LAN-server

- OS: *Debian 13 Trixie*
- CPU: *Intel Atom N455@1.66GHz*
- Memory: *2 GiB*
- Disk: *128 GB SSD*
- IP: `192.168.1.200/24`
- Access Point: *D-Link DSL-2640B* @ `192.168.1.254`

> Il `/24` dopo l'IP indica che i primi 24 bit = 3 byte = tre numeri sono gli stessi per tutta la rete. Se fosse stato `/16` sarebbero stati uguali i primi due numeri dell'IP. Una rete `/24` quindi è più piccola di una rete `/16`.

Impostazione iniziale

SSH

Per installare il server (nel caso non lo avessi fatto durante l'installazione):

```
apt install openssh-server
```

Per collegarsi al server:

```
ssh mattia@192.168.1.200
```

Se l'utente è lo stesso basta:

```
ssh 192.168.1.200
```

Fonte: debian wiki.

sudo

Mi autentico come root e installo sudo:

```
su -l # per avere tutta la lista dei binari e poter eseguire adduser
apt install sudo
```

Aggiungo l'utente normale al gruppo sudo:

```
adduser mattia sudo
```

Torno utente normale (`exit` o `CTRL+D`), **logout** per applicare la modifica.

Pacchetti utili

All'inizio conviene installare alcuni pacchetti, se non già presenti:

```
sudo apt install git unzip rsync curl bat lm-sensors
```

Gestione schermo

Voglio fare in modo che il portatile non vada in sospensione con lo schermo chiuso. Devo modificare la configurazione di `logind`.

```
sudo nano /etc/systemd/logind.conf
```

Modifico `HandleLidSwitch`, decommento e imposto `ignore` invece di `suspend`. Modifico anche:

```
HandleLidSwitchExternalPower=ignore
HandleLidSwitchDocked=ignore
```

Salvo e riavvio `systemd-logind`:

```
sudo systemctl restart systemd-logind
```

LAMP Stack

Guida video: LAMP Stack.

Linux

Partendo da una nuova installazione di Debian **con SSH**, aggiorniamo i repository e se necessario aggiorniamo i pacchetti:

```
sudo apt update
sudo apt upgrade
```

Le impostazioni di **connessione** della macchina sono contenute nel file `/etc/network/interfaces`.

Apache

Installo `apache2` e le sue dipendenze:

```
sudo apt install apache2
```

Controllo se Apache sta funzionando:

```
sudo systemctl status apache2.service
```

`enabled` mi informa che Apache si attiva al boot. `q` per uscire.

Per controllare che funzioni, posso anche visitare con un browser l'indirizzo IP della macchina (`ip a` per avere tutte le info sulla connessione). Dovrei vedere la pagina di benvenuto di Apache.

Vedi anche come gestire l'upload di grandi files.

Firewall

Se Apache non funziona, potrebbe essere necessario aprire le porte del firewall `ufw` (**non preinstallato** in Debian 13):

```
sudo ufw status
```

Per aprire le porte per HTTP (80) e HTTPS (443) devo aggiungere delle regole al firewall.

Per vedere i servizi possibili:

```
sudo ufw app list
```

E poi:

```
sudo ufw allow Apache # se sono due parole uso le "virgolette così"
```

Ricontrollo lo status per confermare che Apache può attraversare il firewall.

MariaDB

MariaDB è un fork completamente open source di MySQL. Vedi qui una guida video.

Se non è installato (`mysql --version`) installo il server:

```
sudo apt install mariadb-server
```

Per avviare il server:

```
sudo systemctl start mysql.service
```

Controllo dello stato del server:

```
sudo systemctl status mysql.service
```

Per eseguire lo **script post installazione**:

```
sudo mysql_secure_installation
```

> Il file di configurazione di Apache non si chiama `httpd.conf` in Debian, come è invece ad esempio per XAMPP. Si trova in `/etc/apache2/apache2.conf`. Leggere il file con attenzione.

Digitando `mysql` apro il prompt SQL di MariaDB.

PHP

Guida qui.

Installo:

```
sudo apt install php php-mysql
```

Assicurarsi che venga installato anche `libapache2-mod-php8.4` per comunicare con il server web.

Abilito, se già non lo è, il **modulo Apache** per processare codice PHP:

```
sudo a2enmod php8.4
```

Riavvio il server Apache:

```
sudo systemctl restart apache2.service
```

phpMyAdmin

Guida qui.

