Resume

Most of the stuff here comes from the .txt with notes. There are some cases where I copied the information from the man pages or Internet.
Some sections like **`vim`** or **`kickstart`** aren't present, others were reduced (in comparison to the original note files), due printing reasons (paper and ink are expensive).

`ls` & `redirect symbols`

Why two different things share a table? Because I'm trying to save space

ls option	Description	Redirect Symbol	Description
l	Extended output	< <filename>	uses file as stdin
ld	Directory output	>	stodout overwrites file
a	Shows all files, even hidden ones	>>	sdtout appends to file
Z	SELinux context	2> <filename>	stderr to file
R	Recursive	2>1	stderr to stdout
		&> <filename>	stdout and stderr to file name

`touch` command

Create files if they don't exist, otherwise, modify the timestamp.

`touch foo` creates the file **`foo`**

`uname` command

`uname -rms` show the current kernel.

`ln` command

The purpose of **`ln`** is create another name for a file. Reference the same contents of the file but with another name.

If you delete a file with a hard link, the content will be available on the hard link.

If you delete a file with a symbolic link, the symbolic link won't work.

```
ln [source] [name of the link]
```

`ln fileA fileB` creates a link where fileA is the original

`ln -s fileA symfileB` creates a symbolic link

`grep` command

Option	Function
-i	case insensitivity

Option	Function
-v	lines without matches
-r	recursive search
-A `[n]`	display X of lines after the match
-B `[n]`	display X of lines before the match
-e	multiple RegEx can be supplied as OR
-n	display line number

RegEx

Symbol	Usage	Example	Applies for
^	beginning of the line	^cat	category
$	end of the line	$dog	chilidog
.	wildcard single character	c.t	cat/cbt/cct
*	any amount of characters	c*t	cat/cbt/caaaaat
.*	zero to infinitely characters	c.*t	ct/cat/coat/culvert
.\{\}	explicit multiplier	c.\{2\}	coat
\< \>	word boundary	\<ipsum\>	Lorem ipsum et
[ ]	options for a single character	c[abc]t	cat/cbt/cct

`locate` & `find`

`locate [search term]` search every file with the search term on it's name.

`locate -i [search term]` case insensitive.

`locate -n [n] [search term]` search and stops after `n` results.

`updatedb` update the locate database.

`find [directory to start] [search term]`

Option	Function
-user	search files that belong to that username
-uid	same as -user but with the UID
-group	search files that belong to that group
-gid	same as -group but with GID
-perm `[permissions]`	search for permissions based on the operator
	764 only `-rwxrw-r--`
	-324 at least `--wx-w-r--`

Option	Function
	/442 **`u`** **`r--`** OR **`g`** **`r---`** OR **`o`** **`-w-`**
-size **`[n][k,M,G]`**	search by size (round up to single units 995 KiB = 1MiB)
	+10M more than 10 MiB
	-1G less than 1 GiB
-mmin **`[n]`**	modified files since at least **`[n]`** minutes
-type	**`r`** regular file, **`d`** directory, **`l`** symlinks, **`b`** block device
-links	regular files with more names

`find /home -user foo` find all the files that belong to **`foo`**

`find / -type l -links +3` find all the symbolic links with 3 or more names.

Users

`/etc/passwd` contains the local user information.
`/etc/shadow` contains the user's passwords.
`/etc/group` contains the local group information.
`/etc/login.defs` contains the default parameters of accounts, such as password age.

`authconfig --passalgo [algorithm]`

`useradd [username]`

`userdel [username]` (add -r to remove the home directory)

`usermod [username]`

`usermod -s /sbin/nologin [username]` the user won't be able to log in.

Most of these options works for **`useradd`** and **`usermod`**

Option	Description
-a, --append	add the user to the supplementary group(s). use only with -G
-c, --comment **`[COMMENT]`**	full name of the user for the GECOS field.
-d, --home **`[HOME_DIR]`**	specify user's home directory
-e, --expiredate **`[EXPIRE DATE]`**	date on which the user account will be disabled (YYYY-MM-DD)
-f, --inactive **`[INACTIVE]`**	number of day after password expires until the account is disabled
-g, --gid **`[GID]`**	specify primary group
-G, --groups **`[GROUPS]`**	supplementary groups
-m, --move-home	moves the user's home directory to a new location, use with -d
-s, --shell **`[SHELL]`**	specify a new login shell for the user

Option	Description
-L, --lock	lock the user's account
-U, --unlock	unlock the user's account

Groups

`groupadd -g [GID] [group]` adds a group with the specified ID and name.

`groupmod -g [GID] [group]` changes the ID of the specified group (`-n` to change name).

`groupdel [group]` deletes the specified group.

`gpasswd -d [user][group]` remove the user from the group.

Password

```
    ||--------------max days (-M)------------||
    |                |      |                 |
    |                |      |                 |
    ||-min days (-m)-|      |-warn days (-w)-|||-inactive days (-I)-|
time|-----------------------------------------|---------------------|
    |                                         |                     |
last change date (-d)             password expiration date  inactive date
```

`chage -l [username]` list user's current settings.

`chage -E YYYY-MM-DD [username]` makes the account expire n the specified date.

`chage -d 0 [username]` forces a password change on the next login.

`chage -m 0 -M 90 -W 7 -I 14 [username]` change the settings to 0 days required to change password, 90 days for the password to expire, warning of password expiring 7 days before it happens, 14 days before the account inactivation.

Option	Description
-d `[n]`	change the last time the password was changed
-E `[YYYY-MM-DD]`	set date of account's expiration
-I `[n]`	days before the password becomes inactive
-m `[n]`	minimum age/time before changing the password
-M `[n]`	maximum age of the password
-W `[n]`	warning before the password expiration date

Permissions

Standard permissions

Word model

`chmod WhoWhatWhich <filename>`

r read, **w** write, **x** execute.

`chmod g=rw- foo` sets read and write for the group of the file *foo*.

