Linux

Gilles De Praeter

# Notities uit de les:

* ‘mysql -**u**root -**p**R2rrbLV02TA1hAjN’
  + optie ‘-u’ en ‘-p’ moeten aan de gebruikersnaam en paswoord hangen
* Bij itereren over de meegegeven variabelen gebruik je best ‘{@}’ ipv ‘{\*}’
  + ‘{\*}’ wordt geïnterpreteerd als een lange string van alle argumenten
* Bij gebruik van ‘[‘ moet er een spatie voor en na komen want dit is een commando
  + ‘ [ ] ’ ter vervanging van ‘test’
  + Let dus bij Linux goed op de spaties
* Als je ‘if’ en ‘then’ op zelfde regel wil zetten, moet er ‘;’ tussen
  + if [ "${#}" -gt '2' ]; then
* until-lus: doe het zolang het fout is
* while-lus met teller -> weinig gebruikt
  + bash niet gemaakt voor met getallen te werken -> beter python gebruiken voor getallen
* bij arithmetic expansion moet er geen $ meer voor de variabele
  + counter=$((counter + 1))
* redhat inloggen sql:
  + mysql -uroot -p
* !! voert vorig commando uit
  + Sudo !! is handig als je sudo rechten nodig hebt op vorig commando
* Als je met vagrant werkt, schakel de vm’s dan via vagrant uit en aan!
* Map linux in semester 1 openen met visual studio code, dan kan je de scripts zien en terminal openen
* “ \ “ op het einde van een regel om één commando op te splitsen over meerdere lijnen
* Inloggen als root **sudo -i**
* Ctrl + c in linux: **ctrl + shift + c** (zelfde voor ctrl + v)
* ALMALINUX:
  + Muis tonen/verdwijnen: host-key
  + Scrollen: SHIFT + PGUP / SHIFT + PGDN
* ssh -l admin 192.168.76.2
  + ssh inloggen almalinux op mint
* vagrant scripting: add mulitple lines to a file:

Graphical user interface, text

Description automatically generated with medium confidence

# Samenvatting Linux Fundamentals:

* Man pages:
  + Type **man** followed by a command (for which you want help) and start reading.
  + Press **q** to quit the manpage. Some man pages contain examples (near the end).
* Man sections:
  + **man man** will explain to you that these are section numbers.

Text, letter

Description automatically generated

* + **man *section file*:**

******

* **pwd**: displays your current directory
* **cd *$path:*** change the current directory
  + **cd** (or cd ~): go to home directory
  + **cd ..** (or cd -): go to the parent directory
* **ls**: list the contents of a directory
  + **ls -a**: show all files, including hidden ones (starting with a ‘.’)
  + **ls -l**: shows the long listing of the files
  + **ls -h**: shows the file sizes in a more readable format
  + combinations of options are also possible:
    - **ls -lh** for example
* **mkdir *$name***: creating directories
  + **mkdir -p *$absolutePath***: mkdir will create parent directories as needed.



* + - this will make all the directories as follows:



* **Rmdir**: deletes a directory
  + **Rmdir -p** ***$absolutePath***: recursively remove directories
* Files:
  + Files are case sensitive!!!
  + Everything is a file in linux, directories are a special kind of file
  + **File *$filename***: determines the type of the file
  + **Touch *$filename***: aanmaken file
  + **Rm** ***$filename***: verwijderen file
    - **Rm -rf *$filename***: verwijderen van alle soorten files indien je de rechten hebt
  + **Cp *$filenameSource $filenameTarget***: to copy files
  + **Cp -r:** copy complete directories
  + **Mv:** rename files
* **Head *$filename*:** displays the first 10 lines
  + **Head -4**: displays the first for lines
* **Tail [-n] *$filename***: same as Head, but for the last 10 lines, -n -> -4 for last 4 lines (optional)
* **Cat**: copy standard input to standard output
  + **Tac**: same as cat but in reverse order
* Linux file tree:
  + The root directory: /
    - Everything that exists on your Linux System can be found below this root directory
  + Binary directories:
    - Binaries are files that contain compiled source code (or machine code). Binaries can be executed on the computer. Sometimes binaries are called executables.
    - The **/bin** directory contains binaries for use by all users.

Table

Description automatically generated with medium confidence

* + - You can find a /bin subdirectory in many other directories. A user named serena could put her own programs in /home/serena/bin.
    - Some applications, often when installed directly from source will put themselves in /opt. A samba server installation can use /opt/samba/bin to store its binaries.
    - **/sbin** contains binaries to configure the operating system. Many of the system binaries require root privilege to perform certain tasks.
    - Binaries found in /bin and /sbin often use shared libraries located in /lib.
  + Configuration directories:
    - The **/boot** directory contains all files needed to boot the computer.
    - **/etc** contains all of the machine-specific configuration files
      * Editable Text Configuration
      * Many times the name of a configuration files is the same as the application, daemon, or protocol with .conf added as the extension.
      * **/etc/sysconfig**: contains a lot of Red Hat Enterprise Linux configuration files.
  + Data directories:
    - **/home**: users can store personal or project data under /home.
    - **/root**: On many systems /root is the default location for personal data and profile of the root user
    - **/tmp**: Applications and users should use /tmp to store temporary data when needed.
  + In memory directories:
    - **/dev:** Device files in /dev appear to be ordinary files, but are not actually located on the hard disk.
      * **/dev/null**: Anything written to /dev/null will be discarded.
    - **/sys:** Basically the /sys directory contains kernel information about hardware.
  + Unix System Resources (usr):
    - The **/usr** hierarchy should contain shareable, read only data.
    - **/usr/bin**: The /usr/bin directory contains a lot of commands.
    - **/usr/lib**: The /usr/lib directory contains libraries that are not directly executed by users or scripts.
    - **/usr/local**: The /usr/local directory can be used by an administrator to install software locally.
    - **/usr/share**: The /usr/share directory contains architecture independent data
      * This directory typically contains **/usr/share/man** for manual pages
    - **/usr/src**: The /usr/src directory is the recommended location for kernel source files
  + Variable data (vr):
    - Files that are unpredictable in size, such as log, cache and spool files, should be located in **/var**.
    - **/var/log**: The /var/log directory serves as a central point to contain all log files.
    - **/var/cache**: The /var/cache directory can contain cache data for several applications.
    - **/var/lib**: The /var/lib directory contains application state information.
* **Echo**: it echoes the input it receives
* Shell expansion: One of the primary features of a shell is to perform a command line scan. When you enter a command at the shell's command prompt and press the enter key, then the shell will start scanning that line, cutting it up in arguments. While scanning the line, the shell may make many changes to the arguments you typed. This process is called shell expansion. When the shell has finished scanning and modifying that line, then it will be executed.
  + Parts that are separated by one or more consecutive white spaces (or tabs) are considered separate arguments, any white space is removed. The first argument is the command to be executed, the other arguments are given to the command. The shell effectively cuts your command into one or more arguments.
* **Type *$command***: to find out if a command is an external or built-in command
* Alias:
  + **Alias *$aliasName*=’*$commandAndOrParameters*’**



* + **Alias *$aliasName***: You can provide one or more aliases as arguments to the alias command to get their definitions.
  + Providing no arguments gives a complete list of current aliases.
  + **Unalias *$aliasName***: to undo an alias
* Control operators:
  + Semicolon ‘;’:
    - You can put two or more commands on the same line separated by a semicolon ‘;’
    - Both series will be executed sequentially with the shell waiting for each command to finish before starting the next one.
  + Ampersand ‘&’:
    - When a line ends with an ampersand ‘&’, the shell will not wait for the command to finish. You will get your shell prompt back, and the command is executed in background. You will get a message when this command has finished executing in background.
  + Dollar question mark ‘**$?**’:
    - The exit code of the previous command is stored in the shell parameter ‘$?’
    - 0 is executed successfully, other than 0 -> failed



* + Double ampersand ‘**&&**’:
    - The shell will interpret && as a logical AND. When using && the second command is executed only if the first one succeeds



* + Double vertical bar ‘**||**’:
    - The || represents a logical OR. The second command is executed only when the first command fails



* + Pound sign ‘**#**’:
    - Shell comments
  + Escaping special characters ‘**\**’:
    - The backslash \ character enables the use of control characters, but without the shell interpreting it, this is called escaping characters.



