**Bachelier en informatique et Systèmes**

**Informatique industrielle**

**3ème année**



**Catégorie technique**

**Charleroi**

**Laboratoire de réseaux**

**Apache**

**2015 – 2016 Haveaux Valentin**

Tables des matières

[1 Serveur Apache 4](#_Toc440963200)

[2 Benchmark Apache 4](#_Toc440963201)

[3 Les commandes de vérification 4](#_Toc440963202)

[4 Environnement globale 4](#_Toc440963203)

[5 Les sites perso 5](#_Toc440963204)

[6 Les redirections simple 6](#_Toc440963205)

[7 Visualiser le contenu d'un répertoire 6](#_Toc440963206)

[8 L'arbre des processus 6](#_Toc440963207)

[9 Protection des sites Web 6](#_Toc440963208)

[9.1 Contrôle sur l'origine du client 6](#_Toc440963209)

[9.2 Directive Order 7](#_Toc440963210)

[9.3 Controller par authentification 8](#_Toc440963211)

[9.3.1 Autorisation de certain utilisateurs 8](#_Toc440963212)

[9.3.2 Autorisation par groupe d'utilisateurs 8](#_Toc440963213)

[10 Hébergement virtuel 9](#_Toc440963214)

[10.1 Hébergement par IP 9](#_Toc440963215)

[10.2 Hébergement par nom 10](#_Toc440963216)

[10.3 UseCanonicalName 10](#_Toc440963217)

[10.4 L'intervenant DNS 10](#_Toc440963218)

[11 Résumé certificat 11](#_Toc440963219)

[12 Tester si une requête fonctionne avec telnet 12](#_Toc440963220)

[13 Structure de l'httpd.conf 13](#_Toc440963221)

[14 Les certificats 13](#_Toc440963222)

[14.1 Clé privé du serveur 13](#_Toc440963223)

[14.2 Création du CSR (Certificat Signing Request) 14](#_Toc440963224)

[14.3 Création du certificat 14](#_Toc440963225)

[14.4 Copie des clés aux bons endroits 15](#_Toc440963226)

[14.5 Connexion 15](#_Toc440963227)

[14.6 Implémentation 16](#_Toc440963228)

[14.6.1 Configurer /etc/httpd/conf/httpd.conf 16](#_Toc440963229)

[14.6.2 Configurer /etc/httpd/conf.d/ssl.conf 16](#_Toc440963230)

Apache

# Serveur Apache

**Daemon du serveur Apache**: httpd

**Fichier de configuration**: /etc/httpd/conf/httpd.conf

**Localisation des modules**: /usr/lib64/httpd/modules

**Localisation du site principal**: /var/www/html

**Lancement/arrêt/redémarrage** : service httpd start/stop/restart

# Benchmark Apache

ab -n 1000 -c 500 [http://www.mysite.be/](http://www.mysite.be/index.html)

watch –n 0 'ps ax | grep httpd' 🡺 sur un autre terminal

# Les commandes de vérification

httpd –t 🡺 vérifie la syntaxe du fichier http.conf

httpd –S 🡺 Liste les VirtualHosts

# Environnement globale

**ServerName** : nom d'hôte sur de la machine sur laquelle apache tourne.

Exemple: ServerName www.mysite.be

**ServerRoot**: Répertoire dans lequel les fichiers de configurations, log et modules sont gardés

Exemple: ServerRoot "/etc/httpd"

**DocumentRoot**: dossier dans lequel les documents du site sont déposés.

Exemple: DocumentRoot "/var/www/html"

**ServerAdmin**: Adresse mail du webmaster.

Exemple: ServerAdmin webmaster@mysite.be

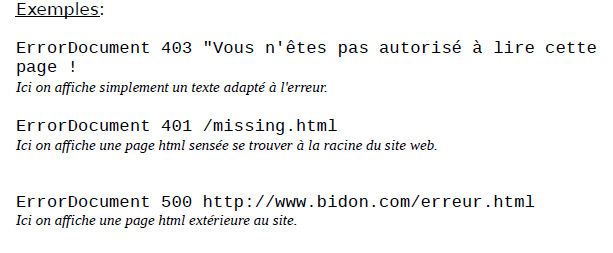
**ServerToken**s: permet de contrôler le contenu de l'en-tête Server inclus dans la réponse envoyée au client (ServerTokens Prod).

