Gestire un server web - Apache

Potete scaricare un server web per il vostro PC da
https://www.apachefriends.org/download.html
Nel pacchetto sono inclusi anche php e MySQL nella versione MariaDB, PERL ed un SMTP server.
Durante l'installazione è possibile scegliere quali elementi installare.

Comandi utili
ps -adef | grep httpd Visualizza se fra i processi in esecuzione c'è Apache (dovrebbero vedersi varie righe con httpd, una per ogni child in esecuzione, oltre al httpd padre di tutti i processi (l'unico eseguito come root).

netstat -natp Visualizza tutte le connessioni Internet esistenti sul sistema, tra cui quelli in LISTENING sul sistema. Per ogni connessione si mostra anche il processo che la gestisce.

ldd /usr/sbin/httpd o **ldd /usr/local/apache/bin/httpd** (o **ldd /path/httpd**) Visualizza tutte le librerie dinamiche utilizzate da Apache. Utile per capirne le dependencies.

strace -p PID (Dove il PID è quello di un child di Apache) Traccia le system call di un singolo processo.

strace apachectl start Traccia le system call del processo che avvia Apache, utile per diagnosticare eventuali problemi all'avvio di Apache (verificare prima i log).

lsof | grep httpd Visualizza tutti i file aperti da Apache. Utile per verificare, tra l'altro dove sono i log e quali moduli sono utilizzati.

La struttura

File name	Description
/usr/sbin/httpd	server binary
/etc/httpd	directory containing server configuration files
/etc/httpd/conf	directory containing main configuration files
/etc/httpd/conf.d	directory containing configuration files for individually packaged modules, like ssl, php, perl etc
/etc/httpd/logs	symbolic link to /var/log/httpd
/etc/httpd/modules	symbolic link to /usr/lib/httpd/modules
/usr/lib/httpd/modules	server modules
/var/log/httpd	server log
/var/www/html	public html files

Il setup di base

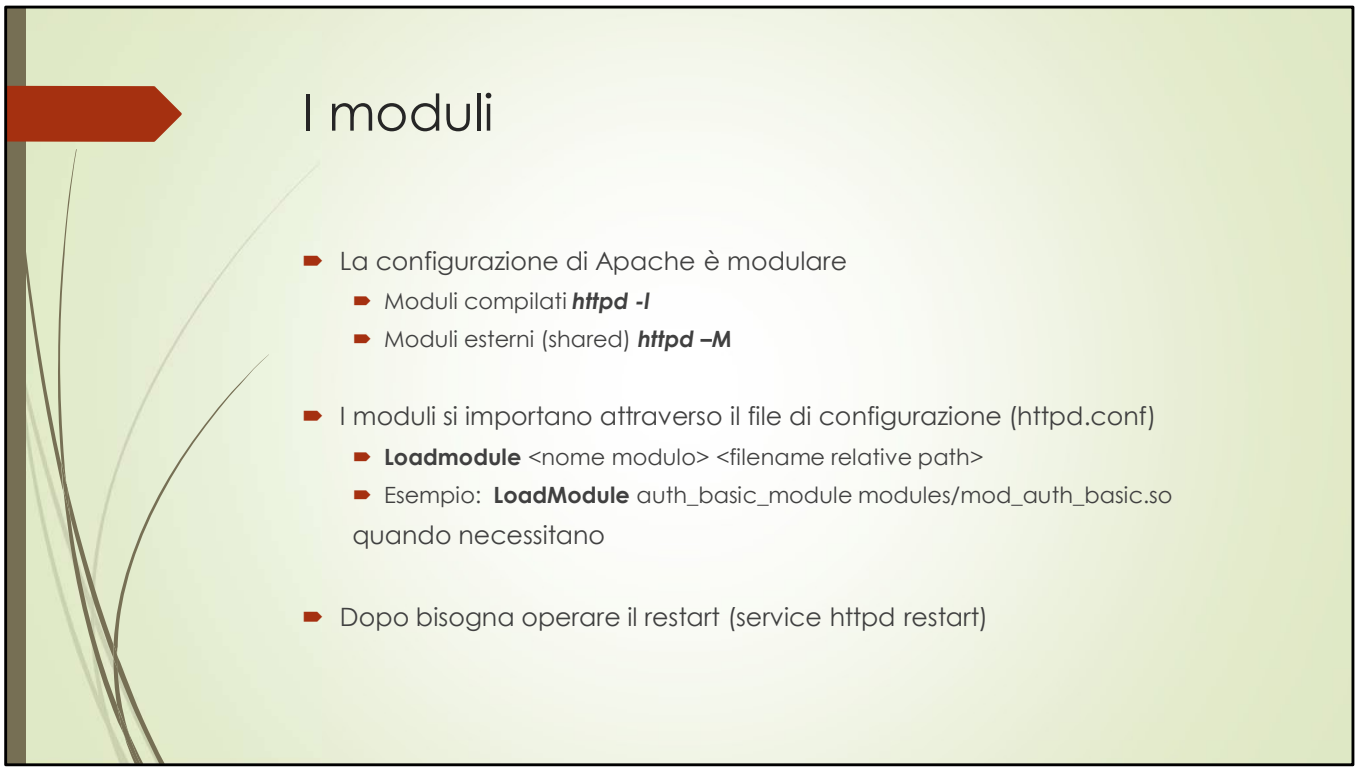
Editando http.conf

- **ServerRoot**, la directory principale in cui trovare i file di gestione (config, log, error)
- **PidFile**, il nome del file utilizzato per la gestione dei processi
- **ServerName**, il nome a dominio del server
- **DocumentRoot**, dove sono collocate le pagine web da mostrare
- **ErrorLog**, quale file utilizzare per gli errori
- **Listen**, la porta su cui attendere le richieste

