GENERAL_COMMANDS

Mittwoch, 2. Dezember 2020 07:25

How to install Linux on Windows

Type below command in windows powershell

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

restart system

search window store and install linux

Linux Command

```
cal                     => shows the calendar
date                    => shows the date
date +'%T'              => shows only the time
date +'%A %d %B %y' => shows the day, date, month, year
hostname            => computer name
hostname -i         => show ip address of hostname
htop                => shows running programs and memory
usuage
man                 => show the manual for command use
man hostname        => show the manual for hostname command
nano                => an application to create text files
uptime                  => show runtime since booting
top                 => shows running programs and memory
usuage
```

```
sudo blkid    : list all connected drives (UUID)
sudo -s       : enables you to run the system as a root
user
su cindy  : using cindy's right to access the computer as a
root user
Alt+ctrl+t    : open new terminal
```

_____
File management system |

```
-------------------------

cd /               => goes to the root directory
cd                      => without anything goes to the
home directory
cd ..                   => returns to the parent directory
cd ../..           => returns to parent of parent
directory
cd sys                  => change to sys directory but
works only user is already in directory (relative path)
cd /sys            => change to sys directory (Absolute
path)
file note.txt      => returns the file type
ll -a                   => list directory contents
including hidden files
pwd                => prints current working directory
realpath note.txt  => returns the full path of file
rm file            => deletes the file with name file
touch file              => creates file with name file
touch my\ file     => creates file with name "my file"
which note.txt          => returns the full path of file or
file location


cat file.txt            => prints the file content to
screen.
cat file.txt > file2.txt => redirect file.txt content to
new file file2.txt
cat > newfile.txt       => create newfile.txt and prompts
user for inputs: Ctrl d to exit
cat newfile.txt | more   => shows file contents in page
format: hit space to scroll
cat newfile.txt | less   => shows file contents in page
format: G endpage g toppage
cp -r folder dest       => copy folder recursively to
destination
less newfile.txt         => same as above
    /joe               => search forward for match with joe
    ?joe                    => search backward for match
    with joe

less -MN newfile.txt     => shows file content with
statistics
ll /usr/bin | less -NM   => shows the folder contents with
statistics
ll -lh --block-size=M    => list the directories with file
size in mb
```

```
ll -rt                        => list directories in a
sorted manner
mkdir folder              => make a folder with name folder
mkdir -p d1/d2/d3         => create folder recursively as
children parent relationship
rm -r folder              => remove folder recursively
rm -ri folder             => remove folder with prompt each
step
yes | rm -ri              => y is pressed as input to the
remove command


df -h                         => return disk management
information


find . -name "hello"                 => search for file
with name hello in current dir
find / -name "hell*"                 => search for files
that contains hell at file name beginning in all file
directory
find / -iname "*ell*"                => search for files
that contains ell within its filename in all directory:
regardless of upper & lower case
find / -ipath "*ell*" 2> /dev/null => search for path that
contains ell within its filename in all directory:
regardless of upper & lower case
                                 => all errors are
                              redirected to another
                              location : /dev/null, which is
                              like a dark pit


_____________________________________
     Compile C program command    |
-------------------------------------
gcc -std=c11 -Wall -fmax-errors=10 -Wextra program.c -o
program => compile c program




hostnamectl set -hostname computername

vi /etc/fstab
    add line
    zzux21/data
```

```
/sbin/ipconfig
```

```
# create archive
tar -cf "filename"
tar -cvf "filename"
```

```
# extract files from archive
tar -xvf "filename"
tar -xf "filename"
```

```
# View folder sizes
```

du -h --max-depth=1 *	Show the folders and the sizes of each

echo with color formatting
https://misc.flogisoft.com/bash/tip_colors_and_formatting

LINUX_SHELL_SCRIPTING_01

Mittwoch, 2. Dezember 2020 15:06

```
#2
# Variables and comments
echo "Hello World" # this is a comment

echo $BASH
echo $BASH_VERSION

name = Mark
echo The name is $name
echo our shell name is $BASH


#3
# Read User input
# Flags: -p -s -a

# for single input
echo "Enter name : "
read name
echo "Entered name : $name"


# for multiple inputs
echo "Enter names : "
read name1 name2 name3
echo "Entered names : $name1, $name2, $name3"


# to give user input in same prompt line
read -p "username : " user_var
read -sp "Password : " pass_var # -s hides the whats the user types in keyboard
echo "username: $user_var"
echo "Password: $pass_var"


# to read input as arrays flag: -a
echo "Enter names: "
read -a names

echo "Names : ${names[0]}, ${names[1]}"


# default read no flags, no variable
# default read variable is $REPLY
echo "Enter name: "
read

echo "Name : $REPLY"




#4
# Pass argument to bash script
# Flags:

# $0 is always the script name
```

```
echo $1 $2 $3 ' > echo $1 $2 $3'


# Passing argument as an array
args=("$@")
echo ${args[0]}, ${args[1]}, ${args[2]}
echo $@ # prints all argument in the array
echo $#  # prints the number of arguments
```

https://www.youtube.com/watch?v=OO2Avn1g5Tw&list=PLS1QulWo1RIYmaxcEqw5JhK3b-6rgdWO_&index=5

```
Integer comparison
-eq  : is equal to - if [ "$a" -eq "$b" ]
-ne  : is equal to - if [ "$a" -ne "$b" ]
-gt  : is greater than - if [ "$a" -gt "$b" ]
-ge  : is greater than or equal to - if [ "$a" -ge "$b" ]
-lt  : is less than - if [ "$a" -lt "$b" ]
-le  : is less than or equal to - if [ "$a" -le "$b" ]
<    : is less than - (("$a" < "$b"))
<=   : is less than or equal - (("$a" <= "$b"))
>    : is greater than - (("$a" > "$b"))
>=   : is greater than or equal - (("$a" >= "$b"))

string comparison
=    : is equal to - if [ "$a" = "$b" ]
==   : is equal to - if [ "$a" == "$b" ]
!=   : is not equal to - if [ "$a" != "$b" ]
<    : is less than, in ASCII alphabetical order - if [[ "$a" < "$b" ]]
>    : is greater than, in ASCII alphabetical order - if [[ "$a" > "$b" ]]
-z   : string is null, that is, has zero length


others
&&   : (-a) => boolean and operation
||   : (-o) => boolean or  operation



#! /bin/bash
# Arithmetric operations

num1=20
num2=5
echo $(( num1 + num2 ))
echo $(( num1 - num2 ))
echo $(( num1 * num2 ))
echo $(( num1 / num2 ))
echo $(( num1 % num2 ))

# alternative
echo $(expr num1 + num2 )
echo $(expr num1 - num2 )
echo $(expr num1 \* num2 )
echo $(expr num1 / num2 )
echo $(expr num1 % num2 )



#! /bin/bash
```

```
#floating point math operations in bash | bc command
echo "25.5+3" | bc
echo "scale=20;25.5+3" | bc

num1=20.5
num2=5
echo "$num1 / $num2" | bc
echo "$num1 % $num2" | bc

num=27
echo "scale=2;sqrt($num)" | bc -l
echo "scale=2;3^3" | bc -l

#more option can be found with man bc
```

```
#! /bin/bash
# the case statement 1

vehicle=$1

case $vehicle in
    "car")
       echo "Rent of $vehicle is 100 dollar";;
    "van")
       echo "Rent of $vehicle is 80 dollar";;
    "bicycle")
       echo "Rent of $vehicle is 5 dollar";;
    "truck")
       echo "Rent of $vehicle is 150 dollar";;
    *)
       echo "Unknown vehicle";;
esac
```

```
#! /bin/bash
# the case statement 2

echo -e "Enter some character : \c"
read value

case $value in
    [a-z] )
       echo "User entered $value a to z" ;;
    [A-Z] )
       echo "User entered $value A to Z" ;;
    [0-9] )
       echo "User entered $value 0 to 9" ;;
    ? )
       echo "User entered $value special character" ;;
    *)
       echo "Unknown input" ;;
esac


#14
#! /bin/bash
```

```
# array variable

os=('ubuntu' 'windows' 'kali')
os[3]='mac'
os[4]='linux'

echo "${os[@]}"
echo "${os[0]}"
echo "${!os[@]}"
echo "${#os[@]}"

unset os[4]
echo "${os[@]}"
echo "${os[0]}"
echo "${!os[@]}"
echo "${#os[@]}"




#15
#! /bin/bash
# while loops

n=1
while [ $n -le 10 ]
#while (( $n <= 10 ))       # alternative
do
    echo "$n"
    n= (( $n + 1 ))
    #(( ++n ))         # alternative
    #(( n++ ))         # alternative
    sleep 1       # slows the computer for 1 sec
done

gnome-terminal &   # open a terminal dependent on shell type
xterm &                 # open a terminal dependent on shell type


#17
#! /bin/bash
# Read a file content in bash

cat hello.sh | while read -r line    # -r flag is used to allow escapes for backslash
do
    echo $line
done

# alternative

while read -r line
do
    echo $line
done < hello.sh


#18
#! /bin/bash
# Until loop condition

n=1
```

```
until [ $n -ge 10 ]
do
    echo $n
    sleep 1
    (( n++ ))
    #n=$(( n+1 )) # alternative
done
```

```
#19
#! /bin/bash
# for loop condition

for i in 1 2 3 4 5
do
    echo $i

done

for i in {1..5}                 # {start..end}
do
    echo $i

done

for i in {1..10..2}             # {start..end..increment}
do
    echo $i

done

echo ${BASH_VERSION}
for (( i=1; i<10; i++ ))    # {start..end..increment}
do
    echo $i

done
```

```
#20
#! /bin/bash
# for loop condition for excuting commands

for command in ls pwd date
do
    echo "------------$command-------------"
    $command

done

for item in *
do
    if [ -d $item ]
    then
        echo $item
    fi
done
```

```
#21
#! /bin/bash
# select loop

select name in mezie harsh philemon stephan
do
    echo "$name has be selected as the winner"

done

select name in mezie harsh philemon aditya
do
    case $name in
    mezie)
        echo mezie is in E1 Department
    ;;
    harsh)
        echo harsh is in Furth im Wald
    ;;
    philemon)
        echo philemon is in material Department
    ;;
    aditya)
        echo aditya is with continental
    ;;
    *)
        echo "Error please enter a valid option"
    esac
done
```

```
#22
#! /bin/bash
# the use of break and continue in loops

#break
for (( i=1; i<10; i++ ))
do
    if [ $i -gt 5 ]
    then
        break
    fi
    echo $i
done

# continue
for (( i=1; i<10; i++ ))
do
    if [ $i -eq 3 -o $i -eq 6 ]
    then
        continue
    fi
    echo $i
done
```

```
#23
#! /bin/bash
# Function and argument

function hello(){
       echo hello

}

quit(){
    exit
}

hello
quit

function print(){
    echo $1
}

print hello
quit
```

```
#24
#! /bin/bash
# Local variable

function print(){
    local name="$1"                        # local keyword prevent variable accessible from
    global namespace
    echo "my name is $name : local variable"
}

name = Tom
echo "my name is $name : global variable"
print(max)
echo "my name is $name : global variable"
```

```
#25
#! /bin/bash
# Local variable
# the second option for the use of function
function usage(){
    echo "You need to provide an argument: "
    echo " "
    echo "usage : $0 file_name"
}
function is_file_exist(){
     file="$1"
    [[ -f $file ]] && return 0 || return 1
```

```
}

[[ $# -eq 0 ]] && usage

if (is_file_exist "$1")
then
    echo "File found"
else
    echo "File not found"
fi
# note that for error with bad interpreter
# open the sh file with vi as "vi scriptFile"
# type the following ":set ff=unix" and press enter
# type the following :x and press enter




#26
#! /bin/bash
# variables and functions can be made readonly
# Hence, variable redefinition and function overwriting is restricted
# This is achieved by the keyword : "readonly"
var=31

readonly var

var=56

echo $var


hello(){
    echo "Hello World"
}

readonly -f hello  # use flag -f with keyword "readonly" to declare function readonly

hello(){
    echo "Hello World Again"
}

hello

readonly -p   # displays all readonly variables
readonly -f   # displays all readonly functions




#27
#! /bin/bash
# signals and traps

# interrupt signal : ctrl + c (signal no: 2)
# stop signal      : ctrl + z (signal no: 15) see man -7 signal
# kill signal      : kill -9 pid (on terminal)
# $$ => provides the pid of a script

file=~/Documents/file.txt
touch $file
```

```
trap "echo signal detected; rm -f $file && echo file deleted; exit" 0 2 15
# note that trap command is similar to python: try and except concept


echo "pid is $$"
while (( COUNT < 10 ))
do
    sleep 10
    (( COUNT ++ ))
    echo $COUNT
done
exit 0




#28
#!/bin/bash -x
# debugging
# the flag: -x in the first line activates the debugg for all code lines
# the flag can be activated during run time: bash -x ./28_00_debug.sh

set -x        # activates the debug

file=~/Documents/file.txt
touch $file

set +x    # deactivates the debug flag above

trap "echo signal detected; rm -f $file && echo file deleted; exit" 0 2 15
# note that trap command is similar to python: try and except concept


echo "pid is $$"
while (( COUNT < 10 ))
do
    sleep 10
    (( COUNT ++ ))
    echo $COUNT
done
exit 0







AWK

# returns the entire line that contains data in $1 which occurs for the fourth time
awk '++A[$1]==4' file
awk '++A[$1, $2]==4' file # more data of interest can be added


------------------------------------------------------------
# how to split string by delimiters or Field Seperators
------------------------------------------------------------
myvar="string1,string2,string3"
```

```
# Here comma is our delimiter value
IFS="," read -a myarray <<< $myvar



------------------------------------------------------------
# How To Assign Output of a Linux Command to a Variable
------------------------------------------------------------
variable_name=$(command)
variable_name=$(command [option ...] arg1 arg2 ...)
OR
variable_name='command'
variable_name='command [option ...] arg1 arg2 ...



------------------------------------------------------------
# How To replace character in string
------------------------------------------------------------

$ sed 's/ /_/g' <<< "$a"
hello_world
```

VIM_COMMANDS

Mittwoch, 2. Dezember 2020 15:08

```
dd  => Delete line
o   => Creates newline after current pos and goes to
insert mode
d   => Cuts line of the current cursor positon
yy  => Copies the current line when cursor is positioned
p   => Paste
v   =>
U
 c -> change
 A -> move cursor to end of the line and switch to insert
mode
ctrl + R
gg  => Go to the top of file
G   => Go to the bottom of file
/text -> search for text in forward direction
?text -> search for text in backward direction
^     -> go the beginning of first word in line
$

!ls
:%s/old/new/g
```

AWK

Mittwoch, 15. September 2021 10:11

`awk '{print $2}' datafile.txt`	Print the data in column 2 for all lines in datafile.txt
`awk 'NR < 2 {print $0}' datafile.txt`	Prints row with row number between 0 and 2

00_01_Utility - cut_tr

Dienstag, 29. März 2022 09:43

$ echo "this is a line" | cut -c 1-10	This is a	Cut returns the first 10 characters
$ cut -c 1-10 {file}		Cut returns the first 10 characters of every line in the file
$ cut -c 11- {file}		Cut returns from the 11th character to the end of the line for all lines in the file
$ cut -d ':' -f5 "ether ac:20:fd:9c:71"	71	Cut uses the delimiter flag to split text into fields and specify which field to return (in this case, it is field 5)

Tr is used to translate or delete characters

It is often referred to as the poor man's sed

$ echo "hello world" | tr 'hw' 'HW'	Hello World	Tr replaces the characters in the first argument which are found in the line with the corresponding character found in the second argument
$ echo "hello world" | tr -d 'hw'	ello orld	tr uses the delete flag to delete all characters in the given argument found in the txt line
$ echo "Thiis issss aaaaa lineeeee" | tr -s 'isae'	This is a line	tr uses the squeeze flag to replace sequential repeated character with single character. It also performs translation when a second argument is provided
$ echo "Hello world" -cd 'el'	elll	tr uses the complement flag and delete flag to delete characters other than the ones specified in the argument
$ head 3 /dev/urandom | tr -cd '[:print:]'		tr uses the complement flag and delete flag to delete all characters that considered not printable
$ head 3 /dev/urandom | tr -cd '[:digit:]'		tr uses the complement flag and delete flag to delete all characters that considered not as digits

Utility - sed

Mittwoch, 30. März 2022 09:51

Substitution	$ sed 's/find/replace/' file	Replace the first instance of find in every line of the file with replace
Substitution	$ sed 's/find/replace/g' file	Replace the all instance of find in every line of the file with replace
Substitution and save back to file.	$ sed -i 's/find/replace/' file	Replace the first instance of find in every line of the file with replace and overwrite the original file

C05-P106::CONNECTING TO RHEL7

Dienstag, 15. Dezember 2020 11:42

Monitor security in realtime

- Open terminal
- Log in as root user
  >> su -
- Type as below
  - >> tail -f /var/log/secure
- To close file
  - Press Ctrl + c

Open Virtual terminal windows (Graphical or nongraphical Environment)

- Press Alt+F1 through Alt+F6 to open terminal windows (nongrahical env)
- Press ctrl+Alt+F1 through Ctrl+Alt+F6 to open terminal windows (grahical env)
- Altenatively use the "chvt" command
  - >> chvt 3
  - # this switches on to virtual terminal 3

```
# all virtual consoles have their device files (non graphical env)
# /dev/tty1 to /dev/tty6

# all virtual consoles have their device files (graphical env)
# /dev/pts/1 to /dev/pts/6
```

