Born 2 be root

Quelques liens de tutos si vous êtes en galères :
https://techdebt.tistory.com/18
https://jjjaeu.tistory.com/33
https://infinitt.tistory.com/39
https://tbonelee.tistory.com/16
https://github.com/pasqualerossi/Born2BeRoot-Guide?tab=readme-ov-file
https://www.youtube.com/results?search_query=born2beroot+coreen

shasum

----------> Partie obligatoire
<----------

Configuration XML pour la création de la VM

Pendant l'installation de la VM :
- Partition disk : Guided - use entire disk and set up encrypted LVM
- Séparation des partitions /home, /var, et /tmp

Mise à jour et installation de sudo
apt-get update -y
apt-get upgrade -y
apt install sudo
groupadd user42
usermod -aG user42, sudo cgorin

Configuration de sudo
sudo visudo
-> Ajouter : ***cgorin   ALL=(ALL:ALL) ALL***
sudo nano /etc/sudoers

```
Defaults    env_reset
Defaults    mail_badpass
Defaults
secure_path="/usr/local/sbin:/usr/local/bin:/usr/bin:/sbin:/bin"
Defaults    badpass_message="Password is wrong, please try again!"
Defaults    passwd_tries=3
Defaults    logfile="/var/log/sudo/sudo.log"
Defaults    log_input, log_output
Defaults    requiretty
```

Configuration de SSH et UFW

Configuration du service SSH
apt install openssh-server

nano /etc/ssh/sshd_config
	-> Remplacer ***#Port 22*** par ***Port 4242***
systemctl restart ssh
systemctl status ssh

Configuration du pare-feu UFW
apt-get install ufw
ufw enable
ufw default deny
ufw allow 4242/tcp
ufw status verbose

Configuration de sudo selon une pratique stricte

Installation de libpam-pwquality
sudo apt install libpam-pwquality

Configuration de /etc/pam.d/common-password
sudo visudo /etc/pam.d/common-password
	-> Remplacer ***Password requisite pam_pwquality.so retry=3*** par:
	Password requisite pam_pwquality.so retry=3 minlen=10 maxrepeat=3 difok=7 lcredit=-1 ucredit=-1 dcredit=-1 reject_username enforce_for_root

Application de la politique de mot de passe
passwd -e cgorin

Configuration de /etc/login.defs
sudo nano /etc/login.defs
	Ajouter :
		PASS_MAX_DAYS 30
		PASS_MIN_DAYS 2
		PASS_WARN_AGE 7

Création du dossier pour les logs sudo
sudo mkdir /var/log/sudo/

Mise en place d'un script monitoring

Installation de net-tools :
sudo apt-get install net-tools
touch /usr/local/bin/monitoring.sh
chmod 777 /usr/local/bin/monitoring.sh

Modification du script
Dans le terminal du pc hote -> ssh cgorin@127.0.0.1 -p 22222
sudo nano /usr/local/bin/monitoring.sh
	#!/bin/bash

```
arc=$(uname -a)
pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)
vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)
fram=$(free -m | awk '$1 == "Mem:" {print $2}')
uram=$(free -m | awk '$1 == "Mem:" {print $3}')
pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')
fdisk=$(df -BG | grep '^/dev/' | grep -v '/boot$' | awk '{ft +=
$2} END {print ft}')
udisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut +=
$3} END {print ut}')
pdisk=$(df -BM | grep '^/dev/' | grep -v '/boot$' | awk '{ut +=
$3} {ft+= $2} END {printf("%d"), ut/ft*100}')
cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk
'{printf("%.1f%%"), $1 + $3}')
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')
lvmu=$(if [ $(lsblk | grep "lvm" | wc -l) -eq 0 ]; then echo no;
else echo yes; fi)
ctcp=$(ss -neopt state established | wc -l)
ulog=$(users | wc -w)
ip=$(hostname -I)
mac=$(ip link show | grep "ether" | awk '{print $2}')
cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)
wall "      #Architecture: $arc
      #CPU physical: $pcpu
      #vCPU: $vcpu
      #Memory Usage: $uram/${fram}MB ($pram%)
      #Disk Usage: $udisk/${fdisk}Gb ($pdisk%)
      #CPU load: $cpul
      #Last boot: $lb
      #LVM use: $lvmu
      #Connections TCP: $ctcp ESTABLISHED
      #User log: $ulog
      #Network: IP $ip ($mac)
      #Sudo: $cmds cmd"
```

