# RHCSA 9 Appendix 1

## H. Stromburg

06/02/2024

# Version History

| Date | Author | Version | Changes made | Rationale |
| --- | --- | --- | --- | --- |
| 07/07/24 | Heidi Stromburg | 0.1 | * Initial draft | Committed to drop box as v0.107\_07\_24 |

Table of Contents

[RHCSA 9 Appendix 1 1](#_Toc171259168)

[H. Stromburg 1](#_Toc171259169)

[Version History 2](#_Toc171259170)

[General 14](#_Toc171259171)

[Terminal Short Cuts 14](#_Toc171259172)

[Putty 14](#_Toc171259173)

[Exam 15](#_Toc171259174)

[Day of the Exam 15](#_Toc171259175)

[Second Sander Video Course 15](#_Toc171259176)

[My Experience 16](#_Toc171259177)

[Recover root - Sander Video Course 1 16](#_Toc171259178)

[Create the two main repos - Sander Video Course 1 17](#_Toc171259179)

[General 17](#_Toc171259180)

[dd 17](#_Toc171259181)

[/etc/fstab 17](#_Toc171259182)

[BaseOS.repo 17](#_Toc171259183)

[Intro - Sander Video 0 18](#_Toc171259184)

[Understanding RHEL - Sander Video 1 18](#_Toc171259185)

[Installing RHEL - Sander Video 2 18](#_Toc171259186)

[Install Pair of Servers for Lab Work - First Sander Video Course 18](#_Toc171259187)

[Installing using custom partitioning 19](#_Toc171259188)

[Lab - Installing RHEL - Sander Video 2 19](#_Toc171259189)

[Lab Solution- Installing RHEL - Sander Video 2 19](#_Toc171259190)

[Using the CLI - Sander Video 3 20](#_Toc171259191)

[Bash Shell 20](#_Toc171259192)

[Gnome 20](#_Toc171259193)

[Virtual Terminal 20](#_Toc171259194)

[Lab - CLI - Sander Video 3 21](#_Toc171259195)

[Lab Solution - CLI - Sander Video 3 21](#_Toc171259196)

[Exploring Essential Tools - Sander Video 4 21](#_Toc171259197)

[man 21](#_Toc171259198)

[vi 22](#_Toc171259199)

[Lab Essential Tools - Sander Video 5 23](#_Toc171259200)

[Lab Solution Essential Tools - Sander Video 5 23](#_Toc171259201)

[Understanding the bash shell - Sander Video 5 24](#_Toc171259202)

[Redirection and Piping 24](#_Toc171259203)

[history 24](#_Toc171259204)

[keyboard shortcuts 25](#_Toc171259205)

[Shell expansion 25](#_Toc171259206)

[Escaping Special Characters 26](#_Toc171259207)

[Applying Variables 26](#_Toc171259208)

[alias 26](#_Toc171259209)

[Tuning the bash environment 27](#_Toc171259210)

[lab - bash shell - Sander Video 5 27](#_Toc171259211)

[lab solution- bash shell - Sander Video 5 27](#_Toc171259212)

[File Management Tools - Sander Video 6 28](#_Toc171259213)

[Filesystem Hierarchy Standard (FHS) 28](#_Toc171259214)

[Finding Files 28](#_Toc171259215)

[mounting filesystems 29](#_Toc171259216)

[Links 29](#_Toc171259217)

[Archiving Files / tar 30](#_Toc171259218)

[Working with Compressed Files 30](#_Toc171259219)

[lab file management tools - sander video 6 30](#_Toc171259220)

[lab solution file management tools - sander video 6 31](#_Toc171259221)

[Managing Text Files - Sander video 7 31](#_Toc171259222)

[grep 31](#_Toc171259223)

[regex 32](#_Toc171259224)

[awk 33](#_Toc171259225)

[sed 34](#_Toc171259226)

[lab solution - Sander video 12 - text files 34](#_Toc171259227)

[lab solution Sander video 7 - text files 34](#_Toc171259228)

[Root Privileges - Sander Video 8 35](#_Toc171259229)

[su and sudo 35](#_Toc171259230)

[General 35](#_Toc171259231)

[switching user with su 35](#_Toc171259232)

[Performing admin tasks with sudo 35](#_Toc171259233)

[Using ssh to log in remotely 36](#_Toc171259234)

[lab 8 using root privilege - sander video 36](#_Toc171259235)

[lab 8 using root privilege - sander video 36](#_Toc171259236)

[Users and Groups - Sander video 9 37](#_Toc171259237)

[Creating and Managing Users 37](#_Toc171259238)

[Defining Default User Settings 37](#_Toc171259239)

[Limiting User Access 38](#_Toc171259240)

[Managing Group Membership 38](#_Toc171259241)

[settings password properties 39](#_Toc171259242)

[Lab for Users and Groups - Sander Video 9 40](#_Toc171259243)

[Lab Solution for Users and Groups - Sander Video 9 40](#_Toc171259244)

[Securing files with Permissions - Sander Video 10 41](#_Toc171259245)

[ownership 41](#_Toc171259246)

[Basic Permission 41](#_Toc171259247)

[Shared Group Directories 42](#_Toc171259248)

[Applying Default Perm - umask 43](#_Toc171259249)

[Lab Permission - Sander Video 10 43](#_Toc171259250)

[Lab Solution - Sander Video 10 43](#_Toc171259251)

[Networking - Sander Video 11 44](#_Toc171259252)

[ipv4 44](#_Toc171259253)

[ipv6 44](#_Toc171259254)

[Nic Naming 45](#_Toc171259255)

[Defining Host Names and Resolution 45](#_Toc171259256)

[Analyzing Network Configuration 46](#_Toc171259257)

[Network Manager 46](#_Toc171259258)

[nmcli for persistent connections 47](#_Toc171259259)

[nmtui 48](#_Toc171259260)

[Troubleshoot Networking 48](#_Toc171259261)

[Lab - Networking - Sander video 11 49](#_Toc171259262)

[Lab Solution - Networking - Sander video 11 49](#_Toc171259263)

[Managing Software - Sander video 12 50](#_Toc171259264)

[rpm commands 50](#_Toc171259265)

[dnf commands 51](#_Toc171259266)

[Modules (in RH8, but not in RH9 so not on exam) and Application Streams 52](#_Toc171259267)

[dnf history 52](#_Toc171259268)

[subscription manager - not on exam 52](#_Toc171259269)

[Lab - Sander Video - Packages 12 53](#_Toc171259270)

[Lab Solution - Sander Video - Packages 12 53](#_Toc171259271)

[Monitoring Activity - Sander Video 13 54](#_Toc171259272)

[Jobs 54](#_Toc171259273)

[Process States 54](#_Toc171259274)

[ps 55](#_Toc171259275)

[Monitoring Memory 56](#_Toc171259276)

[Monitoring cpu load 57](#_Toc171259277)

[top 58](#_Toc171259278)

[Processes 58](#_Toc171259279)

[Repos 59](#_Toc171259280)

[Lab Process / Monitoring - Sander Video 13 59](#_Toc171259281)

[Lab Solution Process / Monitoring - Sander Video 13 59](#_Toc171259282)

[Managing Processes - Sander Video 14 59](#_Toc171259283)

[Signals 59](#_Toc171259284)

[Managing Process Priority 61](#_Toc171259285)

[tuned 62](#_Toc171259286)

[Managing user sessions and processes 64](#_Toc171259287)

[lab managing processes - sander video 14 64](#_Toc171259288)

[lab solution managing processes - sander video 14 65](#_Toc171259289)

[Sander video 15 - systemd 65](#_Toc171259290)

[units 66](#_Toc171259291)

[service units 66](#_Toc171259292)

[modifying systemd service configuration 66](#_Toc171259293)

[managing unit dependencies 67](#_Toc171259294)

[masking services 68](#_Toc171259295)

[Lab systemd - Sander Video 15 68](#_Toc171259296)

[Lab solution systemd - Sander Video 15 68](#_Toc171259297)

[Task Scheduling - Sander Video 16 68](#_Toc171259298)

[systemd timers 68](#_Toc171259299)

[Scheduling Tasks with Cron 70](#_Toc171259300)

[anacron 70](#_Toc171259301)

[at 70](#_Toc171259302)

[Managing Temporary Files 71](#_Toc171259303)

[lab - schduling - Sander Video 16 72](#_Toc171259304)

[lab solution - scheduling - Sander Video 16 72](#_Toc171259305)

[Configuring Logging - Sander Video 17 72](#_Toc171259306)

[Using journald 73](#_Toc171259307)

[Preserving journald and making it persistent 74](#_Toc171259308)

[Configuring rsyslog 74](#_Toc171259309)

[log rotate 75](#_Toc171259310)

[lab logging - Sander Video 17 76](#_Toc171259311)

[lab solution for logging - Sander Video 17 76](#_Toc171259312)

[Partitions and Mounts - Sander Video - Chapter 18 77](#_Toc171259313)

[General 77](#_Toc171259314)

[partitions 77](#_Toc171259315)

[mbr partitions 77](#_Toc171259316)

[gpt partitions 77](#_Toc171259317)

[fdisk 77](#_Toc171259318)

[gdisk 77](#_Toc171259319)

[parted 78](#_Toc171259320)

[block devices 78](#_Toc171259321)

[partition types 78](#_Toc171259322)

[create partitions with fdisk 78](#_Toc171259323)

[File Systems 79](#_Toc171259324)

[/etc/fstab 80](#_Toc171259325)

[UUIDs and Labels 80](#_Toc171259326)

[systemd mounts - don’t use on the exam 82](#_Toc171259327)

[Creating a swap partition 83](#_Toc171259328)

[Partitions and Mounts Lab - Sander Video 18 83](#_Toc171259329)

[Partitions and Mounts Lab Solution - Sander Video 18 84](#_Toc171259330)

[LVMs - Sander Video - Chapter 19 84](#_Toc171259331)

[General 84](#_Toc171259332)

[Overall Concept 85](#_Toc171259333)

[Creating an LVM 85](#_Toc171259334)

[Understanding Device Mapper and LVM Device Names 86](#_Toc171259335)

[Extents 86](#_Toc171259336)

[Resizing LVMs 86](#_Toc171259337)

[Reducing Volume Groups 87](#_Toc171259338)

[LVM Lab - Sander Video 19 89](#_Toc171259339)

[LVM Lab Solution - Sander Video 19 89](#_Toc171259340)

[Stratis - not on the exam (see screencap on phone) - Sander video 20 89](#_Toc171259341)

[How storage is organized 89](#_Toc171259342)

[Creating Stratis Volumes 90](#_Toc171259343)

[stratis snapshots 91](#_Toc171259344)

[lab - Sander stratis video 20 92](#_Toc171259345)

[lab solution - Sander stratis video 20 92](#_Toc171259346)

[Managing the Boot Procedure - Sander Video 21 93](#_Toc171259347)

[Steps 93](#_Toc171259348)

[grub2 - changing run time params 93](#_Toc171259349)

[grub2 - changing persistent params 94](#_Toc171259350)

[managing systemd targets 94](#_Toc171259351)

[Lab - Managing the Boot Procedure - Sander Video 21 95](#_Toc171259352)

[Lab Solution- Managing the Boot Procedure - Sander Video 21 95](#_Toc171259353)

[Troubleshooting RHEL - Sander Video 22 96](#_Toc171259354)

[Using Troubleshooting Modes 96](#_Toc171259355)

[Changing the lost root password 97](#_Toc171259356)

[Using the root debug shell 98](#_Toc171259357)

[Troubleshooting File System Issues 98](#_Toc171259358)

[Fixing Network Issues 99](#_Toc171259359)

[Managing Performance Issues 99](#_Toc171259360)

[Troubleshooting software issues 100](#_Toc171259361)

[Fixing Memory Shortage 100](#_Toc171259362)

[Consulting Red Hat Websites for Tips 100](#_Toc171259363)

[Lab Troubleshooting - Sander Video 22 101](#_Toc171259364)

[Lab Solution Troubleshooting - Sander Video 22 101](#_Toc171259365)

[Bash Scripts - Sander Video 23 101](#_Toc171259366)

[Lab for Bash Scripting - Sander Video 23 103](#_Toc171259367)

[Bash Lab Solution - Sander Video 23 104](#_Toc171259368)

[Managing ssh - Sander Video 24 104](#_Toc171259369)

[Key Pairs 104](#_Toc171259370)

[Caching Keys - Not on the Exam 105](#_Toc171259371)

[Caching Demo 105](#_Toc171259372)

[Defining SSH Client Config 105](#_Toc171259373)

[ssh server options 106](#_Toc171259374)

[scp 106](#_Toc171259375)

[sftp 106](#_Toc171259376)

[rsync 106](#_Toc171259377)

[Lab - Managing ssh - Sander Video 24 107](#_Toc171259378)

[Lab Solution - Managing ssh - Sander Video 24 107](#_Toc171259379)

[http - sander video 25 107](#_Toc171259380)

[General 107](#_Toc171259381)

[Lab - Sander Video 108](#_Toc171259382)

[Lab Solution - Sander Video 108](#_Toc171259383)

[SELinux - sander video 26 108](#_Toc171259384)

[selinux components 109](#_Toc171259385)

[File Context Labels 110](#_Toc171259386)

[Finding the Right Context 112](#_Toc171259387)

[Managing port access 113](#_Toc171259388)

[selinux booleans 113](#_Toc171259389)

[selinux log messages 114](#_Toc171259390)

[Troubleshooting selinux 114](#_Toc171259391)

[SE Linux Lab - Sander Video 26 115](#_Toc171259392)

[SE Linux Lab Solution - Sander Video 26 115](#_Toc171259393)

[Firewalls - Sander Video 27 116](#_Toc171259394)

[ss 116](#_Toc171259395)

[rhel firewalling 116](#_Toc171259396)

[firewalld 116](#_Toc171259397)

[firewall-cmd 117](#_Toc171259398)

[lab - firewalls sander video 27 117](#_Toc171259399)

[lab solution - firewalls sander video 27 117](#_Toc171259400)

[Automatic Installs - Sander Video 28 118](#_Toc171259401)

[kickstart 118](#_Toc171259402)

[Installing virtual machines 119](#_Toc171259403)

[Lab - Video 28 119](#_Toc171259404)

[Lab Solution - Video 28 120](#_Toc171259405)

[Configuring Time Services - Sander Video 29 120](#_Toc171259406)

[General 120](#_Toc171259407)

[timedatectl 120](#_Toc171259408)

[Managing an NTP Client - chronyd 121](#_Toc171259409)

[Lab Configuring Time Services - Sander Video 29 121](#_Toc171259410)

[Lab Solution - Configuring Time Services - Sander Video 29 121](#_Toc171259411)

[NFS - Sander book or video? 30 122](#_Toc171259412)

[General 122](#_Toc171259413)

[Manually mount nfs 122](#_Toc171259414)

[Unmount your manual mount and mount with /etc/fstab 123](#_Toc171259415)

[simple autofs 123](#_Toc171259416)

[Set Up 123](#_Toc171259417)

[autofs for home directories 125](#_Toc171259418)

[Set Up 125](#_Toc171259419)

[Containers - Sander - confirm video 31 126](#_Toc171259420)

[General 126](#_Toc171259421)

[Informational 126](#_Toc171259422)

[Troubleshooting Containers 127](#_Toc171259423)

[Build Image from containerfiles 128](#_Toc171259424)

[Persistent Storage / Bind Mounting & weird 27:27 thing 128](#_Toc171259425)

[starting containers as services - rootless 129](#_Toc171259426)

[starting containers as services - rootful 130](#_Toc171259427)

[Sander main example 130](#_Toc171259428)

[container lab - confirm this is from the video - don’t think this worked for me, suspect 27:27 because ghori doesn’t use that and his works 131](#_Toc171259429)

[solution to container lab 131](#_Toc171259430)

[Containers - Ghori 132](#_Toc171259431)

[General 132](#_Toc171259432)

[Set Up 132](#_Toc171259433)

[Search, Examine and Download and Image 133](#_Toc171259434)

[Delete an image 133](#_Toc171259435)

[Build an image from a container file 133](#_Toc171259436)

[run a container interactively and interact with it and delete it 134](#_Toc171259437)

[run a container and auto-remove it after running it’s command 134](#_Toc171259438)

[container with port mapping for httpd 134](#_Toc171259439)

[start, restart and remove a container 135](#_Toc171259440)

[set environmental variables 135](#_Toc171259441)

[Configure Persistent Storage 135](#_Toc171259442)

[configure a rootful container as a systemd service 136](#_Toc171259443)

[General 136](#_Toc171259444)

[configuration 136](#_Toc171259445)

[configure a rootless container as a systemd service 137](#_Toc171259446)

[General 137](#_Toc171259447)

[configuration 137](#_Toc171259448)

[Lessons 4 - Essential Tools - Sander Second Video 138](#_Toc171259449)

[Configuring Remote Repo Access 138](#_Toc171259450)

[Configuring Local Repo Access 139](#_Toc171259451)

[Finding Files 140](#_Toc171259452)

[Managing Permissions 140](#_Toc171259453)

[Lesson 5 - Creating a Shell Script - Sander Second Video 141](#_Toc171259454)

[Looping Constructs 141](#_Toc171259455)

[Conditionally Executing Code 143](#_Toc171259456)

[Lesson 6 - systemd - Sander Second Video 144](#_Toc171259457)

[systemd 144](#_Toc171259458)

[tuning 145](#_Toc171259459)

[Lesson 7 - Configuring Local Storage - Sander Second Video 146](#_Toc171259460)

[Managing MBR Partitions 146](#_Toc171259461)

[Managing GPT Partitions 147](#_Toc171259462)

[Managing LVM 148](#_Toc171259463)

[Managing Swap 148](#_Toc171259464)

[Lesson 8 - Managing File Systems - Sander Second Video 149](#_Toc171259465)

[Mounting Filesystems 149](#_Toc171259466)

[Managing autofs 150](#_Toc171259467)

[Resizing LVM Volumes 151](#_Toc171259468)

[Configuring Directories for Collaboration 152](#_Toc171259469)

[Lesson 9 - Deploying, Configuring and Maintaining Systems - Sander Second Video 153](#_Toc171259470)

[Scheduling Tasks 153](#_Toc171259471)

[Configuring Time Services 153](#_Toc171259472)

[Lesson 10 - Basic Management - Sander Second Video 154](#_Toc171259473)

[Restricting Network Access 154](#_Toc171259474)

[Configuring hostname resolution 155](#_Toc171259475)

[Lesson 11 - Users and Groups 155](#_Toc171259476)

[Managing user accounts and groups 155](#_Toc171259477)

[managing passwords 156](#_Toc171259478)

[Configuring super user access 157](#_Toc171259479)

[Managing Security - Sander Second Video 158](#_Toc171259480)

[Permissions 158](#_Toc171259481)

[Managing selinux 159](#_Toc171259482)

[Configuring ssh 160](#_Toc171259483)

[Managing selinux port context 161](#_Toc171259484)

[Lesson 13 - Managing Containers - Sander Second Video 162](#_Toc171259485)

[Container Management 162](#_Toc171259486)

[Managing Storage 163](#_Toc171259487)

[Containerfile 165](#_Toc171259488)

[systemd integration for containers 165](#_Toc171259489)

[Lesson 14 - Practice Exam - Sander Second Video 166](#_Toc171259490)

[Prepare the Practice Exam Environment 166](#_Toc171259491)

[Exam Tasks 167](#_Toc171259492)

[Practice Exam Solutions 172](#_Toc171259493)

[Task 1 172](#_Toc171259494)

[Task 2 173](#_Toc171259495)

[Task 3 173](#_Toc171259496)

[Task 4 175](#_Toc171259497)

[Task 5 176](#_Toc171259498)

[Task 6 177](#_Toc171259499)

[Task 7 177](#_Toc171259500)

[Task 8 178](#_Toc171259501)

[Task 9 179](#_Toc171259502)

[Task 10 180](#_Toc171259503)

[Task 11 181](#_Toc171259504)

[Task 12 182](#_Toc171259505)

[Task 13 183](#_Toc171259506)

[Task 14 184](#_Toc171259507)

[Grade the Exam 184](#_Toc171259508)

# General

* do the work as student or user1 with sudo -i
  + they will need to be added to the wheel group to do this
  + useradd -G wheel student
* His github
  + <https://github.com/sandervanvugt/rhcsa9>
    - it’s a hodge podge of stuff
    - has 2 mini exams
* InformIT
  + books and videos
  + hylie
  + 6l9

# Terminal Short Cuts

* copy
  + shift-control-c
* paste
  + shift-control-v
* find
  + shift-control-f
* new tab
  + shift-control-t
* new window
  + shift-control-n
* settings, key board, custom, custom, add short cut whatever
  + command: name gnome-terminal
  + press ctrl-alt-t
* increase font of the terminal
  + shift-control-+
  + need to redo this every time you log in

# Putty

* copy/paste
  + entirely mouse driven
  + select the text, let go, auto copied to clip board, no ctrl-c needed
  + to paste just right click
* if it double pastes
  + settings, terminal, local echo = force off

# Exam

* Note the exam objectives now
* Check them when you are about to take the exam
  + make sure they didn't change
  + make sure what Sander covers on the final video collection matches the current objectives because he says he covers them all
* What's not on the exam
  + stratis
  + vdo
* You can't take the exam on an apple computer because you need to boot from the thumb drive
* You will probably have to set up access to repos
* Grading per Sander
  + 300 total points
    - need 210 to pass
  + Results provided within 3 business days
    - usually much sooner
      * often within a few hours

# Day of the Exam

* run mandb
* settings, privacy, screenlock
  + blank screen delay => never
  + auto screen lock => never
* Set shortcuts for terminal

# Second Sander Video Course

* Grade scripts to check your work
* 2.5 hours for practice exam
  + run exam-grade.sh to grade it
* GH repo for course
  + https://gihub.com/sandervanvugt/rhcsa9-lab
* VM four course
  + 2 cpu
  + 4 GB ram
  + 20 GB disk as scsi
  + don't use nvme, vmware often uses that
  + After install, increate the size of the disk to 30GB
  + Add a second disk of size 20GiB
* Definitely use RH Linux!!!
* Post Install
  + create an account
    - student
    - password = password
      * do this for the grading thing
      * make him admin
  + power off
    - he's using vmware fusion
  + resize to 30 GB
    - later on you have free space on the hard drive
  + add 20 GB drive
    - checked advanced
    - pick scsi, drive names are easier
  + start vm

# My Experience

* I struggled to get container exercises done the first time, nfs too
* It felt like Wordscapes a little
  + You start off and think, wow I’m fucked, I can’t get started and I don’t know what to do
  + Then you get one thing, and that implies the next one
  + Like a hard Wordscapes. It’s hard until you get the first word, then there are hints and alot of the other words become obvious

# Recover root - Sander Video Course 1

* Reset the virtual machine or reboot
* at kernel menu do: e
* at end of the linux line add this param
  + init=/bin/bash
* ctrl-x
  + mount -o remount,rw /
  + passwd
    - change password
  + touch /.autorelabel
* Run this command to be elegant
  + exec /usr/lib/systemd/systemd
  + to not be elegant
    - use vmware reset
* At reboot should be able to log in

# Create the two main repos - Sander Video Course 1

## General

* Now is a good time to set the host name if you didn’t set it in the install
  + hostnamectl set-hostname blah.example.com
  + validate
    - hostnamectl => blah.example.com
    - prompt won’t update until later
* Do the below on both servers or have this built into a snapshot
* If you have to redo the two repo files
  + just delete them
  + dnf clean all
  + start again

## dd

* dd if=/dev/sr0 of=/rhel9.iso bs=1M
  + if you forget what sr0 is
    - lsblk
* allow ~10 minutes for this
* dd stuff appears as loop in lsblk

## /etc/fstab

* Make the directory to mount the iso on
  + mkdir /repo
* vi /etc/fstab
  + /rhel9.iso /repo iso9660 defaults 0 0
* mount it
  + mount -a
  + systemctl daemon-reload
  + mount -a
* validate
  + ls /repo
  + confirm you see stuff

## BaseOS.repo

* cd /etc/yum.repos.d
* Name it 100% consistently
* vi BaseOS.repo
  + [BaseOS]
  + name=BaseOS
  + baseurl=file:///repo/BaseOS
    - repo is the dir you used as a mount point
  + gpgcheck=0
    - 0 means don’t check it
  + validate
    - dnf repolist
    - should show the name
* vi AppStream.repo
* Name it 100% consistently
* vi AppStream.repo
  + [AppStream]
  + name=AppStream
  + baseurl=file:///repo/AppStream
    - repo is the dir you used as a mount point
  + gpgcheck=0
    - 0 means don’t check it
  + validate
    - dnf repolist
    - should show the name
    - with both repos installed
      * dnf install vim
        + vim and vi are separate apps

which vim

which vi

# Intro - Sander Video 0

* Nothing

# Understanding RHEL - Sander Video 1

* ubi (universale base image) is a subset of RHEL for containers
* developers.redhat.com
  + can use 16 instances for free

# Installing RHEL - Sander Video 2

## Install Pair of Servers for Lab Work - First Sander Video Course

* esxi
  + 20 GB
  + network connection
  + extra 10 GB drive
  + 2 cpu
  + 2 GB ram
  + Connect dvd to datastore iso file
  + do a legacy bios install
* use server with gui install
* In the first video class he used automatic

## Installing using custom partitioning

* use custom partitions
* select custom, done
  + then it takes you to the configuration screen
  + he uses all partitions, no lvms
  + add /
    - mount /
    - capacity = 12G
    - XFS
  + add swap
    - mount swap
    - capacity = 1G
  + add /boot
    - 500M
    - XFS
  + add /home
    - 3GB
    - XFS
  + This leaves you a few extra GB to use out of the 20 GB of the main disk
  + Done
* root and student users with passwords
  + should add whatever admin user you create during install to the wheel group

## Lab - Installing RHEL - Sander Video 2

* Install a rhel server
  + server with gui
  + 10 GiB root partition
  + 1 GiB swap partition
  + Make sure 4 GiB of disk space remains unused
  + set the root password to password
  + create a user student with password of password
  + configure the nic for dhcp

## Lab Solution- Installing RHEL - Sander Video 2

* Install a rhel server
  + server with gui
  + select custom, done
  + 10 GiB root partition
  + add /
    - mount /
    - capacity = 12G
    - XFS
  + add swap
    - mount swap
    - capacity = 1G
  + He was going to add a boot partition, but then said it wasn't in the requirements so he stopped. Does the installer give you a default one then?
  + 1 GiB swap partition
  + Make sure 4 GiB of disk space remains unused
  + set the root password to password
  + create a user student with password of password
  + configure the nic for dhcp
    - ensure it says connected
    - ensure it selected an ip for you
  + didn't see option for creating an admin user
  + he reboots after install
    - student
    - password

# Using the CLI - Sander Video 3

## Bash Shell

* ls --help

## Gnome

* search bar
  + power savings
  + screen blank, never

## Virtual Terminal

* virtual terminals can be used to start additional terminal sessions
  + use ctrl[-alt]-Fn
  + GUI
    - ctrl-alt-f2
* See who is active on which terminals
  + who
  + w
    - gives a little more info than who
* Use chvt to change between virtual terminals
  + I don't think I got that to work
  + must be root
    - sudo chvt 3
* tty is the name of the terminal
  + it's a device name

## Lab - CLI - Sander Video 3

* Log into the gui with your default user account
* open a terminal
  + who
  + who --help

## Lab Solution - CLI - Sander Video 3

* If you are already logged in, log out
* student
* who
* who --help

# Exploring Essential Tools - Sander Video 4

## man

* numbers / sections
  + 3 sections matter the most
    - 1 = executable programs or shell commands
    - 5 = file formats and conventions, config files
    - 8 = sysadmin commands
* man rpm
  + /query
  + n for next
  + opens to section 8
* man man describes the sections
  + shows ways to search man
* For info about a section
  + man <section number> intro
    - for an intro to the topic
* man who
  + opens to section 1
* man passwd
  + this one has multiple sections
  + to ensure you see all sections
    - man -a passwd
    - when you quit it tells you there are more, do you want to see this other section
* if you know what section number you want, pass it in
  + man <section number> <name>
  + man 5 passwd
* all man pages are indexed in mandb
  + search mandb via keybord
    - man -k or apropos
    - mandb is built thru a scheduled task
    - sudo mandb
* on a new box, mandb hasn't been started yet
  + sudo mandb
* man man
  + /-k
* man apropos
* man mandb
* mandb
  + no perm if not root
* sudo mandb
  + fine
  + updates the index cache
* man -k user
  + shows results and the section they come from
* man -k user | grep 8
  + shows only descriptions from section 8