`chmod u+x script` adds the execute permission for the owner of the file *script*

Shared table for the Word model

Word	Operator	Permission	Special bit
u (owner)	+ (add permission)	r (read)	s (suid, using u)
g (group)	- (remove permission)	w (write)	s (sgid, using g)
o (other)	= (set permission)	x (execute)	t (sticky, only directories)

Numeric model

Number	Permission	Special bit
4	read	suid
2	write	sgid
1	execute	sticky

`chmod 0700 foo` equivalent to `-rwx------`

`chmod 4554 script` equivalent to `-r-sr-xr--`

`chmod` supports `-R` for recursive operations.

`chown [user]:[group]` change file ownership.

`umask`

Change the default permissions applied to a new created file/directory using `umask`.

Write the value for the permissions excluded.

`umask 0022` new files will be created as `-rwxr--r--` and `drwxr--r--`.

ACLs

Check if a file has ACLs using `ls -l [file]`. If a `+` symbol is present next to the permissions column, then it contains ACLs.

You can set explicit permissions for users and groups that aren't the owner or primary group of the file.

Each ACL has a mask that gets recalculated every time you modify the ACL settings of a file.

The mask limits what permissions are effective (if the mask is `r--`, ACLs with `rw-` won't make use of the write permission).

`getfacl <filename>` get the ACL settings of the specified file. The command still works even if the file doesn't have any ACL settings.

`setfacl [option] [permissions]`

Option	Description
-m	modify the ACL of a file or directory
-x	remove the ACL entry of a file or directory
--set-file=	apply the ACL from another file (use the **`getfacl`** output)

`setfacl -m u:foo:r notes.txt` add (modify if it's already present) an entry specifying that the user **`foo`** has read permision on the file.

`setfacl -m o:: notes.txt` changes the **`others`** permissions to **`---`**

`setfacl -x u:foo: notes.txt` removes the entry for the user **`foo`**. Note that you don't need to specifiy any permissions, just leave the last field empty.

`getfacl fileWithACL | setfacl --set-file=- newFile` uses the output from the **`getfacl`** command and uses it to set the ACLs on **`newFile`**.

`setfacl -m m::r <filename>` modify the mask to only allow the read permission.

`setfacl -m d:u:rx <directory>` modify the default ACLs of the directory.

`setfacl -k <directory>` remove all default settings on a directory.

`setfacl -b <directory>` remove all ACLs on a directory.

Processes

`ps aux` processes with **`USER PID %CPU %MEM TTY STATUS`**.

`ps lax` long listing style, avoid username lookup.

`ps -ef` display all processes.

`ps j` jobs running.

Process status

Name	Flag	Kernel state	Description
Running	R	TASK_RUNNING	executing on a CPU or waiting to run
Sleeping	S	TASK_INTERRUPTIBLE	waiting for some condition (hw request, resources, signal)
	D	TASK_UNINTERRUPTIBLE	sleeping but won't respond to signals
	K	TASK_KILLABLE	like D but waiting for a signal to be killed
Stopped	T	TASK_STOPPED	stopped by being signaled (by user or another process)
Zombie	Z	EXIT_ZOMBIE	child process signals it's parent as it exists. Free resources
	X	EXIT_DEAD	parent reaps the remaining child process structure. Now free

Jobs

Useful when you have access to only ONE terminal.

`[command] &` the ampersand moves the program to the background automatically.

`jobs` display running jobs on the background.

`fg %[job ID]` bring job to the foreground.

`bg %[job ID]` resume stopped process in the background.

`Ctrl + Z` suspends the process and send it to the background (use before `bg`).

`kill %[job ID]` kill the job running in the background.

`kill` command

`man 7 signal` for more details.

Number	Name	Definition	Purpose
1	SIGHUP	Hangup	report termination of the controlling process of a terminal
2	SIGINT	Keyboard interrupt	interrupt from keyboard (`Ctrl + C`)
3	SIGQUIT	Keyboard quit	quit from keyboard (`Ctrl + \`)
9	SIGKILL	Kill, unblockable	abrupt program termination. Always fatal
15	SIGTERM	Terminate	termination signal, process should close properly
18	SIGCONT	Continue	resume process if stopped
19	SIGSTOP	Stop, unblockable	suspend the process
20	SIGTSTP	Keyboard stop	can be blocked or handled (`Ctrl + Z`)

`kill [PID]` kill the process with the default signal (SIGTERM,15).

`kill -[signal] [PID]` send the specified signal (name or number).

`kill -l` list all the available signals.

`killall [command pattern]` kill all the processes that matches the command pattern.

`killall -[signal] [command pattern]` send the specified signal to all the process that matches the command pattern.

`killall -[signal] -u [username] [command]` same as before but only those that belong to the specified user.

`pkill` command

It's like killall, and uses an advanced selection criteria.

Use `pgrep` to check which processes will be affected

Option	Name	Description
`[command]`	Command	processes matching that command
-U	User ID	processes owned by that user
-G	Group ID	processes owned by that group
-P	Parent	processes belonging to that parent process
-t	Terminal	processes controlled by that terminal

`pkill [command pattern]`

`pkill -U 1000` kill all the processes that belong to the user with ID 1000.

`pgrep -l -u foo` display all the processes running by the user **`foo`**

`w -f` display who's logged into the system and their activities.

`pstree -p [username]` tree representation of the processes running by the specified user.

Process activity

`uptime` display the load average of the last 1, 5 and 15 minutes.

`grep "model name" /proc/cpuinfo | wc -l` Count the cores of the machine (both physical and hyperthread ones).

Divide each number by the amount of cores. If the result is greater than 1 (>1), the CPU is overloaded.

`top` real-time process monitoring

List of columns

Name	Description
USER	process owner
VIRT	virtual memory is all the memory that the process is using
RES	physical memory used by the process
S	process state.
	[D] uninterruptable sleeping [R] Running or Runnable
	[S] Sleeping [T] Stopped or Traced [Z] Zombie
TIME	total processing time since the process started
COMMAND	process command name

Keystrokes

Key	Purpose
? / h	help for interactive keystrokes
l t m	toggles for load, threads and memory header lines

Key	Purpose
1	toggle showing individual CPUs or a summary in header
s	refresh rate in decimal seconds (0.5,1,5)
b	reverse highlighting for Running processes; default = bold
B	enables use of bold in display
H	toggle threads
u,U	filter for username
M	sort by memory usage
P	sort by processor utilization
k	kill a process, ask for PID and signal
r	renice a process, ask for PID and nice_value
W	save the current display configuration for the next restart
q	quit

`nice` & `renice`

Nice levels of a process goes from -20 to 19 for users.