To see all user currently logged in

Open a terminal and type "w"
Observe the tty number and pts number

System shutdown

```
>> systemctl reboot or reboot
>> systemctl poweroff or poweroff
>> systemctl halt or halt

Emergency reset
# data will be lost
# to be used as a last resort
>> echo b > /proc/sysrq-trigger
```

Access servers on network using ssh non- graphical

No firewall should be active
Port : 22 # default port

If different port is configured
Use the following command
>> ssh -p {port_number}

For use on windows install putty on windows machine

To login as user or root
>> ssh remoteserver -l root
>> ssh remoteserver -l user

Or
>> ssh root@server

Example

```
>>systemctl status sshd   # to see status
>>systemctl stop firewall # to stop firewall and prevent problems
>>ip a | grep "inet "     # fetch ip address of server to log onto
>>ssh 192.168.0.1 -l root # log onto server as root
>> yes                   # respond to security prompt
# enter the root password to log in
>> w                     # to see a new terminal session as ssh
>> exit                  # log out
```

Access servers on network using ssh graphical interface

```
ssh -X 192.168.0.1 -l root # option -X enables support for graphical environment

# to connect to a server(server2) as linda with graphical environment
ssh -X linda@server2 -l root # option -X enables support for graphical environment
```

```
# you can include the following option
-v verbose to show event details while establishing connection
-p to specify port, default port is 22
```

Securely transfer of files between Systems

Use command scp :to copy files
Scp /etc/host server2:/tmp

Scp root@server2:/etc/password ~ => copies file from server2 to home directory as rootuser

Scp -r server2:/etc/ /tmp => copy entire subdirectory structure from server to tmp

Scp -P 23 /etc/host server2:~ => copy to specific server with port (-P)

Connecting to Remote Server with Public/Private keys

Serverl - open terminal shell
Type ssh-kegen
- Prompt >> use paraphase: press enter
- Prompt >> filename to store private key: accept default ~/.ssh/id_rsa.pub file
  >> enter a passphrase: press Enter twice

```
    Private key will be written and stored in pub file

    >> ssh-copy-id server2 : copy public key over to server2
                           : password will be requested for last time

# Verify authentication
>> ssh server2 : connecting to server2 and authentication should work without
password
```

C06-P123::USER AND GROUP MANGEMENT

Donnerstag, 11. Februar 2021 11:42

USER TYPES

>>> id demo	To see details about current user (demo)

METHODS TO RUN TASKS WITH ELEVATED PRIVILEDGES

>>> su	To open a subshell as advantest user
>>> su -	gain access to root user after providing password.
>>> sudo mkdir	Allows an unpriveleged user to execute command (mkdir) as a root user or administrator

SETTING UP SUDO PRIVILEGDES AFTER INSTALLATION

>>> usermod -aG wheel user	Make the admin user account member of the group wheel
>>> visudo	After execution,
%wheel ALL=(ALL) ALL	ensure that the command is included in a line

SWITCHING USER ACCOUNT

>>> whoami	To see current user account
>>> useradd -G wheel demo	Create user demo who is a memeber of group wheel
>>> id demo	Verify that demo is created
>>> passwd demo	Set password for demo. Enter passwd twice
# log out and log in as demo	
>>> sudo useradd albert	Enter passwd and notice the account created

MANAGING USER ACCOUNTS

System Account
Normal Account
Both user share common properties in /etc/passwd file
The file contains 7 columns (field)

/etc/passwd/	
Username	User unique name
Password	User password store in /etc/shadow
UID	Unique user ID is numeric ID that determines what a user can do (Permission) UID = 0 : Root user UID = 1-999 : system accounts user UID = 1000- above : regular user UID range for regular user accounts can be set in /etc/login.defs
GID	Group which the user belongs to for permission management
Comment Field	For extra information. Eg. Reason why the account was create
Directory	Home directory of the user account for storing personal files and programs
Shell	Program that starts after successful login of user most user: /bin/bash system user: /sbin/nologin

/etc/shadow	Has 9 fields or columns
Login name	Df
Encryted password	Enough to show passwd in secured way
Days	That passwd was last changed
Days	Before passwd may be changed. Default:0

Days	After which passwd must be changed default:99999
Days	Before user is warned that passwd expires: default:7
Days	After passwd expires account is disabled: enforce password change
Days	Account is disabled
Reserved	For future

Creating Users

- By editing the files - should be avoided due to error while editing (vipw -s /etc/passwd or /etc/shadow)
  - /etc/passwd
  - /etc/shadow
- Best approach use the following utility commands
  - useradd
  - userdel : to remove users
  - userdel -r : to remove a user and ist user environment
  - Useradd -m -u 1201 -G sales, ops linda : creates a user linda who is a member of the group sales and ops with UID 1201 and add a home directory to the user account as well(- m)
- /etc/skel : This directory contains configuration files that determine how the user environment is set up. Example: selecting files that needs to be in the home directories of all users.

Managing user properties

vipw	This can be done by modifying the configuration files using **vipw**
usermod	Command utility to modify user properties. Except for setting passwords which always encryption. Best approach to set user password is **passwd** command as root

Configuration files for user Management Defaults

Useradd	When used default values are set in two configuration files /etc/login.defs /etc/defaults/useradd
/etc/defaults/useradd	This file has the following default values
GROUP	100
HOME	/home
INACTIVE	-1
EXPIRE	
SHELL	/bin/bash
SKEL	/etc/skel
CREATE_MAIL_SPOOL	yes

/etc/login.defs	This file sets up the environment for new user
MOTD_FILE	Define file for "Message of the day". It includes message displayed to user after successful login
ENV_PATH	Defines the $PATH variable, a list of directories that should be searched for executable files after login
PASS_MAX_DAYS PASS_MIN_DAYS PASS_WARN_AGE	Define users default password expiration properties
UID_MIN	The first UID to use when creating new user
CREATE_HOME	Indicates if home directory for new user is to be created
USERGROUPS_ENAB	If "yes": creates private group for all new user (new user has a group with same name as user) if "No", all users are made a member of the group users

Managing Password Properties

Passwd -n 30 -w 3 -x 90 linda	Sets the password for user linda to min of 30days, expiry of 90days and warning 3 days before expiry
Chage -E 2015-12-31 bob	The user account bob expires on 31.12.2015
Chage -l Chage -l demo	This shows the current password management settings of an account

Creating a User Environment

The following files are used to set up the user environment. The files are read in the following order as well.

/etc/profile	Default settings for all user when starting a login shel
/etc/bashrc	Define defaults for all users when starting a subshell
~/.profile	Specific setting for one user applied when starting a login shell
~/.bashrc	Specific setting for one user when starting a subshell

Creating User Accounts

$ vim /etc/login.defs	Open configuration file, modify parameters CREATE_HOME : yes USERGROUPS_ENAB : yes

$ cd /etc/skel	Navigate to the skel directory and create the following folders
	$ mkdir Pictures
	$ mkdir Documents

$ vim .bashrc	Modify the file content .bashrc by including the line below for default text file edit application. export EDITOR=/usr/bin/vim
$ useradd linda	Creates the account for linda
$ id linda	Verifies that account was created successfully Also check that document, picture folders were created as well
$ passwd linda	Create password for linda "default -> password"
$ passwd -n 30 -w 3 -x 90 linda	Change linda account password properties
$ for i in lucy, lori, bob; do useradd $i; done	Create a more account
$ grep lori /etc/passwd /etc/shadow /etc/group	Check the 3 critical files and confirm that setup was correctly done

Creating and Managing Group Accounts

Every account user belong to at least one member group (primary group).

/etc/password	User primary group is defined in this file
/etc/group	User group configuration is stored in this file

/etc/group	This file has 4 fields (demo:x:1000)

GROUP NAME	The name of the group
GROUP PASSWORD	The password for the group membership. Obsolete
GROUP ID	A unique numeric group id number
MEMBERS	Not shown

Creating Group

There are two ways to add users

$vigr ->	opens an editor interface directly on the /etc/group configuration file
$groupadd	Utility command

$ groupadd sales	Creates group sales
$ groupadd account	Create group account
$ usermod -aG sales demo	Add user demo to sales group
$ usermod -aG account linda	Add user linda to account group
$ id demo	Verify if operation was successful

C07-P155::MANAGING FILE OWNERSHIP

Montag, 18. Oktober 2021 11:45

Displaying Ownership

$ls -l	To see current ownership (user and group owner)
$find / -user demo	To see a list of all files on the system that belongs to user
$find / -group users	Search for all files that have a specific group as their owner

Changing User Ownership

$chown demo report.txt	change the ownership for the file report.txt to user(demo)
$chown -R demo /home/demo	Change the ownership for the directory /home and everything beneath it to user(demo)

Changing Group Ownership

$chown .account /home/account	Set group account as group owner of account folder without changing the user owner
$chown lisa.sales myfile	Set user (lisa) as user owner and group (sales) as group owner
$chown lisa:sales myfile	Set user (lisa) as user owner and group (sales) as group owner
$chown .sales myfile	Sets group (sales) as group owner of myfile without changing the user owner
$chown :sales myfile	Sets group (sales) as group owner of myfile without changing the user owner
Chgrp account /home/account	Sets the group ownership for the directory /home/account to the group account (utility function)
Chgrp -R account /home/account	Sets the group ownership for the directory /home/account to the group account (utility function) recursively

Default Ownership

$groups >>> lisa account sales	list all groups the current user belongs to. The first being the primary group
$newgrp sales >>> sales lisa account	this command sets the primary group of user to sales (as primary group) if user belongs to more than one group
$gpasswd	Sets the password of a group for non group members to use

Managing Basic Permission

Read	4
Write	2
Execute	1
$chmod 755 /somefile	Absolute mode configuration

	Set read, write, execute for user; Set read, execute, for the usergroup; Set read, execute, for the other group;

Relative Mode
(u) user
(g) group
(o) other

Permission
Read [r], write [w], execute[x]

$chmod +x somefile	Relative mode: adds execute permission to all user
$chmod g+w,o-r somefile	Relative mode: write permission to group; remove read permission from other
$chmod -R o+rx /data	Set execute permission on all directories using recursive option (-R)
$chmod -R o+rX /data	The uppercase X ensures that only directory get the execution permission. Files are excludes from execution permission

Pg.164 Managing Advanced Permission

This applies SUID; SGID & sticky bit using chmod utility command

PERMISSION	NUMERIC VALUE	RELATIVE VALUE	FILES	DIRECTORIES
SUID	4	u+s	User exec file with file owner permission	No meaning
SGID	2	g+s	User exec file with group owner permission	Files created in dir get the same group owner
Sticky bit	1	+t	No meaning	Prevent users from deleting files from other users

Chmod 2755 /somedir	Adds the SGID permission to dir and set rwx for user and rx for group and others (drwxr-sr-x)
Chmod g+s, o+t /data/sales	Set the group ID bit as well as stickybit on a shared group directory

Pg.166 Managing ACL (Access Control List)

Setting ACL should be done as root user

$ getfacl -R /directory > file.acls	To create backup of ACL
$ setfacl --restore=file.acl	To restore the settings from backup file
$ getfacl /dir	Checks the permission for different entries
$ setfacl -m g:sales:rx /dir	The command gives read and execute permission to sales group -m -> indicates modify current ACL settings

	g:sales:rx -> indicates the ACL to read and execute
$ setfacl -R -m u:linda:rwx /data	Gives permission to user linda on the /data directory and all files within without making her owner
$ setfacl -m d:g:account:rwx,g:sales:rx /data/sales	Set default "d:" ACL permission for the sales directory (groups: account, sales). This applies to new directory or files created within the directory "sales" for both groups

Prepare file system for ACLs

/etc/fstab	Add "acl mount" to the /etc/fstab file

Pg.169 Setting Default Permissions with unmask

Maximum setting

666	Files
777	Directories

Value	Files	Directory
0	rw	everything
1	rw	rw
2	r	rx
3	r	r
4	w	wx
5	w	w
6	-	x
7	-	-

The unmask setting value is always subtracted from the default permission for a file or directory to get the effective permission.

Unmask setting	Description
022	This gives 644 for all new files and 755 for all directories

Option 1	/etc/profile	Set the unmask for all users
Option 2	Create unmask.sh and specify permission. store file in /etc/profile.d directory	
Option 3	Modify unmask setting in .profile stored in user home directory	

Pg.170 Working with User Extended Attributes

Attributes works regardless of the user that accesses the file

A	Ensure file access time of a file is not modified
a	Allow a file to be added to but not to be removed
c	Ensures that the file is compressed the first time the compression engine gets active
D	Ensure that changes to file are written to disk immediately and not cache first

d	Ensures that the file is not backed up in the backups where the dump utility is used
i	Makes a file immutable. No changes can be made to the file
s	Overwrites the blocks where the file was stored with 0s after file is deleted. Therefore recovery of file is not possible after it has been deleted
u	Saves undelete information. Allows utility to be developed that works with that information to salvage deleted files

Chattr +s somefile	Applies attribute to somefile
Chattr -s somefile	Remove attribute to somefile
$ lsattr	List all attributes that are currently applied

C08-P180:: NETWORKING FUNDAMENTALS

Dienstag, 7. Dezember 2021 12:15

Networking Fundamentals

IPv4 (32bits address) -> 192.168.10.100
IPv6 (128bits address) -> fe80:badb:abe01:45bc:34ad:6723:8798

IP Address	Address assigned to devices (computer, router) on a network. Devices are also referred to as node
Host	A server providing services on the network
Router	Used for computers to communicate with another computer on another network. it connects network to another network

Private IP address
These addresses are used for internal use only
Computer on a private network cannot access the internet and cannot be accessed from the internet.

10.0.0.0/8	Single class A network
172.16.0.0/12	16 class B networks
192.168.0.0/16	256 Class C networks

Pg.238/(181)NAT: Network Address Translation
This enables computer in a private network to access the internet by replacing the private IP address with that of the NAT router.

The NAT router uses table to keep track of all connections that are existing for the hosts in the network. This is how the NAT router helps computer with private IP address to connect to hosts on the internet

Network Masks

Subnet mask	Indicates the network address and that of the node or computer on the network
CIDR	Classless interdomain routing is the form or notation in which network address are specified.
192.168.10.100/24	Indicates a 24bit network address is used
192.168.10.100/255.255.255.0	Indicates same as the address above

192.168.10.xxx	Network part Therefore,
192.168.10.0	Network address
xxx.xxx.xx.100	Host part on the network

Broadcast address	The address that is used to address all nodes in the network. Therefore all bits for the host part are set to 1 192.168.10.255

Example 2

Ip address (Network addr + Node addr)	212.209.113.33/27
Ip address (Network addr + Node addr)	11010100.11010001.00001010.00100001
Subnet mask addr /27	11111111.11111111.11111111.11100000
Subnet mask addr /27	255.255.255.224
Network address	11010100.11010001.00001010.001xxxxx
Network address	212.209.113.32/27
Broadcast address (Network addr + all Node addr)	11010100.11010001.00001010.00111111
Broadcast address (Network addr + all Node addr)	212.209.113.63
Number of ip address for network (Network and broadcast addr)	2
Number of ip address for Host ip addr	30
Total number of ip address	32

MAC ADDRESS - Each network card has an address called MAC address

- They are mainly used for local network
- They cannot be used for communication between nodes on a different network
- They help to find the specific network card that an IP address belongs to

Protocol and PORTs

In addition to IP address, port address are needed to identify the services offered on the network.

