Job 01 - Installation d'une VM Debian avec interface graphique et configuration de SSH

Installation et configuration de SSH :

- Ouvrez un terminal et mettez à jour la liste des paquets :

```
sudo apt-get update
```

- Installez le paquet OpenSSH-server :

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

- Après l'installation, activez et démarrez le service SSH :

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

- Vérifiez l'état du service SSH :

```
sudo systemctl status ssh
```

- Il est conseillé de changer le numéro de port par défaut (22) pour des raisons de sécurité. Pour cela, éditez le fichier de configuration de SSH :

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

- Cherchez la ligne "#Port 22" et changez-la en "Port <numero_port>", où "<numero_port>" représente un nombre entier entre 1025 et 65535.
- Enregistrez et fermez le fichier.
- Redémarrez le service SSH pour appliquer les modifications :

```
sudo systemctl restart ssh
```

- Notez le nouveau numéro de port, car vous en aurez besoin pour vous connecter ultérieurement.

Vous avez correctement installé Debian avec une interface graphique et configuré SSH sur votre VM. Vous pouvez désormais vous connecter à distance à votre VM à l'aide d'un client SSH tel que Putty.

Job02: Installer un serveur Web Apache2

1. Premièrement, mise à jour des sources de paquets et installation d'Apache2 :

```
sudo apt update
sudo apt install apache2
```

2. Après l'installation, activez et démarrez le service Apache2 :

```
sudo systemctl enable apache2
sudo systemctl start apache2
```

