

Born2BeRoot	3
Téléchargement	3
1. Initialisation de la machine virtuelle	3
2. Premier lancement de la machine virtuelle	5
Partitionnage	6
fdisk	6
parted	7
Chiffrement	10
LUKS	10
Création groupe de volume	12
Volumes physiques (PV (Physical Volumes)) :	12
Groupes de volumes (VG (Volume Groups)) :	12
Volumes logiques (LV (Logical Volumes)) :	12
Avantages de LVM	12
3. Installation de Rocky	15
4. Configuration du serveur	19
Comparaison entre SELinux et AppArmor	19
Gestion de SSH dans SELinux	19
Installation d'un paquet	20
Ajuster SELinux pour SSH	20
Changement du port SSH	21
5. Secure Shell (SSH)	21
Configuration de la machine virtuelle pour SSH	22
Configuration du pare-feu pour SSH	25
6. Configuration du HostName	26
Utilisation de hostnamectl pour Configurer le Nom d'Hôte	26
7. Configuration de Politiques de Mot de Passe et de Droits Sudo Sécurisées sur Rocky Linux	28
C'est quoi TTY?	31
Pourquoi activer le Mode TTY ?	31
8. Automatiser la Surveillance Système avec Bash et Cron	32
Création de monitoring.sh	32
Différence entre CPU physique et vCPU (processeur virtuel)	36
Automatiser les tâches avec Cron	37
Comment vérifier ces paramètres ?	38
Partie Bonus	39
Configuration de WordPress avec Lighttpd, MariaDB et PHP	39
Lighttpd : un serveur Web léger	39
MariaDB : une base de données robuste	39
PHP : traitement du contenu dynamique	40
Configurer le pare-feu	40
Configurer lighttpd	41
Configurer PHP-FPM	43

Sécuriser l'installation de MariaDB	44
Créer une base de données et un utilisateur WordPress'	45
Installation WordPress	47
Configurer WordPress	48
Ajout d'un service supplémentaire : Fail2Ban	49
Ajouter SFTP	50
Tu veux tester ton Born2BeRoot?	51
ANNEXE	52
Commande pour créer la signature	52
Qu'est-ce qu'une signature ?	52
Caractéristiques de la signature	52
Comparer les signatures	52
Résumé des différences basiques entre Rocky Linux et Debian	53
Différences entre apt, aptitude et dnf	54
Comment tester si pas d'environnement graphique installé	54
Comment savoir si Firewalld est lancé	55
Comment savoir si SSHD (Secure Shell Daemon) est lancé	55
Comment tester les groupes d'appartenance de l'utilisateur	55
Comment créer un utilisateur	56
Fichier de politique de mot de passe	57
Comment changer le HostName	58
sudo vi /etc/hosts	58
Les partitions et LVM (Logical Volume Management)	59
Configuration des partitions et volumes logiques	59
Comment vérifier sudo	60
Tester Firewalld	61
Tester SSHD (Secure Shell Daemon)	63
Tester le monitoring.sh	64
La partie bonus	66
Supprimer une entrée GRUB suite a une MAJ	66
Résumé des commandes	67
Commandes liées à la signature :	67
Commandes liées aux environnements graphiques :	67
Commandes liées à Firewalld :	67
Commandes liées à SSHD :	68
Commandes liées aux groupes et utilisateurs :	68
Commandes liées au fichier de politique de mot de passe :	68
Commandes générales diverses :	68
Commandes liées au monitoring.sh :	69
Commandes liées à Fail2Ban :	69
Commandes liées à la gestion des noyaux :	69
Commandes bonus liées à Lighttpd, MariaDB, PHP-FPM :	69
Répertoires mentionnés :	69

Born2BeRoot

Téléchargement

Téléchargement de Rocky-9.4-x86_64-minimal.iso :

https://download.rockylinux.org/pub/rocky/9/isos/x86_64/Rocky-9.5-x86_64-minimal.iso

1. Initialisation de la machine virtuelle

- Creation de la machine virtuelle "**Born2BeRoot**" dans le repertoire "/sgoinfre/goinfre/Perso/mdemare/Born2BeRoot/" avec "Rocky-9.4-x86_64-minimal.iso"
- Skip unattended Install : **true**
- hostname "**mdemare42**"
- Nom d'utilisateur : **mdemare**
- Mot de passe : **Machinevirtuelle42**
- Hostname/domain name : mdemare42
- install in background : **false**
- install guest additions : **false**
- Le mot de passe doit être de 10 caractères minimums dont une majuscule, une minuscule et un chiffre, et ne devra pas comporter plus de 3 caractères identiques consécutifs.
- Mémoire vive : **2048**
- Processors : **4**
- Disk size : **31.00 GB**
- pre-allocate full size : **false**

Récapitulatif

The following table summarizes the configuration you have chosen for the new virtual machine. When you are happy with the configuration press Finish to create the virtual machine. Alternatively you can go back and modify the configuration.

Machine Name and OS Type

Machine Name	Born2BeRoot
Machine Folder	/sgoinfre/goinfre/Perso/mdemare/Born2BeRoot
ISO Image	/home/mdemare/Downloads/Rocky-9.4-x86_64-boot.iso
Guest OS Type	Red Hat (64-bit)
Skip Unattended Install	true

Hardware

Mémoire vive	2048
Processor(s)	4
EFI Enable	false

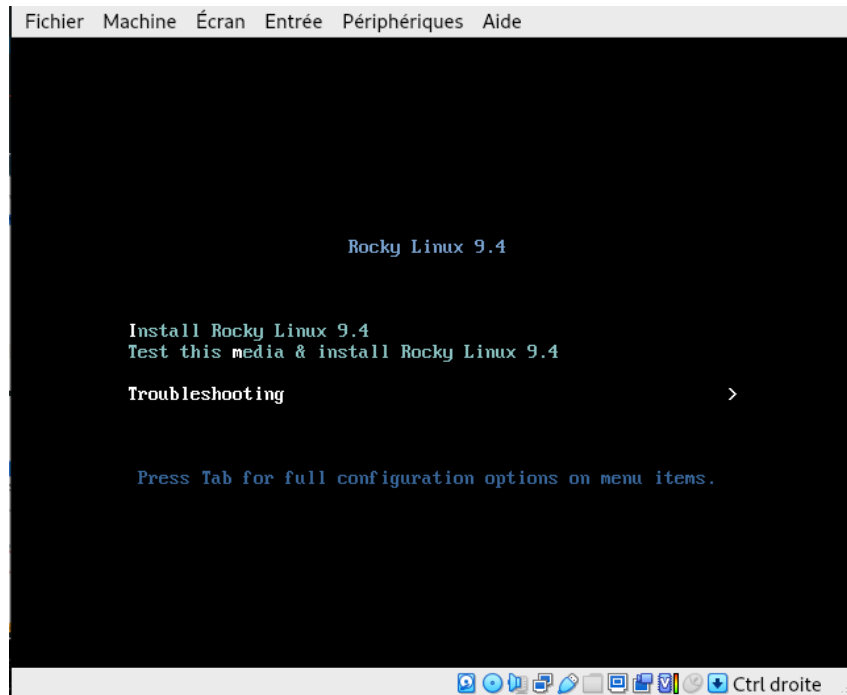
Disk

Disk Size	31.00 Gio
Pre-allocate Full Size	false

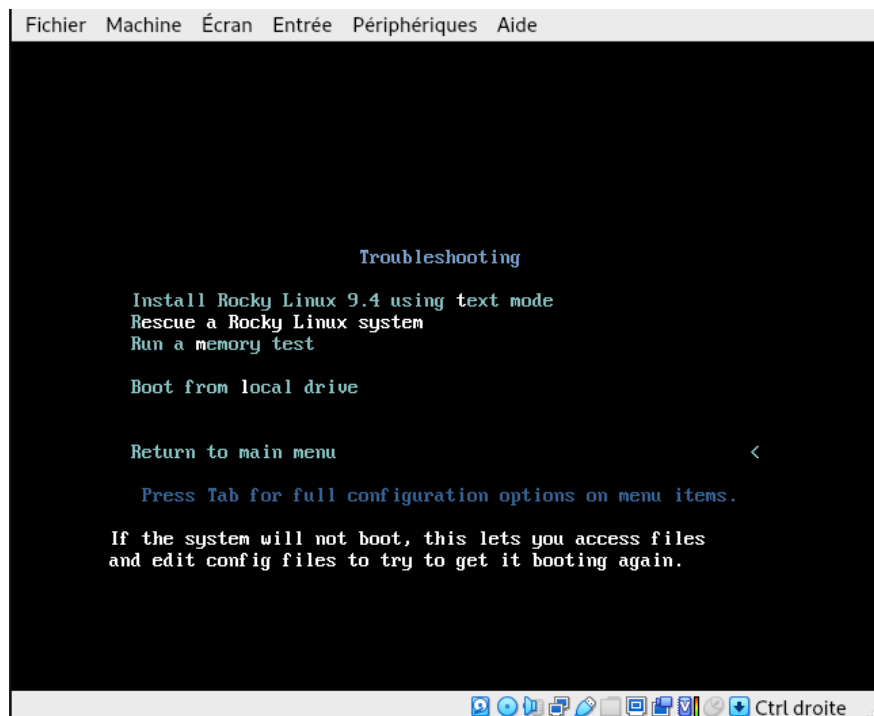
[Précédent](#)[Finish](#)

2. Premier lancement de la machine virtuelle

Au démarrage, à l'affichage de l'écran Rocky Linux 9.4, aller sur Troubleshooting, à l'aide des flèches,



puis sur Rescue a Rocky Linux system,



ensuite choisir (1) continue et appuyer sur ENTER pour continuer.

```
Starting installer, one moment...
anaconda 34.25.4.9-1.el9_4.rocky.0.3 for Rocky Linux 9.4 started.
 * installation log files are stored in /tmp during the installation
 * shell is available on TTY2
 * when reporting a bug add logs from /tmp as separate text/plain attachments
=====
Rescue

The rescue environment will now attempt to find your Linux installation and
mount it under the directory : /mnt/sysroot. You can then make any changes
required to your system. Choose '1' to proceed with this step.
You can choose to mount your file systems read-only instead of read-write by
choosing '2'.
If for some reason this process does not work choose '3' to skip directly to a
shell.

1) Continue
2) Read-only mount
3) Skip to shell
4) Quit (Reboot)

Please make a selection from the above: 1
=====
Rescue Shell

You don't have any Linux partitions.
When finished, please exit from the shell and your system will reboot.

Please press ENTER to get a shell:
bash-5.1# _
```

Partitionnage

Deux possibilité pour créer les partitions :

fdisk ou **parted**.

Fdisk et **parted** sont des outils puissants et largement utilisés pour le partitionnement des disques dans les systèmes Linux. Ils ont leurs avantages et leurs cas d'utilisation spécifiques, mais ils diffèrent également sur certains aspects clés.

fdisk

fdisk est un utilitaire textuel largement utilisé pour gérer les partitions de disque sur les systèmes Linux. Il prend en charge la création, la suppression et la modification de partitions sur un disque dur. Bien qu'il fonctionne principalement avec les tables de partition MBR (Master Boot Record), il offre également une prise en charge limitée des tables de partition GPT (GUID Partition Table). Voici quelques-unes des raisons de le choisir **fdisk** :

Familiarité : **fdisk** existe depuis longtemps et de nombreux utilisateurs sont habitués à l'utiliser.

Simplicité : **fdisk** fournit une interface simple pour la gestion des partitions, ce qui permet aux utilisateurs d'accomplir plus facilement leurs tâches.

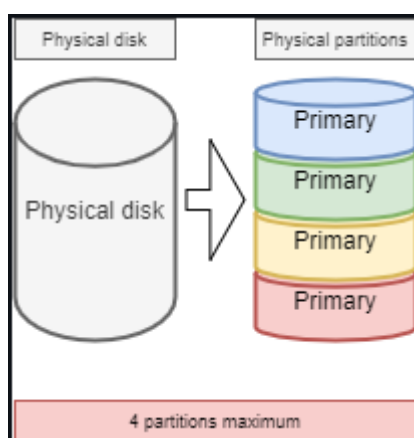
parted

parted est un autre utilitaire de ligne de commande pour le partitionnement des disques durs sur les systèmes Linux. Il est plus avancé fdisk et prend en charge une gamme plus large de formats de table de partition, notamment MBR et GPT. Voici quelques raisons de choisir **parted** :

Plus grande compatibilité : **parted** fonctionne avec une plus grande variété de tables de partition, ce qui le rend plus polyvalent pour la gestion des disques durs modernes.

Fonctionnalités avancées : **parted** offre des fonctionnalités plus avancées, telles que le redimensionnement des partitions sans perte de données.

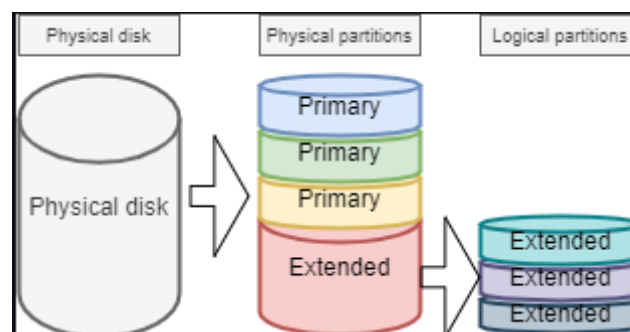
Partitions primaires



Les partitions primaires sont les partitions principales d'un disque. Elles peuvent héberger directement un système de fichiers, ce qui signifie que vous pouvez y installer un système d'exploitation ou les utiliser pour le stockage de données. Selon le schéma de partitionnement MBR, vous pouvez avoir jusqu'à quatre partitions primaires sur un disque.

La limitation à quatre partitions principales peut être restrictive, en particulier sur les disques durs de grande taille. C'est là que les partitions étendues entrent en jeu.

Partitions étendues



Une partition étendue est un type spécial de partition qui agit comme un conteneur pour des partitions supplémentaires appelées partitions logiques. Vous ne pouvez avoir qu'une seule partition étendue sur un disque, mais cette partition étendue peut être subdivisée en plusieurs partitions logiques.

Cette structure vous permet de contourner efficacement la limite de quatre partitions principales. Au lieu de créer plusieurs partitions principales, vous pouvez créer une seule partition principale, une partition étendue, puis autant de partitions logiques que nécessaire au sein de la partition étendue.

En résumé, les partitions primaires sont les principales divisions de votre disque dur et peuvent héberger directement un système de fichiers ou un système d'exploitation. D'autre part, une partition étendue agit comme un conteneur pour plusieurs partitions logiques, vous permettant de créer plus de quatre partitions sur un disque.

Il faut configurer correctement les partitions pour obtenir une structure similaire à celle ci-dessous, avec la prise en compte des partitions pour le bonus :

```
# 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
```

Lancer fdisk : `fdisk /dev/sda`

- "n" pour nouvelle partition

Créer une partitions primaire de 512M :

- Select : p (primary)
- Partition number : 1
- First sector : laisser vide
- Last sector : +512M


```
Born2BeRoot [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
[anaconda root@localhost ~]# fdisk /dev/sda

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xadfeaad8.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-62914559, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62914559, default 62914559): +512M

Created a new partition 1 of type 'Linux' and of size 512 MiB.
```

Ensuite, créer une partition étendue qui occupera tout l'espace restant.

- “n” pour nouvelle partition
- Select : e (extended)
- Partition number : laisser vide
- First sector : laisser vide
- Last sector : laisser vide

```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): e
Partition number (2-4, default 2):
First sector (1050624-62914559, default 1050624):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1050624-62914559, default 62914559):

Created a new partition 2 of type 'Extended' and of size 29.5 GiB.

Command (m for help):
```

Entrez ensuite *n* à nouveau et fdisk créera automatiquement une nouvelle partition logique sda5.

