Maintenir et Exploiter un Serveur Linux Debian 11

Table des matières

Module 1 : Installation d'un serveur Debian

Dans ce module, vous apprendrez comment installer Debian 11 sur un serveur, étape par étape. Debian est une distribution Linux stable, populaire et idéale pour les serveurs. L'installation sur un serveur peut se faire de différentes manières, mais nous allons décrire la méthode la plus courante et simple : l'installation depuis une image ISO.

Objectif :

- Installer un serveur Debian 11 depuis une image ISO.
- Configurer les paramètres réseau, disque et utilisateurs pour un serveur fonctionnel.

---

1. Prérequis pour l'installation

Avant de commencer l'installation de Debian 11, vous devez vous assurer de disposer des éléments suivants :

- **Une machine physique ou virtuelle (VM)** : Vous devez disposer d'un serveur physique ou d'une machine virtuelle (VM) pour installer Debian.
- **Une clé USB bootable** (si vous installez sur une machine physique) ou un serveur virtuel (si vous installez sur une machine virtuelle).
- **Une image ISO de Debian 11** : Vous pouvez télécharger la dernière version stable de Debian depuis le site officiel de Debian : Télécharger Debian 11.
- **Accès à Internet** pour télécharger les mises à jour ou installer des paquets supplémentaires pendant l'installation.

---

2. Création d'une clé USB bootable avec l'ISO de Debian

Si vous installez Debian sur un serveur physique, vous devez d'abord créer une clé USB bootable. Pour cela, suivez ces étapes :

1. **Téléchargez l'image ISO** de Debian 11 depuis le site officiel.
2. **Utilisez un outil comme Rufus (Windows)** ou **Etcher (Linux/Mac)** pour créer une clé USB bootable :
    - **Rufus (Windows)** : Sélectionnez la clé USB et l'image ISO téléchargée, puis cliquez sur "Démarrer".
    - **Etcher (Linux/Mac)** : Sélectionnez l'image ISO et la clé USB, puis cliquez sur "Flash!".
3. Une fois la clé USB créée, insérez-la dans votre serveur et redémarrez-le en configurant le BIOS ou UEFI pour démarrer depuis la clé USB.

---

3. Étapes de l'installation de Debian 11

L'installation de Debian se fait via un programme d'installation texte ou graphique. Pour simplifier, nous allons utiliser l'installation graphique, plus conviviale.

Étape 1 : Démarrage à partir de la clé USB

1. **Démarrer le serveur à partir de la clé USB** : Vous verrez le menu d'installation de Debian avec plusieurs options. Choisissez la première option "Install" pour une installation graphique standard.

   Note : Si vous utilisez un serveur virtuel, il suffit de spécifier l'ISO comme média d'installation et démarrer la VM.

Étape 2 : Sélectionner la langue et la disposition du clavier

1. Lors du démarrage, l'installateur vous demandera de choisir votre **langue** (français, dans ce cas) et la **disposition du clavier** (par exemple, "Français").
2. Confirmez ces choix en cliquant sur "Continuer".

Étape 3 : Configurer le réseau

1. **Configuration de l'interface réseau** : Si vous êtes connecté à un réseau local, choisissez l'option pour utiliser DHCP (généralement automatique). Si vous configurez un serveur avec une IP statique, entrez les informations nécessaires (adresse IP, masque de sous-réseau, passerelle et serveur DNS).
2. **Nom de l'hôte** : L'installateur vous demandera de choisir un nom pour votre serveur. Il s'agit généralement d'un nom simple comme `serveur1` ou `webserver`. Ce nom sera utilisé pour identifier le serveur sur le réseau.

Étape 4 : Partitionnement du disque

1. **Choix de la méthode de partitionnement** : L'installateur propose plusieurs méthodes de partitionnement :
   - **Guidé** : Utilise tout l'espace du disque et partitionne automatiquement. Cette option est la plus simple et recommandée si vous n'avez pas de besoins spécifiques.
   - **Partitionnement manuel** : Cette option vous permet de créer des partitions personnalisées, mais elle est recommandée pour les utilisateurs avancés.
2. **Sélection de la méthode de partitionnement** : Choisissez "Guidé - Utiliser tout le disque" pour une installation simple.
   - Si vous avez plusieurs disques durs, l'installateur vous demandera de sélectionner le disque sur lequel vous souhaitez installer Debian.
3. **Sélection des partitions** :
   - Le partitionnement automatique va créer les partitions nécessaires, telles que `/`, `swap`, et `home`. En général, les partitions recommandées sont suffisantes.
4. **Validation du partitionnement** : Après avoir choisi vos partitions, l'installateur vous demandera de confirmer. Tapez "oui" pour valider.

Étape 5 : Installation du système de base

L'installateur va maintenant installer les fichiers de base de Debian. Cela peut prendre un certain temps en fonction de la vitesse de votre matériel.

1. Une fois l'installation terminée, il vous sera demandé de configurer le gestionnaire de paquets.

Étape 6 : Configuration de l'utilisateur et du mot de passe

1. **Mot de passe de l'utilisateur root** : Debian ne vous permet pas de vous connecter directement en tant que root pour des raisons de sécurité. Cependant, vous devrez entrer un mot de passe pour le superutilisateur (root).
2. **Création d'un utilisateur normal** : Créez un utilisateur avec des droits sudo. Ce compte vous permettra d'exécuter des commandes administratives sans vous connecter en tant que root.

   Exemple :

   - Nom : `admin`
   - Mot de passe : Choisissez un mot de passe sécurisé.

Étape 7 : Sélection des logiciels à installer

1. **Sélection des logiciels** : L'installateur vous permettra de choisir certains logiciels à installer, comme le serveur web Apache, Nginx, ou OpenSSH pour l'accès à distance.
    - Si vous prévoyez d'administrer le serveur à distance, choisissez "Serveur SSH".
    - Si vous avez l'intention de configurer un serveur web, vous pouvez choisir "Serveur web" ou installer un paquet comme Nginx ou Apache plus tard.
2. **Installation des logiciels** : L'installateur va télécharger et installer les logiciels que vous avez choisis.

Étape 8 : Installation du chargeur de démarrage GRUB

Le chargeur de démarrage GRUB permet de charger le noyau Linux lors du démarrage de votre serveur.

1. Lorsqu'on vous le demande, choisissez d'installer GRUB sur le disque dur principal (habituellement `/dev/sda`).
2. Confirmez et l'installation sera terminée.

---

4. Finalisation de l'installation

Une fois l'installation terminée, l'installateur vous proposera de redémarrer votre serveur. Assurez-vous de retirer la clé USB pour éviter de redémarrer l'installation.

1. **Redémarrage** : Cliquez sur "Redémarrer maintenant". Le serveur va redémarrer, et vous devriez voir le système Debian 11 démarrer sur votre serveur.
2. **Connexion** : Une fois le serveur redémarré, vous pouvez vous connecter avec l'utilisateur que vous avez créé précédemment. Vous serez invité à entrer le nom d'utilisateur et le mot de passe.

