BORN2BEROOT CHEATSHEET🛠️
@egalindo

Prompts

Commands

Technical explanations

Possible errors

Technical Curiosities

🖥️ VIRTUAL MACHINE CONCEPTS

VM BASIC FUNCTIONING

HYPERVISOR⚙️

The software that manages and translates physical hardware resources to the VM (like VirtualBox)

ISOLATION🛡️

The VM runs in a secure partition ensuring actions inside the guest OS don't harm the host OS

FILES💾

The entire VM, including its virtual disk and configuration, is simply stored as files on your host computer's drive

INSTALLATION DOUBTS

WHY DEBIAN?

	Debian	Rocky
Pros	SSH installation simpler and based on Linux standards. Huge repos.	Server stability-focused, minimal bloatware. Aligned with company standards.
Cons	Older requiring more manual configuration.	Need to dive deep on more terminology. Use of SELinux (more commands).

SELinux = Security-Enhanced Linux) Kernel mechanism for Mandatory Access Control for system processes

ISO file: it's a type of disk image file used to replicate the content and structure of a physical disk, like a CD, DVD, or Blu-ray disc, in a single file.

Difference between primary and logical partitions:

A primary partition can be booted directly by the operating system, while a logical partition is created inside an extended partition and cannot boot the operating system

check partitions --> lsblk

apt	--> Easier for daily usage, modern interface
aptitude	--> TUI based, can suggest solutions to complex dependency problems. For advanced tasks.
apparmor	--> Default on Debian Mandatory Access Control (MAC) on Debian, system for the Linux kernel that proactively restricts what individual programs can do.

BASIC SYSTEM COMMANDS

Install sudo	-->	apt install sudo
Reboot system	-->	sudo reboot
Check sudo version	-->	sudo -V
Update system	-->	sudo apt update
Install vim (optional)	-->	sudo apt install vim
Change Hostname	-->	sudo hostnamectl set-hostname new-hostname
Verify Hostname	-->	hostname

if you get an error due to system time mismatch between your VM and main computer, use `sudo date -s "YYYY-MM-DD HH:MM:SS"` with real time to synchronize it

👤USERS👤

Create new user	-->	sudo adduser <username>
Check if user exists	-->	getent passwd <username>
Change password	-->	sudo passwd <username>
Change own password	-->	passwd

👥👥GROUPS👥👥

Create Group	-->	sudo addgroup <groupname>
Check if group exists	-->	getent group <groupname>
Add user to group	-->	sudo adduser <username> <groupname>
Check groups of a user	-->	groups <username>
Add user to sudo group	-->	sudo adduser <username> sudo

📶 SSH

What is SSH?

SSH stands for Secure Shell, and it is a network protocol that allows you to securely connect to another computer over a network. It allows you to access the terminal of another machine as if you were using its keyboard and local screen.
It encrypts all communication, including passwords and data.

Install tool for SSH control	-->	sudo apt install openssh-server (yes on confirmation)
Check if it is correctly installed	-->	sudo service ssh status
Check SSH config	-->	vim /etc/ssh/sshd_config

If any configuration has been changed, then sudo service ssh restart is needed. After that, check again with sudo service ssh status

To create a new SSH connection, we go to Setting --> Network and we create a new NAT assigning Host Port 4241 and Guest Port 4242.
We could use any Host Port not being used, to check if it is being used, for the host port 4242, use command lsof -i :4242. If it does not return an output, port is free, if it is being used it will return the next values (as an example):

```
COMMAND   PID    USER FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd      1038   root 6u  IPv4   8383     0t0  TCP *:4242 (LISTEN)
sshd      1038   root 7u  IPv6   8385     0t0  TCP *:4242 (LISTEN)
```

COMMAND	-->	Process (ssh server)
PID	-->	Process ID
USER	-->	User executing the process
FD	-->	File Descriptor used to listen connection
TYPE	-->	Internet Protocol version
DEVICE	-->	These are internal ID numbers for the sockets in the kernel
SIZE/OFF	-->	Offset or file size for the socket
NODE	-->	Packet transmission protocol
NAME	-->	Port name and status

*IPv4 --> Common internet protocol
*IPv6 --> Increasingly used on modern networks
*Virtual machine will usually open both automatically when enabling a port
*(LISTEN) = Mode: waiting for incoming connections
*If port is free, we can connect from our main computer with
ssh <user>@localhost -p <port>

A socket is the object (maintained by the kernel) that your program creates to communicate.
For the socket to communicate over the network, it needs to be associated with an IP and a port:
(e.g., IP: 192.0.2.10 and port 443)
When you combine the IP + port, you get what is called a network endpoint, for example:
192.0.2.10:443

A PORT is a 16-bit number that uniquely identifies a network connection within a host.

🔥 UFW (FIREWALL) 🧱

What is UFW?

