Boumrah Samy

DEBIAN - Chapitre 1 : Installer SUDO ET SSH sur DEBIAN 13

Ce chapitre décrit la mise en place de sudo et du serveur OpenSSH sur une installation minimale de **Debian 13**, sans interface graphique, intégrée à un réseau PFsense.
L'objectif est de permettre l'administration distante de la machine via SSH depuis un poste Windows, via une règle de pare-feu PFsense.

Contexte de l'environnement

La machine Debian 13 est installée depuis l'ISO officiel, sans environnement de bureau.
Elle est rattachée au LAN interne géré par PFsense, via un réseau virtuel **VMnet0**.
Une adresse IP statique a été attribuée à la VM selon le schéma d'adressage du lab soit :

- IP : `192.168.10.112`

Les deux comptes créés lors de l'installation sont :

- Super-administrateur :
  - Login : root
  - Mot de passe : `19181716`
- Utilisateur standard :
  - Login : debian
  - Mot de passe : `19181716`

L'adresse IP et l'adresse MAC peuvent être vérifiées via la commande suivante :

```
bash
ip a
```

Ces informations permettent de fixer une adresse IP réservée dans le DHCP ou via une règle statique dans PFsense.

Boumrah Samy

Mise à jour du système

Avant d'installer de nouveaux paquets, il est recommandé de mettre à jour le système. Connectez-vous en tant que **root** puis exécutez :

```
bash
apt update && apt upgrade -y
```

Cette commande met à jour la liste des paquets disponibles et met à niveau les paquets installés sur Debian 13.

Installation et configuration de SUDO

Debian 13 n'installe pas `sudo` par défaut.

Pour l'installer, exécutez la commande suivante (en tant que `root`) :

```
bash
apt install sudo -y
```

Une fois le paquet installé, il faut ajouter l'utilisateur `debian` au groupe `sudo` afin qu'il puisse utiliser la commande `sudo` pour élever ses privilèges :

```
bash
usermod -aG sudo debian
```

Vous pouvez vérifier que l'utilisateur `debian` appartient bien au groupe `sudo` avec :

```
bash
groups debian
```

La sortie doit notamment contenir `sudo` dans la liste des groupes.

Installation du serveur OPEN SSH

Pour permettre l'administration distante de la machine, installez le serveur `OpenSSH` :

```
bash
apt install openssh-server -y
```

Cette commande installe le service SSH côté serveur, nécessaire pour accepter les connexions SSH entrantes.

Vérification de l’état du service SSH

Après l’installation, vérifiez que le service `ssh` est actif et démarré :

```
bash
systemctl status ssh
```

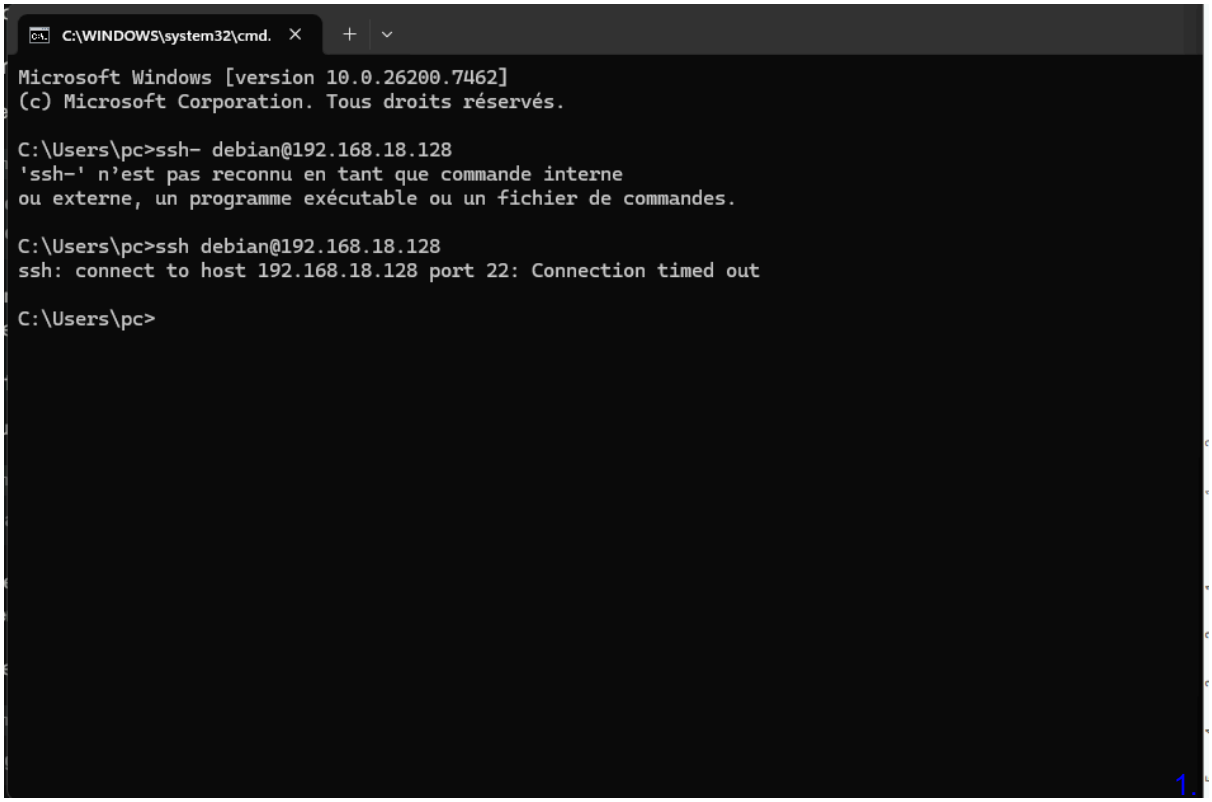
Si le service n’est pas encore activé au démarrage, vous pouvez le lancer et l’activer définitivement avec :

```
bash
systemctl enable --now ssh
```

Le service doit apparaître dans l’état `active (running)` dans la sortie de `systemctl status ssh`.