The port addresses is used between nodes in same or different network (sender
Sender : source port address
Receiver: destination port address

HTTP	80
SSH	22

Services offered on the network using protocols between the ip address and port address. The protocols are as follows

TCP	Transport control protocol	Used when network communication must be reliable guaranteed Delivery
UDP	User datagram protocol	Must be fast guaranteed delivery is not necessary

Pg.183 MANAGE NETWORK ADDRESSESS AND INTERFACES

Fixed ip address	Mainly used for servers, where ip addr stays same
Dynamic assigned ip address	Mainly used for endusers and devices. (DHCP) Dynamic host configuration protocol

Network card names consist of the follow
Example: eno1677764

en	Ethernet
wl	WLAN
ww	WWAN

Type of adapter

o	Onboard
s	Hotplug slot
p	Pci location

Number which represent index, ID, or port
1677764

p4p1 -> PCI slot 4, port 1

Pg.242(185) Validating Network Configuration

$ ip addr show	To show current network settings
$ ip a s	To show current network settings

Important part is the ethernet onboard card configuration

CURRENT STATE	UP
MAC ADDRESS CONFIG	00:0C:29:B8:8C:EB
BROADCASE ADDRESS	FF:FF:FF:FF:FF:FF
IPV4	192.168.4.220/24
Ipv4 Broadcast addr	192.168.4.255
Ipv6	Fe80::20c:29ff:feb8:8ceb/64

```
[root@server2 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
  pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:b8:8c:eb brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.168.4.220/24 brd 192.168.4.255 scope global eno16777736
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb8:8ceb/64 scope link
       valid_lft forever preferred_lft forever
```

$ ip link show	shows only the link state of the network interfaces which are also shown when $ip addr show is executed
$ ip link set dev eno16777736 up	Temporary bring the link up if down
$ ip -s link	Shows all existing network connections, packets statistics (sent, receive) & error messages

Validating Routing

Routing is required on network that needs to communicate to other networks.
Every network has a default router (default gateway) for communicating with nodes on a different network.

```
[root@server2 ~]# ip route show
default via 192.168.4.2 dev eno16777736  proto static  metric 1024
192.168.4.0/24 dev eno16777736  proto kernel  scope link  src
192.168.4.220
```

Validating Routing
Routing is required on network that needs to communicate to other networks.
Every network has a default router (default gateway) for communicating with nodes on a different network.
At all times the default router should be on the same network as the local ip addr That the network card uses.

```
[root@server2 ~]# ip route show
default via 192.168.4.2 dev eno16777736  proto static  metric 1024
192.168.4.0/24 dev eno16777736  proto kernel  scope link  src
192.168.4.220
```

$ ip route show	To see which router is used as default
$ ss -lt	Display all the listening ports on the local system

Pg.244(187) Validating Ports and Services Availability
Network problems could be as follows

- Network ports not availabe on server or on a remote server

Listing 8.4 Use **ss -lt** to Display All Listening Ports on the Local System

```
[root@server2 ~]# ss -lt
State      Recv-Q Send-Q     Local Address:Port       Peer Address:Port
LISTEN     0      100        127.0.0.1:smtp          *:*
LISTEN     0      128        *:56601                 *:*
LISTEN     0      128        .0.0.1:x11-ssh-offse    *:*
LISTEN     0      128        *:sunrpc                *:*
LISTEN     0      128        *:ssh                   *:*
LISTEN     0      128        127.0.0.1:ipp           *:*
LISTEN     0      100        ::1:smtp                :::*
LISTEN     0      128        ::1:x11-ssh-offset      ::: *
LISTEN     0      128        :::sunrpc               :::*
LISTEN     0      128        :::34449                :::*
LISTEN     0      128        :::ssh                  :::*
LISTEN     0      128        ::1:ipp                 :::*
```

Notice where the port is listening on. Some ports are only listening on the IPv4 loopback address 127.0.0.1 or the IPv6 loopback address ::1, which means that they are locally accessible only. Other ports are listening on *, which stands for all IPv4 addresses, or on :::*, which represents all ports on all IPv6 addresses.

$netstat	Verify port availability on your server
$ss	Same as the command above
$ ss -lt	Display all the listening ports on the local system

Pg.246/(189)Configuring Network Configuration with nmtui and nmcli
Network configuration scripts -> /etc/sysconfig/network-scripts/ifcfg…

- A device is a network interface card
- A connection is the configuration that is used on a device (stored as a configuration file)
  /etc/sysconfig/network-scripts/ifcfg-{network_interface_name}
  /etc/sysconfig/network-scripts/ifcfg-eno16777736

Exercise 8.2 Verifying Network Settings

1. Open a root shell to your server and type **ip addr show**. This shows the current network configuration. Note the IPv4 address that is used. Notice the network device names that are used; you need these later in this exercise.
2. Type **ip route show** to verify routing configuration.
3. If your computer is connected to the Internet, you can now use the **ping** command to verify the connection to the Internet is working properly. Type **ping -c 4 8.8.8.8**, for instance, to send four packets to IP address 8.8.8.8. If your Internet connection is up and running, you should get "echo reply" answers.
4. Type **ip addr add 10.0.0.10/24 dev <yourdevicename>**.
5. Type **ip addr show**. You'll see the newly set IP address, in addition to the IP address that was already in use.
6. Type **ifconfig**. Notice that you do not see the newly set IP address (and there are no options with the **ifconfig** command that allow you to see it). This is one example why you should not use the **ifconfig** command anymore.
7. Type **ss -tul**. You'll now see a list of all UDP and TCP ports that are listening on your server.

Network Configuration with nmcli

$ nmcli con show	List and show all network connections
$ nmcli con show eno16777736	Show all properties of particular connection
$ nmcli	Shows an overview of currently configured devices and status of the devices
$ man 5 nm-settings	Explains the network properties function

Exercise 8.3 Managing Network Connections with nmcli

1. Create a new network connection using **nmcli con add con-name "dhcp" type ethernet ifname eth0**.
2. Create a connection with the name static to define a static IP address and gateway: **nmcli con add con-name "static" ifname eth0 autoconnect no type ethernet ip4 10.0.0.10/24 gw4 10.0.0.1**. The gateway might not exist in your configuration, but that does not matter. (Make sure to change the ifname eth0 into the interface name that matches your hardware!)
3. Type **nmcli con show** to show the connections, and use **nmcli con up "static"** to activate the static connection. Switch back to the DHCP connection using **nmcli con up "dhcp"**.

In this exercise, you created network connections using **nmcli con add**. You can also change current connection properties by using **nmcli con mod**. In Exercise 8.4, you use **nmcli** to change some connection parameters.

Exercise 8.4 Changing Connection Parameters with nmcli

1. Make sure that the static connection does not connect automatically by using **nmcli con mod "static" connection autoconnect no**.
2. Add a DHCP server to the static connection by using **nmcli con mod "static" ipv4.dns 10.0.0.10**. Notice that while adding a network connection you used **ip4**, but while modifying parameters for an existing connection, you'll often use **ipv4** instead. This is not a typo; it is just an inconsistency in the command.
3. To add a second item for the same parameters, use a + sign. Test this by adding a second DNS server, using **nmcli con mod "static" + ipv4.dns 8.8.8.8**.
4. Using **nmcli con mod**, you can also change parameters such as the existing IP address. Try this by using **nmcli con mod "static" ipv4.addresses "10.0.0.20/24" 10.0.0.100**.
5. And to add a second IP address you use the + sign again: **nmcli con mod "static" +ipv4.addresses 10.20.30.40/16**.
6. After changing connection properties, you need to activate them. To do that, you can use **nmcli con up "static"**.

Pg.251/(194) Network Configuration with nmtui

It contains three option

Edit a connection	To create new and edit existing connections (very useful)
Activate a connection	Activate or reactivate a connnection
Set system Hostname	Use to set the hostname of your computer

$ nm-connection-editor	Starts the nmtui application
$ cat /etc/sysconfig/network-scripts/ifcfg-eno16777736	View configuration file
$ nmcli con reload	Activate configuration file after config file modification
Set BOOTPROTO to dhcp	Sets fixed ip addr to dynamic ip in a network connection. While specifying IP address and network prefix
Same above can be achieved	Set the IPV4 configure to

with the nmtui uitility	automatic and not manual and specify an Ip addr

Pg.253/(194) Setting Up Hostname and Name Resolution

Hostnames are used to communicate with other host, access servers and the services they offer.

The system hostname is also known as (Fully qualified domain name FQDN) and it consist of two parts
- Name of host
- DNS domain

FQDN -> server1.example.com

The hostname can be changed in three various ways
- Use nmtui and select the **Change Hostname** option
- Use **hostnamectl set-hostname**
- Edit the contents of the configuration file
  - /etc/hostname

$ hostnamectl set-hostname myhost.example.com	Sets the hostname
$ hostnamectl status	Show current hostname

DNS are also used to set the hostname by configuring the hostname resolution in the file
- /etc/hosts

Settings implemented in the file above applies prior to that in DNS. This is ensured by configuration in the following file

/etc/nsswitch.conf	hosts: files dns

/etc/hosts description
It should contain at least two columns

First column	IP address of the specific host {127.0.0.1}
Second column	Specifies the hostname hostname can be shortname "server1" hostname can be FQDN "server1.example.com" or both in the following order; FQDN shortname -> "server1.example.com" "server1"

Pg.255/(198) DNS RESOLVING

Set the DNS server via Network manager by using nmtui
- Set DNS1 & DNS2 in ifcfg network connection configuration file
  - /etc/sysconfig/network-scripts
- Use a DHCP Server that is configured to hand out the address of DNS name server
- Use **nmcli con mode <connection-id> [+]ipv4.dns <ip-of-dns>**

Deactivate the use of DHCP
- Edit the ifcfg configuration file to include the option
  - PEERDNS=no
- Use **nmcli con mode <con-name> ipv4.ignore-auto-dns yes**

Verify hostname resolution
- Use **getent hosts <servername>**

C09-P206::MANAGING PROCESSES

Mittwoch, 9. Februar 2022 09:33

Shell jobs are commands started from the command line

Daemons are processes that provides services and started normally during boot often with root privileges

Pg.266/(209) Managing shell jobs

Ctrl+Z	Temporarily stops job but retained in memory
Ctrl+C	Stops current job and remove from memory
Ctrl+D	Sends EOF character to current job
$ init.py &	Start job immediately in background
$ fg	Brings back the last job back to the foreground
$ bg	Continue frozen (ctrl+Z) job in the background
$ jobs	Shows all jobs currently running from shell and their job numbers which can be used as arguments to the commands fg & bg
$ top	Show runnning shell job use "k" to kill job Use "r" to adjust process priority
$ ps	Show list of processes started only by the current user
$ ps aux	Show a short summary of the active processes
$ ps -ef	Show active process and the exact command that started them
$ ps fax	Shows hierarchical relationships btw parent and child processes
$ ps aux | grep dd	Show process detail about dd, including ist pid
$ pgrep dd	Alternative to command above

Pg.271/(215) Adjusting Process Priority with nice

Default priority for all regular process is 20
The priority ranges from -20(highest) to 19(lowest)

$ nice -n 5 dd if=/dev/zero of=/dev/null &	start a process with an adjusted priority of 5

$ ps aux | grep dd
 	
$ renice -n 10 -p 1234
	obtain the pid for renice
	
	Change priority of a current active process using the pid.

$ top	List active process use "r" to change priority

Pg.273/(216) Sending Signals to Processes with kill, killall, and pkill

$ man 7 signal	Shows complete overview of all available signals
$ kill -l	List available signals that can be used with kill
$ kill {pid}	Sends SIGTERM signal
$ kill -9 {pid}	Sends SIGKILL signal (leads to data loss)
$ killall	
$ pkill	Takes the name of the process as argument and not pid

Pg.275/(218) Using top to Manage Process

The top utility provides the process states as follows

Running	R	Process is running and active
Sleeping	S	Process is waiting for an event to occur
Uninterruptable Sleep	D	Process is in sleep state that cannot be stopped
Stopped	S	Process was stopped by user (ctrl+Z)
Zombie	Z	Process has stopped and cannot be removed by parent (unmanageble state)

The following functions can be executed using top

Kill	Type "k"	1. Top prompts for the PID of the process

Exercise 9.2 Managing Processes from the Command Line

In this exercise, you learn how to work with ps, nice, kill, and related utilities to manage processes.

1. Open a root shell. From this shell, type **dd if=/dev/zero of=/dev/null &**. Repeat this command three times.
2. Type **ps aux | grep dd**. This shows all lines of output that have the letters dd in them; you will see more than just the dd processes, but that should not really matter. The processes you just started are listed last.
3. Use the PID of one of the dd processes to adjust the niceness, using **renice -n 5 <PID>**. Notice that in top you cannot easily get an overview of processes and their current priority.
4. Type **ps fax | grep -B5 dd**. The **-B5** option shows the matching lines, including the five lines before that. Because **ps fax** shows hierarchical relationships between processes, you should also find the shell and its PID from which all the dd processes were started.
5. Find the PID of the shell from which the dd processes were started and type **kill -9 <PID>**, replacing **<PID>** with the PID of the shell you just found. You will see that your root shell is closed, and with it, all of the dd processes. Killing a parent process is an easy and convenient way to kill all of its child processes

		removed by parent (unmanageble state)

The following functions can be executed using top

Kill	Type "k"	1. Top prompts for the PID of the process
		2. top ask which signal to send
		  1. Signal "15" -> SIGTERM
		  2. Signal "9" -> SIGKILL
		3. Press enter to terminate the process

Renice	Type "r"	1. Top prompts for the PID of the process
		2. top prompts for nice value
		  1. Positive number -> increase process priority
		  2. Negative number -> decrease process priority
		3. Press enter to terminate the process

5. Find the PID of the shell from which the dd processes were started and type **kill -9 <PID>**, replacing **<PID>** with the PID of the shell you just found. You will see that your root shell is closed, and with it, all of the dd processes. Killing a parent process is an easy and convenient way to kill all of its child processes also.

C11-P249::MANAGING SOFTWARE

Montag, 14. Februar 2022 12:02

Pg.306/(249)

Yum	Utility used to manage software packages
Rpm	Utility used to manage software by querying new and installed software packages

Pg.309/(252) Managing Software Packages with Yum

Yum	Yellowdog update manager create and manage repositories

Create repository file with a name that ends in .repo
The .repo file should have the following contents

[label]	The .repo file can contain different repositories with each starting with a label that identifies the specific repository
name=	Specifies the name of the repository to use
baseurl=	Contains the url that points to the specific repository location (most important)

Examples can be found by reading the manual "man yum.conf"

Listing 11.1 Repository File Example

```
[root@server1 yum.repos.d]# cat CentOS-Base.repo
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and
the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.  You should use this for CentOS
updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try
the
# remarked out baseurl= line instead.
#
#

[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$bas
earch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$bas
earch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$bas
earch&repo=extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$bas
earch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/
centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

In the repository configuration file from Listing 11.1, you can see that some options are used. Table 11.2 summarizes these options.

Table 11.2 Key Options in .repo Files

Option	Explanation
[label]	The label used as an identifier in the repository file.
name=	The name of the repository.
mirrorlist=	Refers to a URL where information about mirror servers for this server can be obtained. Typically used for big online repositories only.
baseurl=	The base URL where to go to find the RPM packages
gpgcheck=	Set to 1 if a GPG integrity check needs to be performed on the packages. If set to 1, a gpgkey is required.
gpgkey=	Specifies the location of the GPG key that is used to check package integrity.

Table 11.3 Repository Types and Their Support Status

Type	Description
base	This is the base repository that contains all essential Red Hat software. Its packages are fully supported.
updates	A specific repository that contains updates only.
optional	This repository contains packages that are provided for the convenience of Red Hat customers. The packages in this repository are open source and not supported by Red Hat.
supplementary	This repository contains packages that are provided for the convenience of Red Hat customers. The packages in this repository are proprietary and not supported by Red Hat.
extras	This repository contains packages that are provided for the convenience of Red Hat customers. Software in this repository comes from different sources and is not supported by Red Hat.

Gpg key for security during installation from the internet repositories are stored in the path

- /etc/pki/rpm-gpg

Exercise 11.1 Creating Your Own Repository

In this exercise, you learn how to create your own repository. To perform this exercise, you need to have access to the CentOS installation disk or ISO file.

1. Insert the installation disk in your virtual machine. This mounts it on the directory /run/media/user/CentOS 7 x86_64. Alternatively, you can manually mount the ISO on the /mnt directory, using **mount -o loop /path/to/centos.iso /mnt**.
2. Type **mkdir /repo** to create a directory /repo that can be used as repository.
3. If you want to create a complete repository, containing all the required files, type **cp $MOUNTPATH/Packages/* repo**. (Replace $MOUNTPATH with the name of the directory on which the installation disk is mounted.) If you do not need a complete repository, you can copy just a few files from the installation disk to the /repo directory.
4. Type **yum install -y createrepo** to ensure that the createrepo RPM package is installed.
5. Type **createrepo /repo**. This generates the repository metadata, which allows you to use your own repository.
6. Now that you have created your own repository, you might as well start using it. In the /etc/yum.repos.d directory, create a file with the name my.repo. Make sure this file has the following contents:

```
[myrepo]
name=myrepo
baseurl=file:///repo
```

7. Type **yum repolist** to verify the availability of the newly created repository. It should show the name of the myrepo repository, including the number of packages that is offered through this repository (see Listing 11.3).

Using yum

At this point, you should have operational repositories, so it is time to start using them. To use repositories, you need the **yum** command. This command enables you to perform several tasks on the repositories. Table 11.4 provides an overview of common yum tasks.

Table 11.4 Common yum Tasks

Task	Explanation
search	Search for the exact name of a package
[what]provides */name	Perform a deep search in the package to look for specific files within the package
info	Provide more information about the package

Task	Explanation
install	Install the package
remove	Remove the package
list [all | installed]	List all or installed packages
group list	List pacakge groups
group install	Install all packages from a group

Task	Explanation
install	Install the package
remove	Remove the package
list [**all** | **installed**]	List all or installed packages
group list	List pacakge groups
group install	Install all packages from a group
update	Update packages specified
clean all	Remove all stored metadata

For remaining part see E-book

C12-P282::SCHEDULING TASKS

Donnerstag, 17. Februar 2022 11:57

Services_atd : for scheduling future tasks Once only
Services_crond : for scheduling recurring regular tasks

$ systemctl status crond -l	Monitor current status of the cron service

Pg.342/(285)Understanding cron timing

TIP No need trying to remember all this, **man 5 crontab** shows all possible constructions.

Table 12.2 cron Time and Date Fields

Field	Values
minute	0–59
hour	0–23
day of month	1–31
month	1–12 (or names which are better avoided)
day of week	0–7 (Sunday is 0 or 7, or names [which are better avoided])

In any of these fields, you can use an * to refer to any value. Ranges of numbers are allowed, as are lists and patterns. Some examples are listed next:

- *** 11 * * *** Any minute between 11:00 and 11:59 (probably not what you want)
- **0 11 * * 1-5** Every day at 11 a.m. on weekdays only
- **0 7-18 * * 1-5** Every hour on weekdays on the hour
- **0 */2 2 12 5** Every 2 hours on the hour on December second and every Friday in December

Format can be seen in the file
$ cat /etc/crontab
$ man 4 crontabs

Managing cron configuration files

/etc/cron.d	Config file directly
/etc/cron.hourly, cron.daily, cron.weekly, cron.monthly	Config file 2
User specified files that are created with **crontab -e**	

$ crontab -e	For users

$ crontab -e -u username	For root
/var/spool/cron	Created cron jobs are stored in this location after editing with Crontab command
/etc/cron.d	Or simply add a file in the crontab directory and ensure it meets the syntax of a crontab job or simply use the config files eg.cron.hourly, cron.daily e.t.c

Pg.345/(288) Purpose of anacron

Anacron service ensures the regular execution of cron by starting the hourly, daily, weekly, monthly cron jobs.

This is achieves using the following file

/etc/anacrontab	Specify how anacrontab jobs should be executed
$ cat	To see file details -> contains three fields
$ first field	Specifies the execution frequency
$ second field	Specifies how long anacron waits before executing job
$ third field	Specifies the command that should be executed

Managing cron security

/etc/cron.allow	Allows users listed within to file to use cron
/etc/cron.deny	User listed within are not allowed to use cron

Exercise 1

Exercise 12.1 Running Scheduled Tasks Through cron

In this exercise, you apply some of the cron basics. You schedule cron jobs using different mechanisms.

1. Open a root shell. Type **cat /etc/crontab** to get an impression of the contents of the /etc/crontab configuration file.

2. Type **crontab -e**. This opens an editor interface that by default uses vi as its editor. Add the following line:

```
0 2 * * 1-5 logger message from root
```

3. Use the vi command **:wq!** to close the editing session and write changes.
4. Use **cd /etc/cron.hourly**. In this directory, create a script file with the name **eachhour** that contains the following line:

```
logger This message is written at $(date)
```

5. Use **chmod +x eachhour** to make the script executable; if you fail to make it executable, it will not work.
6. Now enter the directory /etc/crond.d and in this directory create a file with the name **eachhour**. Put the following contents in the file:

```
11 * * * *  root  logger This message is written from /etc/cron.d
```

7. Save the modifications to the configuration file and go work on the next section. (For optimal effect, perform the last part of this exercise after a couple of hours.)
8. After a couple of hours, type **grep written /var/log/messages** and read the messages that have been written which verifies correct cron operations.

Configuring "at" to Schedule one time Future Task

This utility is used for services that needs to be executed only once.

$ at 14:00	Command "at" followed by time opens a shell for type the commands to be executed at the given time
Ctrl+D	Quit the "at" shell after entering the commands to execute
$ atq	Verify that the job has been queued and show the overview of all queued jobs
$ atrm {job_number}	Command "atrm" followed by the job number from "atq" command removes the job from the queue.

TIP The **batch** command works like **at**, but it's a bit more sophisticated. When using **batch**, you can specify that a job is only started when system performance parameters allow. Typically, that is when system load is lower than 0.8. This value is a bit low on modern multi-CPU systems, which is why the load value can be specified manually when starting atd, using the **-l** command-line option. Use for instance **atd -l 3.0** to make sure that no batch job is started when system load is higher than 3.0.

Exercise 12.2 Scheduling Jobs with at

In this exercise, you learn how to schedule jobs using the atd service.

1. Type **systemctl status atd**. In the line that starts with Loaded:, this command should show you that the service is currently loaded and enabled, which means that it is ready to start receiving jobs.
2. Type **at 15:00** (or replace with any time near to the time at which you are working on this exercise).
3. Type **logger message from at**. Use **Ctrl+D** to close the at shell.
4. Type **atq** to verify that the job has indeed been scheduled.

C13-P295::CONFIGURING LOGGING

Mittwoch, 23. Februar 2022 10:19

Pg.355(298) Understanding system logging

Two main type of log exist on linux redhat

-rsyslog	enhance version of syslog for managing centralized logging
-journald	very advanced integrated with systemd which allow detailed information of the system by journal collects messages from kernel, boot procedure and services and store them in an event journal as binary format - allows configuration of log services and remote logging -handles writing log info to specific files

When detail information about the ongoing event on machine is required, the following steps are recommended.

/var/log	Monitor files written by rsyslog in this directory
"journalctl"	Use "journalctl" utility to get more information from a journal
$ systemctl status {unit}	Use the command to show service status and the most recent log. Example -> $ systemctl status httpd Example -> $ systemctl status sshd

Table 13.2 System Log Files Overview

Log File	Explanation
/var/log/messages	The most commonly used log file, it is the generic log file where most messages are written to.
/var/log/dmesg	Contains kernel log messages.
/var/log/secure	Contains authentication related messages. Look here to see which authentication errors have occurred on a server.
/var/log/boot.log	Look here for messages that are related to system startup.
/var/log/audit/audit.log	Contains audit messages. SELinux writes to this file.
/var/log/maillog	Look here for mail-related messages.
/var/log/samba	Provides log files for the Samba service. Notice that Samba by default is not managed through rsyslog, but writes directly to the /var/log directory.
/var/log/sssd	Contains messages that have been written by the sssd service, which plays an important role in the authentication process.
/var/log/cups	Contains log messages that were generated by the print service CUPS.
/var/log/httpd/	Directory that contains log files that are written by the Apache web server. Notice that Apache writes messages to these files directly and not through rsyslog.

Each log message (example /var/log/messages) contains the following,

- TIMESTAMP (date and time);
- HOST(server1);
- SERVICE_OR_PROCESSNAME(systemd);
- MESSAGE CONTENT

```
Mar 25 05:27:21 server1 chronyd[851]: Selected source 65.19.178.219
Mar 25 05:27:52 server1 systemd: Time has been changed
Mar 25 05:28:54 server1 systemd: Time has been changed
Mar 25 05:29:56 server1 systemd: Time has been changed
Mar 25 05:30:03 server1 systemd: Starting Session 2 of user root.
Mar 25 05:30:03 server1 systemd: Started Session 2 of user root.
Mar 25 05:30:10 server1 NetworkManager[1058]: <info> NetworkManager
  state is now CONNECTED_GLOBAL
Mar 25 05:30:11 server1 goa[3009]: goa-daemon version 3.8.5 starting
  [main.c:113, main()]
```

$ tail -f <logfile>	Monitor services in linux by observing the log file in real time.
$ tail -f /var/log/messages	Same as above and can be close with ctrl+c
$ logger <message>	This writes the message into log file
$ logger hello	Same as above This writes the message "hello" into log file (/var/log/messages)

Exercise 13.1 Using Live Log Monitoring and logger

In this exercise, you use **tail -f** to monitor a log file in real time. You also use **logger** to write messages to a log file.

1. Open a root shell.
2. From the root shell, type **tail -f /var/log/messages**.
3. Open a second terminal window. In this terminal window, type **su - user** to open a subshell as user.
4. Type **su -** to open a root shell, but enter the wrong password.
5. Notice that nothing appears in /var/log/messages. That is because login-related errors are not written here.
6. From the user shell, type **logger hello**. You'll see the message appearing in the /var/log/messages file in real time.
7. In the **tail -f** terminal, use **Ctrl+C** to stop tracing the messages file.
8. Type **tail -n 20 /var/log/secure**. This shows the last 20 lines in /var/log/secure, which also shows the messages that the **su -** password errors have generated previously.

Pg.360 (303) Configuring rsyslogd

/etc/rsyslog.conf	Contain different section to specify where info should be written.
/etc/rsyslogd.conf	Central location to configure rsyslog
/etc/rsyslogd	Directory: content of the directory can be included from rsyslogd.conf -it contains specific log information
/etc/sysconfig/rsyslog	Used to pass specific options to rsyslod service on startup By default contains only one line {SYSLOGD_OPTIONS=""}

Configurations are mainly done with the /etc/rsyslog.conf file and contains sections

/etc/rsyslod.conf

MODULES	For enhanced functions
GLOBAL DIRECTIVES	Specifies the global parameters, such as location for aux files, default timestamp format
**RULES	Specifies what information should be logged and the destination as well --> *Facilities, Priorities and log destinations are specified*

i

```
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                 /dev/console
```

```
# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
```

Facilities	Specify the kind of information that is logged
Priorities	Defines the severity of the message to be logged. * by default: messages of specified priority and those of higher priorities are logged
Destination	This is a file where logged messages are stored start filename without hypen (commit log to file): /var/log/secure start filename with hypen (no commit to file but buffered): -/var/log/secure

Facility	Used by
auth / authpriv	Messages related to authentication.
cron	Messages generated by the crond service.
daemon	Generic facility that can be used for nonspecified daemons.
kern	Kernel messages.
lpr	Messages generated through the legacy lpd print system.
mail	Email-related messages.
mark	Special facility that can be used to write a marker periodically.
news	Messages generated by the NNTP news system.
security	Same as auth / authpriv. Should not be used anymore.
syslog	Messages generated by the syslog system.
user	Messages generated in user space.
uucp	Messages generated by the legacy UUCP system.
local0-7	Messages generated by services that are configured by any of the local0 through local7 facilities.

The daemon and local0-7 facility can be used for user defined (specific log purpose)
By adding a rule to the rsyslog.conf file

Table 13.4 rsyslogd Priorities

Priority	Used for
debug	Debug messages that will give as much information as possible about service operation.
info	Informational messages about normal service operation.
notice	Used for informational messages about items that might become an issue later.
warning / warn	Something is suboptimal, but there is no real error yet.
err /error	A noncritical error has occurred.
crit	A critical error has occurred.
alert	Used when the availability of the service is about to be discontinued.
emerg / panic	Message generated when the availability of the service is discontinued.

=debug	To log messages of specified priority only by adding equal (=) infront of the priority cron.=debug /var/log/cron.debug
$ man 5 rsyslog.conf	Access the manual pages

Exercise 13.2 Changing rsyslog.conf Rules
In this exercise, you learn how to change rsyslog.conf. You configure the Apache service to log messages through syslog, and you create a rule that logs debug messages to a specific file.

1. By default, the Apache service does not log through rsyslog, but keeps its own logging. You are going to change that. To start, type **yum install -y httpd** to install the Apache service.
2. After installing the Apache service, open its configuration file /etc/http/conf/httpd.conf and add the following line to it:

```
ErrorLog   syslog:local1
```

3. Type **systemctl restart httpd**.
4. Now create a line in the rsyslog.conf file that will send all messages that it receives for facility local1 (which is now used by the httpd service) to the file /var/log/httpd-error.log. To do this, include the following line:

```
local1:error                    -/var/log/httpd-error.log
```

5. Tell rsyslogd to reload its configuration, by using **systemctl restart httpd**.
6. All Apache error messages will now be written to the httpd-error.log file.
7. From the Firefox browser, go to http://localhost/nowhere. Because the page you are trying to access does not exist, this will be logged to the Apache error log.
8. Now let's create a snap-in file that logs debug messages to a specific file as well. To do this, type **echo "*.debug /var/log/messages/messages-debug" > /etc/rsyslogd/debug.conf**.
9. Again, restart rsyslogd using **systemctl restart rsyslogd**.
10. Use the command **tail -f /var/log/messages-debug** to open a trace on the newly created file.
11. Type **logger -p daemon.debug "Daemon Debug Message"**. You'll see the debug message passing by.
12. Use **Ctrl+C** to close the debug log file.

Pg.365 (308) Rotating Log Files

Create a new log file and save current as old when a threshold is reached. This is performed by the crond service periodically.

/etc/logrotate.conf	Configuration file for rotating log
$ man logrotate	Manual information

The most important configuration in the logrotate.conf are

Rotate log files weekly	Weekly
Keep 4 weeks worth of backlogs	Rotate 4

Specific files can have their own rotate configuration within the /etc/logrotate.d/logrotate.conf file.
The specific configuration superseeds the default settings found within the file.

Listing 13.4 /etc/logrotate.conf Sample Content

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
        minsize 1M
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}
```

Pg.367 (310) Working with journald

The journal is a binary file that stores log messages in /run/log

$ journalctl	To show all event since last server start up * press G to navigate to end page to see recent messages * use / to search in forward direction -> /error * use ? To search in reverse direction -> ?error
$ journalctl -f	Shows the most recent messages logged in the journal
$ journalctl -o verbose	Shows detailed output of all items logged, which includes PID, ID of user and group account, commands and more…

Pg.371 (314) Preserving the systemd Journal

/varlog/journal	Default storage file location

Exercise 13.3 Discovering journalctl

In this exercise, you learn how to work with different journalctl options.

1. Type **journalctl**. You'll see the content of the journal since your server last started, starting at the beginning of the journal. The content is shown in **less**, so you can use common **less** commands to walk through the file.
2. Type **q** to quit the pager. Now type **journalctl --no-pager**. This shows the contents of the journal without using a pager.
3. Type **journalctl -f**. This opens the live view mode of journalctl, which allows you to see new messages scrolling by in real time. Use **Ctrl+C** to interrupt.
4. Type **journalctl** and press the **Tab** key twice. This shows specific options that can be used for filtering. Type, for instance, **journalctl _UID=0**.
5. Type **journalctl -n 20**. The **-n 20** option displays the last 20 lines of the journal (just like **tail -n 20**).
6. Now type **journalctl -p err**. This command shows errors only.
7. If you want to view journal messages that have been written in a specific time period, you can use the **--since** and **--until** commands. Both options take the time parameter in the format YYYY-MM-DD hh:mm:ss. Also, you can use **yesterday**, **today**, and **tomorrow** as parameters. So, type **journalctl --since yesterday** to show all messages that have been written since yesterday.
8. **journalctl** allows you to combine different options, as well. So, if you want to show all messages with a priority err that have been written since yesterday, use **journalctl --since yesterday -p err**.
9. If you need as much detail as possible, use **journalctl -o verbose**. This shows different options that are used when writing to the journal (see Listing 13.3). All these options can be used to tell the **journalctl** command which specific information you are looking for. Type, for instance, **journalctl _SYSTEMD_UNIT=sshd.service** to show more information about the sshd systemd unit.

Preserving the systemd Journal

By default, the journal is stored in the file /run/log/journal. The entire /run directory is used for current process status information only, which means that the journal is cleared when the system reboots. To make the journal persistent between system restarts, you should make sure that a directory /var/log/journal exists.

Even when the journal is written to the permanent file in/var/log/journal, that does not mean that the journal is kept forever. The journal has built-in log rotation that will be used monthly. Also, the journal is limited to a maximum size of 10% of the file system size that it is on, and it will also stop growing if less than 15% of the file system is still free. If that happens, the oldest messages from the journal are dropped automatically to make place for newer messages. To change these settings, you can modify the file /etc/systemd/journald.conf. You'll see that in this file some other parameters can be set also (see Listing 13.5).

Listing 13.5 Setting journald Parameters Through /etc/systemd/journald.conf

```
[Journal]
#Storage=auto
#Compress=yes
#Seal=yes
#SplitMode=login
#SyncIntervalSec=5m
#RateLimitInterval=30s
#RateLimitBurst=1000
#SystemMaxUse=
#SystemKeepFree=
#SystemMaxFileSize=
#RuntimeMaxUse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
```

parameters can be set also (see Listing 13.5).

Listing 13.5 Setting journald Parameters Through /etc/systemd/journald.conf

```
[Journal]
#Storage=auto
#Compress=yes
#Seal=yes
#SplitMode=login
#SyncIntervalSec=5m
#RateLimitInterval=30s
#RateLimitBurst=1000
#SystemMaxUse=
#SystemKeepFree=
#SystemMaxFileSize=
#RuntimeMaxUse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
#MaxRetentionSec=
#MaxFileSec=1month
#ForwardToSyslog=yes
#ForwardToKMsg=no
#ForwardToConsole=no
#TTYPath=/dev/console
#MaxLevelStore=debug
#MaxLevelSyslog=debug
#MaxLevelKMsg=notice
#MaxLevelConsole=info
```

Making the journal permanent is not hard to do. Exercise 13.4 shows how to proceed.

Exercise 13.4 Making the journald Journal Permanent

In this exercise, you learn how to make the journald journal permanent.

1. Open a root shell and type **mkdir /var/log/journal**.
2. Before journald can write the journal to this directory, you have to set ownership. Type **chown root:systemd-journal /var/log/journal**, followed by **chmod 2755 /var/log/journal**.
3. Next, you can either reboot your system (restarting the systemd-journald service is not enough) or use the **killall -USR1 systemd-journald** command.
4. The systemd journal is now persistent across reboots. If you want to see the log messages since last reboot, use **journalctl -b**.


```
#SystemMaxUse=
#SystemKeepFree=
#SystemMaxFileSize=
#RuntimeMaxUse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
#MaxRetentionSec=
#MaxFileSec=1month
#ForwardToSyslog=yes
#ForwardToKMsg=no
#ForwardToConsole=no
#TTYPath=/dev/console
#MaxLevelStore=debug
#MaxLevelSyslog=debug
#MaxLevelKMsg=notice
#MaxLevelConsole=info
```

Making the journal permanent is not hard to do. Exercise 13.4 shows how to proceed.

Exercise 13.4 Making the journald Journal Permanent

In this exercise, you learn how to make the journald journal permanent.

1. Open a root shell and type **mkdir /var/log/journal**.
2. Before journald can write the journal to this directory, you have to set ownership. Type **chown root:systemd-journal /var/log/journal**, followed by **chmod 2755 /var/log/journal**.
3. Next, you can either reboot your system (restarting the systemd-journald service is not enough) or use the **killall -USR1 systemd-journald** command.
4. The systemd journal is now persistent across reboots. If you want to see the log messages since last reboot, use **journalctl -b**.