* + - End of line backslash: lines ending in a backslash are continued on the next line

Text

Description automatically generated

* Shell variables:
  + Case sensitive
  + Dollar sign ‘**$**’:
    - The shell will look for an environment variable named like the string following the dollar sign and replace it with the value of the variable
  + Creating variables:
    - ***$varName*=*$varValue***



* + Double quotes vs single quotes: Double quotes still allow the parsing of the variables, whereas single quotes prevent this
  + **Set:** displays a list of environment variables
  + **Unset**: to remove a variable from the shell environment

**Text, letter

Description automatically generated**

* + The **$PS1** variable:
    - Determines your shell prompt

**Text, letter

Description automatically generated**

* + - **\u** for user and **\w** for the working directory
  + **$PATH:**
    - The $PATH variable determines where the shell is looking for commands to execute (unless the command is builtin or aliased).
  + **Env:**
    - The **env** command without options will display a list of exported variables. The difference with set with options is that set lists all variables, including those not exported to child shells.
    - But env can also be used to start a clean shell (a shell without any inherited environment). The **env -i** command clears the environment for the subshell.
  + **Export *$varname***: to export shell variables to other shells (child shells)
  + **$():** 
    - Until now, we have seen that bash interprets a variable starting from a dollar sign, continuing until the first occurrence of a non-alphanumeric character that is not an underscore. This issue can be resolved with curly braces

**Text

Description automatically generated**

* + Unbound variables:
    - the **nounset** shell option that you can use to generate an error when a variable does not exist.
    - In the bash shell **set -u** is identical to **set -o nounset** and likewise **set +u** is identical to **set +o nounset**.

Text, letter

Description automatically generated

* Shell embedding:
  + Shells can be embedded on the command line, or in other words, the command line scan can spawn new processes containing a fork of the current shell.
  + Can be done using backticks `` or $()





* Shell history:
  + **!!**: to repeat the last command
  + **!*letters***: will repeat the last command that started with those letters
  + **History *n***: will show the history of the n last commands
  + **!*n***: will execute the n’th command

**Text

Description automatically generated**

****

* + **$histsize:**
    - The $HISTSIZE variable determines the number of commands that will be remembered in your current environment. Most distributions default this variable to 500 or 1000.

****

* + **$histfile:**
    - The $HISTFILE variable points to the file that contains your history. The bash shell defaults this value to ~/.bash\_history.

****

* + **$histsizefile:**
    - The number of commands kept in your history file can be set using $HISTFILESIZE.

****

* File globbing:
  + Asterisk ‘**\***’:
    - matching to any combination of characters (even none).
    - When no path is given, the shell will use filenames in the current directory.

Graphical user interface, text

Description automatically generated

* + Question mark ‘**?**’:
    - Matches with exactly one character



* + Square brackets ‘**[]**’:
    - Matches any of the characters between the brackets
    - Each pair of brackets is replaced by exactly one character
    - Order between the brackets is not relevant

Text

Description automatically generated

* + - The shell will also understand ranges between the brackets

Text

Description automatically generated

* + Preventing globbing:

Text, letter

Description automatically generated

* I/O redirection:
  + Stdin, stdout, stderr: The bash shell has three basic streams; it takes input from stdin (stream 0), it sends output to stdout (stream 1) and it sends error messages to stderr (stream 2) .

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

* + **>** stdout: stdout can be redirected with a greater than sign. While scanning the line, the shell will see the > sign and will clear the file. (even if the command fails, the file will have been cleared)

Text

Description automatically generated

Text

Description automatically generated

* + Noclobber:
    - Erasing a file while using > can be prevented by setting the noclobber option
    - The noclobber can be overruled with **>|**

Text, letter

Description automatically generated

* + Append to file using ‘**>>**’

Text

Description automatically generated

* + Error redirection: Redirecting stderr is done with **2>**



* + Redirect error and standard output: **2>&1**

****

* + Join stdout en stderr: **&>**

Text

Description automatically generated

* + Stdin redirection: **<**



* + Here document: **<<**
    - Use ‘\_EOF\_’ to use more than one line

Graphical user interface

Description automatically generated

* + Quick file clearing: **> *$fileName***



* + ***Command1* | *command2***: stdout from command 1 is used as stdin for command 2
* Filters (often used with a pipe ‘**|**’):
  + **Grep**: to filter lines of text (returns the line containing the character(s)) **ZIE PAGINA 15 VOOR MEER!!!**
    - **Grep -i**: filters in a case insensitive way
    - **Grep -v**:outputsthe lines that do not match the filter

Graphical user interface, text

Description automatically generated

* + - **Grep -A1**: one line after the result is also displayed
    - **Grep -B1:** one line before the result is also displayed
    - **Grep -C1:** one line before and after the result are displayed

Text

Description automatically generated with medium confidence

* + **Cut**:
    - The cut filter can select columns from files, depending on a delimiter or a count of bytes.

**A picture containing chart

Description automatically generated**

-> It uses the colon as a delimiter, and selects fields 1 and 3.

* + - When using a space as the delimiter for cut, you have to quote the space.

**A picture containing text

Description automatically generated**

-> display the 2nd to 7th character

* + **Tr:** You can translate characters with tr
    - All letters to uppercase:

Text

Description automatically generated

* + - Newlines to spaces:

Text

Description automatically generated

* + - **Tr -s**: used to squeeze multiple occurrences of a character to one.

Graphical user interface, text

Description automatically generated

* + - **Tr -d**: to delete characters

Text

Description automatically generated with medium confidence

* + **Wc**: to count words, lines and characters

Text, letter

Description automatically generated

* + **Sort**: by default it will sort alphabetical

Text

Description automatically generated

* + - **Sort -n**: sorts numerical
    - **Sort -k1**: sorts based on column 1, can be another number

Text

Description automatically generated

* + **Uniq**: remove duplicates from a sorted list
    - **Uniq -c**: to count occurancies

Text

Description automatically generated

Text

Description automatically generated with low confidence

* + **Comm**: comparing streams
    - By default comm will output three columns.
    - In this example, Abba, Cure and Queen are in both lists, Bowie and Sweet are only in the first file, Turner is only in the second.

**Text

Description automatically generated**

* + - The output of comm can be easier to read when outputting only a single column. The digits point out which output columns should not be displayed.

**Text, letter

Description automatically generated**

* + **Od:** 
    - European humans like to work with ascii characters, but computers store files in bytes.
    - The example below creates a simple file, and then uses od to show the contents of the file in hexadecimal bytes

**Text

Description automatically generated**

* + - Octal bytes**:**

****

* + - Ascii characters:



* + **Sed**: can perform editing functions in the stream, using regular expressions.

Text

Description automatically generated

* + - Add **g** for global replacements

Text

Description automatically generated with medium confidence

* + - With **d** you can remove lines from a stream containing a character

Graphical user interface, text

Description automatically generated

* + Pipe examples:

Text

Description automatically generated

A picture containing text

Description automatically generated

Text, letter

Description automatically generated

* **Locate *$filename*:** uses an index to locate files, faster than traversing all the directories, but it also means that it is always outdated.
  + If the index does not exist yet, then you have to create it (as root on Red Hat Enterprise Linux) with the **updatedb** command (most Linux systems schedule this command to run once every day).
* **Date**: display the time, time zone and more

****

****

****

****

* **Find**: can be very useful at the start of a pipe to search for files
  + You might want to add 2>/dev/null to the command lines to avoid cluttering your screen with error messages.

