



SERVICE Web APACHE

Serveur Linux CentOS

Table des matières

1	Introduction.....	3
2	Installation et contrôle du service Apache	4
3	Configuration de base du service Apache.....	7
4	Directives globales.....	8
5	Directives de configuration du serveur principal.....	10
6	Directives des serveurs virtuels (vhost)	13
7	Emplacement des pages web	14
8	Travail Pratique 1.....	15
9	Travail pratique 2.....	17
10	Installation de plusieurs démons.....	19
11	Travail pratique 3.....	20

1 Introduction

Pourquoi Apache est-il devenu un standard ?

- Coût nul
- Code source disponible et modifiable permet un développement rapide du serveur, la création de modules spécifiques et une très grande réactivité dans la correction des bogues.
- Très grande flexibilité du serveur grâce à sa structure modulaire l'ajout d'un nouveau module permet d'ajouter de nouvelles fonctionnalités.

Sa flexibilité combinée à sa stabilité, à ses performances, ainsi qu'à la disponibilité du code source a fait du serveur **Apache**, le serveur WEB le plus populaire sur Internet.

Le protocole HTTP

HTTP (HyperText Transfer Protocol) est un protocole de requêtes et de réponses. Le dialogue entre un client web (un navigateur tel que Netscape) et un serveur (**Apache**) se traduit par une requête du client à laquelle le serveur répond en effectuant le traitement intermédiaire adéquat.

Par exemple lorsque le navigateur transmet l'**URL**(Uniform Resource Locator) <http://www.orabec.ca/index.html> au serveur Web.

Ce dernier commence par extraire la partie initiale, qui définit le protocole à utiliser (http), ainsi que le nom du serveur à contacter (www.orabec.ca). Ensuite il se connecte à cette machine en utilisant le port TCP approprié (80 dans le cas du protocole http).

Le protocole SSL (Secure Sockets Layer)

Parfois, l'**URL** utilisée pour accéder à une page commence par **https://**. Cela signifie que la connexion se fait par le biais du protocole **SSL**.

Dans la pratique, cela correspond à plusieurs choses.

D'abord, votre navigateur se connecte au serveur sur un port différent (**443**) au lieu du port http classique (**80**).

Ensuite, toutes les informations transmises entre votre navigateur et le serveur sont chiffrées à l'aide d'un algorithme complexe qui vous protège contre toute interception de l'échange.

Enfin, le serveur est identifié formellement, afin de vous protéger des imposteurs. Le protocole **SSL** n'est pas spécifique à **HTTP**.

2 Installation et contrôle du service Apache

Sur **Redhat/CentOS**, installer le paquetage **httpd** :

```
[root@localhost ~]# yum install httpd
```

Le service Web écoute par défaut sur le port **HTTP** (port **TCP 80**).

Pour afficher la version du service **Apache** :

```
[root@localhost ~]# httpd -v  
Server version: Apache/2.4.6 (CentOS)  
Server built: Apr 24 2019 13:45:48
```

Pour vérifier la configuration du service **Apache** :

```
[root@localhost ~]# httpd -t  
Syntax OK
```

Pour démarrer le service **Apache** avec la commande **httpd**:

```
[root@localhost ~]# httpd
```

Ou bien :

```
[root@localhost ~]# httpd -k start
```

Pour afficher les processus du service **Apache** :

```
[root@localhost ~]# ps -ef|grep httpd  
root      17344      1    0 19:51 ?        00:00:00 httpd  
apache    17345    17344    0 19:51 ?        00:00:00 httpd  
apache    17346    17344    0 19:51 ?        00:00:00 httpd  
apache    17347    17344    0 19:51 ?        00:00:00 httpd  
apache    17348    17344    0 19:51 ?        00:00:00 httpd  
apache    17349    17344    0 19:51 ?        00:00:00 httpd  
root      17377    17216    0 19:57 pts/0    00:00:00 grep --  
color=auto httpd
```