3. Vérification de l'activation du service Apache2 :

```
sudo systemctl status apache2
```

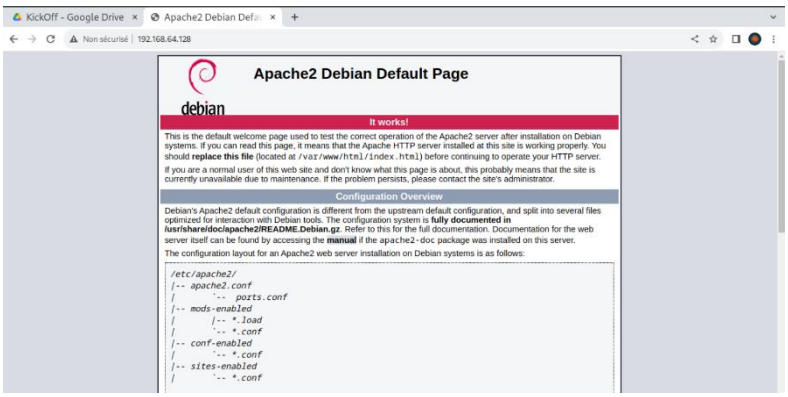
```
dome@debian12:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2023-10-27 09:50:50 CEST; 23s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2112 (apache2)
      Tasks: 55 (limit: 4581)
     Memory: 9.6M
        CPU: 52ms
     CGroup: /system.slice/apache2.service
             ├─2112 /usr/sbin/apache2 -k start
             ├─2114 /usr/sbin/apache2 -k start
             └─2115 /usr/sbin/apache2 -k start

oct. 27 09:50:50 debian12 systemd[1]: Starting apache2.service - The Apache HTTP Server...
oct. 27 09:50:50 debian12 apachectl[2111]: AH00558: apache2: Could not reliably determine the serve>
oct. 27 09:50:50 debian12 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

Le statut retourné doit mentionner que le service est actif (active) et qu'il attend des requêtes (listening).
4. Maintenant, testez l'accessibilité du serveur web Apache2 via un navigateur internet. Sur la machine hôte (PC physique), copiez et collez l'URL suivante dans un navigateur web :

http://<ip_du_serveur>

Par défaut, Apache2 renvoie une page de test intitulée "It Works!" accompagnée d'une illustration représentant un ours polaire.



Note : Remplacez "<ip_du_serveur>" par l'adresse IP privée de la VM Debian, attribuée dynamiquement par le routeur Vmware (si le DHCP est activé sur celui-ci) ou fixée manuellement (si le DHCP est désactivé).

Job03 différents serveurs Web existants

Il existe plusieurs serveurs web populaires, chacun ayant ses propres avantages et inconvénients en fonction des besoins spécifiques de l'utilisateur. Voici quelques-uns des serveurs web les plus couramment utilisés, avec un bref aperçu de leurs caractéristiques, avantages et inconvénients :

1. Apache HTTP Server (Apache) :

- ***Avantages :***
  - *Grande part de marché, largement utilisé et bien documenté.*
  - *Prise en charge d'une large gamme de modules et d'extensions.*
  - *Configuration flexible via des fichiers de configuration.*
  - *Supporte les langages de script tels que PHP, Python, et Perl.*
- ***Inconvénients :***
  - *La configuration peut être complexe pour les utilisateurs débutants.*

- *Peut être gourmand en ressources dans certains cas.*

2. Nginx :

- ***Avantages :***
  - *Conçu pour la performance, il est léger et rapide.*
  - *Excellente gestion de la charge (load balancing) et de la mise en cache.*
  - *Peut gérer un grand nombre de connexions simultanées.*
  - *Faible consommation de mémoire.*
- ***Inconvénients :***
  - *La configuration basée sur des fichiers de configuration peut être intimidante pour les débutants.*
  - *Moins flexible que Apache en termes de modules.*

3. Microsoft Internet Information Services (IIS) :

- ***Avantages :***
  - *Intégré aux systèmes d'exploitation Windows Server.*
  - *Interface utilisateur graphique conviviale pour la configuration.*
  - *Bonne intégration avec d'autres produits Microsoft.*
- ***Inconvénients :***
  - *Moins couramment utilisé pour les hébergements web non-Windows.*
  - *Certains modules peuvent nécessiter des licences supplémentaires.*

4. LiteSpeed Web Server :

- ***Avantages :***
  - *Haute performance et faible utilisation des ressources.*
  - *Excellente gestion des pics de trafic.*
  - *Prise en charge de l'OpenLiteSpeed, une version open source.*
- ***Inconvénients :***
  - *La version commerciale peut nécessiter une licence payante pour certaines fonctionnalités avancées.*

5. Caddy :

- ***Avantages :***
  - *Configuration automatique des certificats SSL (Let's Encrypt) par défaut.*
  - *Interface utilisateur web pour simplifier la configuration.*
  - *Prise en charge du protocole HTTP/2 par défaut.*
- ***Inconvénients :***
  - *Certaines fonctionnalités avancées peuvent nécessiter une mise à niveau vers la version commerciale.*

Job 04 : Mettre en place un DNS sur votre serveur Linux qui fasse correspondre l'adresse IP de votre serveur au nom de domaine local suivant : "dnsproject.prepa.com"

1. *Installer BIND9, le serveur DNS de référence sur Linux Debian :*

   ```
   sudo apt-get update
   sudo apt-get install bind9 bind9utils bind9-doc
   ```

2. *Créer un fichier de configuration pour votre domaine :*

   ```
   sudo touch /etc/bind/db.dnsproject.prepa.com
   sudo nano /etc/bind/db.dnsproject.prepa.com
   ```

3. *Ajouter les informations relatives à votre domaine dans le fichier db.dnsproject.prepa.com. Remplacez les valeurs par celles adaptées à votre situation :*

   ```
   $TTL    604800
   @       IN      SOA     ns.dnsproject.prepa.com. admin.dnsproject.prepa.com. (
                 1         ; Serial
            604800         ; Refresh
             86400         ; Retry
           2419200         ; Expire
            604800 )       ; Negative Cache TTL;
   @       IN      NS      ns.dnsproject.prepa.com.
   @       IN      A       192.168.xxx.yyy  ; Adresse IP de votre serveur
   ns      IN      A       192.168.xxx.yyy  ; Adresse IP de votre serveur
   ```

4. *Créer un fichier de zone pour votre domaine :*

   ```
   sudo nano /etc/bind/named.conf.local
   ```

   Ajouter la déclaration de zone :

   ```
   //
   // Do any local configuration here
   //

