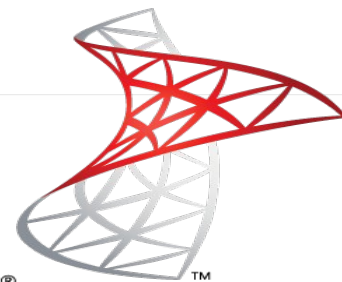




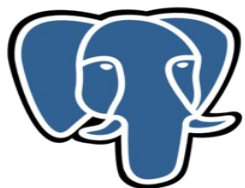
mongoDB



Microsoft®
SQL Server®

ORACLE®

DATA BASES
IDBMS



PostgreSQL

MySQL™



Esempio: mysql su ubuntu

- Per installare mysql server su Ubuntu:

```
# sudo apt install mysql-server mysql-client
```

```
# sudo mysql
```

```
mysql> UNINSTALL COMPONENT "file://component_validate_password";
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH  
mysql_native_password BY '<sceglieteunapassword>';
```

- Utilizziamo una comoda utility di Ubuntu/Debian per effettuare una configurazione iniziale di MySQL:

```
# sudo mysql_secure_installation
```

Esempio: mysql su ubuntu

- La configurazione e il tuning di **MySql** avviene editando il file `/etc/mysql/conf.d/mysql.cnf`.
- Qui potete, ad esempio specificare quali reti possono provare ad accedere a **MySql**:

```
bind-address = 127.0.0.1; # ammette solo localhost
```

- Una volta cambiati i parametri di `mysql.cnf` dovete riavviare il servizio

```
# service mysql restart
```

```
# systemctl restart mysql
```

Esempio: mysql su ubuntu

- Per la gestione e l'utilizzo è disponibile una console da riga di comando:

```
# mysql -u root -p
```

- Una volta avviata la console potete ad esempio vedere quali database siano installati:

```
# mysql> show databases;
```

- Per creare un db è sufficiente usare la clausola create database:

```
# mysql> create database <vostrogruppo>_db;
```

```
# mysql> CREATE USER 'sysadmin'@'127.0.0.1' IDENTIFIED BY  
'Un4p4ssw0rd4c4s0m4d3c3nt3!';
```

```
# mysql> GRANT ALL PRIVILEGES ON <vostrogruppo>_db.* TO  
'sysadmin'@'127.0.0.1';
```

```
# mysql> FLUSH PRIVILEGES;
```

```
# mysql> show databases;
```

The background features a perspective view of a grid of squares, with a bright light source in the upper right corner creating a lens flare effect. The text 'IL WEB' is centered in a bold, red, sans-serif font with a black outline.

IL WEB

Esempio: Installare LAMP(PP) Server 1/2

- Un **LAMP** server è un host che funge da server web costruito per ospitare applicazioni che utilizzano **php** e **mysql** oltre a **html** e **css**.
- Il vostro progetto di Basi di dati gira su un **LAMP** server.
- Su Ubuntu (per comodità a qualsiasi richiesta password mettete sempre `laslaslas`):

```
# sudo apt install lamp-server^ ; metapacchetto
# sudo apt install phpmyadmin
# sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-enabled/phpmyadmin.conf
# sudo service apache2 restart
```

- Al termine dell'installazione abbiamo a disposizione:
 - un web server Apache 2,
 - il database server MySQL,
 - PHP 7 con il modulo per Apache e la libreria gd,
 - phpMyAdmin per la gestione dei database.

Esempio: Installare LAMP(PP) Server 2/2

Puntando il browser su `http://192.168.100.xx` e su `http://192.168.100.xx/phpmyadmin` si ottiene:

The image shows two browser windows from Mozilla Firefox. The top window displays the 'Apache2 Ubuntu Default Page: It works - Mozilla Firefox' with the Ubuntu logo and the text 'It works!'. The bottom window displays the 'phpMyAdmin 4.5.4.1deb2ubuntu2' interface. The address bar shows the URL '192.168.203.130/phpmyadmin/index.php?token=280d8f5440226cd2ec1e567a1221b518'. The interface includes a sidebar with 'Recent' and 'Favorites' buttons, and a main area with a 'General settings' tab. A login box is visible in the bottom right corner of the phpMyAdmin interface.