---

5. Accéder au serveur via SSH

Si vous avez installé le serveur SSH durant l'installation, vous pouvez vous connecter à votre serveur à distance en utilisant un autre ordinateur.

1. Depuis un autre ordinateur, ouvrez un terminal (Linux/Mac) ou un client SSH (comme PuTTY pour Windows).
2. Tapez la commande suivante pour vous connecter :

```
ssh utilisateur@ip_du_serveur
```

Si tout est configuré correctement, vous serez connecté à votre serveur Debian 11 à distance.

---

6. Vérification de l'installation

Pour vérifier que tout fonctionne bien, vous pouvez :

- **Vérifier la version de Debian** :

```
lsb_release -a
```

- **Vérifier la connectivité réseau** :

```
ping google.com
```

Module 2 : La gestion de son système Linux

Dans ce module, vous apprendrez à gérer et administrer un système Debian à travers des commandes de gestion de système essentielles. L'objectif est de vous rendre autonome dans la gestion de votre serveur Debian, de la mise à jour du système à la gestion des utilisateurs et des services.

Objectif :

- Gérer les utilisateurs, les paquets, les services, et surveiller le système Debian de manière efficace.
- Utiliser les outils essentiels de Debian pour maintenir et administrer un serveur de manière optimale.

2.1 Gestion des utilisateurs et des groupes

Dans un système Linux, les utilisateurs et les groupes sont des concepts fondamentaux pour gérer les permissions et les accès.

2.1.1 Ajouter un utilisateur

Pour créer un nouvel utilisateur, utilisez la commande `adduser` :

```
sudo adduser nom_utilisateur
```

Cette commande va :

- Créer l'utilisateur.
- Demander un mot de passe pour l'utilisateur.
- Demander des informations supplémentaires comme le nom complet (vous pouvez appuyer sur Entrée pour les ignorer).
- Créer le répertoire personnel de l'utilisateur (`/home/nom_utilisateur`).

2.1.2 Ajouter un utilisateur à un groupe

Dans Linux, chaque utilisateur appartient à un ou plusieurs groupes. Par défaut, chaque utilisateur appartient à un groupe du même nom. Pour ajouter un utilisateur à un groupe spécifique (par exemple, le groupe `sudo` pour les privilèges administratifs), utilisez la commande `usermod` :

```
sudo usermod -aG groupe nom_utilisateur
```

Exemple pour ajouter un utilisateur au groupe `sudo` :

```
sudo usermod -aG sudo nom_utilisateur
```

2.1.3 Supprimer un utilisateur

Si vous souhaitez supprimer un utilisateur, vous pouvez utiliser la commande `deluser` :

```
sudo deluser nom_utilisateur
```

Pour supprimer aussi le répertoire personnel de l'utilisateur, ajoutez l'option `--remove` :

```
sudo deluser --remove-home nom_utilisateur
```

2.1.4 Changer de mot de passe

Pour changer le mot de passe d'un utilisateur, utilisez la commande `passwd` :

```
sudo passwd nom_utilisateur
```

Cette commande vous demandera de saisir un nouveau mot de passe.

2.1.5 Vérification des utilisateurs et groupes

Pour vérifier les utilisateurs existants, vous pouvez consulter le fichier `/etc/passwd` :

```
cat /etc/passwd
```

Pour voir les groupes existants, vous pouvez consulter le fichier `/etc/group` :

```
cat /etc/group
```

2.2 Gestion des paquets

Les paquets sont des éléments essentiels dans Debian. La gestion des paquets est réalisée à l'aide de l'outil `APT` (Advanced Package Tool).

2.2.1 Mettre à jour les paquets du système

Avant de commencer à installer de nouveaux logiciels, il est important de mettre à jour la liste des paquets disponibles et les paquets installés sur le système.

1. **Mettre à jour la liste des paquets** :

```
sudo apt update
```

2. **Mettre à jour les paquets installés** :

```
sudo apt upgrade
```

Cela installera les dernières versions des paquets déjà installés.

2.2.2 Installer un paquet

Pour installer un nouveau paquet, utilisez la commande `apt install`. Par exemple, pour installer `curl` :

```
sudo apt install curl
```

L'outil `apt` va chercher le paquet dans les dépôts et l'installer automatiquement.

2.2.3 Désinstaller un paquet

Si vous souhaitez supprimer un paquet, vous pouvez utiliser la commande `apt remove`. Par exemple, pour supprimer `curl` :

```
sudo apt remove curl
```

Cela désinstalle le paquet, mais conserve les fichiers de configuration. Si vous souhaitez supprimer complètement le paquet (y compris les fichiers de configuration), utilisez :

```
sudo apt purge curl
```

2.2.4 Rechercher un paquet

Si vous ne connaissez pas le nom exact du paquet, vous pouvez le rechercher dans les dépôts en utilisant la commande `apt search`. Par exemple, pour rechercher tous les paquets contenant "nginx" :

```
apt search nginx
```

2.2.5 Vérification des paquets installés

Pour voir la liste des paquets installés sur votre système, utilisez la commande :

```
dpkg --get-selections
```

Pour obtenir des informations détaillées sur un paquet spécifique (comme `nginx`), utilisez :

```
apt show nginx
```

2.3 Gestion des services

Linux utilise le système `systemd` pour la gestion des services. Vous pouvez démarrer, arrêter, redémarrer ou vérifier l'état des services en utilisant la commande `systemctl`.

2.3.1 Vérifier l'état d'un service

Pour vérifier si un service est actif, utilisez la commande `systemctl status` suivie du nom du service. Par exemple, pour vérifier l'état du service `nginx` :

```
sudo systemctl status nginx
```

Cette commande affiche si le service est en cours d'exécution, son état actuel, ainsi que les journaux associés.

2.3.2 Démarrer, arrêter et redémarrer un service

- **Démarrer un service** :

  ```
  sudo systemctl start nginx
  ```

- **Arrêter un service** :

  ```
  sudo systemctl stop nginx
  ```

- **Redémarrer un service** :

  ```
  sudo systemctl restart nginx
  ```

2.3.3 Activer ou désactiver un service au démarrage

- **Activer un service au démarrage** (c'est-à-dire démarrer automatiquement lors du démarrage du serveur) :

  ```
  sudo systemctl enable nginx
  ```

- **Désactiver un service au démarrage** (cela empêche le service de démarrer automatiquement lors du démarrage) :

  ```
  sudo systemctl disable nginx
  ```

2.3.4 Voir les journaux des services

Vous pouvez consulter les journaux des services gérés par `systemd` avec la commande `journalctl` :

```
sudo journalctl -u nginx
```

Cela vous montre tous les logs du service `nginx`.

2.4 Surveillance du système

Surveiller les ressources du système est essentiel pour assurer le bon fonctionnement d'un serveur. Debian fournit plusieurs outils pour vérifier l'utilisation de la mémoire, du processeur et du disque.

