

A thick, dark green vertical bar runs down the left side of the page. A light green arrow points from the right edge of this bar towards the title.

Projet Epreuve E4

Serveur LAMP - HaProxy

Several thin, curved lines in shades of green and grey originate from the bottom left corner and sweep upwards and to the right, creating a decorative, organic feel.

Dorian Laporte
BTS SIO

Sommaire :

➤ Situation n°1 :

- Installation des serveurs LAMP
- Sécurisation d'accès à l'espace personnel d'un utilisateur
- Connexion avec Pam_LDAP à l'annuaire

➤ Situation n°2 :

- Installation du serveur HaProxy
- Mise en place de la réplication avec « DRBD »

Objectifs :

- Installer deux serveurs LAMP.
- Sécuriser l'accès par mot de passe à l'espace personnel.
- Mettre en place une connexion PAM_LDAP avec l'annuaire.
- Installer un serveur avec le service HaProxy.
- Mettre en place une réplication entre les serveurs LAMP avec DRBD.

I. Installation des serveurs LAMP :

LAMP signifie « **L**inux **A**pache **M**ySQL **P**hp ».

Nous allons créer deux machines virtuelles qui utiliseront le système d'exploitation Linux.

Image de l'OS utilisée :

Distribution	Version	Nom machine
Debian	8.3	LAMP1
Debian	8.3	LAMP2

Les deux serveurs se nommeront respectivement « LAMP1 » & « LAMP2 » :

```
root@LAMP1:~# hostname LAMP1_
```

```
root@LAMP2:~# hostname LAMP2_
```

La commande « *nano /etc/network/interfaces* » nous permet d'accéder et de modifier les paramètres réseaux d'une machine Linux.

Les configurations réseau des deux serveurs seront comme ci-dessous :

Serveur LAMP1 :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.130
netmask 255.255.255.0
gateway 192.168.1.254
```

Serveur LAMP2 :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.131
netmask 255.255.255.0
gateway 192.168.1.254
```

Maintenant que les deux machines possèdent une configuration réseau, nous pouvons commencer à télécharger du contenu.

Nous commençons par mettre à jour les services et paquets présents sur notre distribution :

```
root@LAMP2:~# apt-get update_
```

Cette commande est à faire sur les deux machines.

Nous allons ensuite commencer par installer les services nécessaires au serveur LAMP.

Pour installer le service apache, il faut taper la commande suivante :

```
root@LAMP2:~# apt-get install apache2_
```

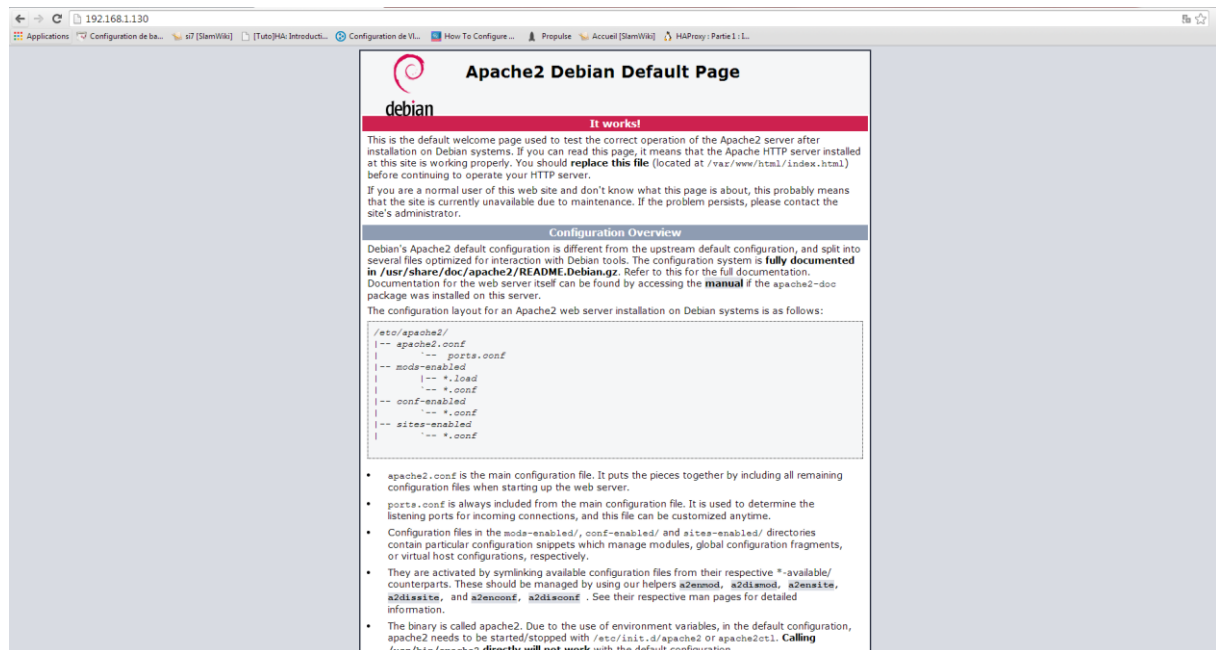
Pour installer le service MySQL, il faut taper la commande suivante :

```
root@LAMP2:~# apt-get install mysql-server_
```

