

Projet Linux

VANGEEBERGEN Augustin

August 5, 2024



Table des matières

1	Introduction	3
1.1	Consignes (pratiques) professeur	4
1.2	Roadmap	5
2	Avant de commencer	6
2.1	Hosting	6
2.2	Partitionnement et RAID	6
2.3	VirtualBox	7
2.4	Réseau	15
2.5	Changer le layout clavier	16
2.6	Snapshots de la machine	16
2.7	Ajout des disques virtuels	16
3	Description du logiciel	18
3.1	Clonage du repo git	18
3.2	Menu principal	18
3.3	RAID	18
3.4	Partage sans authentification	18
3.5	Connexion par SSH	18
3.6	Gestion des utilisateurs	19
3.6.1	IP fixe	19
3.6.2	Gestion DNS interne	19
3.7	Serveur temps	19
3.8	Plan de sauvegarde	19
3.9	Sécurité	19
3.10	Journal de bord	19
4	Conclusion	20

1 Introduction

Dans le cadre de ce projet, l'objectif est de configurer un serveur GNU/Linux. Cet exercice consiste à mettre en application la matière vue en classe, et à se préparer à un environnement réel.

Les objectifs globaux sont donc l'application et la compréhension profonde des mécanismes permettant d'héberger les différents services souhaités, de la gestion des utilisateurs, ainsi que de la sécurité, que ce soit au niveau des attaques ou des sauvegardes.

Nous avons le choix d'utiliser n'importe quelle distributon RedHat-like, par exemple Fedora, ou bien Alma, sur laquelle nous avons travaillé en cours. Le choix se porte sur Fedora, qui a une plus grosse communauté (dont je fais partie pour la partie desktop) et a ma préférence (utiliser alma ne changerait bien sûr quasiment rien, les deux distributions étant très similaires).

Dans la deuxième itération de ce projet linux, la barre est automatiquement mise plus haut.

Les consignes professeur sont toujours les mêmes, et sont reprises dans le sous-point suivant. Cependant, je souhaite faire tourner un serveur php pour mon homelab (et ainsi faire d'une pierre deux coups). L'option est donc présente en supplément.

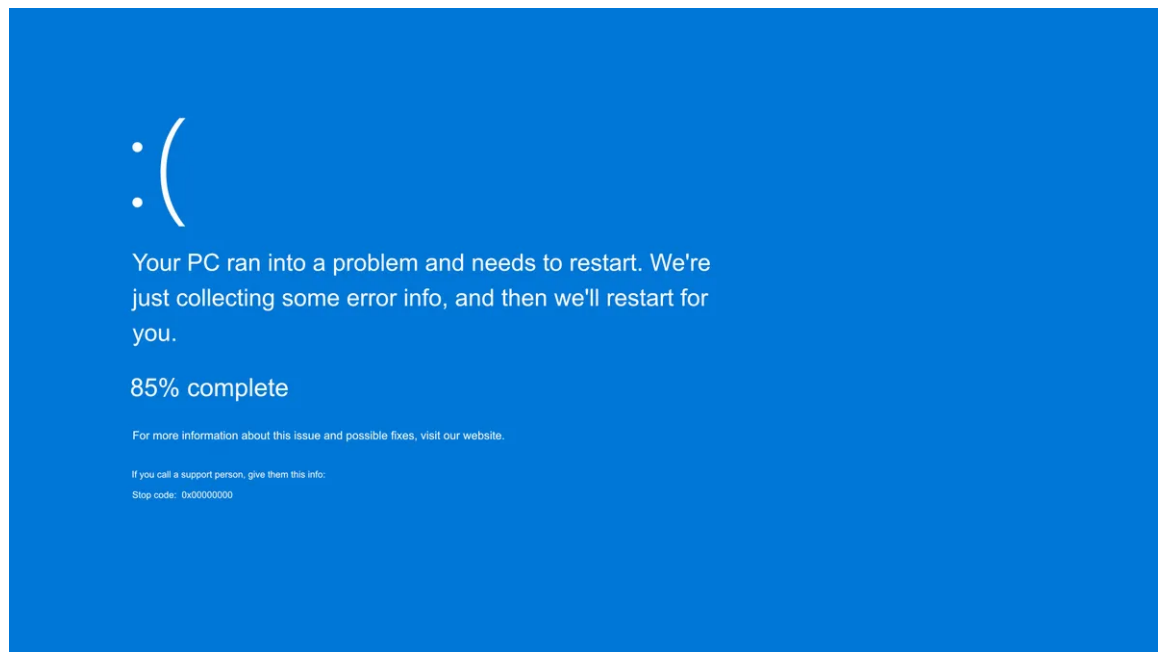
L'OS de test est une machine virtuelle, hébergée sur VirtualBox.

Le but est aussi de pouvoir gérer en ssh le serveur avec plein d'outils utiles, et de pouvoir installer/désinstaller les services à souhait, diminuant ainsi la surface d'attaque.