- First sector : laisser vide
- Last sector : laisser vide

```

Command (m for help): n
All space for primary partitions is in use.
Adding logical partition 5
First sector (1052672-62914559, default 1052672):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1052672-62914559, default 62914559):

Created a new partition 5 of type 'Linux' and of size 29.5 GiB.

Command (m for help): _

```

Appuyer sur *w* puis *enter* pour écrire les modifications.

```

Created a new partition 5 of type 'Linux' and of size 30.5 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[anaconda root@localhost ~]# _

```

lsblk dans le terminal affiche les partitions.

```

[anaconda root@localhost ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
loop0 7:0 0 819.2M 1 loop /run/rootfsbase
sda 8:0 0 31G 0 disk
├─sda1 8:1 0 512M 0 part
├─sda2 8:2 0 1K 0 part
└─sda5 8:5 0 30.5G 0 part
sr0 11:0 1 949M 0 rom /run/install/repo
zram0 252:0 0 1.9G 0 disk [SWAP]
[anaconda root@localhost ~]# _

```

Les partitions de sda5 doivent être chiffrées, pour cela j'ai utilisé LUKS.

Chiffrement

LUKS

LUKS (Linux Unified Key Setup) est une spécification de chiffrement de disque offrant un format indépendant de la plateforme, compatible et sécurisé. Il chiffre entièrement les blocs de données, idéal pour les supports mobiles et les disques d'ordinateurs portables. LUKS stocke les configurations dans l'en-tête de partition, sécurise les mots de passe par salage et hachage PBKDF2, et permet jusqu'à huit clés par partition pour un accès multi-utilisateurs. LUKS1, largement pris en charge et testé, est recommandé pour Born2beRoot pour sa compatibilité et stabilité, malgré les améliorations de LUKS2.

PBKDF2 (Password-Based Key Derivation Function 2) est une méthode qui renforce les mots de passe en ajoutant un "sel" unique et en répétant le hachage plusieurs fois. Cela

rend les mots de passe beaucoup plus difficiles à casser, car chaque tentative de déchiffrement demande plus de calculs.

Pour démarrer le chiffrement, entrez la commande suivante :

```
cryptsetup luksFormat --type luks1 /dev/sda5
```

Are you sure? Ecrire **“YES”**

Enter passphrase for /dev/sd5 : **Monmdpcrypt42**

```
[anaconda root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0 819.2M 1 loop /run/rootfsbase
sda          8:0      0   30G  0 disk
├─sda1       8:1      0   512M  0 part
├─sda2       8:2      0     1K  0 part
└─sda5       8:5      0  29.5G  0 part
sr0         11:0      1   949M  0 rom  /run/install/repo
zram0       252:0      0   1.9G  0 disk [SWAP]
[anaconda root@localhost ~]# cryptsetup luksFormat --type luks1 /dev/sda5

WARNING!
=====
This will overwrite data on /dev/sda5 irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/sda5:
Verify passphrase:
[anaconda root@localhost ~]#
```

déverrouille la partition chiffrée pour la rendre accessible avec la commande :

```
cryptsetup open /dev/sda5 sda5_crypt
```

Maintenant la partition sda_crypt est accessible :

```
[anaconda root@localhost ~]# cryptsetup open /dev/sda5 sda5_crypt
Enter passphrase for /dev/sda5:
[anaconda root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0 819.2M 1 loop /run/rootfsbase
sda          8:0      0   30G  0 disk
├─sda1       8:1      0   512M  0 part
├─sda2       8:2      0     1K  0 part
└─sda5       8:5      0  29.5G  0 part
    └─sda5_crypt 253:0      0  29.5G  0 crypt
sr0         11:0      1   949M  0 rom  /run/install/repo
zram0       252:0      0   1.9G  0 disk [SWAP]
[anaconda root@localhost ~]#
```

Création groupe de volume

Créons un groupe de volumes logiques dans notre conteneur `sda_crypt` et plusieurs partitions logiques à l'aide de LVM.

La gestion des volumes logiques (LVM ***logical volume management***) est une méthode avancée et flexible pour gérer les disques sous Linux.

Elle permet de regrouper et de manipuler les disques et autres périphériques de stockage en créant des "volumes logiques" abstraits au lieu de manipuler chaque disque individuellement.

Voici les principaux composants de LVM :

Volumes physiques (PV (Physical Volumes)) :

Ce sont les disques ou partitions physiques réels qui constituent la base du stockage dans LVM.

Groupes de volumes (VG (Volume Groups)) :

Ils sont constitués de plusieurs volumes physiques combinés. Un VG peut être vu comme un disque unique contenant l'espace de tous les PV inclus. Cela rend le redimensionnement simple, car des PV peuvent être ajoutés ou retirés dynamiquement du groupe.

Volumes logiques (LV (Logical Volumes)) :

Ces volumes, créés à partir d'un VG, sont similaires aux partitions de disque traditionnelles, mais offrent plus de flexibilité. Contrairement aux partitions classiques, les LV peuvent s'étendre sur plusieurs PV et être redimensionnés facilement. Ils agissent comme des partitions sur lesquelles on peut créer des systèmes de fichiers.

Avantages de LVM

LVM offre une grande flexibilité pour gérer l'espace de stockage. Par exemple, si un **LV** manque d'espace, il est possible d'ajouter un **PV** supplémentaire dans le **VG** et de l'étendre sans arrêter les services en cours. Les données peuvent également être migrées d'un **PV** à un autre de manière transparente.

Ainsi, une fois le conteneur déverrouillé (**sda5_crypt**), on peut y créer un **VG** et plusieurs **LV** pour organiser et gérer efficacement l'espace de stockage.

La commande suivante sert à initialiser la partition déverrouillée (`/dev/mapper/sda5_crypt`) en tant que volume physique (PV) dans le système LVM:

```
pvccreate /dev/mapper/sda5_crypt
```

Une fois initialisé pour être utilisé dans LVM, nous pouvons créer un groupe de volumes LVMGroup et y ajouter le volume précédemment créé avec la commande :

```
vgcreate LVMGroup /dev/mapper/sda5_crypt
```

```
[anaconda root@localhost ~]# vgcreate LVMGroup /dev/mapper/sda5_crypt
Physical volume "/dev/mapper/sda5_crypt" successfully created.
Creating devices file /etc/lvm/devices/system.devices
Volume group "LVMGroup" successfully created
```

La commande suivante permet de créer des volumes logiques dans un groupe de volume :

lvcreate

Voici la signification de chaque partie pour root :

lvcreate : Commande utilisée pour créer un volume logique (LV) dans LVM.

-L 10G : Indique la taille du volume logique à créer, ici 10 gigaoctets.

LVMGroup : Spécifie le nom du groupe de volumes (VG) dans lequel le volume logique sera créé. Dans cet exemple, le groupe s'appelle LVMGroup.

-n root : Attribue un nom au volume logique, ici root. Cela permettra de le monter facilement sous le nom /dev/LVMGroup/root.

```
lvcreate -L 10G LVMGroup -n root
```

```
lvcreate -L 2.3G LVMGroup -n swap
```

```
lvcreate -L 5G LVMGroup -n home
```

```
lvcreate -L 3G LVMGroup -n var
```

```
lvcreate -L 3G LVMGroup -n srv
```

```
lvcreate -L 3G LVMGroup -n tmp
```

```
lvcreate -L 4G LVMGroup -n var-log
```

en cas d'erreur de taille :

```
lvresize -L 2.3G /dev/LVMGroup/swap
```

Formater en ext4 le volume logique pour la partition principale :

```
mkfs.ext4 /dev/sda1
```

```
mkfs.ext4 /dev/LVMGroup/root
```

```
mkfs.ext4 /dev/LVMGroup/home
```

```
mkfs.ext4 /dev/LVMGroup/var
```

```
mkfs.ext4 /dev/LVMGroup/srv
```

```
mkfs.ext4 /dev/LVMGroup/tmp
```

```
mkfs.ext4 /dev/LVMGroup/var-log
```

mkfs.ext4 : Cette commande formate le volume logique en utilisant le système de fichiers ext4, adapté pour les partitions de type Linux.

mkswap : Formate le volume logique en tant que partition de swap, utilisée par le système pour étendre la mémoire virtuelle :

```
mkswap /dev/LVMGroup/swap
```

swapon : Active le volume logique formaté en swap, permettant au système de commencer à l'utiliser pour la gestion de la mémoire :

```
swapon /dev/LVMGroup/swap
```

Tester `lsblk` (ListBlock) pour voir si les volumes logiques apparaissent, sinon utiliser la commande `vgdisplay` ou `lvdisplay`.

Modifier le fichier de configuration avec `vi /etc/lvm/lvm.conf` et changer la valeur de `use_devicesfile` à 0.

Cette modification permet à LVM de rechercher tous les périphériques et de détecter les volumes physiques même s'ils sont chiffrés.

Le fichier `/etc/lvm/lvm.conf` est le fichier de configuration principal pour LVM sous Linux, influençant le comportement de toutes les commandes LVM.

Un paramètre important, `use_devicesfile`, est défini par défaut à 1, ce qui fait que LVM utilise le fichier `/etc/lvm/devices/system.devices` pour identifier les périphériques.

Cependant, dans une configuration LVM chiffrée, cela peut poser problème, car les volumes chiffrés risquent de ne pas être détectés.

En définissant `use_devicesfile` à 0, LVM analyse tous les périphériques pour détecter les volumes physiques, incluant les volumes chiffrés.

Ce problème est fréquent sur Rocky Linux et peut également se produire sur d'autres distributions supportant LVM2 et le chiffrement de disque.

Créer le répertoire `/var/log/` avec : `mkdir /var/log`

Ensuite, spécifiez les points de montage pour chaque volume logique créé dans le fichier `/etc/fstab` en ajoutant les entrée suivante : `vi /etc/fstab`

<code>/dev/mapper/LVMGroup-root</code>	<code>/</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-home</code>	<code>/home</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-var</code>	<code>/var</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-srv</code>	<code>/srv</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-tmp</code>	<code>/tmp</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-var--log</code>	<code>/var/log</code>	<code>ext4</code>	<code>defaults 0 0</code>
<code>/dev/mapper/LVMGroup-swap</code>	<code>none</code>	<code>swap sw</code>	<code>0 0</code>

Explication des options de chaque ligne :

Système de fichiers : le volume logique (`/dev/mapper/LVMGroup-root` pour root, etc.).

Point de montage : où le volume est monté (`/`, `/home`, etc.).

Type : le type de système de fichiers (`ext4` ou `swap` pour le swap).

Options : `defaults` pour des options de montage par défaut.

Dump et Pass : définis respectivement à 0 et 0 (pas de sauvegarde ni de vérification par `fsck`).

`none` : Il n'y a pas de point de montage pour swap, car ce n'est pas un système de fichiers.

`swap` : Ceci spécifie le type de l'entrée (dans ce cas, `swap`).

`sw` : Il s'agit de l'option de montage, indiquant qu'il s'agit d'une partition d'échange.

`0` : ce champ est destiné à l'utilitaire de vidage. Étant donné que swap n'est pas un système de fichiers, il est défini sur 0.

`0` : ce champ est destiné à `fsck` (File System Consistency Check) qui est un utilitaire sous Linux et Unix utilisé pour vérifier et réparer les systèmes de fichiers. Il détecte et corrige les erreurs de structure, d'allocation de fichiers, et d'autres incohérences dans les systèmes de fichiers pour assurer leur intégrité.

Encore une fois, comme swap n'est pas un système de fichiers, il est défini sur 0.

```
systemctl daemon-reload
```

Vérifier que les partitions sont correctement montées en exécutant la commande `mount -a`. Cette opération tentera de monter toutes les partitions répertoriées dans le fichier fstab.

Après avoir monté les partitions, vérifier et ajuster les permissions si nécessaire :

```
chown -R root:root /home
chown -R root:root /var
chown -R root:root /srv
chown -R root:root /tmp
chown -R root:root /var/log
```

L'installateur active normalement le swap automatiquement. Si besoin, vous pouvez le faire manuellement :

```
swapon -a
```

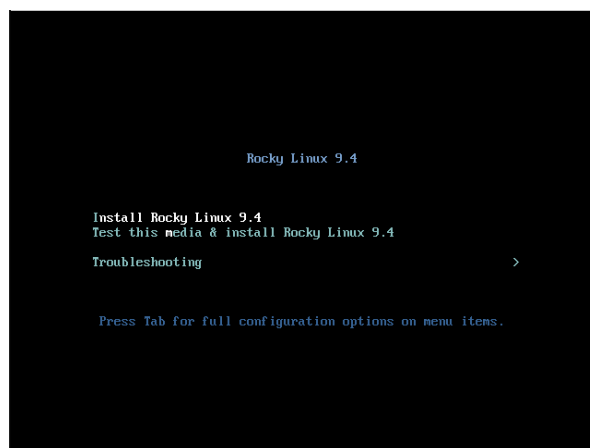
Vérifiez ensuite l'état de la mémoire et du swap :

```
free -h
```

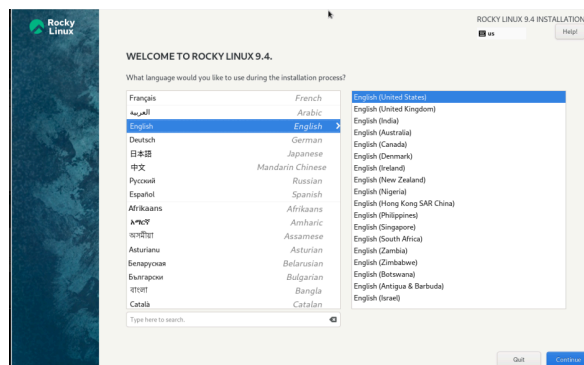
ensuite redémarrer la vm pour retourner à l'installateur.

3. Installation de Rocky

Une fois l'installateur relancé, il faut sélectionner Install Rocky Linux.

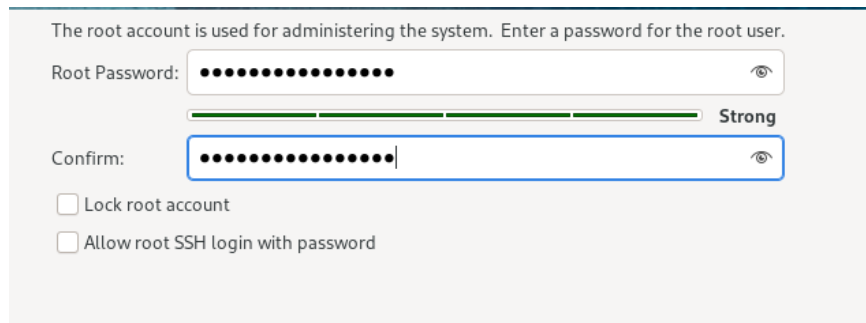


Il faut ensuite choisir la langue.



Il faut ensuite ajouter un mot de passe pour Root.

Root password : **Jesappeller00t42**



The root account is used for administering the system. Enter a password for the root user.

Root Password: [password field] [eye icon]

[progress bar] Strong

Confirm: [password field] [eye icon]

☐ Lock root account

☐ Allow root SSH login with password

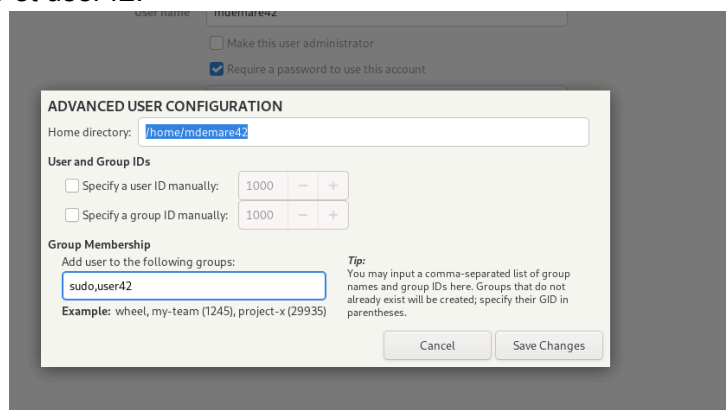
Créer un utilisateur.

