



# Rapport d'installation et configuration d'Arch Linux avec environnement graphique XFCE et Slim

Rochdi DARDOR | Administration Système | IATIC 5

**Enseignants :**

M. Franck TALBART

M. Max HOFFER

# Table des matières

<b>Introduction :</b>	<b>2</b>
<i>Etape 1 Téléchargement De L'iso Et Vérification De La Signature</i>	<i>2</i>
<i>Etape 2 : Création De La Machine Virtuelle :</i>	<i>3</i>
<i>Etape 3 : Démarrage Depuis L'iso Et Configuration Initiale</i>	<i>4</i>
<i>Etape 4 : Configuration Des Partitions Et Système De Fichiers</i>	<i>5</i>
<i>Etape 5 : Installation Du Systeme De Base</i>	<i>7</i>
<i>Etape 6 : Configuration Du Système De Base :</i>	<i>9</i>
<i>Etape 7 : Configuration Du Nom D'hote Et Du Fichier Hosts</i>	<i>11</i>
<i>Etape 8 : Création De L'utilisateur Et Configuration Du Compte Root</i>	<i>11</i>
<i>Etape 9 : Configuration Du Démarrage Du Système :</i>	<i>12</i>
<i>Etape 10 : Configuration Réseau Et Résolution Des Problèmes</i>	<i>15</i>
<i>Etape 11 : Installation Et Configuration De L'environnement Graphique</i>	<i>17</i>
<b>Evaluation Des Compétences Acquisées :</b>	<b>22</b>
<b>Conclusion :</b>	<b>23</b>

## Introduction :

Ce projet s'inscrit dans le cadre de ma formation en administration système et vise à approfondir ma compréhension des concepts fondamentaux liés à l'installation, la configuration et la personnalisation d'un environnement Linux. Arch Linux a été choisi comme plateforme pour ce projet en raison de son approche minimaliste, offrant une flexibilité et une maîtrise complète du système, tout en nécessitant une implication active de l'utilisateur dans chaque étape du processus.

L'objectif principal de ce projet était de réaliser l'installation complète et fonctionnelle d'Arch Linux sur une machine virtuelle, en mettant en place un environnement graphique léger et efficace, ainsi qu'une gestion optimisée des services réseau. En choisissant XFCE4 comme environnement graphique et SLiM comme gestionnaire de sessions, j'ai créé une interface utilisateur adaptée à mes besoins tout en relevant plusieurs défis techniques.

Ce rapport présente, étape par étape, les différentes phases de ce projet, incluant :

- La configuration initiale du matériel virtuel et des partitions.
- L'installation du système de base.
- La mise en place des services réseau et la résolution des problèmes rencontrés.
- L'installation et la personnalisation de l'environnement graphique.

Chaque étape est illustrée par des captures d'écran et enrichie de commentaires techniques pour fournir un guide clair et détaillé. Cette démarche, méthodique et progressive, vise à documenter non seulement les succès, mais également les défis et les solutions qui ont permis d'atteindre un résultat final stable et performant.

Ce projet a été une opportunité de développer des compétences pratiques essentielles en administration système, en diagnostic et en résolution de problèmes techniques. Ces acquis constituent une base solide pour mes futurs projets académiques et professionnels.

## ÉTAPE 1 : TELECHARGEMENT DE L'ISO ET VERIFICATION DE LA SIGNATURE

Avant d'entamer l'installation, il était essentiel de télécharger l'image ISO officielle d'Arch Linux et de vérifier son intégrité. Cette étape garantit que le fichier téléchargé est complet, authentique et sécurisé contre toute tentative de modification malveillante.

### Téléchargement de l'ISO :

L'image ISO officielle d'Arch Linux a été téléchargée depuis le [site officiel](#). La version utilisée pour ce projet est : archlinux-2024.12.01-x86\_64.iso.

Pour compléter cette étape, le fichier de signature correspondant (archlinux-2024.12.01-x86\_64.iso.sig) a également été téléchargé depuis la même source.

### Vérification de la signature :

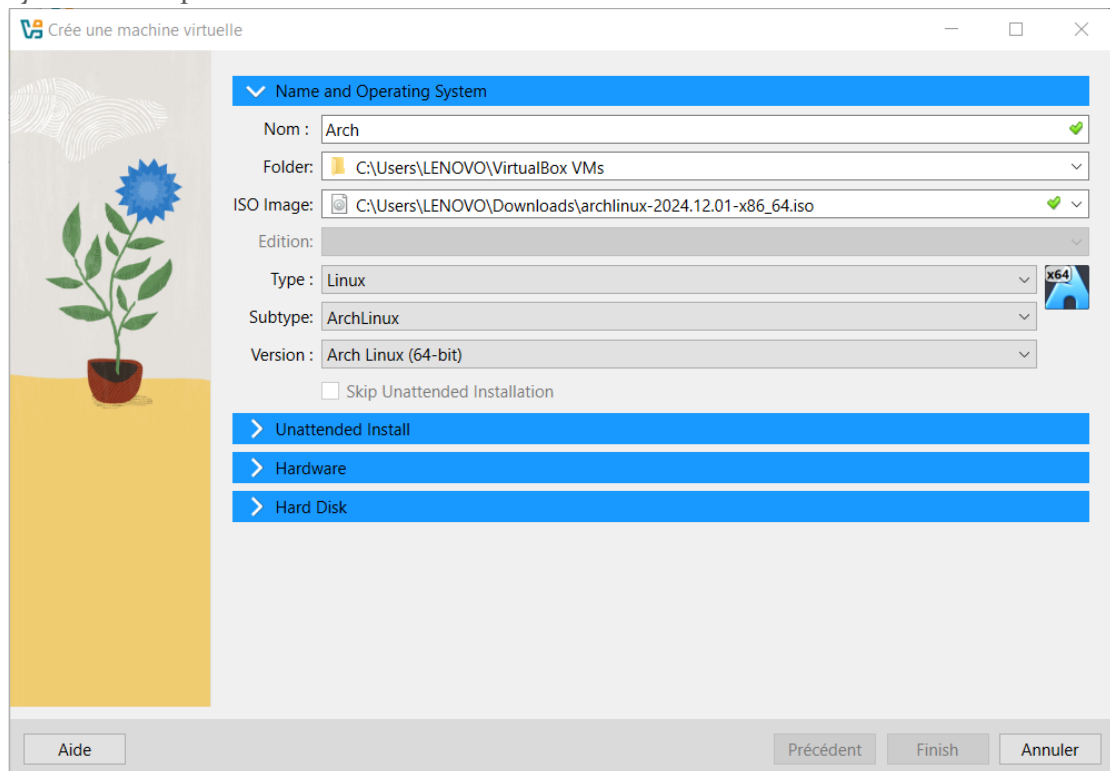
Une fois l'ISO et son fichier de signature récupérés, la vérification a été effectuée à l'aide de GPG (GNU Privacy Guard) , Pour garantir la sécurité et l'intégrité du fichier téléchargé Cette étape permet de s'assurer que le fichier n'a pas été altéré et qu'il provient d'une source de confiance.

```
C:\Users\LENOVO\Downloads>gpg --verify archlinux-2024.12.01-x86_64.iso.sig archlinux-2024.12.01-x86_64.iso
gpg: Signature faite le 12/01/24 06:50:47
gpg: avec la clef EDDSA 3E80CA1A8B89F69CBA57D98A76A5EF9054449A5C
gpg: issuer "pierre@archlinux.org"
gpg: Bonne signature de « Pierre Schmitz <pierre@archlinux.org> » [inconnu]
gpg: alias « Pierre Schmitz <pierre@archlinux.de> » [inconnu]
gpg: WARNING: The key's User ID is not certified with a trusted signature!
gpg: Rien n'indique que la signature appartient à son propriétaire.
Empreinte de clef principale : 3E80 CA1A 8B89 F69C BA57 D98A 76A5 EF90 5444 9A5C
```

Cette étape est essentielle pour prévenir toute tentative de compromission. Bien que la clé ne soit pas certifiée comme étant de confiance, elle est vérifiable et correspond à la clé officielle du mainteneur d'Arch Linux.

