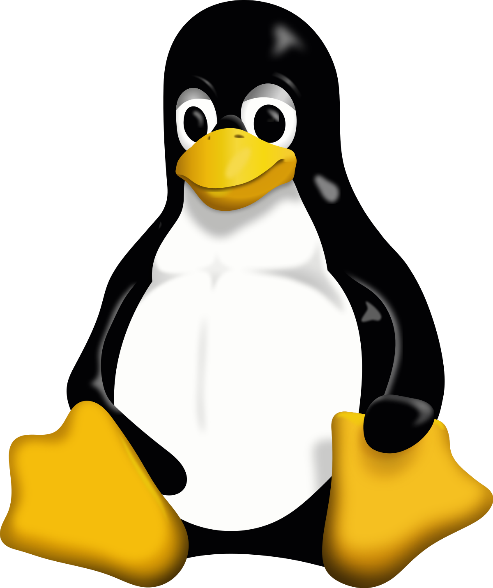
**Guide de l’utilisateur.**



Projet de Linux/UNIX

Projet LINUX Matrix Collège.

Équipe : Kevin Clovel, Bruno de Santana, Mario Camelo.

Prof : William Herrera.

Date : 24/01/2022.

INDICE

[INSTALLEZ DEUX SYSTÈMES D’EXPLOITATION LINUX SUR ORACLE VM. 3](#_Toc93572923)

[PRÉREQUIS : 3](#_Toc93572924)

[SERVEUR : 3](#_Toc93572925)

[CLIENT : 19](#_Toc93572926)

[COMMUNIQUEZ LES DEUX MACHINES À L’AIDE DE DEUX ADAPTATEURS RÉSEAU. 23](#_Toc93572927)

[CRÉEZ L’UTILISATEUR « PROJET ». 28](#_Toc93572928)

[GÉNÉREZ UNE CLÉ SSH SUR LE SERVEUR POUR L’UTILISATEUR DU PROJET. 29](#_Toc93572929)

[Copiez la clé publique sur le client. 31](#_Toc93572930)

[Validez qu`il y a bien communication entre leS MACHINES du client au serveur. 32](#_Toc93572931)

[LE SCRIPT. 34](#_Toc93572932)

[PRÉREQUIS: 34](#_Toc93572933)

[MENU SUIVANT SOUS LINUX. 36](#_Toc93572934)

[LA SCRUCTURE DU SCRIPT. 39](#_Toc93572935)

[LES CODES. 40](#_Toc93572936)

[MAIN.SH 40](#_Toc93572937)

[CREATE\_FILE.SH 43](#_Toc93572938)

[VALIDE.SH 44](#_Toc93572939)

[CKSUM.SH 45](#_Toc93572940)

[SEND.SH 46](#_Toc93572941)

# INSTALLEZ DEUX SYSTÈMES D’EXPLOITATION LINUX SUR ORACLE VM.

## PRÉREQUIS :

Télécharger et installé le « ORACLE VM VIRTUALBOX »

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

Télécharger l’IMAGE ISO Ubuntu 20.04

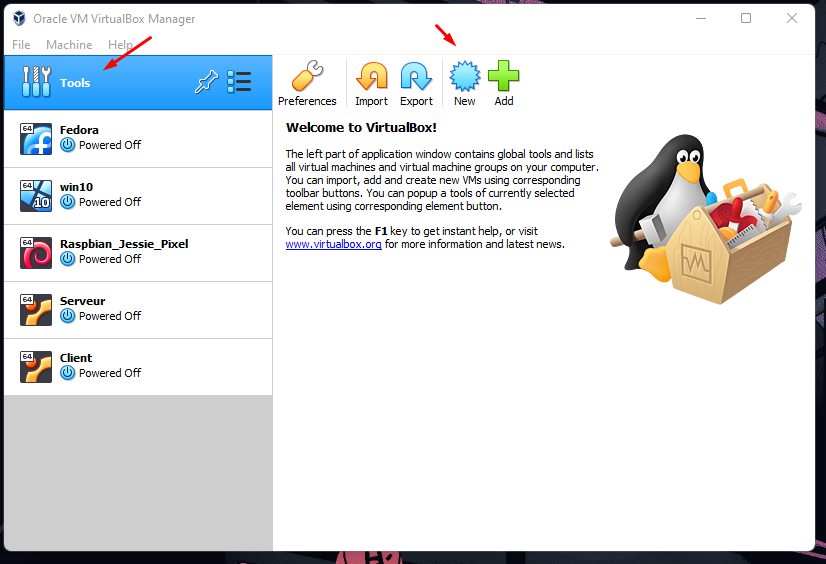
Lien <https://ubuntu.com/download/>

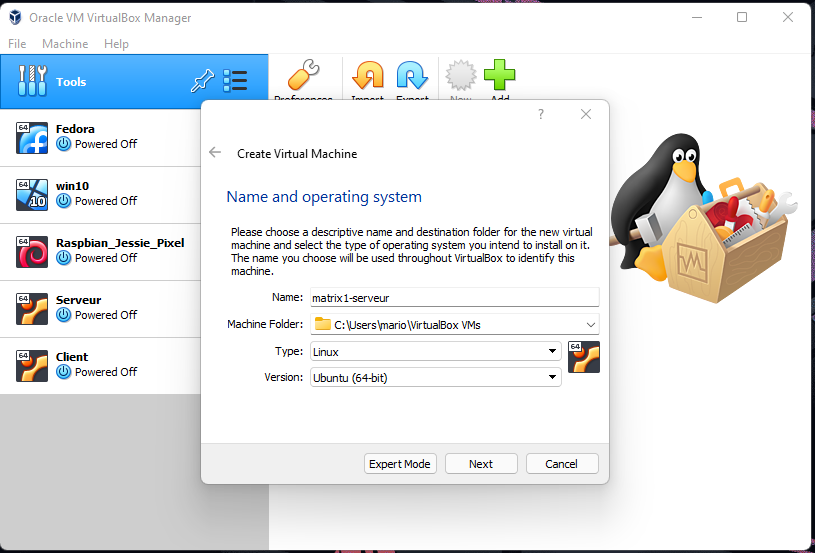
* Serveur Linux – Ubuntu
* Nom : matrix1-serveur
* Matériel : 3Gb RAM ; Disque 20GB
* Client Linux – Ubuntu
* Nom : matrix1-client
* Matériel : 1Gb RAM ; Disque 10GB

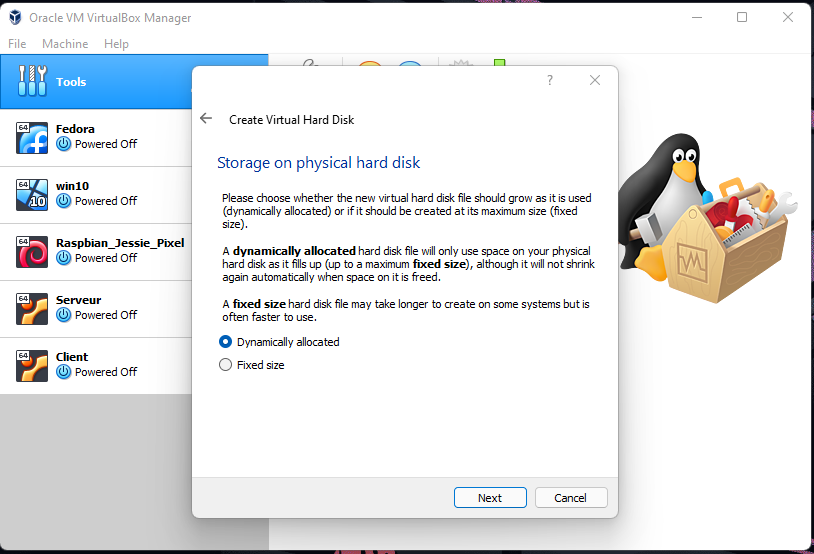
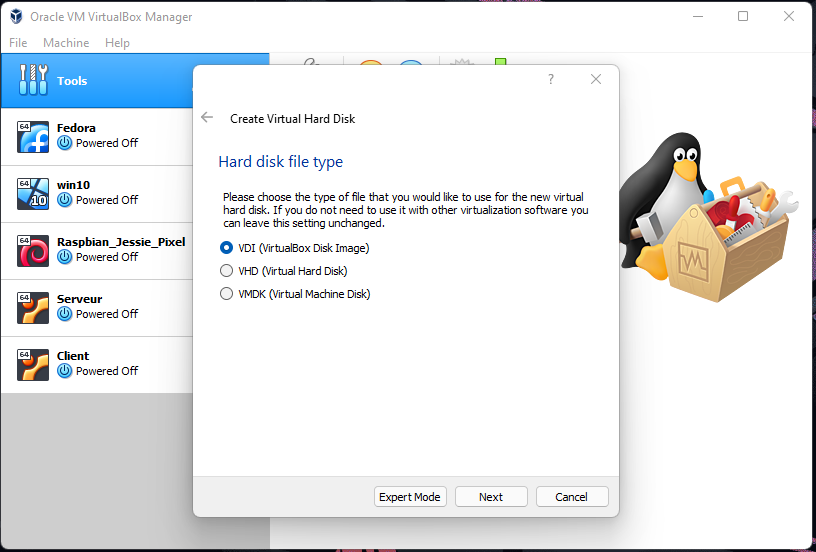
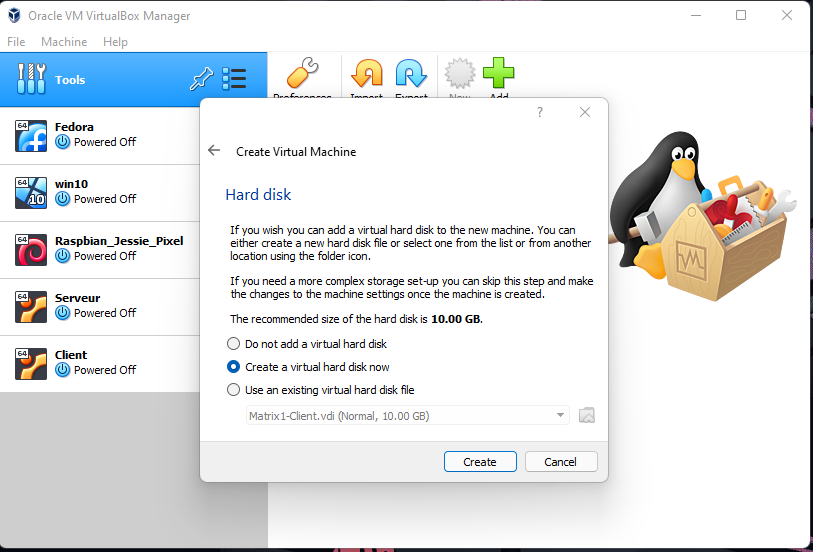
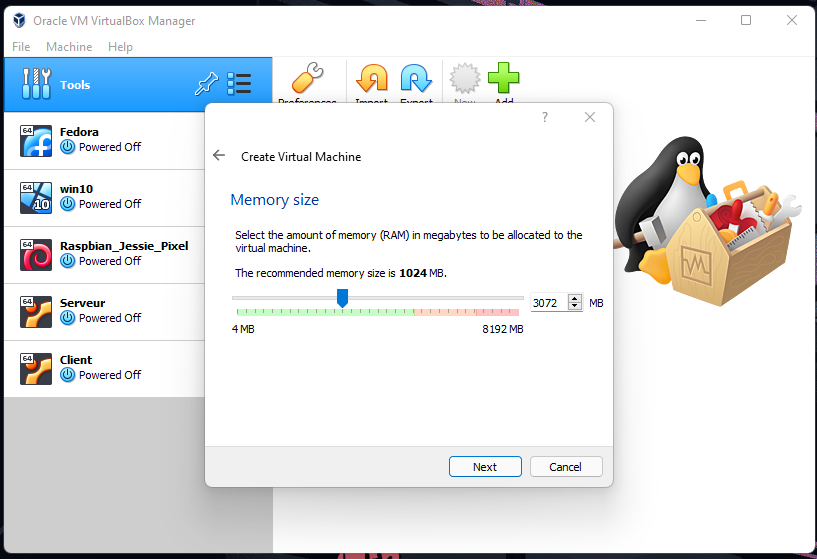
### SERVEUR :

