



# PRÉSENTATION DE PROJET

## **SYSTÈME DE FICHIERS ET DISTRIBUTION PERSONNALISÉ LINUX**



Ayoub Gorry  
Rim Agrouaz  
Youssef El Kahaloui  
Zineb Chagiri



Pr.Mohamed  
Cherradi

# SOMMAIRE

## 1 INTRODUCTION

1.1 DESCRIPTION

1.2 OUTILS

1.3 BIBLIOTHEQUES

## 2 ARCHITECTURE

2.1 DESIGN

2.2 FONCTIONNEMENT

## 3 DEMARRAGE DE RAZY-SF

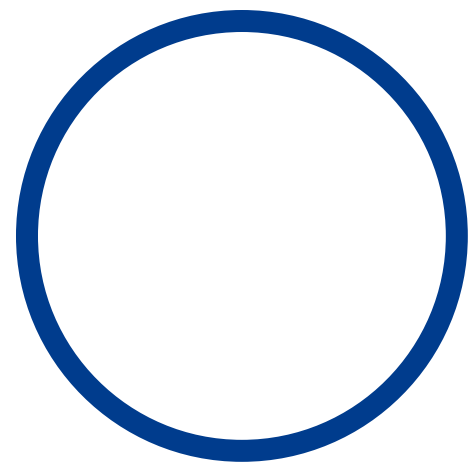
## 4 COMMANDES

## 5 GESTION DES UTILISATEURS

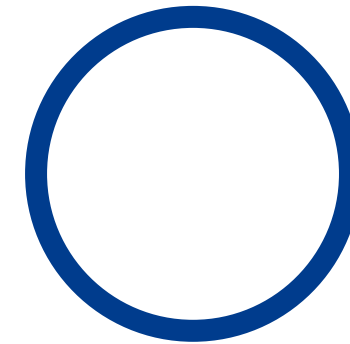
## 6 CREATION DE L'IMAGE

6.1 CHROOT

6.2 INSTALLATION SUR VM



# INTRODUCTION



# DESCRIPTION

La base de toute distribution est un système de fichier bien distribué, sécurisé . Dans cet optique, nous avons crée un système de fichier nommé "RAZY-SF" , c'est un système de fichier distribué et qui gère le stockage des données grâce à certaines bibliothèques.

Ce système de fichier se distingue des autres avec la possibilité d'utiliser ses propres commandes sans ceux prédéfinis dans le système d'exploitation Linux.

Il assure aussi la gestion des utilisateurs avec le cryptage du mot de passe afin de bien sécurisé les données.

# Les Outils:

Le langage de programmation utilisé pour coder se système de fichier est **le langage c** qui est un langage polyvalent, à simple usage, claire syntaxe, ayant une vaste base de code et différentes bibliothèques.





# Cubic

Cubic est une application GUI (Graphical User Interface) pour créer une image personnalisée en direct à base d'une image iso d'une distribution Linux.

Elle dispose d'un environnement Chroot où toutes les personnalisations sont effectuées:

- ajout des packages

- personnalisation de l'interface graphique

- insertion des commandes en utilisant git qui contient le code de toutes les commandes.

# Les Bibliothèques

## Usuelles:

**<stdio.h>** : Elle fournit des fonctions d'entrée/sortie standard, permettant la lecture et l'écriture sur la console ou dans des fichiers.

**<stdlib.h>** : Cette bibliothèque fournit des fonctions liées à la gestion de la mémoire dynamique, la conversion de chaînes en nombres (atoi(), atof()), et des fonctions pour le contrôle du programme (exit()).

**<string.h>** : contient des fonctions pour la manipulation de chaînes de caractères, y compris la copie, la concaténation, la comparaison

## Nouvelles:

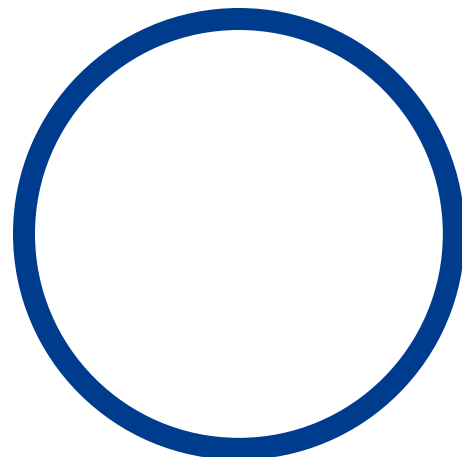
**<dirent.h>** : Elle offre des fonctions pour ouvrir, lire et fermer des répertoires, permettant la gestion des fichiers dans un système de fichiers.