Pour installer le service PHP, il faut taper la commande suivante :

```
root@LAMP2:~# apt-get install php5_
```

Une fois ces commandes tapées, les services nécessaires au serveur LAMP sont installés. La configuration de ces services est pour le moment la configuration par défaut. Par exemple, en tapant l'adresse IP d'un des serveurs dans un navigateur, nous obtiendrons le résultat suivant :



Il s'agit de la page que met le service apache par défaut. Cette page se trouve dans le dossier « `/var/www/html` » des serveurs.

Nous allons la modifier sur chaque serveur afin de différencier les serveurs lorsque l'on tape leur adresse IP dans un navigateur.

Après modifications des pages « index.html » se trouvant dans le dossier « /var/www/html » de chaque serveur, nous obtenons un résultat qui nous permet de les différencier :

Page du serveur LAMP1 :



Page du serveur LAMP2 :



II. Sécurisation d'accès à l'espace personnel d'un utilisateur :

Nous allons d'abord commencer par créer un utilisateur commun aux deux serveurs LAMP :

```
root@LAMP1:~# adduser dorian
```

Nous avons donc créé l'utilisateur « dorian ».

Nous allons ensuite créer au sein du dossier personnel (« home ») de cet utilisateur un dossier nommé « public_html » :

```
root@LAMP1:/home/dorian# mkdir public_html
```

On assigne après un mot de passe qui sera crypté. Pour cela, nous tapons la commande suivante :

```
root@LAMP1:/home/dorian/public_html# htpasswd -c .privpasswd dorian
New password:
Re-type new password:
Adding password for user dorian
```

La commande « *htpasswd -c .privpasswd dorian* » nous demande de taper un mot de passe qui sera crypté dans le fichier « .privpasswd ».

Lorsque l'on ouvre le fichier « .privpasswd », on obtient la ligne suivante :

```
GNU nano 2.2.6          Fichier : .privpasswd
dorian:$apr1$2UyTk2gq$2pGSZ1aIQoWA1RNOpayPa0
```

Toujours dans le dossier « public_html », nous allons créer un fichier qui permettra de sécuriser l'espace personnel. Le fichier s'appellera « .htaccess » et contiendra les lignes suivantes :

```
root@LAMP1:/home/dorian/public_html# touch .htaccess
```

```
GNU nano 2.2.6          Fichier : .htaccess
AuthType Basic
AuthName "Bonjour, Entrez vos identifiants de connexion : "
AuthUserFile /home/dorian/public_html/.privpasswd
Require valid-user
```

