PARTITIONS

- **`lsblk`** –> Partitions info

HOSTNAME

- **`hostnamectl`** –> To check the host name
- **`hostnamectl set-hostname <new name>`** –> to change the name of the host
- **`sudo nano /etc/hosts (reboot)`** –> to change the name of the host

USERS MANAGEMENT

- **`groups <username>`** –> Know what groups the current user belongs to
- **`sudo useradd <username>`** –> Create a user WITHOUT a home directory
- **`sudo useradd -m <username>`** –> Create a user WITH a home directory
- **`sudo passwd <username>`** –> The user will be prompted to set a new password.
- **`sudo useradd -m <username> -p PASSWORD`** –> Creates the user and set the password “PASSWORD”
- **`sudo groupadd <group_name>`** –> Creates a new group
- **`sudo usermod -a -G <group_name> <username>`** –> Adds the username to the group named- (-a appends / -G what follows is a group)
- **`getent group sudo`** –> Check if the user is in the Sudo group
- **`su -`** –> to change to root user
- **`sudo userdel <username>`** –> to delete a user
- **`sudo chage -l <username>`** –> Check the password configuration of the user
- **`cut -d. -f1 /etc/passwd`** –> Show the local user

PASSWORDS POLICIES

- **`sudo vim /etc/login.defs`** –> To configure the passwords rules
- **`sudo apt install libpam-pwquality`** –> To instal the package libpam-pwquality to set the password policies (PAM module)
- **`dpkg -l | grep libpam-pwquality`** –> To check the package installation.
- **`sudo vim /etc/pam.d/common-password`** –> To set the password strength.

```
minlen=10 –> to set a minimum of 10 characters
credit= -1 –> to set a minimum of 1 upper character
credit=-1 –> to set a minimum of 1 digit
maxrepeat=3 –> to set a rule to avoid more than 3 similar characters together
```

```
reject_username –> to reject the password if it contains the username
difok=7 –> 7 characters should be different than previous password
enforce_for_root –> to set the same policies for the root user.
```

SSH

- **`sudo apt install openssh-server`** –> Install the ssh server
- **`dpkg -l | grep ssh`** –> Check if the SSH has ben installed correctly.
- **`sudo nano /etc/ssh/sshd_config`** –> To change the SSH config
- **`sudo service ssh status`** –> Check the ssh status
- **`systemctl status ssh`** –> Check the ssh status
- **`ip a s`** –> to know the ip address of the virtual machine

CONECTIONS

- **`ssh username@ip-address -p 4242`** –> To connect to the ip-address machine through the 4242 port using the username login.
- **`logout`** –> to disconnect
- **`exit`** –> to close the ssh session

UFW (FIREWALL)

- **`sudo apt install ufw`** –> Instals the UFW firewall
- **`Dpkg -l | grep ufw`** –> Check the UFW installation
- **`sudo ufw enable`** –> Activates the firewall service. Disabled by default.
- **`sudo ufw allow 4242`** –> Tell the firewall the port 4242 is valid for connections.
- **`sudo ufw deny 4242`** –> Disable the 4242 port
- **`sudo ufw delete <port nb>`** –> Removes the port from the allowed list.
- **`sudo ufw status`** –> The the ufw status.

MONITORING.SH

The script is created under the root directory to avoid problems with its execution due to in some instructions, it is needed to execute as root user.

```
#!/bin/bash
#Arquitecture
echo "#Architecture:" $(uname -a)
```

```
#CPU Physical
echo "#CPU physical:" $(lscpu | awk ' NR==5 {print $2}')
#Virtual CPUrm
echo "#vCPU :" $(lscpu | grep Socket\(s\) | awk '{print $2}')
#Memory Usage
free --mega | awk 'NR==2{printf "#Memory Usage: %s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2
}'
#Disk Usage
df -h | awk '$NF=="/"{printf "#Disk Usage: %d/%dGB (%s)\n", $3,$2,$5}'
#CPU Load
top -bn1 | grep load | awk '{printf "#CPU Load: %.2f%s\n", $(NF-2), "%"}'
#Last Boot
echo "#Last boot:" $(who -b | awk '{print($4 " " $5)}')
#LVM
echo "#LVM use:" $(if [ $(lsblk | grep "lvm" | wc -l) -eq 0 ]; then echo no; else echo
yes; fi)
#Connections TCP
echo "#Connections TCP:" $(ss -s | grep TCP | awk 'NR==2 {printf "%d ESTABLISHED\n",
$3}')
#User log
echo "#User log:" $(who | wc -l)
#Network IP
echo "#Network: IP" $(hostname -I) $(ip a | grep link/ether | awk '{printf " (%s)\n",
$2}')
#Sudo
echo "#Sudo : " $(cat /var/log/sudo/sudo.log | grep USER | wc | awk '{printf "%s
cmd\n", $1}')
```

CRON

- **`sudo crontab -u root -e`** —> Open the crontab configuration file
- **`sudo service cron stop`** —> Stop the cron service
- **`sudo service cron start`** —> Start the cron service
- **`sudo service cron status`** —> show the cron status

BONUS

LIFHTTPD

- **```sudo apt install lighttpd —> Install the web server lighttpd
- **```sudo uff allow 80 —> open the 80 port in the firewall

MARIADB (To review)

- **`sudo apt install mariadb-server`** —> Install MariaDB Database server
- **`sudo mariadb`** —> to enter into MariaDB environment
- ***In mariadb environment*** —> **`CREATE USER <username> @localhost IDENTIFIED BY <password>;`**

- ***In mariadb environment*** –> **`GRANT ALL PRIVILEGES ON <database_name>.* TO * <username>@localhost IDENTIFIED BY <password>;`** –> To give all DB privileges to a user.
- ***In mariadb environment*** –> **`FLUSH PRIVILEGES;`** –> To reload the privileges without restarting.
- ***In mariadb environment*** –> **`exit`** –> To exit from mariadb environment.
- ***In mariadb environment*** –> **`SHOW DATABASES;`** –> To show all the DB where the username has access to.

PHP

- **`sudo apt install php-cgi php-mysql`** –> To install PHP

WORDPRESS

- **`sudo apt install wget`** –> To install wordpress
- **`sudo wget http://wordpress.org/latest.tar.gz -P /var/www/html`** –> To download Wordpress in /var/www/html
- **`sudo tar -xzvf /var/www/html/latest.tar.gz`** –> Extract the content
- **`sudo rm /var/www/html/latest.tar.gz`** –> To remove the tarball
- **`sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php`** –> Create a config file for wordpress
- **`sudo nano /var/www/html/wp-config.php`** –> To modify the config file connecting wordpress to mariadb

FTP

- **`sudo apt install vsftpd`** –> To install an FTP server
- **`sudo ufw allow 21`** –> Open 21 port in the firewall for FTP
- **`sudo vim /etc/vsftpd.conf`** –> To configure the FTP server (uncomment the line #write_enable=YES)
- **`sudo mkdir /home/<username>/ftp`** –> Create a folder for ftp transferred files for the user
- **`sudo mkdir /home/<username>/ftp/files`** –> Create a folder for ftp transferred files for the user
- **`sudo chown nobody:nogroup /home/”<username>/ftp`** –> Change the owner of a file
- **`sudo chmod a-w /home/<username>/ftp`** –> give permissions

TO PREPARE THE DEFENSE

https://www.codequoi.com/en/born2beroot-02-configuring-a-debian-virtual-server/#monitoringsh_for_born2beroot