Pour arrêter le service **Apache** avec la commande **httpd**:

```
[root@localhost ~]# httpd -k stop
```

Pour démarrer le service **Apache** en utilisant la commande **systemctl**:

```
[root@localhost ~]# systemctl start httpd
```

Pour arrêter le service **Apache** en utilisant la commande **systemctl**:

```
[root@localhost ~]# systemctl stop httpd
```

Il existe également un outil nommé **apachectl** qui permet de contrôler le service **Apache**:

Démarrer le service **Apache**:

```
[root@localhost ~]# apachectl start
```

Arrêter le service **Apache**:

```
[root@localhost ~]# apachectl stop
```

Redémarrer le service **Apache**:

```
[root@localhost ~]# apachectl restart
```

Redémarrer le service **Apache**: élégamment :

```
[root@localhost ~]# apachectl graceful-stop
```

Lancer automatiquement le serveur au démarrage de l'ordinateur

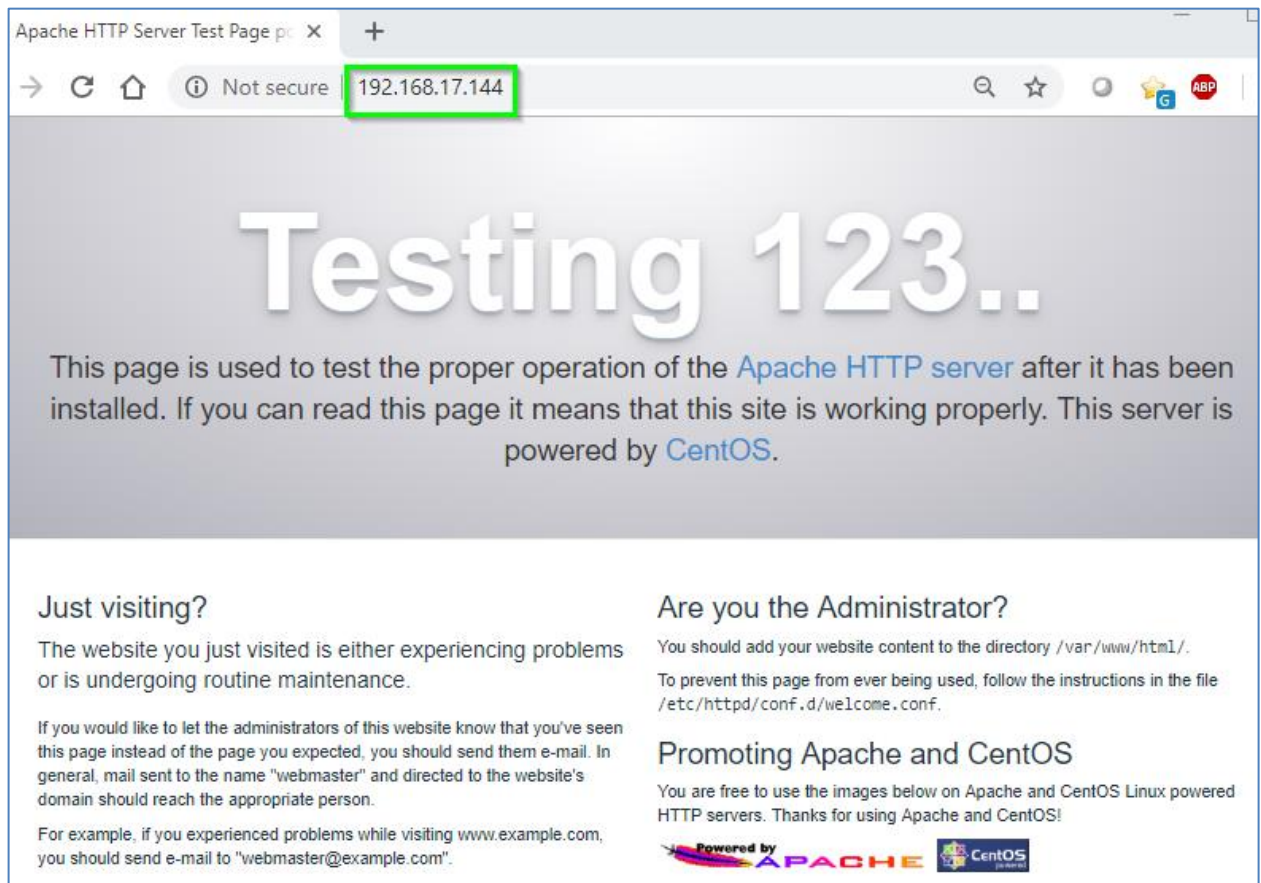
Dans le cas d'une panne d'électricité prolongée dépassant la durée d'efficacité de l'onduleur, le serveur s'éteint. Lorsque le courant est rétabli, le serveur redémarre. Il est crucial comme tout autre service réseau que le service **Apache** soit redémarré automatiquement afin que le temps d'indisponibilité des sites hébergés soit le plus court possible.