Exemple: ServerTokens Prod[uctOnly] 🡺 Le serveur renvoie (par ex.): Server: Apache

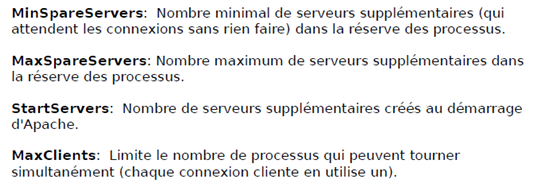
**Listen**: adresse IP et numéro de ports sur lesquels Apache attend et reçoit les connexions des clients.

Exemple: Listen 10.0.0.7:80 🡺 Apache écoute sur le port 80 sur l'interface d'IP 10.0.0.7.

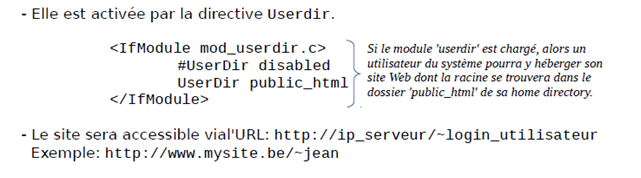
**ErrorDocument**: pour remplacer les pages d'erreurs envoyées au client en cas de problème.

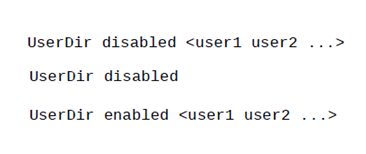


**DirectoryIndex**: page renvoyé lors d'un accès à la racine (par défaut index.html).



# Les sites perso





Pour les sites perso faite bien attention aux droit d'accès des dossiers !

chmod 755 /home/login\_user

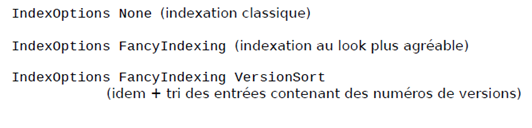
Pour avoir plus de facilité et ne pas devoir créer de dossier public\_html à chaque fois:

mkdir /etc/skel/public\_html

# Les redirections simple

Alias /CentOS/ "/usr/share/doc/HTML"

