



Pôle Systèmes et Réseaux (C3)

S3.A.01 – Développement d’une application

Tuteur : Pr. Richard Chbeir

Projet : Hego Lagunak, une application de parrainage pour le BDE

Équipe 3 : BRIERRE Titouan (TP1), DARGAZANLI Nicolas (TP1), ERREZARET Leho (TP2) et MAURICE Alexandre (TP1), en BUT Informatique, 2022, Semestre 3, Parcours A.

1^{er} jalon : analyse

1. Expression des besoins

Dans le cadre de cette SAE nous devons mettre en place toute une infrastructure réseau permettant la mise en service de notre application.

Notre application doit pouvoir afficher des pages web, réaliser des requêtes entre les clients et l'application, appliquer des traitements à des informations relatives aux clients, répondre à des contraintes de sécurité et éventuellement disposer de sauvegardes en cas de problème.

2. Les différents services

Un serveur Apache / NGINX : serveurs Web destinés à afficher les pages, avec PHP installé pour exécuter les scripts PHP.

Un serveur MariaDB / MySQL / Oracle : serveur de base de données pour stocker les informations nécessaires (utilisateurs, formulaires etc.).

Un pare-feu : pour faire respecter la politique de sécurité du réseau.

Un répartiteur de charge : éventuellement, pour rendre le traitement global plus efficace.

Un serveur de clonage de données : pour sauvegarder les données utilisées par l'application.

3. Les machines

Chaque service aura une machine dédiée, mais nous devons prendre en compte l'existence une machine administrateur et une machine client (à l'extérieur). Faire ce choix nous permet d'avoir plus de sécurité, sans « planter » toute l'application si une machine ne fonctionne plus. De plus cela empêchera certains serveurs d'avoir un accès vers le réseau internet (en particulier le SGBD).

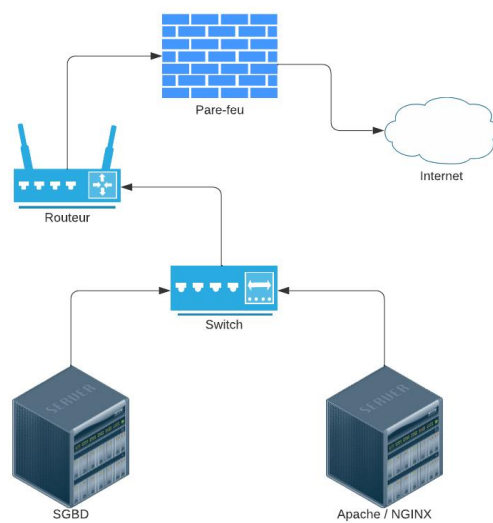
4. Réseaux

Nous n'aurons besoin que d'un seul réseau qui comprendra l'ensemble des serveurs avec une seule passerelle vers le monde extérieur. Nous ne considérons pas les machines de test client comme appartenant à notre réseau.

En vue du nombre maximum d'utilisateurs (environ 200 utilisateurs avec une fréquence d'utilisation basse sur une période de 1 mois), nous ne nous servirons probablement pas d'un répartiteur de charge.

5. Plan d'adressage

Nous avons choisi un sous-réseau de classe C : 192.168.27.0/24.



2^{ème} jalon : création de l'infrastructure réseau

Les clients

Nous devons paramétrer les deux clients pour qu'ils récupèrent leur adresse IP via le protocole DHCP.

Pour cela, nous ajoutons le paramétrage de l'interface eth0, comme nous le montrons sur la capture ci-dessous.

```
GNU nano 2.2.6      File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Le serveur DHCP

Le serveur DHCP permettra de desservir des adresses IP ainsi que le routage par défaut.

Nous définissons de manière locale les adresses IP.

eth0 : 192.168.0.252/24

eth1 : 192.168.1.252/24

Nous paramétrons le service DHCP pour qu'il écoute sur l'interface eth0, grâce à la ligne `INTERFACES="eth0"`.

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"
```

Nous allons maintenant configurer le réseau que le serveur DHCP va redistribuer.

```
GNU nano 2.2.6      File: /etc/dhcp/dhcpd.conf

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.1 192.168.0.250;
    option routers 192.168.0.254;
    option broadcast-address 192.168.0.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Les routeurs

Routeur1 :

On attribue les dernières IP des réseaux disponibles et nous pointons le Routeur2 comme routeur par défaut.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

ifconfig eth1 192.168.1.254/24
ifconfig eth0 192.168.0.254/24
route add default gw 192.168.1.1
echo 1 > /proc/sys/net/ipv4/ip_forward
exit 0
```

Routeur2 :

On attribue les premières IP des réseaux disponibles et nous pointons la passerelle comme routeur par défaut. Etant donné que le Routeur2 est le routeur par défaut des serveurs, il faut indiquer le routage vers le réseau LAN1, ce qui engendre une redirection de Routeur2 vers Routeur1.

```
GNU nano 2.2.6      File: /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.1.1/24
ifconfig eth1 192.168.255.1/24
route add default gw 192.168.255.2
route add -net 192.168.0.0/24 gw 192.168.1.254
echo 1 > /proc/sys/net/ipv4/ip_forward
exit 0
```

Serveur de Base de Données

Nous faisons en sorte à ce que le serveur de Base de Données ne puisse communiquer avec personne sauf le serveur Web, pour des raisons de sécurité.

Nous ajoutons aussi une exception pour 127.0.0.1 dans le cas où le ServeurBD devrait faire une requête à lui-même, ce qui ne crée pas de failles de sécurité.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.1.20/24
iptables -P INPUT DROP
iptables -I INPUT -s 192.168.1.10 -j ACCEPT
iptables -I INPUT -s 127.0.0.1 -j ACCEPT
exit 0
```

Serveur Web

Le serveur Web communiquera par défaut avec le Routeur2. Pour la sécurité du serveur Web, nous devrions bloquer l'accès aux ports non-utilisés, en gardant simplement les ports 20,21,22,80,443, mais nous rencontrons actuellement quelques problèmes car le serveur Web ne sera plus capable de faire des requêtes extérieures vers lparla par exemple, car ces ports-là sont « aléatoires » et ne peuvent pas être débloqués en conséquence sans paramétrage avancé.

```
GNU nano 2.2.6      File: /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.1.10/24
route add default gw 192.168.1.1
exit 0
```