

Vous allez créer votre première machine en respectant des consignes précises et en utilisant VirtualBox (ou UTM si VirtualBox ne fonctionne pas sur votre machine).

L'utilisation de VirtualBox (ou UTM si VirtualBox ne fonctionne pas sur votre machine) est obligatoire.

Installer Oracle VirtualBox 7.1.4 pour Windows (Oracle VirtualBox 7.0.14 est déjà installé sur les PC de 42 sous Ubuntu 22.0.4)

<https://www.virtualbox.org/wiki/Downloads>

Vous devez utiliser comme système d'exploitation, au choix : Debian latest stable (pas de testing/unstable), ou Rocky latest stable. L'utilisation de Debian est fortement conseillée pour quelqu'un débutant dans ce domaine.

Télécharger Debian 12.8.0 (amd64) sur <https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/>

[debian-12.8.0-amd64-netinst.iso](#)

Dans VirtualBox, installer la VM

New

Name : born2beroot

Folder (42 PC) : goinfre/juduchar/born2beroot

ISO Image : [debian-12.8.0-amd64-netinst.iso](#)

Cocher Skip Unattended Installation

Suivant

Hardware :

Mémoire vive : 2048 Mb (ou plus)

Processors : 1 CPU

Create a virtual hard disk now

Ne pas cocher Pre-allocate Full Size (pour disque dur virtuel dynamique)

Disk size : 10,00 Gio (pour la partie obligatoire) ou 35.00 Gio (pour la partie bonus)

Suivant

Finish

Start

Une interface graphique n'a pas d'utilité ici. Il est donc interdit d'installer X.org ou tout autre serveur graphique équivalent.

Install (pas Graphical Install)

Language : English

other

Europe

France

Locale : United States

Keyboard : American English

Votre machine aura pour hostname votre login suivi de 42 (exemple : wil42)

Hostname : juduchar42

Domain name : laisser vide

Continue

Root password : *****

Un utilisateur sera présent avec pour nom votre login en plus de l'utilisateur root.

Full name for the new user : juduchar

Username for your account : juduchar

Password for the new user : *****


```
| sda1      8:1    0  487M  0 part  /boot
```

sda1 est une partition physique (partition primaire) (TYPE : part)

sda1 est la première partition de sda (NAME : sda1 : première partition de sda)

8:1 indique aussi que sda1 est la première partition de sda

Sa taille est de 487 MB (SIZE : 487M)

Le point de montage de cette partition est /boot (MOUNTPOINT : /boot)

/boot est monté sur une partition séparée des autres, contenant les fichiers nécessaires au démarrage du système, comme le noyau

sda1 :

SCSI2 (0,0,0) (sda) – 21.5 GB ATA VBOX HARDDISK

Yes

pri/log 21.5 GB FREE SPACE

Create a new partition

500 MB

Continue

Primary

Beginning (les premiers secteurs doivent être réservés à /boot et à /)

Use as: Ext4 journaling file system

Mount point

/boot – static files of the boot loader

Done setting up the partition

#1 primary 499.1 MB f ext4 /boot est indiqué, ce qui signifie la 1ere partition primaire de taille 499.1 MB, bootable (f signifiant que cette partition est marquée pour être utilisée comme partition de démarrage), montée sur /boot

sda5 :

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
└─sda5	8:5	0	7.5G	0	part	

8:5 signifie que sda5 est un disque SCSI/SATA, et c'est la première partition logique de sda (car les partitions logiques commencent par 5)

Elle a une taille de 7.5 G

Elle n'est pas montée

Pour créer cette partition, il faut choisir **pri/log 20.0 GB FREE SPACE** (car c'est une partition du disque sda) : on remarque que la taille disponible pour ce disque a diminuée de 500 MB

Create a new partition

max (pour allouer tout le reste de l'espace disponible) (ainsi, en cryptant sda5, tout l'espace restant sur le disque sera crypté)

Logical (sda5 correspond à une partition logique)

Mount point

Do not mount it

Done setting up the partition

#5 logical 21.0 GB f ext4 est indiqué, ce qui signifie la 1ere partition logique (les partitions logiques commencent par 5) de taille 21.0 GB, bootable (f signifiant que cette partition est marquée pour être utilisée comme partition de démarrage), non montée

Il faut ensuite encrypter le volume sda5

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
└─sda5_crypt	254:0	0	7.5G	0	crypt	

Configure encrypted volumes

Yes

Create encrypted volumes

Choose sda5 (space)

Enter

Tout laisser par défaut

Done setting up the partition

Finish

Yes

Passphrase : born2beRoot-42*

#5 logical 21.0 GB K crypto / est indiqué, ce qui signifie que la 1ere partition logique (les partitions logiques commencent par 5) de taille 21.0 GB, non montée, a été transformée en conteneur chiffré

La notation K est utilisée pour indiquer qu'une couche de chiffrement a été appliquée au volume

Crypto signifie que le volume est désormais chiffré (toutes les données qui y sont stockées sont protégées et inaccessibles sans déchiffrement)

sda5_crypt indique que sda5 est formaté avec un système de chiffrement via dm_crypt

Le conteneur nécessite une passphrase pour être monté et utilisé

Une fois la partition **sda5** déverrouillée (en fournissant la passphrase), **dm-crypt** crée un volume chiffré appelé **sda5_crypt**.

Montage et accès : sda5_crypt est ensuite monté comme un volume standard (ou un **PV** dans LVM) pour permettre l'accès aux données. Le système traite **sda5_crypt** comme un volume non chiffré après déverrouillage, mais toutes les opérations sont chiffrées/déchiffrées en arrière-plan.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
wil--vg-root	254:1	0	2.8G	0	lvm	/
wil--vg-swap_1	254:2	0	976M	0	lvm	[SWAP]
wil--vg-home	254:3	0	3.8G	0	lvm	/home

Nous allons maintenant créer le VG (Volume Group) juduchar

Configure the Logical Volume Manager

Yes

Create volume group

juduchar

dev/mapper/sda5_crypt (on veut créer le groupe dans le conteneur chiffré sda5_crypt)
(sélectionner avec espace, puis entrée)

Le device mapper (gestionnaire de périphériques) est un sous-système Linux qui crée les périphériques mappés pour faciliter le chiffrement et d'autres opérations comme la gestion des volumes logiques avec LVM

dev/mapper/sda5_crypt est un périphérique virtuel pour l'accès au volume chiffré

il fonctionne comme une interface vers le volume déchiffré, ce qui permet au système de lire et d'écrire sur le volume chiffré de manière transparente