On peut activer le service **Apache** pour qu'il soit lancé au démarrage :

```
[root@localhost ~]# systemctl enable httpd
```

Tester le service Apache

L'approche la plus simple et la plus naturelle pour tester le bon fonctionnement du service **Apache** consiste à lancer un navigateur web et d'utiliser **http://adresse-ip** comme URL. Après l'installation et le lancement du service **Apache**, vous devriez voir apparaître la page web par défaut d'**Apache**.



Si aucune page n'apparaît, vérifier tout d'abord si le serveur est bien lancé :

```
[root@localhost ~]# systemctl status httpd
```

Dans le cas où le fichier de configuration du service **Apache** comporterait une erreur, un message signale l'emplacement de l'erreur lors du lancement du serveur.

Le protocole **HTTP** reposant sur des commandes textuelles, vous pouvez simuler l'action de votre navigateur à l'aide de **telnet** :

```
[root@localhost ~]# telnet www.orabec.ca http
Trying 127.0.0.1...
Connected to www.orabec.ca.
Escape character is '^]'.
GET /index.html
<Affichage du contenu de index.html>..."
```

Une fois la commande appropriée transmise au serveur, le navigateur reçoit le fichier demandé et l'affiche sur votre écran.

3 Configuration de base du service Apache

La configuration du service **Apache** peut s'effectuer soit en modifiant manuellement son fichier de configuration avec un éditeur de texte soit en utilisant un outil web tel que **WebMin**. Même si la configuration du service **Apache** à partir d'un fichier texte peut paraître fastidieuse, cette méthode présente l'avantage de contrôler exactement et de façon exhaustive le fonctionnement du service.

Le fichier de configuration du service **Apache** se nomme **httpd.conf** et est placé dans le répertoire **conf/** du répertoire du service **Apache**. Le chemin complet du fichier est : **/etc/httpd/conf/httpd.conf**

Comment **Apache** structure-t-il sa configuration ?

- Les directives de conteneur **Apache** ont une portée limitée ;
- Les directives peuvent être utilisées à portée globale ou locale ;
- Il est possible d'outrepasser une directive à l'aide d'une configuration par répertoire.

Vous trouverez la liste complète de toutes les directives reconnues par le service **Apache** à l'adresse :

<http://httpd.apache.org/docs/2.4/mod/quickreference.html>

La documentation complète du service **Apache** version **2.4** se trouve à l'adresse :

<http://httpd.apache.org/docs/2.4/>

4 Directives globales

ServerRoot

Syntaxe : `ServerRoot` *nom répertoire*

Valeur par défaut : `ServerRoot /etc/httpd`

La directive **ServerRoot** définit le répertoire dans lequel se situe le serveur. Typiquement, ce répertoire contiendra les sous-répertoires **conf/** et **logs/**. Les chemins d'accès relatifs pour d'autres fichiers de configuration seront considérés relativement à ce répertoire.