**<sys/stat.h>** : Elle fournit des fonctions pour obtenir des informations sur les fichiers (comme les permissions, la taille) avec la fonction stat().

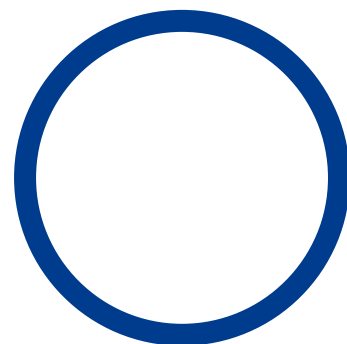
**<sys/types.h>** : fournit des définitions de types de données spécifiques à chaque système d'exploitation comme le type ssize\_t.

**<errno.h>** : définit une variable qui stocke le code d'erreur après l'exécution de certaines fonctions.

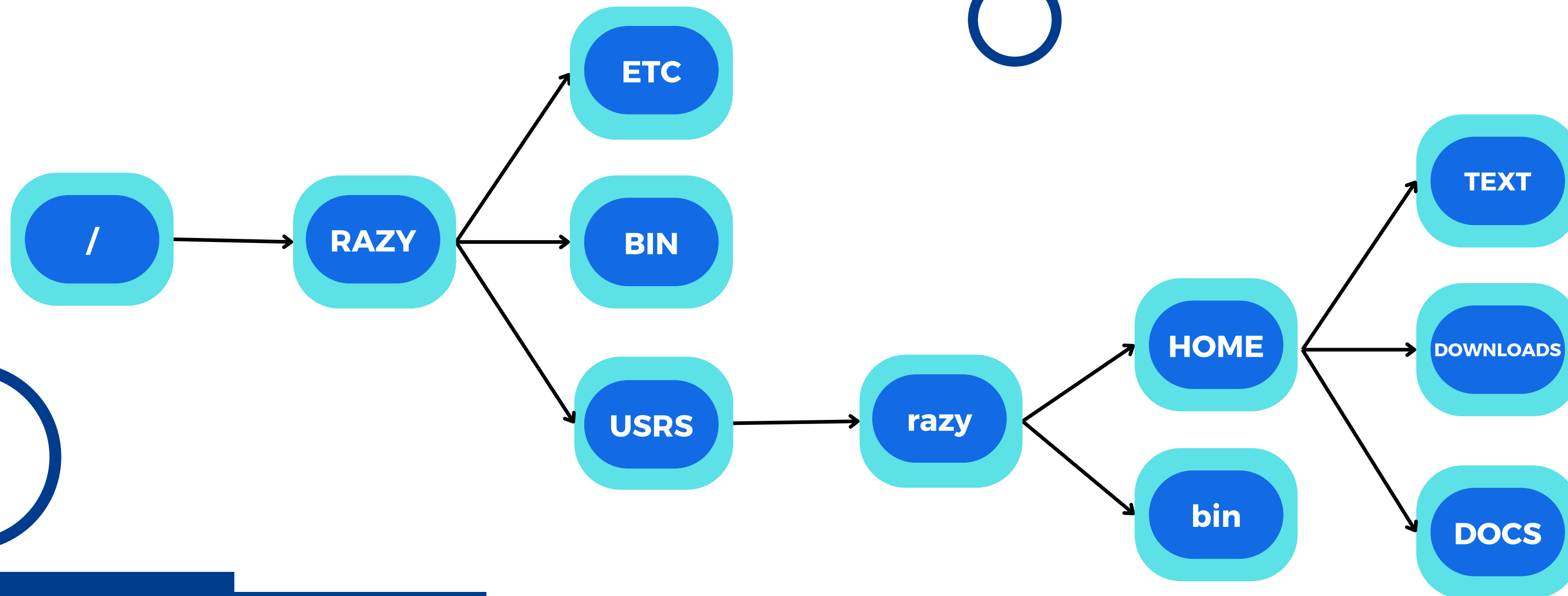




# ARCHITECTURE



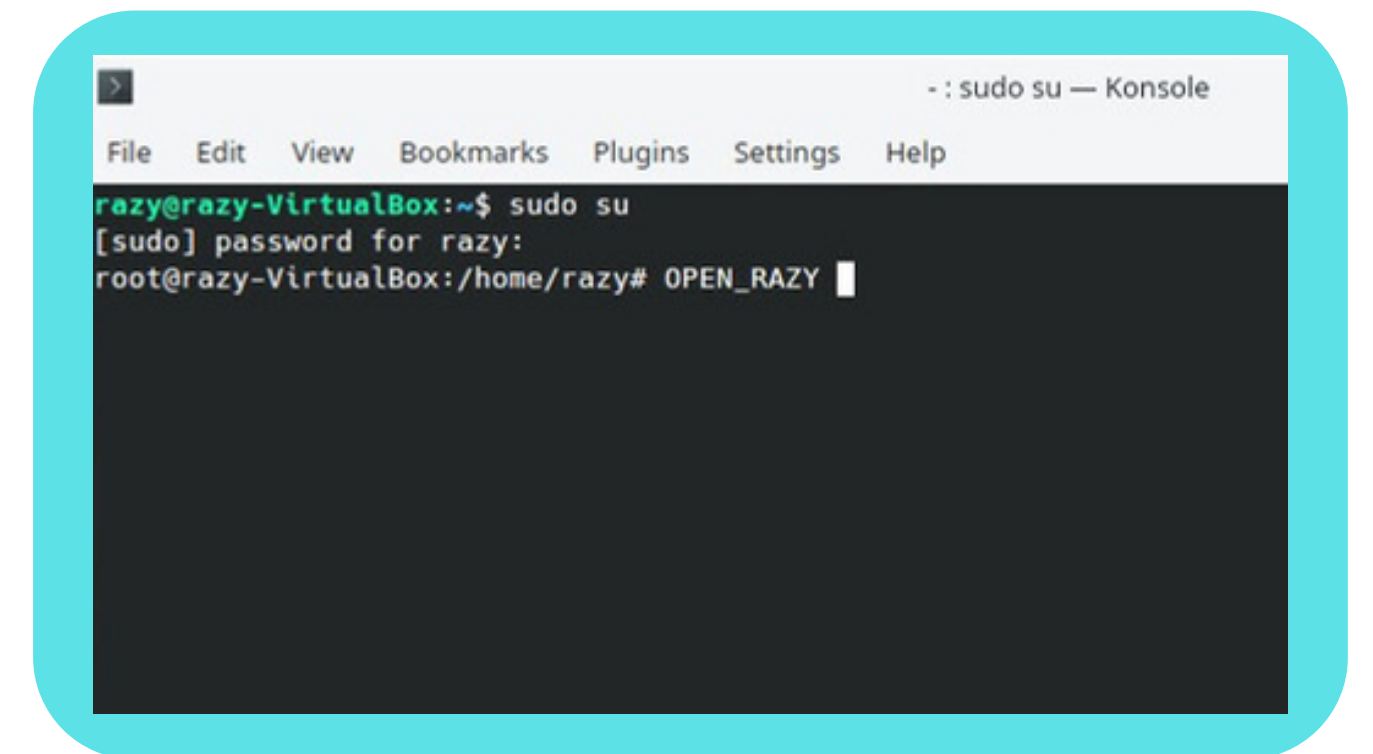
# | DESIGN



# FONCTIONNEMENT

La communication avec notre environnement peut être réalisée via le terminal.

Et puis, cela nous apparait des statistiques de notre systeme, comme vous allez voir dans la phase pratique.


