



FINTECH

Unità Formativa (UF): Linux Server

Docente: Wolfgang Cecchin

Titolo argomento: Installazione di un http server



Negoziazione Client-Server e status code

Riferimenti:

Architettura

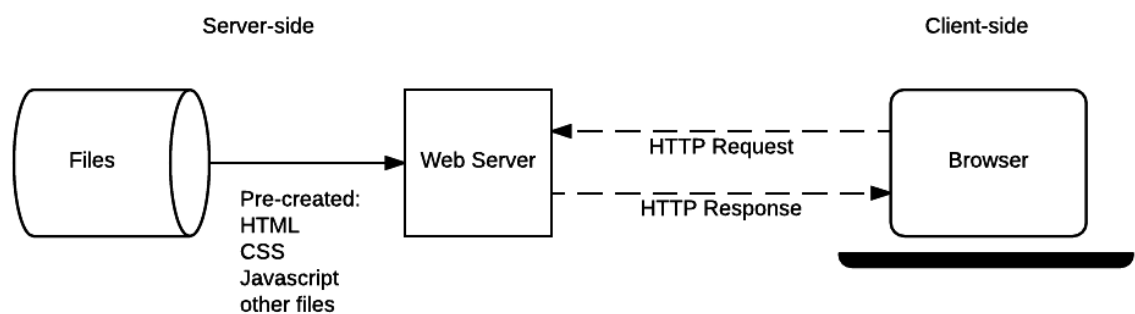
https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview

Codici di stato

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

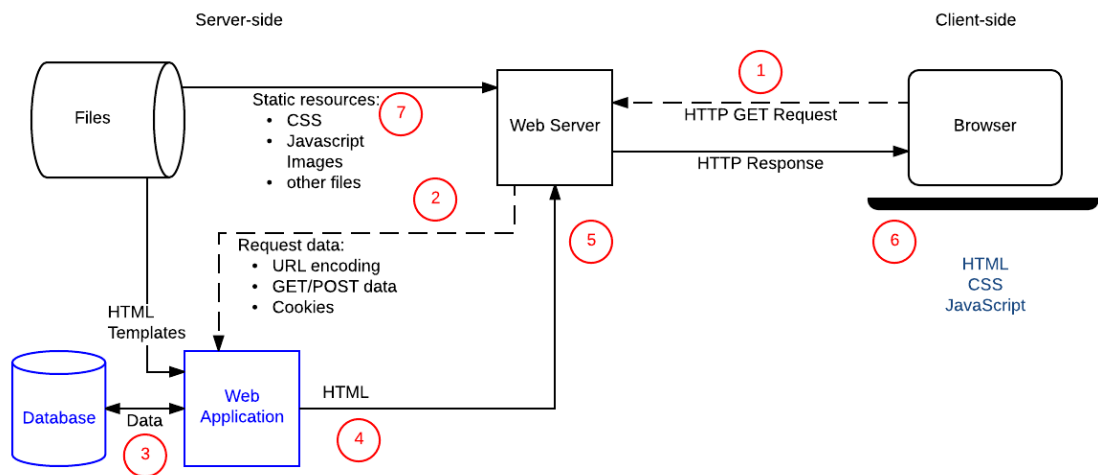
https://it.wikipedia.org/wiki/Codici_di_stato_HTTP

Architettura statica:





Architettura dinamica:





“Configurazione” di un webserver

- Assegniamo un nome al server;
- Troviamo uno spazio per mettere i files;
- Diciamo al webserver come associare il nome ai files da servire;
- Configuriamo i log.



Installazione nginx

```
$ sudo apt update
```

```
$ sudo apt install nginx
```

Far partire, riavviare, fermare, conoscere lo stato di Nginx

```
$ sudo systemctl start nginx
```

```
$ sudo systemctl restart nginx
```

```
$ sudo systemctl stop nginx
```

```
$ sudo systemctl status nginx
```

Di default Nginx è in ascolto sulla porta 80

(di default anche Apache!).