Esempio

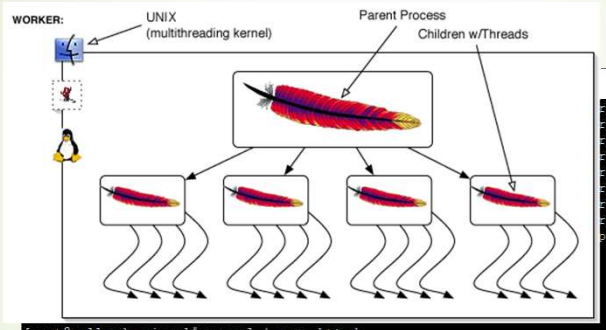
```
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
KeepAlive Off
Listen 8080
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule rewrite_module modules/mod_rewrite.so
...
Include conf.d/*.conf
ServerAdmin tech@digitalcontest.it
ServerName collaudo.digitalcontest.it
DocumentRoot "/var/www/html"
<Directory />
   Options FollowSymLinks
   AllowOverride None
</Directory>
DirectoryIndex index.html index.html.var index.shtml index.cfm index.php index.htm
AccessFileName .htaccess
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
   MIMEMagicFile conf/magic
</IfModule>
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D \"%{Host}i\" \"%{X-
Forwarded-For}i\"" custom
```

https://httpd.apache.org/docs/2.4/mod/mod_log_config.html

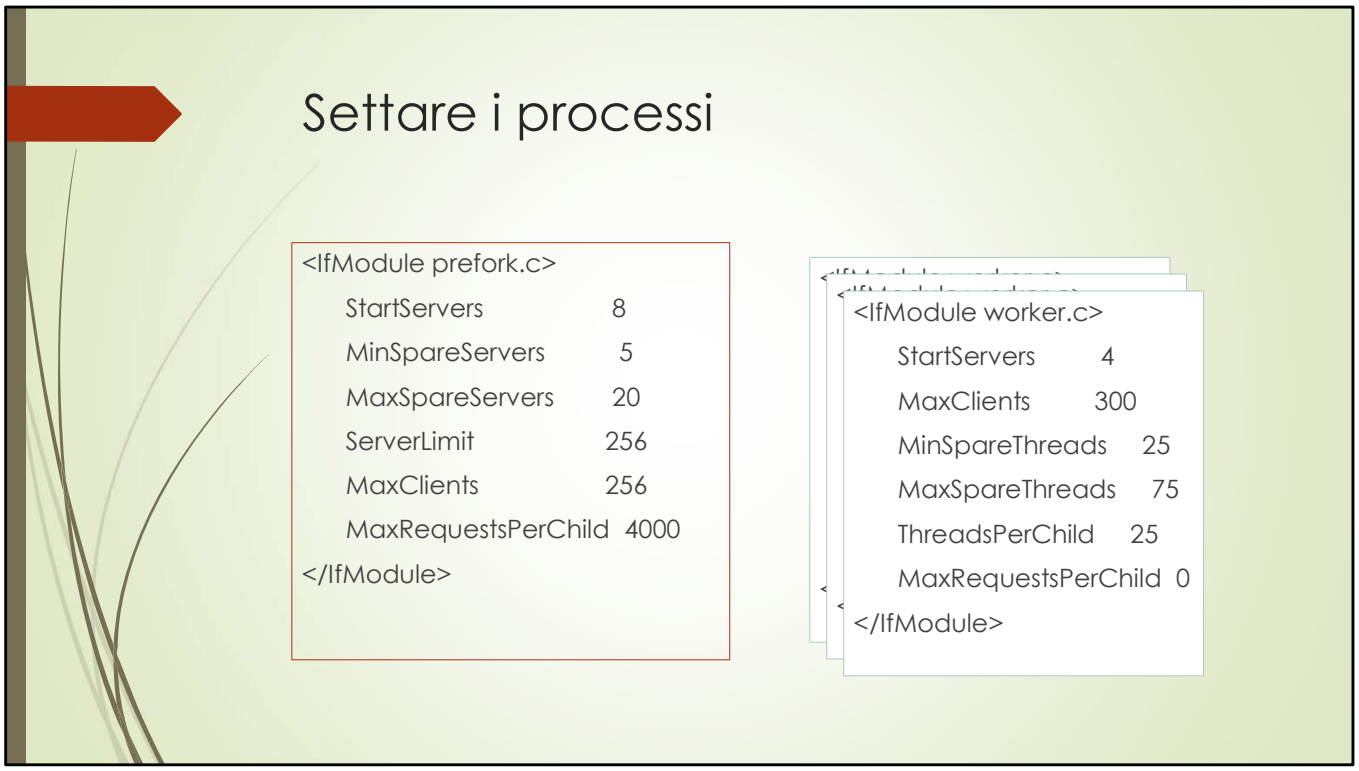


mod_access Questo modulo fornisce il controllo dell'accesso basato sul nome host del client, Indirizzo IP o altre caratteristiche della richiesta del client.
mod_dir Questo modulo fornisce l'interfaccia per i reindirizzamenti e la directory di servizio indici.
mod_perl Questo modulo consente di creare contenuto dinamico prodotto da script Perl servito alle richieste in entrata senza utilizzare l'interprete Perl ogni volta, riducendo il sovraccarico e il carico di sistema. Questo viene fatto incorporando un interprete Perl in il server Apache. Il modulo può anche emulare un ambiente CGI, permettendo il riutilizzo degli script CGI (Common Gateway Interface) senza alcuna modifica il set up.
mod_python Consente l'integrazione del linguaggio di programmazione Python nel server Apache. Ha lo scopo di sostituire CGI come metodo di esecuzione Script Python su un server web. Offre un'esecuzione molto più veloce e consente dati da mantenere su più sessioni.
mod_ssl Questo modulo fornisce un'interfaccia alla libreria OpenSSL, consentendo l'uso di Secure Socket Layer (SSL) e Transport Layer Security (TSL) secure protocolli di comunicazione. Ciò consente di eseguire un server Web che verrà eseguito sessioni crittografate con i client, consentendo uno scambio sicuro di potenzialmente sensibili dati.
mod_auth_basic Questo modulo fornisce la basic autenthication
Per un elenco dettagliato dei moduli disponibili e delle relative funzionalità, fare riferimento a: https://httpd.apache.org/docs/2.2/mod/.