Accès SSH depuis le poste Windows

Une fois le service SSH configuré sur Debian 13, il est possible de se connecter depuis le terminal Windows (par exemple via PowerShell ou Terminal) :



1.

Conclusion

Ce chapitre a permis de préparer la machine **Debian 13** en mode serveur minimal (sans interface graphique) pour une administration distante via SSH.
Les étapes clés ont été réalisées :

- Mise à jour du système avec apt update && apt upgrade -y.
- Installation de sudo et ajout de l'utilisateur debian au groupe sudo pour une élévation de privilèges contrôlée.
- Installation et activation du service openssh-server, puis vérification de son bon fonctionnement avec systemctl status ssh.

Fin du chapitre 1

Boumrah Samy

DEBIAN - Chapitre 2 : Configuration IP fixe et Accès SSH sur DEBIAN.

Objectif

Ce chapitre vise à établir une connexion SSH fiable depuis le **poste Windows hôte** vers une **VM Debian 13** (IP `192.168.10.112`) à travers un pare-feu **pfSense**, dans un environnement **VMware Workstation**.
L'adresse IP de la VM Debian est fixée de manière stable via une **réservation DHCP** sur pfSense, tout en activant et sécurisant le service SSH côté Debian.

Architecture réseau mise en place

Topologie logique

- **Windows hôte :** `192.168.10.200/24` (adaptateur VMware VMnet0).
- **pfSense** :
  - WAN : `192.168.18.128/24`.
  - LAN : `192.168.10.1/24`.
- **Debian VM :** `192.168.10.112/24` (adresse IP fixe obtenue via réservation DHCP).

Réseaux VMware

- **VMnet0** : mode *Host-Only* sur le sous-réseau `192.168.10.0/24`.
- Les machines **Debian** et **pfSense (interface LAN)** sont connectées sur le même segment VMnet0, ce qui permet une communication directe en LAN.

Configuration étape par étape

1. Configuration de Debian

Installation et activation du serveur SSH

Connectez-vous en tant que `root` sur la Debian 13 puis mettez à jour les paquets et installez `openssh-server` :

```
bash
sudo apt-get update
sudo apt-get install openssh-server -y
sudo systemctl start ssh
sudo systemctl enable ssh
sudo systemctl status ssh
```

Le service `ssh` (OpenSSH) doit apparaître dans l'état `active (running)` et être activé au démarrage.

Ajout de l'utilisateur au groupe sudo

Si ce n'est pas déjà fait, installez `sudo` et ajoutez l'utilisateur `debian` au groupe `sudo` :

```
bash
su -
apt install sudo -y
usermod -aG sudo debian
exit
```

Vous pouvez ensuite vérifier l'appartenance avec :

```
bash
groups debian
```

Récupération de l'adresse MAC:

```
bash
ip a
# MAC address: 00:0c:29:3a:44:00
```

2. Configuration de pfSense

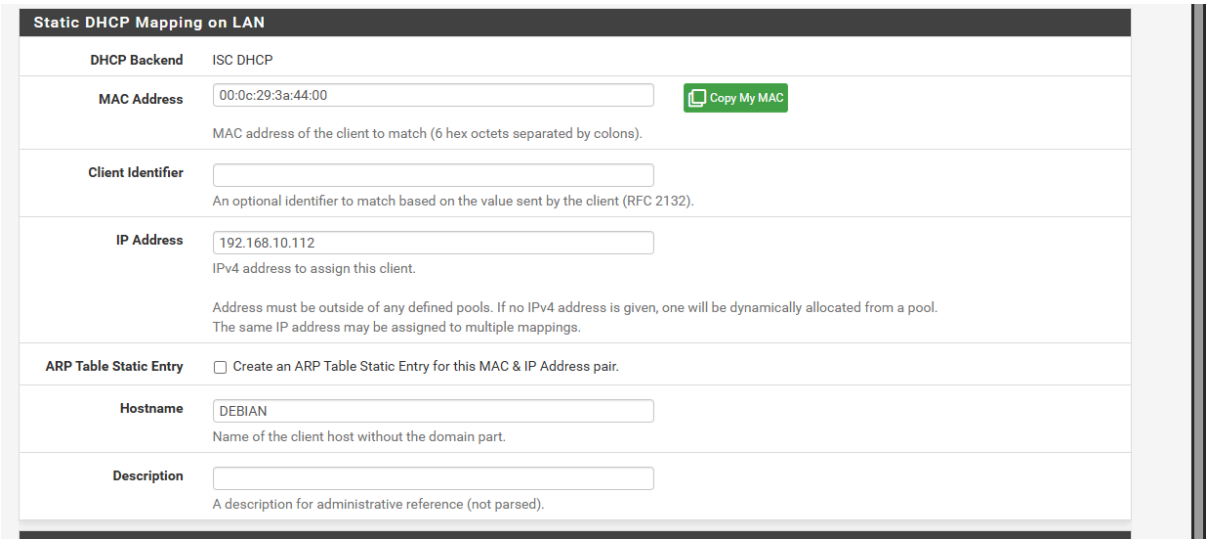
Réservation DHCP statique

Cette configuration garantit que la VM Debian reçoit toujours l'adresse `192.168.10.112`.

1. Ouvrir l'interface web de pfSense.
2. Naviguer dans : **Services > DHCP Server > LAN**.

3. Dans la section **Static mappings**, ajouter une entrée :
    - **MAC address** : `00:0c:29:3a:44:00`
    - **IP address** : `192.168.10.112`
    - **`Hostname` :** `debian`
4. Cliquer sur **Save** puis **Apply changes**.

Cette réservation DHCP affecte une adresse IP fixe à la VM Debian tout en conservant une gestion centralisée de l'adressage côté pfSense.



Règles de pare-feu (LAN → LAN)

Par défaut, pfSense autorise tout le trafic **LAN → LAN** sans règle explicite supplémentaire.