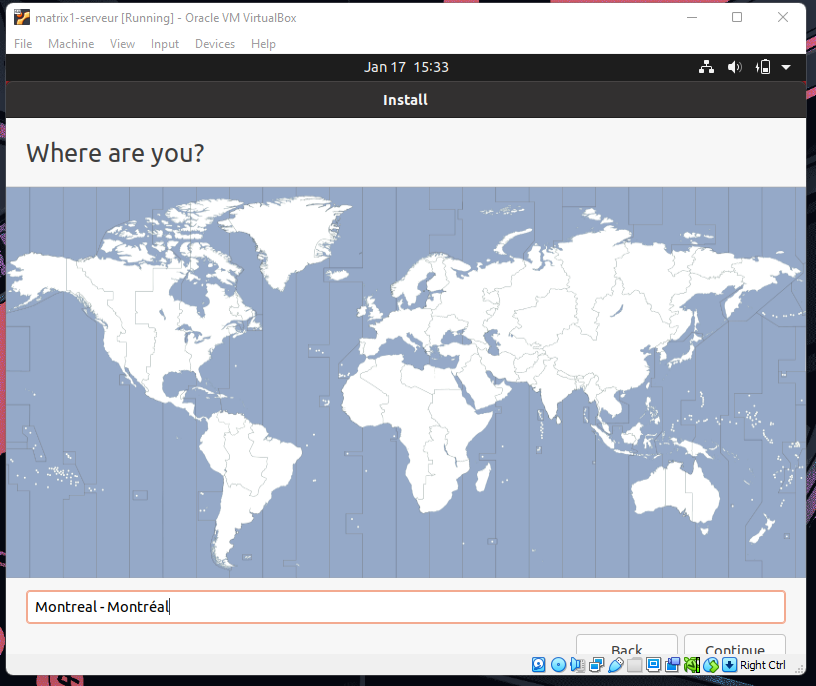
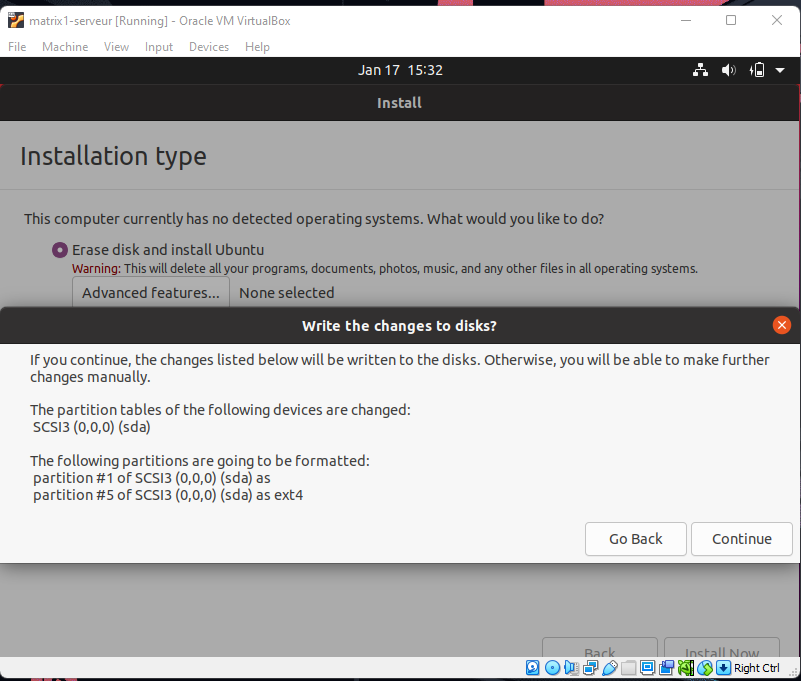
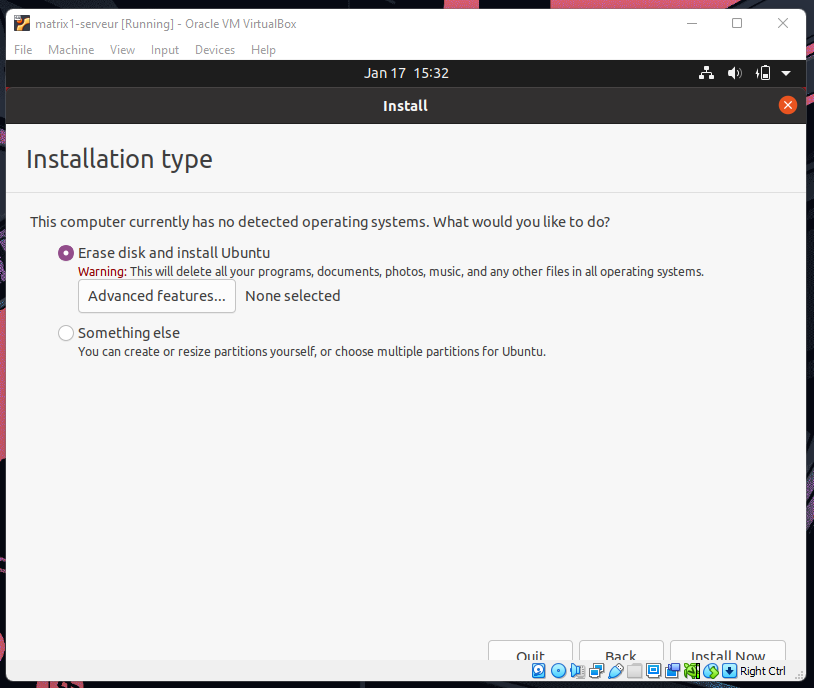
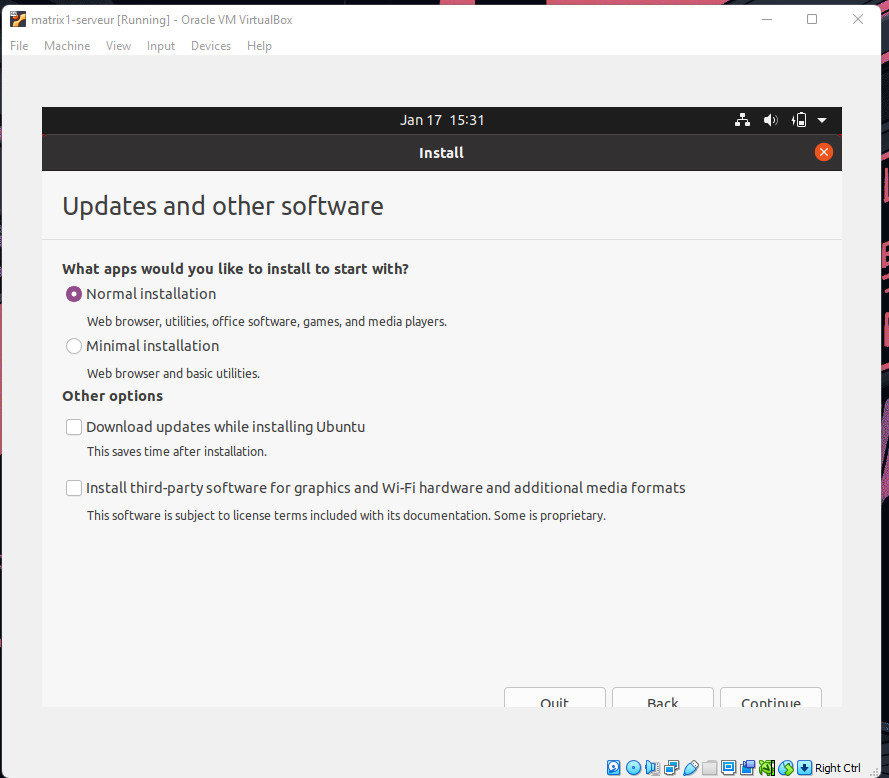
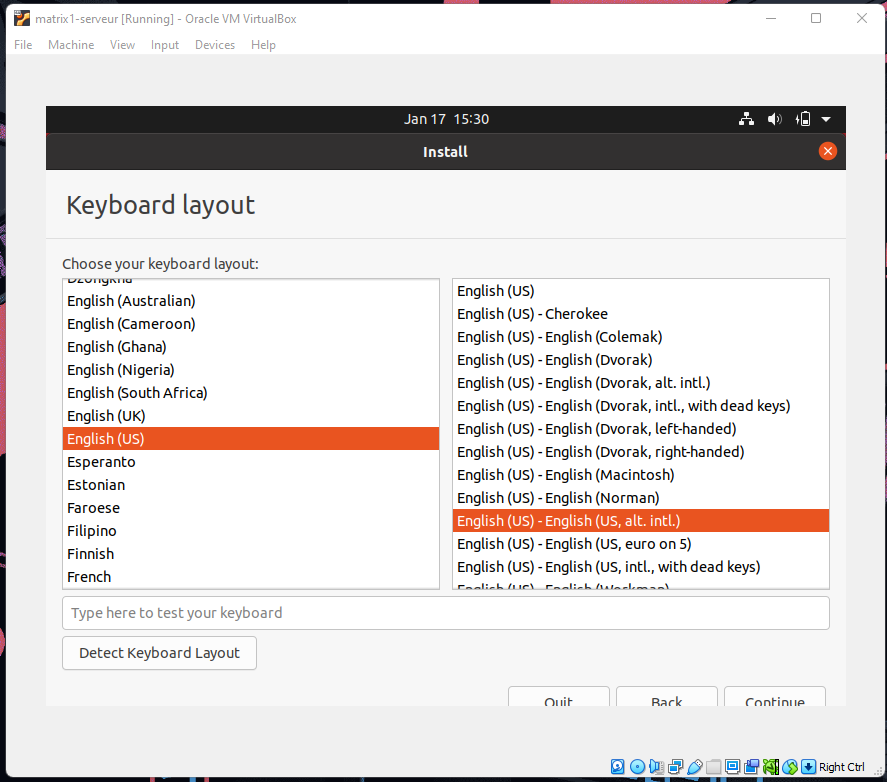
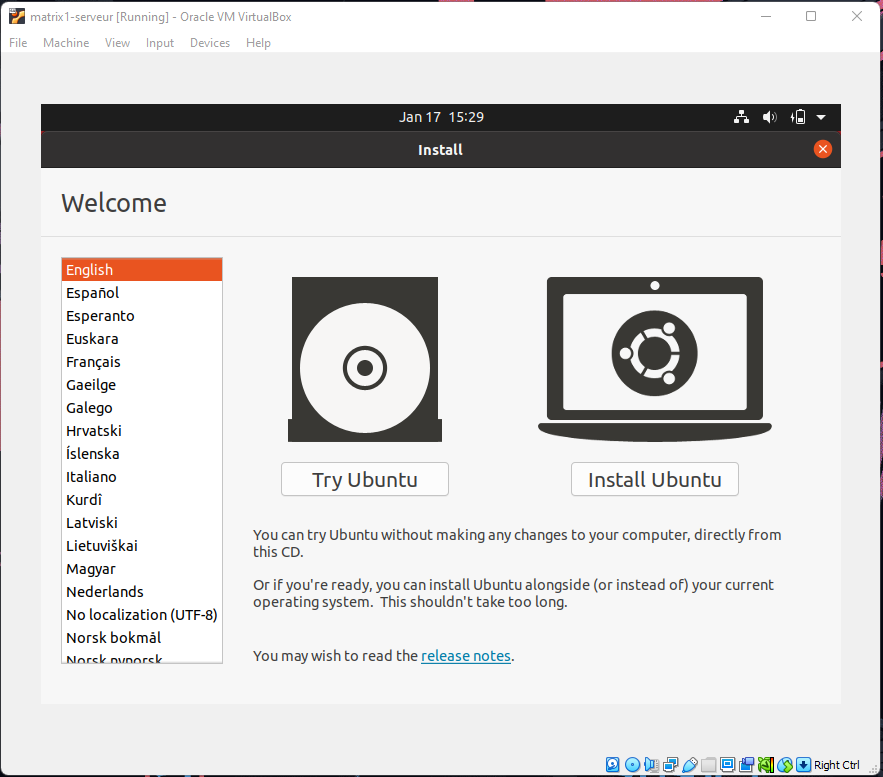
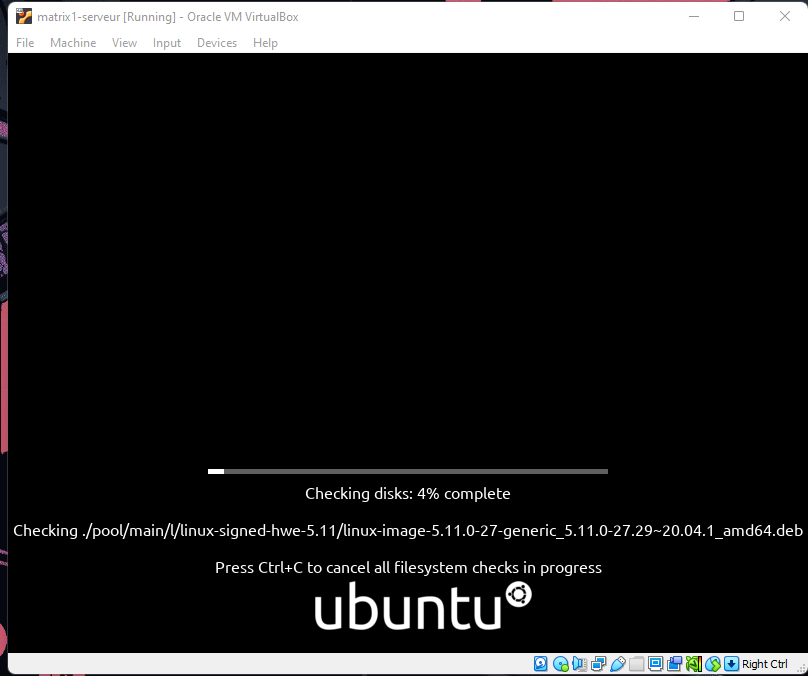
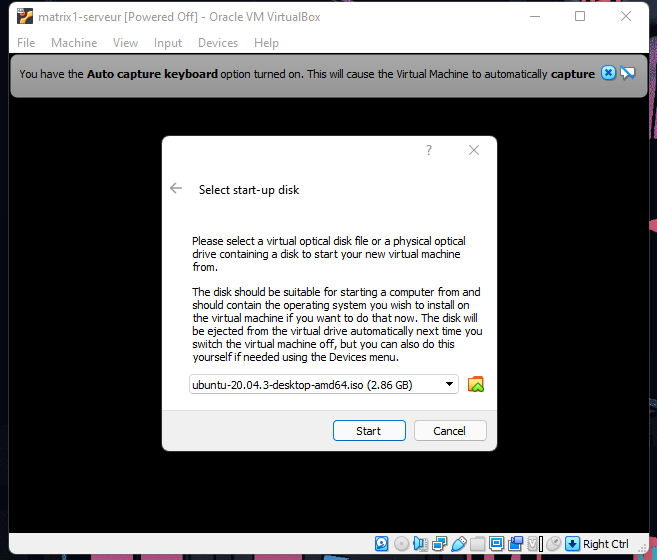
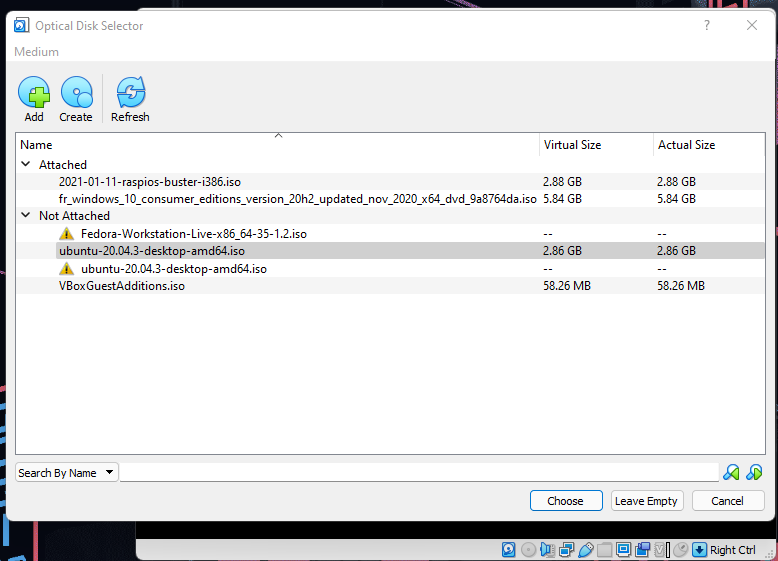