`top` displays them from RT,-99 to 39.

Nice level of 20 for users translates as 0 for `top`.

Use `nice` for run programs, `renice` for already running programs.

`nice -n [nice level] [command]` run the program with the specified nice level.

`renice -n [nice level] [PID]` renice the process that is already running.

systemd & boot process

`systemctl -l` show what's running on the system without abbreviate the names.

`systemctl [option] [unit]`

The most common units: `service`, `socket`, `path`. Some processes has different units (like the `cups` process)

Option	Function	Option	Function
start	starts the unit	reload	reload the configuration of the unit (keep PID)
stop	stops the unit	restart	restarts the unit (new PID)
enable	allow unit to run at boot time	disable	prevent unit from running at boot time
is-enabled	check if the unit is enabled	is-active	check if the unit is active

Option	Function	Option	Function
status	display the status of the unit	mask	disable and hide unit

`systemctl` is also used for the boot targets.

A target is used to declare that we reached certain point in the boot process. Their names ends with **`.target`**

`systemctl list-units --type=target --all` display all the available targets and their current status.

`systemctl list-dependencies [target].target | grep target` display all the dependencies for that target.

`systemctl isolate [target].target` stops all the services that aren't required for the specified target. Not all targets can be isolated, only those with the **`AllowIsolate=yes`** flag.

Important targets

Name	Usage
graphical	system supports multiple users, graphical and text-based logins
multi-user	system supports multiple users, text-based logins only
rescue	sulogin prompt, basic system initialization completed
emergency	sulogin prompt, initramfs pivot complete and system root mounted on / read-only

`systemctl set-default [target].target` change the default target.

You can override the default target at boot time by appending **`systemd.unit=[target].target`** to the kernel line.

Changing the root password

1. Edit the GRUB entry of the system.
2. Search the line that starts with **`linux16`**
3. Append **`rd.break`** to the end of the line.
4. Press **`Ctrl + X`** to boot with the changes.
5. System will load and present a root shell. The actual boot system is mounted as read-only on /sysroot.
6. Remount the system with read-write permissions **`mount -oremount,rw /sysroot`**.
7. Use **`chroot`** to treat **`/sysroot`** as the root of the file system tree **`chroot /sysroot`**.
8. Change the password of **root** **`passwd root`**.
9. Create the file **`.autorelabel`** to relabel the whole system with the right SELinux context **`touch /.autorelabel`**
10. Execute **`exit`** twice and the system will finish the boot process.

GRUB (GRand Unified Bootloader)

grub2 is the default boot loader on RHEL 7.

The main configuration is located at **`/boot/grub2/grub.cfg`** but you're not supposed to edit that file directly.

`grub2-mkconfig` generates a new config file.

`grub2-mkconfig > /boot/grub2/grub.cfg` generates a new config file and applies the changes permanently.

It's recommended to send the output to another file and review the changes before apply them.

`grub2-install` reinstalls the boot loader in case it's corrupt.

SELinux

`/etc/selinux/config`

Recommended packages

Package	Description
`policycoreutils-python`	adds the **`semanage`** command
`selinux-policy-devel`	more man pages related to SELinux
`setroubleshoot-server`	adds the **`sealert`**

`sepolicy manpage -a -p /usr/local/man/man8` creates the SELinux man pages.

Security Enhanced Linux (SELinux) is an additional layer of system security.
Every single file in the system has a tag or context assigned.
SELinux labels have several contexts: user, role, type, and sensitivity.
RHEL uses the targeted policy by default, bases it's rules rules on the third context: type.

Every process goes through the SELinux vector table to look up what is allowed to do and which files are going to be used.
If the process is not allowed to do certain action or use certain file, an alert will be emitted.

By default, everything on Linux is denied. You can allow processes to do their stuff with policy rules.
There are three modes for SELinux:

Mode	Description
Enforcing	denies access to everything without explicit policies for that behaviour
Permissing	used to troubleshoot. Allow any interaction and logs the ones that should be denied.
Disabled	turns off SELinux. Requires a reboot to remove the labeling of SELinux.

It's better to use permissive mode than disable SELinux. The kernel will automatically maintain SELinux file system labels as needed, avoiding the need of relabeling during the system reboot.

`getenforce` shows the current SELinux mode.

`setenforce [Enforcing|Permissive|1|0]` changes the SELinux mode. Or we can edit the **`/etc/selinux/config`** file.

SELinux also has Booleans that can be used to tune the policy doing selective adjustments.

`getsebool -a` display all the current Booleans and their values.

Changing SELinux contexts

We can change contexts with the command `chcon` but it's not persistent.

`chcon -t [context] <filename>` changes the context of the specified file.

Using the `semanage` command we can do persistent changes.

`semanage` is part of the package `policycoreutils-python`, maybe you'll have to install it.

`semanage fcontext -l` show all the contexts on the database (supports RegEx).

`semanage fcontext -a -t [context] [folder]` add a new rule on the SELinux database. From now, every time you restore the context of the files inside the specified folder, the specified context will be applied.

`semanage fcontext -a -t httpd_sys_content_t '/virtual(.*)?'` set the context `httpd_sys_content_t` to the files inside of `/virtual`.

`restorecon -Rv [directory]` restores the context of the directory.

Remember to use `restorecon` after changing the directory's context.

`getseboolean -a` list all the current booleans and their current status.
`getseboolean [Boolean name]` shows the status of the specified Boolean.

`setsebool [Boolean] [on|off]` toggles the Boolean.

`setsebool -P httpd_enable_homedirs on` set the `httpd_enable_homedirs` Boolean `on` and makes the change persistent (`-P`).

`semanage boolean -l` list all the Booleans with their current status, default value and description (use `grep` to filter what you're looking for).

