BTS SIO1 – Bloc 1
Hébergement d'un site web

Présentation
Un serveur web (ou serveur HTTP) permet d'héberger un site et de le rendre disponible aux utilisateurs en font la demande. Les utilisateurs utilisent en général un navigateur web et saisissent l'adresse IP du serveur web ou un nom de domaine pointant vers l'adresse IP du serveur). Le navigateur envoie une requête (depuis un port dynamique) de type HTTP, donc à destination du port 80 (port réservé pour http) du serveur web (parce qu'il « écoute » sur le port 80 pour répondre). Le port peut aussi être 443 s'il s'agit d'un site sécurité via HTTPS (protocole SSL/TLS).

Objectif
Nous allons ici installer un serveur web (logiciel qui héberge des sites web pour les rendre accessibles à distance) avec un outil nommé « apache » sur une machine Linux Debian. Apache est le serveur web le plus populaire et le plus utilisé sur Internet. C'est un logiciel libre. D'autres concurrents sont disponibles comme Nginx ou IIS pour l'environnement Windows.
Le serveur devra être accessible via les protocoles HTTP et HTTPS (à l'aide d'un certificat SSL), à partir d'une adresse IP.

1) Découverte du fonctionnement d'Apache (A LIRE UNIQUEMENT, PAS DE MANIPULATION)
Cette étape permet de se familiariser un peu plus avec Apache (commandes, arborescences des dossiers et fichiers de configuration, etc.).

Commandes de base :

- Arrêter Apache : **`systemctl stop apache2`**
- Lancer Apache : **`systemctl start apache2`**
- Redémarrer Apache : **`systemctl restart apache2`**
- Recharger la configuration d'Apache : **`systemctl reload apache2`**
- Voir la version d'Apache utilisée : **`apache2ctl -v`**
- Tester l'ensemble de la configuration d'Apache : **`apache2ctl -t`**
- Tester la configuration des hôtes virtuels : **`apache2ctl -t -D DUMP_VHOSTS`**
- Voir les modules d'Apache chargés : **`apache2ctl -M`**
- Voir les ports ouverts pour apache : **`ss -luntp | grep apache`**

Fonctionnement sommaire :
Lorsqu'Apache démarre, il charge les fichiers de configuration et attend des requêtes de clients sur ses interfaces réseaux (sur les ports d'écoute, typiquement les ports 80 et 443). Il écoute donc (listen) sur certains ports.
Quand un client fait une demande (requête) au serveur web pour accéder aux pages web, il saisit l'adresse IP du serveur ou le nom de domaine ou clique sur une URL. Dès lors :

1. Le client fait une demande DNS pour résoudre le nom de domaine pour obtenir l'adresse IP du serveur.
2. Il envoie une requête HTTP avec la méthode GET vers le serveur, depuis un de ses ports dynamiques vers le port 80 ou 443 du serveur web.
3. Le serveur écoute sur le port 80 et/ou 443 (ou même un autre port 😉). Il reçoit la demande et va chercher la page à afficher.
4. Il renvoie la page (depuis le même port : 80 ou 443) au client, qui s'affiche enfin sur le navigateur.

Configuration des ports d'écoute :
Le fichier /etc/apache2/ports.conf permet de spécifier les ports à écouter. Par défaut, il s'agit des ports 80 (port par défaut pour HTTP) et 443 (port par défaut pour HTTPS) si le module SSL est activé. Si on veut proposer un site web accessible sur un autre port que le 80 ou 443, il faudra modifier ce fichier et ajouter une autre ligne telle que :
Listen *numeroPort*.

Fichiers de configuration :
Un serveur Apache permet d'héberger un ou plusieurs sites. La configuration ses sites se fait dans plusieurs fichiers de configuration (qui se situent dans le répertoire : /etc/apache2) :

1. sites-available : contient les fichiers de configuration des sites disponibles.
2. conf-available : contient les fichiers de configuration des autres services disponibles.
3. mods-available : contient les fichiers de configuration des modules d'Apache.