A screenshot of a terminal window titled "- : sudo su — Konsole". The window has a menu bar with "File", "Edit", "View", "Bookmarks", "Plugins", "Settings", and "Help". The terminal text shows a user named "razy" at a "razy-VirtualBox" prompt running the command "sudo su". It then prompts for a password, which is entered, and the prompt changes to "root@razy-VirtualBox:/home/razy#". The command "OPEN\_RAZY" is then entered at the root prompt.

```
> - : sudo su — Konsole
File Edit View Bookmarks Plugins Settings Help
razy@razy-VirtualBox:~$ sudo su
[sudo] password for razy:
root@razy-VirtualBox:/home/razy# OPEN_RAZY
```

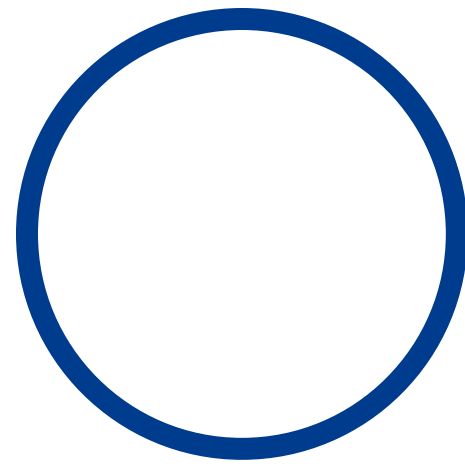


# **DEMARRAGE DE RAZY-FS**

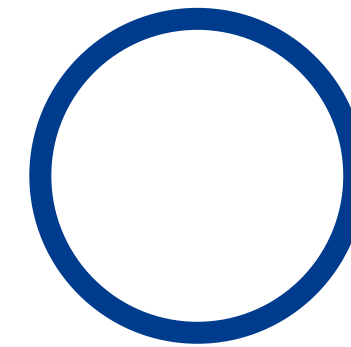




Pour assurer la communication entre l'utilisateur et notre système de fichiers on a créé un environnement personnalisé que l'on accède depuis le terminal en utilisant la commande `OPEN_RAZY` , et dans cet environnement là on peut utiliser juste les commandes inclues dans le répertoire `RAZY/bin` (nos propres commandes).



# COMMANDES



## COMMANDES

## fonctionnalité

razy\_crd

créer un répertoire

razy\_ld

lister un répertoire

razy\_rnd

rennomer un répertoire

razy\_rmd

supprimer un répertoire

razy\_crf

créer un fichier

razy\_lf

lister un fichier

razy\_rmf

supprimer un fichier

razy\_rnf

rennomer un fichier

razy\_w

écrire dans un fichier



## COMMANDES

## fonctionnalité

razy\_head

affiche un certain nombre de lignes depuis le début d'un fichier

razy\_tail

affiche un certain nombre de lignes depuis la fin d'un fichier

./add.sh

crée un utilisateur

./read.sh

login d'un utilisateur



# **GESTION DES UTILISATEURS**

# Gestion des utilisateurs:

```
RAZY : bash —
View  Bookmarks  Plugins  Settings  Help
bash ./add.sh
ame: RaZy
ord:
word forrazy:
ease enter a username matching the regular expressi
E_REGEX[_SYSTEM] configuration variable. Use the
elax this check or reconfigure NAME_REGEX.
dded successfully.
ords.txt: No such file or directory
rectory: /home/youssef/Desktop/cc/users/RaZy
lid user: 'RaZy'
home/youssef/Desktop/cc/users/RaZy created and lock
passwords.txt:
DFUDA$XaQUbp/uheKEChosHo9301
█
```

Dans le but de gestionner les utilisateurs, deux commandes sont créées:

“./add.sh” afin d’ajouter un utilisateur en cryptant son mot de passe.

“./read.sh” afin de faire le login et d’accéder aux dossiers et fichiers de cet utilisateur.