Maintenant, nous allons créer le premier LV (Logical Volume) (lvm) root du VG (Volume Group) juduchar

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
└wil--vg-root	254:1	0	2.8G	0	lvm	/

Create Logical Volume

juduchar

root

2.8G

Puis le deuxième LV (Logical Volume) (lvm) swap_1 du VG (Volume Group) juduchar

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
└wil--vg-swap_1	254:2	0	976M	0	lvm	[SWAP]

Create Logical Volume

juduchar

swap_1

1 G

Enfin, le troisième LV (Logical Volume) (lvm) home du VG (Volume Group) juduchar

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
wil--vg-home	254:3	0	3.8G	0	lvm	/home

Create Logical Volume

juduchar

home

3.8 G

Finish

Les 3 LV (home, root et swap_1), du VG LVM juduchar, apparaissent maintenant

Linear indique un agencement linéaire des données (elles sont écrites séquentiellement sur le disque, dans l'ordre de stockage)

C'est le mode de base pour créer des LV, cela permet d'augmenter l'espace total disponible en combinant des disques de manière séquentielle (si l'on alloue de la mémoire à un LV, les données seront continues du premier au second disque)

Il faut maintenant monter ces 3 LV dans les points de montage correspondants :

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
wil--vg-root	254:1	0	2.8G	0	lvm	/
wil--vg-swap_1	254:2	0	976M	0	lvm	[SWAP]
wil--vg-home	254:3	0	3.8G	0	lvm	/home

Sous LV VG juduchar, LV home - ...

Sélectionner #1 3.8 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/home – user home directories

Done setting up the partition

/home est utilise pour monter une partition dediee aux fichiers des utilisateurs, pour que leurs donnees personnelles soient separees du systemee principal

Sous LV VG juduchar, LV root - ...

Sélectionner #1 2.8 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/ - the root file system

Done setting up the partition

/ est le point de montage racine, ou le système de fichiers principal est monte : toutes les autres partitions ou peripheriques montes apparaissent comme des sous-repertoires de /

Sous LV VG juduchar, LV swap_1 - ...

Sélectionner #1 2.8 GB

Use as: swap area

Done setting up the partition

La partition swap est utilisee comme mémoire virtuelle pour compléter la RAM (mémoire vive) lorsqu'elle est saturée

Le swap est utile pour les operations necessitant beaucoup de mémoire ou pour éviter les plantages en cas de surcharge de la RAM

Les donnees sont stockees temporairement dans la partition swap lorsque la RAM est epuisee, pour éviter les erreurs de mémoire insuffisante

Elle peut aussi être utilisée pour l'hibernation : le contenu de la RAM peut être enregistré dans le swap pour pouvoir être restauré au démarrage

Lors de la création des volumes logiques (LV) dans sda5, la partition principale sda2 sera aussi créée puis transformée en partition étendue (de taille 1K, structure pour organiser les partitions logiques)

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda2	8:2	0	1K	0	part	

sr0 correspond au premier lecteur optique (sr0) (lecteur de CD-ROM ou de DVD-ROM) (de type rom et RM 1 (périphérique amovible))

Le CD-ROM ou DVD-ROM est de taille 1024 Mo (1 Go) (à moins que cela soit une taille indicative par défaut)

Ce lecteur optique peut écrire des données sur disque (si celui-ci est inscriptible). En effet, R0 est à 0, ce n'est pas un périphérique en lecture seule

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sr0	11:0	1	1024M	0	rom	

Il n'est donc pas nécessaire de s'occuper de ces deux périphériques

Finish partitioning and write changes to disk

Vérifier les informations

Yes

PARTIE BONUS UNIQUEMENT :

Mettre correctement en place des partitions afin d'obtenir une structure proche de cet exemple :

```
# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0 30.8G  0 disk
├─sda1                              8:1    0   500M  0 part  /boot
├─sda2                              8:2    0     1K  0 part
├─sda5                              8:5    0 30.3G  0 part
│   └─sda5_crypt                    254:0    0 30.3G  0 crypt
│       ├─LVMGroup-root              254:1    0   10G  0 lvm    /
│       ├─LVMGroup-swap              254:2    0   2.3G  0 lvm    [SWAP]
│       ├─LVMGroup-home              254:3    0     5G  0 lvm    /home
│       ├─LVMGroup-var               254:4    0     3G  0 lvm    /var
│       ├─LVMGroup-srv               254:5    0     3G  0 lvm    /srv
│       ├─LVMGroup-tmp               254:6    0     3G  0 lvm    /tmp
│       └─LVMGroup-var--log          254:7    0     4G  0 lvm    /var/log
sr0                                  11:0    1 1024M  0 rom
```

sda1 :

SCSI3 (0,0,0) (sda) – 37.6 GB ATA VBOX HARDDISK

Yes

pri/log 37.6 GB FREE SPACE

Create a new partition

500 MB

Continue

Primary

Beginning (les premiers secteurs doivent être réservés à /boot et à /)

Use as: Ext4 journaling file system

Mount point

/boot – static files of the boot loader

Done setting up the partition

pri/log 37.1 GB FREE SPACE

Create a new partition

max (pour allouer tout le reste de l'espace disponible) (ainsi, en cryptant sda5, tout l'espace restant sur le disque sera crypté)

Logical (sda5 correspond à une partition logique)

Mount point

Do not mount it

Done setting up the partition

Configure encrypted volumes

Yes

Create encrypted volumes

Choose sda5 (space)

Enter

Tout laisser par défaut

Done setting up the partition

Finish

Yes

Passphrase : born2beRoot-42*

Configure the Logical Volume Manager

Yes

Create volume group

LVMGroup

dev/mapper/sda5_crypt (on veut créer le groupe dans le conteneur chiffré sda5_crypt)
(sélectionner avec espace, puis entrée)

Create Logical Volume

LVMGroup

root

10 G

Create Logical Volume

LVMGroup

swap

2.3 G

Create Logical Volume

LVMGroup

home

5 G

Create Logical Volume

LVMGroup

var

3 G

Create Logical Volume

LVMGroup

srv

3 G

Create Logical Volume

LVMGroup

tmp

3 G

Create Logical Volume

LVMGroup

var-log

4 G

Finish

Sous LV VG LVMGroup, LV root - ...