### ETAPE 2 : CREATION DE LA MACHINE VIRTUELLE :

Une fois l'image ISO téléchargée et vérifiée, la prochaine étape consistait à préparer un environnement virtuel pour l'installation d'Arch Linux. **VirtualBox** a été choisi comme hyperviseur en raison de sa simplicité d'utilisation et de sa compatibilité avec plusieurs systèmes d'exploitation.



## Paramétrage de la machine virtuelle :

- Nom : Arch
- Dossier : Emplacement personnalisé pour les fichiers de la VM.
- Type de système : Linux
- Sous-type : ArchLinux
- Version : Arch Linux (64-bit)
- Image ISO sélectionnée : archlinux-2024.12.01-x86\_64.iso

## Configuration matérielle :

- **Processeur** : Allocation de 2 cœurs pour assurer une bonne fluidité lors de l'installation et de l'utilisation.
- **RAM** : 3 Go (recommandé pour XFCE et les tâches courantes).
- **Stockage** : 20 Go, dont :
  - Partition /boot de 512 Mo (système de fichiers FAT32 pour la compatibilité EFI).
  - Partition principale / de 19,5 Go. (Système de fichiers EXT4.)
- **Mode réseau** : NAT, permettant à la VM d'accéder à Internet via la machine hôte.

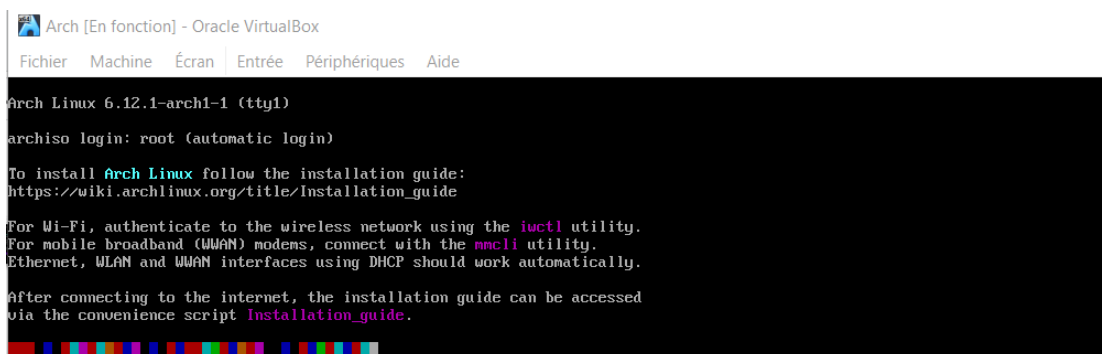
Cette étape a permis de préparer un environnement virtuel optimal pour l'installation d'Arch Linux. Nous sommes maintenant prêts à lancer l'installation depuis l'ISO.

## ETAPE 3 : DEMARRAGE DEPUIS L'ISO ET CONFIGURATION INITIALE

Après la création de la machine virtuelle, l'image ISO téléchargée a été montée, et le système Arch Linux a été démarré en mode live. Cette étape nous a permis de préparer l'environnement pour l'installation complète du système.

### Démarrage depuis l'ISO :

- Une fois la machine virtuelle démarrée, le menu d'accueil de l'ISO d'Arch Linux s'est affiché.

The image shows a screenshot of a virtual machine window titled "Arch [En fonction] - Oracle VirtualBox". The window contains a terminal window with the following text:

```
Arch Linux 6.12.1-arch1-1 (tty1)
archiso login: root (automatic login)

To install Arch Linux follow the installation guide:
https://wiki.archlinux.org/title/Installation_guide

For Wi-Fi, authenticate to the wireless network using the iwctl utility.
For mobile broadband (WWAN) modems, connect with the mmcli utility.
Ethernet, WLAN and WWAN interfaces using DHCP should work automatically.

After connecting to the internet, the installation guide can be accessed
via the convenience script Installation_guide.
```

Cette étape prépare le terrain pour configurer les partitions, les systèmes de fichiers et l'installation de base. Les captures d'écran ci-jointes illustrent ces étapes pour référence.

## Configuration initiale :

- Vérification de l'état du réseau pour assurer une connexion Internet fonctionnelle :

```
root@archiso ~ # ping -c 3 archlinux.org
PING archlinux.org (95.217.163.246) 56(84) bytes of data.
64 bytes from archlinux.org (95.217.163.246): icmp_seq=1 ttl=255 time=91.8 ms
64 bytes from archlinux.org (95.217.163.246): icmp_seq=2 ttl=255 time=90.9 ms
64 bytes from archlinux.org (95.217.163.246): icmp_seq=3 ttl=255 time=91.3 ms

--- archlinux.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 90.944/91.352/91.834/0.367 ms
root@archiso ~ # _
```

Cette commande a confirmé que la connexion était active, essentielle pour télécharger les paquets nécessaires.

## Synchronisation de l'horloge système :

```
root@archiso ~ # timedatectl set-ntp true
root@archiso ~ # timedatectl status
          Local time: Fri 2024-12-27 22:02:07 UTC
          Universal time: Fri 2024-12-27 22:02:07 UTC
             RTC time: Fri 2024-12-27 22:02:08
             Time zone: UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
root@archiso ~ #
```

Cela garantit que l'heure et la date sont synchronisées, une étape critique pour éviter les erreurs liées aux certificats ou aux paquets.

## ETAPE 4 : CONFIGURATION DES PARTITIONS ET SYSTEME DE FICHIERS

Après avoir vérifié la connectivité réseau, la prochaine étape était de configurer les partitions du disque pour accueillir le système Arch Linux. Pour ce faire, nous avons utilisé l'outil `cfdisk`, qui offre une interface simple et intuitive.

### 1. Sélection de la table de partitions :

Lors du lancement de `cfdisk`, il a été demandé de choisir un type de table de partitions. Le type **GPT (GUID Partition Table)** a été sélectionné pour sa compatibilité avec les systèmes modernes et l'utilisation d'un démarrage UEFI.

