EX-200 - Red Hat Certified System Administrator

Setup

home vms
192.168.50.164
192.168.50.105

work vms
192.168.17.185
192.168.17.208

Hyper V
Disable secureboot

Proxmox (host) (wget command from red hat site) > download > f12 > network > copy cURL > trim it
curl -L \
  -H "Cookie: rh_user_id=59291558; rh_sso_session=1" \
  -o /var/lib/vz/template/iso/rhel-9.8-x86_64-dvd.iso \
"
https://access.cdn.redhat.com/content/origin/files/sha256/c0/c0dd53b73406b85b40d6168d1748e605d71361b2992d282c408b7d7d2e1d2c80/rhel-9.8-x86_64-dvd.iso?_auth_=1782441220_ea8eb57265125e48f98b84775988df3d"

- 64gb disk
- 2048mb memory
- 1 socket 2 cores cpu
- Disable screen timeout

RHC Connect (needed package commands)
rhc connect
username: nicksotelo95
password: RHELsolomon5!!

Set Hostname

hostnamectl set-hostname <hostname>

Set Time Zone

timedatectl
timedatectl list-timezones
timedatectl set-timezone Australia/Sydney
timedatectl set-ntp true
chronyd

/etc/chronyd.conf
/var/log/chrony
systemctl start/restart chronyd
chronyc
rpm -qa | grep chrony
ps -ef | grep chrony
yum install chrony

Subscription Manager Repositories

sudo dnf repolist
subscription-manager repos --enable codeready-builder-for-rhel-10-x86_64-rpms
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-10.noarch.rpm
sudo dnf install --nogpgcheck https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -E %rhel).noarch.rpm https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-$(rpm -E %rhel).noarch.rpm

Remote access without password (SSH keys)

- Needed for repetitive logins or automation via scripts
- Keys are generated at the user level (nsotelo, root)
- Install Git Bash for ssh-copy-id to work
- Powershell will then connect without password

ssh-keygen
ssh-copy-id root@192.168.1.X
ssh root @192.168.1.X
ssh -1 root 192.168.1.X

File Management

Linux Filesystems: ext3 ext4 xfs	Windows Filesystems: NTFS FAT

Basic Commands

pwd	print working directory
cd	home directory (/home/
~	
/	root directory (absolute
~	user directory (/home/u
.	current directory (relati
..	parent directory (relativ
ls	show directory contents
ls -l	show directory contents
ls -ltriah	show directory contents -l long -t time sorted (-S for siz -r reverse -i inode -a show hidden files -h human readable file

Type	# of Links	User Owner	Group Owner	Size	Month
drwxr-xr-x	21	root	root	4096	Feb
lrwxrwxrwx	1	root	root	7	Feb
-rw-r--r--	1	root	root	0	Mar

d	directory
l	link
• •	file

Creating Files and Directories

touch filename	create file
rm filename	remove file
cp filename newfilename	copy file and name it

cp filename /tmp	copy file to /tmp
mv filename newfilename	renames the file
mv filename /tmp	move file to /tmp
mkdir dirname	create directory
rmdir dirname	remove directory
rm -r dirname	remove directory
chgrp groupowner filename	change the group owne
chown userowner filename	change the user owner
chown userowner:groupowner filename	change the user and gr
echo "text" > filename	insert the text into the
cat filename	display the text inside a
rm -rf	remove directory (recu

Searching Files and Directories

nsotelo@rhel-pc:~$ find / -name "*ksh*.rpm" 2> /dev/null	search file or directory .rpm stderr to /dev/null
grep -ri error /var/log/ 2> /dev/null | wc -l	search file contents for -c count -r recursive -i ignore case sensitive -n display matched line -v all except specified wc -l word count lines o
egrep -i "keyword|keyword2" /var/log	search file contents for
find /etc -name "*.conf*" | xargs grep "localhost"	combination of above c

Soft and Hard Links

- inode - pointer of number of a file on the disk
- soft link - link will be removed if the file is removed or renamed
- hard link - deleting, renaming or moving the original file will not affect the hard link

ln /location/file	creates a hard link to a
ln -s /location/file	creates a soft link to a f

Soft Link Example (Hard link may return an error has both directories must be on the same partition)

touch hulk
cd /tmp
/tmp $ ln -s /home/nsotelo/hulk

Input and Output Redirects (3 Redirects)

standard input (stdin) with file descriptor 0	ls >> listings
standard output (stdout) with file descriptor 1	cat < listings

standard output (stdout) with file descriptor 1	mail -s "office memo" allusers@abc.com < memoletter
standard error (stderr) with file descriptor 2	ls -l /root 2> errorfile