Apache2 Ubuntu Default Page

It works!

phpMyAdmin

192.168.203.130 / localhost | phpMyAdmin 4.5.4.1deb2ubuntu2 - Mozilla Firefox

192.168.203.130/phpmyadmin/index.php?token=280d8f5440226cd2ec1e567a1221b518

Server: localhost

Databases SQL Status User accounts Export Import

General settings

Change password

Login: **phpmyadmin**
Passwd: **laslaslas**

VirtualHost - Esempio

- Si vuole che il server Web risponda alle richieste per `hpyyweb.<vostrogruppo>.localdomain`
- Anzitutto è necessario definire una entry nel server **DNS** che faccia puntare `lasweb` al server web:

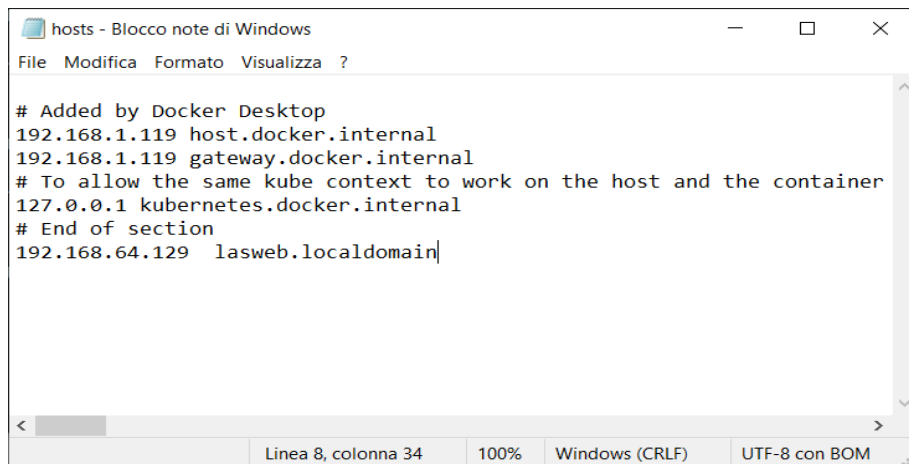
```
hpyyweb A 192.168.100.xx #in questo caso \
                                bisogna inserire anche il reverse
```

oppure

```
hpyyweb CNAME hpxx
hpyy A 192.168.100.xx
```


VirtualHost - Esempio

- Oppure, ancora più semplicemente editate il vostro file `c:\windows\system32\etc\drivers\hosts` (`o /etc/hosts`) con notepad come amministratore e inserite la riga (soluzione valida solo in lab):
- `192.168.100.xx hpyyweb.<vostrogruppo>.localdomain hpyyweb`



```
# Added by Docker Desktop
192.168.1.119 host.docker.internal
192.168.1.119 gateway.docker.internal
# To allow the same kube context to work on the host and the container
127.0.0.1 kubernetes.docker.internal
# End of section
192.168.64.129 lasweb.localdomain|
```

Linea 8, colonna 34 100% Windows (CRLF) UTF-8 con BOM

VirtualHost - Esempio

- Successivamente bisogna creare la directory dove il sito sarà ospitato e i relativi file di log:

```
# sudo mkdir /var/www/hpyyweb
```

```
# sudo mkdir /var/log/apache2/hpyyweb
```

- Ricordiamoci di dare gli accessi alla directory del sito all'utente `www-data`:

```
# sudo chown -R www-data:www-data /var/www/hpyyweb
```

- Creiamo il file di configurazione del VirtualHost:

```
# sudo [vi|nano] /etc/apache2/sites-available/hpyyweb.conf
```

VirtualHost – Esempio

- E ci copiamo dentro questa configurazione:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName hpyyweb.<vostrogruppo>.localdomain
    DocumentRoot /var/www/lasweb/
    ErrorLog /var/log/apache2/hpyyweb/hpyyweb.error.log
    LogLevel warn
    CustomLog /var/log/apache2/hpyyweb/hpyyweb.access.log combined
    ServerSignature Off
</VirtualHost>
```

- Ora non resta che attivare il sito e riavviare apache:

```
# sudo a2ensite hpyyweb.conf
# sudo systemctl restart apache2
```

VirtualHost – Esempio

- A questo punto puntando il browser verso
`http://hpyyweb.<vostrogruppo>.localdomain`

Dovremo vedere il nostro sito.... Ma non funzionerà
nulla come sempre :(

HTTPS - Esempio

- Attiviamo il modulo ssl e riavviamo apache:

```
# sudo a2enmod ssl  
# sudo systemctl restart apache2
```
- Creiamo una directory dove mettere il certificato appena generato:

```
# sudo mkdir /etc/apache2/ssl
```
- Creiamo il certificato:

```
# sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048  
-keyout /etc/apache2/ssl/apache.key -out  
/etc/apache2/ssl/apache.crt
```
- Configuriamo apache creando un nuovo sito:

```
# sudo nano /etc/apache2/sites-available/default-ssl.conf
```

HTTPS- Esempio

- E inseriamo la configurazione nel file:

```
# sudo nano /etc/apache2/sites-available/default-ssl.conf:
```

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key

        BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0

        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>
```

HTTPS - Esempio

- Abilitiamo il sito:

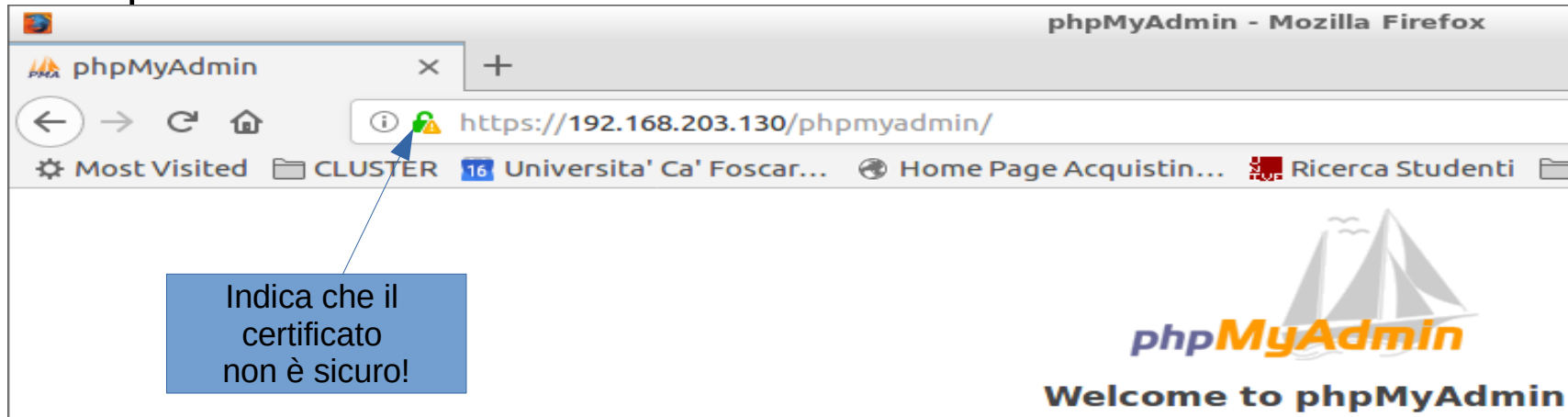
```
# sudo a2ensite default-ssl.conf
```

- Riavviamo apache:

```
# sudo systemctl restart apache2
```

- A questo punto puntando il browser verso

`https://192.168.100.xx/phpmyadmin` Dovremo vedere il nostro sito protetto con SSL....