Sélectionner #1 10 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/ - the root file system

Done setting up the partition

/ est le point de montage racine, ou le système de fichiers principal est monte : toutes les autres partitions ou peripheriques montes apparaissent comme des sous-repertoires de /

Sous LV VG LVGroup, LV swap - ...

Sélectionner #1 2.3 GB

Use as: swap area

Done setting up the partition

La partition swap est utilisée comme mémoire virtuelle pour compléter la RAM (mémoire vive) lorsqu'elle est saturée

Le swap est utile pour les opérations nécessitant beaucoup de mémoire ou pour éviter les plantages en cas de surcharge de la RAM

Les données sont stockées temporairement dans la partition swap lorsque la RAM est épuisée, pour éviter les erreurs de mémoire insuffisante

Elle peut aussi être utilisée pour l'hibernation : le contenu de la RAM peut être enregistré dans le swap pour pouvoir être restauré au démarrage

Sous LV VG LVGroup, LV home - ...

Sélectionner #1 5 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/home – user home directories

Done setting up the partition

/home est utilisé pour monter une partition dédiée aux fichiers des utilisateurs, pour que leurs données personnelles soient séparées du système principal

Sous LV VG LVGroup, LV var - ...

Sélectionner #1 3 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/var – variable data

Done setting up the partition

Le point de montage /var (Variable Data) stocke les données variables qui changent fréquemment ou qui sont générées dynamiquement par le système

Sous LV VG LVGroup, LV srv - ...

Sélectionner #1 3 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/srv – data for services provided by this system

Done setting up the partition

Le point de montage /srv contient les données spécifiques aux services fournis par le serveur. Cela inclut les fichiers utilisés par les services zeb, FTP ou autres

Sous LV VG LVGroup, LV tmp - ...

Sélectionner #1 3 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

/tmp – temporary files

Done setting up the partition

Le point de montage /tmp permet de stocker les fichiers temporaires créés par le système ou les applications (par les programmes ou les scripts, les sessions de navigation, les fichiers de travail intermédiaires...)

Sous LV VG LVGroup, LV var-log - ...

Sélectionner #1 4 GB

Use as: changer do not use pour Ext4 journaling file system

Mount point

Enter manually

/var/log

Done setting up the partition

Le point de montage /var/log est dédié au stockage des logs (journaux système et applicatifs). C'est un sous-répertoire de /var

Finish partitioning and write changes to disk

Vérifier les informations

Yes

Configure the package manager :

Scan extra installation media ?

No

Mirror :

France

deb.debian.org

Proxy information :

Laisser vide

Continue

Participate in the package survey ?

No

Software selection :

Décocher Debian desktop environment, GNOME et standard system utilities

Continue

Configure grub-pc :

Install the GRUB boot loader to your primary drive ?

Yes

Device for boot loader installation :

/dev/sda (ata-VBOX-HARDDISK-VB02d809df-3fce6ae1)

Installation complete :

It is time to boot into your new system

Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation

Continue

Please unlock disk sda5_crypt :

Entrer la passphrase

Debian GNU/Linux 12 juduchar tty1

juduchar login :

juduchar

puis entrer le mot de passe utilisateur

Résultat attendu :

juduchar@juduchar42

(juduchar : user ; juduchar42 : host)

Taper la commande :

lsblk

Résultat attendu (partie bonus) :

```
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                  8:0      0   35G  0 disk
├─sda1                              8:1      0   476M  0 part  /boot
├─sda2                              8:2      0     1K  0 part
├─sda5                              8:5      0  34.5G  0 part
│   └─sda5_crypt                   254:0     0  34.5G  0 crypt
│       ├─LVMGroup-root             254:1     0   9.3G  0 lvm    /
│       ├─LVMGroup-swap             254:2     0   2.1G  0 lvm    [SWAP]
│       ├─LVMGroup-home             254:3     0   4.7G  0 lvm    /home
│       ├─LVMGroup-var              254:4     0   2.8G  0 lvm    /var
│       ├─LVMGroup-srv              254:5     0   2.8G  0 lvm    /srv
│       ├─LVMGroup-tmp              254:6     0   2.8G  0 lvm    /tmp
│       └─LVMGroup-var--log         254:7     0   3.7G  0 lvm    /var/log
sr0                                  11:0     1 1024M  0 rom
```

Taper la commande :

cat /etc/os-release

Résultat attendu :

```
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

whoiam

Resultat attendu : juduchar

Login as root :

\$ su root

Taper le mot de passe root

whoiam

Resultat attendu : root

Vous allez installer et configurer sudo selon une pratique stricte.

Install sudo :

apt update

apt upgrade

apt install sudo

Cet utilisateur appartiendra au groupe sudo

sudo usermod -aG sudo juduchar

groups juduchar

Resultat attendu :

cdrom floppy sudo audio dip video plugdev users netdev bluetooth

Sortir du mode root pour revenir à juduchar :

exit

Se déconnecter de juduchar

exit

Se reconnecter :

juduchar

puis entrer le mot de passe utilisateur

\$ whoami

Resultat attendu : juduchar

\$ sudo whoami

Resultat attendu : root

Activer la synchronisation avec le NTP timesyncd :

timedatectl

sudo apt install systemd-timesyncd

sudo systemctl enable systemd-timesyncd

sudo systemctl start systemd-timesyncd

sudo timedatectl set-ntp true

timedatectl

Mettre a jour les paquets :

sudo apt update

sudo apt upgrade

Cet utilisateur appartiendra au groupe user42

Ajouter le groupe user42 :

```
sudo groupadd user42
```

Ajouter l'utilisateur juduchar au groupe user42 :

```
sudo usermod -aG user42 juduchar
```

Vérifier que l'utilisateur soit bien ajouté au groupe :

```
groups juduchar
```

Se déconnecter :

```
exit
```

Se reconnecter :

```
juduchar
```

puis entrer le mot de passe utilisateur

Configurer sudo selon une pratique stricte :

Pour mettre en place une configuration stricte dans votre groupe sudo, il faudra remplir les conditions suivantes :

- L'authentification en utilisant sudo sera limitée à 3 essais en cas de mot de passe erroné.

Taper la commande :

```
sudo visudo
```

Ajouter la ligne :

```
Defaults    passwd_tries=3
```

CTRL + o

Enter

CTRL + x

Tester en se reconnectant puis en tapant sudo whoami en tapant 3 fois un mauvais mot de passe

· Un message de votre choix s'affichera en cas d'erreur suite à un mauvais mot de passe lors de l'utilisation de sudo.

Taper la commande :

sudo visudo

Ajouter la ligne :

Defaults badpass_message="You have entered a bad password"

CTRL + o

Enter

CTRL + x

Tester en se reconnectant puis en tapant sudo whoami en tapant un mauvais mot de passe

· Chaque action utilisant sudo sera archivée, aussi bien les inputs que les outputs.

Archiver les inputs :

Taper la commande :

