**Excercise Notes Zip**

[[https://www.evernote.com/images/file-generic.png](https://www.evernote.com/shard/s555/res/3b87f259-36c5-4912-8f05-52c7073cf977/lfcs-linux-operation-essentials.zip)**lfcs-linux-operation-essentia...**2.4 MB](https://www.evernote.com/shard/s555/res/3b87f259-36c5-4912-8f05-52c7073cf977/lfcs-linux-operation-essentials.zip)

PluralSight - LCFS Linux Operation Essentials Course Notes

**Introducing the Linux Foundation and LCFS Certification Module**

Preparation for both the Linux Foundation Certified SysAdmin and Red Hat Certified System Administrator

Linux Foundation and LCFS

Founded 2000

LCFS Linux SysAdmin taken on CentOS 7, openSUSE 13.1 or Ubuntu 14.04 2

Exam lasts for 2 hours

Course Map

LCFS: Linux Essentials

LCFS: Linux Operation Essentials

LCFS: Linux User and Groups Management

LCFS: Linux Storage Management

LCFS: Linux Networking

LCFS: Linux Service Management

LCFS: Linux Virtualization Management

Map for this course

Boot Process

Scheduled Tasks

Software Management

Process Management

SELinux

System Monitoring

**Diagram of Lab Environment**

**Reading Operating System Data**

cat /etc/system-release

This command will universally display the version of the installed operating system

lsb\_release -d

Another command that will display the version of the operating system. An additional package would need to be installed for this command

rpm -qf $(which lsb\_release)

Query file arguments

Package management command

$(which lsb\_release) - results with the package call

uname -r

Provides the kernel version

cat /proc/version

Command will also display the version of the kernel and additional details

cat /proc/cmdline

displays details of a separate boot partition which includes the root partition used by the grub loader. Root of boot, all of the options that have been passed to the kernel during boot phase

lsblk

List all of the block devices and the hierarchy

This command will identify the partitions of each block

**Introduction and Using wall**

Exam Objectives

Boot, reboot and shutdown a system safely

Boot systems into different runlevels manually

Manipulate the Linux system during the boot process

Demo

Messaging Users  
Shutdown commands  
Changing runlevels  
Setting the default runlevel  
Booting to rescue.target

**Messaging Users**

write command overview

Can be used to write from one user to another. Anything typed after entering the command will appear in the other terminal

Crtl + d and the write command

Using Crtl + d to exit the message. the phrase "EOF" will appear in the receivers terminal

wall command overview

Messages are sent to everyone

cat > message <<END

The server is offline at 5pm

END

wall <message

This string of inputs will send the message to every terminal

mesg command overview

service that can be turned on or off by typing mesg n or y. the mesg command can allow or prevent write access to the terminal.

mesg will give status of whether or not its activated

**Shutdown Commands**

Specific Commands

reboot

halt

poweroff

init

String of commands that can be executed most machines

telinit

Same as init but can be accesses on legacy machines

shutdown -r

reboot

shutdown -h 10 "The system will self-destruct in 10 minutes..."

provide length of time before shutdown using the second argument. The message will be sent vial wall to all terminals on the server

shutdown -c

This can cancel the shutdown process and will prompt that the future shutdown has been cancelled

/run/nologin

This file will appear after 50 percent of the time has passed and only root will not have this file and will be the only one that can login. once the login file appears  and will read system is going down

**Changing Runlevels**

Runlevel Defined

process of managing the various ways of running the machine and with different specs depending on the number chosen

Known as targets for system d based machines

CentOS 7 is system d based system

who -r

display the runlevel the terminal is on

runlevel 5 is the same as graphical.target as both of the terms mean the same

runlevel

this command by itself will also display the run level the terminal is on

systemctl get-default

This command will display the default runlevel at boot

systemctl set-default multi-user.target

Command that can change the default boot level

systemctl isolate multi-user.target

This will disconnect the current daemon and reconnect, can be used to change the run levels

systemctl isolate rescue.target

This will change the run level to rescue level or rescue mode or run level 1

telinit 0

Command that will power off session

**Selecting Runlevels at Boot**

Rescue.Target

Only root user can use this runlevel and has minimal functionality

Start -> Normal Start

Start VM on normal mode

When booting the grub menu choose esc and press the e key to edit the first entry and teh boot parameters

Navigate to the end of the boot item and add systemd.unit=rescue.target

This will reboot in the rescue mode

**RECAP Introcution and Using Wall Module**

init 0 or telinit 0 or poweroff

Power off machine

init 6 or telinit 6 or reboot

Reboot machine

Require root privileges

shutdown -r 10 "Message to display"

Shutdown command that can send message and has parameters with reboot within 10 minutes. once less than half of time, will prohibit users from logging in unless root

wall command

directs messages to all users terminals

who -r or runlevel

displays run level

systemctl get-default

Display default runlevel

systemctl set-default multi-user.target

This command will set the default runlevel

**Managing GRUB Recovery Module Introduction**

Exam Objectives

Manipulate the Linux system during the recovery process

Develop system disaster recovery tests

Demo

Enable recovery mode  
Recover from lost root passwords

Enable Recovery Mode

Overview

e key will edit the boot process in GRUB

Edit GRUB file Set Disable Recover to False

vi /etc/default/grub and edit

Set GRUB\_DISABLE\_RECOVERY="false"

Update GRUB Configuration

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

reboot

Once rebooted, the GRUB will provide a recovery mode for each GRUB option.

who -r (this will display the run level currently and previously)

**Recover Lost Root Passwords**

Overview

Use the up or down arrow keys to interrupt the automated boot process in GRUB

e key to edit an entry in grub

Scroll down to the end of line beginning with "linux16"

delete "rhgb quiet"

insert the two option "rd.break enforcing=0"

press Ctrl + x to reboot

Remount the root file system as read write

mount -o remount,rw /sysroot

Generate a root set a false root that points to the root directory

chroot /sysroot

Create a new password for root user

passwd

prompted "all authentication tokens updated successfully"

Exit chroot environment

exit

Remount as read only using the command and exit to continue boot process

mount -o remount,ro /sysroot

exit

Restore the security context for the password file

restorecon /etc/shadow

Set the enforcing back to original using command

setenforce 1

**Recap Boot Process Module**

Grub Recovery

Edit etc grub file

vi /etc/default/grub

Set Grub disable recovery to false

GRUB\_DISABLE\_RECOVERY='"false"

Regenerate grub configuration for addition recovery mode

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

Recover Lost Root Passwords

Boot to initramfs

Edit linux16 entry

Remove rhgb and quiet add rd.break enforcing=0

switch\_root# mount -o remount,rw /sysroot ; chroot /sysroot

sh-4.3$ passwd ; exit  
switch\_root# mount -o remount,ro /sysroot ; exit

Login and reconfigure

restorecon /etc/shadow

setenforce 1

**Managing GRUB2**

Exam Objectives

Install, configure and troubleshoot the bootloader

Change kernel runtime parameters, persistent and non-persistent

Demo

Reinstall GRUB2

Manage GRUB2 defaults

Manage GRUB2 with grubby

Password protecting GRUB2

Custom GRUB2 entries

**Reinstall GRUB2**

Login as root and install grub 2 with master location parameters usually located in /dev/sda

grub2-install /dev/sda

For efi machines (not i386)

yum reinstall grub2-efi shim

**Manage GRUB2 Defualts**

Edit the grub file

vi /etc/default/grub

GRUB\_TIMEOUT=x

Length of time in seconds it takes for the menu to display before default process

GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"