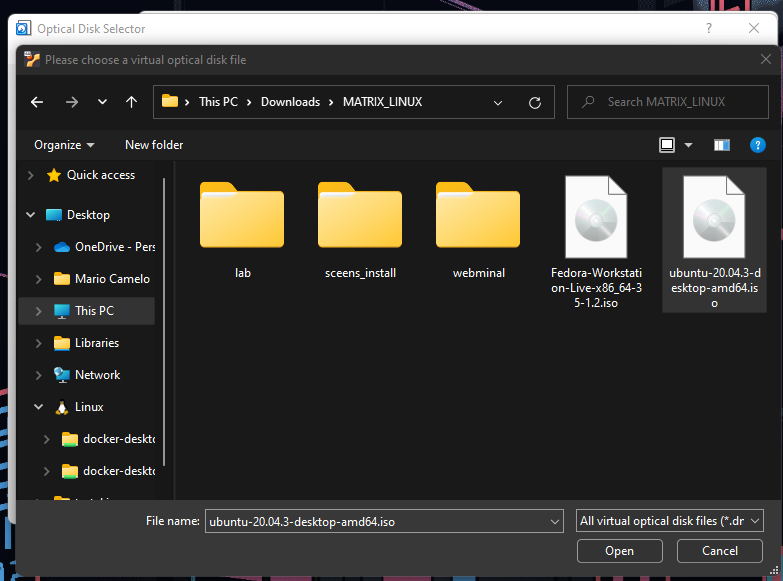
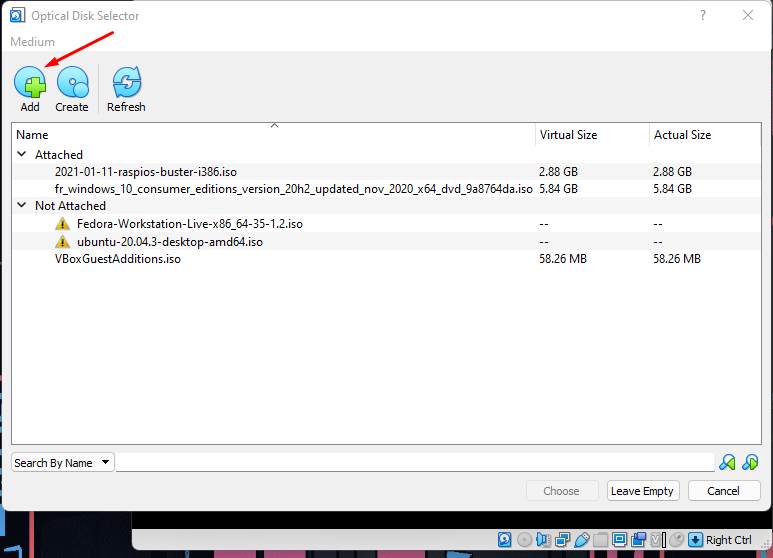
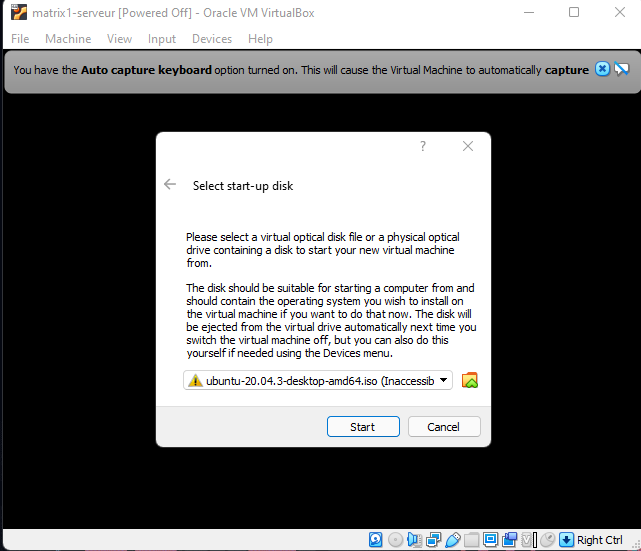
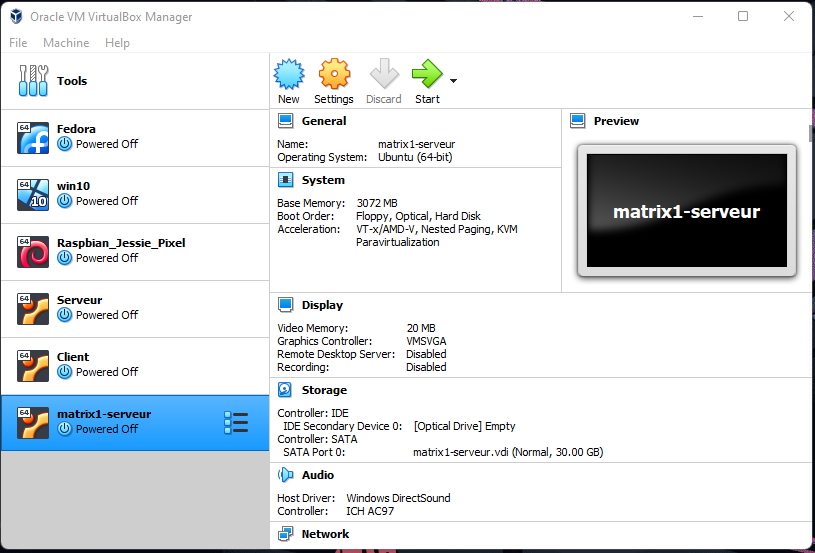
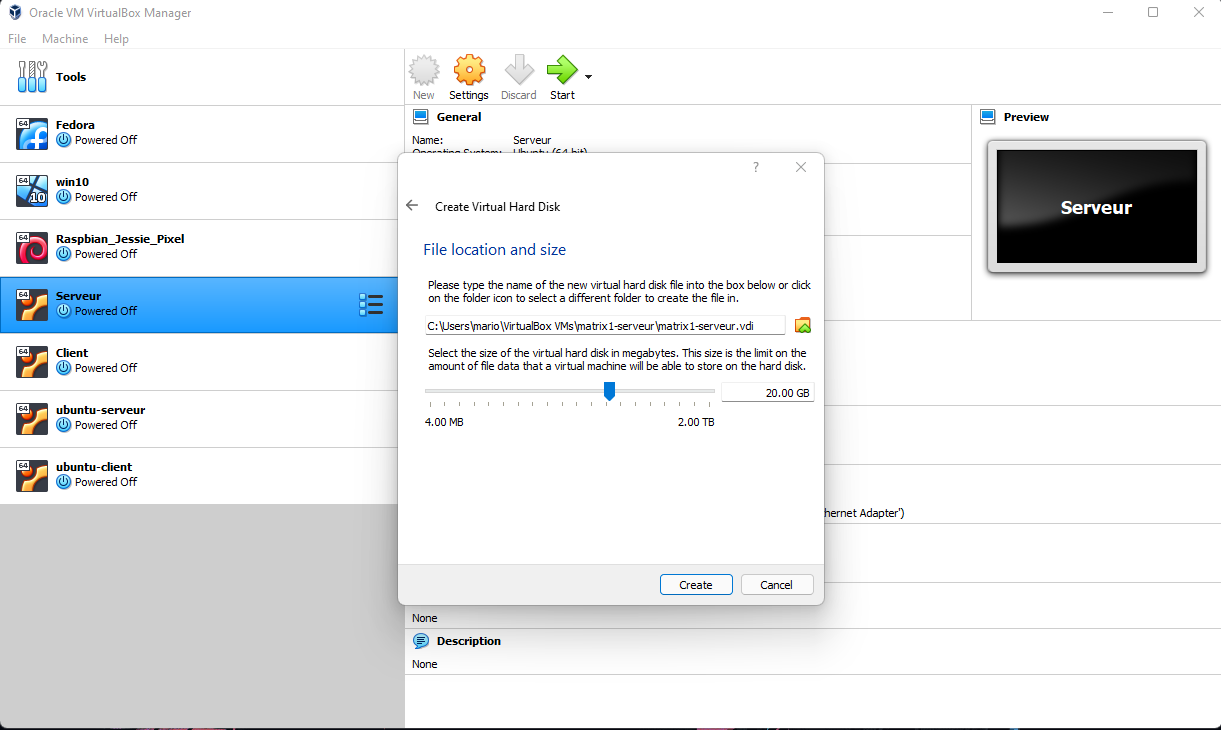
Ici on vas faire le même procediment du « serveur », choisissez les propriétés que nous avez été demandé, comme le nom de la machine, mémoire ram, mémoire du disque, le nom de l’utilisateur.