HTTPS – Esempio per hpyyweb

- Aggiungiamo questa configurazione nel file `/etc/apache2/sites-available/hpyyweb.conf`

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerAdmin webmaster@ocalhost
    ServerName hpyyweb.<vostrogruppo>.localdomain
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key
    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    DocumentRoot /var/www/hpyyweb/
    ErrorLog /var/log/apache2/hpyyweb/hpyyweb.error.log
    LogLevel warn
    CustomLog /var/log/apache2/hpyyweb/hpyyweb.access.log combined
    ServerSignature Off
</VirtualHost>
</IfModule>
```


HTTPS – Esempio per lasweb

- Riavviamo apache:

```
# sudo systemctl restart apache2
```

- A questo punto puntando il browser verso

```
https://hpyyweb.<vostrogruppo>.localdomain
```

Dovremo vedere il nostro sito protetto con SSL....Ma non funzionerà` nulla come sempre :'(

The NGINX logo is a green hexagon with a white stylized 'N' inside. The text 'NGINX (Engine X)' is written in a bold, dark red font with a black drop shadow, centered horizontally and partially overlaid by the logo.

NGINX (Engine X)

Nginx - Installazione

- Fermate apache2 se lo avete installato e configurato come in precedenza.

```
# sudo systemctl stop apache2
```

- Per installare la chiave pubblica del repository, digitare il comando:

```
# sudo wget https://nginx.org/keys/nginx_signing.key  
# sudo apt-key add nginx_signing.key
```

L'aggiunta del repository avviene invece modificando il file `sources.list`, presente nella directory `/etc/apt/`, aggiungendo le righe (Ubuntu 22.04):

```
deb https://nginx.org/packages/mainline/ubuntu/ kinetic  
nginx  
deb-src https://nginx.org/packages/mainline/ubuntu/  
kinetic nginx
```

Nginx - Installazione

- Dopo aver modificato il file `sources.list`, aggiornare la lista di pacchetti con il comando:

```
# sudo apt update
```

- Quindi procedere con l'installazione, mediante il comando:

```
# sudo apt install nginx nginx-extras nginx-core
```

Nginx – Virtual Hosting per hpyyweb

- Creare tanti file in `/etc/nginx/sites-available/` quanti sono i siti da ospitare, usando ovviamente nomi significativi e abilitarli in `/etc/nginx/sites-enabled/` (nel caso non esistessero queste cartelle crearle manualmente)

```
# sudo nano /etc/nginx/sites-available/hpyyweb.conf
server {
    listen 80;
    root /var/www/hpyyweb;
    index index.html;
    server_name hpyyweb.<vosrtrgruppo>.localdomain;
    access_log /var/log/nginx/hpyyweb/hpyyweb.access.log;
    error_log /var/log/nginx/hpyyweb/hpyyweb.error.log;
}
```

- Attiviamo il nuovo sito :

```
# sudo ln -s /etc/nginx/sites-available/hpyyweb.conf
/etc/nginx/sites-enabled/
```

- Riavviamo nginx:

```
# sudo systemctl restart nginx
```

Nginx – Virtual Hosting per myvhost

- A questo punto puntando il browser verso `http://hpyyweb<vostrogruppo>.localdomain`
Dovremo vedere il nostro sito.... Ma non funzionerà
nulla come sempre :'(

Nginx – Configurazione https per hpyyweb

- L'attivazione di **HTTPS** per un **server block** avviene specificando il parametro “**ssl**” per la direttiva **listen**.
- I percorsi dei file certificato e chiave privata sono specificati dalle direttive **ssl_certificate** e **ssl_certificate_key**, anch'esse contenute nel blocco server.
- Esempio:

```
server {  
    listen 80;  
    listen 443 ssl;  
    root /var/www/hpyyweb;  
    index index.html;  
    server_name hpyyweb.<vostrgruppo>.localdomain;  
    access_log /var/log/nginx/hpyyweb/hpyyweb.access.log;  
    error_log /var/log/nginx/hpyyweb/hpyyweb.error.log;  
    ssl_certificate    /etc/apache2/ssl/apache.crt;  
    ssl_certificate_key /etc/apache2/ssl/apache.key;  
}
```

Nginx – Configurazione https per myvhost

- Riavviamo nginx:

```
# systemctl restart nginx
```

- A questo punto puntando il browser verso
`https://hpyyweb.<vostrogruppo>.localdomain`

Dovremo vedere il nostro sito protetto con SSL....Ma non funzionerà` nulla come sempre :'(