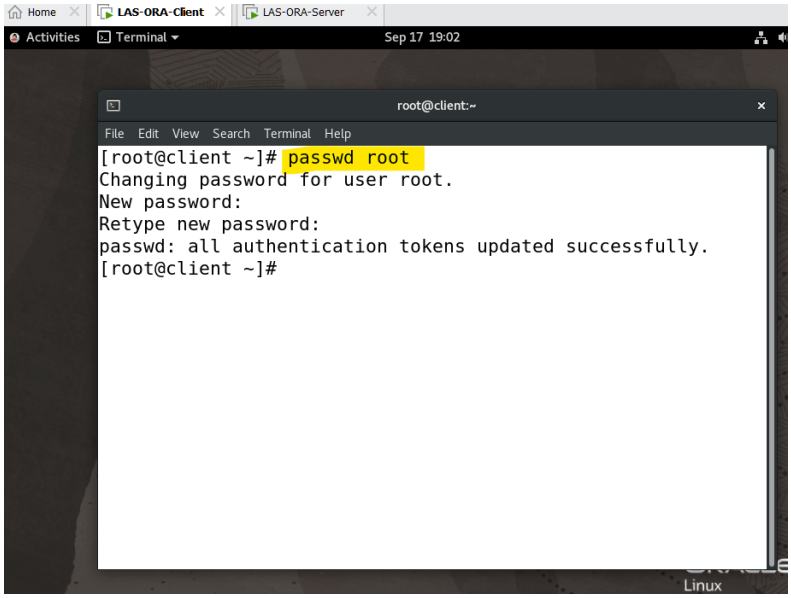
```
init 6
```

to reboot the system.

9.



As shown in the above, the system restarts normally and showing the warning messages that it has to go through the relabeling process. (It will take some time to enable the SELinux context for all the files.) After the system is up and running you may login to root using the password 'P@ssw0rd!'. You may want to change the root password back to '1wer$#@!' after login successfully.



2. Grub and Linux Boot Process

GRUB stands for Grand Unified Bootloader. Grub boot loader can be configured dynamically, which means a user has an option to make changes while booting. Even users can also easily alter the current boot entries, they can add new entries, select multiple kernels or even they can modify initrd. GRUB has also got the support of Logical Block Address. GRUB can be installed and executed from any type of device like hard disk, CD and USB. GRUB and GRUB2 are two different versions. Modern day linux distros (Ubuntu, RedHat, Oracle Linux) are all using GRUB or GRUB2. For Oracle 8, it is using GRUB2.

ref: https://linoxide.com/best-difference-between-linux-grub-and-grub2-bootloader/

1. Refer to https://www.thegeekstuff.com/2011/02/linux-boot-process to identify the six stages of the Linux boot process.

   1) _____
   2) _____
   3) _____
   4) _____
   5) _____
   6) _____

   More ref: https://linoxide.com/boot-process-of-linux-in-detail/

2. In the previous section, we have learned how to alter the boot process by modifying the GRUB parameter and reset a lost root password. It is a nice recovery technique, but it also imposing a security threat. To prevent unauthorized password reset attempt, we may configure our Grub with password protection.
   (ref: How to set grub2 password in RHEL/CentOS 7/8 (Step by Step Guide) - Edumotivation)

On Server:

Ensure you have powered on your Server. (It serves as the repository for your client)

On Client:

Login as root via GUI and start a new terminal.

Type the following command

```
grub2-setpassword
```

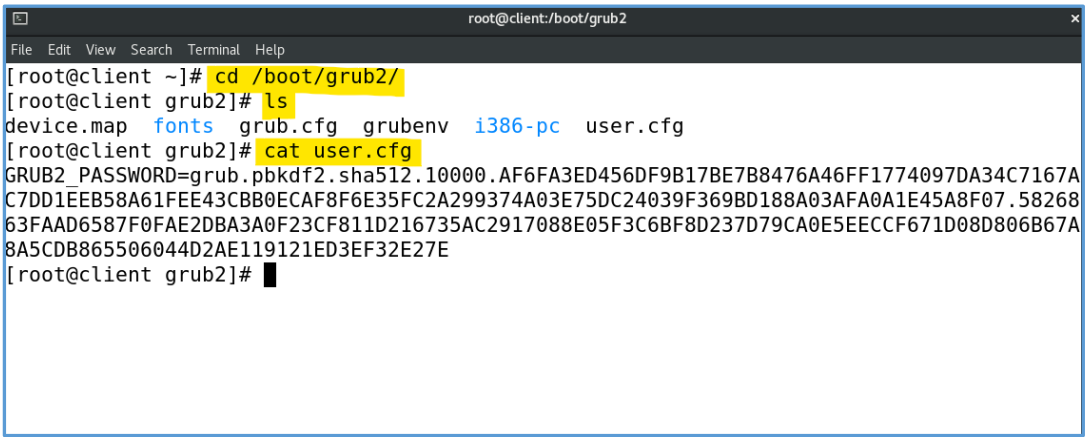
to set a grub2 password for the root user.

When prompt for the password, type in '`Oracle86`'

```
[root@client ~]# grub2-setpassword
Enter password:
Confirm password:
[root@client ~]#
```

The above command generates a hashed password that is stored in the /boot/grub2/user.cfg file. (At this point, the new grub2 password is not deployed yet.)

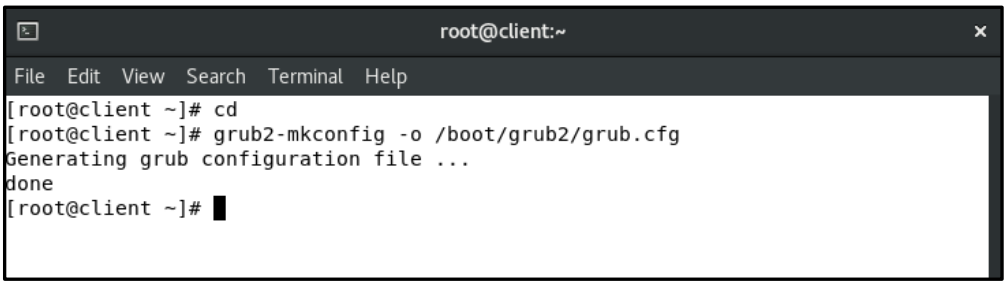
3. You may display the content of the user.cfg to show the generated password:



The password is a hash value based on your input. 'pbkdf2' stands for 'Password-Based Key Derivation Function 2'. This hashing scheme makes password cracking much more difficult. (ref: https://en.wikipedia.org/wiki/PBKDF2)

4. Recreate the GRUB2 Configuration files by running the following command:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```



5. To verify if the Grub menu is password protected. Restart your client and try to repeat the root Password recovery steps. In this case, you should not be able to do so unless you have entered the correct Grub user id ('root') and password ('0racle86')

   Although the security level of your Oracle Linux system has been improved with the password protection on the Grub menu your root file system is still vulnerable when it is against other type of boot related attacks. We will revisit this issue in the later part of this module.
   Let's move on to some other essential Linux tools and utilities.

3. Test ssh access connectivity between your systems

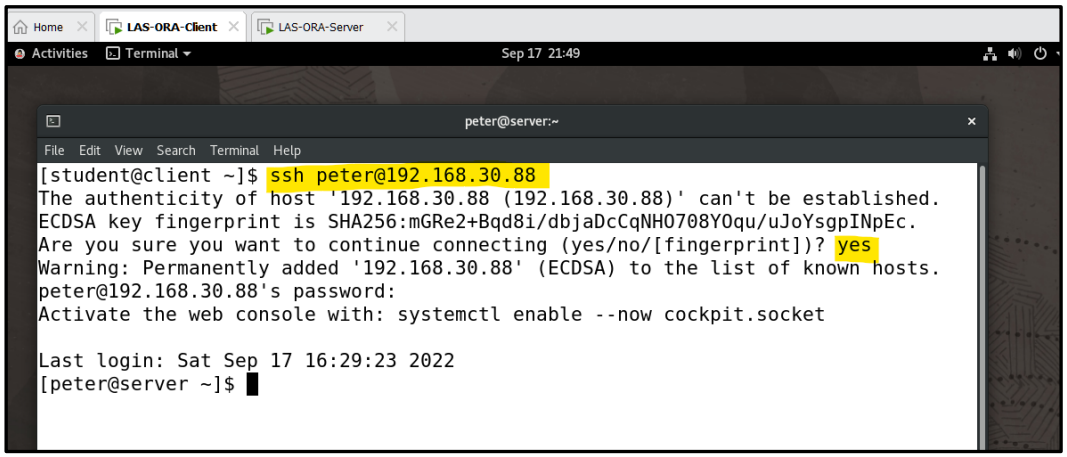
Secure shell (ssh) is a common remote login tools for Linux administrators.
SSH provides a secure channel over an unsecured network in a client–server architecture, connecting an SSH client application with an SSH server. SSH was designed as a replacement for Telnet and for unsecured remote shell protocols such as the Berkeley rlogin, rsh, and rexec protocols.
Thus, one of the common practices is to disable/disallow telnet, rlogin, rsh and rexec in a modern-day Linux system.
SSH services (and the firewall settings (**TCP port 22**) is enabled and started by default in Oracle Linux.

1. Power on both of your server and client VMs.

On client:

2. Login as student (password is 'user'), use ssh at a terminal to do a remote login to your server as user peter. You will need to enter the password of the user peter on the server system.

ssh *peter@<serverIP>*



Take note that in the above screen shot, you only type in the part that highlighted in yellow.

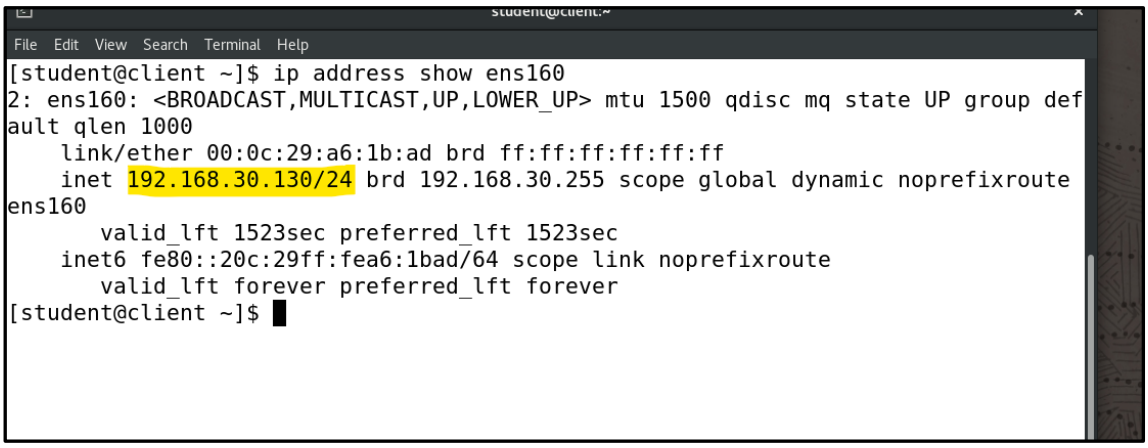
For an initial ssh connection, the user will be prompted to accept the public key that is offered by the server, you have to response with a 'yes' to proceed. This prompt will not appear in the subsequent connections.

Note: The message: "Activate the web console with: systemctl enable --now cockpit.socket" refers to a secure web based remote administrative console, cockpit. It is recommended to enable the cockpit interface for remote access to the system for administrative tasks. We may cover cockpit later.

3. Type “exit” to close the remote connection.

4. Check the current ip address of your client system. Type:

```
Ip address show ens160
```

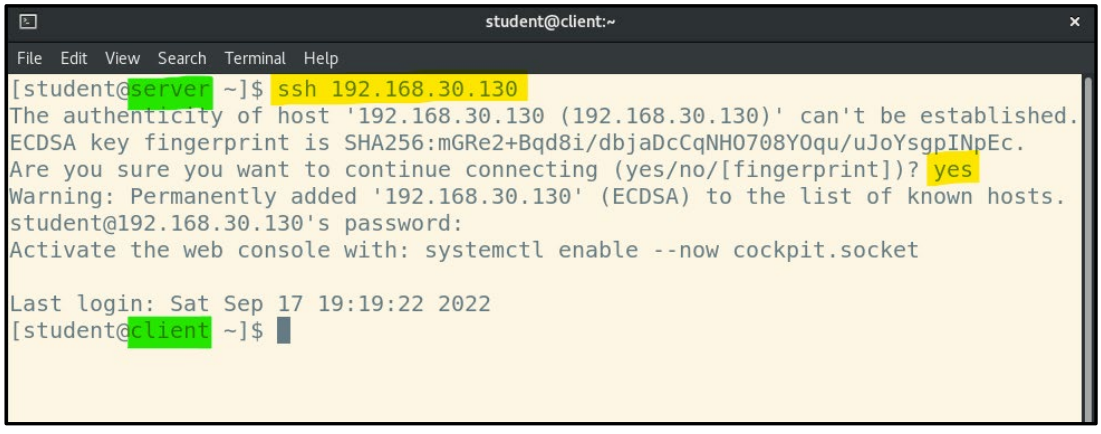


The above depicts the current ip address for the device ens160 is assigned as 192.168.30.130 (with 24-bit mask).

On server:

5. Login as student and use ssh to do a remote login to your client without specifying any user name. The client side will assume you want to login to the student account of the client system. Thus, you will need to enter the password of the student on the client system when it is prompted.

```
ssh <clientIP>
```



Observe that the changes on the Terminal title and the command prompt after the ssh session is successfully connecting to the client VM.