Message regarding distribution

GRUB\_DEFAULT=saved

Message indicating default

GRUB\_DISABLE\_SUBMENU=true

Disable submenu switch

GRUB\_TERMINAL\_OUTPUT="console"

Output switch

GRUB\_CMDLINE\_LINUX=

Default entry for kernel

GRUB\_DISABLE\_RECOVERY

Default entry for kernel

Update grub configuration to overwrite grub.cfg

grub2-mkconfig -o /boot/grub2grub.cfg

reboot

**Manage Persistent Setting with grubby**

Using grubby to check the default kernel value

grubby --default-kernel

Setting the default with grubby

grubby --set-default /boot/vmlinuz-3.10.0-327.18.2.e17.x86\_64

Check to see what is listed using command

grubby --info=ALL

Check info for only default kernel

grubby --info /boot/vmlinuz-3.10.0-327.18.2.e17.x86\_64

grubby to remove arguments in prompt for last kernel without updating file

grubby --remove-args="rhgb quiet" --update-kernel !$

Verify the changes are in effect to specified kernel

grubby --info /boot/vmlinuz-3.10.0-327.18.2.e17.x86\_64

reboot

Once rebooted the default kernel will be a different value and the new value will display all messages during boot due to configuration changes

**Password Protect GRUB2**

Set the default kernel back to the latest for the guest additions as they work only for latest updated distribution

grubby --set-default /boot/vmlinuz-3.10.0-327.18.2.e17.x86\_64

reboot

Set Unencrypted Password

Make a copy of the file before changing in the home directory

cp /etc/grub.d/01\_users .

Edit the file in /etc

cd /etc/grub.d/

vi 01\_users

Remove all but 1 line

set superusers="root"

Change the line

set superusers="dany"

Add password

password dany L1nux

save and quit

reboot configuration using mkconfig

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

Set Encrypted Password

Overview

grub2-mkpasswd-pbkdf2

type password L1nux

copy the output using ssh and mouse

Go back edit the configuration file

!v or vi /etc/grub.d/01\_users

Edit password line with the new data

password\_pbkdf2 dany ""

save file

reboot the mkconfig

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

**Custom GRUB2 Entries**

Create a custom file using vi

menuentry 'CentOS Custom' {

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

linux16 /vmlinuz-3.10.0-327.3.1.el7.x86\_64 initrd16 /initramfs-3.10.0-327.3.1.el7.x86\_64.img  
}

Edit 40\_custom file

vi 40\_custom

add the information underneath the visible information in file

Save and exit

reboot mkconfig

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

**Recap Managing GRUB2 Module**

Reinstalling GRUB2

grub2-install /dev/sda

Configure GRUB defaults

/etc/default/grub

Regenerate main configuration file after any changes are made to GRUB configs

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

Use grubby to determine default GRUB boot

grubby --default-kernel

Use grubby to set the default boot

grubby --set-default /boot/vmlinuz...

Use grubby to find information of GRUB

grubby --info=ALL

Use grubby to find detailed information on a single boot type

grubby --info /boot/vmlinuz...

Use grubby to remove arguments on a single GRUB boot type

grubby --remove-args “rhgd” --update-kernel /boot/vmlinuz

Manage GRUB2 user accounts

vi /etc/grub.d/01\_users

!/bin/sh -e

cat << EOF

set superusers=“andrew”

password andrew L1nux

EOF

Adding Grub2 Custom entries using grub.d in 40\_custom

menuentry 'CentOS Custom' {

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

linux16 /vmlinuz-3.10.0-327.3.1.el7.x86\_64

initrd16 /initramfs-3.10.0-327.3.1.el7.x86\_64.img

}

**Managing Linux Processes Module Introduction**

Exam Objectives

Identify resource utilization by process

Verify the integrity and availability of key processes

Demo

Using the process status or ps command

Monitor process usage from /proc and use the $$ variable

Manage processes with kill

Use shortcuts with pgrep and pkill

Monitor resource usage with top

**Listing Processes with ps**

Overview

ps - process status

ps command displays the process id, tty attached, cpu time and the command that is running

Accepts 3 options

ps -e (display all processes)

ps aux (detailed list of processes)

ps -e --forest (GNU added displays process tree and grouping processes together in their heirarchy)

pstree (more detailed easier to see process tree)

ps -f (full listing of the current shell)

ps -F (more than full listing with additional including size of program and resident set size)

ps -l (long listing including user name and id and priority)

ps -ly (replaces a column for resident set size, how much memory is being used)

ps -elf (long and full listing)

ps -elf | grep sshd (long and full listing of a specific process

**Using the proc directory and the $$ Variable**

Overview

/proc directory

Represents running processes in the system

Maintains files with information of the system

Numbered folders indicate process PID number

ps -p1 -f

Will provide a full listing or process ID 1

echo $$

This will provide the PID of the bash process

ps -p $$ -F

Displays information related to current bash shell in extra full listing

cd $$ or cd and the PID

Navigate to the process directory

ls -l exe

Provide the exe file associated

cat loadavg

create a new process and provide a load average for 1 mins, 5 mins and 15 minutes and the number of processes running in column 3

**Sending Signals with kill Command**

stty -a

Provides list of signals and their short cut commands for services

kill -l

Provide the list of signals that can be sent messages

kill <PID>

kill 2756 is an example of a command that will kill the bash

kill -15 <PID> or kill -term <PID> or kill -sigterm <PID>

Represents the same action as kill <PID>

kill -9 <PID> or kill -kill <PID> or kill -sigkill <PID>

will force kill the PID associated. This may be necessary as some processes will not be killed by the standard. This also will depend on the user and can only be performed on processes associated to the user

**Shortcuts with pgrep, pkill and top**

Using pgrep and pkill

pgrep sshd

Provides a list PID's to services that are sshd type

ps -F -p $(pgrep sshd)

Provide a full and wide listing of the sshd services running

Create sleep processes in the background

sleep 100& (create three sleep services)

pgrep sleep

Display the PID's for the active sleep services

pkill sleep

Terminates all the sleep services

top

Sorted list by type of all resources, memory and CPU utilisation

top - pressing the f and s key

Enter the chart menu and choose between which columns to sort using the s key. Instructions are listed above the menu for other commands

top q key

quit top menu

**RECAP Managing Linux Processes Module**

ps Command Overview

Display current process status

UNIX ps -elf

Long and full listing

BSD ps aux

Different method of producing same results

GNU ps --forest

Long switch of GNU tree layout

/proc

Directory containing the process information

Sub directories represents running processes

cd /proc/$$/

Current running process id

cat /proc/load

Provides the load average for the previous process and the last field for file

kill

Default signal is to terminate

kill -l

List signals

kill -15 PID

Terminate the particular signal by PID

kill -9 PID/ kill -kill PID/kill -sigkill PID

Force kill signal by PID

stty -a

List keyboard shortcuts

p-tools

piping with pgrep

pgrep

pkill

ps -F -p $(pgrep sshd)

provide a full and wide list of the sshd type service

pkill sleep

Process will terminate all services with the labeled sleep

**Managing Process Priority Module Introduction**

Exam Objectives

Change the priority of a process

Demo

Backgrounding tasks

Configuring the process priority

**Backgrounding Tasks**

Start a job in the background using &

sleep 1000&

jobs

lists all jobs working in the background

Without "&" job will run in the foreground

sleep 1000

Crtl +z will suspend current running job

bg will resume the job in the background

fg will bring the job back to the foreground

stty will list the same controls to view

to choose a particular job instead of the current job, use the placement number after the command

fg 1

the job with the "+" has the particular focus for the commands unless specified

ps -F

full listing of processes

**Using Nice to control Process Priority**

when using ps -l command,

PRI

Priority

Range 60 - 99

The lower the number, the higher it is in priority

NI

Nice value

Default value of 0

range -20 - 19

Higher the number, the less CPU time the system will take

nice command

Used to start an application

nice -n  19 sleep 1000

-n switch indicates nice value

The job type is after the nice command

Only root users can set a negative nice value

renice command

Alter priority of running processes

renice -n 10 -p 4022

Command will change the priority to 10 for job

Only root can set a nice value lower than what the level is when the job is running

