

LINUX SERVER ADMINISTRATION

DOCUMENTATIE CURS

DOCUMENTATIE

INTREABA PROFESORUL

CURSURILE MELE

7 Serverul Apache » 7.5 Configurare Apache » 7.5.1 Sectiuni de configurare

- 1. Shell Scripts
- 2. Linux Kernel
- 3. Serverul DHCP
- 4. Serverul FTP
- 5. NFS - Network File System
- 6. Serverul DNS
- 7. Serverul Apache
 - 7.1 Protocolul HTTP
 - 7.2 Prezentare generala server
 - 7.3 Compilare si instalare
 - 7.4 Structura Apache
 - 7.5 Configurare Apache
 - 7.5.1 Sectiuni de configurare
 - 7.5.2 Configurare Minimala
 - 7.5.3 Directive principale
 - 7.5.4 Virtual Hosting
 - 7.5.5 Controlul accesului la resurse
 - 7.6 PHP
 - 7.7 Securitate Web
- 8. Serverul MySql
- 9. NETFILTER
- 10. Sistemul de e-Mail
- 11. Serverul Postfix
- 12. Serverul POP/IMAP
- 13. Managementul Logurilor
- 14. Exemple practice (Ubuntu 14.04 LTS)
- 15. Webmin

Sectiuni de configurare

Directivele (optiunile) de configurare pot avea efect pentru intreg serverul sau doar pentru o serie de directoare, fisiere, hosturi sau URL-uri. Exista astfel mai multe sectiuni de configurare. Acestea sunt un fel de container si contin directivele ce se vor aplica contextului respectiv.

1. Server config

Sunt optiunile permise in `httpd.conf` sau in alt fisier inclus din `httpd.conf`, in contextul global, si care au efect asupra intregului serverul. Unele nu sunt permise in alte contexte precum `<Directory>`.

Exemplu: nr. maxim de conexiuni care vor fi procesate simultan (`MaxRequestWorkers` number).

In documentatia serverului, pentru fiecare directiva, este indicat contextul in care directiva este permisa.

MaxRequestWorkers Directive

Description: Maximum number of connections that will be processed simultaneously

Syntax: `MaxRequestWorkers number`

Default: See usage for details

Context: `server config`

Status: MPM

Module: `event, prefork, worker`

2. <Directory>

Directivele care se gasesc intre `<Directory PATH>` si `</Directory>` vor fi aplicate doar directorului respectiv, subdirectoarelor acestuia precum si fisierele din respectivele directoare. Nu toate directivele sunt permise in contextul `<Directory>`, iar **contextul `<Directory>` se poate gasi doar in server config sau in sectiunea `<VirtualHost>`**. `PATH` reprezinta o cale absoluta (incepe cu `/`) la sistemul de fisiere al sistemului de operare.

Optiunile sunt valabile pentru fisiere, doar daca sunt accesate cu respectiva cale si nu altfel (Ex: via symlinks sau redirectari).

Se recomanda setarea de permisiuni foarte restrictive pentru directorul radacina `"/` al OS-ului, fiindca toate directoarele din sistem (inclusiv toate `VirtualHost`-urile) vor mosteni apoi.

Exemplu:

```
<Directory />
  AllowOverride none
  Options none
  Order Deny,Allow
  Deny from All
</Directory>
```

Nota



`<Directory>` trebuie sa se gaseasca in contextul global si nu `VirtualHost` ca sa devina default pt. orice subdirector.

`<DirectoryMatch PATH>` - se foloseste la fel ca `<Directory>` doar ca se poate folosi o expresie regulata pt. `PATH`. Nu se poate folosi `$` si anume end-of-line.

Exemplu: `<DirectoryMatch "^/www/(.+)?[0-9]{3}">` - face match la directoarele din `/www` care au numele format din 3 cifre.

3. <Files>

Directivele care se gasesc intre `<Files FILENAME>` si `</Files>` se vor aplica doar fisierele cu acel nume (`FILENAME`).