```
sudo visudo
```

Ajouter la ligne :

```
Defaults    log_input
```

```
CTRL + o
```

```
Enter
```

```
CTRL + x
```

Vérifier en tapant :

```
sudo -i
```

```
sudo whoami
```

```
sudoreplay -l
```

Noter la valeur de TSID pour la dernière ligne affichée (exemple 000009)

```
sudoreplay 000009 (par exemple)
```

Resultat attendu :

```
Replaying sudo session : /usr/bin/whoami
```

Archiver les inputs :

Taper la commande :

```
sudo visudo
```

Ajouter la ligne :

```
Defaults    log_output
```

```
CTRL + o
```

```
Enter
```

```
CTRL + x
```

Vérifier en tapant :

```
sudo whoami
```

```
sudoreplay -l
```

Noter la valeur de TSID pour la dernière ligne affichée (exemple 00000A)

```
sudoreplay 00000A (par exemple)
```

Resultat attendu :

```
Replaying sudo session : /usr/bin/whoami
```

```
root
```

Le journal se trouvera dans le dossier /var/log/sudo/

Taper la commande :

```
sudo visudo
```

Ajouter la ligne :

```
Defaults    iolog_dir=/var/log/sudo
```

```
CTRL + o
```

```
Enter
```

```
CTRL + x
```

Vérifier en tapant :

```
sudo -i
```

```
sudo whoami
```

```
cd /var/log/sudo
```

```
ls
```

```
cd 00
```

```
cd 00
```

```
cd 01
```

cat log.json

Vérifier que la valeur pour « command » est /usr/bin/whoami

sudoreplay -d /var/log/sudo -l

sudoreplay -d /var/log/sudo 000001

Resultat attendu :

Replaying sudo session : /usr/bin/whoami

root

Taper la commande :

sudo visudo

Ajouter la ligne :

Defaults logfile="/var/log/sudo/sudo.log"

CTRL + o

Enter

CTRL + x

· Le mode TTY sera activé pour des questions de sécurité.

Taper la commande :

sudo visudo

Ajouter la ligne :

Defaults requiretty

CTRL + o

Enter

CTRL + x

· Les paths utilisables par sudo seront restreints, là encore pour des questions de sécurité. Exemple :

`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

Taper la commande :

`sudo visudo`

Editer la ligne : Defaults secure_path comme ceci (pour y ajouter /snap/bin) :

Defaults secure_path=

`"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"`

CTRL + o

Enter

CTRL + x

Vérifier que les commandes qui ne sont pas dans le secure_path ne fonctionnent pas :

`echo «echo Hello World» > /tmp/test-script`

`chmod +x /tmp/test-script`

`sudo test-script`

Resultat attendu :

`"sudo: test-script: command not found"`

`rm /tmp/test-script`

Vérifier que les commandes se trouvant dans le `secure_path` fonctionnent bien :

```
mkdir snap
```

```
cd snap
```

```
mkdir bin
```

```
echo «echo Hello World» > /snap/bin/test-script2
```

```
chmod +x /snap/bin/test-script2
```

```
sudo test-script2
```

Résultat attendu :

Hello World

```
rm /snap/bin/test-script2
```

```
rmdir /snap/bin
```

```
rmdir /snap
```

Un service SSH sera obligatoirement actif sur le port 4242 dans votre machine virtuelle.

Installer SSH :

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install openssh-server
```

Vérifier le statut du serveur SSH :

```
sudo systemctl status ssh
```

Resultat attendu :

Loaded : loaded ... enabled .. enabled

Active : active (running)

Vous allez configurer votre système d'exploitation avec le pare-feu UFW (ou pare-feu pour Rocky) et ainsi ne laisser ouvert que le port 4242 dans votre machine virtuelle.

Installer UFW :

`sudo apt update`

`sudo apt upgrade`

`sudo apt install ufw`

Activer UFW :

`sudo ufw enable`

Le message : Firewall is active and enabled on system startup doit s'afficher

Verifier le statut d'UFW :

`sudo ufw status verbose`

Status : active

Autoriser la connexion via le port SSH :

`sudo ufw allow ssh`

Les lignes suivantes devraient s'afficher :

Rule added

Rule added (v6)

`sudo ufw status`

Resultat attendu :

22/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)

Ajouter la règle pour le port 4242 :

`sudo ufw allow 4242`

`sudo ufw status`

Resultat attendu :

22/tcp	ALLOW	Anywhere
4242	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
4242 (v6)	ALLOW	Anywhere (v6)

Un service SSH sera obligatoirement actif sur le port 4242 dans votre machine virtuelle.

Configurer SSH :

Modifier le port SSH 22 (par défaut) pour le port 4242 :

`sudo nano /etc/ssh/sshd_config`

Modifier la ligne :

#Port 22

Par :

Port 4242

Vous allez configurer votre système d'exploitation avec le pare-feu UFW (ou pare-feu pour Rocky) et ainsi ne laisser ouvert que le port 4242 dans votre machine virtuelle.

Supprimer l'accès au port 22 pour UFW :

```
sudo ufw delete allow ssh
```

```
sudo ufw status
```

Resultat attendu :

4242	ALLOW	Anywhere
4242 (v6)	ALLOW	Anywhere (v6)

Pour des questions de sécurité, on ne devra pas pouvoir se connecter par SSH avec l'utilisateur root

Interdire la connexion en SSH avec le compte root :

Modifier la ligne :

```
#PermitRootLogin prohibit-password
```

Par

```
PermitRootLogin no
```

CTRL + o

Enter

CTRL + x

Redémarrer le service SSH :

```
sudo service ssh restart
```

Afficher les ports ouverts :

```
sudo ss -tuln
```

Resultat attendu :

```
LISTEN 0 128 0.0.0.0:4242 0.0.0.0:*
```

```
LISTEN 0 128 [::]:4242 [::]:*
```

Votre pare-feu devra être actif au lancement de votre machine virtuelle.