2.4.1 Utilisation de `top`

La commande `top` est un outil puissant pour surveiller l'utilisation du processeur et de la mémoire en temps réel. Il montre les processus en cours et leur consommation de ressources.

```
Top
```

2.4.2 Utilisation de `htop`

`htop` est une version améliorée de `top`, offrant une interface plus lisible et interactive. Pour l'installer :

```
sudo apt install htop
```

Une fois installé, lancez-le avec la commande :

```
htop
```

2.4.3 Utilisation de `free`

Pour voir l'utilisation de la mémoire sur votre système, vous pouvez utiliser la commande `free` :

```
free -h
```

Cela affiche la quantité de mémoire utilisée, libre et la mémoire swap.

2.4.4 Utilisation de `df`

La commande `df` permet de voir l'espace disque disponible sur votre système :

```
df -h :Cela affiche l'espace disque de chaque partition de manière lisible (en Go ou To).
```

2.4.5 Vérification de l'espace disque utilisé par un répertoire

Si vous voulez savoir combien d'espace est utilisé par un répertoire spécifique, utilisez la commande `du` :

```
du -sh /chemin/du/répertoire
```

2.5 Gestion des mises à jour et des patchs de sécurité

Une partie essentielle de la gestion d'un serveur est de garder le système à jour pour éviter les failles de sécurité.

2.5.1 Mise à jour régulière du système

Il est fortement recommandé de mettre régulièrement à jour votre système pour corriger les vulnérabilités de sécurité. Utilisez les commandes suivantes pour mettre à jour votre serveur Debian :

```
sudo apt update
sudo apt upgrade
```

2.5.2 Mettre à jour uniquement les paquets de sécurité

Si vous souhaitez mettre à jour uniquement les paquets de sécurité, vous pouvez utiliser le paquet `apt-get` avec l'option `dist-upgrade` :

```
sudo apt-get dist-upgrade
```

Cela mettra à jour tous les paquets installés et apportera les corrections de sécurité nécessaires.

Module 3 : Installation d'OpenSSH

Dans ce module, vous apprendrez à installer et à configurer OpenSSH sur un serveur Debian. OpenSSH (Open Secure Shell) est un ensemble d'outils permettant de se connecter à un serveur à distance de manière sécurisée. Il est principalement utilisé pour les connexions SSH (Secure Shell), mais il inclut également des outils comme SFTP (Secure File Transfer Protocol) pour le transfert de fichiers sécurisé.

Objectif :

- Installer OpenSSH sur un serveur Debian.
- Configurer OpenSSH pour une utilisation sécurisée.
- Se connecter à un serveur Debian via SSH.
- Apprendre à sécuriser l'accès SSH pour éviter les vulnérabilités.

3.1 Présentation de OpenSSH

OpenSSH est une version libre et sécurisée du protocole SSH, qui permet d'accéder à un serveur à distance de manière sécurisée en cryptant les communications. Il est largement utilisé pour administrer les serveurs de manière distante.

Les fonctionnalités principales d'OpenSSH incluent :

- **Connexion sécurisée** : Pour accéder à un serveur à distance de manière cryptée.
- **Transfert de fichiers** : Utilisation de SFTP ou SCP pour transférer des fichiers de manière sécurisée.
- **Tunnel SSH** : Création de tunnels sécurisés pour contourner des pare-feu ou accéder à des réseaux privés.

3.2 Installation d'OpenSSH sur Debian

L'installation d'OpenSSH sur un serveur Debian est simple grâce au gestionnaire de paquets APT.

Étape 1 : Mise à jour du système

Avant d'installer OpenSSH, assurez-vous que votre système est à jour pour éviter d'installer des versions obsolètes ou vulnérables des paquets.

```
sudo apt update
sudo apt upgrade
```

Étape 2 : Installer le paquet OpenSSH

Pour installer le serveur OpenSSH, utilisez la commande suivante :

```
sudo apt install openssh-server
```

Cela installera le serveur OpenSSH ainsi que les outils nécessaires pour se connecter à distance via SSH.

Étape 3 : Vérifier que OpenSSH est bien installé

Une fois l'installation terminée, vérifiez que le service SSH fonctionne correctement. Utilisez la commande `systemctl` pour vérifier l'état du service :

```
sudo systemctl status ssh
```

Vous devriez voir une sortie indiquant que le service SSH est **actif (en cours d'exécution)** :

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-04-24 12:30:56 UTC; 10min ago
```

Si le service n'est pas en cours d'exécution, démarrez-le avec la commande suivante :

```
sudo systemctl start ssh
```

Étape 4 : Activer le démarrage automatique d'OpenSSH au démarrage

Pour que OpenSSH démarre automatiquement lors du démarrage du système, utilisez la commande suivante :

```
sudo systemctl enable ssh
```

3.3 Se connecter au serveur via SSH

Une fois OpenSSH installé et configuré sur votre serveur Debian, vous pouvez vous connecter à ce dernier depuis un autre ordinateur à l'aide du client SSH.

Étape 1 : Se connecter à un serveur Debian via SSH

Depuis un autre ordinateur, utilisez la commande suivante pour établir une connexion SSH avec votre serveur Debian :

```
ssh utilisateur@ip_du_serveur
```

- **utilisateur** : Le nom d'utilisateur sur le serveur Debian auquel vous vous connectez (par exemple `root` ou un utilisateur standard comme `admin`).
- **ip_du_serveur** : L'adresse IP de votre serveur Debian.

Par exemple, si votre serveur a l'adresse IP `192.168.1.10` et que vous utilisez l'utilisateur `admin` :

```
ssh admin@192.168.1.10
```

Si vous vous connectez pour la première fois, vous recevrez un avertissement de sécurité concernant l'empreinte de la clé du serveur. Tapez **"yes"** pour continuer. Ensuite, il vous sera demandé de saisir le mot de passe de l'utilisateur.

3.4 Configuration d'OpenSSH

Une fois OpenSSH installé, vous pouvez personnaliser plusieurs options de configuration selon vos besoins. Le fichier principal de configuration d'OpenSSH est `/etc/ssh/sshd_config`.

Étape 1 : Accéder au fichier de configuration

Pour modifier la configuration, ouvrez le fichier `/etc/ssh/sshd_config` avec un éditeur de texte comme `nano` :

```
sudo nano /etc/ssh/sshd_config
```

Étape 2 : Options de configuration importantes

Voici quelques paramètres de configuration courants que vous pouvez ajuster dans ce fichier pour sécuriser votre serveur SSH :

- **Port** : Par défaut, SSH écoute sur le port 22. Il est recommandé de changer ce port pour un autre (par exemple, 2222) afin de réduire les attaques automatisées.

  ```
  Port 2222
  ```

- **PermitRootLogin** : Désactivez la connexion directe en tant que root pour des raisons de sécurité. Il est recommandé d'utiliser un utilisateur normal et d'employer `sudo` pour des privilèges élevés.

  ```
  PermitRootLogin no
  ```

- **PasswordAuthentication** : Désactivez l'authentification par mot de passe pour renforcer la sécurité. Utilisez des clés SSH pour vous connecter.

  ```
  PasswordAuthentication no
  ```

- **AllowUsers** : Si vous souhaitez limiter l'accès SSH à des utilisateurs spécifiques, vous pouvez ajouter la directive `AllowUsers`.

  ```
  AllowUsers admin
  ```

- **PermitEmptyPasswords** : Assurez-vous que cette option est désactivée pour interdire les mots de passe vides.

