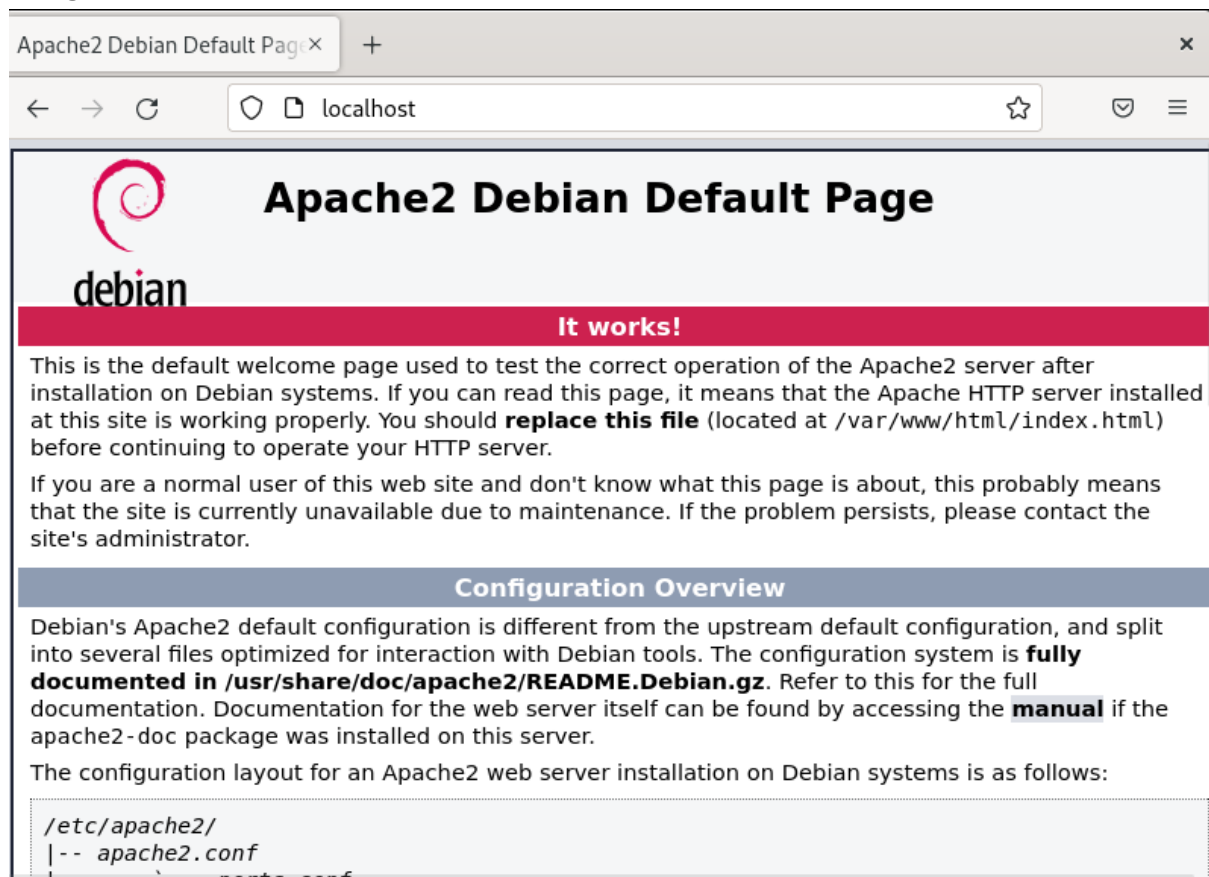


### Job 2)

On commence d'abord par installer Apache 2 avec la commande `sudo apt install apache2`

Pour vérifier que le serveur web Apache 2 fonctionne correctement, on utilise un navigateur web pour se connecter dessus via l'url localhost.



### Job 3)

**Apache HTTP Server:** disponible sur Windows Mac OS et Linux, l'un des serveur le plus utiliser pour les serveurs HTTP, il est open source et facile à installer ainsi que permet une modification de la configuration rapide, mais il supporte mal la charge sur les serveurs à gros trafic.

**NGINX:** disponible sur système Unix (Linux MacOS) et Windows, il est le second serveur le plus utilisé, il est open source et permet un meilleur traitement pour les serveurs avec des configurations fixe, mais moins pour les configurations dynamique.

#### Job 4)

On commence par installer bind9 permettant de créer un serveur DNS avec `sudo apt install bind9*`

On commence d'abord par définir l'emplacement de notre fichier de configuration du nom de domaine `dnsproject.prepa.com` dans le fichier `named.conf.local`

Pour lier une page web à un nom de domaine, il nous faut récupérer l'IP du serveur web

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

Pour lier une page web à un nom de domaine, il nous faut récupérer l'IP du serveur web

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:19:00:11 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute
```

Nous utiliserons donc 192.168.56.101 pour nos entrées DNS.

Voici notre entrée DNS pour le nom de domaine `dnsproject.prepa.com`

On ajoute donc une entrée A pour l'IPv4 et une IPv6 localhost avec AAAA