# Visualiser le contenu d'un répertoire



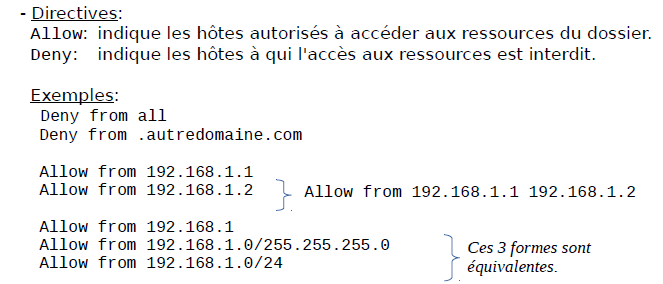
# L'arbre des processus

ps –ef | grep apache

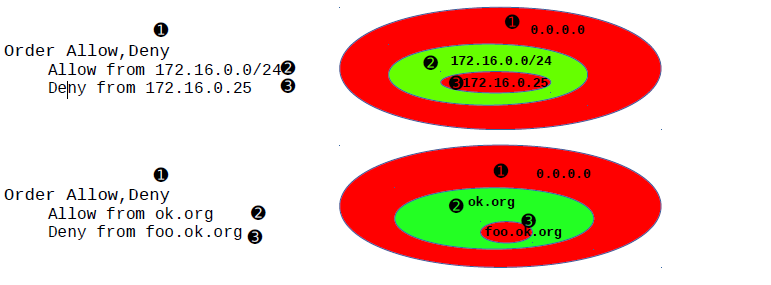
# Protection des sites Web

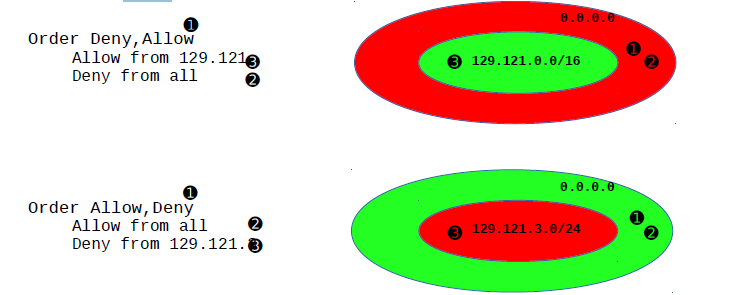
## Contrôle sur l'origine du client

* Géré par le module mod\_authz\_host
* Ses directives agissent dans un contexte répertoire (bloc **<Directory… >**)
* Les restrictions sont imposées à tous les fichiers du répertoire.

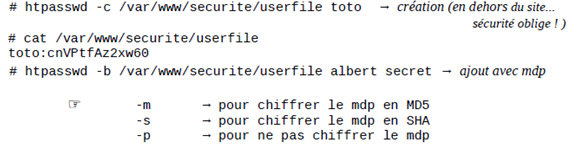


## Directive Order





## Controller par authentification

Activation de l'autorisation



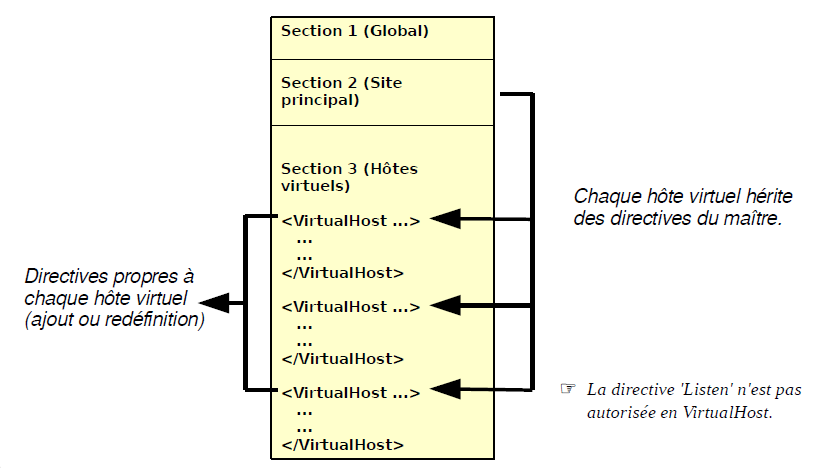
### Autorisation de certain utilisateurs



### **Autorisation par groupe d'utilisateurs**

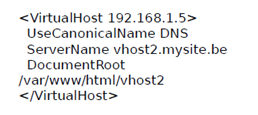


# Hébergement virtuel



## Hébergement par IP

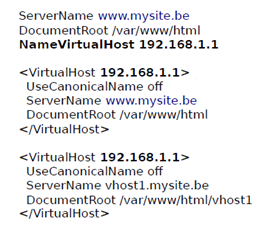




Création d'une alias: ifconfig ethx:y ***Ip***netmask ***masque***

Si une adresse IP peut être atteinte mais qu'aucun hôte n'est défini sur celle-ci, c'est le site principal qui répondra à la requête.

## Hébergement par nom



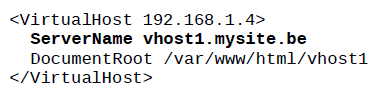
Une seule directive NameVirtualHost par adresse Ip.

## UseCanonicalName

DNS: adressage par IP

Off: hôte virtuel par nom

## L'intervenant DNS



Il est conseillé de l'écrire de cette manière, car faire intervenir DNS peut provoquer la désactivation de l'hôte.