Dans ce scénario, la connexion SSH depuis le poste Windows hôte (`192.168.10.200`) vers Debian (`192.168.10.112`) reste dans le même segment et est donc permise par la règle par défaut **LAN default allow**.

Si vous souhaitez expliciter ou restreindre l'accès SSH, vous pouvez ajouter une règle spécifique sur l'interface LAN :

- **Protocol** : *TCP*
- **Port de destination** : `22`
- **Adresse de destination** : `192.168.10.112`
- **Action** : *Pass*

3. Configuration VMware Workstation

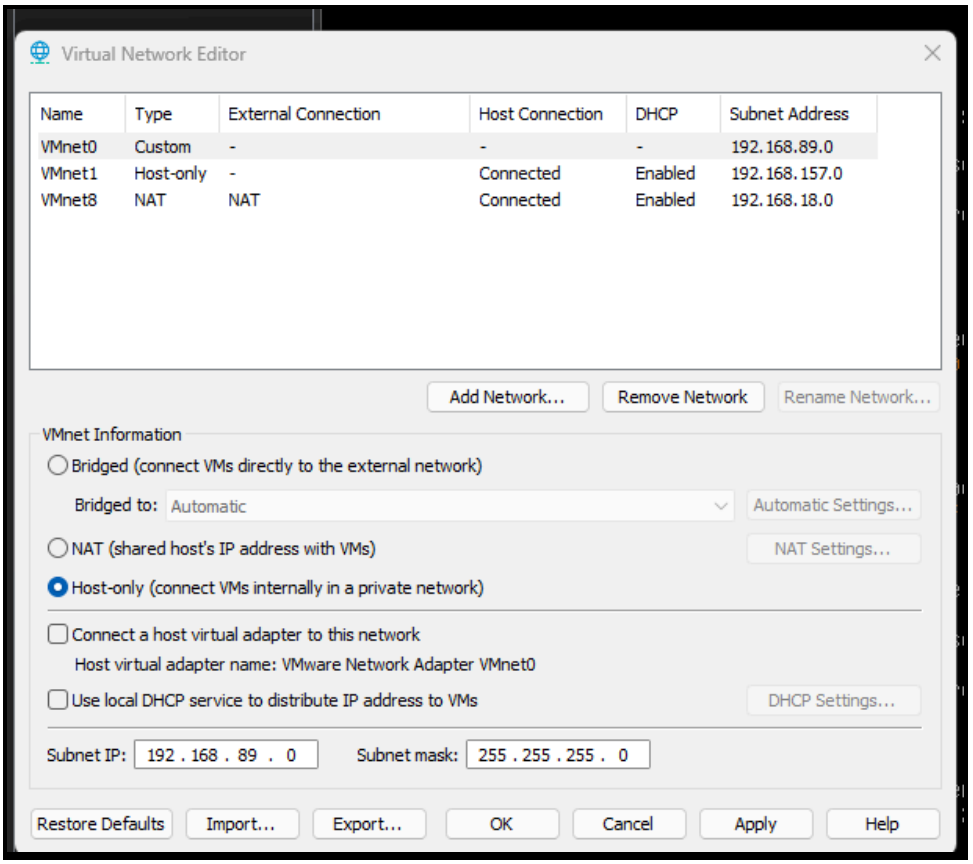
3. Configuration VMware Workstation

Paramétrage du réseau VMnet0

La configuration du réseau virtuel **VMnet0** dans **VMware Workstation** est réalisée via l'outil **Virtual Network Editor**, à exécuter en tant qu'administrateur.

1. Ouvrir **Virtual Network Editor** (élevé en droits administrateur).
2. Sélectionner le réseau VMnet0.
3. Vérifier les paramètres suivants :
    - Type : *Host-only*
    - ✓ **Connect a host virtual adapter to this network** (coché).
    - ☐ **Use local DHCP service** (décoché, le service DHCP est géré par pfSense).
    - Subnet IP : `192.168.10.0`
    - Subnet mask : `255.255.255.0`

Cette configuration crée un réseau interne isolé entre le poste Windows hôte (via l'adaptateur VMnet0) et les machines virtuelles (Debian et pfSense LAN), tout en laissant la gestion de l'adressage IP à pfSense.



4. Configuration Windows hôte

4.1 Adaptateur VMware Network Adapter VMnet0

Le poste **Windows hôte** se connecte au réseau interne via l'adaptateur **VMware Network Adapter VMnet0**.
Les paramètres réseau sont configurés en adresse IP statique sur le segment `192.168.10.0/24`.

Paramètres de l'interface **VMware Network Adapter VMnet0** :

- **Adresse IP** : `192.168.10.200`
- **Masque de sous-réseau** : `255.255.255.0`
- **Passerelle par défaut** : `192.168.10.1` (adresse LAN de pfSense).
- **Serveur DNS** : `192.168.10.1` (DNS fourni par pfSense).

Cette configuration permet au poste Windows hôte d'être sur le même segment réseau que Debian (`192.168.10.112`) et pfSense (`192.168.10.1`), tout en utilisant pfSense comme passerelle et résolveur DNS.

5.1 Installation et utilisation de dhclient

Pour forcer le renouvellement de l'adresse DHCP côté Debian (utile après une mise à jour de la configuration pfSense ou de VMnet0), installez le client DHCP et relancez le bail.

Connectez-vous en tant qu'utilisateur avec privilèges `sudo` puis exécutez les commandes suivantes :

```
bash
sudo apt-get install isc-dhcp-client
sudo dhclient -r ens33
sudo dhclient ens33
```

- `dhclient -r ens33` : libère le bail DHCP actuel sur l'interface `ens33`.
- `dhclient ens33` : demande un nouveau bail DHCP sur l'interface `ens33`.

L'interface concernée (`ens33`) peut être différente selon la VM (par exemple `ens192`, `ens32`, etc.) ; adaptez-la à votre cas.