C14-P379::MANAGING PARTITIONS

Montag, 30. Mai 2022 07:49

BIOS	Basic Input and output systems
MBR	MASTER BOOT REGISTER
GUID	Global Unique ID (128 bit number)
GPT	GUID Partition Table (Recent)
UEFI	Unified Extensible Firmware Interface (Replaces the BIOS and used for GPT)

Table 14.2 Disk Size Specifications

Symbol	Name	Value	Symbol	Name	Value
KB	Kilobyte	1000^1	KiB	Kibibyte	1024^1
MB	Megabyte	1000^2	MiB	Mebibyte	1024^2
GB	Gigabyte	1000^3	GiB	Gibibyte	1024^3
TB	Terabyte	1000^4	TiB	Tebibyte	1024^4
PB	Petabyte	1000^5	PiB	Pebibyte	1024^5
EB	Exabyte	1000^6	EiB	Exbibyte	1024^6
ZB	Zettabyte	1000^7	ZiB	Zebibyte	1024^7
YB	Yottabyte	1000^8	YiB	Yobibyte	1024^8

Table 14.3 Common Disk Device Types

Device Name	Description
/dev/sda	A hard disk that uses the SCSI driver. Used for SCSI and SATA disk devices. Common on physical servers but also in VMware virtual machines.
/dev/hda	The (legacy) IDE disk device type. You will seldom see this device type on modern computers.
/dev/vda	A disk in a KVM virtual machine that uses the virtio disk driver. This is the common disk device type for KVM virtual machines.
/dev/xvda	A disk in a Xen virtual machine that uses the Xen virtual disk driver. You see this when installing RHEL as a virtual machine in Xen. RHEL 7 cannot be used as a Xen hypervisor, but you might see RHEL 7 virtual machines on top of the Xen hypervisor using these disk types.

As you can see in Table 14.3, all disk device names end with the letter a. That is because it is the first disk that was found in your server. The second SCSI disk, for instance, would have the name /dev/sdb. If many disks are installed in a server, you can have up to /dev/sdz and even beyond. After /dev/sdz, the kernel continues creating devices with names like /dev/sdaa and /dev/sdab.

TIP In the end-of-chapter labs, you have to create partitions again. It will be a lot easier if at that point you can start from a clean installation. The following two steps help you to revert easily to a system on which you have not created any partitions yet.

1. Type **dd if=/dev/vda of=/root/diskfile bs=1M count=1**. (If your disk is /dev/sda and not /dev/vda, change the disk name accordingly.) Using this command allows you to create a backup of the first megabyte of raw blocks and write that to the file /root/diskfile. This file allows you to easily revert to the situation that existed at the start of this exercise.
2. Type **cp /etc/fstab /root/fstab** to make a backup of the /etc/fstab file as well.

At this point you are ready to start working on the exercise.

1. Open a root shell and run the **fdisk** command. This command needs the name of the disk device where you want to create the partition as its argument. In this exercise I'll use /dev/vda. Change if needed according to your hardware.

```
[root@localhost ~]# fdisk /dev/vda
Welcome to fdisk (util-linux 2.23.2).
```

   Changes will remain in memory only until you decide to write them. Be careful before using the **write** command.

2. Before doing anything, it is a good idea to check how much disk space you have available. Press **p** to see an overview of current disk allocation.

```
Command (m for help): p

Disk /dev/vda: 6442 MB, 6442450944 bytes, 12582912 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
Disk label type: dos
Disk identifier: 0x000a056b

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048      514047      256000   83  Linux
/dev/vda2          514048     8984575     4235264   8e  Linux LVM
```

   In the output of this command, in particular look for the total number of sectors and the last sector that is currently used (marked in bold in the above command output). If the last partition does not end on the last sector, you have available space to create a new partition.

3. Type **n** to add a new partition.

```
Command (m for help): n
Partition type:
   p   primary (2 primary, 0 extended, 2 free)
   e   extended
Select (default p):
```

4. Assuming you have a /dev/vda1 and a /dev/vda2 partition and nothing else, select **p** to create a primary partition. Accept the partition number that is now suggested.
5. Specify the first sector on disk that the new partition will start on. The first available sector is suggested by default, press Enter to accept.
6. Specify the last sector that the partition will end on. By default, the last sector available on disk is suggested. If you use that, after this exercise you will not have any disk space left to create additional partitions or logical volumes, so you should use another last sector. To use another last sector, you can do the following:
   - Enter the number of the last sector you want to use.
   - Enter **+number** to create a partition that sizes a specific number of sectors.
   - Enter **+number(K,M,G)** to specify the size you want to assign to the partition in KiB, MiB, or GiB.
   - Type **+100M** to make this a 100 MiB partition.

```
Partition number (3,4, default 3):
First sector (8984576-12582911, default 8984576):
Using default value 8984576
```

   - 82: Linux swap
   - 83: Linux
   - 8e: Linux LVM

   Press **Enter** to accept the default Linux partition type **83**.

8. If you are happy with the modifications, press **w** to write them to disk and exit fdisk. If you have created a partition on a disk that is already in use, you now see the following message:

```
Command (m for help): w
The partition table has been altered!
```

- Type **+100M** to make this a 100 MiB partition.

```
Partition number (3,4, default 3):
First sector (8984576-12582911, default 8984576):
Using default value 8984576
Last sector, +sectors or +size{K,M,G} (8984576-12582911,
default 12582911): +100M
Partition 3 of type Linux and of size 100 MiB is set
```

After you enter the partition's ending boundary, fdisk will show a confirmation.

7. At this point, you can define the partition type. By default, a Linux partition type is used. If you want the partition to be of any other partition type, use **t** to change it. Common partition types include the following:

exit fdisk. If you have created a partition on a disk that is already in use, you now see the following message:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16:
Device or resource busy.
The kernel still uses the old table. The new table will be used
at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@localhost ~]#
```

9. This message indicates that the partition has successfully been added to the partition table, but the in-memory kernel partition table could not be updated. You can see that by comparing the output of **fdisk -l /dev/vda** with the output of the command **cat /proc/partitions**, which shows the kernel partition table.
10. Type **partprobe /dev/vda** to write the changes to the kernel partition table. The partition has now been added and you can create a file system on it as described in the section "Creating File Systems."

NOTE You see the "re-reading the partition table failed with error 16" message only if you are adding partitions to a disk that already has some mounted partitions. If you are working on a new disk that does not have any mounted partitions, you will not see this error and you will not have to use the **partprobe** command.

Page 387 (330) Using Extended and Logical Partitions on MBR

Using Extended and Logical Partitions on MBR

In the previous procedure, you learned how to add a primary partition. If three partitions have been created already, there is room for one more primary partition, after which the partition table is completely filled up. If you want to go beyond four partitions on an MBR disk, you have to create an extended partition. Following that, you can create logical partitions within the extended partition.

Using logical partitions does allow you to go beyond the limitation of four partitions in the MBR; there is a disadvantage as well, though. All logical partitions exist within the extended partition. If something goes wrong with the extended partition, you have a problem with all logical partitions existing within it as well. If you need more than four separate storage allocation units, you might be better off using LVM instead of logical partitions.

NOTE An extended partition is only used for the purpose of creating logical partitions. You cannot create file systems directly on an extended partition!

Exercise 14.2 Creating Logical Partitions

1. To create a logical partition, when fdisk prompts which partition type you want to create, enter **e**.

```
Command (m for help): n
Partition type:
   p   primary (3 primary, 0 extended, 1 free)
   e   extended
Select (default e):
```

2. If the extended partition is the fourth partition that you are writing to the MBR, it will also be the last partition that can be added to the MBR. For that reason, it should fill the rest of your computer's hard disk. Press **Enter** to accept the default first sector and press **Enter** again when **fdisk** prompts for the last sector.

```
Using default response e
Selected partition 4
First sector (9189376-12582911, default 9189376):
Using default value 9189376
Last sector, +sectors or +size{K,M,G} (9189376-12582911, default
12582911):
Using default value 12582911
Partition 4 of type Extended and of size 1.6 GiB is set
```

3. Now that the extended partition has been created, you can create a logical partition within it. Still from the fdisk interface, press **n** again. The utility will prompt that all primary partitions are in use now and by default suggests adding a logical partition with partition number 5.

```
Command (m for help): n
All primary partitions are in use
Adding logical partition 5
```

4. Press **Enter** to accept the default first sector. When asked for the last sector, enter **+100M** (or any other size you want to use).

```
First sector (9191424-12582911, default 9191424):
Using default value 9191424
Last sector, +sectors or +size{K,M,G} (9191424-12582911, default
12582911): +100M
Partition 5 of type Linux and of size 100 MiB is set

Command (m for help):
```

5. Now that the logical partition has been created, enter **w** to write the changes to disk and quit fdisk. To complete the procedure, enter **partprobe** to update the kernel partition table. The new partition is now ready for use.

TIP The **fdisk** utility writes changes to disk only when you enter **w**, which is the **fdisk** write command. If you have made a mistake and want to get out, press **q** to quit.

Creating partition with GPT (GUID portable table)

NOTE fdisk has some support for managing GPT partitions also. At the time of this writing, the GPT support in fdisk is not stable. For that reason, it is recommended to use gdisk on GPT partitions and fdisk on MBR partitions.

WARNING! Do not ever use gdisk on a disk that has been formatted with fdisk and already contains fdisk partitions. Gdisk will detect that an MBR is present and it will convert this to a GPT (see the following code listing). Your computer will most likely not be able to boot after doing this!

```
[root@localhost ~]# gdisk /dev/vda
GPT fdisk (gdisk) version 0.8.6

Partition table scan:
  MBR: MBR only
```

NOTE fdisk has some support for managing GPT partitions also. At the time of this writing, the GPT support in fdisk is not stable. For that reason, it is recommended to use gdisk on GPT partitions and fdisk on MBR partitions.

WARNING! Do not ever use gdisk on a disk that has been formatted with fdisk and already contains fdisk partitions. Gdisk will detect that an MBR is present and it will convert this to a GPT (see the following code listing). Your computer will most likely not be able to boot after doing this!

```
[root@localhost ~]# gdisk /dev/vda
GPT fdisk (gdisk) version 0.8.6

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present


***************************************************************
Found invalid GPT and valid MBR; converting MBR to GPT format.
THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by typing 'q' if
you do not want to convert your MBR partitions to GPT format!
***************************************************************

Command (? for help):
```

To save you the hassle of going through this, I verified it does what it says. After converting an MBR to a GPT your machine will not start anymore.

Exercise 14.3 Creating GPT Partitions with gdisk

To apply the procedure in this exercise, you need a new disk device. Do *not* use a disk that contains data that you want to keep, because this exercise will delete all data on it. If you are using this exercise on a virtual machine, you may add the new disk through the virtualization software. If you are working on a physical machine, you can use a USB thumb drive as a disk device for this exercise. Note that this exercise works perfectly on a computer that starts from BIOS and not EFI, all you need is a dedicated disk device.

1. To create a partition with gdisk, type **gdisk /dev/vdb**. (Replace /dev/vdb with the exact device name used on your computer.) Gdisk will try to detect the current layout of the disk, and if nothing has been detected, it will create the GPT partition table and associated disk layout.

```
[root@localhost ~]# gdisk /dev/vdb
GPT fdisk (gdisk) version 0.8.6

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.

Command (? for help):
```

2. Type **n** to enter a new partition. You can choose any partition number between 1 and 128, but it is wise to accept the default partition number that is suggested.

```
Command (? for help): n
Partition number (1-128, default 1):
```

3. You now are asked to enter the first sector. By default, the first sector that is available on disk will be used, but you can specify an offset as well. This does not make sense at all, so just press **Enter** to accept the default first sector that is proposed.

```
First sector (34-2097118, default = 2048) or {+-}size{KMGTP}:
```

4. When asked for the last sector, by default the last sector that is available on disk is proposed (which would create a partition that fills the entire hard disk). You can specify a different last sector, or specify the disk size using +, the size, and KMGTP. So to create a 2 TiB disk partition, use **+2TiB**.

```
Last sector (2048-2097118, default = 2097118) or {+-}size{KMGTP}:
+100M
```

5. You now are asked to set the partition type. If you do not do anything, the partition type is set to 8300, which is the Linux file system partition type. Other options are available as well. You can press **l** to show a list of available partition types.

The relevant partitions types are as follows:

- **8200:** Linux swap
- **8300:** Linux file system
- **8e00:** Linux LVM

Notice that these are the same partition types as the ones that are used in MBR, with two 0s added to their names. You can also just press **Enter** to accept the default partition type 8300.

6. The partition is now created (but not yet written to disk). Press **p** to show an overview, which allows you to verify that this is really what you want to use.

```
Command (? for help): p
Disk /dev/vdb: 2097152 sectors, 1024.0 MiB
Logical sector size: 512 bytes
Disk identifier (GUID): 870DF067-6735-482E-83CE-5123E20509E0
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 2097118
Partitions will be aligned on 2048-sector boundaries
Total free space is 1892285 sectors (924.0 MiB)

Number  Start (sector)  End (sector)  Size       Code  Name
   1    2048            206847  100.0 MiB  8300  Linux filesystem
```

7. If you are satisfied with the current partitioning, press **w** to write changes to disk and commit. This gives a warning, after which the new partition table is written to the GUID partition table.

```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL
OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/vdb.
The operation has completed successfully.
```

8. If at this point you get an error message indicating that the partition table is in use, type **partprobe** to update the kernel partition table.

Pg.391 (334) Creating file system

Creating File Systems

At this point, you know how to create partitions. A partition all by itself is not very useful. It only becomes useful if you decide to do something with it. That often means that you have to put a file system on top of it. In this section, you learn how to do that.

Different file systems can be used on RHEL 7. Table 14.4 provides an overview of the most common file systems.

Table 14.4 File System Overview

File System	Description
XFS	The default file system in RHEL 7.
Ext4	The default file system in previous versions of RHEL. Still available and supported in RHEL 7.
Ext3	The previous version of Ext4. On RHEL 7, there is no real need to use Ext3 anymore.
Ext2	A very basic file system that was developed in the early 1990s. There is no need to use this file system on RHEL 7 anymore.
BtrFS	A relatively new file system that was not yet supported in RHEL 7.0 but will be included in later updates.
NTFS	Not supported on RHEL 7.
VFAT	A file system that offers compatibility with Windows and Mac, it is the functional equivalent of the FAT32 file system. Useful to use on USB thumb drives that are used to exchange data with other computers but not on a server's hard disks.

`$ mkfs -t xfs dev/vda2`	Format a partition using flag (-t) to specify file system type. -> input the file system -> input the partition to format
`$ mkfs.xfs /dev/vda3`	-> Alternative to above command
`$ tune2fs -l /dev/vda`	List the file system properties
`$ tune2fs -o /dev/vda1`	Set the file system mount options
`$ tune2fs -o acl,user_xattr`	Switch on access control list switch on user extended attributes
`$ tune2fs -o ^acl,user_xattr`	Switch of both acl and user extended attributes
`$ tune2fs -O {feature}`	To switch on file system feature place ^ infront of feature to switch it off
`$ tune2fs -L`	Set a label on the file system alternative command: **e2label**
`$ xfs_admin -L mylabel`	Used mainly for xfs file system to set file system label to mylabel

To change any of the default file system options, the **tune2fs** command enables you with other parameters. Some common usage examples are listed below:

- Use **tune2fs -o** to set default file system mount options. When set to the file system, the option does not have to be specified while mounting through /etc/fstab anymore. Use, for instance, **tune2fs -o acl,user_xattr** to switch on access control lists and user extended attributes. Use a ^ in front of the option to switch it off again, as in **tune2fs -o ^acl,user_xattr**.
- Ext file systems also come with file system features that may be enabled as a default. To switch on a file system feature, use **tune2fs -O** followed by the feature. To turn a feature off, use a ^ in front of the feature name.
- Use **tune2fs -L** to set a label on the file system. As described in the section "Mounting File Systems" later in this chapter, you can use a file system label to mount a file system based on its name instead of the device name. Instead of **tune2fs -L**, the **e2label** command enables you to do so.

Page 395 (338)

Adding Swap Partitions

The swap is a memory space allocation which can be allocated on disk device (partition or logical volume) or a file.
It is used to extend the computer memory usage when there is physical ram memory shortage.

Exercise 14.5 Creating a Swap Partition

1. Use **fdisk /dev/vda** to open your disk in fdisk. (Use gdisk if you are using a disk with a GUID partition table.)
2. Press **n** to add a new partition. Specify start and stop cylinders and size.
3. Type **t** to change the partition type. If you are using fdisk, use partition type 82. If you are using gdisk, use partition type 8200.
4. Use mkswap to format the partition as swap space. Use, for instance, **mkswap /dev/vda6** if the partition you have just created is /dev/vda6.
5. Type **free -m**. You see the amount of swap space that is currently allocated.
6. Use swapon to switch on the newly allocated swap space. If, for instance, the swap device you have just created is /dev/vda6, use **swapon /dev/vda6** to activate the swap space.
7. Type **free -m** again. You see that the new swap space has been added to your server.

Adding Swap Files

If you do not have free disk space to create a swap partition and you do need to add swap space urgently, you can use a swap file as well. From a performance perspective, it does not even make that much difference if a swap file is used instead of a swap device such as a partition or a logical volume, and it may help you fixing an urgent need in a timely manner.

To add a swap file, you need to create the file first. The **dd if=/dev/zero of=/swapfile bs=1M count=100** command would add 100 blocks with a size of 1 Mebibyte from the /dev/zero device (which generates 0s) to the /swapfile file. The result is a 100 MiB file that can be configured as swap. To do so, you can follow the same procedure as for swap partitions. First use **mkswap /swapfile** to mark the file as a swap file, after which you can use **swapon /swapfile** to active it.

Pg.397 (340) Mounting the File System

After a partition is created, it is required to be formatted with a file system and then mounted onto a file directory for ist contents to be accessible.

$ mount /dev/vda1 /mnt	Mounts the device (/dev/vda1) onto the directory (/mnt)
$ unmount /dev/vda1	Unmounts the device
$ unmount /mnt	Unmounts the directory. Alternatively unmounts the device as well.
$ blkid	Shows the overview of the current file system and UUID used by the file system
$ mount UUID="42f419c4-633f-4ed7-b161-519a4dadd3da" /mnt	Preferred alternative to mount device to directory
$ mount LABEL=mylabel /mnt	To mount device to directory temporarily

Mounts can be performed automatically by using the /etc/fstab file

Listing 14.4 Sample /etc/fstab file Contents