## vi

* released in 1976
* dd = delete
* yy = copy
* p = paste
* v = visual mode
  + then select text
  + either delete with dd or copy with yy
* u = undo
* ctrl-r = redo
* gg = go to the top
* G = go to the bottom
* ?blah
  + search backwards
* ^ start of line
* $ end of line
* w move to the next word
* search and replace
  + :%s/old/new/g
* :set number = show line numbers
* visual mode example
  + v
  + highlight 3 lines
  + you will see VISUAL in the bottom left
  + dd to delete the 3 lines
* he does :wq!
  + he says ! means don't ask me any other questions, just do it

## Lab Essential Tools - Sander Video 5

* find the man page that shows how to set a password
* use the man page for useradd, create a user anna
* set anna password to password
* use vi
  + create a file with the name users
  + add these names on separate lines
    - alex
    - alexander
    - linda
    - belinda

## Lab Solution Essential Tools - Sander Video 5

* find the man page that shows how to set a password
  + man -k passwd | less
    - that's a lot of info
    - man -k passwd | grep 1
      * nothing good here
    - man -k passwd | grep 8
  + nothing good comes up at all
  + try this
    - man -k authentication
    - now you see passwd
* use the man page for useradd, create a user anna
  + man useradd
    - it's in section 8
    - only root can do stuff from section 8
    - sudo useradd anna
  + synopsis shows how to use it
* set anna password to password
  + sudo passwd anna
    - password
* use vi
  + create a file with the name users
  + add these names on separate lines
    - alex
    - alexander
    - linda
    - belinda

# Understanding the bash shell - Sander Video 5

## Redirection and Piping

* STDIN
* STDERR
  + Write to stderr
    - 2> /dev/null
  + Redirect errors example
  + ls poopomloochichoo
    - error message
  + ls poopomloochichoo 2> /dev/null
    - no error message
  + if you ls a bad file and a good file, the error goes to /dev/null and it lists the good file
    - ls poopomloochichoo goodfile 2> /dev/null
  + input redirection
    - cat < output
      * shows what is in output
    - but cat works without that
      * cat output
* STDOUT

## history

* commands are stored in
  + ~/.bash\_history
* two variables re history
  + HISTSIZE
    - default = 1000
  + HISTFILESIZE
    - default = 1000
* ctrl-r
  + reverse i search
  + reverse search based on entered pattern
* !nn
  + repeats that line number
* !a
  + repeats the last command that starts with a
    - seems dangerous to me
* manage history
  + history -w synchronizes the current history from memory to history file
  + history is synchronized automatically when you log out or exit properly
* history -c clears all the history from memory, but if you want to clean the file too, also do history -w to remove everything
* it's more convenient to just remove a line
  + history -d nn
  + it readjust the line numbers
* if you have two different tabs open, they have different history if you type history
  + history is kept local to your terminal session!!!
* if you close both terminals and open a new one, when you come back the history from both terminal sessions will be merged into the one history file
  + in this case it might be saved out of order

## keyboard shortcuts

* alt = option key on mac
* ctrl-a move to start of line
* ctrl-e move to end of line
* ctrl-l clear screen
* ctrl-u remove current line
* alt-b more one word backward
* alt-f move one work forward

## Shell expansion

* globbing = wildcards
  + ls \*
  + ls [a-e]\*
    - files that start with a, b, c, d, e
* brace expansion
  + touch file{1..9}
  + can do it with comma separate lists!!!
    - useradd {lisa,linda,anna}
* command substitution
  + ls -l $(which ls)
* variable substitution
  + echo $PATH
* ls vs. ls \*
  + ls \* shows directory names
* ls -d \*
  + shows the same as ls \*
* ls -d /etc/\*
* ls -d /etc/a\*
* ls -d /etc/[a-e]\*
* Not sure what the -d is doing, it looks like more than directories to me???
* which passwd
  + ls -l $(which passwd)
  + this way you don't have to type /usr/bin/passwd, it autofills it in for you
* echo $PATH

## Escaping Special Characters

* Three ways to escape
  + double quotes
    - Doubles suppress globbing and shell expansion, but allow command and variable substitution
  + single quotes
    - Singles take away the special meaning of any characters
  + backslash
    - protects the following character from expansion
* ls -l myfile output
  + the ls command with two args
* echo $PATH
  + here $PATH is interpreted
* echo '$PATH'
  + here $PATH is not interpreted
* double quotes do not take away the meaning of the $echo
* \$PATH
  + escape the $

## Applying Variables

* show environment variables
  + env
* define a variable
  + [export] key=value
* to make a variable persist it needs to be in a bash start up file
* Create a local variable
  + color=purple
  + echo $color
* if you start a subshell by typing bash, and do echo $color it wouldn't display anything because the variable was local to the other shell
* if you want it available for a subshell you would need to do
  + export color=magenta
  + this doesn't work for new shells
  + if you want this, set it in the bash start up file

## alias

* to see what is alredy aliased
  + alias
* create an alias
  + alias key=value
  + alias del='rm -rf /tmp'
* remove an alias
  + unalias blah
* some aliases are stored in the bash start up files

## Tuning the bash environment

* I love how he does this. Two pairs perfectly mapped with the place they are applied!!!
* /etc/profile
  + LOGIN: generic login bash start up file with system settings processed at login
  + shell code exists to include/source /etc/bashrc if it exists
* /etc/bashrc
  + SUBSHELL: generic file processed upon opening a subshell
* ~/.bash\_profile
  + LOGIN: user specific version of /etc/profile
* ~/.bashrc
  + SUBSHELL: user specific version of /etc/bashrc
* /etc/profile.d/
  + ls
  + drop in files for bash environment

## lab - bash shell - Sander Video 5

* he was not clear at all, but he means to do this just for user student
* set the variable color to the value red
  + ensure it's available every time your account logs in
* create an alias that runs the command ls -tlr when you type dir
* ensure the bash history file can grow to the maximum size of 2500
  + HIST\_FILE\_SIZE is a generic system variable, but even generic system variables can be overwritten at the user level!!!

## lab solution- bash shell - Sander Video 5

* set the variable color to the value red
  + ensure it's available every time your account logs in
  + as student
  + vi ~/.bash\_profile
  + export color=red
    - you need export, even if it's in the file!!!
  + alias dir='ls -tlr'
  + export HIST\_FILE\_SIZE=2500
  + To activate or make these changes effective
    - source .bash\_profile
  + source
    - when you use source you include the contents of a shell script
* create an alias that runs the command ls -tlr when you type dir
* ensure the bash history file can grow to the maximum size of 2500
* validate
  + dir
    - see the stuff
  + echo $HIST\_FILE\_SIZE
    - 2500
  + echo $color
    - red

# File Management Tools - Sander Video 6

## Filesystem Hierarchy Standard (FHS)

* man hier
* man filehierarchy

## Finding Files

* which
  + looks for binaries in $PATH
* locate
  + uses a database built by the command updatedb to find files
* find
* Demo
  + find / -name "hosts"
    - / is good for starting a find operation, start at the root directory
    - can take a long time
    - he says it's always good to put your find argument between double quotes
      * he doesn't say why
    - Get rid of the errors
      * find / -name "hosts" 2>/dev/null
    - if you wanted files containing hosts, not just exact match, do wildcards
      * find / -name "\*hosts\*" 2>/dev/null
  + find / -type f -size +100M
    - -type f is for ordinary files, not directories ro special files
  + find /etc -exec grep -l student {}\; -exec cp {} find/contents/\; 2>/dev/null
    - start in /etc
    - mkdir -p find/contents
    - -exec lets you run a command on the result of the find command
      * the command is grep -l student
      * the {} refer to the result of the previous part of the find command
      * \; closes the exec statement
      * in regular bash ; is the command separator
      * you can use multiple -exec statements
        + each one needs it's own \;
      * after finding and grepping the files copy them to find/contents/
  + find /etc -name "\*" -type f | xargs grep "127.0.0.1"
    - xargs takes the output of the find command and passes it to be used by the grep command
    - shows all files containing the string 127.0.0.1
    - "\*" means all files
      * may be '\*', couldn't see

## mounting filesystems

* to access a device, it must be connected to a directory
* Ex:
  + mount /dev/<device name> /<directory>
  + mount /dev/sdb1 /mnt
* findmnt
  + shows all currently mounted devices and their place in the FS
  + left column is directories
  + then what's mounted on them
* lsblk
  + list block devices

## Links

* Links ar pointers to files in a different location
* like a windows shortcut
* hard links
  + ln
* soft links
  + ln -s
* inode => blocks
  + name the inode
* the symlink points to a hard link
* the hard link points to the inode directly
* ls -li /etc/hosts
  + first column is inode number
  + ln /etc/hosts myhosts
  + ls -l
    - they have the same inode
  + ectho hello >> myhosts
  + both /etc/hosts and myhost have become bigger because you added hello to them
* ln -s myhosts symhosts
* ls -il myhosts symhosts
* it's dangerous
  + symlinks point to a relatvie file name that is the contents of the symlinks
  + if you move it you will be in trouble
  + symlinks have their own inodes
  + they have no perm because it's set in the target file
  + if you delete the original, the symlink isn't good

## Archiving Files / tar

* tar up some stuff
  + tar -cvf my\_archive.tar /home /etc
  + c for create
  + v for verbose
  + f filename
* show contents of an archive
  + tar -tvf
* extract to the current directory
  + use -C to switch the output path
  + tar -xvf my\_archive
* To add compression use any of these
  + -z gzip
  + -j bzip2
  + -J xz
* Demo
  + tar something
    - tar -cvf /tmp/archive.tar /home /etc
  + now do it with compression and gzip
    - tar -czvf /tmp/archive.tgz /home /etc
    - gzip
    - extensions have no meaning but change it here to help identify
  + now do it with compression and bzip2
    - tar -cjvf /tmp/archive.bz2 /home /etc
  + now do it with compression and xz
    - tar -cJvf /tmp/archive.xz /home /etc
  + ls -hal
  + xz is twice as good as gzip, but much slower
  + tar -tvf /tmp/archive.tar
    - shows you what's in the tarball
  + now extract the file
    - tar -xvf /tmp/archive.tar

## Working with Compressed Files

* gzip (-z) is the most commong
  + fastest
* bzip2 (-j)
* zip
* xz (-J)
  + best compression

## lab file management tools - sander video 6

* Use tar to create a compressed archive of all files in /etc and /opt
  + write the archive to your home dir
* Create a sym link to the archive in /tmp
* Remove your archive from your home dir, what happens to the sym link?

## lab solution file management tools - sander video 6

* Use tar to create a compressed archive of all files in /etc and /opt
  + write the archive to your home dir
  + tar czvf lab6.tgz /etc/ /opt
  + validate
    - tar tvf lab6.tgz
* Create a sym link to the archive in /tmp
  + ln -s lab6.tgz /tmp/lab6.link
* Remove your archive from your home dir, what happens to the sym link?
  + rm lab.tgz
  + ls -hal /tmp/lab6.link
    - link is invalid

# Managing Text Files - Sander video 7

* head / tail
  + first 10 / last 10
* tail -3 /etc/passwd | head -n 1
* cat
  + cat -A shows non-printable characters
  + cat -b shows lines numbers
  + cat - s suppresses repeated empty lines
* tac
  + cat but in reverse
* cut
  + cut -d : -f 1 /etc/passwd
    - -d is delimiter
    - here it's the colon
      * -f field
      * n field number 1
  + pipe it to sort
* tr
  + echo hello | tr [:lower:] [:upper:]
  + echo hello | tr a-z A-Z
    - output for both of those is HELLO
  + echo hello | tr a-l n-z
    - outputs tqxxa

## grep

* generic regular expression parser
* grep linda \*
* grep -i linda
* grep -A5 linda /etc/passwd
* grep -R root /etc
* grep -R root \*
  + recursive grep
* ps aux | grep ssh | grep -v grep
  + removes the bogus grep line
* grep student \* 2>/dev/null
  + get rid of errors
* grep -A 5 -B 5 Allow /etc/ssh/sshd\_config
  + shows 5 after
  + showb 5 before
* regular grep shows the file names and the match
* see just files names
  + grep -Rl root \*

## regex

* it's different from globbing
  + put your regex in single quotes so they don't get confused with globbing
* man 7 regex
  + too much info
* there are several sets of regular expressions
  + extended regular expressions don't always work
    - grep -E blah
* perl has its own regex
* Demo
  + To work thru these, have this group each on it's own in a file called users
    - leanna
    - anna
    - annabelle
    - anna bella
    - bananna
    - space
    - bit
    - bet
    - bot
    - boat
    - boot
    - boooot
  + grep 'l' users
  + grep '^l' users
  + grep 'anna$' users
  + grep 'anna\b' users
    - confusing. matches
      * leanna
      * anna
      * anna bella
      * bananna
  + grep 'b.\*t' users
    - bit
    - bet
    - bot
    - boat
    - boot
    - booot
    - the dot means one single character
    - the start means 0 or more times
  + grep Match beginning of line
    - grep '^l' myfile
  + Match end of line
    - grep 'anna$' myfile
  + Match lines starting with lea at the end of a word, so it will fine lea, but not leanne
    - grep '^lea\b' myfile
  + Match zero or more times
    - \*
    - grep 'n.\*x' myfile
  + Match one or more times (an extended regex)
    - +
    - grep -E 'bi+t' myfile
  + Match zero or one time (an extended regex)
    - ?
    - grep -E 'bi?t' myfile
  + Match but string must be a word
    - grep '\banna\b' myfile
  + Match one character
    - .
    - grep '^.$' myfile
  + Match either
    - grep -E '( svm | vmx )' /proc/cpuinfo
  + xxx
    - grep 'bo\{4\}t' users
      * 4 is the number of times to match
      * each curl needs to be escaped \{ and \}
      * it must match o 4 times, boooot

## awk

* it focuses on working with individual lines, it's line based
* performs actions based on selectors
* a lot of awk has lost its usefulness
* awk -F : '{ print $4 }' /etc/passwd
  + -F is the field separator which is a colon
  + this prints out all the UIDs
* awk -F : '/linda/ { print $4 }' /etc/passwd
  + it prints the 4th column only for lines that match on linda
* awk -F : '{ print $NF }' /etc/passwd
  + prints the last field, which is the shell

## sed

* means stream editor
* use the file from the grep exercise
* it focuses on searching for text and then performing an operation on matching text
* sed -n 5p /etc/passwd
  + prints line 5
* sed -i s/boot/beet/g ~/users
  + g is global, change all of them on the line
* sed -i -e '10d' ~/users
  + -i means interactive
    - it means the change is written immediately to the file
    - -e means edit
    - it deleted the 10th line
    - d must mean delete

## lab solution - Sander video 12 - text files

* use head and tail to display the 5th line of /etc/passwd
* use sed to disply the 5th line of /etc/passwd
* use awk with a pipe to filter the last column out of the results of ps aux
* use grep to show the names of all files in /etc that have lies that contain the text 'root' as a word
* use grep to show all lines from all files in /etc/ that contain exactly 3 chars
* use grep to find all files that contain the string 'alex' but make sure that alexander is not included

## lab solution Sander video 7 - text files

* use head and tail to display the 5th line of /etc/passwd
  + head -5 /etc/passwd | tail -1
* use sed to disply the 5th line of /etc/passwd
  + sed -n 5p /etc/passwd
* use awk with a pipe to filter the last column out of the results of ps aux
  + ps aux | awk '{ print $NF}'
* use grep to show the names of all files in /etc that have lies that contain the text 'root' as a word
  + grep 'root\b' \*
    - this has a lot of error messages so remove them
    - grep 'root\b' \* 2>/dev/null
* use grep to show all lines from all files in /etc/ that contain exactly 3 chars
  + grep '^...$'
* use grep to find all files that contain the string 'alex' but make sure that alexander is not included
  + grep '\banna\b' users
  + \b for the beginning and \b for the end
  + needs to both start as a word and end as a word

# Root Privileges - Sander Video 8

# su and sudo

## General

* the root user operates in kernel space, that's why it has unlimited access to all parts of the system
* When installing RH9, the root user can be activated or not
  + if the root user is not activated, the administrative user is used
    - created during install
* To acitvate the root user, the only thing you need to do is set the password
* During the install you also need to decide if the root user can ssh into the box

## switching user with su

* su -
  + switches the user account with the complete environment
* Doing su - is considered bad practice !!!
  + use sudo -i instead
    - you don't need to know the root password
    - it's for authorized users only
* sudo -i opens a root shell
* With su it's all or nothing, you are in a root shell or you are not
  + with sudo, you could have just partial perm

## Performing admin tasks with sudo

* sudo works with /etc/sudoers
  + an alaternative is to use drop in files in /etc/sudoers.d
  + because /etc/sudoers is installed from mpackages, there is a chance it could be overwritted, but the drop in files are never overwritten
* edit with visudo
* prompts for current user password
* to open a root shell
  + sudo -i
* the administrative user you create during install will always have sudo privilegles automatically
* users who are a member of whell get full sudo access
  + here is the needed line
  + %wheel ALL=(ALL) ALL
* Here is how you add a user to wheel
  + usermod -aG wheel blah
    - he insists you need both -a and -G or it won't work
* There is an example line that should never be enabled, because it would give root passwordless access
  + %whell ALL=(ALL) NOPASSWD:ALL
* If you don't like entering your password every 5 minutes, add this to increase your authentication token expiration
  + where do you add this?
  + Defaults timestamp\_type=global,timestamp\_timeout=60
    - now you are good for 1 hour
* Provide access to specific tasks
  + lisa ALL=/sbin/useradd, /usr/bin/passwd
* Use args to make the commands more specific
  + %users ALL=/bin/mount /dev/sdb, /bin/umount /dev/sdb
    - all users can mount/unmount /dev/sdb
  + linda ALL=/usr/bin/passwd, space]/usr/bin/passwd root
    - linda can change passwords for all but root
    - Always put the generic rule first, and then the exception!!!

## Using ssh to log in remotely

* root access is denied by default in RH9

## lab 8 using root privilege - sander video

* create user linda
* create a sudo config that allows linda to peform
  + allow using useradd, usermod, and userdel
  + allow changing passwords, but not the root password
* ensure she needs to enter a password for sudo only 1x every 60 minutes

## lab 8 using root privilege - sander video

* create user linda
* create a sudo config that allows linda to peform
  + allow using useradd, usermod, and userdel
  + allow changing passwords, but not the root password
  + vi /etc/sudoers.d/linda
  + linda ALL=/usr/bin/passwd, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel, ! /usr/bin/passwd root
* ensure she needs to enter a password for sudo only 1x every 60 minutes
  + sudo visudo
    - shows some good examples in the default file!!!
    - At the top of the file under the line that says defaults add this
    - Defaults timestamp\_type=global, timestamp\_timeout=60

# Users and Groups - Sander video 9

* processes use system accounts
* people use regular user accounts
* /etc/passwd properties
  + name
  + password
    - may be disabled
    - system users should be disabled and not have passwords
  + uid
  + gid
  + gecos
  + home directory
  + shell
    - normal users /bin/bash
    - system users
      * /bin/false
      * disallow login
      * /sbin/nologin
  + people users normally start at 1,000
  + halt user shell is /sbin/halt
  + sync user shell is /bin/sync

## Creating and Managing Users

* useradd
  + useradd --help | less
  + good options
  + useradd lori -u 2000 -s /sbin/nologin
    - -u uid
    - -s shell
* usermod
* userdel
  + userdel -rf lori
  + remove the user and force them off
* passwd

## Defining Default User Settings

* Specify default settings
* useradd -D
  + He said don't do that
* /etc/default/useradd
  + only applies to useradd
  + don't do stuff here
* Do modify /etc/login.defs
  + PASS MAX DAYS 99999
    - 270 years
    - change it to something, 99
  + HOME\_MODE 0700
    - he likes this
    - only the user has access to the home dir
* Files in /etc/skel are created to the user home dir upon creation
  + touch /etc/skel/hello
  + useradd bill
  + passwd bill
    - blah
  + ls -hal /home/bill
    - hello is there

## Limiting User Access

* Lock/Unlock accounts
  + usermod --help
  + usermod -L anna
  + usermod -U anna
  + See if an account is locked
    - passwd -S bill
      * Output: Password locked
    - grep <user> /etc/shadow
      * if the long password string starts with a !, then it's locked
  + You could even edit /etc/passwd and put the exclamation mark there to lock it
  + Could also lock with passwd
* Set account to expire
  + expire user account bill on 01/01/23
    - usermod -e 2023-01-01 bill
* No login shell
  + usermod -s /sbin/nologin blah
  + use this for all system users
* /etc/passwd
  + ! before long string means account is locked
  + !! after name: means password was never set

## Managing Group Membership

* Primary group membership is managaed via /etc/passwd
* Secondary group membership is managed via /etc/groups
* You can temporarily set primary group membership using newgrp
* If you change user properties, you must log out/login for them to take effect
* See what groups a user is a member of
  + id blah
* add groups
  + groupadd
* groupdel
* groupmod
  + groupmod --help
  + has a limited number of options
* List all users that are a member of a group
  + lid -g <group name>
* group membership can be administered in two different files
  + /etc/group
  + /etc/passwd
* Demo
  + id anna
  + groupadd sales
  + to add anna to the sales group
  + groupmod -U anna sales
  + id anna
  + see all members of sales
    - lid -g sales
  + Try lid -g wheel

## settings password properties

* passwords are stored in /etc/shadow
* the encrypted string shows 3 pieces of info
  + hashing algo
  + random salt
  + encrypted hash of the pw
* Basic password requirements are set in /etc/login.defs
* Use PAM for advanced password properties
* change password!!!
  + chage bob
    - do it this interactive way
    - 3 minimum
    - 7 max
    - last password change just enter
      * it's set to the day the user was created
    - expiration warning 1
    - password inactive
      * it's not inactive so -1
    - expiration date 2024-12-12
* chage --help
  + -l
  + chage -l Bob
    - shows all params
* passwd can do the same, but stick with chage

## Lab for Users and Groups - Sander Video 9

* make sure new users require a password with a maximum validity of 90 days
* ensure that while creating users, an empty file with the name newfile is created in their home directory
* create the users: anna, audrey, linda, and lisa
* set passwords for anna and audrey to password
* disable the passwords for linda and lisa
* create the groups profs and students
  + make anna and audrey members of profs
  + make linda and lisa members of students

## Lab Solution for Users and Groups - Sander Video 9

* make sure new users require a password with a maximum validity of 90 days
  + vi /etc/login.defs
  + edit PASS\_MAX\_DAYS to 90
* ensure that while creating users, an empty file with the name newfile is created in their home directory
  + touch /etc/skel/newfile
* create the users: anna, audrey, linda, and lisa
  + useradd anna
  + useradd audrey
  + useradd linda
  + useradd lisa
* set passwords for anna and audrey to password
  + for automation consider
    - echo password | passwd --stdin anna
* disable the passwords for linda and lisa
  + you can do this two ways
  + passwd --help
  + passwd -l linda
  + passwd -l lisa
* create the groups profs and students
  + make anna and audrey members of profs
  + make linda and lisa members of students
  + groupadd profs
  + groupadd students
  + groupadd --help
  + groupmod -U anna,audrey profs
  + groupmod -U linda,lisa students
  + validate
    - lid -g profs
    - lid -g students

# Securing files with Permissions - Sander Video 10

## ownership

* ugo
* linux permissions aren't additive
  + if you are the owner, perms are applied and that's all
* chown user[:group] file
* chown lisa newfiles
* chown linda:sales newfiles
* chgrp blahgroup newfiles

## Basic Permission

* read / file
  + read, open the file
* read / directory
  + list files in directory
* write / file
  + modify
* write / directory
  + create or delete files
* execute / file
  + run the file
* execute / directory
  + cd
* read = 4
* write = 2
* execute = 1
* special X
  + when x is applied recursively, it would make directories as well as files executable
    - that's bad
  + in recursive commands, you should use X, upper case X
    - directories will be granted execute perm
    - files will only get execute perm if it's already set on the file
* chmod
  + change mode
  + change permissions
  + use it in absolute or relative mode
    - absolute
      * use numbers
        + chmod 750 myfile
    - relative
      * letters
      * good if you want to apply an individual perm to a file
        + chmod +x myscript
* Demo
  + chmod 750 newfiles/
  + chmod g+w newfiles/
  + chmod o-w newfiles/
* root lives in a world where permissions don't apply
* if root makes a file in linda's home dir even though the perm is rw-r--r-- and then linda wants to delete the file, will she be able to do it since root created it?
  + she can delete it
    - deleting a file requires write perm on the directory
    - it's Linda's home dir

## Shared Group Directories

* SGID ensures that all files created in a shared goup directory are owned by the group owner of the directory
* the sticky bit ensures that only the user how is owner of the file, or the directory that contains the file, is allowed to delete the file
* chmod g+s mydir applies SGID
* chmod +t assigns the sticky bit to a directory
* in absolute mode, a four digit number is used, the first of which is for special perm
  + chmod 3770 mydir assigns SGID and sticky bit, as well as rwx for user and group
* Demo
  + id ana
    - member of profs
  + id audrey
    - member of profs
  + su - anna
    - cd /data/profs
      * touch anna{1..5}
    - ls -hal
    - you don't want them owned by group anna, you want them owned by group profs
    - chmod 3770 /data/profs
    - ls -hal
    - rwxrws--T
      * there is a s now where the group x was
      * lower s means set group id + group execute
      * T for the sticky bit

## Applying Default Perm - umask

* The umask subtracts the umask from the default perm
  + default config file = /etc/bashrc
  + override for individual user = ~/.bashrc
* Default perm for a file is 666
* Default perm for a directory is 777
  + the difference between 6 and 7 is execute perm
* Example
  + imagine umask is 027
    - 640 for files
      * 666 - 027
      * the third number would be a negative, but since you can't do that, do 0
    - 750 for dirs
* Demo
  + just type umask and it shows you the curren setting
    - it's 4 digits
    - ignore the leftmost one, it deals with special perms
      * you will never change it
  + change it
  + umask 027
  + umask
    - it's changed, but it's non-persistent
    - add it to ~/.bashrc
      * somewhere near the bottom
    - to source your change quickly without logging out and back in
      * . .bashrc
      * the dot command is the source command!!!