Redemarrer la VM, et taper la commande :

```
sudo ufw status
```

Resultat attendu :

Status : active

Configurer le port forwarding de la VM :

Arrêter la VM

Aller dans VirtualBox, puis dans Settings, Network, Adapter 1, puis Port Forwarding

Créer une nouvelle règle de redirection

Indiquer 4242 pour Port hôte, et 4242 pour Port invité

Vérifier que le port forwarding fonctionne correctement :

Relancer la VM

Une demande d'autorisation pour la redirection réseau devrait apparaître, accepter

Taper :

```
sudo systemctl restart ssh
```

```
sudo service sshd status
```

Les lignes suivantes devraient apparaître :

```
Starting ssh.service
```

```
Server listening on 0.0.0.0 port 4242
```

```
Server listening on :: port 4242
```

```
Started ssh.service
```

Ouvrir la console en mode admin sur l'hôte (avec un terminal 42, ou invite de commande Windows)

```
ssh juduchar@localhost -p 4242
```

```
yes
```

```
juduchar@juduchar42 devrait s'afficher
```

```
whoami
```

```
juduchar devrait s'afficher
```

Une commande avec sudo (sudo whoami par exemple) devrait s'afficher dans les logs du serveur

Taper exit dans la console pour quitter la connexion ssh

AppArmor pour Debian devra également rester actif.

Vérifier que AppArmor est bien actif :

```
sudo systemctl status apparmor
```

Resultat attendu :

- apparmor.service - Load AppArmor profiles

Loaded: loaded (/lib/systemd/system/apparmor.service; enabled; vendor preset: enabled)

Active: active (exited) since ...

Pour mettre en place une politique de mot de passe fort, il faudra remplir les conditions suivantes :

Règles d'expiration du mot de passe :

```
nano /etc/login.defs
```

Votre mot de passe devra expirer tous les 30 jours.

```
PASS_MAX_DAYS 30
```

· Le nombre minimum de jours avant de pouvoir modifier un mot de passe sera configuré à 2.

PASS_MIN_DAYS 2

· L'utilisateur devra recevoir un avertissement 7 jours avant que son mot de passe n'expire.

PASS_WARN_AGE 7

Appliquer ces mêmes règles de sécurité à l'utilisateur déjà existant et au root :

Expire tous les 30 jours :

sudo chage -M 30 juduchar

sudo chage -M 30 root

Vérifier avec :

sudo chage -l juduchar

sudo chage -l root

Resultat attendu :

Maximum number of days between password change : 30

2 jours minimum avant de pouvoir modifier un mot de passe :

sudo chage -m 2 juduchar

sudo chage -m 2 root

Vérifier avec :

sudo chage -l juduchar

sudo chage -l root

Resultat attendu :

Mimimum number of days between password change : 2

Envoyer un avertissement à l'utilisateur 7 jours avant que son mot de passe n'expire :

```
sudo chage -W 7 juduchar
```

```
sudo chage -W 7 root
```

Vérifier avec :

```
sudo chage -l juduchar
```

```
sudo chage -l root
```

Resultat attendu :

Number of days of warning before password expires : 7

Installer pwquality :

```
sudo apt install libpam-pwquality
```

Editer le fichier /etc/security/pwquality.conf :

```
nano /etc/security/pwquality.conf
```

Votre mot de passe sera de 10 caractères minimums

```
minlen = 10
```

Dont une majuscule, une minuscule et un chiffre

dcredit = -1

lcredit = -1

ucredit = -1

Ne devra pas comporter plus de 3 caractères identiques consécutifs

maxrepeat = 3

Le mot de passe ne devra pas comporter le nom de l'utilisateur.

usercheck = 1

Le mot de passe devra comporter au moins 7 caractères qui ne sont pas présents dans l'ancien mot de passe :

difok = 7

La règle suivante ne s'applique pas à l'utilisateur root : le mot de passe devra comporter au moins 7 caractères qui ne sont pas présents dans l'ancien mot de passe.

· Bien entendu votre mot de passe root devra suivre cette politique.

nano /etc/pam.d/common-password

Au dessus de :

```
password [success=1 default=ignore] pam_unix.so obscure use_authok  
try_first_pass yescrypt
```

Ajouter les lignes suivantes :

```
password [success=1 default=ignore] pam_succeed_if.so user = root
```

password requisite pam_pwquality.so retry=3 difok=0

password requisite pam_pwquality.so retry=3 difok=7

Redémarrer la VM

sudo chage -m 0 juduchar

Changer le mot de passe de l'utilisateur avec la commande passwd

Tester avec plusieurs combinaisons de mot de passe (sans majuscule, sans minuscule, sans chiffre, avec moins de 10 caractères, avec plus de 3 caractères identiques consécutifs, et contenant le login de l'utilisateur

Enfin, tester avec moins de 7 caractères différents de l'ancien mot de passe

Tous ces tests devraient échouer

Tester la même chose avec le compte root (se connecter en root pour cela), puis :

sudo chage -m 0 root

Le test avec moins de 7 caractères différents de l'ancien mot de passe ne devrait pas échouer

Modifier les mots de passe root et juduchar

sudo chage -m 2 juduchar

sudo chage -m 2 root

Enfin, vous devrez mettre en place un petit script nommé monitoring.sh. Ce dernier sera à développer en bash :

```
cd /root
```

```
nano monitoring.sh
```

```
#!/bin/bash
```

```
CTRL + o
```

```
Enter
```

```
CTRL + x
```

```
chmod 700 monitoring.sh
```

. L'architecture de votre système d'exploitation ainsi que sa version de kernel.

· Le nombre de processeurs physiques.

· Le nombre de processeurs virtuels.

· La mémoire vive disponible actuelle sur votre serveur ainsi que son taux d'utilisation sous forme de pourcentage.

· La mémoire disponible actuelle sur votre serveur ainsi que son taux d'utilisation sous forme de pourcentage.

· Le taux d'utilisation actuel de vos processeurs sous forme de pourcentage.

· La date et l'heure du dernier redémarrage.

· Si LVM est actif ou pas.

· Le nombre de connexions actives.

· Le nombre d'utilisateurs utilisant le serveur.

· L'adresse IPv4 de votre serveur, ainsi que son adresse MAC (Media Access Control).