Testare e ricaricare la configurazione di nginx:

```
$ sudo nginx -t
```

```
$ sudo systemctl reload nginx
```



Configurazione di nginx

I file di configurazione si trovano in:

`/etc/nginx`

La configurazione dei siti si trova in:

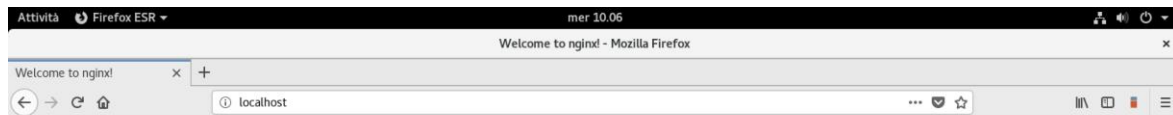
`/etc/nginx/sites-available`

Esiste una configurazione di default, quindi se io digito nel browser della mia macchina locale un sito che ho impostato in `/etc/hosts`, ad esempio:

<http://localhost>

<http://mercoledì.local>

mi risponde la pagina di default di Nginx.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Creiamo una directory in cui inserire i file del sito.

Ad esempio:

`/var/www/mercoledì.local`

oppure:

`/srv/www/mercoledì.local`

La configurazione del sito devo inserirla in:

`/etc/nginx/sites-available`

ma il sito va attivato con link simbolico in:

`/etc/nginx/sites-enabled`

Configurazione minima del file:

`/etc/nginx/sites-available/mercoledì.local.conf`

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name mercoledì.local;  
  
    root /var/www/mercoledì.local;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

In questo modo sto dicendo che chiamando

<http://mercoledì.local>

mi vengono serviti i files che si trovano nella directory:

`/var/www/mercoledì.local`

Attenzione! Non abbiamo ancora attivato il sito!



Configuriamo i log del web server

/var/log è la directory in cui si trovano i files di log dei principali servizi su Linux. I log sono gestiti dal servizio (detto anche “demone”) syslogd, che tra le altre cose si occupa anche della compressione.

Aggiungiamo i log specifici per questo sito:

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name mercoledi.local;  
  
    root /var/www/mercoledi.local;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
    access_log /var/log/nginx/mercoledi.local.log;  
    error_log /var/log/nginx/mercoledi.local.error.log;  
}
```



Codici di stato http

All'interno dei log sono memorizzate le risposte e gli errori del web server.

Esempio:

```
127.0.0.1 - - [21/Oct/2019:22:35:08 +0200] "GET / HTTP/1.1" 200 3437 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/65"
127.0.0.1 - - [21/Oct/2019:22:35:08 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:22:35:08 +0200] "GET /favicon.ico HTTP/1.1" 404 143 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101$
127.0.0.1 - - [21/Oct/2019:22:35:45 +0200] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.52.1"
127.0.0.1 - - [21/Oct/2019:22:43:17 +0200] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.52.1"
127.0.0.1 - - [21/Oct/2019:22:45:24 +0200] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.52.1"
127.0.0.1 - - [21/Oct/2019:23:02:31 +0200] "GET / HTTP/1.1" 200 24165 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/$
127.0.0.1 - - [21/Oct/2019:23:02:31 +0200] "GET /favicon.ico HTTP/1.1" 404 143 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101$
127.0.0.1 - - [21/Oct/2019:23:02:50 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:23:02:50 +0200] "GET /favicon.ico HTTP/1.1" 404 143 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101$
127.0.0.1 - - [21/Oct/2019:23:03:31 +0200] "GET / HTTP/1.1" 200 3437 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/65$
127.0.0.1 - - [21/Oct/2019:23:03:31 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:23:03:31 +0200] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"$
127.0.0.1 - - [21/Oct/2019:23:03:31 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:23:03:32 +0200] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"$
127.0.0.1 - - [21/Oct/2019:23:03:32 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:23:03:32 +0200] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"$
127.0.0.1 - - [21/Oct/2019:23:03:32 +0200] "GET /icons/openlogo-75.png HTTP/1.1" 404 143 "http://merc.apache.local/" "Mozilla/5.0 (X11; Lin$
127.0.0.1 - - [21/Oct/2019:23:10:55 +0200] "GET / HTTP/1.1" 200 106 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60$
127.0.0.1 - - [21/Oct/2019:23:20:02 +0200] "GET / HTTP/1.1" 200 110 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60$
127.0.0.1 - - [21/Oct/2019:23:20:02 +0200] "GET /favicon.ico HTTP/1.1" 404 243 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101$
```

Approfondimenti sui codici di risposta http:

https://it.wikipedia.org/wiki/Codici_di_stato_HTTP



Attivazione di un sito

Attiviamo il sito.

Ricordate?

La configurazione del sito devo inserirla in:

`/etc/nginx/sites-available`

ma il sito va attivato con link simbolico in:

`/etc/nginx/sites-enabled`

Per creare un link usiamo la sintassi:

```
$ sudo ln -s source nomeLink
```

Creiamo un link simbolico:

```
$ sudo ln -s /etc/nginx/sites-available/mercoledì.conf /etc/nginx/sites-enabled/mercoledì.conf
```

Verifico la configurazione e ricarico nginx:

```
$ sudo nginx -t
```

```
$ sudo systemctl reload nginx
```



Riepilogo

- Configuriamo `/etc/hosts`;
- Abbiamo creato una directory per il sito:
`/var/www/mercoledì.local`;
- Abbiamo configurato il sito inserendo la configurazione in:
`/etc/nginx/sites-available/mercoledì.local.conf`;
- Abbiamo impostato i log su:
`/var/log/nginx/mercoledì.local.log` e `mercoledì.local.error.log`
- Abbiamo attivato il sito creando un link che punta
da `sites/available` a `sites/enabled`;

Abbiamo verificato la configurazione e riavviato Nginx.



Configurazione di Apache 2

Normalmente si sceglie di utilizzare Nginx o Apache2. Tuttavia in casi particolari possiamo scegliere di averli entrambi sulla stessa macchina:

```
$ sudo apt install apache2
```

Modifichiamo le porte su cui ascolta Apache:
`/etc/apache2/ports.conf`

Modificando la seguente riga:
`Listen 8080`

Andiamo nella directory:
`/etc/apache2/sites-available`

Cambiamo la configurazione di default, cioè il file:
`000-default.conf`
Specificando:

```
<VirtualHost *:8080>
```

(Nota: se - al contrario - voglio cambiare le porte su cui ascolta Nginx devo invece modificare i seguenti file:

`nginx.conf`

alla riga:

```
listen 80 default_server
```

e la configurazione di default di Nginx).



Far partire, riavviare, fermare, conoscere lo stato di Apache2:

```
$ sudo systemctl start apache2
```

```
$ sudo systemctl restart apache2
```

```
$ sudo systemctl stop apache2
```

```
$ sudo systemctl status apache2
```

Testare e ricaricare la configurazione di Apache2:

```
$ sudo apachectl -t
```

```
$ sudo systemctl reload apache2
```



Ma chi è in ascolto sulla porta 80 e chi sulla 8080?

Installiamo il seguente pacchetto Debian:

```
$ sudo apt install net-tools
```

e lanciamo:

```
$ sudo netstat -tulpn
```

```
itsuser@debianITS:~$ su
Password:
root@debianITS:/home/itsuser# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      680/nginx: master p
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      547/sshd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      1138/exim4
tcp6       0      0 :::80                 :::*                   LISTEN      680/nginx: master p
tcp6       0      0 :::8080               :::*                   LISTEN      617/apache2
tcp6       0      0 :::22                 :::*                   LISTEN      547/sshd
tcp6       0      0 :::1:25               :::*                   LISTEN      1138/exim4
udp        0      0 0.0.0.0:1900           0.0.0.0:*               LISTEN      857/minissdpd
udp        0      0 0.0.0.0:34798          0.0.0.0:*               LISTEN      439/avahi-daemon: r
udp        0      0 0.0.0.0:68             0.0.0.0:*               LISTEN      585/dhclient
udp        0      0 0.0.0.0:5353           0.0.0.0:*               LISTEN      439/avahi-daemon: r
udp6       0      0 :::60054              :::*                   LISTEN      439/avahi-daemon: r
udp6       0      0 :::5353               :::*                   LISTEN      439/avahi-daemon: r
```



Configurazione di Apache 2

Vogliamo adesso creare un caso pratico:

vogliamo che all'host **merc.apache.local** (inventato or ora) risponda Apache (ma prima lo configureremo perché risponda con <http://merc.apache.local:8080>)

Creiamo in /etc/hosts un entry:

```
127.0.1.1      merc.apache.local
```

Chiamando:

<http://merc.apache.local>

Chi mi risponde?

Posso analizzare gli headers http per verificarlo.

Ad esempio, installando cURL (client for URL):