Root can also set negative values on startups

nice  -n -20 sleep &

ps -l

Root can set the default nice value and based by users and or groups

vi /etc/security/limits.conf

dansaf - priority 10

will set the default priority to 10 for dansaf

Can also be set to another value to provide more usage for specific users

**Recap Managing Process Priority**

Background

&

Run process in background

^z bg

Suspend application and resume using bg

jobs

See all the jobs using command

fg

Foreground jobs and "+" displays current job

Nice

Nice value -20 to +19

Default 0

Process priority 60 - 99

nice -n 5 firefox&

Start Firefox with less priority

renice -n 10 -p 14721

switches have to be used in the specified value

/etc/security/limits.conf

Control the root as default

tux - priority 10

Specify the group by using @GroupName

**Monitor Linux Performance Module Introduction**

Exam Objectives

Identify and resolve system performance bottlenecks and platform instability

Demo

List contents of the procps-ng package

Investigate and use tools such as free, tload, top, and vmstat

**Listing Standard Tools in procps-ng**

rpm

Defined as redhat package manager

Using this command is like query to the RPM database

rpm -ql procps-ng command

Display what is included in the procps-ng package

rpm -qf /usr/bin/top

Query file switch which displays which file the executable belongs to

rpm -qd procps-ng

Query documents that reside in procps-ng

Basically the man pages

-qc switch checks for configuration files

rpm -ql procps-ng | grep '^/usr/bin/'

This command will pipe and grep list only the items in the /usr/bin

rpm -ql procps-ng | grep '^/usr/bin/' | wc -l

Command to count the amount of items

You can use the pipes to continue to build up

**Using pwdx and pmap**

free command

Display the amount of free and used memory in the system. The -m switch to display the free memory in megabytes

pmap

Report memory map of a process

pmap 4170

Report memory map of 4170 process

pwdx

Report current working directory of a process

pwdx $(pgrep sshd)

Report current working directory of sshd association. May require root or sudo privileges

**Working with uptime and tload**

uptime command

Tell how long the system has been running

Load average is by 1 minute, 5 minutes and 15 minutes

Load average depends on how many CPU's

w command

Display who is logged on and what they are doing

Will also display the user's load average

lscpu

Displays information about the CPU architecture

Will display the amount of CPU's

cat /proc/uptime

Read uptime and is displayed in seconds

First column displayed in seconds system has been up. Seconds column displays idle time

cat proc/loadavg

Read load average and displayed in seconds

watch command

Executes a program periodically, showing output fullscreen

By default, runs a command every 2 seconds

watch -n 4 uptime

Execute uptime periodically every 4 seconds

Top left corner of shell will display the interval and the command

tload command

Graphic representation of system load average

**Recording Performance with top and vmstat**

top command

Display Linux processes

top -b -n1

-b switch will run in batch mode

-n switch is the amount of iterations

top -b -n1 > file1

">" switch will send the information to file1 in the home directory

vmstat command

Report virtual memory statistics

vmstat -S K

-S switch will specify unit type

k switch is using the default memory type in kilobytes

si and so columns

swap in and swap out amount

bi and bo columns

byte in and byte out amount

cs column

context switch per second

vmstat 5 3

Executing command with a delay of 5 seconds and 3 iterations

**Recap Monitor Linux Performance Module**

Inside procps-ng

rpm -qf /usr/bin/ps

Query package manager database

rpm -ql procps-ng

List complete content of the package

rpm -ql procps-ng | grep ‘^/usr/bin/’

List all the binaries in teh package using grep

pmap and pwdx

pmap $$

Check memory map of previously selected package

pwdx $$

Check working directory of a process

sudo pwdx $(grep sshd)

Check specific processes. May need to use sudo. This command specifically checks sshd

uptime and tload

/proc/uptime

Lists uptime and idle time in seconds

/proc/ loadavg

Lists load aveerage from the load average file

w

List output of who's logged in and output of uptime

watch uptime

List in  intervals the uptime

tload

List load average in a graph, display values as more CPU is utilised

top and vmstat

top -b -n1

List in batch mode and saved to a file

vmstat -S k

List default in kilobyte

vmstat -S m 5 3

Execute 3 times and for 5 seconds in megabytes

**Using Sysstat to Monitor Performance Module Introduction**

Exam Objectives

Produce and deliver reports on system use (processor, memory, disk, and network), outages, and user requests

Demo

Install sysstat and view initial configuration

Using additional tools

Using the sar tool to report on system usage

**Installing Sysstat**

Overview

sysstat configuration file used to configure sysstat logging. The details are similar to vmstat.

Install as root su -

yum install -y sysstat

cat /etc/cron.d/sysstat

Where the cron daemon lives. Looks at scheduled tasks regularly

The file contains system activity information for sa1 and sa2 scripts scheduled to run every 10 minutes and once a day

cat /etc/sysconfig/sysstat

Contains the sysstat configuration file

systemctl start sysstat

This command will start the sysstat service

systemctl enable sysstat

Turned the service on so the configuration files will be executed on boot

systemctl status sysstat

Checking the status of the service

**Using Additional Tools**

iostat defined

Report Central Processing Unit (CPU) statistics and input/output statistics for devices, partitions and network filesystems (NFS).

dm-0

Root file system on sda

dm-1

Logical volume used for swap on sda

iostat -m

-m switch will display in megabytes

iostat -m 5 3

will list in megabytes 3 times every 5 seconds

dd if=/dev/zero of=test

open a new session and execute the test command in parallel with the iostat to test the performance.

rm test

Remove test. This command can fill up memory

pidstat defined

Report statistics for Linux tasks

pidstat -p $$ 5 3

List process statistics for the current task with a delay of 5 seconds 3 times

mpstat defined

Report processors related statistics

mpstat -P ALL 2 3

List all the processors with 2 second delay 3 times

**Creating System Activity Reports**

Reporting using sar

sar defined

Collect, report, or save system activity information

sar command with no switch

overall usage CPU as a whole

sar -u

-u switch is user specific

sar -r

-r switch reports memory utilisation

sar -b

-b switch reports disk IO utilisation

sar -n DEV

-n switch reports network activity for the DEV keyword

sar -q

-q switch reports queue load average

cd /var/log

Log directory

/var/log/sa

System administration directory

the files will begin with "s" than provide a number which represents the day "s15" represents 15 day of the month

sar -s 10:00:00 -e 11:00:00

-e switch sets the end time of the report

This command will search for stats from 10:00 till 11:00 am of today

sar -s 10:00:00 -e 11:00:00 -f /var/log/sa/sa15

-f switch will check from file. This is particularly helpful for specific days and files associated

**Recap Using Sysstat to Monitor System Performance Module**

Installing sysstat

Tool recommended to be installed in all machines initially

yum install -y sysstat

sudo su or root

cat /etc/cron.d/sysstat

Configuration file for cron daemon

cat /etc/sysconfig/sysstat

Main configuration file and history. By default is 28 days

systemctl enable sysstat

CentOS 7 is systemctl for process. This may be different depending on OS used. Set to execute on boot

systemctl start sysstat

Activate it immediatley

Additiional Tools

iostat -m 5 3

Disk activity displayed in MB, delay of 5 sec reading 3 times

pidstat -p $$

Disk activity specific to process ID

mpstat -P ALL

Specify the processor, this command is using ALL which indicates everything

System Activity Reports

/var/log/sa/

Log destination for CentOS specific distribution packages

sar -u (CPU)

