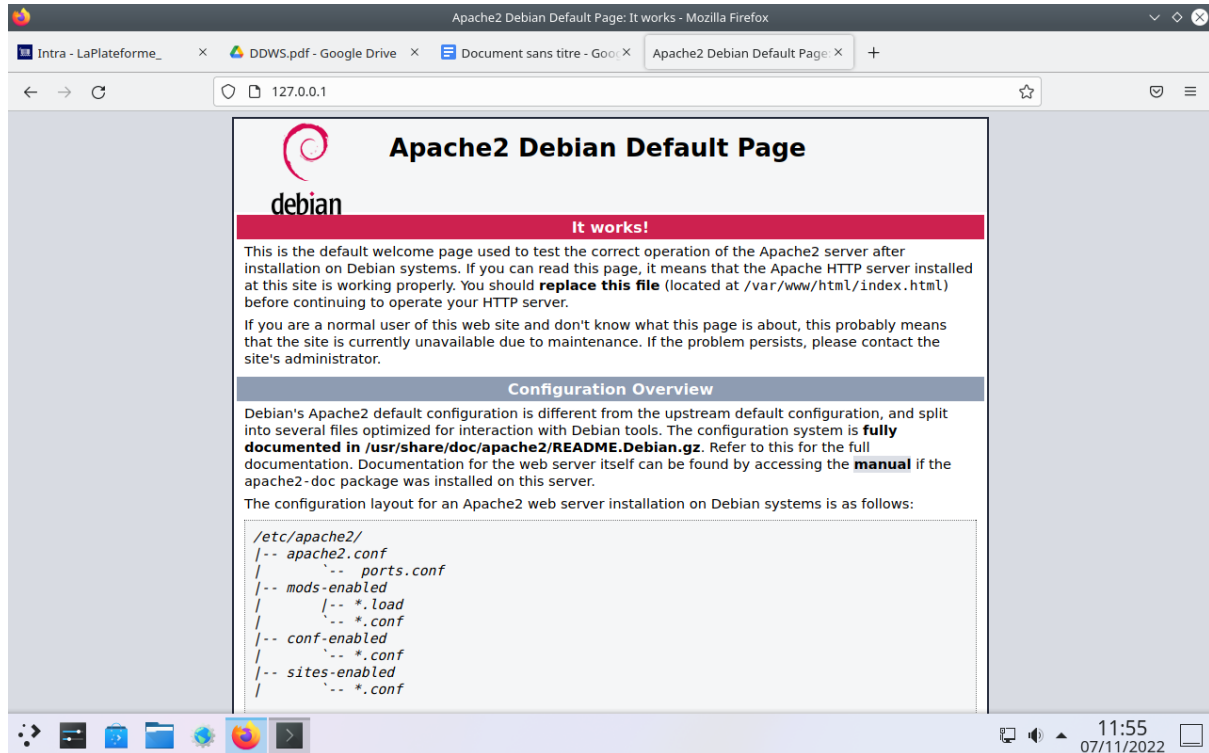


Job 02 :

Je commence par installer apache2 avec `sudo apt-get install apache2`

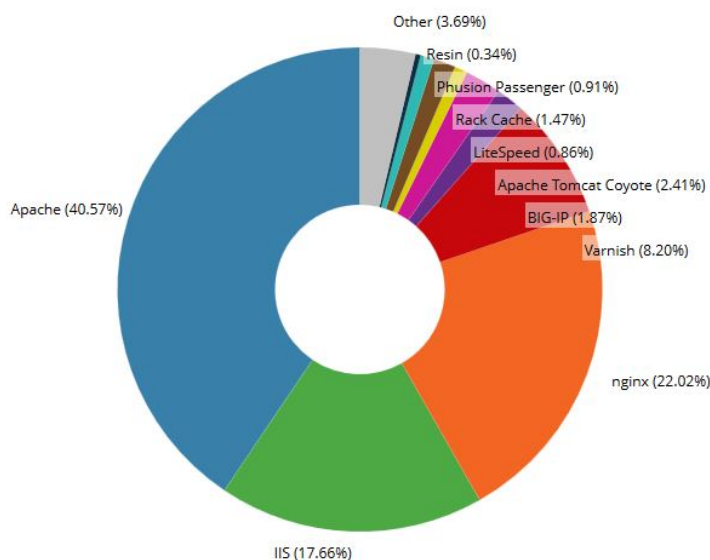
Je vérifie que ça à fonctionné en allant sur mon adresse locale 127.0.0.1 et la page affichée confirme bien que apache2 est bien installé.



Pour accéder à cette page depuis mon hôte je remplace juste mon adresse locale par l'adresse ip de ma vm linux (en l'occurrence 192.168.64.4).