Nous devons donc gérer des services de partage de fichiers, serveur DNS, serveur Web ainsi que serveur temps.

La dernière étape sera de sécuriser le serveur correctement, notamment en utilisant SELinux, et en définissant les polices d'utilisation correctes.

linux meme (19 juillet 2024) :



1.1 Consignes (pratiques) professeur

Les consignes sont les suivantes :

- Chaque groupe devra mettre en place un serveur linux selon les règles de l'art et devra respecter les bonnes méthodologies pour le faire.
- Le serveur devra permettre de partager un dossier sans authentification aussi bien pour l'environnement Linux que Windows à l'aide de NFS et Samba.
- Une connexion SSH judicieusement sécurisée permettra à l'administrateur de configurer le serveur et d'exécuter des scripts sur le serveur.
- Le serveur devra permettre la mise à disposition pour un client : d'un nom de domaine dans notre domaine, d'un serveur web, d'un accès FTP et Samba à son dossier web et d'une base de données différente pour chaque utilisateur. Le tout devra être automatisé à l'aide de scripts de configurations. Bien sûr chaque client aura un dossier web, une base de données et un domaine différent.
- En bonus, chaque utilisateur devra posséder une adresse mail dans notre domaine ainsi qu'une interface web pour consulter ses mails.
- Le serveur de domaine devra également faire cache pour les requêtes, être maitre dans sa zone et également posséder une zone inverse.
- Le serveur devra permettre aux ordinateurs de son réseau de pouvoir mettre à jour l'heure de leurs machines.
- Le plan de sauvegarde établi devra être mis en place.
- Une attention particulière sera portée sur la sécurisation du serveur et des services à l'aide des outils disponibles. (FW, antivirus, SELinux, ...)
- Toutes les installations et configurations seront notées dans le journal de bord de votre serveur.

1.2 Roadmap

L'ensemble sera scripté pour coller à l'ensemble des cas d'utilisation. Voici donc la liste prévue de ces scripts (pour installer/désinstaller):

- Menu de sélection
- SSH config wizard
- File sharing install wizard
- Web server install wizard
- FTP server install wizard
- MySQL server
- DNS server (+ cache + création de zone + serveur cache)
- Time server
- Sécurisation
- Backup
- Updates
- Partitionnement

Il est donc indispensable de se former au BASH, afin de savoir faire un bon TUI, ainsi que des commandes conditionnelles, selon les features qui sont/ne sont pas déjà installées.

Le service SSH est indépendant des autres services. Le DNS, time server également.

Cependant, les Serveurs Web, SQL et FTP doivent fonctionner en symbiose. C'est aussi le cas du NFS et SMB. Ce sera donc une personne qui s'occupera de ces services deux à deux.

2 Avant de commencer

2.1 Hosting

Hyper-V n'ayant pas apporté satisfaction (principalements bugs de corruption de checkpoints), je me suis tourné vers d'autres alternatives :

- Gnome boxes qui offre peu de flexibilité au niveau du réseau
- VMware Workstation qui a cassé lors d'une mise à jour du kernel de la machine hôte (Fedora 39 vers Fedora 40)

La seule solution correcte restante étant donc Oracle VirtualBox.

Virtualbox permet en outre d'associer un ou plusieurs disques virtuels à une machine virtuelle, ce qui est plutôt intéressant au vu du contexte, nous y viendrons dans la sous-section suivante.

2.2 Partitionnement et RAID

Il est assez compliqué de séparer le partitionnement et la gestion des disques en deux sous sections, car ces deux concepts sont intimement liés.

Il faut premièrement assurer la préservation des données, dans n'importe quel scénario.

Il faut également s'assurer que la hiérarchie du stockage a du sens et qu'elle est pratique.

Sur la machine virtuelle, il y aura :

- sur un disque, le système d'exploitation contenant les partitions suivantes :
 - /boot
 - /swap
 - /
 - /home
- sur un autre disque, ou plutôt un array de disques en RAID :
 - /share
 - /web
- et enfin, sur un ou plusieurs disque(s) additionnel(s) :
 - /backup

Petite liste des niveaux de RAID (redundant array of independent disks) les plus courants :

- RAID 0 : volume agrégé par bandes (ou striping).

Performances en lecture et écriture extrêmement élevées (jusqu'à n fois pour un nombre n de disques en lecture et écriture), mais aucune redondance, et donc non pertinent pour notre serveur.

- RAID 1 : volumes miroirs.

Meilleure redondance des informations (n-1 disques peuvent être retirés). Pire performance niveau vitesse d'écriture (égale à la vitesse d'écriture d'un disque seul) et vitesse de lecture jusqu'à la somme de la vitesse de chaque disque dans l'array (meilleur scénario). Choix rejeté car en pratique on recherche un milieu entre performance et sécurité/redondance.

- RAID 5 : volume agrégé par bandes à parité répartie.