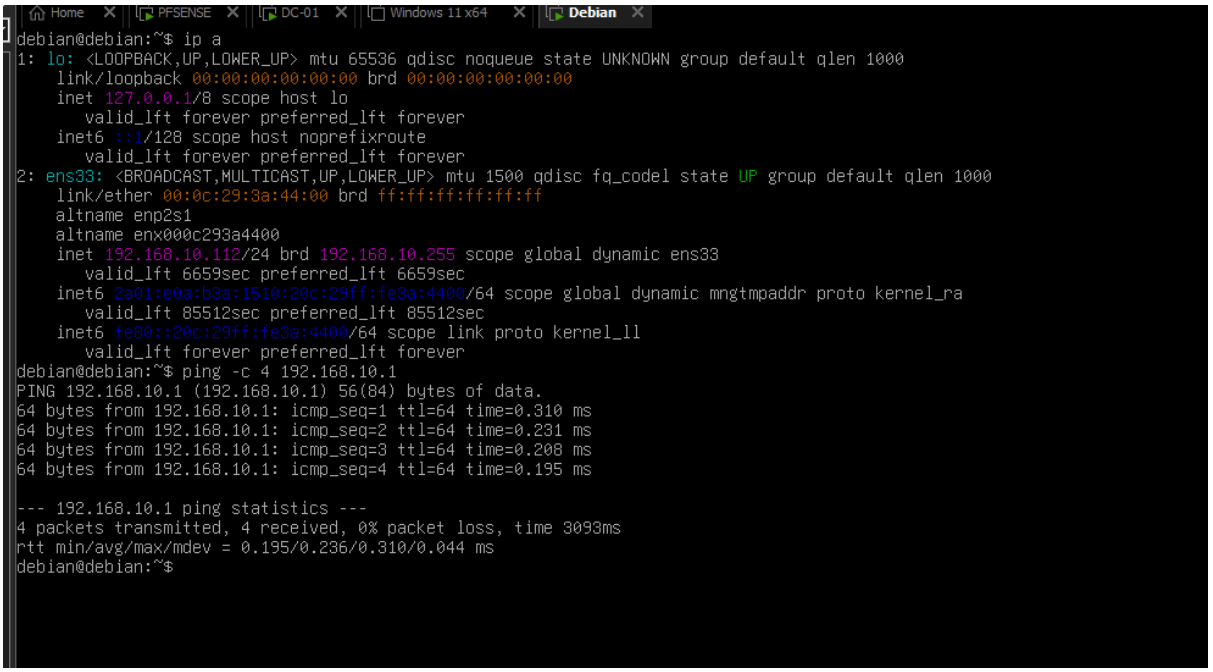
6. Tests de connectivité

6.1 Vérification de la configuration réseau sur Debian

Depuis Debian, vérifiez la configuration IP de la machine avec la commande suivante :

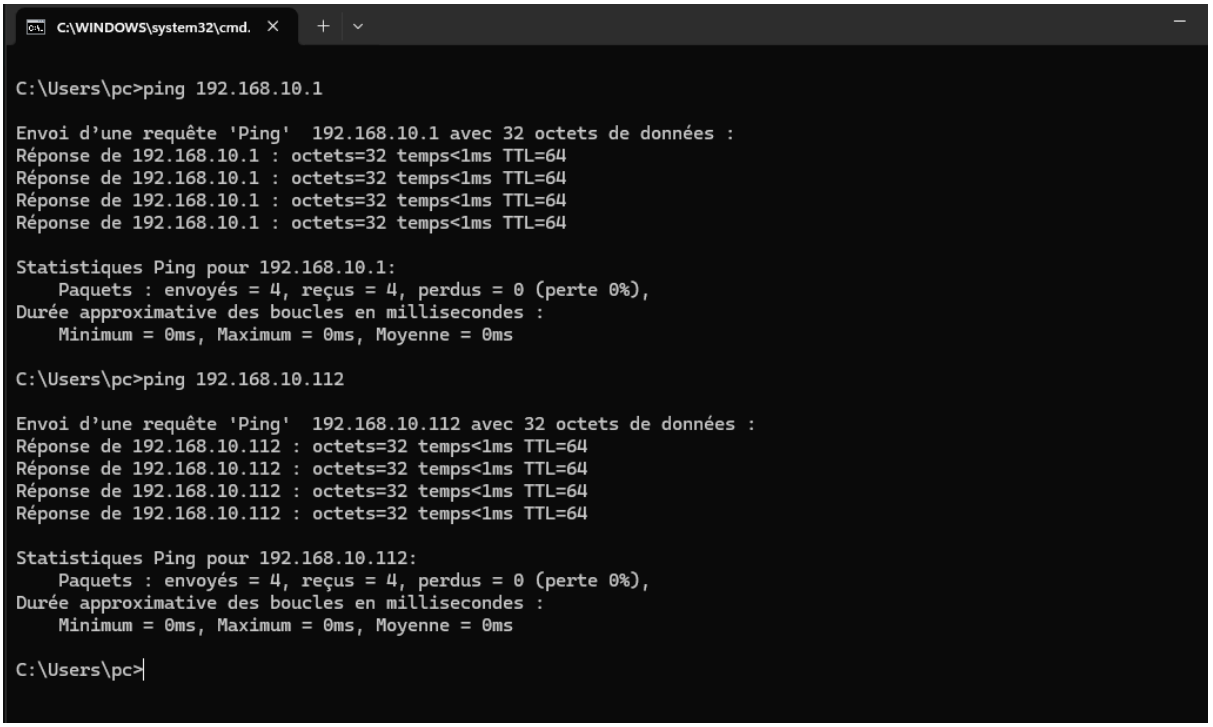
```
bash
ip a
ping -c 4 192.168.10.1
```

La sortie doit montrer que l'interface connectée au réseau `192.168.10.0/24` possède l'adresse `192.168.10.112`, conforme à la réservation DHCP définie sur pfSense.
Cette commande permet de confirmer que Debian obtient bien l'adresse IP fixe attendue dans le lab.