```
$ sudo apt install curl
```

```
$ curl -I merc.apache.local
```

HTTP/1.1 301 Moved Permanently

Server: nginx/1.16.0

Date: Mon, 21 Oct 2019 20:36:07 GMT

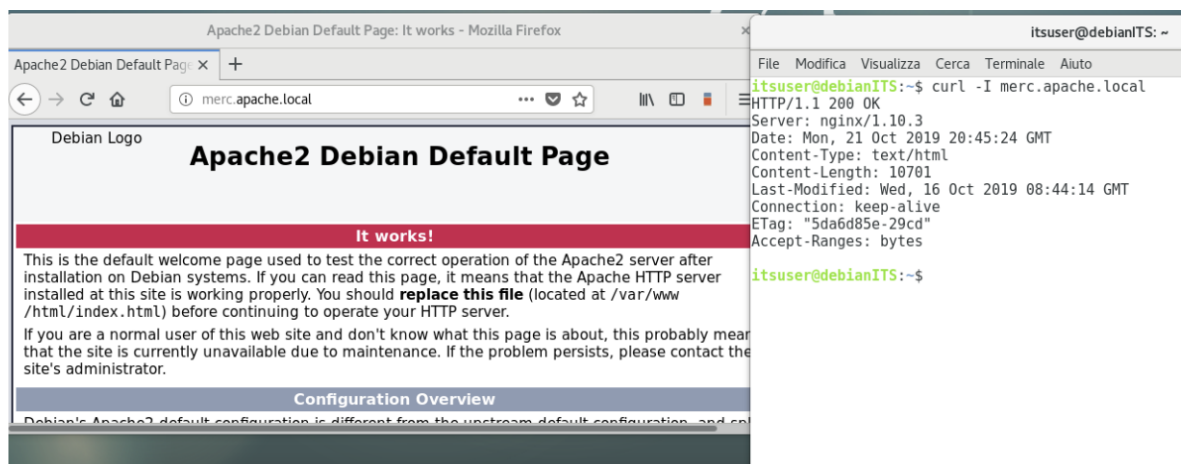
Mi sta rispondendo Nginx e non Apache.



Attenzione! L'installazione di Apache appena effettuata è andata a sovrascrivere la pagina di default di Nginx in

/var/www/html/index.html

In realtà solo il file HTML è stato sovrascritto: ci sta rispondendo Nginx che è in ascolto sulla porta 80, porta di default per il webserver.



Per evitare di essere di nuovo ingannati, creiamo una pagina html generica di default:

/var/www/html/index.html

```
<html>
  <head>
    <title>
      Pagina di default
    </title>
  </head>
  <body>
    <h1>Pagina di default</h1>
  </body>
</html>
```



Ma come configuriamo un sito con Apache?

Con la stessa procedura di Nginx, cioè:

- Configuriamo `/etc/hosts`;
- Creiamo una directory per il sito:
`/var/www/merc.apache.local`;
- Configuriamo il sito inserendo la configurazione in:
`/etc/apache2/sites-available/merc.apache.local.conf`;
- Impostiamo i log su:
`/var/log/apache2/merc.apache.local.log` e `merc.apache.local.error.log`
- Attiviamo il sito creando un link che punta
da `sites/available` a `sites/enabled`;

Verifichiamo la configurazione e riavviamo Apache.



Creiamo una directory in cui inserire i file del sito.

Ad esempio:

`/var/www/merc.apache.local`

La configurazione del sito devo inserirla in:

`/etc/apache2/sites-available`

ma il sito va attivato con link simbolico in:

`/etc/apache2/sites-enabled`

Configurazione minima del file:

`/etc/Httpd2/sites-available/merc.apache2.local.conf`

`<VirtualHost *:8080>`

`ServerName merc.apache.local`

`ServerAdmin webmaster@localhost`

`DocumentRoot /var/www/merc.apache.local`

`ErrorLog ${Apache2_LOG_DIR}/merc.apache.local.error.log`

`CustomLog ${Apache2_LOG_DIR}/merc.apache.local.access.log combined`

`</VirtualHost>`

In questo modo sto dicendo che chiamando

<http://merc.apache.local:8080>

mi vengono serviti i files che si trovano nella directory:

`/var/www/merc.apache.local`

Attenzione! Non abbiamo ancora attivato il sito!



Attiviamo il sito.

La configurazione del sito va inserita in:
`/etc/apache2/sites-available`

ma il sito va attivato con link simbolico in:
`/etc/apache2/sites-enabled`

Per creare un link uso la sintassi:

`$ sudo ln -s source nomeLink`

Creo un link simbolico:

`$ sudo ln -s /etc/apache2/sites-available/mer.apache.conf /etc/apache2/sites-enabled/mer.apache2.conf`

Verifico la configurazione e ricarico Apache2:

`$ sudo apachectl -t`

`$ sudo systemctl reload apache2`



Per evitare di annoiarci con operazioni ripetitive, il team di sviluppo di Apache ha creato un comando che esegue per noi queste operazioni:

```
$ sudo a2ensite merc.apache.local.conf
```

```
$ sudo a2dissite merc.apache.local.conf
```



Quindi adesso sullo stesso host ci rispondono:

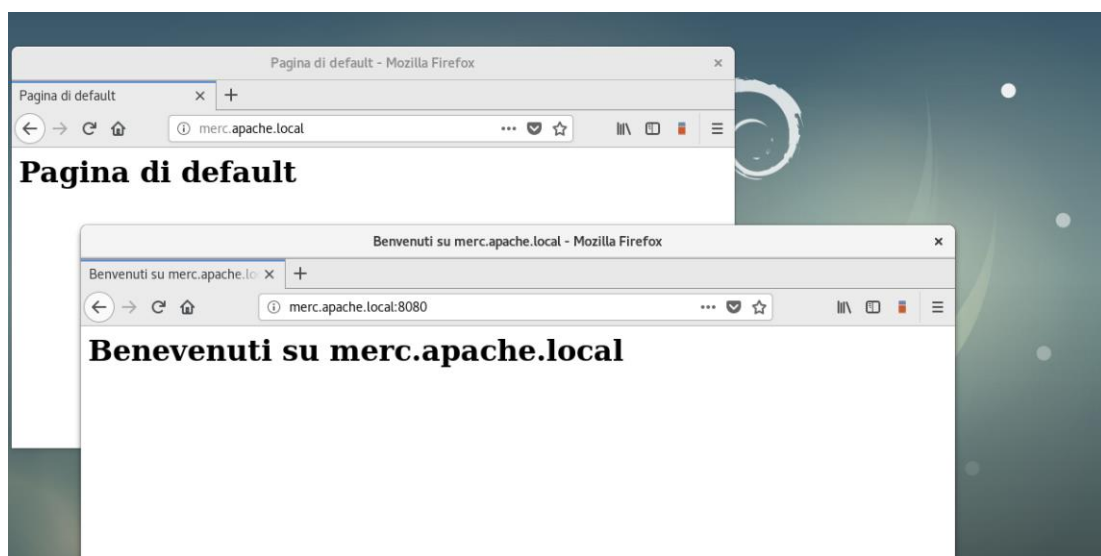
nginx sulla porta 80

<http://merc.apache.local>

e

Apache2 sulla porta 8080

<http://merc.apache.local:8080>



```
itsuser@debianITS:~$ curl -I merc.apache.local
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 21 Oct 2019 21:38:59 GMT
Content-Type: text/html
Content-Length: 144
Connection: keep-alive
Last-Modified: Mon, 21 Oct 2019 21:00:28 GMT
ETag: "90-59571f932864a"
Accept-Ranges: bytes
Vary: Accept-Encoding
```

```
itsuser@debianITS:~$ curl -I merc.apache.local:8080
HTTP/1.1 200 OK
Date: Mon, 21 Oct 2019 21:39:03 GMT
Server: Apache/2.4.25 (Debian)
Last-Modified: Mon, 21 Oct 2019 21:00:28 GMT
ETag: "90-59571f932864a"
Accept-Ranges: bytes
Content-Length: 144
Vary: Accept-Encoding
Content-Type: text/html
```



Ma come facciamo a far sì che su un host mi risponda Apache, quando c'è Nginx di mezzo?

Dobbiamo configurare Nginx come reverse proxy.

Posso ad esempio crearmi una configurazione `/etc/nginx/sites-available/apache.conf` che contenga:

```
server {  
    listen 80;  
    server_name merc.apache.local;  
  
    location / {  
        proxy_pass http://merc.apache.local:8080;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

(Ovviamente perché sia attiva va linkato in `/etc/nginx/sites-enabled/`).

Approfondimenti: <https://www.digitalocean.com/community/tutorials/how-to-configure-nginx-as-a-web-server-and-reverse-proxy-for-apache-on-one-ubuntu-18-04-server>