Apache

Table des matières

1 favicon

Placer le fichier *favicon.ico* là où pointe la directive `DocumentRoot`. Recharger la configuration du serveur apache puis relancer le navigeteur.

```
$ giftopnm logo.gif |
  pnmscale -width=16 -height=16 |
  ppmtowinicon -output favicon.ico
```

2 Http + Https

Plutôt que d'utiliser apache-ssl on peut utiliser les virtuals hosts :

- Activation des modules SSL

  ```
  # cd /etc/apache2/mods-enabled
  # ln -s ../mods-available/ssl.load .
  # ln -s ../mods-available/ssl.conf .
  ```

- fichier */etc/apache2/ports.conf*

  ```
  ServerName narval.tk
  Listen 80
  Listen 443
  ```

- fichier */etc/apache2/mods-enabled/rewrite.load*

  ```
  LoadModule rewrite_module /usr/lib/apache2/modules/mod_rewrite.so
  #RewriteLog /etc/apache2/rewriteLog.log
  #RewriteLogLevel 9
  ```

- fichier */etc/apache2/sites-enables/000-default*

  ```
  # ln -s ../site-availables/default 000-default
  ```

  modifier à la main :

```
<VirtualHost *:80>
  ServerAdmin root@narval.tk   # <=
  DocumentRoot /var/www
    <Directory />
      Options FollowSymLinks
      AllowOverride None       # <= this must be changed to allow htaccess
    </Directory>
...
  # redirection sur le port HTTPS pour certains sites
  RewriteEngine On
  RewriteCond %{SERVER_PORT} !^443$
  RewriteCond %{REQUEST_URI} ^/(omegai|prologuei)(.*)
  RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI} [QSA,R=301,L]
</VirtualHost>
```

- fichier */etc/apache2/sites-enables/001-default-ssl*

```
# ln -s ../site-availables/default-ssl 001-default-ssl
```

- fichier */etc/apache2/sites-availables/narvali*

```
Alias /narvali /home/intranet-latex/htdocs/narvali
<Directory /home/intranet-latex/htdocs/narvali>
        AllowOverride None
</Directory>

<Directory /home/intranet-latex/htdocs/narvali/admin>
        AuthType Basic
        AuthName "Restricted Files"
        AuthUserFile /etc/apache2/narvali.passwd
        Require valid-user

        AllowOverride None
</Directory>
```

- fichier */etc/apache2/sites-availables/omegai*

```
Alias /omegai /home/intranet-latex/htdocs/omegai
<Directory /home/intranet-latex/htdocs/omegai>

  SSLRequireSSL

  AuthType Basic              # http acces will result on: ‘‘Forbidden’’
  AuthName "Restricted Files"
  AuthUserFile /etc/apache2/omegai.passwd
  Require valid-user

  AllowOverride None
  #order deny,allow
  #allow from all
</Directory>
```

3 webDAV : .ics et syncPlaces

WebDAV (Web-based Distributed Authoring and Versioning) est un protocole (plus précisément, une extension du protocole HTTP).
iCalendar est une norme (RFC 2445) pour les échanges de données de calendrier. Cette norme est aussi connue sous le nom d'iCal.
syncPlaces permet de synchroniser ses marques pages.

- Activation des modules dav

```
# cd /etc/apache2/mods-enabled
# ln -s ../mods-available/dav.load .
# ln -s ../mods-available/dav_fs.load .
# ln -s ../mods-available/dav_fs.conf .
```

- Création du fichier lock

```
# cd /var/lock/apache2
# touch DAVLock
# chown www-data: DAVLock
```

- Création des couples user/mot de passe.

```
# cd /etc/apache2
# htpasswd -c /etc/apache2/webdav.passwd xxx
# htpasswd /etc/apache2/webdav.passwd yyy
```

- Création du dossier contenant les calendriers

```
# cd /var/www
# mkdir -p DAVdocs/ics
# chown -R www-data: DAVdocs
# chmod 755 -R DAVdocs
```