```
[root@server3 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Jan 16 10:28:41 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root /                       xfs     defaults        1 1
UUID=02305166-840d-4f74 /                       xfs     defaults        1 2
/dev/mapper/centos-swap swap                    swap    defaults        0 0
```

In the /etc/fstab file, everything is specified to mount the file system automatically. For this purpose, every line has six fields, as summarized in Table 14.5.

Table 14.5 /etc/fstab Fields

Field	Description
Device	The device that must be mounted. A device name, UUID, or label can be used.
Mount Point	The directory or kernel interface where the device needs to be mounted.
File System	The file system type.
Mount Options	Mount options.
Dump Support	Use 1 to enable support to backup using the dump utility. This may be necessary for some backup solutions.
Automatic Check	Specifies if the file system should be checked automatically when booting. Use 0 to disable automated check, 1 if this is the root file system and it has to be checked automatically, and 2 for all other file systems that need automatic checking while booting. Network file systems should have this option set to 0.

Based on what has previously been discussed about the **mount** command, you should have no problem understanding the device, mount point, and file system fields in /etc/fstab. Notice that in the mount point not all file systems use a directory name. Some system devices such as swap are not mounted on a directory, but on a kernel interface. It is easy to recognize when a kernel interface is used; its name does not start with a / (and does not exist in the file system on your server).

The Mount Options field defines specific mount options that can be used. If no specific options are required, this line will just read "defaults." To offer specific functionality, a large number of mount options can be specified here. Table 14.6 gives an overview of some of the more common mount options.

Table 14.6 Common Mount Options

Key Topic

Option	Use
auto/ noauto	The file system will [not] be mounted automatically.
acl	Adds support for file system access control lists (see Chapter 7, "Configuring Permissions").
user_xattr	Add support for user extended attributes (see Chapter 7).
ro	mounts the file system in read-only mode.
atime / noatime	Disables or enables access time modifications.
noexec / exec	Denies or allows execution of program files from the file system.
_netdev	Use this to mount a network file system. This tells fstab to wait until the network is available before mounting this file system.

The fifth column of /etc/fstab specifies support for the dump utility. This is a utility that was developed to create file system backups. It is good practice to switch this feature on by specifying a 1 for all real file systems, and switch it off by specifying 0 for all system mounts.

The last column indicates if the file system integrity needs to be checked while booting. Put a 0 if you do not want to check the file system at all, a 1 if this is the root file system which needs to be checked before anything else, and a 2 if this is a nonroot file system that needs to be checked while booting.

WARNING If a file system through /etc/fstab is flagged for automatic file system check and something prevents the file system to be checked correctly, your system stops booting and prompts "enter root password to enter maintenance mode." To prevent this from ever happening, you could choose to disable automated checks while booting. See Chapter 19, "Troubleshooting the Boot Procedure," for more information on how to fix this specific case.

Exercise 14.6 Mounting Partitions Through /etc/fstab

In this exercise, you mount the XFS formatted partition /dev/vda5 that you have created in previous exercises.

1. From a root shell, type **blkid**. Use the mouse to copy the UUID="nnnn" part for /dev/vda5.
2. Type **mkdir -p /mounts/data** to create a mount point for this partition.
3. Open /etc/fstab in an editor and add the following line:

```
UUID="nnnn"         /mounts/data        xfs         defaults  1 2
```

4. Before attempting an automatic mount while rebooting, it is a good idea to test the configuration. Type **mount -a**. This mounts everything that is specified in /etc/fstab and that has not been mounted already.
5. Type **df -h** to verify that the partition has been mounted correctly.

Summary

In this important chapter, you learned how to work with partitions and file systems on RHEL 7. You learned how to create partitions for MBR and GPT disks, and how to put a file system on top of the partition. You also learned how to mount these partitions manually and automatically through /etc/fstab.

Exam Preparation Tasks

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 14.7 lists a reference of these key topics and the page numbers on which each is found.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 14.7 lists a reference of these key topics and the page numbers on which each is found.

C15-P409::MANAGING LVM LOGICAL VOLUMES

Montag, 20. Juni 2022 07:13

The use of logical volume provides flexibility to storage Resources. Hence, the possibility to dynamically grow a partition That is running out of disk space.

Advantages of LVM

- Flexible disk sizing - ease of resizing the harddisk size
- Snapshots support - by copying the logical volume administrative data (metadata) that describe the current state of files to a snapshot volume.
- Offers the ease of replacing failing hardware - by using the **pvmove** utility command, data can be moved within the volume group as backup for the failed hardware to be replaced.

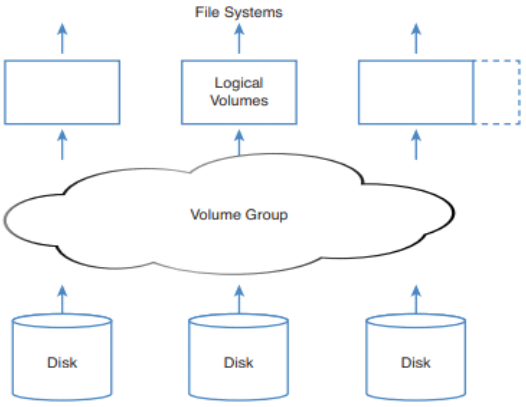


Figure 15.1 LVM architecture overview.

Creating the LV

Requires the three layers as shown above to be created

- Physical volume (PV)
- Volume group (VG)
- Logical volume (LV)

Page 413 - Creating the physical volume

Creating the Physical Volumes

Before the LVM tools can be used to create physical volumes, you need to create a partition marked as the LVM partition type. This is basically the same procedure as described in the preceding chapter, with the only difference that before writing changes to disk in fdisk or gdisk, you need to press **t** to change the partition type. (Exercise 15.1 shows exactly what you need to do.) If you are using an MBR disk, the partition type is 8e. If you are using a GUID disk, use the partition type 8300.

After creating the partition and flagging it as an LVM partition type, you need to use **pvcreate** to mark it as a physical volume. This writes some metadata to the partition, which allows it to be used in a volume group. The entire procedure is summarized in Exercise 15.1.

Exercise 15.1 Creating the Physical Volume

In this exercise, you create a physical volume. To do this exercise, you need a hard disk that has free (unpartitioned) disk space available. The recommended method to make disk space available is by adding a new hard disk in your virtual machine environment. In this exercise, I use a clean /dev/vdb device to create the partition. You may have to change the device name to match your configuration. If you do not have a dedicated hard disk available to create this configuration, you might want to consider attaching a USB key to your machine.

1. Open a root shell and type **fdisk /dev/vdb**.
2. Type **n** to create a new partition. Select **p** to make it a primary partition, and use the partition number that is suggested as a default. If you are using a clean device, this will be partition number 1.
3. Press **Enter** when asked for the first sector and type **+100M** to accept the last sector.
4. Once you are back on the fdisk prompt, type **t** to change the partition type. Because there is one partition only, fdisk does not ask which partition to use this partition type on. You may have to select a partition if you are using a different configuration.
5. The partitioner asks for the partition type you want to use. Type **8e**. Then, press **w** to write changes to disk and quit fdisk. Listing 15.2 shows an overview of all commands that have been used so far. If you are getting a message that the partition table could not be updated while writing the changes to disk, reboot your system.

Listing 15.2 Creating an LVM Partition in fdisk

```
[root@localhost ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xe39ca22b.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-2097151, default 2097151):
+100M
Partition 1 of type Linux and of size 100 MiB is set

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

6. Now that the partition has been created, you need to flag it as an LVM physical volume. To do this, type **pvcreate /dev/vdb1**. You should now get this prompt: Physical volume "/dev/vbd1" successfully created.
7. Now type **pvs** to verify that the physical volume has been created successfully. The output may look like Listing 15.3. Notice that in this listing another physical volume already exists; that is because RHEL uses LVM by default to organize storage.

Listing 15.3 Verifying the Physical Volume

```
[root@localhost ~]# pvs
  PV         VG     Fmt  Attr PSize   PFree
  /dev/vda2  centos lvm2 a--    3.51g      0
  /dev/vdb1         lvm2 a--  100.00m 100.00m
```

As an alternative to the **pvs** command, which shows a summary of the physical volumes and their attributes, you can also use the **pvdisplay** command to show some more details. Listing 15.4 shows an example of the output of this command.

Listing 15.4 Example **pvdisplay** Command Output

```
[root@server1 ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/vda2
  VG Name               centos
  PV Size               3.51 GiB / not usable 3.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              898
  Free PE               0
  Allocated PE          898
  PV UUID               CILii7-DzOd-w4L0-yOxi-9NXg-D3nP-ZugJIj
```

If you want a very synthetic overview of the current storage configuration, you might also like the **lsblk** command. As can be seen in Listing 15.5, this command gives a hierarchical overview of which disks and partitions are used in what LVM volume groups and logical volumes.

Listing 15.5 Use **lsblk** for a Synthetic Overview of the Current Configuration of Storage on Your Server

```
[root@localhost ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0                   2:0    1    4K  0 disk
sda                   8:0    0    8G  0 disk
├─sda1                8:1    0  200M  0 part /boot
├─sda2                8:2    0  6.9G  0 part
| ├─centos-swap     253:0    0  256M  0 lvm  [SWAP]
| └─centos-root     253:1    0  5.9G  0 lvm  /
├─sda3                8:3    0  100M  0 part
└─sda4                8:4    0  887M  0 part
  └─vgik-lvgroups   253:2    0  440M  0 lvm  /groups
sr0                  11:0    1 1024M  0 rom
```

Page 416 - Creating the Volume groups

C16-P429::BASIC KERNEL MANAGEMENT

Dienstag, 5. Juli 2022 07:41

Understanding the Role of the Linux Kernel

The Linux kernel is the heart of the operating system. It is the layer between the user who works with Linux from a shell environment and the hardware that is available in the computer on which the user is working. The kernel is doing so by managing the I/O instructions it receives from the software and translating those to processing instructions that are to be executed by the central processing unit and other hardware in the computer. The kernel also takes care of handling essential operating system tasks. One example of such a task is the scheduler that makes sure that processes that are started on the operating system are handled by the CPU.

Understanding the use of Kernel Threads and Drivers

The operating system tasks that are performed by the kernel are implemented by different kernel threads. Kernel threads are easily recognized with a command like **ps aux**. The kernel thread names are listed between square brackets (see Listing 16.1).

Understanding Hardware Initialization

The loading of drivers is an automated process that roughly goes like this:

1. During boot, the kernel probes available hardware.
2. Upon detection of a hardware component, the **systemd-udevd** process takes care of loading the appropriate driver and making the hardware device available.
3. To decide how the devices are initialized, systemd-udevd reads rules files in /usr/lib/udev/rules.d. These are system provided udev rules files that should not be modified.
4. After processing the system provided udev rules files, systemd-udevd goes to the /etc/udev/rules.d directory to read any custom rules if these are available.
5. As a result, required kernel modules are loaded automatically and status about the kernel modules and associated hardware is written to the sysfs file system which is mounted on the /sys directory.

The systemd-udevd process is not a one-time only process; it continuously monitors plugging and unplugging of new hardware devices. To get an impression of how this works, as root you can type the command **udevadm monitor**. This is all events that are processed while activating new hardware devices. Use Ctrl+C to close the udevadm monitor output.

Listing 16.4 shows output of the **udevadm monitor** command. In this command, you can see how features that are offered by the hardware are discovered automatically by the kernel and udev working together. Each phase of the hardware probing is concluded by the creation of a file in the /sys file system. Once the hardware has been fully initialized, you can also see that some kernel modules are loaded.

Analyzing What the Kernel Is Doing

To help analyze what the kernel is doing, some tools are provided by the Linux operating systems:

- The **dmesg** utility
- The /proc file system
- The uname utility

The first utility to consider whether detailed information about the kernel activity is required is dmesg. This utility shows the contents of the kernel ring buffer, an area of memory where the Linux kernel keeps its recent log messages. An alternative method to get access to the same information in the kernel ring buffer is by using the **journalctl --dmesg** command, which is equivalent to **journalctl -k**. In Listing 16.2, you can see a part of the result of the **dmesg** command.

Another valuable source of information is the /proc file system. The /proc file system is an interface to the Linux kernel, and it contains files with detailed actual status information on what is happening on your server. Many of the performance-related tools mine the /proc file system for more information.

As an administrator, you will find that some of the files in /proc are very readable and contain actual status information about CPU, memory, mounts, and more. Take a look, for instance, at /proc/meminfo, which gives detailed information about each memory segment and what exactly is happening in these memory segments.

A last useful source of information that should be mentioned here is the **uname** command. This command gives different kinds of information about your operating system. Type, for instance, **uname -a** for an overview of all relevant parameters of **uname -r** to see which kernel version currently is used. This information also shows when using the **hostnamectl status** command.

Managing Kernel Modules

Linux kernel modules normally are loaded automatically for the devices that need them, but you will sometimes have to load the appropriate kernel modules manually. A few commands are used for manual management of kernel modules. Table 16.2 provides an overview.

An alternative method of loading kernel modules is by doing this through the /etc/modules-load.d directory. In this directory, you can create files to load modules automatically that are not loaded by the udev method already.

Table 16.2 Linux Kernel Module Management Overview

Command	Use
lsmod	Lists currently loaded kernel modules
modinfo	Displays information about kernel modules
modprobe	Loads kernel modules, including all of their dependencies
modprobe -r	Unloads kernel modules, considering kernel module dependencies

The first command to use when working with kernel modules is **lsmod**. This command lists all kernel modules that currently are used, including the modules by which this specific module is used. Listing 16.5 shows the output of the first 10 lines of the **lsmod** command.

To manually load and unload modules, you can use the **modprobe** and **modprobe -r** commands. On earlier Linux versions, you may have used the **insmod** and **rmmod** commands. These should be used no longer because they do not consider kernel module dependencies. In Exercise 16.1, you learn how to manage kernel modules using these commands.

Exercise 16.1 Managing Kernel Modules from the Command Line

In this exercise, you work with the basic commands that are used for managing Linux kernel modules from the command line.

1. Open a root shell and type **lsmod | head**. This shows all kernel modules currently loaded.
2. Type **modprobe ext4** to load the ext4 kernel module. Verify that it is loaded, using the **lsmod** command again.
3. Type **modinfo ext4** to get information about the ext4 kernel module. Notice that it does not have any parameters.
4. Type **modprobe -r ext4** to unload the ext4 kernel module again.
5. Type **modprobe -r xfs** to try to unload the xfs kernel module. Notice that you get an error message as the kernel module currently is in use.

6. Type **dmesg**. For some kernel module, load information is written to the kernel ring buffer which can be displayed using the **dmesg** command. Unfortunately this is not the case for the cdrom kernel module.
7. Create a file with the name /etc/modprobe.d/cdrom and give it the following contents:

```
options cdrom debug=1
```

This will enable the parameter every time the cdrom kernel module will be loaded.

Updating the kernel

`$ yum install kernel`	`Updates the kernel`
`$ yum upgrade kernel`	`Same as above`

C17-P446::CONFIGURE BASIC APACHE SERVER

Montag, 18. Juli 2022 16:08

3 basic steps are required

1. Installing the required software

$ yum search http	Gives lots of packages
$ yum install httpd	Install the base package
$ yum groups list	Gives overview of all yum groups available
$ yum groups install "Basic Web Server"	Install the Apache web server and it core requirements

2. Identify the Main configuration file

/etc/httpd/conf/httpd.conf	The main Apache configuration file
DocumentRoot	This is the most important parameter- it specifies the default location where the Apache web server looks for ist contents
ServerRoot	Another important parameter. it specifies the default location where the Apache web server looks for ist configuration files /etc/httpd

3. Create some web server content

By default the web server will look for files in folder.

/var/www/html	Default folder for web server contents

Exercise 17.1 Setting Up a Basic Web Server

In this exercise, you learn how to set up a basic Apache web server. Nothing fancy, just enough to get you going and test web server functionality.

1. Type y**um groups install "Basic Web Server"**. This installs the httpd package, and some of the most commonly used additional packages as well.
2. Open the main Apache configuration file with an editor, and look up the line that starts with DocumentRoot. This identifies the location where the Apache server will look for the contents it will service. Confirm that it is set to /var/www/html.
3. In the directory /var/www/html, create a file with the name index.html. In this file, type **"Welcome to my web server"**.
4. To start and enable the web server, type **systemctl start httpd; systemctl enable httpd**. This starts the web server and makes sure that it starts automatically after restarting the server. Use **systemctl status httpd** to check that the web server is up and running. In Listing 17.2 you can see what the result of this command should look like.
5. Type **yum install elinks** to install the elinks text-based browser. Type **elinks http://localhost** to connect to the web server and verify it is working.