6. Type “exit” to close the remote connection.

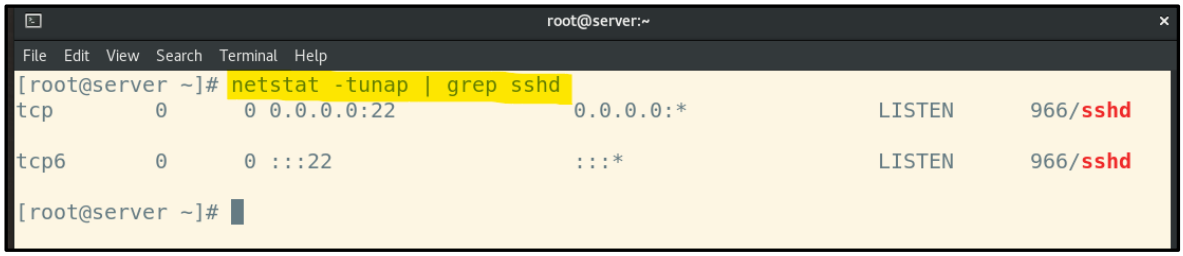
Recap: By default, the SSH service is installed and enabled in the Oracle Linux System, the SSH service runs on port 22. The default firewall setting also allows port 22 traffic. We

will now try running SSH on a different port number.

On server: (login as root)

1. Use netstat to see which port the SSH service is running on. It should be port 22. Type:

```
netstat -tunap | grep sshd
```



The output of the netstat displays the essential information of all the network services in various columns.

There are two entries related to sshd. One is running for tcp v4 the other is for tcp v6.

For the first entry (tcp v4), in column 4, **0.0.0.0:22**, states that the sshd service is accepting request from any of the network interface(s) (0.0.0.0) and the port number it is listening on is 22.

In the column 7, 966/sshd , denotes the PID (process ID - 966) / service program (sshd).

On Client (login as root)

2. Check the current repos are only two offered by your LAS server, type:

```
dnf repolist
```

```
[root@client ~]# dnf repolist
repo id                          repo name
las_ol8_appstream                Oracle Linux 8 Application Stream (x86_64)
las_ol8_u6_baseos_base           Oracle Linux 8.6 BaseOS (x86_64)
[root@client ~]#
```

3. Install the popular nmap port scanning utility, type:

```
dnf install nmap  -y
```

(Do you know what is the '-y' option for?)

After the nmap installation is completed successfully you can try to use nmap to scan your server:

nmap <serverIP>

```
[root@client ~]# nmap 192.168.30.88
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-17 22:04 +08
Nmap scan report for 192.168.30.88
Host is up (0.00055s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:0C:29:0A:99:A9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 10.43 seconds
[root@client ~]#
```

As shown, a simple nmap scan can reveal that the server is offering ssh login. There is a common practice that the administrator may change the configuration of the sshd services to let it listen on different tcp port number. This simple modification may help to improve the security level against port sweeping attacks on standard ports.

Other than the ssh service, nmap also reported the ftp service (tcp port 21) is running and the tcp port 9090 is closed. (It implies the firewall allows traffic to pass thru tcp port 9090 but there is no service running on the port. You will soon find out what is tcp port 9090 for.)

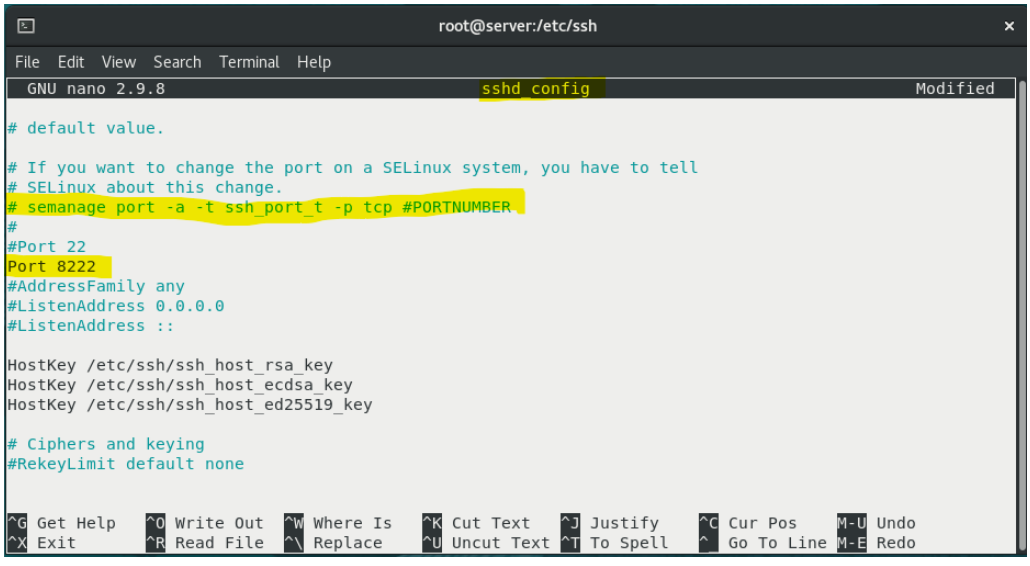
On server: (login as root)

4. Edit the SSH service config file /etc/ssh/sshd_config and change the port to 8222.

   Locate the commented line: #Port 22 and add the following line

   Port 8222

   Below it.



The above shows the content of the /etc/ssh/sshd_config file is editing by using nano editor.

The default Port no for sshd is 22 (If the Port entry is undefined). In the above, we have defined Port to be 8222. (which is not a standard port number, a standard nmap scan will not scan this port.)

Caveat: Please ensure you are editing the /etc/ssh/sshd_config file but not the /etc/ssh/ssh_config file !

5. Save the file and exit the editor.

6. The current(default) SELinux policy only allows SSH to run on certain port numbers. To allow SSH to listen on tcp Port 8222, you need to run the following command. (SELinux will be covered in more detail later. BTW, the sshd_config file also reminds you to run this command.)

```
semanage port  -a  -t  ssh_port_t  -p tcp 8222
```

7. Restart the SSH service to let the sshd to listen on port 8222.

```
systemctl restart sshd
```

Use the netstat command again to check the current sshd listening port.

```
[root@server ~]# semanage port  -a  -t  ssh_port_t  -p tcp 8222
[root@server ~]# systemctl restart sshd
[root@server ~]#
[root@server ~]# netstat -tunap | grep sshd
tcp        0      0 0.0.0.0:8222            0.0.0.0:*               LISTEN      12160/sshd
tcp6       0      0 :::8222                 :::*                    LISTEN      12160/sshd
[root@server ~]#
```

The above confirms the sshd is now serving at port 8222. Do you also notice the process ID has been changed to 12160 as we have restarted the sshd process ?

8. Since we are not listening on Port 22 for ssh connections. We should configure the firewall to block the port 22 (ssh service), type:

```
firewall-cmd --remove-service=ssh --permanent
firewall-cmd --reload
```

```
[root@server ~]# firewall-cmd --remove-service=ssh --permanent
success
[root@server ~]# firewall-cmd --reload
success
[root@server ~]#
```

(We will cover more on firewall-cmd in the later lessons)

<u>On Client (login as root)</u>

9. Rerun nmap to scan your server:

```
[root@client ~]# nmap 192.168.30.88
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-17 22:15 +08
Nmap scan report for 192.168.30.88
Host is up (0.00073s latency).
Not shown: 998 filtered ports
PORT     STATE  SERVICE
21/tcp   open   ftp
9090/tcp closed zeus-admin
MAC Address: 00:0C:29:0A:99:A9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 12.45 seconds
[root@client ~]#
```

As shown, a standard nmap scan may report that the server is not running sshd.

On client:

10. See if you can connect to the ssh service on the server from your client.

    ssh –p 8222 user@*serverIP*

```
[root@client ~]# ssh -p 8222 peter@192.168.30.88
ssh: connect to host 192.168.30.88 port 8222: No route to host
[root@client ~]#
```

You should not be successful. Why?

Task 1

Try to configure the system(s) so that your client can connect to the sshd service that is running on your server on port 8222. Hint: firewall

Work with your classmates and/or consult your tutor for the solution if needed.

```
[root@client ~]# ssh -p 8222 peter@192.168.30.88
The authenticity of host '[192.168.30.88]:8222 ([192.168.30.88]:8222)' can't be estab
lished.
ECDSA key fingerprint is SHA256:mGRe2+Bqd8i/dbjaDcCqNHO708YOqu/uJoYsgpINpEc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.30.88]:8222' (ECDSA) to the list of known hosts.
peter@192.168.30.88's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Sep 17 21:49:43 2022 from 192.168.30.130
[peter@server ~]$
```

After you have resolved the issue at the server side, you should be able to access to the server from your client via ssh via port 8222 as shown at the above.

Task 2

On server:

Revert the sshd configuration back to the default state. Configure the SSH service to run on the default port 22.
Reset back any changes you have done to the firewall configuration.
Use `semanage` command to remove the port 8222 from ssh_port_t association.

Work with your classmates and/or consult your tutor for the solution if needed.

Hint: use `semange port -l`
to list out the current SELinux port type with port number association.
You may apply the grep command to display the output related to ssh
`semanage port -l | grep ssh`

11. Briefly describe and list all the commands/configurations that you have used/applied to revert the system to its default state:

12. Verify your system configuration by applying the following commands as shown in the following screenshots:

    (on server)

```
[root@server ~]# netstat -tunap | grep sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      12566/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      12566/sshd
[root@server ~]# semanage port -l | grep ssh
ssh_port_t                     tcp      22
[root@server ~]#
```

sshd is running and listening on tcp port 22.
Only tcp port 22 is associated with selinux port ssh_port_t.

(on client)

```
[root@client ~]# nmap 192.168.30.88
Starting Nmap 7.70 ( https://nmap.org ) at 2022-09-17 22:42 +08
Nmap scan report for 192.168.30.88
Host is up (0.00055s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
9090/tcp closed zeus-admin
MAC Address: 00:0C:29:0A:99:A9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.51 seconds
[root@client ~]#
```

nmap running on client detects the ssh service is running at the server.

4. Secure Copy (scp) and Secure FTP (sftp)

Description: On top of ssh clients, SSHD services are accepting connections from several other SSH clients like secure copy (scp) and secure ftp (sftp).

On server: (Assume /tmp/practice is not created yet.)

1. Login as root and open a new terminal. Create an empty file, practice, at the /tmp folder

```
touch /tmp/practice
```

```
File  Edit  View  Search  Terminal  Help
[root@server ~]# touch /tmp/practice
[root@server ~]# ls -l /tmp/practice
-rw-r--r--. 1 root root 0 Sep 18 17:19 /tmp/practice
[root@server ~]#
```

On client:

2. Login as root, open a terminal, stay at the default folder path and do a secure copy of a file from your server to the client at a terminal prompt.

server file to copy

Destination to copy to
("." refers to current directory)

```
scp serverIP:/tmp/practice .
```

Similar to ssh command, you may be prompted for acceptance of the key and the root password to complete the remote login.

3. Check that the file has been copied over.

```
ls -l
```

```
[root@client ~]# pwd
/root
[root@client ~]# scp 192.168.30.88:/tmp/practice .
The authenticity of host '192.168.30.88 (192.168.30.88)' can't be established.
ECDSA key fingerprint is SHA256:mGRe2+Bqd8i/dbjaDcCqNHO708YOqu/uJoYsgpINpEc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.30.88' (ECDSA) to the list of known hosts.
root@192.168.30.88's password:
practice                      100%    0     0.0KB/s   00:00
[root@client ~]# ls -l
total 8
-rw-------. 1 root root 1098 Sep 11 19:08 anaconda-ks.cfg
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Desktop
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Documents
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Downloads
-rw-r--r--. 1 root root 1558 Sep 11 19:41 initial-setup-ks.cfg
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Music
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Pictures
-rw-r--r--. 1 root root    0 Sep 18 17:22 practice
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Public
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Templates
drwxr-xr-x. 2 root root    6 Sep 14 17:23 Videos
[root@client ~]#
```

4. Use secure ftp to connect to the server as user peter.

```
sftp peter@serverIP
```

5. At the sftp> prompt, check which directory you are in and list the contents of the directory.

```
pwd
ls -a
```

6. Download the file .bash_profile and name it as peter_profile. Close the connection

```
get .bash_profile   peter_profile
exit
```

7. Check that the file has been copied over.

```
ls -l p*
```

```
[root@client ~]# whoami
root
[root@client ~]# pwd
/root
[root@client ~]# sftp peter@192.168.30.88
peter@192.168.30.88's password:
Connected to peter@192.168.30.88.
sftp> ls -a
.              ..             .bash_history  .bash_logout   .bash_profile  .bashrc        .config
.esd_auth      .mozilla
sftp> get .bash_profile peter_profile
Fetching /home/peter/.bash_profile to peter_profile
/home/peter/.bash_profile                                   100%  141     9.2KB/s   00:00
sftp> quit
[root@client ~]# ls -l p*
-rw-r--r--. 1 root root 141 Sep 18 17:28 peter_profile
-rw-r--r--. 1 root root   0 Sep 18 17:22 practice
[root@client ~]#
```

In this section, we have tried 3 different types of SSH clients. They are all based on the same ssh protocol.
ssh - provide secure remote terminal session for command line based shell.
scp - provide secure file copy feature.
sftp - provide secure file transfer feature via an ftp like session command shell. (Note: sftp is running on ssh protocol, it is different from the ftp protocol.)

5. Introduction to cockpit web console

Cockpit is a web-based graphical interface for servers, intended for everyone, especially those who are: new to Linux (including Windows admins) familiar with Linux and want an easy, graphical way to administer servers.

Ref: https://cockpit-project.org/

The following material is based on: https://docs.oracle.com/en/operating-systems/oracle-linux/8/obe-cockpit-install/

On server (login as root)

By default, cockpit should be installed but not yet enabled nor running in an Oracle Linux 8 system. You may run the dnf install cockpit to ensure you have installed with the latest version cockpit.

1. Try to install the cockpit web console, type:

```
dnf install cockpit
```

```
[root@server ~]# dnf install cockpit
Last metadata expiration check: 20:56:20 ago on Sat 17 Sep 2022 08:39:06 PM +08
.
Package cockpit-264.1-1.0.1.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@server ~]#
```

As shown at the above, the cockpit is already installed.