## 2. Création des partitions

Deux partitions ont été configurées pour répondre aux besoins du projet .:

### 1. Partition pour le démarrage (/boot) :

```

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: gpt, identifier: 3a2a8715-d68c-4ca9-a880-50ffff4486e

Device      Start          End          Sectors      Size Type
/dev/sda1   2048           1050623      1048576      512M EFI System
[ Enter ]

Partition UUID: C99E2D97-676F-43FD-BFC3-DB96E3659FB0
Partition type: EFI System (C12A7328-F81F-11D2-B04B-0000C93EC93B)

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]

Quit program without writing changes

```

- Taille : 512 Mo
- Type : EFI System
- Système de fichiers : FAT32

**Raison :** Cette partition est essentielle pour un démarrage avec UEFI

### 2. Partition racine (/) :

```

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: gpt, identifier: 3a2a8715-d68c-4ca9-a880-50ffff4486e

Device      Start          End          Sectors      Size Type
/dev/sda1   2048           1050623      1048576      512M EFI System
/dev/sda2   1050624        41940991     40890368     19.5G Linux filesystem
[ Enter ]

Partition UUID: B08F713F-2665-48EF-A154-41B01021D170
Partition type: Linux filesystem (0FC63D0F-8483-4772-8E79-3B690477D0E4)

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]

Write partition table to disk (this might destroy data)

```

- Taille : 19,5 Go (le reste de l'espace disque)
- Type : Linux Filesystem
- Système de fichiers : EXT4

**Raison :** La partition racine contient le système d'exploitation et tous les fichiers nécessaires à son fonctionnement. Une fois les partitions créées, les modifications ont été enregistrées en sélectionnant **Write** dans le menu

### 3. Formatage des partitions :

Une fois les partitions créées chaque partition a été formatée avec le système de fichiers approprié comme suite :

```
root@archiso ~ # mkfs.fat -F32 /dev/sda1
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkfs.ext4 /dev/sda2
mke2fs 1.47.1 (20-May-2024)
Creating filesystem with 5111296 4k blocks and 1277952 inodes
Filesystem UUID: a735111e-pa63-44e1-abfb-e023f5aa665a
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1695632, 2654208,
    4096000
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
root@archiso ~ # _
```

Partition `/boot` formatée en FAT32 pour la compatibilité UEFI

Partition racine (`/`) formatée en ext4

### 4.Montage des partitions :

Les partitions ont ensuite été montées sur leurs points de montage respectifs :

- Partition racine montée sur `/mnt` avec la commande **mount** :

```
root@archiso ~ # mount /dev/sda2 /mnt
```

- Partition `/boot` montée sur `/mnt/boot` :

```
root@archiso ~ # mount /dev/sda1 /mnt/boot
```

### Justification des choix

Le type de table de partitions GPT a été choisi pour sa compatibilité avec les systèmes modernes et pour prendre en charge le démarrage UEFI. La partition `/boot`, formatée en FAT32, est indispensable pour garantir la compatibilité avec le firmware UEFI. De plus, une partition racine `/`, formatée en EXT4, a été configurée pour assurer un système de fichiers performant, stable et adapté aux environnements Linux. Ces choix permettent une flexibilité et une fiabilité optimales, tout en respectant les standards actuels d'installation.

## ETAPE 5 : INSTALLATION DU SYSTEME DE BASE

Avec les partitions correctement configurées et montées, l'installation du système de base d'Arch Linux a été effectuée. Cette étape constitue le cœur de l'installation et fournit les composants essentiels nécessaires au fonctionnement du système

### Installation des paquets essentiels :

La commande suivante a été utilisée pour installer les paquets de base : **pacstrap /mnt base linux linux-firmware**



## Accès au système installé via arch-chroot :

Une fois les partitions montées et le système de base installé, la commande **arch-chroot** a été utilisée pour accéder au système installé et effectuer les configurations nécessaires.

```
root@archiso ~ # arch-chroot /mnt  
[root@archiso /]#
```

Cette commande crée un environnement chroot (Change Root), simulant le système installé comme s'il était déjà démarré. Cela permet de réaliser les configurations essentielles directement dans l'environnement final.

## ÉTAPE 6 : CONFIGURATION DU SYSTEME DE BASE :

Après l'installation du système de base, plusieurs configurations essentielles ont été réalisées pour préparer Arch Linux à fonctionner de manière optimale. Ces étapes incluent la configuration du fuseau horaire, des locales, et du clavier.

### Configuration du fuseau horaire :

Pour configurer le fuseau horaire, la commande suivante a été utilisée pour afficher tous les fuseaux horaires disponibles :

```
root@archiso /]# ls /usr/share/zoneinfo  
Africa  Australia  Cuba      Etc      GMT+0    Iceland  Kwajalein  NZ       Poland   Turkey    WET      posixrules  
America  Brazil     EET       Europe  GMT-0    Indian   Libya      NZ-CHAT  Portugal  UCT      Zulu     right  
Antarctica  CET      EST      Factory  GMT0     Iran     MET        Nava jo   ROC      US       iso3166.tab  tzdata.zi  
Arctic     CST6CDT   EST5EDT  GB       Greenwich  Israel   MST        PRC       BOK      UTC      leap-seconds.list  zone.tab  
Asia      Canada    Egypt    GB-Eire  HST      Jamaica  MST7MDT   PST8PDT  SECURITY  Universal  leapseconds  zone1970.tab  
Atlantic  Chile     Eire     GMT      Hongkong  Japan    Mexico    Pacific  Singapore  U-SU     posix      zonenow.tab
```

Cela nous a permis de naviguer dans les différents fuseaux horaires et de choisir celui qui convient à notre emplacement géographique. Dans notre cas, nous avons sélectionné le fuseau horaire **Europe/Paris**, comme illustré dans la capture d'écran.

Ensuite, un lien symbolique a été créé pour définir le fuseau horaire sélectionné

### Synchronisation de l'horloge système :

Après avoir défini le fuseau horaire, l'horloge système a été synchronisée avec l'horloge matérielle en utilisant la commande suivante :

```
[root@archiso /]# hwclock --systohc
```

Cela garantit que l'heure système correspond à l'heure définie par le matériel, ce qui est essentiel pour la cohérence des logs et des services.



## ÉTAPE 7 : CONFIGURATION DU NOM D'HÔTE ET DU FICHIER HOSTS

### Configuration du nom d'hôte :

Pour définir le nom de la machine, le fichier `/etc/hostname` a été modifié. Nous avons utilisé l'éditeur **nano** pour y inscrire le nom d'hôte souhaité, ici `archlinux`. Cela permet d'identifier clairement la machine sur le réseau.

```
GNU nano 8.2 /etc/hostname
archlinux
```

### Configuration du fichier `/etc/hosts` :

Le fichier `/etc/hosts` a été configuré pour associer l'adresse IP locale au nom d'hôte défini précédemment. Cela garantit une résolution correcte des noms d'hôtes sur la machine.

```
GNU nano 8.2 /etc/hosts
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1 localhost
::1 localhost
127.0.1.1 archlinux.localdomain archlinux
```

Ce fichier assure que les noms d'hôtes de la machine soient correctement reconnus, même en l'absence de serveur DNS.

## ÉTAPE 8 : CRÉATION DE L'UTILISATEUR ET CONFIGURATION DU MOT DE PASSE ROOT

Pour garantir la sécurité et permettre une gestion efficace du système, un mot de passe pour le compte `root` a été défini, et un utilisateur standard a été créé avec des privilèges administratifs. Ces configurations permettent une gestion sûre et isolée des droits d'accès.