Full Name : **Demare Mickael**

Nom d'utilisateur : **mdemare**

Mot de passe : **Monmdpuser42**

Une fois les informations utilisateur créées, cliquer sur advanced pour ajouter l'utilisateur aux groupes sudo et user42.



User name: mdemare42

☐ Make this user administrator

☒ Require a password to use this account

ADVANCED USER CONFIGURATION

Home directory: /home/mdemare42

User and Group IDs

☐ Specify a user ID manually: 1000 [minus] [plus]

☐ Specify a group ID manually: 1000 [minus] [plus]

Group Membership

Add user to the following groups:

[text box with 'sudo,user42']

Tip: You may input a comma-separated list of group names and group IDs here. Groups that do not already exist will be created; specify their GID in parentheses.

Example: wheel, my-team (1245), project-x (29935)

[Cancel] [Save Changes]

Dans Software Selection, sélectionner Minimal Install.

Désactiver KDUMP si pas besoin.

Dans Network & Host Name il faut ajouter Hostname :

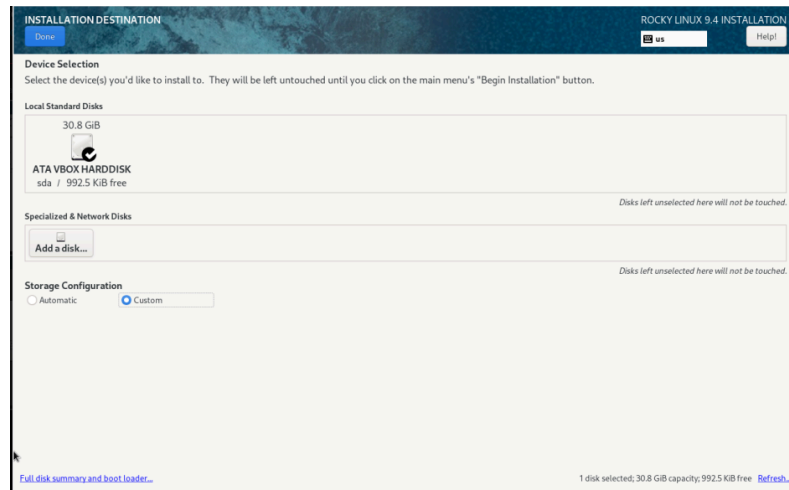


[+ -]

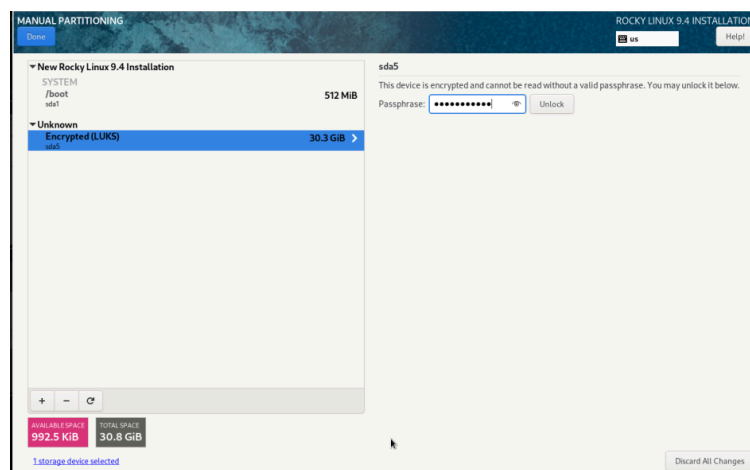
Host Name: mdemare [Apply]

Enfin, cliquer sur Installation Destination.

A Storage configuration il faut cocher Custom.



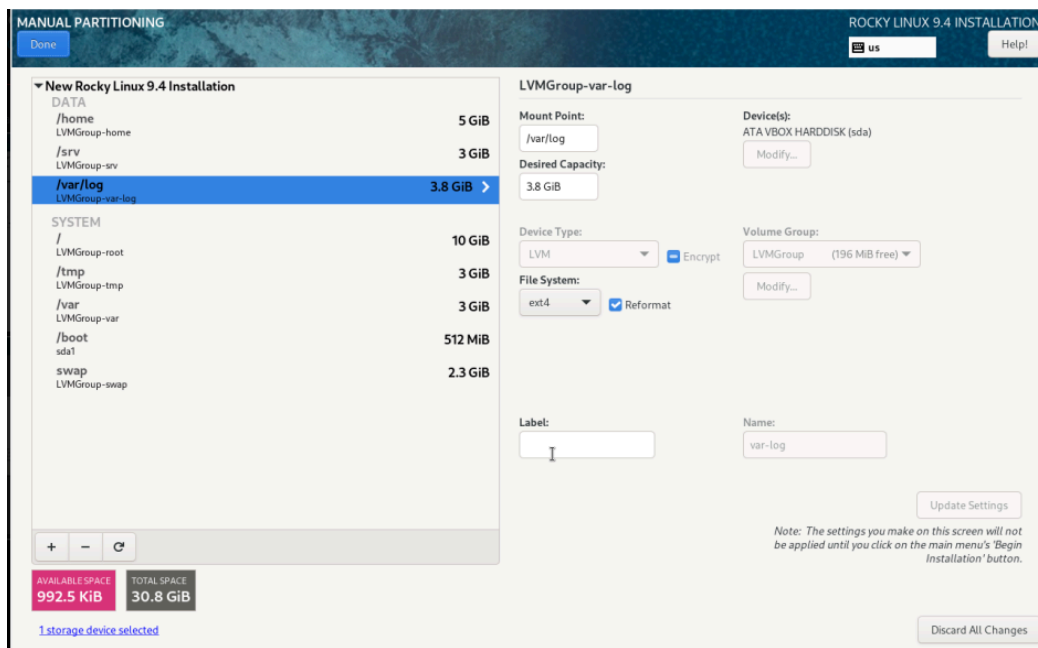
A la fenêtre suivante, il faut déverrouiller sda5 avec le mot de passe utilisé lors du cryptage luks.



Ensuite il faut assigner les points de montage et cocher les cases Reformat.

/dev/sda1 : Mount point : /boot
 Format : ext4
home : Mount point : /home
 Format : ext4
root : Mount point : /
 Format : ext4
srv : Mount point : /srv
 Format : ext4
swap : Mount point : none
 Format : swap
tmp : Mount point : /tmp
 Format : ext4
var : Mount point : /var

Format : ext4
var--log : Mount point : /var/log
Format : ext4



Ensuite appuyer sur Done.

Redémarrer la VM si pas automatique.

4. Configuration du serveur

S'ajouter a sudoers.

Se connecter en root avec la commande `su` et le mot de passe root, modifier le fichier `/etc/sudoers` grâce à l'outil `visudo`.

Rechercher la ligne `## Allow root to run any commandes anywhere` et ajouter :
`username ALL=(ALL) ALL ##` remplacer username par le nom utilisateur>

Comparaison entre SELinux et AppArmor

SELinux (Security-Enhanced Linux) renforce la sécurité du système en contrôlant finement les permissions des applications, empêchant ainsi les actions non autorisées même en cas de compromission. Développé initialement par la NSA et contribué en open source, SELinux permet un contrôle précis des interactions système grâce à des politiques détaillées, bien que cela rende parfois sa gestion complexe.

AppArmor, quant à lui, applique des restrictions à des applications spécifiques via des profils, offrant une approche simplifiée en comparaison avec SELinux, qui impose un contrôle à l'échelle du système. Bien que moins puissant et granulaire que SELinux, AppArmor est plus facile à configurer et entraîne moins d'effets secondaires indésirables.

Tous deux sont des systèmes de contrôle d'accès obligatoire (MAC), qui dépassent les permissions classiques utilisateur/groupe/autre (DAC) en restreignant l'accès aux ressources système de manière stricte.

Gestion de SSH dans SELinux

Par défaut, SELinux autorise le trafic SSH sur le port 22. Toutefois, si le port SSH est modifié, SELinux pourrait bloquer le démon SSH (sshd) sur le nouveau port. Pour ajuster cela, il est nécessaire de mettre à jour les politiques de SELinux pour autoriser le démon SSH à utiliser ce port personnalisé.

sshd (Secure Shell Daemon) est le service qui gère les connexions SSH (Secure Shell) sur un système. C'est lui qui écoute les demandes de connexion SSH entrantes et gère l'authentification des utilisateurs pour établir des sessions sécurisées.

Voici quelques points clés sur sshd :

- **Rôle** : sshd permet aux utilisateurs distants de se connecter de manière sécurisée à un serveur via le protocole SSH. Cela inclut la connexion à distance à une console ou la copie sécurisée de fichiers.
- **Configuration** : Le comportement de sshd est configuré dans le fichier `/etc/ssh/sshd_config`. Ce fichier permet de définir des paramètres tels que le

port d'écoute (par défaut 22), les méthodes d'authentification, les permissions utilisateur, et bien plus encore.

- **Lancement** : sshd s'exécute généralement comme un service en arrière-plan, démarré au démarrage du système et géré par systemd sur les distributions Linux modernes.

Installation d'un paquet

Pour installer un paquet dans Rocky, il faut utiliser dnf, contrairement à Debian qui utilise apt ou aptitude.

dnf (Dandified YUM) est le gestionnaire de paquets par défaut pour les distributions Linux basées sur RPM (Red Hat Package Manager), telles que Rocky Linux, Fedora, Red Hat Enterprise Linux (RHEL) et CentOS (à partir de la version 8). Il remplace l'ancien gestionnaire de paquets yum avec des améliorations en termes de performance, de résolution de dépendances, et de gestion des dépôts.

Ajuster SELinux pour SSH

Pour ajouter un nouveau port SSH à la politique SELinux, il faut installer le package policycoreutils-python-utils qui fournit la commande `semanage` qui fait partie de l'outil SELinux Management et est souvent utilisée pour configurer et gérer les politiques de SELinux.

```
sudo dnf install policycoreutils-python-utils
```

Il faut aussi installer selinux-policy-targeted qui fournit une politique SELinux ciblée pour des services critiques, en appliquant des règles de sécurité pour limiter l'accès aux ressources système de manière stricte.

```
sudo dnf install selinux-policy-targeted
```

selinux-policy-targeted est l'un des principaux modules de politique pour SELinux. Il applique un ensemble de règles de sécurité spécifiques à des services et applications ciblés tout en laissant le reste du système relativement libre. Cela permet de limiter l'impact de SELinux aux services les plus critiques pour la sécurité sans imposer des restrictions sur tout le système.

Changement du port SSH

On peut ensuite ajouter le nouveau port à la politique SELinux avec la commande `semanage` :

```
sudo semanage port -a -t ssh_port_t -p tcp 4242
```

Voici une description détaillée de chaque partie de cette commande :

`semanage` : C'est un outil de gestion SELinux qui permet d'ajuster les paramètres de sécurité sans avoir besoin de modifier directement les fichiers de politique de SELinux.

`port` : Indique que l'on veut gérer les paramètres de ports sous SELinux.

`-a` : Ajoute une nouvelle règle dans les politiques SELinux. Ici, cette option permet d'ajouter le port spécifié (4242) comme port autorisé pour un service particulier.

`-t ssh_port_t` : Spécifie le type de port SELinux.

`ssh_port_t` est le type de port associé au service SSH dans SELinux. En associant le port 4242 à ce type, on informe SELinux que ce port est désormais valide pour le service SSH, autorisant ainsi `sshd` à écouter sur ce port.

`-p tcp` : Spécifie le protocole réseau utilisé, ici TCP, qui est le protocole standard pour les connexions SSH.

`4242` : Indique le numéro de port personnalisé que l'on souhaite autoriser pour SSH.

La commande suivante permet de vérifier que le nouveau port est ajouté :

```
sudo semanage port -l | grep ssh
```

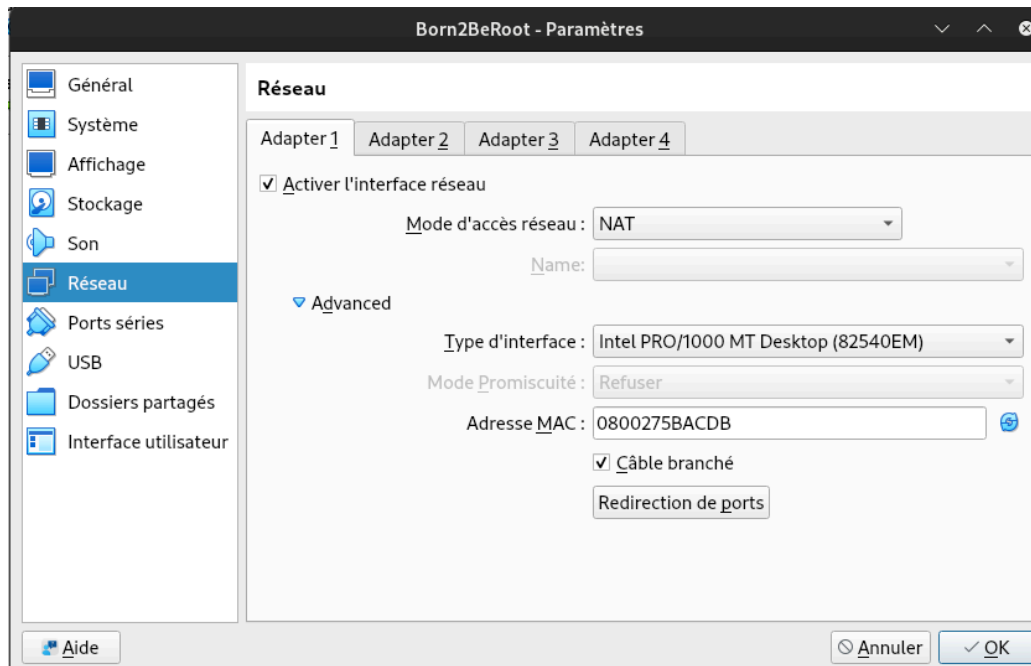
5. Secure Shell (SSH)

Secure Shell (SSH) est un protocole réseau cryptographique permettant des connexions sécurisées sur des réseaux non sécurisés. Il est principalement utilisé pour se connecter à distance à des machines et exécuter des commandes, mais il prend également en charge le tunneling, le transfert de ports, les connexions X11, ainsi que le transfert de fichiers via SCP et SFTP.