Exemple de découverte :
Dans chaque fichier de configuration, on retrouve le port d'écoute et un paramètre DocumentRoot. Par exemple : « DocumentRoot /var/www/html » signifie que le site web affiché correspondra à un fichier nommé « index.html » (ou .php) situé dans le dossier « html » (du dossier parent « www », lui-même enfant du dossier parent « var »).

Hôtes virtuels :
Chaque site web correspond à un hôte virtuel (VirtualHost). Chaque hôte virtuel est défini par un fichier de configuration stocké dans le répertoire /etc/apache2/sites-available/. Donc, c'est le dossier "sites-available" qui contient la configuration des sites web hébergés sous forme de fichiers de configuration.

Le 1er virtual host déjà créé est décrit dans le fichier "000-default.conf" → c'est le site web par défaut répondant sur le port 80.
Le 2ème virtual host déjà créé est décrit dans le fichier "default-ssl.conf" → c'est le site web par défaut répondant sur le port 443.

Il est recommandé de supprimer les virtual host déjà créés et de créer un nouveau virtual host (de A à Z) via un nouveau fichier de configuration (nommé, par exemple, par le nom de domaine auquel il correspond, suivi de l'extension ".conf" ; exemple : /etc/apache2/sites-available/damico.com.conf) pour chaque site web. Mais on peut aussi utiliser les fichiers déjà créés par défaut et les modifier… 😉

Directives présentes éventuellement dans les fichiers de configuration :

- <VirtualHost *:80> : on accepte les connexions sur n'importe quelle IP du serveur (*) sur le port 80.
- ServerName damico.com : cet hôte virtuel sera seulement appelé pour le nom de domaine example.com…
- ServerAlias www.damico.com : …ainsi que pour le sous-domaine www.damico.com. On peut spécifier ici d'autres noms de domaine en les séparant par un espace. On peut aussi utiliser *.damico.com pour inclure tous les sous-domaines.
- DocumentRoot "/var/www/damico" : on placera les fichiers du site dans le répertoire /var/www/damico.
- <Directory "/var/www/damico"> : on spécifie dans cette section des règles pour le répertoire /var/www/damico sous cet hôte virtuel.
- Options +FollowSymLinks : apache suivra les liens symboliques qu'il trouvera dans ce répertoire (et ses descendants).
- AllowOverride all : on pourra inclure une configuration personnalisée via un fichier .htaccess.
- Require all granted : tous les visiteurs pourront accéder au contenu de ce répertoire. Voir la documentation officielle pour modifier ce comportement. Pour des raisons de sécurité ou de privacité on peut par exemple limiter l'accès au serveur à seulement une ou certaines adresses IP avec une directive du type Require ip 192.168.1.10.
- ErrorLog /var/log/apache2/error.damcio.com.log | CustomLog /var/log/apache2/access.damico.com.log combined : il est pratique d'avoir des logs séparés pour chaque hôte virtuel, afin de ne pas mélanger toutes les informations.

Après avoir créé un fichier de configuration, il faut l'activer avec la commande **`a2ensite`** suivie du nom du fichier sans son extension et recharger Apache : **`systemctl reload apache2`**.

2) Préparation du serveur (machine virtuelle)

Cette étape permet d'avoir une machine Linux Debian opérationnelle qui servira de serveur Web. La machine ne sera pas sur un réseau isolé mais sera une machine virtuelle sur le réseau local (donc avec un accès aux autres machines et à Internet).

1. Sur VirtualBox, déployer une nouvelle machine virtuelle Debian, avec un adaptateur réseau en mode pont (pour accéder au réseau local et Internet).

2. Mémoriser l'adresse IP obtenue (via le DHCP du réseau SIO) avec la commande : **`ip a`**
   (sur ma machine, dans cet exemple, l'adresse IP est 10.100.0.115/8 sur la carte réseau ens192 ; il faudra donc adapter les commandes avec l'adresse IP de VOTRE Debian 😉).

   →Noter l'adresse IP qui vous a été attribuée : ……

   ⓘ Ça sera l'adresse IP du serveur, donc l'adresse IP que les clients devront contacter pour accéder aux sites hébergés sur le serveur.