****

* **Cal**: displays the current month, with the current day highlighted

**Text

Description automatically generatedText

Description automatically generated**

* **Sleep**: used to wait for a number of seconds

****

* **Time**: displays how long it takes to execute a commanda

**Graphical user interface, text, application

Description automatically generated**

* **Gzip**: used for compression so files take up less space
  + **Gunzip:** to get the original file back

Text

Description automatically generated

* **Zcat or zmore**: files that are compressed with gzip can be viewed with zcat and zmore

Text, letter

Description automatically generated

* **Bzip2**: compresses file better than gzip, but takes more time
  + **Bunzip2**: to get the original file back (unzipping)
  + **Bzcat or bzmore**: to display the files compressed with bzip2
* Regex versions:



* **Grep**:
  + **-E** option to force a string to be read as ERE
  + **-G** forces BRE
  + **-P** forces PRCE
  + Match one or the other character:
    - BRE needs an ‘\’ before the ‘|’, ERE doesn’t

Text

Description automatically generated with medium confidenceText

Description automatically generated with medium confidence

* + Match one or more:
    - **\*** signifies zero, one or more of the previous
    - **+** signifies one or more of the previous

**Text, letter

Description automatically generated**

* + Match the end of a string:
    - **$** character to match the end of a string

Text

Description automatically generated

* + Match the start of a string:
    - **^** character to match the start of a string

Text

Description automatically generated

* + Separating words:
    - Regular expressions use a **\b** sequence to reference a word separator
    - Surrounding the searched word with spaces is not a good solution (because other characters can be word separators).

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

* + - The **-w** option can also be used

Text, letter

Description automatically generated

* **Sed**:
  + = stream editor, uses regex for stream editing
  + Replace a string: (other characters can be used than ‘/’)

Graphical user interface, text

Description automatically generated

* + Can be used on a file:

Text, letter

Description automatically generated

* + The ‘**&**’ character can be used to reference the searched (and found) string

Graphical user interface, text

Description automatically generated

* + ‘**()**’ are used to group sections of a regex so they can be referenced later

Text

Description automatically generated

* + ‘**.**’ For any character

A screenshot of a computer

Description automatically generated with low confidence

* + Multiple back referencing (called grouping):



* + The ‘**\s**’ can refer to white space such as a space or a tab.

Text

Description automatically generated

* + ‘**?**’ signifies that the previous is optional.

Text, letter

Description automatically generated

* + Exactly n times the previous has to occur:

Text

Description automatically generated

* + Between n and m times:

Text, letter

Description automatically generated

* Introduction to **vi**:
  + The vi editor starts in command mode. In command mode, you can type commands. Some commands will bring you into insert mode. In insert mode, you can type text.
  + The escape key will return you to command mode.
  + Start typing:

Table

Description automatically generated

* + Replace and delete a character:

Graphical user interface, text, application, email

Description automatically generated

* + Undo and repeat the last action:

Graphical user interface, text, application, email

Description automatically generated

Table

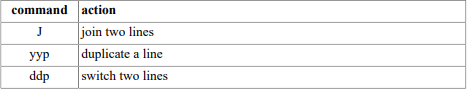
Description automatically generated

* + Start and end of a line:

Table

Description automatically generated

* + Join two lines and more:



* + Save (or not) and exit:

Graphical user interface, text, application, email

Description automatically generated

* + Searching:



Graphical user interface, text, application, email

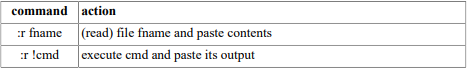
Description automatically generated

* + Replace all:

Graphical user interface, text, application, Word

Description automatically generated

* + Reading files:



* + Text buffers:

Graphical user interface, text, table

Description automatically generated

* + Multiple files:

Table

Description automatically generated

* + Abbreviations:

Graphical user interface, text, application

Description automatically generated

* Scripting:
  + **Chmod +x *$filename***: to make a file executable
  + ‘**#!**’: She-bang
  + **‘#!/bin/bash**’: to ensure your script will run in a bash shell
  + ‘**#**’: comment
  + ‘***varName*=*varValue***’: creating a variable
    - These only exist in the script itself (because this is another shell)

Graphical user interface, text, application

Description automatically generatedText, letter

Description automatically generated

* + - Sourcing a script: running the script in the same shell

Text

Description automatically generatedText

Description automatically generated

* Troubleshooting:
  + **Bash -x *$filename***: allows you to see the commands the shell is executing

Text

Description automatically generated

* Scripting loops:
  + **Test**: can test if something is true or false
    - 1 -> fails, 0 -> success
    - Can also be written by ‘ **[ ]** ‘

****

**Text

Description automatically generated with medium confidence**

* + **If then else**

**A picture containing calendar

Description automatically generated Logo

Description automatically generated with medium confidence**

* + **For loop**

**A picture containing text

Description automatically generated**

**A picture containing graphical user interface

Description automatically generated**

**Diagram

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated**

* + **While loop**

**Chart, scatter chart

Description automatically generated Text

Description automatically generated**

* + **Until loop**

**Chart, scatter chart

Description automatically generated with medium confidenceText

Description automatically generated**

* + **Case**:

Chart, scatter chart

Description automatically generated Graphical user interface, text, application

Description automatically generated

* Script parameters:
  + Bij itereren over de meegegeven variabelen gebruik je best ‘{@}’ ipv ‘{\*}’
    - ‘{\*}’ wordt geïnterpreteerd als een lange string van alle argumenten

Text, letter

Description automatically generated Text, letter

Description automatically generated

* + ‘**$0**’ contains the name of the script

Text

Description automatically generated

* + The **shift** parameter:

Graphical user interface, text

Description automatically generated Text

Description automatically generated

* + **Read** command to ask the user for input

**Text

Description automatically generated**

* + **Getopts**: allows you to parse options given to a command

**Graphical user interface, text

Description automatically generated** **Text, letter

Description automatically generated**

* More scripting:
  + **Eval**: reads arguments as input to the shell (the resulting commands are executed). This allows using the value of a variable as a variable.

Text

Description automatically generated

* + - Sometimes the eval is needed to have correct parsing of arguments.

Text

Description automatically generated

* + **(( ))**: allows for evaluation of numerical expressions

Text, letter

Description automatically generated

* + **Let**: The let built-in shell function instructs the shell to perform an evaluation of arithmetic expressions. It will return 0 unless the last arithmetic expression evaluates to 0.

Text

Description automatically generated

A picture containing text, screen

Description automatically generated

* + Functions:
    - Een functie gedraagt zich als een script!
    - oproepen: functie\_naam arg1 arg2 arg3
    - positionele parameters: ${1}, ${2}, enz.
    - return STATUS ipv exit

A picture containing text

Description automatically generated

Text, letter

Description automatically generated Text

Description automatically generated

* Users:
  + **Whoami**: tells you your username



* + **Who**: will give you information about who is logged on the system

Text

Description automatically generated

* + **Who am i**: will display only the line pointing to your current session



* + **w**: shows you who is logged on and what they are doing

Graphical user interface, text, application, Word

Description automatically generated

* + **id**:gives your user id, primary group id, and a list of the groups that you belong to

****

* + **su**: allows a user to run a shell as another user
    - The root user can become any other user without their password

Text

Description automatically generated

* + - **Su - *$username***: To become another user and also get the target user's environment

***Text, letter

Description automatically generated***

* + - **Su -** : the command will assume the target is root
* User management:
  + **/etc/passwd**: the local user database on Linux
    - Root user always has user id 0
  + **Useradd**:
    - The example below shows how to add a user named yanina (last parameter) and at the same time forcing the creation of the home directory (-m), setting the name of the home directory (-d), and setting a description (-c).



* + **/etc/default/useradd**: a file that contains some default user options
    - Can be accessed with cat or wit useradd -D
  + **Userdel**: to delete a user
    - **Userdel -r**: will also remove the home directory



* + **Usermod**: modify the properties of a user

Text

Description automatically generated

* + Making home directories:
    - Easiest way: **useradd -m** (likely to be a default option on Linux)
    - Less easy way: using **mkdir**, **chown**  and **chmod**

**Text

Description automatically generated**

* + **/etc/skel/**: When using useradd the -m option, the /etc/skel/ directory is copied to the newly created home directory. The /etc/skel/ directory contains some (usually hidden) files that contain profile settings and default values for applications. In this way /etc/skel/ serves as a default home directory and as a default user profile.