Si un disque lâche, il suffit de remplacer celui-ci, et il peut être reconstitué à partir des autres disques et de la parité stockée sur ceux-ci.

Le choix va se porter sur le RAID 5, qui combine performance et efficacité, en offrant une sécurité sur la casse d'un disque à la fois. On peut donc finir la liste des disques virtuels :

- 1 disque pour l'OS et les fichiers de configuration (disk1)
- 3 disques (c'est à dire le minimum requis pour un RAID 5) pour le stockage (disk2-3-4)
- 1 disque pour la sauvegarde (disk5)

2.3 VirtualBox

Comme dit précédemment, voici les différents disques virtuels créés et leur taille, dans l'interface de VirtualBox :



Ensuite, il faut créer la machine virtuelle, en prenant soin de sélectionner le bon ISO :



On sélectionne la quantité de ram optimale pour le système :



Et on selectionne le disque virtuel créé précédemment :



Et cliquer sur Finish ou Terminer.

Pour lancer la machine virtuelle, il suffit de double-cliquer sur le nom de la machine, dans le côté gauche :



On installe en Anglais, parce que c'est la langue universelle et la seule utilisée en programmation.



On sélectionne "Continue", et on arrive sur le menu principal d'installation :



Le clavier est correctement configuré, puisque c'est une machine virtuelle. La langue est également correctement configurée, ainsi que la date qui dépend de ma machine hôte. Nous allons aller sélectionner le software dont on a besoin. Dans notre cas, nous pouvons conserver la Server Edition.



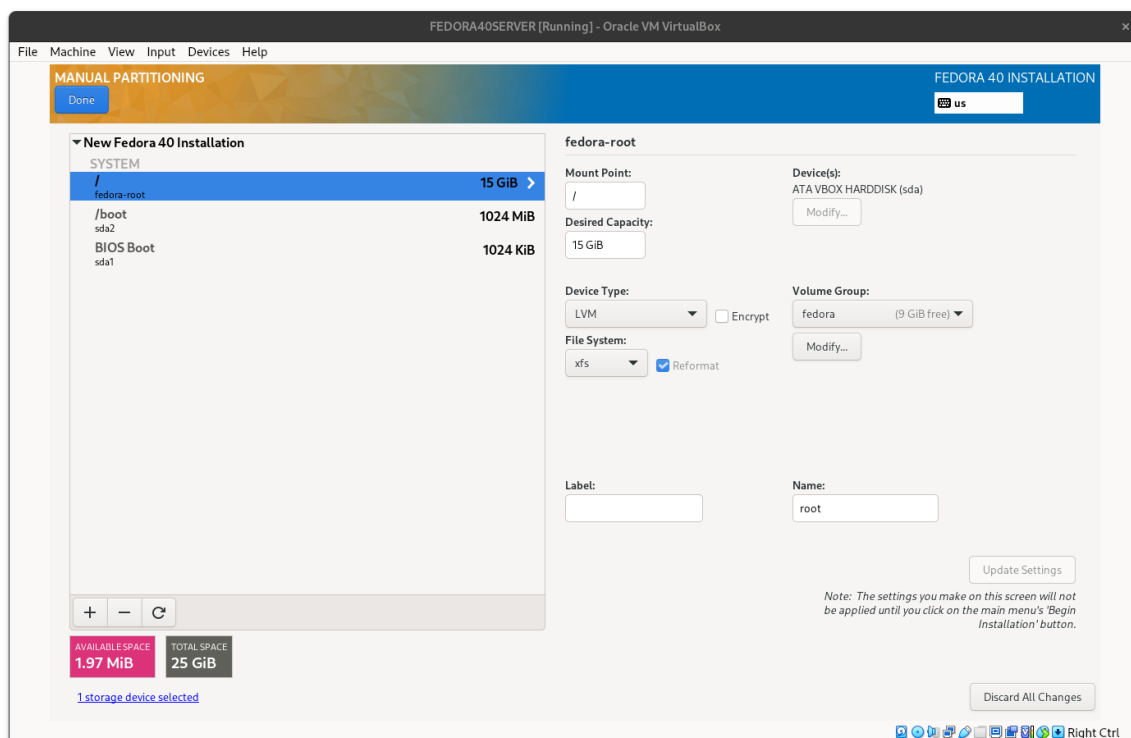
On sélectionne une installation custom :