Running : Service httpd start




```
[root@collaudo views]# ps axl | grep httpd
5     0 23472     1  20   0 504812 19568 poll_s Ss   ?          0:00 /usr/sbin/httpd
5    48 23474 23472  20   0 214408  4348 poll_s S    ?          0:00 /usr/sbin/httpd
5    48 23475 23472  20   0 522036 21076 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23476 23472  20   0 521488 20512 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23477 23472  20   0 522036 21012 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23478 23472  20   0 521360 20308 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23479 23472  20   0 521492 20520 ep_pol S    ?          0:00 /usr/sbin/httpd
5    48 23480 23472  20   0 523716 22652 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23481 23472  20   0 522164 21060 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 23482 23472  20   0 508512 17188 sys_se S    ?          0:00 /usr/sbin/httpd
5    48 25258 23472  20   0 508512 17060 sys_se S    ?          0:00 /usr/sbin/httpd
0     0 26221 19064  20   0 107440   920 pipe_w S+   pts/0      0:00 grep httpd
[root@collaudo views]#
```



prefork MPM
StartServers: number of server processes to start
MinSpareServers: minimum number of server processes which are kept spare
MaxSpareServers: maximum number of server processes which are kept spare
ServerLimit: maximum value for MaxClients for the lifetime of the server
MaxClients: maximum number of server processes allowed to start
MaxRequestsPerChild: maximum number of requests a server process serves

worker MPM
StartServers: initial number of server processes to start
MaxClients: maximum number of simultaneous client connections
MinSpareThreads: minimum number of worker threads which are kept spare
MaxSpareThreads: maximum number of worker threads which are kept spare
ThreadsPerChild: constant number of worker threads in each server process
MaxRequestsPerChild: maximum number of requests a server process serves

La configurazione di una directory istruisce Apache su come gestire quella specifica directory

Opzioni :

- FollowSymLinks – indica che il ws può seguire i link simbolici sul file system
- Indexes - istruisce il ws di creare un'indice della directory se non esiste un index.html (da usare con molta cautela ed evitare se possibile)

Direttive :

- AllowOverride (All | None | Option) – Indica se considerare direttive nei file .htaccess presenti nelle varie directory, default della 2.4 none, default precedenti all
- Order indica come gestire l'accesso ai file della directory, in quale ordine
  - allow, deny indica che primariamente si applicano le regole di allow e poi quelle di deny (esempio blacklist)
  - deny, allow, al contrario applica prima le regole di deny e poi quelle di allow, esempio white-list

La possibile dichiarazione di client consentiti o negati nella direttiva può essere fatta tramite nome host, dominio nome, indirizzo IP, indirizzo IP parziale e altro. A esempio qui limitiamo l'accesso a directory negando l'accesso a tutti - e solo permettendo di

accedere da un'altra macchina sulla LAN :

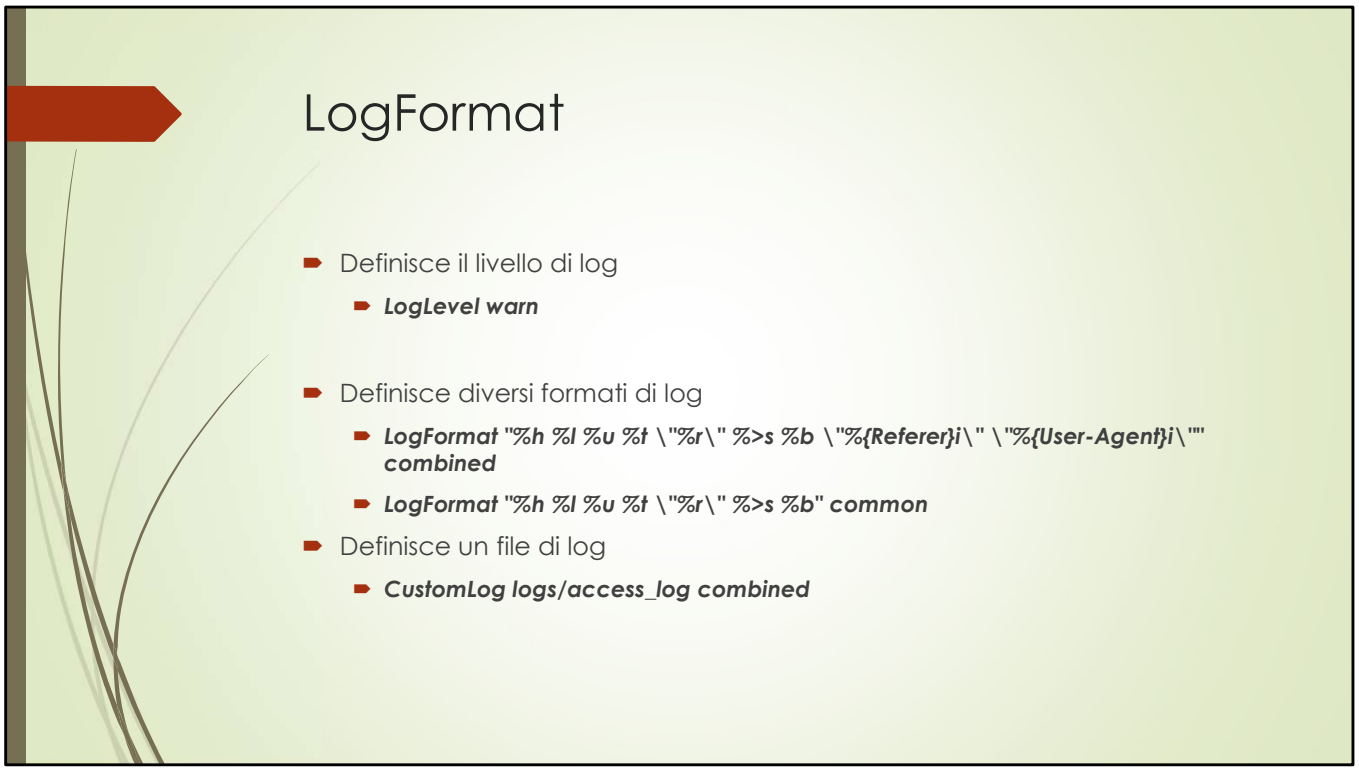
Order deny, allow
Deny from all
Allow from 192.168.1