### Configuration du mot de passe root

Le compte `root` (administrateur) a besoin d'un mot de passe sécurisé pour gérer le système. La commande suivante a été utilisée pour définir ce mot de passe :

```
root@archiso /l# passwd
New password:
Retype new password:
passwd: password updated successfully
root@archiso /l#
```

## Création d'un utilisateur standard

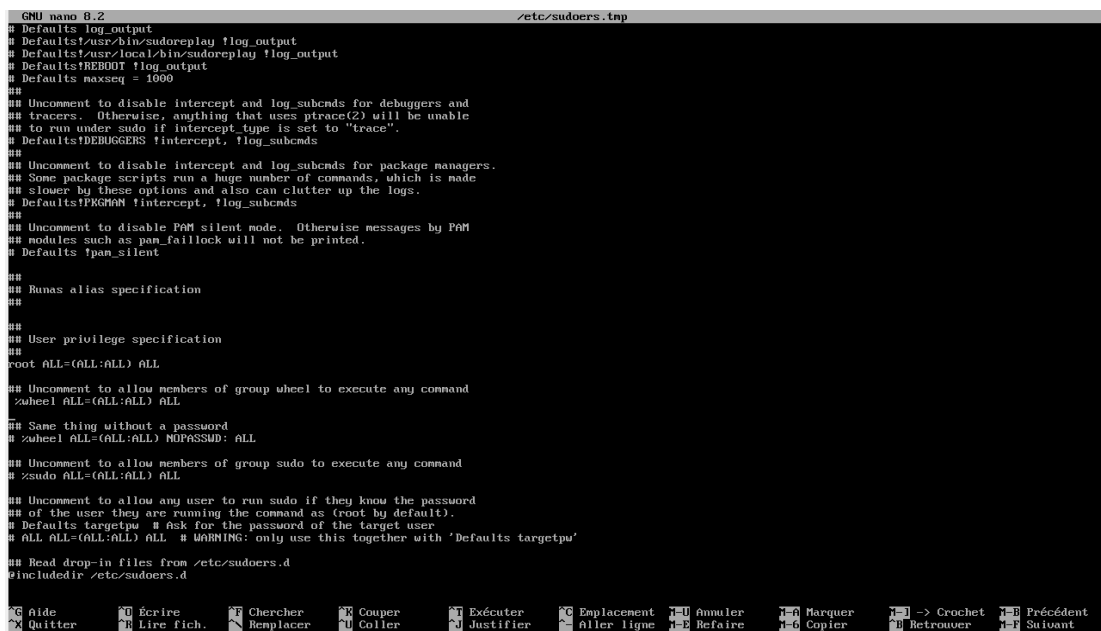
Pour des raisons de sécurité et de gestion, il est recommandé de ne pas utiliser directement le compte `root` pour les tâches quotidiennes. Un utilisateur standard a donc été créé à l'aide de la commande suivante :

`useradd -m -G wheel -s /bin/bash iatic`, et ensuite un mot de passe a été défini pour le nouvel utilisateur

```
[root@archlinux ~]# useradd -m -G wheel -s /bin/bash iatic
[root@archlinux ~]# passwd iatic
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
```

## Activation des privilèges sudo pour l'utilisateur

Pour permettre à l'utilisateur standard d'exécuter des commandes avec des privilèges administratifs, le fichier de configuration `sudoers` a été modifié. Nous avons utilisé la commande suivante pour éditer le fichier en toute sécurité avec l'éditeur `visudo` :



```
GNU nano 0.2 /etc/sudoers.tmp
# Defaults log_output
# Defaults!user/bin/sudoedit !log_output
# Defaults!user/local/bin/sudoedit !log_output
# Defaults!REBOOT !log_output
# Defaults maxseq = 1000
###
### Uncomment to disable intercept and log_subcmds for debuggers and
### tracers. Otherwise, anything that uses ptrace(2) will be unable
### to run under sudo if intercept_type is set to "trace".
# Defaults!DEBUGGERS !intercept, !log_subcmds
###
### Uncomment to disable intercept and log_subcmds for package managers.
### Some package scripts run a huge number of commands, which is made
### slower by these options and also can clutter up the logs.
# Defaults!PROGRAM !intercept, !log_subcmds
###
### Uncomment to disable PAM silent mode. Otherwise messages by PAM
### modules such as pam_faillock will not be printed.
# Defaults !pam_silent
###
### Runas alias specification
###
###
### User privilege specification
root ALL=(ALL:ALL) ALL
###
### Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL
###
### Same thing without a password
# %wheel ALL=(ALL:ALL) NOPASSWD: ALL
###
### Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL:ALL) ALL
###
### Uncomment to allow any user to run sudo if they know the password
### of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'
###
### Read drop-in files from /etc/sudoers.d
#includedir /etc/sudoers.d

^G Aide      ^O Écrire   ^F Chercher  ^K Couper   ^I Exécuter  ^C Emplacement ^U Annuler  ^A Marquer  ^J -> Crochet ^B Précédent
^X Quitter   ^R Lire fich. ^N Remplacer ^U Coller   ^J Justifier ^L Aller ligne ^Z Refaire  ^G Copier    ^D Retrouver  ^F Suivant
```

## ETAPE 9 : CONFIGURATION DU DÉMARRAGE DU SYSTÈME :

Pour permettre à Arch Linux de démarrer correctement, un gestionnaire de démarrage (bootloader) a été installé et configuré. GRUB (GRand Unified Bootloader) a été choisi en raison de sa compatibilité et de sa flexibilité.

## Installation de GRUB

La commande suivante a été utilisée pour installer GRUB et les dépendances nécessaires :

```
[root@archiso /]# pacman -S grub efibootmgr
resolving dependencies...
looking for conflicting packages...

Packages (3) efivar-39-1 efibootmgr-18-3 grub-2:2.12-3

Total Download Size: 6.98 MiB
Total Installed Size: 34.35 MiB

:: Proceed with installation? [Y/n]
```

Ensuite, GRUB a été installé sur le disque principal. La commande suivante a été utilisée : **grub-install --target=x86\_64-efi --efi-directory=/boot --bootloader-id=GRUB**

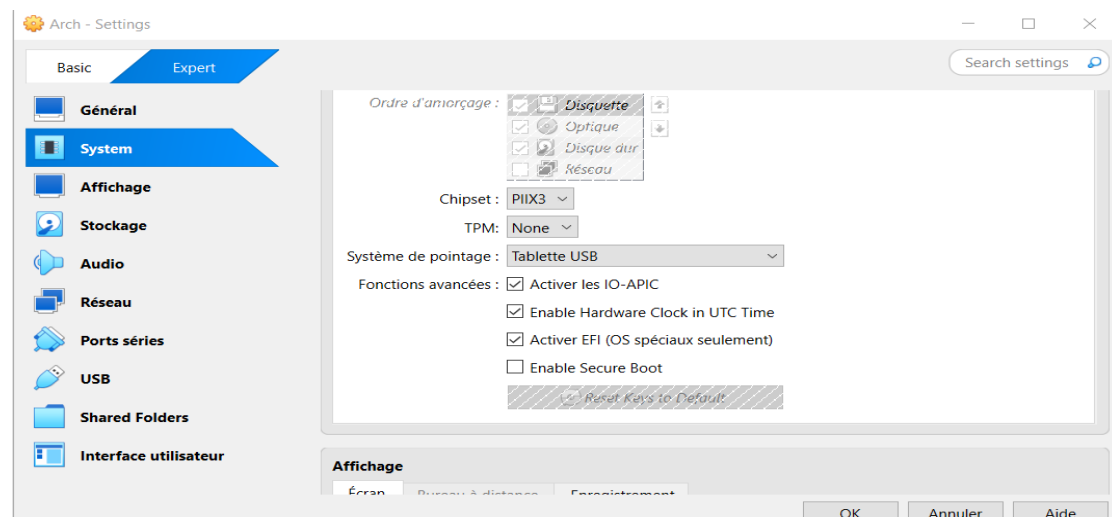
- target=x86\_64-efi : Indique que le système est basé sur UEFI.
- efi-directory=/boot : Spécifie le point de montage EFI.
- bootloader-id=GRUB : Attribue un identifiant au chargeur de démarrage.

