

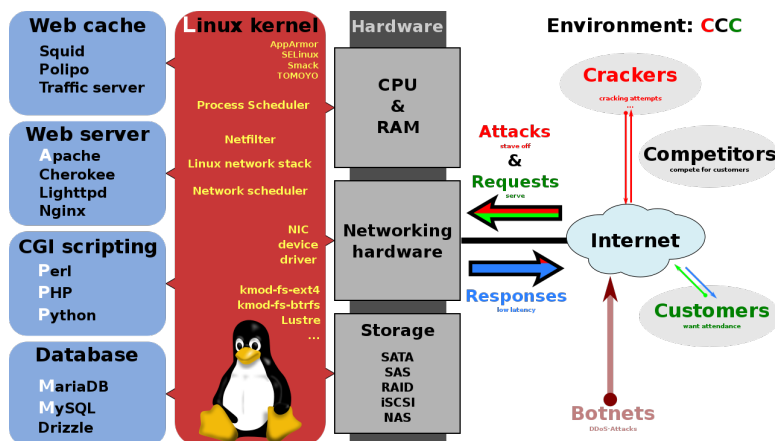


TP LAMP

24.01.2023

Cela va vous éclairer sur ce qu'est un service, les sockets etc ... !!! ☺

Auteur : Pascal Fougeray



Source : <https://fr.wikipedia.org/wiki/LAMP>

1 Introduction

Dans ce TP, je vous propose d'installer un service ou disons un ensemble de services très utilisés dans le domaine du développement.

Il existe de nombreux tutos sur Internet à ce sujet et vous pouvez aller les lire si vous désirez approfondir le sujet.

Comme vous êtes de très bons développeurs vous pourrez l'utiliser pour développer un site Web avec plein de belles choses, même mettre plein de sites virtuels etc...

Dans ce TP vous allez voir

- Comment on installe des **services** et on les active avec la commande **systemctl** (voir cours **Systemd**)
- Revoir ce qu'est un processus **daemon**, son utilité, qui est son père et pourquoi le tuer le relance ☺
- Voir ce qu'est le scan de ports (**nmap**)
- Voir quels ports sont ouverts (**netstat**)
- Voir ce qu'est une **socket** local
- **Vous devez câbler le câble rouge sur l'interface BRO et vérifiez que vous avez une IP sur sur réseau 192.168.128.0/22**

2 La structure

Elle est simple, nous allons partir d'une debian 11 avec toujours les mêmes comptent root et étudiant.

Un accès en pont sur le réseau 192.168.128.0/22 donc vos serveurs sont tous accessibles par tout le monde se trouvant sur ce réseau.



3 Préparation

1. **Lancez** la VM de base qui ne contient que le serveur **ssh** !
2. **Vérifiez** qu'elle ping l'extérieur, **ping -c 2 8.8.8.8**
C'est quoi cette @IP ?
3. **Mettez** à jour le système : **apt update && apt full-upgrade**
C'est quoi ces 2 commandes ?
4. **Visualisez** ce qu'elle écoute en TCP : **netstat -lnpt4**

```
root@debian-10-etu:~# netstat -lnpt4
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale      Adresse distante    Etat      PID/Program name
tcp        0      0 0.0.0.0:ssh          0.0.0.0:*            LISTEN    736/sshd
tcp        0      0 localhost:6010       0.0.0.0:*            LISTEN    1391/sshd: etudiant
tcp6       0      0 [::]:ssh            [::]:*              LISTEN    736/sshd
tcp6       0      0 localhost:6010      [::]:*              LISTEN    1391/sshd: etudiant
root@debian-10-etu:~#
```

Si vous voyez les ports 25, 110 et 143, c'est normal, **souvenez** vous du TP intitulé Protocoles ☺

4 LAMP

LAMP est un acronyme :

- **Linux** : Le système d'exploitation
- **Apache** : Le serveur Web
- **MySQL / MariaDB** : Le serveur de Base de données
- **PHP** : Le langage de script

Avec LAMP, on peut

- mettre en place un serveur Web,
- hébergeant un site web dynamique écrit en PHP,
- tout en allant chercher des données dans une base MySQL devenu MariaDB, son **fork**.