## Lab Permission - Sander Video 10

* Create shared group directories
  + /data/profs
  + /data/students
  + Member of the groups have full read and write access to their directories, others have no perm
* Modify default perm settings to that normal users have a umaks that allows the user and group to write, create, and execute files and directories while denying all others access

## Lab Solution - Sander Video 10

* Create shared group directories
  + Go thru this slowly, I don't think he did SGID or Sticky, so what does this really do
  + /data/profs
  + /data/students
  + mkdir -p /data/profs
  + mkdir -p /data/students
  + Member of the groups have full read and write access to their directories, others have no perm
  + chgrp profs profs
  + chgrp students students
  + ls -hal
  + chmod 770 students
  + chmod 770 profs
  + ls -hal
  + don't care about user owner, since it's root
  + if exam question doesn't ask about user, just leave it
* Modify default perm settings to that normal users have a umaks that allows the user and group to write, create, and execute files and directories while denying all others access
  + vi /etc/bashrc
    - there is an if statement about if uid is xxx set the umask to blah, else blah
      * it menas ordinary users have a umask of 002
      * root has 022
    - edit the if line for the user to 007

# Networking - Sander Video 11

## ipv4

* ipv4 and ipv6 can coexist on the same nic

## ipv6

* 128 bit numbers
* expressed as 8 colon separated groups of 4 hex numbers
* leading zeros are omitted
* long strings of zeros can be replaced by ::
* using colons for separators is problematic with ip / port
  + use square brackets around ipv6 ip
    - [2001:bad::210]:80
* standard subnet is 64 bits
* network providers usually provide a /48 which leaves 16 bits to the customer for subnets
* ![A white and black address list

  Description automatically generated with medium confidence

## Nic Naming

* ip link show for devices
  + note the naming convention used
* ip addr show for config
* every system has a lo device (LO) for internal networking
  + 127.0.0.1
  + ::1
* BIOS Device Names vs. Classical Names
  + classical naming is using stuff like eth0, eht1
  + These names don't reveal any info about physical device location
  + bios naming is based on hardware properties
    - bios naming has 3 conventions
      * em[1-N] for embedded nics
      * eno[nn] for embedded nics
      * p<slot><port> for nics on pci bus
  + if the driver doesn't reveal network device properties, classic naming is used
* Demo
  + ip link show
    - lo
    - ens160
      * can see mac address
      * altname enp3s0 is the bios name
  + ip addr
    - ipv4 address and mask

## Defining Host Names and Resolution

* hostnamectl hostname server1.example.com
* hostname is written to /etc/hostname
* to resolve hostnames, use /etc/hosts
  + 10.0.0.1 server2.example.com server2
  + good for local use only
* /etc/resolv.conf has dns client config
* There are 2 systems for host name resolution
  + dns
  + /etc/hosts
  + the order of host name resolution is determined thru /etc/nsswitch.conf
* Demo
  + hostname => localhost.localdomain
  + hostnamectl hostname server1.example.com
  + hostname => server1.example.com
  + exit and reopen shell to see name reflected there
  + vi /etc/hosts
    - 192.168.29.211 server1.example.com server1

## Analyzing Network Configuration

* None of the ip commands are persistent, need to use network manager
* ip -s link
  + shows link properties
* ip route
  + manage route properties
  + add default gw
    - ip route add default via 10.0.0.1
* ip addr
  + manage address properties
* Demo
  + ip a
    - ip a a dev ens160 10.0.0.10/24
    - ip a a = ip address add
  + ifconfig
    - don't use it
    - doesn't show secondary ip
    - it's obsolete
    - mac os still uses ifconfig!!!
  + ip -s link
    - stats
    - this info doesn't matter any more
      * collissions
      * not relevent
      * better to analyze switch
  + ip route show
    - default via = dgw

## Network Manager

* It's the systemd service that manages networks
* configuration is at /etc/NetworkManager/system-connections
  + this is a new location from past RHEL versions
  + legacy files are in /etc/sysconfig/network-scripts
    - supported but deprecated
* nmcli
* nmtui
  + tui = text user interface
* gnome tools
* devices
  + network interfaces
* connections
  + collections of config settings for a device stored in a config file at /etc/NetworkManager/system-connections
  + only 1 connection can be active for a device
* Permissions to modify settings in NM are applied thru dbus
* Non-privileged users that are logged in on the console can change network settings
  + Non-privileged users that are logged in via ssh can't
* See an overview of current permissions
  + nmcli general permissions
* Demo
  + systemctl status NetworkManager
  + nmcli has great tab completion
  + nmcli general status
  + nmcli general permissions
    - if you run this as user linda and root, you see different perm
  + ssh student@localhost
    - weird to ssh to localhost
    - nmcli general permission
      * totally different perm, lots of no and auth
      * thru an ssh connection there is a very limited features that can be changed

## nmcli for persistent connections

* Don't use it on the exam!!!
* Use nmtui on the exam!!!
* show current connections
  + nmcli con show
* show current devices
  + nmcli dev status
* Add a new connection
  + nmcli con add con-name mynewconnection ifname ens33 ipv4.address 10.0.0.10/24 ipv4.gateway 10.0.0.1 ipv4.method manual type ethernet
  + still need to activate it
    - nmcli con up mynewconnection
* Demo
  + nmcli connection show
  + nmcli dev status
  + on the command below he got everything from tab completion except for ipv4.method
  + nmcli connection add con-name mycon ifname ens160 ipv4.addresses 192.168.29.33/24 ipv4.gateway 192.168.29.2 ipv4.method manual ethernet
    - activate it
      * nmcli connection up mycon
  + nmcli con show
  + ip a
    - it's there now
* show all connection settings
  + nmcli con show mynewconnection
* modify connection settings, good tab completion
  + nmcli con mod
* reload the modified connection
  + nmcli con reload
* all connections not using dhcp need to use ipv4.method manual

## nmtui

* Use this on the exam instead of wasting time remembering nmcli options
* 3 options
  + edit
    - ip
    - gateway
    - dns server
    - check automatically connect
      * double check before rebooting
  + activate
    - it's a toggle
  + set host name

## Troubleshoot Networking

* ping
  + ping -c 4 blah
* ip route
* tracepath example.com
  + shows the whole path
* ss
  + socket statistics
    - replaces netstat
    - ss -tu
      * t = tcp
      * u = udp
      * established connections
    - ss -tuna
    - ss -tunap
      * tcp
      * udp
      * ports
* Demo
* ping fails
  + check ip a
    - make sure your ip address is good
  + ip route
    - default route is missing
    - fix it with nmtui
      * edit
      * looks ok
      * deactivate
      * reactivate
      * ip route looks good now
  + ss -tunap
    - under local address column
      * 127.0.0.1:631
        + you probably want port 631 bound to all addresses 0.0.0.0 or a specific ip, not the loopback

## Lab - Networking - Sander video 11

* Set the hostname for your server to rhcsaserver.example.com
* set the ip to a fixed ip that's good on your network
* set a second ip 10.0.0.10/24 on the same nic
* enable hostname resolution for your local server hostname
* reboot and verify it works

## Lab Solution - Networking - Sander video 11

* Set the hostname for your server to rhcsaserver.example.com
  + nmtui
  + set hostname
  + rhcsaserver.example.com
  + ok
* set the ip to a fixed ip that's good on your network
  + nmtui
  + edit
  + set ipv4 configu to manual, not automatic
  + show, then edit
    - address = 192.168.29.133/24
    - gateway = 192.168.29.2
      * his gw is also a dns and dhcp server
    - dns server = 192.168.29.2
* set a second ip 10.0.0.10/24 on the same nic
  + still in nmtui, add
  + 10.0.0.10/24
* now you are done with nmtui so activate
  + confirm it's not deactivate, activate all the time
* Check your work
  + ip a
    - are both ips good
* enable hostname resolution for your local server hostname
  + vi /etc/hosts
    - d$ in vi deletes to the end of the line!!!
    - 192.168.29.133 rhcsaserver.example.com rhcsaserver
* reboot and verify it works
  + ip a
  + both ips are there

# Managing Software - Sander video 12

* RPMs
  + include a compressed archive and metadata
  + installed packages are registered in the rpm database
  + packages can be extracted to the current directory without installing
    - rpm2cpio blah.rpm | cpio -tv
    - shows the contents
  + rpm2cpio blah.rpm | cpio -idmv
    - extracts the package contents to the current directory
* DNF
  + need repos set up to use it
  + a dnf group is a collection of packages
  + an environment group is used to install a specific usage pattern, and may consist of packages and groups

## rpm commands

* show all packages currently installed
  + queries the rpm database
  + rpm -qa
* show from which package a file was installed
  + rpm -qf <filename>
  + rpm -qf /bin/ls
* list files installed from a package
  + rpm -ql <blah>
  + rpm -ql coreutils
  + rpm -ql coreutils | grep bin
* show scripts executed while installing the package
  + rpm -q --scripts
  + rpm -q --scripts coreutils
* show the changelog for a package
  + rpm -q --changelog
* if you want to query a package file instead of the rpm database, add -p to any of the commands above

## dnf commands

* List installed and available packages
  + dnf list
  + if it's available it will show the repo
  + if it's installed it will show @blah
  + if it shows the word repo it's not installed yet
* List all installed and available packages starting with selinux
  + dnf list 'selinux\*'
* Search in name and summary and description
  + dnf search all seinfo
* Search only in name and summary
  + dnf search seinfo
* Search for the package that provides a file
  + dnf provides \*/Containerfile
  + \*/ is mandatory, without it you won't find what you are looking for!!!
  + not sure why you need \*/, but remember to use it!!!
* Show info about a package
  + dnf info <package>
* install package
  + dnf install
* remove package
  + dnf remove
* update version of package
  + dnf update
* install the new kernel
  + dnf update kernel
  + keeps the old kernel as a back up
* See partial list of groups
  + dnf group list
    - environment groups don't show up here
* See list of hidden groups including environmental groups
  + dnf group list hidden
* See what packages are in a group
  + dnf group info <groupname>
* Install a group
  + out of mandatory, default and optional, only mandatory and default are insalled with
    - dnf group install
* Install an optional group
  + dnf group install --with-optional
* Use double quotes if name includes spaces

## Modules (in RH8, but not in RH9 so not on exam) and Application Streams

* dnf uses modularity, meaning different versions of the same package can be maintained in the same repo
* To deal with modularity, RH 9 offers 2 repos
  + BaseOS has core OS content which shares the OS lifecycle
  + AppStream is for packages that don't have the same lifecycle as RHEL
    - these packages are offered as packages or modules
* In RH 9 no modules are provided by default, they may be added separately later!!!
  + In RH 8 there was the dnf module command, but it's not relevant in RH 9!!!
* In a module, different streams are offered, where each package version has it's own stream
* module profiles provide common installation patterns like client and server
* In the AppStream repo you could go to the next version

# dnf history

* all transaction that dnf performs are logged to /var/log/dnf.rpm.log!!!
* For a summary of all installation and removals
  + dnf history
* Undo a transaction
  + dnf history undo blahnumber
  + you can't always undo if certain other stuff came after

## subscription manager - not on exam

* To use rhel you need to reigster and attach a subscription
* to register
  + subscription-manager register
  + asks for your RH credentials
* to attach a subscription
  + subscription-manager attach --auto
* to unregister
  + subscription-manager unregister
* validate
  + dnf repolist
* After attaching subscriptions to a system, entitlement certs are created
  + they are located in /etc/pki/product
  + RH account for registration
    - /etc/pki/consumer
  + Indication of what subscription is attached
    - /etc/pki/entitlement
* Check current entitlements
  + rct cat-cert /etc/pki/entitlement/xyxxxyyzzz.pem

## Lab - Sander Video - Packages 12

* Ensure your system is using a repo for base and app stream
* find the package that contains the seinfo program file and install it
* download the httpd package from the repos without installing it
  + query to see if there are scripts in it

## Lab Solution - Sander Video - Packages 12

* delete anything in /etc/yum.repos.d before starting
* See naming in my example at the top of the file
  + he makes it confusing here by using all sorts of different names
    - base
    - baseos
    - BaseOS
* /repo should still have the iso mounted
  + it has the AppStream and BaseOS directories
  + just crate the files
* cd /etc/yum.repos.d
* vi base.repo
  + any name.repo
  + [baseos]
  + name=BaseOS
  + baseurl=file:///repo/BaseOS
  + gpgcheck=0
    - don't do it
  + He didn't add the enable line
* validate
  + dnf repolist
* vi appstream.repo
  + any name.repo
  + [appstream]
  + name=AppStream
  + baseurl=file:///repo/AppStream
  + gpgcheck=0
    - don't do it
  + He didn't add the enable line
* dnf search all seinfo
  + setools-console
  + could have also used dnf provides
* dnf install setools-console
* download the httpd package without installing
  + cd /repo/AppStream/Packages
  + ls http\*
  + rpm -qp --scripts httpd-blah

# Monitoring Activity - Sander Video 13

## Jobs

* tasks started from a shell can be managed as jobs
* shell jobs can be started in the foreground or background
* start a job in the background
  + <command> &
* move a job to the background
  + ctrl-Z to stop the job
  + bg to move it to the background
  + get an overview of the running jobs
    - jobs
  + move the last job back to the foreground
    - fg [job number]
* Example
  + sleep 1000
  + ctrl-Z
  + bg
  + run another starting in the background
    - sleep 2000 &
  + jobs
  + move job 1 back to the foregound
    - fg 1

## Process States

* When a new process is started, it's called forking. Then it's scheduled and after being scheduled it will be in a runnable state, R
  + In this state it is waiting in the queue to be scheduled
  + Runnable processes will get a time slice, which allows them to get a running state, in either kernel space or user space
  + Runnable processes can get preempted or rescheduled
    - They will return to a runnable state and wait in the queue for a new time slice
  + A runnable process can be stopped with ctrl-z and will show as TASK\_STOPPED (T)
    - after being stopped it can receive another signal to resume and return to a runnable state
* While running the process may have to wait
  + this is referred to as blocking, blocking isn't an official state in the linux kernel
  + TASK\_INTERRUPTIBLE (S)
    - The process is waiting for hardware request, system resource access or a signal
  + TASK\_UNINTERRUPTIBLE (D)
    - The process is waiting but does not respond to signals
  + TASK\_KILLABLE (K)
    - The process is waiting but may be killed
  + TASK\_REPORT\_IDLE (I)
    - Used for kernel threads, this process will not count for the load average
  + When a process exits, it will briefly enter the EXIT\_ZOMBIE (Z) state. This is where it signals the parent process that it exits and all resources except for the PID are relased
  + In the next stage the process will enter the EXIT\_DEAD (X) state. In this state it will be reaped and all remaining processes are cleand up
  + A process becomes a Zombie when it has completed its task, but the parent process hasn't collected its execution status
  + Zombies are already dead so they can't and don't have to be killed
  + the most import disadvantags is that zombies occupy a PID
  + To get rid of the zombie, the parent process must collect the child execution status
    - send SIGCHLD to the parent to ask the parent to reap the zombie
    - kill the parent process
    - when the parent is killed, the zombie becomes an orphan and will be adopted by the init process

## ps

* it has two dialects
  + BSD
    - no leading -
  + System V
    - -
* commands look similar, but are completely different
  + ps -L and ps L are different
* good way to start
  + ps aux
    - overview of all processes
  + ps -fax
    - shows hierarchical relations between processes
  + ps -fU linda
    - shows all processes owned by linda
  + ps -f --forest -C sshd
    - shows a process tree for a process
  + ps L
    - shows format specifiers
  + ps -eo pid,ppid,user,cmd
* stuff in square brackets [blah] are kernel threads

## Monitoring Memory

* Linux places as many files as possible n cache to guarantee fast access to files
* for that reason, linux memory often shows as saturated
* the kernel moves inactive application memory to swap first
* inactive cache memory will be dropped
* For memory info
  + free -m
    - -m for megabytes
* If you want more detailed memory info
  + /proc/memino
* The use of RAM needs to be balanced between cache and application memory
  + cache and app memory can be
  + used and unused which is active and inactive
* free -m
  + He has total Mem 3,696, basically 4G of memory
  + free means completely unused, nothing happening there
  + shared is memory shared between different processes
  + buff/cache is memory used to speed things up
  + buff/cache + free memory = available
    - buff/cache can be dropped as needed
* reboot
  + e
  + add mem=1G to the linux line
  + ctrl-x
  + starts with only 1G of memory via a kernel parameter
* free -m
  + now it's using swap
  + less /proc/meminfo
    - look at the lines re /active/inactive
    - anonymous memory = app memory
    - file memory = cache
    - active application memory should not use swap
    - inactive anonymous memory can go in swap
    - active file = the cache we are using
    - inactive file = cache that can be discarded if we get short on memory
* write cache
  + it's part of what free calls buff/cache
  + while writing files a write cache (buffers) is used
  + if you write a file and first it's committed to memory, then from memory it's written to disk, it will be handled by the kernel thread pdflush
    - if you see pdflush being active alot, your system is writng alot
  + as a result of the write cache, after committing a file write, it's not immediately secure
  + if you write a file it will be temporarily stored in your write cache, but if at that moment your server crashes, the file is lost
  + If you want to ensure the file is committed to disk immediately, use the sync command

## Monitoring cpu load

* cpu load is checked via uptime
* it's expressed as the average number of runnable process over the last 1, 5 and 15 minutes
  + good trending info
* as a guideline, the number should not exceed the number of cpu cores on a system
  + Ex: if you have a 4 core system, the load should not be more than 4
* use lscpu to check the number of cpu cores
  + On line CPU list
    - 0,1
      * means you have 2 cpus
* Say you have 2 cpus
  + c1
  + c2
* the scheduler makes sure your tasks end up on one of these cpus
* any process that needs attention is in the queue before the scheduler
* say there are 3 tasks
  + t1
  + t2
  + t3
* it's like an air plane and there can be prioirty boarding
* say 1, 2, 3 is the order of entrance
* but you want 3, 1, 2
  + this is where nice comes in
* once a task is scheduled, every process gets a time slice
  + when the time slice is over, the process needs to be removed from the cpu and put back in the queue again
* demo
  + lscpu
    - Online cpu list
      * 0,1
        + means you have 2 cpus
  + uptime
    - load is low
  + create workload
    - dd if=/dev/zero of=/dev/null &
    - repeat this 5x
  + dd is a program you can use to clone devices
  + he clones the /dev/zero device to the /dev/null device
  + he's moving nothing to nowhere
  + run uptime again and the should be high

## top

* it's a dashboard that allows you to monitor system activity
* f to show and select display fields
* M to filter on memory
* W to save new display settings
* htop is an alternative to top, but it's not installed by default
* top
  + shows load average
  + shows tasks
    - he has one zombie
    - you don't see them that often
* quit top and deal with the zombie first
  + ps aux | grep defunct
  + he is going to let it fix itself, which is what usually happens]
* top
  + %cpu line
    - us: starts with the percentage of time the cpu spends in user space
    - sy: system space is kernel processes, like his dd command
    - id: idle is the percentage of time the system spends doing nothing
    - wa: waiting, the cpu is waiting for io
  + shows the same info as free does at the top
  + default sort = cpu usage
    - to sort on memory instead: M
    - use small indent sign on keyboard to go back to sorting on cpu?
  + f will let you show additional fields
    - do f and it shows a list
    - do PPID
    - use arrow keys and select with space bar
    - q to exit
    - if you like what you added, use W to write and save

## Processes

* All tasks are started as processes
* Some processes are starting multiple threads, individual threads can't be managed

## Repos

* a repo is a collection of rpm packages with an index
* To acces third party repos, you need to add a repo file in /etc/yum.repos.d or use dnf config-manager
* dnf config-manager --add-repo="file:///repo/BaseOS"
* dnf config-manager --add-repo="file:///repo/AppStream"
* ls /etc/yum.repos.d
  + note how they are named
  + confirm the params are what I used manually
* The autogenerated files don't come with gpgcheck=0, so be sure to add them
  + this disables gpg checks
* dnf repolist
* you can install packages now
* See my notes at the top of this file, they are simplified
* Exam: They will probably provide a repo server so yo won’t have to set up a repo from scratch
* He goes thru creating a repo here, but his book instructions were way better and are at the top of the file
  + there are options
    - first
      * do dd and make an iso and then put it in /etc/fstab
      * /rhel9.iso /repo iso9660 defaults 0 0
    - second
      * if you don’t have room, use the cd and put it in /etc/fstab
      * /dev/sr0 /repo iso960 defaults 0 0

## Lab Process / Monitoring - Sander Video 13

* Use any tools to find out if your machine performance is in good shape

## Lab Solution Process / Monitoring - Sander Video 13

* top
* load average
  + 5.02
* press 1
  + now you have one line for each cpu!!!
  + there are 2 lines, so you have a 2 core system
    - 5.02 is too high for 2 cores
* only 61 mb of memory free, that's not good

# Managing Processes - Sander Video 14

## Signals

* a signal allows the os to interrupt a process from software and ask it to do something
* interrupts are comparable to signals, but are generated from hardware
* a limited amount of signals can be used and is documented in man 7 signals
* not all signals work in all cases
* you will only use about 2 of them
* kill
  + the kill command is used to send signals to PIDs
  + you can also us k from top
* there are other kill type commands like
  + pkill
  + killall
* Demo - Killing a zombie
  + run zombie from https://github.com/sandervanvugt/rhcsa/zombie
    - dnf install git -y
  + git clone https://github.com/sandervanvugt/rhcsa/zombie
    - no auth required
  + cd rhcsa
  + ./zombie &
  + ps aux | grep defunct
  + kill <child pid>, it will fail
    - kill 11231
  + ps fax
    - get the pid for parent and child
  + kill -SIGCHLD <parent pid>, it will be ignored
  + kill <parentpid>, the zombie will get adoped and init will reap it after a few seconds
  + top
  + he has 4 dd processes running
  + man 7 signal
    - shows them all
      * SIGKILL
        + ordinary kill
      * SIGTERM
        + force kill

it's kill -9

* + you can kill from top
    - k
    - it suggests killing the busiest process
    - he's ok with that
    - signal 9
    - a dd process is gone
  + kill from the cli
    - kill 8992
  + normal kill vs. kill -9
  + killall dd
    - kills by name

## Managing Process Priority

* cgroups - don't need to know for exam, but should understand it
  + cgroups provide a framework to apply resource restrictions to Linux systems
    - they can limit the amount of cpu cycles, available memory
  + if processes are equal from the cgroup perspective, the linux nice and renice commands can be used to manage priority
  + in cgroups, the linux system is divided into 3 slices
    - system: all systemd processes
    - user: all user processes
    - machine: virtual machines and containers
  + each slice has an equal cpu weight
  + that means that if one or more processes within a slice requests a maximum amount of cpu cycles, each slice will get an equal amount of cpu shares
    - so 20 systemd processes together gets as much as one user process that claims full cpu usage
  + In systemd, the CPUWeight can be set on individual systemd units
* nice
  + if no specific cgroups are defined, linux nice and renice can be used to define cpu priority
  + to change priorities of non-realtime processes, the nice and renice commands can be used
  + values range from -20 to 19
  + negative nice value indciates an increated priority, a positive nice value indicates decreased priority
  + users can set their processes to a lower priority, to increase priorities you need root access
  + Ex:
    - nice -n 19 dd if=/dev/zero of=/dev/null
  + priority is always relative to other processes
  + Demo
    - nice -n 10 dd if=/dev/zero of=/dev/null &
    - nice -n -10 dd if=/dev/zero of=/dev/null &
    - top
    - you can see the nice value in column 4
    - press 1
      * 1 line for each cpu now
    - cd /sys/bus/cpu/devices/cpu1
    - cat cpu1/onlinem => 1
    - echo 0 > cpu1/online
    - that takes one of the two cpus offline
    - top
      * now it's looking different
    - killall dd
    - enable the second cpu again
      * echo 0 > cpu1/online

## tuned

* kernel tunables are provided thru the /proc/sys directory
* different files in the /proc/sys directory contain the current setting as its value
* change the current value by echoing a new value into the file
  + cat /proc/sys/vm/swappiness
  + echo 40> /proc/sys/vm/swappiness
  + this is not persistent
  + swappiness defines how eager your kernel is to swap out data if more memory is needed
  + if you want to make the change persistent
    - write them to a file in /etc/sysctl.d
    - cat >> swappiness.conf <<EOF
    - vm.swappiness = 40
    - EOF
* Demo
  + cat /proc/sys/vm/swapiness
  + you can't use vi to edit this file
  + echo 40 > /proc/sys/vm/swappiness
  + sysctl -a
    - sysctl is responsible for managing many kernel tuneables
    - note this is not systemctl
  + sysctl vm.swappiness
  + cd /etc/sysctl.d
  + cat >> swappiness.conf <<EOF
    - this is a here document
    - vm.swappiness = 40
    - EOF
  + cat swappiness.conf
    - => vm.swappiness = 40
  + There are 1,110 tuneable parameters
    - that's a lot
  + use tuned instead
* to make system tuning easier, tuned is provided
* tuned is a systemd service that works with different profiles
* tuned-adm list
  + shows current profiles
* tuned-adm profile virtual-guest
  + sets another profile as default
* each profile contains a file with the name tuned.conf that has a wide range of performance related settings
* how does tuned relate to sysctl
  + the reapply\_sysctl = 1 parameter in /etc/tuned/tunedmain.conf ensures that in case of conflict, the sysctl param wins
    - if you set this to 0 the tuned param wins
  + tuned and sysctl do the same thing
* Demo
* tuned-adm --help
  + not found
  + it asks if you want to install it and it installs the tuned package
  + the autoinstall errored out for him so he did it manually
* dnf install tuned -y
* tuned-adm --help
  + positional arguments
  + tuned-adm list
* tuned-adm profile virtual-guest
* you can do custom tuned profiles
  + they are stored in /etc/tuned
  + each profile should have a tuned.conf with the requested performance settings
  + after creating the directory with tuned.conf, it will automatically be picked up
* Demo
  + dnf install tuned -y
    - he never started it, but it's already running when he does systemctl status tuned
    - read the output
    - it says vendor present enabled
    - you install the package and it will automatically be enabled
  + systemctl enable --now tuned
  + tuned-adm list
    - shows all the profiles
  + echo vm.swappiness = 33 > /etc/sysctl.d/swappiness.conf
  + sysctl -p /etc/sysctl.d/swappiness.conf
    - activates your change
  + sysctl -a | grep swappiness
    - confirm it's your value you set
  + mkdir /etc/tuned/myprofile
  + cat >> /etc/tuned/myprofile/tuned.conf <<EOF
  + [sysctl]
  + vm.swappiness = 66
  + EOF
  + tuned-adm list
    - can see your new profile here now
  + tuned-adm profile myprofile
  + tued-adm profile
    - last line shows the current active profile is myprofile
  + sysctl -a | grep swappiness
  + cat /etc/tuned/tuned-main.conf
    - this file has reapplysysctl - 1
      * this makes sysctl win
      * if you don't like it set it to 0???

## Managing user sessions and processes

* show processes owned by a specific user
  + ps -u username
* remove processes owned by a specific user
  + pkill -u username
* loginctl
  + it's part of systemd
  + one user can have multiple sessions open simultaneously
    - show users
      * loginctl list-users
    - show sessions
      * loginctl list-sessions
  + show a tree of processes owned by a user
    - loginctl user-status <UID>
  + terminate a user or a session
    - loginctl terminate-session
    - loginctl terminate-user
* Demo
  + chvt 5
  + log in
  + vi myfile
  + :shell
  + chvt 2
  + w
    - shows users and tty
  + loginctl list-users
  + loginctl list-sessions
  + loginctl user-status <uid>
  + loginctl terminate-user <user name>
  + loginctl list-users
    - still shows them but
  + loginctl user-status <uid>
    - shows them not logged in or lingering
  + chvt 5
  + if suspicious activity is going on and you need to terminate it fast, use loginctl

## lab managing processes - sander video 14

* create user linda
* open a shell as linda
* as linda run two background processes for sleep 600
  + run one at the highest possible priority
  + run one at the lowest possibly priority
  + use the most efficient way to terminate all sessions for linda

## lab solution managing processes - sander video 14

* linda already exists
* have two windows
* su - linda in one
* root in the other
* nice -n 10 sleep 600 &
  + he says this is the lowest priority
    - double check, I thought it was like 19
* nothing to do for highest priority because I regular user can't bump up their priority
  + sleep &
* another way to terminate all linda stuff is pkill
  + pkill -u linda
    - that seems like it would kill all processes for linda, does that include sessions?
  + validate
    - w

# Sander video 15 - systemd

* systemd is started as the first process after loading the kernel and is the manager of everything
* it's used for starting services
* items started by systemd are called units
* systemctl is the main management tool
* systemctl -t help
  + shows all the available types of units
* sysctl is only for performance parameters
* systemctl tab tab
* systemctl list-units
  + units that are currently loaded
    - what does loaded mean?
* systemctl list-units -t timer
  + look for timer units that are active now
* systemctl list-unit files
  + lists the unit files and their current state
* systemctl cat blah.service
  + shows you all the configuration for the unit even if it's spread over different files!!!