### Problème rencontré : Absence de prise en charge EFI

Lors de la première tentative d'installation, une erreur s'est produite indiquant que les variables EFI n'étaient pas prises en charge sur le système. Voici le message d'erreur affiché :

```
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=GRUB
Installing for x86_64-efi platform.
EFI variables are not supported on this system.
EFI variables are not supported on this system.
grub-install: error: efibootmgr failed to register the boot entry: No such file or directory.
[root@archiso /]#
```

Après un petit tour sur la wiki et vérification, il s'est avéré que la configuration de la machine virtuelle n'incluait pas la prise en charge d'EFI. Pour résoudre ce problème, nous avons activé l'option **EFI** dans les paramètres de VirtualBox, comme illustré ci-dessous :



Après avoir activé EFI, nous avons relancé la commande d'installation de GRUB, qui s'est terminée avec succès :

```
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=GRUB
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso /]#
```

**Génération du fichier de configuration GRUB :**Après avoir installé le chargeur d'amorçage GRUB avec succès, la commande suivante a été utilisée pour générer le fichier de configuration de GRUB :

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot:  initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@archiso /]#
```

Cette commande a localisé les images du noyau et les a incluses dans le fichier de configuration, comme montré dans la capture d'écran ci-dessus. Le fichier généré assure que GRUB peut charger Arch Linux correctement.

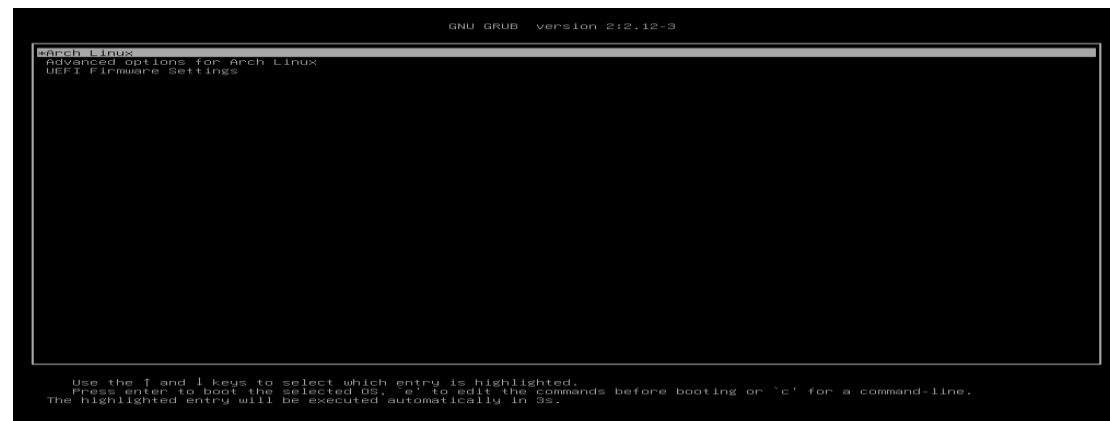
**Sortie de l'environnement chroot et démontage :**

Une fois toutes les configurations terminées dans l'environnement chroot, la commande **exit** a été utilisée pour quitter cet environnement, Ensuite, toutes les partitions montées ont été démontées pour préparer le redémarrage.

```
[root@archiso /]# exit
exit
root@archiso ~ # umount -R /mnt
root@archiso ~ #
```

Ces étapes sont illustrées dans les captures fournies. Cela garantit que le système est prêt à démarrer avec les paramètres configurés.

**Test de Démarrage avec GRUB :**Après le redémarrage de la machine virtuelle, le menu GRUB s'est affiché, comme illustré dans la capture ci-dessus. Les options disponibles incluent :



- **Arch Linux** : Pour démarrer le système.
- **Advanced options** : Pour des démarrages avancés, comme le mode de secours.
- **UEFI Firmware Settings** : Pour accéder aux paramètres EFI si nécessaire.

L'option Arch Linux a été sélectionnée pour vérifier que le système installé démarre correctement. Cette étape a confirmé que l'installation de GRUB était réussie et fonctionnelle.

## ETAPE 10 : CONFIGURATION RESEAU ET RESOLUTION DES PROBLEMES

### Problème initial de connexion réseau

Après le redémarrage, une tentative de ping vers archlinux.org a échoué en raison d'un problème temporaire de résolution DNS. L'analyse de l'état des interfaces réseau avec la commande ip link a révélé que l'interface enp0s3 était désactivée (état DOWN).

```
root@archlinux ~# ping -c 3 archlinux.org
ping: archlinux.org: Échec temporaire dans la résolution du nom
root@archlinux ~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
   link/ether 08:00:27:c9:47:3e brd ff:ff:ff:ff:ff:ff
   altname enx080027c9473e
root@archlinux ~#
```

### Activation de l'interface réseau

Pour résoudre ce problème, nous avons activé l'interface réseau en exécutant :

```
[root@archlinux ~]# ip link set enp0s3 up
[root@archlinux ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
   link/ether 08:00:27:c9:47:3e brd ff:ff:ff:ff:ff:ff
   altname enx080027c9473e
[root@archlinux ~]#
```

Après activation, la commande ip link a confirmé que l'interface était désormais en état UP.

```
root@archlinux ~# systemctl restart systemd-networkd
root@archlinux ~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
     valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
     valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:c9:47:3e brd ff:ff:ff:ff:ff:ff
   altname enx080027c9473e
   inet6 fd00:a00:27ff:fec9:473e/64 scope global dynamic mngtmpaddr proto kernel_r
     valid_lft 85763sec preferred_lft 13763sec
   inet6 fe80:a00:27ff:fec9:473e/64 scope link proto kernel_l1
     valid_lft forever preferred_lft forever
root@archlinux ~#
```

### Configuration manuelle de l'adresse IP et de la passerelle :

Avant d'installer dhcpd, nous avons configuré manuellement une adresse IPv4 pour résoudre temporairement le problème :

```
[root@archlinux ~]# ip addr add 10.0.2.15/24 dev enp0s3
[root@archlinux ~]# ip route add default via 10.0.2.2
```

Ensuite, nous avons ajouté les serveurs DNS Google dans le fichier /etc/resolv.conf :

```
GNU nano 8.2 /etc/resolv.conf
# Resolver configuration file.
# See resolv.conf(5) for details.
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Un ping vers la passerelle (10.0.2.2), Google DNS (8.8.8.8), et enfin archlinux.org a confirmé que la connectivité était désormais opérationnelle.

```
[root@archlinux ~]# ping -c 3 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) octets de données.
64 octets de 10.0.2.2 : icmp_seq=1 ttl=255 temps=0.728 ms
64 octets de 10.0.2.2 : icmp_seq=2 ttl=255 temps=0.574 ms
64 octets de 10.0.2.2 : icmp_seq=3 ttl=255 temps=0.590 ms

--- statistiques ping 10.0.2.2 ---
3 paquets transmis, 3 reçus, 0% packet loss, time 2028ms
rtt min/avg/max/mdev = 0.574/0.630/0.728/0.069 ms
[root@archlinux ~]# ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) octets de données.
64 octets de 8.8.8.8 : icmp_seq=1 ttl=255 temps=103 ms
64 octets de 8.8.8.8 : icmp_seq=2 ttl=255 temps=44.8 ms
64 octets de 8.8.8.8 : icmp_seq=3 ttl=255 temps=45.4 ms

--- statistiques ping 8.8.8.8 ---
3 paquets transmis, 3 reçus, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 44.769/64.223/102.502/27.068 ms
[root@archlinux ~]# ping -c 3 archlinux.org
PING archlinux.org (95.217.163.246) 56(84) octets de données.
64 octets de archlinux.org (95.217.163.246) : icmp_seq=1 ttl=255 temps=251 ms
64 octets de archlinux.org (95.217.163.246) : icmp_seq=2 ttl=255 temps=270 ms
64 octets de archlinux.org (95.217.163.246) : icmp_seq=3 ttl=255 temps=193 ms

--- statistiques ping archlinux.org ---
3 paquets transmis, 3 reçus, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 193.302/238.264/270.180/32.712 ms
[root@archlinux ~]#
```

Les captures d'écran montrent chaque étape, des configurations manuelles à la validation du ping vers des hôtes externes.

