Job 01

Installation VM

I. Installation

Afin de créer un serveur Apache2 nous allons installer une machine virtuelle debian 12 avec interface graphique. Je vais utiliser VMWare Workstation Pro.

II. Accès SSH

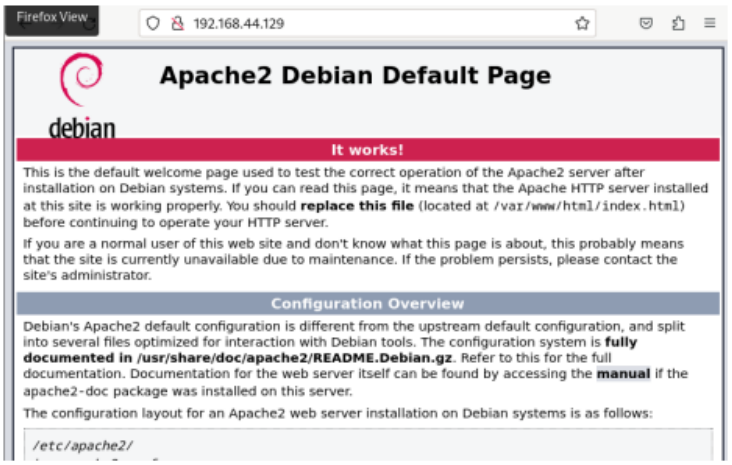
Que ce soit en configurant notre serveur web depuis la machine virtuelle ou en établissant une connexion SSH depuis notre machine hôte, les deux approches sont envisageables. Dans mon cas, j'ai opté pour la configuration directe depuis la machine virtuelle. Commençons par vérifier que le service SSH est opérationnel en utilisant la commande `systemctl status ssh`. Une fois cette vérification effectuée, sur notre machine hôte, nous pouvons entrer la commande suivante : `ssh 192.168.44.129`, qui représente l'adresse IP de ma machine virtuelle. Ensuite, je saisis le mot de passe et voilà, nous sommes connectés.

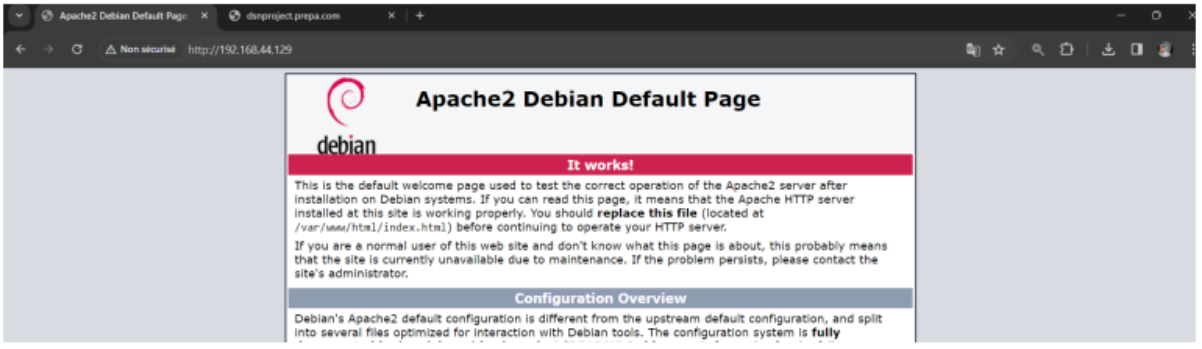
Job 02

Installation serveur Apache

I. Apache2

Pour mettre en place Apache2, nous procédons en utilisant la commande `apt install apache2`. Après son installation, aucune configuration supplémentaire n'est nécessaire, les paramètres par défaut conviennent parfaitement à cet exercice. Pour visualiser la page web, il suffit d'entrer l'adresse IP de notre machine virtuelle dans le navigateur.





Nous remarquons que nous pouvons accéder à la page web depuis notre VM et depuis notre machine hôte.

