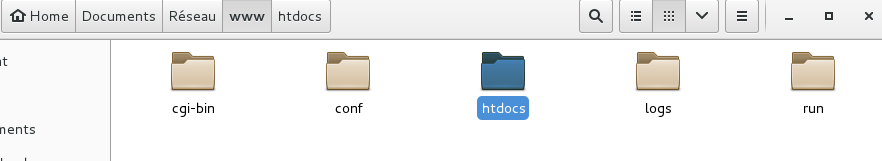
**TP 3 :**

Le but de ce TP est de créer un serveur HTTP !

Pour ça on part d’un modèle tout fait en décompressant une archive dans un dossier nommé “www”. On va ensuite utiliser Apache (à travers des commandes) pour configurer le serveur.



* htdocs contient juste les fichiers du site web (les pages elles-mêmes codées en html, et les images utilisées par ces pages html)
* conf contient simplement le fichier ( httpd.conf) qui sert à configurer notre serveur. Par exemple, la première chose à faire d’ouvrir ce fichier et de modifier le port d’écoute (Listen 80) par un port autorisé (8080 par exemple) (petit rappel sur les ports grâce à wikipedia [ici](https://fr.wikipedia.org/wiki/Port_(logiciel)) )
* logs contient des fichiers textes avec l’historique de toutes les connexions à notre serveur
* cgi-bin contient des scripts CGI

I) Lancement du serveur avec Apache

On commence par lancer le serveur, qui tournera alors en arrière plan, avec :

**httpd -f ~/www/conf/httpd.conf**

Et pour le quitter on fait : **killall httpd** (pour le redémarrer on combine donc ces 2 commandes,)

Maintenant que le serveur est lancé, on peut y accéder avec un navigateur :

<http://localhost:8080/index.html> (car la page qu’on veut ouvrir est index.html) ou bien en tapant le nom de son pc : [http://ENSIPCXXX:8080/index.html](http://localhost:8080/index.html) ou bien avec son IP local :

http://129.88.242.218:8080/index.html

Il faut juste bien penser à indiquer le numéro du port d’écoute sur le fichier conf (ici 8080). (entre 1 et 1024 ils sont tous attribués)

II) Scripts CGI

Un script CGI (Common Gateway Interface) permet juste au serveur HTTP de lancer des exécutables externes.

**http://localhost:8080/cgi-bin/test?plein+de+parametres**

Ici on ajoute les paramètres du script test du répertoire cgi-bin en les plaçant après le “?” et en les séparant par des + (comme une recherche google en fait https://www.google.fr/search**?q=ceci+est+un+test**)

Dans le TP, il y a deux fonctions ping qui utilisent des scripts CGI avec deux méthodes différente GET (ajoute les données à l'URL en utilisant l’entrée standard) et POST (utilise une variable) expliquées [ici](https://www.xul.fr/ecmascript/get-post.php).

III) Hôtes virtuels

Ici, on veut juste héberger PLUSIEURS sites sur le MEME serveur ! Il faut donc commencer par dupliquer le dossier htdocs et renommer la nouvelle copie comme l’on veut ! Ici on va l’appeler ensipcXXX. Il suffit ensuite de modifier le fichier conf comme indiquer dans le TP et ainsi les URL suivantes : <http://localhost:8080/index.html> et [http://ENSIPCXXX:8080/index.html](http://localhost:8080/index.html)

ne mènent plus vers le même site internet (modifier le fichier HTML dans le dossier ensicpcXXX pour s’en rendre compte). C’est maintenant le serveur qui gère les 2 différents hôtes.

IV) URL et systèmes de fichier

Les redirections :

En ajoutant cette commande dans conf :

**Redirect /ensiwikiFAQ http://ensiwiki.ensimag.fr/index.php/FAQ**

au fichier httpd.conf, l’accès à l’adresse http://localhost:8080/ensiwikiFAQ est créé une redirection vers <http://ensiwiki.ensimag.fr/index.php/FAQ>

Alias :

On crée un Alias de cette façon :

Alias /logfiles "${HOME}/www/logs"

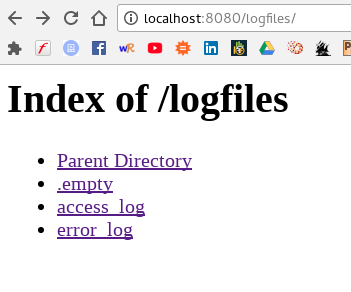
<Directory "${HOME}/www/logs">

Require all granted

Options Indexes

</Directory>

Alias [chemin URL] chemin fichier|chemin répertoire

Un alias permet simplement de rendre un dossier accessible en local grâce à une url comme on peut le voir ci-dessous.

Réécriture :

**RewriteEngine on**

**RewriteRule ^/?fping/([^/]\*) /cgi-bin/fping?$1 [L,PT]**

Après cette réécriture dans le fichier conf, l’url <http://localhost:8080/fping/google.fr/>

La réécriture d’URL permet de renommer des URL plus de manière plus propre et facilement indexable par les robots Google. <http://ensipc525>

V) Authentification

Pour créer un système d’authentification, entrer simplement :

**htpasswd -c pass.txt nom\_utilisateur**

En utilisant Wireshark, on peut lire facilement le mdp dans le paquet HTTP : il faut utiliser du HTTPS

VI) HTTPS

Dans ~/www/conf/, générez la clé de chiffrement privée de votre site à l’aide de la commande suivante :

**/usr/bin/openssl genrsa > privkey.pem**

Générez une requête de certificat à l’aide de la commande suivante :

**/usr/bin/openssl req -new -key privkey.pem -out cert.csr**