   zone "dnsproject.prepa.com" {
      type master;
      file "/etc/bind/db.dnsproject.prepa.com";
   };
   ```

5. *Valider la syntaxe de votre fichier de configuration :*

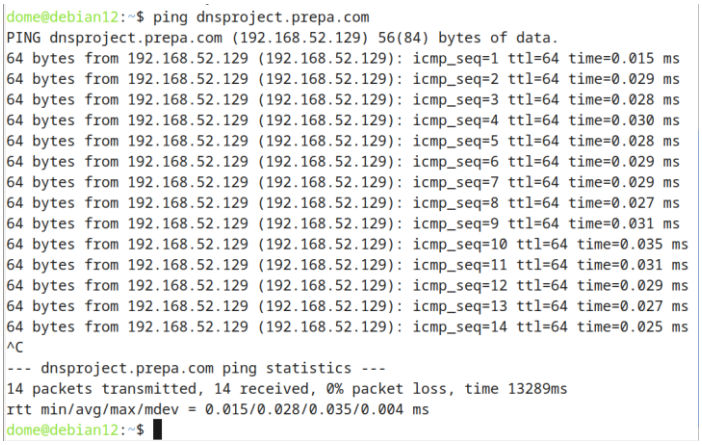
   ```
   sudo named-checkconf
   ```

6. *Redémarrer le service BIND9 pour prendre en compte les nouveaux paramètres :*

   ```
   sudo systemctl restart bind9
   ```

7. *Vérifier le fonctionnement du service DNS :*

   ```
   ping dnsproject.prepa.com
   ```



La commande précédente doit renvoyer une réponse contenant l'adresse IP que vous avez spécifiée dans le fichier de configuration db.dnsproject.prepa.com.

8. *Vérifier la résolution inverse (mapping IP -> Nom de domaine) :*

   ping -x 192.168.xxx.yyy

La commande doit renvoyer une réponse contenant le nom de domaine que vous avez spécifié dans le fichier de configuration db.dnsproject.prepa.com.

Vous avez maintenant mis en place un serveur DNS sur votre serveur Linux qui fait correspondre l'adresse IP de votre serveur au nom de domaine local "dnsproject.prepa.com". Vous pouvez également vérifier la résolution inverse en utilisant la commande "dig -x IP_ADDRESS".

Job 05 Comment obtient-on un nom de domaine public et quelles sont les spécificités que l'on peut avoir sur certaines extensions

L'obtention d'un nom de domaine public implique plusieurs étapes, et les spécificités peuvent varier en fonction du type d'extension de nom de domaine (TLD). Voici les étapes générales et quelques spécificités courantes :

Étapes pour obtenir un nom de domaine public :

1. ***Choix du Nom de Domaine :** Choisissez un nom de domaine qui reflète votre identité, votre entreprise ou vos objectifs en ligne.*
2. ***Vérification de la Disponibilité :** Utilisez un service d'enregistrement de domaines pour vérifier si le nom de domaine que vous avez choisi est disponible.*
3. ***Sélection de l'Extension :** Choisissez le type d'extension de domaine de premier niveau (TLD) qui convient à vos besoins, par exemple, .com, .org, .net, .tech, etc.*
4. ***Enregistrement :** Enregistrez le nom de domaine via un registraire de domaines. Vous devrez fournir des informations telles que votre nom, votre adresse, et des détails de contact.*
5. ***Configuration DNS :** Configurez les enregistrements DNS pour faire correspondre votre nom de domaine à l'adresse IP de votre serveur web ou autre service en ligne.*
6. ***Renouvellement :** Les noms de domaine sont généralement enregistrés pour une période d'un an et doivent être renouvelés périodiquement.*

Spécificités des Extensions de Domaine (TLD) :

1. ***Extensions Génériques (gTLD) :***
   - ***.com :** Généralement utilisé pour des sites web commerciaux.*
   - ***.org :** Souvent utilisé par des organisations à but non lucratif.*

- ***.net :*** *À l'origine destiné aux fournisseurs de services réseau, mais ouvert à tous.*

2. ***Extensions Géographiques (ccTLD) :***
   - ***.us (États-Unis), .uk (Royaume-Uni), .fr (France), etc. :*** *Associés à des pays spécifiques.*
3. ***Extensions de Premier Niveau Sponsorisées (sTLD) :***
   - ***.gov, .edu, .mil :*** *Restreintes à des entités gouvernementales, éducatives et militaires aux États-Unis.*
4. ***Extensions Nouvelles ou Spécifiques (nTLD) :***
   - ***.blog, .app, .guru, .tech :*** *Introduites récemment, souvent liées à des secteurs spécifiques.*
5. ***Extensions de Marque (Brand TLD) :***
   - *Exemples : .apple, .google, .microsoft.*
   - *Souvent utilisées par des grandes marques pour renforcer leur présence en ligne.*
6. ***Extensions Réservées (Reserved TLD) :***
   - *Certaines extensions sont réservées et ne peuvent pas être enregistrées par le public.*

Chaque registraire peut avoir ses propres règles et tarifs, il est donc important de choisir un registraire de confiance et de lire attentivement leurs conditions d'utilisation.

Job 06 : Connecter votre hôte Windows 11 au nom de domaine local de votre serveur

Pour accomplir cette tâche, vous allez modifier le fichier hosts de votre ordinateur hôte Windows 11. Le fichier hosts permet de mapping manuel entre des noms d'hôtes et des adresses IP.

1. *Ouvrez l'Invite de Commande (cmd) ou PowerShell en tant qu'administrateur.*
2. *Naviguez vers le répertoire system32 en utilisant la commande cd.*

   ```
   cd C:\Windows\System32\drivers\etc
   ```

3. *Éditez le fichier hosts en utilisant Notepad++ ou tout autre éditeur de texte de votre choix en administrateur. Par exemple, vous pouvez utiliser Notepad++ :*

   ```
   notepad++ hosts
   ```

4. *Ajouter une ligne en bas du fichier dans le format suivant :*

   ```
   <192.168.1.xx>     <dnsproject.prepa.com>
   ```

Remplacez "192.168.1.xx" par l'adresse IP de votre machine virtuelle. Enregistrer et fermer le fichier hosts.

Maintenant, quand vous pingerez "dnsproject.prepa.com" sur votre hôte Windows 11, il devrait pointer vers l'adresse IP de votre machine virtuelle.

Vous pouvez également essayer d'accéder à votre serveur Apache en utilisant le nom de domaine local dans un navigateur web sur votre hôte Windows 11 :