Pipes

Connects the output of one command directly to the input of another command

ls -ltr | more

ll | tail -l

File Editor

Text Editors

ed (1969)	original, line oriented standard for Unix
ex (1976)	extended, more powerful version of ed, introduced visual mode for full screen editing
vi (1976)	classic, efficient, visual full-screen editor built on top of ex
emacs (1976)	extensible, feature-rich ecosystem known for customisation
pico (1992)	simple, menu-driven, for non-technical users
vim (1991)	improved and advanced version of vi, with modern features
nano (1999)	free open source clone of pico

echo 'export EDITOR=nano' >> ~/.bashrc	set nano as default text editor

vi commands

mode	command	function
esc	up down left right	move around
esc	shift g	move to end
esc	**i**	**insert**
insert	a	move right (and insert)
insert	o	line break (and insert)
insert	**esc**	**escape current mode**
esc	x	remove character
esc	r	replace character
esc	d	delete line
esc	u	bring line back
esc	:q!	quit
esc	:wq!	save and quit

nano commands

ctrl o	save current file (write out)
ctrl s	save current file (newer versions)
ctrl x	exit
ctrl w	search for specific text
ctrl \	find and replace
ctrl k	cut line
ctrl u	paste line
alt 6	copy line
alt n	toggle line numbers
alt u	undo
alt e	re-do

Shell Scripting

Basic Commands

echo $0	display current shell
cat /etc/shells	display available shells
cat /etc/passwd	shows user's shell e.g. /bin/bash)
chmod a+x scriptname	allow execute permission for all
./script.bash	execute from current location
/home/userdir/script.bash	execute from absolute location

Script Components

#!/bin/bash	Shell
# This script does X	Comment
echo/cp/grep	Command
if/while/for	Statement

Basic Scripts

nano output-screen

#!/bin/bash # made by Nicolas Sotelo whoami echo pwd echo hostname echo ls -ltr echo	shell comment command space

Variables

nano variable-command

#!/bin/bash a=Nicolas b=Sotelo c="Linux Class" echo "My first name is $a" echo "My surname is $b" echo "My class is $c"	 set variable echo variable

Input / Output

#!/bin/bash echo Hello, my name is Nicolas Sotelo echo echo What is your name? read namevar echo echo Hello $namevar echo	 read sets variable to the user's input echo variable
#!/bin/bash hnvar=`hostname` echo Hello, my hostname is $hnvar echo echo What is your name? read namevar echo echo Hello $namevar echo	 set variable as a `command` echo the output

If-then scripts

#!/bin/bash count=100 if [$count -eq 100] then echo Count is 100 else echo Count is not 100 fi	 end if statement
#!/bin/bash clear if [-e /home/nsotelo/error.txt] then echo "File exists" else echo "File does not exist" fi	

#!/bin/bash clear echo echo "What is your name?" echo read namevar echo echo Hello $namevar sir echo echo "Do you like working in IT? (y/n)" read Like echo if ["$Like" == "y"] then echo "You are cool" elif ["$Like" == "n"] then echo "You should try IT, it's a good field" echo fi	

For Loops

- Runs repeatedly until specified number of variables is met

#!/bin/bash for numv in 1 2 3 4 5 do echo "Welcome $numv times" done	assign 1 2 3 4 5 to numv repeatedly display for each of the specified
#!/bin/bash for actionv in eat run jump play do echo "See Nick $actionv" done	assign eat run jump play to actionv repeatedly display for each of the specified
#!/bin/bash for filev in {1..5} do touch $filev done	for loop to create 5 files named 1-5

#!/bin/bash for filev in {1..5} do rm $filev done	for loop to delete 5 files named 1-5
#!/bin/bash i=1 for day in Mon Tue Wed Thu Fri do echo "Weekday $((i++)): $day" done	specify days in for loop
#!/bin/bash a=1 for username in `awk -F: '{print $1}' /etc/passwd` do echo "Username $((a++)) : $username" done	list all users one by one from /etc/passwd fil

Scheduled Tasks

Scheduled Tasks

crontab -e	scheduled jobs -e edit
at.	scheduled one-off jobs
* * * * *	minute (0-59), hour (0-23), day (1-31), month (week day (0-6)
0 * * * * /home/user/script	run every hour (at minute 0)
0 0 * * * /home/user/script	run every day (at midnight)
30 4 * * 1 /home/user/script	run every Monday at 4:30 AM

Access Control

File Permissions

- Files that are not scripts do not have x
- Directories have x because users can cd into them

d	r	w	x	r	w	x	r	w	x
file type	user read	user write	user execute	group read	group write	group execute	others read	others write	others execute

Numerical Permissions

- 7 = rwx
- 6 = rw
- 5 = rx
- 4 = r
- 2 = w
- 1 = x
- 0 = -

Owner of parent directory can still delete a file or subdirectory

chmod u+rw filename	user owner add rw
chomd g-w filename	group owner remove w
chmod o+r filename	others owner add r
chmod a-r filename	all owners remove r
chmod 755 filename	set to rwxr-xr-x
chmod 644 filename	set to rw-r--r--
chmod 777 filename	set to rwxrwxrwx
chown root filename	change user owner to root
chgrp root filename	change group owner to root