OU

```
#!/bin/bash
arch=$(uname -a);

socket=$(lscpu | grep -E '^Socket\(' | rev | cut -d' ' -f1 |
rev);
vcpu=$(lscpu | grep -E '^CPU\(' | rev | cut -d' ' -f1 | rev);

mem_used=$(free -m | head -n 2 | tail -1 | awk '{print $3}');
mem_total=$(free -m | head -n 2 | tail -1 | awk '{print $2}');
mem_used_percent=$(($mem_used*100/$mem_total));

sda_size=$((($(df /boot | awk 'NR > 1 {print $2}') + $(df / | awk
'NR > 1 {print $2}') + $(df /srv | awk 'NR > 1 {print $2}') +
$(df /home | awk 'NR > 1 {print $2}') + $(df /tmp | awk 'NR > 1
{print $2}') + $(df /var | awk 'NR > 1 {print $2}') + $(df
```

```
/var/log | awk 'NR > 1 {print $2}')) / 1024 + $(free -m | head -n
3 | tail -1 | awk '{print $2}')));
sda_used=$((($(df /boot | awk 'NR > 1 {print $3}') + $(df / | awk
'NR > 1 {print $3}') + $(df /srv | awk 'NR > 1 {print $3}') +
$(df /home | awk 'NR > 1 {print $3}') + $(df /tmp | awk 'NR > 1
{print $3}') + $(df /var | awk 'NR > 1 {print $3}') + $(df
/var/log | awk 'NR > 1 {print $3}')) / 1024 + $(free -m | head -n
3 | tail -1 | awk '{print $3}')));
sda_used_percent=$(($sda_used*100/$sda_size));

cpu_loaded_user=$(top -bn1 | grep '%Cpu(s)' | awk '{print $2}' |
sed 's:\.[^|]*::g');
cpu_loaded_sys=$(top -bn1 | grep '%Cpu(s)' | awk '{print $4}' |
sed 's:\.[^|]*::g');
cpu_loaded=$((cpu_loaded_user+cpu_loaded_sys));

last_boot=$(who -b | cut -d ' ' -f13-15);

raw_lvm=$(lsblk -f | grep "sda2_crypt" | awk '{print $3}');
is_lvm="no";
if [ $raw_lvm = "LVM2" ]
then
      is_lvm="yes";
fi

esta_connections=$(ss -s | grep "estab" | awk '{print $4}' | rev
| cut -c2- | rev);
user_log=$(who | cut -d' ' -f1 | sort | uniq | wc |  awk '{print
$1}');

ipv4=$(sudo ifconfig enp0s3 | grep "inet " | awk '{print $2}');
mac_address=$(sudo ifconfig enp0s3 | grep "ether " | awk '{print
$2}');

raw_cmd=$(cat /var/log/sudo/sudo.log | wc | awk '{print $1}');
sudo_cmd=$(($raw_cmd / 2));

clear;
echo
"------------------------------------------------------------------
---------------
# Architecture: $arch
------------------------------------------------------------------
--------------
# CPU physical: $socket
# vCPU: $vcpu
# CPU load: $cpu_loaded%
# Memory Usage: $mem_used MB / $mem_total MB ($mem_used_percent%)
# Disk Usage: $sda_used MB / $sda_size MB ($sda_used_percent%)
------------------------------------------------------------------
--------------
```

```
      # Last boot: $last_boot
      # LVM use: $is_lvm
      -------------------------------------------------------------
      --------------
      # Connection(s) TCP: $esta_connections
      # User(s) log: $user_log
      # Network: IP $ipv4 ($mac_address)
      -------------------------------------------------------------
      --------------
      # Sudo history: $sudo_cmd
      -------------------------------------------------------------
      --------------"| wall -n;
sudo visudo
      ALL=(ALL) NOPASSWD: /usr/local/bin/monitoring.sh
sudo reboot
sudo /usr/local/bin/monitoring.sh
sudo crontab -u root -e
*/10 * * * * /usr/local/bin/monitoring.sh
```