System activity reporter CPU specific sar without a switch is the same as using the -u switch

sar -r (Memory)

System activity reporter memory specific

sar -q (Load Averages)

System activity reporter load average specific

sar -n DEV (Network)

System activity reporter network specific

sar -s 14:00:00 - e 15:00:00

Specify start and end times. Must be placed in the proper format

sar -f /var/log/sa/sa15

Specify files which assists in tracking history and trends

**Managing Shared Libraries Module Introduction**

Exam Objectives

Manage shared libraries

Demo

Viewing shared libraries

Shared locations

Shared library cache

**Viewing Shared Libraries**

ldd command

List shared libraries for particular program

ldd /usr/bin/ls

Command will check libraries in ls folder

pcre

POSIX compliant regular expression. In shared libraries ending in pcre will comply with regular expressions

ldd /usr/bin/grep

grep also makes the same reference in lib64 library and contains a code to process, meaning the grep is compatible with using regular expressions

.so

Shared libraries use the ".so" extension, meaning "standard module" (they do not abbreviate by the sound)

pmap $$ or pmap and process ID

List all the running libraries in shell. This can be confirmed with the .so extension as they are included in the list

Identical to using lss /usr/bin/bash except this will give you more precise results as it's specific to libraries

**Setting the Location of Shared Libraries**

Shared Library Locations

/lib/lib64

In CentOS 7, the library is located in the /usr directory and have symbolic links from root as normally these files are located in root or can be located in the root folder "/". This includes the bin and sbin directories. This means that these directories will reside in "/usr" instead of "/"

ld.so.conf.d

The configuration files for libraries are located in /etc/ld.so.conf.d

mkdir /usr/local/lib/pluralsight

Create a new directory for the location of shared library. Substitute the "pluralsight"

cp /root/libdisplayuid.so /usr/local/lib/pluralsight

Copy the libdisplayuid library over to the newly created directory

cp /root/libdisplayuid.so /usr/local/lib/pluralsight

Create the libdisplayuid.so file and copy it from where it lives to the directory associated

ls -l !$

Verify that is in fact in the proper directory

chmod +x !$/displayuid.so or chmod /usr/local/lib/pluralsight/displayuid.so

Set Executable permissions on the file using chmod

ls -l /usr/local/lib/pluralsight

Verify that the file has the proper permissions

/etc/ld.so.conf.d vi pluralsight.conf

Create a pluralsight configuration file in the ld.so.conf.d folder

/usr/local/lib/pluralsight/

Add a directory where the system will search for the modules on the top of file, save and close with "x"

This creates the library configuration as the destination of the library is set.

This creates the library and the configuration which is pointing to the destination of the files

Steps in high level

1. Build the folder/directory in the library location and name it appropriately

2. Create and add the .so file and place it in the newly created directory

3. Once the standard module file is created and in the proper place, create the configuration file and associate the file to the destination of the directory

echo $LD\_LIBRARY\_PATH

This command allows users to set a path temporarily to add and manage files

**Rebuilding the Library Cache**

Verifying the module using ls

ls -l /usr/local/lib/pluralsight/

cat shouid.c

This file was a demo file that was used as an example

ldconfig command defined

ldconfig creates the necessary links and cache to the most recent shared libraries found in the directories specified on the  command line, in the file /etc/ld.so.conf, and in the trusted directories (/lib and /usr/lib). The cache is used by the run-time linker, ld.so or ld-linux.so. ldconfig checks the header and filenames of the libraries it encounters when determining which versions should have their links updated

ldconfig -p

Investigate the ld cache content in it's entirety by using this command

ld.so.cache and ls -l /etc/ld.so.cache

These commands will inform the user of the time and whether or not the cache was updated

ldconfig

Execute this command to cache the library links. The library will now be updated

./showuid

Execute any commands that reference the library

This file was created to demonstarte and display the type of permission the user has

chown dansaf showid

Changes the permissions so that the user can access and execute

ldconfig -p | grep display

Once the ldconfig has been cached, the link should be committed to the proper location of the file

ldconfig -v

-v verbose switch will update the cache and provide a long list

**Recap Managing Shared Libraries Module**

View Shared Libraries

ldd /usr/bin/bash

Location of the library modules

pmap $$

Display the use of the modules

Shared Library Locations

/usr/lib

Location of shared libraries

/usr/lib64

Location of shared libraries

/etc/ld.so.conf.d/

Creating directory structures, extension directories. Then creating a config file directing to the proper location

LD\_LIBRARY\_PATH

Independant locations provided by the system can be configured using this path

Shared Library Cache

/etc/ld.so.cache

Update the library cache file which is located here

ldconfig -p

Read from the ldconfig file

ldconfig

Update the cache and write can also be executed by using ldconfig -v

**Scheduling Tasks in Linux Module Introduction**

Exam Objectives

Schedule tasks to run at a set date and time

Verify completion of scheduled jobs

Use scripting to automate system maintenance tasks

Demo

Using scripts for maintenance tasks

Scheduling jobs with cron

Scheduling jobs with anacron

Scheduling jobs with at

**Using Scripts for System Maintenance**

df -h

Check disk space and list

df -h > dr.txt

Creating a text file with the output of df -h in the home directory

mail -s "df" dansaf < df.txt

Create a mail instance with the subject "df" and attached text file that was previously created to dansaf user

Log in as user mail

open the mail using the number associated

vi df.sh

Build the command into a script with the below information

#!/usr/bin/bash

FILE=/tmp/df.txt

df -h > FILE

mail -s "df" < $FILE && rm $FILE

This script will designate the variable FILE to the df.txt location. FILE will include the df.txt which will be the disk utilisation. Then the script will send a mail to dansaf and remove the file only if the mail was successfull otherwise teh script finishes

chmod +x df.sh

Ensure that the execution permission is set to the newly created file

Execute script and fix error

There is a missing switch for the script indicating the recipient to send the mail to. change the line and re-execute the script

mail -s "df" dansaf < $FILE && rm $FILE

Check location of created file to ensure it was removed

**Scheduling Tasks with crond**

cron Defined

Daemon that executes scheduled commands. **Cron** is started automatically from /etc/init.d on entering multi-user runlevels. **Cron** searches its spool area (/var/spool/**cron**/crontabs) for **crontab** files (which are named after accounts in /etc/passwd); crontabs found are loaded into memory

su -

When performing changes to cron, its best practice to have sudo privileges

ls /etc/cron\*

Listing all the cron by by directory

/etc/crontab

System file root users create jobs in this location

/etc/cron.d

Extension directory, Where the sysstat job lives

/etc/cron.daily, /etc/cron.hourly, /etc/cron.monthly, /etc/cron.weekly

Scheduled jobs live in these directories

vi /etc/crontab

In the bottom of the file fill in the necessary info to create a job for root

45 8-18 \* \* 1-5 root df -h

crontab -e

As a user check for crontab jobs by using the cron -l

To edit the crontab use the crontab -e

Insert jobs to test the process

MAILTO=dansaf

\*/5 \* \* \* \* ls /etc

0 15 \* \* 5 /home/dansaf/df.sh

Back to root user cat /etc/crontab

Check the mail from the root user, new mail from cron job

Check the mail from the other user new mail from cron job

crontab -r

remove the cron jobs specific to user

**Using anacron as a job scheduler**

Anacron Command Defined

anacron is used to execute commands periodically, with a frequency specified in days