UFW is collection of rules in the Linux Kernel used to filter incoming or outgoing packets. Each packet contains data such as: source/destination IP && Port, protocol (TCP, UDP, ICMP), etc. Firewall decides if these transmissions are allowed or not with the rules set.
If we do not define protocol (e.g. sudo ufc allow <port>), both TCP and UDP will be included.

Install firewall	-->	apt install ufw		
Enable firewall (auto enable on start)	-->	sudo ufw enable		
Allow connections on <port><protocol> (protocol is optional)	-->	sudo ufw allow <port>	-->	This is the Guest Port on VM NAT config.
Connect to <port>	-->	ssh <user>@localhost -p <port>	-->	From main computer, we will connect to host port, VM will redirect us to guest port is NAT configuration is set
Check active rules	-->	sudo ufw status numbered		
Remove rule	-->	sudo ufw delete <number>		

📜 SUDO POLICIES 👨🏻‍⚖️

- passwd_tries=3: Total tries for entering the sudo password.		
- badpass_message="message": The message that will show when the password failed.		
- logfile="/var/log/sudo/sudo_config": Path where will the sudo logs will be stored.	-->	Register which commands where executed and general data (user, date, success or fail)
- log_input, log_output.	-->	Register all inputs and outputs. More detailed than logfile (similar to a screen record). More detailed for audits or similar
- iolog_dir="/var/log/sudo": Diretory	-->	Stores log_inputs and outputs
- requiretty: TTY become required	-->	TTY = "Teletype terminal". sudo can only be executed from a terminal (no scripts or similar)
secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin": Folders that will be excluded of sudo	-->	Only allows sudo commands in directories (separated by ":"), avoiding unwanted binaries

🔑Password Policy🔒

All password expiration and modification rules should be find and modified on vi /etc/login.defs

For password complexity rules installation--> sudo apt install libpam-pwquality

For password complexity rules modification--> vi /etc/pam.d/common-password

PAM = Pluggable Authentication Modules

```
retry=3 minlen=10 ucredit=-1 dcredit=-1 lcredit=-1 maxrepeat=3 reject_username enforce_for_root
```

- retry=3 ➤ Number of retries
- minlen=10 ➤ The minimum characters a password must contain.
- ucredit=-1 ➤ The password must contain at least one capital letter. We must write it with "-" sign, as this is how it knows that it refers to minimum characters; if we put a + sign it will refer to maximum characters.
- dcredit=-1 ➤ The password must contain at least one digit.
- lcredit=-1 ➤ The password must contain at least one lowercase letter.
- maxrepeat=3 ➤ The password cannot have the same character repeated three consecutive times.
- reject_username ➤ The password cannot contain the username within itself.
- difok=7 ➤ The password must contain at least seven different characters from the last password used. → ONLY applied for non root on pam_unix.so **difok=7 non_root**
- enforce_for_root ➤ We will implement this password policy for root.

Check password conditions for a user --> chage -l <username>

The password rules changes are not retroactive; to change them, we can proceed with:

```
sudo chage -M $MAX_DAYS -m $MIN_DAYS -W $WARN_DAYS <username>
```

🤖Script

To exectute script--> `sh <script_name>`

Architecture

```
uname -a
```

uname shows system information.
-a shows everything: kernel name, version, architecture, hostname, etc.

Physical Cores

```
grep "physical id" /proc/cpuinfo | wc -l
```

/proc/cpuinfo contains detailed CPU information.
grep "physical id" filters lines containing “physical id”, identifying each physical CPU.
wc -l counts the lines, giving the number of physical cores.

Virtual Cores

```
grep processor /proc/cpuinfo | wc -l
```

Similar to the previous command, but looks for “processor”.
Each line with “processor” represents a logical core

awk processes text recognizing separators and gets the $element indicated

RAM

```
free --mega | awk '$1 == "Mem:" {print $3}'
free --mega | awk '$1 == "Mem:" {print $2}'
free --mega | awk '$1 == "Mem:" {printf("(%.2f%%)\n", $3/$2*100)}'
```

free --mega shows memory in MB.
awk selects columns: column 3 = used, column 2 = total.
Percentage formula: (used/total)*100. %.2f%% formats with 2 decimals and adds %

DISK MEMORY

```
df -m | grep "/dev/" | grep -v "/boot" | awk '{use += $3} {total
+= $2} END {printf("(%d%%)\n"), use/total*100}'
```

df -m shows disk usage in MB.
grep "/dev/" filters real storage devices.
grep -v "/boot" excludes the /boot partition.
awk '{use += $3; total += $2} END {...}' sums used and total space, then calculates the percentage used.

CPU USAGE PERCENTAGE

```
vmstat 1 4 | tail -1 | awk '{print $15}'
```

vmstat 1 4 shows system statistics 4 times, at 1-second intervals.
tail -1 takes the last line.
$15 in vmstat is the percentage of idle CPU. To get active CPU usage, usually calculate 100 - $15

LAST REBOOT

```
who -b | awk '$1 == "system" {print $3 " " $4}'
```

who -b shows the date of the last boot.
awk prints the date and time of the last reboot.