Access Control Lists (ACLs)

ACLs are used to grant rwx to a specific user, separate from the file owners

- As you assign ACL to a file/directory it adds + to the permissions
- w permission does not include delete (owner only)

getfacl	get file acl
setfacl -m u:user:rwx filename	add permissions for user
setfacl -m g:group:rw filename	add permissions for group
setfacl -rm u:user:rwx dirname	add permissions for directory -r recursive
setfacl -x u:user filename	remove entry for user

setfacl -b filename	remove all entries

SELinux

User Management

useradd username	create a user
groupadd groupname	create a group
userdel username	delete user
groupdel groupname	delete group
usermod username	modify user

Important files

/etc/passwd
/etc/group
/etc/shadow

Examples

useradd -g superheros -s /bin/bash -c "User Description" -m -d /home/spiderman spiderman
(Creates spiderman in group superheros)

usermod -G superheros spiderman
(adds spiderman to superheros group)

chgrp -R superheros spiderman
(change the group owner of user to superheros)

cat /etc/passwd	display users, passwords (x=encrypted), UserID, GroupID, etc.

defaults file located in /etc/login.defs

Example:

chage -m 5 -M 90 -W 10 -I 3 -E 20656 babubutt

-d	(lastchanged) days since 01/01/1970 that password was last changed
-m	minimum number of days between password changes
-M	maximum number of days before user must change password
-W	number of days before password is to expire before user is warned

-I	number of days after password expires that account is disabled
-E	days since 01/01/1970 that account is disabled (account end date)

grep babu /etc/shadow
babubutt:!:20626:5:90:10:3:20656:

Switching users and sudo access

su -	switch to root
su - username	switch user
sudo	perform action as super user
visudo	

Add user to sudoers

su -	switch to root
usermod -aG wheel nsotelo	adds nsotelo to wheel group
grep wheel /etc/group	verify
su - nsotelo	switch back to nsotelo
sudo fdisk -l	run sudo command to confirm access

Processes & Services

Basic Definitions

- Application
- Script
- Process
- Daemon
- Thread
- Job

Monitoring and Management

command	example	fu
fdisk	fdisk -l	di -l
df	df -h	fil
du	du -sh /var/log	si
uptime	uptime -p	sh pa
top	top -h	di
free	free -h	di (h
lsof	lsof -l :80	lis
tcpdump	tcpdump -i eth0	ca
ps	ps -ef --forest	cu
kill	kill 34448	te

top

PID	USER	PR	NI	VIRT	RES	SHR	S	%

top columns

PID	id
USER	owner
PR	focus priority
NI	the nice value
VIRT	amount of virtual memory i
RES	amount of resident memory
SHR	amount of shared memory

S	status of the process
%CPU	share of CPU time since last
%MEM	share of physical memory in
TIME+	total cpu time used (hundre
COMMAND	name of the command or pr

Services & Daemons

systemctl --version	Check if systemd is installe
ps -ef | grep system	check if systemd is running
systemctl --all	check all running services
systemctl status/stop/start/restart application.service	status/start/stop/restart app
systemctl reload application.service	reload the config of a servic
systemctl enable/disable application.service	enable or disable at boot
systemctl mask/unmask application.service	enable or disable completel

Logs

/var/log	log directory

Log Examples

- boot (overwrites upon boot)
- chonyd / ntp
- cron
- maillog
- secure
- messages
- httpd

Technical Support and Analyse Server

sos report	collect logs and configuratio enter case ID compressed archive saved t mm-dd-filename.tar.xz
cockpit	web based server administr can monitor system resourc and perform other tasks also available centos redhat
ping www.google.com	verify internet connectivity
rpm -qa | grep cockpit	verify cockpit package is ins
yum/dnf install cockpit -y	install cockpit (Red Hat, Cer
apt-get install cockpit	install cockpit (Debian, Ubu

systemctl start/enable cockpit.socket	relies on socket activation t enable socket listener whicl when someone tries to conr
systemctl status cockpit	verify
systemctl status cockpit.socket	verify
https://192.168.50.164:9090	firewall may need to be disa access the web-interface log in with root or system u can be used to manage eve