/etc/anacrontab

Location of the anacrontab file

Enter job at the end of anacrontab file

1 45 backup tar -cf /tmp/backup /etc

Job to happen daily (1) 45mins after system startup (45)  backup as the job identifier (backup) tar -cf /tmp/backup /etc command

Will create a compressed copy of the backup folder and place it in the /etc location daily, 45 minutes after boot

systemctl status crond

Check the current status of the crond daemon

Navigate to /etc/cron.hourly then ls command

Confirm that the anacron daemon will run hourly as it is in the cron.hourly folder

cat 0anacron

Investigating this file and found that if the script already executed in the da or if the machine is running on battery, will not execute. Otherwise, the script will execute anacron daily

**Irregular Job Scheduling with at**

at defined

 at and batch read commands from standard input or  a  specified  file which are to be executed at a later time, using /bin/sh

systemctl status atd

Check to ensure atd the service is running. This command displays if the system is up and whether or not the setting is enabled during boot

at noon

Command will begin the job process for noon if 12 has passed it would run the next day at noon

at>

Once schedule has been set, commands can be added in each line

at> ls /etc

at> ls /tmp

<EOT>

Crtl + d

pressing Crtl + d key will close the schedule menu

atq

List jobs in queue

at wednesday

Job schedule menu will appear and will execute on the next wednesday that occurs

at>ls /etc

<EOT

Add a simple task and close

atq

List the current tasks including the newly created task

atrm 4

Command will remove the specific job by id. in This case, it will remove the 4 job

at 13:23 Jun 23

This command will do as previosuly with a more specific time. This job uses switches for time and the date

ls /etc/\*.deny

Limit access to configuring the jobs specific to the .deny file in the /etc directory

By default everyone can access and make changes to the jobs unless added to the .deny file

.allow

files can be created in the same directory to only allow specific users access to the jobs. If no file is listed than the restrictions will be handled in the .deny file

.allow file sets precedence over the .deny file if both are created for the same job

**Recap Scheduling Tasks in Linux Module**

Scripts

#!/usr/bin/bash -shebang

(First line of script)

FILE=/tmp/df.txt -variable

(No extra spaces beside the "=")

df-h > $FILE

mail-s “df” dansaf < $FILE && rm$FILE -command line lists

First half of statement includes mailing user dansaf and reading the $File variable

Middle of statement checking to see if the command succeeds using the "&&"

"&&" is a half way point and if the first half of the statement succeeds, then the second half of the stament (after the "&&") will execute, otherwise, second half is ignore

Second half is removing the contents of the file variable which is the .txt file that was declared

"||" can also be used. The condition is if the first hallf fails then run the second half instead of succeeding

cron

# systemctlstatus crond

Check for the status of the crond job

/etc/crontab

Configuration directory

/etc/cron.d

Extension directory

$ crontab-l

List any cron job

$ crontab-e

Editing cron file

$ crontab-r

Remove cron file

MAILTO

Variable, output mail to the user. User will receive email copy of the output

min hour d-o-m month dow

Time configurations, minute, hour, day of month, month and day of week. Using the "\*" would incremently execute every time (day, month, year etc, for days the range in numbers is 0-7 with 0 and 7 both being sunday

# anacron

Using anacron for tasks that requrie the computer be tunred on. This will allow to customizer the delay time after boot to ensure the job runs

/etc/anacrontab

Directory where the configurations are made

period delay job-id command

Period of time to execute, job ID and the command

# at

# systemctlstatus atd

Daemon

$ at 17.45 23 May 2018

Ability to use the specific date

$ atq

List job

$ atrm

Remove jobs

**Log Files and Logrotate Module Introduction**

Exam Objectives

Locate and analyzesystem log files

Monitor security and conduct audits

Script automation tools to make work faster and more accurate

Demo

Audit login events

Audit access sudoand su

Scripting awk to analyze logs

Configuring system logging  
Rotating log files with logrotate

Using systemdand journalctl

**Auditing User Login**

lastlog command defined

 Reports the most recent login of all users or of a given user

$ lastlog

List to see who logged in

Each user is listed once

grep -v

Search for line that does not match switch

$ lastlog | grep -v "Never"

this will list all lastlog lines that do not match "Never"

Last command defined

Last searches back through the file /var/log/wtmp (or the file designated by the -f flag) and displays a list of all users logged in (and out)

$ last

Listing all users

$ last -n 10

List all users in the first 10 rows

-n switch indicates how many lines  
The output with the latest on top

(5+04:45) indicates system was logged on for that period of time.  5 days 4 hours and 45 minutes

$ last | grep "still"

This command will list all users still logged in, searching for "still" keyword

$ last reboot

List all reboot events in system

$ last dansaf

List all events by user

-n switch can also be applied last -n 10 dansaf

lastb command

Similar to the last command but reads from a different file and will require sudo privileges

List all bad login attempts

**Auditing Root Access**

var/log directory as root user

Location of several types of log files

btmp file in log directory

Temp log of all bad login attempts

wtmp file in log directory

Temp file of good login attempts

$ ls secure\*

Listing secure files from the log directory

secure file holds all current secure outputs, other files from previous dates rotated with a date extension

$ less secure

List all events including sudo events

$ grep sudo secure\*

List all sudo events from any of the secure files'

This assists in creating an audit trail of the events that occurred with the sudo action. Displays the user who attempted, date, time and the action

$ grep su: secure\*

This command is similar except isolates the sudo type to "su:"

**Scripting with AWK to Analyse Logs**

awk defined

Short for "Aho, Weinberger, and Kernighan," **AWK** is an interpreted programming language which focuses on processing text

awk [ -F fs ] [ -v var=value ] [ 'prog' | -f progfile ] [ file ... ]

awk process

The essential organization of an AWK program follows the form:

*pattern* { action}

The pattern specifies when the action is performed. Like most UNIX utilities, AWK is line oriented. That is, the pattern specifies a test that is performed with each line read as input. If the condition is true, then the action is taken. The default pattern is something that matches every line. This is the blank or null pattern. Two other important patterns are specified by the keywords "BEGIN" and "END". As you might expect, these two words specify actions to be taken before any lines are read, and after the last line is read

/var/log directory as root

tail -n1 secure

List the last section of the secure file

$ awk '/sudo/ { print $0 } ' secure

Command that can be included in a script to search for lines that contain "sudo" in the secure file. $0, referred to as filed 0, is supposed to the whole line

$ awk '/sudo/ { print $5 } ' secure

Same as above but would display only field 5

$ awk '/sudo/ { print $5, $6 } ' secure

List will include field 6

Possibly of including more fields using comma as delimiter

$ vi secure.sh in root home directory (Perform this on the master server)

#!/usr/bin/bash

awk "/$1/ { print \$5,\$6,\$14 }" $2

chmod +x secure.sh

Setting executable rights

./secure.sh su: /var/log/secure

This will execute the script with su: /var/log/secure

**Configuring System Logging**

rsyslogd defined

Rsyslogd  is a system utility providing support for message logging.  Sup-port of both internet and unix domain sockets enables this utility to sup-port both local and remote logging.

