

DDWS

Job 01

- Pour commencer à configurer SSH, on va vérifier si le serveur SSH est déjà installé. La commande **"dpkg -l"** liste les paquets installés sur Debian. En utilisant **"grep openssh-server"**, on recherche spécifiquement le paquet "openssh-server" parmi la liste. Si le paquet est installé, il s'affiche dans les résultats, vérifiant ainsi la présence du service SSH sur le système.
- Ensuite vu que le serveur SSH ne sera pas installé, on va l'installer en faisant comme commande : **"sudo apt update"** et **"sudo apt install openssh-server"**, **"sudo apt update"** met à jour la liste des paquets disponibles, s'assurant qu'il est à jour. **"sudo apt install openssh-server"** installe le serveur SSH (openssh-server) sur le système Debian, permettant ainsi la configuration du service SSH pour les connexions distantes sécurisées.
- En dernier lieu, nous allons utiliser la commande **"ssh azerty@10.0.2.15-p 22"**, cette commande **"ssh azerty@10.0.2.15 -p 22"** est utilisée pour se connecter à un serveur distant via SSH. On spécifie l'utilisateur, l'adresse IP du serveur, et éventuellement le port ("22" dans ce cas-là). Cette commande permet d'établir une connexion sécurisée et nécessite une authentification avant d'accéder au serveur distant.

Port 22

#AddressFamily any

#ListenAddress 0.0.0.0

#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key

#HostKey /etc/ssh/ssh_host_ecdsa_key

#HostKey /etc/ssh/ssh_host_ed25519_key

Ciphers and keying

#RekeyLimit default none

Logging

#SyslogFacility AUTH

#LogLevel INFO

Authentication:

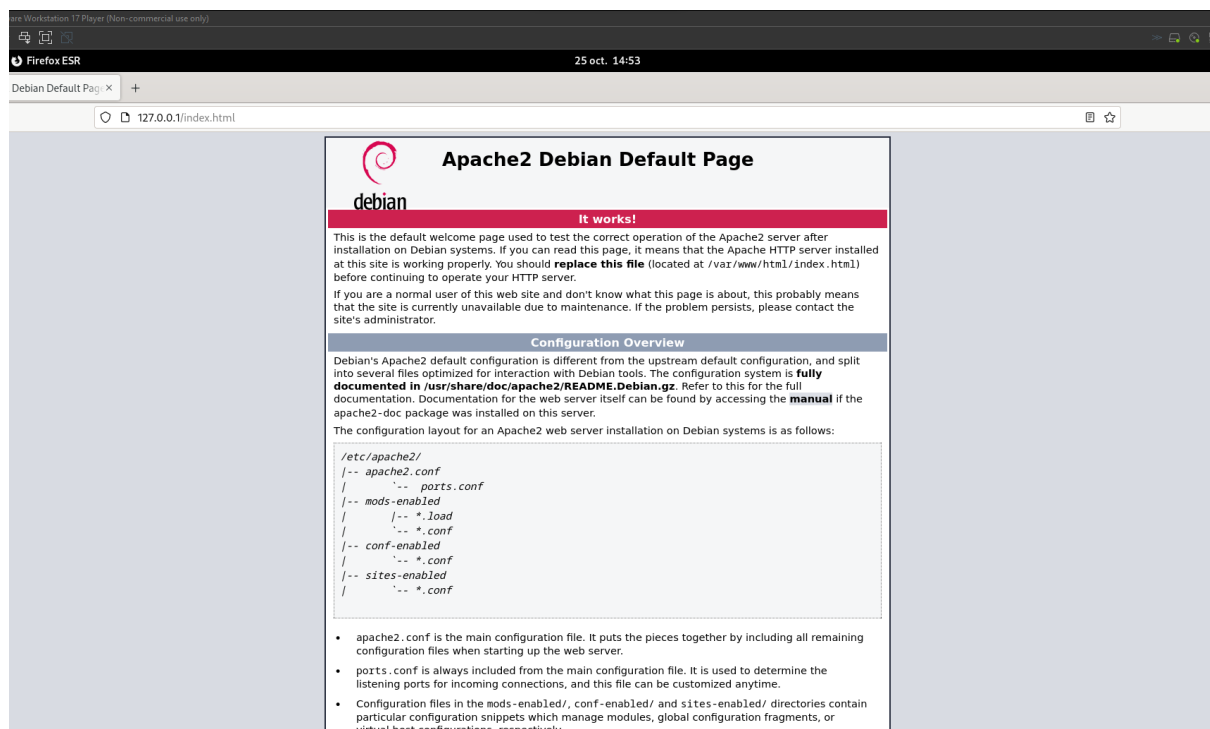
#LoginGraceTime 2m

PermitRootLogin yes

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

Job 02

- Pour commencer nous allons installer Apache grâce à la commande suivante : **"sudo apt install apache2"**, la commande "sudo apt install apache2" installe le serveur web Apache2 sur le système Linux basé sur Debian, tel qu'ubuntu. Apache2 permet d'héberger des sites web et de fournir des services web. Le préfixe "sudo" est utilisé pour exécuter la commande en tant qu'administrateur, car l'installation de logiciels nécessite des privilèges élevés.
- Ensuite nous allons utiliser la commande : **"sudo systemctl status apache2"** pour vérifier l'état du service Apache. La commande "sudo systemctl status apache2" est utilisée pour vérifier l'état du service Apache2 sur un système Linux. Il affiche des informations sur l'exécution, les erreurs éventuelles et l'état du service. Cette commande est utile pour le dépannage et la surveillance du serveur web Apache2.
- Nous allons maintenant aller dans la destination : **"/var/www/html"**, qui est l'emplacement par défaut où les fichiers et les pages web d'un serveur web, tel qu'Apache, sont stockés sous Linux.
- Une fois dans la destination **"/var/www/html"** nous allons taper **"firefox 127.0.0.1/index.html"** qui nous ouvrira la page de lancement d'Apache sur firefox.



Job 03

Renseignez-vous sur les différents serveurs Web existants.

Il existe plusieurs serveurs web populaires, chacun avec ses avantages et inconvénients.

Apache HTTP Server, par exemple, est renommé pour sa modularité et sa grande communauté de soutien, mais il peut parfois **consommer beaucoup de ressources et sa configuration peut sembler complexe pour les débutants**.

D'un autre côté, **Nginx** est salué pour ses performances exceptionnelles et sa gestion efficace de nombreuses connexions simultanées. Cependant, **il peut être moins adapté aux applications nécessitant un traitement côté serveur complexe**.

LiteSpeed est un serveur performant, offrant un support de **LiteSpeed Cache et une interface web conviviale**, mais sa version gratuite a des limitations et des fonctionnalités réduites.

Caddy se distingue par sa facilité d'utilisation, **sa configuration automatique, et le support intégré de HTTP/2 et Let's Encrypt pour le chiffrement HTTPS**. Cependant, il offre moins de modules tiers que certains concurrents.

Microsoft Internet Information Services (IIS) est intégré à Windows, ce qui en fait un choix naturel pour les environnements Windows et **prend en charge diverses technologies Microsoft**. Néanmoins, il peut être moins performant que certains serveurs open source et moins couramment utilisé en dehors de l'écosystème Windows.

Cherokee propose une interface conviviale et une **gamme de fonctionnalités avancées**, en plus de performances solides pour les sites web dynamiques. Cependant, il est moins populaire que d'autres alternatives, ce qui signifie une communauté plus restreinte.

Enfin, **lighttpd** (Lighty) est un serveur **léger et rapide**, idéal pour servir du contenu statique, bien qu'il puisse manquer de flexibilité pour des scénarios plus complexes.

Le choix du serveur web dépend de divers facteurs, notamment des besoins spécifiques, il est essentiel de prendre en compte des considérations telles que **la performance, la sécurité, la facilité de configuration, et la disponibilité des fonctionnalités lors de la sélection du serveur web** le mieux adapté à l'application.