## Installation et démarrage de dhcpd

Pour automatiser la gestion du réseau, nous avons installé le gestionnaire DHCP

```
[root@archlinux ~]# pacman -S dhcpd
résolution des dépendances#
recherche des conflits entre paquets#

Paquets (1) dhcpd-10.1.0-1
Taille totale du téléchargement : 0,20 MiB
Taille totale installée : 0,45 MiB

:: Procéder à l'installation ? [O/n] o
:: Récupération des paquets#
dhcpd-10.1.0-1-x86_64 206,0 KiB 209 KiB/s 00:01 [#####] 100%
(1/1) vérification des clés dans le trousseau [#####] 100%
(1/1) vérification de l'intégrité des paquets [#####] 100%
(1/1) chargement des fichiers des paquets [#####] 100%
(1/1) analyse des conflits entre fichiers [#####] 100%
(1/1) vérification de l'espace disque disponible [#####] 100%
:: Traitement des changements du paquet#
(1/1) installation de dhcpd [#####] 100%
Dépendances optionnelles pour dhcpd
openresolv: resolvconf support
:: Exécution des crochets (« hooks ») de post-transaction#
(1/4) Creating system user accounts...
Creating group 'dhcpd' with GID 972.
Creating user 'dhcpd' (dhcpd privilege separation) with UID 972 and GID 972.
(2/4) Reloading system manager configuration...
(3/4) Creating temporary files...
(4/4) Arming ConditionNeedsUpdate...
[root@archlinux ~]#
```

Ensuite, nous avons activé et démarré le service avec les commandes suivantes :

```
root@archlinux ~]# systemctl enable dhcpcd
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpcd.service' → '/usr/lib/systemd/system/dhcpcd.service'.
root@archlinux ~]# systemctl start dhcpcd
root@archlinux ~]# systemctl status dhcpcd
● dhcpcd.service - DHCP/ IPv4LL/ IPv6RA/ DHCPv6 client on all interfaces
   Loaded: loaded (/usr/lib/systemd/system/dhcpcd.service; enabled; preset:
 disabled)
   Active: active (running) since Sat 2024-12-28 05:08:20 CET; 20s ago
  Invocation: 1e75d1bf40a24239ac86f0d23c0c2c71
    Main PID: 476 (dhcpcd)
      Tasks: 4 (limit: 2285)
     Memory: 3.1M (peak: 5.2M)
        CPU: 139ms
    CGroup: /system.slice/dhcpcd.service
            └─476 "dhcpcd: {manager} [ip4] [ip6]"
              └─477 "dhcpcd: {privileged proxy}"
                └─478 "dhcpcd: {network proxy}"
                  └─479 "dhcpcd: {control proxy}"

déc. 28 05:08:21 archlinux dhcpcd[477]: emp0s3: offered 10.0.2.15 from 10.0.2.2
déc. 28 05:08:21 archlinux dhcpcd[477]: emp0s3: leased 10.0.2.15 for 86400 seconds
déc. 28 05:08:21 archlinux dhcpcd[477]: emp0s3: adding route to 10.0.2.0/24
déc. 28 05:08:21 archlinux dhcpcd[477]: emp0s3: adding default route via 10.0.2.2
déc. 28 05:08:21 archlinux dhcpcd[477]: emp0s3: pid 0 deleted route to 10.0.2.0/24
déc. 28 05:08:22 archlinux dhcpcd[477]: emp0s3: soliciting an IPv6 router
déc. 28 05:08:22 archlinux dhcpcd[477]: emp0s3: Router Advertisement from fe80::2
déc. 28 05:08:22 archlinux dhcpcd[477]: emp0s3: adding address fa00::2590:243c:2fa9:ce9b/64
déc. 28 05:08:22 archlinux dhcpcd[477]: emp0s3: adding route to fa00:::64
déc. 28 05:08:22 archlinux dhcpcd[477]: emp0s3: adding default route via fe80::2
root@archlinux ~]#
```

Le service a attribué automatiquement une adresse IP et configuré la passerelle ainsi que les serveurs DNS

```
root@archlinux ~]# ping -c 3 archlinux.org
PING archlinux.org (95.217.163.246) 56(84) octets de données:
64 octets de archlinux.org (95.217.163.246) : icmp_seq=1 ttl=255 temps=197 ms
64 octets de archlinux.org (95.217.163.246) : icmp_seq=2 ttl=255 temps=105 ms
64 octets de archlinux.org (95.217.163.246) : icmp_seq=3 ttl=255 temps=106 ms

--- statistiques ping archlinux.org ---
3 paquets transmis, 3 reçus, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 105.457/136.326/197.258/43.086 ms
```

## Note sur la configuration réseau

La configuration réseau manuelle effectuée initialement a permis de résoudre temporairement le problème d'accès au réseau. Une fois le gestionnaire DHCP dhcpcd installé et activé, il a pris en charge la gestion automatique des paramètres réseau, remplaçant la configuration manuelle. Ce service assure une connectivité réseau fiable et persistante après chaque redémarrage.

## ÉTAPE 11 : INSTALLATION ET CONFIGURATION DE L'ENVIRONNEMENT GRAPHIQUE

Pour offrir une interface utilisateur fluide et légère, l'environnement graphique XFCE4 a été installé et configuré, accompagné d'un gestionnaire de sessions. Cette étape inclut l'installation, le test, et la résolution des problèmes liés au choix initial du gestionnaire de sessions.

### Installation de l'environnement XFCE et des paquets nécessaires

Pour fournir une interface utilisateur légère et réactive, l'environnement graphique XFCE a été choisi pour sa légèreté et sa réactivité, particulièrement adaptées à un environnement minimaliste comme Arch Linux. Les commandes suivantes ont été utilisées

```
pacman -S xfce4 xfce4-goodies
```

- **xfce4** contient les composants principaux de l'environnement de bureau.

- **xfce4-goodies** inclut des outils supplémentaires pour enrichir l'expérience utilisateur.

Après l'installation, un test manuel a confirmé que XFCE4 fonctionnait correctement avec la commande **startxfce4**

**Résultat** : XFCE s'est chargé avec succès, confirmant que les composants graphiques étaient correctement installés.

## Gestion des sessions : Choix et configuration

### Problèmes rencontrés avec LightDM :

Dans un premier temps, LightDM a été choisi comme gestionnaire de sessions pour son intégration large et ses fonctionnalités étendues. Cependant, dès les premières configurations, des instabilités majeures ont été constatées, notamment :

- Écrans noirs persistants au démarrage.
- Échecs répétés lors de la connexion aux sessions.

L'analyse approfondie des journaux système avec **journalctl** a mis en évidence des erreurs récurrentes liées à l'authentification PAM et à la gestion des sessions. Malgré des ajustements, notamment la réinstallation et la modification des configurations, LightDM s'est révélé inadapté dans ce contexte particulier.

### Transition vers SLiM :

Pour surmonter ces limitations, SLiM (Simple Login Manager) a été choisi comme alternative. Ce gestionnaire de sessions est connu pour sa simplicité et sa compatibilité avec des configurations légères.

Les raisons principales du basculement vers SLiM incluent :

- **Légereté** : SLiM consomme moins de ressources système.
- **Simplicité de configuration** : Une interface minimaliste avec peu de dépendances.
- **Compatibilité** : Fonctionne de manière fiable avec XFCE4.