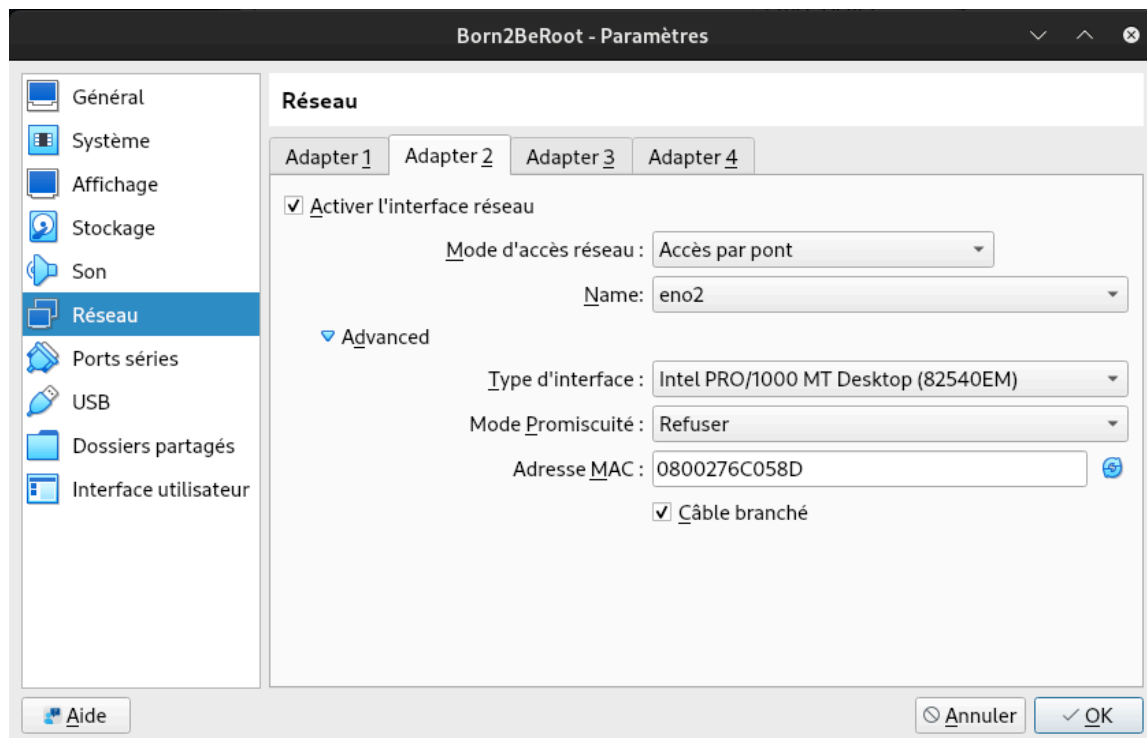
SSH utilise la cryptographie à clé publique pour authentifier les machines et protéger la confidentialité et l'intégrité des données échangées. Il a été conçu pour remplacer Telnet et d'autres protocoles non sécurisés qui exposaient les informations en texte clair, vulnérables à l'interception.

Configuration de la machine virtuelle pour SSH

Pour pouvoir utiliser le port 4242 il faut rediriger le port dans les paramètres de la Machine virtuelle, sous Réseau, Advanced, Redirection de ports :



Pour éviter les soucis liés au port 4242 déjà utilisé par l'ordinateur, utiliser Adapter 2 et le configurer en accès par pont :



La configuration SSH est contrôlée par le fichier `/etc/ssh/sshd_config` sur votre système. Dans ce fichier, vous pouvez personnaliser divers paramètres tels que le port SSH, les méthodes d'authentification et les options liées à la sécurité.

Conformément aux exigences de notre projet, nous devons modifier la configuration SSH pour la rendre plus sécurisée et répondre à des critères spécifiques. Voici un guide étape par étape sur la façon de procéder :

Générer une paire de clés SSH sur la machine virtuelle : exécutez-la simplement `ssh-keygen -t rsa` sur votre machine virtuelle.

Acceptez l'emplacement de fichier par défaut et fournissez éventuellement une phrase secrète pour plus de sécurité. mon identifiant est enregistré par défaut dans :

`/home/mdemare/.ssh/id_rsa`

et ma clé publique dans :

`/home/mdemare/.ssh/id_rsa.pub`

ma clé fingerprint est :

SHA256 : 15sCHnG6PCXeWrAaLECUAVJGxEEiIQ2LwcnwZvUPQk
mdemare@mdemare42

Changer le port par défaut .

Ouvrez le fichier de configuration SSH `/etc/ssh/sshd_config` en su si besoin avec un éditeur de texte, localisez la ligne `#Port 22`, supprimez le `#`, et changez le numéro de port en `4242`. Ainsi, la ligne devrait ressembler à ceci : `Port 4242`. Cela obligera SSH à écouter sur le port 4242 au lieu du port par défaut 22.

Rechercher la ligne `ListenAddress 0.0.0.0` supprimez le `#`, ceci permet de pouvoir se connecter à la machine sur n'importe quelle ip si la machine a plusieurs cartes réseaux.

```
# /etc/ssh/sshd_config.d/ which will be automatically included
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 4242
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
```

Désactiver la connexion root. Pour empêcher l'accès direct au compte root via SSH, recherchez la ligne `#PermitRootLogin`, supprimer le `#` et définissez sa valeur sur `'no'`. Cela devrait donc ressembler à ceci : `PermitRootLogin no`.

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

Authentification par mot de passe .

Vérifier que l'authentification par mot de passe est activée. Rechercher une ligne indiquant `PasswordAuthentication yes`. Si cette ligne est commentée, décommentez-la. Si vous ne la voyez pas, ajoutez-la au fichier.


```
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes
```

Redémarrer le service pour prendre en compte les changements :
`sudo systemctl restart sshd`

Configuration du pare-feu pour SSH

Le pare-feu protège votre système en contrôlant le trafic réseau entrant et sortant. Dans ce projet, nous utiliserons FirewallD.

firewall-cmd est l'interface en ligne de commande de FirewallD, un pare-feu dynamique qui utilise des zones réseau pour définir le niveau de confiance des connexions et des interfaces réseau.

FirewallD permet de configurer des règles de manière temporaire ou permanente afin d'autoriser ou de bloquer le trafic entrant sur les ports de votre serveur.

firewall-cmd fait partie de l'application firewalld, qui est installée par défaut sur certaines distributions Linux.

Voici quelques commandes courantes avec firewall-cmd :

`firewall-cmd --state` : Vérifie si le pare-feu est en cours d'exécution.

`firewall-cmd --reload` : Recharge la configuration sans interrompre les connexions existantes.

`firewall-cmd --get-default-zone` : Affiche la zone par défaut utilisée pour les connexions et les interfaces sans zone spécifique attribuée.

`firewall-cmd --get-active-zones` : Montre les zones actuellement actives, avec leurs interfaces réseau et sources.

`firewall-cmd --zone=public --list-all` : Affiche tous les paramètres de la zone spécifiée (public peut être remplacé par une autre zone).

Il faut configurer le pare-feu pour autoriser le trafic SSH via le port 4242.

Pour ouvrir le port avec firewallD il faut utiliser la commande suivante :

```
firewall-cmd --permanent --add-port=4242/tcp
firewall-cmd --reload
```

La suite de protocoles Internet, appelée TCP/IP, est basée sur le **protocole TCP** (Transmission Control Protocol) et **IP** (Internet Protocol). TCP est orienté connexion,

assurant une transmission fiable et ordonnée des données entre applications sur des réseaux IP, ce qui le rend essentiel pour des services comme le Web, la messagerie et le transfert de fichiers.

Parmi les autres protocoles clés de la suite :

- **UDP** (User Datagram Protocol) : protocole sans connexion, rapide mais sans garantie de livraison, utilisé dans le streaming et les appels VoIP.
- **ICMP** (Internet Control Message Protocol) : envoie des messages d'erreur et d'état entre périphériques réseau.
- **HTTP/HTTPS** : protocoles de transfert d'hypertexte, HTTPS incluant un chiffrement via SSL/TLS pour la sécurité.
- **FTP** : permet le transfert de fichiers entre client et serveur.
- **DNS** (Domain Name System) : système de nommage associant noms de domaine et adresses IP.

Le choix du protocole dépend des besoins de l'application : vitesse, fiabilité ou sécurité.

Maintenant que SSH est configuré il faut redémarrer la machine : `reboot`

Pour obtenir l'ip de la machine, utiliser `hostname -I`

`10.11.200.103`

Essayer de se connecter : `ssh mdemare@10.11.200.103 -p 4242`

```
Host key verification failed.
mdemare@c1r1p11 ~ % ssh mdemare@10.11.200.103 -p 4242
The authenticity of host '[10.11.200.103]:4242 ([10.11.200.103]:4242)' can't be
established.
ED25519 key fingerprint is SHA256:HgBmuyXD0uZKf4Ehf8vj0fTrZ/TabQoThmrjCJC7mTw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.11.200.103]:4242' (ED25519) to the list of known
hosts.
mdemare@10.11.200.103's password:
Last login: Thu Nov 14 17:22:35 2024
```

6. Configuration du HostName

Le nom d'hôte de votre machine agit comme une carte d'identité unique, la distinguant des autres appareils sur le même réseau. Dans certains contextes, comme dans cet exercice pédagogique, il peut être nécessaire de suivre une convention spécifique pour le nom d'hôte, afin de refléter un identifiant de connexion, un identifiant personnel ou un numéro de tâche.

Utilisation de `hostnamectl` pour Configurer le Nom d'Hôte

Sur Rocky Linux, l'outil `hostnamectl` permet de contrôler et de modifier le nom d'hôte de manière simple. Pour définir le nom d'hôte souhaité, vous pouvez utiliser la commande suivante :

`sudo hostnamectl set-hostname mdemare42`

Pour vérifier que le Hostname a changé :

hostnamectl

```
[root@mdemare42 mdemare]# hostnamectl
Static hostname: mdemare42
    Icon name: computer-vm
    Chassis: vm
    Machine ID: 530e2ad2fb164859b25a894567a82e5d
    Boot ID: 3eec9d12a13744e6a38a2d4293f04cb2
    Virtualization: oracle
Operating System: Rocky Linux 9.4 (Blue Onyx)
    CPE OS Name: cpe:/o:rocky:rocky:9::baseos
    Kernel: Linux 5.14.0-427.13.1.el9_4.x86_64
    Architecture: x86_64
Hardware Vendor: innotek GmbH
    Hardware Model: VirtualBox
Firmware Version: VirtualBox
```

Pour que le changement de nom d'hôte soit appliqué dans tout le système, il faut également mettre à jour le fichier **hosts**.

sudo vi /etc/hosts

Trouvez la ligne qui suit ce format : 127.0.1.1. Remplacez votre-ancien-nom-d'hôte par votre nouveau nom d'hôte, votrelogin42. Si cette ligne n'existe pas, ajoutez-la juste en dessous de la ligne 127.0.0.1 localhost.

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4 mdemare42
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6 mdemare42
```

Enregistrer les modifications et fermer le fichier.

executer la commande :

sudo hostnamectl set-hostname new_hostname

Enfin, pour nous assurer que nos modifications prennent effet, nous allons redémarrer le système avec la commande : **reboot**

7. Configuration de Politiques de Mot de Passe et de Droits Sudo Sécurisées sur Rocky Linux

Maintenir un environnement Linux sécurisé implique de définir des politiques de mot de passe et de sudo fortes. Ces politiques protègent non seulement votre système contre les accès non autorisés, mais garantissent également sa stabilité globale.

Pour définir des politiques de mot de passe fortes, vous devez généralement modifier les fichiers de configuration associés à PAM (Pluggable Authentication Modules) et login.defs. Voici comment procéder :

Ouvrez le fichier `/etc/security/pwquality.conf` et modifiez les valeurs suivantes :

Au moins 7 caractères différents de l'ancien mot de passe :

`difok = 7`

Longueur minimale de 10 caractères :

`minlen = 10`

Inclure au moins une majuscule, une minuscule et un chiffre :

`dcredit = -1 # Nombre minimum de chiffres`

`ucredit = -1 # Nombre minimum de majuscules`

`lcredit = -1 # Nombre minimum de minuscules`

Pas plus de 3 caractères identiques consécutifs :

`maxrepeat = 3`

Interdire les mots de passe contenant le nom d'utilisateur :

`usercheck = 1`

Applique des contrôles de qualité pw sur le mot de passe de l'utilisateur root.

`enforce_for_root`

ensuite modifier `/etc/login.defs` pour la gestion de l'expiration des mots de passe et des avertissement :

```
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_MIN_LEN     Minimum acceptable password length.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   30
PASS_MIN_DAYS   2
PASS_WARN_AGE   7
```

Expiration du mot de passe tous les 30 jours :

`PASS_MAX_DAYS 30`

Délai minimum de 2 jours avant de pouvoir changer le mot de passe :

`PASS_MIN_DAYS 2`

Avertissement 7 jours avant l'expiration :
`PASS_WARN_AGE 7`

Pour garantir des mots de passe complexes, mettez à jour la ligne `pam_pwquality.so` dans les fichiers `/etc/pam.d/system-auth` et `/etc/pam.d/password-auth`. Vous devez définir la longueur minimale à 10, exiger des lettres majuscules et des chiffres différents et limiter les caractères répétitifs. Voici un exemple de ligne :

```
password requisite pam_pwquality.so try_first_pass local_users_only retry=3  
authtok_type= minlen=10 ucredit=-1 lcredit=-1 dcredit=-1 difok=7 reject_username  
enforce_for_root
```

Voici une explication détaillée des options de cette ligne de configuration **PAM** (Pluggable Authentication Module) utilisée pour renforcer les règles de mot de passe.

`[success=1 default=ignore]` = si root alors sauter 1 règles

Décomposition et Explication

1. Type : `password`

Définit le type de module PAM. Ici, il s'agit de la gestion des mots de passe.

2. Contrôle : `requisite`

Indique que ce module doit réussir pour que le processus d'authentification continue. Si ce module échoue, le processus s'arrête immédiatement.

3. Module : `pam_pwquality.so`

Module utilisé pour appliquer des règles de qualité sur les mots de passe. C'est l'outil principal pour contrôler la complexité des mots de passe.

4. Options du Module

`try_first_pass`

Tente d'utiliser le mot de passe saisi précédemment (utile si un autre module PAM a déjà demandé un mot de passe). Si le mot de passe est incorrect, il sera demandé à nouveau.

`local_users_only`

Applique les restrictions uniquement aux utilisateurs locaux (pas aux utilisateurs définis dans des bases externes comme LDAP).

`retry=3`

Définit le nombre de tentatives autorisées pour entrer un mot de passe conforme aux règles avant que le processus ne soit annulé.

`authtok_type=`

Cette option est normalement utilisée pour spécifier le type d'authentification (comme `password`). Ici, elle est vide, ce qui indique que la valeur par défaut est utilisée.

`minlen=10`

Le mot de passe doit contenir au moins 10 caractères. Ce nombre inclut les majuscules, minuscules, chiffres et symboles.

`ucredit=-1`

Exige au moins une lettre majuscule dans le mot de passe.

Une valeur négative indique une obligation minimale.

`lcredit=-1`

Exige au moins une lettre minuscule dans le mot de passe.

`dcredit=-1`

Exige au moins un chiffre dans le mot de passe.

`difok=7`

Exige que le nouveau mot de passe contienne au moins 7 caractères différents par rapport à l'ancien mot de passe.

`reject_username`

Interdit que le mot de passe contienne le nom de l'utilisateur.

`enforce_for_root`

Applique également ces règles au compte root.

Par défaut, certaines règles ne s'appliquent pas au superutilisateur.

Il faut également modifier le fichier `/etc/sudoers` en utilisant `visudo` pour éviter les erreurs de syntaxe et ajouter ce qui suit.

Limiter les tentatives d'authentification à 3 essais :

```
Defaults    passwd_tries=3
```

Afficher un message personnalisé en cas d'erreur :

```
Defaults    badpass_message="Bad password, it's sad!"
```

Pour enregistrer les entrées et les sorties sudo, ajoutez les lignes suivantes :

```
Defaults    logfile="/var/log/sudo/sudo_config"
Defaults    log_input
Defaults    log_output
Defaults    iolog_dir="/var/log/sudo/
```

Forcer le mode TTY pour les sessions sudo :

```
Defaults    requiretty
```

Restreindre les chemins utilisables par sudo :

```
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
```

```
# Defaults specification
Defaults    passwd_tries=3
Defaults    badpass_message="Bad password, it's sad!"
Defaults    log_input
Defaults    log_output
Defaults    iolog_dir="/var/log/sudo/
Defaults    requiretty
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
#
# Refuse to run if unable to disable echo on the tty.
#
```

Après avoir effectué ces modifications, utilisez la commande `passwd` pour modifier les mots de passe de tous les utilisateurs (`sudo passwd username`), y compris root, afin de respecter les nouvelles politiques.

N'oubliez pas de modifier soigneusement ces fichiers, car des modifications incorrectes peuvent entraîner une instabilité du système. Sauvegardez toujours ces fichiers avant d'effectuer des modifications. Utilisez la commande `visudo` pour modifier en toute sécurité le fichier `sudoers`, car elle vérifie la syntaxe avant de l'enregistrer.

SI LE MESSAGE : `"/var/log/sudo exists but is not a directory"` apparaît, essayer de le supprimer et de le recréer avec :