4.1 Apache

Le serveur Web

On installe le serveur apache et son module permettant de gérer PHP :

1. Avant on fait une mise à jour **apt update && apt upgrade**
Cela peut être un peu long, mais pas certain..., cela dépend des nouveaux paquets, je ne peux le savoir et le gérer d'une année sur l'autre.
Vous devez obtenir quelque chose comme ça

```
root@debian-11-GNS3:~# apt update && apt upgrade
Réception de :1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Réception de :2 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]
Réception de :3 http://deb.debian.org/debian bullseye-updates InRelease [44,1 kB]
Réception de :4 http://deb.debian.org/debian unstable InRelease [161 kB]
Réception de :5 http://deb.debian.org/debian bullseye/main Sources [8634 kB]
Réception de :6 http://security.debian.org/debian-security bullseye-security/main Sources [176 kB]
Réception de :7 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [176 kB]
Réception de :8 http://security.debian.org/debian-security bullseye-security/main Translation-en [176 kB]
Réception de :9 http://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
...
Etc ...
```

2. **Lancez** la commande **netstat -lnpt4** et **constatez** que les **ports 80 et 443 sont fermés** !
Pour **stopper** les **3 serveurs de mails** que nous avons installés lors du **TP Protocoles** !
Lancez éventuellement les commandes
 - **systemctl stop postfix**
 - **systemctl stop dovecot.service**
 - **systemctl stop popa3d.service**



3. Installation du serveur Web : **apt install apache2 libapache2-mod-php**

Le système va vous suggérer d'installer aussi les paquets :

libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline

```
root@debian-11-GNS3:~# apt install apache2 libapache2-mod-php
```

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances... Fait

Lecture des informations d'état... Fait

apache2 est déjà la version la plus récente (2.4.54-1~deb11u1).

apache2 passé en « installé manuellement ».

Les paquets supplémentaires suivants seront installés :

libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline

Paquets suggérés :

php-pear

Les NOUVEAUX paquets suivants seront installés :

libapache2-mod-php libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache

0 mis à jour, 8 nouvellement installés, 0 à enlever et 0 non mis à jour.

Il est nécessaire de prendre 4079 ko dans les archives.

Après cette opération, 17,9 Mo d'espace disque supplémentaires seront utilisés.

Souhaitez-vous continuer ? [O/n]

4. **Expliquez** les dernières lignes renvoyées lors de l'installation **Created symlink**. Voir le cours sur **Systemd**.

Enabling conf localized-error-pages.

Enabling conf other-vhosts-access-log.

Enabling conf security.

Enabling conf serve-cgi-bin.

Enabling site 000-default.

Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.

Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.

Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...

5. **Validez** le lancement automatique de ce service au démarrage du serveur

systemctl enable apache2

6. **Visualisez** son état et **expliquez** ce que vous comprenez !

systemctl status apache2

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-01-17 18:23:00 CET; 2min 49s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 20640 (apache2)
    Tasks: 55 (limit: 14288)
   Memory: 11.0M
      CPU: 60ms
   CGroup: /system.slice/apache2.service
           └─20640 /usr/sbin/apache2 -k start
             └─20642 /usr/sbin/apache2 -k start
               └─20643 /usr/sbin/apache2 -k start

janv. 17 18:23:00 debian-11-GNS3 systemd[1]: Starting The Apache HTTP Server...
janv. 17 18:23:00 debian-11-GNS3 apachectl[20639]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, us
janv. 17 18:23:00 debian-11-GNS3 systemd[1]: Started The Apache HTTP Server.
```

Le but du TP n'est pas de configurer en détails le serveur Web, cela sera fait en M1, donc on ne perd pas de temps à modifier la conf !

Un peu de réseau et de système !

Nous allons voir des choses intéressantes ☺

7. **Lancez** la commande **nmap -A -T4 localhost** (En **root** ou comme **étudiant** peu importe !) et **expliquez** ce que vous obtenez, **interprétez** les résultats

Si la commande **nmap** n'est pas installée, on l'installe : **apt install nmap**

Vous devez obtenir quelque chose comme cela