## units

* there are different types of units
* service units
* socket units
* timer units
* path units
  + path units can start service units when activity is detected in the file system
* mount units
  + used to mount file systems

## service units

* used to start daemons
* there are other types of service units
  + Type=oneshot
    - start any command via systemd
* systemctl status sshd
  + The Active: line
    - shows status
  + The Loaded: line
    - shows which config is loaded, and whether the unit is enabled to autostart
    - shows that vendor preset is enabled as well as regular enabled
* start and enable at the same time
  + systemctl enable --now sshd
  + systemctl disable --now sshd
* systemctl reload
  + reloads the unit configuration without restarting the unit
  + does a service go at the end
* Demo
  + systemctl stop sshd
  + systemctl status sshd
    - it's still enabled, but active line is inactive now
  + systemctl disable sshd
    - it's still loaded because systemctl is aware of it
    - now it's set to disabled
    - inactive
  + systemctl enable --now sshd
    - starts and enables

## modifying systemd service configuration

* default system provided systemd unit files are in /usr/lib/systemd/system
  + don't touche these, they come from rpms
* if you want to create custome configs
  + /etc/systemd/system
  + don't modify /usr/lib/systemd/system, but create a custom file in /etc/systemd/system that is used as an overlay file
  + even better
    - use systemctl edit unit.service to edit unit files
  + systemctl show
    - shows available params
  + systemctl reload may be needed
* there are run time auto generated unit files in /run/systemd
  + you won't change these
* Demo
  + systemctl cat httpd.service
  + dnf install httpd -y
  + systemctl cat httpd-service
    - it's clear it's showing you just one file
    - the file it shows has many tuneables
    - for more info
      * systemctl show httpd.service
      * shows a long list of tuneables
  + systemctl edit httpd.service
  + it's nano
  + export EDITOR=/usr/bin/vim
  + now it's vi
  + Add a service section
    - [Service]
    - Restart=always
    - RestartSec=5s
      * if the process is killed or whatever, it waits 5 seconds and restarts it
  + systemctl cat httpd.service
    - it concatenates the contents of the two config files!!!
    - new file was created in /etc/systemd/system/httpd.service.d/override.conf
    - it creates a sub dir for the service name and then an override.conf file
  + if the change is not picked up automatically, d
    - systemctl daemon-reload
    - systemctl restart httpd
  + ps aux | grep httpd
  + kill -9 13231
  + systemctl status httpd

## managing unit dependencies

* units have dependencies on other units
* For an overview of all loaded units and their dependencies
  + systemctl list-dependencies
* For a specific unit
  + systemctl list-dependencies sshd.service

## masking services

* some units can't work simultaneously on the same system
* to prevent accidentally starting them
  + systemctl mask
    - this links a unit to /dev/null which ensures it can't start
  + systemctl unmask
* Demo
  + systemctl stop httpd
  + systemctl mask httpd
  + systemctl start httpd
    - fails
  + systemctl unmask httpd

## Lab systemd - Sander Video 15

* make sure httpd auto starts
* edit it's config so that on failure it will continue after 1 minute

## Lab solution systemd - Sander Video 15

* systemctl edit httpd.service
* At the top of the file add
  + [Service]
  + Restart=always
  + RestartSec=60s
* systemctl restart httpd.service
* systemctl status httpd
  + it's not enabled, enable it
  + systemctl enable httpd
* Validate it waits 1 minute and restarts
* killall httpd
* systemctl status httpd
  + Active: activating
    - this means it's waiting x seconds before restarting

# Task Scheduling - Sander Video 16

* systemd timers are the primary solution for RH 9 and they are the default
* cron is still supported
* at for non-recurring

## systemd timers

* a systemd timer always belongs to a systemd service
* systemd provides unit.timer files that go with unit.service files
* if you use a systemd timer, the timer should be enabled / started not the service unit!!!
* systemd timers are often installed from rpm packages
* in the timer unit file, the OnCaledar option specifies when the service should start
* Demo
  + systemctl list-units -t timer
  + systemctl list-unit-files logrotate.\*
    - it's enabled
    - vendor preset is enabled
  + systemctl status logrotate.service
    - shows inactive dead
    - TriggeredBy: logrotate.timer
  + systemctl status logrotate.timer
  + dnf install -y sysstat
    - sysstat grabs info
    - it's used by sar
  + systemctl list-unit-files sysstat.\*
    - sysstat-collect.timer
    - sysstat summary.timer
  + systemctl cat sysstat-collect.timer
    - runs every 10 minutes
    - OnCalendar-\*:00/10
      * \*:00 means any hour minute zero
      * every 10 minutes
* The systemd timer OnCalendar has it's own language to express when it should activate
  + OnCalendar=\*:00/10
    - every 10 minutes
  + OnCalendar=2023-\*-\* 9:9,19:30 and 9:29:30
    - runs every day in 2023
    - at 9:09:30, 9:19:30, and 9:29:30
* Could start the unit a specific time after the unit was last activated. Use:
  + OnUnitActivateSec
* Other options
  + OnBootSec
  + OnStartupSec
* man 7 systemd-time
* Demo
  + the service and timer file names must match
  + cat >> /etc/systemd/system/touchfile.service << EOF
  + [Unit]
  + Description=demo unit
  + [Service]
  + Type=oneshot
  + ExecStart=/usr/bin/touch /tmp/myfile.txt
  + EOF
  + cat >> /etc/systemd/system/touchfile.timer << EOF
  + [Unit]
  + Description=demo timer
  + [Timer]
  + OnCalendar=\*:00/01
    - this is every minute
  + EOF
  + systemctl daemon-reload
  + systemctl start touchfile.timer
  + systemctl status touchfile.service
  + watch ls -l /tmp/myfile.txt
    - every xxx seconds runs ls
  + systemctl stop touchfile.timer

## Scheduling Tasks with Cron

* use crond
  + checks it's config every minute
* /etc/crontab and /etc/cron.d
  + don't edit /etc/crontab, put stuff in /etc/cron.d
* /etc/cron.{hourly,daily,weekly,monthly}
  + drop in for scripts that need to be scheduled on a regular basis
  + just need execute perm set
* minute, hour, dom (day of month), month, dow (day of week)
* For a reminder of the syntax, /etc/crontab!!!
* Ex: 0 \* \* dec 1-5
* Demo
  + 10 \* \* \* \* logger HELLO
    - it's not a full path to the coomand
    - a good example for testing

## anacron

* a service behind cron that ensure jobs are executed on a regular basis, but not at a specific time
* it handles jobs in /etc/cron.{hourly,daily,weekly,monthly}
* config file = /etc/anacrontab
* don't 'change anything in /etc/anacrontab
  + use systemd timers instead

## at

* atd service must be running to run once only jobs using at
* use at <time> to schedule a job
  + opens an interactive shell
  + ctrl-d to close
* see the list of scheduled jobs
  + atq
* remove a job from the list
  + atrm
* Demo
  + at teatime
  + touch /tmp/teatime
  + ctrl-d
  + atq

## Managing Temporary Files

* systemd-tmpfiles is started whie booting and manages temporary files and directories
  + it's a service
  + works with related services
    - systemd-tmpfiles-setup.sservice
      * creates and remove tmp files according to the config
    - systemd-tmpfiles-clean.timer calls systemd-tmpfiles-clean.service to remove tmp files
      * by default this happens 15 minutes after booting
      * and also on a daily basis
  + it will create and delete tmp files automatically, according to config files in these locations
    - /usr/lib/tmpfiles.d/
      * for rpm provided tmp files
    - /etc/tmpfiles.d
      * for stuff you create yourself
    - /run/tmpfiles.d
      * for system generated tmp files
* man tmpfiles.d
* In a config file you would see something like
  + d /run/myfiles 0750 root root
    - creates the dir /run/files if needed
  + D /run/myfiles 0750 root root 1d
    - creates the dir if necessary, and wipes its contents if it already exists
    - files older than 1 day are eligible for automatic removal
* Demo
  + echo "q /tmp 1777 root root 7d" > /etc/tmpfiles.d/tmp.conf
    - creates a config that creates the /tmp dir and removes all unused files after 7 days
    - view this file to see what it looks like
  + systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf
    - checks the syntax
  + echo "d /tmpfiles 0770 root root 30s" > /etc/tmpfiles.d/myfiles.conf
  + systemd-tmpfiles --create /etc/tmpfiles.d/myfiles.conf
    - why does he do this for myfiles.conf but not tmp.conf???
  + ls -ld /tmp/myfiles
  + touch /tmp/myfiles/test
  + ls -ld /tmp/myfiles/test
  + sleep 30
  + ls -ld /tmp/myfiles/test
  + systemd-tmpfiles --clean /etc/tmpfiles.d/myfiles.conf
  + ls -ld /tmp/myfiles/test
    - now the file is gone
  + the timer needs to be enabled for the cleaning to happen automatically, otherwise you need to manually clean it
* What about the defaults in ghori, it's more like xxx number of days for /tmp and xxx number of days for /var/tmp

## lab - schduling - Sander Video 16

* ensure the systemd timer that cleans up tmp files is enalbed
  + systemctl list-unit-files -t timer
  + systemctl cat systemd-tmpfiles-clean.timer
    - review the timings inside
* run a cron job
  + issue the command touch /tmp/cronfile 5 minutes from now
    - crontab -e
    - 13 15 \* \* \* touch /tmp/cronfile
  + sometimes it takes cron a long time to pick up a config change, just restart it to speed it up!!!
    - systemctl restart crond
* use at to schedule a job to power off your box at a convenient time later today
  + at 5 pm[enter]
    - poweroff
    - ctrl-d
  + atq

## lab solution - scheduling - Sander Video 16

* ensure the systemd timer that cleans up tmp files is enalbed
  + systemctl list-unit
* run a cron job
  + issue the command touch /tmp/cronfile 5 minutes from now
* use at to schedule a job to power off your box at a convenient time later today

# Configuring Logging - Sander Video 17

* systemd-journald
  + receives log messages from different locations
    - kernel
    - early boot procedure
    - syslog events
    - standard output and errors from daemons
  + it's not persistent by default
* rsyslog reads syslog messages and writes them to different locations
  + /var/log
  + output modules
* services may write to /var/log

## Using journald

* it's not really journald that shows the last messages most easily, it's
  + systemctl status name.unit
* show entire journal
  + journalctl
    - important messages are in red
* show only messages with a priority of error and higher
  + journalctl -p err
* shows the last 10 files and adds new messages as they come in, like follow
  + journalctl -f
* show messages for 1 service only
  + journalctl -u sshd.service
* specify time
  + journalctl --since "-1 hour"
  + journalctl --since today
* verbose
  + journalctl -o verbose
* view boot logs
  + current boot log
    - journalctl -b
  + add explanation texts to boot log messages
    - journalctl -xb
  + show all boot that have been logged, on persistent journal only
    - journalctl --list-boots
  + show messages from a specific boot
    - journalctl -b 3
* journald receives input from
  + services
  + kernel
  + early boot
  + systemd
* rsyslog receives input from
  + facilities
  + journald
* rsyslog sends output based on OM, output modules, and it goes to files in /var/log
  + output could also go to a database
* /etc/rsyslog.conf
* rsyslog can do remote logging

## Preserving journald and making it persistent

* it's not persistent by default
* persistency is taken care of by rsyslog
* rsyslog offers good filtering
* review /etc/systemd/journal.conf
  + Storage=auto
    - ensure persistent storage is happening once the directory /var/log/journal is created
    - other options for Storage are
      * persistent
        + stores journals in /var/log/journal
      * volatile
        + stores journals temporarily in /run/log/journal
      * none
        + doesn't use any storage for the journal at all
* consider size of the journal
  + ensure it doesn't grow in an unlimited way
    - log rotation triggers monthly
    - could set it so no more than 10% of the fs is used
    - or no more than 15% of the file system free size can be used
* To check the settings of /etc/systemd/journal.conf
  + journalctl | grep -E 'Runtime Journal|SystemJournal'
* Demo
  + journalctl | grep -E 'Runtime Journal|SystemJournal'
  + grep -i 'Storage='/etc/systemd/journal.conf'
    - shows a # in front of Storage=auto
    - he says it's the default
  + mkdir /var/log/journal
  + ls /var/log/journal
  + systemctl restart systemd-journald
  + this syncs it with /var/log/journal
    - it can take 1 - 2 minutes

## Configuring rsyslog

* uses rsylogd
* main file
  + /etc/rsyslog.conf
* drop in files
  + /etc/rsyslog.d
* logger lines contain 3 items
  + facility
    - the facility the log is created for
  + severity
    - the severity from which should be logged
  + destination
    - the file or other destination the log should be written to
* use logger to write messages to rsyslog
* facilities
  + rsyslogd must be backwards compatible with the ancient syslog service
  + syslog defined a fixed number of facilities
    - For services that don't have their own facility, use
      * local{0..7}
    - Because of the lack of facilities, some services take care of their own logging and don't use rsyslog
      * httpd is an example
* Demo
  + vi /etc/rsyslog.conf
    - moduleload='injournal'
    - this is the line that allows systemd to receive messages from the systemd journal
  + rules
    - authpriv logs are /var/log/secure
    - \*.info
      * anything with priority of info or higher
    - \*.emerg
      * sends to :omusrmsg:\*
        + goes to all users currently connected

## log rotate

* started by a systemd time to prevent them from growing too large
* after rotation the file is renamed to a file with the rotation date as an extension
* when too many file are rotated, the oldest will be discarded
* configure here
  + /etc/logrotate.conf
  + /etc/logrotate.d
* Demo
  + systemctl list-unit-files -t timer
  + logrotate.time is enabled
  + systemctl cat logrotate.service
    - a lot of hardening options
    - ExecStart
      * it's running logrotate with /etc/logrotate.conf
  + vi /etc/logrotate.conf
    - weekly
    - rotate 4
    - those are the defaults

## lab logging - Sander Video 17

* make sure systemd journal is logged persistently
* create an entry in rsyslog that writes all messages with a severity of error or higher to /var/log/error
* ensure /var/log/error is rotated on a montly basis, and the last 12 logs are kept before they are rotated out

## lab solution for logging - Sander Video 17

* make sure systemd journal is logged persistently
  + mkdir /var/log/journal
  + crazy long string file
  + vi /etc/systemd/journal.conf
  + Storage=auto
    - it's ok for it to have a # in front
* create an entry in rsyslog that writes all messages with a severity of error or higher to /var/log/error
  + vi /etc/rsyslog.conf
    - \*.err /var/log/error
    - systemctl restart rsyslog
  + test it with logger
  + logger --help
    - logger -p err hello sweet priority
  + cat /var/log/error
    - confirm message is logged
* ensure /var/log/error is rotated on a montly basis, and the last 12 logs are kept before they are rotated out
  + vi /etc/logrotate.conf
    - he didn't edit anything here, he will use a drop in file
  + cd /etc/logrotate.d
  + cp chrony error
  + vi error
    - swap error for chrony in the top line
    - delete all of the listed params
    - add these params (decide what you want from logrotate.conf
      * monthly
      * create
      * rotate 12
      * dateext

# Partitions and Mounts - Sander Video - Chapter 18

## General

* There are handwritten notes, delete them after I type these up
* Partition schemes
  + mbr
  + gpt
* lsblk
  + lists block devices
  + lsblk info comes from /proc/partitions
* loop devices can be used to mount iso files
* For exam do practice mbr stuff

## partitions

* sd and vd begin numbering parts at 1
  + sda1, vda1
* nvme adds a p and then starts at 1
  + nvme0n1p1
  + the number after n is the disk number

## mbr partitions

* can only do 4 partitions
  + /boot/
  + /
  + /home
  + you take the 4th partition and make it extended
    - inside of the extended you make logical partitions

## gpt partitions

* can make 128
* no extended and logical partitoins
* guid = global partition table

## fdisk

* can use it for mbr and gpt now
* for gpt parts use g
* list the parts
  + fdisk -l /dev/sda
* t to change type
* l
  + display available partition types
* m for menu/help

## gdisk

* for gpt partitions

## parted

* scriptable
* advanced features are hidden

## block devices

* use lsblk to see them
* /dev/sdx is scsi and sata
* /dev/vdx is a kvm virtual machine
* /dev/nvmexny is nvme
  + nvme0n1

## partition types

* currently partitions will work, even if you set the wrong type
  + needs to be right on the exam
* hex codes or aliases
* aliases
  + linux
    - standard linux
  + swap
  + uefi
    - uefi boot
  + lvm

## create partitions with fdisk

* lsblk
* fdisk /dev/sda
  + warning about disk in use, no worries
  + p print table
  + n for new
  + discusses primary or extended, so you know it’s an mbr disk
  + p for primary
  + default number
  + default first sector
  + +1G last sector
  + p print table
  + w write table
* validate
  + lsblk
* extended partition
  + fdisk /dev/sda
  + e for extended
  + since it’s extended allocate all of the remaining space
  + an extended partition by itself is useless
    - it’s just a vessel for logical partitions
  + n for new
    - adding logical parition 5
    - +2G
  + p to print
  + w to write
* validate
  + lsblk
  + sda4 is 1k partition
    - it’s not useable
    - it’s the extended partition
* do the file systems
* lsblk
* mkfs.xfs /dev/sda3
* mount /dev/sda3 /mnt
* validate
  + lsblk
  + mount | grep ‘/^’
    - avoide all the garbage and see real file systems only
* findmnt
  + shows the hierarchy
* Can’t mount it if you forgot to put a fs on it
* Can’t mount an extended partition

## File Systems

* XFS uses CoW to guarantee data integrity
  + copy on write
  + if you write a file you copy the old file so if anything goes wrong you can easily go back
* vfat offers multi-OS support
  + use it for shared devices like thumb drives
* btrfs
  + butter fs
  + butter comes from a cow, copy on right
* mkfs.<type>
  + mkfs.xfs
  + mkfs.ext4
  + mkfs.vfat
  + mkfs.[tab][tab]
  + mkfs by itself is ext2
* mount /dev/vdb1 /mnt
  + mount it on /mnt, the directory for temporary mounts
* mount
  + by itself it shows all mounts
* umount
  + use lsof /mnt if you get an error re open files
* findmnt
  + shows which file system is mounted where in the directory structure

## /etc/fstab

* /etc/fstab is used to generate systemd mounts by the sytemd-fstab-generator utility
* run this after changing /etc/fstab
  + systemctl daemon-reload
* then mount -a
  + mounts anything not mounted yet
* could do this to verify syntax of /etc/fstab
  + findmnt --verify
* if you jack up /etc/fstab your system will drop into a troubleshooting shell when it boots
* Example of mounting
* mkdir mydata
  + what you will mount on
* vi /etc/fstab
  + /dev/sda3 /mydata xfs default 0 0
* can do tabs or spaces in the file
* For practice, introduce an error into /etc/fstab
* mount -a will error out
* findmnt --verify will issue warnings
* reboot
* e while booting
* edit linux line and remove rhgb quiet
* ctrl-x
* wait 1.5 minutes it says a start job is running for /dev/sd3
* drops you into root shell
* enter root password
* if you leave the exam with the box in a root shell, you get 0 points for the exam
* mount
  + confirm you are in a read / write file system
* lsblk
* vim /etc/fstab
  + fix your error
* exit
* it will boot

## UUIDs and Labels

* block device names may change
* UUID is unique
* Label
  + best to do: mkfs.<type> -L <label>while creating the file system and add a label
  + if you forget
    - tune2fs -L <label> for ext fs
    - xfs\_admin -L <label> for xfs fs
* /dev/disk/\* has unique file names
* blkid
  + shows all devices with their naming attributes
  + shows labels and uuids
* XFS UUIDs on cloned devices - not on exam
  + cloning clones the UUID as well
  + you can create a new UUID
    - xfs\_admin -U generate /dev/sdb1
* labels

Demo

* mkfs.xfs /dev/sda5
* mkfs.ext4 /dev/sda6
* mkdir /xfs /ext4
* vi /etc/fstab
  + /dev/sda5 /xfs xfs defaults 0 0
  + /dev/sda6 /ext4 ext4 defaults 0 0
* mount -a
* lsblk
* create a test file
  + touch /ext4/ext4file
  + touch /xfs/xfsfile
* he wants to remove partition sda5
* umount /xfs
* vi /etc/fstab
  + comment out the line for sda5 with a #
* fdisk /dev/sda
  + p
  + d to delete
  + 5 partition 5
  + w to write
* reboot
  + having trouble finding /dev/sda6 now that you just deleted sda5
  + this is how device names change, so use uuid or label
* in troubleshooting shell
* enter root pw
* lsblk
* blkid | grep sda5 | awk ‘{ print $2 }’
  + shows uuid
* vi /etc/fstab
  + comment out /dev/sda6
  + do a new line with the UUID
  + UUID=”long string” /ext4 ext4 defaults 0 0
* mount -a
* lsblk
* ctrl-d to exit troubleshooting shell
* Labels Demo
  + fdisk /dev/sda
  + p
  + he accepts defaults to use existing space
  + Do you want to remove the signature
    - say no
    - sda6 will be what sda5 was before
  + w to write
* set a label
* xfs\_admin -l <label name> /dev/sda6
* vi /etc/fstab
  + LABEL=cow /xfs xfs defaults 0 0
* mount -a
* ls /xfs
* Best practice = UUIDs

## systemd mounts - don’t use on the exam

* lines in /etc/fstab are converted to systemd mounts
* check /run/systemd/generator for auto generated files
* you don’t have to use /etc/fstab, mounts can be created using systemd .mount files
  + using .mount files lets you be more specific in defining dependencies
* look at this for an example
  + systemctl cat tmp.mount
  + [Mount] section shows What=tmpfs
    - tmpfs is a ram drive
    - mounted on /tmp
    - need to enable and start it
    - systemctl enable --now tmp.mount
    - you won’t see it in /etc/fstab at all
    - THIS COULD COME UP AT WORK IF SOMEONE STARTS MOUNTING WITH SYSTEMD, THEY WON’T SEE STUFF MOUNTED
    - systemctl status tmp.mount
* cd /run/systemd/generator
  + ls
  + these are all the mount files that have been created
  + vi mydata.mount
    - [Mount] section show what and where mounted

## Creating a swap partition

* swap can be created on any block device, including files
* use linux-swap as partition type
* use mkswap to create the swap fs
* activate it with swapon
* Swap Demo
  + fdisk /dev/sda
  + n for new
  + +2G
  + p
  + t for type
  + 7
  + swap
  + x for expert mode
  + f to fix partition order
  + r to return
  + p
  + now order is right
  + w
  + reboot
  + lsblk
  + free -n
  + mkswap /dev/sda7
  + free -n
    - still same
  + swapon /dev/sda7
  + vi /etc/fstab
    - could use a uuid, but since he’s not gonna change anything
      * /dev/sda7 none swap defaults 0 0
      * you can use none or swap, swap is older
  + swapoff /dev/sda7
  + swapon -a
    - activate all swap not currently activate
    - free -m
    - now you see the new swap

## Partitions and Mounts Lab - Sander Video 18

* Create a 10GiB additional hard dirk
* Create a 1Gib primary part
* format it with ext4
* mount it persistently on /mounts/files using its uuid
* create an extended partition that includes all remaining disk space
  + this would be a 500MiB XFS partition
  + mount it persistently on /mounts/xfs via label myxfs
* create a 500MiB swap partition and mount it persistently

## Partitions and Mounts Lab Solution - Sander Video 18

* lsblk
* fdisk /dev/nvme0n1
  + n for new
  + 1
  + p for primary
  + +1G
  + n for new
  + e for extended
  + n for new
  + +500M
  + first logical partition is always partition 5
  + n
  + +500M
  + t for type
  + 6
  + swap
  + p
  + w
* mkdir -p /mounts/files /mounts/xfs
* mkfs.ext4 /dev/nvme0n1p1
* mkfs.xfs -L myxfs /dev/nvme0n1p5
* mkswap /dev/nvme0n1p6
* need uuid
  + blkid
* vi /etc/fstab
  + UUID=”long string” /mounts/files ext4 defaults 0 0
  + LABEL=myxfs /mounts/xfs xfs defaults 0 0
  + UUID=”long string” none swap defaults 0 0
* findmnt --verify
  + warnings are cool to ignore
* reboot to validate
* lsblk
* usually partition 4 is extended, but here it’s 2

# LVMs - Sander Video - Chapter 19

## General

* They are used during the default install of RHEL
* stratis
  + it’s implemented in user space which makes api access possible
* virtual data optimizer - NOT ON THE EXAM!!!
  + now it’s integrated in lvm

## Overall Concept

* Start with the Volume Group
  + It’s made up of physical volumes
    - disks or partitions, any block device
* Create logical volumes from the volume group
  + add file system and mount them

## Creating an LVM

* Create a partition and set the partition type to lvm
  + You can also create lvms on disk devices
* use pvcreate /dev/sdb1 to create the physical volume
* use vgcreate vgdata /deve/sdb1 to create the volume group
* use lvcreate -n lvdata -L 1G vgdata to create the logical volume
* use mkfs /dev/vgdata/vldata to create a filesystem
* put it in /etc/fstab to mount it persistently
  + You don’t need to use labels or uuids with lvms in /etc/fstab because lvm names don’t change
* Example creation of LVM with gdisk
  + gdisk isn’t really needed anymore because fdisk can do both
  + gdisk /dev/sdb
  + ? for help
  + n for new partition
  + use the proposed starting sector
  + for ending sector +2G
  + L to show hex codes
  + enter a string to search
  + lvm, it shows you that it’s 8e00 for lvm
  + p to print and verify
  + w to write
  + pvcreate /dev/sdb1
    - adds metadata
  + vgcreate -S 8M vgdata /dev/sdb1
    - extent size
  + vgs to validate
  + vgdispslay
    - check extent size
    - PE Size
  + lvcreate -L 1G -n lvdata /dev/vgdata
  + validate with lvs
  + add the filesystem
    - mkfs.vfat /dev/vgdata/lvdata
  + create a mount point
    - mkdir /lvdata
  + add it to /etc/fstab
    - vi /etc/fstab
      * /dev/vgdata/lvdata /lvdata vfat default 0 0
  + mount -a
  + reboot to make sure it’s persistent

## Understanding Device Mapper and LVM Device Names

* Device Mapper is the system the kernel uses to interface storage devices
  + It’s used by lvm, stratis, multipath and some other storage systems
* DM generates non-persistent names like /dev/dm-0 and /dev/dm-1
  + Don’t use these non-persistent names
* Persistent names are provided as symbolic links thru /dev/mapper
  + Ex: /dev/mapper/vgdata-lvdata
* Alternatively use the LVM generated symbolic links
  + /dev/vgdata/lvdata
* mount
* shows the example we created above
* what mount shows is in the one format, and what /etc/fstab shows is the other
* they are both just different sym links. To see them
  + ls -hal /dev/mapper/vgdata-lvdata
  + ls -hal /dev/vgdata/lvdata

## Extents

* Extents are the building blocks of lvm allocation
* Define the size while defining the volume group
  + Ex: vgcreate -s 8M vgdata /dev/sdb1
    - where 8M is the extne tsize
* All LVMs from the same VG will have the same extent size
* vgdisplay shows the extent size

## Resizing LVMs

* Confirm the VG has unused disk space
  + vgs
  + If it doesn’t use vgextent to add one or more PVs to the VG
* Use lvextend -r -L +1G go grown the LVM including the file system!!!
  + -r takes care of the file system
  + if you forget to use -r and still need to grow the fs, use
    - resize2fs
      * for ext file systems
    - xfs\_growfs
      * for xfs file systems
* Sander’s Resizing Demo
  + create 2 partitions with a size of 1GiB each and set the lvm partition type
    - lsblk
    - he will use sdg
    - fdisk /dev/sdg
    - g
      * gives it a gpt disk label
    - n
    - +1G
    - t
      * lvm
    - n
      * for second partition
    - +1G
    - t
      * lvm
    - w
      * write it
  + vgcreate vgfiles /dev/sdg1
    - He’s not doing pvcreate separately because vgcreate detects that the pv still needs to be flagged as a pv
      * the output confirms this
  + lvcreate -l 255 -n lvfiles /dev/vgfiles
    - -l <number of extents>
    - 255 extents
    - can’t use 256 because 1 is used for metadata, so 255
  + mkfs.ext4 /dev/vgfiles/lvfiles
  + mount /dev/vgfiles/lvfiles /mnt
  + df -h
  + vgs
  + vgextend vgfiles /dev/sdg2
    - adds a new pv to the vg
  + vgs
    - see more xxx
  + lvextend -r -l +50%FREE /dev/vgfiles/lvfiles
    - -r to grow the file system too
  + df -h
    - shows the new available space of the file system

## Reducing Volume Groups

* If a VG consists of multiple PVs, a PV can be removed if there is sufficient space remaining
* Won’t work if the remaining PVs are fully used already
* Overview
  + use pvmove to move used extents from the volume you want to remove to the remaining volumes
  + use vgreduce to complete the removal
* Sander Demo for removing a vg from a pv
  + use fdisk to create 2 partitions with a size of 2 GB each and set type to lvm
    - fdisk /dev/sdh
    - g for gpt disk
    - n for new
    - 1
    - start at suggested sector
    - +2G
    - t is lvm
    - n for new second partition
    - start at suggested sector
    - +2G
    - t is lvm
    - w to write
  + vgcreate vgdemo /dev/sdh1
  + lvcreate -L 1G -n lvdemo /dev/vgdemo
    - Do not use all the space
  + vgs
    - see free space
  + vgextend vgdemo /dev/sdf2
  + pvs
  + vgextend vgdemo /dev/sdh2
    - pvmove only works if you have multiple pvs so you extend
    - pvs
  + lvextend -L +500M /dev/vgdemo/lvdemo /dev/sdf2
  + pvs
  + mkfs.ext4 /dev/vgdemo/lvdemo
  + mount /dev/vgdemo/lvdemo /mnt
  + df -h
  + dd if=/dev/zero of=/mnt/bigfile bs=1M count 1100
    - ensures that data is on both PVs
    - 1100 ensures data is on both
  + pvmove -v /dev/sdh2 /dev/sdh1
    - moves all used extents from sdh2 to sdh1
    - can take awhile
  + pvs
    - /dev/sdh2 is now unused
  + vgreduce vgdemo /dev/sdh2
    - is now allowed
  + pvs

## LVM Lab - Sander Video 19

* To do this, add a new 10GiB disk to your vm
* create an lvm with the name lvdb and a size of 1GiB
* create the vg and pv that are needed for the lv
* format the lv with xfs and mount it persistently on /mounts/lvdb