```
http://dnsproject.prepa.com
```

Et voilà, votre hôte Windows 11 est connecté à votre nom de domaine local.

Remarque pour vous guider sur la configuration de la redirection de port sur le routeur VMware pour que votre hôte Windows 11 puisse accéder à la page Apache sur votre serveur Debian.

1. *Ouvrez l'application VMware Workstation sur votre hôte Windows 11.*
2. *Sélectionnez la machine virtuelle Debian dans la liste et cliquez sur "Modifier les paramètres virtuels".*
3. *Dans la fenêtre des paramètres, naviguez vers l'onglet "Réseau".*
4. *Sous l'onglet "Adapter 1", assurez-vous que le mode de connexion est défini sur "NAT".*
5. *Cochez la case "Rediriger le trafic TCP" et "Rediriger le trafic UDP" dans la section "Paramètres avancés".*
6. *Dans les champs "Host Port(s)" et "Guest Port(s)", entrez respectivement le numéro de port de votre choix (par exemple, 8080) et le numéro de port d'Apache sur votre machine virtuelle Debian (généralement 80).*
7. *Cliquez sur OK pour enregistrer les modifications.*

Maintenant, vous devriez pouvoir accéder à la page Apache sur votre serveur Debian en utilisant l'adresse IP de votre hôte Windows 11 et le numéro de port que vous avez choisi (par exemple, http://<IP_address>:8080).

Important : Avant d'effectuer ces étapes, assurez-vous que votre machine virtuelle Debian a une adresse IP fixe. Sinon, vous risquez de perdre l'accès à la page Apache chaque fois que l'adresse IP change.

Job 07 : Mettre en place un serveur DHCP sur le serveur principal qui attribuera des adresses IP aux machines virtuelles présentes sur son réseau local.

Pour ce job, vous allez installer le serveur DHCP (Dynamic Host Configuration Protocol) sur votre serveur principal afin d'assigner dynamiquement des adresses IP à vos machines virtuelles.

1. *Mettez à jour le gestionnaire de paquets :*

   sudo apt update

2. *Installez le serveur DHCP :*

   sudo apt install isc-dhcp-server

3. *Arrêtez le serveur DHCP pour le moment :*

   sudo systemctl stop isc-dhcp-server

4. *Éditez le fichier de configuration du serveur DHCP :*

   sudo nano /etc/dhcp/dhcpd.conf

5. *Ajoutez les lignes suivantes à la fin du fichier de configuration. Changez l'adresse IP de début et de fin en fonction de votre plage d'adresses IP disponible. Ces adresses doivent être hors de la plage utilisée par votre serveur DNS.*