2. Enable and start the cockpit services, type:

```
systemctl enable --now cockpit.socket

systemctl start cockpit
```

```
[root@server ~]# systemctl enable --now cockpit.socket
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket → /usr/
lib/systemd/system/cockpit.socket.
[root@server ~]# systemctl start cockpit
[root@server ~]#
```

3. Check if cockpit is running, type:

```
systemctl status cockpit

netstat -tunap | grep 9090
```

```
[root@server ~]# systemctl status cockpit
● cockpit.service - Cockpit Web Service
   Loaded: loaded (/usr/lib/systemd/system/cockpit.service; static; vendor pre>
   Active: inactive (dead) since Sun 2022-09-18 17:40:25 +08; 1min 54s ago
     Docs: man:cockpit-ws(8)
  Process: 2997 ExecStart=/usr/libexec/cockpit-tls (code=exited, status=0/SUCC>
 Main PID: 2997 (code=exited, status=0/SUCCESS)

Sep 18 17:38:54 server.example.com systemd[1]: Starting Cockpit Web Service...
Sep 18 17:38:55 server.example.com systemd[1]: Started Cockpit Web Service.
Sep 18 17:40:25 server.example.com systemd[1]: cockpit.service: Succeeded.
[root@server ~]# netstat -tunap | grep 9090
tcp6       0      0 :::9090                 :::*                    LISTEN      1/systemd
[root@server ~]#
```

As shown at the above, it seems that the cockpit service is inactive! Actually the service is running under the systemd process. As the default listen port number of cockpit is 9090, thus, we use netstat to check if the port 9090 is being used.

4. Configure firewall to allow cockpit, type: (optional, as port 9090 is open by default)

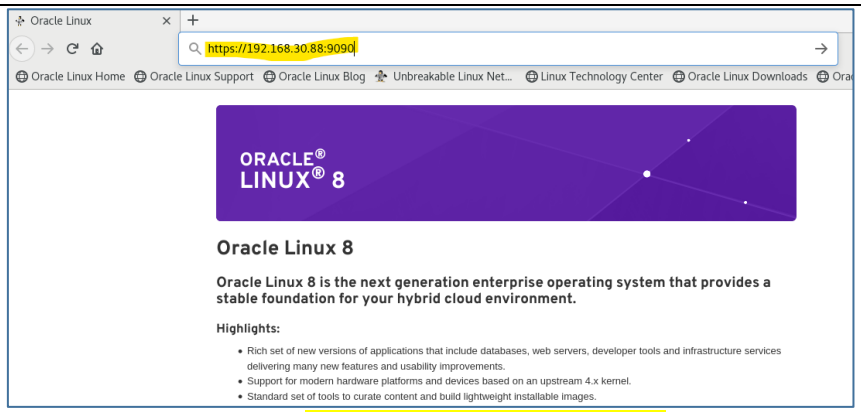
```
firewall-cmd --add-service=cockpit --permanent
firewall-cmd --reload
```

On client (login as student)

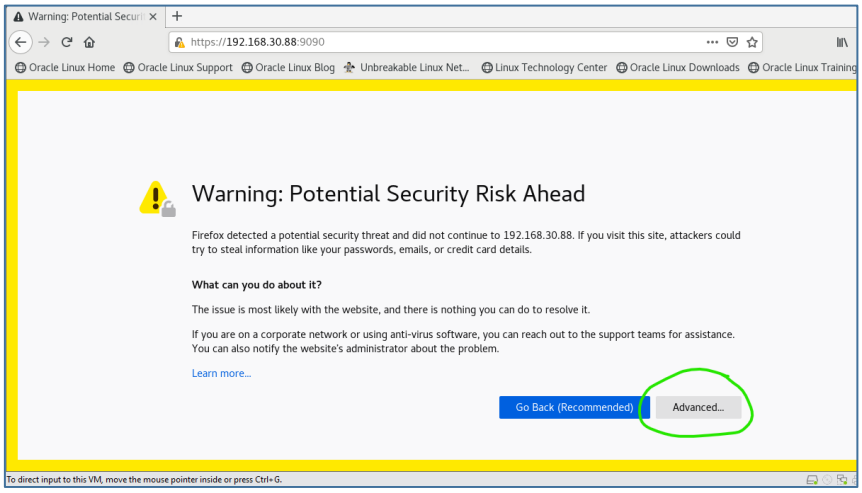
5. Access and Logging into the Cockpit web site that runs on the server.

   Open a firefox browser, go to the Cockpit web console page using the IP address of the server at port 9090 via https connections.

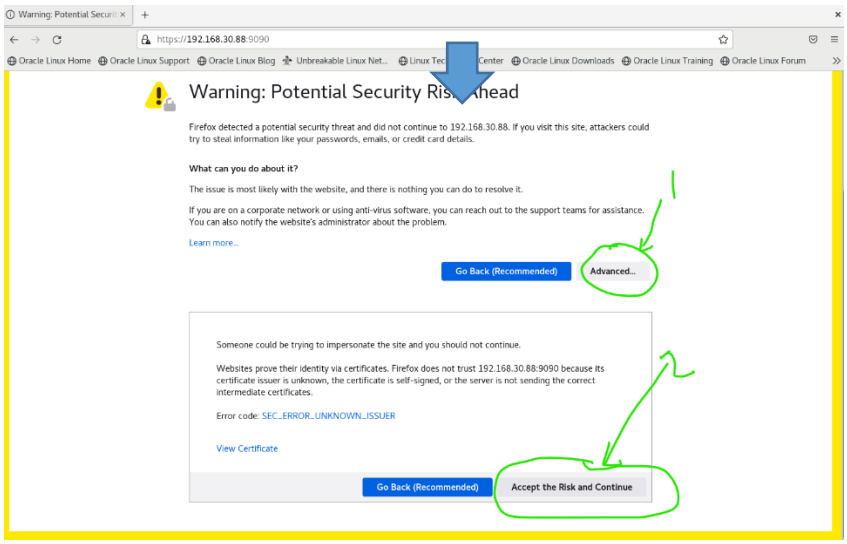
   Type: https://<server ip>:9090 at the URL box of the firefox.



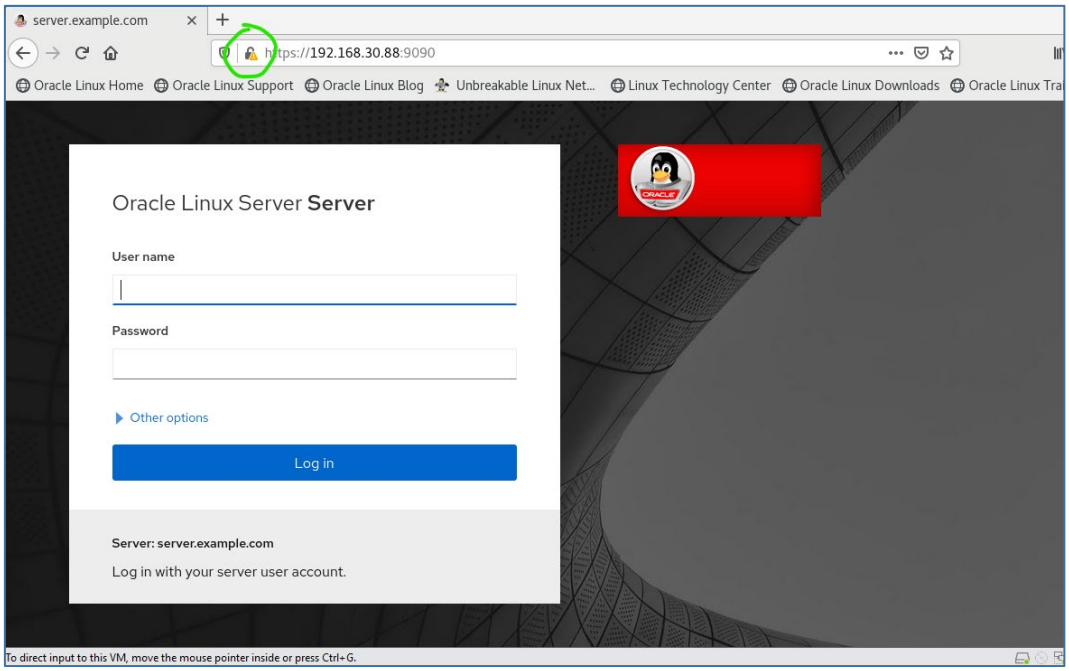
(Note: The URL used is https://192.168.30.88:9090)



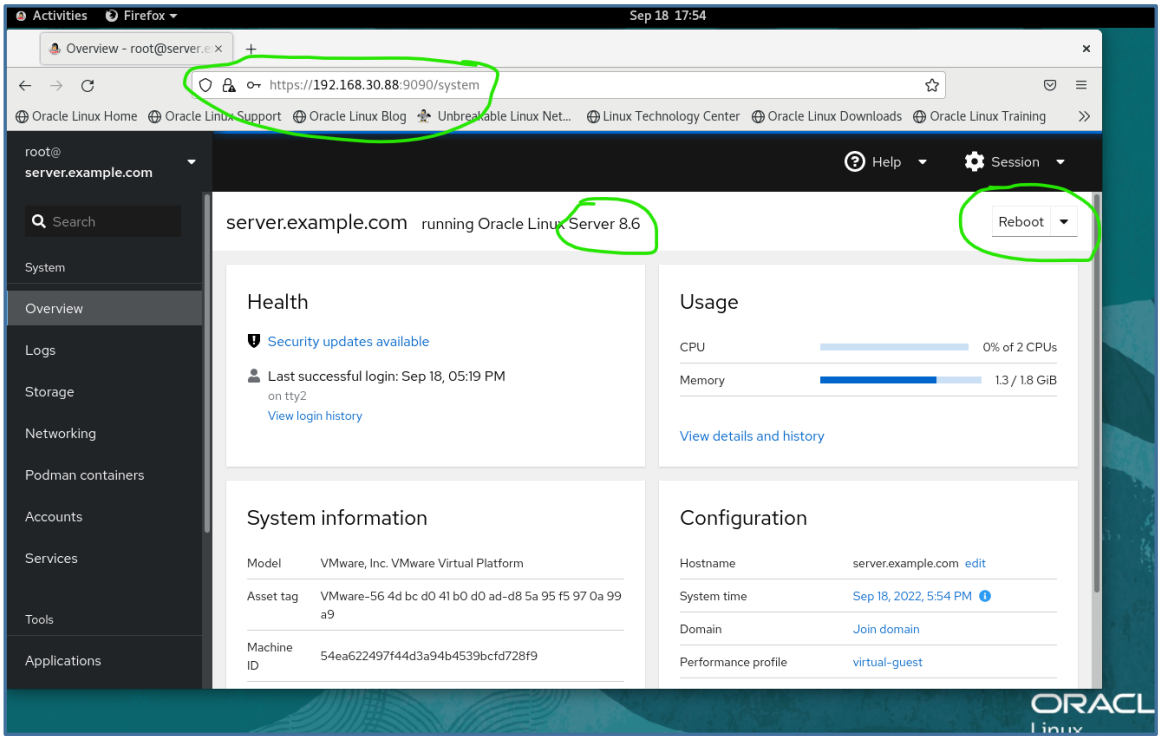
You will encounter a Warning page and you may click on the Advanced… button to proceed.



Click on Accept the Risk and Continue button.



Now you will see the login page of the web console and you may login to the server with the root credential.