Creating Apache Virtual Hosts

Many companies host more than one website. Fortunately, it is not necessary to install a new Apache server for every website that you want to run. Apache can be configured to work with virtual hosts. A virtual host is a distinguished Apache configuration file that is created for a unique hostname. When working with virtual hosts, the procedure to access the host is roughly like the following:

1. The client starts a session to a specific virtual host, normally by starting a browser and entering the URL to the website the client wants to use.
2. DNS helps resolving the IP address of the virtual host, which is the IP address of the Apache server that can host different virtual hosts.
3. The Apache process receives requests for all the virtual hosts it is hosting.
4. The Apache process reads the HTTP header to analyze which virtual host this request needs to be forwarded to.
5. Apache reads the specific virtual host configuration file to find which document root is used by this specific virtual host.
6. The request is forwarded to the appropriate contents file in that specific document root.

When working with virtual hosts, there are a few things to be aware of:

- If your Apache server is configured for virtual hosts, all servers it is hosting should be handled by virtual hosts. To create a catch all entry for all HTTP requests that are directed to this host but that do not have a specific virtual host file, you can create a virtual host for _default_:80.
- Name-based virtual hosting is the most common solution. In this solution, virtual hosts are using different names but the same IP address.
- IP-based virtual hosts are less common, but are required if the name of a web server must be resolved to a unique IP address. IP-based virtual hosts do require several IP addresses on the same machine and are common in configuration where the Apache server uses TLS to secure connections.

TIP Configuring virtual hosts is not an RHCSA objective, but it is useful to know how to configure them anyway. If you are preparing for the RHCE exam, you absolutely do need to know how to configure virtual hosts. Exercise 17.2 walks you through the procedure. If you are interested in RHCSA exam-related contents only, you are welcome to skip this exercise.

Exercise 17.2 Installing Apache Virtual Hosts

In this exercise, you create two virtual hosts. To help you setting up virtual hosts, you first set up name resolution, after which you create the virtual hosts configuration as well. Because SELinux has not been discussed yet, you temporarily switch off SELinux.

NOTE I later tell you that you should never switch off SELinux. For once, I make an exception to this important security rule. To focus on what needs to be done on the Apache web server, it is easier to focus just on Apache and not to configure SELinux as well.

1. On both server1 and server2, open the file /etc/hosts with an editor and add two lines that make it possible to resolve the names of the virtual host you are going to create to the IP address of the virtual machine:

```
192.168.122.210    server1.example.com          server1
192.168.122.220    server2.example.com          server2
192.168.122.210    account.example.com          account
12.168.122.210     sales.example.com            sales
```

2. On server1, open a root shell and add the following to the /etc/httpd/conf/httpd.conf file. (You can leave all other settings as they are.)

```
<Directory/www/docs>
        Required all granted
        AllowOverride None
</Directory>
```

3. On server1, open a root shell and create a configuration file with the name **account.example.com.conf** in the directory /etc/httpd/conf.d. Give this file the following content:

```
<VirtualHost *:80>
        ServerAdmin webmaster@account.example.com
        DocumentRoot /www/docs/account.example.com
        ServerName account.example.com
        ErrorLog logs/account/example.com-error_log
        CustomLog logs/account.example.com-access_log common
</VirtualHost>
```

4. Close the configuration file and from the root shell use **mkdir -p /www/docs/account.example.com**.
5. Create a file with the name index.html in the account document root, and make sure its contents read "Welcome to account."
6. Temporarily switch off SELinux using **setenforce 0**.
7. Use **systemctl restart httpd** to restart the Apache web server.
8. Use elinks http://account.example.com. You should now see the account welcome page. (You may have to install elinks, using **yum install -y elinks**.)
9. Back on the root shell, copy the /etc/httpd/conf.d/account.example.com.conf file to a file with the name /etc/httpd/conf.d/sales.example.com.conf.
10. Open the sales.example.com.conf file in vi, and use the vi command **:%s/account/sales/g**. This should replace all instances of account with the text sales.
11. Create the /www/docs/sales.example.com document root, and create a file index.html in it, containing the text "Welcome to the sales server."
12. Restart httpd and verify that the account and the sales servers are both accessible.

Summary

In this chapter, you learned about Apache basics. The information in this chapter helps you configure a basic Apache web server, which helps testing advanced topics like firewall configuration or SELinux configuration that are covered in later chapters in this book.

C18-P465::MANAGING AND UNDERSTANDING THE BOOT PROCEDURE

Dienstag, 2. August 2022 07:22

PG.465 Working with Systemd
The systemd system is used to manage units (eg. services)
Services are processes that provide specific functionality
and allow connections from external clients coming in.

$ systemctl -t help	List all available unit types

/usr/lib/systemd/system	System default unit files
/etc/systemd/system	System specific modification (overrides the defaults)
/run/systemd/system	Runtime configuration generated automatically are stored here

Each systemd service unit file consists of at least three sections

Unit	Describes the unit and define dependencies it contains sometimes the following statement **After:** list unit to start after the unit starts **Before:**list unit to start before the unit starts **Conflicts:** list units that cannot be used together
Service	Describes how to start, stop the service and request status.
install	Here, the wants are taken care of. **WantedBy:** defines where the unit has to be started

See page 467 for more examples for UNITS - mount, socket

Listing 18.2 Example of the Vsftpd Unit File

```
[Unit]
Description=Vsftpd ftp daemon
After=network.target

[Service]
Type=forking
ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

[Install]
WantedBy=multi-user.target
```

$ systemctl show {unit} $ systemctl show sshd	Shows all the options available for configuring a specific unit

PG.469 Understanding Target Units
The target unit is a group of units used to ensure the unit files are loaded in the right order and in the right moment.
The target contains wants.

$ cd /etc/systemd/system/…	Stores subdirectories of various targets

PG.470 Understanding Target Units
Wants : Define which units systemd wants when starting a specific target. They are created when systemd units are enabled and are stored as symbolic links within the target directory

$ cd /etc/systemd/system/basic.target.wants/…	Stores the wants as symbolic link to other target

PG.470 Managing Units with systemctl

TIP Memorizing all the different arguments that can be used with the **systemctl** command might seem hard, but you don't have to do that. Instead, just type **systemctl** and press the **Tab** key twice to use command autocompletion. This will show you all available commands.

Exercise 18.1 Managing Units with systemctl

1. Type **yum -y install vsftpd** to install the Very Secure FTP service.
2. Type **systemctl start vsftpd**. This activates the FTP server on your machine.
3. Type **systemctl status vsftpd**. You'll get an output as in Listing 18.7 and see that the vsftpd service is currently operational. You can also see in the Loaded line that it is currently disabled, which means that it will not be activated on a system restart.
4. Type **systemctl enable vsftpd**. This creates a symbolic link in the wants directory for the multi-user target to ensure that the service gets back after a restart.
5. Type **systemctl status vsftpd** again. You'll now see that the unit file has changed from being disabled to enabled.

Listing 18.7 Requesting Current Unit Status with **systemctl status**

```
[root@server202 system]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled)
   Active: active (running) since Sun 2014-09-28 08:42:59 EDT; 2s ago
  Process: 34468 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
(code=exited, status=0/SUCCESS)
 Main PID: 34469 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─34469 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 08:42:59 server202.example.com systemd[1]: Starting Vsftpd ftp
daemon...
Sep 28 08:42:59 server202.example.com systemd[1]: Started Vsftpd ftp
daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

Table 18.3 Systemctl Unit Overview Commands

Command	Description
systemctl --type=service	Shows only service units
systemctl list-units --type=service	Shows all active service units (same result as the previous command)
systemctl list-units --type=service --all	Shows inactive service units as well as active service units
systemctl --failed --type=service	Shows all services that have failed
systemctl status -l your.service	Shows detailed status information about services

`$ systemctl list-dependencies {unit}` `$ systemctl list-dependencies vsftpd`	List units that the unit depends on for it service to start
`$ systemctl list-dependencies --reverse {unit}`	List out the units that are dependent on this unit for their service to start

Page 474. Managing Target conflicts

Some units have conflict with other units, hence they cannot be started at the same time.(iptables & firewalld, mount & unmount). To avoid such case from occuring. Units can be masked from being loaded as follows

`$ systemctl mask {unit}`	Mask a unit, mainly to avoid conflict with another
`$ systemctl mask iptables`	Same as above. (example for command above)

C19-P489::UNDERSTANDING THE BOOT PROCEDURE

Donnerstag, 25. August 2022 07:37

Table 19.2 Boot Phase Configuration and Troubleshooting Overview

Boot Phase	Configuring It	Fixing It
POST	Hardware configuration (F2, Esc, F10, or another key)	Replace hardware.
Selecting the bootable device	BIOS/UEFI configuration or hardware boot menu	Replace hardware or use rescue system.
Loading the boot loader	**grub2-install** and edits to /etc/defaults/grub	GRUB boot prompt and edits to /etc/defaults/grub, followed by grub2-mkconfig.
Loading the kernel	Edits to the GRUB configuration and /etc/dracut.conf.	GRUB boot prompt and edits to /etc/defaults/grub, followed by grub2-mkconfig.
Starting /sbin/init	Compiled into initramfs	**init= kernel** boot argument, **rd.break** kernel boot argument.
Processing initrd.target	Compiled into initramfs	Not typically required.
Switch to the root file system	/etc/fstab	/etc/fstab.
Running the default target	/etc/systemd/system/default.target	Start the rescue.target as a kernel boot argument.

Passing Kernel Boot Arguments

If your server does not boot normally, the GRUB boot prompt offers a convenient way to stop the boot procedure and pass specific options to the kernel while booting. In this section, you learn how to access the boot prompt and how to pass specific boot arguments to the kernel while booting.

Accessing the Boot Prompt

When your server boots, you briefly see the GRUB 2 menu. Look fast because it will only last for a few seconds. From this boot menu you can type **e** to enter a mode where you can edit commands, or **c** to enter a full GRUB command prompt, as shown in Figure 19.1. To pass boot options to a starting kernel, use **e**.

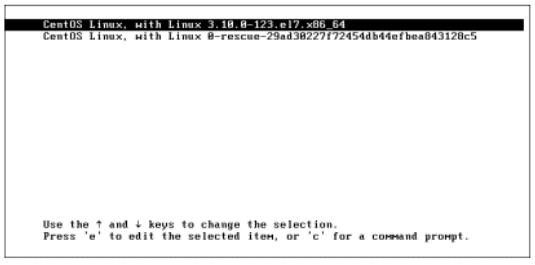


Figure 19.1 Entering the GRUB boot prompt.

After passing an **e** to the GRUB boot menu, you'll see the interface that is in Figure 19.2. From this interface, scroll down to locate the section that begins with linux16 /vmlinuz followed by a lot of arguments. This is the line that tells GRUB how to start a kernel, and by default it looks like this:

```
linux16 /vmlinuz-0-rescue-5dea58df1a3b4cb5947ddb6c78a6773f
root=UUID=432d640e-3339-45fa-a66d-89da9c869550 ro rd.lvm.lv=centos/
swap vconsole.font=latarcyrheb-sun16 rd.lvm.lv=centos/root
crashkernel=auto  vconsole.keymap=us rhgb quiet
```

To start, it is a good idea to remove the **rhgb** and **quiet** parts from this line; these arguments hide boot messages for you, and typically you do want to see what is happening while booting. In the next section you learn about some troubleshooting options that you can enter from the GRUB boot prompt.

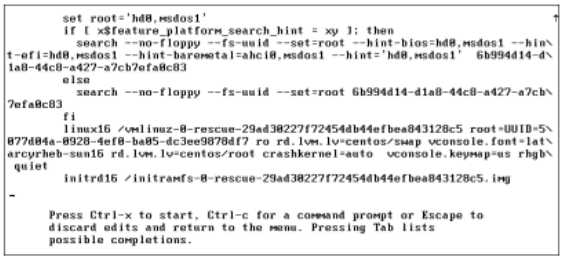


Figure 19.2 Enter boot arguments on the line that starts with linux16.

After entering the boot options you want to use, press **Ctrl+X** to start the kernel with these options. Notice that these options are used one time only and are not persistent. To make them persistent you must modify the contents of the /etc/default/grub configuration file and use **grub2-mkconfig -o /boot/grub2/grub.cfg** to apply the modification.

Pg.496 Reinstalling GRUB using Rescue disk

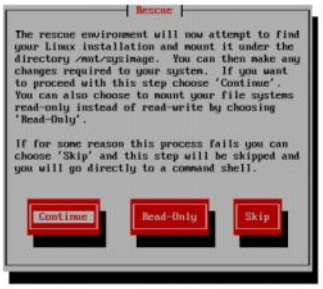
When the bootloader (GRUB 2) is broken, the screen does not show the boot option. The rescue disk can be used to restore it.

Note that recovery process must have been initiated to access the /mnt/sysimage directory. **Pg.493**

$ chroot /mnt/sysimage	Change to the sysimage dir and execute the next command
$ grub2-install /dev/sda	Install grub on physical server
$ grub2-install /dev/vda	Install grub on KVM virtual machine

Exercise 19.2 Using the Rescue Option

1. Restart your server from the installation disk. Select the **Troubleshooting** menu option.
2. From the Troubleshooting menu, select **Rescue a Red Hat System**. This prompts you to press **Enter** to start the installation. Do not worry: This option does not overwrite your current configuration; it just loads a rescue system.
3. The rescue system now prompts you that it will try to find an installed Linux system and mount on /mnt/sysimage. Press **Continue** to accept this option (see Figure 19.4).



	server
$ grub2-install /dev/vda	Install grub on KVM virtual machine

Figure 19.4 The rescue system looks for an installed system image and mount it for you.

4. If a valid Red Hat installation was found, you are prompted that your system has been mounted under /mnt/sysimage. At this point, you can press **Enter** twice to access the rescue shell.
5. Your Linux installation at this point is accessible through the /mnt/sysimage directory. Type **chroot /mnt/sysimage**. At this point, you have access to your root file system and you can access all tools that you need to repair access to your system.
6. Type **exit** and **reboot** to restart your machine in a normal mode.

Pg.496 Recreating the Initramfs using Rescue disk

Server cannot boot into normal operational mode when the initramfs is damaged.
To repair boot into the rescue environment **see Pg.493**

$ dracut	Creates new initramfs for the kernel currently loaded
$ dracut --force	Creates new initramfs and overwrite existing one
$ /usr/lib/dracut/dracut.conf.d/*.conf	Contains the system default configuration files
/etc/dracut.conf.d	Contains custom dracut configuration files
/etc/dracut.conf	Is used as the master configuration files

Pg.500 Recovering from FILE SYSTEM issues

This occurs due to misconfiguration in the file system file /etc/fstab
Case as below could occur
Device referred to by the fstab file does not exist
Error in the UUID that is used to mount the device.
During startup, there would be a prompt "Give root password for maintenance" if the cases exist.

Give password	Type in password when prompt appears
$ journalctl -xb	Type the command to see if there is relevant messages that provides information about what went wrong.
$ mount -o remount, rw	Type command to ensure the root file system is mounted read-only and analyze what is wrong in the /etc/fstab file and fix it.

Pg.500 Resetting the Root Password

To recover the root password the os must boot into minimal mode. (minimal mode allows you to login without a password)

so, follow these steps:

1. On system boot, press **e** when the GRUB 2 boot menu is shown.
2. Enter **rd.break** as boot argument to the line that loads the kernel and press **Ctrl+X** to boot with this option.
3. You'll now be dropped at the end of the boot stage where initramfs is loaded, just before a mount of the root file system on the directory /.
4. Type **mount -o remount,rw /sysroot** to get read/write access to the system image.
5. At this point, make the contents of the /sysimage directory your new root directory by typing **chroot /sysroot**.
6. Now you can enter **passwd** and set the new password for the user root.
7. Because at this very early boot stage SELinux has not been activated yet, the context type on /etc/shadow will be messed up. If you reboot at this point, no one will be able to log in. So you must make sure that the context type is set correctly. To do this, at this point you should load the SELinux policy by using **load_policy -i**.

8. Now you can manually set the correct context type to /etc/shadow. To do this, type **chcon -t shadow_t /etc/shadow**.
9. Reboot. You can now log in with the changed password for user root.

NOTE In the preceding procedure you have read how to use the **load_policy -i** and **chcon** commands to correct the labels on the /etc/shadow file. An alternative (and easier) method is to create a file with the name /.autorelabel which will force SELinux to restore labels that are set on the entire file system.

C20-P511::xxx_USING KICKSTART_SERVER INSTALLATION

Donnerstag, 8. September 2022 08:31

The Kickstart is a configuration file needed by an installer to specify exactly how an installation is to be performed.

It requires and installation to provide access to the repository to the installation files.

It requires a configured PXE boot server to provide access to the boot image for automating the installation process.

- Configure a network server as an installation server

Exercise 20.1 Setting Up the Network Installation Server

In this exercise, you set up the network installation server by copying over all files required for installation to a directory that is offered by an HTTP server. After doing this, you test the installation from a virtual machine. To perform this exercise, you need the account.example.com virtual Apache web server that you created in Chapter 17, "Configuring a Basic Apache Server."

1. Insert the Red Hat Enterprise Linux installation DVD in the optical drive of your server and navigate to the Packages directory on the installation disk.
2. Use **mkdir /www/docs/account.example.com/install** to create a subdirectory in the Apache document root for account.example.com.
3. Use **cp -R * /www/docs/account.example.com/install** from the directory where the Red Hat Enterprise Linux installation DVD is mounted to copy all files on the DVD to the install directory in your web server document root.
4. Modify the configuration file for the server1 virtual host in /etc/httpd/conf.d/account.example.com and make sure that it includes the line Options Indexes. Without this line, the virtual host will only show contents of a directory if it contains an index.html file.
5. Use service httpd restart to restart the Apache web server.
6. Start a browser and browse to http://account.example.com/install. You should now see the contents of the installation DVD.
7. Start Virtual Machine Manager and create a new virtual machine. Give the virtual machine the name **testnetinstall** and select Network Install when asked how to install the operating system.
8. When asked for the installation URL, enter http://account.example.com/install (and verify that this URL can be resolved). The installation should now be started.
9. You can now interrupt the installation procedure and remove the virtual machine. You have now seen that the installation server is operational, and it is time to move on to the next phase in the procedure.

- Set up a TFTP and DHCP server for PXE Boot

C21-P533::MANAGING SELinux

Montag, 26. September 2022 11:07

If SELinux is enabled and nothing else has been configured, all system calls are denied. To specify what exactly is allowed, a policy is used.
In this policy, rules define which source domain is allowed to access which target domain

Table 21.2 SELinux Core Elements

Element	Use
Policy	A collection of rules that define which source has access to which target.
Source domain	The object that is trying to access a target. Typically a user or a process.
Target domain	The thing that a source domain is trying to access. Typically a file or port.
Context	A security label that is used to categorize objects in SELinux.
Rule	A specific part of the policy that determines which source domain has which access permissions to which target domain.
Labels	Same as context label, defined to determine which source domain has access to which target domain.

SELinux is interwoven with the kernel hence, changing mode from enabled to disabled requires a reboot.

ENABLED	enforcing	Fully operation and enforces all SELinux rules
ENABLED	permissive	All SELinux activities is blocked but no access is blocked. useful
DISABLED	-	-

```
[root@server1 ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected
processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

/etc/sysconfig/selinux	This file is read during the booting to detect the operating mode of selinux. The mode can be modified in the file accordingly
$getenforce	This command checks the mode state (enforcing or permissive)
$setenforce 1	Sets the selinux in enforcing mode
$setenforce 0	Sets the selinux in permissive mode
$sestatus -v	Shows detailed information about current selinux status. This includes policy version, context labels

Pg.538 Understanding Context Settings and the Policy.