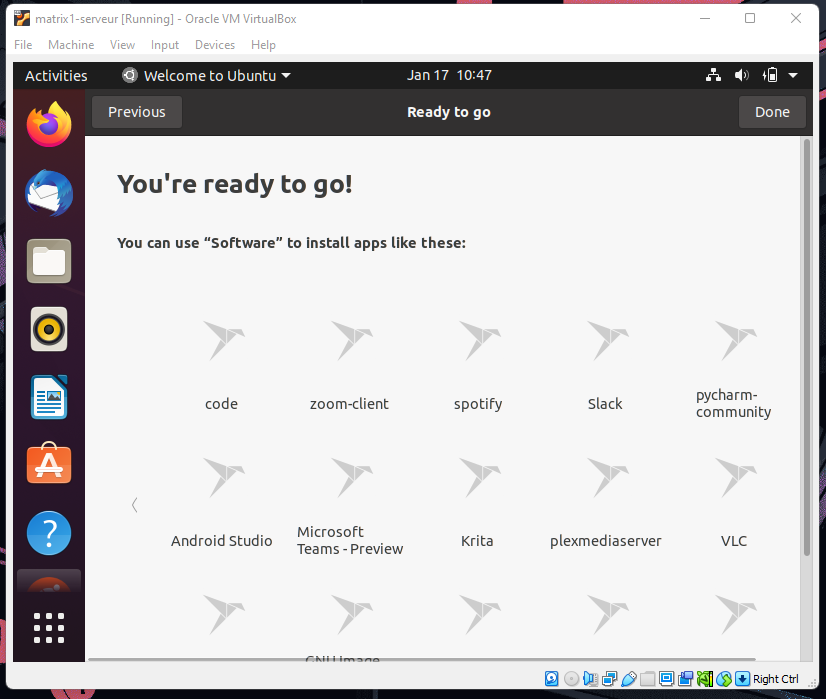
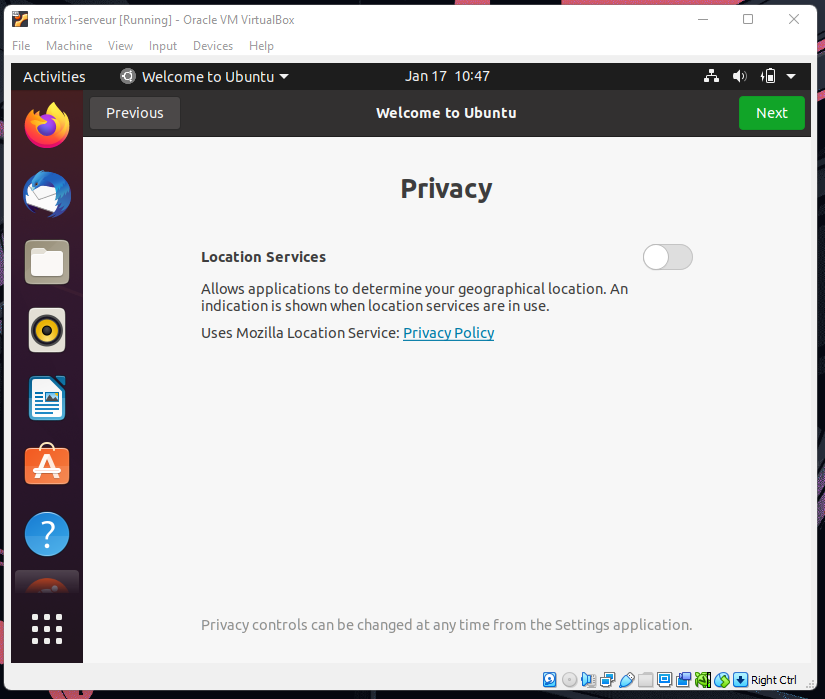
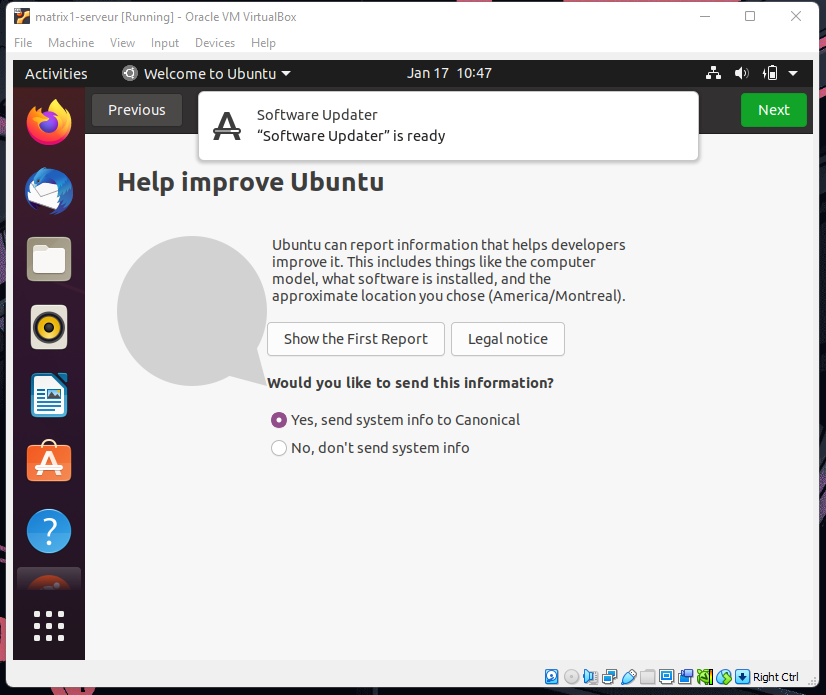
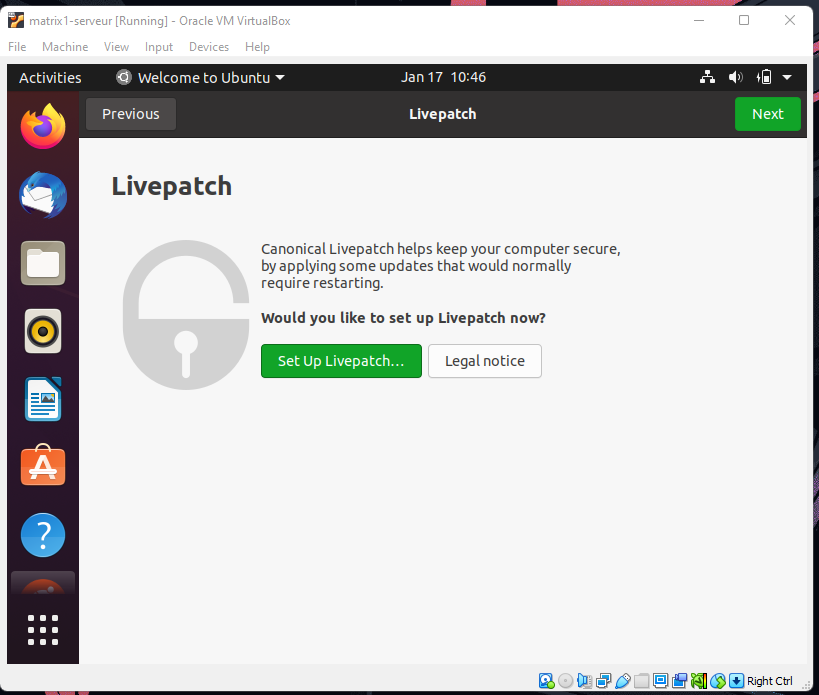
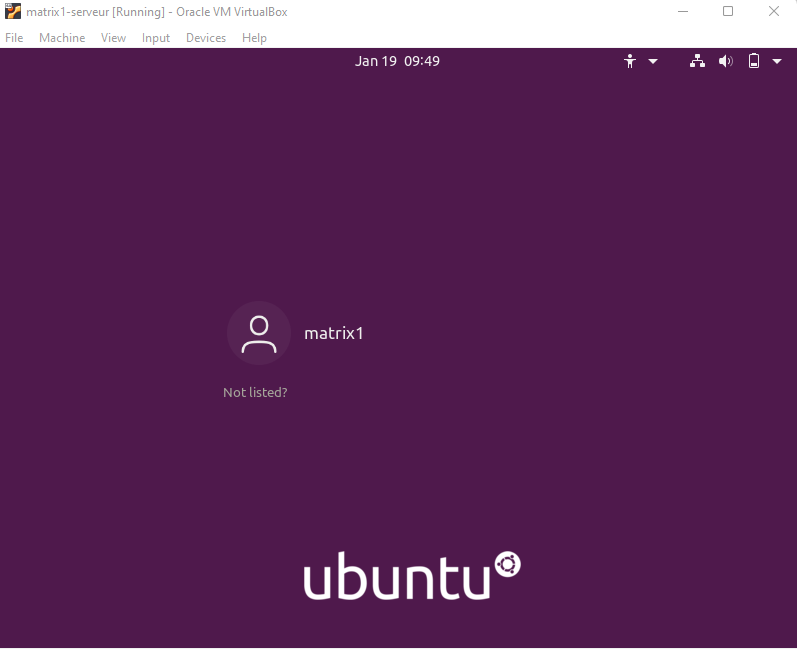
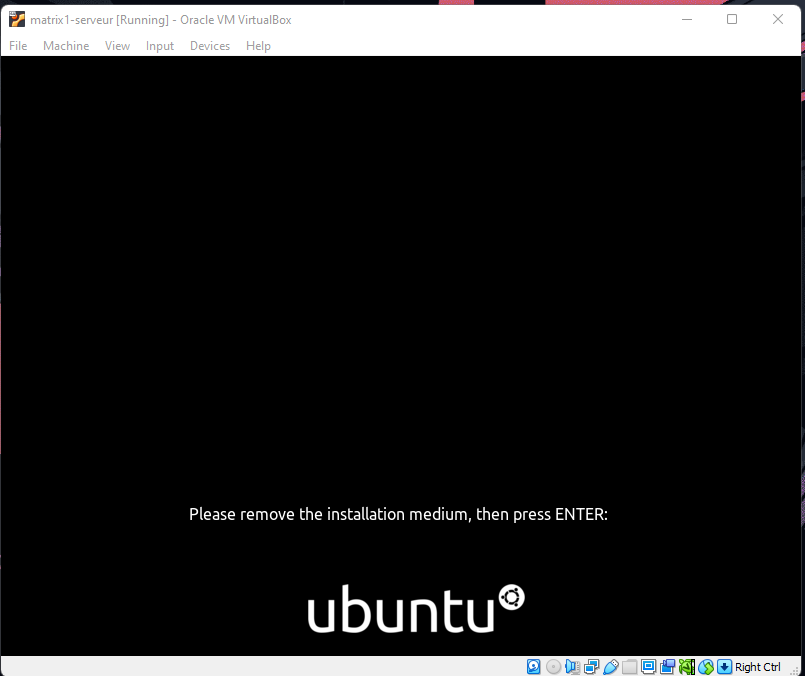
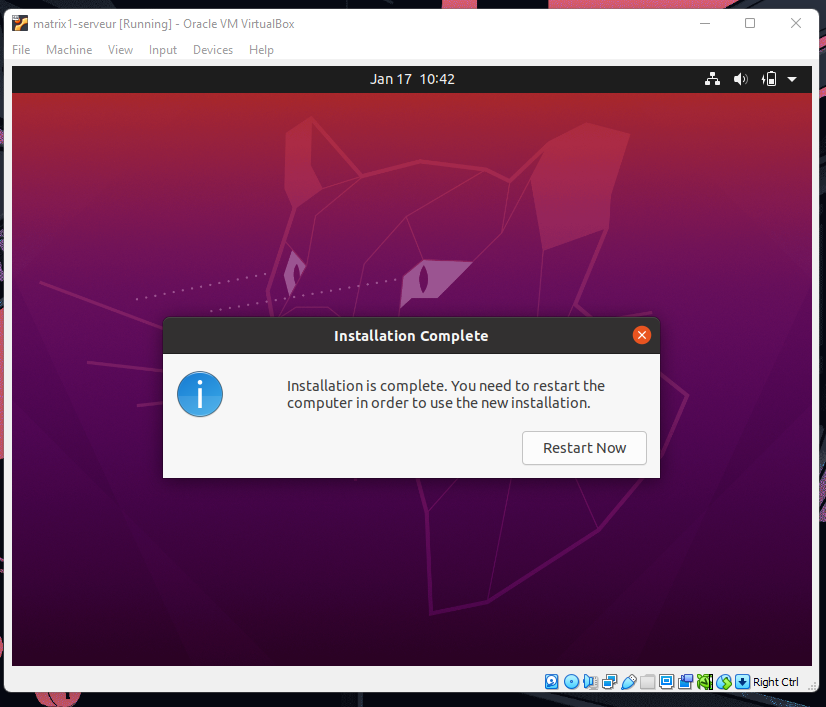
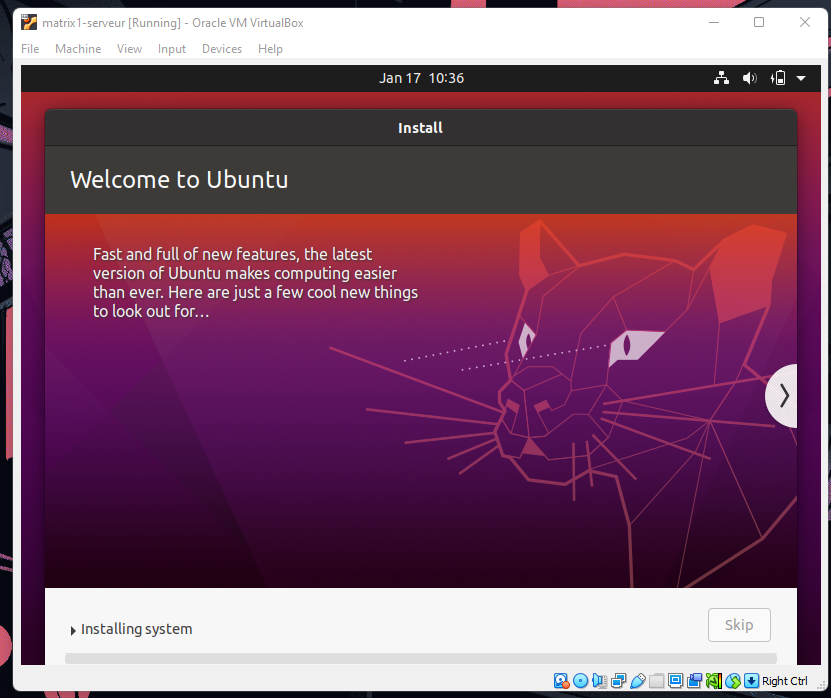
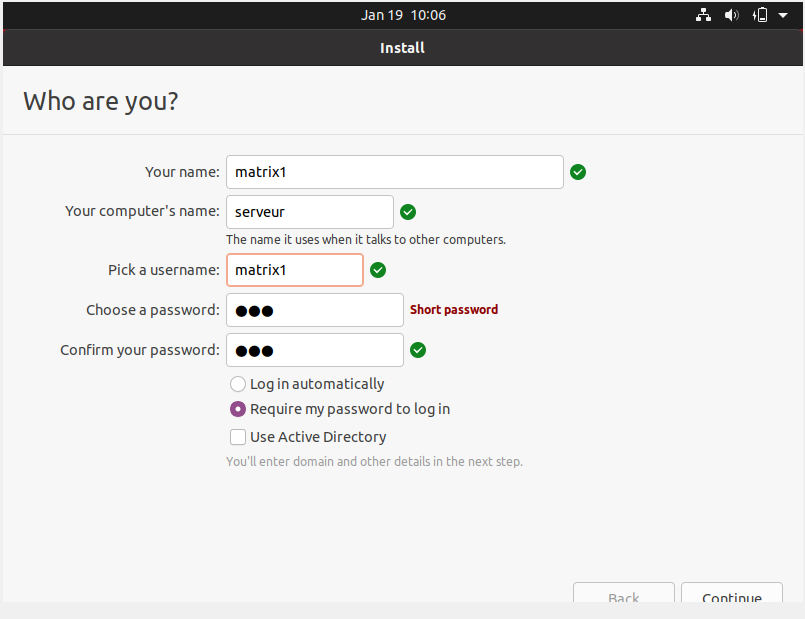
* Serveur Linux – Ubuntu
* Nom : matrix1-serveur
* Matériel : 3Gb RAM ; Disque 20GB