## LVM Lab Solution - Sander Video 19

* lsblk
* fdisk /dev/nvme0n2
  + g for gpt partition table
  + n for new
  + 1
  + use first sector
  + +1G
  + t lvm
  + p
  + w
* vgcreate vgdb /dev/nvme0n2p1
  + creates a physical volume and a vg at the same time
* lvcreate -l 255 -n lvdb /dev/vgdb
  + Can’t use -L 1G because not enough extents since one is used for metadata
* mkfs.xfs /dev/vgdb/lvdb
* mkdir /mounts/lvdb
* vi /etc/fstab
  + /dev/vgdb/lvdb /mounts/lvdb xfs defaults 0 0
* mount -a

# Stratis - not on the exam (see screencap on phone) - Sander video 20

## How storage is organized

* lowest level
  + block devices: disks
    - scsi, nvme, fc, virtual
* next layer
  + multipath
    - an optional component that is used in storage network to provide redundancy to different block devices
  + only used when you use external storage networks
* next layer
  + raid
* next layer
  + partitions
* next layer
  + file systems
  + LVM
    - if this then file systems on next layer
  + stratis
    - if this then file systems on next layer

## Creating Stratis Volumes

* stratis volumes always use XFS
  + if you create a stratis volume, you get xfs for free
* thin provisioned
  + Because it's thin provisioned, stratis tools must be used to monitor available space
* volume storage is allocated from the stratis pool
* each stratis volume needs a minimal size of 4 GiB
* there is a pool, say it has 10 TBs
  + you put your block devices in the pool
    - stratis calls that blockdev
    - there can be multiple blockdevs because it's extensible
  + you create stratis volumes from the pool
    - the volumes are called file systems?
    - You call it a file system because with a stratis volume, you get a FS
      * you can't separate them
  + The file system consumes space from the pool, but it can't claim a specific amount
* To manage stratis you need these packages
  + stratisd
  + stratis-cli
* Use the stratis command to create pools and filesystems
  + Determine the pool has available space
    - stratis pool list
    - it's the stratis alternative to df -h
* To mount stratis volumes
  + only mound by UUID
  + must include this mount option
    - x-systemd.requires=stratisd.service
* Demo
  + dnf install stratis-cli stratisd
  + systemctl enable --now stratisd
  + stratis [tab][tab]
  + stratis pool create mypool /dev/sdc
    - you can use a disk or a partition here
  + stratis pool list
    - overview of what you have so far
  + stratis pool add-data mypool /dev/sdd
  + stratis blockdev list
    - shows all the block devices in stratis
    - the disks in the pool
  + stratis fs create mypool myfs
    - create the volume/fs
  + mkdir /myfs
  + stratis fs list
    - confir it's all ok
  + lsblk --output=UUID /dev/stratis/mypool/myfs
    - creates and displays the uuid for the filesystem
    - could redirect it to /etc/fstab
      * lsblk --output=UUID /dev/stratis/mypool/myfs >> /etc/fstab
  + Edit /etc/fstab and add
    - UUID=longstring /myfs xfs x-systemd.requires=stratisd.service 0 0
      * could double quote the uuid, but it's not mandatory
      * could add defaults, with the option, but since it's the default, it's there anyway
  + mount -a
  + mount
    - review the last entry
  + df -h
    - shows 1 TB, misleading
    - use stratis pool list for the truth
* if the pool has 4GB, it's up to the file systems to decide how to carve out what they need

## stratis snapshots

* a stratis snapshot is a metadata copy that allows you to access the state of the snapshot at any time
* a snapshot is not a backup
  + but it can be helpful in accessing deleted files
  + if you remove the original volume, the snapshot becomes useless
* a snapshot is mounted by it's device name, not by uuid
* Demo
  + dd if=/dev/zero of=/myfs/bigfile bs=1M count=2000
  + stratis pool list
  + stratis fs list
  + stratis fs snapshot mypool myfs myfs-snap
  + stratis fs list
    - see both
  + rm /myfs/bigfile
  + mkdir /myfs-snap
    - snaps need their own mount
  + mount /dev/stratis/mypool/myfs-snap /myfs-snap
  + ls -l /myfs-snap
    - bigfile still exists
  + umount /myfs-snap
  + stratis fs destroy mypool myfs-snap
    - snaps are meant to be used temporarily

## lab - Sander stratis video 20

* lsblk
* create a stratis pool of size 10 GiB
  + name it stratispool
  + the pool should have 2 filesystems
    - myfiles
    - myprograms
* mount the volumes/file systems persistently on /myfiles and /myprograms
* copy all the fiels from /etc/ that start with a, c or f to /myfiles
* create a snapshot of myfiles filesystem
* delete all the files from /myfiles that have a nam starting with a
* verify you can still access these files from snapshot

## lab solution - Sander stratis video 20

* lsblk
* create a stratis pool of size 10 GiB
  + name it stratispool
  + the pool should have 2 filesystems
    - myfiles
    - myprograms
  + stratis pool create pool20 /dev/sde
  + stratis pool list
  + stratis fs create pool20 myfiles
  + stratis fs create pool20 myprograms
  + stratis fs list
  + get the uuid
  + stratis fs list | awk '/progra/ { print $NF }'
    - shows uuid
  + stratis fs list | awk '/progra/ { print $NF }' >> /etc/fstab
    - add the uuid to /etc/fstab
  + stratis fs list | awk '/files/ { print $NF }'
    - shows uuid
  + stratis fs list | awk '/files/ { print $NF }' >> /etc/fstab
    - add the uuid to /etc/fstab
  + vi /etc/fstab
    - see 2 uuids
    - fininsh the line for both
    - UUID=lonstring /myfs xfs x.systemd.requires-stratisd.service 0 0
    - UUID=lonstring /myprograms xfs x.systemd.requires-stratisd.service 0 0
* mount the volumes/file systems persistently on /myfiles and /myprograms
  + mkdir /myfiles /myprograms
  + mount -a
  + df -h
  + lsblk
  + stratis fs list
* copy all the fiels from /etc/ that start with a, c or f to /myfiles
  + cp /etc/[acf]\* /myfiles
  + ls /myfiles
* create a snapshot of myfiles filesystem
  + stratis filesystem snapshot pool20 myfiles-snap
  + mkdir /myfiles-snap
* delete all the files from /myfiles that have a nam starting with a
  + rm -f /myfiles/a\*
  + mount /dev/stratis/pool20/myfiles-snap /myfiles-snap
* verify you can still access these files from snapshot

# Managing the Boot Procedure - Sander Video 21

## Steps

* firmware
* boot device allocated
  + boot sector
* grub
* load kernel
* initramfs or initramrd
* systemd
  + early stage
  + services
  + shell
* login

## grub2 - changing run time params

* at boot, press e to edit
* the linux line defines how things are booted
  + at the end of the linux line, possible edits
    - systemd.unit=emergency.target
    - systemd.unit=rescue.target
    - remove rhgb quiet
    - they hide good boot messages
  + ctrl-x to boot
* could press c eo enter grub2 command mode
  + type: help
  + this is hard to use c

## grub2 - changing persistent params

* edit /etc/default/grub
  + then compile changes to grub.cfg
    - MBR: grub2-mkconfig -o /boot/grub2/grub.cfg
    - UEFI: grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
* How to know if I’m on a UEFI system or not
  + lsblk | less
    - find your boot disk
      * if it only has a /boot partition it’s MBR
      * if you are on efi you see a separate vfat partition mounted on /boot/efi
* vi /etc/default/grub
  + Edit the CMDLINE LINUX line
    - at the end of the line remove rhgb guiet
  + To make that change effective
    - grub2-mkconfig -o /boot/grub2/grub.cfg

## managing systemd targets

* a systemd target is a group of unit files
* some targets are isolatable, which means they define the feinal state a system is starting in
  + emergency.target
  + rescue.target
  + multi-user.target
  + graphical.target
* when enabling a unit, it’s added to a specific target
* systemctl list-dependencies
  + good way to see what target you are in
* systemctl cat sshd-service
  + need more info re systemctl cat
* See what target you are in
  + systemctl get-default
  + he’s in graphical
* Set a new default target
  + systemctl set-default
  + he switches
    - systemctl set-default multi-user.target
* Whenever you change your target, reboot to verify
* Booting into a specific target
  + on the grub 2 boot prompt, use systemctl.unit=xxx.target to boot into a specific target
  + to change between targets on a running system, use systemctl isolate xxx.target
    - systemctl isolate multi-user.target
      * blew away gui stuff and let you log in with text mode
* Example
  + in graphical mode he runs systemctl isolate multi-user.target
  + reboot
  + e
  + add systemd.unit=emergency.target
  + ctrl-x
  + prompts for root password
  + now you are in emergency target
  + systemctl list-dependencies
  + nothing is green, everything else is disabled, you can see how much is disabled with emergency target

## Lab - Managing the Boot Procedure - Sander Video 21

* Configure your system to boot in a multi-user target by default
* Persistently remove the options that hide startup messages while booting

## Lab Solution- Managing the Boot Procedure - Sander Video 21

* He’s doing this from the emergency target
* touch blah
  + errors out, it’s read only
  + mount -o remount,rw /
    - remount root in a read writeable way
  + check to see if you have access to the /boot directory
    - you don’t
  + mount -a
    - should give you contents in the boot directory
    - ignore the other errors, because alot can’t be accessed in emergency target
  + ls -hal /boot
    - stuff is there now
  + systemctl set-default multi-user target
  + vi /etc/default/grub
    - if rhgb and quiet are still there, delete them
  + grub2-mkconfig -o /boot/grub2/grub.cfg
    - no such file error
    - he should not be doing this in emergency mode
      * wtf then!?!
  + reboot and you will be in multi-user mode
  + He already made the change by editing the file, now he has to enable it
    - grub2-mkconfig -o /boot/grub2/grub.cfg
  + Get your graphical target back
  + systemctl set-default graphical.target
  + reboot

# Troubleshooting RHEL - Sander Video 22

## Using Troubleshooting Modes

* A person standing next to a blackboard

  Description automatically generated
* Hist boot sequence, start at bottom and go to top
* shell
* services
* early
* systemd
* initramfs
* kernel
* grub
* boot device
* firmware
* In booting, the first thing that can go wrong is you never reach the boot device
  + solution: external boot disk
* In booting, the second that that can go wrong is you get to your boot device, but don't get to grub
  + solution: grub shell
    - manually do what grub is doing
    - do stuff at the grub prompt or menu
    - he says this could be on the exam
* rd.break option
  + stops the boot right after loading initramfs
* init=/bin/bash
  + this is one step later than rd.break
  + here the root file system is already mounted
* after systemd
  + emergency.target
* if you pass early after systemd you can use
  + rescue.target
  + you get more in rescue than emergency, but you don't get all of your services
  + use rescue.target if you have a problem with your services

## Changing the lost root password

* enter grub menu while booting
* find the line that loads the linux kernel
  + add this to the end of the line
    - init=/bin/bash
    - this starts things with bash as the init / first process, not systemd
  + in RH8 you would use rd.break, but in RH9 it will ask you for the root password, so it won't work for RH9!!!
  + now you are in a read only file system
    - mount -o remount,rw /
  + reset the password
    - passwd root
  + get yourself good with selinux
    - touch /.autorelabel
  + start systemd
  + exec /usr/lib/systemd/systemd
    - exec throws out your current process and lets systemd be the first process like it demands
  + continue with boot
* Demo
  + reboot
  + at grub, do e to edit
  + at the end of the line that says linux add, init=/bin/bash
  + drops you in a root shell
  + touch blah
    - it's ready only
  + mount -o remount,rw /
  + passwd root
  + touch /.autorelabel
  + exec /usr/lib/systemd/systemd
    - if this takes excessively long, reset your vm
  + log in as student
  + validate
    - su -
      * asks for root password
    - can't test with sudo su - because that asks for your password

## Using the root debug shell

* it's better for analyzing than troubleshooting, because it needs to be running or able to be set to run to use it
* if something goes wrong early in the boot process, it's useful to have an option to open a dbug shell
* that is why debug-shell.service is provided
  + it's not active by default
* if this service is active, it starts a root shell that can be accessed on tty9 without having to enter a password!!!
* This is good for troubleshooting, but once it's no longer needed, it needs to be removed
  + systemctl disable --now debug-shell.service
* if you want to use it, you have to enable it
* Demo
  + systemctl enable --now debug-shell.service
  + To use it, you need to jack up your system so it won't boot normally
    - vi /etc/fstab
    - he jacked up a uuid in the file and saved it
  + reboot
  + it will wait 1.5 minutes
  + ctrl-alt-functionkey-f9
  + now he's in a weird shell
  + vi /etc/fstab
    - fix what you jacked up
  + reboot
  + turn off the shell
  + systemctl disable --now debug-shell.service

## Troubleshooting File System Issues

* real corruption does occur, but not often, and it's automatically fixed
* Typos in /etc/fstab is usually the case
* Always do mount -a after editing /etc/fstab
  + mounts all filesystems that haven't been mounted yet
* Always do this to verify syntax
  + findmnt --verify
* Final validation on /etc/fstab changes
  + reb
* The fix
  + remount the filesystem in read/write and edit /etc/fstab
* Fragmentation can be an issue
  + it just makes things slow
  + xfs\_fsr is the xfs file system reorganizer
  + e4defrag is for ext4 defragging

## Fixing Network Issues

* Wrong subnet mask
* the router must always be in the local network
* DNS isn't working
  + check /etc/resolv.conf
* Commands for troubleshooting
  + ip addr show
  + ip route show
  + ping 8.8.8.8
  + cat /etc/resolv.conf
  + ping google.com
  + dig google.com
* Demo
  + ip addr show
    - find ip and subnet mask
  + ip route show
    - default route shoud be in the same network as the box
  + ping 8.8.8.8
    - google that is almost always reachable
  + ping google.com
    - if the ip works, but the name doesn't
      * check /etc/resolv.conf
      * this is the dns client, and it's usually generated by network manager
  + dig google.com
    - if you see nx domain, it means there is no such domain
* Exam will mostly focus on you seeing if the ip and subnet mask and the gateway are good
  + maybe memorize some of the small subnet masks
    - is there anything in the man pages that shows subnet mask info???

## Managing Performance Issues

* four key areas of performance
  + memory
  + cpu load
  + disk load
  + network
* start with top
* Demo
  + while true: do true: done &
  + do it 4 times to stress
  + top
  + load average
    - should not be higher than the number of cpu cores
  + cpus line
    - if idle shows 0 than your box is 100% busy
    - k
    - kill the top 4 consuming processes which are the bash while loops you set up
    - it asks you one at a time

## Troubleshooting software issues

* hardly exists
* library problems
  + if you get any messages about .so files or missing symbols
    - run ldconfig to update the library cache!!!
    - some apps need different versions of the same libraries
    - if you use containers, this won't be an issue

## Fixing Memory Shortage

* if available memory by free is low
* as a first help, could use swap
* use vmstat 2 25 to make sure that adding swap doesn't lead to too much i/o traffic
  + reboot
  + e
  + at end of line add MEM=768M
  + ctrl-x
  + box should be slow now
  + ctrl-alt-functionkey to open a virtual terminal if the gui is slow
    - faster
  + logging is as root this way
  + top
    - low memory, some swap is being used
  + vmstat 2 25
    - 2 for 2 second interval
    - 25 for 25 times
    - see columns in the middle
    - si swap in
    - so swap out
    - if these values are close to block in and block out, that's bad

## Consulting Red Hat Websites for Tips

* access.redhat.com
* log in
* use the developer account
* redhat enterprise linux documentation
* also a link for troubleshooting
  + select the version
  + issue summary
    - reset root password
    - get some links
    - read the articles

## Lab Troubleshooting - Sander Video 22

* You lost the root user password, reset it

## Lab Solution Troubleshooting - Sander Video 22

* reboot
* e at grub
* at the end of the linux line add
  + init=/bin/bash
* ctrl-x
* dropped in bash shell
* read only fs
* mount -o remount,rw /
* touch blah
* passwd
  + change root password
* touch /.autorelabel
* exect /usr/lib/systemd/systemd
  + take 1 - 2 minutes for relabeling
* log in as student
* validate
  + at terminal su -
    - asks for root password

# Bash Scripts - Sander Video 23

* shebang
  + #!/bin/bash
* apple uses zsh by default
* Variables
  + key=value
  + color=red
  + echo $color
  + he likes all caps for variables, they do pop, but I thought all caps was for env variables
* See his git repo
* read
  + read command reads in what you type
* for positional parameters, not set up, just use echo $1 in the script
* user input
  + echo What is your name
    - no quotes needed!!!
  + read name
  + echo Your name is $name
* run the script
  + ./argument.sh one two three
  + Only $1 is used in the script, but it’s cool to pass in more
* Conditionals
  + test
    - used in if statements
      * test condition
  + shorthand for test
    - [ condition ]
  + if is used with then, and closed with fi
* man test
  + good info
  + -z see if length of the string is 0
* Ex if statment:
  + if test -z $1
  + then
  + echo you have to provide arguments
  + exit 3
  + fi
* exit of 0 = success
* exit of non-zero = failure
* Ex if else statement:
  + if test $1==”hello”
    - this could have a type, double check
  + then
  + echo you typed hello
  + else
  + echo you have typed something else
  + fi
* People double quote variables like “$1”
  + it avoids interpretation???
* Ex if with alternate syntax
  + if [ -f $1 ]
  + then
  + echo $1 is a file
  + else
  + echo $1 is not a file
  + run it: ./test.sh /etc/hosts
* Make executable
  + he just does
    - chmod +x tset.sh
* Validate
  + echo $?
    - shows the exit code
* Loops
  + his github coundown.sh

#!/bin/bash

if test -z $1

then

echo Provide the number of minutes

read COUNTER

else

COUNTER=$1

fi

COUNTER=$(( COUNTER \* 60 ))

while [ $COUNTER -gt 0 ]

do

echo $COUNTER seconds remaining

COUNTER=$(( COUNTER - 1 ))

sleep 1

done

echo Break is over

* debug mode: bash -x countdown .sh 1
* He did not demo for loops!!!

## Lab for Bash Scripting - Sander Video 23

* Write a script that copies any \*.txt file
  + copy should be named filename.txt.bak
  + after the copying is done, move it to /tmp
  + pass in the directory name to use when running the script
  + if no args are provided, stop the swcript with exit code 9
  + Is this github lab23.sh?

## Bash Lab Solution - Sander Video 23

#!/bin/bash

if [ -z $1 ]

then

echo You need to provide a dir3ectory as an argument

exit 9

fi

for i in $1/\*.txt

do

cp $i $i.bak

mv $i.bak /tmp

done

* Is selinux cool for this?
* Preparation for testing this script
* as root
  + mkdir /myfiles
  + touch /myfiles/{hello,blue|quedeming.txt}
    - this doesn’t look right, double check
  + ./lab23.sh /myfiles
* a different run as root
  + touch /myfiles/{1..100}.txt
  + ls /tmp/\*bak

# Managing ssh - Sander Video 24

## Key Pairs

* client
  + create public / private key pair
* server
  + copy client public key to server
* use ssh-keygen
  + don’t use a passphrase
* use ssh-copy-id to copy client key to server
* do this for student user on client
  + ssh-keygen
  + says it will generate rsa keypair
  + do nothing for passphrase
  + ssh-copy-id <other server>
  + validate
    - ssh from client to server and proove you can do so without a pw

## Caching Keys - Not on the Exam

* In production most people use a passphrase, which is inconvenient
* ssh-agent /bin/bash allocates space in the bash shell to cache the private key passphrase
* then use ssh-add to add the passphrase to the cache
* they are cached for the duration of the ssh session

## Caching Demo

* He says not to use the GUI terminal
* sudo chvt 4
* Log in as student
* ssh-keygen
  + overwrite currently existing file
  + set a passphrase
* ssh-copy-id <other server>
  + Since the old key was redone, it will ask for a password this time
* On the client server
  + ssh-agent /bin/bash
  + ssh add
  + enter passphrase
* he does chvt 2
  + he’s going back to the gui
* He did all that to say that if you set up the agent in a terminal, it will handle the gui also

## Defining SSH Client Config

* ssh -X
  + enables x forwarding
* ssy -Y
  + enables trusted X forwarding
* Generic client config file
  + /etc/ssh/ssh\_config
* Specific config
  + ~/.ssh/ssh\_config
* ssh server gedit1
  + gui app
  + screen on your machine is handled by x server
  + remote stuff is the client
  + need ssh -X server gedit1

## ssh server options

* /etc/ssh/sshd\_config
* Port 22
* PermitRootLogin
* PubkeyAuthentication
  + on by default
* PasswordAuthentication
* X11Forwarding
* AllowUsers
  + list of users allowed to login
  + do we use this?
  + this option doesn’t exist in the file you need to add it
    - AllowUsers student
* Any time you make changes
  + systemctl restart sshd

## scp

* scp uses the sshd process
* scp file1 file2 student@remoteserver:/home/student
* Recursive copy
  + scp -r root@remoteserver:/tmp/files .
* In RHEL 9 can’t scp ro root@blah because by default root can’t log in

## sftp

* uses ssh
* put /my/file
* get /your/file

## rsync

* Uses ssh
* way more efficient
* -r recursively
* -l sync sym links
* -p preserve sym links
* -n dry run
* iA use archive mode and sync ACLs
* -X sync SELinux
* -a uses archive mode and is equivalent to -rlptgoD
* rsync -s \* server:/home/student
  + what does -s do?

## Lab - Managing ssh - Sander Video 24

* Set up your ssh server so that
  + ssh is on port 22
  + user root can log in
  + graphical sessions can be forwarded
* Verify that X forwarding works from the ssh client
* Use localhost as the server address in the lab

## Lab Solution - Managing ssh - Sander Video 24

* Be root
* vi /etc/ssh/sshd
  + port 22 is good
  + PermitRootLogin prohibit password
  + Change this to
    - PermitRootLogin yes
  + X11Forwrading
    - set it to X11Forwarding yes
* systemctl restart sshd
* Test as root
  + ssh root@localhost
    - logged in fine
  + ssh -X localhost gedit
    - Confirm gedit opens a window

# http - sander video 25

## General

* This is not on the exam, but do this to practice selinux
* as root
* main config file = /etc/httpd/conf/httpd.conf
* The default document root = /var/www/html
  + this is where index.html should go
* Probably already installed, but if not
  + dnf install httpd
* In a browser
  + <http://localhost>
    - should see redhat test page
* With curl
  + curl <http://localhost>
    - should see redhat test page
* cd /var/www/html
  + vi index.html
  + No html mark up just
    - Hello Heidi
* You can’t see it yet because the perm is wrong, it’s just rw-r-----, and t needs to be rw-r--r--
* chmod o+r index.html
* validate with curl
  + curl <http://localhost>

## Lab - Sander Video

* Configure apache to serve a basic website that shows hello world
* validate with curl

## Lab Solution - Sander Video

* The reason index.html is not rw r r is because of the umask which sets perm on new files
* See what it’s set to
  + umask => 0027
* Review /etc/bashrc
  + there is conditional logic based on user id that determines what to set it
* See if it’s been overwritten at /root/.bashrc
  + it hasn’t, but lets overwrite it now
  + umask 022
* touch blah
  + it’s still just rw r
* source ~/.bashrc
* touch blah2
* ls
* now new files are rw r r
* 027 for root is very secure, but it can be inconvenient
* fix index.html
  + chmod o+r index.html
* systemctl restart httpd
* validate
  + curl <http://localhost>
  + hello world

# SELinux - sander video 26

* If it isn’t specifically allowed, it’s denied
* If you have vanilla default stuff, you should be able to leave sel on and it will work
* sel is either
  + enabled
  + disabled
  + if you want to switch from disabled to enabled, you have to reboot
    - verify the direction, he confused me
* When it’s enabled it can be in one of two modes
  + enforcing
  + permissive
    - only logs, doesn’t block
* getenforce shows the current state
* setenforce toggles between enforcing and permissive
  + this does not persist
* Configure the default state of sel
  + edit /etc/sysconfig/selinux
* sel state can be set at boot as a kernel param
  + enforcing=0 is permissive mode
  + enforcing =1 is enforcing mode
  + selinux=0 is disabled
  + selinux=1 is enabled
* Change these in the grub boot loader
* Demo
  + getenforce => enforcing
  + setenforce permissive
  + getenforce => permissive
  + cat /etc/sysconfig/selinux
    - it’s a useless file now because no one wants to set sel to permissive and you can’t disable it here
      * I think you could in other versions
  + reboot
  + e at grub line
  + at the end of the linux line add selinux=0 for disbled
    - confirm this is one time only
  + ctrl-x
  + getenforce => disabled
  + setenforce enforcing
    - it won’t due this and tells you that sel is dissabled
  + to switch between enabled and disabled you need to reboot
  + reboot
  + getenforce => enforcing
  + update what is persistent and what isn’t

## selinux components

* context labels
  + define sel permissions
  + applied to source and target objects
* source objects
  + users
  + processes
* target objects
  + files and directories
  + ports
* sel policy
  + rules to define which source context has access to which target context
  + rules are created by labels
* context management is about applying contexts to mostly files, directories, and ports
* you need to apply a context that matches a specific rule
* booleans
  + allow parts of the sel policy to be rewritten to allow/disallow specific functionality
* a source which is usually a process wants to go to a target, which is usually a file
  + the source would have a label like myapp\_t
  + the target would have a label like my\_app\_files\_t
  + these labels need to be mapped, and the mapping happens in the rules
  + the rules are stored in the sel policy

## File Context Labels

* every object is labeled with a context label
  + user: user specific context - ignore for exam
  + role: role specific context - ignore for exam
  + type: flags which type of operation is allowed on this object
* For the exam, only type matters, so ignore user and role
* Many commands have a -Z option to show the current context
* Context types are used in the rules in the policy to define which source object has access to which target object
* There is usually a default context
  + most services don’t need additional sel configuration if a default exists
* When files are created in a directory, they inherit the context of the parent directory
* When files are copied, they inherit the context of the parent directory
* When files are moved, they keep the original context
* To manage file context labels
  + use semanage fcont3ext to set the label
  + it writes the context to the sel policy, but it’s not written to the fs yet
  + the context must land in the inode as well
* To set a new context label
  + semanage fcontext -a
* If a file already has a label, then do
  + semanage fcontext -m to modify it
* Show only settings that have changed in the policy
  + semanage fcontext -l -C
* semanagae fcontext writes to the policy, not to the inode!
* Use restorecon to enforce the policy setting on the fs
  + alternatively, use touch /.autorelabel to relabel all files to the context stored in the policy
* The man page for semanage-fcontext is great
  + it has good examples
* Never use chcon for anything
  + writes to the inode, not the policy
* Demo of file context management
  + cp /etc/hosts /root/home
  + ls -Z /etc/hosts
    - net\_conf\_t
  + ls -Z /root/hosts
    - admin\_home\_t
  + the labels are based on the directory
  + mv /root/hosts /var/www/html
  + ls -Z /var/www/html/
    - hosts file has admin\_home\_t
    - can it be read by web server
    - all the other files in the dir have the label httpd\_sys\_content\_t
    - because you moved the file it has a mismatched label
    - the easiest way to fix mismatched context labels is with restorecon
    - restorecon -Rv /var/www/html
    - output confirms it did the needful
* Demo of what happens if you change default settings
  + vi /etc/httpd/conf/httpd.conf
  + reset the DocumentRoot to /web
  + mkdir /web
  + vi /web/index.html
    - hello from web
  + systemctl restart httpd
  + validate: curl localhost
  + gives the default welcome page because servre can’t connect to it’s document root
  + getenforce
    - you are in enforcing
  + use setenforce permissive and do some troubleshooting
    - if the error was because of selinux, it should work now because it’s in enforcing
    - curl localhost
      * but it still doesn’t work
  + Edit /etc/httpd/conf/httpd.conf
    - Set the <Directory below DocumentRoot to /web
    - systsemctl restart httpd
  + validate
    - curl localhost
    - now it shows your custom index.html
  + put back sel
    - setenforce enforcing
  + validate
    - curl localhost
    - it doesn’t work
    - this time it doesn’t work because of sel
  + look at the sel logs which are the audit logs
    - grep AVC /var/log/audit/audit.log
    - it’s managed by auditd
    - lots of lines, but most recent one says avc: denied
      * scontext is source context is http\_d
      * target context is default\_t
      * this is not allowed
  + man semanage-fcontext
    - See examples at the bottom
    - copy the web example
  + semanage fcontext -a -t httpd sys content t “/web(/.\*)?”
    - it’s a regex
    - this changed the context, can you see it with ls -Z
    - ls -Zd /web
    - no
      * semanage fcontext writes to the policy not the fs
    - to write it to the fs
      * restorecon -Rv /web
      * this recursively applies the context from the policy to the directory
  + getenforce => enforcing
    - still in enforcing mode
  + crul localhost
    - can see your custom text

