

R3.06 Architecture des Réseaux

TD2

1. INSTALLATION D'UNE MACHINE VIRTUELLE DEBIAN

Créer une machine virtuelle Debian dans VirtualBox (2 Go de Ram, 12Go disque). Installer Debian sur cette VM, depuis une image ISO que vous pouvez récupérer sur le site Debian (<https://www.debian.org/distrib/>).

Installer des Additions Invité dans la VM Debian

2. Installer des Additions Invité dans la VM Debian

- a. Avant de commencer l'installation des Additions Invité, exécutez les commandes suivantes. Expliquez à quoi sert ces 2 commandes :

```
sudo apt update  
sudo apt upgrade
```

- b. Il faut installer les paquets indispensables pour modifier les modules du noyau lors de l'installation des Additions Invité :

```
sudo apt install dkms build-essential module-assistant  
sudo m-a prepare
```

- c. Insérez l'image du CD des Additions Invité, en passant par « Périphériques > Insérer l'image CD des Additions Invité ».
- d. Enfin, pour installer les Additions Invité, exécutez la commande suivante :

```
sudo sh /monchemin/vers/cd/VBoxLinuxAdditions.run (/monchemin/vers/cd :  
est à remplacer par le chemin vers votre CD-ROM)
```

- e. Redémarrez votre VM :

3. INSTALLATION D'UN SERVEUR APACHE

- a. Installer le paquetage apache2 avec votre gestionnaire de paquetage apt-get
- b. Voilà quelques commandes utiles pour la gestion de votre serveur Apache2 :

- **démarrage d'Apache:**

```
apache2 -k start (avec debian, vous pouvez utiliser /etc/init.d/apache2 à la place de « apache»
```

- **arrêt immédiat**

```
apache2 -k stop
```

- **autre manière d'arrêt (arrêt après le traitement des requêtes en cours par les processus enfants):**

```
apache2 -k graceful-stop
```

- **redémarrage : expliquez la différence entre ces 2 commandes suivantes :**

```
apache2 -k restart
```

```
apache2 -k reload
```

- c. Dans apache2, le fichier de configuration principal du serveur est `/etc/apache2/apache2.conf`. ouvrir ce fichier avec un éditeur de texte, et observer son contenu.

Les fichiers de Configuration Apache2 :

Sur certaines distributions, le fichier de configuration principal, `apache2.conf` est organisé de façon à sous-traiter les directives qu'il est sensé contenir à d'autres fichiers. Cela permet de séparer la configuration en blocs facilement repérables et de faciliter ainsi la maintenance d'Apache.

Sur Debian, par exemple, il faut considérer l'organisation suivante :

- le fichier « `apache2.conf` » pour la configuration principale d'Apache ;
 - les sous-répertoires « `conf-available` » pour des morceaux de configurations supplémentaires disponibles ;
 - « `conf-enabled` » pour les morceaux de configurations supplémentaires actifs (c'est-à-dire effectivement pris en compte par Apache) ;
 - les sous-répertoires « `mod-available` » et « `mod-enabled` » pour les modules d'extension disponibles et actifs ;
 - les sous-répertoires « `sites-available` » et « `sites-enabled` » pour la configuration des hôtes virtuels (les sites web) disponibles et actifs.
- d. Voici quelques directives utilisables dans le contexte de configuration globale du serveur :
 - `Servername` : contient le nom d'hôte et le port que le serveur utilise pour s'authentifier lui-même.
 - `ServerRoot` : correspond au répertoire d'installation d'Apache qui contient ses fichiers de configuration (`/etc/apache2` sur Debian). C'est à partir de ce répertoire que seront calculés les chemins relatifs contenus dans les autres directives.
 - `DocumentRoot` : permet de définir la racine des documents distribués par Apache (le répertoire où chercher les fichiers).
 - `ErrorDocument` : permet de changer le comportement par défaut d'Apache lorsqu'une erreur est rencontrée. Cette directive permet une redéfinition par code d'erreur. Cela permet par exemple d'afficher une page d'erreur personnalisée.
 - e. Si vous modifiez les fichiers de configuration Apache, vous pouvez vérifiez si vos fichiers de configuration contiennent des erreurs de syntaxe avec la commande suivante :

```
/usr/sbin/apache2ctl -t
```
 - f. Déterminer les commandes permettant de :
 - Obtenir la version d'Apache
 - Lister des modules chargés par Apache

ATTENTION : si vous souhaitez modifier la configuration Apache pour faire des tests, n'oubliez pas de stocker une version du fichier de configuration avant de le modifier, cela vous permettra d'avoir toujours une configuration propre sous la main.

- g. En se basant sur la documentation officielle, déterminer les directives (dans les fichiers de configuration) qui permettent de :
 - fixer le nombre maximum de requêtes pouvant être traitées simultanément
 - activer les connexions persistantes
 - limiter le nombre de requêtes autorisées par connexion lorsque les connexions persistantes sont actives.

4. TESTER LE SERVEUR APACHE

Exercice1

- a. Déterminez l'adresse ip de votre serveur
- b. Allez sur un navigateur sur votre machine windows, et tapez comme URL l'adresse IP de votre serveur Apache. Commentez.
- c. Arrêter le serveur Apache et refaites la question a).
- d. Démarrer de nouveau Apache.
- e. Allez sur le site <https://www.noip.com/> et créer un compte. Créer ensuite un nom de domaines gratuit et remplacer l'adresse IP associé au hostname en remplaçant celle existante par l'adresse IP de votre machine virtuelle.
- f. Allez sur un navigateur sur votre machine windows, et tapez l'adresse du nom de domaine que vous venez de créer. Commenter.

Exercice2

Nous allons maintenant voir comment créer des répertoires spécifiques pour chaque utilisateur du système.

- a. Si vous n'avez pas d'utilisateur autre que root, créer en un avec les commandes suivantes :

```
useradd -m -d /home/ monNom / -s /bin/bash monNom
```

```
passwd monNom
```
- b. Créer un répertoire public_html dans le homedir de monNom (bien faire cela sur le compte monNom et non le compte root).
- c. Rajouter une page index.html dans le nouveau répertoire public_html.
- d. Revenez sous le compte root. Installer le module userdir, de la même suivante :

`/usr/sbin/a2enmod userdir`

- e. redémarrer le serveur http
- f. Allez sur un navigateur sur votre machine windows, et tapez l'adresse suivante :

```
http://xxx.xxx.xxx.xxx/~ monNom / (remplacer xxx.xxx.xxx.xxx par l'adresse IP de votre serveur)
```

5. INSTALLATION D'UN SERVEUR NGINX

Comme pour Apache, pour Nginx on retrouvera généralement un découpage de la configuration en plusieurs fichiers répartis dans différents répertoires.

Sur Debian :

- **nginx.conf** : fichier de configuration générale ;
 - **modules-available** et **modules-enabled**: dossiers contenant les fichiers permettant l'activation des modules de Nginx ;
 - **conf.d** : contient des fichiers de configuration générale pour le serveur web (Nginx peut aussi faire office de proxy mail mais la configuration de cette fonctionnalité ne se fera pas dans conf.d) ;
 - **sites-available** et **sites-enabled** : dossiers contenant la configuration des hôtes virtuels.
-
- a. Arrêter le serveur Apache sur votre VM et ensuite installer le serveur nginx.
 - b. Déterminer la version installée de Nginx