Job 04

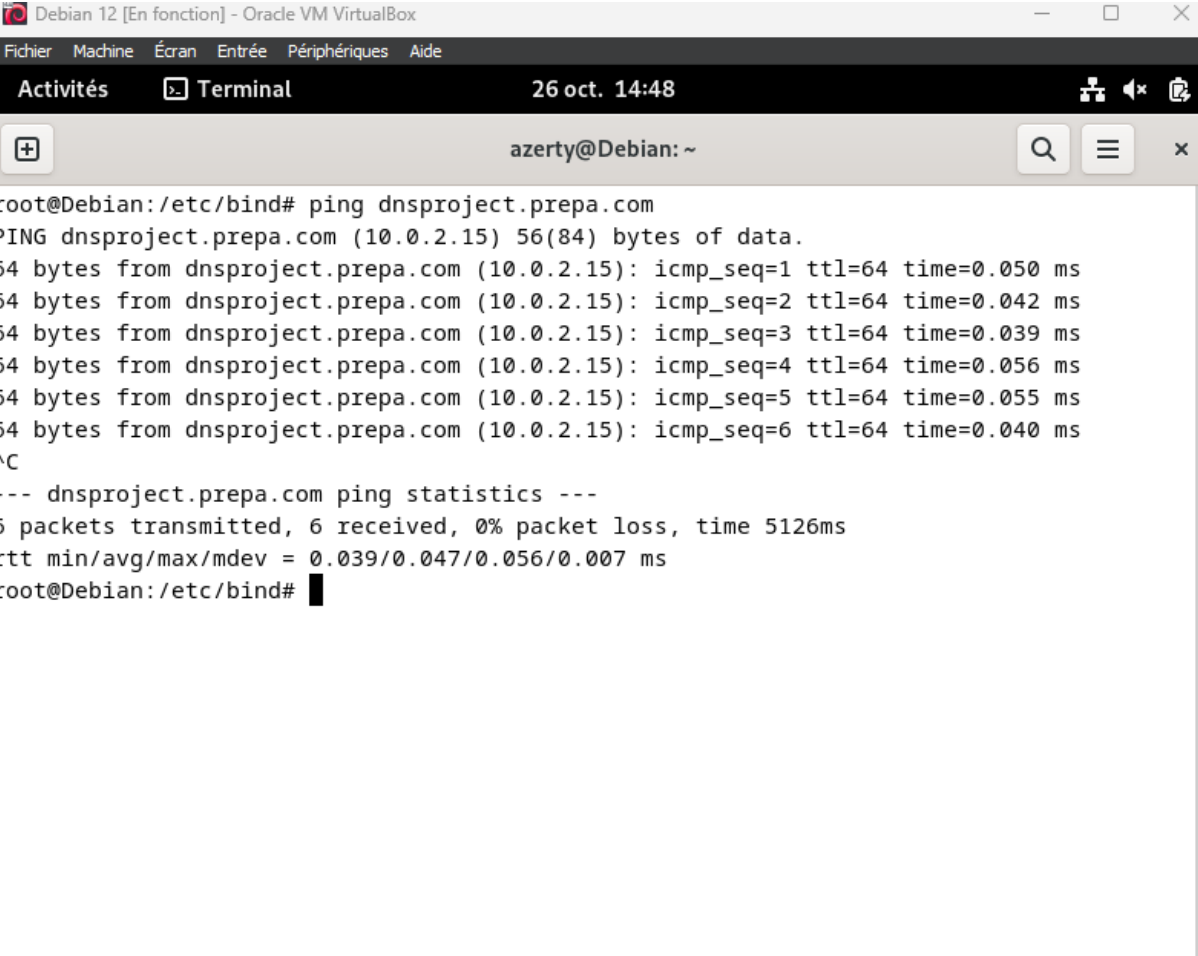
- En premier temps, nous allons faire un **"sudo apt update"** pour mettre à jour la liste des paquets disponibles dans les dépôts de logiciels du système Linux.
- Ensuite, nous allons installer Bind9, pour ce faire exécutons la commande : **"sudo apt install bind9"**, qui sert à installer Bind9 sur Linux.
- Ensuite, on va taper dans le terminal : **"sudo nano /etc/bind/named.conf.options"**, cette commande permet de configurer un forwarder, elle permet donc d'ouvrir le fichier de configuration des options du serveur DNS BIND pour des modifications en tant qu'administrateur.
- Après, nous allons rentrer dans cette commande ce script :
- **"forwarders {**
 10.0.2.15;
 8.8.8.8;
 };"
- Le bloc "forwarders" dans la configuration DNS BIND spécifie les serveurs DNS à consulter en cas d'incapacité à résoudre localement une requête DNS. Dans notre exemple, on utilise d'abord **10.0.2.15**, un serveur local, puis **8.8.8.8**, le serveur DNS public de Google, pour obtenir la réponse. Cela améliore la résolution DNS en utilisant ces serveurs comme relais.
- Ensuite, nous allons utiliser dans le terminal : **"nano /etc/bind/named.conf.local"**, cette commande est utilisée pour éditer le fichier de configuration local de bind, un serveur DNS. Ce fichier contient des directives pour configurer les zones de domaines locaux et les serveurs DNS autoritaires. En l'éditant, on peut définir des paramètres spécifiques à notre domaine, tels que les enregistrements de zone et les autorisations de transfert de zone.
- Une fois rentrés dans le fichier nano, nous allons inscrire :
- **"zone "local.lan" IN {**
 type master;
 file "/etc/bind/dnsproject.prepa.com";
 };"