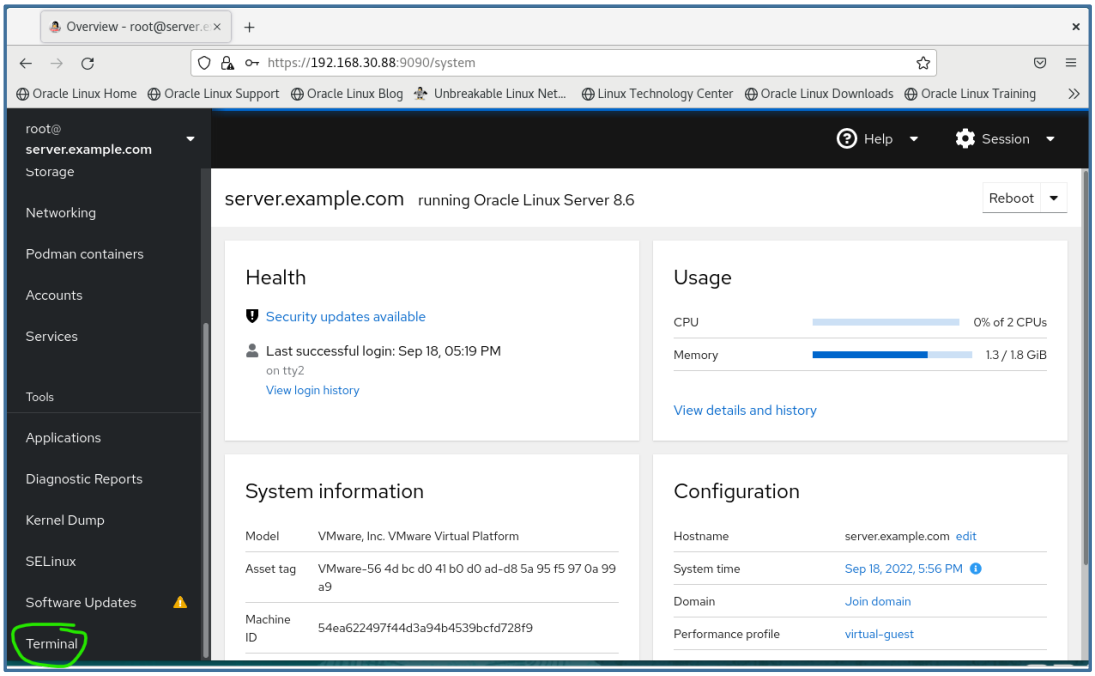


6. Launching a remote terminal shell at cockpit.

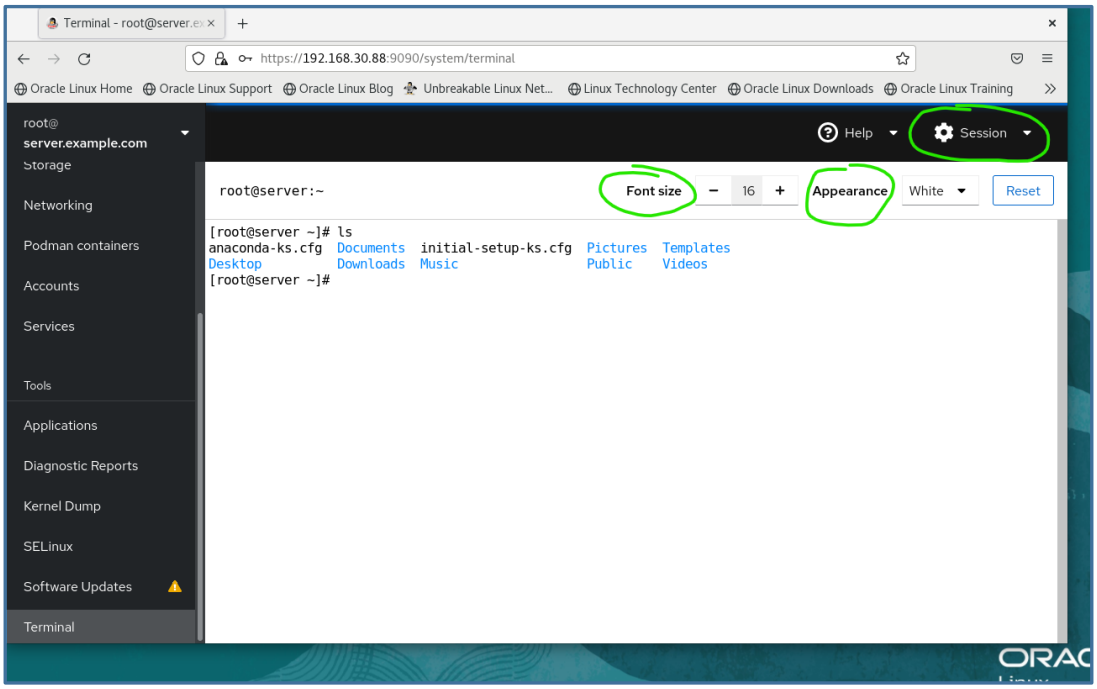
Cockpit web console provides a nice Web based GUI for administrator to carry out common server maintenance and system operation tasks. We will not go into the detail

of these various tasks. We only want to highlight one useful remote access feature, the remote terminal shell.

To start a remote terminal shell, scroll down at the left navigation panel. Find and click on '**Terminal**'.



You will see a terminal appear on the web page.



You may try out some commands at this terminal page. After you have done with the tests. You may just close the browser, or you can find a way to logout from the session.

That's all for cockpit web console.

6. Network Configuration

Overview: Network configuration can be done either by through command line or through GUI. The Network configuration of the Oracle Linux system is managed by the Network Manager Services. Thus, most of the network configuration related commands are started with 'n' and 'm'. The most important one is the 'nmcli' . You will now try some of options that of this command and use them to set static IP address for the client system.

On Client (login as root – open a new terminal to use command line approach):

1. Type “man nmcli” to see the man page for the Network Manager’s Command Line Interface (nmcli)

   Look for the **nmcli connection** and **nmcli device** sections. (These two are commonly used nmcli command options.)

```
       See also nmcli connection monitor and nmcli device monitor to watch for changes in certain devices or
       connections.

CONNECTION MANAGEMENT COMMANDS
       nmcli connection {show | up | down | modify | add | edit | clone | delete | monitor | reload | load |
                         import | export} [ARGUMENTS...]

       NetworkManager stores all network configuration as "connections", which are collections of data (Layer2
       details, IP addressing, etc.) that describe how to create or connect to a network. A connection is
       "active" when a device uses that connection's configuration to create or connect to a network. There may
       be multiple connections that apply to a device, but only one of them can be active on that device at any
       given time. The additional connections can be used to allow quick switching between different networks
       and configurations.

       Consider a machine which is usually connected to a DHCP-enabled network, but sometimes connected to a
       testing network which uses static IP addressing. Instead of manually reconfiguring eth0 each time the
       network is changed, the settings can be saved as two connections which both apply to eth0, one for DHCP
       (called default) and one with the static addressing details (called testing). When connected to the
       DHCP-enabled network the user would run nmcli con up default , and when connected to the static network
```

```
           to standard output.

DEVICE MANAGEMENT COMMANDS
       nmcli device {status | show | set | connect | reapply | modify | disconnect | delete | monitor | wifi |
                     lldp} [ARGUMENTS...]

       Show and manage network interfaces.

       status
           Print status of devices.

           This is the default action if no command is specified to nmcli device.

       show [ifname]
           Show detailed information about devices. Without an argument, all devices are examined. To get
           information for a specific device, the interface name has to be provided.

       set [ifname] ifname [autoconnect {yes | no}] [managed {yes | no}]
           Set device properties.

       connect ifname
           Connect the device. NetworkManager will try to find a suitable connection that will be activated. It
           will also consider connections that are not set to auto connect.
```

   Type 'q' to exit from the man.

2. Type “nmcli device” or “nmcli d” to view the network devices.

```
[root@client ~]#
[root@client ~]# nmcli device
DEVICE  TYPE      STATE      CONNECTION
ens160  ethernet  connected  ens160
lo      loopback  unmanaged  --
[root@client ~]#
```

We are only interested in the 'ethernet' type device. (If you see some additional devices shown, e.g. virbr0, you may have skipped some of the exercises in Lesson 0.)

3. To view more details about of a particular network device, type "nmcli device show <device name> ". In our sample, the device name is 'ens160'.

   Take note of your current IP address, subnet mask, the IPv4 gateway and the IPv4 DNS.

```
[root@client ~]# nmcli device show ens160
GENERAL.DEVICE:                         ens160
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:0C:29:A6:1B:AD
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens160
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.30.130/24
IP4.GATEWAY:                            192.168.30.2
IP4.ROUTE[1]:                           dst = 192.168.30.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 192.168.30.2, mt = 100
IP4.DNS[1]:                             192.168.30.2
IP4.DOMAIN[1]:                          localdomain
IP6.ADDRESS[1]:                         fe80::20c:29ff:fea6:1bad/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024
[root@client ~]#
```

4. A network related command, **ip** , is another useful command, Type "ip address show <device name> will display the current ip address(es) of the particular device. Another network related command, **route**, is also a useful command. Type "**route -n**" is another way to find the current gateway IP. The gateway IP will appear in the Gateway column of the row for “default”. The character “G” will also appear under Flags.

```
[root@client ~]# ip address show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1
000
    link/ether 00:0c:29:a6:1b:ad brd ff:ff:ff:ff:ff:ff
    inet 192.168.30.130/24 brd 192.168.30.255 scope global dynamic noprefixroute ens160
       valid_lft 1346sec preferred_lft 1346sec
    inet6 fe80::20c:29ff:fea6:1bad/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@client ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.30.2    0.0.0.0         UG    100    0        0 ens160
192.168.30.0    0.0.0.0         255.255.255.0   U     100    0        0 ens160
[root@client ~]#
```

This is the IP of the Gateway. You may have a different IP for your Gateway.

5. Under the Network Manager Services, the current DNS settings will be published at the system file /etc/resolv.conf. Thus, to view the current DNS Server settings, type "cat /etc/resolv.conf". Look for the line starting with “nameserver”.

```
[root@client ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search localdomain example.com
nameserver 192.168.30.2
[root@client ~]#
```

This is the IP of the DNS Server. You may have a different IP for your DNS Server.

Take note of the current DNS Server.

It is correct that if your gateway address is the same as your DNS server address. It is because VMWare network is using the sample address to provide two services (gateway and DNS) to the guest VMs.

6. Let's change your client VM to use static IP.

   To set a static IP address for the network interface (in a single line):

```
nmcli connection modify ens160 ipv4.addresses
192.168.30.99/24 ipv4.gateway 192.168.30.2 ipv4.dns
192.168.30.2
```

Set to your current DNS Server

Set to a new IP with the same subnet mask

Set to your current Gateway

7. To change from dynamic IP to static IP address for the network interface:

```
nmcli connection modify ens160 ipv4.method manual
```

(Take note: You may define a static address for your NIC, it will only be in effect if the current ipv4.method is set to 'manual' – the other choice of ipv4.method is 'auto')

```
[root@client ~]# nmcli connection modify ens160 ipv4.addresses 192.168.30.99/24 ipv4.gateway 192.168.30.2 ipv4.dns
192.168.30.2
[root@client ~]# nmcli connection modify ens160 ipv4.method manual
[root@client ~]#
```

8. At this point if you run the ip address show ens160 you will see the following:

```
[root@client ~]# ip address show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:fb:76:13 brd ff:ff:ff:ff:ff:ff
    inet 192.168.30.128/24 brd 192.168.30.255 scope global dynamic noprefixroute ens160
       valid_lft 1402sec preferred_lft 1402sec
    inet6 fe80::20c:29ff:fefb:7613/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@client ~]#
```

The current ip address remains unchanged !

9. For the changes to take effect, disconnect and connect the network device:

```
nmcli device disconnect ens160
nmcli device connect ens160
```

```
[root@client ~]# nmcli device disconnect ens160
Device 'ens160' successfully disconnected.
[root@client ~]# nmcli device connect ens160
Device 'ens160' successfully activated with '115487a8-005c-4998-9a14-5cc0e6e0c088'.
[root@client ~]#
```

10. You may use 'nmcli d show ens160' or simply type "ip addr show ens160" to verify your current IP address and Subnet mask of ens160 has been updated.

```
[root@client ~]# nmcli device show ens160
GENERAL.DEVICE:                         ens160
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:0C:29:FB:76:13
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens160
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/4
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.30.99/24
IP4.GATEWAY:                            192.168.30.2
IP4.ROUTE[1]:                           dst = 192.168.30.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 192.168.30.2, mt = 100
IP4.DNS[1]:                             192.168.30.2
IP6.ADDRESS[1]:                         fe80::20c:29ff:fefb:7613/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255
[root@client ~]#
```

11. View the file `/etc/sysconfig/network-scripts/ifcfg-ens160` to see the network settings that you just made.

    Note : The DNS Server setting is stated in the file /etc/sysconfig/network-scripts/ifcfg-ens160, it will override the settings in the /etc/resolv.conf.

12. Type the following and check if your ip address is changed back to dynamic.

```
nmcli connection modify ens160 ipv4.method auto
nmcli connection delete ens160
nmcli device connect ens160
```

Note: To change from dynamic IP to static IP only requires a connection reset. To change from static IP back to dynamic IP (and erase the static settings) requires restarting the NIC (using nmcli connection delete <connection name> and nmcli device connect <device name>).

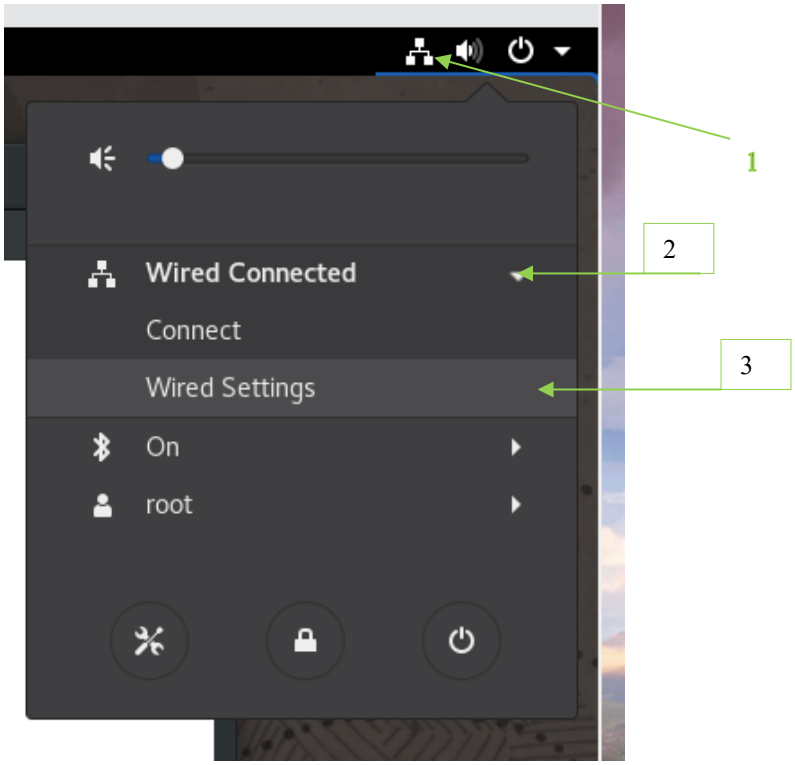
```
[root@client ~]# nmcli connection modify ens160 ipv4.method auto
[root@client ~]# nmcli connection delete ens160
Connection 'ens160' (01f9bb6b-f636-40b9-97ef-67d3ca440277) successfully deleted.
[root@client ~]# nmcli device connect ens160
Device 'ens160' successfully activated with '3e150f2c-8eff-44b3-84a4-36705a77dd2d'.
[root@client ~]#
```

