~$ DDWS

Mise en place et configuration de différents types de serveurs

Ilian Bonsens
LA PLATEFORME

Table des matières

Job N°1 2

Job N°2 3

Job N°3 3

Job N°4 4

Job N°5 6

Job N°7 7

Job N°1

Après avoir installé notre VM Debian et mis à jour nos paquets en utilisant successivement `sudo apt-get update` et `sudo apt-get upgrade`, commençons par installer un serveur SSH sur notre système. Pour le faire, nous avons installé le logiciel OpenSSH.

```
ilian@debian:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Une fois les paquets installés, il suffit d'activer le serveur en utilisant `systemctl start ssh.` Pour s'assurer qu'il se remette en route à chaque démarrage, on peut utiliser `systemctl enable ssh`. On peut vérifier si le serveur SSH s'est bien activé en utilisant la commande systemctl status ssh, si c'est le cas cela donnera :

```
ilian@debian:~$ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Thu 2023-10-26 10:12:48 CEST; 1min 37s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3443 (sshd)
      Tasks: 1 (limit: 2285)
     Memory: 1.7M
        CPU: 57ms
     CGroup: /system.slice/ssh.service
             └─3443 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Enfin, pour tester le dispositif, il nous suffit de retourner sur notre système Windows et d'y rentrer la commande suivante dans le PowerShell : `ssh ilian@192.168.1.24`. Si la connexion s'effectue bien, on devrait avoir ce résultat.

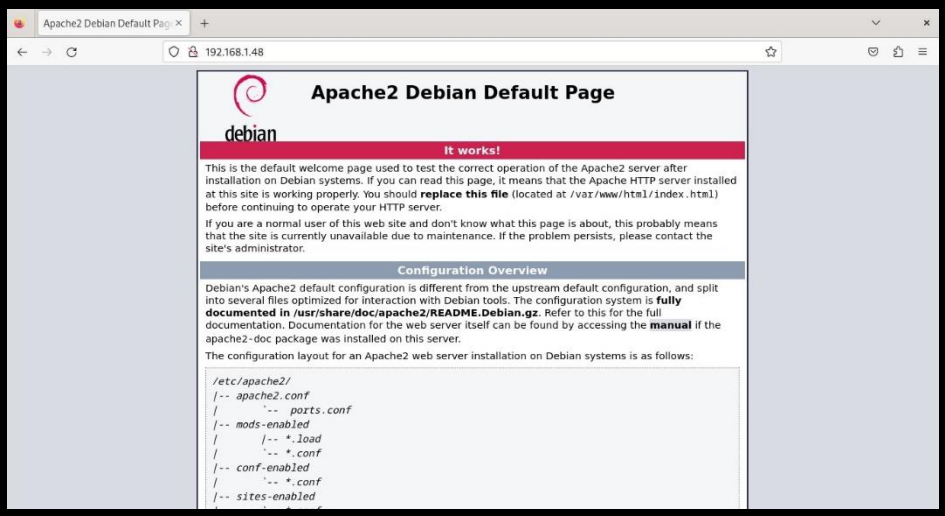
```
ilian@debian12:~$
```

Le protocole SSH est idéal pour se connecter et gérer à distance son serveur (ici notre VM) depuis d'autres systèmes. Il garantit la sécurité de la connexion en chiffrant les données et en donnant la possibilité de mettre en place une authentification lors de la connexion au serveur.

Job N°2

Passons maintenant à l'installation de notre serveur Web. Nous allons cette fois utiliser Apache2 que nous allons installer en utilisant la commande `sudo apt-get install apache2`. Comme pour l'étape précédente, nous allons démarrer le service en utilisant `systemctl start apache2` et nous allons faire en sorte qu'il se remette toujours en marche à chaque démarrage en utilisant `systemctl enable apache2`.

Pour vérifier que le serveur est bien configuré nous pouvons utiliser `systemctl status apache2` pour voir s'il est actif. Finalement, il suffit de récupérer l'adresse IP de notre serveur en utilisant `hostname -I`; puis dans un navigateur il faut rechercher l'adresse http://192.168.1.48. Si le serveur est bien en route cela nous renvoie sur la page suivante.



Job N°3

Il existe de nombreux services de serveur Web, qui peuvent correspondre chacun à différents besoins et différents utilisateurs. En voici une liste, non exhaustive car il en existe une variété trop importante pour pouvoir tous les lister de manière concise.

