

# LINUX SERVER ADMINISTRATION

## DOCUMENTATIE CURS

DOCUMENTATIE

INTREABA PROFESORUL

CURSURILE MELE

- 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
          - 7.5.5.1 Acces in functie de IP
          - 7.5.5.2 Acces in functie de user si parola
          - 7.5.5.3 Autorizare acces (Apache 2.4)
  - 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

7 Serverul Apache » 7.5 Configurare Apache » 7.5.5 Controlul accesului la resurse » 7.5.5.3 Autorizare acces (Apache 2.4)

## Autorizare acces (Apache 2.4)

Odata cu versiunile serverului mai recente de 2.4, sistemul de autorizare al accesului s-a modificat. Directivele Order, Allow si Deny au devenit inechite (deprecated) si au fost inlocuite de directivele Require.

### Nota



In continuare pot fi folosite directivele vechi, doar ca nu mai sunt recomandate din cauza complexitatii crescute si posibilitatii mai mari de a gresi. Este posibil ca in versiuni viitoare ale serverului acestea sa dispara.

Au fost astfel introduse:

a) directiva Require

b) containerele:

<RequireAll> - toate conditiile din interiorul containerului trebuie indeplinite simultan

<RequireAny> - orice conditie poate fi indeplinita

<RequireNone> - nicio conditie nu trebuie sa fie indeplinita.

Pentru un control al accesului cat mai granular, directiva Require si containerele de mai sus vor fi folosite in cadrul containerelor

<Directory>, <Files>, <VirtualHost>, <Location> etc.

### Important

Implicit se considera ca toate directivele Require se gasesc in interiorul containerului <RequireAny>. Acest lucru inseamna ca daca o singura conditie este satisfacuta, accesul este permis.

Astfel:

```
<RequireAny>
Require ip 192.168.0.0/24
Require local
</RequireAny>
```

Este acelasi lucru cu:

```
Require ip 192.168.0.0/24
Require local
```

Trecem acum la cateva exemple practice:

1. Nicio cerere nu este permisa

Varianta veche (2.2):

```
Order deny,allow
Deny from all
```

Varianta noua (>2.4):

```
Require all denied
```

2. Toate cererile sunt permise

Varianta veche:

```
Order allow,deny
Allow from all
```

Varianta noua:

```
Require all granted
```

3. Doar cererile de la un anumit IP sau retea sunt permise pentru directorul `/var/www/html/protect`

Varianta veche:

```
<Directory /var/www/html>
Order Deny,Allow
Deny from all
Allow from 80.0.0.1
Allow from 100.0.0.0/26
</Directory>
```

Varianta noua:

```
<Directory /var/www/html>
Require all denied
Require ip 80.0.0.1
Require ip 100.0.0.0/26
<Directory>
```

4. Orice cerere pentru un fisier dintr-un director numit `admin` sau `secure` va fi permisa doar daca utilizatorul este autentificat cu username si parola.

```
SetEnvIf REQUEST_URI "/(admin|secure)/" PROTECTED

<RequireAny>

  <RequireAll>
    Require not env PROTECTED
    Require all granted
  </RequireAll>
  <RequireAll>
    AuthType Basic
    AuthName "Password Protected"
    AuthUserFile <path_to_your_htpasswd_file>
    Require valid-user
  </RequireAll>

</RequireAny>
```

Explicatii:

- variabila speciala `REQUEST_URI` contine calea catre resursa ceruta, relativa la `DocumentRoot`.

**Exemplu:** daca utilizatorul acceseaza `http://www.crystalmind.ro/proiecte/linux.pdf` atunci `REQUEST_URI` va contine valoarea `/proiecte/linux.pdf`

Astfel directiva `SetEnvIf REQUEST_URI "/(admin|secure)/" PROTECTED` presupune ca in cazul in care `REQUEST_URI` contine stringul `admin` sau `secure` sa fie creata o variabila numita `PROTECTED`

- accesul este permis daca oricare dintre cele 2 containere `<RequireAll>` returneaza adevarat, cu alte cuvinte daca nu exista variabila `PROTECTED` (adica nu s-a accesat directorul `admin` sau `secure`) sau daca utilizatorul se autentifica (al 2-lea container `<RequireAll>`)

- containerul `<RequireAny>` exterior poate fi sters deoarece este oricum default. A fost trecut doar pt. claritate:

**Nota**



Directiva `Require not`, folosita pentru negare, trebuie mereu sa se gaseasca intr-un container `<RequireAll>` altfel va fi logat un mesaj de alerta.

**Nota**



Pentru a permite accesul local si anume la `localhost` sau `127.0.0.1` la o resursa (site, director etc) se foloseste `Require local`.

5. Accesul la site este permis doar de la localhost (127.0.0.1), rețeaua locală 10.0.0.0/24 sau dacă userul de autentificare are username și parolă (alte ip-uri).

```
<Directory "/path/to/your/website">  
Options FollowSymLinks  
AllowOverride None  
Require local  
Require ip 192.168.1  
Require valid-user  
</Directory>
```

#### Resurse:

- [Upgrading to 2.4 from 2.2](#)