  ```
  PermitEmptyPasswords no
  ```

Étape 3 : Redémarrer le service SSH

Après avoir effectué des modifications dans le fichier de configuration, vous devez redémarrer le service SSH pour appliquer les changements :

```
sudo systemctl restart ssh
```

3.5 Utilisation des clés SSH pour une connexion sans mot de passe

Une des meilleures pratiques de sécurité consiste à se connecter à un serveur SSH sans mot de passe, en utilisant une paire de clés SSH. Voici comment configurer l'authentification par clé SSH :

Étape 1 : Générer une paire de clés SSH

Sur votre ordinateur local (le client), générez une paire de clés SSH avec la commande suivante :

```
ssh-keygen -t rsa -b 4096
```

Cela va générer deux fichiers :

- **id_rsa** : La clé privée (à ne jamais partager).
- **id_rsa.pub** : La clé publique (à copier sur le serveur).

Étape 2 : Copier la clé publique sur le serveur

Pour ajouter la clé publique à votre serveur Debian, utilisez la commande `ssh-copy-id`. Cela ajoute automatiquement votre clé publique au fichier `~/.ssh/authorized_keys` de l'utilisateur sur le serveur.

```
ssh-copy-id utilisateur@ip_du_serveur
```

Cela vous demandera le mot de passe de l'utilisateur sur le serveur. Une fois que vous avez entré le mot de passe, la clé publique sera copiée dans le fichier autorisé, et vous pourrez vous connecter sans mot de passe.

Étape 3 : Vérifier la connexion sans mot de passe

Maintenant, vous pouvez vous connecter au serveur sans entrer de mot de passe en utilisant :

```
ssh utilisateur@ip_du_serveur
```

3.6 Sécuriser l'accès SSH

Il existe plusieurs bonnes pratiques pour sécuriser votre accès SSH. Voici quelques recommandations supplémentaires :

1. **Limiter les tentatives de connexion SSH avec `fail2ban`** : Cela permet de bloquer les adresses IP qui tentent trop de se connecter sans succès.
2. **Utiliser des pare-feu pour restreindre les accès SSH** : Utilisez un pare-feu pour limiter l'accès SSH uniquement à certaines adresses IP ou sous-réseaux.
3. **Configurer des VPN ou des tunnels SSH pour plus de sécurité** : Cela permet de chiffrer encore plus les connexions en ajoutant un niveau de sécurité supplémentaire.

Module 4 : Les commandes de base Linux Debian 11

Ce module vous introduit aux commandes de base de Linux, spécifiquement pour la distribution Debian 11. Ces commandes sont essentielles pour interagir avec le système, gérer les fichiers, les utilisateurs, et effectuer des tâches administratives. Vous apprendrez à utiliser le terminal efficacement pour gérer votre serveur Debian.

Objectif :

- Apprendre les commandes essentielles pour naviguer dans le système Debian 11.
- Comprendre comment manipuler les fichiers, les processus et les permissions.
- Acquérir des compétences de base pour administrer un serveur Debian en ligne de commande.

4.1 Navigation dans le système de fichiers

La gestion des fichiers et des répertoires est une partie fondamentale de l'administration système. Voici les commandes de base pour naviguer dans le système de fichiers.

4.1.1 `pwd` (Print Working Directory)

Cette commande affiche le répertoire dans lequel vous vous trouvez actuellement. C'est un moyen rapide de vérifier votre emplacement dans le système de fichiers.

```
Pwd
```

4.1.2 `ls` (List)

La commande `ls` liste les fichiers et répertoires dans le répertoire courant. Voici quelques options courantes :

- `ls` : Liste les fichiers et répertoires.
- `ls -l` : Affiche une liste détaillée avec des informations sur les fichiers (permissions, propriétaire, taille, etc.).
- `ls -a` : Affiche également les fichiers cachés (ceux dont le nom commence par un point).
- `ls -lh` : Affiche la taille des fichiers de manière lisible (par exemple, 1K, 2M, 3G).

Exemple :

```
ls -l
```

4.1.3 `cd` (Change Directory)

La commande `cd` vous permet de changer de répertoire.

- `cd /chemin/du/répertoire` : Vous déplace dans le répertoire spécifié.
- `cd ..` : Vous déplace un niveau en arrière (vers le répertoire parent).
- `cd` : Si utilisée seule, vous ramène dans votre répertoire personnel.

Exemple :

```
cd /home/user
```

4.1.4 `mkdir` (Make Directory)

Pour créer un nouveau répertoire, utilisez la commande `mkdir` :

```
mkdir mon_dossier
```

4.1.5 `rmdir` (Remove Directory)

Pour supprimer un répertoire vide, utilisez la commande `rmdir` :

```
rmdir mon_dossier
```

4.1.6 `rm` (Remove)

La commande `rm` est utilisée pour supprimer des fichiers ou des répertoires. Par défaut, elle supprime les fichiers sans demander de confirmation.

- Pour supprimer un fichier :

```
rm fichier.txt
```

- Pour supprimer un répertoire et son contenu :

```
rm -r mon_dossier
```

- Pour forcer la suppression sans confirmation :

```
rm -rf mon_dossier
```

4.2 Gestion des fichiers

Les fichiers dans Linux peuvent être manipulés de diverses manières. Voici les commandes pour lire, modifier, copier, et déplacer des fichiers.

4.2.1 `cat` (Concatenate)

La commande `cat` permet de visualiser le contenu d'un fichier.

```
cat fichier.txt
```

4.2.2 `nano` / `vim` (Éditeurs de texte)

- `nano` : Un éditeur de texte simple en ligne de commande.

  Exemple pour modifier un fichier avec `nano` :

```
nano fichier.txt
```

- `vim` : Un éditeur de texte plus puissant et complexe, mais plus flexible.

  Exemple pour ouvrir un fichier avec `vim` :

```
vim fichier.txt
```

4.2.3 `cp` (Copy)

La commande `cp` permet de copier des fichiers ou des répertoires. Par exemple :

- Copier un fichier :

```
cp fichier.txt /chemin/destination/
```

- Copier un répertoire et son contenu :

```
cp -r dossier/ /chemin/destination/
```

4.2.4 `mv` (Move)

La commande `mv` permet de déplacer ou renommer des fichiers et répertoires.

- Déplacer un fichier :

```
mv fichier.txt /chemin/destination/
```

- Renommer un fichier :

```
mv ancien_nom.txt nouveau_nom.txt
```

4.2.5 `rm` (Remove)

Déjà abordée précédemment, cette commande permet de supprimer des fichiers et des répertoires.

4.3 Gestion des permissions

Dans Linux, chaque fichier et répertoire a des permissions d'accès qui déterminent qui peut lire, écrire ou exécuter un fichier. Voici les commandes associées à la gestion des permissions.

4.3.1 `chmod` (Change Mode)

La commande `chmod` est utilisée pour modifier les permissions des fichiers et répertoires. Les permissions peuvent être définies en utilisant des chiffres ou des lettres.

- Pour donner des permissions en mode numérique (exemple : 755) :

```
chmod 755 fichier.txt
```

  Cela donne les permissions suivantes :

  - 7 : Lecture, écriture et exécution pour le propriétaire.
  - 5 : Lecture et exécution pour le groupe.
  - 5 : Lecture et exécution pour les autres.
- Pour donner des permissions en mode symbolique (exemple : `+x` pour ajouter l'exécution) :

```
chmod +x script.sh
```

4.3.2 `chown` (Change Owner)

La commande `chown` permet de changer le propriétaire et/ou le groupe d'un fichier.

- Pour changer le propriétaire :

```
sudo chown utilisateur fichier.txt
```

- Pour changer le groupe :

```
sudo chown :groupe fichier.txt
```

- Pour changer à la fois le propriétaire et le groupe :

```
sudo chown utilisateur:groupe fichier.txt
```

4.3.3 `chgrp` (Change Group)

La commande `chgrp` permet de changer le groupe d'un fichier sans toucher au propriétaire.