`semanage boolean -l -C` show all the Booleans which value has been changed.

Troubleshooting SELinux

There are times where SELinux may deny something. Most of the time the issue is an incorrect file context.
Check SELinux messages on `/var/log/audit/audit.log` using the command `sealert -l`.

The package `setroubleshoot-server` must be installed in order to use `sealert`

`sealert -a /var/log/audit/audit.log` search and display SELinux messages in the `audit.log` file.

`sealert -l [UUID]` display more information about the SELinux violation.

scontext is the source of the problem
tcontext is the target that the service was trying to do something to.

`grep [service] /var/log/audit/audit.log | audit2allow -M mypol` generate a local policy module.

`semodule -i mypol.pp` enable the policy we created.

`tar` command

`tar [options]`

Option	Description	Option	Description
c	create an archive	x	extract an archive
f	name of the archive to work with	t	list the contents of the archive
p	preserve the permissions of files	P	don't strip leading `/` from absolute paths
v	verbosity	compression	`z` gzip, `j` bzip2, `J` xz

`tar cf [resulting file name] [files to add]` this will create an archive.
Even if we don't use extensions on UNIX, it's good to add **`.tar`** at the end of the file.

`tar czf /root/foo.tar.gz /etc` creates a gzip-compressed tar archive, using the contents of the **`/etc`** folder.

`tar cjf /root/backup.tar.bz2 /var/log` creates a bzip2 archive.

`tar cJf /root/bar.tar.xz /etc/selinux` creates a xz archive.

`tar xzf /root/foo.tar.gz` extracts the content of the archive.

Logfiles

rsyslogd

/var/log	Description
messages	most syslog messages are logged here (except auth and email processing)
secure	security and authentication-related messages and errors (permissions and stuff)
maillog	mail server-related messages
cron	periodically executed tasks
boot.log	system startup-related messages (check first for troubleshooting boot problems)

Every message comes from a facility with a level of priority

Code	Priority	Severity	Code	Priority	Severity
0	emerg	system is unusable	4	warning	warning condition
1	alert	action must be taken immediately	5	notice	normal but significant event
2	crit	critical condition	6	info	informational event
3	err	non-critical error condition	7	debug	debugging-level message

`man 1 logger` for more information.

`/etc/rsyslog.conf` contains predefined rules.

New rules must be created on files inside of `/etc/rsyslog.d` and end with `.conf`

`auth.* /var/log/mostsecure.log` all messages from the `auth` facility will be logged on `/var/log/mostsecure.log`.

`*.info;mail.none;authpriv.none;cron.none /var/log/messages` all the messages with priority above `info` (6) will be logged on `/var/log/messages`, except those that comes from the `mail`,`auth` and `cron` facilities.

Syslog entries have a defined format based on `timestamp:host:process:message` (you can add your own format on `/etc/rsyslog.conf`).

`logger -p [facility].[level] [message]` sends a fake message (useful to test configurations).

`journalctl` command

Provided by **systemd**, writes the log on `/run` so it won't be saved by default.
`mkdir /var/log/journal` this will make `journalctl` logs persistent. Remember to assign the right permissions to this folder:
`chown root:systemd-journal /var/log/journal`
`chmod 2755 /var/log/journal` equivalent to `rwxr-sr-x`.
Still won't be permanent, you need to change the rotation time on `/etc/systemd/journald.conf`, then send the `USR1` signal to `systemd-journald`.

`journalctl -n [n]` display `n` amount of lines.
`journalctl -p [priority name or number]` display the messages with the specified priority.

`journalctl -f` real time output.

`journalctl --since [date (today| YYYY-MM-DD HH:MM:SS)] --until [date (today) | YYYY-MM-DD HH:MM:SS]` display the messages since the `--since` date to the `--until` date.

`journalctl -o verbose` shows more information like:

Verbose	Description		
_COMM	name of the command	_EXE	path of the executable for the process
_PID	PID of the process	_UID	UID of the user running the process
_SYSTEMD_UNIT	**systemd** unit that started the process		

`journalctl _SYSTEM_DUNIT=[unit].[type of unit] _PID=[PID]` display the logs of the specified process.

`journalctl -b` display the last boot messages.
`journalctl -b -1` output of the previous boot.

Time & date

Make sure that your system's time is accurate.

`timedatectl` display information about how the system time is configured.

timedatectl option	Description		
list-timezones	list available timezones	set-ntp	enable or disable NTP synchronization
set-timezone	set the time to the selected timezone	set-time	set time using `YYYY-MM-DD hh:mm:ss`

`tzselect` select timezone interactively.

`chrony` & NTP

`chronyd` is used to synchronize our system with an NTP server.
It uses servers from the NTP Pool Project (it can be changed to local servers).

In order to add an NTP server, we have to add a line on `/etc/chrony.conf`
`server classroom.example.com iburst` the option `iburst` uses four measurements in a short period of time for a more accurate initial clock synchronization.
Restart `chronyd` after making changes.
`chronyc sources -v` list the NTP servers that we're connected to.

Scheduling tasks

`at` command

The `at` is a small and powerful command that let us schedule tasks that won't be repeated

`at <TIMESPEC> [command]`

The `<TIMESPEC>` is quite flexible. You can use many different combinations.

`echo touch /root/hello | at now +1min` add a job to create the file `hello` in 1 minute from the moment it's executed.

`at noon +4 days < myscript` add a job to execute the file `myscript` at noon in four days since today.

`at <TIMESPEC> -q [queue] [command]` you have 26 queues (from a to z) to schedule tasks.

`at -l` shows the current queue.
`atq` same as `at -l`.
`atrm [job]` remove the specified job.

`crontab` command

The benefit of `crontab` is that you can schedule recurring tasks.

Option	Description
-e	edit jobs for the current user
-l	list the jobs for the current user
-r	remove all jobs for the current user
-u	manage the jobs of another user (only **root**)

`crontab <filename>` if you specify a file, all the jobs will be removed and replaced by the jobs of that file. If no filename is specified, `stdin` will be used.