## Finding the Right Context

* if you apply a non-default config, check the default config context setting
* you could also
  + man selinux-policy-doc
  + use sealert
* dnf search selinux
* dnf install -y selinux-policy-doc
* man -k selinux
* man -k selinux | wc -l => 853 selinux manpages
* if you want to know about httpd
  + man -k selinux | grep httpd
  + man httpd\_selinux
  + look for examples or commands at bottom
* For the exam it’s fasteset to use default contexts with sealert

## Managing port access

* network ports are also provided with a context label
* there is some default port access
* for non-default port access, use semanage port to apply the right label
* See man semanage-port for examples
* demo
  + vi /etc/ssh/sshd\_config
    - this file states if you change the port you need to update sel and it provides an example!
  + switch Port to 2022
  + systemctl restart sshd
    - errors out
  + systemctl status sshd
  + setenforce permissive
  + systemctl restart sshd
    - now it works
    - this proves it’s a sel problem
  + grep -i avc /var/log/audit/audit.log | grep ssh
  + the source and target contexts don’t match up
  + copy the command in /etc/ssh/sshd\_config and edit the port number
  + setenforce enforcing
  + systemctl restart sshd
  + validate port config
    - ss -tunap
    - shows sshd is listening on 2022
    - man semanage-port shows examples
  + if -a doesn’t work do -m for modify

## selinux booleans

* an easy to use config switch to enable/disable parts of the sel policy
* for an overview of booleans
  + semanage boolean -l
  + getsebool -a
  + to set a boolean
    - setsebool -P <boolean> [on|off]
    - -P makes it persistent
  + use semanage blloean -l -C to see all booleans with a non-default setting
* boolean demo
  + getsebool -a | grep ftp
  + setsebool -P ftpd\_anon\_write on
  + semanage boolean -l -C
    - shows all booleans with non-default settings

## selinux log messages

* sel uses auditd to write messages to the audit log
* sealert interprets messages from audit log and applies sel AI and writes meaningful messages to /var/log/messages
* run sealert including UUID message to get advice on how to troubleshoot
* sealert demo
  + journalctl | grep sealert
  + it shows you sealert -l <longstring> | less
    - you execute that as a command
  + he had a rsyslog issue
  + run the suggested command
  + touch /.autorelabel
    - applies the policy settings to the entre system and ensure the file system matches the policy settings
    - it’s never a problem to do an /.autorelabel!!!
  + reboot
    - can see the relabeling in boot messages
    - it will autorelabel and then boot again, could be multiple times
  + if it’s rebooting 5 times, that’s a problem
  + he does e at the boot menu
  + adds systemd.unit=emergency.target to the end of the linux line
  + log in with the root password
  + ls -la /
    - see if .autorelabel is still there
    - it is
    - there was a message about read only file system before, which means it can’t delete .autorelabel
    - rm -rf /.autorelabel
    - it won’t delete it because the fs is read only, so mount it
    - mount -o remount,rw /
    - rm -rf /.autorelabel
    - getenforce => enforcing
    - if autorelabel fails, you can do this
    - restorecon -Rv /
    - reboot
    - confirm all is well
    - he says autorelabel is glitchy on rh9

## Troubleshooting selinux

* if you think sel is blocking access, start by using setenforce 0 and trying again
  + if it works now, you confirmed that sel is blocking
* use grep -i avc /var/log/audit/audit.log to see the messages and look at the source/target contexts
* even if you use sealert, get good at the avc messages
* install selinux-policy-doc for more man pages
  + dnf install selinux-policy-doc
  + try not to use this on the exam because it provides pages and pages of info that will slow you down
* journalctl | grep sealert
* dnf provides \*/sealert
  + if you don’t see sealert stuff in the log
  + dnf install setroubleshoot-server
    - restart auditd or reboot to make it fully functional

## SE Linux Lab - Sander Video 26

* configure apache to bind to port 82
* do: mv /etc/hosts /var/www/html
  + ensure the file gets a sel context that makes it readable by apache

## SE Linux Lab Solution - Sander Video 26

* /vi /etc/httpd/conf/httpd.conf
  + Edit DocumentRoot and set it to /var/www/html
  + Below that, at <Directory ...
    - set that back to /var/www/html
  + Edit the Listen parameter to 82
* if from practing you have anything in /var/www/html, delete it now
* mv /etc/hosts /var/www/html
* systemctl restart httpd
* fails with errors
* journalctl | grep sealert
  + lots of lines, find the one that has bind and one about not reading a file
  + it tells you a command to run for more info
    - copy it and pipe it to less
  + it suggests a fix with a confidence level
  + paste the suggestion into the cli
    - semanage port -a -t http port t -p tcp 82
* he will fix the next issue without sealert
  + cd /var/www/thml
  + ls - Z
  + what it shows won’t work
  + restorecon -Rv /var/www/html
    - output makes sense
  + systemctl restart httpd
* journalctl | grep sealerts
  + no new alerts
* He doesn’t really validate specifically that it’s good, I wish there was more for that

# Firewalls - Sander Video 27

## ss

* ss = sockets statistics
* a network socket is a connection endpoint, consisting of an ip address followed by a port
* sockets also exist as unix sockets, which are endpoints in communication with services
* here are the good ss commands
  + show all connections, a ton of info
    - ss
  + show connected tcp and udp sockets
    - ss -tu
  + show connected tcp and udp sockets in a listening state
    - ss -tua
  + show only tcp sockets in a listening state without resolving host names
    - ss -tln
  + show tcp and udp sockets in a listening state and add the process name or pid
    - ss -tulpn
    - local ip of 0.0.0.0 means it's listending on all ips
    - local ip of 127.0.0.1 means it's listening on loopback
    - :: is ipv6 stuff
    - why can't i access chronyd from the outside
      * in his example, it's only listending on the loopback ip

## rhel firewalling

* the linux krenel provides the netfilter framework to take care of firewall stuff
  + packet filtering
  + network address translation
  + port forwarding
* netfilter forwards specific operations to kernel modules
* nftables is the framework that applies firewalling
* firewalld is a service which rhel uses as the front end to manage nftables firewalls

## firewalld

* firewalld components
  + service
    - contains one or more ports as well as optional kernel modules that should be loaded
* zone
  + default config to which network cards can be assigned to apply specific settings
* ports
  + optional elements to allow access to specific ports
* additional components are available as well, but not frequently used in a base firewall config
* list all components
  + firewall-cmd --list-all
    - shows all of the configuration
    - first line of output is the zone
  + services
    - shows the services the firewall is currently handling

## firewall-cmd

* make it persist, but not be used for runtime
  + --permanent
* make the change for runtime, but not for permanent
  + don't use --permanent
* to make it both runtime and permanent
  + you need to run your commands twice
  + or you write everything to permanent and then reload the firewall
* Demo
  + systemctl status firewalld
  + firewall-cmd --list-all
    - shows all of the configuration
  + firewall-cmd --get-services
    - tons of stuff
    - grep for what you want
  + firewall-cmd --add-service http
    - validate it's good for run time
      * firewall-cmd --list-all
      * should see it listed under services now
      * but if you restart the service or the server it will disappear because it's not permanent
      * run it again with --permanent
  + firewall-cmd --add-service http --permanent
  + validate it's good permanently
    - reload the firewall
      * xxx
    - or reboot
    - firewall-cmd --list-all
      * confirm you see it

## lab - firewalls sander video 27

* configure firewalld so that ssh and ftp are allowed
* the configuration should work for runtime and permanently

## lab solution - firewalls sander video 27

* check what's up
  + firewall-cmd --list-all
    - ssh is already enabled, so nothing to do there
* firewall-cmd --add-service ftp
  + for runtime
* firewall-cmd --add-service ftp --permanent
  + for permanent
* validate runtime with
  + firewall-cmd --list-all
* validate permanent with a reload or reboot
  + reload xxx
* if you don't know the name of the service
  + firewall-cmd get-services

# Automatic Installs - Sander Video 28

* Options
  + vagrant
  + cloud-init
  + kickstart with pxe boot

## kickstart

* Let’s you reproduce installations
* after install you are left with anaconda-ks.cfg
  + you can edit it
  + To check it
  + ksvalidator /root/anaconda-ks.cfg ks-validator to check it
* common options
  + url --url=”<http://myserver/...>” defines which URL to access for the install media
  + repo --name=”myrepo” --baseurl=xxx
  + text
    - force text mode install
  + vnc --password=hermes enables the vnc viewer for remote access to the install
  + clearpart --all --drive=sda,sdb
    - removes all partition on drives sda, and sdb
  + part /home --fstype=ext4 --label=home --size=2048 --maxsize=4096 --grow
    - creates and mounts a parttion
  + autopart
    - automatically creates boot, root and swap partitions
  + network --device=ens33 --bootproto=dhcp
    - configure the primary network interface
  + firewall --enabled --service=ssh,http
    - open the firewall
  + timesource --ntp-server pool.ntp.org
    - set up ntp
  + rootpw --plaintext hermes
    - not secure but convenient
  + selinux --enforcing
    - activates selinux
* it’s used on an installation server
* The client indicates where to get the file
  + ks=http://somewhere/ks.cfg
    - web server
  + ks=hd:LABEL=MYLABEL:/directory/file
    - to access from a device like USB
    - Only works with LABEL= or UUID=
  + use the interface provided by the install program like virtual machine manager

## Installing virtual machines

* KVM VMs
* See if the box can handle virtualization
  + grep -E “svm|vmx” /proc/cpuinfo
* dnf group install “Virtualization Host”
* Validate the host meets all the requirements
  + virt-host-validate
* Install the vm from the cli
  + dnf install virt-install
  + virt-install --name testvm --memory 2048 --vcpus 2 --disk size 20 --os-type linux --cdrom /rhel9.iso
* Install from the web console
  + dnf install cockpit-machines
  + systemctl start cockpit.socket
  + systemctl enable cockpit.socket
  + <https://localhost:9090>
    - login as root
  + From administrative mode, select Virtual Machines => Create VM
    - local media
    - /rhel9.iso is on the filesystem of the host
* Recent versions of MAC OS don’t support nested virtualization well, you will get a failure message

## Lab - Video 28

* Create a kickstart file named my-ks.cfg
  + installer needs to prompt for a password
  + network connectivity is enabled at boot
  + the local machine name = server10.example.com

## Lab Solution - Video 28

* copy the kickstart created when your box was installed
  + cp anaconda-ks my-ks.cfg
* vi my-ks.cfg
  + Edit the network information and add the hostname at the end
    - --hostname=server10.example.com
  + remove the rootpw line
    - it will now prompt you for the password
* dnf provides \*/ksvalidator
  + dnf install pykickstart -y
* validate your file
  + ksvalidator my-ks.cfg

# Configuring Time Services - Sander Video 29

## General

* While booting, the box gets its time from the hardware clock
* system time is set according to the hardware clock
* NTP services
  + chrony
  + systemd-timesync.service
    - not used on rhel
* hwclock is used to set hardware time
  + you can use it to synchronize time to
  + hwclock --systohc
  + hwclock --hctosys
  + hwclock by itself shows you a long string
* date
  + used to show and set time
  + date by itself shows a long string

## timedatectl

* used to manage time and time zones
* a new tool that lets you manage all aspects to system time
* timedatectl status
  + shows all the time properties currently used
* timedatectl set-time
  + change the time
* timedatectl set-timezone
  + change the time zone
* timedatectl set-ntp
  + enables / disables ntp time sync
* timedatectl status
  + shows good stuff
    - local time
    - utc
    - rtc time
    - timezone
    - is it synchronized
* list the timezones
  + timedatectl list-timezones
* change a timezone
  + timedatectl set-timezone Africa/Cairo
  + use the same format as the list command
  + validate
    - timedatectl status
      * zone will change
      * times will change

## Managing an NTP Client - chronyd

* it’s the default ntp service
* use /etc/chrony.conf to specify sync params
* pool 2.rhel.pool.ntp.org iburst
  + configures a pool of ntp servers
* configure a single ntp time source
  + server myserver.example.com
* use iburst to permit fast synchronization
* Once the config is modified
  + systemctl restart chronyd
* validate proper synch
  + chronyc sources
  + The + is on the one you are using
* to know if ntp is active
  + timedatectl
    - NTP service active

## Lab Configuring Time Services - Sander Video 29

* Configure your system to sync with the servers in pool.ntp.org

## Lab Solution - Configuring Time Services - Sander Video 29

* timedatectl status
  + confirm NTP service = active
* vi /etc/chrony.conf
  + it’s already set to use pool.ntp.org
  + pretend you edited it
* systemctl restart chronyd
* validate
  + chronyc sources
    - shows 4 servers
    - none have an asterisk or a +, but he says his server just synchronized with several servers

# NFS - Sander book or video? 30

## General

* Check /etc/auto.misc for examples
* mount
  + by itself, it puts the most recently mounted item at the bottom
* Creating a nfs server is not on the exam, but you need it to practice

## Manually mount nfs

* server1 = nfs server
* server2 = nfs client
* update /etc/hosts on both servers with both names
* This is a combination of Sander and Ghori that I got working

**server1**

* dnf install nfs-utils
* mkdir /nfsdata; cd nfsdata; touch blah{1..3};
* chmod 777
* firewall
  + for i in nfs muontd rpc-bind; do firewall-cmd –add-service $i –permanent; done
  + firewall-cmd –reload
  + firewall-cmd –list-all
* systemctl start nfs-server
* systemctl enable nfs-server
* systemctl status nfs-server
  + enabled, active, exited
    - should it really say exited? It does work...
* vi /etc/exports
  + /nfsdata server2(rw)
    - server2 is who you want to share it to
* exportfs -av

**server2**

* dnf install nfs-utils
* mkdir /local
* mount server1:/nfsdata
* validate
  + cd /local, confirm you see the 3 files you made
  + df -h | grep local

## Unmount your manual mount and mount with /etc/fstab

* Do the previous section and carry on from here

**server2**

* Unmount the manual mount
  + umount /local
* vi /etc/fstab
  + server1:/nfsdata /local nfs sync 0 0
* mount -a
* systemctl daemon-reload
* mount -a

**server 1**

* validation
* touch a new file in /nfsdata

**server 2**

* cd /local
* confirm you see the new file
* df -h
* showmount -e server1
* reboot
* confirm persistence
  + cd /local
  + confirm you see the new file
  + df -h
  + showmount -e server1

## simple autofs

## Set Up

* Start with clean snapshots
  + could leave server1 if you are doing all of these at once
  + for sure blow out server2
* server1 = nfs server
* server2 = nfs client
* update /etc/hosts on both servers with both names
* This is a combination of Sander and Ghori that I got working

**server1**

* dnf install nfs-utils
* mkdir /nfsdata; cd nfsdata; touch blah{1..3};
* chmod 777
* firewall
  + for i in nfs muontd rpc-bind; do firewall-cmd –add-service $i –permanent; done
  + firewall-cmd –reload
  + firewall-cmd –list-all
* systemctl start nfs-server
* systemctl enable nfs-server
* systemctl status nfs-server
  + enabled, active, exited
    - should it really say exited? It does work...
* vi /etc/exports
  + /nfsdata server2(rw)
    - server2 is who you want to share it to

exportfs -av

**server2**

* Most of the work is done here
* dnf install autofs -y
* showmount -e server1
* vi /etc/auto.master
  + after the misc line
  + /nfsdata /etc/auto.nfsdata
* vi /etc/auto.nfsdata
  + mydir -rw server1:nfsdata
  + mydir is a brand new dir that doesn’t exist yet, but that is where your files will be found. This is the dir that you will need to cd into or ls to see the nfs files
* systemctl start autofs
* systemctl enable autofs
* systemctl status autofs
  + enabled, active, running
* ls -hal /
  + there is no /nfsdata directory
* cd /nfsdata/mydir
* you will see the files now
* validate
  + mount
  + read the last entry for autofs info
  + reboot
    - cd /nfsdata/mydir
    - confirm you still see the files
* autofs does not get manually mounted or go into /etc/fstab

## autofs for home directories

## Set Up

* Start with clean snapshots
  + could leave server1 if you are doing all of these at once
  + for sure blow out server2
* server1 = nfs server
* server2 = nfs client
* update /etc/hosts on both servers with both names
* This is a combination of Sander and Ghori that I got working
* When practicing, don’t ever use /home as that would jack up the regular home

**server1**

* dnf install nfs-utils
* mkdir -p /users/user1 /users/user2
* firewall
  + for i in nfs muontd rpc-bind; do firewall-cmd –add-service $i –permanent; done
  + firewall-cmd –reload
  + firewall-cmd –list-all
* systemctl start nfs-server
* systemctl enable nfs-server
* systemctl status nfs-server
  + enabled, active, exited
    - should it really say exited? It does work...
* vi /etc/exports
  + /users \*(rw,no\_root\_squash)
    - server2 is who you want to share it to
* exportfs -av

**server2**

* Most of the work is done here
* dnf install nfs-utils
* dnf install autofs -y
* showmount -e server2.example.com
* vi /etc/auto.master
  + Below the misc line
    - /users /etc/auto.users
* vi /etc/auto.users
  + \* -rw server1:/users/&
  + this is called a wildcard mount
  + Here server1 is the name of the nfs server that stuff is being exported from
    - It’s the opposite direction and server from what would go in /etc/exports
* systemctl start autofs
* systemctl enable autofs
* systemctl status autofs
  + enabled, active, running
* ls -hal /
  + cd /users/user1
  + There is nothing in here, but you can cd in. No where on this server was a dir named user1 manually created

# Containers - Sander - confirm video 31

## General

* Sanders GH for RH 9 includes 2 mini exams
  + <https://github.com/sandervanvugt/rhcsa9>
* Figure out his 27:27 thing
  + mysql id
* log in / ssh in as the user who starts the container, don’t su -
* man podman-gererate-systemd
  + search EXAMPLE
* 27 id default uid for mysql
* 3206 is default port for mysql

## Informational

* Containers need a UID to be started
* rootless containers are started as non-root user
  + can generate a uid dynamically or be preconfigured to use a specific UID
* can’t bind to privileged network ports
* podman is an app similar to docker
* quay.io is a RH community site with images optimized for RH
* If you just want to browse images
  + <https://catalog.redhat.com>
* Exam
  + Need to use, but not write a containerfile
* He clones the course repo
  + as root cd rhcsa
  + cat Containerfile
    - it downloads an ubi image
    - installs nmap
    - runs an nmap command
* Build a custom container
  + podman build -t <tag> .
    - tag is like a name
    - . is current directory
    - he has an error
      * if you do dnf install blah you need to have -y on the end because a container build is not interactive with a container file
  + podman ps -a
    - shows containers built from container files as localhost/blah
* Not sure what this is
  + podman 27
    - it’s a unique path that allows podman to identify the container id
      * podman stop 27
    - is this related to the mysql uid?
* pass a command to run on a container
  + podman run --name sleepy docker.io/redhat/ubi9 sleep 3600

## Troubleshooting Containers

* podman run -it
  + run it this way so it doesn’t disappear after running the default command
  + podman logs
  + podman run busybox
    - busybox is a tiny linux
* container exit status
  + posman ps -a
    - status could show Exited (0)
      * this means it did the needful successfully and exited
    - If it shows Exited (1) it failed
* logs
  + podman logs f31
    - he uses the first 3 chars of the container id
  + didn’t write this out fully, but it works too
    - podman logs <container name>
* skopeo
  + skopeo inspect <docker://quay.io/centos7/mariadb-103-centos7
    - use docker:// just like http://, even if the registries are redhat
  + review the usage line
* any environmental variables must come before the image name
* be careful with mariadb/mysql images
  + envs can be the same or different
* must set up ports at creation, you can’t add them later
* rootless containers can only map to ports higher than 1024
* he works as student and runs a container
  + errors because the port is a privileged port
  + configure port forwarding
    - -p 8080:80
  + firewalld --add-port=80/tcp --permanent
  + firewall-cmd --reload

## Build Image from containerfiles

* sudo dnf provieds \*/Containerfiles
  + why do you need \*/ ???
* sudo dnf install buildah-tests
  + installing this provides example Containerfiles
* sudo find / -name “Containerfile” -exec ls -l {} \;
* gives you sizes of the files
* look at the biggest one
  + He has example container files in his repo
  + I don’t think the exam covers writing one, just using one

## Persistent Storage / Bind Mounting & weird 27:27 thing

* Ghori makes no mention of the 27:27 thing
* -v /hostdir:/containerdir
* if a container is started by root, UIDs/GIDs on the host match UIDs / GIDs on the container
* If a container is rootless, the UID of the user running the container must be the owner of the bind mounted directory
* use podman inspect on the image to detminer which user it is
* **rootless containers are launched in a namespace**
  + **different UIDs are used inside the namespace than outside**
  + **UIDs are mapped between the namespace and the host OS**
  + **podman unshare can be used to run commands inside the container namespace**
* **To get the UID mapping**
  + **podman unshare cat /proc/self/uid\_map**
    - **this shows the container root user maps to your host ordinary UID**
    - **this is his 27:27 thing that I don’t understand**
* Additional work for rootless
  + Find the UID of the user that runs the main app
    - podman inspect <imagename>
    - if that doesn’t work do
    - podman exec <containername> grep username /etc/passwd
    - podman inspect is best cuz it shows the UID
    - podman unshare chown nn:nn <directory name>
      * where nn is the uid of the user who should be owner of the directory
      * the directory must be in the users home directory otherwise it wouldn’t be part of the name space
  + validate
    - podman unshare cat /proc/self/uid\_map
      * validated the mapped user is the owner on the host
      * ls -hal /dirname
* Example with mariadb
  + he runs a mariadb
  + then: podman exec <container name> grep mysql /etc/passwd
  + userid for mysql in running container is 27:27
  + mkdir /home/student/mydb
  + podman unshare chown nn:nn mydb
  + podman unshare cat /proc/self/uid\_map
    - hard to read
    - user ids are mapped to 100,000 - 1
    - so 27 => 100,026
    - ls -Z
    - need to handle selinux
  + Two options
    - if ownership of host dir is right use:Z to autoset the context, else
    - use container\_file\_t context
    - podman run ... -v /home/student/mydb:/var/lib/mysql:Z ...
    - validate
    - ls -Z /home/student
    - should see container\_file\_t
    - cd mydb
    - see data and mysql.sock

## starting containers as services - rootless

* create systemd user unit files with systemctl for rootless containers
* systemd user services start when a user sesh is opened and closed when the user sesh stops
* to fix this, as root
  + loginctl enable-linger linda
  + validate
    - loginctl show-user linda
* could do loginctl disable-linger linda
* create a regular user to manage all containers, in this case linda
* use podman to generate a user systemd file
* mkdir ~/.config/systemd/user
* cd ~/.config/systemd/user
* file is generated in the current directory
  + xxx
* use podman generate to create user specific unit files in ~/.config/systemd/user
  + Look at the generated file and confirm that WantedBy=default.target
    - it was every time that I looked at it
* Manage with
  + systemtcl --user daemon-reload
  + systemctl --user start myblah.service
  + systemctl --user enable myblah.service
  + systemctl --user status myblah.service
* **systemctl --user only works on console or ssh and doesn’t work in sudo or su sessions**
  + **just open a new tab and ssh in as that user**
  + **see Ghori, he does this better**

## starting containers as services - rootful

* for a root container it would be different
  + cd /etc/systemd/system
  + podman generate systemd --name myweb --files --new
  + --new creates a new container when systemd unit is started and deletes it when unit is stopped
    - you don’t have to keep the initial container around
  + without --new it looks like it’s running, but it’s not

## Sander main example

* he is student
* sudo useradd linda
* sudo passwd linda
* sudo loginctl enable-linger linda
* ssh linda@localhost
  + not sure about this, may be better to just open a new tab for linda
* mkdir -p .config/systemd/user
* cd there
* podman run -d --name mynginx -p 8081:80 nginx
* select docker.io/library/nginx/latest
* podman ps -a
* pwd
* podman generate systemd --name mynginx --files --now
* ls
  + confirm you see the file
  + vi container-mynginx.service
  + confirm WantedBy=default.target or edit it
* systemctl --user daemon-reload
* systemctl --user start container-mynginx.service
* systemctl --user enable container-mynginx.service
* systemctl --user status container-mynginx.service
* exit linda
* reboot as student
* sudo shutdown -r now
* login as student to prove that the container starts even if linda isn’t logged in
* sudo -i
* ps faux
  + look for linda
  + look for /usr/bin/conmon/see that nginx started

## container lab - confirm this is from the video - don’t think this worked for me, suspect 27:27 because ghori doesn’t use that and his works

* do it on a clean snap
* ensure you are logg in to RH
* run a mariadb container
  + rootless started by student
    - I think he is in wheel group
  + accessible at host port 3206
  + set db root pw to hermes
  + name container mydb
  + accessible bind-mount dir
    - /home/student/mariadb for host
    - /var/lib/mysql for container
  + autostarts as a user systemd unit on restart
* swap in this registry
  + quay.io/centos7/mariadb-103-centos7

## solution to container lab

* as student
  + sudo su -i
* podman login registry.redhat.io
* sudo loginctl enable-linger student
* sudo loginctl show-user student
  + confirm linger was yes
* in a new tab ssh as student
* mkdir -p /home/student/mydb
* for 27:27 thing
  + podman unshare chows 27:27 mydb
  + ls -hald mydb
    - UID and GID are now 100026
      * this is correct for his formula mentioned above
      * formula: xxx
* podman search mariadb-105
* podman run -d -p 3206:3206 --name mydb -v /home/lisa/mydb:/var/lib/mysql:Z -e MYSQL\_ROOT\_PASSWORD=hermes quay.io/centos7/mariadb-103-centos7
* podman ps -a
* ls -l mydb
  + confirm you see data and the symlink mysql.lock
* podman-logout
  + if you used the original registry
* mkdir -p .config/systemd/user
* cd there
* podman generate systemd --name mydb --files --new
  + it tells you this command is deprecated for quadlets
    - see podman-systemd.unit(5)
    - it did do the needful though
  + double check if you need --user in here somewhere
* in the newly generated file, confirm WantedBy=default.target
* **systemctl --user daemon-reload**
  + **this is where it fails**
  + **do you need to do this, could you just start, enable, and status it?**
  + **failed to connect to bus**
* systemctl --user start container-mydb.service
* systemctl --user enable container-mydb.service
* systemctl --user status container-mydb.service
* sudo -i reboot
* validation
  + don’t log in as student
  + you want to test that the container is up without student loggin in
* log in as root
  + ps faux | less
  + look for /usr/bin/conmon
  + look for student

# Containers - Ghori

## General

* I had a lot of trouble with Sander container stuff in the end
  + I never understood the 27:27 thing
* Most examples will be from Ghori
* If I figure out the mother of all container labs, I will add it, but as of 06/02/24 I can’t get it to work

## Set Up

* You only need one server
* dnf install container-tools -y
  + validate dnf list container-tools
* Check what registries are available
  + cat /etc/containers/registries.conf or ~/.config/containers/registries.conf
  + find the line that starts with unqualified, and the array behind it shows the registries
    - registry.access.redhat.com requires no log in
    - registry.redhat.io requires a log in
    - docker.io – not sure
    - for the exam, there should be a registry that is on the network listed here
  + Get info on your system
    - podman info
      * shows the registries here too
* I had issues with registry.redhat.io and creds getting wonky
  + I found images from non-login repos to swap

## Search, Examine and Download and Image

* Login
  + podman login registry.redhat.io
    - blancabasura
    - 6H9!
  + podman login registry.redhat.io -–get-login => blancabasura
* Search for mysql-80 in registry.redhat.io and don’t truncate the info
  + podman search registry.redhat.io/mysql-80 –no-trunc
* Inspect the image without downloading it
  + skopeo inspect docker://registry.redhat.io/rhel9/mysql-80
    - this is different, see Ghori notebook
    - xxx
* Download an image without installing it
  + podman pull docker://registry.redhat.io/mysql-80
  + need docker:: even if it’s redhat
  + validate
    - podman ps -a
* Now that the image is downloaded, display the details locally
  + podman inspect mysql-80

## Delete an image

* list the images
  + podman images
* podman rmi mysql-80
* Validate
  + podman images => not there