Choisissez 3GB de mémoire RAM, pour le machine virtuelle « matrix1-serveur ».

Choisissez 20.GB de mémoire DISK, pour le machine virtuelle « matrix1-serveur ».



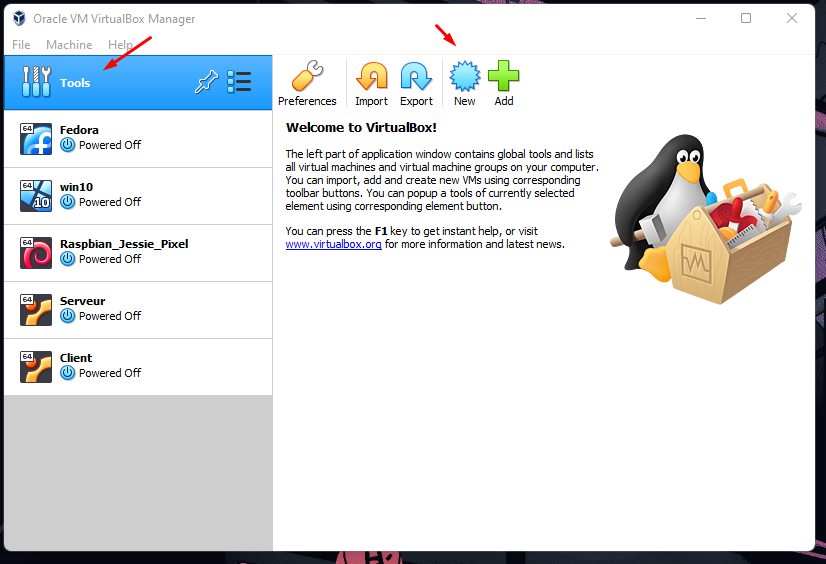
### CLIENT :

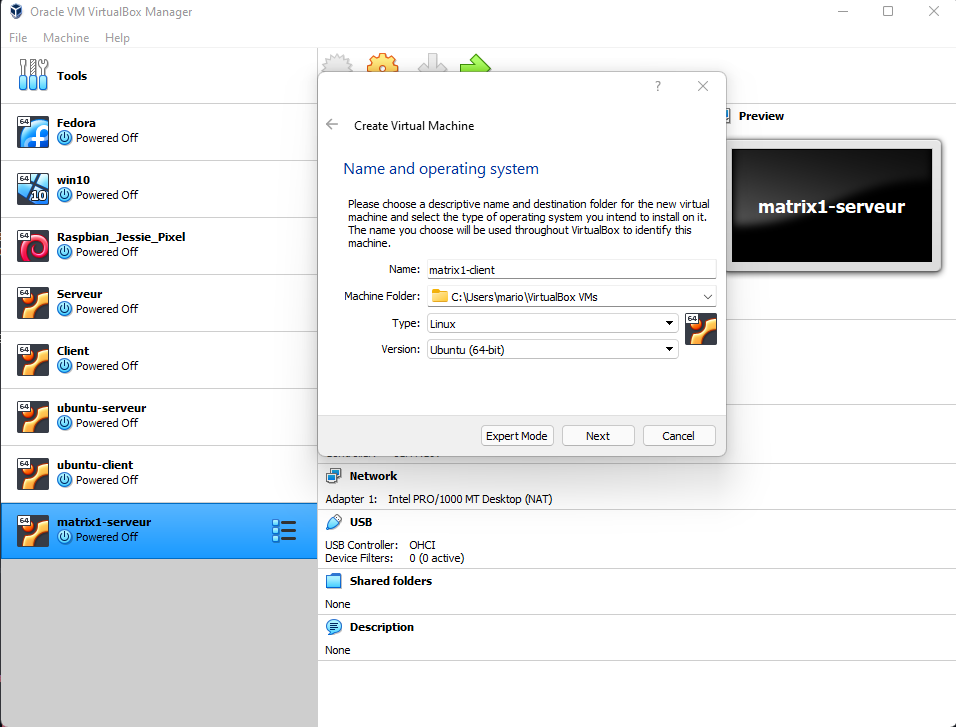
Ici on vas faire le même procediment du « serveur », mais maintenant choisissez les propriétés du « client » que nous avez été demandé, comme le nom de la machine, mémoire ram, mémoire du disque, le nom de l’utilisateur.

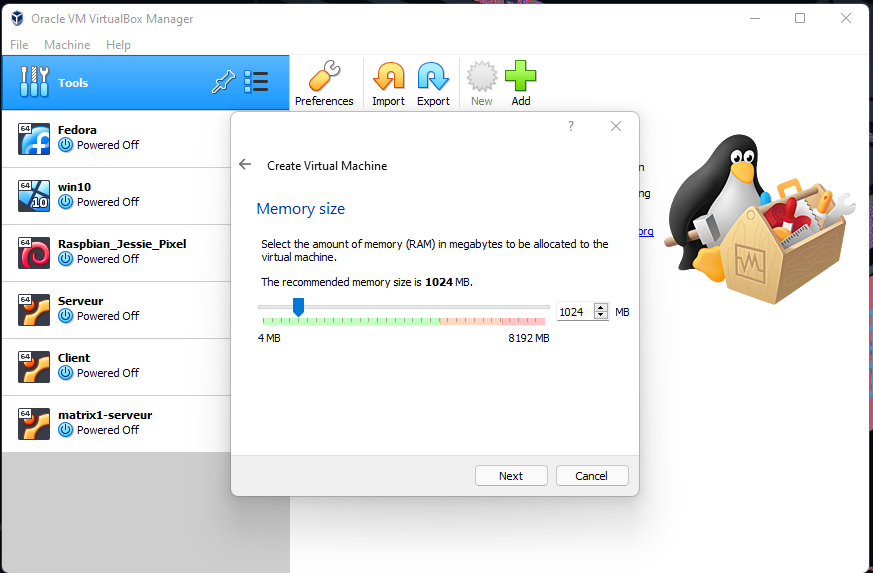
Client Linux – Ubuntu

Nom : matrix1-client

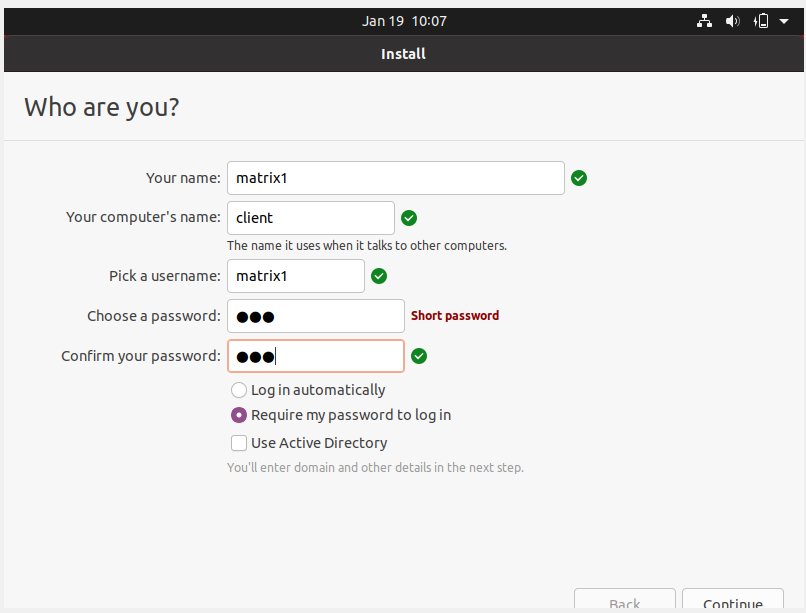
Matériel : 1Gb RAM ; Disque 10GB



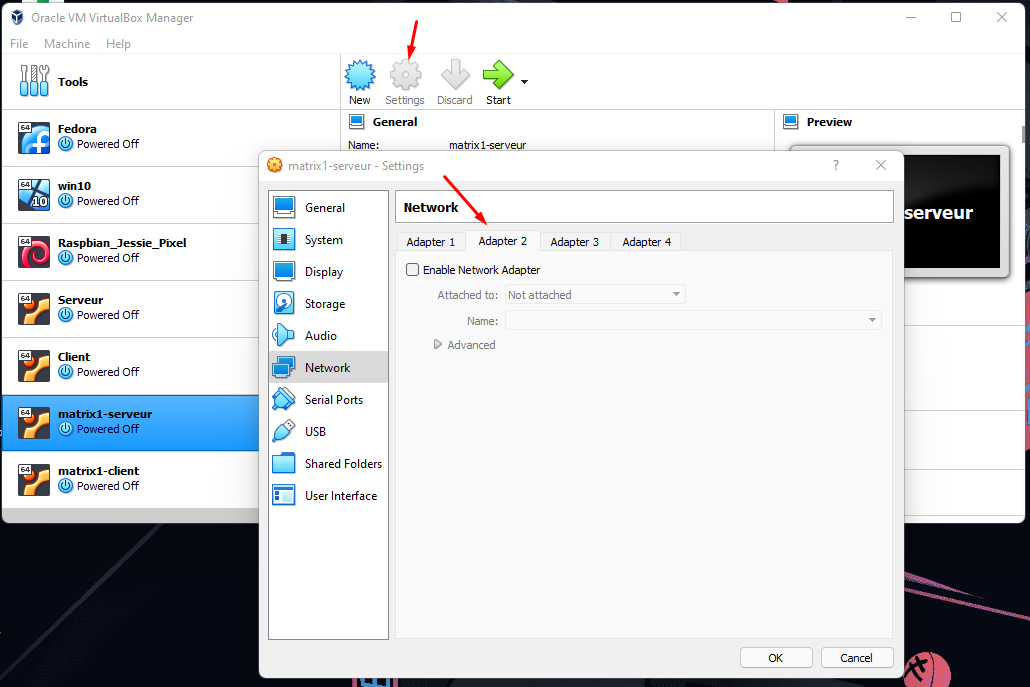
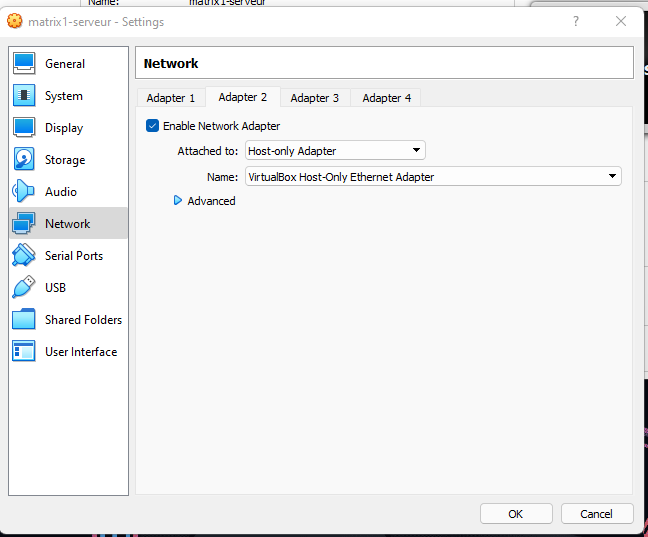


Choisissez 1GB de mémoire RAM, pour le virtual machine matrix1-client.

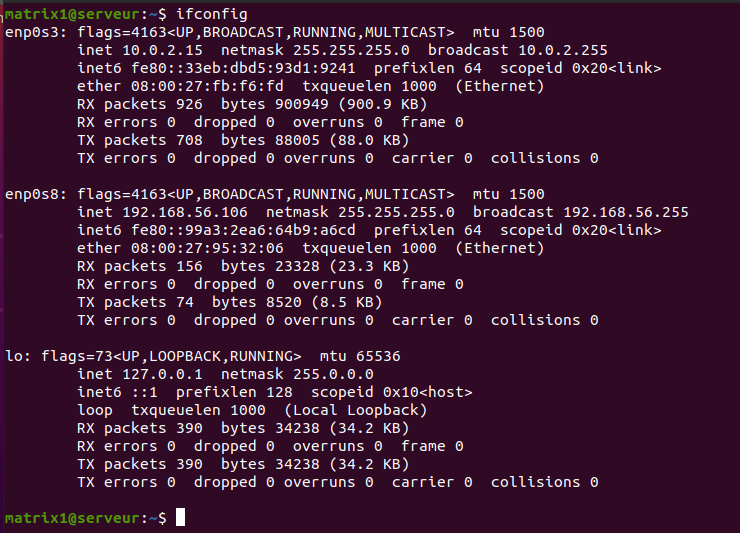
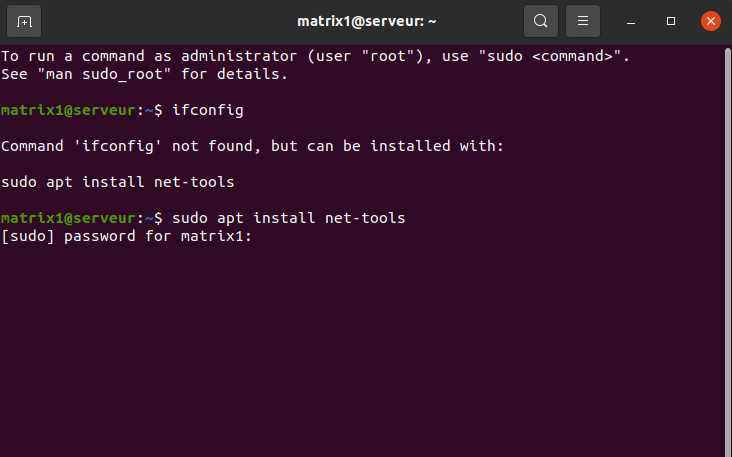
Choisissez 10GB de mémoire DISK, pour le virtual machine matrix1-client.

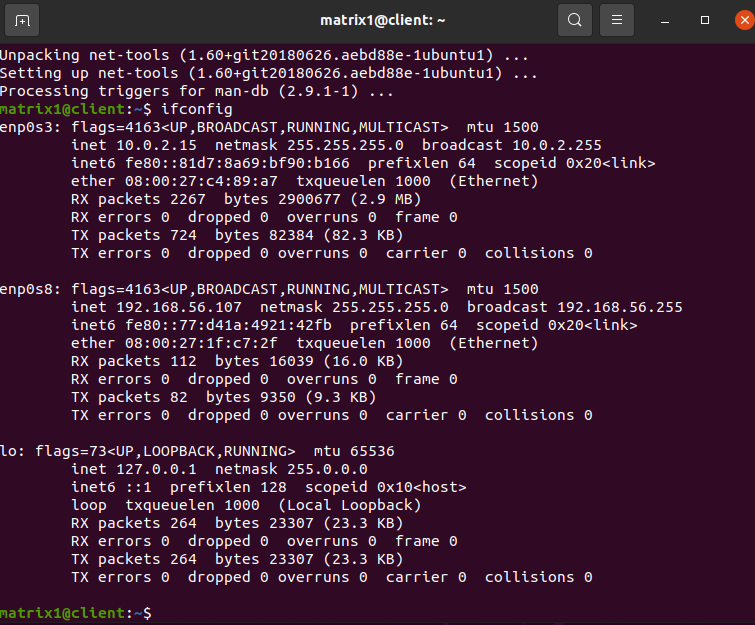


# **COMMUNIQUEZ LES DEUX MACHINES (SERVEUR ET CLIENT) À L’AIDE DE DEUX ADAPTATEURS RÉSEAU.**

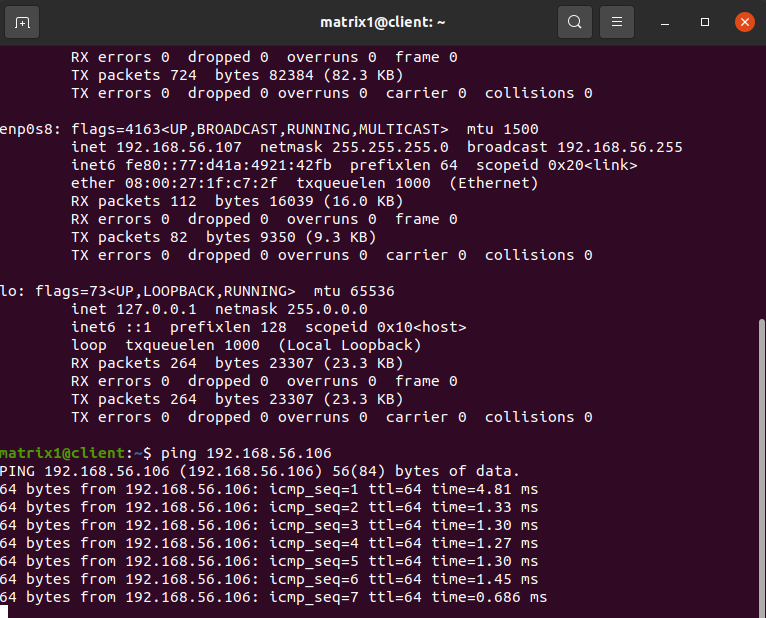
On va ajouter une carte de réseau sur les machines virtuelles, afin d’elles se communiquent. Cliquez sur « SETTINGS », après vous allez au « Network ».  Cliquez sur option « ADAPTER 2 » et marquez la boite « ENABLE NETWORK ADPTER ». Finalement, choisissez « HOST ONLY ADAPTER ».