Attenzione ! Se si rovescia l’ordine di esecuzione la directory viene bloccata

Files si applica a singoli nomi di file come descritti da un nome od un'espressione regolare (utilizzando il carattere ~). Analogamente FilesMatch gestisce espressioni regolari.
Nell'esempio si blocca l'accesso ai file di configurazione .htaccess ed .htpassword

Le espressioni regolari sono utilizzate per descrivere la struttura di una stringa, il confronto tra la stringa e l'espressione determina se questa è corretta.
Rapida spiegazione delle espressioni regolari
https://support.google.com/a/answer/1371415?hl=it
Se volete giocare con le espressioni regolari : https://regex101.com/

Esempio la descrizione di un numero cellulare corretto ([+39|0039]){0,1}3([0-9]){2}([0-9]){6,7}



%h (host) Hostname del cliente o indirizzo IP .
%l (*ident)* se è abilitato mod_ident l'idientificativo dell'utente
%u (*authuser*) se il documento è protetto da password l'identità dell'utente
%t (*date)* la data in formato :[*day/month/year:hour:minute:second tzoffset*].
%r (*request)* il testo della richiesta
%s (status) status code inviato al client su tre cifre
%s stato della prima richiesta , %>s stato dell'ultima richiesta in caso di redirect
%b (*bytes)* I byte trasmessi , header esclusi

%{Stringa}i , il contenuto della Stringa trasmessa al server

Ogni formato può essere accompagnato da una condizione sullo stato della richiesta , ad esempio %400r logga la richiesta solo per uno stato 400, %!200r logga la richiesta solo se lo stato è diverso da 200

LogFormat "%400,501{User-agent}i" browserlog
LogFormat "%!200,304,302{Referer}i" refererlog

Log Condizionali

- Si impostano variabili di ambiente basandosi sulla request
  - ***SetEnvIf*** *(Remote_Host , Remote_Addr, Server_Addr , Request_Method, Request_Protocol, Request_URI)*

```
SetEnvIf Request_URI "\.gif$" is_image
SetEnvIf Request_URI "\.jpg$" is_image
```

- Si condiziona il log ad una variabile di ambiente preimpostata

```
CustomLog logs/access_log common env=!is_image
```

- *Esiste anche la possibilità di condizionare il log allo status https*

```
LogFormat "%400,501{User-agent}i" browserlog
LogFormat "%!200,304,302{Referer}i" refererlog
```

Basic authentication

- .htaccess nella directory

```
AuthType Basic
AuthName "Per favore autenticatevi"
AuthUserFile "/etc/httpd/conf/.htpasswd
require valid-user
```

- Nel file di configurazione, anche di un virtual host (vedi dopo)

```
<Directory "/var/www/html/esempio/report">
        AuthType Basic
        AuthName "Autenticazione Richiesta"
        AuthUserFile "/etc/httpd/conf/.esempiopwd"
        Require valid-user
        Order allow,deny
        Allow from all
        DirectoryIndex <file>.php
 </Directory>
```

- La Basic Authentication si usa solo in https

Per prima cosa, accederemo alla directory in cui intendiamo posizionare il file degli utenti.
Può essere qualsiasi directory, **ma deve essere al di fuori di DocumentRoot**, quindi è così quindi non visibile dai tuoi clienti. Dovrebbe avere le autorizzazioni impostate su 0644, in modo da essere modificabile da root e da nessun altro.
Vengono aggiunti utenti e password il file eseguendo il comando **htpasswd**:
htpasswd -c nome utente .htpasswd

Attenzione per gli utenti successivi si deve omettere il –c, che serve a creare il file, altrimenti si perdono gli utenti precedenti

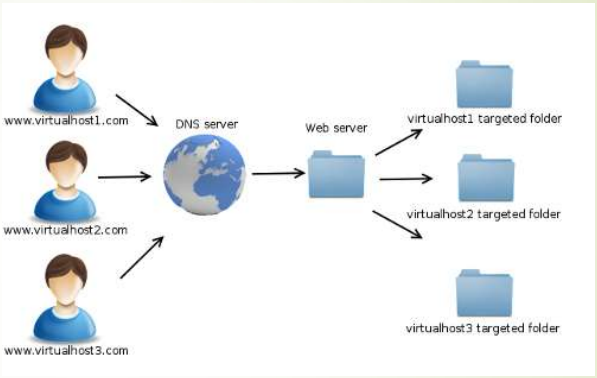
Per cancellare un utente : ***htpasswd -D [filename] [username]***

Esempio di htpasswd

balaton:$apr1$rIDolRej$ji694ElgfK0CrKKuvsnzl.
gianni:$apr1$RDWaURC8$vFgBQKslKnBLMoGkMuqbJ.