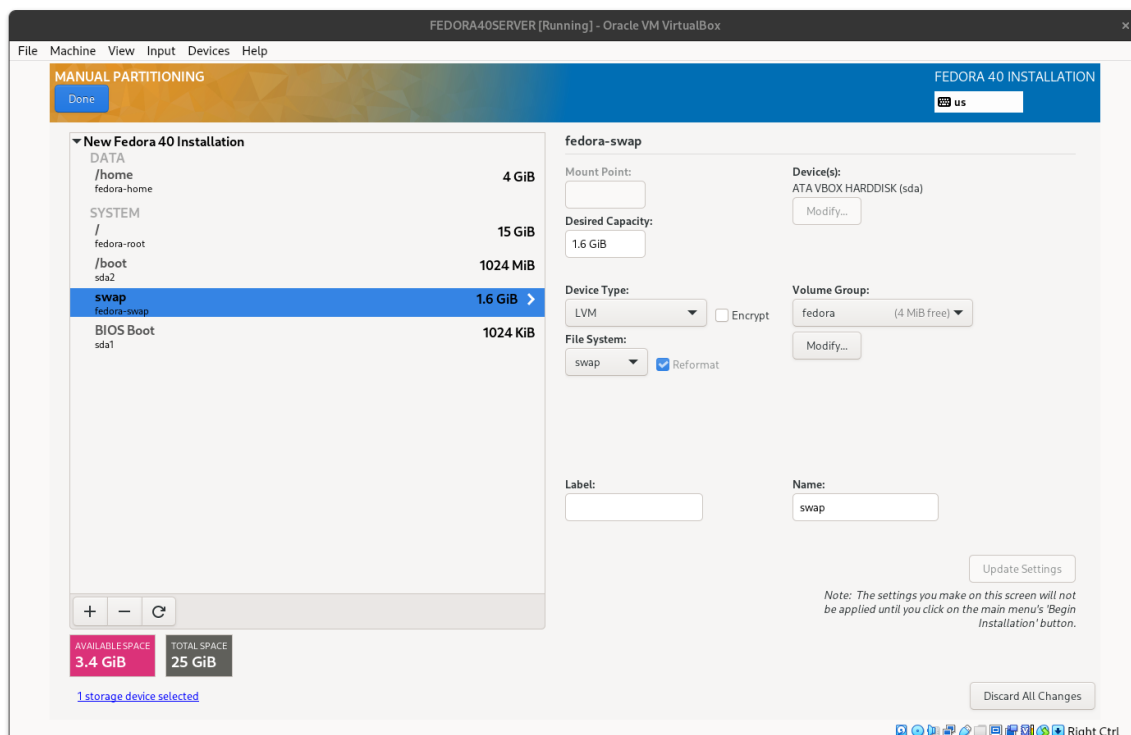
Puis on crée automatiquement les partitions de base.



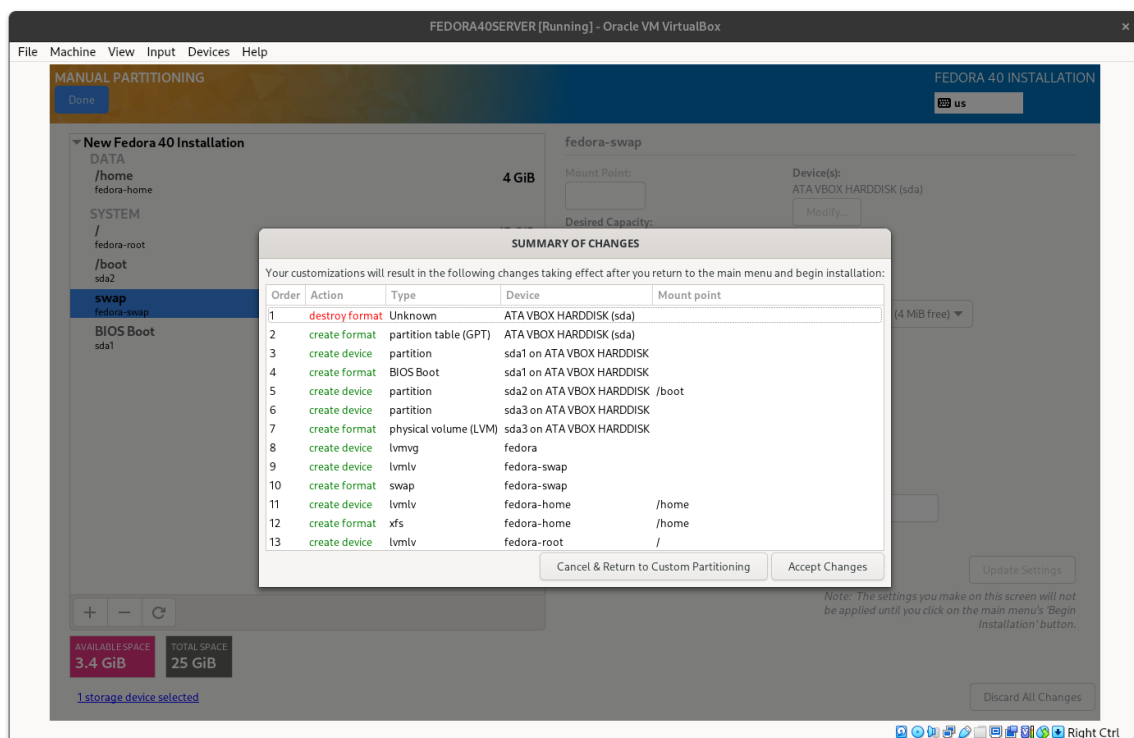
On ne touche pas à la partition automatique :