6.1 Vérification de la configuration réseau sur Windows

```
# Depuis Windows
ping 192.168.10.1
ping 192.168.10.112
```



7. Connexion SSH finale

7.1 Connexion depuis Windows hôte

Une fois l'ensemble de la configuration réseau (VMware, Windows hôte, pfSense, Debian) en place, ouvrez une session SSH depuis le terminal Windows vers la VM Debian.

```
# Depuis Windows
ssh debian@192.168.10.112
password : ****
```

```
C:\Users\pc>ssh debian@192.168.10.112
debian@192.168.10.112's password:
Linux debian 6.12.57+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.57-1 (2025-11-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan  2 23:32:42 2026 from 192.168.10.200
debian@debian:~$
```

La connexion est alors établie entre le poste Windows hôte et la VM Debian, via le réseau `192.168.10.0/24` et en passant par le pare-feu pfSense.

7.2 Intérêt de la configuration

Cette configuration permet un **accès SSH** depuis le poste Windows hôte vers la VM Debian en passant par le pare-feu **pfSense**, dans un environnement **isolé** et **contrôlé**.
Elle est particulièrement adaptée à un **lab d'apprentissage** en administration système et réseau, où la compréhension du flux réseau et du rôle du pare-feu est essentielle.

Conclusion du chapitre

Ce chapitre a permis de mettre en place une configuration réseau cohérente entre le poste **Windows hôte**, le pare-feu **pfSense** et la **VM Debian 13**, afin d'autoriser un accès SSH fonctionnel depuis l'hôte vers la machine Debian.

Les points clés sont :

- La définition du réseau **VMnet0** en mode *Host-Only* avec **pfSense comme serveur DHCP**.
- L'attribution d'une adresse IP fixe à la VM Debian via une **réservation DHCP** sur pfSense.
- La configuration de l'adaptateur **VMware Network Adapter VMnet0** sur le Windows hôte avec une IP statique sur le même segment.
- Le renouvellement du bail DHCP sur Debian et la vérification de la configuration IP avec ip a.
- L'établissement d'une **connexion SSH** finale (ssh debian@192.168.10.112) depuis Windows vers Debian, via pfSense, dans un environnement isolé adapté à un lab de formation.

Cette base réseau et cette configuration SSH constituent un socle stable pour la suite de la documentation (services de l'infrastructure, scripts d'administration, sécurité avancée, etc.).

Fin du chapitre 2

Boumrah Samy

DEBIAN - Chapitre 3 : Installer un environnement LAMP sur DEBIAN 13

Contexte du chapitre

Dans ce lab, l'objectif est de préparer Debian 13 pour l'installation ultérieure de GLPI (helpdesk / gestion d'actifs), qui nécessite un environnement LAMP :

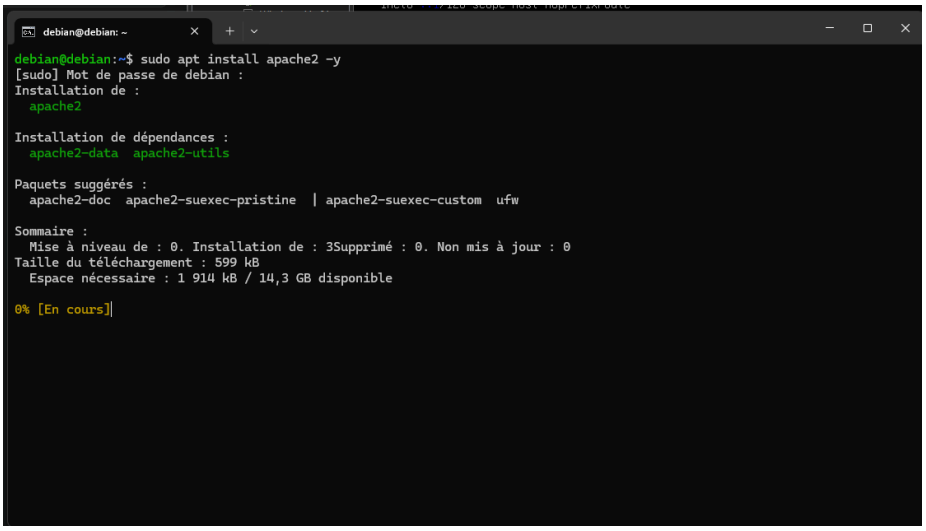
- **L : Linux (**Debian 13**).**
- **A : Apache2 (**serveur web**).**
- **M : MariaDB (**base de données**).**
- **P : PHP (**langage de script serveur**).**

Ce chapitre décrit l'installation et la configuration minimale du serveur web Apache2 sur Debian 13, dans le cadre de la mise en place de l'environnement LAMP.

Installation du serveur web Apache2

Depuis la VM Debian 13, connectez-vous avec un utilisateur disposant des privilèges `sudo` (par exemple debian), puis exécutez la commande suivante pour installer Apache2 :

```
bash
sudo apt install apache2 -y
```



Cette commande installe le paquet `apache2` et dépendances associées.
Apache2 est automatiquement lancé et activé au démarrage du système par défaut sur Debian 13.

Vous pouvez vérifier l'état du service avec :

```bash
systemctl status apache2
```