Virtual Host

```
.....
    ServerAdmin webmaster@digitalcontest.it
    HostnameLookups Off
    UseCanonicalName Off
    ServerSignature Off
.....
<VirtualHost *:80>
    ServerName: corso.dcinfo.it
    DocumentRoot /var/www/html/corso
    ErrorLog logs/corso.dcinfo.it-error_log
    CustomLog logs/corso.dcinfo.it-access_log common
    Header set Access-Control-Allow-Origin "*"
    <Directory "/var/www/html/corso/">
        Options FollowSymLinks
        AllowOverride All
        Order allow,deny
            Allow from all
    </Directory>
</VirtualHost>
<VirtualHost *:80>
    ServerName: test.dcinfo.it
    DocumentRoot /var/www/html/test
    ....
</VirtualHost>
```



In realta davanti c'e un NGINX (poxy gateway) che manda il traffico destinato a corso.dcinfo.it sulla porta 8092

```
<VirtualHost *:8092>
..
    ServerName corso.dcinfo.it
    ErrorLog logs/corso.dcinfo.it-error_log
    CustomLog logs/corso.dcinfo.it-access_log common
    HostnameLookups Off
    UseCanonicalName Off
    ServerSignature Off
    Header set Access-Control-Allow-Origin "*"
    <Directory "/var/www/html/corso/">
        Options FollowSymLinks
        AllowOverride All
        Order allow,deny
            Allow from all
    </Directory>
</VirtualHost>
```

Secure Web Server

- Un client tenta di connettersi alla porta 443 sul server Web protetto.
- Il client invia un elenco dei metodi di crittografia disponibili che supporta; TSL o SSL.
- Il server sceglierà il metodo di crittografia più potente disponibile per entrambi i lati possono supportare.
- Il server restituirà quindi al client il suo certificato e la chiave di crittografia pubblica.
- La chiave pubblica verrà utilizzata dal client per generarla proprio hash di crittografia se sceglie di accettare il certificato del server
- Se il client accetta la connessione, invierà indietro un hash crittografato con la chiave pubblica del server. Questo hash verrà utilizzato per crittografare tutte le comunicazioni tra il server e il client la sessione.
- Prerequisiti:
  - Mod_ssl
  - OpenSLL (? rpm –q OpenSSL)

Serve un certificato ..

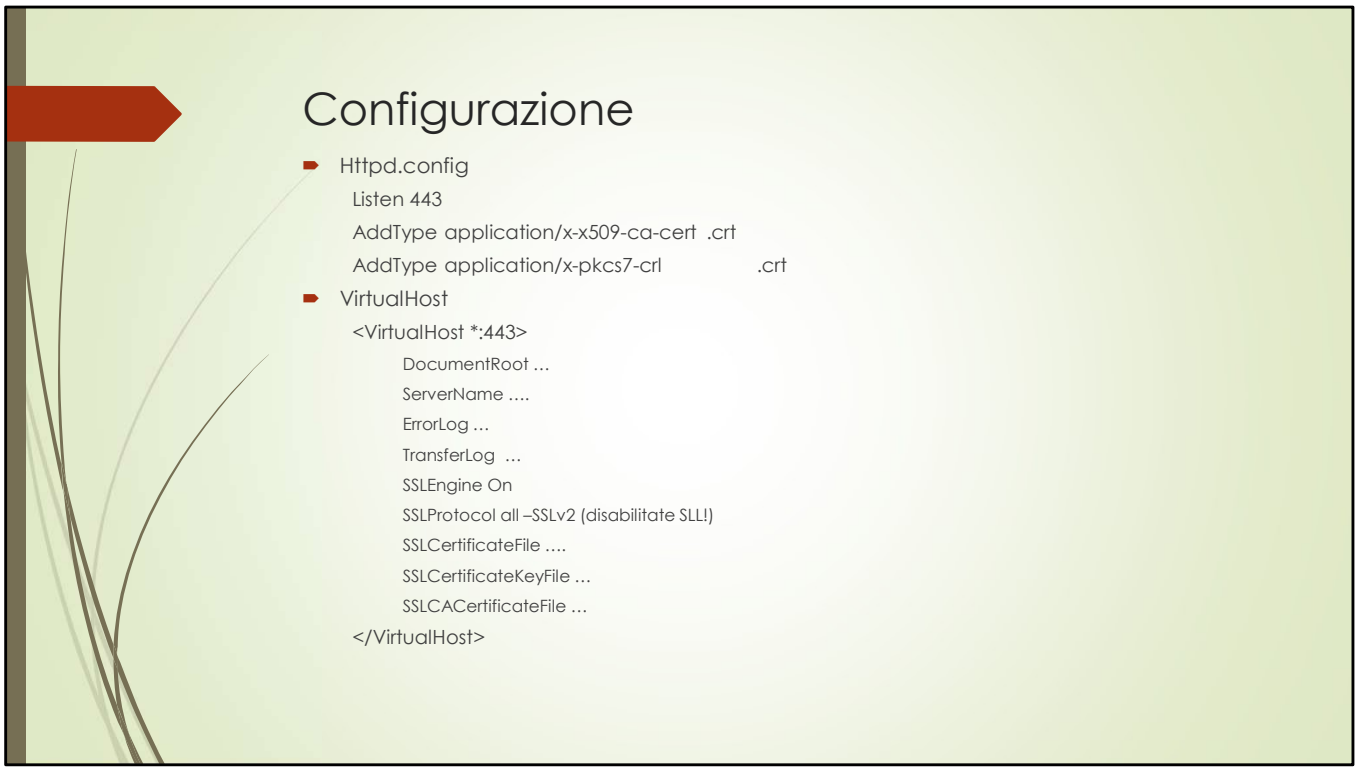
Let's Encrypt

- CA 'Open Source' fornisce certificati gratuitamente
- Utilizza il protocollo ACME (Automatic Certificate Management Environment)
- Si appoggia su Certbot
- https://letsencrypt.org/it/getting-started/