System Performance

Tuned - dynamic system duning daemon

rpm -qa | grep tuned	check if tuned package is installed
yum install tuned	install tuned
systemctl status/enable/disable tuned	check tuned status
tuned --version	check tuned version
tuned-adm	change setting for tuned daemon
tuned-adm active	check which profile is active
tuned-adm list	list available profiles
tuned-adm profile profile-name	change profile
tuned-adm recommend	recommended profile
tuned-adm off	turn off tuned daemon
https://192.168.X.X:9090	change via web console (check firewall)

nice and renice

- Prioritise CPU tasks
- Range from -20 (highest priority) to 19 (lowest priority), 0 = default
- System priorities range 0-139, (0-99 real time), (100-139 users)

top	check active process priority PR = System priority (-99 to 39) NI = nice value (-20 to 19)
ps axo pid,comm,nice,cls --sort=-nice	show active processes nice, by nice
nice -n # process-name	set process value
nice -n -12 top	set top nice to -12
renice -n -15 <PID>	change active PID nice to -15 (open 2nd terminal to check)

Networking

Basic Commands

ip a	Display network inerfaces and detail
ip addr	
ifconfig	
hostnamectl set-hostname <hostname>	set hostname

NetworkManager

systemctl status NetworkManager	NetworkManager status
ps -ef | grep NetworkManager	verify the process is running

Network Configuration Methods

nmcli	network manager cli (non gui) can b
nmtui	text user interface
nm-connection-editor	full gui (via desktop/console)
GNOME Settings	

Creating a New Connection Profile

nmcli con add type ethernet ifname enp0s3 con-name myprofile1

Use nmcli to set a static ip address

nmcli connection	connected interface UUIDs
nmcli device	connected interface status
nmcli connection modify eth0 ipv4.addresses 192.168.50.X/24	set static ip address & subnet mask
nmcli connection modify eth0 ipp4.gateway 192.168.50.1	set gateway
nmcli connection modify eth0 ipv4.method manual	configure eth0 to use static IP
nmcli connection modify eth0 ipv4.dns 8.8.8.8	set dns to google.com
nmcli connection down eth0 && nmcli connection up eth0	restart nmcli connection
ip address show eth0	

Use nmcli to configure a secondary static ip address

nmcli device status	
nmcli connection show --active	

ifconfig	
nmcli connection modify eth0 +ipv4.addresses 192.168.50.X/24	set secondary static ip address & su
nmcli connection reload	
systemctl reboot	
ip address show	

nmtui (as root), nm-connection-editor or GNOME settings can also be used to activate a connection or change hostname

Network Files and Basic Commands

/etc/sysconfig/network-scripts	
/etc/hosts	
/etc/hostname	
/etc/resolv.conf	DNS
/etc/nsswitch.conf	Set to use Files or DNS first (per com
ping	verify network connectivity to a host
ifconfig	Display network interfaces and IP ad
ip addr	
ip a	
ifup	brings up a specific network interfac
ifdown	takes down a specific network interf
traceroute	track and display path (router/hops) destination host
tcpdump -i eth0	capture and monitor live network tra
nslookup	queries DNS servers to find ip adddr
dig	
ethtool eth0	display and modify physical hardwar speed & duplex

Network Firewall

Software Firewall (Runs on operating system)
Hardware Firewall (Dedicated network appliance)

Modern Linux uses firewalld as front-end & nftables under the hood

- organises rules by "zones" (like public, home, work)
- exposes "services" (ssh, http, samba, nfs, etc.)

- temporary changes are active immediately but vanish on reload
- permanent changes are written to disk and take effect after reload

- unspecified zone will be applied to the default (public) zone
- services not on --list-all can be enabled by opening specific ports

Examples

- allow SSH from jump hosts
- open http/https on a web server
- open NFS only to specific subnets

Exam

- start/enable firewall, list zones, see active zone
- add/remove a service, open/close a port, make changes permanent
- reload, create, or use a custom XML service file
- basic rich rules for blocking a specific source IP
- temporary vs permanent

Basic Commands

dnf install firewalld	install firewalld
systemctl enable --now firewalld	enable now
systemctl status firewalld	details & uptime
firewall-cmd --state	status
firewall-cmd --get-active-zones	
firewall-cmd --get-zones	list predefined zone profiles • public - locked down defa • home/work - more relaxe • trusted - allow all • dmz/external - edge scen
firewall-cmd --list-all	shows all zones, rules, service (only a few enabled by default
firewall-cmd --get-services	list of all services that can be a