PidFile

Syntaxe : `PidFile` *nom répertoire*

Valeur par défaut : `PidFile run/httpd.pid`

Fichier dans lequel le serveur enregistre son le **PID** du service **apache**. Le chemin est relatif à **ServerRoot**.

Timeout

Syntaxe : `Timeout` *duré en secondes*

Valeur par défaut : `Timeout 120`

Duré d'attente en secondes avant que le serveur ne retourne une erreur.

Listen

Syntaxe : `Listen` *adresse IP:numéro de port*

Valeur par défaut : `Listen 80`

```
Listen 80
```

numéro est un nombre compris entre **0** et **65535**; certains numéros de ports (surtout en dessous de **1024**) sont réservés pour des protocoles spécifiques. Une liste des ports prédéfinis est consultable dans **/etc/services**; le port standard assigné au protocole HTTP est le **port 80**.

La directive **Listen** définit le port réseau que le serveur écoute.

Le Port 80 est l'un des ports prédéfinis de Linux. Tous les ports numérotés en dessous de **1024** sont réservés à un usage système, c'est-à-dire. Que des utilisateurs non privilégiés (non-root) ne peuvent les utiliser ; ces derniers peuvent par contre utiliser des ports de plus haut rang.

```
Listen 192.168.13.11:80
```

Group

Syntaxe : Group *groupeUnix*

Valeur par défaut : Group apache

La directive Group définit le groupe dont les requêtes seront traitées par le serveur.

User

Syntaxe : User *utilisateurUnix*

Valeur par défaut : User apache

La directive User définit l'utilisateur associé au serveur.

5 Directives de configuration du serveur principal

ServerAdmin

Adresse email de l'administrateur

Syntaxe : `ServerAdmin adresseEMail`

Valeur par défaut : `ServerAdmin root@localhost`

La directive **ServerAdmin** définit l'adresse e-mail que le serveur inclut dans tout message d'erreur retourné au client.

Il peut être utile de dédier une adresse réservée à cet usage, par exemple:

```
ServerAdmin webmaster@orabec.ca
```

ServerName

Nom du serveur

Syntaxe : `ServerName nom de domaine entièrement qualifié:port`

La directive **ServerName** définit le nom d'hôte du serveur et le numéro de port que le serveur utilise que s'identifier lui-même.

DocumentRoot

Emplacement par défaut des pages html

Syntaxe : `DocumentRoot directory-filename`

Valeur par défaut : `DocumentRoot /var/www/html`

Cette directive définit le répertoire racine à partir duquel **httpd** va distribuer les fichiers. Le serveur ajoute le chemin relatif mentionné dans l'**URL** présentée à cette racine pour établir le chemin complet jusqu'au document.

Un accès à <http://www.my.host.com/index.html> se réfère au document </var/www/html/index.html>

UserDir

Syntaxe: UserDir *répertoire*

Valeur par défaut: UserDir disabled

La directive **UserDir** définit le sous-répertoire contenu dans répertoire personnel de l'utilisateur qui servira de location de la page personnelle de l'utilisateur (**~utilisateur**).

Le paramètre qui suit **UserDir** peut prendre l'une des valeurs suivantes:

- Le nom du sous-répertoire pour la page personnelle.
- Le mot-clé **disabled**. Ce qui permet de désactiver la translation usager répertoire.
- Le mot-clé **disabled** suivi de la liste des utilisateurs dont la translation usager répertoire est désactivée.
- Le mot-clé **enabled** suivi de la liste des utilisateurs dont la translation usager répertoire est activée.