Quicker than syslogd and easily integrated with other systems

# cd /etc

Where the rsyslogd resides

# ls rsyslogd

Lists items with "rsyslog"

# vi rsyslog.conf

This is the configuration file for rsyslogd daemon

Update file to include a demo condition

local1.info /var/log/Dany

Restart and check status of service

# systemctl restart rsyslog

# systemctl status rsyslog

Create a log entry for testing purposes

# logger -p local1.warn "Test Message"

# tail /var/log/messages

This log entry will be accessible to view in messages file

**Rotating Log Files**

Particularly usefule for rotating files so the size doesn't get too large to manage

/etc/cron.daily

Location of the daily log rotate executable

/etc/logrotate.conf

Configuration file used for defaults and create new defaults along with specifying permissions

vi !$ or vi /etc/logrotate.conf

Specify the file name with the parameters in the brackets

/var/log/dany {

missingok

notifempty

size 10

compress

}

Activate logrotate process using the logrotate.conf as the parameter

logrotate /etc/logrotate.conf

**Managing Logs with journalctl**

journalctl Defined

journalctl may be used to query the contents of the systemd journal as written by systemd-journald.service. If called without parameters, it will show the full contents of the journal, starting with the oldest entry collected

# journalctl

Display all the messages with the ability to scroll through using page up and down

# journalctl -n

List the last 10 messages

# journalctl -n 15

List the last 15 messages

# journalctl -f

-f switch will list and follow the messages in real time as they happen until the job is stopped or terminated

# journalctl -b

-b switch lists information since the last boot which is also the default behavior

# journalctl -u sshd.service

-u switch to list all messages by the service indicated. In this case, the sshd service

# journalctl --since "2016-06-01 12:00:00"

--since switch will list all messages from the date and or time indicated

year-day-month hours:mins:sec

# journalctl --since "10 minutes ago"

List messages from shorter periods if needed

# journalctl --until

The --until switch can be used with the same parameters as "since"

# mkdir /var/log/journal

Create a directory to place log files

vi /etc/systemd/journald.conf

Location of the journald configuration file. Make changes to the file to manage the location of where the log resides

Change the storage configuration

Storage=persistent

# journalctl --list-boots

List all boot messages. Current file is referred to as file "0", previous files are before and will contain a minus

# journalctl -b -1

-1 switch indicates to list from the minus 1 file

no switch will indicate to look at the current file

**Log Files and Logrotate Module Recap**

Audit Logins

lastlog

List all users and when they last logged in

last

Login activity and system restarts during a period of time

lastb

List all failed login and bad login attempts

Audit Root Access

/var/log/secure

Location of file

sudo:

Search for keyword sudo:

su:

Search for su: keywords

awk

awk ‘/sudo/ { print $5,$6,$14 }’ /var/log/secure

search for lines that contain sudo and print lines 5, 6, 14

rsyslogd

/etc/rsyslog.conf

Configuration file location

local1.info /var/log/service\_log

Specifying the local1

systemctl restart rsyslogd.service

Any changes require restart

logger -p local1.warn “Script ended”

Using the logger command on the local1 for }script ended"

Rotate Logs

/etc/logrotate.conf

Configuration file location

/etc/cron.daily/

Script setup to run logrotate once daily

logrotate /etc/logrotate.conf

Changes to the time on which to run logrotate can be configured in this location

journalctl

# journalctl

List all messages

# journalctl -n

List specific amount

# journalctl --since “10 minutes ago”

List specific amount

# journalctl -b

Choose latest or whichever boot file

# mkdir /var/log/journal

Make files persisitent

**Introducing SELinux Module Introduction**

Exam Objectives

List and identify SELinux/AppArmorfile and process

Configure and modify SELinux/AppArmorpolicies

Demo

View SELinux status and understand modes and context

View SELinux logs

Working with SELinux booleans

Working with SELinux ports

**View SELinux Modes and Context**

SELinux Defined

NSA Security-Enhanced Linux (SELinux) is an implementation of a flexible mandatory access control architecture  in  the Linux  operating system. The  SELinux  architecture provides general support for the enforcement of many  kinds  of  mandatory  access  control  policies, including  those  based  on  the concepts of Type Enforcement®, Role-Based Access Control, and Multi-Level Security

# ls -Z

 Display  security  context  so it fits on most displays.  Displays only mode, user, group, security context and file name

# getenforce

Get the current mode of SELinux

#sestatus

This  tool  is used to get the status of a system running SELinux. It displays data about whether SELinux is enabled or disabled,  location of key directories, and the loaded policy with its status

Current mode setting can be "enforcing" or "permissive" and "disabled"

# cat /etc/selinux/config

Configuration file that controls the state of SELinux on the system

Default SELinux type is targeted

# setenforce 0 or setenforce permissive

Configure the mode to permissive

# setenforce 1 or setenforce enforcing

Configure the mode to enforcing

# id -Z

List settings, similar to ls -Z but more detailed

# ps -Z

List permissions based on processes

# ps -Zp $(pgrep sshd)

List setting by service, in this case, sshd

**Audit Logs and SELinux Commands**

# tail /var/log/audit/audit.log

Audit log which contains all the information regarding SELinux

In Permissive mode, the error message will be added as permissions are available

ausearch Defined

ausearch  is  a  tool  that  can  query the audit daemon logs based for events based on different  search  criteria

# ausearch -m avc

List by message type, in this case it will execute for "avc" message type

chcon Defined

Change the SELinux security context of each FILE to CONTEXT. With --reference, change the security context of each FILE to that of RFILE

# chcon -t unlabeled\_t /etc/shadow

Change the context of the type to unlabeled in shadow directory

restorecon Defined

This program is primarily used to reset the security context (type) (extended attributes) on one or more files

# restorecon /etc/shadow

Restore defualt setting to ssh back in as dansaf

semanage Defined

semanage is used to configure certain elements of SELinux policy without requiring modification to or recompilation from policy sources

# semanage fcontext -l

List file contexts

# semanage fcontext -l | grep /etc/shadow

**Working with SELinux Booleans**

getsebool Defined

Reports where a particular SELinux boolean or all SELinux booleans are on or off In certain situations a boolean can be in one state with a pending change to the other state. getsebool will report this as a pending change. The pending value indicates the value that will be applied upon the next boolean commit

# getsebool -a

List all SELinux booleans

# getsebool -a | wc -l

List the count of boolean types

# semanage boolean -l

Descriptive list which includes the state, default state and description of the boolean type

# getsebool httpd\_read\_user\_content

List specific by boolean type, in this case, list the httpd\_read\_user\_content type

setsebool Defined

sets the current state of a particular SELinux boolean or a list of booleans to a given value. The value may be 1 or true or on to enable the boolean, or 0 or false or off to disable it

# setsebool httpd\_read\_user\_content on

Switch the httpd\_read\_user\_content to on

This will  only work in the current session. There is no persistence with this configuration change

# getsebool httpd\_read\_user\_content

Check to ssee if the change was made to the boolean value

# ls -l /etc/selinux/targeted/policy

This confirms the change will only be applied to the session as the policy file itself remains to be updated

# setsebool -P httpd\_read\_user\_content on

-P switch will permanantly change the boolean value

Policy will be updated once the change is made. This may take alittle longer

# ls -l /etc/selinux/targeted/policy

Verify the policy file

**Working with SELinux Ports**

semanage port Defined

semanage port controls the port number to port type defitions

# semanage port -l

List port type definitions

# semanage port -l | grep ssh

List port specific to keyword search, in this case searching for port ssh

# semanage port -a -t ssh\_port\_t -p tcp 2222

This will set the port from "22" to "2222"

