

Documentation : DDWS

Job 1 :

Pour le premier job il nous faut installer une VM Debian avec une interface graphique et lors de l'installation on lui configure le SSH.



Job 2 :

Nous allons installer le serveur Web Apache2 sur le terminal de Debian. On oublie de se mettre en tant que sudoers.

Puis on entre la commande :

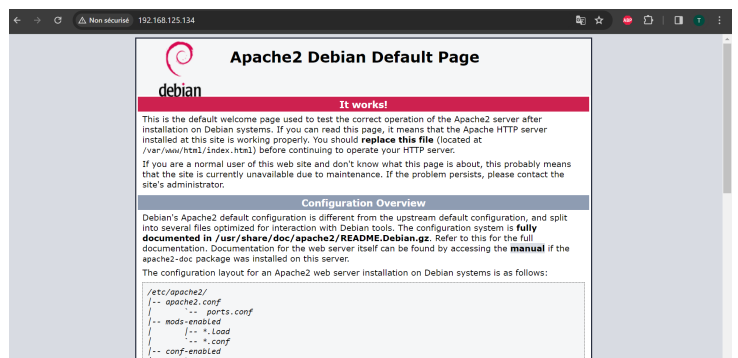
sudo apt-get install apache2

Une fois installé on entre la commande **sudo systemctl start apache2**

Puis on prend son IP avec la commande **ip a** et on entre son ip(192.168.125.134) dans l'url d'un navigateur .

Ici le serveur est atteignable directement depuis notre hôte.

```
tom@debian: ~  
tom@debian: $ sudo apt-get install apache2  
[sudo] Mot de passe de tom :  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  apache2-data apache2-utils  
Paquets suggérés :  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom  
Les NOUVEAUX paquets suivants seront installés :  
  apache2 apache2-data apache2-utils  
0 mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.  
Il est nécessaire de prendre 577 ko dans les archives.  
Après cette opération, 1 890 ko d'espace disque supplémentaires seront utilisés.  
Souhaitez-vous continuer ? [O/n] O  
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 apache2-data all 2.4.57-2 [160 kB]  
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 apache2-utils amd64 2.4.57-2 [202 kB]  
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 apache2 amd64 2.4.57-2 [215 kB]  
577 ko réceptionnés en 0s (3 191 ko/s)  
Sélection du paquet apache2-data précédemment désélectionné.  
(Lecture de la base de données... 154879 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de .../apache2-data 2.4.57-2 all deb ...
```





Job 3 :

Les serveurs webs les plus connus et utilisés sont :

- Apache



- NGINX



- IIS



- Lighttpd



- Caddy



Apache :

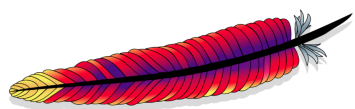
Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est maintenu et développé par Apache Software Foundation.

Il permet aux propriétaires de sites web de servir du contenu sur le web – d'où le nom **serveur web** -. C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version sortie il y a plus de 20 ans, en 1995.

Il a plusieurs avantages comme :

-Open Source : Apache est un logiciel open source, ce qui signifie qu'il est gratuit à télécharger, à utiliser et à personnaliser. Cela le rend accessible à un large public.

-Fiabilité : Apache est connu pour sa stabilité et sa fiabilité. Il peut gérer de grandes charges de trafic sans crash ni problème majeur.



Apache Web Server

-Modularité : Apache est extensible grâce à des modules tiers. Vous pouvez ajouter des fonctionnalités supplémentaires en installant des modules adaptés à vos besoins.

-Support communautaire : Il existe une grande communauté d'utilisateurs d'Apache qui partagent des ressources, de l'aide et des conseils en ligne. Cela facilite la résolution de problèmes et la mise en place de configurations avancées.

-Plateforme multiplateforme : Apache est compatible avec de nombreuses plates-formes, y compris Unix, Linux, Windows, MacOS, etc.

Inconvénients :

-Configuration complexe : La configuration d'Apache peut être complexe, en particulier pour les utilisateurs novices. Il nécessite des connaissances techniques pour une configuration avancée.

-Surconsommation de ressources : Apache peut consommer plus de ressources système que d'autres serveurs web, ce qui peut affecter les performances si mal configurées.