Mi posiziono nella cartella che preferisco e scarico phpMyAdmin:

```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-
languages.tar.gz
```

Entro nel *docroot* di Apache e creo la cartella per PMA:

```
sudo mkdir /var/www/html/phpmyadmin
```

Estraggo l'archivio nella cartella creata:

```
sudo tar xvf phpMyAdmin-latest-all-languages.tar.gz --strip-components=1 -C
/var/www/html/phpmyadmin
```

Creo il file di configurazione a partire da quello di esempio:

```
sudo cp /var/www/html/phpmyadmin/config.sample.inc.php
/var/www/html/phpmyadmin/config.inc.php
```

Modifico il file:

```
sudo nano /var/www/html/phpmyadmin/config.inc.php
```

Cerco la stringa:

```
$cfg[‘blowfish_secret’] = ''
```

e inserisco una password complessa.

Devo fare in modo che la cartella `phpmyadmin` sia di proprietà dell'utente `www-data` (che è l'owner dei servizi httpd) e che il file modificato sia leggibile dall'utente proprietario.

```
sudo chown -R www-data:www-data /var/www/html/phpmyadmin
sudo chmod 660 /var/www/html/phpmyadmin/config.inc.php
```

> Leggi qui e qui.

> **Nota:** a installazione funzionante, i permessi sulla cartella `/var/www/html/phpmyadmin/` sono:
>
> ```
> drwxrwxr-x 13 www-data www-data   4096 13 ott 00.22 phpmyadmin
> ```
>
> All'interno:
>
> ```
> -rw-rw-r--  1 www-data www-data   2587  7 ott 22.40 CONTRIBUTING.md
> drwxrwxr-x  3 www-data www-data   4096  7 ott 22.40 doc
> ```

Riavvio Apache:

```
sudo systemctl restart apache2.service
```

Controllo se ci sono errori in Apache e apro la pagina:

```
http://[IP-del-server]/phpmyadmin
```

Se mi dice che manca l'estensione `php-mysqli` risolvo installando il pacchetto `php-mysql`.

> Dei **temi** sono disponibili qui. `unzip` l'archivio e lo sposto nella cartella `themes` di PMA.

Altri strumenti

pihole

Installazione di base:

```
curl -sSL https://install.pi-hole.net bash
```

Per installare saltando il controllo di compatibilità del sistema:

```
curl -sSL https://install.pi-hole.net | sudo PIHOLE_SKIP_OS_CHECK=true bash
```

Problema.

Dopo aver installato pihole e Apache2, la pagina di login di pihole viene mostrata come index di file perché entrambi usano la porta 80. Bisogna lavorare sulle porte di pihole.

Modifico il file `/etc/pihole/pihole.toml` alla sezione `webserver.port`:

```
port = "81o,444os,[::]:81o,[::]:444os"
```

Alzo tutte le porte di 1.

> Utile una sessione di debug: `pihole -d`.

L'interfaccia web è ora disponibile a:

```
http://[IP-del-server]:81/admin/login
```

Se non ho segnato la pass di primo accesso:

```
sudo pihole setpassword
```

> Da *Settings / Local DNS Records* posso impostare il rerouting da un indirizzo a mia scelta ad un indirizzo IP o altro.

L'interfaccia web di pihole è in `/var/www/html/admin/`.

Cockpit

Semplicemente:

```
sudo apt install cockpit
```

> Tra le dipendenze c'è `network-manager`.

L'interfaccia web è disponibile su:

```
http://[IP-del-server]:9090
```

Nome utente e password sono quell'utente normale sul server.

Nel dettaglio qui per la spiegazione sui backports (letteralmente porting all'indietro delle nuove versioni dei pacchetti, contenute in Debian testing).

Server FTP

Usare `vsftpd`: video.