3. Effectuer un test de connectivité réseau PING vers un serveur de notre réseau : le 10.0.0.1

   →Dire quel est l'utilité de faire ce test : ……

4. Effectuer un test de connectivité réseau PING vers un serveur situé sur Internet (typiquement, le serveur DNS ouvert proposé par Google qui répond tout le temps 😉) : le 8.8.8.8

   →Dire quel est l'utilité de faire ce test : ……

5. Effectuer un test de connectivité réseau PING vers un serveur situé sur Internet mais à partir de son nom : par exemple google.com

   →Dire quel est l'utilité de faire ce test : ……

6. Afficher les ports ouverts sur la machine avec la commande : **`ss -lntup`** (ou bien **`ss -lntp`** pour les ports TCP et **`ss-lnup`** pour les ports UDP)

3) Installation du service Apache2

Cette étape permet d'installer le logiciel permettant de proposer un service d'hébergement web. Cela transforme le serveur en serveur web (et ouvre le port numéro 80 en TCP).

1. Mettre à jour le système et les paquets (logiciels installés)
   ```
   apt update && apt upgrade
   ```

2. Installer le paquet Apache2
   ```
   apt install apache2
   ```

3. Vérifier que le service est démarré et opérationnel
   ```
   systemctl status apache2
   ```
   (appuyer sur les touches CTRL+C pour quitter)

4. Afficher les ports TCP ouverts sur la machine.

   →Quel port est désormais ouvert ? : ……

5. Effectuer un test pour accéder au serveur web (depuis le poste physique, sur un navigateur) : *http://adresseIPdeVOTREserveur*

   Une page « It's works » devrait s'afficher. Cela signifie que le serveur web est fonctionnel. Il s'agit d'une page web d'exemple par défaut proposée par Apache. Elle se nomme « index.html » et se trouve dans le dossier suivant : /var/www/html/.

6. On souhaite maintenant modifier le contenu de la page affichée. Se rendre dans le dossier du serveur web avec la commande :
   ```
   cd /var/www/html
   ```

7. Editer le fichier index.html via le logiciel nano (équivalent au bloc note de Windows) avec la commande :
   ```
   nano index.html
   ```

8. Supprimer la page d'exemple (index.html) avec la commande :
   ```
   rm index.html
   ```

9. Créer une nouvelle page web (fichier nommé « index.html ») :
   ```
   touch index.html
   ```

10. Ajouter le contenu suivant dans le fichier index.html :
    ```
    nano index.html
    ```

    ```
    <html><body>Site en construction par les SLAM – Serveur géré par les SISR</body></html>
    ```
    (enregistrer et quitter, avec les touches : CTRL+X, puis la touche O pour confirmer l'enregistrement puis Entrée pour confirmer le nom du fichier)

11. Vérifier que la nouvelle page web s'affiche lorsqu'on consulte le site web.

4) Hébergement d'un nouveau site HTTP accessible sur un autre port, par exemple le port 40001 (le port 80 est déjà utilisé)

Lorsqu'on accède à un site web HTTP depuis un navigateur, on a souvent juste à saisir l'adresse IP du site ou le nom de domaine (et c'est le serveur DNS qui traduire en adresse IP). Donc on ne précise pas sur quel port il faut « toquer »… Car par défaut, c'est le port 80. Mais, il se peut qu'un site web se cache derrière un autre port. Souvent, c'est pour héberger discrètement des pages web telles que des pages réservées aux administrateurs. Dans ce cas, pour y accéder, il faudra saisir dans le navigateur : http://IPduSiteWeb:portAcontacter (exemple : http://10.100.15.15:8081).

1. Créer un dossier nommé « monSite » dans /var/www/ et se déplacer dans le dossier créé
   ```
   mkdir /var/www/monSite
   cd /var/www/monSite
   ```

3. Créer un fichier nommé « index.html » dans le dossier monSite

```
touch index.html
```

4. L'éditer en y intégrant des balises HTML (ou alors demander à un développeur de créer une nouvelle page HTML 😉)

```
nano index.html
```

```
<html><body>Ceci est la page d'un site discret</body></html>
```

5. Créer un nouveau fichier nommé « sitePerso.conf » dans /etc/apache2/sites-available/ (ce dossier permet de déclarer/lister tous les sites web à héberger)

```
cd /etc/apache2/sites-available
touch sitePerso.conf
```