13. Now we will try to set the Client to use Static IP one more time via the GUI approach.

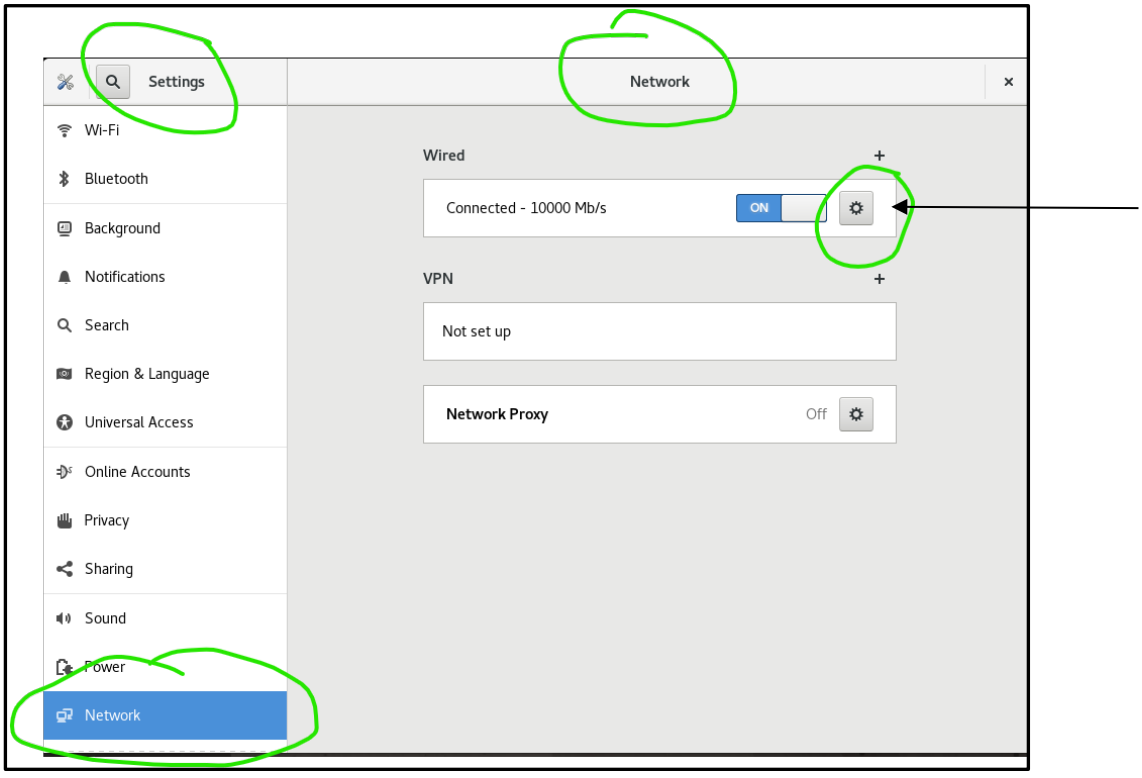
<u>On client (login as root):</u>

From the GUI, click on the Network Connection icon ( ) in the **top right-hand corner**, click to expand the Wired Connected section and finally click on the Wired Settings to

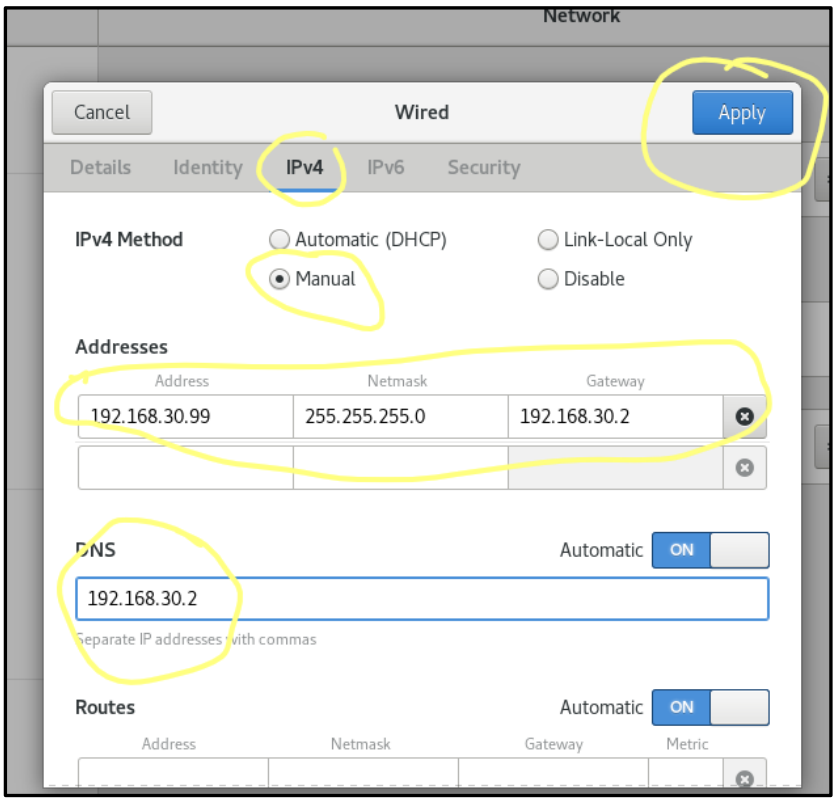
bring up the Network setting menu.



14. At the Network Settings Menu, click on the gear wheel to configure the Wired connection.



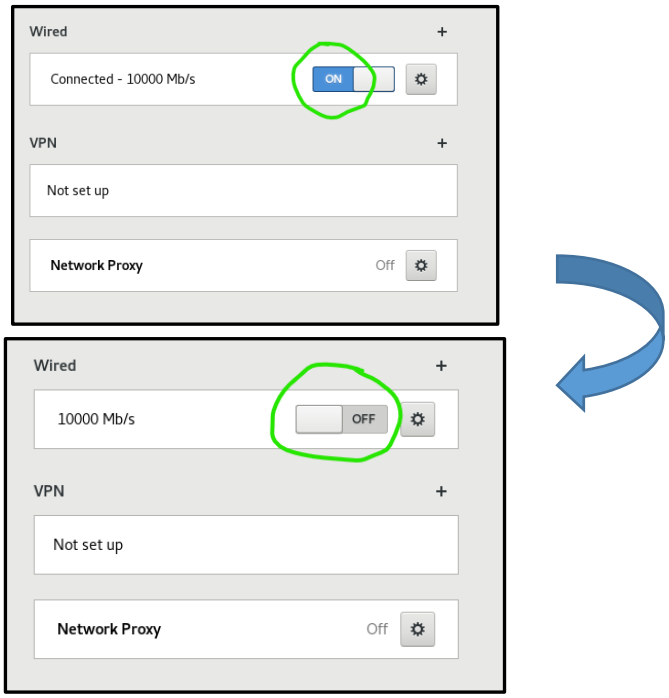
15. Explore the GUI (The IPv4 Tab) on how to set your client to toggle between using static IP and dynamic IP.



Apply the new settings in two steps:

a. Press the Apply button to save and exit from the setting menu.

b. Must toggle the connection state (on to off then off to on) by clicking the on off switch button:



In each case, you need to check for two operations:

1) Your server can still ping to your client and vice versa. (To prove the LAN connection is working)
2) Your client can ping www.google.com (To prove the DNS and Gateway is working)

Hints: The 192.168.30.99/24 combined IP/Netmask notation have to be broken into the traditional IP and Netmask notation: 192.168.30.99 , 255.255.255.0.

Seek for help from your tutor if you have trouble to configure your network settings.

7. Kernel parameters

Description: The Linux kernel parameters can be used to change some core resources allocation or configuration settings of the Linux kernel, for example, to prevent the Linux kernel from responding to ping packets.

On server (login as root) :

1. View the list of available kernel parameters and their current values.

```
sysctl -a
```

2. View the list of available kernel parameters associated with ICMP (Internet Control Message Protocol)

```
sysctl -a | grep icmp
```

```
[root@server ~]# sysctl -a | grep icmp
net.ipv4.icmp_echo_ignore_all = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_errors_use_inbound_ifaddr = 0
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.icmp_msgs_burst = 50
net.ipv4.icmp_msgs_per_sec = 1000
net.ipv4.icmp_ratelimit = 1000
net.ipv4.icmp_ratemask = 6168
net.ipv6.icmp.echo_ignore_all = 0
net.ipv6.icmp.echo_ignore_anycast = 0
net.ipv6.icmp.echo_ignore_multicast = 0
net.ipv6.icmp.ratelimit = 1000
net.ipv6.icmp.ratemask = 0-1,3-127
net.netfilter.nf_conntrack_icmp_timeout = 30
net.netfilter.nf_conntrack_icmpv6_timeout = 30
[root@server ~]#
```

3. Set the kernel parameter so that the Linux kernel will ignore ipv4 ping packets.

```
sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

(Note: This is not related to the firewall setting. This feature is implemented at the kernel level. Ignore ping can reduce the attack surface of the system.)

On client:

4. Try to ping the server from the client. You should not be successful.

```
time ping -c 3 <Server IP>
```

```
[root@client ~]# time ping -c 3 192.168.30.88
PING 192.168.30.88 (192.168.30.88) 56(84) bytes of data.

--- 192.168.30.88 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2070ms


real    0m12.104s
user    0m0.000s
sys     0m0.007s
[root@client ~]#
```

Take note that, in Linux, we can measure the execution time of a command with the time utility. As shown at the above, ping issues icmp_echo_request datagram to the target in a 1 second interval continuously. The -c option, will limit the number of requests. For each request, the default time out is 3 seconds. That's matches with the output from the 'time' utility: The real time took about 12.104 seconds. And the actual CPU time spent (user code + sys code) is less than 0.007 seconds (7 milliseconds).

On server:

5. Let's try another way to update the kernel parameter. Edit the file /etc/sysctl.conf and set the kernel parameter back to accepting icmp_echo_request. (Add the entry if it does not already exist in the file)

```
net.ipv4.icmp_echo_ignore_all=0
```

6. Load the settings from /etc/sysctl.conf.

```
sysctl -p
```

On client:

7. Try to ping the server from the client. You should be successful this time.

```
[root@client ~]# time ping -c 3 192.168.30.88
PING 192.168.30.88 (192.168.30.88) 56(84) bytes of data.
64 bytes from 192.168.30.88: icmp_seq=1 ttl=64 time=0.965 ms
64 bytes from 192.168.30.88: icmp_seq=2 ttl=64 time=0.800 ms
64 bytes from 192.168.30.88: icmp_seq=3 ttl=64 time=0.460 ms

--- 192.168.30.88 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.460/0.741/0.965/0.212 ms

real    0m2.011s
user    0m0.002s
sys     0m0.003s
[root@client ~]#
```

When there is no timeout involved, the command be completed in slightly more than 2 seconds. Since the user code has to do the output it took up 2 milliseconds.

Extra reference:

a. https://docs.oracle.com/en/operating-systems/oracle-linux/7/security/E54670.pdf (Check out section 2.8 from the above to learn a few more security related Kernel

Parameters settings.)

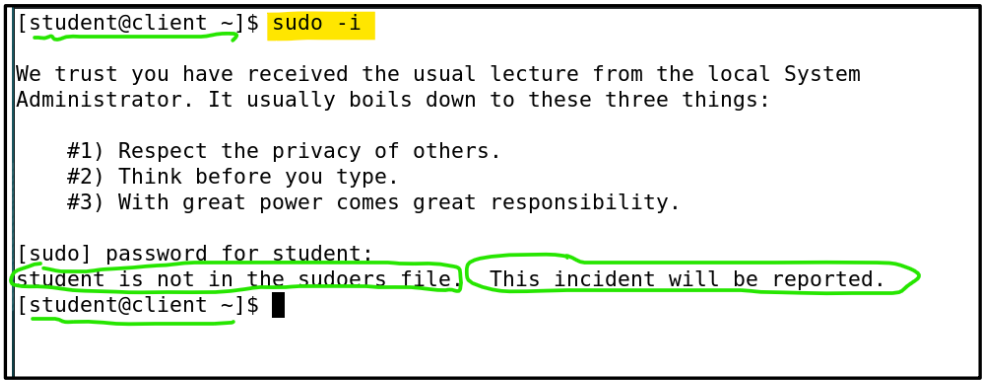
b. https://docs.oracle.com/en/operating-systems/oracle-linux/8/security/hardening-guidelines.html

(Check out the above security guidelines for kernel parameters and many other recommendations.)

8. Prevent root login

Administrators may want to prevent users from logging in directly as root. This is to prevent any of the root account holders accidentally entering wrong commands or configurations. Typical, the system administrators' logon to the system using accounts with normal privilege. If they need to carry out any administrative tasks, they need to use “sudo” command to escalate their privilege temporarily.
*However, not all the user account can run the sudo command.

On you client:

1. Login as ‘student’, open a terminal session and run '**sudo -i**' to check if it is allowed or not.
2. 

   The above shown the student account is not allowed to run sudo.
3. Allow student to run administrator tasks using sudo.
4. Login as user root, (or using '**su - root**'), edit the /etc/sudoers filer or type "visudo" (which will start a vi session with /etc/sudoers file pre-loaded).

   Add the following line to the end of the file.

   student ALL=ALL

```
  GNU nano 2.9.8                    /etc/sudoers                    Modified


## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
student  ALL=ALL
```

5. Save your changes and exit the editor.
6. Now login as student and check if sudo is allowed now

   sudo ls

```
[student@client ~]$ sudo ls
[sudo] password for student:
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[student@client ~]$
```

The above shows that student can run a command with sudo now. Apparently, the above is not a good example, as ls command itself does require root privilege to run.

7. Let's use sudo to enable student to have the root privileges. Type sudo -i :

```
[student@client ~]$ sudo -i
[root@client ~]# ps
    PID TTY          TIME CMD
   2730 pts/0    00:00:00 sudo
   2732 pts/0    00:00:00 bash
   2758 pts/0    00:00:00 ps
[root@client ~]# ps -t pts/0
    PID TTY          TIME CMD
   2475 pts/0    00:00:00 bash
   2730 pts/0    00:00:00 sudo
   2732 pts/0    00:00:00 bash
   2759 pts/0    00:00:00 ps
[root@client ~]#
```

Bash session with root privileges. (Started by sudo -i)

The original bash session for student login.

The above shows that sudo -i actually start a new bash session and associate this bash session with root account.

8. With sudo, we do not need to login to the root account anymore. Let's disable the root account to disallow it to be used for login.

   Remain in the sudo/root enabled bash. Edit the file /etc/passwd and edit the root account to have a non-interactive shell (change highlighted in bold). This will disallow the root for direct login.

   `root:x:0:0:root:/root:`**`/sbin/nologin`**

```
root:x:0:0:root:/root:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

9. Exit from the sudo mode (Type exit.) and Log off from the system.Try to login as root. You should not be successful.
10. Login as student. Type “su -“ and type root’s password. You should not be successful neither. (su is not allowed to use to substitute as the root account).

```
[student@client ~]$ su - root
Password:
This account is currently not available.
[student@client ~]$
```

11. The only way to run admin tasks is to use sudo. Type "sudo nano /etc/passwd" or "sudo vi /etc/passwd" to change /etc/passwd and edit the root account to have an interactive shell again (change highlighted in bold).