```

root@debian-11-GNS3:~# nmap -A -T4 localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-17 18:39 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000087s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
23/tcp    open  telnet   Linux telnetd
80/tcp    open  http     Apache httpd 2.4.54 ((Debian))
|_ http-server-header: Apache/2.4.54 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.47 seconds
root@debian-11-GNS3:~#

```

8. **Lancez** cette commande sur la VM d'un voisin en lui demandant son adresse **IP 192.168.128.xx/22** et **vérifiez** que vous obtenez la même chose !

Si vous désirez approfondir cette commande nmap :

<https://nmap.org/man/fr/index.html>

Si vous désirez avoir une belle interface graphique vous pouvez installer et utiliser : **nmapi4 (pas en TP, chez vous !)**

9. **Lancez** les commandes

(a) **ps axjf | grep apache2**

(b) **pstree -lp | grep apache2**

(c) Que **constatez**-vous ?

10. **Expliquez** ce que vous comprenez en répondant aux question suivantes

(a) Qui est le père du premier apache2 ? Pourquoi est-ce lui ? **Relevez** son PID !

(b) Qui est le père des suivants ? Pourquoi ?

11. **Relancez** la commande **netstat -lpta4** et **relevez** le PID du processus qui écoute

12. **Comparez** cette valeur à celle **relevée** précédemment

Expliquez pourquoi c'est comme cela !

13. **Tuez** le premier **kill -9 PID** et **relancez** la commande **ps axjf | grep apache2**

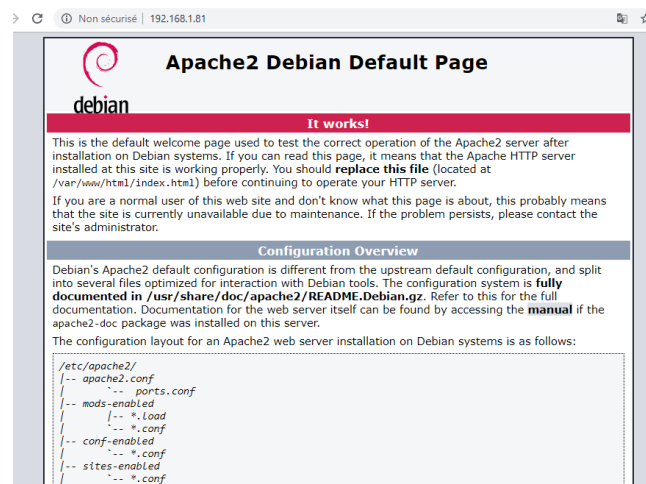
(a) **Expliquez** ce que vous obtenez, pourquoi est-ce comme cela ?

(b) Quel type de processus est apache 2

(c) Pourquoi est-il de ce type ?

14. On teste Apache en lançant un **feuilleteur** et en tapant l' @IP de votre VM.

Vous devez avoir une page web comme celle-ci



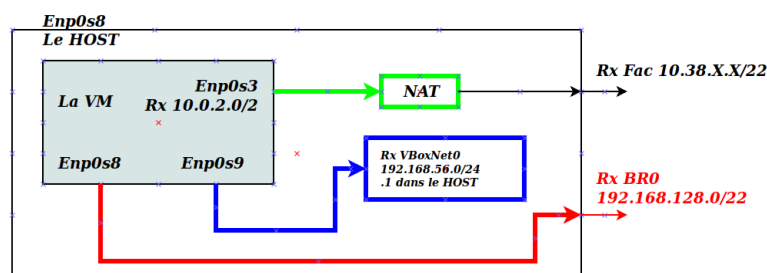
Votre serveur Web fonctionne ☺

Allez je ne résiste pas de voir si vous avez compris le problème d'accès à une machine, ici un serveur !



15. Dans le HOST, oui oui dans le HOST, **lancez** un navigateur web et **essayez** de voir la page Web de votre VM

Je rappelle la structure !



Si vous y arrivez du premier coup, félicitations ☺

Sinon et bien je vais vous expliquer ☺

4.2 PHP

Je ne vais pas vous faire l'affront de vous expliquez ce qu'est PHP etc ...

1. Installation de PHP : **apt install php php-cli**

2. **Expliquez** ce qu'est php-cli !