## Build an image from a container file

* Log in
  + podman login registry.redhat.io
  + blancabasura
  + 6H9!
  + validate
    - podman login registry.redhat.io –-get-login
* Create a containerfile
  + vi containerfile
    - FROM registry.redhat.io/ubi9/ubi
    - CMD echo “RHCSA exam is hand on” | wc
    - COPY ./testfile /tmp
* vi testfile
  + add rando text
* Build the image
  + podman image build -f containerfile -t ubi9-simple-image .
  + -f <container file name>
  + -t <tag – descriptor>
  + . local directory
  + validate
    - podman image ls
    - you can do podman image ls or podman images
    - output indicates localhost/ubi9-simple-image
      * localhost indicates you built it with a container file

## run a container interactively and interact with it and delete it

* podman run -it –name rhel8-base-os ubi8
  + -it is interactive
  + ubi8 is the image name
  + you don’t have to name it
* prompt changes to long string, the container id
  + pwd
  + ls
  + cat /etc/redhat-release
  + date
  + exit
  + back at host prompt
* delete the container
  + podman rm rhel8-base-os
  + validate
    - podman ps -a

## run a container and auto-remove it after running it’s command

* podman run –rm ubi7 ls
  + Use the ubi7 image and create a container that runs ls and then autodeletes
* validate auto-removal
  + podman ps -a

## container with port mapping for httpd

* search for an httpd image in registry.redhat.io
  + podman search registry.redhat.io/rhel7/httpd
  + log in
    - podman login registry.redhat.io
* Download the latest version
  + podman pull registry.redhat.io/rhscl/httpd-24-rhel7
    - validate
      * podman images
* podman run -dp 10000:8000 –name rhel7-port-map httpd-24-rhel7
  + -d is daemon style
  + -p is ports
  + host:container
* validate
  + podman ps -a
  + podman port rhel7-port-map

## start, restart and remove a container

* podman ps -a
  + pick a container that is UP and running
  + podman stop rhel7-port-map
  + validate ps -a
    - Will show Exited (0)
* start the container again
  + podman start rhel7-port-map
  + validate
  + podman ps -a
    - should show Up
* stop again, and then remove
  + podman stop rhel7-port-map
  + podman rm rhel7-port-map
* validate deletion
  + podman ps -a

## set environmental variables

* pass each variable in with it’s own -e
* podman run -it -e HISTSIZE -e SECRET=”secret123” –name rhel9-env-vars ubi9
  + don’t have to have a value
* prompt changes to long string
  + echo $HISTSIZE $SECRET
    - 1000 secret123
* Because it’s an interactive container, disconnect with:
  + exit
* stop the container
  + podman stop rhel9-env-vars
* remove the container
  + podman rm rhel9-env-vars

## Configure Persistent Storage

* mkdir /host\_data
* chmod 777 /host\_data
* podman run –name rhel9-persistent-data -v /host\_data:/container\_data:Z -it ubi9
* prompt changes to long string
* validate the directory inside the container
  + ls -ldZ /container\_data
  + vi testfile
  + This is persistent storage
  + Validate creation of file
    - ls -ldz /container\_data
* exit the container
* Confirm you see the testfile outside of the container
  + ls -lZ /host\_data
* Stop and remove the container
  + podman stop rhel9-persistent-data
  + podman rm rhel9-persistent-data
* Create a new container, and attach to the same host storage
  + podman run -it –name rhel8-persistent-data -v /host\_data:/container\_data2:Z ubi8
  + the name for the data in the container is incremented to 2
* prompt changes to long string
* ls -ldZ /container\_data2
  + confirm you see testfile
  + create testfile2
    - This is persistent storage 2
  + ls -lZ /container\_data2
* exit the container
* Confirm you see testfile and testfile2 outside of the container
  + ls -lZ /host\_data
* Stop and remove the container
  + podman stop rhel8-persistent-data
  + podman rm rhel8-persistent-data
* Confirm you see the testfile and testfile2 outside of the container
  + ls -lZ /host\_data

## configure a rootful container as a systemd service

## General

* rootful and rootless are different
* config file = /etc/systemd/system/xxx
* no need to use podman start / podman stop with systemd

## configuration

* podman run -dt –name rootfule-container ubi9
* validate it was created
  + podman ps -a
* generate a service unit file named /etc/systemd/system/rootful-container.service
  + use the –new param to ensure the container is launched and won’t require the source container to work
* podman generate systemd --new --name rootful-container | sudo tee /etc/systemd/system/rootful-container.service
  + ensure WantedBy=default.target
* stop and delete the original container
  + before deleting note the container id
  + podman stop rootful-container
  + podman rm rootful-container
* systemctl daemon-reload
* systemctl start rootful-container
* systemctl enable rootful-container
* systemctl status rootful-container
* Because you started the service, a new container should have been launched
  + validate
  + podman ps -a
  + note the new container id
* reboot
  + confirm there is a container with yet another container id

## configure a rootless container as a systemd service

## General

* config file = ~/.config/systemd/user
* no need to use podman start / podman stop with systemd
* elevated privleges are not needed for the most part
* confirmed 27 is the default uid for mariadb
* probably need to swap out any example that uses registry.redhat.io
  + it works and then it doesn’t work
  + better to practice with a repo that doesn’t need log in

## configuration

* as root
  + make a user just for this
    - useradd conuser1
  + passwd conuser1
    - hermes
* open a new putty tab and login as conuser1
  + mkdir -p ~/.config/systemd/user
* Create a new container
  + podman run -dt --name rootless-container ubi8
  + confirm it was created
    - podman ps -a
  + Create a service unit file ~/.config/systemd/user/rootless-container.service
  + podman generate systemd --new --name rootless-container > ~/.config/systemd/user/rootless-container.service
    - this used tee with the rootful one
  + cat ~/.config/sysemd/user/rootless-container.service
    - confirm WantedBy=default.target
* stop and remove the original container
  + podman stop rootless-container
  + podman rm rootless-container
  + note the container id
  + Confirm deletion
    - podman ps -a
* systemctl --user daemon-reload
* systemctl --user start rootless-container
* systemctl --user enable rootless-container
* systemctl --user status rootless-container
* starting the service should have created a new container
  + compare the container ids
    - podman ps -a
* Enable the container to start and stop with out the user being logged in
  + as root
  + loginctl enable-linger
  + loginctl show-user conuser1
* Reboot the container
  + Confirm that it starts a container with another new user id

# Lessons 0 - Sander Second Video

* nothing

# Lessons 1 - Sander Second Video

* nothing

# Lessons 2 - Prepare VM for Labs - Sander Second Video

* VM
* Server with GUI
* Default Installation
* No access to any repos
  + remove them if necessary
* Use VMWare and increase the size of the primary disk by 10 GiB
  + do this for free disk space
* Add a second disk with 20 GiB
* Add student as the admin user
  + set password to password
* He doesn't say, but I'm doing
  + 20 GiB
    - VMWare does 20 GB by default
    - It is doing sda by default
  + 2 cpu
  + 2 GB ram
  + BIOS
  + 10,000 ms
  + myserver.example.com
* The VM is named saunderlabs2 in vmware
  + once it's built snapshot it
    - right click, snapshot, take snapshot
  + When you are ready to blow it out and go again
    - right click, snapshot, manage snapshot, restore
* What this looks like
  + auto install used /dev/mapper/rhel\_myserver-root for /
  + /dev/sda1 is 1G on /boot

[student@myserver ~]$ df -h

Filesystem Size Used Avail Use% Mounted on

devtmpfs 4.0M 0 4.0M 0% /dev

tmpfs 888M 0 888M 0% /dev/shm

tmpfs 356M 7.0M 349M 2% /run

/dev/mapper/rhel\_myserver-root 17G 4.3G 13G 26% /

/dev/sda1 960M 304M 657M 32% /boot

tmpfs 178M 104K 178M 1% /run/user/1000

/dev/sr0 11G 11G 0 100% /run/media/student/RHEL-9-4-0-BaseOS-x86\_64

* what lsblk looks like

[student@myserver ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

sda 8:0 0 30G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─rhel\_myserver-root

│ 253:0 0 17G 0 lvm /

└─rhel\_myserver-swap

253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 10G 0 disk

sr0 11:0 1 10.3G 0 rom /run/media/student/RHEL-9-4-0-BaseOS-x86\_64

* what fdisk /dev/sda looks like
  + want to see what adding 10 GB looks like after I installed the OS

: p

Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors

Disk model: Virtual disk

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xf1ad5344

Device Boot Start End Sectors Size Id Type

/dev/sda1 \* 2048 2099199 2097152 1G 83 Linux

/dev/sda2 2099200 41943039 39843840 19G 8e Linux LVM

* what fdisk /dev/sdb looks like

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0xfae48fd0.

Command (m for help): p

Disk /dev/sda2: 19 GiB, 20400046080 bytes, 39843840 sectors

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0xfae48fd0

# Lessons 3 - Sander Second Video

* nothing

# Lessons 4 - Essential Tools - Sander Second Video

## Configuring Remote Repo Access

* Key elements for remote repo task
  + On the exam, all servers are offline, so by default there will be no repo access
  + On the exam, a URL to a repo will be provided
  + To configure repo access you need to add a file that ends in .repo to /etc/yum.repos.d
  + To create a skeleton repo file
    - dnf config-manager --add-repo
* Task
  + Configure your system to use the repo https://repository.example.com
  + Ensure no gpg checks will be done while accessing it
  + Ensure the client will not actually use this repo
  + Validateion
    - The repo should not show while using dnf repolist, but it's configuration should exist
* Task Solution
  + You can't use the validation script until after the next video when you configure a local repo and get git installed with his repo
  + dnf config-manager --add-repo https://repository.example.com
  + vi /etc/yum.repos.d/repository.example.com.repo
    - this was the name generated by the dnf command
    - Change enable to 0 so that no one uses the file
    - add this line
      * gpgcheck=0
      * no gpg check will take place
  + validate
    - dnf repolist
  + Score
    - run scoring script later

## Configuring Local Repo Access

* Key elements for local repo task
  + Use dd to create an iso file
  + Edit /etc/fstab and use iso9660 as the file system type
    - do not create a Systemd mount unit
  + Use dnf config-manager --add-repo or edit the file manually in /etc/yum.repos.d
  + In the baseurl, use file:// as the resource type identifier
    - xxx will have file:///
* Task
  + Make an ISO file from your install disk at /rhel9.iso
  + mount it persistently on /repo
  + configure your box to use the mounted disk as a repo
  + Verify you can install packages from the repo
* Task Solution
  + Check your virtualization software to confirm your installation iso is connected
    - He looks in the menu and sees an option to disconncect cd/dvd so he knows it's connected
    - Edit the settings in vmware
      * find cd/dvd
      * should say Datastore ISO file and connected should be checked
  + dd if=/dev/sr0 of=/rhel9.iso bs=1M
    - it would work without bs=1M, but then it would streams bytes and that would be much slower
  + df -h
    - make sure you have space, you need 20GB
  + vi /etc/fstab
  + /rhel9.iso /repo iso9660 default 0 0
  + mkdir /repo
  + mount -a
    - errors because you did dd in your home dir, so rhel9.iso is there
    - you need to move it so /etc/fstab works and you meet the requirements
  + mv rhel9.iso /
  + mount -a
  + ls -hal /repo
    - should see all the stuff
  + dnf config-manager --add-repo file:///repo/BaseOS
    - //is from the URI
    - the third / is for the file location in /
  + dnf config-manager --add-repo file:///repo/AppStream
  + Edit the repo files, because by default gpgcheck is enabled
  + vi /etc/yum.repos.d/repo\_BaseOS.repo
    - gpgcheck=0
  + vi /etc/yum.repos.d/repo\_AppStream.repo
    - gpgcheck=0
  + Validate you can use the repo
  + dnf install -y git
  + Set up git for grading with this repo
  + As the root user in /root
    - git clone https://github.com/sandervanvugt/rhcsa-labs
  + ./lab42-grade.sh
  + Run it for lab41 since you couldn't do that before without the repo
    - ./lab41-grade.sh

## Finding Files

* Key Elements
  + To perform a command on the result set of a find command, add
    - -exec blahcommand {} \; to the end of the command
    - the empty curlies symbolize the results of the find command
  + pipe grep to awk and then print the field of a returned line
* Task
  + Find all files bigger than 100 MiB
  + Write a long listing of them to /tmp/big files
* Task Solution
  + find / -size +100M -type f -exec ls -hal {}\;
    - run it like this first to make sure it looks good
    - there are some errors, but they will be handled by the redirect
    - -type f
      * files only
  + find / -size +100M -type f -exec ls -l {}\; > /tmp/bigfiles
  + validate
    - ls -hal /tmp/bigfiles
  + run the scoring script

## Managing Permissions

* Key elements
  + Basic perm is checked
    - the check will exit on match
    - if a user is user owner, group perm and other perm are not checked
* Task
  + The ask was confusing to me
  + Create the directory /data/profs
  + Create the group profs
  + Create the user linda
  + Configure perm so linda
    - can't read or write files in /data/profs
    - can change perm on the dir /data/profs
  + members of profs should be able to read and write files in /data/profs
  + nobody else should have access to /data/profs
* Task Solution
  + mkdir -p /data/profs
    - verify current settings
      * ls -ld /data/profs
      * root:root
  + groupadd profs
  + useradd linda -G profs
    - puts her in the profs group
  + id
    - verify groups
  + Linda perm
    - chown linda:profs /data/profs
    - ls -ld data/profs
    - linda:profs for user and group
  + chmod 070 /data/profs
  + ls -ld /data/profs
  + Grad your script

# Lesson 5 - Creating a Shell Script - Sander Second Video

## Looping Constructs

* Key elements for Looping task
  + for statement ise used to evaluate a range of items
  + It's good for evaluating multiple files, arguments, usernames, etc.
  + Remove a part from a string by using pattern matching operators
    - ${name##\*/}
      * removes the longest match of the pattern \*/ from the left side of the value of the variable
    - ${name#\*/}
      * removes the shortest match of the pattern \*/ from the left side
    - ${name%/\*}
      * removes the shortest match of the pattern /\* from the right side
    - ${name%%/\*}
      * removes the longest match of the pattern /\* from the right side
* Create a range
  + put the items in between {}
* Task
  + Write a script named /root/lab51.sh
  + Make a backup copy of files
    - Prompt for the directory in which files need to be backed up
    - Back up all files that have the extension .txt only
    - The backup should replaced the txt extension with bak
  + To test the script
    - create files testfile{1..9} in /tmp
* Task Solution
  + vi /root/lab51.sh
  + Use ${DIR} instead of $DIR because it's easier to read next to stuff
  + complete the for loop before filling in the body
    - for
    - do
    - done
* SHORTNAME=${i%.txt}
  + this makes testfile1.txt into testfile1
* Below he is just echoing to confirm the naming, not actually doing it

#!/bin/bash

echo what directory?

read DIR

for i in ${DIR}/\*.txt

do

SHORTNAME=${i%.txt}

echo The short name is $SHORTNAME

echo mv $i ${SHORTNAME}.bak

done

* then
* touch /tmp/testfile{1..9}.txt
* ls -hal /tmp/tst
* chmod +x /root/lab51.sh
* /root/lab1.sh
* It prompts for the directory, use /tmp
* It outputs 9 files
* now that you know it's good, go edit again for phase 2
* Rewrite it like this

#!/bin/bash

echo what directory?

read DIR

for i in ${DIR}/\*.txt

do

SHORTNAME=${i%.txt}

mv $i ${SHORTNAME}.bak

done

* Run it again
  + /root/lab51.sh
* Run the grading script
  + ./lab51-grade.sh

## Conditionally Executing Code

* Key Elements
  + - if ... then ... else ... fi
  + $1 one cli arg
  + $@ multiple cli args with a while loop
  + use read to prompt for input and store it in a variable
  + Task
    - Write a script named /root/lab52.sh
      * check is a user exists
      * it is exists, print the login shell for the user
      * if the user doesn't exist, print a message that it doesn't exist
    - Ensure that scripts allows you to provide the user name as a cli argument
    - if no cli arg is provided, the script should prompt for it
    - The script should work on multiple users if provided
* Task Solution

#!/bin/bash

if [ -z $1 ]

then

echo Provide one or more user names

read USERS

else

USERS=$@

fi

echo print each user

for i in $USERS

do

grep $i /etc/passwd &> /dev/null && echo user $i uses the shell ${grep $i /etc/passwd | awk -f : '{print $7}'; || echo the user $i doesn't exist

done

Use checker script

* ./lab520grade.sh

# Lesson 6 - systemd - Sander Second Video

## systemd

* Key Elements
  + systemd is configured with directives
  + to see all the possible directives do
    - systemctl show
  + man systemd.directives
  + systemctl uses tab completion
  + systemctl subcommand --help
* Task
  + install httpd and nginx
  + configure systemd so nginx can never be started
  + httpd should auto start at boot
  + if httpd should stop, it needs to auto restart after 21 seconds
* Task Solution
  + dnf install -y httpd nginx
  + systemctl status httpd
  + systemctl status nginx
  + both httpd and nginx want to listen on port 80
  + systemctl mask nginx
  + validate
    - systemctl start nginx
      * should say it's masked
  + systemctl enable httpd
  + Here is a good example for restart on failure
    - systemctl cat sshd.service
      * copy the two lines re restart on failure
  + systemctl edit httpd.service!!!
    - it's nano by default
    - Create this section
      * [Service]
      * Restart on failure - can't read the font
      * RestartSec=42s
  + systemctl status httpd
  + systemctl cat httpd.service
  + systemctl start httpd
  + systemctl status httpd
    - 35169 is the pid
  + kill -9 35169
  + systemctl status httpd
    - activating auto restart
  + run the check script from rhcsa-labs
    - ./lab61-grade.sh

## tuning

* Key Elements
  + the kernal can be tweaked by modifying params in /proc/sys
  + to make persistent modifications, changed params should be included in /etc/sysctl.conf or /etc/sysctl.conf.d
  + use tuned service and tuned-adm
* Task
  + Configure your system for optimal power usage efficiency
* Task Solution
  + systemctl status tuned
    - there is a bug where it could say enabled and dead
    - just start it
      * systemctl start tuned
  + systemctlctl status tuned
    - should be good now
  + tuned-adm tab tab
  + see all the profiles
    - tuned-adm list
  + tuned-adm profile powersave
  + check the current active profile
    - tuned-adm list
    - powersave
* from rhcsa-labs run the grading script
  + ./lab62-grade.sh

# Lesson 7 - Configuring Local Storage - Sander Second Video

## Managing MBR Partitions

* Key Elements
  + mbr = master boot record
  + place for 4 partitions, if more are needed the fourth should be an extended part
  + within the extended part, create logical parts
  + after formatting mount the partition
  + when mounting use a uuid
  + to see labels and uuids
    - blkid
* Task
  + On the primary disk, add an extended partition that includes all remaining disk space
  + Within this extended part, create a 1GiB logical part
  + Format the part with ext4
  + mount the part persistently on /mnt/data using the label EXTFILES
* Task Solution
  + fdisk /dev/sda
  + p to print
  + n for new
  + e for extended
  + it suggests partition 3, enter
  + since you are using the remaining space
    - first sector enter
    - second sector enter
  + p to print
  + n for new
  + adding logical partition number 5
  + first sector enter
  + second sector +1G
  + p to print
  + w to write
  + mkfs.ext4 -L EXTFILES /dev/sda5
  + mkdir /mnt/data
  + vi /etc/fstab
  + LABEL=EXTFILES /mnt/data ext4 defaults 0 0
  + validate
    - mount -a
  + For the exam, consider rebooting any time you mount something
  + reboot
  + lsblk
  + sudo -i
  + cd rhcsa/labs
    - ./lab71-grade.sh

## Managing GPT Partitions

* Key Elements
  + introduced in 2010
  + can create 128 parts
  + no differences between primary, extended and logical
  + vfat can be mounted on linux, windows and mac
    - it doesn't support labels
  + use blkid to find uuids
* Task
  + create a 2 GiB gpt partition on your second disk
  + Format it with vfat
  + mount it on /mnt/files using it's uuid
* Task Solution
  + fdisk /dev/sdb
  + m for help
  + by default fdisk creates mbr partitions, if you want gpt you need to use g to create a label for a gpt partition !!!
  + g for gpt before doing n for new!!!
  + n for new
  + if it prompts for a number between 1 and 128 you know you have a gpt part
  + partition 1
  + accept first sector with enter
  + second sectore +2G
  + w
  + validate
    - lsblk
  + mkfs.vfat /dev/sdb1
  + blkid
    - find the uuid and copy it
  + vi /etc/fstab
    - UUID="paste string" /mnt/files vfat defaults 0 0
  + mkdir /mnt/files
  + mount -a
  + in rhcsa-labs
    - ./lab72-grade.sh

## Managing LVM

* Key elements
  + LVM logical volumes are allocated from a volume group
  + the VG is composed of one or more physical volumes
  + if a partition is used as a physical volume it should be marked with the lvm partition type
  + while creating a volume group, the physical extent size is used to specify the minimal allocation unit
  + each volume group uses one physical extent to store metadata
* Task
  + On the second disk, add a 2 GiB partition to be used for creating LVMs
  + Create a volume group named vglabs and add the partition to it
  + In the volume group, create a logical volume named lvlabs
    - it should use half of the available disk space
  + format the LV with xfs
* Task Solution
  + fdisk /dev/sdb
  + n for new
  + accept first sectore
  + for second sector +2G
  + t for type
    - 2
    - lvm
  + p to print
  + w to write
  + vgcreate vglabs /dev/sdb2
    - vgcreate --help
    - man vgcreate
      * has good examples
  + vgs
    - confirm name and free space
  + lvcreate -n lvlabs -l 50%FREE vglabs
  + lvs
  + vgs
  + mkfs.xfs /dev/vglab/lvlabs
  + it didn't say make it persist so he didn't add it to /etc/fstab
  + Grade it
  + cd rhcsa-labs
  + ./lab73-grade.sh

## Managing Swap

* Key Elements
  + swap can be allocated on a block device or on a swap file
  + after creating an empty file with dd, it can be treated like any other swap device
  + Add it to /etc/fstab to persist
* Task
  + Create a 1GiB swap file named /swapfile
  + Mount it persistently
* Task Solution
  + dd if=/dev/zero of=/swapfile bs=1M count=1024
    - count=1024 for a 1GB file
  + format it
    - mkswap /swapfile
    - it gives you a message that you have insecure perm on the file
    - don't fix this, see what the grading script does
  + vi /etc/fstab
    - there is already a swap line in there but you can add yours
    - /swapfile none swap defaults 0 0
      * used to use swap instead of none
  + free -m
  + swapon -a
    - complains re insecure permissions but it still works
  + cd rhcsa-labs
  + ./lab74-grade.sh
    - fails you for wrong perm
  + fix it
    - chmod 600 /swapfile
  + ./lab74-grade.sh
    - this time it passes

# Lesson 8 - Managing File Systems - Sander Second Video

## Mounting Filesystems

* Key Elements
  + options are available for mounting file systems
    - some are file system specific, some are generic
    - man mount
* Task
  + mount the lvlabs LVM created in lesson 7 persistently on /lvlabs so that no executable files can be started from it
  + ensure the file access time is not updated while files are accessed
* Task Solution
  + mkdir /lvlabs
  + vi /etc/fstab
    - /dev/vglabs/lvlabs /lvlabs xfs noexec,noatime 0 0
  + mount -a
    - if you use the wrong file system type, it will complain with an error message
  + cd rhcsa-labs
  + ./lab81-grade.sh
    - he edited the script, I hope it works

## Managing autofs

* Key Elements
  + His example is not an nfs example, this is rare
  + autofs is a service that mounts file systems on demand
  + It's mostly used for nfs shares, but it can be configured for local device mounting (rarely used)
  + man autofs
  + review /etc/auto.misc
    - it has good examples!!!
  + /etc/auto.master is the main file and it refers to /etc/auto.misc for additional configuration
* Task
  + Configure autofs so that /dev/sda1 is mounted on /start/files when the directory is accessed
* Task Solution
  + vi /etc/auto.master
    - empty
    - you need to install it
      * dnf install -y autofs
  + now there should be something in the file
  + vi /etc/auto.master
    - 5 lines down from the top, add this below /misc
      * /start /etc/auto.start
      * you don't have to make the directory, autofs will do it for you
  + vi /etc/auto.misc
    - copy the boot line
  + vi /etc/auto.start
    - paste the boot line
    - remove #boot and type files
      * this is the relative directory that will be created in the /start directory
      * edit fstype to be xfs
      * edit the device to be sda1
  + ls /
    - there should be no /start directory
  + systemctl enable --now autofs
  + ls /
    - you should see /start now that autofs is running
    - there is also a new misc directory and a couple other directories related to autofs
  + cd /start
  + ls
    - there is nothing there
  + if you then cd into files you see stuff
    - it's the content of sda1
  + mount
    - look at the bottom
  + cd /rhcsa-labs
    - ./lab82-grade.sh

## Resizing LVM Volumes

* Key Elements
  + Free extents must be available in the volume group if you want to grow the size of a logical volume
  + If no free extents are available, additional extents can be allocated by adding physical volumes to the volume group
  + always use -r to resize the filesystem as well when resizing the logical volume
    - you could damage the file system this way
* Task
  + Add 10GiB to the logical volume that the root file system is mounted on
  + Configure any additional required device up to your discretion
* Task Solution
  + vgs
    - shows rhel
    - this is where the root file system is
  + start by creating a pv and add it to the vg
  + vgs
  + fdisk /dev/sdb
  + n for new
  + it will be partition #3
  + accept first sector with enter
  + second sectore +10G
  + t for type
  + 3 for partition number 3
  + lvm
  + w to write
  + lvs
    - shows rhel volume group
    - this is the root file system
  + vgextend rhel /dev/sdb3
  + vgs
    - now see 10GB available
  + lvextend -r -L +10G /dev/rhel/root
    - errors out, not enough extents
    - he says for the exam, it will be worded that it's ok to be missing / over by just a little bit!!!
  + redo it this way to get around the error
  + lvextend -r +2559 /dev/rhel/root
    - + means to add it to it, not make it the exact size
  + lvs
  + df -h
    - root should be 27G
  + cd rhcsa-labs
  + lab83-grade.sh

## Configuring Directories for Collaboration

* Key Elements
  + collaboration always means use special permissions
  + the set-group ID, if set on a directory, ensure that all files created in that directory will be group owned by the group owner of that directory
  + sticky bit, if set on a directory, ensures that only the user that is the owner of the file is allowed to delete the file
* Task
  + Create a group sales with users lisa and lori as members
  + Ensure the group sales has full access to /data/sales
  + All files created in /data/sales should be group owned by the group sales
  + Ensure that files can only be deleted by the user who created the file, as well as user lisa who is a member of group sales
* Task Solution
  + always create the groups before the users
  + groupadd sales
  + useradd lisa -G sales
  + useradd lori -G sales
  + mkdir -p /sales/data
  + chgrp sales /data/sales
  + need set gid
    - chmod g+s /data/sales
  + need sticky bit too
    - chmod +t /data/sales
  + chmod g+w /data/sales
  + chown lisa /data/sales
  + cd rhcsa-labs
  + ./lab84-grade.sh

# Lesson 9 - Deploying, Configuring and Maintaining Systems - Sander Second Video

## Scheduling Tasks

* Key Elements
  + use cron and logger
  + Neither cron nor systemd timers are connected to stdout so commands like echo don't work!!!
* Task
  + Create a schedule tasks that runs as user linda
    - weekdays at 2am
    - it should write greetings from linda to the logging service
* Task Solution
  + crontab -e -u linda
  + 0 2 \* \* 1-5 logger greetings from linda
  + it's good to do this until you know it works
  + \* \* \* \* \* logger testing
    - writes once a minute
  + validate
    - journalctl
    - should see testing at the bottom so you know that basic functionality with logger works
    - remove the testing line at the exam
  + cd rhcsa-lab
    - ./lab91-grade.sh

## Configuring Time Services

* Key Elements
  + While booting, Linux obtains its time from the hardware clock, and then sets the system clock
  + Don't use systemd-timesyncd or ntpd
    - they are not supported on RHEL
    - use chronyd
* Task
  + Configure the server to fetch time from pool.ntp.org
  + set the timezone to Africa/Lusaka
* Task Solution
  + timedatectl tab tab
  + timedatectl show
  + timedatectl set-timezone Africa/Lusaka
  + validate
    - timedatectl show
  + vi /etc/crony.conf
    - Leave the pool line that is there and add this one
      * pool pool.ntp.org
      * I think I would want to comment that other one out, hope it's not that ambiguous
  + systemctl restart chronyd
  + cd rhcsa-lab
    - ./lab92-grade.sh