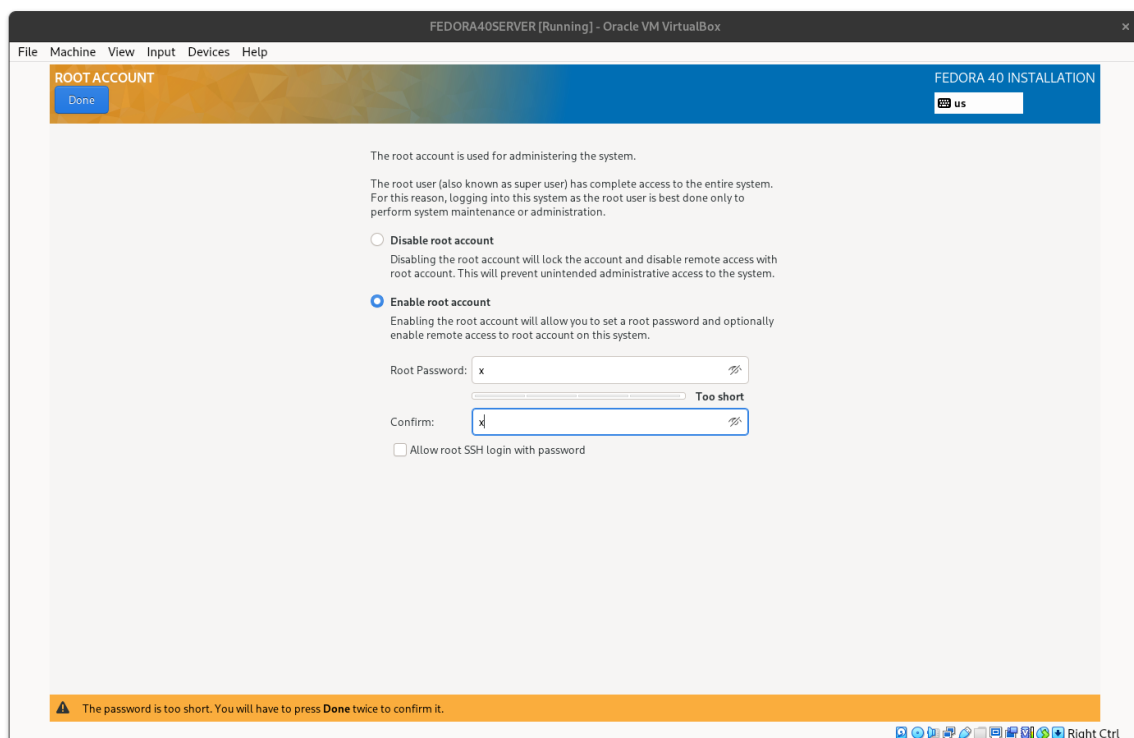
On rajoute les éléments manquants swap (20% de la RAM) et le /home puisqu'on a un utilisateur dont il faut stocker les données) :



Ensuite on valide les changements :



Je choisis personnellement d'avoir un compte root au cas où, mais il n'est pas conseillé en production.



Ensuite, on crée un utilisateur a (admin), avec pour l'exemple, le mot de passe a. Il est non-sécurisé mais facile et rapide à taper.



Une fois que tous les paramètres ont été réglés, il suffit de lancer l'installation, puis redémarrer la machine virtuelle (Begin Installation).



2.4 Réseau

Par défaut, le réseau virtuel sur lequel se trouve la machine est le NAT. Il est donc inaccessible depuis l'extérieur.

Il faut donc aller dans "Devices" → "Network" → "Network Settings", et changer le "Attached To" en "Bridged".



Lorsqu'on lance la machine, on peut voir que son adresse n'est pas 10.10.etc mais bien 192.168.etc, et que l'on est bien en bridge mode. (De plus, ma machine hôte n'a pas la même adresse.)