The context is a label that is applied to different elements like.

- Files and directories
- Port
- Processes
- Users

SELinux rules are created to match define context labels of source objects to that of target objects.

Option -Z shows the current context label settings

$ ls -Z	Show the current context label for files and directories
$ ps Zaux	Show the current context label settings for all processes
$ netstat Ztulpen	Shows all network ports and their associate context label setting.

```
lrwxrwxrwx. root root system_u:object_r:lib_t:s0       lib -> usr/lib
lrwxrwxrwx. root root system_u:object_r:lib_t:s0       lib64 -> usr/
lib64
drwxr-xr-x. root root system_u:object_r:mnt_t:s0       media
drwxr-xr-x. root root system_u:object_r:mnt_t:s0       mnt
drwxr-xr-x. root root system_u:object_r:usr_t:s0       opt
dr-xr-xr-x  root root ?                                proc
dr-xr-x---. root root system_u:object_r:admin_home_t:s0 root
drwxr-xr-x  root root ?                                run
lrwxrwxrwx. root root system_u:object_r:bin_t:s0       sbin -> usr/sbin
drwxr-xr-x. root root system_u:object_r:var_t:s0       srv
dr-xr-xr-x  root root ?                                sys
drwxrwxrwt. root root system_u:object_r:tmp_t:s0       tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0       usr
drwxr-xr-x. root root system_u:object_r:var_t:s0       var
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 web
```

Every context label always consists of three different parts:

- **User:** The user can be recognized by _u in the context label; it is set to system_u on most directories in Listing 21.3. SELinux users are not the same as Linux users, and they are not important on the RHCSA or RHCE exams.
- **Role:** The role can be recognized by _r in the context label. In Listing 21.3, most objects are labeled with the object_r role. In advanced SELinux management, specific SELinux users can be assigned permissions to specific SELinux roles. For the RHCSA and RHCE exams, you do not have to know how to configure these.
- **Type:** The type context can be recognized by _t in the context label. In Listing 21.3, you can see that a wide variety of context types is applied to the directories in the / file system. Make sure that you know how to work with context types, because they are what it is all about on the exams.

Each context label consists of three parts-> user, role, type

Pg.540 Setting context type

To set the context type, the command "semanage" is required. This command writes the new context to the SELinux policy which is then applied to the file system.

$ yum whatprovides */semanage	To install semanage

In order to apply context setting, it is possible and easier

To apply the default setting of an existing item to another item.
For file systems;

$ ls -Z /var/www	List the context of all content within the directory /var/www/
$ semanage fcontext -a -t httpd_sys_context_t "/mydir(/.*)?"	Apply the context type to all files and folders in the directory "/mydir" -a : used to add a context type -t : change the context type * This command only writes to the policy
$ restorecon -R -v /mydir	This command applies the policy to the file system
$ man semanage-fcontext	View manual page and type "/example" to see application

```
[root@server1 /]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0
  cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
```

Exercise 21.2 Setting a Context Label on a Nondefault Apache Document Root

1. Open a root shell and type **yum install httpd elinks –y**.
2. Still from the root shell, type **mkdir /web**.
3. Type **vim /web/index.html** and put the following contents in the file: **welcome to my web server**.
4. Type **vim /etc/httpd/conf/httpd.conf** to open the Apache configuration file and find the **DocumentRoot** parameter. Change it so that it reads **DocumentRoot "/web"**.
5. In the same httpd.conf configuration file, add the following section:

```
<Directory "/web">
   AllowOverride None
   Require all granted
</Directory>
```

6. Type **systemctl restart httpd; systemctl enable httpd** to start and enable the httpd service.
7. Type **elinks http://localhost**. You'll see the default Red Hat web page and not the contents of the index.html file you have just created.
8. Type **setenforce 0** to switch SELinux to permissive mode.
9. Repeat Step 7. You'll now get access to your custom web page, which proves that SELinux was doing something to block access.
10. Type **semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"** to apply the new context label to /web.
11. Type **restorecon -R -v /web**. The **-v** (verbose) option ensures that you see what is happening and that you will see the new context being applied to /web.
12. Set SELinux back in enforcing mode, using **setenforce 1**.
13. Type **elinks http://localhost**. You'll now get access to your custom web page.

Pg.543 Finding the right context type
Three approach exist for this,

- Look at the default environment
- Read the configuration files
- Use "man -k _selinux" (most recommended)

Note that the selinux manpage is not installed by default. Hence must be installed before use------------------------->

Exercise 21.3 Installing SELinux-Specific Man Pages

1. Type **man -k _selinux**. You'll probably see just one or two man pages.
2. Type **yum whatprovides */sepolicy**. This shows you the name of the RPM that contains the sepolicy binary, which is policycoreutils-devel.
3. Type **yum -y install policycoreutils-devel** to install this package.
4. Type **sepolicy manpage -a -p /usr/share/man/man8** to install the man pages.
5. Type **man -k _selinux**. You'll see no changes yet.
6. Type **mandb** to update the database that contains names and descriptions of all man pages that are installed.
7. Once the **mandb** command has finished (this can take a few minutes), type **man -k _selinux**. You'll now see a long list of man pages scrolling by.
8. Type **man -k _selinux | grep http** to find the man page that documents SELinux settings for the httpd service and scroll through it. Notice that it is a complete list of all that you can do with SELinux on the httpd service.

Restore default file context
It is important to note the following context behaviour.

- New file created, inherits the context settings of ist parent directory.
- Copied file are considered as new file, unless the (cp -a) is specified during copy.

$ restorecon -Rv /	Restore the default context of all files
	Or simply create a file "/.autorelabel" then the file system will automatically be relabel during the new restart.

Exercise 21.4 Using *restorecon* to Relabel Files

1. From a root shell, type **ls -Z /etc/hosts**. You'll see the file has the net_config_t context label.
2. Type **cp /etc/hosts ~** to copy the file to the root home directory. Because copying is considered the creation of a new file, the context setting on the ~/hosts file is set as admin_home_t. Use **ls -Z ~/hosts** to verify this.
3. Use **mv ~/hosts /etc** and confirm that you want to overwrite the existing file.
4. Type **ls -Z /etc/hosts** to confirm that the context type is still set to admin_home_t.
5. Use **restorecon -v /etc/hosts** to reapply the correct context type. The **-v** option shows you what is happening.
6. Type **touch /.autorelabel** and restart your server. While restarting, make sure to press the **Escape** key on your keyboard so that you'll see boot messages. You'll see that the file system is automatically relabeled.

Pg.546 Using Boolean settings to Modify SELinux Settings

`$ getsebool -a`	List all the boolean selinux settings
`$ getsebool -a | grep ftp`	List the boolean selinux settings for ftp service.
`$ semanage boolean -l`	List both the current and default boolean setting
`$ setsebool ftpd_anon_write on`	Set the boolean setting (ftpd_anon_write) to only at runtime.
`$ setsebool -P ftpd_anon_write on`	Set the boolean setting (ftpd_anon_write) to on permanently.

Exercise 21.5 Working with SELinux Booleans

1. From a root shell, type **getsebool -a | grep ftp**. You'll see the ftpd_anon_write boolean, with its current value off.
2. Type **setsebool ftpd_anon_write on**. This changes the value in runtime.
3. Type **getsebool ftpd_anon_write**. It shows the value of the Boolean as on.
4. Type **semanage boolean -l | grep ftpd_anon**. Notice that this command shows the runtime configuration set to on, but the permanent setting is still set to off.
5. Use **setsebool -P ftpd_anon_write on** to switch the runtime and the default setting for the Boolean to on.
6. Repeat **semanage boolean -l | grep ftpd_anon**. Notice that it is now set to on, on.

Pg.548 Diagnosing and Addressing SELinux Policy Violation

Selinux logs everything thing it does in /var/log/audit/audit.log with type=AVC

`$ grep AVC /var/log/audit/audit.log`	View all the actions performed by SELinux
`$ sealert`	Alternative way to easily understand the log messages from SELinux. it must be installed manually -> yum -y install setroubleshoot-server

At first sight, the SELinux log messages look complicated. If you look a bit closer, though, they are not that hard to understand. Let's take a closer look at the last line in the log file:

```
type=AVC msg=audit(1414933365.304:14): avc:  denied  { getattr } for
pid=1330
comm="alsactl" path="/var/lib/alsa/asound.state" dev="dm-1"
ino=72731037
scontext=system_u:system_r:alsa_t:s0-s0:c0.c1023 tcontext=system_u:obj
ect_r:file_t:s0
tclass=file
```

The first relevant part in this line is the text **avc: denies { gettattr }**. That means that a **getattr** request was denied, so some process has tried to read attributes of a file and that was denied. Following that message, we can see **comm=alsactl**, which means that the command trying to issue the **getattr** request was **alsactl**, and we can see **path="/var/lib/alsa/asound.state"**, which is the file that this process has tried to access.

In the last part of the log line, we can get information about the source context and the target context. The source context (which is the context setting of the **alsactl** command) is set to **alsa_t**, and the target context (which is the context setting of the asound.state file) is set to **file_t**. And apparently, SELinux did not like that too much.

C22-P556::CONFIGURING FIREWALL

Mittwoch, 26. Oktober 2022 11:41

Pg.559 Understanding Linux Firewalling

The linux firewalling is performed kernel modules through the NETFILTER, which allows the kernel to inspect incoming, outgoing and forwarding packets and then allows or block them.

Interaction with the netfilter can be performed by the following means

- Iptables (not recommended anymore)
- Firewalld

It is discouraged to use both methods. Only one should be used at a time.
Firewalld is a system service that can configure firewall rules by using different interfaces and manage netfilter firewall configuration.

Understanding firewalld Zones.
The firewalld service works with zones which are a collection of rules That applies to incoming packets by matching a specific source address or network interface.

By default, it applies only to incoming packets and not outgoing packets.

Table 22.2 Firewalld Default Zones

Zone name	Default Settings
Block	Incoming network connections are rejected with an "icmp-host-prohibited" message. Only network connections that were initiated on this system are allowed.
Dmz	For use on computers in the demilitarized zone. Only selected incoming connections are accepted, and limited access to the internal network is allowed.
Drop	Any incoming packets are dropped and there is no reply.
External	For use on external networks with masquerading (Network Address Translation [NAT]) enabled, used especially on routers. Only selected incoming connections are accepted.
Home	For use with home networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.
Internal	For use in internal networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.

Some defined services allows administrator to deny and allow access to specific port on server.

$ firewall-cmd --get-services	List all the services available for firewalld

Working with firewall requires adding the right services to the right zones.
The services files are stored in the following directory

/usr/lib/firewalld/services	Directory for firewall service files
/etc/firewalld/services	Directory for firewall services files

Listing 22.2 Contents of the ftp Service File

```
[root@server1 services]# cat ftp.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>FTP</short>
  <description>FTP is a protocol used for remote file transfer. If you plan to make your FTP
server publicly available, enable this option. You need the vsftpd package installed for this
option to be useful.</description>
  <port protocol="tcp" port="21"/>
  <module name="nf_conntrack_ftp"/>
</service>
```

The following tools are used for configuring the firewall service.

$ firewall-cmd	Command line tool
$ firewall-config	Graphics user interface tool

$ systemctl mask iptables	Disables the use of iptables prior to configuring the firewall with firewall-cmd as they are both not compatible

The exercise below teaches how to add services and port to the default zones.

Exercise 22.1 Managing the Firewall with Firewall-cmd

1. Open a root shell. Type **firewall-cmd --get-default-zone**. This shows the current default zone. You'll see the current default zone, which is by default set to public.
2. To see which zones are available, type **firewall-cmd --get-zones**.
3. Now show the services that are available on your server by using **firewall-cmd --get-services**. Notice that the **firewall-cmd --get** options show what is available on your server.
4. To see which services are available in the current zone, type **firewall-cmd --list-services**. You'll see a short list containing a Dynamic Host Configuration Protocol (DHCP) client as well as Secure Shell (SSH).
5. Now type **firewall-cmd --list-all**. Look at the output and compare the output to the result of **firewall-cmd --list-all --zone=public**. Both commands show a complete overview of the current firewall configuration, as shown in Listing 22.3. Notice that you see much more than just the zone and services that are configured in that zone; you also see information about the interfaces and more advanced items.

Listing 22.3 Showing Current Firewall Configuration

```
[root@localhost ~]# firewall-cmd --list-all
public (default, active)
  interfaces: eno16777736
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

6. Type **firewall-cmd --add-service=vnc-server** to add the VNC server to the configuration of the firewall. Verify using **firewall-cmd --list-all**.
7. Type **systemctl restart firewalld** and repeat **firewall-cmd --list-all**. Notice that the vnc-server service is no longer listed.
8. Add the vnc-server service again, but make it permanent this time, using **firewall-cmd --add-service vnc-server --permanent**.
9. Type **firewall-cmd --list-all** again to verify. You'll see that VNC server is not listed. Services that have been added to the on-disk configuration are not added automatically to the runtime configuration. Type **firewall-cmd --reload** to reload the on-disk configuration into runtime configuration.
10. Type **firewall-cmd --addport=2022/tcp --permanent**, followed by **firewall-cmd --reload**. Verify using **firewall-cmd --list-all**. You'll see that a port has now been added to the firewalld configuration.

Other firewall-cmd options include the following

Table 22.3 Common **firewall-cmd** Options

Firewall-cmd Options	Explanation
--get-zones	Lists all available zones
--get-default-zone	Shows the zone currently set as default zone
--set-default-zone=<ZONE>	Changes the default zone
--get-services	Shows all available services
--list-services	Shows services currently in use
--add-service=<service-name> [--zone=<ZONE>]	Adds a service to the current default zone or the zone that is specified
--remove-service=<service-name>	Removes a service from the configuration
--list-all [--zone=<ZONE>]	Lists all configurations in a zone

Firewall-cmd Options	Explanation
--add-port=<port/protocol> [--zone=<ZONE>]	Adds a port and protocol
--remove-port=<port/protocol> [--zone=<ZONE>]	Removes a port from the configuration
--add-interface=<INTERFACE> [--zone=<ZONE>]	Adds an interface to the default zone or a specific zone that is specified
--remove-interface=<INTERFACE> [--zone=<ZONE>]	Removes an interface from a specific zone
--add-source=<ipaddress/netmask> [--zone=<ZONE>]	Adds a specific IP address
--remove-source=<ipaddress/netmask> [--zone=<ZONE>]	Removes an IP address from the configuration
--permanent	Writes configuration to disk and not to run-time
--reload	Reloads the on-disk configuration

C23-P576::CONFIGURING REMOTE MOUNTS AND FTP

Mittwoch, 26. Oktober 2022 11:41

Pg.579 Understanding Linux Firewalling

BUILT-IN::ARCHIVE->zip

Dienstag, 27. September 2022 09:31

```
$ zip <output_file.zip> file1 file2

$ zip -r <output_file.zip> folder1 file1

$
```