- Alternativa certificati autoprodotti -> diventate CA

Certificati autoprodotti

- Verifica esistenza OpenSSL
  - rpm -q openssl
- Creare la CA
  - Creare la private key
    - openssl genrsa -des3 -out myca.key 4096
  - Creare la CA
    - openssl req -new -x509 -days 365 -key myca.key -out myca.crt
- Creare il certificato server
  - Creare la private key
    - openssl genrsa -des3 -out server.key 4096
  - Creare il certificato server
    - openssl req -new -key server.key -out server.csr
  - Firmare il certificato server con la CA
    - Openssl X509 –req –days 365 –in server.csr –CA myca.crt –Cakey myca.key –set_serial 01 –out server.crt
- Testare tutto quanto
  - openssl x509 -noout -text -in <file da testare>



Potete usare Let's Encrypt per generare un certificato (https://letsencrypt.org/) od installare un certificato autofirmato (sconsigliato, vedi www.dedoimedo.com-apache-web-server-lm.pdf capitolo 5)

Il processo di rilascio e gestione dei certificati LetsEncrypt può essere automatizzato con Certbot (https://certbot.eff.org/)

Un caso d'uso – http to https

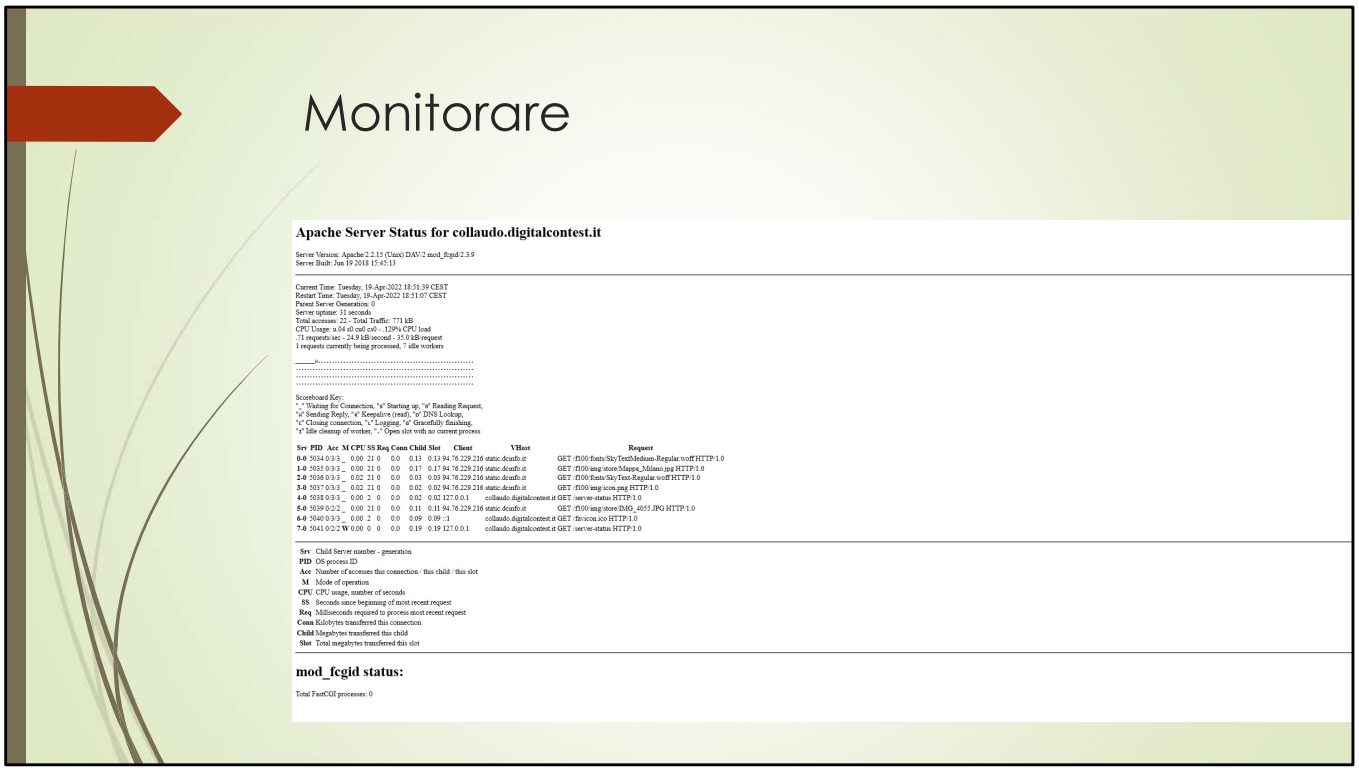
Vogliamo un sito che risponda solo in https

```
<VirtualHost *:80>
    ServerName  example.com
    ServerAlias www.example.com
    Redirect  permanent / https://example.com/
</VirtualHost>

<VirtualHost *:443>
    ServerName example.com
    ServerAlias www.example.com
    SSLProtocols  TLSv1.1 TLSv1.2

    # SSL Configuration
    # Other Apache Configuration
</VirtualHost>
```

SSL Configuration

Certificato del server : SSLCertficateFile <path al certificato>
Chiave privata del certificato: SSLCertificateKeyFile <path della file di chiave privata>
Certificato della CA : SSLCerticateFile <path del file>