Le bloc de configuration "zone" indique au serveur DNS BIND de gérer la zone **"local.lan"** en tant que zone maître. Les informations de cette zone, telles que les enregistrements DNS, sont stockées dans le fichier **"/etc/bind/dnsproject.prepa.com"**. Le serveur BIND est responsable de résoudre les noms de domaines de cette zone à l'aide de ce fichier.

- Ensuite nous allons taper dans le terminal : **"sudo cp /etc/bind/db.local /etc/bind/dnsproject.prepa.com"**, cette commande copie un modèle de configuration de zone DNS (**"db.local"**) vers un fichier (**"dnsproject.prepa.com"**) que l'on peut personnaliser pour configurer la zone DNS du domaine **"dnsproject.prepa.com"**.
- Ensuite, on va également taper dans le terminal : **"nano /etc/bind/dnsproject.prepa.com"**, cette commande sert à éditer le fichier de

configuration de zone pour le domaine "**dnsproject.prepa.com**" et à personnaliser la résolution des noms de domaine pour le domaine inscrit précédemment (**dnsproject.prepa.com**) ainsi que les enregistrements DNS de ce domaine.

- Avant d'appliquer les modifications, il nous reste à utiliser cette commande sur le terminal : "**nano /etc/bind/named.conf.options**" cette commande permet de changer à l'intérieur "**any;**" en "**::1;**".
- Il nous manque donc à redémarrer Bind en tapant : "**sudo service bind9 restart**" dans le terminal.
- Pour tester et voir si ça marche, il nous reste plus qu'à taper "**ping dnsproject.prepa.com**" sur le terminal.



The screenshot shows a terminal window titled "Debian 12 [En fonction] - Oracle VM VirtualBox". The window has a menu bar with "Fichier", "Machine", "Écran", "Entrée", "Périphériques", and "Aide". Below the menu bar is a toolbar with "Activités", "Terminal", and a clock showing "26 oct. 14:48". The terminal prompt is "azerty@Debian: ~". The user has entered the command "ping dnsproject.prepa.com". The output shows six successful ping requests with varying times, followed by a summary of the statistics.

```
root@Debian:/etc/bind# ping dnsproject.prepa.com
PING dnsproject.prepa.com (10.0.2.15) 56(84) bytes of data:
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from dnsproject.prepa.com (10.0.2.15): icmp_seq=6 ttl=64 time=0.040 ms
^C
--- dnsproject.prepa.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5126ms
rtt min/avg/max/mdev = 0.039/0.047/0.056/0.007 ms
root@Debian:/etc/bind#
```

Job 05

- Faites des recherches sur comment obtient-on un nom de domaine public ?