    `root:x:0:0:root:/root:`**`/bin/bash`**

```
 GNU nano 2.9.8                    /etc/passwd                         Modified

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
geoclue:x:997:995:User for geoclue:/var/lib/geoclue:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line
```

12. Test that you can su to root. You should be successful now.

<u>On Server:</u>

13. Login as root and enable sudo for the user account 'student'.

Note: In production systems, we shall avoid logging on using root account directly.

9. Anonymous access to the vsftpd service

Here we will start to revisit the basic configurations of the vsftpd services.

On server:

1. Use one of the following commands to check if the vsftpd package is already installed. (It should be if you have completed lesson 0 properly)

```
rpm -qa | grep vsftpd
dnf info vsftpd
```

2. Install the vsftpd service if it is not installed yet

```
dnf install vsftpd
```

3. Check if the vsftpd service will be automatically started on bootup.

```
systemctl is-enabled vsftpd
```

4. Set the vsftpd service to be automatically started on every bootup if it is not enabled. Remember to start it too.

```
systemctl enable vsftpd
systemctl start vsftpd
```

5. View a list of all the services (installed only) and the output will show their states.

```
systemctl list-unit-files --type service
```

```
UNIT FILE                                   STATE
accounts-daemon.service                     enabled
alsa-restore.service                        static
alsa-state.service                          static
anaconda-direct.service                     static
anaconda-fips.service                       static
anaconda-nm-config.service                  static
anaconda-noshell.service                    static
anaconda-pre.service                        static
anaconda-shell@.service                     static
anaconda-sshd.service                       static
anaconda-tmux@.service                      static
anaconda.service                            static
arp-ethers.service                          disabled
atd.service                                 enabled
auditd.service                              enabled
auth-rpcgss-module.service                  static
autovt@.service                             enabled
avahi-daemon.service                        enabled
blivet.service                              static
blk-availability.service                    disabled
bluetooth.service                           enabled
```

The three possible states of an installed service: enabled, disabled, or static. 'static' service is a dependency of an enabled service.

6. We can narrow down the list to show only enabled services with the '|' and grep trick.

```
systemctl list-unit-files --type service | grep enabled
```

```
[root@server ~]# systemctl list-unit-files --type service | grep enabled
accounts-daemon.service                    enabled
atd.service                                enabled
auditd.service                             enabled
autovt@.service                            enabled
avahi-daemon.service                       enabled
bluetooth.service                          enabled
chronyd.service                            enabled
crond.service                              enabled
cups.service                               enabled
dbus-org.bluez.service                     enabled
dbus-org.fedoraproject.FirewallD1.service  enabled
dbus-org.freedesktop.Avahi.service         enabled
dbus-org.freedesktop.ModemManager1.service enabled
dbus-org.freedesktop.nm-dispatcher.service enabled
dbus-org.freedesktop.timedate1.service     enabled
display-manager.service                    enabled
firewalld.service                          enabled
gdm.service                                enabled
```

: : :

```
smartd.service                             enabled
sshd.service                               enabled
sssd.service                               enabled
syslog.service                             enabled
systemd-pstore.service                     enabled
timedatex.service                          enabled
tuned.service                              enabled
udisks2.service                            enabled
vdo.service                                enabled
vgauthd.service                            enabled
vmtoolsd.service                           enabled
vsftpd.service                             enabled
[root@server ~]#
```

On server (Login as student)

7. Now, we should have the vsftpd running on the server. The default files download folder is set as at /var/ftp/pub directory.

   You may create a new testing text file `/var/ftp/pub/ftpserver1.txt` by cloning the content from the /etc/passwd file. (You may need sudo or login as root if the operations require root privileges. You should know how to enable sudo for student by now.)

```
[student@server ~]$ cd /var/ftp/pub
[student@server pub]$ pwd
/var/ftp/pub
[student@server pub]$ cp /etc/passwd ftpserver1.txt
cp: cannot create regular file 'ftpserver1.txt': Permission denied
[student@server pub]$ sudo cp /etc/passwd ftpserver1.txt

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for student:
[student@server pub]$ ls
ftpserver1.txt  local_repo
[student@server pub]$
```

8. Edit the vsftp configuration file /etc/vsftpd/vsftpd.conf. Allow anonymous access to your vsftpd service by setting the following in the config file. (If it is not existed yet.)

   ```
   anonymous_enable=YES
   ```

9. Start the vsftpd service if it is not running yet.

```
systemctl start vsftpd
```

On client: (login as student)

10. Install the ftp client program by using sudo or su -

```
su -
dnf install ftp -y
exit
```

```
[root@client ~]# dnf install ftp -y
Last metadata expiration check: 21:39:01 ago on Sun 18 Sep 2022 06:00:38 PM +08.
Dependencies resolved.
================================================================================
 Package     Architecture   Version             Repository                  Size
================================================================================
Installing:
 ftp         x86_64         0.17-78.el8         las_ol8_appstream           70 k

Transaction Summary
================================================================================
Install  1 Package
```

: : : : : :

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : ftp-0.17-78.el8.x86_64                                 1/1
  Running scriptlet: ftp-0.17-78.el8.x86_64                                 1/1
  Verifying        : ftp-0.17-78.el8.x86_64                                 1/1

Installed:
  ftp-0.17-78.el8.x86_64

Complete!
[root@client ~]#
```

(In case if the 'su -' is not working you must be forgotten to re-enable the root login in the client VM.)

11. Type "cd" to change back to the default directory of student. Run the ftp client program to connect to your server.

```
cd
pwd
ftp serverIP
```

12. Note the version number of the FTP service on the server.
13. Enter “anonymous” for the username. Press the 'Enter' key when asked for the password.

```
[student@client ~]$ pwd
/home/student
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 (vsFTPd 3.0.3)
Name (192.168.30.88:student): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Upon the successful login as an anonymous user, you will be entering to the ftp client session shell.

14. Type “help” to view the available commands in the ftp prompt.
15. Type "pwd" to check your current working directory.
16. Type "ls" to view the directory listing of the ftp server.
17. Type "cd pub" to enter the pub directory of the ftp server.
18. Type "ls" to view the directory listing.
19. If you can see the file "ftpserver1.txt" type “get ftpserver1.txt” to download the file.
20. Type "bye" or "quit" or "exit" to exit the FTP client shell.
21. Check that the “ftpserver1.txt” file has been downloaded to your client.

Task :

On server

Change the ftpd_banner settings in the /etc/vsftpd/vsftpd.conf file so that the version number of the FTP service is not displayed when clients connect to it. Remember to restart the FTP service after changing the configuration file.

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

The banner message is customized, and the ftp server version number is hidden.

ChallengingTask : (Try to work on it only after your complete the entire practical)

On server

Change the single line banner message to a multiple line message similar to the sample shown below:

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220-__        __   _                            _          _   _
220-\ \      / /__| | ___ ___  _ __ ___   ___  | |_ ___   | |_| |__   ___
220- \ \ /\ / / _ \ |/ __/ _ \| '_ ` _ \ / _ \ | __/ _ \  | __| '_ \ / _ \
220-  \ V  V /  __/ | (_| (_) | | | | | |  __/ | || (_) | | |_| | | |  __/
220-   \_/\_/ \___|_|\___\___/|_| |_| |_|\___|  \__\___/   \__|_| |_|\___|
220-
220- ____ _____ ____  _  _   _ ____    _____ _____ ____
220-/ ___|_   _|___ \| || | / |___ \  |  ___|_   _|  _ \
220-\___ \ | |   __) | || |_| | __) | | |_    | | | |_) |
220- ___) || |  / __/|__   _| |/ __/  |  _|   | | |  __/
220-|____/ |_| |_____|  |_| |_|_____| |_|     |_| |_|
220-
220- ____
220-/ ___|  ___ _ ____   _____ _ __
220-\___ \ / _ \ '__\ \ / / _ \ '__|
220- ___) |  __/ |   \ V /  __/ |
220-|____/ \___|_|    \_/ \___|_|
220-
220
Name (192.168.30.88:student): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

(Hints: oracle-epel-release-el8, figlet)

10. chroot vsftpd users to their home directories

For security concern, an anonymous user account is only allow to access files / folders under the /var/ftp/ path. To ensure this, vsftpd will apply a technique, change root (chroot), to map the top level folder from the "/" to "/var/ftp/". Therefore, the 'root' in this context is referring to the root of the file system.

Based on the similar concept to tighten the security measure, we will chroot named vsftpd user login sessions to their own home directories. In this way, a successful ftp connection with named user id will set the initial folder at the home directory of the corresponding user. Subsequently, it is not possible for the user to use the cd command to change to any other folder. This configuration limits the ftp users to stay within the sub-folders of their own home directories.

<u>On client:</u>

1. As user student, do a ftp to your server. Verify if the ftp server allows normal user logon:

```
ftp <serverIP>
```

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): student
530 This FTP server is anonymous only.
Login failed.
ftp>
```

Based on our configurations, you should encounter an issue as shown at the above.
The current vsftpd server configurations only allows anonymous login.
you need to update the ftp server configuration at your server to enable named user account to login to the vsftd:

On server:

Login as root, edit the config file /etc/vsftpd/vsftpd.conf, set the 'local_enable' option to 'Yes', this setting enables local user account logon.

```
  GNU nano 2.9.8                    vsftpd.conf                         Modified

# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
#anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

You also need to restart the vsftpd service to let the new configuration takes effect:

```
systemctl restart vsftpd
```

On Client

Try again at the client side to verify the ftp server is now allowing local user account login:

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

On Server

2. As user student (you can use su - student), create a sub folder, name it as XXtestdir*, in the /home/student folder.

   (*XX is your initial, in the following example, the new folder is named as KKtestdir.)

```
[root@server ~]# su student
[student@server root]$ pwd
/root
[student@server root]$ cd
[student@server ~]$ exit
exit
[root@server ~]# su - student
[student@server ~]$
[student@server ~]$ pwd
/home/student
[student@server ~]$ mkdir KKtestdir
[student@server ~]$ ls
Desktop    Downloads  Music     Public     Videos
Documents  KKtestdir  Pictures  Templates
[student@server ~]$
```

On client:

3. Do a ftp to your server. Login as student.
4. Type “pwd” to view the present working directory.
5. Type “ls” to view a listing of the current directory.

   This is to verify that your default current directory is /home/student.
6. Type “cd /etc” to go to the /etc directory.

   This is to verify that you can traverse outside your home directories and explore the entire system file system. (Which is what an adversary would like to do.)
7. Type “ls” to view a listing of the /etc/ directory on your server. Currently user student can view the whole file system on the server.

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/home/student" is the current directory
ftp> ls
227 Entering Passive Mode (192,168,30,88,251,72).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Desktop
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Documents
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Downloads
drwxrwxr-x    2 1000     1000            6 Sep 16 12:23 KKtestdir
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Pictures
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Public
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Templates
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Videos
226 Directory send OK.
ftp> cd /etc
250 Directory successfully changed.
ftp>
```

: : : : :

```
ftp> ls
227 Entering Passive Mode (192,168,30,88,173,43).
150 Here comes the directory listing.
-rw-r--r--    1 0        0            4536 Aug 20  2020 DIR_COLORS
-rw-r--r--    1 0        0            5214 Aug 20  2020 DIR_COLORS.256color
-rw-r--r--    1 0        0            4618 Aug 20  2020 DIR_COLORS.lightbgcolor
-rw-r--r--    1 0        0              94 Mar 15  2019 GREP_COLORS
drwxr-xr-x    2 0        0              76 Aug 29 07:58 PackageKit
drwxr-xr-x    2 0        0              25 Aug 29 07:54 UPower
drwxr-xr-x    7 0        0             121 Aug 29 07:56 X11
drwxr-xr-x    3 0        0              65 Aug 29 07:58 alsa
drwxr-xr-x    2 0        0            4096 Aug 29 07:58 alternatives
drwxr-xr-x    4 0        0              58 Aug 29 07:57 anaconda
-rw-r--r--    1 0        0             541 Nov 11  2019 anacrontab
-rw-r--r--    1 0        0              55 Apr 07 19:17 asound.conf
-rw-r--r--    1 0        0               1 Mar 07  2019 at.deny
drwxr-xr-x    3 0        0             228 Aug 29 08:02 authselect
drwxr-xr-x    4 0        0              71 Aug 29 07:56 avahi
drwxr-xr-x    2 0        0             150 Sep 16 09:40 bash_completion.d
-rw-r--r--    1 0        0            3019 Jul 31  2020 bashrc
-rw-r--r--    1 0        0             429 Sep 13 20:07 bindresvport.blacklist
drwxr-xr-x    2 0        0               6 Aug 10 15:47 binfmt.d
-rw-r-----    1 0        985            33 Aug 29 07:56 brlapi.key
```

Now we shall try to tighten the system security by limiting the named ftp users to stay within their own home directories.

There is an interesting term to refer to this: 'ftp chroot jail'.

<u>On server:</u>

8. Edit /etc/vsftpd/vsftpd.conf (as root, or use sudo) to enable the chroot jail for all local users (connection made by local user login).

```
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
```

In the above setting. All local users will be chroot to theirs home directories (jailed there).

Since the chroot_list_enable is YES, any user ID listed in the chroot_list_file will be exempted from the jail.

9. Add the following two lines to the end of the /etc/vsftpd/vsftpd.conf.

```
allow_writeable_chroot=YES
passwd_chroot_enable=YES
```

> Note:
> allow_writeable_chroot=YES is not recommended for an FTP server that with 'write_enable=YES' setting. It is okay for our case, as we do not allow file upload to the server from the client.
> passwd_chroot_enable=YES implies the user must change to the home directory defined at the /etc/passwd entry as the root directory. I.e the user will be jailed in there.

10. Create the file /etc/vsftpd/chroot_list and include peter and paul into this file.