-Vitesse : Bien qu'Apache soit fiable, il peut être moins performant en termes de vitesse par rapport à d'autres serveurs web, en particulier lorsque des options de configuration incorrectes sont utilisées.

NGINX :

NGINX, prononcé comme « engine-ex », est un serveur web open-source qui, depuis son succès initial en tant que serveur web, est maintenant aussi utilisé comme reverse proxy, cache HTTP, et load balancer.

Nginx a été créé à l'origine par Igor Sysoev, avec sa première sortie publique en octobre 2004. Igor a d'abord conçu le logiciel comme une réponse au problème du C10k, qui est un problème de performance lié à la gestion de 10.000 connexions simultanées.





Avantages :

-Haute performance : Nginx est réputé pour sa capacité à gérer un grand nombre de connexions simultanées avec une utilisation efficace des ressources système, ce qui en fait un choix solide pour les sites web à fort trafic.

-Léger : Nginx a une empreinte mémoire réduite par rapport à certains autres serveurs web, ce qui le rend adapté à des environnements avec des ressources limitées.

-Extensibilité : Il offre des modules qui peuvent être ajoutés pour étendre ses fonctionnalités, ce qui permet de le personnaliser en fonction des besoins spécifiques de votre application.

-Équilibrage de charge : Nginx est souvent utilisé comme proxy inverse pour répartir la charge entre plusieurs serveurs, ce qui améliore la disponibilité et les performances.

-Configuration simple et lisible : Sa configuration est généralement considérée comme plus simple et plus lisible que celle d'Apache, ce qui peut faciliter l'administration.

Inconvénients :

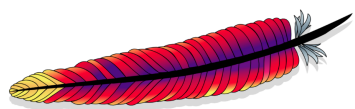
-Configuration différente : La syntaxe de configuration de Nginx est différente de celle d'Apache, ce qui peut rendre la transition difficile si vous êtes habitué à Apache.

-Applications dynamiques : Bien que Nginx puisse servir des fichiers statiques rapidement, il nécessite généralement un serveur d'application supplémentaire pour gérer des applications web dynamiques, ce qui peut compliquer la configuration.

-Moins de modules : Nginx a moins de modules tiers disponibles que Apache, bien que cela puisse être atténué par des modules tiers de qualité.

-Documentation moins abondante : Par rapport à Apache, la documentation de Nginx peut sembler moins abondante en raison de sa relative nouveauté, bien que la situation se soit améliorée au fil du temps.





IIS :

Microsoft IIS est le serveur web fonctionnant sous Windows Server. IIS permet de gérer une application web avec une prise en charge avancée des langages de programmation au travers des modules CGI. IIS s'installe et s'administre via le gestionnaire de serveur comme tous les rôles Windows Server.

Avantages :

-Intégration avec les produits Microsoft : IIS est étroitement intégré avec d'autres produits Microsoft tels que Windows Server, ASP.NET, et SQL Server, ce qui peut simplifier le développement et la gestion des applications web sur des infrastructures Microsoft.

-Gestion conviviale : IIS est connu pour sa convivialité en matière de gestion, avec une interface utilisateur graphique (IIS Manager) qui facilite la configuration et la surveillance du serveur.

-Sécurité : IIS propose diverses fonctionnalités de sécurité, telles que l'authentification intégrée de Windows, la gestion des certificats SSL, et des options de filtrage de demande pour renforcer la sécurité de vos sites web.

-Performances : IIS a fait des progrès en termes de performances au fil des versions, ce qui en fait une option viable pour les sites web à trafic élevé.

Inconvénients :

-Plateforme Windows uniquement : IIS est exclusivement conçu pour les systèmes Windows, ce qui limite son utilisation aux environnements Microsoft. Il n'est pas aussi portable que des serveurs web comme Apache ou Nginx.

-Coût : IIS est généralement associé à des systèmes d'exploitation Windows Server, qui sont payants, ce qui peut augmenter les coûts par rapport à des solutions open source.

-Moins populaire : Par rapport à des serveurs web open source bien établis, IIS est moins populaire, ce qui signifie qu'il peut y avoir moins de ressources communautaires et moins de tutoriels en ligne pour vous aider en cas de besoin.