```
# Option subnet declaration
subnet 192.168.52.0 netmask 255.255.255.0 {
   range 192.168.52.10 192.168.52.150;
   option broadcast-addr 192.168.1.255;
   option routers 192.168.1.1;
   default-lease-time 600;
   max-lease-time 7200;
   }
```

Remarque : Remplacez "192.168.1." par l'adresse IP de votre réseau local. Enregistrez et fermez le fichier de configuration.

6. *Réinitialisez les droits d'accès au fichier de configuration :*

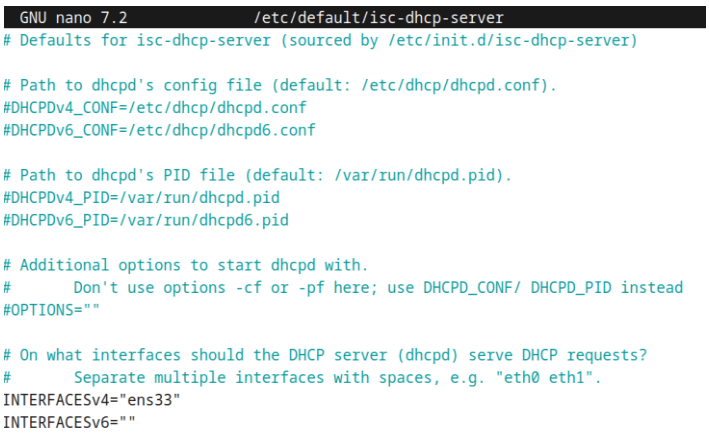
   sudo chmod +r /etc/dhcp/dhcpd.conf

7. *Éditez le fichier d'interfaces réseau :*

   sudo nano /etc/default/isc-dhcp-server

8. *Modifiez la variable INTERFACES en fonction de l'interface réseau de votre serveur principale. Habituellement, elle s'appelle eth0, mais elle peut varier selon votre configuration.*

   INTERFACES="eth0"



   Enregistrez et fermez le fichier de configuration.

9. *Redémarrez le service DHCP :*

   sudo systemctl start isc-dhcp-server

10. *Vérifiez le statut du service DHCP :*

    sudo systemctl status isc-dhcp-server

Maintenant, votre serveur DHCP est configuré et attribuera des adresses IP aux machines virtuelles qui rejoignent votre réseau local.

Job 08 : Faire en sorte que votre serveur principal serve de Gateway à vos autres machinesvirtuelles.

1. Activer le partage de connexion sur le serveur principal :

Assurez-vous que l'IP forwarding est activé. Vous pouvez le faire en modifiant le fichier /etc/sysctl.conf. Recherchez la ligne suivante et assurez-vous qu'elle est décommentée :

net.ipv4.ip_forward=1

Vous pouvez également activer cela temporairement en utilisant la commande suivante :

sudo sysctl -w net.ipv4.ip_forward=1

2.Configurez les règles iptables pour le partage de connexion

Assurez-vous d'ajuster les interfaces réseau (eth0, ens33, etc.) en fonction de votre configuration. Assurez-vous également d'ajuster l'interface qui a une connexion Internet.

sudo iptables -A FORWARD -i <interface_connexion_internet> -o <interface_locale> -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i <interface_locale> -o <interface_connexion_internet> -j ACCEPT