· Le nombre de commande exécutées avec le programme sudo.


```
#!/bin/bash

architecture=$(uname -a)

cpu_physical=$(grep "physical id" /proc/cpuinfo | sort -u | wc -l)

v_cpu=$(grep "processor" /proc/cpuinfo | wc -l)

used_memory_mb=$(free -m | grep "Mem:" | awk '{print $3}')
total_memory_mb=$(free -m | grep "Mem:" | awk '{print $2}')
used_memory_kb=$(free -k | grep "Mem:" | awk '{print $3}')
total_memory_kb=$(free -k | grep "Mem:" | awk '{print $2}')

memory_usage_rate=$(awk "BEGIN {printf \"%.2f\\\", ($used_memory_kb / $total_memory_kb) * 100}")
used_disk_memory=$(df -h --total | grep "total" | awk '{print $3}')
total_disk_memory=$(df -h --total | grep "total" | awk '{print $2}')
disk_memory_usage_rate=$(df -h --total | grep "total" | awk '{print $5}')

us_load=$(top -bn1 | grep "%Cpu(s)" | awk '{print $2}')
sy_load=$(top -bn1 | grep "Cpu(s)" | awk '{print $4}')
ni_load=$(top -bn1 | grep "Cpu(s)" | awk '{print $6}')
cpu_load=$(awk "BEGIN {printf \"%.1f\\\", $us_load + $sy_load + $ni_load}")

last_boot_date=$(who -b | awk '{print $3}')
last_boot_time=$(who -b | awk '{print $4}')

lvm_active_logical_volumes=$(sudo lvs | wc -l)
lvm_active_volume_groups=$(sudo vgs | wc -l)
lvm_active_mounted_partitions=$(df | grep "^/dev/mapper" | wc -l)
lvm_active_total=$((lvm_active_logical_volumes + lvm_active_volume_groups + lvm_active_mounted_partitions))
if [ $lvm_active_total -eq 0 ]; then
    lvm_active="no"
else
    lvm_active="yes"
fi

tcp_connections=$(ss -s | grep "TCP:" | awk '{print $4}' | sed 's/,//')

user_log=$(who -q | awk -F '=' '/# users=/ {print $2}')

network_ip=$(hostname -I | awk '{print $1}')
main_network_interface=$(ip route show default | grep -o 'dev [^]*' | awk '{print $2}')
network_mac_address=$(ip link show $main_network_interface | grep 'link/ether' | awk '{print $2}')

sudo_command_count=$(grep -o "COMMAND=[^]*" /var/log/sudo/sudo.log | wc -l)
```

```
echo "#Architecture: $architecture
#CPU physical : $cpu_physical
#vCPU : $v_cpu
#Memory Usage: ${used_memory_mb}/${total_memory_mb}MB (${memory_usage_rate}%)
#Disk Usage: ${used_disk_memory}/${total_disk_memory} (${disk_memory_usage_rate}%)
#CPU load: ${cpu_load}%
#Last boot: $last_boot_date $last_boot_time
#LVM use: $lvm_active
#Connexions TCP : ${tcp_connexions:-0} ESTABLISHED
#User log: $user_log
#Network IP $network_ip (${network_mac_address})
#Sudo : ${sudo_command_count:-0} cmd"
```

Dès le lancement de votre serveur, le script écrira des informations toutes les 10 minutes sur tous les terminaux (jetez un oeil du côté de wall)

```
systemctl enable cron
```

```
systemctl status cron
```

```
crontab -e
```

```
*/10 * * * * bash /root/monitoring.sh | wall
```

```
sudo systemctl restart cron
```

Mettre en place un site web WordPress fonctionnel avec, comme services, lighttpd, MariaDB et PHP.

Avant d'utiliser apt update, synchroniser la date du serveur :

```
sudo systemctl restart systemd-timesyncd
```

Mettre à jour la liste des paquets :

```
sudo apt update
```

Mettre à jour les paquets :

```
sudo apt upgrade
```

```
sudo apt install wget
```

```
sudo apt install curl
```

```
sudo apt install php
```

```
sudo apt install php-common
```

```
sudo apt install php-cgi
```

```
sudo apt install php-cli
```

```
sudo apt install php-mysql
```

```
sudo apt install php-curl
```

```
sudo apt install php-gd
```

```
sudo apt install php-zip
```

Vérifier si apache2 est actif :

```
systemctl status apache2
```

Désinstaller Apache2 et supprimer ses fichiers de configurations associées (plus complet que `sudo apt remove`, qui ne fait que désinstaller le paquet) :

```
sudo apt purge apache2
```

Supprimer les dépendances inutilisées (après la désinstallation d'Apache 2) :

```
sudo apt autoremove
```

Installer Lighttpd :

```
sudo apt install lighttpd
```

Vérifier si Lighttpd a bien été installé :

```
sudo lighttpd -v
```

Démarrer le serveur web Lighttpd :

```
sudo systemctl start lighttpd
```

Permet de lancer automatiquement Lighttpd au démarrage du serveur :

```
sudo systemctl enable lighttpd
```

Vérifier si Lighttpd est actif :

```
sudo systemctl status lighttpd
```

La commande suivante permet d'activer le module fastcgi pour Lighttpd :

```
sudo lighty-enable-mod fastcgi
```

Redémarrer Lighttpd :

```
sudo systemctl restart lighttpd
```

Configure FastCGI spécifiquement pour PHP, pour qu'il cherche une configuration prête à l'emploi pour traiter les scripts PHP :

```
sudo lighty-enable-mod fastcgi-php
```

Redémarrer Lighttpd :

```
sudo systemctl restart lighttpd
```

Recharger la configuration de Lighttpd :

```
sudo service lighttpd force-reload
```

Autoriser le port 80 avec UFW pour permettre l'accès au serveur web depuis une machine cliente en utilisant le protocole TCP :

```
sudo ufw allow 80/tcp
```

Recharger UFW pour qu'il prenne en compte cette modification :

```
sudo ufw reload
```

Voir les ports autorisés ou interdits :

```
sudo ufw status
```

Résultat attendu :

```
Status: active

To Action From
--
4242 ALLOW Anywhere
80/tcp ALLOW Anywhere
4242 (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
```

Après avoir arrêté la VM, aller dans VirtualBox, puis dans Settings, Network, Adapter 1, puis Port Forwarding

Créer une nouvelle règle de redirection

Indiquer 8080 pour Port hôte, et 80 pour Port invité

Redémarrer la VM, et accéder à Apache2 sur le système invité via navigateur avec l'URL suivant :

<http://localhost:8080>

Installer MariaDB Server :

```
sudo apt install mariadb-server
```

Démarrer MariaDB :

```
sudo systemctl start mariadb
```

Executer un script de sécurité fourni avec MariaDB (et MySQL), pour renforcer la sécurité du serveur de base de données, en désactivant les configurations par défaut potentiellement vulnérables, en appliquant les paramètres de sécurité essentiels :

```
sudo mysql_secure_installation
```