-Configuration plus complexe pour certaines tâches : Bien que l'IIS Manager facilite la gestion, certaines tâches avancées ou de configuration spécifique peuvent être plus complexes dans IIS que dans des serveurs web open source.



Lighttpd :

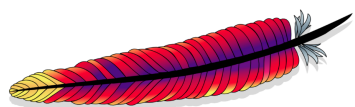
Lighttpd est un logiciel de serveur Web sécurisé, rapide et flexible. C'est un logiciel libre écrit en C et distribué selon les termes de la licence BSD. Sa rapidité vient du fait qu'il a une plus petite empreinte mémoire que d'autres serveurs HTTP ainsi qu'une gestion intelligente de la charge CPU.

Avantages :

- Léger** : Lighttpd est extrêmement léger en termes de consommation de ressources, ce qui en fait un excellent choix pour les systèmes avec des ressources limitées.
- Haute performance** : Il est capable de gérer un grand nombre de connexions simultanées, ce qui le rend adapté aux sites web à fort trafic.
- Configuration simple** : La configuration de Lighttpd est généralement considérée comme plus simple et plus lisible que celle de certains autres serveurs web, ce qui facilite l'administration.
- Rapidité de traitement des requêtes** : Lighttpd est conçu pour traiter rapidement les requêtes HTTP, ce qui en fait un choix solide pour les sites nécessitant une faible latence.

Inconvénients :

- Moins de modules tiers** : Par rapport à des serveurs web plus établis comme Apache ou Nginx, Lighttpd a moins de modules tiers disponibles, ce qui peut limiter la flexibilité en termes de personnalisation.
- Complexité pour les configurations avancées** : Bien que la configuration de base soit simple, la mise en place de configurations avancées peut être plus complexe.
- Plateforme Linux/Unix** : Lighttpd est principalement conçu pour les systèmes Linux/Unix, ce qui limite son utilisation dans les environnements Windows.



Apache Web Server

Job 4 :

Pour le Job 4 on doit mettre en place un DNS sur notre serveur Linux pour faire correspondre l'adresse IP du serveur au nom du domaine local : **dnsproject.prepa.com**

Pour configurer un serveur DNS on a besoin de l'outil **Bind9** et ses utilitaires.

Donc on installe avec la commande :

sudo apt -y install bind9 bind9utils dnsutils

Puis je change le mode d'accès réseau de la VM et je le mets en host.

Pour connaître mon IP qui a changé je rentre la commande **hostname -I**

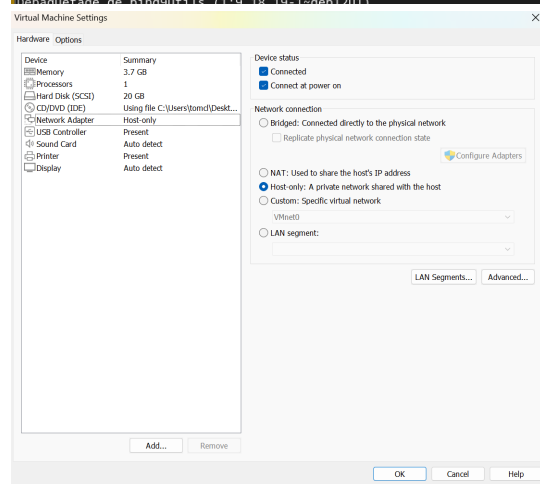
Puis on se rend dans le dossier des configs de Bind

```
tom@debian:~$ hostname -I
192.168.58.128
tom@debian:~$ cd /etc/bind
```

Une fois dedans on va modifier les fichiers de configuration pour avoir son adresse IP associée au nom du serveur dnsproject et au nom du domaine prepa.com.