Pour obtenir un nom de domaine public, vous devez choisir un registraire de domaine accrédité, comme **GoDaddy** ou **Namecheap**. Ensuite, on vérifie la disponibilité du nom de domaine qu'on souhaite et on sélectionne l'extension de domaine appropriée, comme **.com**, **.org**, ou d'autres. On enregistre le nom de domaine en fournissant nos informations et en payant les frais d'enregistrement. Une fois enregistré, on configure les enregistrements DNS pour **diriger le nom de domaine vers notre site web ou nos services**. On s'assure de gérer notre nom de domaine **via le tableau de bord du registraire**, y compris le renouvellement annuel. Il faut respecter les politiques et règles d'enregistrement pour maintenir la propriété du nom de domaine. Une fois obtenu, notre nom de domaine peut être utilisé pour créer un site web, des adresses e-mail personnalisées et bien plus encore.

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

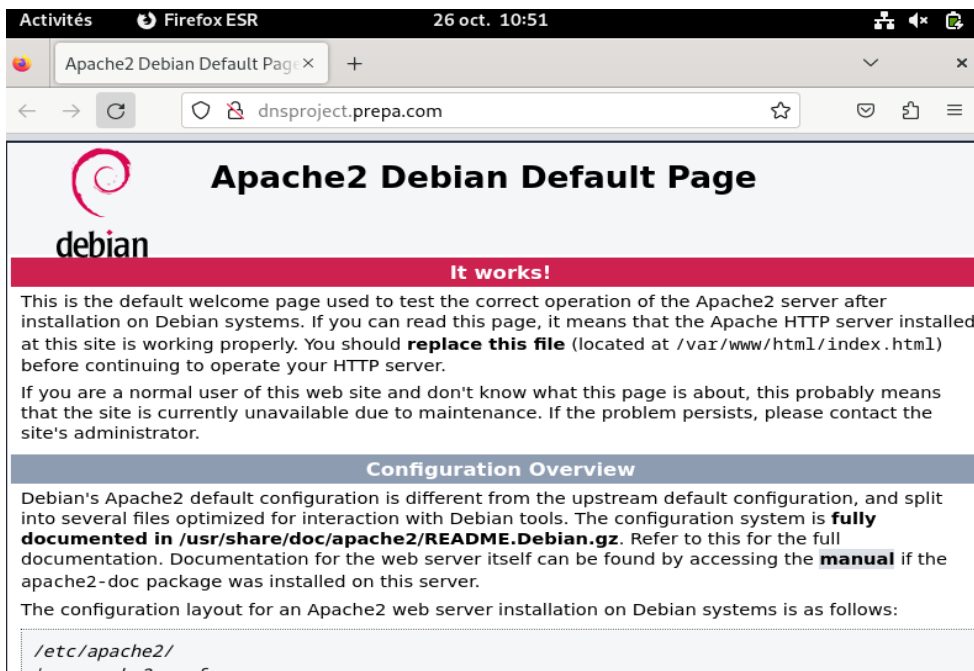
Les extensions de nom de domaine (**TLD**) présentent des spécificités en fonction de leur objectif d'utilisation. Par exemple, le populaire **".com"** (.com = dans le monde entier) est **polyvalent, sans restrictions majeures**, tandis que **".org"** est traditionnellement associé **aux organisations à but non lucratif**. Des extensions spécifiques existent, telles que **".gov"** pour les **entités gouvernementales aux États-Unis**, ou **".edu"** pour les **institutions éducatives**. D'autres extensions, comme **".io"**, sont appréciées par **les startups technologiques**. Certaines, comme **".fr"**, sont des **extensions nationales soumises à des exigences géographiques**. Chaque **TLD** a ses **propres règles**, adaptées à des fins particulières, et il est essentiel de les comprendre avant d'enregistrer un nom de domaine.

Job 06

- Pour commencer nous allons taper dans le terminal : "**sudo nano /etc/apache2/sites-available/dnsproject.conf**"
- Ensuite à l'intérieur de nano on tape : "<**VirtualHost *:80**>
ServerName dnsproject.prepa.com
DocumentRoot /var/www/html
</VirtualHost>"