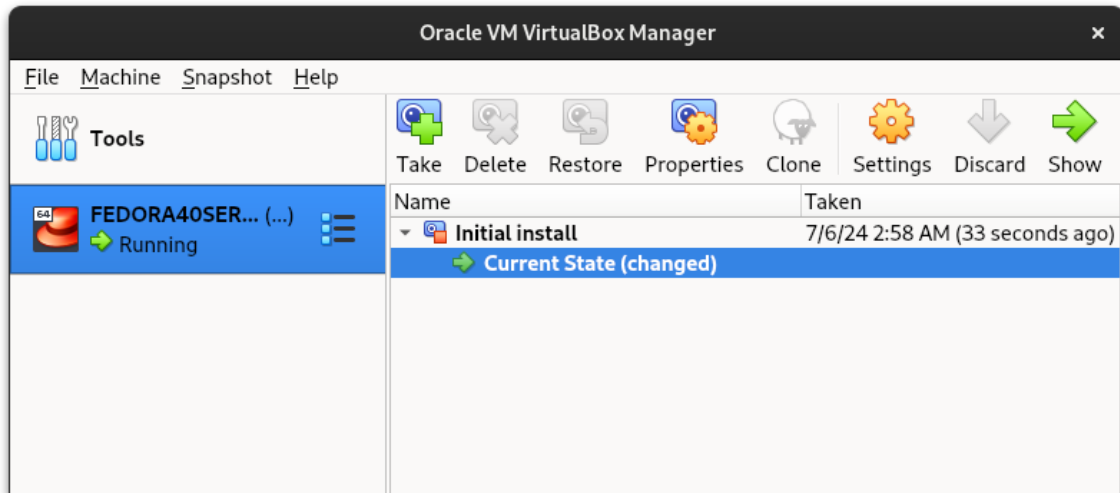
2.5 Changer le layout clavier

On utilise "localectl list-keymaps" pour avoir la liste des layouts disponibles.

Pour sélectionner un layout, par exemple, le "fr" : "localectl set-keymap fr"

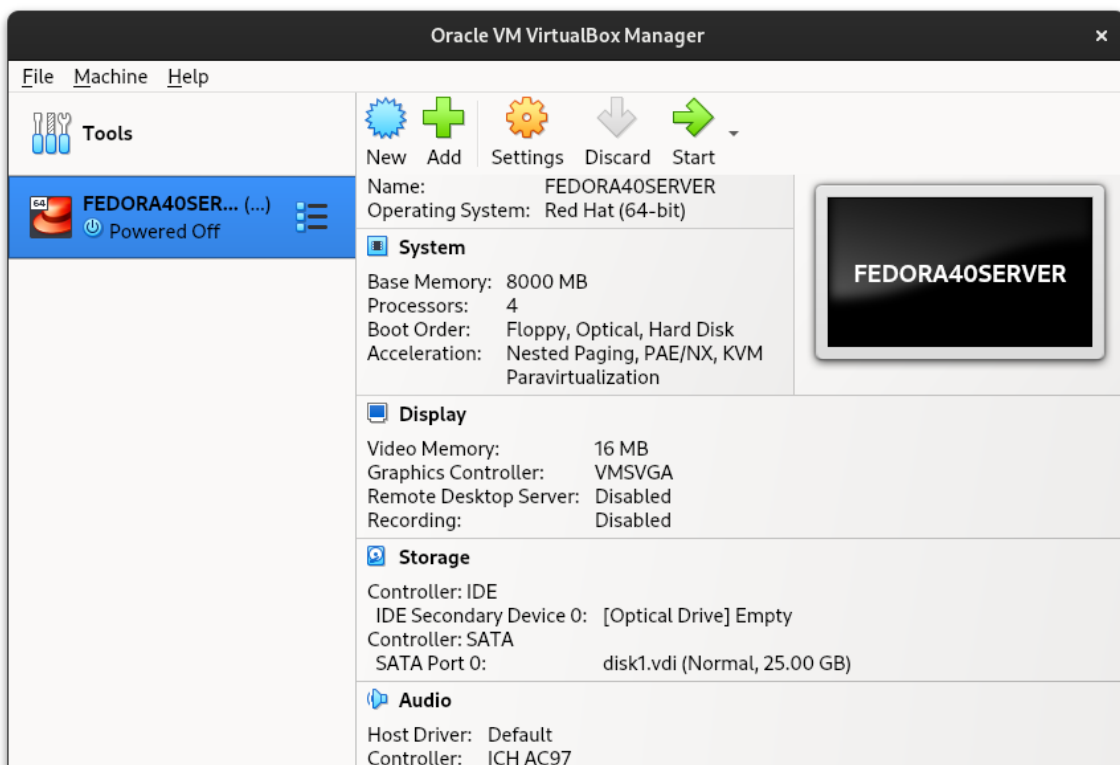
2.6 Snapshots de la machine

Pour faire une sauvegarde de l'état de la machine, il suffit de sélectionner la machine, puis de cliquer sur "Take" pour créer un snapshot.