Job 03:

Serveurs web:

Il existe de nombreux serveurs web parmi lesquels les 5 les plus utilisés sont:

- Apache HTTP Server (Apache) : Apache représente l'un des serveurs web les plus anciens et les plus largement adoptés. Il s'agit d'un logiciel open source qui offre une extensibilité grâce à l'utilisation de modules. Les atouts de son utilisation résident principalement dans sa stabilité éprouvée et sa documentation exhaustive. Il se distingue par sa polyvalence et intègre un module de sécurité robuste pour contrer les attaques. Cependant, les inconvénients incluent sa complexité potentielle pour les débutants, en particulier lors de la sécurisation, et la nécessité de redémarrages fréquents lors de modifications de la configuration.

- Nginx : Nginx se distingue par ses performances élevées et sa consommation réduite de ressources. Il est fréquemment choisi comme serveur proxy ou pour la répartition de charge. Ses points forts incluent une grande efficacité, capable de gérer de multiples connexions simultanées, ainsi qu'un équilibrage de charge et un proxy inversé très performants. De plus, il se montre peu gourmand en ressources. Toutefois, ses inconvénients résident dans sa courbe d'apprentissage potentiellement abrupte pour les débutants, l'absence de support natif pour certains langages serveur tels que PHP, et la nécessité d'utiliser des proxies pour pallier cette limitation.

- Microsoft Internet Information Services (IIS) : IIS constitue le serveur web de Microsoft, prédominant sur les serveurs Windows. Ses atouts résident dans son intégration native avec les systèmes Windows Server, sa prise en charge des technologies Microsoft telles que ASP.NET, et sa facilité d'administration via une interface graphique conviviale. Cependant, ses limitations se manifestent dans son exclusivité aux environnements Windows, le rendant moins répandu en dehors de cet écosystème.

- LiteSpeed : Célèbre pour sa rapidité et sa fiabilité en termes de sécurité, LiteSpeed se démarque. Il est compatible avec Apache et peut être une alternative sans nécessiter des modifications majeures de configuration. Sa performance est exceptionnelle, particulièrement adaptée aux sites à fort trafic. La transition depuis Apache peut se faire sans difficulté notable, et il offre des fonctionnalités de sécurité robustes. Son principal inconvénient réside dans le fait qu'il n'est pas en open source, et la version gratuite propose des fonctionnalités limitées.

- Caddy : En tant que serveur web contemporain, Caddy se démarque par sa facilité d'installation et de configuration. Sa principale force réside ainsi dans sa simplicité d'installation et de configuration. Il offre également un support natif du chiffrement HTTPS et inclut une interface de gestion web. Cependant, ses limitations résident dans son niveau de flexibilité inférieur par rapport à certains autres serveurs, et il peut afficher des performances légèrement moindres dans certaines situations comparé à des serveurs web tels que Nginx.

Job 04:

DNS:

I. Installation et configuration bind9

Pour établir un serveur DNS, nous pouvons utiliser Bind9. Son installation se fait à l'aide de la commande `apt install bind9`.

Une fois l'installation terminée, la configuration nécessite l'ajustement de différents fichiers pour créer notre propre domaine.

Pour commencer, la configuration du fichier de zone directe est essentielle. Nous procédons en copiant le fichier `db.local` vers `direct` avec la commande `sudo cp /etc/bind/db.local /etc/bind/direct`.

Une zone directe, également appelée zone "forward" dans un serveur DNS, s'occupe de la résolution des noms de domaine vers des adresses IP. Concrètement, elle établit des correspondances entre des noms de domaine spécifiques et leurs adresses IP respectives. Dans notre contexte, elle associera le nom `dnsproject.prepa.com` à l'adresse IP de notre machine. À noter que les zones inverses effectuent l'opération inverse.

La prochaine étape implique la modification du fichier de zone directe à l'aide de la commande `sudo nano /etc/bind/direct`.

```
 GNU nano 7.2                        /etc/bind/direct
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     prepa.com. dnsproject.prepa.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsproject.prepa.com.
dnsproject      IN      A       192.168.0.15
www     IN      CNAME   dnsproject.prepa.com.
```

Une fois le direct configuré nous allons le copier dans inverse avec sudo cp /etc/bind/direct /etc/bind/inverse.