Monitorare



Apache Server Status for collaudo.digitalcontest.it

Server Version: Apache/2.2.15 (Unix) DAV/2 mod_fcgid/2.3.9
Server Built: Jun 19 2018 15:45:13

Current Time: Tuesday, 19-Apr-2022 18:51:39 CEST
Restart Time: Tuesday, 19-Apr-2022 18:51:07 CEST
Parent Server Generation: 0
Server uptime: 31 seconds
Total accesses: 22 - Total Traffic: 771 kB
CPU Usage: u.04 s0 cu0 cs0 - .129% CPU load
.71 requests/sec - 24.9 kB/second - 35.0 kB/request
1 requests currently being processed, 7 idle workers

_____W.....
.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	5034	0/3/3	_	0.00	21	0	0.0	0.13	0.13	94.76.229.216	static.dcinfo.it	GET /f100/fonts/SkyTextMedium-Regular.woff HTTP/1.0
1-0	5035	0/3/3	_	0.00	21	0	0.0	0.17	0.17	94.76.229.216	static.dcinfo.it	GET /f100/img/store/Mappa_Milano.jpg HTTP/1.0
2-0	5036	0/3/3	_	0.02	21	0	0.0	0.03	0.03	94.76.229.216	static.dcinfo.it	GET /f100/fonts/SkyText-Regular.woff HTTP/1.0
3-0	5037	0/3/3	_	0.02	21	0	0.0	0.02	0.02	94.76.229.216	static.dcinfo.it	GET /f100/img/icon.png HTTP/1.0
4-0	5038	0/3/3	_	0.00	2	0	0.0	0.02	0.02	127.0.0.1	collaudo.digitalcontest.it	GET /server-status HTTP/1.0
5-0	5039	0/2/2	_	0.00	21	0	0.0	0.11	0.11	94.76.229.216	static.dcinfo.it	GET /f100/img/store/IMG_4055.JPG HTTP/1.0
6-0	5040	0/3/3	_	0.00	2	0	0.0	0.09	0.09	::1	collaudo.digitalcontest.it	GET /favicon.ico HTTP/1.0
7-0	5041	0/2/2	W	0.00	0	0	0.0	0.19	0.19	127.0.0.1	collaudo.digitalcontest.it	GET /server-status HTTP/1.0

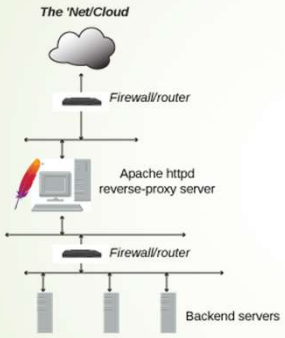
Srv Child Server number - generation
PID OS process ID
Acc Number of accesses this connection / this child / this slot
M Mode of operation
CPU CPU usage, number of seconds
SS Seconds since beginning of most recent request
Req Milliseconds required to process most recent request
Conn Kilobytes transferred this connection
Child Megabytes transferred this child
Slot Total megabytes transferred this slot

mod_fcgid status:

Total FastCGI processes: 0

```
<Location "/server-status">
        SetHandler server-status
        Require ip xxx.xxx.xxx.xxx
</Location>
```

Reverse Proxy (mod_proxy)




```
ProxyPass "/"          "http://worker1.net/"
ProxyPassReverse "/"          "http://worker1.net/"

ProxyPass "/images"          "http://local.images.net/"
ProxyPassReverse "/images" "http://local.images.net/"
```

Bilanciamento

```
ProxyPass "/" "balancer://myset/"
```

Balancer (mod_balancer)

```
<Proxy balancer://myset>
      BalancerMember http://www2.example.com:8080
     BalancerMember http://www3.example.com:8080
     ProxySet lbmethod=bytraffic
 </Proxy>

ProxyPass "/images/" "balancer://myset/"
ProxyPassReverse "/images/" "balancer://myset/"
```

Bilanciamento non simmetrico

```
<Proxy balancer://myset>
     BalancerMember http://www2.example.com:8080
     BalancerMember http://www3.example.com:8080 loadfactor=3 timeout=1
     ProxySet lbmethod=bytraffic
</Proxy>
```

Failover (mod_proxy_hcheck)

```
<Proxy balancer://myset>
    BalancerMember http://www2.example.com:8080
    BalancerMember http://www3.example.com:8080 loadfactor=3 timeout=1
    BalancerMember http://spare1.example.com:8080 status=+R
    BalancerMember http://spare2.example.com:8080 status=+R
    BalancerMember http://hstandby.example.com:8080 status=+H
    BalancerMember http://bkup1.example.com:8080 lbset=1
    BalancerMember http://bkup2.example.com:8080 lbset=1
    ProxySet lbmethod=byrequests
</Proxy>


ProxyPass "/images/"            "balancer://myset/"
ProxyPassReverse "/images/"  "balancer://myset/"
```

Permessi consigliati* (D: directory, F:file)

/usr/sbin/httpd F: 511
/etc/httpd D: 751
/etc/httpd/conf D: 751, F: 644 / 600
/etc/httpd/conf.d D: 751, F: 644 / 600
/etc/httpd/logs symbolic link, 755 / 711
/etc/httpd/modules symbolic link, 755 / 711
/etc/httpd/run symbolic link, 755 / 711
/usr/lib/httpd/modules D: 751, F: 644 / 600
/var/log/httpd D: 751, F: 644 / 600
/var/run D: 751, F: 644 / 600
/var/www/html D: 755, F: 755

***vedi appendice per comprensione**

Security 2

- Bloccate gli accessi alla root ('/')

```
<Directory />
    Order Deny, Allow
    Deny from all
</Directory
```

- Disabilitate gli accessi agli .htaccess, ai file di log, alle directory di servizio (esempio include), a file utilizzati per i parametri di accesso (esempio .inc)
- Togliete il listing delle directory
- Disabilitare i CGI

```
Options -ExecCGI
```

- Rendete sicuri i certificati

I file *server.crt* e *server.key* devono essere accessibili solo da root, meglio se si disabilita la possibilità di modica (permissions a 400) .
Il file *ssl.conf* dovrebbe avere permessi 640 o 600.