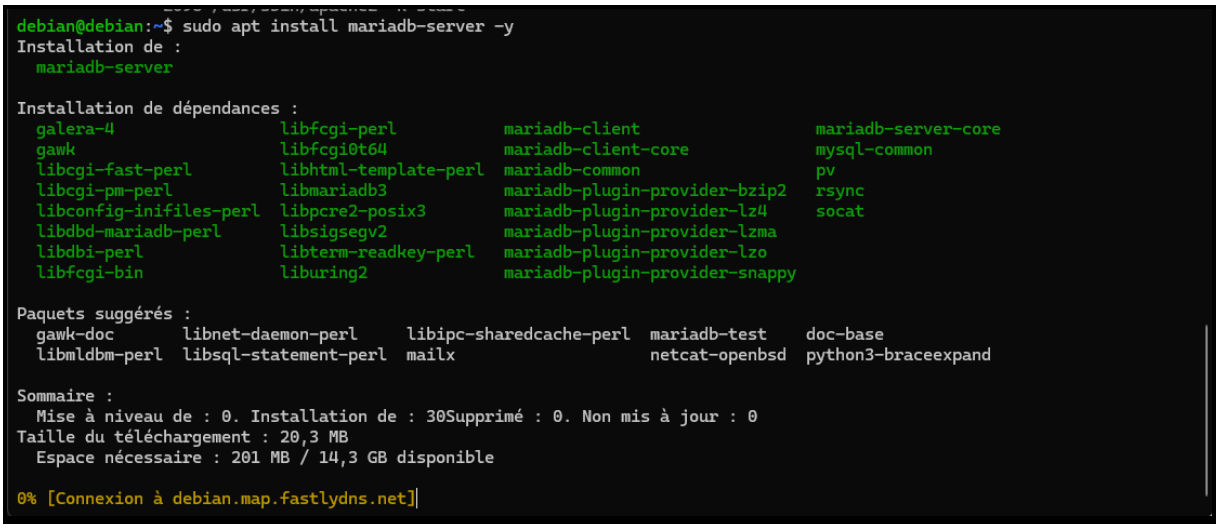
Le service doit apparaître dans l'état `active (running)`.

Installation du serveur MariaDB

Une fois Apache2 installé, il faut mettre en place le serveur de base de données **MariaDB**, qui jouera le rôle du composant **M** dans l'environnement LAMP.

Depuis la VM Debian 13, exécutez la commande suivante en tant qu'utilisateur disposant des privilèges `sudo` :

```bash
sudo apt install mariadb-server -y
```



Cette commande installe le paquet `mariadb-server` ainsi que ses dépendances.
MariaDB est généralement démarré et activé au démarrage du système après l'installation.

Vous pouvez vérifier l'état du service avec :

```
bash
systemctl status mariadb
```

```
debian@debian:~$ systemctl status mariadb
● mariadb.service - MariaDB 11.8.3 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
     Active: active (running) since Sat 2026-01-03 00:01:12 CET; 6s ago
 Invocation: 3ef8a08c14814683b3a65b998e021f2b
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 3155 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 13 (limit: 14861)
     Memory: 125.4M (peak: 129.9M)
        CPU: 902ms
     CGroup: /system.slice/mariadb.service
             └─3155 /usr/sbin/mariadbd
debian@debian:~$
```

Installation du module PHP

Une fois Apache2 et MariaDB installés, il faut ajouter **PHP** pour compléter l'environnement LAMP.

Dépuis la VM Debian 13, exécutez la commande suivante en tant qu'utilisateur disposant des privilèges `sudo` **:**

```
bash
sudo apt install php -y
```

```
debian@debian:~$ sudo apt install php -y
Installation de :
  php

Installation de dépendances :
  libapache2-mod-php8.4  php-common  php8.4  php8.4-cli  php8.4-common  php8.4-opcache  php8.4-readline

Paquets suggérés :
  php-pear

Sommaire :
  Mise à niveau de : 0. Installation de : 8Supprimé : 0. Non mis à jour : 0
Taille du téléchargement : 4 944 kB
  Espace nécessaire : 23,8 MB / 13,9 GB disponible

Réception de : 1 http://deb.debian.org/debian trixie/main amd64 php-common all 2:96 [13,3 kB]
Réception de : 2 http://deb.debian.org/debian trixie/main amd64 php all 2:8.4+96 [3 936 B]
5% [Connexion à debian.map.fastlydns.net (2a04:4e42:6a::644)]
```

Cette commande installe le paquet PHP de base sur Debian 13, ce qui permet d'exécuter des scripts PHP côté serveur web.

Vous pouvez vérifier la version de PHP installée avec :

```
bash
php -v
```

La sortie de cette commande doit afficher la version de PHP active sur le système (par exemple `PHP 8.x.x`), confirmant que l'environnement est prêt pour l'exécution de scripts PHP via Apache2.

Sécurisation de MariaDB avec mariadb-secure-installation

Description

La commande `mariadb-secure-installation` est un script de sécurisation automatique fourni par MariaDB.
Elle permet de configurer rapidement les paramètres de sécurité essentiels d'un serveur MariaDB après son installation.

Objectif

L'objectif de ce script est de transformer la configuration par défaut de MariaDB (volontairement peu sécurisée pour faciliter les tests) en une configuration plus robuste, adaptée à un environnement de lab ou de production.
Parmi les actions courantes réalisées par le script figurent :

- La suppression des comptes anonymes.
- L'interdiction de la connexion distante au compte `root` (ou leur restriction).
- La suppression des bases de données de test.
- La mise en place d'un mot de passe pour le compte `root` MariaDB.

Commande à exécuter

Depuis la VM Debian 13, exécutez la commande suivante avec les privilèges `sudo` :

```
bash
sudo mariadb-secure-installation
```

```
debian@debian:~$ sudo mariadb-secure-installation

NOTE: MariaDB is secure by default in Debian. Running this script is
      useless at best, and misleading at worst. This script will be
      removed in a future MariaDB release in Debian. Please read
      mariadb-server.README.Debian for details.

Enter root user password or leave blank:

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!
```

Note : Sur Debian 13, utilisez la commande `mariadb-secure-installation` et non `mysql_secure_installation`, conforme au paquet MariaDB fourni par la distribution.

Le script vous guidera pas à pas. Répondez aux questions selon les règles de sécurité souhaitées pour votre lab (par exemple, suppression des comptes anonymes, suppression de la base de données de test, configuration stricte pour les connexions distantes).