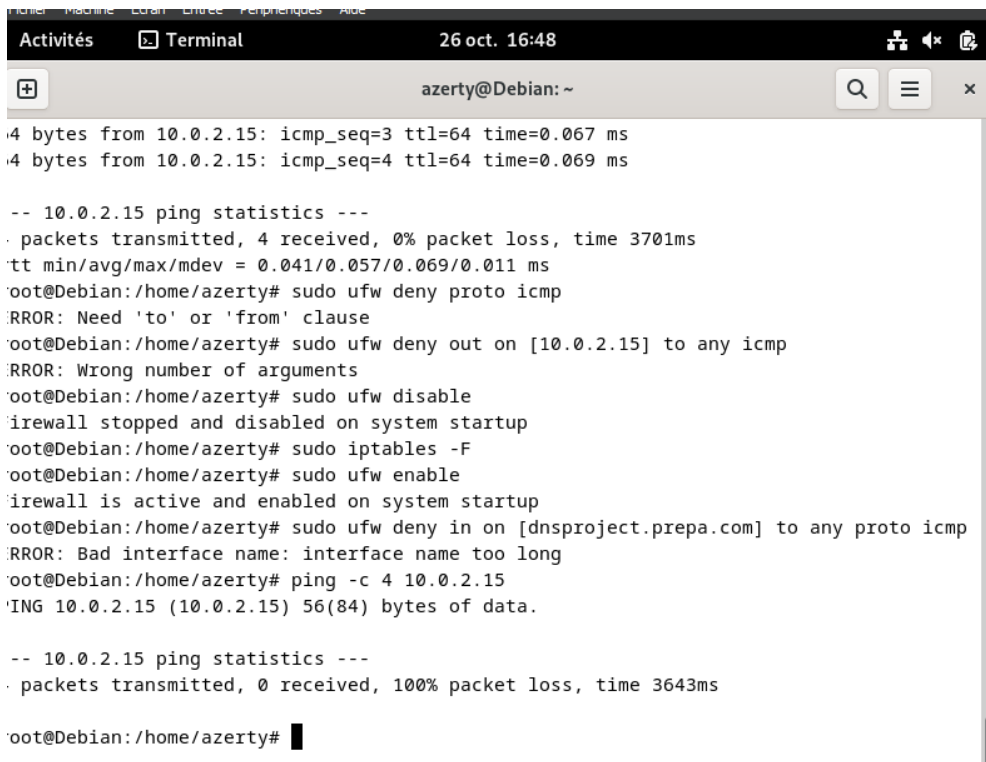
Ensuite on ferme nano en enregistrant

- On active notre Apache, pour ce faire on tape dans le terminal : "**sudo a2ensite dnsproject.conf**", cela permet d'activer Apache.
- On continue en redémarrant Apache en utilisant la commande : "**sudo systemctl restart apache2**".
- Puis pour finir on ajoute une entrée DNS locale en faisant comme commande dans le terminal : "**sudo nano /etc/hosts**".
- Et pour finir on ajoute dans nano : "**10.0.2.15 dnsproject.prepa.com**", on quitte ensuite en enregistrant.
- Et pour tester on tape dans la barre de recherche internet : "**http://dnsproject.prepa.com**" et on devrait avoir la page Apache qui s'ouvre.