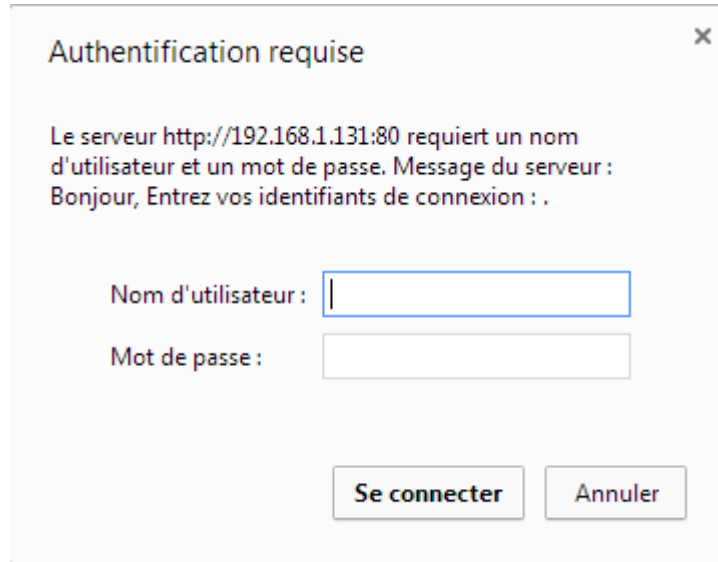
Nous allons maintenant activer la publication de documents via la commande suivante :

```
root@LAMP1:/home/dorian/public_html# a2enmod userdir
Module userdir already enabled
```

Il est nécessaire de redémarrer le service apache :

```
root@LAMP1:/home/dorian/public_html# service apache2 restart_
```

On teste ensuite l'accès par un navigateur avec l'adresse <http://192.168.1.130/~dorian> pour accéder à l'espace personnel de l'utilisateur « dorian » :



Authentification requise

Le serveur <http://192.168.1.131:80> requiert un nom d'utilisateur et un mot de passe. Message du serveur : Bonjour, Entrez vos identifiants de connexion : .

Nom d'utilisateur :

Mot de passe :

Nous indiquons les identifiants que nous avons enregistré lors de la commande « `htpasswd -c .privpasswd dorian` » et nous obtenons ensuite ce résultat :

Index of /~dorian

Name	Last modified	Size	Description
----------------------	-------------------------------	----------------------	-----------------------------



[Parent Directory](#)

-



Apache/2.4.10 (Debian) Server at 192.168.1.131 Port 80

On constate qu'il n'y a aucun fichier car nous n'en avons pas encore créé dans le répertoire personnel de l'utilisateur. Nous allons donc créer une page web que nous allons appeler « test.html » :

```
root@LAMP1:/home/dorian/public_html# nano test.html_
```

Le résultat sur le navigateur devient différent, on voit désormais affiché la page que nous avons créé précédemment et nous y avons accès :

Index of /~dorian

Name	Last modified	Size	Description
 Parent Directory			-
 test.html	2016-04-25 11:23	54	

Apache/2.4.10 (Debian) Server at 192.168.1.130 Port 80



On a pu constater que l'accès aux fichiers situés dans un espace personnel était désormais sécurisé par un mot de passe. Nous pouvons donc maintenant commencer à installer le service HaProxy.