```
sudo chgrp groupe fichier.txt
```

4.4 Gestion des processus

Linux est un système multitâche, et il est essentiel de savoir comment gérer les processus qui s'exécutent.

4.4.1 `ps` (Process Status)

La commande `ps` permet de lister les processus en cours d'exécution.

- Pour afficher les processus de l'utilisateur actuel :

```
ps
```

- Pour afficher tous les processus :

```
ps aux
```

4.4.2 `top` (Task Manager)

La commande `top` affiche les processus en cours d'exécution en temps réel, ainsi que l'utilisation des ressources (CPU, mémoire).

```
Top
```

4.4.3 `kill` (Terminer un processus)

La commande `kill` permet d'envoyer un signal à un processus, souvent pour le terminer.

- Pour tuer un processus par son PID (identifiant de processus) :

```
kill 1234
```

- Pour forcer l'arrêt d'un processus (avec le signal `-9` qui tue immédiatement) :

```
kill -9 1234
```

4.4.4 `htop` (Enhanced top)

`htop` est une version améliorée de `top` qui fournit une interface interactive plus agréable pour surveiller les processus en cours. Vous devez l'installer avec la commande :

```
sudo apt install htop
```

Ensuite, lancez-le avec :

```
htop
```

4.5 Gestion des utilisateurs

Les utilisateurs sont une composante essentielle dans la gestion d'un serveur. Voici les commandes pour gérer les utilisateurs et les groupes.

4.5.1 `adduser` (Ajouter un utilisateur)

Pour ajouter un nouvel utilisateur :

```
sudo adduser nouvel_utilisateur
```

Cela crée un nouvel utilisateur et configure son répertoire personnel.

4.5.2 `usermod` (Modifier un utilisateur)

La commande `usermod` permet de modifier un utilisateur existant.

- Ajouter un utilisateur à un groupe :

```
sudo usermod -aG groupe utilisateur
```

4.5.3 `deluser` (Supprimer un utilisateur)

Pour supprimer un utilisateur et son répertoire personnel :

```
sudo deluser --remove-home utilisateur
```

4.6 Gestion des paquets

Debian utilise le gestionnaire de paquets APT pour installer, mettre à jour et supprimer des logiciels.

4.6.1 `apt update` (Mettre à jour la liste des paquets)

Avant d'installer de nouveaux paquets, vous devez mettre à jour la liste des paquets disponibles :

```
sudo apt update
```

4.6.2 `apt upgrade` (Mettre à jour les paquets installés)

Pour mettre à jour tous les paquets installés vers leur dernière version disponible :

```
sudo apt upgrade
```

4.6.3 `apt install` (Installer un paquet)

Pour installer un paquet :

```
sudo apt install nom_du_paquet
```

4.6.4 `apt remove` (Supprimer un paquet)

Pour supprimer un paquet :

```
sudo apt remove nom_du_paquet
```

Module 5 : Installation du serveur Nginx

Dans ce module, vous apprendrez à installer et configurer le serveur web **Nginx** sur un serveur Debian 11. Nginx est un serveur web léger, très performant et largement utilisé pour héberger des sites web, des applications et des services. Il est souvent préféré pour sa capacité à gérer un grand nombre de connexions simultanées avec une consommation de ressources relativement faible.

Objectifs :

- Installer Nginx sur un serveur Debian 11.
- Configurer les fichiers de base de Nginx.
- Vérifier le bon fonctionnement du serveur.
- Gérer Nginx via systemd.
- Déployer une page web simple.

5.1 Présentation de Nginx

Nginx (prononcé "Engine-X") est un serveur HTTP open-source qui peut aussi être utilisé comme reverse proxy, équilibrage de charge (load balancer) et serveur de cache. Il est conçu pour être hautement performant, stable et léger, ce qui en fait un choix populaire pour les serveurs web modernes.

Les principaux avantages de Nginx sont :

- **Haute performance** : Grâce à sa capacité à gérer des milliers de connexions simultanées avec une faible empreinte mémoire.
- **Reverse proxy et équilibrage de charge** : Nginx peut être configuré pour distribuer le trafic entre plusieurs serveurs d'applications.
- **Configuration simple** : Les fichiers de configuration sont basés sur du texte, ce qui facilite leur modification.

5.2 Installation de Nginx sur Debian 11

Étape 1 : Mettre à jour votre système

Avant d'installer Nginx, il est recommandé de mettre à jour la liste des paquets et de mettre à jour les paquets existants pour éviter les conflits.

```
sudo apt update
sudo apt upgrade
```

Étape 2 : Installer Nginx

L'installation de Nginx est simple sur Debian grâce à APT, le gestionnaire de paquets.

```
sudo apt install nginx
```

Cela va télécharger et installer Nginx ainsi que ses dépendances nécessaires.

Étape 3 : Vérifier l'installation de Nginx

Une fois Nginx installé, vous pouvez vérifier son statut pour vous assurer qu'il fonctionne correctement.

```
sudo systemctl status nginx
```

Cela devrait afficher une sortie indiquant que le service Nginx est **actif (en cours d'exécution)**. Si ce n'est pas le cas, vous pouvez démarrer le service avec :

```
sudo systemctl start nginx
```

Étape 4 : Activer Nginx au démarrage

Pour que Nginx démarre automatiquement lors du démarrage du système, vous devez activer le service avec la commande suivante :

```
sudo systemctl enable nginx
```

5.3 Vérification du fonctionnement de Nginx

Étape 1 : Accéder à la page d'accueil de Nginx

Par défaut, une page d'accueil est fournie avec l'installation de Nginx. Pour vérifier que Nginx fonctionne, ouvrez votre navigateur et entrez l'adresse IP de votre serveur dans la barre d'adresse :