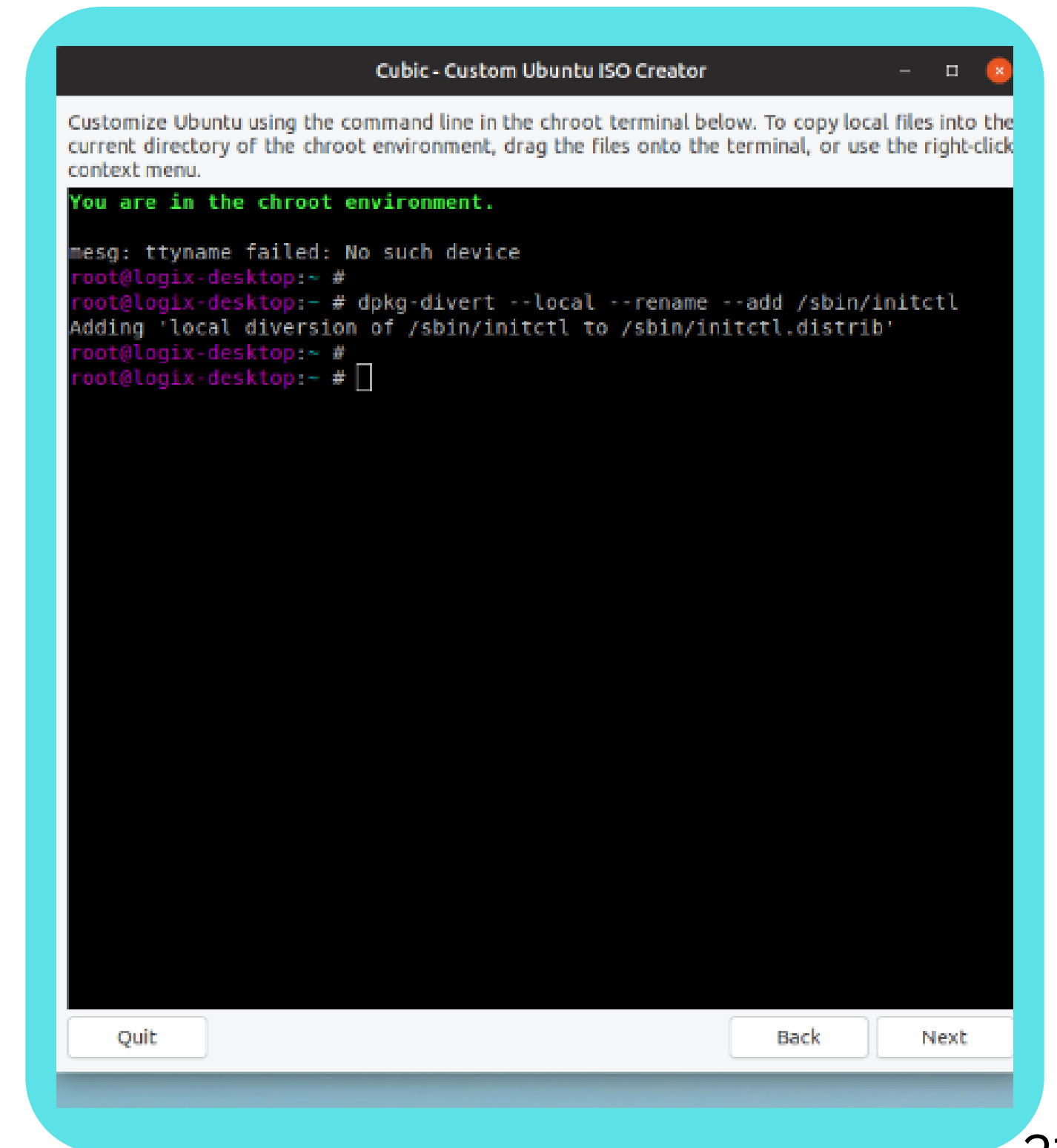
Le seul bémol rencontré est l’exécution des scripts avec la commande bash qui ne marche pas dans notre environnement d’où ce gestionnaire ne peut être utilisé qu’en dehors de l’environnement.