```
sudo rm -f /var/log/sudo
sudo mkdir /var/log/sudo
sudo chmod 0700 /var/log/sudo
sudo chown root:root /var/log/sudo
```

Pour appliquer ces règles à un utilisateur, utilisez la commande chage : `sudo chage -M 30 -m 2 -W 7 [nom_utilisateur]`

Exécutez la commande suivante pour vérifier que les paramètres sont appliqués : `sudo chage -l [nom_utilisateur]`

C'est quoi TTY?

TTY signifie Teletype, un terme historique qui fait référence aux terminaux physiques utilisés pour interagir avec un ordinateur. Aujourd'hui, dans un contexte moderne, TTY désigne un terminal virtuel ou une session interactive avec le système.

Dans le cadre de sudo, l'activation du mode TTY signifie que l'utilisateur doit être connecté à une session interactive pour exécuter des commandes sudo. Cela ajoute une couche de sécurité en empêchant des processus ou des scripts non interactifs de tenter d'exécuter des commandes avec des privilèges administratifs.

Pourquoi activer le Mode TTY ?

Sécurité accrue : En activant le mode TTY pour sudo, vous exigez qu'un utilisateur soit physiquement (ou logiquement) présent dans une session interactive pour exécuter des commandes administratives. Cela empêche :

Les scripts ou programmes automatisés de tenter des commandes sudo.

Les attaques distantes ou non autorisées qui exploiteraient des services automatisés.

Traçabilité : Lorsque le mode TTY est activé, chaque commande sudo est liée à une session interactive spécifique, ce qui facilite l'audit des actions administratives.

8. Automatiser la Surveillance Système avec Bash et Cron

La surveillance du système est essentielle pour un administrateur, permettant de suivre des paramètres comme l'utilisation du processeur, de la mémoire ou du disque. En utilisant Bash, un interpréteur de commandes courant sur Linux, vous pouvez écrire des scripts pour automatiser ces vérifications. Ces scripts combinent des commandes classiques et des structures de contrôle spécifiques à Bash. Leur exécution peut être planifiée régulièrement à l'aide de Cron, garantissant une surveillance continue et automatisée.

Création de monitoring.sh

Téléchargez le fichier monitoring.sh avec la commande suivante :

```
curl -o /root/monitoring.sh  
https://raw.githubusercontent.com/42mdemare/Born2BeRoot/main/monitoring.sh
```

Le script que nous avons créé pour la surveillance du système, monitoring.sh, collecte diverses informations système importantes. Voici comment cela fonctionne :

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

Cette ligne, appelée shebang, indique au système que ce script doit être exécuté à l'aide de l'interpréteur de commandes bash.

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

La commande `export PATH` garantit que le script peut trouver toutes les commandes nécessaires, même dans des environnements restreints ou minimaux. En exportant explicitement le PATH, elle assure la compatibilité et la sécurité en limitant l'exécution des commandes aux répertoires standards.

```
while true; do
```

C'est le début d'une boucle qui fonctionnera indéfiniment car la condition de la boucle (vrai) est toujours remplie.

```
ARCH=$(uname -m)  
KERNEL=$(uname -r)
```

Ces lignes utilisent la commande `uname` pour capturer l'architecture du système (par exemple, x86_64, i386, armv7l, etc.) et la version du noyau.

```
DISTRIBUTION=$(awk '{print $1}' /etc/redhat-release)
```

`cat /etc/redhat-release` Cette commande imprime le contenu du fichier `/etc/redhat-release` sur la sortie standard. | Il s'agit d'un opérateur de canal qui prend la sortie de la commande précédente (`cat /etc/redhat-release`) et la transmet comme entrée à la commande suivante (`awk '{print $1}'`). `awk '{print $1}'` Cette commande `awk` est utilisée pour manipuler des fichiers texte. Ici, il lui est demandé d'imprimer le premier champ (séparé par des espaces) de chaque ligne. La sortie de cette commande sera le premier mot de la première ligne du fichier `/etc/redhat-release`, qui indique généralement le nom de la distribution.


```
VERSION=$(awk '{print $4}' /etc/redhat-release)
```

Similairement à la ligne "DISTRIBUTION", cette commande récupère le contenu du fichier `/etc/redhat-release`. `awk '{print $4}'` Ici, `awk` est à nouveau utilisé pour imprimer un champ spécifique de chaque ligne. Dans ce cas, il imprime le quatrième champ de chaque ligne. Cela correspond généralement aux informations de version dans le fichier `/etc/redhat-release`.

```
CPU_PHYSICAL=$(lscpu | awk '/Socket\s\):/ {print $2}')  
VCPU=$(lscpu | awk '/^CPU\s\):/ {print $2}')
```

Ici, la commande `lscpu` est utilisée pour capturer le nombre de sockets CPU physiques et le nombre total de CPU virtuels (`VCPU`).

```
RAM_USAGE=$(free -m | awk '/Mem:/ {printf "%.0f/%.0fMB (%.2f%%)", $3, $2, $3*100/$2}')
```

Cette ligne utilise la commande `free` pour récupérer des informations sur l'utilisation de la RAM du système en Mo, y compris le pourcentage de RAM utilisé. `free` fournit la mémoire totale et utilisée en Mo, ainsi que le pourcentage d'utilisation.

```
DISK_USAGE=$(df -BG --output=size,used,pcent / | awk 'NR==2 {printf "%s/%s (%s)\n", $2, $1, $3}')
```

Ici, `df` est utilisé pour afficher l'utilisation du disque du répertoire racine en Go, y compris le pourcentage de disque utilisé.

```
CPU_LOAD=$(top -b -n1 | grep "Cpu(s)" | awk '{printf "%.1f%%", $2+$4}')
```

Cette ligne utilise la commande `top` pour calculer la charge du processeur, qui est le pourcentage de temps pendant lequel le processeur fonctionne.

```
LAST_BOOT=$(who -b | awk '{print $3, $4}')
```

La commande `who` avec l'option `-b` est utilisée ici pour récupérer la date et l'heure du dernier démarrage du système.

```
LVM_STATUS=$(lsblk | grep -q "lvm" && echo "yes" || echo "no")
```

Cette ligne vérifie si des volumes Logical Volume Manager (LVM) actifs existent.

```
TCP_CONNECTIONS=$(ss -tan | grep ESTAB | wc -l)
```

La commande `ss` est utilisée pour compter le nombre de connexions TCP actives.

```
USER_LOG=$(who | wc -l)
```

Cette ligne utilise la commande `users` pour compter le nombre d'utilisateurs actuellement connectés au système.

```
IP=$(hostname -I | awk '{print $1}')
```

La commande `ip` est utilisée ici pour afficher l'adresse IPv4 et

```
MAC=$(ip link show | awk '/ether/ {print $2}')
```

l'adresse MAC de l'interface réseau active.

```
SUDO_CMDS=$(journalctl _COMM=sudo | grep COMMAND | wc -l)
```

Cette commande compte combien de fois des commandes ont été exécutées avec sudo sur le système et enregistre ce nombre dans la variable SUDO_CMDS.

Les commandes wall et journalctl nécessitent des privilèges root pour être utilisées.

```
echo "$MESSAGE" | wall 2>/dev/null
```

Les informations collectées sont formatées dans un message clair et diffusées à tous les terminaux connectés avec wall.

```
2 > /dev/null
```

Redirige la sortie standard (stdout) vers `/dev/null`, un fichier spécial qui supprime tout contenu envoyé.

Toutes les 10 minutes selon la configuration cron, le script exécute ces commandes, regroupe les résultats dans un message bien formaté, puis diffuse ce message à tous les terminaux à l'aide de la commande wall.

Ce script collecte et affiche une multitude d'informations sur le système, notamment l'architecture, la version du noyau, l'utilisation du processeur, de la RAM et du disque, l'heure du dernier démarrage, l'état LVM, les connexions TCP, les utilisateurs connectés, les détails du réseau et les commandes sudo exécutées. Il utilise divers utilitaires Linux courants pour collecter ces informations.

Pour rendre le script exécutable, enregistrez-le dans un fichier, disons « monitoring.sh », puis modifiez ses autorisations à l'aide de la commande chmod comme suit :

```
sudo chmod +x /root/monitoring.sh
```

```

#####
##                                     ##
##                               SYSTEM MONITORING REPORT                               ##
##                                     ##
#####
##                                     ##
## [SYSTEM]                                     ##
## Distribution : Rocky                                     ##
## Kernel Version : 5.14.0-427.13.1.el9_4.x86_64         ##
## System Version : 9.4                                   ##
## Hardware Architecture : x86_64                         ##
##                                     ##
## [PROCESSOR]                                     ##
## Physical CPUs : 1                                       ##
## Virtual CPUs (Threads) : 4                             ##
## CPU Load : 0.0%                                         ##
##                                     ##
## [MEMORY AND STORAGE]                             ##
## RAM Usage : 294/1966MB (14.95%)                       ##
## Disk Usage : 2G/10G (12%)                             ##
##                                     ##
## [BOOT AND STATUS]                                 ##
## Last Boot : 2024-11-16 14:30                          ##
## LVM Active : yes                                       ##
##                                     ##
## [NETWORK]                                         ##
## IPv4 Address : 10.11.200.103                           ##
## MAC Address : 08:00:27:6c:05:8d                       ##
##                                     ##
## [SECURITY]                                         ##
## Sudo Commands Executed : 0                             ##
##                                     ##
## [CONNECTIONS]                                     ##
## Active TCP Connections : 0                             ##
## Logged In Users : 2                                    ##
##                                     ##
#####

```

N'oubliez pas que vous devez disposer des autorisations appropriées pour exécuter ces scripts et planifier des tâches cron.

Et voilà ! Vous avez créé un script qui surveille les paramètres clés du système et qui le configure pour qu'il s'exécute automatiquement. Non seulement cela vous épargnera beaucoup de travail manuel, mais cela vous permettra également d'être constamment informé de l'état de votre système.

N'oubliez pas de tester votre script de manière approfondie pour vous assurer qu'il fonctionne comme prévu. Lorsque vous êtes prêt, vous pouvez le déployer sur votre système de production en toute confiance.

Différence entre **CPU physique** et **vCPU** (processeur virtuel)

La différence entre **CPU physique** et **vCPU** (processeur virtuel) réside dans la structure matérielle et la manière dont les ressources de traitement sont allouées et utilisées. Voici une explication détaillée :

1. CPU Physique

Définition : Un **CPU physique** représente un processeur matériel réel installé sur la carte mère. Chaque CPU physique peut avoir plusieurs cœurs.

Mesure avec `lscpu` : Le nombre de CPU physiques est généralement déterminé par le nombre de sockets sur la carte mère.

- Une **socket** est un emplacement physique pour un processeur.
- Exemple : Si votre carte mère à deux sockets et que chaque socket contient un processeur, vous avez **2 CPU physiques**.

2. vCPU (Processeur Virtuel)

Définition : Un **vCPU** est une unité logique de traitement créée par le système pour représenter un thread d'exécution.

- Un cœur physique peut gérer plusieurs vCPU grâce à une technologie appelée **hyperthreading**.
- Exemple : Si un cœur physique peut exécuter 2 threads simultanément, alors 1 cœur physique = 2 vCPU.

Mesure avec `lscpu` : Le nombre total de vCPU correspond au nombre total de **threads disponibles**.

- Cela inclut tous les cœurs physiques et les threads supplémentaires ajoutés par l'hyperthreading.

Exemple Pratique

Configuration matérielle :

- Une machine a **2 CPU physiques** (2 sockets).
- Chaque CPU a **4 cœurs physiques**.
- L'hyperthreading est activé, avec **2 threads par cœur**.

Résultats :

- **CPU physiques** = 2
- **Cœurs physiques** = 8 (2 CPU x 4 cœurs)
- **vCPU (threads)** = 16 (8 cœurs x 2 threads)

Pourquoi cette différence est importante ?

Performance :

Les vCPU augmentent la capacité de traitement, mais ils partagent les ressources physiques (cœurs et caches). Leur performance peut être inférieure à un cœur physique en charge lourde.

Virtualisation :

Dans les environnements cloud ou virtualisés, les **vCPU** sont attribués aux machines virtuelles (VM), mais plusieurs vCPU peuvent être associés à un seul cœur physique, selon la charge.

Automatiser les tâches avec Cron

Cron est un utilitaire de planification de tâches basé sur le temps dans les systèmes d'exploitation de type Unix. Les utilisateurs peuvent planifier des tâches (commandes ou scripts) à exécuter à des heures spécifiques et à des jours précis.

Dans notre cas, nous souhaitons que `monitoring.sh` s'exécute au démarrage du système, puis toutes les 10 minutes. Le script lui-même contient une boucle infinie qui garantit que le script continue à s'exécuter une fois démarré, avec une pause de 600 secondes (10 minutes) entre les itérations.

Bien qu'il existe de nombreuses façons d'exécuter un script au démarrage, une méthode courante consiste à utiliser le mot-clé spécial `@reboot` de cron, qui exécute une tâche une fois au démarrage.

Pour ajouter une tâche cron pour démarrer au lancement et toutes les 10 minutes, il faut ouvrir Crontab.

Crontab est un outil qui permet de lancer des applications de façon régulière, pratique pour un serveur pour y lancer des scripts de sauvegardes, etc.

Pour l'ouvrir, il faut exécuter la commande suivante :

```
sudo crontab -e ou vi /etc/crontab
```

Ensuite ajouter les lignes suivante :

```
*/10 * * * * root bash /root/monitoring.sh >/dev/null 2>&1
```

```
*/10 * * * * Exécute le script toutes les 10 minutes.
```

Explications des champs :

Premier champ (*) : Chaque minute (0–59).

Deuxième champ (*) : Chaque heure (0–23).

Troisième champ (*) : Chaque jour du mois (1–31).

Quatrième champ (*) : Chaque mois (1–12).

Cinquième champ (*) : Chaque jour de la semaine (0–7, où 0 et 7 représentent le dimanche).

`*/` : tout les

`root` : Exécute la commande en tant qu'utilisateur root.

`bash` /root/monitoring.sh : Exécute le script avec bash.

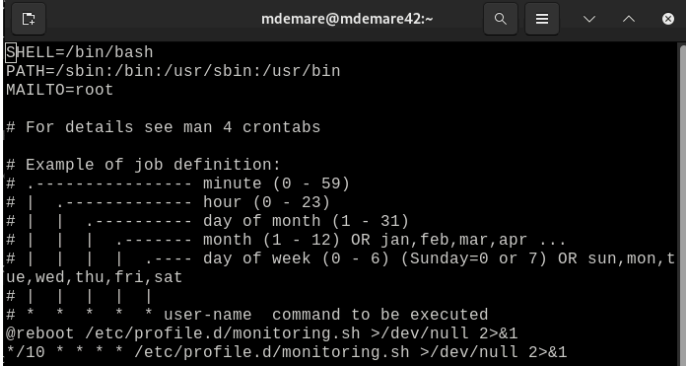
`2 > /dev/null` :

Redirige la sortie standard (stdout) vers `/dev/null`, un fichier spécial qui supprime tout contenu envoyé.

Cela évite que les messages réguliers du script apparaissent dans les journaux de cron ou sur le terminal.

2>&1 :

Redirige les erreurs standard (stderr) vers le même endroit que la sortie standard (stdout). En combinaison avec `> /dev/null`, cela garantit que ni les sorties normales ni les messages d'erreur ne s'affichent. >/dev/null 2>&1

A terminal window titled 'mdemare@mdemare42:~' showing the contents of a crontab file. The text includes environment variables (SHELL, PATH, MAILTO), a reference to man 4 crontabs, an example of job definition with fields for minute, hour, day of month, month, and day of week, and two cron jobs: one to reboot at every reboot and another to run a monitoring script every 10 minutes. Both jobs are configured to redirect output to /dev/null (2>&1).