```
http://votre-ip
```

Si tout est correctement installé, vous devriez voir la page par défaut de Nginx, qui ressemble à ceci :

Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working.

Étape 2 : Vérifier avec `curl`

Alternativement, vous pouvez vérifier en utilisant la commande `curl` depuis un terminal : `curl http://localhost`

5.4 Configurer Nginx

Étape 1 : Emplacement des fichiers de configuration

Les fichiers de configuration principaux de Nginx sont situés dans le répertoire `/etc/nginx/`. Les deux fichiers principaux sont :

- **/etc/nginx/nginx.conf** : Le fichier de configuration principal.
- **/etc/nginx/sites-available/** : Répertoire contenant les configurations des sites web individuels.
- **/etc/nginx/sites-enabled/** : Répertoire contenant les liens symboliques vers les configurations actives des sites.

Étape 2 : Configurer un site Web (vhost)

Supposons que vous souhaitez configurer un site web simple, tel qu'un site de test.

1. **Créer un fichier de configuration pour le site**

   Créez un fichier de configuration pour votre site dans le répertoire `/etc/nginx/sites-available/`. Par exemple, pour un site appelé "mon-site.com", créez un fichier `mon-site.com` :

   ```
   sudo nano /etc/nginx/sites-available/mon-site.com
   ```

   Ajoutez la configuration suivante pour un site basique :

   ```nginx
   server {
       listen 80;
       server_name mon-site.com www.mon-site.com;

       root /var/www/mon-site.com;
       index index.html;

       location / {
           try_files $uri $uri/ =404;
       }
   }
   ```

   - **listen 80** : Spécifie que le serveur écoute sur le port 80 pour HTTP.
   - **server_name** : Le nom de domaine (ou l'adresse IP) de votre site.
   - **root** : Le répertoire racine de votre site, où les fichiers HTML sont stockés.
   - **location /** : La section qui définit comment les requêtes sont traitées. Dans ce cas, nous utilisons `try_files` pour rechercher le fichier demandé et renvoyer une erreur 404 si le fichier n'existe pas.

2. **Créer un répertoire pour le site**

   Créez le répertoire où vous placerez les fichiers du site Web.

   ```
   sudo mkdir -p /var/www/mon-site.com
   ```

   Vous pouvez maintenant ajouter un fichier index HTML à ce répertoire :

   ```
   echo "<h1>Bienvenue sur mon-site.com</h1>" | sudo tee /var/www/mon-site.com/index.html
   ```

3. **Activer le site**

   Créez un lien symbolique vers ce fichier de configuration dans le répertoire `sites-enabled` pour activer le site.

   ```
   sudo ln -s /etc/nginx/sites-available/mon-site.com /etc/nginx/sites-enabled/
   ```

4. **Vérifier la configuration de Nginx**

   Avant de redémarrer Nginx, il est toujours prudent de vérifier la configuration pour vous assurer qu'il n'y a pas d'erreurs :

   ```
   sudo nginx -t
   ```

   Si tout est correct, vous verrez un message indiquant que la syntaxe est correcte et que le test est réussi.

5. **Redémarrer Nginx pour appliquer les changements**

   Une fois la configuration vérifiée, redémarrez Nginx pour appliquer les modifications :

   ```
   sudo systemctl restart nginx
   ```

6. **Accéder au site**

   Vous pouvez maintenant accéder à votre site en utilisant l'adresse IP ou le nom de domaine configuré (dans ce cas, `mon-site.com`).

5.5 Gestion des logs Nginx

Nginx génère des logs qui peuvent être utilisés pour surveiller le trafic et dépanner les problèmes.

Étape 1 : Accéder aux logs d'accès

Les logs d'accès de Nginx sont généralement situés dans le répertoire `/var/log/nginx/` :

- **/var/log/nginx/access.log** : Contient les informations sur chaque requête reçue par le serveur.

Étape 2 : Accéder aux logs d'erreur

Les logs d'erreur de Nginx sont également situés dans `/var/log/nginx/` :

- **/var/log/nginx/error.log** : Contient les erreurs générées par Nginx.

Vous pouvez consulter ces logs avec la commande `cat`, `less` ou `tail`. Par exemple, pour surveiller les logs d'accès en temps réel :

```
sudo tail -f /var/log/nginx/access.log
```

5.6 Sécuriser Nginx avec SSL (HTTPS)

Pour sécuriser votre site Web avec SSL, vous pouvez utiliser **Let's Encrypt**, qui fournit des certificats SSL gratuits.

Étape 1 : Installer Certbot

Certbot est un outil qui permet d'obtenir et de renouveler automatiquement des certificats SSL de Let's Encrypt.

```
sudo apt install certbot python3-certbot-nginx
```

Étape 2 : Obtenir un certificat SSL

Utilisez Certbot pour obtenir et configurer automatiquement le certificat SSL pour Nginx :

```
sudo certbot --nginx -d mon-site.com -d www.mon-site.com
```

Certbot va configurer Nginx pour utiliser SSL et rediriger les connexions HTTP vers HTTPS.

Module 6 : Installation du serveur LAMP

Dans ce module, nous allons installer et configurer un serveur LAMP sur un serveur Debian 11. LAMP est un acronyme pour **Linux, Apache, MySQL (ou MariaDB), et PHP**, qui sont les composants nécessaires pour exécuter des applications web dynamiques sur un serveur. Ce module vous guidera tout au long du processus d'installation de chacun des composants et de leur configuration.

Objectifs :

- Installer Apache, MySQL (ou MariaDB), et PHP sur un serveur Debian 11.
- Configurer chaque composant pour qu'il fonctionne ensemble.
- Tester le serveur LAMP pour vérifier qu'il fonctionne correctement.
- Déployer une application PHP simple.

6.1 Présentation de LAMP

LAMP est une pile de logiciels open-source utilisée pour héberger des sites web dynamiques. Les composants sont les suivants :

- **Linux** : Le système d'exploitation, ici Debian 11.
- **Apache** : Le serveur web qui gère les requêtes HTTP.
- **MySQL/MariaDB** : Le système de gestion de base de données relationnelle pour stocker les données.
- **PHP** : Le langage de programmation utilisé pour générer des pages web dynamiques.

Cette combinaison de logiciels est très populaire car elle est robuste, bien supportée et dispose de nombreuses ressources et communautés pour aider à sa gestion.

6.2 Installation d'Apache

Étape 1 : Mettre à jour le système

Avant de commencer, mettez à jour votre serveur Debian pour garantir que tous les paquets sont à jour :

```
sudo apt update
sudo apt upgrade
```

Étape 2 : Installer Apache

Apache est un serveur web populaire et fiable. Pour l'installer, utilisez la commande suivante :

```
sudo apt install apache2
```

Étape 3 : Vérifier l'installation d'Apache

Après l'installation, Apache devrait démarrer automatiquement. Vous pouvez vérifier son statut avec la commande :

```
sudo systemctl status apache2
```

Si Apache est en cours d'exécution, vous devriez voir une ligne indiquant que le service est actif.

Étape 4 : Accéder à Apache

Pour vérifier qu'Apache fonctionne, ouvrez un navigateur web et accédez à l'adresse IP de votre serveur ou à `localhost:`

```
http://votre-ip
```

Vous devriez voir la page d'accueil par défaut d'Apache qui indique que le serveur web est correctement installé et en fonctionnement.

6.3 Installation de MySQL (ou MariaDB)

MySQL est un système de gestion de base de données relationnelle. Cependant, MariaDB est un fork de MySQL et est souvent recommandé pour sa compatibilité et ses performances améliorées. Dans ce module, nous allons utiliser **MariaDB**, qui est désormais le choix par défaut sur Debian.

Étape 1 : Installer MariaDB

Installez MariaDB en utilisant la commande APT :