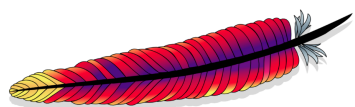
```
tom@debian:/etc/bind$ sudo nano direct
```

```
tom@debian:~$ sudo apt -y install bind9 bind9utils dnsutils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  bind9-utils
Paquets suggérés :
  bind-doc resolvconf ufw
Les NOUVEAUX paquets suivants seront installés :
  bind9 bind9-utils bind9utils dnsutils
0 mis à jour, 4 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 1 418 ko dans les archives.
Après cette opération, 2 565 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 bind9-utils amd
64 1:9.18.19-1-deb12u1 [406 kB]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 bind9 amd64 1:9
.18.19-1-deb12u1 [494 kB]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 bind9utils all
1:9.18.19-1-deb12u1 [259 kB]
Réception de :4 http://deb.debian.org/debian bookworm/main amd64 dnsutils all 1:
9.18.19-1-deb12u1 [259 kB]
1 418 ko réceptionnés en 1s (2 555 ko/s)
Sélection du paquet bind9-utils précédemment désélectionné.
(Lecture de la base de données... 158959 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../bind9-utils_1%3a9.18.19-1-deb12u1_amd64.deb ...
Dépaquetage de bind9-utils (1:9.18.19-1-deb12u1) ...
Sélection du paquet bind9 précédemment désélectionné.
Préparation du dépaquetage de .../bind9_1%3a9.18.19-1-deb12u1_amd64.deb ...
Dépaquetage de bind9 (1:9.18.19-1-deb12u1) ...
Sélection du paquet bind9utils précédemment désélectionné.
Préparation du dépaquetage de .../bind9utils_1%3a9.18.19-1-deb12u1_all.deb ...
Dépaquetage de bind9utils (1:9.18.19-1-deb12u1)
```



```
GNU nano 7.2                                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       192.168.58.128
www       IN      CNAME    dnsproject.prepa.com.

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacem
^X Quitter   ^R Lire fich ^M Remplacer ^U Coller    ^J Justifier ^_ Aller li
```



Apache Web Server

Ensuite on va copier le contenu du fichier direct dans le fichier inverse avec la commande **cp direct inverse** puis on édite le fichier inverse.

```
tom@debian:/etc/bind$ sudo cp direct inverse
tom@debian:/etc/bind$ sudo nano inverse
```

```
GNU nano 7.2 inverse
; BIND data file for local loopback interface
$TTL 604800
@ IN SOA prepacom. dnsproject.prepa.com. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS dnsproject.prepa.com.
dnsproject IN A 192.168.58.128
213 IN PTR dnsproject.prepa.com.
```

[Lecture de 14 lignes]

^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller ^J Justifier ^_ Aller ligne

Et on va éditer le fichier local :

```
tom@debian:/etc/bind$ sudo nano named.conf.local
```

```
GNU nano 7.2 named.conf.local
zone "prepacom" IN {
    type master;
    file "/etc/bind/direct";
};
zone "29.10.10.in-addr-arpa" IN {
    type master;
    file "/etc/bind/inverse";
};
```

[Lecture de 8 lignes]

^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller ^J Justifier ^_ Aller ligne

Puis pour finir on va associer le nom de domaine à l'adresse IP.

```
tom@debian:/etc/bind$ sudo nano /etc/resolv.conf
```

```
GNU nano 7.2 /etc/resolv.conf
# Generated by NetworkManager
search prepacom
nameserver 192.168.58.128
```

[Lecture de 3 lignes]

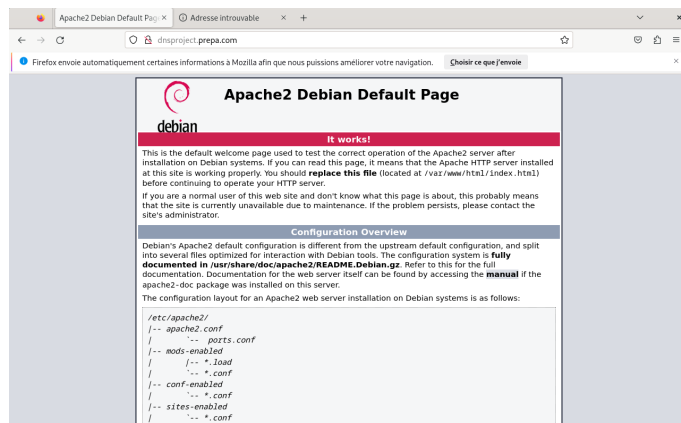
^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller ^J Justifier ^_ Aller ligne

Il nous reste plus qu'à restart le bind pour appliquer les changements et on peut faire le ping.

```
tom@debian:/etc/bind$ systemctl restart bind9
tom@debian:/etc/bind$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (192.168.58.128) 56(84) bytes of data:
64 bytes from debian (192.168.58.128): icmp_seq=1 ttl=64 time=3.02 ms
64 bytes from debian (192.168.58.128): icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from debian (192.168.58.128): icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from debian (192.168.58.128): icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from debian (192.168.58.128): icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from debian (192.168.58.128): icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from debian (192.168.58.128): icmp_seq=7 ttl=64 time=0.057 ms
64 bytes from debian (192.168.58.128): icmp_seq=8 ttl=64 time=0.041 ms
```



Puis on entre le nom de domaine dans l'url d'un navigateur sur la VM.