Installa i pacchetti:

```
sudo apt install vsftpd ftp
```

Controlla se `vsftpd` è in esecuzione:

```
systemctl status vsftpd
```

Creazione di un utente FTP:

```
sudo useradd -m [username]
sudo passwd [username]
```

Io non lo faccio perché uso il mio utente `mattia`. Controllo del funzionamento da locale:

```
ftp localhost
```

Config file di default:

```
/etc/vsftpd.conf
```

Faccio un backup:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.backup
```

Modifico il file originale, se ne ho bisogno:

```
sudo nano /etc/vsftpd.conf
```

Una modifica da fare:

```
write_enable=YES
```

Riavviare il server ftp:

```
sudo systemctl restart vsftpd
```

Per accesso, porta vuota oppure 20, 21.

Per impostare correttamente i **permessi**:

```
cd /var/www/
sudo chown -R www-data: html/
sudo find html -type f -exec chmod 664 {} + -o -type d -exec chmod 775 {} +
```

Assegna i permessi 664 ai file e i permessi 775 alle cartelle in `html`.

> Fonti in merito alla questione dei permessi:
>
> - https://www.andreaminini.com/linux/permessi-numerici-su-linux
> - https://www.andreaminini.com/linux/come-cambiare-i-permessi-su-linux

> - https://supporthost.com/it/comando-chown-linux/

Altre fonti su `vsftpd`:

- https://wiki.debian.org/vsftpd#User_access_control
- https://security.appspot.com/vsftpd.html
- http://vsftpd.beasts.org/vsftpd_conf.html
- https://askubuntu.com/questions/354178/what-is-ftp-username-and-password-for-vsftpd
- https://www.reddit.com/r/linuxquestions/comments/phkwtf/granting_access_for_ftp_user_to_upload_files_to/

Backup

https://github.com/sukria/Backup-Manager

Copiare file sul server

In alternativa a FTP posso usare uno degli strumenti seguenti.

scp

Posso usare `scp`:

```
scp -r /home/mattia/Desktop/welcome-page mattia@192.168.1.200:/home/mattia
```

e poi, sul server:

```
sudo mv /home/mattia/welcome-page /var/www/html
```

rinomina la cartella:

```
cd /var/www/html
sudo mv welcome-page/ home/
```

rsync

Uno strumento più potente è `rsync`, tra i pacchetti raccomandati.

Video:

- LearnLinuxTV
- bread
- Veronica

sftp

La home dell'utente ftp è accessibile anche da un file manager grafico tramite `sftp`:

```
sftp://192.168.1.200
```

Comandi interessanti

Ultimi pacchetti installati

Per vedere gli ultimi pacchetti installati:

```
ls -tl /var/lib/dpkg/info/*.list
```

Per vedere gli ultimi 10:

```
ls -tl /var/lib/dpkg/info/*.list | head -n 10
```

Fonte.

Temperatura del server

Tramite il pacchetto `lm-sensors`:

```
sensors
```

Programmi utili

Editor di testo: nano

Contenuto di `~/.nanorc`:

```
set linenumbers
set mouse
set softwrap
```

Posso impostare lo stesso anche per l'utente root creando il file in `/root/.nanorc`.

Per avere le scorciatoie moderne, in `~/.profile`:

```
    export EDITOR="nano --modernbindings"
```

Gestione file: nnn

Scorciatoie:

- `e` per modificare un file
- `spazio` per selezionare un file
- `p` per copiare *coPy* i file selezionati
- `v` per spostare *moVe* i file selezionati
- `/` per filtrare/type to nav
- `x` per eliminare
- `w` per spostare/copiare cambiando il nome del file
- `d` toggle dettagli
- `.` toggle file nascosti

Opzioni interessanti:

- `n` start in type to nav mode
- `o` open files only on Enter key
- `d` detail mode
- `U` show user and group names in status bar

Per attivarle, in `~/.profile`:

```
export NNN_OPTS="nodU"
```

pandoc

Permette di convertire da md a pdf.

```
sudo apt install pandoc
```

Comando di esempio per convertire tutti gli md in una cartella in corrispondenti html:

```
for i in *.md ; do echo "$i.md --> $i.html " && pandoc -s $i -o $i.html ; done
```

Migliorabile, ad esempio togliendo l'estensione .md nel salvare il file html.

Fonte: itsfoss.com.

openEMR

Guida all'installazione qui.

Dipendenze

Installazione delle dipendenze (fonte):

```
sudo apt-get install apache2 mariadb-server libapache2-mod-php libtiff-tools
php php-mysql php-cli php-gd php-xsl php-curl php-soap php-json php-gettext
imagemagick php-mbstring php-zip php-ldap
```

Molte dovrebbero essere già installate. Non trova candidati per `php-gettext`: provo a installare `php-gettext-languages`, non so se è giusto ma poi l'installazione parte comunque (in una pagina non correlata viene citato insieme alla localizzazione di un software, potrebbe essere corretto).

Riavvia i server:

```
sudo service apache2 restart
sudo service mariadb restart
```

Download e spostamento file

Sposta il file sul server oppure scaricalo (link diretto estratto da sourceforge):

```
wget
https://downloads.sourceforge.net/project/openemr/OpenEMR%20Current/7.0.0.2/ope
nemr-7.0.0.tar.gz
```

Estrai l'archivio:

```
tar -pxvzf openemr-7.0.0.tar.gz
```

Sposta tutto nella *docroot* del webserver:

```
mv openemr-7.0.0 /var/www/html/openemr
```

La procedura di installazione via web è ora disponibile su:

```
[IP-del-server]/openemr
```

> Nota sui permessi: all'inizio della procedura guidata viene controllato che `/var/www/html/openemr/sites/default/documents` e le sue sottocartelle abbiano i permessi corretti. Prima di iniziare l'installazione, i permessi sono:
>
> ```
> drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 certificates
> drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 couchdb
> ```

```
drwxrwxrwx 3 mattia mattia 4096  6 dic  2022 custom_menus
drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 edi
drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 era
drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 letter_templates
drwxrwxrwx 3 mattia mattia 4096  6 dic  2022 logs_and_misc
drwxrwxrwx 3 mattia mattia 4096  6 dic  2022 mpdf
drwxrwxrwx 3 mattia mattia 4096  6 dic  2022 onsite_portal_documents
drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 procedure_results
drwxrwxrwx 4 mattia mattia 4096  6 dic  2022 smarty
drwxrwxrwx 2 mattia mattia 4096  6 dic  2022 temp
```

> La prima `d` indica che si tratta di directory. Le cartelle sono proprietà di `mattia`, con gruppo `mattia` e la possibilità per tutti di leggere, scrivere, ed eseguire (nel caso delle cartelle, il permesso di esecuzione corrisponde al permesso di apertura). In termini numerici, questi permessi corrispondono a `777`.

Upload database precedenti

Prima di iniziare l'installazione, provo a caricare il vecchio database (+backup) sul server. L'ho salvato come unico file sql, molto grande.

Se necessario, bisogna aumentare il **limite massimo di upload** e post di Apache, che ha effetto sul limite mostrato da phpMyAdmin. Modifico `php.ini`:

```
sudo nano /etc/php/8.4/apache2/php.ini
```

Modifico le impostazioni a riga `419`, `699` e `851`:

```
max_input_time = 300        # in secondi; def 60
post_max_size = 200M        # maggiore di upload_max_size; def 8MB
upload_max_filesize = 100M  # funziona per la dimensione dei miei db; def 2MB
```

Riavvio il server apache:

```
sudo systemctl restart apache2
```

Uso ora la pagina di importazione di phpMyAdmin per caricare il file di backup. Ci vuole un po'!

Installazione

La procedura guidata è raggiungibile da 192.168.1.200/openemr.

Dati per l'installazione:

1. Se ho già un database importato, allora devo prima creare l'utente MySQL da usare per gestire il db. Lo creo da phpMyAdmin con i dati qui sotto. Devo ovviamente anche fornire **tutti** i privilegi sul database `openemr` all'utente appena creato.

> Per poter riuscire a fare login da un altro computer, devo sul server, da phpMyAdmin, modificare il nome host per `root` permettendo il login da ogni posizione.

> Avrei potuto prima creare l'utente e contestualmente il database con lo stesso nome e solo poi fare un dump dei dati **evitando l'istruzione** `CREATE DATABASE`.

nome	commento	valore
server host	IP del server MySQL	`192.168.1.200`
server port	la porta usata per comunicare in rete	`3306`
database name	il nome del db in phpMyAdmin	`openemr`
login name	username per MySQL	`openemr`
password	pass dell'utente sopra	`administrator`
root account	utente root di MySQL, crea l'utente sopra e il db	`root`
root password	vedi tabella utenti in phpMyAdmin	solitalunga
user hostname	IP del server PHP	`192.168.1.200`
initial login name	nome amministratore	`administrator`
initial user password	password amministratore	`administrator`

> Se l'installazione non inizia perché fallisce il login come utente `openemr`, andare nella tabella utenti di phpMyAdmin e modificare il nome host dell'utente `openemr` per funzionare da **qualsiasi host** (per poter gestire il db da remoto).

Questo passaggio dell'installazione richiede un po' di tempo, attendere senza ricaricare la pagina.

2. Al passo 4 dell'installazione viene richiesto di modificare un file di configurazione di Apache, `php.ini`, pertanto:

   sudo nano /etc/php/8.4/apache2/php.ini

e modifico i valori richiesti (leggi con attenzione!).

3. Al passo 5 dell'installazione viene richiesta una modifica al file di configurazione di apache.

   sudo nano /etc/apache2/apache2.conf

Aggiungere queste righe alla fine del file:

```
```php
<Directory "/var/www/html/openemr">
    AllowOverride FileInfo
    Require all granted
```
```

```
</Directory>
<Directory "/var/www/html/openemr/sites">
    AllowOverride None
</Directory>
<Directory "/var/www/html/openemr/sites/*/documents">
    Require all denied
</Directory>
```

4. Procedere con l'installazione. Al termine riavviare il server Apache.

   sudo systemctl restart apache2

> Nel caso di errori, i log di apache sono in `/var/log/apache2/`.

5. Il gestionale è disponibile all'indirizzo:

   ```
   https://localhost/openemr
   ```

   oppure:

   ```
   http://[IP-del-server]/openemr
   ```

6. **Problema:** dopo l'installazione, la pagina di login restituisce un generico errore 500. Controllo il log di apache in `/var/log/apache2/error.log` e trovo:

   PHP Parse error: syntax error, unexpected token ":" in /var/www/html/openemr/vendor/twig/twig/src/Environment.php(358) : eval()'d code on line 48

Sembra esserci un problema con `twig`. scarico la versione più aggiornata da qui e la sposto sul server:

```
scp -r /home/mattia/Downloads/twig/ mattia@192.168.1.200:/home/mattia
```

I file originali erano in `/var/www/html/openemr/vendor/twig/twig`, rispettare la struttura. (Fonte) [https://community.open-emr.org/t/syntax-error-on-first-start-in-environment-php/24048/6].

7. **Altro problema:** i moduli all'interno della pagina non vengono caricati. Provo la soluzione qui, lavorando sul file:

   sudo nano /etc/apache2/conf-available/ssl-params.conf

e commentando la riga:

```
#Header always set X-Frame-Options DENY
```

Riavvio il server apache:

```
sudo systemctl restart apache2
```

Impostazione

1. Creazione degli account di base (medici e front desk), prima i medici e poi il desk:

user	nome	cognome	id
dott-rossi	Antonella	Rossi	5
dott-enoki	Nobuo	Enoki	6
dott-mellino	Francesco	Mellino	7
desk-spada	Alice	Spada	8
desk-salas	Carla	Salas	9
desk-pietra	Filippo	Pietra	10

Gli account utenti si trovano nella tabella `users_secure`.

2. Per importare i dati degli utenti da altri database, importo da phpMyAdmin nella tabella `patient_data` il file `patient-dump.sql`. Il campo che contiene l'ID del medico assegnato è `providerID` e oscilla tra 5 e 7.

3. Per configurare gli orari di presenza in studio bisogno creare un nuovo evento di calendario per gli operatori e indicare come tipo "In ufficio", "Fuori ufficio" o "Pranzo".

Permessi

I permessi su `/var/html/www/openemr/` sono:

```
drwxr-xr-x 29 www-data www-data   4096  6 dic  2022 openemr
```

All'interno, i permessi per file e cartelle sono come in questo estratto:

```
-rw-r--r--  1 www-data www-data  28612  6 dic  2022 API_README.md
drwxr-xr-x  2 www-data www-data   4096  6 dic  2022 apis
```

Backup

Una volta impostato, scarico un backup completo, `emr_backup.tar`. Contiene:

- un file SQL per il database
- la cartella `/var/www/html/openemr/`

Welcome page

La pagina si trova in `var/html/www/index.html` e tutti i materiali correlati si trovano nella stessa cartella.

I permessi sono:

```
drwxrwxr-x  2 www-data www-data   4096 19 ott 20.36 risorse
-rw-rw-r--  1 www-data www-data  11036 19 ott 20.36 risorse.html
```