EXEMPLE

La requête pour **http://www.orabec.ca/~bob/index.html** sera dirigée vers:

```
UserDir public_html->~bob/public_html/index.html
UserDir /usr/web->/usr/web/bob/index.html
UserDir /home/*/www->/home/bob/www/index.html
```

Les directives suivantes provoqueront les redirections suivantes :

```
UserDir http://www.fo.com/users->http://www.fo.com/users/bob/index.html
UserDir http://www.fo.com/*/usr->http://www.fo.com/bob/usr/index.html
UserDir http://www.fo.com/~*/-> http://www.fo.com/~bob/index.html
```

Pour permettre à seulement certains utilisateurs d'avoir **UserDir**:

```
UserDir disabled
UserDir enabled user1 user2 user3
```

Pour interdire à seulement certains utilisateurs d'avoir **UserDir** :

```
UserDir enabled
UserDir disabled user4 user5 user6
```

DirectoryIndex

Syntaxe : DirectoryIndex fichier1 fichier2

Valeur par défaut : DirectoryIndex index.html index.html.var

Définit l'ordre de recherche de la page par défaut à afficher.

```
DirectoryIndex index.htm index.html index.php
```

ErrorLog

Journal d'erreur par défaut.

Syntaxe : ErrorLog *nomfichier*

Valeur par défaut : ErrorLog logs/error_log

Cette directive définit le nom du fichier dans lequel le serveur marque la trace des erreurs rencontrées (**error_log**).

ErrorDocument

Syntaxe : `ErrorDocument code d'erreur document`

Dans l'éventualité d'un problème ou d'une erreur, **Apache** peut exécuter l'une des quatre actions suivantes :

- Sortie d'un message d'erreur simple standard
- Sortie d'un message personnalisé
- Redirection vers une **URL** locale pour traiter le problème (ou l'erreur)
- Redirection vers une **URL** externe pour traiter le problème (ou l'erreur)

La première option est celle par défaut, les options **2** à **4** seront obtenues en utilisant la directive **ErrorDocument**, suivi du code **HTTP** d'erreur et du message textuel d'erreur, ou une **URL**.

Messages dans ce contexte, commence par un guillemet simple ("), qui ne fait pas partie du message lui-même. **Apache** ajoutera souvent des informations complémentaires explicitant le problème (ou l'erreur).

L'**URL** peut débuter par un slash (/) pour des URL locales, ou être complètement qualifiées. Exemples:

```
ErrorDocument 500 http://foo.example.com/cgi-bin/tester
ErrorDocument 401 /subscription_info.html
ErrorDocument 403 "Sorry can't allow you access today"
```

6 Directives des serveurs virtuels (vhost)

Voir le document "**Hôtes virtuels**"

7 Emplacement des pages web

Une fois le serveur configuré et activé, il paraît logique de s'en servir afin de diffuser des documents. L'emplacement des documents est celui indiqué par la directive **DocumentRoot** du fichier **httpd.conf**. Les fichiers situés dans ce répertoire seront immédiatement visibles et accessibles.

De nombreux serveurs diffusent des pages gérées par différents usagers. Si ces pages devaient toutes se situer dans un même répertoire, avec les risques d'altérations que cela suppose, la situation deviendra rapidement chaotique. Mais, heureusement, ce n'est pas le cas. Dans sa configuration par défaut, le serveur **Apache** autorise la création de pages personnelles dans le répertoire de connexion de chaque utilisateur.

L'emplacement de ces pages est défini par la directive **UserDir**. Par convention, il s'agit du répertoire public **public_html**. Grâce à cette directive, un utilisateur disposant d'un répertoire de connexion sur le serveur peut créer un sous répertoire **public_html** et y insérer ses pages personnelles.

Celles-ci seront alors accessibles à l'aide d'une syntaxe particulière :

<http://serveur/~usager>

Le tilde (~) indique au serveur **Apache** que le mot qui suit est le nom d'un usager, et qu'il faut se diriger vers le répertoire **public_html** situé dans son répertoire de connexion.

8 Travail Pratique 1

- 1) Créer un usager **u1** et lui attribuer un mot de passe.