Job 5 :

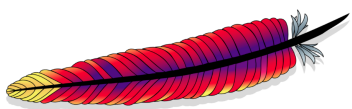
Comment obtient-on un nom de domaine public ?

Pour obtenir un nom de domaine public il faut choisir un bureau d'enregistrement(registrars) de nom de domaine, vérifier si ce nom est toujours disponible, choisir son extension appropriée, acheter le nom de domaine et configurer les serveurs de nom DNS.
Par la suite, il faudra protéger le nom de domaine.

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

Il existe plusieurs types d'extensions qui ont chacune leurs particularités.

Extensions	Exemples nom de Domaines
Génériques gTLDs	.com / .net / .org / .info
De pays ccTLDs	.fr (France) .de (Allemagne) .ru (Russie)
Thématiques nTLDs	.io / .app / .blog

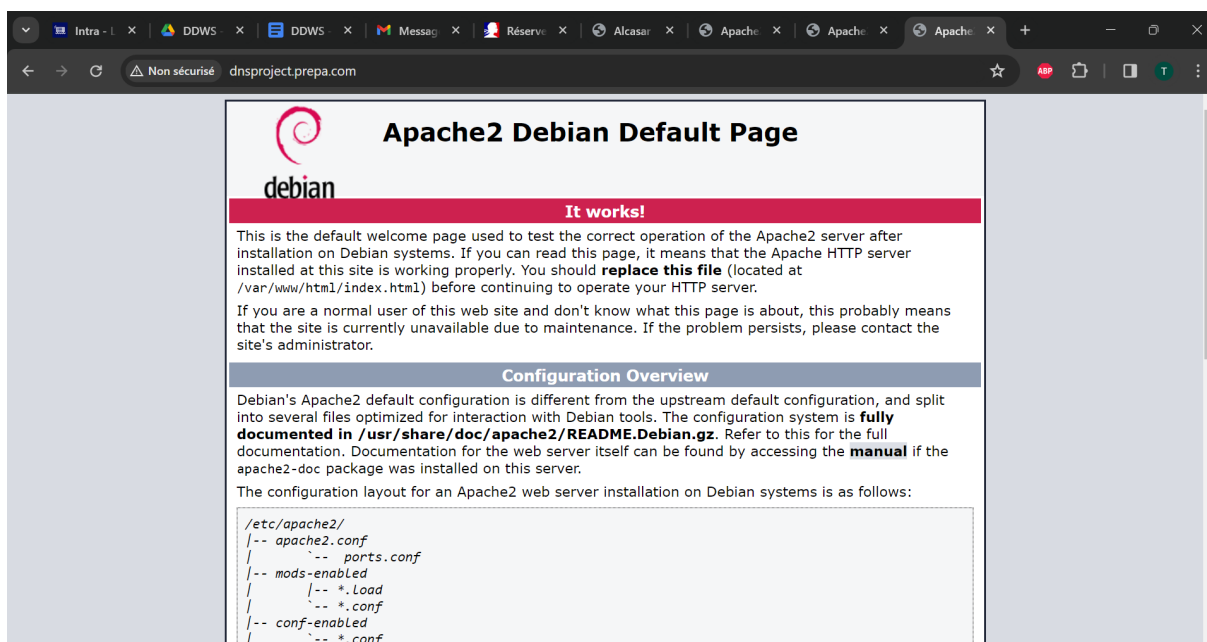
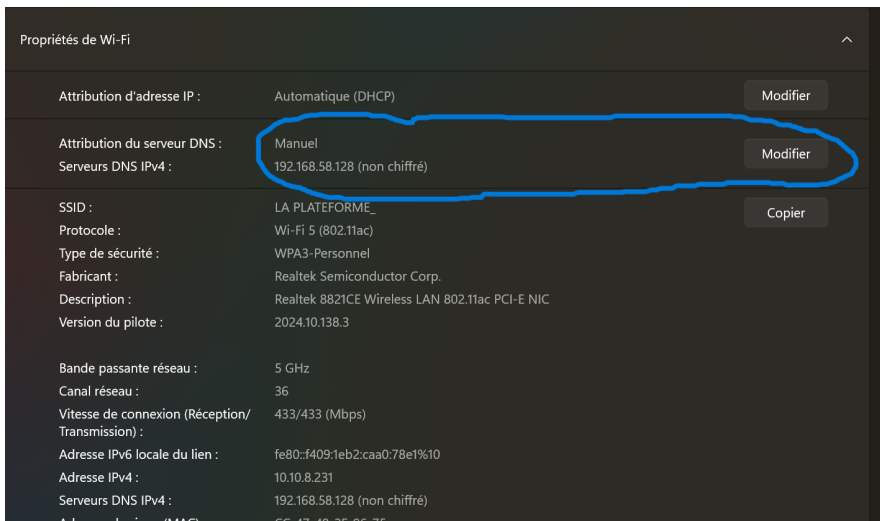
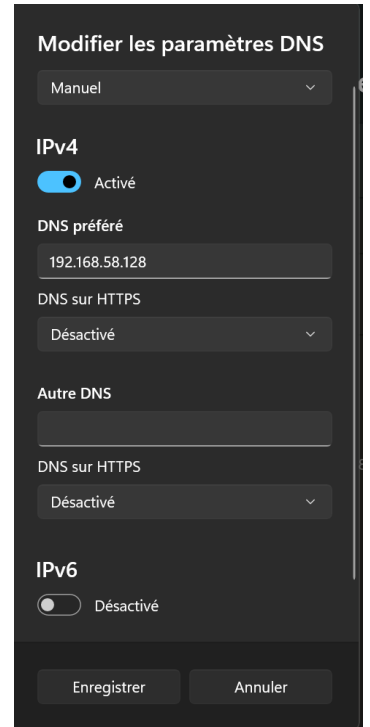
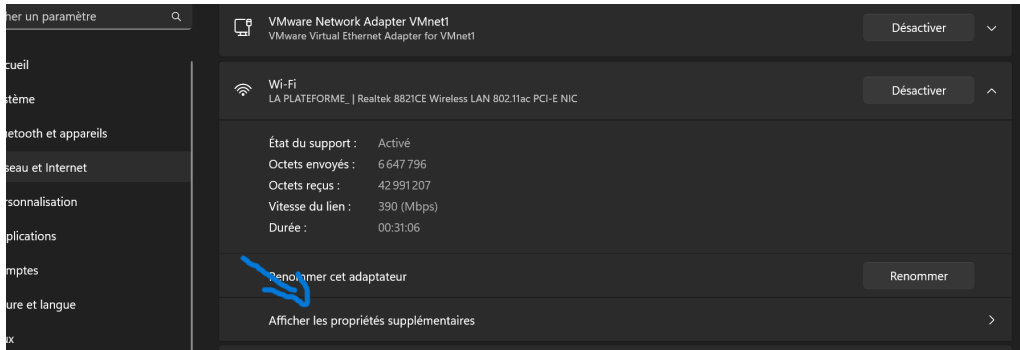


Apache Web Server

Job 6 :

Pour que l'hôte soit connecté au nom de domaine local du serveur il faut changer ses paramètres de DNS dans notre machine hôte.

On change l'attribution du serveur DNS en manuel et on entre l'ip du serveur.



Job 7 :

Pour le Job 7 on doit installer un pare-feu en utilisant **ufw** sur le serveur principale pour que notre hôte accède à apache sans pouvoir le ping.