L'installation de SLiM a été réalisée avec : **pacman -S slim**

Ensuite, SLiM a été activé pour qu'il démarre automatiquement avec le système : **systemctl enable slim**

Le fichier de configuration `/etc/slim.conf` a été ajusté pour définir XFCE comme environnement par défaut :

```

GNU nano 0.2.2 /etc/slim.conf Modifié
# Valid values: true|false
#hidecursor      false

# This command is executed after a successful login.
# You can place the %session and %theme variables to handle launching of
# specific commands in xinitrc depending on chosen session and slim theme.
# Ensure that the command handles an empty %session, as that is the default
#
# NOTE: if your system does not have bash you need to adjust the command
# according to your preferred shell, e.g. for freebsd use:
# login_cmd      exec /bin/sh - ~/.xinitrc %session
login_cmd        exec startxfce4
#
# Commands executed when starting and exiting a session.
# They can be used for registering a X11 session with
# sssreg. You can use the %user variable
#
# sessionstart_cmd  some command
# sessionstop_cmd   some command
#
# Start in daemon mode. Valid values: yes | no
# Note that this can be overridden by the command line options "-d" and "-n"
daemon           yes

# Available sessions:
# The current chosen session name replaces %session in the login_cmd
# above, so your login command can handle different sessions.
# If no session is chosen (via F1), %session will be an empty string. This
# allows the script to handle default in a user-specific manner, if desired.
# See the xinitrc.sample file shipped with slim sources.
#sessions         xfce4,icewm-session,unaker,blackbox

# Alternatively, read available sessions from the contents of a
# directory. The directory can contain either executable scripts,
# or %sessions Desktop files. In the case of Desktop files, the name
# displayed is the Name= value and the string substituted in place of
# %session is the Exec= value -- note that this may provide a full
# path to the session executable!
#sessiondir       /usr/share/xsessions

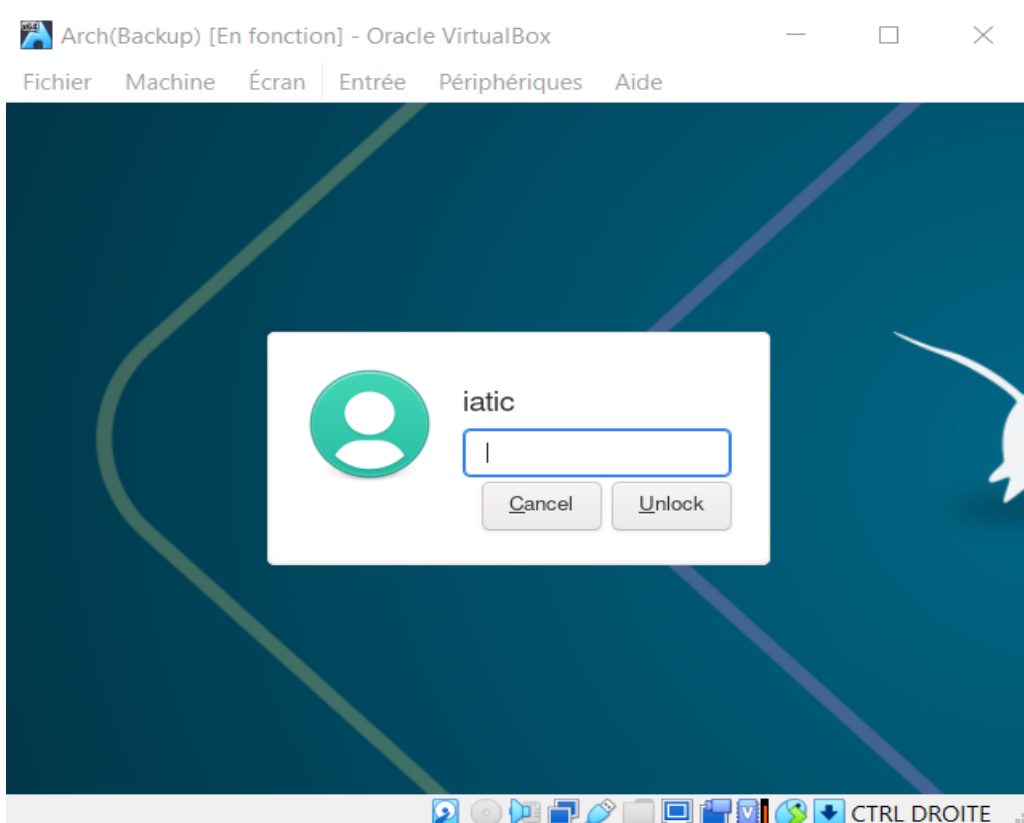
# Executed when pressing F11 (requires imagenagick)
#screenshot_cmd   import -window root /slim.png
# Alternatively using scrot.
#screenshot_cmd   scrot /root/slim.png

# Aide      Écrire      Chercher      Couper      Exécuter      Enplacement      Annuler      Marquer      -> Crochet      Précédent
# Quitter   Lire fich.  Remplacer    Coller      Justifier     Aller ligne     Refaire     Copier      Retrouver    Suivant

```

## Résultat :

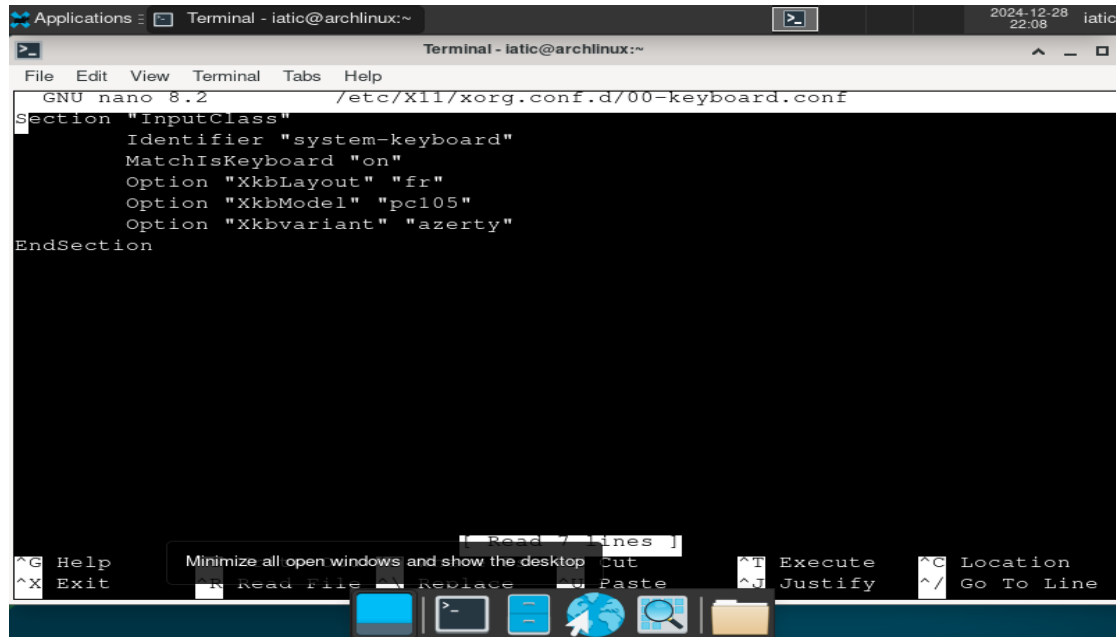
Après ces modifications, SLiM a permis un démarrage des sessions stable et fluide. Contrairement à LightDM, il s'est avéré parfaitement adapté à nos besoins. Bien que SLiM soit moins riche en fonctionnalités, il a offert une expérience utilisateur fiable, en adéquation avec les contraintes techniques du projet.