```
dome@debian12:~$ sudo iptables -A FORWARD -i ens33 -o 192.168.52.0 -m state --state RELATED,ESTABLISHED -j ACCEPT
dome@debian12:~$ sudo iptables -A FORWARD -i 192.168.52.0 -o ens33  -j ACCEPT
dome@debian12:~$ sudo apt-get install iptables-persistent
```

Pour rendre ces règles persistantes après le redémarrage, vous devrez utiliser une méthode appropriée pour votre distribution. Par exemple, avec Debian/Ubuntu, vous pouvez utiliser iptables-persistent :

sudo apt-get install iptables-persistent
sudo service netfilter-persistent save
sudo service netfilter-persistent restart

3.Configurer la passerelle sur VM client

Sur la deuxième machine virtuelle, configurez la passerelle (gateway) pour qu'elle pointe vers l'adresse IP du serveur principal. Vous pouvez le faire en éditant le fichier /etc/network/interfaces sur la deuxième machine virtuelle :

sudo nano /etc/network/interfaces

Assurez-vous que la ligne gateway ressemble à ceci :

gateway adresse_IP_du_serveur_principal

```
  GNU nano 7.2                              /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto ens33
iface ens33 inet dhcp

gateway 192.168.52.129
```

```
dome@debian12:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto ens33
iface ens33 inet dhcp

gateway 192.168.52.129
dome@debian12:~$
```

Sauvegardez le fichier et redémarrez l'interface réseau ou redémarrez la machine virtuelle.

4.Test de la connectivité Internet

Sur la deuxième machine virtuelle, essayez de pinguer une adresse IP sur Internet pour tester la connectivité :

ping 8.8.8.8

```
dome@debian12:~$ sudo systemctl restart networking
dome@debian12:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=524 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=279 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=270 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=277 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=269 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=275 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 6 received, 14.2857% packet loss, time 6006ms
rtt min/avg/max/mdev = 268.756/315.624/524.028/93.271 ms
dome@debian12:~$
```

Ces étapes permettront à la deuxième machine virtuelle de passer par le serveur principal pour accéder à Internet. Assurez-vous de configurer les règles iptables de manière persistante pour qu'elles soient appliquées après le redémarrage du serveur principal.

Job 09 : Mettre en place un pare-feu en utilisant ufw sur votre serveur principal

Un pare-feu est un élément essentiel de la sécurité de votre réseau. Il va filtrer le trafic entrant et sortant de votre serveur. Dans ce job, nous allons utiliser ufw (Uncomplicated FireWall), un pare-feu facile à configurer, pour sécuriser notre serveur Debian.

Étape 1 : Installer ufw

La première étape consiste à installer ufw sur votre serveur. Exécutez la commande suivante pour installer ufw :

sudo apt-get update
sudo apt-get install ufw

Étape 2 : Initialiser ufw

Par défaut, ufw est désactivé après l'installation. Pour l'activer, exécutez la commande suivante :

sudo ufw enable

```
dome@debian12:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
dome@debian12:~$
```

Étape 3 : Définir les règles de base

Nous allons maintenant définir les règles de base pour ufw. Tout d'abord, nous autoriserons les connexions SSH, sinon vous ne pourrez plus accéder à votre serveur via SSH :

sudo ufw allow ssh

Ensuite, nous autoriserons les connexions HTTP et HTTPS pour notre serveur web Apache :

sudo ufw allow http
sudo ufw allow https

Étape 4 : Bloquer les connexions ICMP (facultatif)

Par défaut, ufw accepte les paquets ICMP. Cependant, pour des raisons de sécurité, vous pouvez décider de les bloquer. Si vous souhaitez le faire, exécutez les commandes suivantes :

sudo ufw deny icmp
sudo ufw allow icmp echo-request

Étape 5 : Vérifier les règles ufw

Vous pouvez vérifier les règles ufw en exécutant la commande suivante :

sudo ufw status verbose

Vous devriez voir quelque chose comme ceci :

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         LIMIT IN    Anywhere
80                         ALLOW IN    Anywhere
443                        ALLOW IN    Anywhere
OpenSSH                    ALLOW IN    Anywhere
22 (v6)                    LIMIT IN    Anywhere (v6)
80 (v6)                    ALLOW IN    Anywhere (v6)
443 (v6)                   ALLOW IN    Anywhere (v6)
OpenSSH (v6)               ALLOW IN    Anywhere (v6)
```