**Text, letter

Description automatically generated**

* + **/etc/passwd**: contains the login shell for the user
    - Use **usermod** to change the shell for a user

****

* User passwords:
  + **Passwd**: passwords of users can be set with the passwd command
  + **Etc/shadow**: this file contains information of the user’s their password and the encrypted password itself
  + passwd uses **crypt** function standard for its encryption
  + **/etc/login.defs**: contains some default settings for user passwords
  + **Chage:** set an expiration date for a user account
    - For options check man
  + Editing local files (/etc/passwd or /etc/shadow) is done by using **vipw** instead of vi
    - The vipw tool will do proper locking of the file.
* Groups:
  + **Groupadd**: to add groups

****

* + **/etc/group**: contains the definition of the group memberships
    - 3rd field is the group identification GID

Text

Description automatically generated

* + **Groups**: to see the list of groups



* + **Usermod** can be used to modify group memberships

Text

Description automatically generated

* + **Groupmod**: make changes to the group, for example change the name

Text, letter

Description automatically generated

* + **Groupdel:** permanently remove a group



* + **Gpasswd**: delegate control of group membership to another user
    - In the example below we delegate permissions to add and remove group members to serena for the sports group. Then we su to serena and add harry to the sports group

Text, letter

Description automatically generated

* + **/etc/gshadow**: information about group administrators
  + **Newgrp**: You can start a child shell with a new temporary primary group
  + **Vigr** is used to edit /etc/group
* **32……..**

# Linux slides Cursus

## 1. Linux intro & user management

* Bestandspermissies: toegangsrechten voor bestanden en directories
  + Bestanden zijn eigendom van een gebruiker en groep
  + **u**: gebruiker, user
  + **g**: groep, group
  + **o**: andere gebruikers, others
  + **a**: iedereen, all
  + **r**: lezen, read
  + **w**: schrijven, write
  + **x**: execute
    - Bestand: uitvoeren als commando
    - Directory: toegang met cd
  + Instellen met **chmod** via de symbolische notatie:
    - chmod g+rw bestand
    - chmod +x bestand
    - chmod u+rw,g+r,o-rw bestand
    - chmod a=r bestand
  + Instellen met **chmod** via de octale notatie:
    - chmod 755 script

Calendar

Description automatically generated with medium confidence

* + Merk op:
    - Sommige permissie-combinaties komen nooit voor in de praktijk, bv. 1, 2, 3
    - Directories: altijd lezen (r) en execute (x) samen toekennen of afnemen
    - Permissies owner ≥ group ≥ others
    - Root negeert bestandspermissies (mag alles), vb. /etc/shadow
    - Tip: octale permissies opvragen: **stat -c %a BESTAND**
* Permissies van nieuwe bestanden:
  + **umask** bepaalt permissies van bestand/directory bij aanmaken
  + Huidige waarde opvragen: umask zonder opties
  + Opgegeven in octale notatie
    - Enkel 0, 2 en 7 zijn zinvol
  + Welke permissies afnemen
    - Bestand krijgt nooit execute-permissie
  + umask 0027, wat worden de permissies?

A picture containing text

Description automatically generated

* Speciale permissies: SUID
  + Set user ID (SUID)
  + Op bestanden met execute-permissies
  + Tijdens uitvoeren krijgt de gebruiker de rechten van de eigenaar van het bestand
  + Symbolisch: u+s
  + Octaal: 4
  + **Chmod u+s BESTANDSNAAM**



* Speciale permissies: SGID
  + Set group ID (SGID)
  + Op bestanden met execute-permissies
  + Tijdens uitvoeren krijgt de gebruiker de rechten van de groep van het bestand
  + Symbolisch: g+s
  + Octaal: 2
  + **Chmod g+s BESTANDSNAAM**



* Speciale permissies: restricted deletion
  + restricted deletion, of sticky bit
  + Toegepast op directories
  + Een bestand mag in zo’n directory enkel door de eigenaar verwijderd worden
  + Symbolisch: +t
  + Octaal: 1
  + **Chmod +t BESTANDSNAAM**



* Commando’s voor gebruikers en groepen:
  + Gebruikers: **useradd, usermod, userdel**
  + Groepen: **groupadd, groupmod, groupdel**
  + Info opvragen: **who, groups, id**
* Configuratiebestanden:
  + Gebruikers: /etc/passwd, /etc/shadow
  + Groepen: /etc/group, (/etc/gshadow, van weinig belang)
* Primaire vs aanvullende groepen:
  + Primaire groep: 4e veld van /etc/passwd (group ID)
  + Aanvullende groepen: in /etc/group. Gebruiker staat niet vermeld in de primaire groep!
  + Primaire groep aanpassen:



* + Gebruiker toevoegen aan opgegeven groepen en verwijderen uit alle andere groepen:



* + Gebruiker toevoegen aan opgegeven groep, blijft lid van andere groepen:



* Eigenaar/groep veranderen:
  + Root rechten nodig!

A picture containing graphical user interface

Description automatically generated

* Root of administrator:
  + Root oorspronkelijk de enige administrator van het systeem
  + sudo kan root rechten verlenen aan een gebruiker
    - Voor een bepaald commando
    - Voor alle commando’s
  + /etc/sudoers tells you who can be admin
    - group sudo in Ubuntu/Debian distro
    - group wheel in Fedora/RedHat distro
* Wachtwoord:
  + Eigen wachtwoord aanpassen:



* + Wachtwoord aanpassen van een gebruiker als administrator:



* + Wachtwoord genereren in hash vorm (zoals in /etc/shadow):

Text

Description automatically generated with medium confidence

## 2. Combining commands towards scripting

* I/O redirection:

Table

Description automatically generated

Text

Description automatically generated

* Foutboodschappen afdrukken:



* + Equivalent van System.err.printf()
* Filters:
  + Filter = commando dat:
    1. leest van stdin of bestand,
    2. tekst transformeert, en
    3. wegschrijft naar stdout
  + Combineer filters via | (pipe) om complexe bewerkingen op tekst toe te passen
  + De UNIX-filosofie
  + **cat**: wat binnenkomt op stdin afdrukken op stdout
  + **tac**: idem, maar in omgekeerde volgorde
  + **shuf**: in willekeurige volgorde
  + **head**  en **tail**: toon 10 eerste / laatste regels
  + **cut**: selecteer kolommen uit gestructureerde tekst (bv. CSV)
  + **paste**: voeg bestanden regel per regel samen
  + **join**: voeg bestanden samen ahv gemeenschappelijke kolom

Polygon

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

* + **sort** en **uniq:**

Graphical user interface

Description automatically generated with medium confidence

* + **fmt, nl** en **wc:**

Text, application

Description automatically generated

* + **grep**:

Polygon

Description automatically generated

* + **tr**:
    - TRanslate
    - Teken per teken ipv lijn per lijn
    - Enkel stdin, geen files



* + Sed (=Stream EDitor):

Graphical user interface, text

Description automatically generated

* + AWK:

Text, letter

Description automatically generated

* Een script schrijven:
  + Maak bestand aan (bv. script.sh) met een teksteditor, bv.
  + Maak bestand uitvoerbaar: **chmod +x script.sh**
  + Voer uit: ./script.sh
* De shebang:
  + Eerste regel van een script
  + Begint met #! (# = hash; ! = bang)
  + Absoluut pad naar de interpreter voor het script, bv:
    - #! /usr/bin/python
    - #! /usr/bin/ruby
    - #! /usr/bin/env bash
      * = zoek in ${PATH} naar bash
* Variabelen:
  + Bash-variabelen zijn (meestal) strings.
  + Declaratie: **variabele=waarde**
  + Geen spaties rond =
  + Waarde v/e variabele opvragen: ${variable}
  + Gebruik in strings (met dubbele aanhalingstekens)
* Onbestaande variabelen:
  + Onbestaande variabele wordt beschouwd als lege string.
  + Oorzaak van veel fouten in shell-scripts!
  + **set -o nounset** ⇒ script afbreken
* Scope variabelen:
  + Enkel binnen zelfde “shell”, niet binnen “subshells”
  + Een script oproepen creëert een subshell
  + Maak “globale”, of omgevingsvariabele met **export**