Après avoir installé et activé XFCE4 en tant qu'environnement graphique principal, des ajustements ont été effectués pour optimiser l'expérience utilisateur et répondre aux exigences spécifiques du projet.

### Configuration du clavier pour Slim :

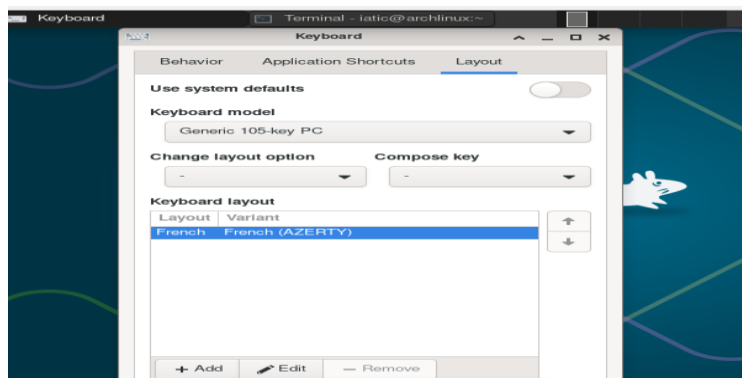
- Le clavier a été configuré pour utiliser la disposition française (AZERTY). Pour cela, un fichier de configuration a été ajouté dans /etc/X11/xorg.conf.d/00-keyboard.conf avec les paramètres suivants :



```
GNU nano 8.2 /etc/X11/xorg.conf.d/00-keyboard.conf
Section "InputClass"
Identifier "system-keyboard"
MatchIsKeyboard "on"
Option "XkbLayout" "fr"
Option "XkbModel" "pc105"
Option "Xkbvariant" "azerty"
EndSection
```

### Configuration des langues pour XFCE4 :

Pour compléter l'expérience utilisateur et assurer une uniformité dans les saisies, nous avons configuré la disposition du clavier en français (AZERTY) au sein de l'environnement XFCE4. Cette configuration a été réalisée à l'aide du gestionnaire de paramètres de XFCE4, dans la section dédiée au clavier.

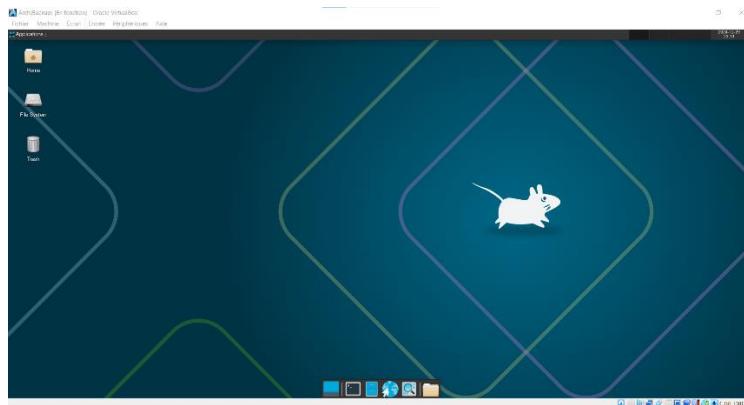


## Vérification finale de l'environnement

1. Démarrage stable avec SLiM : SLiM a permis de lancer XFCE sans problème.

2. Interface utilisateur :

- Après démarrage, le bureau XFCE4 a été correctement chargé, affichant les icônes système par défaut, comme "Home", "Trash", et le gestionnaire de fichiers. Une navigation fluide entre les applications a été confirmée (voir capture d'écran du bureau).



3. Applications :

- Le navigateur Firefox, installé précédemment, a été testé pour vérifier la connectivité réseau et l'accès à des sites externes. Le test a confirmé que la configuration réseau et l'environnement graphique fonctionnent sans anomalies.



Le système Arch Linux configuré intègre désormais :

- Un environnement graphique léger et stable avec XFCE4.
- Une gestion des sessions fiable et fluide via SLiM.
- Une configuration réseau et clavier optimisée pour les besoins de l'utilisateur.

Ce projet a permis d'acquérir une compréhension approfondie des concepts de configuration système, tout en atteignant un résultat final parfaitement fonctionnel et adapté aux exigences initiales. Les captures d'écran fournissent une illustration claire des résultats finaux et de l'expérience utilisateur réussie.

### ÉVALUATION DES COMPÉTENCES ACQUISES :

La réalisation de ce projet a permis de développer des compétences essentielles en administration système et en gestion des environnements Linux. Les principales compétences acquises sont les suivantes :

- 1. Compréhension approfondie des mécanismes d'installation et de configuration d'Arch Linux :**
  - Ce projet a exigé une maîtrise des outils de partitionnement, de gestion des paquets, et des mécanismes de démarrage. Chaque étape, de la configuration des partitions à l'installation de l'environnement graphique, a contribué à une compréhension détaillée des processus internes de Linux.
- 2. Gestion et résolution des problèmes techniques :**
  - Les défis rencontrés, tels que les instabilités de LightDM, le basculement vers SLiM, et les configurations réseau manuelles, ont renforcé les compétences en diagnostic et en résolution de problèmes. L'analyse des journaux système a permis de trouver des solutions adaptées à chaque situation.
- 3. Configuration optimisée de l'environnement utilisateur :**
  - La mise en place d'un clavier en disposition française (AZERTY) et la personnalisation de l'expérience utilisateur avec XFCE4 ont complété une compréhension pratique des outils et fichiers de configuration.
- 4. Documentation détaillée et rigoureuse :**
  - Chaque étape a été documentée de manière claire et illustrée par des captures d'écran. Cette démarche a permis de produire un rapport structuré, pouvant servir de référence pour des installations futures.

## CONCLUSION :

Ce projet d'installation et de configuration d'Arch Linux sur une machine virtuelle a été une expérience enrichissante et formatrice, permettant d'explorer en profondeur les concepts d'administration système et de personnalisation avancée d'un environnement Linux.

L'approche minimaliste d'Arch Linux a exigé une compréhension détaillée des différentes étapes nécessaires pour obtenir un système pleinement fonctionnel. Chaque composant, de la gestion des partitions à l'installation des paquets essentiels, en passant par la configuration réseau et la mise en place d'une interface graphique, a été soigneusement étudié et documenté.

Les défis rencontrés, notamment les problèmes initiaux de réseau et l'instabilité du gestionnaire de sessions LightDM, ont été des opportunités d'apprentissage. La transition vers des solutions plus adaptées, comme SLiM pour la gestion des sessions, illustre l'importance de l'adaptabilité dans l'administration système.

Le résultat final est un système Arch Linux stable et performant, intégrant l'environnement graphique XFCE4, une configuration réseau fonctionnelle et une personnalisation optimisée pour une expérience utilisateur fluide. Ce projet a permis non seulement d'atteindre les objectifs initiaux, mais aussi d'approfondir des compétences essentielles, telles que la résolution de problèmes techniques, l'analyse de journaux système, et la documentation détaillée des étapes et configurations.

En conclusion, cette réalisation marque une étape clé dans le développement de compétences en administration système et gestion d'environnements Linux avancés, offrant une base solide pour des projets futurs et des applications dans des contextes professionnels. Ce rapport, enrichi de commentaires et de captures d'écran, servira de référence précieuse pour des installations similaires pourra servir de référence pour des projets similaires ou des implémentations futures.

## Référence :

<https://archlinux.org/>