# **CREATION IMAGE**

## **:ISO**

# CUBIC:



# L'installation de packages

**avec apt:**

- gnome-tweaks
- dconf-editor
- nano
- vim

- git
- neofetch
- wine
- python
- gcc



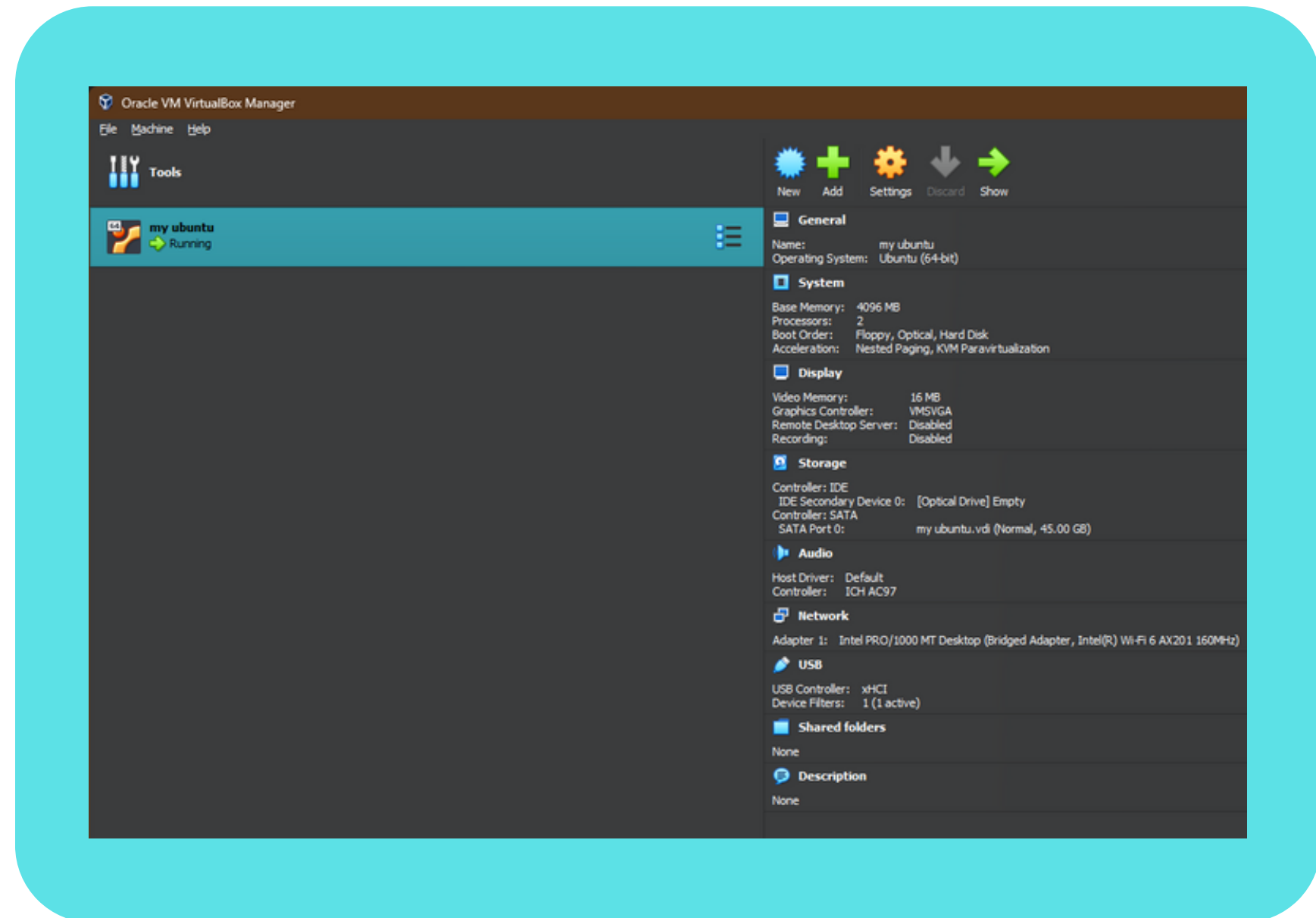
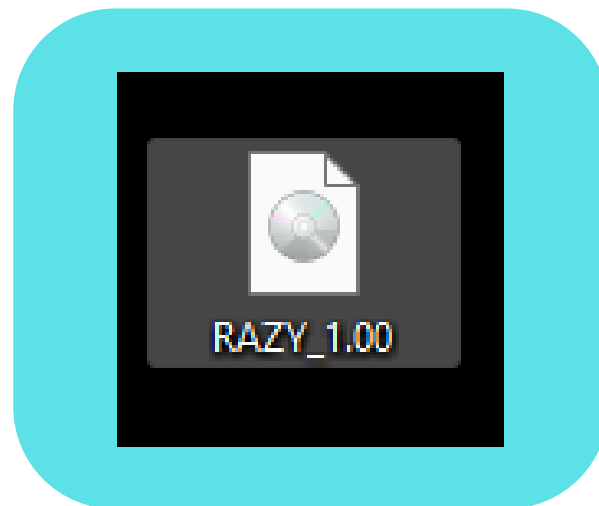
# Les modification packages:

`/usr/share/ply/mouth` : pour modifier les animations de lancement

`/home/.config/config.config` : pour personnaliser les information sur notre system

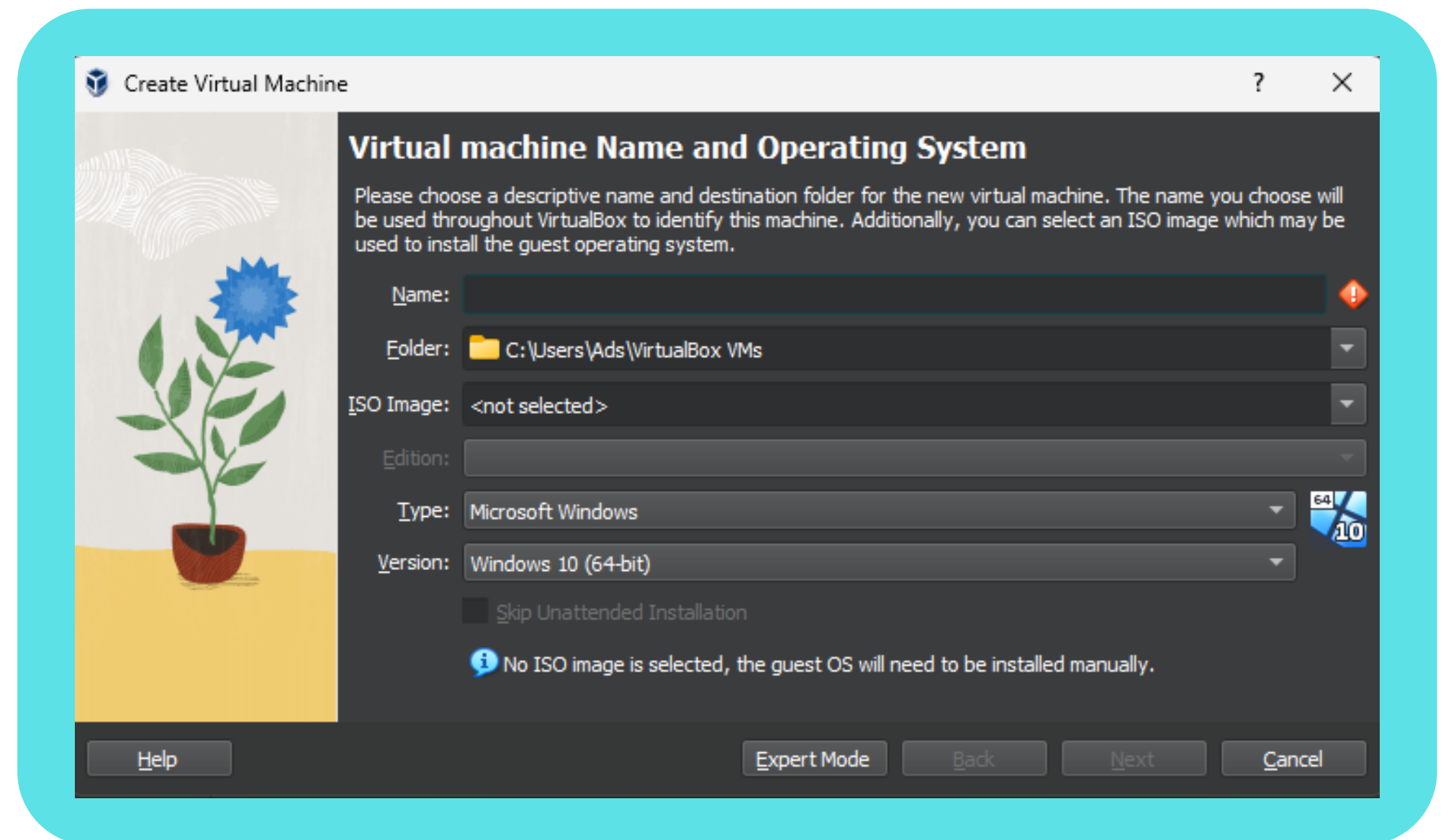