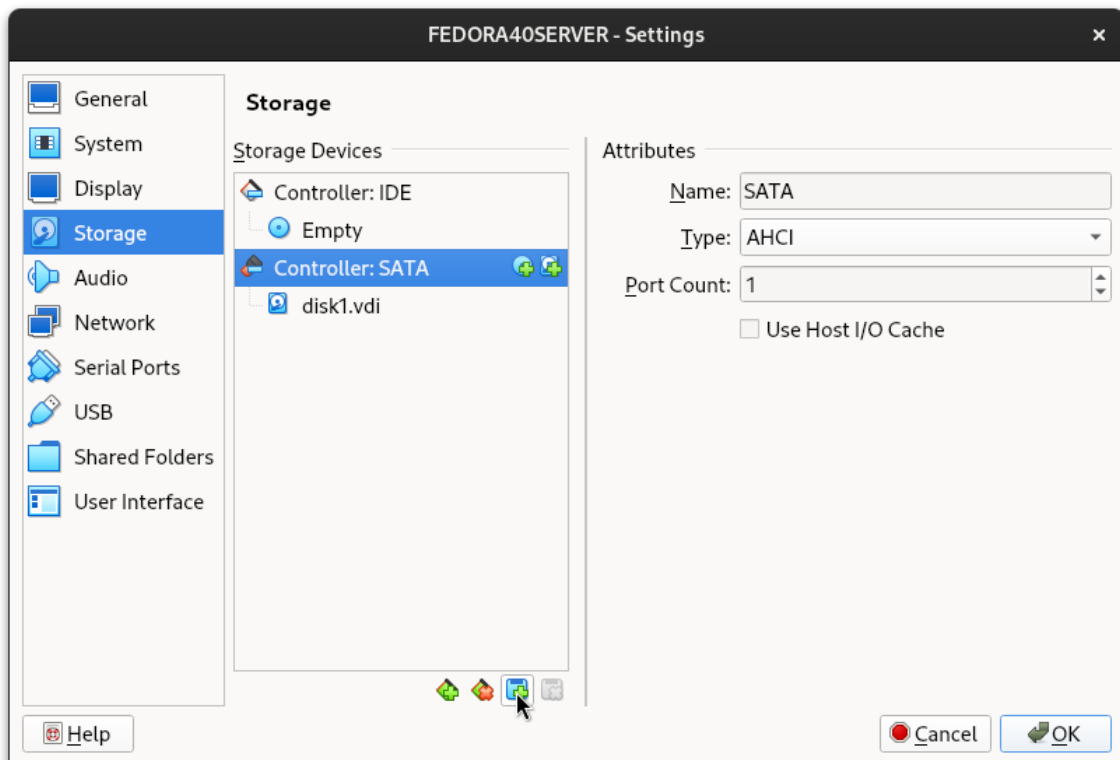


2.7 Ajout des disques virtuels

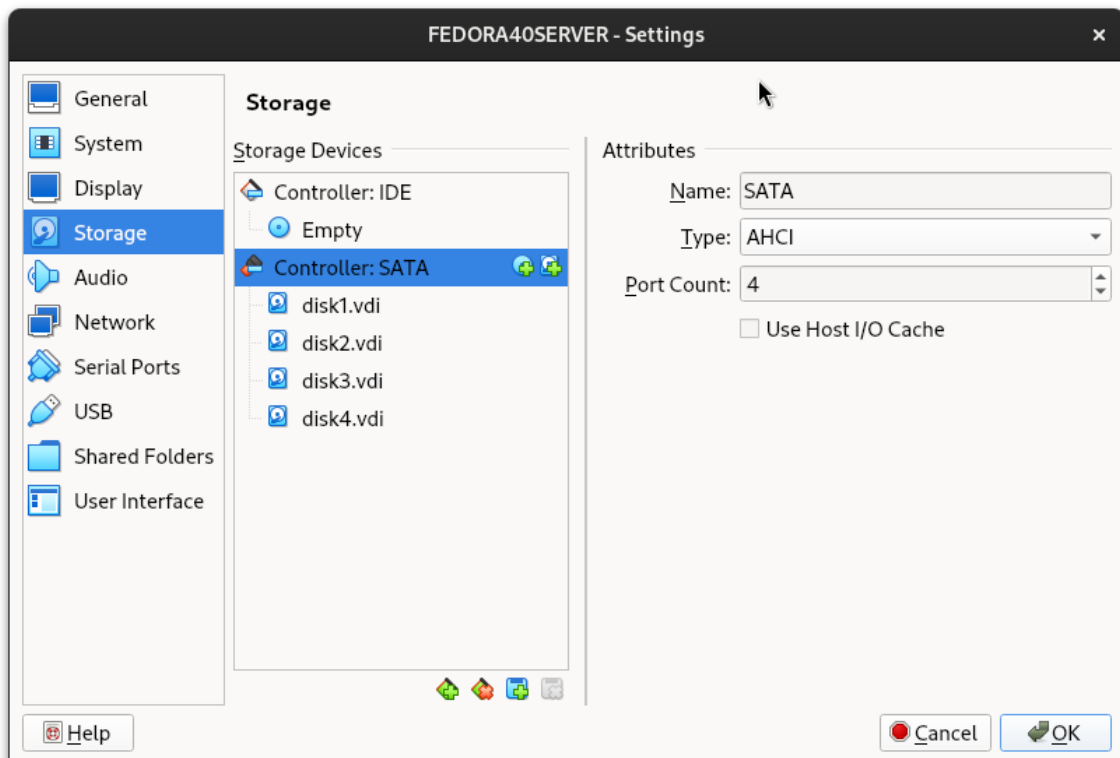
Dans les details de la machine, on selectionne "Storage" :



On va choisir "Add Attachment", "Hard Disk", puis sélectionner un à un les disques pré-crés.