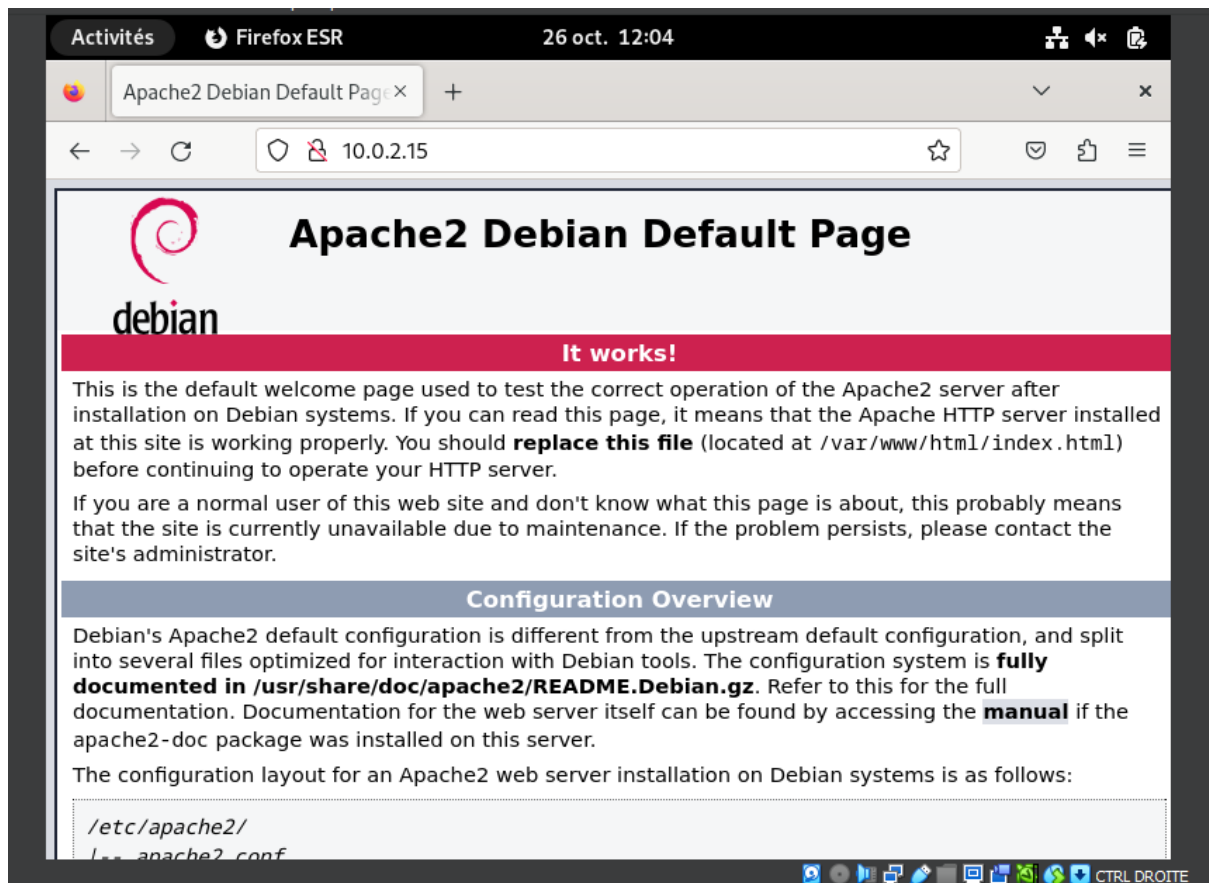


Job 07

- Pour commencer, on utilise **"sudo apt-get update"** pour mettre à jour la liste des paquets disponibles.
- Ensuite on utilise **"sudo apt-get install ufw"** pour installer UFW.
- Après on commence à configurer ufw, pour ce faire on tape **"sudo ufw allow 80/tcp"** qui permet d'autoriser le trafic HTTP (port 80).
- Ensuite on bloque le trafic ICMP (ping) en entrant dans le terminal : **"sudo ufw deny proto icmp"**, sauf qui se peut que cette commande ne marche pas, il faudra alors taper dans le terminal : **"sudo ufw deny 5353/udp"**.
- Il nous manque plus qu'à taper : **"sudo ufw enable"** dans le terminal pour activer UFW.
- En dernier on va tester la configuration UFW pour s'assurer que les règles ont été appliquées en tapant dans le terminal : **"sudo ufw status"**.
- Pour voir si tout ce qu'on a fait marche, on va taper dans le terminal : **"ping 10.0.2.15"** ou (et) encore **"ping dnsproject.prepa.com"** si le ping ne marche pas c'est que c'est bon, on va également voir si la page Apache s'affiche sur internet en rentrant sur internet : **"http://10.0.2.15"** ou (et) encore **"http://dnsproject.prepa.com"**.
- Il se peut que le ping marche toujours, pour régler ce problème-là, il suffit de taper la commande suivante : **"sudo iptables -A INPUT -p icmp-icmp-type echo-request -j DROP"** la commande consiste à bloquer les requêtes de ping entrantes (ICMP Echo Requests) vers un serveur en utilisant IPTables, renforçant ainsi la sécurité, mais peut perturber les outils de diagnostic réseau.



```
azerty@Debian: ~  
4 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.067 ms  
4 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.069 ms  
  
-- 10.0.2.15 ping statistics --  
4 packets transmitted, 4 received, 0% packet loss, time 3701ms  
rtt min/avg/max/mdev = 0.041/0.057/0.069/0.011 ms  
oot@Debian:/home/azerty# sudo ufw deny proto icmp  
RROR: Need 'to' or 'from' clause  
oot@Debian:/home/azerty# sudo ufw deny out on [10.0.2.15] to any icmp  
RROR: Wrong number of arguments  
oot@Debian:/home/azerty# sudo ufw disable  
firewall stopped and disabled on system startup  
oot@Debian:/home/azerty# sudo iptables -F  
oot@Debian:/home/azerty# sudo ufw enable  
firewall is active and enabled on system startup  
oot@Debian:/home/azerty# sudo ufw deny in on [dnsproject.prepa.com] to any proto icmp  
RROR: Bad interface name: interface name too long  
oot@Debian:/home/azerty# ping -c 4 10.0.2.15  
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.  
  
-- 10.0.2.15 ping statistics --  
4 packets transmitted, 0 received, 100% packet loss, time 3643ms  
  
oot@Debian:/home/azerty#
```