Pour validez la bonne installation, on peut utiliser le command « ifconfig » dedans le TERMINAL SHELL.





Pour validez la bonne communication entre les deux, on peut utiliser le command « ping <IP> ».



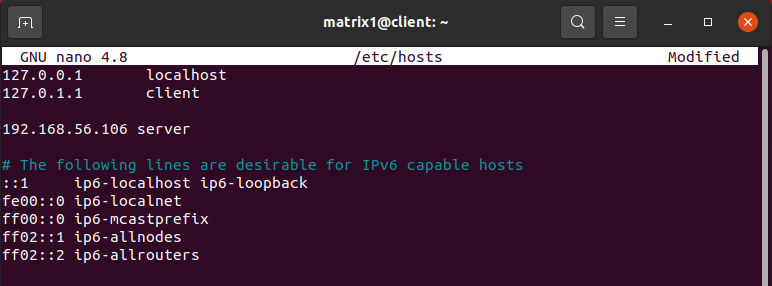
On va créer les variables de hosts, afin d’être plus simples se souvins de l’IP adress.

#CLIENT

sudo pico /etc/hosts

#add <ip-server> <nickName>

192.168.56.106 server

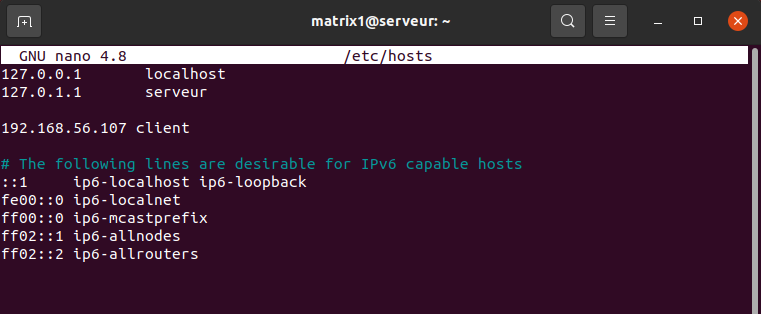


#SERVEUR

sudo pico /etc/hosts

#add <ip-client> <nickName>

192.168.56.105 client



Confirmer avec le command: ping <nickName>

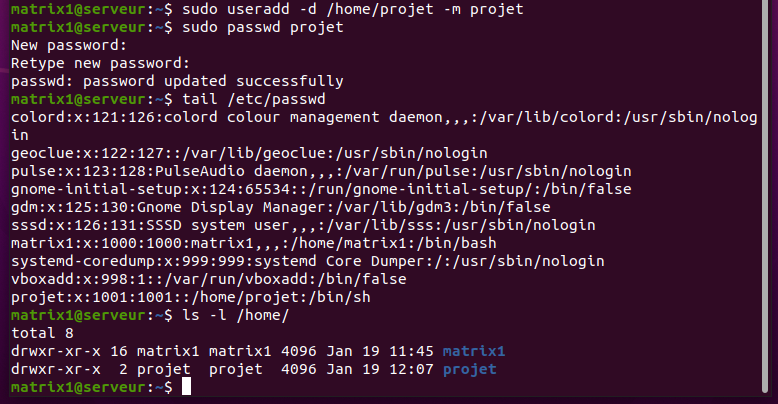
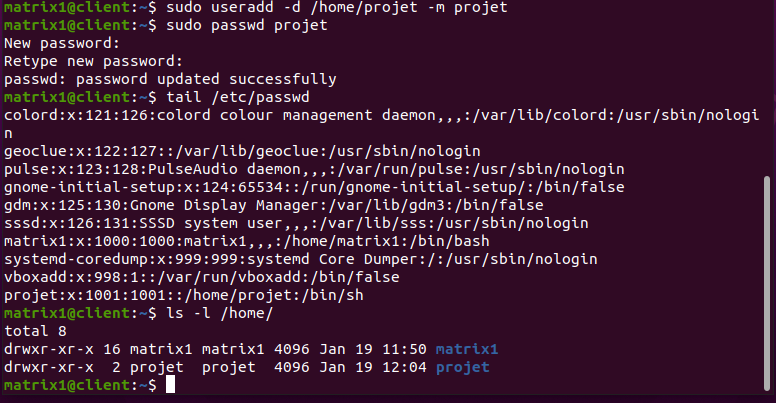
Ex:

ping client

ping server

# **CRÉEZ L’UTILISATEUR « PROJET ».**

On va une nouvelle l’utilisateur. Command « useradd ». Créer le mot de passe avec « passwd ».



# **GÉNÉREZ UNE CLÉ SSH SUR LE SERVEUR POUR L’UTILISATEUR DU PROJET.**

Prérequis:

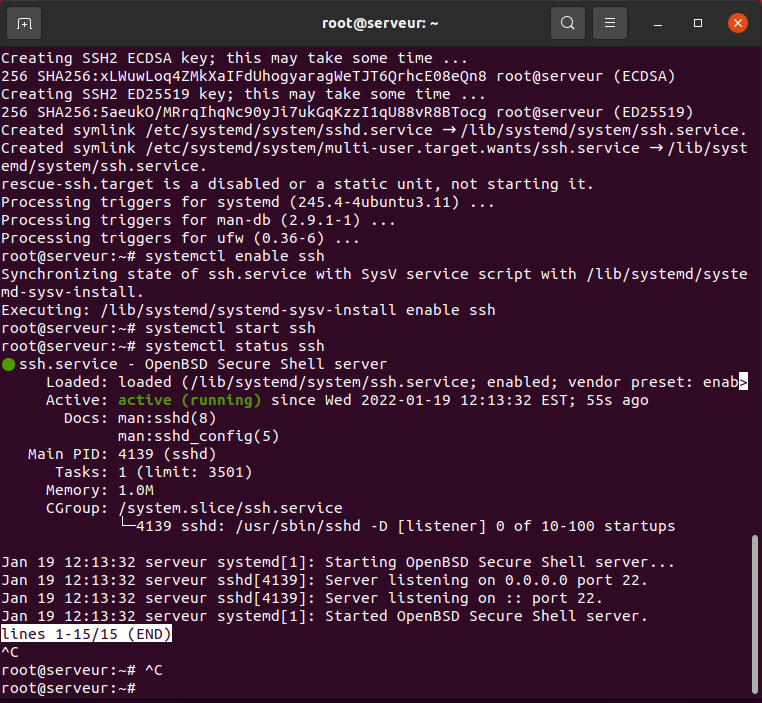
sudo su –

apt install openssh-server

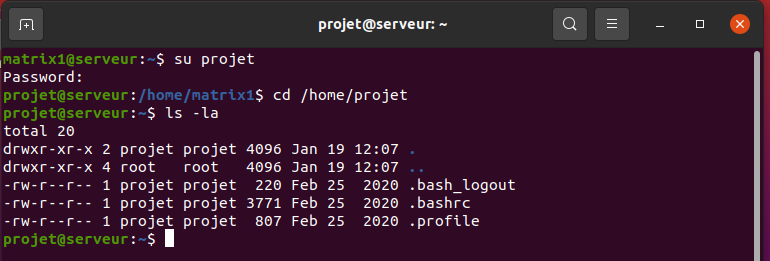
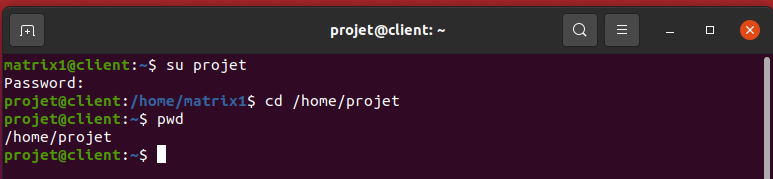
systemctl enable ssh

systemctl start ssh

systemctl status ssh

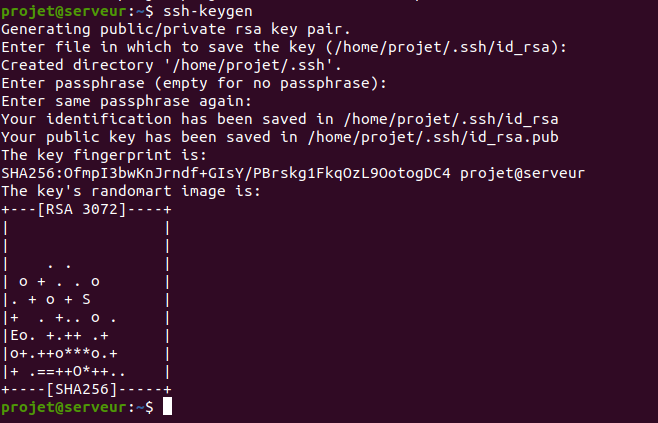


Utilisez le commando « exit », pour sortir de « root ». Après roule le command « su projet » pour utiliser user projet.



ssh-keygen

#Entrez le mot de passe.



# **Copiez la clé publique sur le client.**

Après la création de la clé publique, sur le couté du serveur, on va le copier vers le client. Après la transmission de la clé sur le client, on peut voir le fichier, « authorized\_keys »

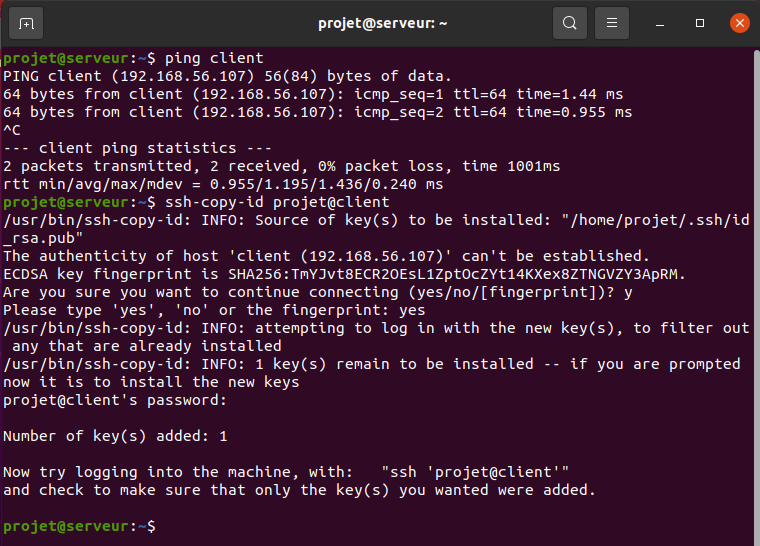
#SERVEUR

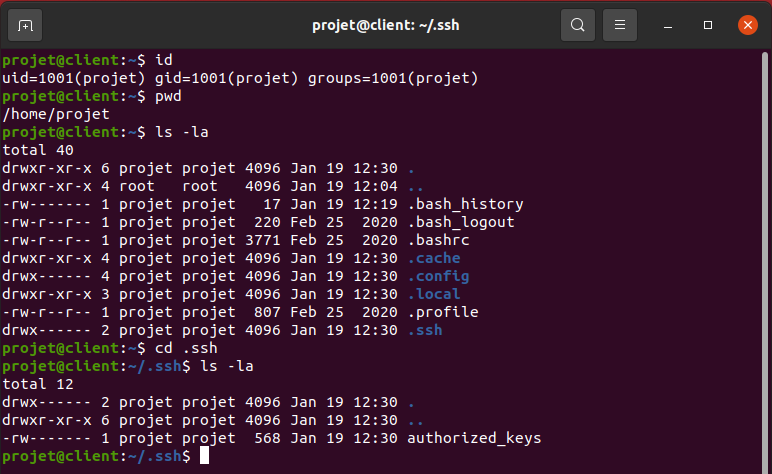
#ssh-copy-id <user-client>@<ip-client>

ssh-copy-id projet@client

# On peut utilizer nom client, parce que on fait l’édition sur /etc/hosts , ou

ssh-copy-id [**projet@192.168.56.107**](mailto:projet@192.168.56.107)





# **Validez qu`il y a bien communication entre le serveur et le client en transmettant le fichier /home/projet/test.txt du client au serveur.**

Prérequis:

###SERVEUR

#Sortir d’utilisateur « projet », avec command « exit ».