```
sudo apt install mariadb-server mariadb-client
```

Étape 2 : Sécuriser l'installation de MariaDB

Une fois MariaDB installé, il est important de sécuriser l'installation pour limiter les risques de sécurité. Exécutez la commande suivante pour configurer les paramètres de sécurité de MariaDB :

```
sudo mysql_secure_installation
```

Vous serez invité à définir un mot de passe root pour MariaDB et à répondre à plusieurs questions pour sécuriser l'installation (désactiver l'accès root à distance, supprimer les bases de données de test, etc.).

Étape 3 : Vérifier l'installation de MariaDB

Vous pouvez tester que MariaDB fonctionne correctement en vous connectant à la base de données avec la commande suivante :

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

Entrez le mot de passe root que vous avez défini précédemment, et vous serez connecté à la ligne de commande de MariaDB. Pour quitter MariaDB, tapez :

```
exit
```

6.4 Installation de PHP

PHP est un langage de programmation utilisé pour générer des pages web dynamiques. Nous allons installer PHP et ses modules nécessaires pour travailler avec Apache et MariaDB.

Étape 1 : Installer PHP et les modules Apache

Installez PHP ainsi que les modules requis pour intégrer PHP à Apache et MariaDB. Utilisez la commande suivante pour installer PHP et ses modules de base :

```
sudo apt install php libapache2-mod-php php-mysql
```

- `libapache2-mod-php` : Ce module permet à Apache de traiter les fichiers PHP.
- `php-mysql` : Ce module permet à PHP d'interagir avec MariaDB/MySQL.

Étape 2 : Vérifier l'installation de PHP

Pour vérifier que PHP est installé et fonctionne avec Apache, créez un fichier de test PHP dans le répertoire racine d'Apache.

```
sudo nano /var/www/html/info.php
```

Ajoutez le contenu suivant :

```
php

<?php
phpinfo();
?>
```

Cela générera une page contenant des informations détaillées sur votre installation PHP. Enregistrez et fermez le fichier.

Ensuite, ouvrez votre navigateur et accédez à la page suivante :

```
http://votre-ip/info.php
```

Si PHP est installé et configuré correctement, vous devriez voir une page contenant des informations sur la version de PHP et les modules activés.

Étape 3 : Supprimer le fichier de test PHP

Après avoir vérifié que PHP fonctionne, supprimez le fichier `info.php` pour des raisons de sécurité :

```
sudo rm /var/www/html/info.php
```

6.5 Tester le serveur LAMP

À ce stade, vous avez installé et configuré Apache, MariaDB, et PHP. Pour tester votre serveur LAMP, nous allons créer une petite application PHP qui interagit avec la base de données MariaDB.

Étape 1 : Créer une base de données et une table

Connectez-vous à MariaDB pour créer une base de données et une table simple :

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

Créez une nouvelle base de données :

```
sql
CREATE DATABASE test_db;
```

Ensuite, sélectionnez cette base de données et créez une table simple :

```
USE test_db;
CREATE TABLE utilisateurs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL
);
```

Étape 2 : Créer un script PHP pour interagir avec la base de données

Créez un fichier PHP pour tester la connexion à la base de données et afficher les données. Dans le répertoire `/var/www/html/`, créez un fichier `test_db.php` :

```
sudo nano /var/www/html/test_db.php
```

Ajoutez le code suivant pour tester la connexion à la base de données et afficher les enregistrements :

```php
<?php
$servername = "localhost";
$username = "root";
$password = "votre_mot_de_passe"; // Remplacez par le mot de passe root de votre base de données
$dbname = "test_db";

// Créer la connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("La connexion a échoué: " . $conn->connect_error);
}

// Exécuter une requête pour récupérer les utilisateurs
$sql = "SELECT id, nom, email FROM utilisateurs";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Afficher chaque ligne
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Nom: " . $row["nom"]. " - Email: " . $row["email"]. "<br>";
    }
} else {
    echo "0 résultats";
}

$conn->close();
?>
```

Étape 3 : Accéder au script PHP

Dans votre navigateur, accédez à la page PHP que vous venez de créer :

`http://votre-ip/test_db.php`

Si tout fonctionne correctement, vous devriez voir la liste des utilisateurs (s'il y en a) dans votre base de données.

6.6 Gestion du serveur LAMP

Étape 1 : Démarrer, arrêter et redémarrer Apache

Utilisez les commandes `systemctl` pour gérer le service Apache :

- Démarrer Apache :

  `sudo systemctl start apache2`

- Arrêter Apache :

  `sudo systemctl stop apache2`

- Redémarrer Apache :

```
sudo systemctl restart apache2
```

Étape 2 : Démarrer, arrêter et redémarrer MariaDB

De même, vous pouvez gérer MariaDB avec ces commandes :

- Démarrer MariaDB :

```
sudo systemctl start mariadb
```

- Arrêter MariaDB :

```
sudo systemctl stop mariadb
```

- Redémarrer MariaDB :

```
sudo systemctl restart mariadb
```

Étape 3 : Démarrer, arrêter et redémarrer PHP

En général, PHP est géré par Apache et redémarre automatiquement avec lui. Il n'y a donc pas de commande dédiée pour PHP, mais si vous modifiez la configuration de PHP, vous devrez redémarrer Apache pour appliquer les modifications.

Conclusion

Dans ce module, vous avez installé un serveur LAMP sur Debian 11, composé des composants suivants :

- **Apache** : Serveur web pour héberger des pages web.
- **MariaDB** : Système de gestion de bases de données.
- **PHP** : Langage de programmation pour créer des pages web dynamiques.

Module 7 : La sauvegarde du système

Dans ce module, vous apprendrez comment réaliser des sauvegardes régulières et fiables de votre serveur Debian 11, en utilisant différentes méthodes pour protéger vos données, vos configurations système et vos applications. La sauvegarde est une tâche essentielle pour garantir la sécurité et la récupération en cas de défaillance du système ou d'incident. Ce module couvre les meilleures pratiques pour effectuer des sauvegardes complètes du système, des données spécifiques et des bases de données.

Objectifs :

- Comprendre les différentes approches de sauvegarde.
- Utiliser les outils de sauvegarde intégrés à Debian.
- Sauvegarder les données spécifiques du système, y compris les fichiers de configuration et les bases de données.
- Automatiser les sauvegardes pour assurer une protection continue.
- Restaurer un système à partir d'une sauvegarde.

7.1 Principes fondamentaux des sauvegardes

Les sauvegardes sont essentielles pour se protéger contre la perte de données. Voici quelques concepts clés à comprendre avant de commencer les sauvegardes :

- **Sauvegarde complète** : Il s'agit d'une copie complète de tous les fichiers, répertoires, et configurations de votre système. Cette sauvegarde est plus longue à effectuer, mais elle vous permet de restaurer rapidement l'intégralité du système en cas de problème.
- **Sauvegarde incrémentielle** : Elle ne sauvegarde que les fichiers qui ont été modifiés ou ajoutés depuis la dernière sauvegarde (qu'elle soit complète ou incrémentielle). Cela permet de gagner du temps et de l'espace de stockage.
- **Sauvegarde différentielle** : Elle sauvegarde les fichiers qui ont changé depuis la dernière sauvegarde complète. Contrairement à l'incrémentielle, chaque sauvegarde différentielle sera plus volumineuse, mais plus rapide à restaurer.
- **Stockage hors site** : Pour éviter de perdre toutes vos données en cas de sinistre (incendie, vol, etc.), il est recommandé de stocker les sauvegardes dans un emplacement différent, soit sur un autre serveur, soit dans le cloud.
- **Planification des sauvegardes** : Pour garantir la sécurité de vos données, il est essentiel de planifier des sauvegardes régulières, adaptées à la fréquence de modification des fichiers.

7.2 Outils de sauvegarde sous Debian 11

Debian 11 offre plusieurs outils pour effectuer des sauvegardes. Nous allons aborder quelques-unes des méthodes les plus courantes :

1. `rsync` – Outil de sauvegarde de fichiers

`rsync` est un utilitaire très utilisé pour copier des fichiers et des répertoires d'une machine à une autre, tout en minimisant l'utilisation de la bande passante en ne copiant que les fichiers modifiés.

Installation de `rsync` :

`rsync` est généralement installé par défaut sur Debian, mais vous pouvez l'installer si nécessaire avec la commande :