Félicitations, vous avez configuré un pare-feu de base sur votre serveur Debian à l'aide d'ufw.

Job10: Création d'un dossier partagé sur le serveur Debian pour permettre le partage de fichiers dans le réseau

Pour ce dernier job, nous allons créer un dossier partagé sur notre serveur Debian qui sera accessible depuis les autres machines virtuelles du réseau. Pour ce faire, nous allons utiliser Samba, un logiciel de partage de fichiers multiplateforme.

Étape 1: Installation de Samba

Installez le paquet Samba en exécutant la commande suivante:

```
sudo apt-get update && sudo apt-get install samba
```

Étape 2: Création du dossier à partager

Créez le dossier à partager dans le répertoire personnel de l'utilisateur que vous souhaitez utiliser pour le partage:

```
mkdir ~/partage
```

Étape 3: Configuration de Samba

Éditez le fichier de configuration de Samba:

```
sudo nano /etc/samba/smb.conf
```

Et ajoutez les lignes suivantes à la fin du fichier:

```
[partage]
comment = Partage de fichiers
path = /home/yourusername/partage
 browsable = yes
 guest ok = yes
 read only = no
 create mask = 0777
 directory mask = 0777
```

Remplacez "yourusername" par votre nom d'utilisateur.

Étape 4: Redémarrage de Samba

Redémarrez le service Samba pour prendre en compte les modifications:

sudo service smbd restart

Étape 5: Montage du dossier partagé sur les autres machines virtuelles

Depuis les autres machines virtuelles, vous pouvez monter le dossier partagé en utilisant l'adresse IP du serveur Debian et le nom du partage défini dans le fichier de configuration Samba:

sudo mount -t cifs //<IP_du_serveur>/partage /mnt -o user=<nom_utilisateur>,password=<mot_de_passe>

Remplacez "<IP_du_serveur>" par l'adresse IP du serveur Debian, "<nom_utilisateur>" par votre nom d'utilisateur et "<mot_de_passe>" par votre mot de passe.

Votre dossier partagé est maintenant monté sur la machine virtuelle et accessible dans le répertoire "/mnt". Vous pouvez y copier des fichiers et y accéder depuis le serveur Debian et les autres machines virtuelles du réseau.

Job Pour aller plus loin... : activation de HTTPS sur votre serveur web Apache

Pour augmenter la sécurité de votre serveur web Apache, vous pouvez activer le protocole HTTPS (Hypertext Transfer Protocol Secure) en installant un certificat SSL (Secure Sockets Layer). Il existe deux types de certificats SSL : auto-signés et signés par une autorité de certification. Les certificats auto-signés sont gratuits, tandis que les certificats signés par une autorité de certification sont payants. Dans ce tutorial, nous allons voir comment installer un certificat auto-signé.

Étape 1 : Installer OpenSSL

Installez OpenSSL en exécutant la commande suivante :

sudo apt-get install openssl

Étape 2 : Générer une clef privée et un certificat SSL

Créez un dossier pour stocker les fichiers de certificat SSL :

sudo mkdir /etc/ssl/private

Générez une clef privée et un certificat SSL en exécutant les commandes suivantes :

cd /etc/ssl/private
sudo openssl req -newkey rsa:2048 -nodes -x509 -days 365 -keyout example.com.key -out example.com.crt

Remplacez "example.com" par votre nom de domaine.