Job Format

```
Minutes Hours Day-of-Month Month Day-of-Week Command
* * * * * command
```

Symbol	Description
`*`	Don't care/always
`0-9`	number to specify a number of minutes or hours,a date or a week day (0 and 7 = Sunday, 1 = Monday)
`x-y`	range starting on `x` and ending with `y` both are included
`x,y`	lists, can include ranges (5,10-13,17)
`*/x`	indicate an interval of `x`
Three letter abbreviation	Month (Aug, Oct, Nov, Dec), weekday (Tue, Thu, Mon, Sun)

For the `command` part, we can use `%` to create a new line. It will be considered `stdin` for the `command` we're executing.

`0 9 2 2 * /usr/local/bin/yearly_backup` execute `yearly_backup` every February 2 at 9:00, doesn't matter the week day.

`*/7 9-16 * Jul 5 echo "Chime"` execute `echo "Chime"` during July but only on Fridays, from 9:00 to 16:59, repeating after 7 minutes.

Scheduling system `cron` jobs

System cron jobs are defined in two locations: `/etc/crontab` and `/etc/cron.d/`.
Some packages install `cron` jobs and place them on `/etc/cron.d/`

Predefined folders for hourly, daily, weekly and monthly jobs can be found on `/etc`.
The directories are `cron.hourly cron.daily cron.weekly cron.monthly`.
Any scripts inside those files must have the execute permission activated.

`/etc/anacrontab` keep track of the scripts and the last time they were executed.

`systemd-tmpfiles` command

`systemd-tmpfiles` reads configuration files located at `/usr/lib/tmpfiles.d/*.conf`, `/run/tmpfiles.d/*.conf` and `/etc/tmpfiles.d/*.conf`.

```
systemd-tmpfiles [option]
```

Option	Description
`--create`	create files and directories specified on the configuration files
`--clean`	remove all files with an age parameter configured

Configuration files format

```
Type Path Mode UID GID Age Argument
```

Column	Description
Type	action that systemd-tmpfiles should take
Path	path to file
Mode	permissions of the file/directory
UID	owner of the file
GID	group of the file
Age	maximum age of the file
Argument	depends on `Type`, written to the new file or used for a symlink

Action	Description
d	create directory if it doesn't exist yet
D	create directory if it doesn't exist yet or empty it if already exists
f	create file if it doesn't exist. `Argument` will be the content of the file
F	create or truncate a file. `Argument` will be the content of the file
L	create a symbolic link. `Argument` will be the file to reference
Z	recursively restore SELinux context and file permissions

`d /run/systemd/seats 0755 root root -` create a directory called `seats` on the `/run/systemd` directory with the permissions `rwxr-xr-x` that belongs to the user and group `root`.
This directory won't be automatically purged.

`D /home/student 0700 student student 1d` create a directory for the user and group `student` with `rwx------` permissions, it will be automatically deleted after 1 day.

`L /run/fstablink - root root - /etc/fstab` create a symbolic link to `/etc/fstab`, it won't be automatically purged.

Configuration files priority

If we have a configuration file that repeats it's name across `/etc/tmpfiles.d`, `/run/tmpfiles.d` and `/usr/lib/tmpfiles.d`, they have certain priority of which file gets to run.

`/etc/tmpfiles.d` -> `/run/tmpfiles.d` -> `/usr/lib/tmpfiles.d`
`/etc/tmpfiles.d` is top priority, then `/run/tmpfiles.d`, and last `/usr/lib/tmpfiles.d`.

Software management

`yum` command

`yum` is a command line tool that knows how to install programs and also knows their dependencies and the relationships between packages.

Option	Description		
help	display usage information	list	list all the packages available to

Option	Description		
			install
repolist	list all the available repositories		`package name` search this package (or another with similar name)
	use the keyword `all` to display all of them, enabled and disabled		`installed` list all the installed packages
search	search a package that matches the keyword	info	display information about the package specified
provide	search the package that provides the specified file	install	install the specified package (can be used with `.rpm` files)
update	update the specified package	remove	removes the specified package
history	show the list of transactions		
	`undo [n]` reverses the `n` amount of transactions		

Group options

You can install whole groups of packages

Option	Description	
`list`	show all the package groups availables	
`install`	install the specified group	
`mark`	marks the group as installed, missing packages will be install on the next update	
`info`	display more information about the group	
	= package was installed with the group	+ will be installed with the group
	–isn't installed and won't be installed with the group	`no marker` is installed but not with the group

`yum update kernel` update the kernel.
`yum install cowsay` install the package `cowsay`

Adding repositories

Repository files are located at `/etc/yum.repos.d/`.

`yum-config-manager --add-repo="[repository URL]"` this will create the proper `.repo` file for that repository.
This command belongs to the `yum-utils` package.

```
[Repository]
name=Super Repo
```

```
baseurl=http://myfirstrepo.com/
## if it's a 0, the repository is defined but not searched by default.
enabled=1
## check the public key when you grab or install a package from that repository.
gpgcheck=1
## where is the public key located
gpgkey=file:///etc/pki/rpm/gpg/RPM-GPGP-KEY
```

`rpm` command

RPM files keep a naming scheme
`name-version-release.architecture`
`httpd-tools-2.4.6-7.el7.x86_64.rpm`

`rpm -q [option] [package/file]` query information about the specified package/file.

Option	Description
-p	display information about the `.rpm` file specified
-f	what packages provides the specified file
-l	list of files installed by the specified package
-c	list of configuration files
-d	list of documentation files
--scripts	list of scripts that may run on install or removal of the package
--changelog	show the changelog of the specified package

`rpm -i [package]` install the package.

Network

We use the TCP/IP standard. TCP is used for large data, UDP for queries.
IPv4 addresses are made out of four octets.
Each IP address has a prefix which take part of the four octets available.

`172.17.5.3/16` means `172.17` is the network and `5.3` the host.
The network is the prefix.
Also, each IP has a netmask:
`255.255.0.0` where `255.255` belongs to the network and `0.0` to the host

Network	Host	Prefix
172.17	.5.3	/16
255.255	.0.0	
192.168.5	.3	/24
255.255.255	.0	

The machine on the subnet connects to the Gateway, which contacts with the rest of the world, for incoming or outcoming connections.
The Gateway connects to the internet using the public IP assigned by the DNS server owned by the

ISP.