Text

Description automatically generated with medium confidence

* Naamgeving variabelen:
  + Lokale variabelen: kleine letters, bv: foo\_bar
  + Omgevingsvariabelen: hoofdletters, bv. FOO\_BAR
* **Vi:**

**Table

Description automatically generated**

## 3. Software-installatie, netwerkconfiguratie

* Waar komt software vandaan?
  + Package = verzameling van …
    - Gecompileerde software voor een versie van Linux
    - Bijhorende bestanden (configuratie, man, …)
    - Informatie over waar deze bestanden terecht horen
    - Eventuele afhankelijkheden van libraries of andere software
    - = dependency (zie verder)
  + Packages worden aangeboden op specifieke servers
    - repository servers
    - Werken zoals ‘Play Store’ (of liever: omgekeerd)
* Debian vs Red Hat
  + Debian
    - .deb packages
    - Ubuntu
    - Mint
    - Raspbian
    - …
  + Red Hat
    - .rpm packages
    - Fedora
    - CentOS
    - AlmaLinux
    - …
* Debian dpkg:
  + Tool die een .deb package installeert op een systeem.
  + Download een .deb package (manueel)
  + Installeer met **dpkg**



* + Los eventuele depencencies manueel op
  + Overzicht van geïnstalleerde packages op Debian



* Dependency:
  + Software wordt gebouwd boven op andere software.
  + Zonder de onderliggende bouwstenen kan dit niet werken.
  + Linux name = dependency

Text

Description automatically generated with low confidence

* Debian **apt**:
  + APT = Advanced Package Tool
  1. Zoek een package op de (aanvaarde) repository servers
  2. Download de .deb package (automatisch)
  3. controlleer depencencies, download eventuele extra packages
  4. Installeert (achterliggend) met dpkg



* + Bijwerken van info op de repo servers



* + Bijwerken van alle packages op jouw systeem



* + Bijwerken van alle packages op jouw systeem



* List of debian repository servers:

Text

Description automatically generated with medium confidence

* Red Hat yum/dnf:
  + Identical system, but different command set.
  + **rpm** = RedHat Package Manager (equivalent van dpkg)
  + **yum** = Yellowdog Update Manager
    - equivalent van apt
    - Yellowdog = oude Linux distro voor Motorola-CPU’s
    - Nog in gebruik op RedHat 6
  + **dnf** = DaNdiFied yum
    - Vervangt yum vanaf RedHat Enterprise Linux 7
    - Ook in Fedora
  + Installatie package
    - **$ dnf install <package\_name>**
  + Bijwerken van info op de repo servers; aftoetsen van wat kan bijgewerkt worden:
    - **$ dnf check-update**
  + Bijwerken van alle packages op jouw systeem
    - **$ dnf upgrade**
  + Bijwerken van een enkele package op jouw systeem
    - **$ dnf upgrade <package\_name>**
  + Andere handige commando's
    - Lijst geïnstalleerde packages
      * **$ dnf list installed**
    - Lijst beschikbare packages
      * **$ dnf list available**
    - Met welke package kan ik het commando fortune installeren?
      * **$ dnf provides \*bin/fortune**
    - Wat zijn de dependencies van curl?
      * **$ dnf deplist curl**
* RedHat repository servers:



* Netwerkconfiguratie:
  + Netwerkinstellingen controleren:
    - Om Internettoegang mogelijk te maken zijn er 3 instellingen nodig:
      * IP-adres en subnetmasker
      * Default gateway
      * DNS-server
  + Netwerkinstellingen opvragen:
    - IP-adress/netmask: **ip address (ip a)**
    - Default gateway: **ip route (ip r)**
    - DNS-server:
      * EL: **cat /etc/resolv.conf**
      * Debian, Fedora: **resolvectl status <interface>**
  + Wat is het IP-adres van...?

Text

Description automatically generated with low confidence

* + Wat is mijn publiek IP-adres?

Word

Description automatically generated with low confidence

* + Controleer eerst netwerkinstellingen

Text

Description automatically generated

* + Netwerkinstellingen
    - lo (loopback): 127.0.0.1/8
    - eth0/enp0s3 = 1e VirtualBox adapter (NAT): 10.0.2.15/24
    - eth1/enp0s8 = 2e VirtualBox adapter (Host-only): 192.168.76.2/24
  + Netwerkinstellingen aanpassen (RedHat)
    - **/etc/sysconfig/network-scripts/ifcfg-eth1**

Text

Description automatically generated

* + Na aanpassingen, netwerk herstarten:



* Dhcp:
  + Configbestand: **/etc/dhcp/dhcpd.conf**
  + Bestand leases: **/var/lib/dhcpd/dhcpd.leases**
  + Opstarten:

Text

Description automatically generated

* Dhcp op LINUX MINT:
  + Interface bestand: /etc/network/interfaces
  + Bring interface up: ifup enp0s8
  + Bring interface down: ifdown enp0s8

## 4. Installatie van een webserver, scripting

* LAMP-stack: Linux + Apache + MariaDB + PHP
* Belangrijke directories:
  + **/etc/httpd/**: configuratie Apache
    - **/etc/httpd/conf/httpd.conf**
    - **/etc/httpd/conf.d/\*.conf**
  + **/var/www/html/**: Apache DocumentRoot
  + **/var/log/httpd/**: logbestanden
    - access\_log
    - error\_log
* services opstarten:
  + **start**: nu opstarten
  + **enable**: Automatisch opstarten bij booten

Graphical user interface, text, application

Description automatically generated

* test services:

A picture containing graphical user interface

Description automatically generated

* + CLI webbrowser op de VM
    - surf naar <http://localhost/> gebruik commando **curl**



* + PHP testen: maak bestand /var/www/html/info.php met inhoud: <?php phpinfo(); ?>
  + Surf naar: <http://localhost/info.php>
* Toon sockets/poorten in gebruik
  + Show sockets: **ss**
  + netstat is obsolete, replaced by ss
    - netstat uses /proc/net/tcp
    - ss directly queries the kernel
  + Similar options

Graphical user interface

Description automatically generated

* + Ss -tlnp
* Logbestanden Apache:

Text

Description automatically generated

* **Journalctl:**
  + journalctl requires root permissions
    - Or, add user to group adm or systemd-journal
  + Some “traditional” text-based log files still exist (for now?):
    - /var/log/messages (gone in Fedora!)
    - /var/log/httpd/access\_log and error\_log
    - …

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

* Website vanaf LINUX MINT bekijken:
  + surf naar http://192.168.76.2/

Scripting:

* positionele parameters:
  + Bij uitvoeren van script zijn opties en argumenten beschikbaar via variabelen, positionele parameters

Table

Description automatically generated

* + Bij itereren over de meegegeven variabelen gebruik je best ‘{@}’ ipv ‘{\*}’
    - ‘{\*}’ wordt geïnterpreteerd als een lange string van alle argumenten
* **Shift**:
  + Het commando shift schuift positionele parameters op naar links:

Text

Description automatically generated

* Exit status:
  + Elk commando heeft een exit-status, numerieke waarde
    - Opvragen met **echo "$?"**
    - 0 => commando geslaagd, logische true
    - 1-255 => commando gefaald, logische false
  + Logische operatoren in Bash zijn gebaseerd op exit-status
  + Booleaanse variabelen bestaan niet
* Scripting loops:
  + **If then else**

**A picture containing calendar

Description automatically generated Logo

Description automatically generated with medium confidence**

* + **For loop**

**A picture containing text

Description automatically generated**

**A picture containing graphical user interface

Description automatically generated**

**Diagram

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated**

* + **While loop**

**Chart, scatter chart

Description automatically generated Text

Description automatically generated**

* + **Until loop**

**Chart, scatter chart

Description automatically generated with medium confidenceText

Description automatically generated**

* + **Case**:

Chart, scatter chart

Description automatically generated Graphical user interface, text, application

Description automatically generated

* Arithmatic expansion:
  + Arithmetic expansion in Linux uses the built-in shell expansion to use parenthesis for mathematical calculations.
  + **$(( arithmentic expression ))**

A picture containing graphical user interface

Description automatically generated

## 5. Hardening van een webserver

Firewall-configuratie:

* Firewall-instellingen aanpassen

Text

Description automatically generated

* Zones:
  + Zone = lijst van regels voor een specifieke situatie
    - bv. in een publieke ruimte (standaard), thuisnetwerk, werk, …
  + Netwerkkaarten worden toegekend aan een zone
  + Vooral nuttig voor laptop
    - Server: public zone is meestal voldoende

Table

Description automatically generated

* Firewall regels instellen:

Table

Description automatically generated

* Persistente wijzigingen:
  + ‘--permanent’ optie wordt niet onmiddellijk toegepast!
  + Twee werkwijzen:
    - Commando 2x uitvoeren, 1x zonder, 1x mét --permanent
    - Commando enkel met --permanent uitvoeren, firewall herladen
* SELinux
  + Security Enhanced Linux
    - “Mandatory Access Control”
    - Ingebouwd in de Linux kernel
    - Vooral op RedHat!
  + AppArmor:
    - Equivalent op Debian/Ubuntu
    - Niet behandeld in deze cursus
* Staat SELinux aan?

A picture containing text

Description automatically generated

* Hoofdconfiguratiebestand: **/etc/selinux/config**

Text

Description automatically generated

* SELinux niet uitzetten op productiesystemen!!!!!
* 3 soorten SELinux-instellingen:
  + Booleans: getsebool, setsebool
  + Contexts, labels: ls -Z, chcon, restorecon
  + Policy modules: sepolicy
    - In de meeste gevallen is configuratie van booleans/context voldoende!
* Controleer de context van een bestand
  + Wat is de SELinux-context van een bestand?
    - **ls -lZ**
  + Wat wordt de SELinux-context bij aanmaken van een nieuw bestand?
    - **/etc/selinux/targeted/contexts/files/files\_contexts**
  + Standaard-context herstellen:
    - **sudo restorecon -R /var/www/**
  + Context instellen naar een bepaalde waarde:
    - **sudo chcon -t httpd\_sys\_content\_t test.php**
* check Booleans:
  + **getsebool -a | grep http**
    - Know the relevant booleans! (RedHat manuals)
  + Enable boolean:
    - **sudo setsebool -P httpd\_can\_network\_connect\_db on**
* Hoe weet ik wat SELinux tegenhoudt?



* Mariadb config file: **/etc/my.cnf.d/**
* semanage fcontext -a -t mysqld\_db\_t "/home/admin/dbdata(/.\*)?"

## 6. Automatiseren webserverinstallatie

* Fouten opsporen:
  + Begin met minimaal script, voer het zo snel mogelijk uit!
  + Werk altijd stap voor stap
    - Iteratieve toevoegingen
  + Test voortdurend het resultaat van elke wijziging
  + Hou minstens 2 vensters open naast elkaar:
    - Editor
    - Terminal voor testen
  + Syntax check: **bash -n script.sh**
  + ShellCheck: **shellcheck script.sh**
    - Gebruik editor-plugin waar mogelijk
  + Druk veel info af (printf of echo)
  + Debug-mode:
    - **bash -x script.sh**
    - In het script: **set -x en set +x**
* Begin elk script met:

A picture containing calendar

Description automatically generated

* Booleans is bash:
  + In Bash bestaan geen booleaanse variabelen!
  + Er bestaan wel gelijknamige commando’s true en false
  + Booleaanse waarden zijn gebaseerd op **exit-status** v/e proces

A picture containing logo

Description automatically generated

* + A-blok wordt uitgevoerd als exit-status van COMMANDO 0 is (geslaagd, TRUE)
  + B-blok wordt uitgevoerd als exit-status van COMMANDO verschillend is van 0 (gefaald, FALSE)
  + Vb:

Text

Description automatically generated

* Operatoren **&&** en **||:**



* + Command2 wordt enkel uitgevoerd als command1 succesvol was (exit 0)



* + Command2 wordt enkel uitgevoerd als command1 niet succesvol was (exit ≠ 0)
* Commando **test**:
  + Evalueren van logische expressies
  + Geeft geschikte exit-status
    - 0 = TRUE
    - 1 = FALSE
  + Alias voor test is **[**
    - [ is een commando, geen “haakje” in de traditionele betekenis
    - spaties vóór en na!

Timeline

Description automatically generated with medium confidence

Automatisering:

* De kracht van automatisering:
  + Installeren van een server is terugkerende taak
  + Moet snel én foutloos kunnen gebeuren!
    - 1x, 10x, 100x, …
  + ⇒ Automatiseren is een noodzaak!
* Kwaliteitsbewaking bij serverinstallatie:
  + Gedetailleerde procedurehandleiding
    - SELab, Linux, System Engineering Project
  + Script voor automatisering
    - Linux, System Engineering Project
  + Configuration Management
    - Infrastructure Automation
* Tool: Vagrant
  + Command-line applicatie
  + Automatiseert aanmaken en configureren van (VirtualBox) VMs
  + Draait op Linux, Windows, MacOS
* Waarom Vagrant?
  + Snel nieuwe VMs aanmaken
  + Reproduceerbaar!
    - Enkel code, geen .ova van +4GB
    - Overdraagbaar naar ander platform
  + Aantal VMs onder controle houden
* Overzicht repo:

Graphical user interface, application

Description automatically generated

* + Vagrantfile: configuratiebestand van Vagrant
    - Hoef je niet aan te komen
  + vagrant-hosts.yml: overzicht VMs in de opstelling
    - incl. eigenschappen zoals IP-adres
  + provisioning/: installatiescripts voor VMs
* Werken met vagrant:
  + Vagrant status
  + Vagrant up db
  + Inloggen: vagrant ssh db

Table

Description automatically generated

* Wat gebeurt er?
  + Een nieuwe VirtualBox-VM wordt aangemaakt
    - Check in de GUI!
  + Een zgn. “base box” wordt zo nodig gedownload
    - = VM image met minimale installatie
    - Hier: bento/almalinux-8
    - Wordt later hergebruikt
  + De VM wordt geconfigureerd:
    - Naam, CPU, RAM, netwerkkaarten, …
    - IP-adres
    - Gedeelde map /vagrant
  + Het installatie-script wordt uitgevoerd
* Idempotentie:
  + Wiskundige eigenschap v/e operatie
  + Een operatie herhaald uitvoeren verandert het resultaat niet meer na de 1e keer
  + Server installatie: handeling wordt enkel uitgevoerd indien nodig
    - Sommige Linux commando’s zijn al idempotent (bv. dnf install)
    - Sommige commando’s zijn dat niet (bv. useradd)
    - ⇒ test of het nodig is bepaalde taak uit te voeren!
* Bekijk cursus voor oef!!!
  + Database gebruiken (bv trialsite): **USE trialsite;**
  + Enkele aanwijzingen:
    - Het provisioning-script moet volledig automatisch lopen en mag geen invoer van de gebruiker vragen. Gebeurt dit toch, dan zal het meteen stoppen.
    - De inhoud van deze Git-repository wordt binnen de VM gemount onder de directory `/vagrant`. Je kan hier gebruik van maken om bestanden te kopiëren naar de VM.
    - Werk in kleine stappen! Maak een kleine wijziging aan het script en voer het uit met `vagrant provision VM`. Controleer of dit het gewenste resultaat heeft. In een aparte terminal kan je inloggen op de VM en een en ander uitproberen.
    - Probeer regelmatig in de webbrowser van je Linux-Mint VM of de website daar kan getoond worden.
    - Als je denkt dat je klaar bent, doe je `vagrant destroy web` en `vagrant up web`. Je webserver zou in één keer moeten geïnstalleerd worden met alle nodige configuratiewijzigingen. Zonder verdere manuele handelingen moet je de website zien in de Linux-Mint VM.
    - **`vagrant provision VM`** - voer het installatiescript van de VM (die opgestart moet zijn) uit. Je vindt dit script in de directory `provisioning`, en het heeft dezelfde naam als de VM (bv. `db.sh` voor de "db"-VM). Als je een VM toevoegt, moet je ook een nieuw provisioning-script voorzien. Een provisioning-script wordt uitgevoerd als root, dus het is niet nodig bij elk commando `sudo` te gebruiken.