```
peter
paul
```

With the above setting. All users will be jailed in their home directories except peter and paul. (ie. student will be jailed but peter and paul are free to traverse to any folder within the filesystem.)

11. Now Type :

```
sudo systemctl restart vsftpd
```

to restart your vsftpd service to verify the effect.

<u>On client:</u>

12. Do an ftp connection to your server. Login as student.
13. Type "pwd" to view the present working directory.

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student):
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is the current directory
ftp> ls
227 Entering Passive Mode (192,168,30,88,56,163).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Desktop
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Documents
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Downloads
drwxrwxr-x    2 1000     1000            6 Sep 16 12:23 KKtestdir
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Pictures
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Public
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Templates
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Videos
226 Directory send OK.
```

The client is now taking /home/student as the '/' (root), so there is no access to any

other part of the file system.

14. Type "cd /etc" to try to go to the root directory. You should not be successful now.

    We call this a jailed ftp session. (limit ftp users to only goes to their home directory)

```
ftp> ls
227 Entering Passive Mode (192,168,30,88,56,163).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Desktop
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Documents
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Downloads
drwxrwxr-x    2 1000     1000            6 Sep 16 12:23 KKtestdir
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Pictures
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Public
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Templates
drwxr-xr-x    2 1000     1000            6 Aug 29 08:28 Videos
226 Directory send OK.
ftp> cd /etc
550 Failed to change directory.
ftp>
```

15. Quit from the ftp connection. Start a new ftp connection to your server. Login as peter.
16. Try to verify that peter can cd to any folder, including /etc.

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): peter
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/home/peter" is the current directory
ftp> ls
227 Entering Passive Mode (192,168,30,88,243,157).
150 Here comes the directory listing.
226 Directory send OK.
ftp> cd /etc
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (192,168,30,88,44,128).
150 Here comes the directory listing.
-rw-r--r--    1 0        0            4536 Aug 20  2020 DIR_COLORS
-rw-r--r--    1 0        0            5214 Aug 20  2020 DIR_COLORS.256color
-rw-r--r--    1 0        0            4618 Aug 20  2020 DIR_COLORS.lightbgcolor
-rw-r--r--    1 0        0              94 Mar 15  2019 GREP_COLORS
drwxr-xr-x    2 0        0              76 Aug 29 07:58 PackageKit
drwxr-xr-x    2 0        0              25 Aug 29 07:54 UPower
drwxr-xr-x    7 0        0             121 Aug 29 07:56 X11
drwxr-xr-x    3 0        0              65 Aug 29 07:58 alsa
```

17. <u>On Server:</u>

    Edit /etc/vsftpd/vsftpd.conf (as root, or use sudo) to disable the chroot jail for any local user, by setting the chroot_local_user to NO.

```
chroot_local_user=NO
```

    Interestingly, the user accounts listed in the /etc/vsftpd/chroot_list will now become the users that will be chroot jailed!

    Restart your vsftpd.

18. <u>On Client:</u>

Repeat the previous tests made. To verify that peter is jailed while student is not.

```
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/home/student" is the current directory
ftp> cd /etc
250 Directory successfully changed.
ftp> bye
221 Goodbye.
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): peter
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is the current directory
ftp> cd /etc
550 Failed to change directory.
ftp>
```

Conclusions: The chroot jail feature is highly recommended for a vsftpd setup. To disable chroot jail, you need to comment out the following three options in the vsftpd.conf file -

```
#chroot_local_user=YES
#chroot_list_enable=YES
#chroot_list_file=/etc/vsftpd/chroot_list
```

11. SELinux basics

The 'SE' in SELinux stands for Security-Enhanced Linux. IT is Standard Linux access control, owner/group + permission flags like rwx, is often called Discretionary Access Control (DAC). Security Enhanced Linux (SELinux) is a parallel security enforcement model. SELinux is basically a labelling system. There are four types of labels: User, Role, Type and Level. With SELinux enabled, an application must be allowed by BOTH SELinux and DAC to do certain activities.
(Recommended ref: https://opensource.com/business/13/11/selinux-policy-guide)

SELinux is an optional feature in a Linux system. It is commonly enabled for production systems that offer Enterprise level services.

On server:

1. Check the general SELinux status.

   sestatus

```
[student@server ~]$ sudo sestatus
[sudo] password for student:
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      31
[student@server ~]$
```

2. Check which SELinux mode the system is in.

   getenforce

3. Set the system to be in Permissive mode.

   setenforce 0

4. Set the system to be in Enforcing mode.

   setenforce 1

5. The setenforce command only change the current SELinux mode at the runtime. To configure the new mode setting for permanent effect. You need to edit the file /etc/selinux/config file to configure the SELinux mode to the target setting. Then the effect will be applied upon the next bootup.

Note:

SELinux has three modes:

Enforcing: SELinux allows access based on SELinux policy rules.

Permissive: SELinux only logs actions that would have been denied if running in enforcing mode.

Disabled: No SELinux policy is loaded.

The true purpose for Permissive mode is that it still logs what it would have denied and as such allows the administrator/developer to get a sense of what would happen if he switched the system from permissive to enforcing mode. In a production system, the system should be booted with enforcing mode and it is not allowed to be changed to permissive mode.

ref: https://wiki.gentoo.org/wiki/SELinux/Tutorials/Permissive_versus_enforcing#Permissive_versus_enforcing

6. View the first 15 SELinux booleans.

```
getsebool -a | head -15
```

```
[student@server ~]$ sudo getsebool -a | head -15
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
auditadm_exec_content --> on
authlogin_nsswitch_use_ldap --> off
authlogin_radius --> off
authlogin_yubikey --> off
awstats_purge_apache_log_files --> off
boinc_execmem --> on
cdrecord_read_content --> off
cluster_can_network_connect --> off
cluster_manage_all_files --> off
cluster_use_execmem --> off
[student@server ~]$
```

What is SELinux Booleans for ?

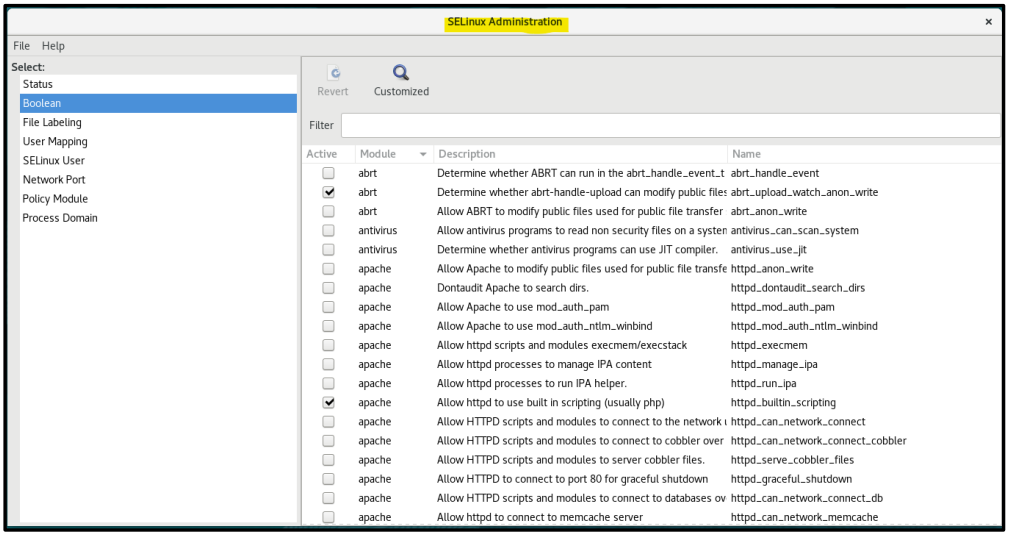
The running behavior of the SELinux Labeling requirement is based on the current activated SELinux policy. The policy itself can be customized by enabling or disabling a set of policy Booleans. Booleans allow parts of SELinux policy to be changed at **run time**, without any knowledge of SELinux policy writing. This allows changes without reloading or recompiling SELinux policy.

Ref: https://www.thegeekdiary.com/understanding-selinux-booleans/

7. To install the SELinux GUI:

```
dnf install policycoreutils-gui
```

8. Login as root. Run “system-config-selinux” to see the mode and the boolean settings.



Close the windows after exploring the list of SELinux Booleans.

9. Open a terminal. View the SELinux file contexts of / directory.

```
ls -lZ /
```

10. View the SELinux file contexts of /var/log directory.

```
ls -lZ /var/log
```

11. View the SELinux file contexts of /var/ftp/pub directory.

```
ls -lZ /var/ftp/pub
```

```
[root@server ~]# ls -lZ /var/ftp/pub
total 8
-rw-r--r--. 1 root root unconfined_u:object_r:public_content_t:s0 2676 Sep 14 10:04 ftpserver1.txt
dr-xr-xr-x. 7 root root unconfined_u:object_r:public_content_t:s0 4096 May 25 01:18 local_repo
[root@server ~]#
```

12. Change the file context of the file /var/ftp/pub/ftpserver1.txt to a wrong file context.

```
chcon -t shadow_t /var/ftp/pub/ftpserver1.txt
```

```
[root@server ~]# chcon -t shadow_t /var/ftp/pub/ftpserver1.txt
[root@server ~]# ls -lZ /var/ftp/pub
total 8
-rw-r--r--. 1 root root unconfined_u:object_r:shadow_t:s0         2676 Sep 14 10:04 ftpserver1.txt
dr-xr-xr-x. 7 root root unconfined_u:object_r:public_content_t:s0 4096 May 25 01:18 local_repo
[root@server ~]#
```

On client:

13. Connect to the FTP server and login as anonymous user. Try to download the ftpserver1.txt file again. You will not be successful this time.

On server:

14. Try to set the current SELinux mode to permissive.

On client:

15. Connect to the FTP server and login as anonymous user. Try to download the ftpserver1.txt file again. This time you will be successful as SELinux is no longer blocking it.

On server:

16. Set the SELinux mode back to enforcing.

17. Reset back the correct file context of the file /var/ftp/pub/ftpserver1.txt

```
restorecon /var/ftp/pub/ftpserver1.txt
```

18. View the file contexts of /var/ftp/pub directory.

```
ls -lZ /var/ftp/pub
```

```
[root@server ~]# restorecon /var/ftp/pub/ftpserver1.txt
[root@server ~]# ls -lZ /var/ftp/pub
total 8
-rw-r--r--. 1 root root unconfined_u:object_r:public_content_t:s0 2676 Sep 14 10:04 ftpserver1.txt
dr-xr-xr-x. 7 root root unconfined_u:object_r:public_content_t:s0 4096 May 25 01:18 local_repo
[root@server ~]#
```

19. Watch this video: https://www.youtube.com/watch?v=tXNr3gOgrn8 to wrap up our learning on SELinux basics.

12. Configuring SELinux to allow anonymous users to upload files to vsftpd server

Normally anonymous users should not be allowed to upload files to an FTP server (what if they upload virus-infected files or Trojans?) If anonymous upload is required, then do not

allow the upload directory to be read by anyone. This will prevent anyone from downloading files that have been uploaded by someone else (to prevent the spreading of malicious files).

On server:

1. Login as root. Create a directory to hold the uploaded files.

```
mkdir -p /var/ftp/incoming
```

2. Change the group owner of the directory to the group ftp.

```
chgrp ftp /var/ftp/incoming
```

```
[root@server ~]# cd
[root@server ~]# mkdir -p /var/ftp/incoming
[root@server ~]# chgrp ftp /var/ftp/incoming/
[root@server ~]# ls -l /var/ftp/incoming/
total 0
[root@server ~]# ls -l /var/ftp
total 0
drwxr-xr-x. 2 root ftp   6 Sep 21 10:52 incoming
drwxr-xr-x. 3 root root 46 Sep 14 10:04 pub
[root@server ~]#
```

3. Set the permissions of the directory to allow full access for the owner root, write and execute by the group ftp and no access for everyone else.

```
chmod 730 /var/ftp/incoming
```

```
or
chmod u=rwx,g=wx,o= /var/ftp/incoming
```

```
[root@server ~]# chmod u=rwx,g=wx,o= /var/ftp/incoming/
[root@server ~]# ls -l /var/ftp
total 0
drwx-wx---. 2 root ftp   6 Sep 21 10:52 incoming
drwxr-xr-x. 3 root root 46 Sep 14 10:04 pub
[root@server ~]#
```

4. Open and view the /etc/vsftpd/vsftpd.conf file and look for the part contains the following configuration line.

```
anon_upload_enable=YES
```

```
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_f
ull_access
anon_upload_enable=YES
#
```

From the given information found, you need to :

- Uncomment the following configurations:
  - write_enable=YES, anon_upload_enable=YES

- Set the following SELinux Booleans to True
  - Allow_ftpd_anon_write, allow_ftpd_full_access

5. Uncomment the two required configruations from the /etc/vsftpd/vsftpd.conf file.

```
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_f
ull_access
anon_upload_enable=YES
#
```

6. Set the two sebool to True using the setsebool command:

```
setsebool -P allow_ftpd_anon_wrtie True
setsebool -P allow_ftpd_full_access True
```

Or

```
setsebool -P allow_ftpd_anon_write=1 allow_ftpd_full_access=1
```

```
[root@server ~]# setsebool -P ftpd_anon_write=1 allow_ftpd_full_access=1
[root@server ~]# getsebool -a | grep ftpd
ftpd_anon_write --> on
ftpd_connect_all_unreserved --> off
ftpd_connect_db --> off
ftpd_full_access --> on
ftpd_use_cifs --> off
ftpd_use_fusefs --> off
ftpd_use_nfs --> off
ftpd_use_passive_mode --> off
[root@server ~]#
```

7. Restart the vsftpd service.

<u>On client:</u>

8. As user student, create a file (any filename) for uploading to the FTP server.
9. Use ftp to establish an ftp session to your server. Login as anonymous.
10. Type “cd incoming” to change to the upload directory.
11. Type “put *filename*” to upload your file, changing *filename* to the name of your file.