# Lesson 10 - Basic Management - Sander Second Video

## Restricting Network Access

* Key Elements
  + to run a service it must be installed
  + if you aren't sure of the name of a service do
    - dnf provides \*/servicename!!!
      * exactly that? or do you insert a name in there?
  + start and enable the service
  + configure the firewall
    - firewalld
    - configure it with firewall-cmd
  + firewalld is not systemd, so the services are separate and can be named differently
* Task
  + Configure your box so httpd is accessible for external users on port 80 and 443 using it's default config
* Task Solution
  + note he never considered doing port stuff for the firewall, just service stuff
  + there was no need to define external anywhere
  + dnf install -y httpd
  + firewall-cmd --list-all
  + firewall-cmd --get-services
    - shows all available services
  + firewall-cmd --add-service http
  + firewall-cmd --add-service https
  + doing the above 2 commands only added it to runtime, make it persistent now
    - firewall-cmd --add-service http --permanent
    - firewall-cmd --add-service https --permanent
  + firewall-cmd --list-all
  + to be safe
    - systemctl restart firewalld
    - firewall-cmd --list-all
    - it's still there even after a restart
    - cd rhcsa-labs
    - ./lab101-grade.sh

## Configuring hostname resolution

* Key Elements
  + hostname is a kernel parameter set in /etc/hostname
    - best to use hostnamectl to set it
  + to configure a dns client, /etc/resolv.conf is used
  + if you are in a dhcp environment you get the resolver as well
  + if no dns is available, use /etc/hosts
  + /etc/nsswitch.conf is used to configure the order in which these are used
* Task
  + Set your hostname to examlabs.local
  + ensure this name is resolved to the IP address your host is using
* Task Solution
  + hostnamectl set-hostname examlabs.local
  + open a new terminal
  + should see student@examlabs at the prompt now
  + cat /etc/hostname
    - should say examlabs.local
  + open a root shell
    - sudo -i
  + ip a
  + copy the ip address
  + vi /etc/hosts
    - 192.168.29.149 examlabs.local
    - validate
      * ping examlabs.local
  + if ping didn't work, review /etc/nsswitch.com
    - find the hosts: line
    - it's set to files, dns, myhostname
    - files checks /etc/hosts
    - myhostname is your hostname
  + cd rhcsa-labs
    - ./lab102-grade.sh

# Lesson 11 - Users and Groups

## Managing user accounts and groups

* Key Elements
  + Default settings for creating users are in /etc/login.defs
* Task
  + Create users sarah and zeina
    - make them members of staff as a secondary group
  + Create users bilal and bob
    - make them members of the group visitors as a secondary group
  + set the password expiration for these users only to 90 days, do not change the standard password expiration
* Task Solution
  + groupadd staff
  + groupadd visitors
  + useradd -G staff zeina
  + useradd -G staff sarah
  + useradd -G visitors bilal
  + useradd -G visitor bob
  + You only want to change the expiration to 90 days for these 4 guys
  + Go edit their lines in /etc/shadow
    - take ou 99999 and set it to 90
    - vi will complain, but do :wq! and it will write it
    - It would have probably been better to use passwd
      * passwd --help
      * -x for maximum password lifetime
      * passwd -x 90 zeina...would do the same thing
    - validate
      * cat /etc/shadow
  + cd rhcsa-labs
  + ./lab111-grade.sh

## managing passwords

* Key Elements
* Task
  + disable login for sarah without deleting her account
  + set bob to expire on 01/01/32
  + ensure the default password hashing algorithm for new users is set to SHA256 (no need to change existing users)
  + enforce a maximum password validity of 120 days for new users (no need to change for existing users)
* Task Solution
  + passwd --help
  + -l will lock an account
  + passwd -l sarah
  + the expire date for bob
    - passwd --help
    - probably --expire, but check the man page
    - man passwd
    - this option is actually used to force the user to set a new password at the next login
    - you don't want the password to expire, you want the account to be expired
    - chage has a better option
    - chage --help
    - -E expire date
    - always consider the date format in linux
    - man chage
    - YYYY-MM-DD
    - chage -E 2032-01-01 bob
    - validate
      * chage -l bob
      * don't validate in /etc/shadow because it's listing the number of days since 01/01/70 that the password will expire
      * fuk that
    - vi /etc/login.defs
      * So, login.defs only affects new users, not old ones!!!
      * Edit PASS\_MAX\_DAYS from 99999 to 120
      * search for SHA
      * Edit ENCRYPT\_METHOD to SHA256
  + cd /rhcsa-labs
    - ./lab112-grade.sh

## Configuring super user access

* Key Elements
  + visudo
    - it opens /etc/sudoers
      * there are good examples there
* Task
  + Configure user sarah so she can perform any tasks with elevated sudo privileges
  + Configure user zeina so she can manage user passwords, but not for root
* Task Solution
  + usermod -aG wheel sarah
    - I think I had trouble with -a, is it good now?
    - -a appends the group wheel to the current secondary membership for Sarah!!!
      * without it, it overwrites the secondary groups that are there already
    - id sarah
    - THERE WAS NOTHING TO DO IN /ETC/SUDOERS FOR THIS, JUST -aG wheel!!!
  + visudo
    - copy the example
    - zeina ALL=/usr/bin/passwd, ! /usr/bin/opasswd root
    - confirm it's space]
    - validate
      * set a password for zeina so you can test
      * passwd zeina
      * => password
      * su - zeina
      * sudo passwd root
      * put in zeina's password
      * get correct error message that you are denied
      * test that she can set a password for another user
      * sudo passwd bob
      * change it
      * cd /rhcsa/labs
      * ./lab113-grade.sh
  + it's interesting that to test this you had to set some passwords
    - you never unset them
    - but it doesn't mention it, so I guess testing artifacts are ok?

# Managing Security - Sander Second Video

## Permissions

* Key Elements
  + UGO
    - Linux moves from user to group to other
    - When it finds a match, it looks no further
  + special perm
    - set uid
    - set gid
      * all files inherit the group owner from the directory
    - sticky bit
  + attributes
    - used to prefent file operations regardless of perm
* Task
  + Configure access to /data/staff directory
    - members of the group staff have full access
    - members of the group can only delete files they created
    - the dir contains the file rootfile, which can be read by all group member, but can't be deleted by anyone
    - otherwise, leave users and perms at their defaults
* Task Solution
  + there is no set gid in this questions
  + mkdir -p /data/staff
  + chgrp staff /data/staff
  + ls -ld /data/staff
  + chmod g+w /data/staff
  + chmod +t /data/staff
  + echo rootfile > /data/staff/rootfile???
    - what does this do?
    - could you have just done touch rootfile?
    - use attributes, are attributes even on the exam?
    - man chattr
      * immutable means it can't be changed
    - chattr +i /data/staff/rootfile
      * ls -hal doesn't show attributes
      * lsattr does
      * lsattr /data/staff/rootfile
        + see i for immutable
  + cd rhcsa-labs
    - ./lab121-grade.sh

## Managing selinux

* Key Elements
  + need to run selinux in exam with enforcing mode
  + file context is used for files and directories
  + man semanage-fcontext
    - good examples
  + booleans are on/off switches to enable/disable specific functionality
  + if sealert is available use it to print interpreted messages about selinux denials
* Task
  + install vsftpd service
  + configure it so anonymous uploads are permitted
  + anonymous\_upload option must be enabled ni /etc/vsftpd/vsftpd.conf
  + perm mode 777 on /var/ftp/pub directory
  + selinux must be configured correctly
  + To test anonymous file upload
    - install lftp client
      * open a session using lftp localhost
      * cd pub
      * put /etc/hosts
* Task Solution
  + dnf install -y vsftpd lftp
  + vi /etc/vsftpd/vsftpd.con
    - tweak anonymous\_enable=YES
    - uncomment anon-upload\_enable=YES
    - the line above this mentions the relevant booleans
      * allow\_ftpd\_anon\_write
        + copy this, we will need it later
      * allow\_ftpd\_full\_access
* ls -ld /var/ftp/pub
  + it's 755
  + chmod 777 /var/ftp/pub
  + getsebool -a | grep ftp
    - shows all bools with ftp in the name
    - ftpd\_anon\_write --> off
      * when he copied this from before, there was an allow at the front???
  + setsebool -P ftpd\_anon\_write on
    - -P persists
  + systemctl restart vsftpd
  + first validate
    - lftp localhost
    - cd pub
    - put /etc/hosts
    - error message
    - journalctl | grep sealert
      * run the command it tells you to run
      * you need some selinux fcontext stuff in addition to the boolean
      * it gives you a semanage command with a relative file name at the end
        + you can't use relative files, it must be absolute
    - at the cli
      * #semanage fcontext -a -t public\_content\_rw\_t
      * it's commented out
    - man semanage-fcontext
      * look at the examples
        + find the regex
      * paste it to the end of the commented out command and edit it
        + take out /web and make it /var/ftp/pub
        + remove the #
        + run the command

it writes it to the policy

restorecon -Tv /var/ftp/pub

apply the context

output indicates it was relabeled

* + - You may need to install the package for sealert
      * dfn provides \*/sealert
      * dnf install -y setroubleshoot-server
      * if it doesn't work, check the status of auditd
      * systemctl status auditd
        + needs to be up and running
    - lftp localhost
    - cd pub
    - put /etc/hosts
    - cd rhcsa-labs
      * ./lab122-grade.sh

## Configuring ssh

* Key Elements
  + By default in RH9, password based root login is denied
  + Main config file
    - /etc/ssh/sshd\_config
* Task
  + Configure ssh so only student and linda are allowed to ssh to your host
* Task Solution
  + man sshd\_config
    - separator is spaces
  + vi /etc/ssh/sshd\_config
  + Add this line anywhere
    - AllowUsers student linda
  + systemctl restart sshd
  + ssh student@localhost
    - if you are logged in, great
  + ssh zaina@localhost
    - you can't log in
  + cd rhcsa\_lab
    - ./lab123-grade.sh

## Managing selinux port context

* Key Elements
  + When services need to be accessed on non-default ports, selinux needs to be configured to allow this
  + For some good examples
    - man semanage-port
  + Consult the sealert logs
  + Look at /var/log/audit/audit.log too
* Task
  + Configure httpd to list on 82 only
* Task Solution
  + vi /etc/httpd/conf/httpd.conf
  + Tweak Listen 80 to
    - Listen 82
  + systemctl restart httpd
    - errors out
  + systemctl status httpd
  + journalct -u httpd
    - Perm denied could not bind to ...82
  + grep AVC /var/log/audit/audit.log
    - avc denied name bind for httpd 82
      * 82 has reserved\_port\_t
    - semanage port -l | grep 80
    - http\_port\_t
  + man semanage-port
    - review examples
    - copy the httpd example, paste it at the cli and change tcp 80 to tcp 80
  + No restore con here
    - that is only for files, and this is ports!!!
  + systemctl restart httpd
  + cd rhcsa-labs
    - ./124-grade.sh

# Lesson 13 - Managing Containers - Sander Second Video

## Container Management

* Key Elements
  + rootless containers are more secure, but in some cases they can't be used
  + limitations of rootless containers
    - they don't have full access to the file system
    - they can't bind to a privileged host port
  + Each user that needs access to container registries needs ao authenticate against the registries
  + man podman-run
* Task
  + Log in to the default RH container regestries
  + start the bitnami nginx container image in detached mode
  + configure access so the container is accessible via host port 81
* Task Solution
  + Where to log in
    - grep -i unqualified /etc/containers/registries.conf
    - this lists the default registries
  + podman login registry.access.redhat.com
    - On the exam, they would have to provide a username and password for this!!!
    - username
    - password
  + Do this for all the redhat regestries
    - podman login registry.redhat.io
    - username
    - password
  + Ignore docker, that's different
  + podman search bitnami | grep nginx
    - it returns one entry, and it's docker.io
  + He pulls the image to figure out what port it's running on
    - podman pull docker.io/binami/nginx
  + Now inspect it
    - podman inspect docker.io/bitnami/nginx:latest | less
    - find exposed ports
    - it's 8080/tcp
  + You need to map the port
    - man podman-run
    - search port
    - podman run -p 81:8080 -d docker.io/bitnami/nginx:latest
  + Validate it's running
    - podman ps
  + Use the grade checker
  + ./lab131-grade.sh

## Managing Storage

* key elements
  + the container user must have access to perm to the mounted directory on the host OS
    - the best way to do this for rootless containers is to do it in the user's home directory
  + **normally the UID within the container is represented by the numeric user id + 99,999 on the host OS**
    - **podman unshare can be used to apply changes in the container namespace**
      * **it allows you to give the user id as it occurs in the container perm on the dir that the container needs access to, without that the container won't have access**
  + SELinux is simple, just add :Z to the mount
* task
  + As user linda
  + create a container named mydb
  + start the mariadb-105 image
  + Make sure it meets the following requirements
  + the container's /var/lib/mysql directory is mounted on the directory mydb in user linda's home directory
  + the container is started in the background
  + the container MYSQL\_ROOT\_PASSWORD variable is set to password
  + SELinux is operational and allowing access
  + the container can be accessed at host port 3306
* task solution
  + **ssh linda@localhost**
    - **he does this because ssh gives a real log in shell**
    - **su - doesn't give you a real log in shell**
  + **When working with containers always ssh user@localhost, otherwise some features won't work**
  + podman search mariadb-105
  + podman run -d --name mydb -v /var/lib/mysql:/home/linda/mydb:Z -e mariadb\_root\_password=password -p 3306:3306 registry.redhat.io/rhel9/mariadb-105
    - He went back and commented out a command line comment!!!
  + Things to take care of first before running the command
  + mkdir mydb
  + podman unshare chown 27:27 mydb
    - this sets the right container owner
    - you know it's 27 because
      * podman login registry.redhat.io
      * podman pull registry.redhat.io/rhel9/mariadb-105:latest
      * podman inspect registry.redhat.io/rhel9/mariadb-105:/latest | less
        + In the Config Section you see User: 27
      * podman unshare ls -ld mydb/
        + here you see 27 27
        + it's fine that there is a number and not a name here, those are users in the container
      * ls -ld mydb/
        + this is without podman unshare

it's 165562

this shows the mapped user

you don't care about the mapped user too much

* + podman run -d --name mydb -v /home/linda/mydb:/var/lib/mysql:Z -e mariadb\_root\_password=password -p 3306:3306 registry.redhat.io/rhel9/mariadb-105
    - Earlier he went back and commented out a command line comment!!!
  + man podman-run
    - list the host directory first and the container directory second
  + It should give you an id after running it
  + podman ps -a
  + podman logs mydb
    - the mariadb images, sometimes they want the mariadb password, and sometimes they want the mysql password, some times they take both
  + Remove the container and do it again
  + podman rm mydb
  + podman run -d --name mydb -v /home/linda/mydb:/var/lib/mysql:Z -e MYSQL\_ROOT\_PASSWORD=password -p 3306:3306 registry.redhat.io/rhel9/mariadb-105
  + podman ps -a
    - it's good now
  + run the exam grade script
    - open another tab and ssh in as student
    - get a root shell
      * sudo -i
    - cd rhcsa-labs
    - ./lab132-grade.sh
    - the script output doesn't say anything about that 27:27 bullshit
      * ghori never mentions that either
      * sander doesn't mention it in the book
      * would it run if you completely ignored that part
  + It's good he's doing the 27 stuff without the systemd stuff on top of it, so you know that problem isnt' with systemd

## Containerfile

* Key Elements
  + RH supports both Containerfile and Dockerfile, but prefers Containerfile
  + to build custom images, use Containerfile
  + When using podman build, use -t to specify a tag to be used as its identifier
  + Since the task doesn't say which user to use, do it as root
  + If you run a container in the foreground, and you need to move on to the next thing, what's the best way to get in the background???
* Task
  + In this GH there is the file lesson13/Containerferile
    - build a container imaged named helloworld:1.0 using this container file
    - start the image once
      * the name is actually the tag
* Task Solution
  + cd rhcsa-labs
  + the containerfile is there
  + vi containerfile
    - does a sleep thing with the ubi image
  + build the image
    - podman build -t helloworld1.0 .
      * dot means to use the container file in the current directory
  + podman images
    - it's there
  + podman run -d helloworld:1.0
  + podman ps -a
    - shows it
  + grade the lab
    - cd rhcsa-labs
    - ./lab133-grade.sh

## systemd integration for containers

* Key Elements
  + Generate systemd unit files from running containers
    - podman generate
    - man podman-generate-systemd
    - while using podman generate, make sure you are in ~/.config/systemd/user
  + to start the container at boot without the user being logged in, enable linger for the user with loginctl linger
* Task
  + create a systemd unit file that starts mydb created earlier as user linda
  + the container should auto start when the system boots and not depend on user login
* Task Solution
  + from a root shell
  + loginctl enable-linger linda
  + loginctl show-user linda
    - confirm linger = yes
  + log out and log in as linda again to pick up this change
    - ssh linda@localhost
  + podman ps -a
  + be in linda's home dir
    - mkdir -p .config/systemd/user
    - cd .config/systemd/user
    - man podman-geberate-systemd
      * too much info
    - podman geneate systemd --help
  + podman generate systemd --new --files mydb
    - it creates a file with a long string
    - vi the file
  + systemctl --user enable container-longstring
  + validate
    - reboot
  + don't log in as linda now, log in as student
  + open a root shell
    - sudo -i
  + ps aux | grep linda
    - find the conmon line
      * container monitor
      * it shows the container's long string, the id
  + check your grade
  + cd rhcsa-labs
  + ./lab134-grade.sh

# Lesson 14 - Practice Exam - Sander Second Video

## Prepare the Practice Exam Environment

* Create a vm
  + server with gui
  + 4 GiB RAM
  + 2 cpu
  + 20 GiB disk that is /dev/sda, not nvme
  + do dnot register for access to RH repos
  + add an extra hard disk with 20 GiB
  + ensure access to the RHEL 9 iso
  + create user student with password of password with full sudo access
* He's using vmware fusion
  + click the configure button
  + add device, new hard disk
  + advanced options
    - disk type
    - default is nvme
    - change to scsi
  + podman run -d --name mydb -v /home/linda/mydb:/var/lib/mysql:Z -e mariadb\_root\_password=password -p 3306:3306 registry.redhat.io/rhel9/mariadb-105

## Exam Tasks

![A close up of a sign

Description automatically generated

A close-up of a text

Description automatically generated

A screen shot of a computer script

Description automatically generated

A white text on a blue background

Description automatically generated

A white box with black text

Description automatically generated

A close-up of a computer program

Description automatically generated

A white and blue text on a white background

Description automatically generated

A close-up of a computer program

Description automatically generated

A close-up of a sign

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A close-up of a text

Description automatically generated

A close-up of a document

Description automatically generated

A close-up of a text

Description automatically generated

* examgrade.sh from git repo will grade the xam
* 2.5 hours

# Practice Exam Solutions

## Task 1

A close up of a sign

Description automatically generated

* Work from a root shell
* Make sure your cdrom disk is connected
* dd -if=/dev/sr0 of=/rhel9.iso bs=1M
* mkdir /repo
* vi /etc/fstab
  + /rhel9.iso /repo iso9660 defaults 0 0
* mount -a
* cd /repo
* ls
  + confirm you see the stuff
* dnf config-manager --add-repo file:///repo/BaseOS
* dnf config-manager --add-repo file:///repo/AppStream
* cd /etc/yum.repos.d
* vi repo.BaseOS.repo
  + at the bottom add
    - gpgcheck=0
* vi repo.AppStream.repo
  + at the bottom add
    - gpgcheck=0
* dnf repolist
* dnf install -y nmap
  + he likes this for his tests
* Questions:
  + if the exam wants you to not use the CD, but a network location, how would that be different?

## Task 2

A close-up of a text

Description automatically generated

* find / -size +200M -type f
  + run this part first to make sure it looks good
* find / -size +200M -type f -exec ls -l {}\;
  + run this much and make sure it looks good
* find / -size +200M -type f -exec ls -l {}\; > /root/bigfiles.txt
  + cat /root/bigfiles.txt

## Task 3

A screen shot of a computer script

Description automatically generated

* name the file and save it first so you have the benefit of syntax highlighting

vi countdown.sh

#!/bin/bash

# if you don't have an argument

if [ -z $1 ]

then

echo Please provide the number of minutes

read COUNTER

else

COUNT=$1

fi

# Debug line

echo Counter is set to #COUNTER

COUNTER=$(( COUNTER \* 60 ))

while [ $COUNTER -gt 0 ]

do

echo $COUNTER seconds remaining

COUNTER=$(( COUNTER - 1 ))

sleep 1

done

echo countdown is now finished

* test it
* chmod +x countdown.sh
* ./coundown.sh
* enter 1 for minutes
  + it starts to count down from 60
* now test it by passing in an arg of 3 seconds
  + ./coundown.sh 3
* remove the debug line
* copy the script to /usr/local/bin
  + any scripts that don't come from packages should go here
  + cp countdown.sh /usr/local/bin

## Task 4

A white text on a blue background

Description automatically generated

* put it in either
  + .bash\_profile
  + .bashrc
* use a clean log in shell for this
  + ssh student@localhost
  + su -
    - this creates a subshell, but you want a login shell
* nice --help | less
* nice -n 19 sleep infinity
* ctrl-c
* vi /home/student/.bash\_profile
  + Add this to the bottom
    - nice -n 19 sleep infinity &
* exit
* ssh student@localhost
* ps aux | grep sleep
  + Confirm you see the sleep command

## Task 5

A white box with black text

Description automatically generated

* Have a root shell
* lsblk
  + Confirm you see /dev/sdb
* fisk /dev/sdb
* m for help
* First partition
  + g for gpt
  + n for new
  + enter to accept first sector
  + last sector = +2G
* Need another partition
  + n for new
  + enter to accept first sector
  + last sectore +5G
  + t for type = lvm
  + p to print and review
  + w to write

## Task 6

A close-up of a computer program

Description automatically generated

* vgcreate --help | less
* vgcreate -s 2M vglabs /dev/sdb2
* vgdisplay vglabs
  + confirm PE size of 2M
* lvcreate -n lvlabs -l 50%FREE vglabs
* vgs to validate
* lvs to validate

## Task 7

A white and blue text on a white background

Description automatically generated

* The way it's written you need to analyze and confirm in your own mind it can only be sdb1
* mkfs.xfs /dev/sdb1
* blkid
  + copy UUID
* vi /etc/fstab
  + UUID="<paste uuid here>" /data xfs defaults 0 0
* mkdir /data
* mount -a
* mkdir /files
* mkfs.ext4 /dev/lvlabs
* it doesn't make sense to use uuid for LVM, just use the LVM name
  + /dev/vglabs/lvlabs /files ext4 defaults 0 0
    - you could use this name or the dev mapper name, both are symlinks
* mount -a
* systemctl tab tab
* systemctl list-unit-files -t mount
* systemctl cat tmp.mount
* systemctl enable tmp.mount
  + there is stuff in /tmp
  + so don't use --now
  + just reboot instead
* reboot to confirm mounts are good
* open a root shell
  + su -i???
* lsblk
* systemctl status tmp.mount
  + it's active so it's good

## Task 8

A close-up of a computer program

Description automatically generated

* We haven't done this before, but you need to know how to do it
* Copy any service file as an example from the default directory
* ls -hal /usr/lib/systemd/system/\*.service
* cp /usr/lib/systemd/system/tuned.service /etc/systemd/system/sleep.service
  + tuned is small and a good one to use
* vi /etc/systemd/system/sleep.service
  + Edit the description to whatever
  + delete after
  + delete requires
  + delete conflicts
  + delete documentation
  + Edit Type to simple
  + delete pid file
  + delete bus name
  + Edit ExecStart to /usr/bin/sleep 3600
  + Save the file
* systemctl cat sshd.service
  + copy the Restart=on-failure and paste under [Service]
* systemctl daemon-reload
  + pick up unit file
* systemctl start sleep.service
* systemctl status sleep.service
* test ability to restart from failure automatically
  + kill -9 <pid from status output>
* systemctl status sleep.service
  + confirm it's running again but with a new pid
* systemctl enable sleep.service

## Task 9

A close-up of a sign

Description automatically generated

* hostnamectl set-hostname examlabs.example.com
* validate
  + cat /etc/hostname
* exit current shell
* sudo -i
  + does he ever do su -i or just sudo -i
* it should be set correctly now
* dnf install -y httpd
* systemctl enable --now httpd
* Get an overview of services
  + firewall-cmd --get-service
  + Note that in firewall-cmd the httpd service is simply named http!!!
* Add it to runtime
  + firewall-cmd --add-service http
* Add it permanently
  + firewall-cmd --add-service http --permanent
* Validate
  + firewall-cmd --list-all
    - should see it

## Task 10

A screenshot of a computer

Description automatically generated

* No special perm in this question
* vi /etc/login.defs
  + Edit PASS\_MAX\_DAYS to 90
* touch /etc/skel/newfile
* If you have special groups for new users, create the groups first
* groupadd students
* groupadd profs
* useradd -G students linda
* useradd -G students lisa
* useradd -G profs anna
* useradd -G profs anouk
* for i in linda lisa anna anouk; do echo password | passwd --stdin $i; done
* mkdir -p /data/profs
* mkdir -p /data/students
* lsblk
  + /data has a separate mount, that's dangerous
  + if this happens in the exam, always do the mount before the permissions
  + check the tasks for any weird dependencies
* cd /data
* chgrp profs profs
* chgrp students students
* chmod 770 \*
* verify
  + ls -hal

## Task 11

A screenshot of a computer screen

Description automatically generated

* vi /etc/httpd/conf/httpd.conf
  + Change Document root to "/web"
  + Change default document root
    - ~5 lines below the other change
      * change the Directory to "/web"
* mkdir /web
* echo hell from web > /web/index.html
* systemctl start httpd
* if it's slow, stop the service and edit /etc/hosts
  + 192.168.29.151 examlabs.example.local
* systemctl start httpd
* systemctl status httpd
* validate
  + curl localhost
  + error message
  + see if it's a se linux problem
  + set enforce permissive
  + curl localhost
  + it works now, so it is an se linux problem
  + set enforce enforcing
  + ls -ldZ /web /var/www/html
  + they are very different
  + man semange-fcontext
  + go to EXAMPLE at the bottom
  + copy the 2 relevant lines to a scratch file
  + semanage fcontext -a -t httpd\_sys\_content\_t "/web(/.\*)?"
  + apply the context to the current file system
    - restorecon -R -v /web
  + curl localhost

## Task 12

A close-up of a text

Description automatically generated

* dnf install git -y
* git clone https://github.com/sandervanvugt/rhcsa-labs
* cd /root/rhcsa-labs
* cat Containerfile
  + super simple
  + prints hello all
* podman build -t greeter .
  + -t is tag, but I guess they say that's the name
* if the question doesn't mention a user name, do it as root
* validate
  + podman images
* podman run --name sleeper greeter:latest
  + should print hello all
* podman ps -a
  + shows created and exited, it does it's thing and then goes away

## Task 13

A close-up of a document

Description automatically generated

* this is rootless so exit your root shell
* be student
* grep unqualified /etc/containers/registries.conf
* podman login registry.access.redhat.com
* log in
* podman login registry.redhat.io
* log in
* he likes to pull before using
  + podman pull registry.redhat.io/rhel9/mariadb-105
* He adds a # to the command line and then adds this stuff
  + #podman run -d -e MYSQL\_ROOT\_PASSWORD=password --name mydb -v /home/student/mariadb:/var/lib/mysql:Z -p 3206:3306 mariadb-105
  + hit enter, and now it's in memory, go take care of the storage
* mkdir mariadb
  + this directory needs the correct ownership
  + podman unshare chown 27:27 mariadb
  + validate
    - podman unshare ls -ld mariadb/
      * you can see 27 27
    - within the container namespace we have correct ownership
* up arrow to your big command and remove the #
* podman ps -a

## Task 14

A close-up of a text

Description automatically generated

* open a root shell
* sudo -i
* loginctl enable-linger student
* open a new tab
* ssh student@localhost
  + do this even if student is logged in already, because you need to come in new to get the loginctl
* podman ps -a
  + container is there but exited
* mkdir -p ./config/systemd/user
* cd .config/systemd/user
* podman generate systemd --new --files mydb
  + it creates container-longstring.service
  + this is your container unit file
* systemctl --user enable container-longstring
* sudo reboot
* log in as anything but student
* ps aux | grep student
  + confirm you see conmon long process

## Grade the Exam

* reboot before running the script
* log in as student
* sudo -i
* cd rhcsa-labs
* ./exam-grade.sh
  + if you have rebooted after your work, say no to the warning and continue
* It can take a minute to fully run
* He got 520 out of 520
* You can fix any failures and rerun the script
  + do that until you get a perfect score
* the script considers passing to be 70% of 520