- 2) Se connecter avec le compte **u1**

- 3) Créer le répertoire **public_html** dans le répertoire personnel de l'utilisateur **u1**.

- 4) Créer le fichier **accueil.html** suivant dans **public_html** :

```
<HTML>
  <HEAD>
    <TITLE><Page Web usager u1></TITLE>
  </HEAD>
  <BODY>
    <H1>Page Web Orabec Inc</H1>
    <P>Liste des produits</P>
  </BODY>
</HTML>
```

- 5) Se connecter comme **root** et modifier le fichier de configuration du service **Apache** de façon à permettre la visualisation de la page Web de l'utilisateur **u1**.

- 6) Redémarrer le service **apache**.

- 7) Changer les droits d'accès au besoin pour permettre aux autres usagers d'accéder à la page Web de l'utilisateur **u1** en lecture seulement.

- 8) Donner l'**URL** qui nous permettra de visualiser le contenu du fichier **accueil.html**.

- 9) Comment faire pour que l'URL <http://localhost/~u1> renvoie le contenu du fichier **accueil.html** ?

- 10) Donner deux façons différentes qui nous permettront de vérifier que le serveur **apache** est opérationnel.

- 11) Donner le nom complet (absolu) du fichier correspond à l'URL suivant:

<http://localhost/vente/jan2014/produit.html>

- 12) Vérifier que le service **Apache** sera bien lancé à chaque démarrage du serveur.

9 Travail pratique 2

Utiliser la directive **ErrorDocument** afin de rediriger les erreurs **404** vers une page d'erreur personnalisée renvoyant sur la page d'accueil du site.

1) Procéder à la désinstallation du service **Apache**

2) Procéder à l'installation du service **APACHE**

3) Quel est l'utilité de l'utilisateur et du groupe **Apache**

4) Installer l'outil **system-config-httpd** qui permet de configurer le service **Apache** en mode graphique

5) Installer le manuel du service **Apache**

6) Quels sont la location et le nom du fichier de configuration du service **Apache**

7) Sauvegarder ce fichier avec l'extension **.default** afin de conserver une copie avec les valeurs par défaut

8) Quelle est la signification du paramètre **Pidfile**

9) Démarrer le service **Apache**

10) Quel est le **PID** du processus maître

11) Écrire un script qui permet d'arrêter le service **Apache** en utilisant la commande **kill**

12) Quel est le paramètre qui limite le nombre maximum de connexions

13) Que se passe-t-il lorsque l'on essaie d'atteindre l'URL : <http://localhost/g1234.html>

14) Utiliser la directive **ErrorDocument** afin de rediriger les erreurs **404** vers une page d'erreur personnalisée renvoyant sur la page d'accueil du site.

10 Installation de plusieurs démons

Il y a deux manières de configurer le service **Apache** pour qu'il supporte plusieurs hôtes. Soit en exécutant un processus démon **httpd** pour chaque nom d'hôte, soit en exécutant un démon unique qui supporte tous les hôtes virtuels.

Pour installer plusieurs démons, créer une installation d'http séparée pour chaque hôte. Pour chaque installation, utiliser la directive **Listen** dans le fichier de configuration pour sélectionner l'adresse IP (hôte) que le démon va servir.

```
Listen 192.168.100.20:80
```

11 Travail pratique 3

- 1) Créer deux configurations distinctes **httpd1.conf** et **httpd2.conf**.
- 2) Lancer une instance de **httpd** par configuration.
- 3) Tester que les deux instances fonctionnent bien.

```
[root@localhost ~]# httpd -f /etc/httpd/conf/httpd1.conf
```

```
[root@localhost ~]# ps -ef|grep httpd
```

```
[root@localhost ~]# httpd -f /etc/httpd/conf/httpd2.conf
```

```
[root@localhost ~]# ps -ef|grep httpd
```

Arrêt du service apache:

```
[root@localhost ~]# httpd -k stop -f /etc/httpd/conf/httpd2.conf
```