----------> Partie bonus (WordPress)
<----------

Installation de PHP

```
# Ajout du dépôt Sury's repository pour obtenir la dernière version de
PHP
sudo apt update
sudo apt install curl
sudo curl -sSL https://packages.sury.org/php/README.txt | sudo bash -x
sudo apt update

# Installation de PHP version 8.1
sudo apt install php8.1
sudo apt install php-common php-cgi php-cli php-mysql
```

Installation de Lighttpd

```
service lighttpd force-reload

# Vérification si Apache est installé et désinstallation pour éviter
les conflits avec Lighttpd
systemctl status apache2
sudo apt purge apache2
```

Installation de Lighttpd
sudo apt install lighttpd

Vérification de la version, démarrage, activation et statut de Lighttpd
sudo lighttpd -v
sudo systemctl start lighttpd
sudo systemctl enable lighttpd
sudo systemctl status lighttpd

Autorisation du port HTTP (port 80) à travers UFW
sudo ufw allow http
sudo ufw status

Activation des modules FastCGI et FastCGI-PHP de Lighttpd
sudo lighty-enable-mod fastcgi
sudo lighty-enable-mod fastcgi-php
sudo service lighttpd force-reload

Pour tester si PHP fonctionne avec Lighttpd :
vi /var/www/html/info.php
<?php
echo "Hello World";
?>
Ouvrir avec un navigateur web l'adresse ***http://localhost:8080/index.php***, le message Hello World! devrait s'afficher.

Installation de MariaDB

sudo apt install mariadb-server
Démarrage, activation et vérification du statut de MariaDB
sudo systemctl start mariadb
sudo systemctl enable mariadb
systemctl status mariadb

Configuration sécurisée de MySQL
sudo mysql_secure_installation

Redémarrage du service MariaDB
sudo systemctl restart mariadb

Connexion à MariaDB
mysql -u root -p

Création d'une base de données pour WordPress
MariaDB [(none)]> ***CREATE DATABASE wordpress_db;***
MariaDB [(none)]> ***CREATE USER 'admin'@'localhost' IDENTIFIED BY 'WPpassw0rd';***

```
MariaDB [(none)]> GRANT ALL ON wordpress_db.* TO 'admin'@'localhost'
IDENTIFIED BY 'WPpassw0rd' WITH GRANT OPTION;
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;

# Vérification que la base de données a été créée avec succès
mysql -u root -p
MariaDB [(none)]> show databases;
```

Installation de WordPress

```
# Installation de wget et tar
sudo apt install wget
sudo apt install tar

# Téléchargement de la dernière version de Wordpress, extraction et
placement des contenus dans le répertoire /var/www/html/
wget http://wordpress.org/latest.tar.gz
tar -xzvf latest.tar.gz
sudo mv wordpress/* /var/www/html/
rm -rf latest.tar.gz wordpress/

# Création du fichier de configuration de WordPress
sudo mv /var/www/html/wp-config-sample.php /var/www/html/wp-config.php

# Édition de /var/www/html/wp-config.php avec les informations de la
base de données
      /** The name of the database for WordPress */
      define( 'DB_NAME', 'wordpress_db' );

      /** Database username */
      define( 'DB_USER', 'admin' );

      /** Database password */
      define( 'DB_PASSWORD', 'WPpassw0rd' );

      /** Database host */
      define( 'DB_HOST', 'localhost' );


# Modification des permissions du répertoire WordPress pour accorder
des droits au serveur web et redémarrage de Lighttpd
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
sudo systemctl restart lighttpd
```

Dans le navigateur de l'hôte, connectez-vous à ***http://127.0.0.1:8080*** et terminez l'installation de WordPress.