## 7. Complexe scripts, cronjobs

* Communicatie script/omgeving:
  + I/O: stdin, stdout, stderr
  + Positionele parameters: $1, $2, enz.
  + Exit-status (0-255)
  + Omgevingsvariabelen, vb:



* Functies in bash:
  + Een functie gedraagt zich als een script!
  + Oproepen: **functie\_naam arg1 arg2 arg3**
  + Positionele parameters: ${1}, ${2}, enz.
  + **Return STATUS** ipv exit

A picture containing text

Description automatically generated

* Vb **functie** in bash:

Graphical user interface, text, application

Description automatically generated

* **Case**:

Chart, scatter chart

Description automatically generated

Text

Description automatically generated with medium confidence

* Tips:
  + Zet positionele parameters om in beschrijvende namen
  + Maak lijnen niet te lang (gebruik \ op het einde van een regel)
  + Gebruik “lange” opties: maakt script leesbaarder
  + Gebruik lokale variabelen in functies
  + Deel script op in (herbruikbare) functies

Plannen van systeembeheertaken: **cronjobs**

* Processen op de achtergrond:

Graphical user interface, text, application

Description automatically generated

* + **Ctrl+Z** zet de uitvoer van een proces stil (nog niet afgesloten!)
  + **bg** start het proces terug op, maar in de achtergrond
  + **&** op het einde van een regel start proces op de achtergrond = combinatie van Ctrl+Z en bg
* Achtergrond processen beheren:

Table

Description automatically generated

* Processen eenmalig plannen:

Text

Description automatically generated

* + **at** zal binnen 2 minuten het opgegeven commando uitvoeren
  + Met **watch** herbekijken we elke 2s de inhoud van het doelbestand
  + Nog at voorbeelden:

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

* Processen herhaald plannen: cron
  + Bekijk **/etc/crontab**
  + Bevat taken die regelmatig gepland worden:
    - Tijdsaanduiding
    - Commando
  + Crontab per gebruiker:
    - Tonen: **crontab -l**
    - Bewerken: **crontab -e**

Table

Description automatically generated

Text

Description automatically generated with low confidence

## 8. Troubleshooting, SSH

* Download trouble-demo from learning path

Table

Description automatically generated

* + On webt, a PHP app runs a query on the dbt
  + dbt is set up correctly, webt is not
* Test the database server:

Graphical user interface, Word

Description automatically generated with medium confidence

* + Should work from
    - your Linux Mint GUI VM (if it is connected to intnet sudo apt install mysql-client
    - from demo VMs (/vagrant/query\_db.sh)
* Use a bottom-up approach:

Table

Description automatically generated

* **Network access layer:**
  + bare metal:
    - test the cable(s)
    - check switch/NIC LEDs
  + VM (e.g. VirtualBox):
    - check virtual network adapter type & settings
  + **ip link**
* **internet layer:**
  + checklist:
    1. Local network configuration
    2. Routing within the LAN
    - Know the expected values!
  + checking Local network configuration:
    1. IP address: **ip a**
       - **Ifup <interfacename>**
       - **Ifdown <interfacename>**
    2. Default gateway: **ip r**
    3. DNS service: **/etc/resolv.conf**
    - IP address?
    - In correct subnet?
    - DHCP or fixed IP?
    - Check configuration: **/etc/sysconfig/network-scripts/ifcfg-\***
    - Example dhcp:

A picture containing text

Description automatically generated

* + - Example static IP:

Text

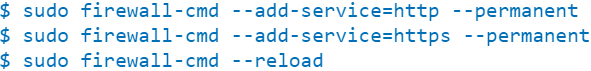
Description automatically generated with medium confidence

* + - Common causes DHCP:
      * No IP
        + DHCP unreachable
        + DHCP won’t give an IP
      * 169.254.x.x
        + No DHCP offer, “link-local” address
      * Unexpected subnet
        + Bad config (fixed IP set?)
      * Watch the logs: **sudo journalctl -f**
    - Common causes (Fixed IP)
      * Unexpected subnet
        + Check config
      * Correct IP, “network unreachable”
        + Check network mask
  + Local configuration: ip route
    1. Default GW present?
    2. In correct subnet?
    - Check network configuration
    - DNS server: **/etc/resolv.conf**
      * nameserver option present?
      * Expected IP?
  + Checking routing within the LAN:
    - Ping between hosts
    - Ping default GW/DNS
    - Query DNS (**dig, nslookup, getent**)

Text

Description automatically generated

* + Lan connectivity DNS:
    - **dig icanhazip.com**
    - **nslookup icanhazip.com**
    - **getent ahosts icanhazip.com**
* **transport layer:**
  + checklist:
    1. Service running? **sudo systemctl status SERVICE**
    2. Correct port/inteface? **sudo ss -tulpn**
    3. Firewall settings: **sudo firewall-cmd --list-all**
  + Is the service running?
    - **systemctl status httpd.service**
    - active (running) vs. inactive (dead)
      * **systemctl start httpd**
      * Fail? See below (Application layer)
    - Start at boot: enabled vs. disabled
      * **systemctl enable httpd**
  + firewall settings:
    - **sudo firewall-cmd --list-all**
    - Is the service or port listed?
      * Use **--add-service** if possible
    - Supported: **--get-services**
    - Don’t use both --add-service and --add-port
    - Add **--permanent**
    - **--reload** firewall rules



* + Correct ports/interfaces?
    - Use ss (not netstat)
      * TCP service: **sudo ss -tlnp**
      * UDP service: **sudo ss -ulnp**
    - Correct port number?
      * See **/etc/services**
    - Correct interface?
      * Only loopback?
* **Application layer:**
  + Checklist:
    - Check the logs: **journalctl**
    - Validate config file syntax
    - Use (command line) client tools
      * e.g. **curl, smbclient** (Samba), **dig** (DNS), etc.
      * **Netcat** (ncat, nc)
    - Other checks are application dependent
      * Read the reference manuals!
  + Check the log files:
    - Either journalctl: **journalctl -f -u httpd.service**
    - Or /var/log/:
      * **tail -f /var/log/httpd/error\_log**
  + check config file syntax:
    - Application dependent, for Apache: **apachectl configtest**
* **SELinux:**
  + SELinux is Mandatory Access Control in the Linux kernel
  + Settings:
    - Booleans: **getsebool, setsebool**
    - Contexts, labels: ls -Z, **chcon, restorecon**
    - Policy modules: **sepolicy**
  + Check file context:
    - Is the file context as expected?
      * **ls -Z /var/www/html**
    - Set file context to default value
      * **sudo restorecon -R /var/www/**
    - Set file context to specified value
      * **sudo chcon -t httpd\_sys\_content\_t test.php**
  + check Booleans:
    - **getsebool -a | grep http**
    - Know the relevant booleans! (RedHat manuals)
    - Enable boolean:
      * **sudo setsebool -P httpd\_can\_network\_connect\_db on**
* **general guidelines:**
  + Back up config files before changing
  + Be systematic, bottom-up
  + Be thorough, don't skip steps
  + Do not assume: test
  + Know your environment
  + Know your log files
  + Read The F\*\*\* Error Message!
  + Open logs in separate terminal
  + Small steps
  + Validate the syntax of config files
  + Reload service after config change
  + Verify each change
  + Keep a cheat sheet/checklist
  + Use a configuration management system
  + Automate tests
  + Don't ping Google!

## 9. Mount

* Pre knowledge:
  + Disk devices and partitions have already surfaced in other courses. We briefly depict how you can explore these devices on Linux
* Disk partitions:
  + Sata disk devices:

Graphical user interface

Description automatically generated with low confidence

* + Every disk:
    - maximum 4 primary/extended partitions
    - one extended partition can host further logical (sub)partitions

Table

Description automatically generated

* **fdisk**:
  + Display and modify partitions of a disk.