firewall-cmd --reload	write permanent and custom r

Adding and Removing Temporary and Permanent Rules

firewall-cmd --add-service=http	add http (temporary)
firewall-cmd --permenent --add-service=http	add http (permanent)
firewall-cmd --remove-service=http	remove http (temporary)
firewall-cmd --permanent --remove-service=http	remove http (permanent)
firewall-cmd --add-port=1110/tcp	add TCP 1110 (temporary)
firewall-cmd --permanent --add-port=1110/tcp	add TCP 1110 (permanent)
firewall-cmd --remove-port=1110/tcp	remove TCP 1110 (temporary)
firewall-cmd --permanent --remove-port=1110/tcp	remove TCP 1110 (permanent)
firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.0.25" reject'	block incoming packets from s
firewall-cmd --remove-rich-rule='rule family="ipv4" source address="192.168.0.25" reject'	remove
firewall-cmd --add-icmp-block=echo-request	block icmp (ping) requests
firewall-cmd --remove-icmp-block=echo-request	remove
firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 203.0.113.10 -j DROP	block outgoing packets to IP (p
firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 0 -d 203.0.113.10 -j DROP	remove
nano /etc/firewalld/services/sap.xml xml version="1.0" encoding="utf-8" <service version=1.0"> <short>SAP</short> <description>Third-party application service</description> <port protocol="tcp" port="3200"/> </service>	add a custom service definitior xml header for firewalld
firewall-cmd --get-services | grep -i sap	verify
firewall-cmd --add-service=sap	add the custom rule (temporar
firewall-cmd --permanent --add-service=sap	add the custom rule (permane

dnf install -y httpd	install apache web service
systemctl enable --now httpd	
curl -I http://localhost	verify (403 Forbidden)
curl -I http://192.168.50.164/	verify (403 Forbidden)

Archive & FTP

Archive and Compress Files

tar cvf nsotelo.tar /home/nsotelo	archive all folders in /home/nsotelo
mkdir new	create a directory 'new'
mv nsotelo.tar new	move archive to 'new'
cd new	change to 'new' directory
gzip nsotelo.tar	compress nsotelo.tar
gzip -d nsotelo.tar.gz	uncompress nsotelo.tar.gz
tar xvf nsotelo.tar	extract archive

File Transfer Protocol

remote host:

rpm -qa | grep ftp	check if the required package is installed
ping www.google.com	verify internet connection
yum install vsftpd	install the remote package
nano /etc/vsftpd	disable anonymous login uncomment ascii_upload_enable=YES uncomment ascii_download_enable=YES ftpd_banner=Welcome to blah FTP sevice add to the end of the file "use_localtime=YES" save & exit
systemctl start vsftpd	
systemctl enable vsftpd	
systemctl stop firewalld	
systemctl disable firewalld	

client host:

yum install ftp	install the client package
ftp 192.168.50.105	log in to the remote host via ftp enter username and password of an active rer account
bi	switch to binary mode (data integrity)
hash	display progress visually
put filename	transfer a file
bye	exit ftp

Secure Copy Protocol (Better security and authentication)

- Uses Port 22 (same as SSH)

client host:

touch filename	create a file
scp filename jsmith@192.168.50.105:/home/jsmith	securely copy the file to /home/jsmith on remo enter jsmith password

Software & Packages

RHC Connect (needed for package commands)

rhc connect -u username -p password	cc

username: nicksotelo95
password: RHELsolomon5!!

Packages are Collections of Files

- RPM = Red Hat Package Manager tool
- CentOS & RHEL packages are in .rpm format

dnf	in
dnf install chrony -y	in
dnf install httpd -y	in
rpm -qa | wc -l	cc -q -a w -l
dnf install bind	in
cat /etc/redhat-release	ch
dnf update -y	up
dnf install ksh -y	in
dnf remove ksh -y	re

Manual Package Installations

wget <download link address>	rp
find / -name *ksh*.rpm 2>/dev/null	ve
rpm -ihv ksh*.rpm	KS -i -h -v
wget <download link for chkconfig depdency>	
rpm -ihv chkconfig*.rpm	in
rpm -qa | grep ksh	ve -q -a
rpm -qi ksh	m

rpm -e ksh	re

Offline Package Installation

- Building a local repository from an iso
- https://mirror.stream.centos.org/9-stream/BaseOS/x86_64/iso/

wget -O ~/iso-downloads/centos-9.iso <.iso url>	dc
mkdir -p /mnt/rhel-dvd	cr -p
mount -t iso9660 -o loop,ro ~/iso-downloads/centos-9.iso /mnt/rhel-dvd	m -t -o ro
df -h	ve -h
mkdir -p /localrepo	cr
cp -rv /mnt/rhel-dvd/BaseOS/Packages/* /localrepo	ur -r -v
cp -rv /mnt/rhel-dvd/AppStream/* /localrepo	ur
dnf install createrepo_c -y	in
createrepo /localrepo/	sc sc
cd /etc/yum.repos.d	gc
mkdir backup	ba
mv *.repo backup	m er
nano local.repo [localrepo] name=localrepo baseurl=file:///localrepo/ enabled=1 gpgcheck=0 metadata_expire=-1	cr re hu ex ht er di Di
dnf clean all	de
dnf repolist	lis
dnf install httpd -y	ve

Containers

Containers allows developers to test and build applications on any computer by putting it in a container bundled with software code, libraries, and config files

Docker

- software used to create and manage containers
- can be installed on Linux - daemon can be controlled natively
- not supported in RHEL 8