On commence par télécharger **ufw** avec la commande **sudo apt-get install ufw**

```
tom@debian:~$ sudo apt-get install ufw
[sudo] Mot de passe de tom :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  iptables libip6tc2
Paquets suggérés :
  firewallld rsyslog
Les NOUVEAUX paquets suivants seront installés :
  iptables libip6tc2 ufw
0 mis à jour, 3 nouvellement installés, 0 à enlever et 6 non mis à jour.
Il est nécessaire de prendre 548 ko dans les archives.
Après cette opération, 3 411 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 libip6tc2 amd64
1.8.9-2 [19,4 kB]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 iptables amd64
1.8.9-2 [360 kB]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 ufw all 0.36.2-
1 [168 kB]
548 ko réceptionnés en 0s (3 221 ko/s)
Préconfiguration des paquets...
Sélection du paquet libip6tc2:amd64 précédemment désélectionné.
```

Puis on entre la commande **ufw default deny incoming** pour rejeter les connexions qui entrent sauf celles qu'on va autoriser.

```
tom@debian:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
```

On fait la même commande pour les connexions sortantes : **ufw default deny outgoing**

```
tom@debian:~$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
```

Puis on va ouvrir certains ports avec la commande : **/tcp**
C'est pour autoriser le trafic TCP.

```
tom@debian:~$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
```

```
tom@debian:~$ sudo ufw allow 139/tcp
[sudo] Mot de passe de tom :
Rules updated
Rules updated (v6)
```

```
tom@debian:~$ sudo ufw allow 445/tcp
Rules updated
Rules updated (v6)
```

On va modifier le fichier le before.rules en remplaçant à la ligne **# ok icmp codes** for INPUT le “ACCEPT” par “DROP”

```
tom@debian: ~  
GNU nano 7.2 /etc/ufw/before.rules *  
  
# allow all on loopback  
-A ufw-before-input -i lo -j ACCEPT  
-A ufw-before-output -o lo -j ACCEPT  
  
# quickly process packets for which we already have a connection  
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  
  
# drop INVALID packets (logs these in loglevel medium and higher)  
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny  
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP  
  
# ok icmp codes for INPUT  
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP  
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP  
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP  
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Il nous reste plus qu'à activer le pare-feu avec la commande **ufw enable**.

```
tom@debian:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
tom@debian:~$
```

Job 8 :

On commence par télécharger samba avec la commande **sudo apt-get install samba**.

```
tom@debian:~$ sudo apt-get install samba  
[sudo] Mot de passe de tom :  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  attr ibverbs-providers libcephfs2 libfmt9 libgfpapi0 libgfrpc0 libgfxdr0  
  libglusterfs0 libibverbs1 librados2 librdmacm1 liburing2 python3-anyio  
  python3-click python3-colorama python3-dnspython python3-gpg python3-h11  
  python3-h2 python3-hpack python3-httpcore python3-httpx python3-hyperframe  
  python3-ldb python3-markdown python3-markdown-it python3-mdurl  
  python3-pygments python3-requests-toolbelt python3-rfc3986 python3-rich  
  python3-samba python3-sniffio python3-talloc python3-tdb python3-yaml  
  samba-ad-provision samba-common samba-common-bin samba-dsdb-modules  
  samba-vfs-modules tdb-tools  
Paquets suggérés :  
  python3-trio python3-aioquic python-markdown-doc python-pygments-doc  
  ttf-bitstream-vera ctdb ldb-tools ntp | chrony winbind heimdal-clients  
Les NOUVEAUX paquets suivants seront installés :  
  attr ibverbs-providers libcephfs2 libfmt9 libgfpapi0 libgfrpc0 libgfxdr0  
  libglusterfs0 libibverbs1 librados2 librdmacm1 liburing2 python3-anyio  
  python3-click python3-colorama python3-dnspython python3-gpg python3-h11  
  python3-h2 python3-hpack python3-httpcore python3-httpx python3-hyperframe  
  python3-ldb python3-markdown python3-markdown-it python3-mdurl
```



Apache Web Server

On va créer notre dossier partagé.

```
tom@debian:~$ sudo mkdir /home/partage
tom@debian:~$
```

Par la suite on modifie la configuration de samba

```
[partage]
comment = partage
path = /home/tom/partage
valid users = @users
force group = users
create mask = 0600
directory mask = 0771
writable = yes
```

Ensuite on crée un utilisateur samba

```
tom@debian:~$ sudo smbpasswd -a tom
New SMB password:
Retype new SMB password:
Added user tom.
tom@debian:~$
```

```
tom@debian:~$ sudo chmod -R 777 /home/partage
tom@debian:~$
```

```
tom@debian:~$ sudo service smbd restart
```