sudo apt install vsftpd -y

systemctl start vsftpd

#Logger avec « projet » : su projet | cd /home/projet

###CLIENT

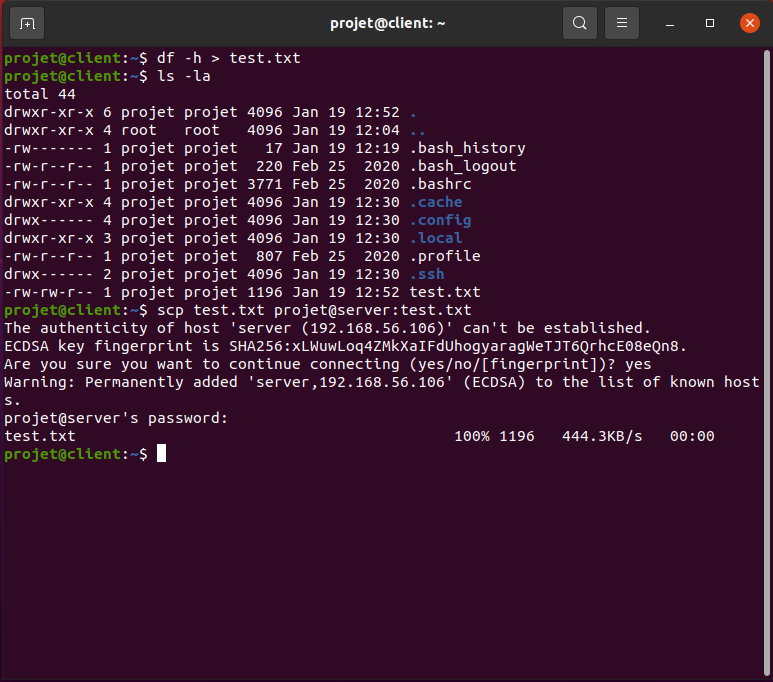
#Creer un fichier.

df -h > test.txt

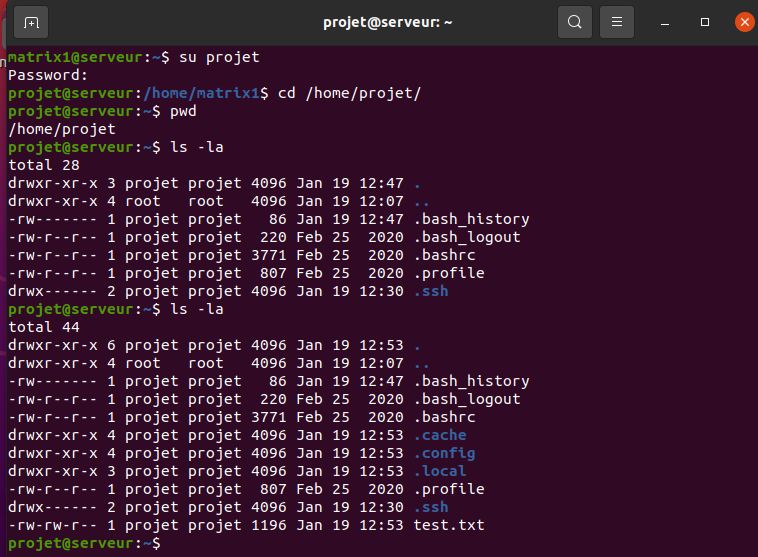
#Envoiez le fichier avec le command « scp ».

#Ex: scp <nomFichier> <nomUtilisateur>@<ipServer> :<nouveuNomeFichier>

scp test.txt [projet@server:test.txt](mailto:projet@server:test.txt)



On peut voir et validé le fichier qui été transféré.

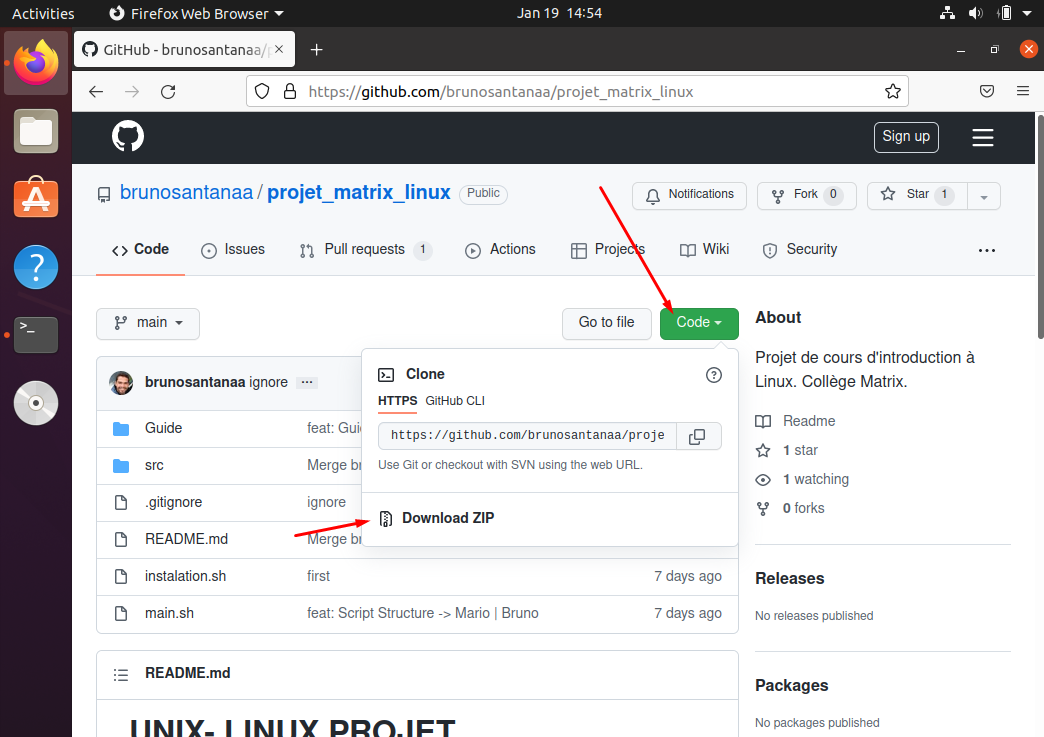


# **LE SCRIPT.**

## PRÉREQUIS:

Télécharger le projet compacté.

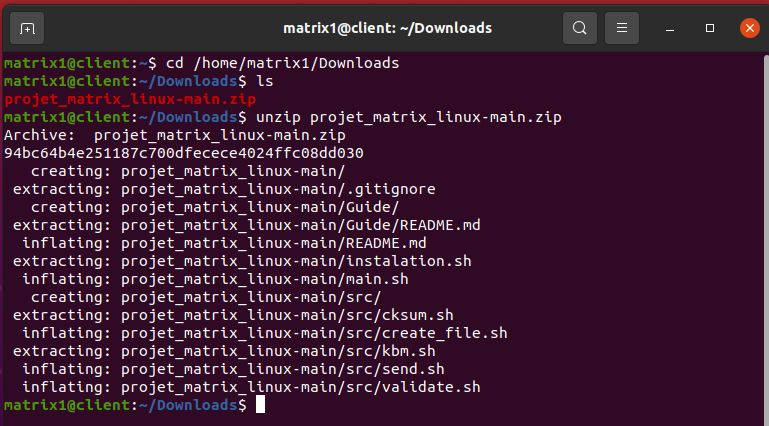
Lien : <https://github.com/brunosantanaa/projet_matrix_linux>



Sur le site, vous allez cliquer sur le Button « CODE », et après sur « Download ZIP » pour faire le téléchargement.

Dans le Terminal, vous allez o PATH de Downloads.

Utilisez le command : cd /home/matrix1/Downloads

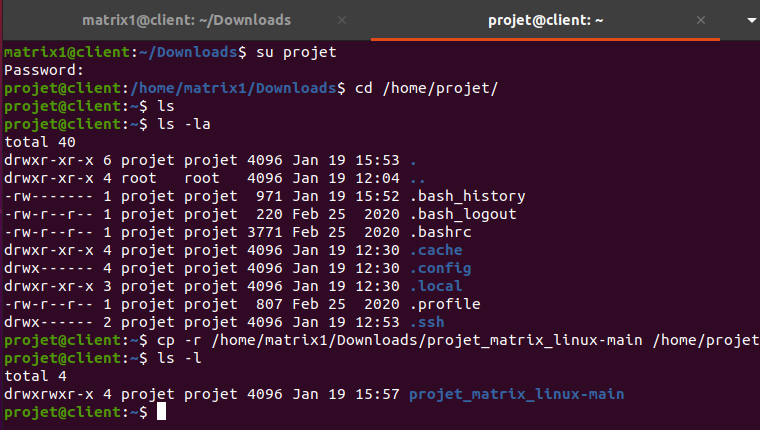


Maintenant on copiera le répertoire de l’utilisateur « matrix1 » à « projet ».

Utilisez le command :

cp -r projet\_matrix\_linux-main /home/projet

Alors, logger avec « projet » et allez o chemin du projet. /home/projet/projet\_matrix\_linux-main/

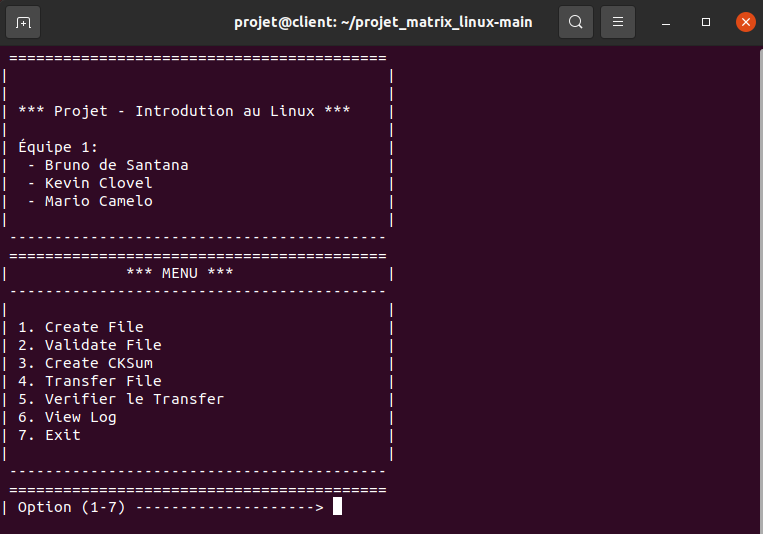


Le but de script c’est, avoir un MENU, et le command automatise pour exécuter les tâches suivantes.

## MENU SUIVANT SOUS LINUX.

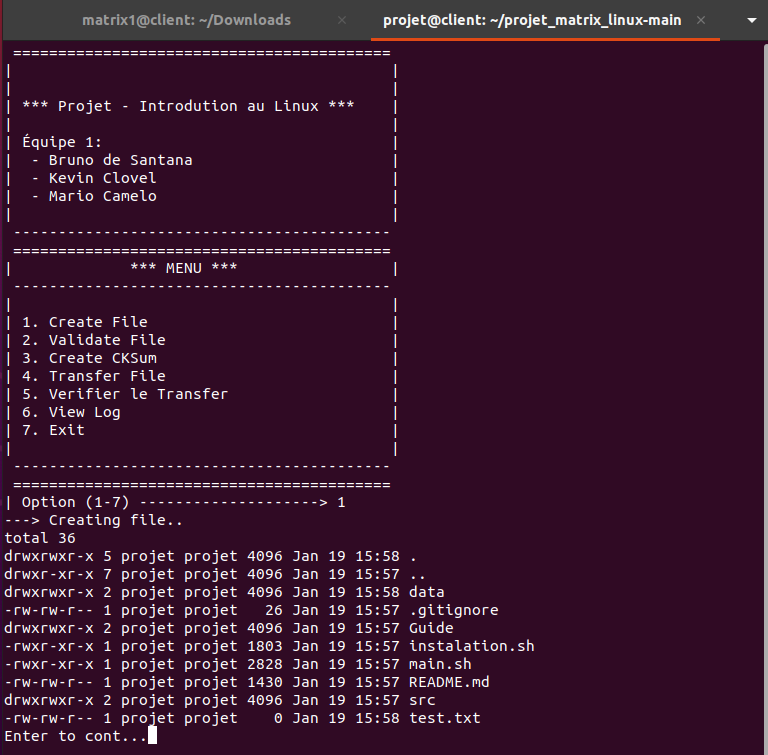
* Générer le fichier à copier.
* Valider le type de fichier généré.
* Générer le cksum à partir du fichier.
* Transmettre le fichier du client au serveur.
* \*Valider le cksum à partir du fichier du serveur.
* Journal de transmission.

Exécutez le script avec le command : ./main.sh

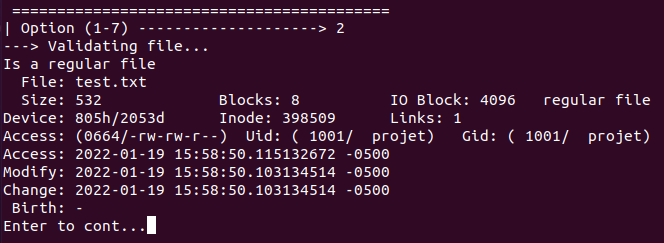


Dans le MENU, vous allez 7 options. Il faut que vous suivissiez la séquence de 1 à 7 pour avoir du succès.

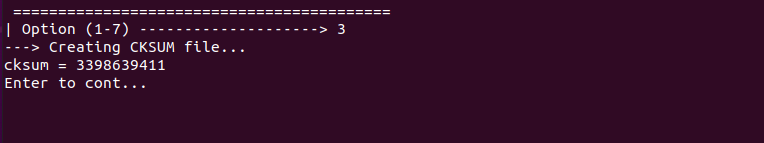
L’option 1, le script va créer un fichier que s’appelle test.txt.



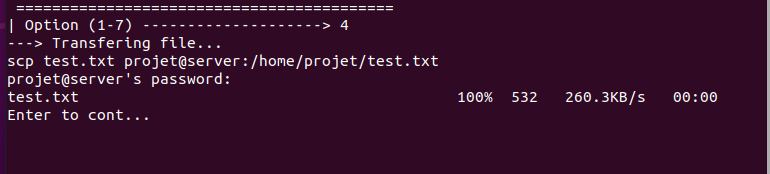
L’option 2, le script va valider le type du fichier test.txt.