Job 03 :

On peut voir sur ce graphique qui représente les parts de marché des différents serveurs web qu'il en existe trois principaux.



Apache est un serveur web open source présenté comme la colonne vertébrale du web 1.0, il a en effet été très important pour le développement d'internet au début du millénaire. Il profitait d'une architecture logicielle simple basée sur une vision du worldwide web qui n'est plus la nôtre aujourd'hui (site internet moins lourd, bande passant plus réduite etc..)

C'est dans ce contexte que nginx est créée en 2004 avec une architecture dite événementielle (au lieu de dédier un processus ou un thread pour traiter chaque requête, il utilise un modèle événementiel. Cela lui permet notamment de tenir un grand nombre de connexions simultanées sans voir sa consommation mémoire s'envoler.)

Table de comparaison

	Apach	Nginx
<b>Approche</b>	Apache utilise une approche multithread pour traiter les requêtes des clients.	Nginx utilise une approche événementielle pour répondre aux requêtes des clients.
<b>Gestion du contenu dynamique</b>	Il gère le contenu dynamique au sein du serveur Web lui-même.	Il ne peut pas traiter le contenu dynamique de manière native.
<b>Traitement de requêtes</b>	Il ne peut pas traiter plusieurs requêtes simultanément avec un trafic Web important.	Il peut traiter plusieurs requêtes clients simultanément et efficacement avec des ressources matérielles limitées.
<b>Modules</b>	Les modules sont chargés ou déchargés dynamiquement, ce qui le rend plus flexible.	Les modules ne peuvent pas être chargés dynamiquement. Ils doivent être compilés dans le logiciel principal lui-même.
<b>Serveur Web/Proxy</b>	Apache est conçu pour être un serveur Web.	Nginx est à la fois un serveur Web et un serveur proxy.
<b>Thread</b>	Un seul thread ne peut traiter qu'une seule connexion.	Un seul thread peut gérer plusieurs connexions.

En résumé, Apache est plus simple et modulaire que nginx mais nginx répond davantage à des problématiques actuelles du web.

IIS quant à lui est un serveur web qui n'est pas open source et qui n'est compatible qu'avec windows. Il est donc moins utilisé.

Job 04 :

Pour mettre en place un DNS sur mon serveur j'installe bind9 avec `apt-get install bind9`. Ensuite il va falloir modifier les fichiers de configuration pour permettre à bind9 d'agir en DNS sur notre serveur. Premièrement je déclare une zone dans `named.conf.local` (une zone est déclarée pour chaque "site" hébergé; ici "dnsproject.prepa.com").

[illegible]

On renseigne le fichier de configuration de la zone qui sera db.local.  
Dans ce fichier on va configurer le dns

[illegible]

L'enregistrement de noms de domaine est géré par des partenaires de l'ICANN. La certification des clients en tant que propriétaire d'un nom de domaine est confiée, sous traitée à des sociétés ou organismes accrédités par l'ICANN via des registres de noms de domaines ou encore le NIC (Network information center)

Le nom de domaine est composé d'une chaîne de caractères (nom propre, marque ou association de mots clés) et d'une extension qui indique l'espace de nommage. Il existe plusieurs types d'extensions :

Des extensions nationales (ccTLD, “Country Code Top Level Domain”), comme le .fr, le .re ou les autres noms de domaine ultramarins gérés par l’Afnic ;

Des extensions génériques (gTLD, “Generic Top Level Domain”) dont les plus connues sont le .com, .net, .info, .biz. Depuis quelques années, de nombreuses nouvelles extensions génériques ont fait leur apparition, comme .paris, .bzh, .alsace, .corsica.

Job 06 :