# ls -l /etc/selinux/targeted/policy

This will verify whether or not the policy has been updated from the last change

**Recap Introduction to SELinux Module**

SELinux Status

# sestatus

Check the status of SELinux currently and the default settings

# getenforce

Check current status

# setenforce 0 | 1

Change during runtime using this command

Both enforcing anf logging SELinux policies

/etc/selinux/config

Mode and default setting in config location

$ ls -Z <file>

List SELinux labels and type values

$ ps -Z <pid>

Processes specific

$ id -Z

ID Specific

Logs and Commands

/var/log/audit/audit.log

FIle location and file that is being logged

# ausearch -m avc

List avc events

# ausearch -m avc -ts recent

Recent avc events

# restorecon /etc/shadow

Restore a file using this command

# chcon -t shadow\_t /etc/shadow

Manually change using this command

Booleans and Ports

# getsebool -a

List all possible booleans

# semanage boolean -l

List by name

# setsebool httpd\_read\_user\_content on

Change value

# -P To make permanent  
# semange port -l

List ports and the values

# semange port -at ssh\_port\_t -p tcp 1111

Change values using this command

Recommended to use value above 1024 for ports

**Managing Software on CentOS 7 Module**

Exam Objectives

Update operating systems to provide required functionality and security

Update software to provide required functionality and security

Update the kernel and ensure the system is bootable

Manage the startup process and services

Update packages from the network, a remote repository, or from the local file system

Install software from source

Demo

Software management with RPM

Using YUM

Working with YUM Repositories

Working with the YUM Cache

Controlling kernel updates

Working with services

Working with source code

**Using the RPM Command**

# yum install httpd

"Is this ok" prompt has three choices yes, no, download only

Packages are downloaded in the yum cache

# tree /var/cache/yum

List of packages for each version

# cd /var/cache/yum/x86\_64/7/base/packages/

Location of version 7 yum packages

# ls

Lists all the packages installed for the version

Each has a name, version, operating system and the platform

.rpm extension (Red Hat Package Manager)

# rpm -qa

Lists all the packages on the system that are currently installed (query all)

# rpm -qa | grep http

Lists the package corresponding to the keyword, in this case, http

# rpm -qa | wc -l

Lists the count of the amount of packages installed in the system

# rpm -qi

-i switch indicates installed packages

-i switch will also list dependencies and if the package has already been installed

# yum install nmap

# rpm -qi nmap

Command will display details of package including dependencies

# rpm -ql nmap

-ql switch will list the contents of the package indicated, in this case, nmap

#rpm -qpi httpd-tools-2.4.6-40.e17.centos.x86\_64.rpm

Query Red Hat package that hasn't been installed using the -i switch to indicate information

#rpm -qpl httpd-tools-2.4.6-40.e17.centos.x86\_64.rpm

Query list using -l switch

# rpm -e nmap

-e switch will remove the package

This command will remove the nmap package

# rpm -qf /etc/hosts

This will check for which package is used for the file

# rpm -qi setup

This will give information on the setup package

# rpm -V setup

The -V switch will verify the package

**Using YUM**

YUM Defined

yum is an interactive, rpm based, package manager. It can automatically perform system updates, including dependency analysis and obsolete processing based on "repository" metadata. It can also perform installation of new packages, removal of old packages and perform queries on the installed and/or available packages among many other commands/services. yum is similar to other high level package managers like apt-get and smart.

# yum install bash-completion

The process always begins with checking the cache for the package

"noarch" for architecture indicates that this package is based off a script

"base" Repository is the repository type

# yum install -y bash-completion

-y switch indicates the "yes" will automatically be filled during install

After the install, the package will not activate until logout and back in using su -l

Pressing the Tab key twice will display the response suggestions and will complete if there is only one

bash-completion package

Very useful package for bash and shell scripting

# yum info bash-completion.noarch

Yum command will provide information on the package

# yum version

Displays information on the version

# yum install httpd

will install the httpd package

# yum install -y nmap

Install the package with auto yes

# yum remove nmap

Command to prompt the removal process of a package

#yum deplist nmap

List all dependencies for the nmap package

# yum list installed

List all installed packages including where the package originated

# yum list available

List all available packages that can be installed through the repository systems

**Configuring YUM Repositories**

# cd /etc/yum.repos.d/

Directory which defines all the repositories

# ls

.repo extension files are in this location

# yum info epel-release.noarch

List details regarding the epel package that was initially installed in the beginning. This package includes all the necessary dependencies for yum package manager

# rpm -ql epel-release

Lists all the dependencies, specifically for "epel-release" including the signing key and repo files

# less CentOS-Base.repo or # cat CentOS-Base.repo

Construction of the package CentOS-Base which contains repository id, name and description, mirror lists, enable public key check

"Enabled=0" indicates that the package was not installed

# yum repolist

List repositories currently installed with repo name, id and description

# yum repolist all

List all enabled and disabled repositories located in configuration

#ls

#mv C\* /root

Moving all CentOS repos to the root directory from /etc/yum.repos.d/

vi local.repo

In the same directory, create a local.repo

Insert the ID, no spaces or special characters between the ID

[CentOS7]

name=CentOS 7.2 Mirror Network

baseurl=http://mirror.cisp.com/CentOS7/7/os/x86\_64/

enabled=1

grgcheck=0

I had trouble configuring the local network and used the mirror network for this example. This is because the original location used in teh demo had an apache server installed which will be required to follow the demo

# yum repolist

Once the configurations are complete, this command will verify the paths and display the total packages

# yum remove nmap

To test the configuration, remove nmap to prepare for reinstall

# yum install -y nmap

This command will take from the newly created repo for CentOS7

# nmap 192.168.56.101

Test to verify the command

**Working with the yum cache**

/etc/yum.repos.d

The configurations for the repos

# tree /var/cache/yum

List out in tree format the yum directory in cache

# yum makecache

 Useful to ensure the cache is up to date with all metadata

Checks all repositories to ensure latest

# yum clean all

Purge repositories for any downloaded packages and cached data

# !t or # tree /var/cache/yum

After investigating both tree commands. The latest tree command has less files and is cleaner

# yum makecache

Metadata is downloaded again from the repositories and added into the cache. Because the clean command was executed

# !t or # tree /var/cache/yum

More files should be added as the makecache command was executed with a cleaner repo list

When yum is executed, the cache is the location in where the system will search which speeds the process significantly

**Controlling Kernel Updates**

Note - Apache server installation and configuration required

The demo requires the repo setup with an Apache server

# ls /var/www/html/centos7/

This was used previously and if created, will list all files

# cd /etc/yum.repos.d

Directoy for configurations

# ls

Verifying current configurations

vi local.repo

Creating a new repo file called "local.repo"

[local]

name=local repo

baseurl=file:///var/www/html/centos7local/

enabled=1

gpgcheck=0

# yum repolist

This is to verify the configurations have been correct. If there is an error, may have to do with the references

# rm local.repo

Once tests have passed, remove the local.repo

# uname -r

Display current kernel

# yum update kernel

Command to install the updated kernel which gives total of 2 kernels

# yum update

Upgrade using yum which installs new kernels. This may have been done already if the vm has already been updated

# vi /etc/yum.conf

Add extra configuration in this file

This file will be used for all yum processes. Any configuration change in this file will apply for everything until the configuration file is updated

exclude=kernel\*

This addition will exclude the kernel as part of this yum process

# yum update --exclude kernel\*

This command will also update excluding all kernels

**Using Source RPM's**

# yum repolist all

List all active and inactive repositories

# grep ^enabledcCentOS-Sources.repo

Search and list all current .repo files for any that lines that begin with "enabled"

sed -i 's/^enabled=0/enabled=1/' CentOS-Sources.repo

This command will automatically set the configurations for CentOS-Sources from enabled=0 to enabled=1