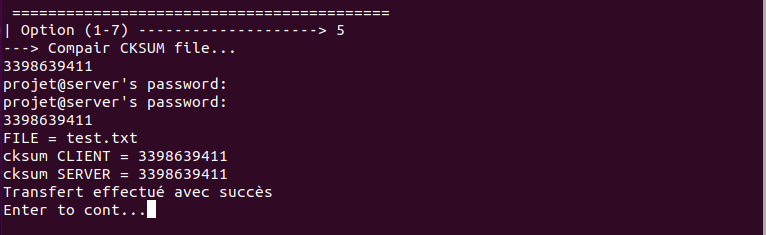
L’option 3, le script va générer le cksum du fichier test.txt.

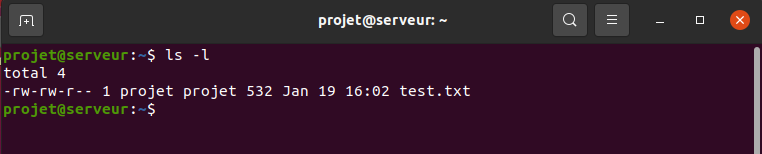


L’option 4, le script va transmettre le fichier test.txt du client au serveur.

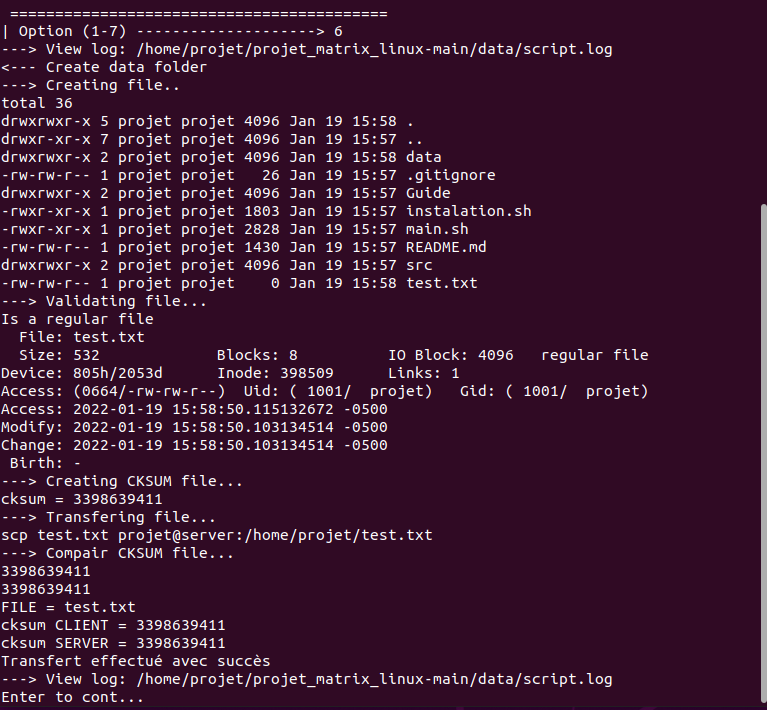


L’option 5, le script va valider si le fichier test.txt a été bien transmettre sans perte de donné.





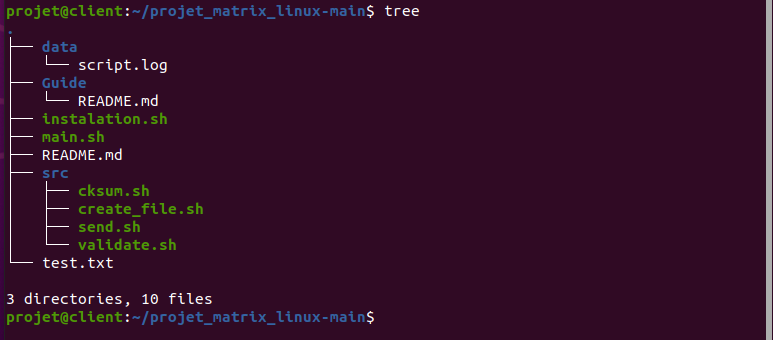
L’option 6, le script va montrer le journal de tout le tâches.



L’option 7, le script va sortir.

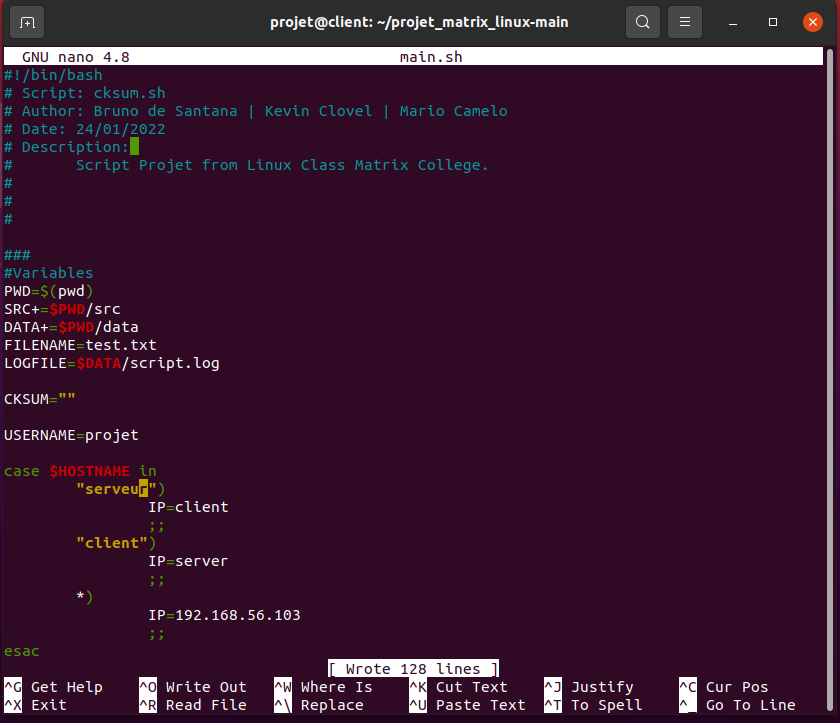


### LA SCRUCTURE DU SCRIPT.



## LES CODES.

### MAIN.SH



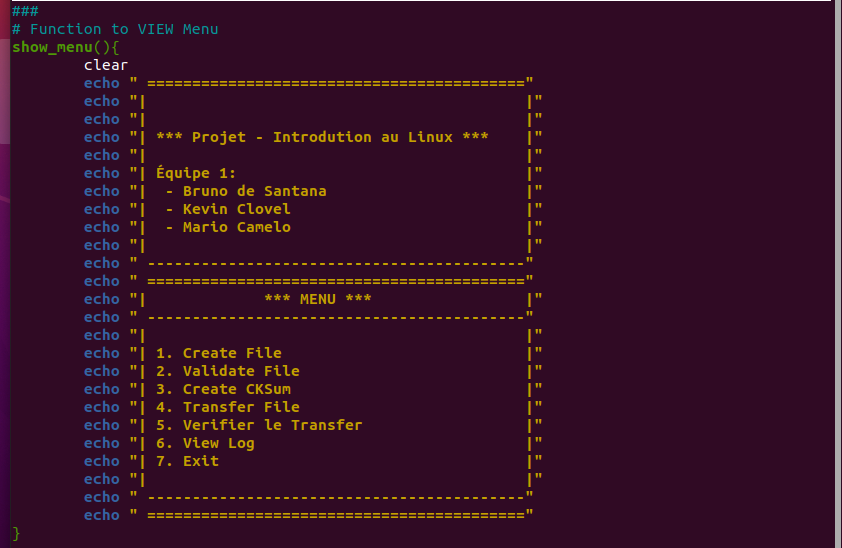
Les variables ont été défini.

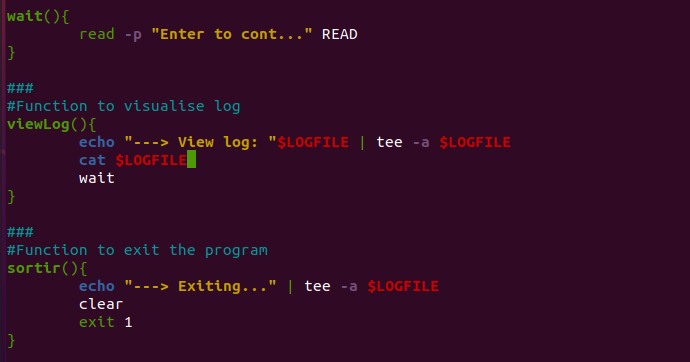


Le command CASE dans ce cas, a été utilisé afin de valider la valeur da variable $HOSTNAME, s’il était client, il ajointerai la variable IP avec le NOM(server) que nous avons créé(/etc/hosts), dans ce cas, le nom égale IP.

show\_menu() :

Efface l’écran (CLEAR) et afficher sur l’écran tout le ligne avec le command ECHO.





wait() :

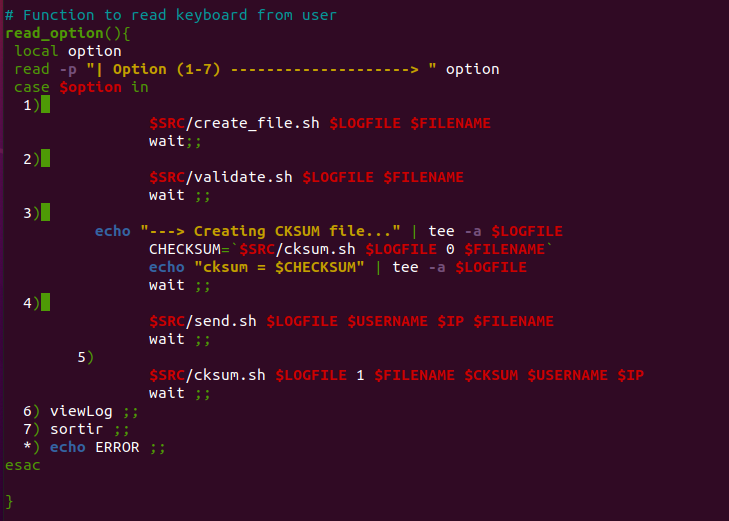
Afin d’attendre l’utilisateur taper quelque chose. (READ)

viewLog() :

Montrez sur l’écran tout le journal d’utilisateur dans le programme. (CAT)

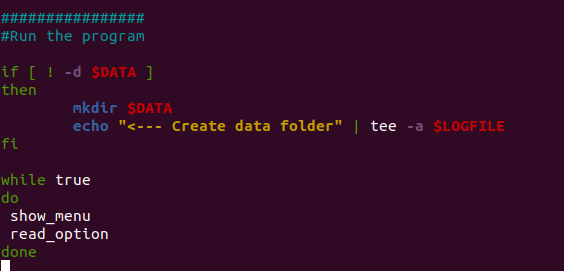
Sortir() :

Utilise le command EXIT, pour sortir du programme.



Read\_option() :

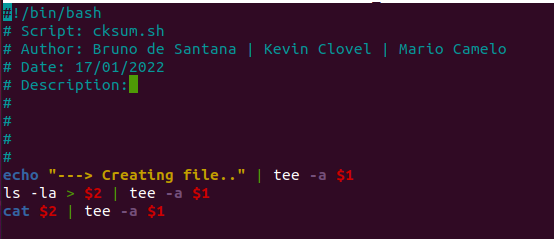
Utilise le command READ pour lire ce que l’utilisateur écrire, et avec le command CASE on peut délimiter l’options, et chaque option il faire appelle à l’autre fonction ou script. Ex : S’il écrivait le nombre 6, appellerais la fonction viewLog.



Ici on a un conditionnel qui valide se le répertoire existe, et si non, il crée.

Et pour fin, le boucle WHILE on exécute la fonction show\_menu et read\_option, le boucle sera infinité (TRUE) jusqu’à l’utilisateur appelle la fonction SORTIR.

### CREATE\_FILE.SH



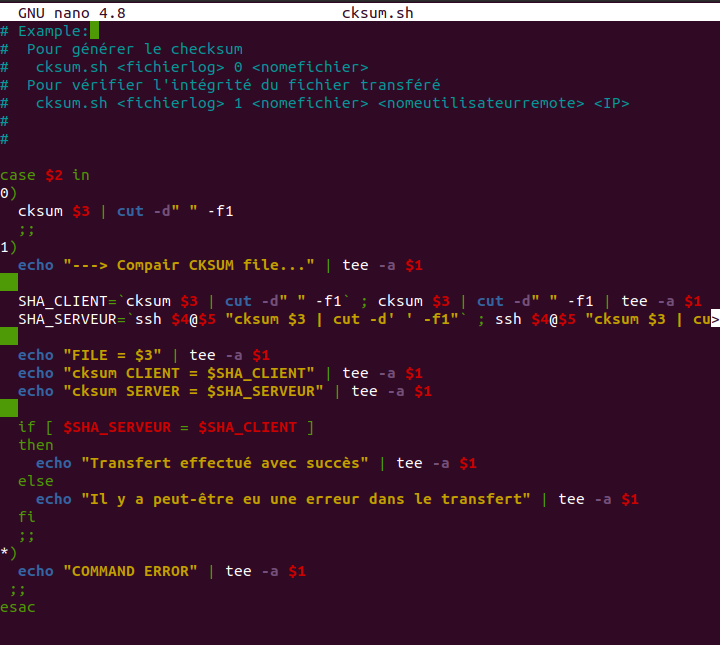
Un script de création de fichier.

### VALIDE.SH



Script avec un boucle IF, pour valider se le fichier existe, s’il est un fichier régulier or répertoire.

### CKSUM.SH



Ce script il y a deux fonctions, la première est générée un cksum, et l’autre est allez au SERVEUR avec le command SSH et prendre le nombre du cksum do fichier do serveur. Après il valide ce le numéro do cksum do serveur est égale au client.

### SEND.SH



Ce script send.sh utilisé le command SCP pour envoyer le fichier du client au serveur.