```
mdemare@mdemare42:~  
SHELL=/bin/bash  
PATH=/sbin:/bin:/usr/sbin:/usr/bin  
MAILTO=root  
  
# For details see man 4 crontabs  
  
# Example of job definition:  
# .----- minute (0 - 59)  
# | .----- hour (0 - 23)  
# | | .----- day of month (1 - 31)  
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...  
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat  
# | | | | | * * * * * user-name command to be executed  
@reboot /etc/profile.d/monitoring.sh >/dev/null 2>&1  
*/10 * * * * /etc/profile.d/monitoring.sh >/dev/null 2>&1
```

Pour vérifier que le script s'exécute toutes les 10 minutes, consulter les logs de cron pour voir si la tâche est bien exécutée avec :

sudo journalctl -u crond

Redémarrer le serveur pour vérifier que le script est bien exécuté au démarrage :

sudo reboot

Comment vérifier ces paramètres ?

Pour vérifier les paramètres de la machine, on peut utiliser les commandes suivantes :

head -n 2 /etc/os-release
sestatus
ss -tunlp
sudo firewall-cmd --list-service
sudo firewall-cmd --list-port
sudo firewall-cmd --state

```
[mdemare@mdemare42 ~]$ head -n 2 /etc/os-release
sestatus
ss -tunlp
sudo firewall-cmd --list-service
sudo firewall-cmd --list-port
sudo firewall-cmd --state
NAME="Rocky Linux"
VERSION="9.4 (Blue Onyx)"
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:        enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.1:323 0.0.0.0:*
udp UNCONN 0 0 [::1]:323 [::]:*
tcp LISTEN 0 128 0.0.0.0:4242 0.0.0.0:*
tcp LISTEN 0 128 [::]:4242 [::]:*
[sudo] password for mdemare:
cockpit dhcpv6-client ssh
4242/tcp
running
[mdemare@mdemare42 ~]$
```

Partie Bonus

Configuration de WordPress avec Lighttpd, MariaDB et PHP

Maintenant c'est le moment de voir comment configurer un site WordPress fonctionnel en utilisant :

Lighttpd : Serveur Web léger.

MariaDB : Base de données robuste.

PHP : Langage de script côté serveur pour le contenu dynamique.

Un service supplémentaire sera également ajouté pour améliorer les performances ou la sécurité.

Lighttpd : un serveur Web léger

Description :

Serveur Web open source, optimisé pour les environnements à haute performance.

Avantages : faible consommation de mémoire, faible charge CPU, rapidité.

Installation :

Installer Lighttpd via le gestionnaire de paquets du système.

Configurez-le pour servir WordPress.

MariaDB : une base de données robuste

Description :

Version communautaire et commerciale de MySQL.

Hautement compatible avec MySQL, idéale pour WordPress.

Installation :

Installer MariaDB via le gestionnaire de paquets.

Créez une base de données dédiée et un utilisateur pour WordPress.

PHP : traitement du contenu dynamique

Description :

Langage de script côté serveur utilisé par WordPress.

Nécessaire pour le traitement dynamique des pages Web.

Installation :

Installer PHP via le gestionnaire de paquets.

Activez le module PHP FastCGI pour l'intégration avec Lighttpd.

Pour commencer il faut ouvrir le port HTTPS sur la VM depuis les paramètres.

Configurer le pare-feu

Autoriser le trafic HTTP et HTTPS :

```
sudo firewall-cmd --add-service=http --permanent  
sudo firewall-cmd --add-service=https --permanent
```

Autoriser le trafic MySQL/MariaDB (en supposant le port par défaut 3306) :

```
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
```

Autoriser le trafic PHP FastCGI (en supposant le port par défaut 9000 pour PHP-FPM) :

```
sudo firewall-cmd --zone=public --add-port=9000/tcp --permanent
```

redémarrez les paramètres du pare-feu en exécutant la commande `sudo firewall-cmd --reload`

Mettre à jour et mettre à niveau le système :

```
sudo dnf update -y  
sudo dnf upgrade -y
```

Vous devez installer le référentiel EPEL (Extra Package for Enterprise Linux) sur votre serveur. Il s'agit d'un référentiel gratuit qui vous permet de connecter de nombreux autres packages logiciels open source. Utilisez la commande ci-dessous pour installer EPEL.

```
sudo dnf install epel-release
```

Appuyez sur y pour accepter la confirmation d'installation et appuyez sur la touche Entrée.

Installez les packages requis :

```
sudo dnf install -y lighttpd lighttpd-fastcgi mariadb mariadb-server php php-mysqlnd  
php-fpm php-gd php-xml php-mbstring wget unzip
```

Démarrer et activer les services :

```
sudo systemctl start lighttpd  
sudo systemctl enable lighttpd  
sudo systemctl start mariadb  
sudo systemctl enable mariadb  
sudo systemctl start php-fpm  
sudo systemctl enable php-fpm
```

Configurer lighttpd

Modifiez le fichier de configuration lighttpd pour inclure la configuration fastcgi :

```
sudo vi /etc/lighttpd/lighttpd.conf
```

désactiver IPv6 :

```
server.use-ipv6 = "disable"
```

Ajoutez la ligne suivante à la fin du fichier :

```
include "conf.d/fastcgi.conf"
```

modifier la ligne suivant :

```
server.document-root = server_root + "/lighttpd"
```

par

```
server.document-root = "/var/www/html"
```

Et assurez-vous que la configuration du module FastCGI dans le fichier `/etc/lighttpd/conf.d/fastcgi.conf` est correcte. Si vous utilisez un socket TCP, cela devrait ressembler à ceci :

```
fastcgi.server += (  
    ".php" => (  
        "localhost" => (  
            "socket" => "/var/run/php-fpm/www.sock",  
            "broken-scriptfilename" => "enable"  
        )  
    )  
)
```

)

Voici ce que chaque ligne signifie :

`fastcgi.server` : Active le support FastCGI pour Lighttpd.

`.php` : Indique que cette configuration s'applique aux fichiers avec l'extension `.php`.

`localhost` : Nom du backend (arbitraire, uniquement interne à Lighttpd).

`"socket" => "/var/run/php-fpm/www.sock"` :

Définit le chemin du socket Unix que PHP-FPM utilise pour communiquer avec Lighttpd.

Ce chemin est configuré dans le fichier `/etc/php-fpm.d/www.conf`.

`"broken-scriptfilename" => "enable"` :

Active une correction nécessaire pour certaines incompatibilités entre PHP-FPM et Lighttpd.

Enregistrez et fermez le fichier.

Configurer PHP-FPM

Modifier le fichier de configuration PHP-FPM :

```
sudo vi /etc/php-fpm.d/www.conf
```

```
listen = /var/run/php-fpm/www.sock  
  
; Set listen(2) backlog.  
; Default Value: 511  
;listen.backlog = 511  
  
; Set permissions for unix socket, if one is used. In Linux, read/write  
; permissions must be set in order to allow connections from a web server.  
; Default Values: user and group are set as the running user  
; mode is set to 0660  
listen.owner = lighttpd  
listen.group = lighttpd  
;listen.mode = 0660  
  
; When POSIX Access Control Lists are supported you can set them using  
; these options, value is a comma separated list of user/group names.  
; When set, listen.owner and listen.group are ignored  
listen.acl_users = apache,nginx  
listen.acl_groups = lighttpd  
  
; List of addresses (IPv4/IPv6) of FastCGI clients which are allowed to connect  
; Equivalent to the FCGI_WEB_SERVER_ADDRS environment variable in the original  
; PHP FCGI (5.2.2+). Makes sense only with a tcp listening socket. Each address
```

Utilisez la commande `chmod` pour définir les autorisations appropriées pour le fichier de socket :

```
sudo chmod 660 /var/run/php-fpm/www.sock
```

Après avoir effectué les modifications, redémarrez le service PHP-FPM pour appliquer les modifications :

```
sudo systemctl restart php-fpm
```

Voici les droits qu'un fichier socket devrait avoir :

```
[mdemare@mdemare42 ~]$ sudo systemctl restart php-fpm  
[mdemare@mdemare42 ~]$ ls -l /var/run/php-fpm/www.sock  
srw-rw----+ 1 root root 0 Nov 16 17:04 /var/run/php-fpm/www.sock  
[mdemare@mdemare42 ~]$
```

Pour vérifier cela, exécuter la commande suivante :

```
ls -l /var/run/php-fpm/www.sock
```

S'assurer que le propriétaire et le groupe sont définis sur `lighttpd`.

Redémarrer les services :

```
sudo systemctl restart lighttpd
```

Sécuriser l'installation de MariaDB

`sudo mysql_secure_installation`

Suivre les instructions pour définir un mot de passe root et supprimer les utilisateurs anonymes, interdire la connexion root à distance et supprimer la base de données de test.

```
[mdemare@mdemare42 ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n]
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

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]
... Success!

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]
... Success!

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]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

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

Reload privilege tables now? [Y/n]
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[mdemare@mdemare42 ~]$
```

Créez une base de données et un utilisateur WordPress'

Connectez-vous à MariaDB :

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

Créer une nouvelle base de données, un nouvel utilisateur et accordez des autorisations :

```
CREATE DATABASE wordpress;  
CREATE USER 'kalicem'@'localhost' IDENTIFIED BY 'Monmdpwordpress42';  
GRANT ALL PRIVILEGES ON wordpress.* TO 'kalicem'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

```
sudo mysql -u root -p USE wordpress; SELECT option_name, option_value FROM  
wp_options WHERE option_name IN ('siteurl', 'home');
```

```
[mdemare@mdemare42 ~]$ sudo mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 14  
Server version: 10.5.22-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> CREATE DATABASE wordpress;  
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists  
MariaDB [(none)]> CREATE USER 'kalicem'@'localhost' IDENTIFIED BY 'Monmdpwordpres  
Query OK, 0 rows affected (0.002 sec)  
  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'kalicem'@'localhost';  
Query OK, 0 rows affected (0.002 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.000 sec)  
  
MariaDB [(none)]> EXIT;  
Bye  
[mdemare@mdemare42 ~]$
```

Télécharger tar avec dnf :

```
sudo dnf install tar
```

Téléchargez et installez WordPress :

```
wget http://wordpress.org/latest.tar.gz  
tar -xzf latest.tar.gz  
sudo mv wordpress/* /var/www/html/  
rm -rf latest.tar.gz wordpress/
```

Définissez la propriété et les autorisations appropriées :

```
sudo chown -R lighttpd:lighttpd /var/www/html  
sudo find /var/www/html/ -type d -exec chmod 755 {} \;
```

```
sudo find /var/www/html/ -type f -exec chmod 644 {} \;
```

Installation WordPress

Copiez le fichier de configuration d'exemple :

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

Modifier le fichier de configuration :

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

Remplacer les détails de la base de données par les valeurs que créées précédemment :

```
define('DB_NAME', 'wordpress');
define('DB_USER', 'kalicem');
define('DB_PASSWORD', 'Monmdpwordpress42');
define('DB_HOST', 'localhost');
```

Enregistrez et fermez le fichier.

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

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

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Maintenant il faut vérifier que la propriété et les autorisations du fichier `wp-config.php` sont correct :

```
sudo chown lighttpd:lighttpd /var/www/html/wp-config.php
sudo chmod 644 /var/www/html/wp-config.php
```

Déterminer le contexte SELinux des fichiers WordPress :

```
sudo ls -Z /var/www/html
```

Définissez le contexte SELinux approprié pour les fichiers WordPress :

```
sudo chcon -R -t httpd_sys_content_t /var/www/html
```

Remplacez-le `/var/www/lighttpd/wordpress` par le chemin réel vers votre installation WordPress.

Confirmez que le contexte SELinux a été mis à jour avec succès :

```
sudo ls -Z /var/www/html
```

Redémarrez le serveur Lighttpd pour appliquer les modifications :

```
sudo systemctl restart lighttpd
```

En définissant le contexte SELinux correct pour les fichiers WordPress, SELinux devrait autoriser l'accès à ces derniers tout en appliquant les politiques de sécurité. Vous pouvez désormais visiter votre site Web à l'adresse `http://localhost:8080/`.

Configurer WordPress

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Born2BeRocky

Username

mdemare

Names can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

Monmdpuser42

Strong

Hide

Important: You will need this password to log in. Please store it in a secure location.

Your Email

mdemare42@gmail.com

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

Maintenant que j'ai accès au tableau de bord WordPress, je peux créer une nouvelle page avec le contenu souhaité.

[Blog](#) [About](#) [FAQs](#) [Authors](#)

Born2BeRocky

Born2BeRoot : Construire un serveur sécurisé et performant

Le projet **Born2BeRoot** a pour objectif de configurer et sécuriser un serveur Linux minimaliste. En utilisant **Rocky Linux 9.4**, ce projet permet d'acquérir les compétences fondamentales en administration système et en sécurité, tout en mettant en place des outils pour surveiller et protéger le serveur.

Objectifs principaux

- Installation minimale :**
 - Mise en place d'un système sans interface graphique dans une machine virtuelle VirtualBox.
 - Utilisation de **LVM** pour une gestion flexible des partitions disques, incluant le chiffrement pour protéger les données sensibles.
- Sécurisation de l'accès utilisateur :**
 - Création d'un utilisateur standard avec des permissions limitées.
 - Configuration de **sudo** pour les privilèges administratifs.
- Renforcement de la sécurité système :**
 - Mise en place d'un pare-feu avec **firewallld** pour filtrer les accès réseau.
 - Désactivation des connexions root via SSH.
- Automatisation de la surveillance :**
 - Désabonnement d'un service **monitoring** chose à afficher

Ajout d'un service supplémentaire : Fail2Ban

Pour améliorer la sécurité du serveur, **Fail2Ban** a été installé. Ce logiciel de prévention des intrusions protège contre les attaques par force brute en surveillant les journaux système pour détecter des comportements suspects, tels que des échecs de connexion répétés.

Comment fonctionne Fail2Ban ?

1. **Surveillance des journaux** : Fail2Ban analyse les fichiers journaux des services (comme SSH) pour détecter les tentatives d'accès non autorisées.
2. **Blocage des IP suspectes** : Lorsqu'une activité suspecte est identifiée, l'adresse IP responsable est temporairement ou définitivement bannie via des règles dynamiques du pare-feu.

Avantages de Fail2Ban :

- Réduit le risque d'accès non autorisé grâce à un blocage automatique des adresses IP malveillantes.
- Améliore la sécurité globale du serveur de manière proactive.

Fail2Ban est un outil essentiel pour tout administrateur souhaitant protéger son serveur contre les menaces courantes tout en renforçant sa posture de sécurité.

Pour installer Fail2ban sur un système Linux, vous pouvez suivre ces étapes générales :

Mettre à jour les listes de paquets : assurez-vous que le cache de votre gestionnaire de paquets est à jour en exécutant :
`sudo dnf update`

Installer Fail2ban : Utilisez le gestionnaire de paquets pour installer Fail2ban :
`sudo dnf install fail2ban`

Démarrer le service Fail2ban : Après l'installation, démarrez le service Fail2ban :
`sudo systemctl start fail2ban`

Activer le service Fail2ban : si vous souhaitez que Fail2ban démarre automatiquement au démarrage, activez le service :
`sudo systemctl enable fail2ban`

Vérifier l'installation : Vérifiez l'état du service Fail2ban pour vous assurer qu'il fonctionne sans erreur :
`sudo systemctl status fail2ban`

Configuration : le fichier de configuration principal de Fail2ban se trouve généralement dans `/etc/fail2ban/jail.conf` ou `/etc/fail2ban/jail.local`.