Podman

- developed by Red Hat 2018 as an alternative to docker
- daemon less, open source, linux native tool to develop, manage and run containers

Red Hat provides a set of CLI tools

- podman - directly managing pods and container images (run, stop, start, ps attach, etc)
- buildah - for building, pushing, and signing container images
- skopeo - copying, inspecting, deleting, signing images
- runc - providing container run and build features to podman and buildah
- crun - optional runtime that can be configured and give greater flexibility, control & security for root

- images - containers can be created through images and containers can be converted to images
- pods - groups of containers deployed together on the host

Basic Commands

dnf install podman -y	install podman
alias docker=podman	used when switching from doc
podman -v	check podman version
podman info	check podman environment (s if loading an image, looks loca

Image --> Run Container --> Remove it

podman search httpd	search container registries for
podman pull docker.io/library/httpd	download image from a registr
podman images	list downloaded images
podman run -d --name web1 -p 8080:80 docker.io/library/httpd	create & start a container from
podman ps	list running containers
curl localhost:8080 (or web) http://localhost:8080	verify application is accessible
podman stop web1	stop a container gracefully
podman rm -f web1	remove a container

Image --> Create Container --> Start Container

podman pull docker.io/library/httpd	download image from a registr
podman create --name web2 -p 8081:80 docker.io/library/httpd	create a container
podman ps -a	verify status is created -a all containers
podman start web2	start an existing container
podman ps	verify container status is runni
curl localhost:8081 (or web) http://localhost:8081	verify application is accessible
podman logs web2	view output generated by cont

Quadlet (Modern RHEL10) systemd container management

mkdir -p /etc/containers/systemd	create directory for quadlet de
nano /etc/containers/systemd/web3.container ... [Unit] Description=Apache Web Container [Container] Image=docker.io/library/httpd ContainerName=web3 PublishPort=8082:80 [Service] Restart=always [Install] WantedBy=multi-user.target	create quadlet container defin define systemd metadata description define container-specific settir specifies container image assigns name maps host port to container pc defines systemd behaviour automatically restart containe definte boot settings allow service to start automati
systemctl daemon-reload	reload systemd configuration
systemctl enable --now web3.service	enable service at boot and sta
systemctl status web3.service	verify service status
podman ps	verify container status is runni
curl localhost:8082 (or web) http://localhost:8082	verify application is accessible

Storage & LVM

Types of Storage

- Local Storage
- SAN (Storage Area Network)
- NAS (Network Attached Storage)

Basic Commands

fdisk -l	disk partition utility (displays disk system n -l list disks
lsblk -f	list block devices (displays UUID) -f forest (tree) hierarchy
df -h	filesystem capacity (displays mounted disks -h human readable

Add a New Disk

insert a new 2GB disk	
fdisk /dev/sdb	enter fdisk utility for the new disk
n	new partition
enter (default)	new standalone disk
enter (default)	first sector 2048 last sector 67108863 (whole disk = 1 partition)
enter (default)	
w	write (apply)
mkfs.xfs /dev/sdb1	create xfs filesystem for the disk
mkdir /data	create directory to mount the disk to
lsblk -f	retrieve UUID
mount /dev/sdb1 /data	mount the disk to the directory
nano /etc/fstab	add the below line to mount during boot (sp ~~/dev/sdb1 /data xfs defaults 0 0~~ UUID=4461ddd4-6c2c-4078-ab90-d451436 xfs defaults 0 0
reboot	verify successful boot
umount /data	to unmount
mount -a	mount disks again according to /etc/fstab

LVM (Logical Volume Management)

- Allows disks to be combined together via software
- Volume groups can be extended by adding additional disks

Physical Volume	Volume Group	Logical Volume	Mounted On
Disk 1	rootvg	system	/
		home	/home
		swap	
Disk 2	datavg	data1	/data1
Disk 3		data2	/data2
Disk 4		data3	/data3
		data4	/data4

Add Disk and Create LVM Partition

File system	datafs		
Logical Volume(s)	datalv		
Volume Group	datavg		
Physical Volume	/dev/sda1	/dev/sdb1	/dev/sdc1
Partitions	/dev/sda1	/dev/sdb1	/dev/sdc1
Hard Disks	/dev/sda	/dev/sdb	/dev/sdc

Basic Commands

pvdisplay	display phsyical volumes
pvs	summary physical volumes
vgdisplay	display volume groups
vgs	summary volume groups
lvdisplay	display logical volumes
lvs	summary logical volumes