`0.0.0.0/0` is the default gateway.

Each network device has a MAC address. Also, their naming scheme on the system depends on how the BIOS recognizes the device:

Interface	Short name	Location	Short name
Ethernet	en	On-board	o
WLAN	wl	Hotplug	s
WWAN	ww	PCI	p

`enp6s0` translates as Ethernet PCI

`ip address` display information about the device and IP address
Note: commands like `ifconfig` and `netstat` are now deprecated.

`ip -s link show` show stats of the interface.
`ip route` display routing information.
`ping -c[n] [ip/domain]` ping the `[ip/domain] n` amount of times.
`tracepath [domain]` traces the path to reach the specified domain.

`ss -ta` socket statistics, `-t` for TCP sockets, `a` for all; display all the services running and what ports they're running on.

Option	Description	Option	Description
-n	numbers instead of names	-t	TCP sockets
-u	UDP sockets	-l	only listening sockets
-a	all sockets	-p	process using the sockets

IP Forwarding

`net.ipv4.ip_forward = 1` add this line to `/etc/sysctl.conf`
After that, you need to apply the changes using `sysctl -p`

NetworkManager

Configuration files on `/etc/sysconfig/network-script`
`man nm-settings`

Use `nmcli` to manage NetworkManager. Any changes to files that you do without using `nmcli` will be overwritten. You must turn on NetworkManager and do a `connection reload`, then down and up the connection.

`nmcli device [option]` manage devices (you can use `d`, `dev` instead of `device`).

Option	Description
status	list all devices
dis	bring down an interface and temporarily disable autoconnect

`nmcli net off` disable all manages interfaces.

`nmcli connection [option] [name of connection]` manage connections (you can use `c`, `conn` instead of `connection`).

Option	Description
show	view basic network information (more if you specify the connection name)
up	activate a connection
down	deactivate a connect (restart if autoconnect is on)
add	add connection
mod	modify a connection
del	delete a connection
reload	reloads configurations based on your manual changes

`nmcli con add help` shows all the options that can be used with this command.

Basic options for connections

Common Options	Description
type	`ethernet wifi wimax ppoe` and more
ifname	device name
con-name	connection name
autoconnect	`yes` (default), `no`

There are many type-specific options, some are better for wired connections, others for wireless.

Note: ipv4 and ipv6 options are accessed using a dot `ipv4.addresses`.

IPv4 Options	Description
addresses	set the IPv4 address and gateway
dns	set the DNS
method	set `auto` for DHCP, `manual` for static
gateway	use when modifying the connection

`nmcli c a con-name "Wired Connection X" ifname enp0s3 type ethernet autoconnect yes ipv4.addresses "192.168.1.10/24" ipv4.gateway "192.168.254.254" ipv4.dns "192.168.254.254" ipv4.method manual` create a new static connection.

`nmcli c m "Wired Connection X" +ipv4.addresses "10.0.0.1/24"` the `+` means we're adding another value instead of replacing the current one.

`nmcli c a con-name "Dynamic" ifname enp0s3 type ethernet autoconnect yes ipv4.method auto` create a new DHCP conection.

Configuration Options for `ifcfg` File

Static	Dynamic	Either
BOOTPROTO=none	BOOTPROTO=dhcp	DEVICE=eth0
IPADDR0=`172.25.x.10`		NAME=`"System eth0"`
PREFIX0=`24`		ONBOOT=`yes`
GATEWAY0=`172.25.x.254`		UUID=`some UUID`
DEFROUTE=`yes`		USERCTL=`yes`
DNS1=`172.25.254.254`		

`USERCTL` allows non-root users to modify the network.

Hostname

Hostnames aren't configured on the `/etc/hosts` file
The static host name is stored on `/etc/hostname`. If the file doesn't exist, a hostname hasn't been defined.

`hostnamectl status` display information about the hostname.

`hostnamectl set-hostname [hostname]` change the hostname of the machine.

`getent hosts [hostname]` test host name resolution with the `/etc/hosts` file.

`host [hostname]` test the DNS server connectivity.

firewalld

Mask iptables.service and ip6tables.service using `systemctl mask`

firewalld replaces `iptables`,`ip6tables` and `ebtables`.

Predefined zones (`man 5 firewalld.zones`)

Zone	Description
home	reject incoming traffic unless related to outgoing traffic or matching `ssh`, `mdns`, `ipp-client`, `samba-client` or `dhcpv6-client`
internal	same as the home zone
work	reject incoming traffic unless related to outgoing traffic or matching `ssh`, `ipp-client` or `dhcpv6-client`
public	used by default, reject incoming trauffic unless related to outgoing traffic or matching `ssh` or `dhcpv6-client`
external	reject incoming traffic unless related to traffic or matching `ssh`, outgoing IPv4 traffic forwarded through this zone is masqueraded
dmz	reject inconming traffic unless related to outgoing traffic or matching `ssh`
block	reject all incoming traffic unless related to outgoing traffic
drop	drop all incoming traffic unless related to outgoing traffic (without sending a response)

Pre-defined services

Service	Description	Ports
ssh	local ssh server	`22/TCP`
dhcpv6-client	local DHCPv6 client	`546/UDP` or `fe80::/64` on IPv6
ipp-client	local IPP printing	`631/UDP`
samba-client	local Windows file and print sharing client	`137/UDP 138/UDP`
mdns	multicast DNS (mDNS) local-link name resolution	`5353/UDP` to the `224.0.0.251` IPv4 or `ff02::fb` IPv6

`firewall-cmd` command

You can use the graphical tool `firewall-config` or `firewall-cmd` for command-line.