Vous pouvez personnaliser la configuration en fonction de vos besoins.

N'oubliez pas de redémarrer Fail2ban après avoir apporté des modifications aux fichiers de configuration :

`sudo systemctl restart fail2ban`

Vérifier les journaux : surveillez les journaux de Fail2ban pour vous assurer qu'ils fonctionnent comme prévu :

`sudo tail -f /var/log/fail2ban.log`

```

[mdemare@mdemare42 ~]$ sudo tail -f /var/log/fail2ban.log
2024-11-16 19:18:13,384 fail2ban.server [6625]: INFO -----
-----
2024-11-16 19:18:13,384 fail2ban.server [6625]: INFO Starting Fail2ban v1.0.2
2024-11-16 19:18:13,384 fail2ban.observer [6625]: INFO Observer start...
2024-11-16 19:18:13,391 fail2ban.database [6625]: INFO Connected to fail2ban persistent data
base '/var/lib/fail2ban/fail2ban.sqlite3'
2024-11-16 19:18:13,392 fail2ban.database [6625]: WARNING New database created. Version '4'

```

Et voilà ! Fail2ban devrait maintenant être installé et exécuté sur votre système, contribuant à protéger votre serveur contre les attaques par force brute. N'oubliez pas de consulter régulièrement les journaux de Fail2ban et d'ajuster sa configuration selon vos besoins pour garantir une sécurité optimale.

Ajouter SFTP

Configurer le SFTP

Modifier la configuration SSH pour activer le SFTP uniquement pour certains utilisateurs :

Éditez le fichier de configuration SSH :

```
sudo vi /etc/ssh/sshd_config
```

Recherchez la ligne suivante et décommentez-la si nécessaire :

```
Subsystem sftp /usr/libexec/openssh/sftp-server
```

Ajoutez cette section en bas pour restreindre certains utilisateurs à SFTP uniquement :

```
Match User sftpuser
```

```
ForceCommand internal-sftp
```

```
ChrootDirectory /home/sftpuser
```

```
PermitTunnel no
```

```
AllowAgentForwarding no
```

```
AllowTcpForwarding no
```

```
X11Forwarding no
```

Créez un utilisateur SFTP sans accès SSH :

```
sudo useradd -m -s /sbin/nologin sftpuser
```

Définissez un mot de passe pour l'utilisateur :

```
sudo passwd sftpuser
```

Changez les permissions du répertoire utilisateur pour le chroot :

```
sudo chmod 755 /home/sftpuser
```

```
sudo mkdir /home/sftpuser/uploads
```

```
sudo chown sftpuser:sftpuser /home/sftpuser/uploads
```

Redémarrer le service SSH :

```
sudo systemctl restart sshd
```

Connectez-vous avec FileZilla ou un client SFTP :
Hôte : L'ip de la machine, pour moi : 10.11.200.162
Port : 4242 (ou si pas modifier le port 22)
Protocole : SFTP
Nom d'utilisateur : sftpuser
Mot de passe : défini lors de la création de l'utilisateur

Tu veux tester ton Born2BeRoot?

```
curl -O  
https://raw.githubusercontent.com/42mdemare/Born2BeRoot/main/Born2BeRoot_Tester.sh
```

Une fois téléchargé, donnez des permissions d'exécution au script si nécessaire :

```
chmod +x Born2BeRoot_Tester.sh
```

Si vous utilisez un compte utilisateur normal, exécutez les fichiers Born2BeRoot_Tester.sh avec la commande `sudo`.

Les fichiers cronjob détectent uniquement le fichier nommé `monitoring.sh`.

Vous pouvez modifier le nom du fichier dans `Born2BeRoot_Tester.sh` pour qu'il corresponde au nom de fichier de votre script de surveillance (uniquement si le vôtre est différent).

Exécutez la commande en utilisant « `sudo bash Born2BeRoot_Tester.sh` » ou « `sudo ./Born2BeRoot_Tester.sh` »

Assurez-vous d'exécuter en utilisant votre nom d'utilisateur. Pas en tant que `ROOT` ou en utilisant `SU`.

ANNEXE

Commande pour créer la signature

Pour créer la signature de la machine virtuelle Born2BeRoot, ouvrez un terminal localisé dans le répertoire contenant le fichier Born2BeRoot.vdi. Ensuite, exécutez la commande suivante :

```
echo $(sha1sum Born2BeRoot.vdi | awk '{print $1}') > signature.txt
```

Qu'est-ce qu'une signature ?

Une signature est un mécanisme cryptographique permettant :

- D'identifier et valider l'intégrité d'une machine virtuelle ou de ses composants.
- De garantir l'authenticité de la VM grâce à une somme de contrôle unique.

Caractéristiques de la signature

- La signature est cryptographique, calculée à l'aide de l'algorithme SHA (Secure Hash Algorithm).
- Elle change à chaque modification de la machine virtuelle (ouverture, modification des fichiers, etc.).
- Elle permet de vérifier que la VM n'a pas été altérée depuis sa création.
- Pour déployer la VM en toute sécurité, une clé peut être associée à cette signature.

La signature correspond donc à un identifiant unique de la machine, garantissant son intégrité et la sécurité de ses composants.

Comparer les signatures

La commande diff permet de comparer les signatures pour vérifier si la machine virtuelle a été modifiée.

Assurez-vous que le fichier signature.txt contenant la signature SHA-1 d'origine se trouve dans le même répertoire que le fichier Born2BeRoot.vdi.

Ouvrez un terminal dans le répertoire contenant Born2BeRoot.vdi et exécutez :

```
diff signature.txt <$(sha1sum Born2BeRoot.vdi | awk '{print $1}')
```

Résumé des différences basiques entre Rocky Linux et Debian

Aspect	Rocky Linux	Debian
Origine	Issue de Red Hat Enterprise Linux (RHEL).	Distribution indépendante.
Type de système	Basée sur RPM (Red Hat Package Manager).	Basée sur DEB (Debian Package Manager).
Public cible	Entreprises et serveurs professionnels.	Serveurs, desktops, et usage général.
Stabilité	Très stable, alignée avec RHEL.	Très stable, mais offre aussi des versions testing/unstable.
Gestion des paquets	dnf ou yum pour RPM.	apt pour DEB.
Cycle de support	Long terme (aligné avec RHEL : ~10 ans).	Varié : versions stables supportées ~5 ans.
Philosophie	Centrée sur les entreprises et la compatibilité RHEL.	Axée sur la liberté et l'ouverture.
Performance	Optimisée pour des environnements professionnels.	Polyvalente pour tous types d'environnements.
Usage principal	Serveurs, datacenters, infrastructures critiques.	Usage général, développeurs, serveurs et desktops.
Communauté	Soutenue par la communauté et des entreprises.	Grande communauté mondiale.

Rocky Linux : Conçue pour les entreprises, compatible avec RHEL, très stable, basée sur RPM avec dnf/yum.

Debian : Polyvalente, adaptée aux serveurs et desktops, stable avec options de testing, basée sur DEB avec apt.

Différences entre apt, aptitude et dnf

apt

- Pour Debian/Ubuntu et dérivés.
- Simplicité, vitesse, et efficacité pour les tâches courantes.
- Gestion des dépendances suffisante pour la plupart des besoins.

aptitude

- Plus avancé que apt, avec une meilleure résolution des conflits.
- Interface interactive en mode texte pour une navigation et une gestion des paquets plus détaillée.
- Idéal pour les administrateurs systèmes ou les scénarios complexes.

dnf

- Pour Fedora, Rocky Linux, RHEL, CentOS et autres systèmes basés sur RPM.
- Plus rapide, plus moderne, et gère mieux les dépendances que son prédécesseur yum.
- Recommandé pour les administrateurs professionnels dans des environnements critiques.

Comment tester si pas d'environnement graphique installé

Sur Debian/Ubuntu :

Recherchez si un environnement graphique est installé (comme GNOME, KDE, XFCE, etc.):

```
dpkg -l | grep -E 'gnome|kde|xfce|mate|lxde|wayland|xorg'
```

Si aucun résultat n'apparaît, il n'y a probablement pas d'environnement graphique installé.

Sur Fedora/Rocky Linux/CentOS :

Utilisez la commande suivante pour rechercher les paquets liés à un environnement graphique :

```
rpm -qa | grep -E 'gnome|kde|xfce|mate|lxde|wayland|xorg'
```

Si aucun résultat n'apparaît, il n'y a probablement pas d'environnement graphique installé.

Utilisez la commande suivante :

```
systemctl get _default
```

Si la sortie est `graphical.target`, cela signifie que le système est configuré pour démarrer avec un environnement graphique.

Si la sortie est `multi-user.target`, cela signifie qu'il n'y a pas d'environnement graphique activé au démarrage.

Comment savoir si Firewalld est lancé

En exécutant la commande suivante :

```
systemctl status firewalld
```

ou vérifier si firewalld est actif uniquement :

```
systemctl is-active firewalld
```

ou vérifier si firewalld est activé au démarrage :

```
systemctl is-enabled firewalld
```

Comment savoir si SSHD (Secure Shell Daemon) est lancé

En exécutant la commande suivante :

```
systemctl status sshd
```

ou vérifier si sshd est actif uniquement :

```
systemctl is-active sshd
```

ou vérifier si sshd est activé au démarrage :

```
systemctl is-enabled sshd
```

Pour se connecter au ssh, dans un terminal utiliser :

recupérer l'ip avec :

```
hostname -I
```

```
ssh mdemare@10.11.200.103 -p 4242
```

Comment tester les groupes d'appartenance de l'utilisateur

Pour voir les groupes de l'utilisateur, il suffit d'utiliser la commande suivante :

```
groups mdemare
```

ou

```
id mdemare
```

groups mdemare : Affiche les groupes secondaires de mdemare.

id mdemare : Fournit des informations supplémentaires comme l'UID, le GID, et les groupes.

uid : Identifiant utilisateur.

gid : Groupe principal de l'utilisateur.

groups : Tous les groupes (principaux et secondaires) auxquels appartient l'utilisateur.

Comment créer un utilisateur

Utilisez la commande `useradd` pour créer un utilisateur.

```
sudo useradd -m <username>
```

Options importantes :

`-m` : Crée automatiquement un répertoire personnel pour l'utilisateur (par exemple, `/home/<username>`).

exemple :

```
sudo useradd -m evaluateur
```

Une fois l'utilisateur créé, vous devez définir son mot de passe avec la commande `passwd`.

Exemple :

```
sudo passwd evaluateur
```

Le système vous demandera d'entrer et de confirmer le mot de passe de l'utilisateur.

Utilisez la commande `groupadd` pour créer un groupe.

Exemple :

```
sudo groupadd evaluating
```

Si vous souhaitez ajouter l'utilisateur à d'autres groupes (secondaires), utilisez la commande `usermod` :

```
sudo usermod -aG evaluating evaluateur
```

`-aG` : Ajoute (a) l'utilisateur aux groupes secondaires spécifiés (G).

Pour le retirer d'un groupe :

```
sudo gpasswd -d evaluating evaluateur
```

Pour supprimer un utilisateur complètement :

```
sudo userdel evaluateur
```

Pour supprimer un utilisateur ainsi que son répertoire personnel et tous les fichiers qu'il contient :

```
sudo userdel -r evaluateur
```


Fichier de politique de mot de passe

`/etc/security/pwquality.conf`, `/etc/pam.d/system-auth` et `/etc/pam.d/password-auth`

Le fichier `/etc/security/pwquality.conf` configure les **politiques de qualité des mots de passe** en spécifiant les exigences minimales pour les nouveaux mots de passe, telles que la longueur, la complexité et la prévention des mots de passe faciles à deviner.

Le fichier `/etc/pam.d/system-auth` configure **l'authentification PAM (Pluggable Authentication Modules)** pour les utilisateurs locaux. C'est un fichier central utilisé pour définir les règles d'authentification, de gestion des mots de passe, des comptes, et des sessions.

Similaire à `/etc/pam.d/system-auth`, le fichier `/etc/pam.d/password-auth` configure également les politiques d'authentification PAM, mais il est utilisé principalement pour **les services distants ou non interactifs**, comme SSH ou des connexions réseau.

Pour garantir des mots de passe solides et sûrs, une politique stricte est mise en place :

Expiration régulière : Les mots de passe expirent tous les 30 jours pour limiter les risques liés à l'utilisation prolongée d'un même mot de passe. Cela réduit la fenêtre d'opportunité pour un attaquant.

Délai avant modification : Un délai de 2 jours entre deux changements empêche les utilisateurs de contourner la politique en réutilisant rapidement un ancien mot de passe.

Avertissement avant expiration : L'utilisateur est informé 7 jours avant l'expiration, garantissant qu'il a le temps de mettre à jour son mot de passe sans interruption de service.

Complexité des mots de passe : Exiger 10 caractères minimum, une majuscule, une minuscule, un chiffre et interdire plus de 3 caractères identiques consécutifs protège contre les mots de passe trop simples ou prévisibles.

Restrictions spécifiques : Interdire l'utilisation du nom d'utilisateur dans le mot de passe réduit les risques liés aux attaques par dictionnaire ou devinettes.

Renouvellement sécurisé (sauf pour root) : Au moins 7 caractères doivent différer de l'ancien mot de passe, empêchant les utilisateurs de créer des variantes trop proches.

En appliquant ces règles, la politique améliore la sécurité des mots de passe en rendant leur découverte ou leur devinette beaucoup plus difficile pour des attaquants potentiels.

Comment changer le HostName

La commande `hostname` ou `hostnamectl` permet de connaître le Hostname actuel.

Pour changer de façon temporaire, jusqu'au redémarrage de la VM :

```
sudo hostname new_name
```

Changer de façon permanente en modifiant le fichier `/etc/hostname` :

```
sudo vi /etc/hostname
```

```
sudo vi /etc/hosts
```

Redémarrez le service `systemd-hostnamed` pour appliquer les modifications :

```
sudo systemctl restart systemd-hostnamed
```

ou

```
sudo hostnamectl set-hostname new_name
```

Cette méthode est **permanente** et ne nécessite pas de modification manuelle de fichiers ou de redémarrage du service.

```
sudo vi /etc/hosts
```

Les partitions et LVM (Logical Volume Management)

Pourquoi LVM pour Born2BeRoot ?

LVM offre des avantages importants par rapport aux partitions traditionnelle

- **Flexibilité** : Permet d'ajuster dynamiquement la taille des volumes logiques sans affecter les données.
- **Gestion simplifiée** : Vous pouvez ajouter de nouveaux disques physiques au groupe de volumes et les utiliser immédiatement.
- **Snapshots** : LVM permet de créer des "snapshots", une copie instantanée d'un volume logique, utile pour les sauvegardes.
- **Optimisation de l'espace** : Les partitions classiques limitent l'utilisation efficace de l'espace. Avec LVM, l'espace est partagé entre les volumes logiques et ajusté en fonction des besoins.
- **Adapté aux serveurs** : Born2BeRoot, qui vise la gestion serveur, profite de cette flexibilité pour s'adapter aux évolutions des besoins.

Configuration des partitions et volumes logiques

La configuration des partitions de Born2BeRoot utilise LVM (Logical Volume Management) et le chiffrement pour garantir sécurité et flexibilité.

Disque principal :

- **sda** (31 Go) contient trois partitions principales :
 - **sda1** : 512 Mo, monté sur /boot, contient les fichiers nécessaires au démarrage.
 - **sda2** : 1 Ko, partition étendue contenant la partition logique sda5.
 - **sda5** : 29,5 Go, chiffrée avec **sda5_crypt** pour protéger les données sensibles.