Creating this Structure

insert a new 1GB disk	
fdisk /dev/sdc	enter fdisk utility for the new disk
n	new partition
enter (default)	new standalone disk
enter (default)	first sector 2048 last sector 67108863 (whole disk = 1 partition)
enter (default)	
p	show partitions
t	change partition type
L	show LVM hex codes
8e	Linux LVM

w	write (apply)
pvcreate /dev/sdc1	create physical volume
vgcreate vg_app /dev/sdc1	create volume group
lvcreate -n lv_app --size 1020M vg_app	create logical volume (allowing overhead)
mkfs.xfs /dev/vg_app/lv_app	create filesystem
blkid /dec/vg_app/lv_app	retrieve UUID
mkdir app	create directory
mount /dev/vg_app/lv_app /app	mount
nano /etc/fstab	UUID=12345678-abcd-1234-efgh-1234567 defaults 0 0

Extending this Space (this can only be done with LVM)

insert a new 1GB disk	
fdisk /dev/sdd	enter fdisk utility for the new disk
n	new partition
enter (default)	new standalone disk
enter (default)	first sector 2048 last sector 67108863 (whole disk = 1 partition)
enter (default)	
p	show partitions
t	change partition type
8e	Linux LVM
w	write (apply)
pvcreate /dev/sdd1	create phsyical volume
vgextend vg_app /dev/sdd1	extend vg_app to new disk
lvextend -L+1020M /dev/mapper/vg_app-lv_app	extend by 1020MB lv_app
xfs_growfs /dev/mapper/vg_app-lv_app	extend filesystem for lv_app

Stratis Pools

Stratis

- Red Hat 8 introduced Stratis volume management
- Uses thin provisioning (starts with 546Mb)
- Combines LVM and filesystems onto one system
- Stratis automatically extends the filesystem if space is available

dnf install stratis-cli stratisd
systemctl enable/start stratisd
lsblk -f
stratis pool list
stratis filesystem list

Creating a Two-Disk Stratis Pool and Filesystem

insert two new 5GB disks
stratis pool create pool1 /dev/sde
stratis pool add-data pool1 /dev/sdf
stratis filesystem create pool1 fs1
mkdir /bigdata
mount /dev/stratis/pool1/fs1 /bigdata
stratis filesystem snapshot pool1 fs1 fs1-snap
nano /etc/fstab … UUID=a6dca899-feab-4663-8028-fe6ce748d268 /bigdata xfs defaults,x-systemd.requires=stratisd.servi 0 0

Network File Shares

Network file sharing (Linux to Linux)

- A client's mounted NFS export appears as a local directory

NFS - Exam Scenario

1. install nfs uitilities
2. enable the nfs server service
3. add a correct line to /etc/exports
4. apply it with exportfs -arv
5. allow the nfs service in the firewall
6. mount on the client
7. mount persistent in /etc/fstab with options like _netdev & vers=4.2

NFS - Server 192.168.17.185

dnf install nfs-utils -y	install nfs utilities
systemctl enable --now nfs-server	start and enable nfs service
mkdir -p /srv/nfsshare	create NFS share directory
chmod -R 0777 /srv/nfsshare	open permissions (0777 for lab)
semanage fcontext -a -t public_content_rw_t "/srv/nfsshare(/.*)?"	assign selinux context
restorecon -Rv /srv/nfsshare	apply selinux context
ls -Zd /srv/nfsshare	verify selinux context
nano /etc/exports ... /srv/nfsshare 192.168.17.0/24(rw,sync,no_root_squash)	export share to subnet with read-write a
exportfs -arv	reload and verify exports
firewall-cmd --add-service=nfs --permanent	allow nfs through firewall
firewall-cmd --add-serivce=rpc-bind --permanent	NFSv4 alone works over port 2049, but and legacy tools need rpc-bind and mou
firewall-cmd --add-service=mountd --permanent	
firewall-cmd --reload	reload firewall rules
exportfs -v | grep /srv/nfsshare	verify export configuration

NFS - Client 192.168.17.208

ip a	confirm subnet
showmount -e 192.168.17.185	verify server is exporting
sudo dnf install nfs-utils -y	install nfs utilities
sudo mkdir -p /mnt/nfsshare	create nfs mount directory

sudo mount -t nfs -o vers=4.2 192.168.17.185:/srv/nfsshare /mnt/nfsshare	mount nfs share
df -h | grep nfsshare	verify
echo "Created from client" | sudo tee /mnt/nfsshare/client.txt >/dev/null	test write access
sudo nano /etc/fstab ... 192.168.17.185:/srv/nfsshare /mnt/nfsshare nfs _netdev,vers=4.2,rw 0 0	persistent nfs mount entry
sudo systemctl daemon-reload	reload systemd config
sudo mount -a	test fstab config
reboot	verify mount survives

Samba (Linux to Windows)

- Implements SMB/CIFS (windows file sharing)
- Allows Linux to share a folder that Windows clients can map
- Linux clients can connect using CIFS
- Define a share in /etc/samba/smb.conf
- With SELinux, you need to label the path you are sharing
- Set proper file permissions and ACLs
- Ensure SELinux is enforcing with right context on shared directories