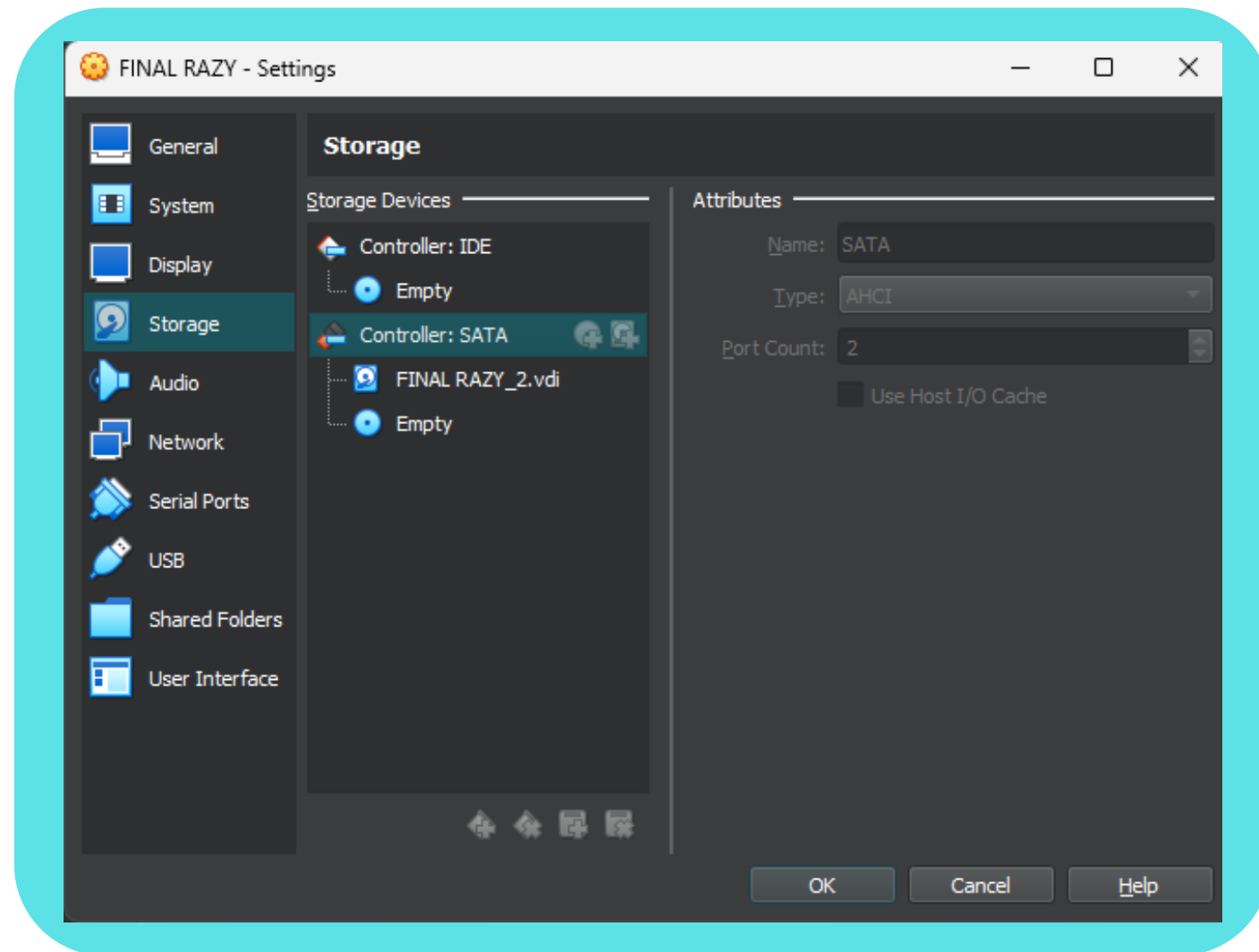
`/RAZY` : pour installer notre système de fichiers de fichiers

# l'installation sur VM :





# problèmes que vous pourriez rencontrer :





**MERCI**