6. Ajouter les balises obligatoires (VirtualHost, ServerName et DocumentRoot)

```
nano sitePerso.conf
```

```
<VirtualHost *:40001>
     ServerName ecrireAdresseIPdeVOTREserveur
     DocumentRoot /var/www/monSite
</VirtualHost>
```

→Dire à quoi sert ce fichier : ……

7. Enregistrer et fermer

8. Activer le nouveau site

```
a2ensite sitePerso
```

→Dire à quoi sert cette commande : ……

9. Modifier le fichier « ports.conf » puis autoriser l'écouter sur le port 40001

```
nano /etc/apache2/ports.conf
```

```
Listen 40001
```
(à mettre juste en dessous de Listen 80)

10. Recharger Apache et tester (via http://ecrireAdresseIPdeVotreServeur:8081)

```
systemctl restart apache2
```

11. Afficher les ports TCP ouverts sur la machine. Si le redémarrage ne fonctionne pas, c'est qu'il y a une erreur de syntaxe dans un fichier de configuration !

→Quel port est désormais ouvert ? : ……

Schéma récapitulatif du fonctionnement :

5) Hébergement de plusieurs sites web derrière un même port, comme le port HTTP 80

Souvent, un serveur web héberge plusieurs sites Web. Pour faciliter la navigation des internautes, tous se trouvent derrière le port 80 HTTP (ou 443 HTTPS s'il est sécurisé) afin que l'internaute n'ait pas à connaître ni saisir le numéro du port correspondant (ça serait pénible voire impossible à retenir le numéro de port pour chaque site web !). Pour ce faire, on va dire à chacun de nos VirtualHost (sites hébergés) à quel nom de domaine ils doivent répondre.

Par exemple : http://gateaux.com et http://voyages.com seront hébergés sur le même serveur, derrière le même port (HTTP 80), mais afficheront des pages web différentes car le VirtualHost appelé sera choisi en fonction du nom de domaine présent dans l'URL 😊.

Schéma de la nouvelle architecte à faire :

1. Supprimer tous les VirtualHost

```
cd /etc/apache2/sites-available
rm *
```

2. Supprimer tous les dossiers et fichiers des sites web

```
cd /var/www/
rm * -r
```

3. Dans le fichier « ports.conf », ne laisser que le port 80 ouvert (toutefois, laisser le port 443 dans les modules).

4. Redémarrer apache.

5. Dans le serveur web (/var/www), créer le dossier « carrefour » et, à l'intérieur, créer une page « index.html » avec le contenu suivant : « Site web de Carrefour – En construction ».

6. Dans le serveur web (/var/www), créer le dossier « casino » et, à l'intérieur, créer une page « index.html » avec le contenu suivant : « Site web de Casino – En construction ».

7. Dans le dossier des VirtualHost (/etc/apache2/sites-available), créer le VirtualHost du site qui gérera Carrefour :

```
touch carrefour.conf
nano carrefour.conf
```

```
<VirtualHost *:80>
     ServerName carrefour
     DocumentRoot /var/www/carrefour
</VirtualHost>
```

8. Dans le dossier des VirtualHost (/etc/apache2/sites-available), créer le VirtualHost du site qui gérera Casino :

```
touch casino.conf
nano casino.conf
```

```
<VirtualHost *:80>
     ServerName casino
     DocumentRoot /var/www/casino
</VirtualHost>
```

9. Activer les 2 sites web :

```
a2ensite casino.conf
a2ensite casino.conf
```

10. Redémarrer Apache.

11. Demander au professeur de vous créer 2 noms de domaine (du type : casinoNOM et carrefourNOM) qui pointent vers l'adresse IP de votre serveur web (donc lui fournir votre adresse IP en 10.100.x.x).

12. A partir du PC physique ou de n'importe quelle machine de la salle, tester via un navigateur en saisissant : http://casinoNOM/ et http://carrefourNOM/ (exemple : http://carrefourDAMICO/). Normalement, selon le nom de domaine saisit, le bon site web apparaîtra 😊.