Important

```
Sectiunea <Files> se poate folosi doar in server config, <Directory>, <VirtualHost> si .htaccess
```

Pentru numele fisierului se poate folosi un string, ? care reprezinta un singur caracter sau * care reprezinta o secventa de oricate caractere.

Exemplu:

```
<Files "contact.html">
  # Optiuni care se aplica fisierelor numite contact.html
</Files>

<Files "?earch.*">
  #optiuni aplicate fisierelor al caror nume incepe cu orice caracter,
  #urmeaza earch si apoi orice extensie (ex: search.html, search.php, gearch.jpg etc).
</Files>
```

<FilesMatch> permite folosirea de expresii regulate extinse.

Este echivalent cu <Files>, dar este preferat. Expresii regulate se pot folosi inclusiv cu <Files> daca stringul de "prins" este precedat de ~

Exemplu:

```
#fisierle al caror nume incepe cu .ht nu sunt servite niciunui client
<FilesMatch "\.ht">
  Order allow,deny
  Deny from all
</FilesMatch>

#fisierle cu extensia gif, jpg, jpeg sau png sunt accesibile doar de la 192.168.0.2
<Files ~ "\.(gif|jp?g|png)$">
  Order allow,deny
  Allow from 192.168.0.2
</Files>
```

4. <Location>

Limiteaza actiunea directivelor cuprinse intre <Location URL> si </Location> in functie de URL-ul cu care sunt accesate paginile web. Se foloseste in special impreuna cu directiva SetHandler. Exista de asemenea <LocationMatch> care accepta regex-uri.

5. <VirtualHost>

Sunt directivele ce se gasesc intre <VirtualHost> si </VirtualHost> si au efect doar pentru un anumit web site identificat in functie de domeniul cu care este accesat.

Detalii despre VirtualHosting se gasesc intr-un capitol viitor.

6. .htaccess

Reprezinta o modalitate prin care se pot folosi optiuni de configurare pentru un anumit director si toate subdirectoarele sale, fara a modifica fisierle de configurare ale serverului Apache, operatiune ce necesita drepturi de root.

Acest mod de lucru se foloseste in special pe serverele de Shared Hosting unde un client nu are permisiuni de root pe server si nu poate modifica fisierul de configurare al serverului Apache.

Exemplu: Presupunem ca administratorul/proprietarul unui web site gazduit pe un server unde mai sunt si alte site-uri ale altor utilizatori ([Shared Hosting](#)) doreste o anumita optiunie pentru site-ul sau, cum ar fi accesarea unui anumit director si a continutului acestuia doar de la propriul IP.

Acest lucru se realizeaza modificand fisierul de configurare al serverului Apache (httpd.conf), adaugand in acesta:

```
<Directory /home/invata-online.ro/admin>
  Order Allow,Deny
  #ip-ul de la care se permite accesul
  Allow from 80.0.10.1
</Directory>
```

Administratorul site-ului nu va putea niciodata realiza acest lucru deoarece nu va avea drepturi de root pe server (este doar unul dintre zecile sau sutele de utilizatori de pe server).

Alternativa oferita de `.htaccess` este ca administratorul site-ului sa creeze un fisier numit `.htaccess` pe care sa-l copieze prin FTP in directorul `admin` (nu sunt necesare drepturi de root). Continutul lui `.htaccess` va fi:

```
Order Allow,Deny
#ip-ul de la care se permite accesul
Allow from 80.0.10.1
```

Serverul Apache va aplica optiunile si deci va permite accesul doar de la IP-ul 80.0.10.1 pentru directorul `admin` si toate directoarele sale.

Important

Recomandarea generala este sa se evite folosirea lui `.htaccess` de fiecare data cand este posibil, deoarece are impact asupra performantei si securitatii intregului server. [Detalii](#).

Resurse

- [Apache Configuration Sections](#)
- [Apache Configuration Files](#)
- [.htaccess](#)