Mesures de sécurité appliquées

Action	Recommandation	Objectif
Définir mot de passe root	Y	Protéger le compte administrateur
Supprimer utilisateurs anonymes	Y	Empêcher les connexions non authentifiées
Désactiver connexion root distante	Y	Limiter l'accès root au localhost uniquement
Supprimer base de données test	Y	Éliminer une surface d'attaque inutile
Recharger les privilèges	Y	Appliquer immédiatement les modifications

Risques liés à un serveur MariaDB non sécurisé

En l'absence d'exécution de `mariadb-secure-installation`, le serveur MariaDB reste configuré avec des paramètres par défaut peu restrictifs, ce qui expose la base de données à plusieurs types de risques :

- Accès root sans authentification : présence éventuelle de comptes root sans mot de passe, permettant une élévation de privilèges immédiate.
- Connexions anonymes autorisées : certains comptes peuvent se connecter sans identifiant, augmentant la surface d'attaque.
- Exploitation possible de la base de test : la base de données test par défaut peut être utilisée pour des tests d'intrusion ou comme point d'entrée.
- Attaques par force brute depuis l'extérieur : si le serveur MariaDB est accessible depuis le réseau, l'absence de restrictions sur les comptes ou les mots de passe facilite ce type d'attaque.

Ces vulnérabilités sont particulièrement critiques dans un environnement réseau ouvert, même dans un lab de formation.

Vérification post-sécurisation

Après l'exécution de `mariadb-secure-installation`, il est recommandé de vérifier que la configuration de sécurité est bien appliquée.

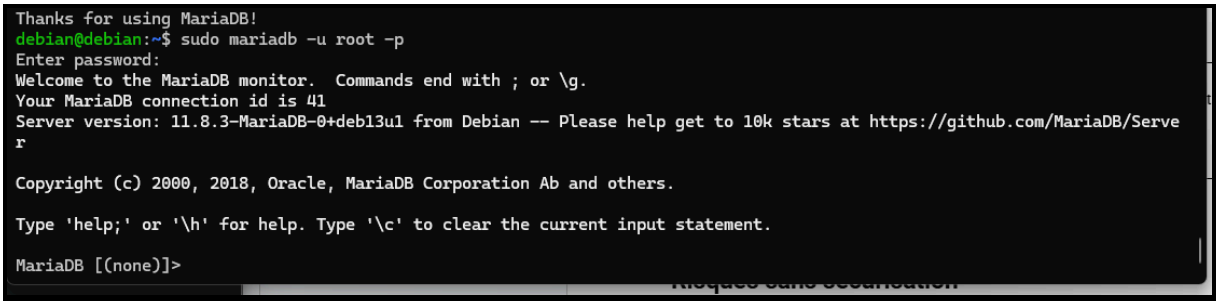
Connectez-vous au serveur MariaDB avec le compte `root` :

```
bash
sudo mariadb -u root -p
```

Saisissez ensuite le **mot de passe root** défini lors de la sécurisation.
Si la connexion est acceptée, cela confirme que :

- Le script `mariadb-secure-installation` s'est correctement exécuté.
- Le compte root est désormais protégé par un mot de passe.
- Le service MariaDB est opérationnel et accessible.

Cette étape permet de valider qu'un niveau de sécurité de base est mis en place avant toute utilisation de MariaDB dans l'environnement LAMP (par exemple, pour GLPI ou d'autres applications web).



Conclusion du chapitre

Ce chapitre a permis de mettre en place un environnement **LAMP complet sur Debian 13** (Linux, Apache2, MariaDB, PHP), nécessaire au fonctionnement de GLPI et d'autres applications web dans le lab.
Les étapes clés ont été réalisées :

- Installation du serveur web **Apache2**.
- Installation du serveur de base de données **MariaDB**.
- Installation du langage **PHP** et vérification de sa version via `php -v`.
- Sécurisation du serveur MariaDB avec le script `mariadb-secure-installation` et vérification de l'accès au compte `root` après la sécurisation.

L'environnement LAMP est désormais opérationnel, avec une base de données sécurisée et un serveur web prêt à servir des applications PHP, ce qui prépare la suite de la documentation (installation de GLPI, configuration de vhosts, etc.).

Fin du chapitre 3

Boumrah Samy

DEBIAN - Chapitre 4 : Configuration Port Forward HTTP dans pfSense

Objectif

Ce chapitre vise à permettre l'accès au **serveur web Apache** hébergé sur la VM **Debian 13** (192.168.10.112:80) depuis l'**interface WAN** de **pfSense**.
L'accès se fait via une règle de **port forwarding** (NAT) sur le port **HTTP 80**, permettant d'atteindre la page par défaut d'Apache depuis un client situé sur le réseau externe.

Configuration NAT (Port Forward)

1. Création de la règle de port forwarding

Connectez-vous à l'interface web de **pfSense**, puis accédez à la configuration NAT :

1. Navigation : **Firewall > NAT > Port Forward**.
2. Cliquez sur **Add** pour créer une nouvelle règle.

Paramètres de la règle :

- **Interface** : `WAN`.
- **Protocol** : `TCP`.
- **Source** : `any` (ou restreindre si nécessaire).
- **Destination** : `WAN address`.
- **Destination port range** : `HTTP (80)`.
- **Redirect target IP** : `192.168.10.112`.
- **Redirect target port** : `HTTP (80)`.
- **Description** : `HTTP vers Debian Apache`.
- **Filter rule association** : ✅ `Add associated filter rule`.

Sauvegardez puis cliquez sur **Apply Changes**.

Cette règle NAT permet de rediriger tout trafic entrant en TCP sur le port 80 de l'interface WAN vers le serveur Apache sur la VM Debian (`192.168.10.112:80`).

Vérification

1. Règle firewall WAN

Après la création de la règle de port forwarding, pfSense génère automatiquement une règle de filtrage associée sur l'interface **WAN** :

- **Firewall > Rules > WAN**
  - Une règle autorisant le trafic TCP sur le port 80 doit être présente.

Cette règle permet le passage du trafic HTTP depuis le réseau externe vers le serveur Debian.