# Résumé certificat

***Configuration de /etc/httpd/conf/httpd.conf***

…

<VirtualHost 192.168.1.100:443>

UseCanonicalName off

ServerName secure.mysite.be

DocumentRoot /var/www/websecure

SSLEngine on

SSLCertificateFile /etc/pki/tls/certs/secure.mysite.be.crt

SSLCertificateKeyFile /etc/pki/tls/private/secure.mysite.be.key

</VirtualHost>

***Installation des modules openssl et mod\_ssl (openssl est certainement déjà installé)***

# yum install openssl mod\_ssl –y

***Implémentation des clés et des certificats***

*Création de la clé privée du serveur*

# cd /tmp

# openssl genrsa -out secure.mysite.be.key 2048

*Création du CSR*

# openssl req -new -key secure.mysite.be.key -out secure.mysite.be.csr

…

*Remplir tous les champs d'information*  *concernant le formulaire.*

*…*

*Création du certifciat auto-signé*

# openssl x509 -in secure.mysite.be.csr -out secure.mysite.be.crt

-req -signkey secure.mysite.be.key -days 3650

*Copie des clés et certificats au bon endroit*

# cp secure.mysite.be.crt /etc/pki/tls/certs

# cp secure.mysite.be.csr /etc/pki/tls/private

# cp secure.mysite.be.key /etc/pki/tls/private

*Suppression des clés et certificats de /tmp*

# rm -f /tmp/secure.mysite.be\*

***Relancer Apache***

# service httpd restart

# Tester si une requête fonctionne avec telnet

mcedit get-page.sh

echo "open $1 80"

sleep 2

echo "GET / HTTP/1.1"

echo "host: $1"