```
[student@client ~]$ cd
[student@client ~]$ touch anyfile.txt
[student@client ~]$ ftp 192.168.30.88
Connected to 192.168.30.88 (192.168.30.88).
220 Welcome to the ST2412  FTP service.
Name (192.168.30.88:student): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/" is the current directory
ftp> ls
227 Entering Passive Mode (192,168,30,88,105,86).
150 Here comes the directory listing.
drwx-wx---    2 0        50              6 Sep 21 02:52 incoming
drwxr-xr-x    3 0        0              46 Sep 14 02:04 pub
226 Directory send OK.
ftp> cd incoming
250 Directory successfully changed.
ftp> put anyfile.txt
local: anyfile.txt remote: anyfile.txt
227 Entering Passive Mode (192,168,30,88,68,72).
150 Ok to send data.
226 Transfer complete.
ftp>
```

Replace this with the ip address of your server.

12. Now turn off the SELinux boolean value ftpd_full_access:

    sudo setsebool -P ftpd_full_access off

    and repeat the same file upload attempt.

    This time you may not be able to upload the file, and it could be due to the SELinux settings.

```
ftp> put anyfile.txt
local: anyfile.txt remote: anyfile.txt
421 Timeout.
Passive mode refused.
ftp>
```

The following steps show how to configure SELinux to allow anonymous uploads when ftpd_full_access is not set. In this approach, we only set specific folder to be writable instead of allowing full access to any folder.

<u>On server:</u>

13. Check and change the file context cdof /var/ftp/incoming to be publicly writable.

```
ls -lZ /var/ftp
chcon -t public_content_rw_t /var/ftp/incoming
```

```
[root@server ~]# ls -lZ /var/ftp
total 0
drwx-wx---. 2 root ftp  unconfined_u:object_r:public_content_t:s0 25 Sep 21 11:19 incoming
drwxr-xr-x. 3 root root system_u:object_r:public_content_t:s0     46 Sep 14 10:04 pub
[root@server ~]# chcon -t public_content_rw_t /var/ftp/incoming
[root@server ~]# ls -lZ /var/ftp
total 0
drwx-wx---. 2 root ftp  unconfined_u:object_r:public_content_rw_t:s0 25 Sep 21 11:19 incoming
drwxr-xr-x. 3 root root system_u:object_r:public_content_t:s0        46 Sep 14 10:04 pub
[root@server ~]#
```

With the context changed from public_content_t to public_content_rw_t , incoming folder should allow file upload operations.

14. Check the SELinux booleans for any setting related to FTP upload.

```
getsebool -a | grep ftpd
```

15. Set the SELinux boolean to allow anonymous FTP writes (the command will take a while to make the setting permanent).

```
setsebool -P ftpd_anon_write on
```

```
[root@server ~]# getsebool -a | grep ftpd
ftpd_anon_write --> on
ftpd_connect_all_unreserved --> off
ftpd_connect_db --> off
ftpd_full_access --> off
ftpd_use_cifs --> off
ftpd_use_fusefs --> off
ftpd_use_nfs --> off
ftpd_use_passive_mode --> off
[root@server ~]#
```

Confirm that the ftpd_full_access is off. Only the ftpd_anon_write is on.

On client:

16. As user student, repeat the anonymous upload again to upload a different file (e.g. anyfile2.txt).
17. After the file upload, type "ls" to view the contents of the incoming directory. You should not be able to see any listing. This is to prevent you from downloading any files that have been uploaded by other people.

```
ftp> put anyfile2.txt
local: anyfile2.txt remote: anyfile2.txt
227 Entering Passive Mode (192,168,30,88,254,239).
150 Ok to send data.
226 Transfer complete.
ftp> ls
227 Entering Passive Mode (192,168,30,88,26,51).
150 Here comes the directory listing.
226 Transfer done (but failed to open directory).
ftp>
```

On server:

18. Verify the file has been uploaded to the incoming directory successfully.
19. Check the owner of the file.

```
[root@server ~]# ls -l /var/ftp/incoming/
total 0
-rw-------. 1 ftp ftp 0 Sep 21 11:38 anyfile2.txt
-rw-------. 1 ftp ftp 0 Sep 21 11:19 anyfile.txt
[root@server ~]#
```

What if Question.

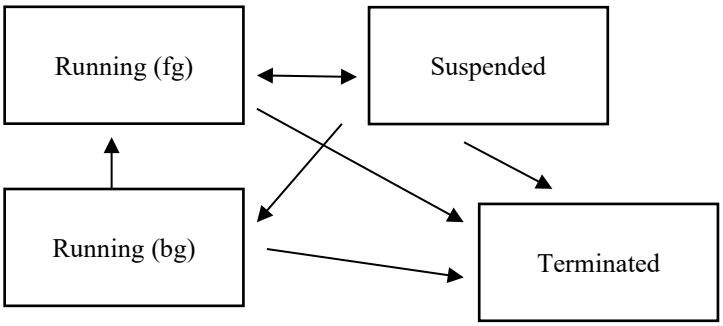
What if you are logging in as peter instead of anonymous? Can you upload a file to the incoming folder?

What if you are logging in as student instead of anonymous? Can you upload a file to the incoming folder?

13. User Process management basics

There are four possible states for a user started process:

- Running (foreground)
- Running (background)
- Suspended
- Terminated (the end of process)



When a user starts a program (or type in a command) at the command prompt, a process will be created and running in the foreground mode. Once the process has completed it will be terminated. While a process is running in the foreground mode, the terminal session will not accept any more user input commands until the process is terminated. There are a few commands you may use to change the state of a process.

Let's try out these few commands.

On any machine (login as student) :

1. At the command prompt, start a non-stop running command, type:

   ```
   ping -i 5 8.8.8.8
   ```

   This ping command should ping 8.8.8.8 once every 5 seconds. It will only be terminated if you press CTRL-C (The interrupt key).

   ```
   [student@server ~]$ ping -i 5 8.8.8.8
   PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
   64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=3.06 ms
   64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=2.56 ms
   ^C
   --- 8.8.8.8 ping statistics ---
   2 packets transmitted, 2 received, 0% packet loss, time 5005ms
   rtt min/avg/max/mdev = 2.565/2.817/3.069/0.252 ms
   [student@server ~]$
   ```

   This illustrates how to change a process from the running (fg) state to terminated state.

2. At the command prompt, start a non-stop running command, type:

   ```
   ping -i 5 8.8.8.8
   ```

   Wait for a few successful pings, press CTRL-Z (The Suspend Key), to suspend the running process. This process is not terminated, instead, it is in suspended (or paused) mode. The command prompt will be shown, and the user can type in other commands.

```
[student@client ~]$ ping -i 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=6.91 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=7.16 ms
^Z
[1]+  Stopped                 ping -i 5 8.8.8.8
[student@client ~]$ ls
anyfile2.txt  Desktop    Downloads  Pictures  Templates
anyfile.txt   Documents  Music      Public    Videos
[student@client ~]$
```

This illustrates how to change a process from the running (fg) state to the suspended state (Paused).

To check if there is any suspended process in your terminal session, you can use jobs command. Type:

```
jobs
```

```
[student@client ~]$ jobs
[1]+  Stopped                 ping -i 5 8.8.8.8
[student@client ~]$
```

The above shows that there is a suspended process. ie. 'ping -i 5 8.8.8.8'. The [1] indicates that we can refer to this job as '%1'.

Caveat: Many Linux beginners are not aware of the difference between the interrupt key (^C) and the suspend key (^Z). A suspended process still holds system resources. If a user keeps suspending his/her processes, it will waste a lot of system resources until the user logout. In addition, many Linux beginners (like LAS students) prefer to use ^Z over ^C because they may misunderstand that ^Z is more powerful than ^C.

3. Continue from the above. To bring back the suspended process to foreground running state, at the command prompt type:

```
fg %1
```

```
[student@server ~]$ jobs
[1]+  Stopped                 ping -i 5 8.8.8.8
[student@server ~]$ fg %1
ping -i 5 8.8.8.8
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=3.45 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=2.36 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=5.11 ms
^Z
[1]+  Stopped                 ping -i 5 8.8.8.8
[student@server ~]$
```

The above illustrates how to use the fg command (with the corresponding job id) to change a suspended process back to running (fg) state.

We can use bg command to change a suspended process to running (background) state. When a process is running in background, it will have a lower priority for getting the system resources, but it is still running, and it may send output to the terminal screen. Users may be confused when multiple background running processes are displaying their output to the terminal.

Moreover, a background running process will not hold up the command prompt input, therefore, the user can type in another command to run in foreground mode. Again, when foreground processes and background processes are running on the terminal session, it is rather confusing to the user:

```
[student@server ~]$ jobs
[1]-  Stopped                 ping -i 5 8.8.8.8
[2]+  Stopped                 ping -i 6 127.0.0.1
[student@server ~]$ bg %1
[1]- ping -i 5 8.8.8.8 &
64 bytes from 8.8.8.8: icmp_seq=18 ttl=128 time=4.73 ms
[student@server ~]$ fg %2
ping -i 6 127.0.0.1
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=128 time=3.85 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.060 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=128 time=2.17 ms
^C
--- 127.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 55748ms
rtt min/avg/max/mdev = 0.060/0.073/0.125/0.023 ms
[student@server ~]$ 64 bytes from 8.8.8.8: icmp_seq=21 ttl=128 time=74.9 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=128 time=2.18 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=128 time=2.15 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=128 time=2.45 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=128 time=5.78 ms
^C
[student@server ~]$ 64 bytes from 8.8.8.8: icmp_seq=26 ttl=128 time=2.96 ms
```

In the above sample terminal session, we have first identified two suspended ping commands. Then we resume one of them to run at foreground, and the other to run at background. Subsequently, by pressing the interrupt key (CTRL-C), we terminated the foreground running process, however, it cannot terminate the background one!

To terminate the background one, we need to know its jobs id then use the fg command to bring it to foreground then can use CTRL-C to terminate it.

4. You can initiate a program to run in background mode by appending a '&' symbol at the end of the command line. For example:

```
ping -i 5 8.8.8.8  &
```

```
[student@server ~]$ ping -i 5 8.8.8.8 &
[1] 6715
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=2.56 ms
[student@server ~]$ 64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=2.47 ms

[student@server ~]$
[student@server ~]$ ls
Desktop    Downloads  mytestfolder  Public     Videos
Documents  Music      Pictures      Templates
```

When we launch a command in background running mode, the system returns two important ids. One is the job number which is enclosed in a pair of [ ]. The other is the process id. In the above sample, the job id is 1, and the process id is 6715.

5. To terminate a background running / suspended process, we can use the kill command with the job id or the process id as the argument. For example:

```
[student@server ~]$ 64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=2.89 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=2.05 ms
kill %1
[1]+  Terminated              ping -i 5 8.8.8.8
[student@server ~]$
```

Note: At the old days, when everyone is using terminal to access to a time-sharing system, some experience users might push a few background jobs to the system to gain more CPU time than those did not aware the process concept.

At present day, we do not use time-sharing system very often, the ^Z, fg, bg commands are no longer that useful. For instance, if we would like to push more jobs to the system, we can simply open a new terminal window to launch a new job. The illustration in this section is to ensure you will not mistakenly missed up CRTL-C (interrupt and kill) with CRTL-Z (suspend and hold) operations.

14. Sed basics

Sed is a stream editor that reads from a file line by line, modifies the data if required and sends the results to the output stream.

<u>On any machine:</u>

1. Create the following text file "/tmp/data.txt". Use the tab key to separate the columns

Kitty	Australia	GroupA	$109	GroupC	Singapore 123456
Peter	New Zealand	GroupA	$94	GroupF	Singapore 654321
Paul	New York	GroupB	$103	GroupG	Singapore 881882
Mary	London	GroupB	$103	GroupG	Singapore 881882

2. To replace the first occurrence of "Group" with "Team", run the following command.

```
sed s/Group/Team/ /tmp/data.txt
```

The "GroupA" and "GroupB" in the third columns will be replaced with "TeamA" and "TeamB". The fifth column remains unchanged.

3. To replace all occurrences of "Group" with "Team", run the following command.

```
sed s/Group/Team/g /tmp/data.txt
```

4. To replace all occurrences of "$" with "SGD$", run the following command.

```
sed 's/\$/SGD\$/' /tmp/data.txt
```

Take note that all the above sed commands only print out the modified content. The data.txt content remains unchanged.

To let the modification to apply to the data.txt, you can use the -i option.

For example:

```
sed -i 's/\$/SGD\$/' /tmp/data.txt
```

15. Awk basics

Awk is an often-used Linux tools among administrators. It is designed for text processing and typically used as a data extraction tool.

On the same machine as the previous exercise:

1. To print the first three columns if the third column contains the letter “A”, and sort alphabetically. The “-F” option tells the awk command that the tab key is the column separator.

```
awk -F "\t" '$3 ~/A/ {print $1, $2, $3}' /tmp/data.txt | sort
```

2. To print the first four columns if the second column starts with the letter "N", and sort alphabetically by the 2nd column. The “-F” option tells the awk command that the tab key is the column separator.

```
awk -F "\t" '$2 ~/^N/ {print $1, $2, $3,$4}' /tmp/data.txt |
sort -k 2
```

3. [optional] You may explore how to use python to carry out the above Sed and Awk tasks.

Note: Both of sed and awk are heavily related to regular expression. Nowadays, administrator may have the more choices of using python scripts instead of tools like sed and awk for text processing.

Your Task: There are many other common utility commands that can help a Linux administrator for their day-to-day jobs. You may try to use the combinations of the following commands: find, wc, ls, grep, and sort to :

1. Display all the file names which are ended with '.conf' under the /etc folder in alphabetical order.
2. Find out the total number of *.conf files under the /etc folder.
3. Locate the exact path of the file 'lvm.conf' under the /etc folder.

*Hints : you may use man command to find out the usages of all these commands.

Additional Reference

- How to recover RHEL 8 / CentOS 8 root password - https://linuxconfig.org/redhat-8-recover-root-password
- vsftp: why is allow_writeable_chroot=YES a bad idea? - https://serverfault.com/questions/743949/vsftp-why-is-allow-writeable-chroot-yes-a-bad-idea

End of Practical