```
GNU nano 5.4 /etc/bind/db.dnsproject.prepa.com
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      dnsproject.prepa.com. dns@prepa.com. (
                                2           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS       dnsproject.prepa.com.
@         IN      A        192.168.56.101
@         IN      AAAA     ::1
ns        IN      A        192.168.56.101
```

On édite notre configuration réseau afin que le seul serveur DNS utilisé soit le nôtre, comme notre configuration est hébergée sur notre PC, nous pouvons utiliser notre adresse localhost.

Annuler

Filaire

Appliquer

Détails
Identité
IPv4
IPv6
Sécurité

**Méthode IPv4**
☒ Automatique (DHCP)
☐ Réseau local seulement
☐ Manuel
☐ Désactiver
☐ Partagée avec d'autres ordinateurs

**DNS**

Automatique ☐

127.0.0.1

Séparer les adresses IP avec des virgules

```

debian@debian:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from localhost (::1): icmp_seq=5 ttl=64 time=0.065 ms

```

Job 5)

Job 6)

Pour ce faire on doit définir l'ip de notre DNS sur un autre ordinateur.

Voici la configuration réseau de l'ordinateur qui interrogera le DNS

```

Vitesse de la connexion  1000 Mb/s
Adresse IPv4  192.168.56.111
Adresse IPv6  fe80::456:1f76:c869:81a4
Adresse matérielle  08:00:27:B9:00:D6
Route par défaut  192.168.56.101
DNS  192.168.56.101

```

Et voici la configuration de notre serveur DNS

**Méthode IPv4** ☐ Automatique (DHCP) ☐ Réseau local seulement  
☒ Manuel ☐ Désactiver  
☐ Partagée avec d'autres ordinateurs

#### Adresses

Adresse	Masque de réseau	Passerelle	
192.168.56.101	255.255.255.0		✕
			✕

#### DNS

Automatique ☐

192.168.56.101

Séparer les adresses IP avec des virgules

```
debian@debian:~$ ping dnsproject.prepa.com
PING dnsproject.prepa.com (localhost (:::1)) 56 data bytes
64 bytes from localhost (:::1): icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from localhost (:::1): icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from localhost (:::1): icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from localhost (:::1): icmp_seq=4 ttl=64 time=0.053 ms
```

L'adresse est bien pingable donc l'autre ordinateur peut accéder au serveur DNS

Job 7)

On commence par installer le serveur DHCP isc-dhcp-server avec apt puis on configure notre serveur DHCP comme ceci dans le fichier /etc/dhcp/dhcp.conf

```
default-lease-time 600;
max-lease-time 7200;
```

```
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;
```

```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

On ajoute authoritative pour indiquer que le serveur est celui qui sera utiliser sur le réseau.

```

subnet 192.168.56.0 netmask 255.255.255.0 {
    range 192.168.56.110 192.168.56.120;
    option routers 192.168.56.101;
    option domain-name-servers 192.168.56.101,8.8.8.8;

```

On ajoute aussi la configuration de notre serveur DHCP avec comme adresse de réseau en 192.168.56.0 qui inclura une plage IP attribuable de 11 hotes, on y ajoute aussi une gateway pour un job ultérieur et on configure 2 DNS d'abord celui que l'on a configuré plus tôt et ensuite celui de google permettant d'accéder aux nom d'hotes qui ne sont pas configurés sur notre serveur DNS

Il nous faut ensuite indiquer au serveur sur quel interface attribué le serveur DHCP pour cela nous allons devoir éditer le fichier `/etc/default/isc-dhcp-server` avec notre interface de notre réseau local ipv4 (dans notre cas `enp0s8`)

```
INTERFACESv4="enp0s8"
```

Une fois ceci effectué, nous démarrons notre serveur avec `service isc-dhcp-server restart`  
 Pour nous, regardons la configuration, réseau attribué par le DHCP à notre ordinateur connecté au DHCP.

```

Vitesse de la connexion 1000 Mb/s
Adresse IPv4 192.168.56.111
Adresse IPv6 fe80::456:1f76:c869:81a4
Adresse matérielle 08:00:27:B9:00:D6
Route par défaut 192.168.56.101
DNS 192.168.56.101 8.8.8.8

```

Job 8)

Pour utiliser notre réseau comme une passerelle réseau, il nous faut modifier la configuration de IPtables

```

GNU nano 5.4 /etc/iptables/rules.v4
*nat
-A POSTROUTING -o enp0s3 -j MASQUERADE
COMMIT

*filter
-A INPUT -i lo -j ACCEPT
-A INPUT -i enp0s3 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i enp0s3 -j DROP
COMMIT

```

Ces commandes permettent à la carte `enp0s3` de récupérer les paquets que les autres cartes réseau ne savent pas où envoyer.

On a aussi ouvert le port 22 afin de laisser la connexion SSH possible.

Il nous faut aussi activer le port forwarding dans le fichier de config de systemctl

```
GNU nano 5.4 /etc/sysctl.conf
net.ipv4.ip_forward=1
```

On peut ensuite ping l'adresse du dns de google qui est 8.8.8.8 pour être sûr de ping une adresse internet

```
debian@debian:~$ 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=113 time=2.46 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=3.23 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=2.43 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=3.12 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.425/2.808/3.229/0.366 ms
```

Le ping par Internet fonctionnant sur notre machine qui n'avait pas accès à Internet avant, on peut considérer ce job comme fini.

Job 9)

Commençons d'abord par installer ufw avec un apt install ufw