- fichier */etc/apache2/sites-availables/webDAV*

```
DAVMinTimeout 600
DAVDepthInfinity On

Alias /DAVdocs /var/www/DAVdocs
Alias /ics /var/www/DAVdocs/ics
Alias /bm /var/www/DAVdocs/bm
<Directory /var/www/DAVdocs/>
      DAV on
      AuthName "WebDAV Storage"
      AuthType Basic
      AuthUserFile /etc/apache2/webdav.passwd
      <Limit PUT POST DELETE PROPFIND PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
            Require valid-user
      </Limit>
      AllowOverride None
</Directory>
```

- Visualisation HTML (parser)

– Récupérer le code PHP

```
lenny# apt-get install phpicalendar
lenny$ tar -zcf ~/phpicalendar.tgz /usr/share/phpicalendar/ /usr/share/doc/phpicalendar/
lenny$ scp /home/nroche/phpicalendar.tgz root@narval.hd.free.fr:install/.

etch# tar -zxvf /root/install/phpicalendar.tgz
```

– fichier */usr/share/phpicalendar/default_config.php*

```
$calendar_path                  = '/var/www/DAVdocs/ics/';
```

– fichier */etc/apache2/sites-availables/phpicalendar*

```
Alias /phpicalendar /usr/share/phpicalendar
<Directory /usr/share/phpicalendar/>
        AllowOverride None
</Directory>
```

– URL

4 Utf-8

Dans le cadre de la migration vers l'encodage UTF-8 il convient d'ajouter la balise MÉTA suivante aux fichiers HTML mis-à disposition.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
```

Apache peut cependant diffuser l'information au browser si on lui indique dans le fichier */etc/apache2/conf.d/charset*

```
AddDefaultCharset on
AddDefaultCharset UTF-8
```

5 Creating Certificates For Each Secure Virtual Host

- Each secure virtual host has to have its own certificate **and** key files. To create SSL certificates and their keys, use the following series of OpenSSL commands in your home directory :

```
## génère la clé 1024 bit RSA (privkey.pem) et le 'certificat request'
$ openssl req -new > new.cert.csr
```

- When you're prompted for the pass phrase, put in anything you want - we're going to strip it out in the next step. Enter the country name, city, and state as prompted. When you are prompted for the "Challenge Password", just leave it empty and press Enter. For the "Optional Company Name", enter the host name the certificate is for (hrothgar.dhs.org in our case.)

```
## new.cert.key
$ openssl rsa -in privkey.pem -out new.cert.key
```

- When you're prompted for the pass phrase, put in exactly the same phrase or word as you did above.

```
## new.cert.cert
$ openssl x509 -in new.cert.csr -out new.cert.cert -req -signkey new.cert.key -days 365
```

- Set the days parameter to the number of days you wish the certificate to be valid. After the certificate "expires", you just have to create a new one. I normally set mine to one year, as above. If you really don't want to be bothered with it, set the number to something absurd like 10 years.
- There now exists in your home directory two files named new.cert.cert and new.cert.key. These are the SSL certificate and key files that need to be copied to /opt/apachessl/conf. In the process, I normally rename them to something reminiscent of the hostname they're for.
- Utiliser ce certificat

```
$ cp new.cert.cert /etc/apache2/ssl/narval.cert
$ cp new.cert.key /etc/apache2/ssl/narval.key
```

- Fichier */etc/apache2/ssl/mods-enabled/ssl.conf*

```
# Point SSLCertificateFile at a PEM encoded certificate.
SSLCertificateFile /etc/apache2/ssl/narval.cert

# If the key is not combined with the certificate, use this directive to
# point at the key file.
SSLCertificateKeyFile /etc/apache2/ssl/narval.key

# Set SSLVerifyClient to:
# 0 - No certificate required.
# 1 - The client may present a valid certificate.
#     If a certificate is presented, it must be from a Certification Authority for which
#     the server holds a certificate.
# 2 - The client must present a valid certificate.
# 3 - The client may present a valid certificate,
#     but not necessarily from a Certification Authority for which the server holds a certifica
SSLVerifyClient 0
```