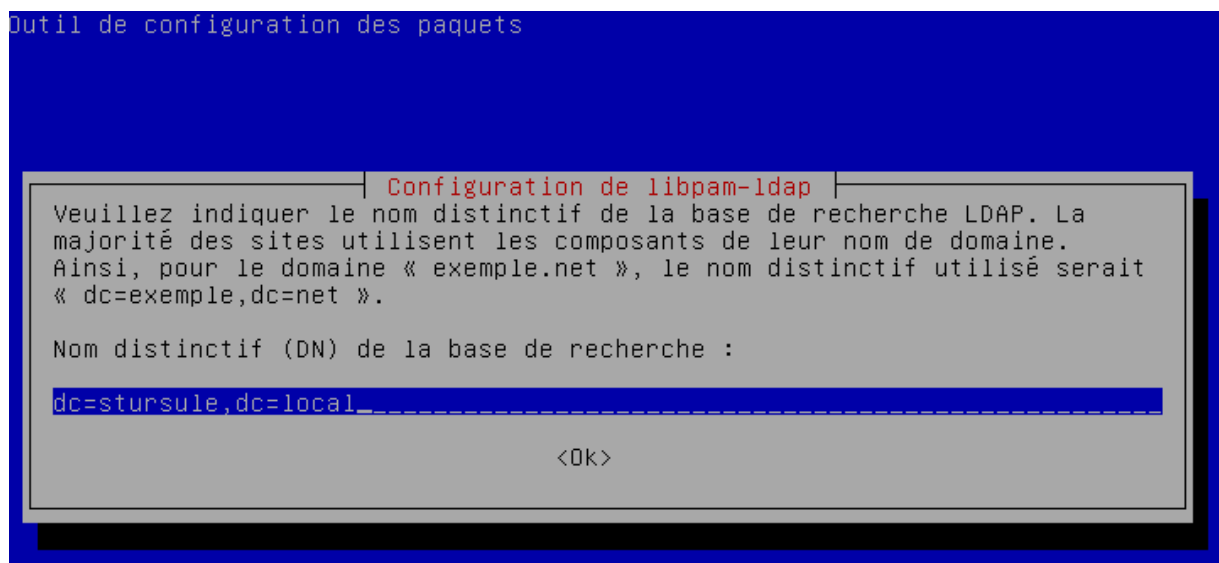
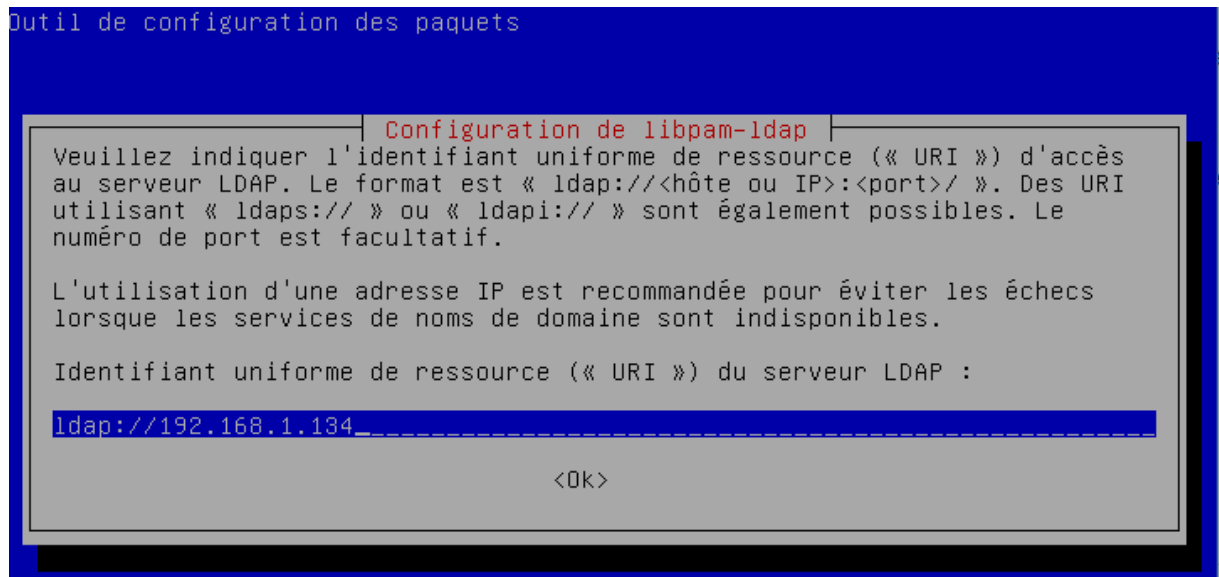
III. Connexion avec Pam LDAP à l'annuaire :

Nous allons mettre en place une connexion entre le serveur LAMP et un annuaire afin de permettre à des utilisateurs possédant un compte de pouvoir se connecter et accéder à leurs pages internet s'ils en possèdent.