PHP est installé, mais on n'a pas grand chose. Il faut installer des modules en fonction des besoins. Les modules sont nommés ainsi : php-xxx. On peut les lister à l'aide la commande

3. **apt search ^php-**

4. **Lancez** cette commande et **comptez** le nombre de modules

pas à la main sinon le TP va être trop long ☺mais : **apt search ^php- | grep php- | wc -l**

Conclusion : on ne les installe pas tous !

5. Les paquets les plus courants à installer sont **curl, gd, intl, memcache, xml, zip, mbstring, json** on les installe juste pour voir une nouvelle commande de apt !

apt install php-{curl,gd,intl,memcache,xml,zip,mbstring,json}

6. Un autre paquet est important pour notre TP !

Celui qui nous intéresse et qui va nous permettre d'interagir avec notre BDD mysql son nom est donc **php-mysql** ☺.

Ce paquet permet d'attaquer une BDD telle MySQL/MariaDB avec **mysqli** ou **pdo**.

7. **Installez le !** : **apt install php-mysql**

8. Comme on a installé un nouveau module pour le serveur Web, il faut lui dire de relire sa conf !

systemctl reload apache2


9. On teste que PHP fonctionne !

(a) On se rend dans le répertoire par défaut de la racine d'apache : **/var/www/html**

(b) On crée un script PHP : **echo "<?php phpinfo(); ?>" > test.php**

(c) Dans votre feuilletteur **saisissez** l'url : **votreIP/test.php** vous devez voir apparaître la page suivante !

Non sécurisé | 192.168.1.68/test.php

PHP Version 7.3.4-2	
	
System	Linux debian-10-etx 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2 (2019-08-28) x86_64
Build Date	Apr 13 2019 19:05:48
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/10-redis.ini, /etc/php/7.3/apache2/conf.d/10-xdebug.ini

Remarque : On peut éditer les options de PHP via le fichier

/etc/php/7.*/apache2/php.ini

ou créer un fichier personnalisé dans

/etc/php/7.*/apache2/conf.d.

Mais cela est un autre TP et pas à faire ici dans ce module !



4.3 Maria-DB

Maria-DB est un fork de **mysql** depuis quelques années et c'est celui qui doit être utilisé. Maintenant, il ne reste plus que le moteur de BDD à installer.

1. Installation du paquet ! : **apt install mariadb-server**
2. **Vérifiez** que le serveur écoute : **netstat -lntp4**
3. **Relevez** le numéro de port de mariaDB !
4. **Vérifiez** l'état du serveur de BDD : **systemctl status maria-db** qui doit vous renvoyer quelque chose comme :