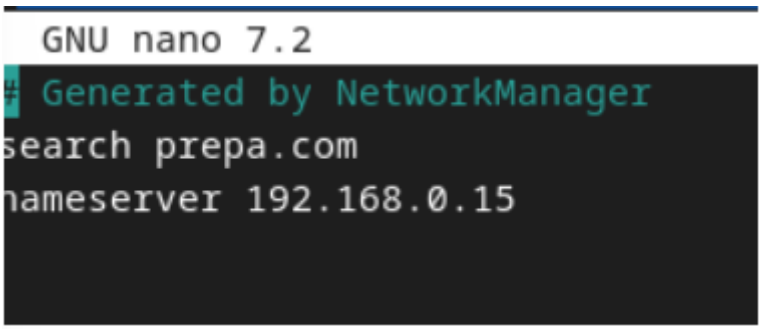
```
  GNU nano 7.2                        /etc/bind/inverse
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     prepa.com. dnsproject.prepa.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dsnproject.prepa.com.
dnsproject      IN      A       192.168.0.15
10      IN      PTR     dnsproject.prepa.com.
```

Ensuite nous allons éditer le fichier named.conf.local situé dans /etc/bind/.

```
  GNU nano 7.2                 /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "prepa.com" IN {
        type master;
        file"/etc/bind/direct";
};
zone "9.10.10.in-addr-arpa" IN {
        type master;
        file"/etc/bind/inverse";
};
```

Une fois tout ceci réalisé nous allons configurer le fichier resolv.conf situé dans le dossier etc.



Une fois réalisé ces étapes nous pouvons redémarrer le service bind9.

II. Test

Nous allons donc maintenant tester que notre configuration soit fonctionnelle sur notre VM et que nous puissions ping et afficher sur firefox notre site à partir de l'adresse dnsproject.prepa.com.

Job 05

Domaine public

En règle générale, l'acquisition d'un nom de domaine implique de choisir un fournisseur de services de noms de domaine, tels que GoDaddy, Namecheap, Google Domains, parmi d'autres.

Ensuite, il est nécessaire de vérifier la disponibilité du nom de domaine sur le site du prestataire, car deux sites distincts ne peuvent pas partager le même nom de domaine.

Le choix de l'extension de domaine (Top-Level Domain), comme .com, .net, .org, .fr, .io, etc., doit également être effectué. Chaque extension de domaine présente ses propres caractéristiques et restrictions (voir fin du Job05). Les étapes suivantes consistent à suivre les instructions du site du prestataire.

Une fois le nom de domaine enregistré, la configuration des enregistrements DNS est nécessaire. Ces enregistrements pointent vers les serveurs de votre site web ou de votre service de messagerie.

Enfin, des frais d'enregistrement annuels doivent être réglés pour maintenir la propriété du nom de domaine.

Il existe divers types d'extensions de domaine (TLD) :

1. gTLD (Generic Top-Level Domain) : Ces extensions, telles que .com, .org, .net, sont génériques et largement utilisées. Elles sont disponibles pour tous, sans restrictions particulières.

2. ccTLD (Country Code Top-Level Domain) : Ces extensions sont associées à des pays ou territoires, comme .fr (France), .uk (Royaume-Uni), .de (Allemagne), etc. Les ccTLD sont souvent soumis à des restrictions géographiques, nécessitant parfois une présence ou une adresse dans le pays pour l'enregistrement.

3. TLD de Second Niveau Restreints : Certains TLD de second niveau, tels que .gov, .edu, sont réservés à des entités spécifiques comme les organismes gouvernementaux ou les établissements d'enseignement.

4. TLD de Second Niveau Génériques : Certains pays et registres proposent des TLD de second niveau génériques, accessibles à tous sans restriction particulière.

5. Nouvelles Extensions (gTLD) : Ces extensions, comme .app, .io, .blog, sont plus récentes et spécifiques. Elles offrent parfois des opportunités intéressantes pour des noms de domaine pertinents.

Job 06

DNS machine hôte

Afin de réaliser ce job, il est conseillé de configurer l'adaptateur réseau de notre VM en bridge ou en host-only. Sachant qu'avec host-only seuls notre hôte et notre VM communiqueront. Dans mon cas, je vais relancer la VM en mode host-only.