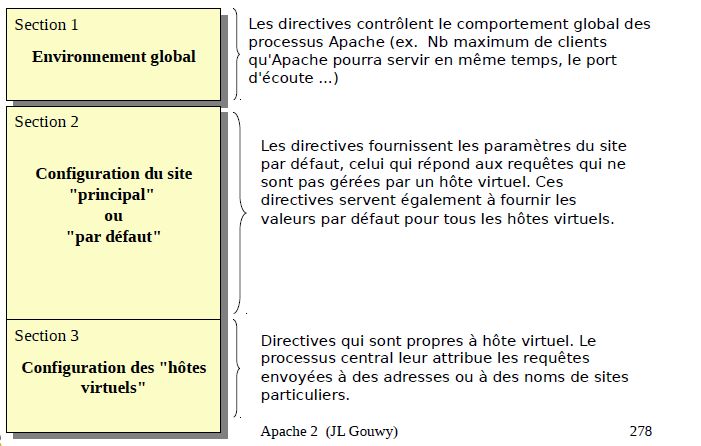
echo

echo

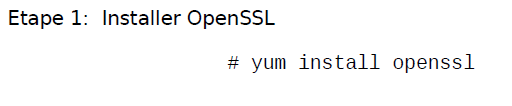
sleep 2

sh get-page.sh [www.helha.be](http://www.helha.be) | telnet > get-page.out

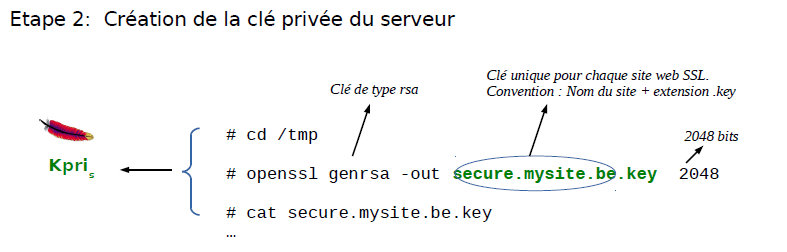
# Structure de l'httpd.conf



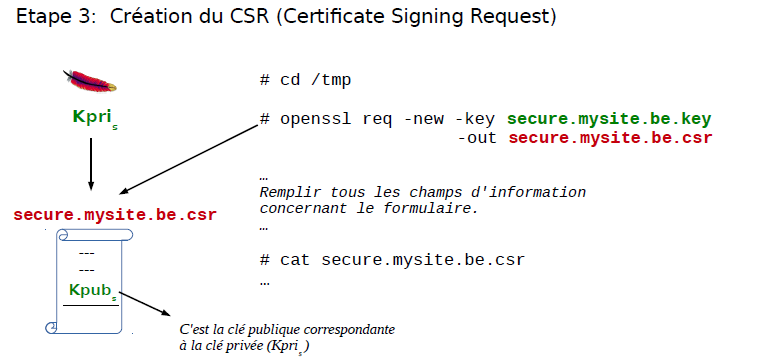
# Les certificats



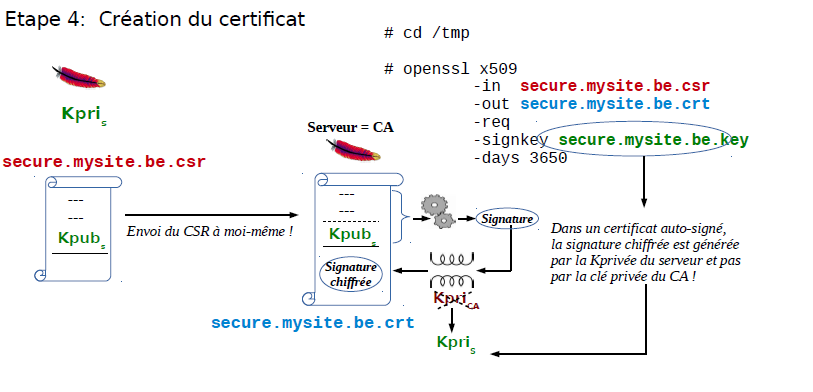
## Clé privé du serveur



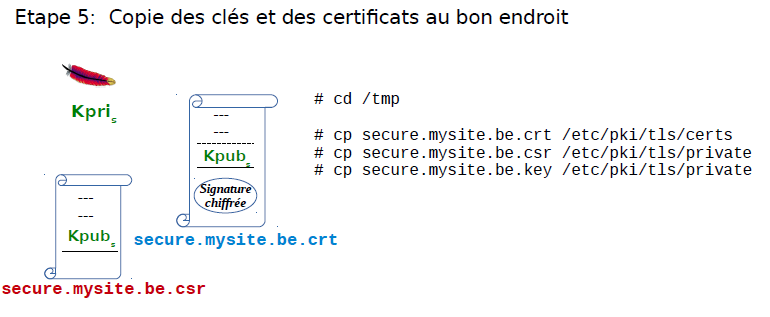
## Création du CSR (Certificat Signing Request)



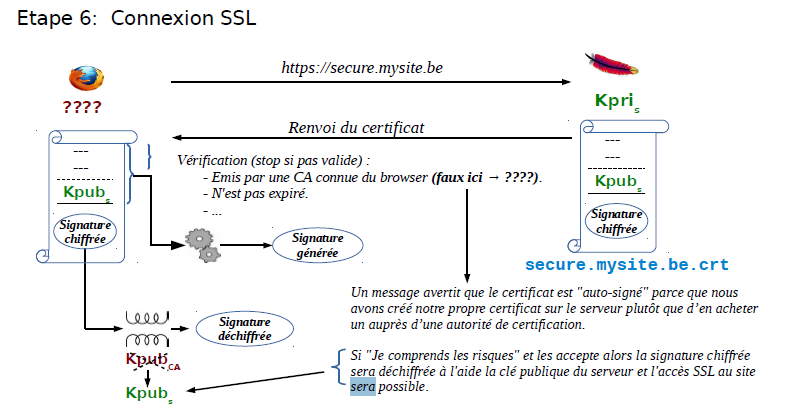
## Création du certificat



## Copie des clés aux bons endroits



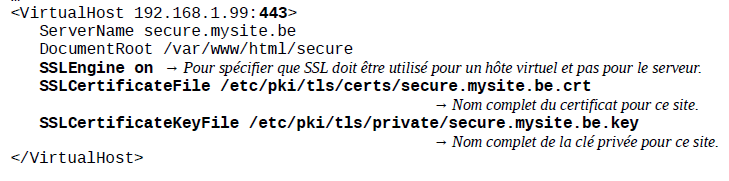
## Connexion



## Implémentation

### Configurer /etc/httpd/conf/httpd.conf

Installer le module mod\_ssl: # yum install mod\_ssl



### Configurer /etc/httpd/conf.d/ssl.conf