A picture containing text

Description automatically generated

* Linux file systems (common):
  + ext2
  + ext3 - with journaling
  + ext4 - latest version, with journaling
  + xfs
  + Other file systems:
    - vfat
    - ntfs
    - iso9660
* formatting a partition:
  + **mkfs** = MaKe FileSystem

Graphical user interface, text

Description automatically generated

* changing defaults of a file system:
  + Every partition has e.g. reserved blocks for root user only:

Graphical user interface, Word

Description automatically generated with medium confidence

* + Update with tune2fs e.g. reduce to 3% reserved blocks

Text

Description automatically generated with medium confidence

Mount:

* Manual mount:
  + mount = Making a partition available in the file tree
  1. make a mount point ~ a mount directory, e.g.



* 1. bind the partition to the mount point



* display mount points:

Graphical user interface

Description automatically generated

* permanent mounts:
  + Partitions which will be mounted at boot: **/etc/fstab**

Graphical user interface, text

Description automatically generated with medium confidence

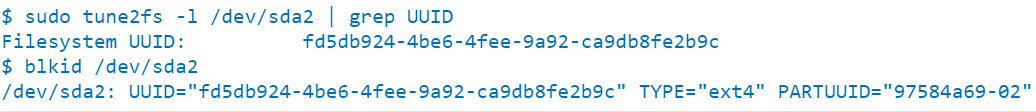
* Mount options:
  + **Mount** has some useful options:
    - **ro** - read-only
    - **rw** - read-write
    - **remount** - mount an already mounted device with new options

Graphical user interface, text, application

Description automatically generated

UUID

* UUID = universally unique identifier
  + 128 bits
  + generated while formating
* lookup UUID



* fstab with UUID
  + unique indication of partition in case e.g. sda and sdb get switched when booting

Graphical user interface, text, application

Description automatically generated

## 10. DNS met BIND

* DNS is in principe eenvoudig:
  + Vertaling van hostnaam naar IP in een tekstbestand
  + DNS query = opzoeking in dat tekstbestand
  + Queries kunnen over het netwerk gestuurd worden
* Poor man’s DNS:
  + Hosts-bestand: **/etc/hosts**

Text

Description automatically generated

* + On Windows: c:\Windows\System32\Drivers\etc\hosts
* Hierarchical domains:

Diagram

Description automatically generated

* + Iterative resolver:

A picture containing text, wrench, tool

Description automatically generated

* Root DNS servers:
  + 12 root servers: [a-m].root-servers.net
  + Verschillende instanties per root server!
    - Totaal: 1000+
    - vb. Brussels: e, f, i, l
  + Instanties v/e root server delen hun IP address!
    - Routers sturen trafiek naar dichtstbijzijnde instantie
* Types van een DNS server:
  + Authoritative: “bron van waarheid” voor een zone
    - Zonebestand
  + Forwarding/caching: stuurt requests door naar andere servers
  + Primary/Secondary (master/slave): voor “high availability”
    - Enkel primaire heeft zonebestand
    - Secundaire vraagt regelmatig “zone transfer”
* Best practices in productie:
  + Authoritative-only
    - Caching & authoritative niet mengen
  + DNS only
    - Geen andere services op die machine
  + Een typische AD DC overtreedt beide regels!

Interactie met DNS

* **Nslookup**:
  + Stuur vraag naar de DNS-server in /etc/resolv.conf



* + Stuur vraag naar specifieke DNS-server



* **Dig**:
  + Forward lookups

Graphical user interface

Description automatically generated

* + Reverse lookup



* + IPv6

Graphical user interface

Description automatically generated with medium confidence

* + Domain info

Graphical user interface, text, application

Description automatically generated

* 1. Wie is de authoritative name server voor hogent.be?
  2. Wie is de mail server voor hogent.be?
  3. Vraag de Start-of-Authority section voor hogent.be op
  4. Geef gelijk welk record met de opgegeven naam
  5. Vraag een domain transfer aan

BIND:

* Berkeley Internet Name Domain
  + Implementatie van het DNS protocol
  + Meest gebruikte
  + De facto standaard op Unix-achtige systemen
  + Lees DNS for rocket scientists!
* Installatie op Enterprise Linux:
  + Package: **bind**
  + Configuratie: **/etc/named\***
  + Zonebestanden, enz: **/var/named/**
* Hoofdconfiguratiebestand
  + /etc/named.conf

A picture containing graphical user interface

Description automatically generated

* Belangrijkste opties:
  + listen-on: port number + network interfaces
    - any;
    - 127.0.0.0/8; 192.168.76.0/24
  + allow-query: welke hosts mogen queries sturen?
  + recursion: recursieve queries toelaten
    - Zou no moeten zijn op een authoritative name server!
* Forwarding name server:
  + Als je de service nu opstart, heb je een forwarding name server

Table

Description automatically generated

* Configuratie zone:
  + Forward lookup zone voor example.com

Graphical user interface, text

Description automatically generated with medium confidence

* Zonebestand voor example.com
  + **/var/named/example.com**

**Table

Description automatically generated**

* Resource Records (RR):

Logo, company name

Description automatically generated

* Types Resource Records:

Graphical user interface, text, application

Description automatically generated

* Start of Authority:



* + srv.example.com.: primaire DNS-server
  + hostmaster.example.com.: email adres v/d sysadmin
    - hostmaster@example.com
  + 21120120: serial
    - LET OP secundaire servers zullen alleen update uitvoeren als serial verhoogd is
* Start of Authority: timeouts



* + **1D**: waneer zal secundaire ns proberen de zone te synchroniseren
  + **1H**: tijd tussen update-pogingen
  + **1W**: wanneer zijn zonegegevens niet langer “authoritative” (enkel op secondaire)
  + **1D**: hoe lang kan een NAME ERROR resultaat gecached worden
* Shortcuts
  + **$ORIGIN**: domeinnaam
    - Wordt toegevoegd aan alle namen die niet eindigen op .
    - @: wordt vervangen door waarde van $ORIGIN
  + **$TTL**: time to live (in seconden)
    - Hoe lang mag een record gecached worden
* (Un)qualified domain names:
  + Fully Qualified Domain Name (FQDN): eindigt met een punt
  + Unqualified: zonder punt
    - $ORIGIN toegevoegd aan het einde
* Tijdsaanduidingen:
  + default = seconden
  + M = minuten
  + H = uren
  + D = dagen
  + W = weken
    - Ook combinaties, bv. 2H30M
* Reverse lookup zone:

Graphical user interface, text

Description automatically generated with medium confidence

* Namen van een “reverse lookup zone”:
  1. Neem het IP-adres: 192.0.2.0/24
  2. Laat het host-deel vallen: 192.0.2
  3. Keer volgorde om: 2.0.192
  4. Voeg in-addr.arpa. toe
  + Resutaat: 2.0.192.in-addr.arpa.
* Zonebestand:
  + **/var/named/2.0.192.in-addr.arpa**

**Table

Description automatically generated**

* Root hint:
  + Elke (forwarding) name server moet een lijst bijhouden van de root name servers



Timeline

Description automatically generated

Labo-opdracht:

* Nuttige commando’s:
  + Controleer de logs: **journalctl -f -l -u named**
  + Valideer configbestanden:
    - Hoofdbestand: **named-checkconf**
    - Zonebestanden: **named-checkzone ZONE FILE**

Text

Description automatically generated

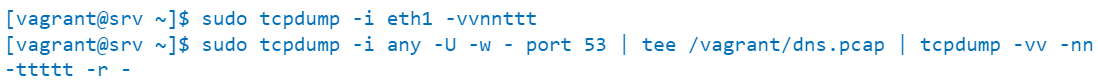
* Tips:
  + Query log aanzetten:



* + BIND logs tonen:



* Netwerkverkeer opvangen:



* 1. Interactie met de Linux Mint VM
  2. Alle verkeer op poort 53, opslaan in dns.pcap
  + Bestand dns.pcap kan je openen met Wireshark!