On commence par installer les paquets nécessaires à l'installation du service *PAM_LDAP* sur les serveurs LAMP.

```
root@LAMP2:~# apt-get install libpam-ldap libpam0g libldap-2.4-2 libpam-cracklib
```

La configuration que nous allons appliquer à ce service est la suivante :



Nous pouvons retrouver ces paramètres et les modifier dans le fichier
« /etc/pam_ldap.conf ».

Les fichiers de configuration du service *PAM_LDAP* se trouvent dans le chemin
« /etc/pam.d ».

```
root@LAMP2:/etc/pam.d# ls
atd                common-auth        login              runuser-1
chfn               common-password    newusers          sshd
chpasswd           common-session     other             su
chsh               common-session-noninteractive  passwd           systemd-user
common-account     cron               runuser
```

Nous allons maintenant configurer certains de ces fichiers tel quel :

« /etc/pam.d/common-account » :

```
account requisite          pam_deny.so
account required          pam_unix.so
account sufficient        pam_ldap.so
```

« /etc/pam.d/common-auth » :

```
auth    requisite          pam_deny.so
auth    required          pam_unix.so nullok_secure use_first_pass
auth    sufficient        pam_ldap.so
```

« /etc/pam.d/common-password » :

```
password    requisite          pam_cracklib.so retry=3 minlen=$
password    [success=2 default=ignore] pam_unix.so obscure use_authok$
password    [success=1 user_unknown=ignore default=die] pam_ldap.so use$
# here's the fallback if no module succeeds
password    requisite          pam_deny.so
password    required          pam_unix.so nullok obscure min=$
password    sufficient        pam_ldap.so_
```

« /etc/pam.d/common-session » :

```
session requisite          pam_deny.so
session required          pam_unix.so
# and here are more per-package modules (the "Additional" block)
session required          pam_mkhomedir.so skel=/etc/skel/_
session optional          pam_ldap.so
```

Il nous est nécessaire aussi de modifier le fichier se trouvant dans le chemin « /etc/nsswitch.conf » pour qu'il ressemble à ceci :

```
passwd:      files  compat  ldap
group:       files  compat  ldap
shadow:     files  compat  ldap
gshadow:     files
-
hosts:       files  dns
networks:    files

protocols:   db  files
services:    db  files
ethers:      db  files
rpc:         db  files

netgroup:    nis
```

Une fois la configuration terminée, nous pouvons redémarrer les machines afin d'appliquer les modifications et de tester la connexion à l'annuaire.

Lorsque l'on tente de se connecter en tant que « *root* » dorénavant, on obtient une demande de mot de passe ldap :

```
LAMP2 login: root
Password:
LDAP Password: _
```

IV. Installation du serveur HaProxy :

Nous allons installer ce service sur un autre serveur.

Image de l'OS utilisée :

Distribution	Version	Nom Machine
Debian	8.3	HaProxy

Nous allons commencer par assigner les paramètres réseaux à la machine comme ceci :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.132
netmask 255.255.255.0
gateway 192.168.1.254
```

Afin de pouvoir télécharger les paquets nécessaires à l'installation de HaProxy, nous allons devoir ajouter une ligne dans le fichier « */etc/apt/sources.list* » :

```
# Pour HaProxy
deb http://ftp.debian.org/debian/ wheezy-backports main
```

Une fois cette ligne ajoutée, nous pouvons lancer la commande d'installation du service HaProxy :

```
root@HaProxy:/# apt-get install haproxy_
```

Le fichier de configuration de HaProxy se trouve dans « `/etc/haproxy/haproxy.cfg` » et ressemble à ceci :