Changes can be made only runtime or permanent (adding the `--permanent` option).
You can also specify the zone using `--zone` (it's required for some commands).
CIDR = IP

Option	Description
`--get-default-zone`	query the current default zone
`--set-default-zone=<ZONE>`	change the default zone (runtime and permanent)
`--get-zones`	list all zones
`--get-active-zones`	list all zones currently in use
`--list-all`	list all configured interfaces, sources, services and ports for `--zone=<ZONE>` (otherwise default)
`--list-all-zones`	retrieve information for all zones
`--reload`	drop the runtime configuration and apply the persistent configuration

Zone commands (any of these command uses `--zone=<ZONE>`)

Option	Description
`--add-source=<CIDR>`	route all traffic coming from the `<CIDR>`
`--remove-source=<CIDR>`	remove the rule routing all trafic from the `CIDR` specified
`--add-interface=<INTERFACE>`	route all traffic from `<INTERFACE>` to the specified zone
`--change-interface=<INTERFACE>`	associate the interface with `<ZONE>`
`--add-service=<SERVICE>`	allow traffic to `<SERVICE>`
`--remove-service=<SERVICE>`	remove `<SERVICE>` from the allowed list for the zone
`--add-port=<PORT/PROTOCOL>`	allow traffic to the `<PORT/PROTOCOL>` for the zone

Option	Description
`--remove-port=<PORT/PROTOCOL>`	remove the `<PORT/PROTOCOL>` from the allowed list

`firewall-cmd --set-default dmz` change the default zone to `dmz`.
`firewall-cmd --permanent --zone=internal --add-source=192.186.0.0/24` assign traffic from `192.168.0.0/24` to the `internal` zone.
`firewall-cmd --permanent -add-service=mysql` open the network ports for `mysql` on the `internal` zone.

`ssh` command

Configuration file: `/etc/ssh/sshd_config`

`ssh [remote username]@[remote host]` connect through SSH to another machine.

`ssh [remote username]@[remote host] [command]` connects and automatically executes the specified command.

Wanna connect without passwords? You need a SSH key.

`ssh-keygen` generate a set of public and private keys.
The private key is stored at the file `~/.ssh/id_rsa` and the public key at the file `~/.ssh/id_rsa.pub`.
You can also set a passphrase that you'll have to enter when connecting.

`ssh-agent` it will enter the passphrase for you during the time you're connected.

`ssh-copy-id [remote user]@[remote host]` copy the public key to the remote machine. Once it's done, we can use the password-less system to connect.

Disable root access

1. Edit the file `/etc/ssh/sshd_config`
2. Search and uncomment the line `PermitRootLogin`
3. Change the `yes` for `no` (you can also set it to `without-password` for users that already copied their public key).

Disable Password Authentication

1. Edit the file `/etc/ssh/sshd_config`
2. Search the line `PasswordAuthentication`
3. Replace `yes` for `no`.

Copying files between systems

`scp` command

Send files through SSH.

You can use the `-r` flag with `scp` to copy files recursively.

`scp [files to send] [remote user]@[remote host]:/path/to/put/files`
`scp /etc/hosts root@rmachine1:/root/copied` sends the local file `hosts` to the directory `/root/copied` on the remote machine.
`scp [remote user]@[remote host]:/file/to/copy /path/to/put/files` send a remote file to our machine.

`sftp` command

SSH FTP interactive interface.

`sftp [remote user]@[remote host]` start an **`sftp`** session on the remote server.
You can use commands such as **`ls`**, **`cd`**, **`mkdir`**, **`rmdir`**, **`pwd`** to navigate.
`put` and **`get`** can be used to upload and download files.

`rsync` command

Quite useful when you need to **synchronize** files.

Important use the **`-n`** option to simulate the **`rsync`** changes without applying them.

`rsync` copy files the first time, then it will only modify those that were affected/copy new files.

Option	Description		
v	verbosity output	a	archive mode
r	sync recursively the whole directory	l	sync symbolic links
p	preserve permissions	t	preserve timestamps
g	preserver group ownership	o	preserve files owner's
D	sync device files (only for troubleshoot)	H	preserve hard links
A	sync ACLs	X	sync SELinux context

`rsync [option] [files to synchronize] [/path/to/place/them]`
`rsync -av /etc/ /etcbackup` synchronize all the files from **`/etc`** with the ones on **`/etcbackup`**.

`rsync -av /home/student/foo.bar student@desktop1:/home/student/` synchronize the local files at the remote machine.

LDAP users

`Lightweight Directory Access Protocol`, used in Active Directory and IPA Server.

Install these packages: **`authconfig-gtk`**, **`sssd`** and **`krb5-workstation`**.
There's also a terminal version of **`authconfig-gtk`** but it's deprecated.

In order to connect to a central LDAP Server, **`authconfig`** needs:

- The host name of the LDAP server(s).
- The base DN (Distinguished Name) of the part of the LDAP tree where the system should look for users (**`dc=example dc=com`**).
- If SSL/TLS is used to encrypt communications with the LDAP server, a root CA certificate that can validate the certificates is offered offered by the LDAP server.

Necessary Kerberos parameters:

- The name of the Kerberos realm to use.
- One or more key distribution centers (KDC). This is the host name of your Kerberos server(s).
- The host name of one of more admin servers.

`getent passwd <username>` test the LDAP + Kerberos configuration.

Partitions & File Systems

Useful commands

Command	Description
`df -h`	display filesystems with space on human readable format
`du -h`	display disk usage on human readable format
`blkid`	show all file systems with their UUIDs
`lsof`	show the processes using the specified directory/file
`free -m`	display memory usage in MiB

`mount` command

`mount [device file or UUID] [mount point]`
`mount -a` mount all the file systems specified on `/etc/fstab`.
`mount -o remount,rw /foo` remounts `/foo` with read-write permissions.

`umount` command

`umount [mount point]`
`umount /filesystem-mounted` unmount the filesystem mounted on `/filesystem-mounted`.
If the mount point is being accessed by a process, you can't unmount it (check with `lsof`).

Partitions

MBR (`Master Boot Record`)

- 4 partitions (maximum, 15 by using extended and logical partitions).
- Partition size of 2 TiB.
- Located at the first part of the scheme (boot block).
- `fdisk`

GPT (`GUID Partition Table`)

- Support for 128 partitions.
- Partition size of 8 ZiB.
- First block is the protective MBR, then the partitions table (backup at the end of the disk).
- `gdisk`

`fdisk` & MBR partitions

`fdisk [device]`
`fdisk /dev/sdb` create MBR partitions on `/dev/sdb`.