And voilà ! (Je laisse le disque de backup en attente pour pouvoir identifier facilement les disques à mettre en RAID. Il suffit de répéter cette étape pour le disque de backup)



3 Description du logiciel

3.1 Clonage du repo git

git clone <https://github.com/trifoil/School-LINUX-PROJECT.git>

3.2 Menu principal

Pour lancer le script, il suffit de se rendre dans le directory School-LINUX-PROJEC, et d'exécuter le script intitulé "install.sh" en sudo.

3.3 RAID

!! A finir

Les volumes dans lesquels viennent se mettre les partages et fichiers web sont `/mnt/raid5_share` et `/mnt/raid5_web`.

3.4 Partage sans authentification

Le partage sans authentification, donc avec accès libre en écriture et lecture, doit être accessible depuis Linuw et Windows. NFS sera donc utilisé pour linux et Samba pour Windows.

Le script offre la possibilité de créer un dossier dans le chemin préétabli, ou de choisir le chemin. Il offre aussi la possibilité de créer séparément un partage NFS et un partage SMB. Il gère également les autorisations pour que SELinux autororise l'accès.

!! Ajouter limites de poids du dossier

Pour monter le directory en NFS :

```
sudo mount -t nfs 192.168.1.195:/mnt/raid5_share/unauth_share /mnt
```

Pour démonter le directory en NFS :

```
sudo umount /mnt
```

3.5 Connexion par SSH

3.6 Gestion des utilisateurs

De loin la partie la plus difficile, qui comporte plusieurs parties :

- Mise en place du DNS local
- Installation de la DB
- Installation du serveur mail
- Ajout d'un utilisateur
 - Creation de son dossier web
 - Creation de sa base de données
 - Creation de sa configuration web (http et php)
 - Creation de sa zone dans le DNS et ajout de son sous-domaine
 - Ajout du mail

3.6.1 IP fixe

!! FAUX, MODIFIER

Pour avoir un DNS fonctionnel, il est impératif d'avoir une adresse IP fixe, dans j'utiliserai donc toujours la même IP tout au long du projet. J'ai donc choisi 192.168.1.69, avec comme réseau local : 192.168.1.0/24.

3.6.2 Gestion DNS interne

On utilise le package BIND. Son fichier de configuration est `/etc/named.conf`.

Le script va donc renommer le fichier de configuration original avec un timestamp comme backup, et ajouter le nouveau fichier de configuration.

Comme on n'utilise que de l'IPv4, le script rajoute la ligne `OPTIONS="-4"` à la fin du fichier `/etc/sysconfig/named`.

Le fichier de configuration `/etc/named.conf` doit renvoyer vers deux fichiers de configuration de zones : une zone en forward et une zone en reverse.

La zone en forward contient un A record afin de pouvoir associer une IP locale avec le nom de domaine.

La zone en reverse quant à elle contient un PTR record afin d'associer le nom de domaine à l'IP choisie.

Comme nom de domaine en local, nous utiliserons `trifoil.local`.

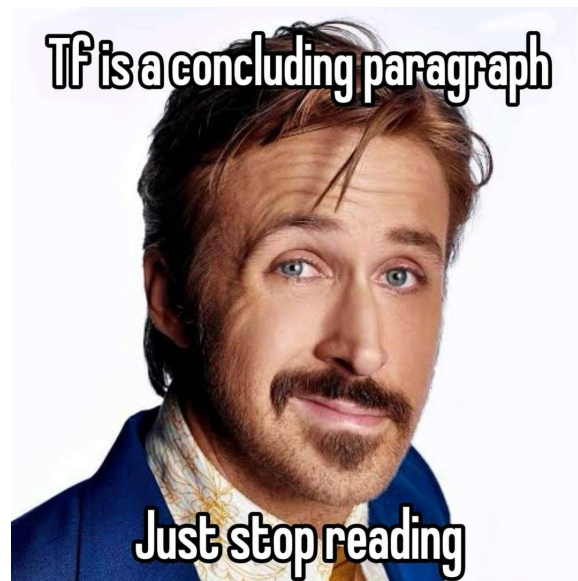
3.7 Serveur temps

3.8 Plan de sauvegarde

3.9 Sécurité

3.10 Journal de bord

4 Conclusion



References

- [1] Author, A. (Year). Title of the article. *Journal Name*, Volume(Issue), Pages.
- [2] Raid sous Oracle VirtualBox <https://youtu.be/ZHVMGfteHCg>
- [3] Configuration de Fedora Server 40 <https://www.server-world.info>

Remerciements

Remerciements à Pauline M. pour ses encouragements et son aide à la concentration :)