```
GNU nano 2.2.6      Fichier : /etc/haproxy/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:EC$
    ssl-default-bind-options no-sslv3

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout connect 5000
    timeout client 50000
    timeout server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http
```

Nous allons devoir ajouter des lignes dans ce fichier afin de le configurer selon notre besoin, celui de faire de la répartition de charge sur les serveurs LAMP que nous avons créé au préalable. Nous allons devoir ajouter les lignes suivantes :

- [*Listen Serveurs_LAMP 192.168.1.132 :80*](#)

Cette ligne permet d'indiquer sur quelle adresse IP la machine utilisant HaProxy va fonctionner ainsi que son port d'écoute (port 80 : http). Nous pourrions le configurer avec un port tcp si nous utiliserions du contenu tel que du mysql.

- [*Mode http*](#)

Cela permet de spécifier que le balancement de charge se fait sur du contenu web http

- *Balance roundrobin*

La méthode round-robin correspond à la répartition équitable de la charge entre les serveurs d'un cluster.

- *Server LAMP1 192.168.1.130:80 check*
- *Server LAMP2 192.168.1.131 :80 check*

Permet de déclarer les différents serveurs web qui vont être utilisés pour la répartition de charge.

- *Stats ...*

Les lignes commençant par « stats » permettent de configurer la page de statistiques de HaProxy. Nous allons configurer ces lignes afin d'avoir accès à la page de statistiques via l'adresse <http://192.168.1.132/stats> avec les identifiants root/root.

Une fois la page de configuration haproxy.cfg modifiée, on obtient ceci en ajout :

```
listen Serveurs_LAMP 192.168.1.132:80

    mode http

    balance roundrobin

    server LAMP1 192.168.1.130:80 check
    server LAMP2 192.168.1.131:80 check

    stats enable
    stats hide-version
    stats refresh 30s
    stats show-node
    stats auth root:root
    stats uri /stats
```

Pour démarrer HaProxy, il faut taper la commande suivante :

