

TP 3 Configuration d'un service web

Objectifs du TP

L'objectif du TP est de comprendre les principes de fonctionnement des services HTTP (service web). Notamment la configuration des *Virtual Host*.

Le principe des serveurs virtuels consiste à faire fonctionner un ou plusieurs serveurs web (p. ex. `www.youpi.tralala.com` et `www.pokemon.go.org`) sur un seul serveur web physique. Ces serveurs virtuels peuvent être configurés de deux manières : en mode "par-IP", où chaque serveur web possède une adresse IP dédiée, ou en mode "par-nom", où plusieurs noms de domaine coexistent sur une même adresse IP. L'utilisateur final ne remarque pas qu'en réalité, tous ces sites sont hébergés sur un seul serveur physique.

Les *virtual hosts* sont décrits dans les fichiers `.conf` du répertoire `sites-available/`. Par exemple, la configuration suivante décrit la configuration de deux sites web sur un même serveur.

```
<VirtualHost *:80>
    ServerName www.youpi.tralala.com
    ServerAlias youpi.tralala.com
    DocumentRoot "/www/youpi"
</VirtualHost>

<VirtualHost *:80>
    ServerName www.pokemon.go.org
    DocumentRoot "/www/pokemon"
</VirtualHost>
```

Pour toute question sur la configuration, consultez la [documentation apache](#).

1 Mise en œuvre et configuration d'un service HTTP

La mise en œuvre du service web se fera à l'aide de deux machines virtuelles (voir TP précédent).

1.1 Préparation des sites sur la VM serveur

1. Démarrer la machine virtuelle serveur et démarrer le service `udhcpd` (voir TP précédent)
2. Avec les droits `root`, dans `/var/www/`, créer 2 répertoires nommés « site1 » et « site2 ».
3. Dans chacun de ces répertoires, créer des fichiers texte simples avec des noms différents.
4. Les permissions des fichiers doivent être telles que Apache, qui tourne sous l'identité `www-data`, soit autorisé à les lire. Cela se résume donc à des permissions `"r-x"` sur les répertoires et des permissions `"r--"` sur les fichiers pour les utilisateurs de la catégorie `"other"`. En principe, ce sont les permissions par défaut, mais il est utile de les vérifier.

1.2 Configuration de Virtual Hosts dans Apache

1. Placez vous dans `/etc/apache2/sites-available/`, et créez une copie du fichier `"000-default.conf"` en la renommant `"site1.conf"`.
2. Dans `"site1.conf"` faire les éditions suivantes :
 - dans la section `VirtualHost`, ajouter la directive `ServerName site1`
 - changer le `DocumentRoot` en `/var/www/site1/`
Il n'est pas nécessaire de faire d'autres modifications.
3. Activer le `site1` avec la commande

```
a2ensite site1
```

4. Vérifier qu'un lien symbolique a bien été créé dans `/etc/apache2/sites-enabled/`
5. Demander à Apache de prendre en compte la nouvelle configuration avec la commande :

Si apache n'est pas démarré (utiliser la commande `ps -aux | grep apache2`)

```
systemctl start apache2
```

Si apache est déjà démarré :

```
systemctl reload apache2
```

6. Faire de même avec `site2`.

1.3 Configuration et tests sur le client

1. Démarrer la machine virtuelle client. La VM client doit avoir une adresse IP obtenue via DHCP. Sinon, configurer son adresse IP "à la main" (avec `ip addr`, cf 1re année de BUT). La VM client doit pouvoir faire un `ping` sur la VM serveur.
2. Éditer le fichier `/etc/hosts` du client et rajouter 2 lignes pour que les noms `site1` et `site2` pointent vers l'adresse IP du serveur.
3. Vérifier avec la commande `ping` que la machine peut accéder aux machines ayant, respectivement, pour nom `site1` et `site2`.
4. Dans firefox, vérifier que les URLs `http://site1/` et `http://site2/` pointent bien chacune vers un site différent.
5. Vers quel site pointe l'URL `http://ADRESSE_IP_DU_SERVEUR/` ?

1.4 Fonctionnement des Virtual Hosts dans HTTP (VM Serveur)

1. Avec wireshark, sur la VM serveur, capturez les requêtes HTTP du navigateur Web vers chacun des 2 sites (utiliser les filtres pour repérer la requête).
2. Quel en-tête HTTP de la requête permet au serveur d'aiguiller la requête vers le bon site ?

2 Gestion d'alias DNS pour chaque site web

Dans cette partie, il s'agit de réaliser la résolution de nom pour les noms `site1` et `site2` avec votre serveur DNS au lieu d'utiliser le fichier `/etc/hosts`.

1. Commenter les deux entrées du fichier `/etc/hosts` que vous avez ajoutées.
2. En reprenant le TP2, augmentez le serveur DNS pour ajouter deux alias (CNAME) `site1` et `site2` au domaine `jaune.fr`. correspondant aux 2 sites web de telle sorte que l'on puisse accéder à ces deux sites avec les URL suivantes `http://site1.jaune.fr` et `http://site2.jaune.fr`. Ces deux alias (`site1` et `site2`) doivent correspondre au serveur `fruits.jaune.fr`.
3. Modifier les configurations Apache pour que les deux sites (directive `ServerName`) correspondent au nom `site1.jaune.fr` et `site2.jaune.fr`.
4. Testez.