Entrez les informations demandées. La question "Common Name" doit contenir votre nom de domaine complet (FQDN).

Étape 3 : Configurer Apache pour utiliser le certificat SSL

Modifiez le fichier de configuration d'Apache pour activer le SSL :

sudo nano /etc/apache2/sites-enabled/000-default.conf

*Ajoutez les lignes suivantes dans la section <VirtualHost *:80> :*

Redirect permanent / https://yourdomain.com/

Enregistrez et quittez le fichier.

Créez un nouveau fichier de configuration pour le SSL :

sudo nano /etc/apache2/sites-available/example.com-ssl.conf

Copiez et collez le bloc de configuration suivant :

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin webmaster@example.com
        ServerName example.com
        ServerAlias www.example.com
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLCertificateFile /etc/ssl/private/example.com.crt
        SSLCertificateKeyFile /etc/ssl/private/example.com.key
</VirtualHost>
</IfModule>
```

Remplacez "example.com" par votre nom de domaine.

Enregistrez et quittez le fichier.

Activez le SSL et le nouveau site web :

```
sudo a2ensite example.com-ssl.conf
sudo a2enmod ssl
sudo systemctl restart apache2
```

Testez votre site web en accédant à l'URL "<https://yourdomain.com>".

Attention : les navigateurs web considèrent les certificats auto-signés comme non fiables et afficheront un avertissement de sécurité. Les certificats signés par une autorité de certification sont reconnus par les navigateurs web et évitent ces avertissements.

Job Pour aller plus loin...activation de HTTPS sur votre serveur web Apache

Pour augmenter la sécurité de votre serveur web Apache, vous pouvez activer le protocole HTTPS (Hypertext Transfer Protocol Secure) en installant un certificat SSL (Secure Sockets Layer). Il existe deux types de certificats SSL : auto-signés et signés par une autorité de certification. Les certificats auto-signés sont gratuits, tandis que les certificats signés par une autorité de certification sont payants. Dans ce tutorial, nous allons voir comment installer un certificat auto-signé.

Étape 1 : Installer OpenSSL

Installez OpenSSL en exécutant la commande suivante :

sudo apt-get install openssl

Étape 2 : Générer une clef privée et un certificat SSL

Créez un dossier pour stocker les fichiers de certificat SSL :

sudo mkdir /etc/ssl/private

Générez une clef privée et un certificat SSL en exécutant les commandes suivantes :

cd /etc/ssl/private
sudo openssl req -newkey rsa:2048 -nodes -x509 -days 365 -keyout example.com.key -out example.com.crt

Remplacez "example.com" par votre nom de domaine.

Entrez les informations demandées. La question "Common Name" doit contenir votre nom de domaine complet (FQDN).

Étape 3 : Configurer Apache pour utiliser le certificat SSL

Modifiez le fichier de configuration d'Apache pour activer le SSL :

sudo nano /etc/apache2/sites-enabled/000-default.conf

*Ajoutez les lignes suivantes dans la section <VirtualHost *:80> :*

Redirect permanent / https://yourdomain.com/

Enregistrez et quittez le fichier.

Créez un nouveau fichier de configuration pour le SSL :

sudo nano /etc/apache2/sites-available/example.com-ssl.conf

Copiez et collez le bloc de configuration suivant :

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin webmaster@example.com
        ServerName example.com
        ServerAlias www.example.com
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLCertificateFile /etc/ssl/private/example.com.crt
        SSLCertificateKeyFile /etc/ssl/private/example.com.key
</VirtualHost>
</IfModule>
```

Remplacez "example.com" par votre nom de domaine.

Enregistrez et quittez le fichier.

Activez le SSL et le nouveau site web :

```
sudo a2ensite example.com-ssl.conf
sudo a2enmod ssl
sudo systemctl restart apache2
```

Testez votre site web en accédant à l'URL "<https://yourdomain.com>".

Attention : les navigateurs web considèrent les certificats auto-signés comme non fiables et afficheront un avertissement de sécurité. Les certificats signés par une autorité de certification sont reconnus par les navigateurs web et évitent ces avertissements.