Enter current password for root or enter id you've just installed MariaDB :

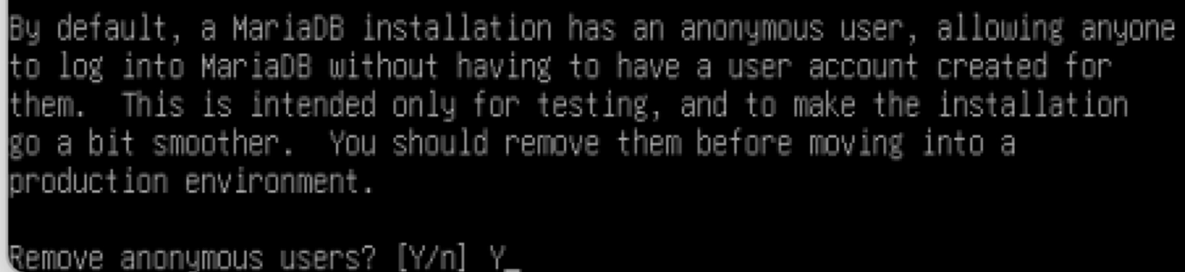
Appuyer sur entrée

Setting the root password (no) or using the unix_socket (ensure that nobody can log into the MariaDB root user without the proper authorization) (yes)

Confirmer avec Y

Change the root password ?

Appuyer sur Y pour définir un mot de passe pour le root de MariaDB



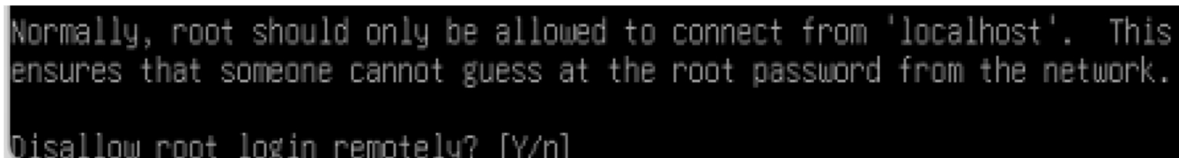
```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
```

Supprimer les utilisateurs anonymes de MariaDB (qui peuvent se connecter à MariaDB sans compte spécifique) (oui) :

Y

Désactiver l'accès à la connexion root de MariaDB à distance (oui) :



```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
```

Y

Supprimer la base de données « test » (oui) :

```
By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.
```

```
Remove test database and access to it? [Y/n]
```

Y

Recharger les tables de privilèges pour appliquer immédiatement ces modifications (oui) :

```
Reloading the privilege tables will ensure that all changes made so far will take effect immediately.
```

Y

Se connecter à MariaDB en tant que root (-u root : en tant que l'utilisateur root), en demandant le mot de passe avant d'établir la connexion (-p : password) :

```
sudo mysql -u root -p
```

Créer la base de données WordPress (remplacer wordpress par un nom plus sécurisé !) :

```
CREATE DATABASE wordpress;
```

Créer l'utilisateur de cette base de données, avec son mot de passe (remplacer wordpress_user et password par un nom et un mot de passe plus sécurisé !) :

```
CREATE USER 'wordpress_user'@'localhost' IDENTIFIED BY 'password';
```

Accorder tous les privilèges sur cette base de données à l'utilisateur wordpress_user (remplacer par le nom défini plus haut) :

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress_user'@'localhost';
```

Appliquer les modifications :

```
FLUSH PRIVILEGES;
```

Quitter MariaDB :

```
EXIT;
```

Télécharger et configurer WordPress :

Se déplacer dans le répertoire par défaut de Lighttpd :

```
cd /var/www/html
```

Télécharger la dernière version de WordPress depuis le site officiel :

```
sudo wget https://wordpress.org/latest.tar.gz
```

Extraire le contenu de l'archive téléchargée :

```
sudo tar -xzf latest.tar.gz
```

Déplacer les fichiers de WordPress dans le répertoire racine /var/www/html :

```
sudo mv wordpress/* /var/www/html/
```

Supprimer l'archive téléchargée et le dossier WordPress vide :

```
sudo rm -rf latest.tar.gz wordpress
```

Accorder les permissions nécessaires pour que Lighttpd puisse accéder aux fichiers :

```
sudo chown -R www-data:www-data /var/www/html
```

Accorder toutes les permissions au propriétaire des fichiers, le droit de lecture et d'exécution pour les autres utilisateurs, au contenu (-R) de /var/www/html :

```
sudo chmod -R 755 /var/www/html
```


Renommer le fichier de configuration d'exemple pour pouvoir le personnaliser et l'appliquer à WordPress :

```
sudo mv /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
```

Editer le fichier de configuration de WordPress :

```
sudo nano /var/www/html/wp-config.php
```

```
// Remplacez 'wordpress' par le nom de votre base de données.
```

```
define('DB_NAME', 'wordpress');
```

```
// Remplacez 'wordpressuser' par votre nom d'utilisateur MySQL.
```

```
define('DB_USER', 'wordpressuser');
```

```
// Remplacez 'password_here' par votre mot de passe MySQL.
```

```
define('DB_PASSWORD', 'password_here');
```

Aller sur cet URL depuis le navigateur du client :

<http://localhost:8080/>

Cliquer sur Install WordPress

Indiquer le nom du site, le nom de l'utilisateur (admin) WordPress, son mot de passe, son email

Cocher la case Discourage search engines from indexing this site

Cliquer sur Install WordPress

Se connecter

Mettre en place un service qui vous semble utile (NGINX/Apache2 exclus !). Durant la soutenance, vous aurez à justifier ce choix.

Installation de Gitea :

Préparation :

Avant d'utiliser apt update, synchroniser la date du serveur :

```
sudo systemctl restart systemd-timesyncd
```

Mettre à jour la liste des paquets :

```
sudo apt update
```

Mettre à jour les paquets :

```
sudo apt upgrade
```

```
sudo apt install --reinstall ca-certificates
```

```
sudo update-ca-certificates
```

```
cd /usr/local/bin
```

```
wget -O gitea https://dl.gitea.io/gitea/1.22.3/gitea-1.22.3-linux-amd64
```

```
chmod +x gitea
```

```
sudo adduser --system --shell /bin/bash --gecos 'Git Version Control' --group --disabled-password --home /home/git git
```