2. Test d'accès depuis un navigateur externe

Depuis un poste situé sur le réseau externe (ou depuis le réseau WAN simulé), ouvrez un navigateur et saisissez :

```
http://IP_WAN_pfSense
```

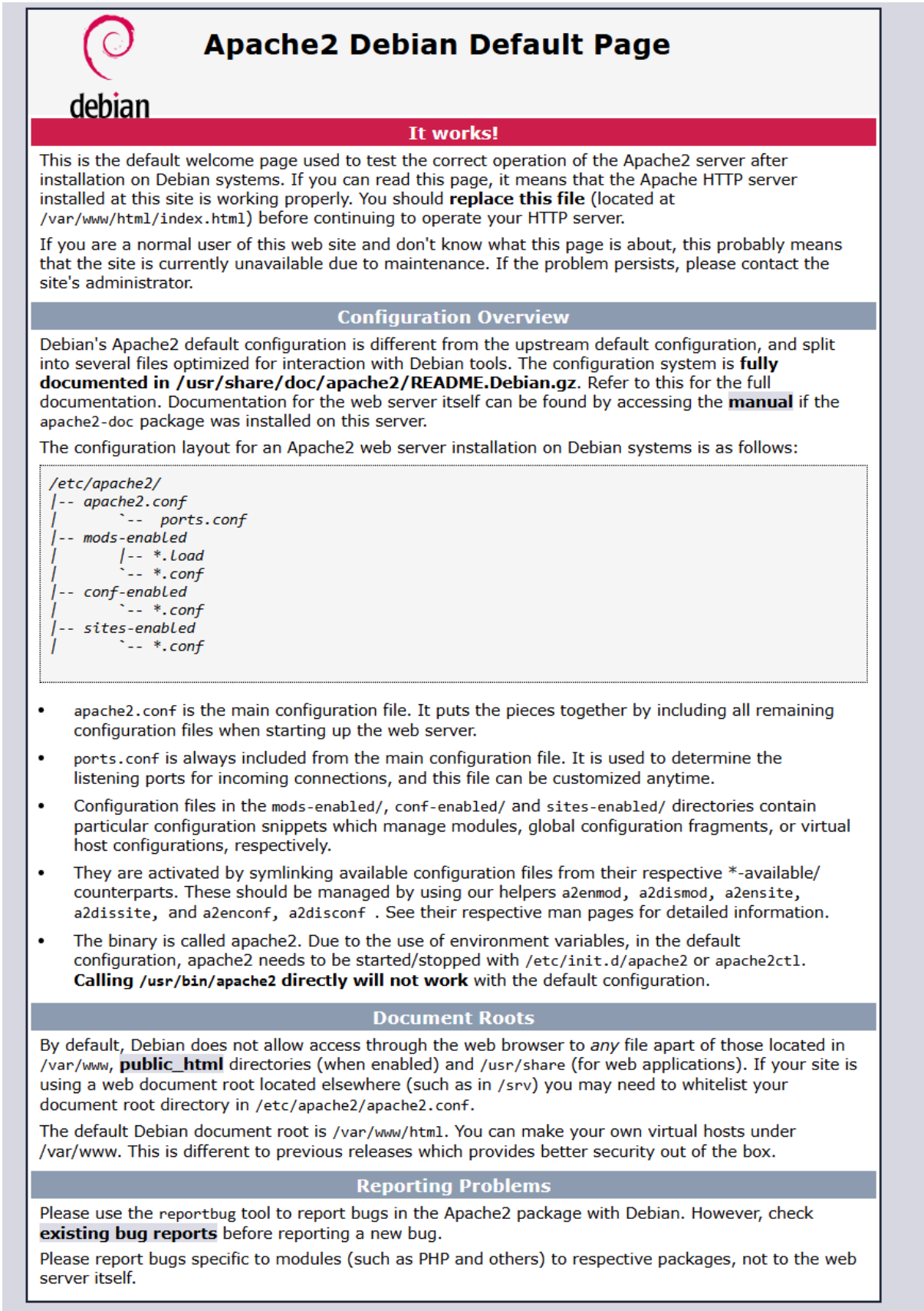
- Remplacez `IP_WAN_pfSense` par l'adresse IP publique (ou IP WAN) de pfSense.
- Si la règle de port forwarding et la règle firewall WAN sont correctement configurées, la **page par défaut d'Apache** hébergée sur Debian devrait s'afficher dans le navigateur.

Notes techniques

- Le serveur **Apache écoute sur le port 80** sur Debian. Vous pouvez vérifier cela avec la commande suivante sur la VM Debian :

```
bash
sudo ss -tlnp | grep :80
```

- La sortie doit montrer que apache2 écoute sur *:80.
- Le port **80** est le port HTTP standard. Il n'est donc pas nécessaire de le spécifier dans l'URL :
  - `http://IP_WAN_pfSense` est équivalent à `http://IP_WAN_pfSense:80`.
- **Point de vigilance** :
  Si l'**interface web de pfSense** utilise le port 80, il sera nécessaire de modifier ce port pour éviter le conflit.
  Cette modification se fait via :
  - **System > Advanced > Admin Access**
  - Changer le port d'administration (par exemple en 8443) si le port 80 est déjà utilisé par Apache.



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Debian does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Debian document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `reportbug` tool to report bugs in the Apache2 package with Debian. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Conclusion du chapitre

Ce chapitre décrit la configuration d'un **port forwarding HTTP** sur **pfSense** afin d'exposer le serveur web **Apache** hébergé sur la VM Debian 13 (192.168.10.112:80) à travers l'interface WAN.
La règle NAT de type **Port Forward** redirige le trafic TCP sur le port 80 vers la machine Debian, tandis qu'une règle de filtrage associée sur l'interface WAN autorise ce trafic entrant.

La vérification via un navigateur externe (http://IP_WAN_pfSense) permet de confirmer que la page par défaut d'Apache est bien accessible depuis le réseau externe, ce qui valide la configuration du tunnel HTTP à travers pfSense.
Cette base est essentielle pour la suite de la documentation, notamment pour l'exposition de services web comme GLPI ou d'autres applications LAMP dans le lab.

Fin du chapitre 4.