Appendice – diritti sui file

- I file appartengono sempre ad un utente che appartiene ad un gruppo
- I file hanno una descrizione binaria dei diritti raggruppata in tre gruppi
  - Owner
  - Group
  - World
- Per ogni gruppo esistono diritti di lettura, modifica, esecuzione
- I tre gruppi sono rappresentati da tre numeri binari, ogni numero rappresenta in modo posizionale un diritto con un bit accesso spento

OWNER			GROUP			WORLD		
R	W	X	R	W	X	R	W	X
1	1	1	1	0	1	1	0	0

→ **754**

Appendice - comandi Linux utili

- List di una directory : **ls [nome file] opzioni utili : -ltr , -ls , -s**
- Paginazione di un file : **more [nome file]**
- Listing di un file : **cat [nome file]**
- Monitoraggio continuo di un file : **tail [nome file] opzioni utili -xxxf**
- Ricerca in un file : **grep –i [stringa] [nome file]**
- Cambio directory : **cd [path], path particolari:**
  - **. directory corrente**
  - **.. directory padre**
- Directory corrente : **pwd**
- Manuale su un comando : **man [comando]**
- Storia dei comandi : **history**
- Cambio privilegi : **chmod [opzioni] [file]**
- Cambio owner : **chown [owner]:[group] [nome file]**
- **Pipelining tra comandi |**
- **Ridirezione di un output >, >>**
- **Task in background &**

.. Un sacco di siti, ad esempio https://www.hostinger.com/tutorials/linux-commands, fanno il listing dei comandi più utili

Appendice – Regular Expression (RegEx)

- Un'**espressione regolare** è una sequenza di simboli che identifica un insieme di stringhe
  - **.** Trova un singolo carattere
  - **[ ]** Trova un singolo carattere contenuto nelle parentesi. Ad esempio, [abc] trova o una "a", "b", o "c". [a-z] è un intervallo
  - **[^]** Trova ogni singolo carattere non incluso nelle parentesi. Ad esempio, [^abc] trova ogni carattere diverso da "a", "b", o "c".
  - **^** Corrisponde all'inizio della stringa (o di ogni riga della stringa, quando usato in modalità multilinea)
  - **$** Corrisponde alla fine della stringa o alla posizione immediatamente precedente un carattere di nuova linea (o alla fine di ogni riga della stringa, quando usato in modalità multilinea)
  - **( )** Definisce una "sottoespressione marcata". Il risultato di ciò che è incluso nell'espressione, può essere richiamato in seguito.
  - **\n** Dove n è una cifra da 1 a 9; trova ciò che la nesima sottoespressione ha trovato. Tale costrutto, detto backreference, estende le potenzialità delle regexp oltre i linguaggi regolari e non è stato adottato nella sintassi estesa delle regexp.
  - ***** Un'espressione costituita da un singolo carattere seguito da "*", trova zero o più copie di tale espressione. Ad esempio, "[xyz]*" trova "", "x", "y", "zx", "zyx", e così via.
  - **\n***, dove n è una cifra da 1 a 9, trova zero o più iterazioni di ciò che la nesima sottoespressione ha trovato. Ad esempio, "(a.)c\1*" trova "abcab" e "accac" ma non "abcac".
  - **{x,y}** Trova l'ultimo "blocco" almeno x volte e non più di y volte. Ad esempio, "a{3,5}" trova "aaa", "aaaa" o "aaaaa".

Esempio `\.jpg$`
`\.(jpg|png|bmp)$`

RegEx - 2 - Quantificatori

***** Cerca l'occorrenza (zero o più volte) del carattere o insieme di caratteri cui segue:

- abc* identifica ab seguito da zero o più c come in ab, abc, abcc, abccc

+ Cerca l'occorrenza (una o più volte) del carattere o insieme di caratteri cui segue:

- ab[ce]+ identifica ab seguito da una o più c oppure una o più e come in abc, abec, abccc, abcceeeccccce

? Cerca l'occorrenza (zero o una volta) del carattere o insieme di caratteri cui segue:

- abc? identifica ab seguito o meno da una c come in abc e ab

{m, n} Cerca l'occorrenza (da m a n volte; m lasciato vuoto è zero, n lasciato vuoto infinito) del carattere, insieme di caratteri o sotto-regex cui segue:

- (ab){1,2} identifica le sequenze di uno o due ab come in ab e abab

https://regex101.com – disegna il RegEx del codice fiscale

RegEx - 3 – Classi predefinite

Classe POSIX	sintassi normale	significato
[:upper:]	[A-Z]	lettere maiuscole
[:lower:]	[a-z]	lettere minuscole
[:alpha:]	[A-Za-z]	lettera sia maiuscole che minuscole
[:alnum:]	[A-Za-z0-9]	numeri e lettere maiuscole e minuscole
[:digit:]	[0-9]	numeri
[:xdigit:]	[0-9A-Fa-f]	numeri in formato esadecimale
[:punct:]	[!\-"#$%&'()*+,.\/:;<=>?@[\]\^_`{|}~\\]	segni di interpunzione
[:blank:]	[\t]	spazio o TAB
[:space:]	[\t\n\r\f\v]	caratteri vuoti
[:cntrl:]	[\x00-\x1F\x7F]	caratteri control
[:graph:]	[^ \t\n\r\f\v]	caratteri non vuoti
[:print:]	[^\t\n\r\f\v]	caratteri non vuoti e spazi