```
sudo mkdir -p /var/lib/gitea/{custom,data,indexers,public,log}
```

```
sudo chown git:git /var/lib/gitea/{data,indexers,log}
```

```
sudo chmod 750 /var/lib/gitea/{data,indexers,log}
```

```
sudo mkdir /etc/gitea
```

```
sudo chown root:git /etc/gitea
```

```
sudo chmod 770 /etc/gitea
```

```
sudo nano /etc/systemd/system/gitea.service
```

```
[Unit]
```

```
Description=Gitea (Git with a cup of tea)
```

```
After=syslog.target
```

```
After=network.target
```

```
[Service]
```

```
RestartSec=2s
```

```
Type=simple
```

```
User=git
```

```
Group=git
```

```
WorkingDirectory=/var/lib/gitea/
```

```
ExecStart=/usr/local/bin/gitea web -c /etc/gitea/app.ini
```

```
Restart=always
```

```
Environment=USER=git HOME=/home/git GITEA_WORK_DIR=/var/lib/gitea
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Activer Gitea et le démarrer :

```
sudo systemctl enable gitea
```

```
sudo systemctl start gitea
```

Accéder à l'interface web Gitea :

<http://127.0.0.1:3000/>

Configuration de Gitea :

Créer la base de données gitea :

Se connecter à MariaDB en tant que root :

```
sudo mysql -u root -p
```

```
CREATE DATABASE gitea CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

```
CREATE USER 'gitea'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON gitea.* TO 'gitea'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

```
cd /usr/local/bin
```

```
mkdir custom
```

```
cd custom
```

```
mkdir conf
```

Puis attribuer les bonnes permissions :

```
sudo chown -R git:git /usr/local/bin/custom/conf
```

```
sudo chmod -R 750 /usr/local/bin/custom/conf
```

Configuration initiale

Si vous exécutez Gitea dans Docker, veuillez lire la [documentation](#) avant de modifier les paramètres.

Paramètres de la base de données

Gitea nécessite MySQL, PostgreSQL, MSSQL, SQLite3 ou TiDB (avec le protocole MySQL).

Type de base de données *	MySQL ▼
Hôte *	127.0.0.1:3306
Nom d'utilisateur *	gitea
Mot de passe *	
Nom de base de données *	gitea

Configuration générale

Titre du site *	Gitea: Git with a cup of tea
	Entrez ici le nom de votre société.
Emplacement racine des dépôts *	/var/lib/gitea/data/gitea-repositories
	Les dépôts Git distants seront stockés dans ce répertoire.
Répertoire racine Git LFS	/var/lib/gitea/data/lfs
	Les fichiers suivis par Git LFS seront stockés dans ce dossier. Laissez vide pour désactiver LFS.

Modifier le port du serveur SSH par 4242

Exécuter avec le compte d'un autre utilisateur *

git

Le nom d'utilisateur du système d'exploitation sous lequel Gitea fonctionne. Notez que cet utilisateur doit avoir accès au dossier racine du dépôt.

Domaine du serveur *

127.0.0.1

Domaine ou adresse d'hôte pour le serveur.

Port du serveur SSH

22

Port d'écoute du serveur SSH. Laissez le vide pour le désactiver.

Port d'écoute HTTP de Gitea *

3000

Port sur lequel le serveur web Gitea attendra des requêtes.

URL de base de Gitea *

http://127.0.0.1:3000/

Adresse HTTP(S) de base pour les clones git et les notifications par courriel.

Chemin des journaux *

/var/lib/gitea/log

Les fichiers de journalisation seront écrits dans ce répertoire.

☐ Activer la vérification des mises-à-jour

Vérifie les mises à jour régulièrement en se connectant à gitea.io.

Paramètres facultatifs

► Paramètres de Messagerie

► Paramètres Serveur et Tierce Parties

► Paramètres de compte administrateur

- Activer la vérification des mises à jour
- Activer le mode hors-ligne (empêche Gitea d'effectuer des connexions externes à Internet)
- Désactiver Gravatar
- Ne pas activer les avatars unifiés
- Ne pas activer l'authentification OpenID (empêche les utilisateurs de pouvoir se connecter avec leur compte Google, Facebook, etc)
- Désactiver le formulaire d'inscription (ainsi, les comptes ne pourront être créés que par un administrateur, cela empêche les nouveaux utilisateurs de se créer un compte via le formulaire d'inscription standard et limite l'accès à Gitea à des utilisateurs spécifiques)
- Ne pas autoriser l'inscription avec des services externes
- Ne pas activer l'inscription OpenID
- Ne pas activer le CAPTCHA d'inscription

- Exiger la connexion à un compte pour voir les pages publiques (les visiteurs connectés ne pourront pas voir les pages publiques de Gitea, comme les dépôts publics, les wikis, etc) (Installation privée et sécurisée)
- Masquer les adresses courriels par défaut (empêche de voir les adresses mail des utilisateurs dans l'interface publique, dans les profils ou les commits visibles publiquement) (améliore la confidentialité des utilisateurs)
- Ne pas autoriser la création d'organisations par défaut (les organisations sont des groupes de collaboration pour les dépôts et projets partagés)
- Activer la fonctionnalité de suivi du temps sur Gitea pour les tickets et les issues, pour suivre et enregistrer le temps passé sur les tâches

Créer un compte administrateur :

Choisir un nom d'utilisateur administrateur et un mot de passe sécurisé

Pour créer un utilisateur sur Gitea (remplacer username par le nom d'utilisateur, mail@example.com par le mail de l'utilisateur, et password par son mot de passe)

--config permet d'indiquer où se trouve le fichier de config app.ini

Il n'est pas correctement paramétré (bug ?), il faut donc le préciser :

```
sudo -u git /usr/local/bin/gitea --config /etc/gitea/app.ini admin user create --username
username --email mail@example.com --password password
```

VERIF :

```
head -n 2 /etc/os-release
```

Résultat attendu :

```
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
```

```
NAME="Debian GNU/Linux"
```


/usr/sbin/aa-status

Résultat attendu :

apparmor module is loaded

ss -tunlp

Résultat attendu :

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
tcp	LISTEN	0	128	0.0.0.0:4242	0.0.0.0:*	users(("sshd",pid=654,fd=3))
tcp	LISTEN	0	128	:::4242	:::*	users(("sshd",pid=654,fd=4))

/usr/sbin/ufw status

Résultat attendu :

```
root@juduchar42:~# /usr/sbin/ufw status
Status: active

To Action From
--
4242 ALLOW Anywhere
4242 (v6) ALLOW Anywhere (v6)
```