Volumes logiques LVM : À l'intérieur de sda5_crypt, le groupe de volumes LVMGroup est divisé en plusieurs volumes logiques pour isoler les différents répertoires :

- **root** : 10 Go, monté sur /, contient le système principal.
- **swap** : 2,3 Go, utilisé comme mémoire virtuelle.
- **home** : 5 Go, pour les répertoires utilisateurs.
- **var** : 3 Go, pour les fichiers variables comme les caches et les journaux.
- **srv** : 3 Go, pour héberger des services comme un serveur web.
- **tmp** : 3 Go, pour les fichiers temporaires.
- **var-log** : 4 Go, dédié aux journaux système, pour éviter la saturation de /var.

Avantages de cette configuration :

- **Sécurité** : Le chiffrement via **sda5_crypt** protège les données sensibles.
- **Isolation** : Les répertoires critiques (/var, /tmp, /srv, etc.) sont séparés pour éviter les conflits ou saturations.
- **Flexibilité** : Grâce à LVM, la taille des volumes peut être ajustée dynamiquement selon les besoins.
- **Robustesse** : La répartition optimise la gestion des ressources et garantit la stabilité du système.

Cette organisation est idéale pour un système sécurisé comme Born2BeRoot, où la protection des données et la gestion efficace des ressources sont primordiales.

Comment vérifier sudo

Pour vérifier que sudo est installé, utiliser la commande suivante pour retourner la version installée :

```
rpm -q sudo
```

Voici une commande pour tester sudo :

```
sudo ls /root
```

liste le contenu du répertoire /root .

Pour voir les règles imposées par le sujet, regarder dans le fichier `/etc/sudoers` de manière sécurisée avec la commande :

```
sudo visudo
```

Pour vérifier que le dossier `/var/log/sudo` existe, on peut regarder ce qu'il contient :

```
sudo ls /var/log/sudo/
```

Pour voir toutes les commandes sudo utilisées, on peut accéder au journal :

```
sudo journalctl _COMM=sudo
```

Tester Firewalld

En exécutant la commande suivante :

```
systemctl status firewalld
```

ou vérifier si firewalld est actif uniquement :

```
systemctl is-active firewalld
```

ou vérifier si firewalld est activé au démarrage :

```
systemctl is-enabled firewalld
```

Résumé des avantages de firewalld

Gestion simplifiée :

- Modifications dynamiques sans redémarrage nécessaire.
- Interface claire avec des zones prédéfinies adaptées à différents environnements réseau (public, privé, etc.).

Sécurité améliorée :

- Support des services et protocoles courants.
- Journalisation des connexions autorisées ou bloquées.

Flexibilité :

- Règles temporaires ou permanentes.
- Redirection de ports, blocage d'adresses IP spécifiques, etc.

Compatibilité :

- Intégration avec iptables, nftables, et des environnements cloud/conteneurs.
- API D-Bus pour une gestion automatisée.

Zones de confiance :

- Différentes zones pour contrôler les niveaux de sécurité des interfaces réseau.

Pour vérifier les port ouvert dans Firewalld :

```
sudo firewall-cmd --list-ports
```

Pour lister les règles permanents :

```
sudo firewall-cmd --permanent --list-all
```

Pour lister les redirections de port :

```
sudo firewall-cmd --list-forward-ports
```

Ajouter un port temporairement (ne sera pas conservé après redémarrage)

```
sudo firewall-cmd --zone=public --add-port=<port>/tcp
```

Exemple pour le port 8080

```
sudo firewall-cmd --zone=public --add-port=8080/tcp
```

Ajouter un port de manière permanente (conservé après redémarrage)

```
sudo firewall-cmd --zone=public --add-port=<port>/tcp --permanent
```

Exemple pour le port 8080

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

Recharger la configuration après un changement permanent

```
sudo firewall-cmd --reload
```

Supprimer un port temporairement

```
sudo firewall-cmd --zone=public --remove-port=<port>/tcp
```

Exemple pour le port 8080

```
sudo firewall-cmd --zone=public --remove-port=8080/tcp
```

Supprimer un port de manière permanente

```
sudo firewall-cmd --zone=public --remove-port=<port>/tcp --permanent
```

Exemple pour le port 8080

```
sudo firewall-cmd --zone=public --remove-port=8080/tcp --permanent
```

Recharger la configuration après un changement permanent

```
sudo firewall-cmd --reload
```

Ajouter une redirection temporairement

```
sudo firewall-cmd --zone=public  
--add-forward-port=port=<source_port>:proto=tcp:toport=<destination_port>
```

Exemple pour rediriger le port 80 vers 8080

```
sudo firewall-cmd --zone=public --add-forward-port=port=80:proto=tcp:toport=8080
```

Ajouter une redirection de manière permanente

```
sudo firewall-cmd --zone=public  
--add-forward-port=port=<source_port>:proto=tcp:toport=<destination_port> --permanent
```

Exemple pour rediriger le port 80 vers 8080

```
sudo firewall-cmd --zone=public --add-forward-port=port=80:proto=tcp:toport=8080  
--permanent
```

Recharger la configuration après un changement permanent

```
sudo firewall-cmd --reload
```

Supprimer une redirection temporairement

```
sudo firewall-cmd --zone=public  
--remove-forward-port=port=<source_port>:proto=tcp:toport=<destination_port>
```

Exemple pour supprimer la redirection du port 80 vers 8080

```
sudo firewall-cmd --zone=public --remove-forward-port=port=80:proto=tcp:toport=8080
```

Supprimer une redirection de manière permanente

```
sudo firewall-cmd --zone=public  
--remove-forward-port=port=<source_port>:proto=tcp:toport=<destination_port> --permanent
```

Exemple pour supprimer la redirection du port 80 vers 8080

```
sudo firewall-cmd --zone=public --remove-forward-port=port=80:proto=tcp:toport=8080  
--permanent
```

Recharger la configuration après un changement permanent

```
sudo firewall-cmd --reload
```

Tester SSHD (Secure Shell Daemon)

En exécutant la commande suivante :

```
systemctl status sshd
```

ou vérifier si sshd est actif uniquement :

```
systemctl is-active sshd
```

ou vérifier si sshd est activé au démarrage :

```
systemctl is-enabled sshd
```

Pour se connecter au ssh, dans un terminal utiliser :

```
ssh mdemare@10.11.200.103 -p 4242
```

Vérifie que ssh utilise uniquement le port 4242 :

```
sudo grep "^Port" /etc/ssh/sshd_config
```

Tester le monitoring.sh

Le script monitoring est dans le dossier root :

```
sudo vi /root/monitoring.sh
```

Ce script Bash extrait des informations détaillées sur le système, son matériel, et son utilisation, en utilisant des commandes Linux standard. Voici une explication résumée de chaque section :

Entête et configuration

`#!/bin/bash` : Spécifie que le script doit être interprété avec Bash.

`export PATH=...` : Définit les chemins des exécutable nécessaires pour garantir que les commandes du script sont accessibles.

La commande `uname` est utilisée pour obtenir des informations sur le système d'exploitation et le noyau Linux. Voici un aperçu des options courantes et leur signification :

La commande `$(awk '{print $1}')` utilise awk pour extraire et afficher la `n` colonne ou le `n` champ d'une entrée donnée. Cela se fait en analysant les données ligne par ligne et en séparant les champs selon un délimiteur (par défaut, un espace ou une tabulation).

Informations sur le système

ARCH : Architecture matérielle (ex. x86_64).

```
ARCH=$(uname -m)
```

DISTRIBUTION et VERSION : Distribution Linux et version.

```
DISTRIBUTION=$(awk '{print $1}' /etc/redhat-release)
```

```
VERSION=$(awk '{print $4}' /etc/redhat-release)
```

KERNEL : Version du noyau Linux.

```
KERNEL=$(uname -r)
```

Informations matérielles et utilisation

CPU_PHYSICAL : Nombre de processeurs physiques (ex. 1).

```
CPU_PHYSICAL=$(lscpu | awk '/Socket(s):/ {print $2}')
```

VCPU : Nombre de CPU logiques ou threads (ex. 4).

```
VCPU=$(lscpu | awk '/^CPU(s):/ {print $2}')
```

RAM_USAGE : Utilisation actuelle de la RAM (ex. 500/2000MB (25.0%)).

```
RAM_USAGE=$(free -m | awk '/Mem:/ {printf "%.0f/%.0fMB (%.2f%%)", $3, $2, $3*100/$2}')
```

DISK_USAGE : Utilisation du disque principal (ex. 10G/100G (10%)).

```
DISK_USAGE=$(df -BG --output=size,used,pcent / | awk 'NR==2 {printf "%s/%s (%s)\n", $2, $1, $3}')
```

CPU_LOAD : Charge CPU actuelle (ex. 5.0%).

```
CPU_LOAD=$(top -b -n1 | awk '/Cpu(s):/ {printf "%.1f%%", $2 + $4}')
```

LAST_BOOT : Date et heure du dernier démarrage.

```
LAST_BOOT=$(who -b | awk '{print $3, $4}')
```


LVM_STATUS : Indique si LVM est utilisé (yes ou no).

```
LVM_STATUS=$(lsblk | grep -q "lvm" && echo "yes" || echo "no")
```

Informations réseau

TCP_CONNECTIONS : Nombre de connexions TCP actives (état ESTABLISHED).

```
TCP_CONNECTIONS=$(ss -tn state established | grep -c ESTAB)
```

USER_LOG : Nombre d'utilisateurs actuellement connectés.

```
USER_LOG=$(who | wc -l)
```

IP : Adresse IP principale de la machine.

```
IP=$(hostname -I | awk '{print $1}')
```

MAC : Adresse MAC de l'interface réseau.

```
MAC=$(ip link show | awk '/ether/ {print $2}')
```

Commandes sudo

SUDO_CMDSD : Nombre de commandes sudo exécutées, enregistré dans les journaux.

```
SUDO_CMDSD=$(journalctl _COMM=sudo | grep COMMAND | wc -l)
```

Résumé

Ce script fournit un rapport complet sur :

Le système : distribution, version, noyau.

Le matériel : architecture, CPU, RAM, disque.

Le réseau : IP, MAC, connexions TCP actives.

L'utilisation : charge CPU, dernier démarrage, commandes sudo.

Le stockage : état de LVM, disque principal.

Le script est exécuté toute les 10 minutes grâce à une règle cron configurée dans le fichier /etc/crontab :

```
sudo vi /etc/crontab
```

avec la règle suivante :

```
*/10 * * * * root bash /root/monitoring.sh >/dev/null 2>&1
```

execute toutes les 10 minutes les script monitoring.sh avec bash en tant que root et redirige les erreurs de la sortie 2 (stdout) vers /dev/null

La partie bonus

pour vérifier les services :

```
systemctl is-active lighttpd
systemctl is-active mariadb
systemctl is-active php-fpm
ou
systemctl status lighttpd
systemctl status mariadb
systemctl status php-fpm
```

php-fpm (PHP FastCGI Process Manager) est une version spécialisée de PHP conçue pour gérer les requêtes PHP dans un environnement serveur performant. PHP-FPM fonctionne en tant que service indépendant. Le serveur web communique avec php-fpm via le protocole FastCGI.

pour voir les partitions :

```
lsblk
```

Service bonus Fail2Ban

```
systemctl status fail2ban
```

Fail2Ban est un outil essentiel pour tout administrateur souhaitant protéger son serveur contre les menaces courantes tout en renforçant sa posture de sécurité.

Supprimer une entrée GRUB suite a une MAJ

Listez les noyaux installés pour identifier ceux qui sont obsolètes

```
sudo rpm -q kernel
```

Pour vérifier le noyau actif :

```
uname -r
```

Si des noyaux obsolètes sont encore installés, vous pouvez les supprimer avec dnf :

```
sudo dnf remove kernel-<version>
```

Par exemple :

```
sudo dnf remove kernel-5.14.0-427.el9.x86_64
```

Résumé des commandes

Commandes liées à la signature :

- `sha1sum Born2BeRoot.vdi | awk '{print $1}'`
 - `echo $(sha1sum Born2BeRoot.vdi | awk '{print $1}') > signature.txt`
 - `diff signature.txt <(sha1sum Born2BeRoot.vdi | awk '{print $1}')`
-

Commandes liées aux environnements graphiques :

- `rpm -qa | grep -E 'gnome|kde|xfce|mate|lxde|wayland|xorg'`
 - `systemctl get-default`
-

Commandes liées à Firewallld :

- `systemctl status firewalld`
- `systemctl is-active firewalld`
- `systemctl is-enabled firewalld`
- `sudo firewall-cmd --list-ports`
- `sudo firewall-cmd --permanent --list-all`
- `sudo firewall-cmd --list-forward-ports`
- `sudo firewall-cmd --zone=public --add-port=<port>/tcp`
- `sudo firewall-cmd --zone=public --add-port=<port>/tcp --permanent`
- `sudo firewall-cmd --zone=public --remove-port=<port>/tcp`
- `sudo firewall-cmd --zone=public --remove-port=<port>/tcp --permanent`
- `sudo firewall-cmd --zone=public --add-forward-port=port=<source_port>:proto=tcp:toport=<destination_port>`
- `sudo firewall-cmd --zone=public --add-forward-port=port=<source_port>:proto=tcp:toport=<destination_port> --permanent`
- `sudo firewall-cmd --zone=public --remove-forward-port=port=<source_port>:proto=tcp:toport=<destination_port>`
- `sudo firewall-cmd --zone=public --remove-forward-port=port=<source_port>:proto=tcp:toport=<destination_port> --permanent`
- `sudo firewall-cmd --reload`

Commandes liées à SSHD :

- `systemctl status sshd`
- `systemctl is-active sshd`
- `systemctl is-enabled sshd`
- `hostname -I`
- `ssh <username>@<IP> -p <port>`
- `sudo grep "^Port" /etc/ssh/sshd_config`

Commandes liées aux groupes et utilisateurs :

- `groups <username>`
- `id <username>`
- `sudo useradd -m <username>`
- `sudo passwd <username>`
- `sudo groupadd <groupname>`
- `sudo usermod -aG <groupname> <username>`
- `sudo gpasswd -d <groupname> <username>`
- `sudo userdel <username>`
- `sudo userdel -r <username>`

Commandes liées au fichier de politique de mot de passe :

- `/etc/security/pwquality.conf`
- `/etc/pam.d/system-auth`
- `/etc/pam.d/password-auth`

Commandes générales diverses :

- `sudo chmod +x Born2BeRoot_Tester.sh`
- `sudo ./Born2BeRoot_Tester.sh`
- `sudo bash Born2BeRoot_Tester.sh`
- `sudo visudo`
- `sudo vi /etc/hostname`
- `sudo vi /etc/hosts`
- `sudo hostnamectl set-hostname new_name`
- `sudo systemctl restart systemd-hostnamed`

Commandes liées au monitoring.sh :

- `sudo vi /root/monitoring.sh`
 - `sudo vi /etc/crontab`
 - `sudo journalctl _COMM=sudo`
-

Commandes liées à Fail2Ban :

- `sudo dnf install fail2ban`
 - `sudo systemctl start fail2ban`
 - `sudo systemctl enable fail2ban`
 - `sudo systemctl restart fail2ban`
 - `sudo tail -f /var/log/fail2ban.log`
 - `systemctl status fail2ban`
-

Commandes liées à la gestion des noyaux :

- `sudo rpm -q kernel`
 - `uname -r`
 - `sudo dnf remove kernel-<version>`
-

Commandes bonus liées à Lighttpd, MariaDB, PHP-FPM :

- `systemctl is-active lighttpd`
 - `systemctl is-active mariadb`
 - `systemctl is-active php-fpm`
 - `systemctl status lighttpd`
 - `systemctl status mariadb`
 - `systemctl status php-fpm`
 - `lsblk`
-

Répertoires mentionnés :

- `Born2BeRoot.vdi`
- `/etc/fail2ban/jail.conf`
- `/etc/fail2ban/jail.local`
- `/var/log/fail2ban.log`
- `/etc/crontab`
- `/var/log/sudo/`
- `/etc/hostname`
- `/etc/hosts`