```
root@debian-11-GNS3:/var/www/html# systemctl status mariadb.service
● mariadb.service - MariaDB 10.5.18 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-01-20 12:30:00 CET; 1min 15s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 8300 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
   Process: 8301 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 8303 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=cd /usr/bin/..; /usr/bin/galera_recovery; [ $? -eq 0 ] && systemctl set-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 8370 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 8372 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 8351 (mariadbd)
    Status: "Taking your SQL requests now..."
     Tasks: 19 (Limit: 14269)
    Memory: 74.8M
       CPU: 570ms
   CGroup: /system.slice/mariadb.service
           └─8351 /usr/sbin/mariadbd

janv. 20 12:30:00 debian-11-GNS3 mariadbd[8351]: Version: '10.5.18-MariaDB-0+deb11u1' socket: '/run/mysqlqld/mysqlqld.sock' port: 3306 Debian 11
janv. 20 12:30:00 debian-11-GNS3 systemd[1]: Started MariaDB 10.5.18 database server.
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8374]: Upgrading MySQL tables if necessary.
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8377]: Looking for 'mariadb' as: /usr/bin/mariadb
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8377]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8377]: This installation of MariaDB is already upgraded to 10.5.18-MariaDB.
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8377]: There is no need to run mysql_upgrade again for 10.5.18-MariaDB.
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8377]: You can use --force if you still want to run mysql_upgrade
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8385]: Checking for insecure root accounts.
janv. 20 12:30:00 debian-11-GNS3 /etc/mysql/debian-start[8389]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
log file: systemctl status maria-db systemctl status maria-db
```

5. **Expliquez** l'utilité du fichier **/var/run/mysqlqld/mysqlqld.sock**
6. **Faites** un **ls -l** de ce répertoire **/var/run/mysqlqld/**, **visualisez** les 2 fichiers !
 -rw-rw---- 1 mysql mysql 5 20 janv. 12 :30 mysqlqld.pid
 srwxrwxrwx 1 mysql mysql 0 20 janv. 12 :30 mysqlqld.sock
7. Vous voulez voir les sockets ? **Allez** à la racine et en tant que root **lancez** la commande **find -type s**
8. Configuration et installation de **mariadb** !
 Dans un souci de sécurité on va modifier le MDP root pour la BDD passer de **Root1** à **RootMysql1**
 - (a) **Lancez** la commande : **mysql_secure_installation**
 - (b) **Changez** le MDP de l'utilisateur root pour la BDD
 - (c) **Répondez** par Y aux 4 questions et **interprétez** les !
 - (d) **Testez** la connexion à la BDD : **mysql -u root -p**
 - (e) **Créez** un utilisateur spécifique, en l'occurrence l'utilisateur **etudiant** ! Cela doit donner le code SQL suivant :

```
CREATE USER 'etudiant'@'localhost' IDENTIFIED BY 'mdp';
GRANT ALL PRIVILEGES ON *.* TO 'etudiant'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

```
root@debian-10-etu:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 56
Server version: 10.3.17-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'etudiant'@'localhost' IDENTIFIED BY 'mdp';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'etudiant'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
Bye
```

- (f) **Déconnectez** vous de la BDD : **exit**
- (g) **Lancez** la commande **ps ajfx | grep mysql** et **interprétez** la !
Connectez vous en tant qu'utilisateur **etudiant** et **jouez** avec votre **BDD, PHP etc... ? non pas maintenant !**

Si vous voulez approfondir le sujet, tout est sur le net ☺



5 Du ménage

On a installé des paquets et ils sont encore sur le serveur, ils occupent donc de la place !

1. **Allez** dans le répertoire **/var/cache/apt/archives**
2. **Listez** les paquets ***.deb** et **interprétez**
3. **Lancez** la commande **df -h** et **interprétez**
4. **Lancez** la commande **apt clean**
5. **Listez** les paquets ***.deb** et **interprétez**
6. **Lancez** la commande **apt autoremove**
7. **Lancez** la commande **df -h** et **interprétez**
8. **Concluez** sur ce ménage ☺

6 Conclusion

Concluez sur la manière, les tests etc... concernant l'installation de services sur un serveur!!!!

L'administration système c'est aussi installer des services, voir s'ils fonctionnent en faisant un peu de réseau et la gestion des utilisateurs ?