Key	Description
d	delete partition
m	help
n	create partition

Key	Description
p	display partitions available in the disk
t	change partition's type (L to see table of types)
w	write changes

Run `partprobe [device]` after writing the changes.

`gdisk` & GPT partitions

`gdisk [device]`
`gdisk /dev/sdb` create GPT partitions on `/dev/sdb`

The keys are like the ones used for `gdisk` except for others that are new.
Use `?` or `m` to see the help list of commands.

Remember to run `partprobe [device]` after you write the changes on the disk.

Creating file systems

After a block device has been created, we need to format it.

`mkfs -t [type] [device]`
`mkfs -t ext4 /dev/sdb1` apply the `ext4` file system to `/dev/sdb1`.
`mkfs -t xfs /dev/sdc3` apply the `xfs` file system to `/dev/sdc3`.

Swap partitions

Swap partitions are like extra RAM.

Create a new partition with `fdisk` or `gdisk`, assigning the type `Linux Swap`.

`mkswap [device]`

`swapon [device]`
`swapon -p [priority] [device]` the priority means which swap partition will be used first (higher value means more priority of use).

`swapon -a` activate all the partitions marked as swap space.
`swapon -s` summary of swap partitions.

/etc/fstab

An incorrect /etc/fstab entry may render the machine unbootable.
Use `mount -a`to check if all the entries are correct.

Entries on `/etc/stab` will be automatically mounted when the system boots.

`UUID=[UUID] [mount point] [file system type] [options during mount] [dump flag and fsco order]`

```
UUID=some-UUID      /mnt/storage    xfs     defaults     0 0
/dev/sda            /               xfs     defaults     0 0
```

You can use the device name instead of UUID. The problem is that device numbers are assigned when disks are discovered during the boot.
If you change a disk, it may take the same device name.

LVM (Logical Volume Management)

Physical Volume (PV)

It's the hardware itself, lowest level of LVM.

Your partitions must have the `Linux LVM` type to be used as PV.

Command	Description
`pvcreate /dev/sda3 /dev/sdb2`	mark `/dev/sda3` and `/dev/sdb2` as PVs
`pvmove /dev/sda4`	move PEs from `/dev/sda4`
`pvremove /dev/sda4`	remove the PV label to `/dev/sda4`
`pvs`	display PVs
`pvdisplay`	display more information about PVs (specify a PV to get more details)

Volume Group (VG)

Made with PVs. It can hold Logical volumes.

Command	Description
`vgcreate [name] [physical volumes]`	create a new volume group
	`-s [n]` define PE size, `-s 16M` define each PE to be 16 MiB
`vgremove [VG name]`	delete the VG, leaving the PV available for other volume group
`vgextend [VG name] [PV]`	extend the size of the VG
`vgreduce [VG name] [PV]`	reduce the size of the VG
`vgs`	display VGs
`vgdisplay`	display more information about VGs (specify a VG to get more details)

Logical Volume (LV)

Logical volumes are created inside of VG.

Command	Description
`lvcreate -n [LV-name] -L [size] [VG-name]`	create a new logical volume
	use `-l` to assign a size in extents
`lvremove /dev/[VG]/[LV]`	remove the LV

Command	Description
`lvextend -L [size] /dev/[VG]/[LV]`	extend the size of the LV. `+300M` add 300 MiB to the LV
	`-l` for increase the size in extents
`lvreduce -L [size] /dev/[VG]/[LV]`	reduce the LV, `[size]` is the new size for the LV (you can use `-l` for PE)
`lvs`	display LVs
`lvdisplay`	display more information about LVs (specify a LV to get more details)

Once a LV has been created, you can format it with `mkfs`. The path will be `/dev/[VG]/[LV]`.

Before reducing or after extending a LV, use the command `resize2fs /dev/[VG]/[LV] [new size]`
The new size is only required for reducing.

NFS & SMB

NFS

We must enable and start the unit `nfs-secure`.

Install `autofs` for automount the shares.
NFS can be protected using Kerberos. It will requiere a `/etc/krb5.keytab` and additional authentication configuration (Kerberos realm).

Security methods	Description
none	anonymous access to the files, writes to the server (if allowed) will be allocated UID and GID of nfsnobody.
sys	standard Linux permissions for UID and GID values. Default if another isn't specified
krb5	client must prove identity using Kerberos and then standard Linux permissions
krb5i	cryptographically strong guarantee that the data in each request hasn't been tampered
krb5p	encryption to all requests between the client and the server. Performance impact

Mount an NFS share

`mount -t nfs -o sync [server]:/share /mountpoint` in this case, the mountpoint should be already created.
We can add the option `sec=` to choose which security method we're using.

`/etc/fstab` entry to automount NFS shares on boot.

```
[server]:/share     /mountpoint     nfs sync    0 0
```

`autofs`

Install `autofs` and activate the unit.

Creating and automount

Create a new file at `/etc/auto.master.d` like `home.autofs`

```
/shares /etc/auto.demo
```

The base point is `/shares` and the information to create it's content can be found at `/etc/auto.demo`.
Note: Those files at `/etc/` follow a convention of using `auto` and then something else at their names.

`/etc/auto.demo`

```
* -rw,sync [server]:/shares/&
```

In this case, the ampersand (&) will match the asterisk at the beginning.
The mount point is an asterisk and the subdirectory on the source location is an ampresand.

`/etc/fstab` entry to automount a NFS share that uses Kerberos

```
[server]:/share /mountpoint nfs sec=krb5p,rw 0 0
```

Mount an SMB share

`mount -t cifs -o guest //[server]/share /mountpoint`
The `-t cifs` option is the file system type for SMB shares and the `-o guest` tells `mount` to try and authenticate as a guest account without a password.

Secure SMB share

We can also specify certain security parameters (like username, password)

`/credentials` file

```
username=username
password=password
domain=domain
```

It should be stored somewhere secure with only root access (0600).

`/etc/fstab` entry for secured SMB share

```
//[server]/share /mountpoint cifs creds=/[credentials] 0 0
```