```
root@HaProxy:/# /etc/init.d/haproxy start_
```

Nous pouvons désormais accéder à la page de statistiques à l'adresse <http://192.168.1.132/stats> :

HAProxy
Statistics Report for pid 14556 on HaProxy

> **General process information**

pid = 14556 (process #1, nproc = 1)
 uptime = 0d 0h24m37s
 system limits: memmax = unlimited; ulimit-n = 4033
 maxsock = 4033; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0; conn rate = 0/sec
 Running tasks: 1/0; idle = 100 %

Legend:
 active UP (green)
 active UP, going down (yellow)
 active DOWN, going up (orange)
 active or backup DOWN (red)
 active or backup DOWN for maintenance (MAINT) (brown)
 active or backup SOFT STOPPED for maintenance (grey)
 backup UP (blue)
 backup UP, going down (light blue)
 backup DOWN, going up (purple)
 not checked (grey)

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

	Queue		Session rate			Sessions			Bytes		Denied		Errors		Warnings		Status	LastChk	Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp			Req	Conn	Resp	Retr	Redis	Wght	Act	Bck	Chk	Dwn	Dwntime
Frontend	0	2	-	1	3	2 000	8			88 101	700 571	0	0	4						OPEN									
LAMP1	0	0	-	0	15	0	1	-	64	64	1m11s	32 103	18 368	0	0	0	0	0	0	1m44s UP	L4OK in 0ms	1	Y	-	3	1	3m16s	-	
LAMP2	0	0	-	0	15	0	2	-	66	66	1m11s	33 311	18 616	0	0	0	0	0	0	24m37s UP	L4OK in 0ms	1	Y	-	0	0	0s	-	
Backend	0	0	-	0	30	0	2	200	130	130	0s	88 101	700 571	0	0	0	0	0	0	24m37s UP		2	2	0		0	0s		

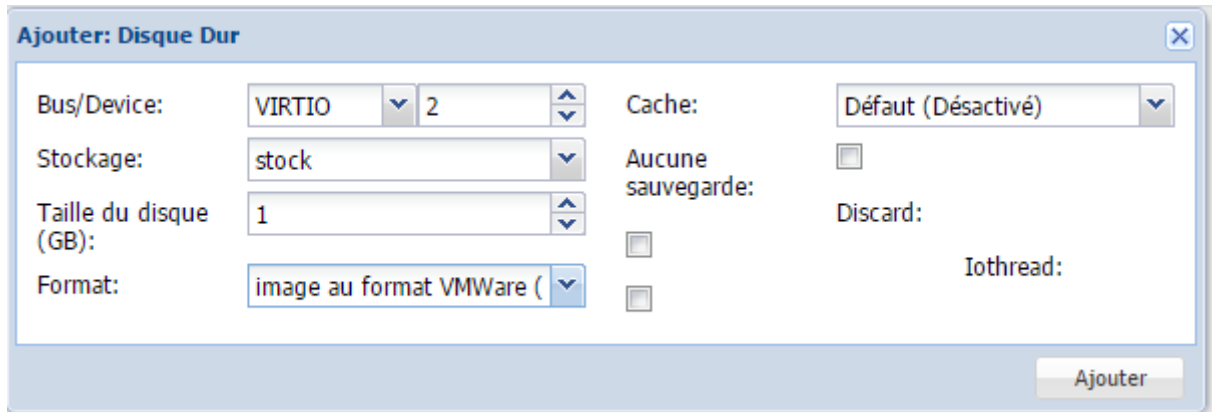
On constate que les lignes qui affichent les statistiques sur LAMP1 & LAMP2 sont en vert ce qui signifie que les serveurs sont opérationnels. Lorsque l'on tape l'adresse 192.168.1.132 sur un navigateur, on obtient l'affichage de la page web d'apache d'un des serveurs :

Ceci est la page du serveur LAMP1

Il s'agit de la page du serveur LAMP1 ou LAMP2 selon la charge sur l'un ou l'autre serveur. Nous avons configuré HaProxy pour envoyer les demandes vers le serveur ayant la charge la plus basse.

V. Mise en place de la réplication avec « DRBD » :

Nous allons commencer par ajouter un disque de 1GB dans proxmox sur chacun des serveurs LAMP que nous avons créés dans le but de les utiliser exclusivement pour notre réplication.



Une fois les disques ajoutés, nous pouvons commencer à créer les partitions. Cette manipulation est à faire sur les deux serveurs :

```
root@LAMP1:/# fdisk /dev/sdb
```

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-130, default 130):
Using default value 130

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

On garde les valeurs par défaut concernant la taille des partitions.

Nous installons ensuite les paquets nécessaires à l'utilisation de DRBD avec la commande « *apt-get install drbd8-utils* ».

Et nous activons ensuite le paquet :

```
root@LAMP1:/# modprobe drbd
```


Toujours sur les deux serveurs, nous allons créer un fichier dans « /etc/drbd.d » que nous appellerons « drbd1.res » :

```
root@LAMP1:~# nano /etc/drbd.d/drbd1.res
```

Nous configurons ensuite ce fichier de la façon suivante :

```
GNU nano 2.2.6 Fichier : /etc/drbd.d/drbd1.res
resource r0 {
    syncer {
        rate 10M;
    }

    on LAMP1 {
        device /dev/drbd0;
        disk /dev/sdb1;
        address 192.168.1.130:7788;
        meta-disk internal;
    }

    on LAMP2 {
        device /dev/drbd0;
        disk /dev/sdb1;
        address 192.168.1.131:7788;
        meta-disk internal;
    }
}
```

Ensuite, nous tapons les commandes suivantes :

```
root@LAMP1:~# drbdadm create-md r0_
```

```
root@LAMP1:~# drbdadm up r0_
```

Nous venons d'exécuter la mise en œuvre de « r0 ».

Nous pouvons vérifier que les serveurs se contactent avec la commande « *drbd-overview* » :