Nom	Avantages	Inconvénients
Apache	Open Source, grande communauté,	Moins performant pour gérer beaucoup de

	documentation abondante en ligne	connexions simultanées, réglages plus complexes que d'autres
Nginx	Gestion des ressources efficaces, grand nombre de connexion en simultané	Configuration complexe,
Microsoft IIS	Excellente intégration dans l'écosystème Microsoft	Principalement limité aux systèmes sous Windows
Caddy	Sécurité renforcée grâce à la prise en charge automatique du SSL, configuration simple	Moins répandu communauté donc plus réduite, version payante pour accéder à toute les fonctions
LiteSpeed	Performances élevées, gestion d'un nombre très important de connexions, convient aux hébergements partagés	Mêmes inconvénients que Caddy

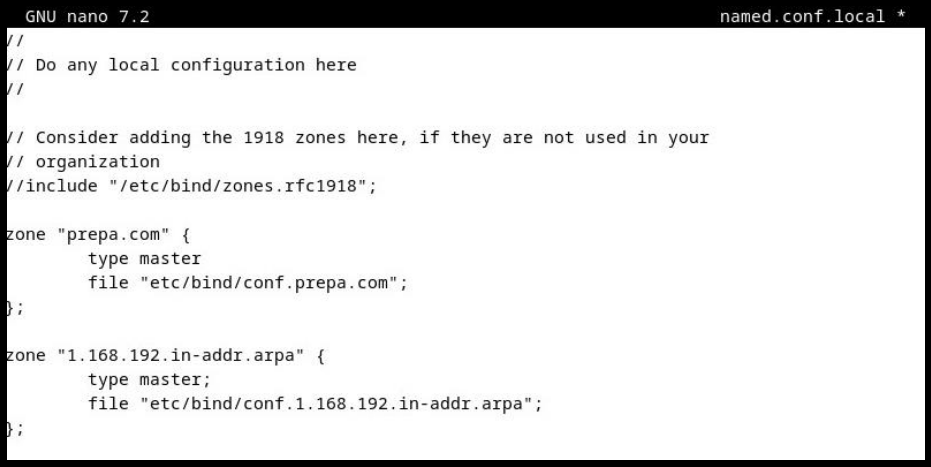
Job N°4

Passons maintenant à l'installation et la configuration d'un serveur DNS sur la VM. Nous allons installer Bind 9 qui est un service de serveur DNS open source, qui offre une grande stabilité et une grande flexibilité dans la gestion des configurations. Néanmoins, son principal inconvénient est sa complexité à l'utilisation, sa configuration se faisant entièrement en lignes de textes brutes, comme nous allons le voir par la suite.

Pour l'installer, pas de changement majeur, on utilise la commande `sudo apt-get install bind9 bind9utils dnsutils`. Allons maintenant dans le répertoire bind en utilisant `cd ../../etc/bind`, on a créé une copie du fichier db.local qui correspond à la base de données locale du serveur, que l'on va ensuite éditer.

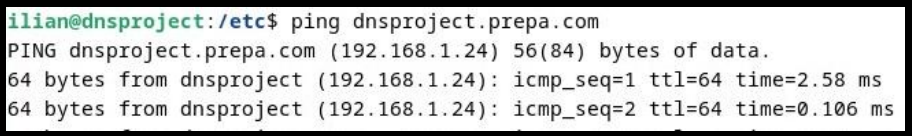
```
 GNU nano 7.2                                                db.local.copie *
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     prepa.com       dnsproject.prepa.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnsproject.prepa.com.
dnsproject      IN      A       127.0.0.1
www     IN      CNAME   dnsproject.prepa.com.
```

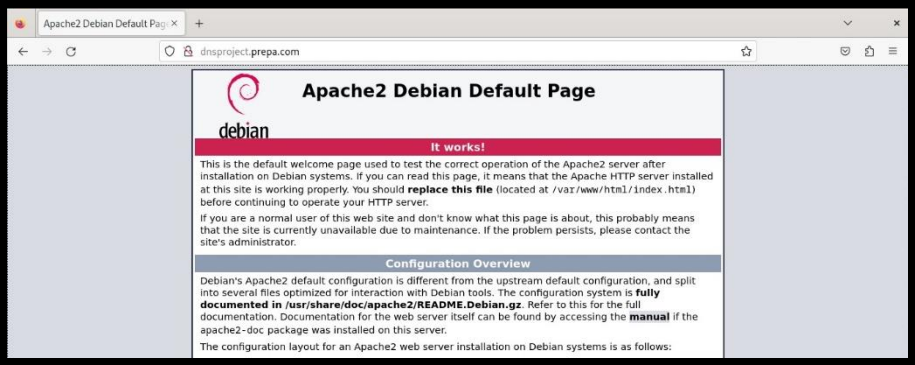
Comme on peut le voir ici, on a rajouté notre adresse IP et notre domaine dans la base de données locale du serveur. Cela va permettre la redirection de la requête du nom de domaine vers l'adresse IP du site Web.
NB : sur l'image l'adresse IP est celle par défaut mais on l'a bien remplacé par notre adresse IP.