```
sudo apt install rsync
```

Utilisation de `rsync` pour une sauvegarde complète

Voici un exemple simple pour sauvegarder le répertoire `/home` vers un autre répertoire ou un périphérique de stockage (par exemple, un disque dur externe ou un autre serveur).

```
sudo rsync -av --delete /home/ /mnt/sauvegarde/home/
```

Explication des options :

- **`-a`** : mode archive, qui préserve les permissions, les liens, les timestamps, etc.
- **`-v`** : mode verbeux, pour afficher les détails de la copie.
- **`--delete`** : supprime les fichiers du répertoire de destination qui ont été supprimés du répertoire source.
-

Vous pouvez également utiliser `rsync` pour sauvegarder sur un serveur distant via SSH :

```
sudo rsync -avz -e ssh /home/ user@remote-server:/backup/home/
```

- `-z` : active la compression des données pendant le transfert.

2. `tar` – Création d'archives de sauvegarde

`tar` est un autre outil très utilisé pour créer des archives compressées de répertoires ou de fichiers. Il peut être utilisé pour effectuer des sauvegardes manuelles ou automatisées.

Sauvegarde avec `tar`

Voici une commande pour créer une archive de votre répertoire `/etc` (contenant les configurations système) :

```
sudo tar czvf /mnt/sauvegarde/etc-backup.tar.gz /etc
```

Explication des options :

- `c` : Crée une nouvelle archive.
- `z` : Compresse l'archive avec `gzip`.
- `v` : Affiche les fichiers archivés pendant l'opération.
- `f` : Spécifie le nom du fichier d'archive (`/mnt/sauvegarde/etc-backup.tar.gz`).

3. `dd` – Sauvegarde de disque entier

L'outil `dd` est utilisé pour faire une copie exacte (bit à bit) d'un disque ou d'une partition. Il peut être utilisé pour créer des images complètes du système, utile pour une sauvegarde complète du disque.

Exemple d'utilisation de `dd`

Pour créer une image de disque entier (par exemple, de `/dev/sda` vers un fichier d'image) :

```
sudo dd if=/dev/sda of=/mnt/sauvegarde/sda-backup.img bs=64K conv=noerror,sync
```

Explication des options :

- `if` : spécifie le disque source.
- `of` : spécifie le fichier d'image de destination.
- `bs=64K` : définit la taille des blocs à 64 Ko.
- `conv=noerror,sync` : continue en cas d'erreurs de lecture et synchronise les blocs.

7.3 Sauvegarde des bases de données

Pour les bases de données, vous pouvez utiliser des outils spécifiques à chaque moteur de base de données.

Sauvegarde de MariaDB/MySQL avec `mysqldump`

`mysqldump` est un utilitaire utilisé pour exporter une base de données MySQL ou MariaDB en fichier texte.

Exemple de sauvegarde avec `mysqldump`

```
sudo mysqldump -u root -p --all-databases > /mnt/sauvegarde/all-databases.sql
```

Explication :

- **`-u root`** : se connecte avec l'utilisateur `root`.
- **`-p`** : vous invite à entrer le mot de passe de l'utilisateur root.
- **`--all-databases`** : sauvegarde toutes les bases de données.

Vous pouvez également sauvegarder une base de données spécifique :

```
sudo mysqldump -u root -p my_database > /mnt/sauvegarde/my_database.sql
```

Sauvegarde automatique avec `cron`

Il est possible d'automatiser vos sauvegardes en les planifiant via `cron`.

Exemple de planification avec `cron`

1. Ouvrez la crontab pour l'utilisateur root :

```
sudo crontab -e
```

2. Ajoutez une ligne pour effectuer une sauvegarde quotidienne à 2h00 du matin, par exemple :

```
0 2 * * * /usr/bin/mysqldump -u root -p'motdepasse' --all-databases >
/mnt/sauvegarde/$(date +\%F)-all-databases.sql
```

Cette ligne crée une sauvegarde de toutes les bases de données chaque jour à 2h00.

7.4 Sauvegarde vers le cloud

De nos jours, de nombreux utilisateurs choisissent de sauvegarder leurs données vers le cloud pour avoir une protection supplémentaire. Voici quelques solutions populaires pour sauvegarder vos données dans le cloud sous Debian 11 :

Utiliser `rclone` pour sauvegarder vers un service cloud

`rclone` est un outil en ligne de commande pour gérer et transférer des fichiers vers des services de stockage cloud comme Google Drive, Dropbox, et autres.

Installation de `rclone` :

```
sudo apt install rclone
```

Configuration de `rclone` :

Lancez la commande suivante pour configurer votre compte cloud :

```
rclone config
```

Suivez les instructions à l'écran pour connecter votre compte cloud.

Sauvegarde vers le cloud avec `rclone` :

Une fois configuré, vous pouvez sauvegarder un répertoire local vers votre cloud en utilisant cette commande :

```
rclone copy /home/ user@remote-cloud:/backup/
```

7.5 Restaurer un système à partir d'une sauvegarde

Restaurer avec `rsync` :

Si vous avez utilisé `rsync` pour faire une sauvegarde, vous pouvez restaurer les fichiers avec la commande suivante :

```
sudo rsync -av /mnt/sauvegarde/home/ /home/
```

Restaurer avec `tar` :

Si vous avez créé une archive avec `tar`, vous pouvez la restaurer avec cette commande :

```
sudo tar xzvf /mnt/sauvegarde/etc-backup.tar.gz -C /
```

Restaurer une base de données avec `mysqldump` :

Pour restaurer une base de données MySQL/MariaDB, utilisez la commande suivante :

```
sudo mysql -u root -p < /mnt/sauvegarde/all-databases.sql
```