```
root@LAMP1:~# drbd-overview
0:r0/0 Connected Secondary/Secondary UpToDate/Diskless
```

```
root@LAMP2:~# drbd-overview
0:r0/0 Connected Secondary/Secondary Inconsistent/Diskless
```

Nous pouvons constater avec la capture d'écran ci-dessus que nos serveurs sont en mode « secondary/secondary », nous allons donc passer le serveur LAMP1 en « primary » et le serveur LAMP2 en « secondary » :

```
root@LAMP1:~# drbdadm -- --overwrite-data-of-peer primary r0
```

```
root@LAMP2:~# drbdadm secondary r0
```

On peut vérifier que le mode « primary/secondary » a bien été mis en place avec la commande « *drbd-overview* » :

```
root@LAMP1:~# drbd-overview
0:r0/0 Connected Primary/Secondary UpToDate/ UpToDate.
```

La synchronisation des fichiers est en cours et l'on peut vérifier l'état de cette synchronisation avec la commande « *cat /proc/drbd* » :

```
root@LAMP1:~# cat /proc/drbd
version: 8.4.3 (api:1/proto:86-101)
srcversion: 1A9F77B1CA5FF92235C2213
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:54572 nr:0 dw:4 dr:58549 al:1 bm:7 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:1696
```

Notre raid réseau local est désormais fonctionnel et il nous est donc nécessaire de créer un système de fichiers ext4 afin de pouvoir écrire dessus. Nous allons donc taper la commande suivante :

```
root@LAMP1:/etc/drbd.d# mkfs.ext4 /dev/drbd0
mke2fs 1.42.12 (29-Aug-2014)
En train de créer un système de fichiers avec 261871 4k blocs et 65536 i-noeuds.
UUID de système de fichiers=4a48c01e-ea82-4538-a0de-44214610218a
Superblocs de secours stockés sur les blocs :
    32768, 98304, 163840, 229376

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (4096 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété
```

Et nous pouvons désormais monter notre disque comme n'importe quel disque dur :

```
root@LAMP1:~# mkdir /mnt/r0_
```

```
root@LAMP1:~# mount /dev/drbd0 /mnt/r0_
```

La commande `df -h` nous permet de vérifier si le disque est bien monté :

```
root@LAMP1:~# df -h
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur
/dev/sda1          2,9G  1,2G  1,5G  45% /
udev              10M    0    10M   0% /dev
tmpfs             150M   4,4M  145M   3% /run
tmpfs             374M    0   374M   0% /dev/shm
tmpfs             5,0M   4,0K   5,0M   1% /run/lock
tmpfs             374M    0   374M   0% /sys/fs/cgroup
/dev/sda6         6,5G   16M   6,2G   1% /home
/dev/drbd0       991M   1,3M   923M   1% /mnt/r0
```

Nous pouvons constater que lorsque nous plaçons un fichier dans le dossier « `/mnt/r0` » du serveur LAMP1 par exemple, le fichier se retrouve dans le même dossier sur le serveur LAMP2.

Exemple : Nous créons un fichier « `test` » dans le dossier avec inscrit « `hello world !` » sur le serveur LAMP2.

```
root@LAMP2:/mnt/r0# nano test_
GNU nano 2.2.6 Fichier : test
hello world !
```

La réplication se fait et l'on retrouve sur le serveur LAMP1 dans le même dossier ce même fichier :

```
root@LAMP1:/mnt/r0# ls
lost+found test
```

Conclusion :

- Les serveurs LAMP sont correctement configurés et fonctionnels.
- L'accès sécurisé des utilisateurs est mis en place.
- L'accès à l'annuaire via PAM_LDAP est configuré mais présente quelques problèmes lors de la tentative de connexion avec des comptes utilisateurs.
- Le serveur HaProxy est configuré et fonctionnel.
- La réplication DRBD est effective entre les serveurs LAMP.