Nous avons par la suite éditer le fichier named.conf.local (en créant une copie au préalable comme nous l'avons fait plus tôt). On y à implémenter l'adresse IP ainsi que le domaine et l'adresse IP inverse suivie du terme in-addr.arpa qui est utilisé pour la résolution inverse des adresses IP. En faisant cela, nous avons déclarer clairement nos zones de DNS.

De retour dans le répertoire etc, nous avons éditer le fichier resolv.conf qui encore une fois permet la bonne résolution des adresses IP pendant les requêtes de noms de domaines. On y a implémenté notre adresse IP ainsi que le domaine de notre page Web. Après toutes ces étapes, nous avons redémarré le serveur DNS pour nous assurer que tous ces changements ont été pris en compte ; en utilisant la commande `systemctl restart bind9`.





Comme on le constate sur ces deux images, la résolution a bien fonctionné, on peut accéder au serveur directement en utilisant le nom de domaine dnsproject.prepa.com.
NB : tout au long du devoir, les adresses IP peuvent parfois différer car certaines captures d'écrans ont été prises à différents moments.

Job N°5

- Comment obtient-on un nom de domaine public ?

Pour obtenir un nom de domaine publique, il y a quelques étapes simples à suivre. Premièrement il faut choisir un bon fournisseur de nom de domaine. Ce sont différentes entreprises qui s'occupent de la gestion et de l'attribution des noms de domaines. En effet, il faut que le nom de domaine désiré soit libre de toute utilisation pour pouvoir être choisi. Il suffit ensuite de procéder au paiement pour le nom de domaine retenu et de configurer les adresses DNS sur notre serveur pour les faire correspondre à notre nouveau nom de domaine. L'utilisation d'un nom de domaine étant payante, il faut renouveler le paiement à intervalle régulière, définie lors de l'acquisition.

- Quelles sont les spécificités que l'on peut avoir sur certaines extensions de nom de domaine ?

Il existe une grande variété de noms de domaines et donc une grande diversité dans les extensions de ces derniers, qui existent pour assurer les différents besoins des différents acteurs présents sur le web. Pour en citer quelques-unes :

- Les extensions géographiques : comme leur nom l'indique, elles sont associées à des pays ou des régions en particulier et peuvent nécessiter une présence physique ou légale de son propriétaire dans la zone en question (.fr / .de / .uk) ;

- Les extensions sectorielles : elles sont réservées à certaines activités particulières comme la .gov pour les gouvernements ou .edu pour les institutions éducatives ;

- Les extensions à caractère internationaux : permettent l'utilisation d'alphabets différents dans les noms de domaines (cyrillique, arabe, etc) ;

- Les extensions de marque : certaines marques peuvent vouloir réserver leur propre extension dans le cadre de la mise en place de leurs différents sites web ;

- Les extensions personnalisées : ce sont de nouvelles extensions qui permettent de mieux s'adapter aux différents buts et thèmes des sites webs créés, par exemple .blog ou encore .app.

Job N°7

Pour mettre en place un pare-feu sur notre serveur, nous avons choisi d'utiliser le service Uncomplicated Firewall, que nous téléchargeons en utilisant `sudo apt-get install uwf`.

```
ilian@dnsproject:/etc$ sudo apt-get install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Cela étant fait, il faut maintenant configurer le pare-feu de sorte qu'il soit possible d'accéder à la page web par son nom de domaine mais qu'il soit impossible de lui envoyer des paquets (ping). Pour le faire, commençons par définir que par défaut, toutes les connexions entrantes ou sortantes seront refusées.

```
ilian@dnsproject:/etc$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
ilian@dnsproject:/etc$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Ping le site est maintenant impossible, il ne nous reste plus qu'à le rendre consultable via son nom de domaine.

Nous allons autoriser les connexions à certains ports spécifiques, qui sont utiliser justement pour les entrées et sorties de trafic en utilisant le protocole TCP. Il est couramment utilisé par les sites pour y accéder en utilisant le protocole HTTP. Nous utilisons donc la commande `sudo uwf allow [port]/tcp`. Ceci avec les ports 80, 139 et 445.

```
ilian@dnsproject:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
ilian@dnsproject:~$ sudo ufw allow 139/tcp
Rule added
Rule added (v6)
ilian@dnsproject:~$ sudo ufw allow 445/tcp
Rule added
Rule added (v6)
```

Enfin, nous avons bloqué le protocole ICMP, qui est responsable du traitement des pings lorsqu'ils arrivent. En faisant cela on s'assure définitivement que les paquets des pings ne pourront plus rentrer. Pour faire cela il faut utiliser la commande : `sudo ufw deny proto icmp.`

Finalement, on peut activer le pare-feu en utilisant ufw enable. Il sera actif et se mettra en route à chaque démarrage.

Job N°8