3 Configuration de virtual hosts en IPv6

1. À l'aide de la commande `ip` trouvez les adresses IPv6 du client et du serveur.
2. testez un ping IPv6 entre les deux machines.
3. Comme il s'agit d'adresses **EUI**, celles-ci sont en link-local et ne sont pas routables. Par ailleurs, certains navigateurs, tels que Chrome et Firefox n'autorisent pas la navigation sur des liens locaux. C'est pourquoi nous allons configurer à la main une adresse IPv6 globale. Sur le serveur, effectuez :

```
ip -6 addr add 2001:0db8:0:f101::1/64 dev ens3
```

Effectuez la même opération pour que le client obtienne l'adresse `2001:0db8:0:f101::90/64` sur `ens3`.

4. Testez la connexion entre les deux machines en utilisant les adresse IPv6 globales.
5. Depuis le client, testez l'accès au serveur web avec l'URL suivante :
`http://[2001:0db8:0:f101::1]:80`
6. Créez le site `site3` et en vous servant de la **documentation apache**
7. Configurez le serveur DNS pour faire en sorte que le nom `mirabelle.jaune.fr` corresponde à l'adresse IPv6 globale du serveur.
8. Ajoutez l'alias `site3` à `mirabelle.jaune.fr` pour que le serveur web réponde correctement à la requête `site3.jaune.fr`
9. Testez cette configuration avec `nslookup` à partir du client.
10. Modifiez la configuration Apache pour que le virtual host `site3.jaune.fr` ne puisse être accédé qu'en IPv6.
11. Testez.
12. À l'aide du mot-clé `ErrorLog`, modifiez la déclaration de virtualhost de `site3` pour que les erreurs soient écrites dans `/var/log/httpd/error_site3_log`.
13. Testez.

4 Configuration automatique d'adresse IPv6 par SLAAC

Plutôt qu'une attribution manuelle, les adresses IPv6 peuvent être attribuées dynamiquement au sein d'un LAN. Cependant, en IPv6, il existe deux méthodes d'attribution dynamique.

- La configuration automatique des adresses sans état (SLAAC)
- La configuration via le protocole DHCPv6 (avec état)

SLAAC (Stateless Address Autoconfiguration) est un mécanisme d'attribution d'adresses IPv6 qui permet aux hôtes d'un réseau de configurer automatiquement leur adresse IPv6 sans nécessiter l'intervention d'un serveur DHCP.

L'obtention d'une adresse par un client suit les échanges ICMPv6 suivants :

1. Un client envoie un message Router Solicitation (découverte de routeur) pour trouver les routeurs SLAAC présents sur le réseau.
2. De leur côté, les routeurs envoient des messages Router Advertisement (publicité de routeur) qui contiennent des informations sur le réseau, notamment le préfixe de réseau.
3. Le client utilise ce préfixe réseau pour générer une adresse IPv6 en combinant ce préfixe avec une adresse machine générée localement (MAC ou une autre valeur aléatoire).
4. Le client vérifie que cette adresse est unique sur le réseau et dans l'affirmative l'utilise pour communiquer avec les autres machines du réseau.

Le processus SLAAC étant sans état, les informations d'adresse réseau ne sont pas maintenues. Ainsi, contrairement à DHCP, le routeur SLAAC ne peut pas dire quelles sont les adresses IPv6 utilisées ou disponibles. Par contre, SLAAC a l'avantage de ne pas nécessiter de serveur central pour maintenir les adresses et d'être très simple à installer.

Nous allons étudier SLAAC en utilisant l'outil **radvd** (Router Advertisement Daemon).

La mise en œuvre se fera à l'aide des deux machines virtuelles

Sur la VM-Serveur

1. Désactivez l'interface ens3, et constatez l'état de l'interface

```
ip link set down ens3
```

2. Modifiez le fichier (créez-le s'il n'existe pas) **/etc/radvd.conf** pour qu'il contienne la configuration suivante

```
interface ens3
{
    AdvSendAdvert on;
    prefix 2001:db8:0:3::/64 {
    };
};
```

3. Démarrez le service radvd et réactivez l'interface

```
systemctl start radvd
```

```
ip link set up ens3
```

4. Après quelques secondes, vérifiez l'état de l'interface. Que s'est-il passé ?

Sur la VM-Client

1. vérifiez le nombre d'adresses IPv6 disponibles sur ens3.

2. Désactivez l'interface ens3, et constatez l'état de l'interface

```
ip link set down ens3
```

3. Sur le **serveur**, lancez une écoute Wireshark sur ens3
4. Activez l'interface ens3

```
ip link set up ens3
```

5. Après quelques secondes vérifiez l'état de l'interface. Que s'est-il passé ?

Analyse des trames

1. Observez le message RS et indiquez type de message ICMPv6, l'adresse source et destination. Quel est le type de chacune des adresses ?
2. Observez le message RA et indiquez type de message ICMPv6, l'adresse source et destination. Quel est le type de chacune des adresses ? Quelles informations sont communiquées ? Pour quelle durée de vie ?
3. Comment le client s'assure-t-il que son adresse est unique ? Repérez les trames sur Wireshark qui correspondent à cette recherche.

Sur la VM-Serveur

1. Modifiez le fichier de configuration pour utiliser le prefix 2001:db8:64::/64 et redemarrez le service radvd
2. Vérifiez, l'effet sur le client.