I. Changer le dns de notre connexion wifi sur notre machine hôte

Afin de pouvoir utiliser le serveur dns de notre machine hôte (ici Windows 11) nous pouvons aller dans réseau, cliquer sur la configuration du wifi et dans la partie dns selectionner manuel et rentrer l'adresse ip de notre serveur dns (notre VM).

II. Test

Maintenant nous pouvons essayer d'accéder à notre serveur web à partir du navigateur web de notre machine hôte

Job 07

Pare-feu ufw (firewall):

Bloquer les pings (ICMP) sur un serveur ou un pare-feu offre des avantages en matière de sécurité, notamment la réduction de la visibilité pour les attaquants et la réduction de la surface d'attaque. Il permet de se protéger des attaques DDOS. Cela peut également améliorer les performances dans certains cas. Cependant, cela peut compliquer le dépannage, avoir un impact sur certains services réseau et ne garantit pas une protection complète.

I. Installation et configuration de ufw (uncomplicated firewall)

Utilisons sudo apt install ufw, pour installer ufw. Une fois installé nous pouvons passer à la configuration de celui-ci.

Nous devons configurer le fichier nommé before.rules situé par défaut dans /etc/ufw.

Nous allons l'éditer avec nano: sudo nano /etc/ufw/before.rules.

Dans # ok icmp codes for INPUT, nous allons remplacer ACCEPT par DROP.

Ensuite nous allons permettre le port 80 pour le protocole HTTP et le port 443 pour le protocole HTTPS.

Nous allons utiliser : sudo ufw allow 80/tcp et sudo ufw allow 443/tcp.

Nous pouvons vérifier les règles appliquées avec sudo ufw status

Pour activer le firewall nous utiliserons sudo ufw enable.

II. Test

Nous pouvons remarquer que les ping ne fonctionnent plus. Tous les paquets envoyés sont perdus. Néanmoins la page web reste accessible.

Job 08

SMB

Pour mettre en place un dossier partagé sur notre serveur, nous pouvons utiliser le protocole SMB (Server Message Block) pour créer un partage réseau. Cela permettra aux autres membres de votre réseau d'accéder aux fichiers et de partager des fichiers dans ce dossier.

I. Installation et configuration de Samba

Nous allons utiliser Samba, nous utiliserons alors sudo apt install samba.

Nous allons le configurer un peu plus tard.

Maintenant nous allons créer un dossier qui va être celui du partage, dans mon cas, j'ai créé ce dossier dans le dossier home. Nous utiliserons sudo mkdir /home/dossier-partage.

Ensuite nous lui accordions tous les droits à tous les utilisateurs. Ceci n'est pas une bonne pratique car ça suppose un problème de sécurité très gros. Par simplicité j'ai choisi de donner tous les droits pour cet exercice. Nous utiliserons sudo chmod -R 777 /home/dossier-partage.

Éditons dès à présent la configuration de samba.

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

```
[Partage]
comment = Dossier de partage
path = /home/dossier-partage
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
writable = yes
```

Nous allons ajouter à la fin du fichier ce bloc. Nous n'avons pas besoin de tous ces attributs, il s'agit de la configuration des règles spécifiques à ce partage. Le nom entre les crochets est le nom pour pouvoir accéder au dossier partagé.

Il ne nous reste plus qu'à configurer un mot de passe pour l'utilisateur avec sudo smbpasswd -a eleat.

Nous pouvons alors restart le service, systemctl restart samba.

II. Configuration ufw

Notre nouveau firewall empêche la connexion par le protocole smb, pour permettre la connexion nous pouvons ouvrir le port 139 et le port 445. Nous utiliserons les commandes sudo ufw allow 139/tcp et sudo ufw allow 445/tcp.

III. Test

Nous pouvons dès à présent tester depuis le navigateur de fichier de notre machine hôte.

Utilisons \\10.10.11.178\Partage dans l'explorateur de fichiers.

Il nous demandera alors de nous connecter. Nous utilisons l'utilisateur de ma vm, eleat, et le mot de passe celui défini dans samba, dans notre cas, root.

Une fois connectés nous pouvons voir que les fichiers se partagent correctement.