ACTIVE LVM

```
if [ $(lsblk | grep "lvm" | wc -l) -gt 0 ]; then echo yes; else
echo no; fi
```

lsblk lists block storage devices.
grep "lvm" searches for LVM volumes.
wc -l counts the results.
If greater than 0 → yes (LVM active), else → no.

TCP CONNECTIONS

```
ss -ta | grep ESTAB | wc -l
```

ss -ta shows active TCP connections.
grep ESTAB filters only established connections.
wc -l counts how many are active.

NUMBER OF USERS

```
users | wc -w
```

users lists currently logged-in users.
wc -w counts the number of users (words).

IP AND MAC ADRESSES

```
ip link | grep "link/ether" | awk '{print $2}'
```

ip link shows all network interfaces.
grep "link/ether" filters only the MAC addresses.
awk '{print $2}' prints the MAC address.

NUMBER OF COMMANDS EXECUTE WITH SUDO

```
journalctl _COMM=sudo | grep COMMAND | wc -l
```

journalctl _COMM=sudo filters logs for sudo commands.
grep COMMAND searches for executed commands.
wc -l counts how many commands were run with sudo.

⏰Crontab

Cron is a software utility found on Unix-like operating systems (including Linux distributions like Debian) that provides a time-based job scheduler

Configure Crontab 1st time	-->	sudo crontab -u root -e
Reopen Crontab	-->	crontab -e

We add `*/MM * * * * sh /path_to_file.sh` at the end of the file

Document saves on /var/spool/cron/crontabs/ , but we never have to open it with a text editor, always use `crontab -e`

Syntax for replacing a value is `*` --> `*/new_value`

```
*    *    *    *    *  command
-    -    -    -    -
|    |    |    |    |
|    |    |    |    +----- Day of week (0-7, Sun=0 or 7)
|    |    |    +------- Month (1-12)
|    |    +--------- Day of month (1-31)
|    +----------- Hour (0-23)
+------------- Minute (0-59)
```

✍️ Signature ✍️

To get signature, on VM installation folder --> `shasum <machinename>.vdi`

What is Shasum?

shasum is a utility that calculates a SHA checksum for a file.
A checksum is a cryptographic hash: a fixed-length string that uniquely represents the file contents.

It outputs a string of 40 hexadecimal characters (SHA-1 hash) followed by the file name.
Example output:

```
d2c7e3a1b9f8e9f2d0c1e4b5a6f7d8e9f0a1b2c3  machinename.vdi
```

The long string is the hash. The filename is shown for reference.
This ensures the file hasn't changed since the hash was generated.

Why on .vdi?

.vdi is virtual disk image
.vbox is VM configuration
.vbox-prev is a config backup

💡 Lighttpd
📰 WordPress

Install Lighttpd --> `sudo apt install lighttpd`

Lighttpd: is a web server designed to be fast, secure, flexible, and standards-compliant. It is optimized for environments where speed is a top priority because it consumes less CPU and RAM than other servers.

Port 80 is the standard for http communication

Install Wordpress --> `sudo apt install wget zip` --> `cd /var/www/` --> `sudo wget https://es.wordpress.org/latest-es_ES.zip` --> `sudo unzip ...` --> rename extracted to html

WordPress: It is a content management system focused on the creation of any type of website.

🦭 Mariadb

Install MariaDB --> sudo apt install mariadb-server

Find mariadb-secure-installation folder --> find / | grep mariadb | grep secure

From that folder, run script to restrict access to the server and remove unused accounts --> sudo mariadb-secure-installation

MariaDB is an open-source relational database management system (RDBMS). It's used to store and manage data, similar to MySQL, PostgreSQL, or SQL Server.

Open databases	-->	mariadb
Create Database	-->	CREATE DATABASE wp_database;
Show Databases	-->	SHOW DATABASES;
Create user	-->	CREATE USER '<user>'@'localhost' IDENTIFIED BY '12345';
		This is the user's password
Give permissions	-->	GRANT ALL PRIVILEGES ON wp_database.* TO 'egalindo'@'localhost';
Update permissions	-->	FLUSH PRIVILEGES;
Exit from Mariadb	-->	exit

🐘 PHP

It is a programming language. It is mainly used to develop dynamic web applications and interactive websites. PHP runs on the server side.

Install PHP --> sudo apt install php-cgi php-mysql

🌐Connecting to Wordpress📰

Configure WP Values (DB NAME, USER AND PWD) --> vi /var/www/html/wp-config.php

After installing WP-CLI, we will be able to change our home adress to http://localhost:8181

📊Install a service (NetData)📊

After sudo apt update, bash <(curl -SsL https://my-netdata.io/kickstart.sh)
Install on custom metrics any service you want (CPU usage in my case)

After sudo apt install stress-ng -y

we can use the following command to stress the CPU:
stress-ng --cpu 4 --timeout 30s

NetData will always use port 19999 by default