# grep ^enabledcCentOS-Sources.repo

Running this command again will give the list and each setting should be changed from 0 to 1 to indicate that the setting is active

# yum repolist or # yum repolist all

after the change the repositories should be displayed after executing the command

All the source rpm's have now been activated. Source RPM's have the source code

# yum install -y yum-utils

This command will install the yum downloader

# cd

Ensure you are as root and in the root directory before using the downloader

yumdownloader Defined

Program for downloading RPMs from Yum repositories

# yumdownloader --source zsh

--source switch is used when requires the use of source files instead of binaries

# ls - Once completed, will create a .src.rpm package in the current directory

# yum install ncurses-devel

ZSH Shell requires this additional file before using

# rpm -i zsh-5.0.2-14.e17.src.rpm

The command is looking for a rpm build and will also build the rpmbuild directory

# cd rpmbuild/SOURCES/

The source code will be located in the SOURCES directory

when checking this directory using # ls, there should be a tar.bz2 file which needs to be extracted

# tar -xjf zsh-5.0.2..tar.bz2

The file will be extracted in the current directory and will have a configure executable script

# cd zsh-5.0.2

# ./configure

This executable will check the environment and setup everything needed

Makefile Defined

Makefiles are special format files that together with the *make* utility will help you to automagically build and manage your projects

Make keeps track of the last time files (normally object files) were updated and only updates those files which are required (ones containing changes) to keep the sourcefile up-to-date

# make

Execute the file for ZSH Shell, will run from the makefile

Compiling the c code

make install Defined

The make program takes the binaries from the previous step and copies them into some appropriate locations so that they can be accessed. Unlike on Windows, installation just requires copying some libraries and executables and there is no registry requirement as such. In short, "make install" just copies compiled files into appropriate locations.

# make install

Compile and deploy code

# zsh

Execute the zsh shell

# which zsh

Check to see that it was installed through the directory

**Managing Services**

# yum install httpd

Checks to see if httpd has been installed

# nmap localhost

Verify that the localhost is up

# systemctl status httpd.service

This should display that the service is not running

# systemctl enable httpd.service

This command is to enable the service

# systemctl status httpd.service

Will still result as inactive, this is because it requires to be started

# systemctl status httpd.service

Start the service

# systemctl status httpd.service

Should indicate the service is active

# nmap localhost

Now should result in the ability to listen on port 80

**Managing Software on CentOS 7 Module Recap**

Using RPM

# rpm -qa

List software

# rpm -i <name.rpm>

Install

# rpm -e nmap

Remove package

# rpm -ql nmap

List contents of installed package

# rpm -qpl <name.rpm>

List what was contained in the rpm file

# rpm -V nmap

Verify installed package

Using YUM

# yum install bash-completion

Install tool for CentOS

# yum install -y bash-completion

Install with auto "yes"

# yum info bash-completion

Info about package

# yum list installed

List all installed packages

# yum remove bash-completion

Remove including all dependencies

YUM Repos

/etc/yum.repos.d/

Location of repositories

# yum repolist

List repositories that are enabled

# yum repolist all

List all repositories

YUM Cache

/var/cache/yum

Location of cache

# yum clean all

Purge all metadata in cache

# yum makecache

New cache

Kernel Updates

When updating kernel yum will install new kernel and does not update the original

Install new kernels

This makes it safe as we can always choose to boot to the original kernel

Allow ability to load the original kernel, the new kernel becomes the default kernel

If we do not want to install new kernels or upgrade the existing we can add exclude=kernel\* to /etc/yum.conf

Exclude Kernel updates using the command

Source RPMs

# Enable repos in CentOS-Source.repo

Source repositories

# yum install -y yum-utils ncurses-devel

Add additional tools

# yum groupinstall “Development Tools”

Perform a group install, several programs

# yumdownloader --source zsh

Download the source files using yumdownloader

# rpm -i zsh...src.rpm

Extract using the -i switch full path

# cd ~/rpmbuild/SOURCES

Location of the source directories

# tar -xjf zsh..tar.bz2

Extraction process for .bz2

# cd zsh... ; ./configure

Create the make file

# make ; make install

Execute the make file move binaries in the proper directories

Services

# systemctl enable httpd.service

Necessary to enable to strat on system boot

# systemctl start httpd.service

Start now

# systemctl status httpd.service

Check status

**Configuration Management Tools Module Introduction**

Exam Objectives

Maintain systems via configuration management tools

Demo

Install puppet

Checking the hostname and domain

Puppet manifests

Running puppet

**Installing Puppet**

Puppet Defined

Central configuration management tool

Open source and runs on most Unix-like systems  
Puppet is model driven requiring little programming knowledge to use

Puppet itself comes from the epel repository

Policies can be created and are called manifest files through a central configuration or local servers

# su -

# cp /root/c\* /etc/yum.repos.d/

Moving back the repositories from teh previous module

# yum install puppet

There is over 17 dependencies and one package

# puppet --version

Determine the version of puppet

**Extracting Data with factor**

hostnamectl Defined

CentOS is system d based and can accept the hostnamectl command

hostnamectl may be used to query and change the system hostname and related settings. This tool distinguishes three different hostnames: the high-level "pretty" hostname which might include all kinds of special characters (e.g. "Lennart's Laptop"), the static hostname which is used to initialize the kernel hostname at boot (e.g. "lennarts-laptop"), and the transient hostname which is a default received from network configuration. If a static hostname is set, and is valid (something other than localhost), then the transient hostname is not used

# hostnamectl

Confirms the hostname configured on system

facter Defined

One of the puppet dependencies

Collect  and  display  facts  about the current system. The library behind Facter is easy to expand, making Facter an easy way to collect information about a system from within the shell or within Ruby

# facter

Command by itself will result in a detailed spec of the system

# facter | grep hostname

Results in providing strictly hostname info

# facter | grep domain

Results in specific domain info

# facter | grep virtual

List virtualization information

**Using and Applying Manifests**

Manifests Defined

Puppet programs are called manifests. Manifests are composed of puppet code and their filenames use the .pp extension. The default main manifest in Puppet installed via apt is /etc/puppet/manifests/site.pp

# cd /etc/puppet/

Location of the configuration

mkdir manifests

Create a manifests directory

# cd manifests

# pwd

Normally these would be created in the puppet server. Created this directory to simulate the server on the local

# vi site.pp

node "server1.example.com" {

file { '/etc/yum.repos.d/local.repo':

ensure => "file",

owner => "root",

group => "wheel",

mode => "644",

content => "[localc7]

name=CentOS 7 Local  
gpgcheck=0

enabled=1

Define a node or where to execute the action or a file that is referenced

Define what is going to happen using the "{}"

Define rules, actions, processes inside the brackets

Use of the colon to add attributes in the file

# cd /etc/yum.repos.d/

# rm local.repo

Remove the current local repo file

# cd -

Take back to previous directory

# puppet apply site.pp

Execute the manifest

# cd -

# ls

Go back and check to see if local.repo was added

# cat local.repo

Verify that the local.repo contains the body

# yum repolist

List the repos including the created repo from the manifest

# puppet apply site.pp

Execute more quickly as it was executed before

**Configuration Management Tools Module Recap**

Install Puppet

Enable EPEL repo

Enable or route

# yum install -y puppet

Ruby based product

Several ruby dependencies

$ puppet --version

Running facter

We can use facter to pass configuration detail to Puppet

# facter | grep hostname

See hostname

# facter | grep domain

Excrapulate domain

Manifests

/etc/puppet/manifests/site.pp

Location of manifests on local

Apply rules for files, packages, services and users

# puppet apply site.pp

Execute where the manifest from the directory