Samba - Exam Scenario

1. install Samba
2. add a share block in smb.conf and verify wih testparm
3. label the directory using semanage fcontext and restorecon
4. allow the samba service through the firewall
5. start and enable the smb daemon
6. test with smbclient
7. mount using CIFS
8. make persistent in /etc/fstab

Samba - Server 192.168.17.185

dnf install -y samba policycoreutils-python-utils	install samba server and selinux tools
mkdir -p /srv/sambashare	create samba share directory
echo "This is a samba file" > /srv/sambashare/readme.txt	create test file
chmod -R 0777 /srv/sambashare	open permissions (0777 for lab)

nano /etc/samba/smb.conf … [sambashare] path = /srv/sambashare browsable = yes writable = yes guest ok = yes read only = no	define samba share configuration
testparm -s	validate samba config
semanage fcontext -a -t samba_share_t "/srv/sambashare(/.*)?"	assign selinux context
restorecon -Rv /srv/sambashare	apply selinux context
ls -Zd /srv/sambashare	verify selinux context
firewall-cmd --add-service=samba --permanent	allow samba through firewall
firewall-cmd --reload	reload firewall rules
systemctl enable --now smb	start and enable samba service
systemctl status smb --no-pager	verify status

Samba - Client 192.168.17.208

sudo dnf install -y samba-client cifs-utils	install samba client tools
smbclient -L //192.168.17.185 -N	list available samba shares
smbclient //192.168.17.185/sambashare -N	connect to samba share
ls	list files in share
get readme.txt	download file from share
quit	exit smbclient session
sudo mkdir -p /mnt/sambashare	create mount directory
sudo mount -t cifs //192.168.17.185/sambashare /mnt/sambashare -o guest	mount samba share
ls -l /mnt/sambashare	verify share contents
echo "Client wrote this via Samba" | sudo tee /mnt/sambashare/client.txt >/dev/null	create file on share
sudo nano /etc/fstab … //192.168.17.185/sambashare /mnt/sambashare cifs _netdev,guest 0 0	configure persistent mount
sudo mount -a	test fstab config
reboot	verify mount survives

Boot Modes & Recovery

New Boot Process (CentOS/RHEL 7 and above):

- systemd manages boot sequence
- backward compatible with SysV init scripts from previous versions

BIOS - Basic Input/Output System (firmware interface)
POST - Power-On-Self-Test started

- MBR - Master Boot Record
  - information saved in first sector of hard disk that indicates where the GRUB2 is located so it can b
    - GRUB2 - Grant Unified Boot Loader v2 - loads linux kernel
      /boot/grub2/grub.cfg
- Kernel - Core of the operating system - loads required drivers from initrd.img starts first OS process
  systemd
- Systemd = System Daemon (PID #1) then starts all required processes
  read /etc/systemd/system/default.target to bring system to run level (7 run levels 0-6)

How to Reboot / Shutdown

systemctl poweroff	stops all services, unmounts fil
poweroff	
systemctl reboot	all of the above plus reboot
reboot	

Select systemd target

- 0-6 run levels now referred to as targets
- important targets:
  - graphical.target - system supports multiple users, graphical and text based logins
  - multi-user.target - system supports multiple users, text based logins only
  - rescue.target - sulogin prompt, basic system initialisation completed
  - emergency.target - sulogin prompt, initramfs pivot complete and system root mounted on / read
- a target can be part of another target (graphical.target includes mult-user.target)

systemctl get-default	check current target
who -r	run level
systemctl list-dependencies graphical.target | grep target	list target dependencies
ls -l /lib/systemd/system/runlevel*	list run levels
systemctl set-default graphical.target	set default target

Recover Root Password (RHEL 9)

Reboot	
Press esc to get to grub	
Find line beginning with Linux, add " rd.break" to the end	RHEL 9 boot into emergency re
Find line beginning with Linux, add "init=/bin/bash" to the end	RHEL 10 boot into emergency
Press Ctrl X	start boot using the modified s
mount -o remount,rw /sysroot	RHEL 9 remount the root filesy
mount -o remount,rw /	RHEL 10 remount the root files
chroot /sysroot	change root to /sysroot (RHEL
passwd	
Enter your new password	
touch /.autorelabel	Force SELinux to relabel files o
exit	RHEL 9 exit back to emergency
mount -o remount,ro /	RHEL 10 reboot cleanly
exit	exit to boot

Repair Filesystem Corruption
Common errors

Corrupt file system	systemd attempts to repair file if it cannot be repaired, go to e
nonexistent device/UUID in /etc/fstab	if device remains unavailable,
nonexistent mount point in /etc/fstab	go to emergency shell
incorrect mount option in /etc/fstab	go to emergency shell

Emergency target can diagnose and fix the issue, because no file systems are mounted before emergency shell is displayed

systemctl daemon-reload	when using emergency shell, r /etc/fstab