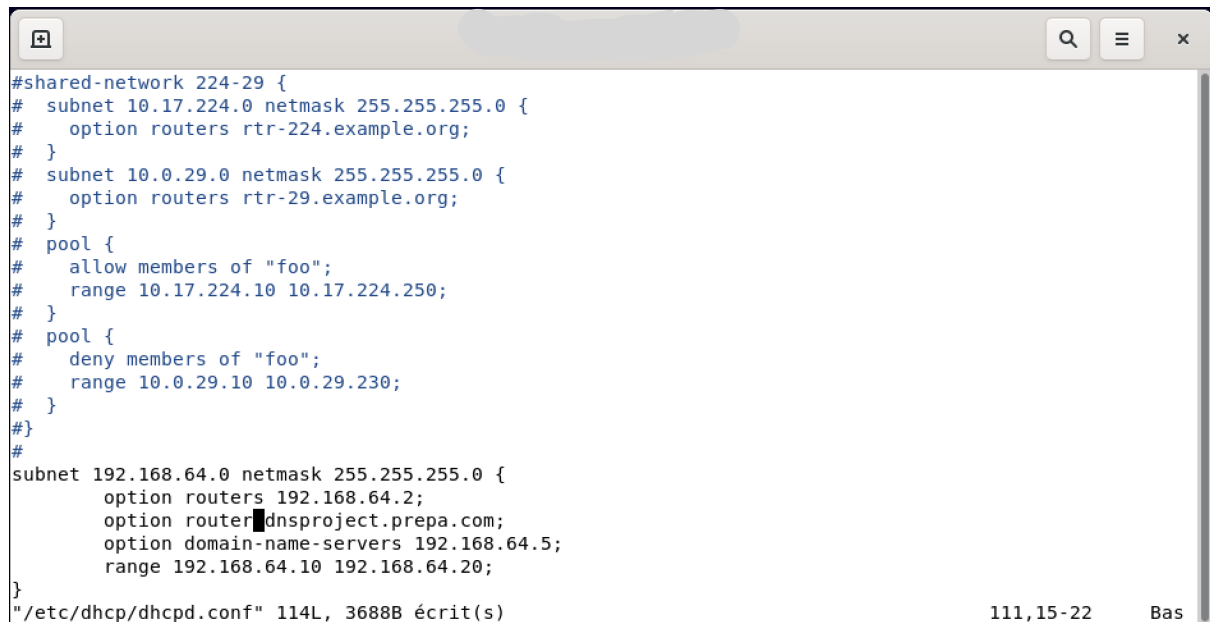
Afin de pouvoir me connecter au serveur apache depuis mon ordinateur hôte je note l'adresse ip de mon serveur que je vais aller ajouter dans le fichier hosts de mon ordinateur hôte. J'ai bien la page par défaut d'apache qui s'affiche sur mon ordinateur hôte. Mais c'est une "fausse" manip qui associe simplement l'ip entrée au nom de domaine donc ça passe toujours par l'ip mais de façon cachée.

[illegible]

En théorie (car je ne peux pas le tester sur la vm de mon mac :) ) Il faudrait changer l'adresse ip de la carte ethernet de l'hôte sur laquelle est "branchée" la vm. Par conséquent, la machine hôte a accès au sous réseau du serveur et peut donc accéder au DNS.

Job 07 :

Pour mettre en place le serveur dhcp j'installe le paquet isc-dhcp-server et je configure l'interface d'écoute avec le nom de la carte réseau du serveur et le fichier dhcpd.conf.



```
#shared-network 224-29 {
# subnet 10.17.224.0 netmask 255.255.255.0 {
#   option routers rtr-224.example.org;
# }
# subnet 10.0.29.0 netmask 255.255.255.0 {
#   option routers rtr-29.example.org;
# }
# pool {
#   allow members of "foo";
#   range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo";
#   range 10.0.29.10 10.0.29.230;
# }
#}
#
subnet 192.168.64.0 netmask 255.255.255.0 {
  option routers 192.168.64.2;
  option router dnsproject.prepa.com;
  option domain-name-servers 192.168.64.5;
  range 192.168.64.10 192.168.64.20;
}
"/etc/dhcp/dhcpd.conf" 114L, 3688B écrit(s) 111,15-22 Bas
```

Job 08 :

La passerelle est spécifiée en même temps que l'interface d'écoute dans /etc/network/interfaces avec gateway "ip de la passerelle"

Job 09 :

Pour installer le pare-feu ufw j'installe le paquet ufw avec apt-get install ufw et je l'active avec sudo ufw enable. Pour interdire le ping il suffit de commenter la ligne `# -A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT` dans /etc/ufw/before.rules

Job 10 :

Pour créer un fichier partager sur mon sous réseau j'installe le paquet samba. Je le configure dans le fichier /etc/samba/smb.conf

Je rajoute un champ [Public] ou je spécifie le dossier que je souhaite partager ici le dossier public de mon utilisateur ainsi que les différents droits qu'auront les utilisateurs qui consulteront le dossier partagé.

**[Public]**

**path = /home/USER/Public**

**browsable = yes**

**writable = yes**

**read only = no**

**force create mode = 0666**

**force directory mode = 0777**

Par défaut, samba n'autorise pas l'accès anonyme il faut donc configurer un utilisateur avec un mot de passe qui permettra au utilisateurs de se connecter au dossier. Ici sudo adduser guestshare.

Ensuite on va lier cet utilisateur a samba et il faudra taper le même mot de passe que pour l'utilisateur linux

**sudo smbpasswd -a guestshare**

Ensuite il faut activer l'utilisateur avec

**sudo smbpasswd -e guestshare**

J'ai bien accès (après une connexion avec guestshare) sur ma deuxième vm au dossier public avec le dossier testsamba