Job 08

- Pour commencer, on doit installer **Samba**, en premier on fait : **"sudo apt-get update"**, ensuite on inscrit dans le terminal : **"sudo apt-get install samba"** cela permettra à Samba de s'installer.
- Ensuite on crée un dossier dans notre répertoire personnel, que l'on nommera **"Partage"**, on lui attribuera les permissions de **lecture/écriture** à tous les utilisateurs en faisant la commande suivante dans le terminal : **"chmod -R 777 ~/Partage"**.
- Ensuite on va commencer à configurer Samba en faisant la commande suivante dans le terminal : **"sudo nano /etc/samba/smb.conf"**.
- Ensuite dans la commande nano, on ajoute tout à la fin cette commande :
"[Partage]
comment = Dossier partagé
path = /home/azerty/Partage
browseable = yes
read only = no
create mask = 0777
directory mask = 0777", ensuite on oublie pas d'enregistrer avant de quitter.
- Ensuite on doit créer un utilisateur Samba qui aura accès au dossier partagé, pour ce faire nous allons taper dans le terminal : **"sudo smbpasswd -a azerty"**, ils vont juste après nous demander de **créer un mot de passe et de le confirmer**.
- Ensuite on redémarre Samba pour appliquer les modifications apportées en tapant dans le terminal : **"sudo service smbd restart"**.
- On peut maintenant essayer de voir si ce qu'on a fait marche, en tapant en premier dans le terminal : **"sudo service smbd status"**, si un message indiquant que Samba est actif (running) apparaît, c'est que ce qu'on a fait marche, ensuite on va dans nos **fichiers sur Linux > Autres emplacements > saisir l'adresse du serveur**, à ce moment-là, on rentrera : **"smb://10.0.2.15/"**, si le dossier **"Partage"** s'affiche c'est que tout marche normalement.

```
chier Machine Écran Entrée Périphériques Aide
26 oct. 12:32
azerty@Debian: ~
azerty@Debian: ~
Added user azerty.
root@Debian:~# sudo service smbda restart
root@Debian:~# sudo service smbda status
• smbda.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbda.service; enabled; preset: enabled)
  Active: active (running) since Thu 2023-10-26 12:24:59 CEST; 2min 17s ago
    Docs: man:smbda(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 8565 ExecCondition=/usr/share/samba/is-configured smb (code=exited, statu>
  Process: 8568 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=ex>
 Main PID: 8577 (smbda)
  Status: "smbda: ready to serve connections..."
    Tasks: 3 (limit: 4622)
  Memory: 5.4M
    CPU: 119ms
  CGroup: /system.slice/smbda.service
          └─8577 /usr/sbin/smbda --foreground --no-process-group
            └─8579 /usr/sbin/smbda --foreground --no-process-group
              └─8580 /usr/sbin/smbda --foreground --no-process-group

oct. 26 12:24:59 Debian systemd[1]: Starting smbda.service - Samba SMB Daemon...
oct. 26 12:24:59 Debian systemd[1]: Started smbda.service - Samba SMB Daemon.
```

