



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea in **INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE**

IMPLEMENTAZIONE SU PIATTAFORMA MIKROTIK DI UN SERVER MYSQL

MYSQL SERVER IMPLEMENTATION ON A MIKROTIK DEVICE

Relatore: Prof.

Ennio Gambi

Tesi di Laurea di:

Gabriele De Bartolomeo

Correlatore: Prof.

Dott. **Adelmo De Santis**

A.A. 2019 / 2020

Indice

INTRODUZIONE	2
Passaggio da RouterOS a OpenWRT	3
Verifica dell'integrità hardware e software del dispositivo.....	3
Salvataggio della licenza RouterOS	5
Installazione di OpenWRT	8
Prima parte: test "live" della build del sistema operativo indicata per l'hardware specifico tramite caricamento in RAM del file <i>initramfs.elf</i>	8
Seconda parte: flash in memoria di massa dell'immagine di OpenWRT come SO permanente ..	14
INSTALLAZIONE E AVVIO DI WEBSERVER (uHTTPd) E INTERPRETE PHP (PHP7)....	17
Collegamento SSH con il router, connessione alla rete WAN e aggiornamento dell'elenco dei pacchetti d'installazione	17
Webserver: uHTTPd.....	23
Installazione e configurazione PHP per uHTTPd	25
Configurazione di MySQLi in PHP	27
MySQL/MariaDB	29
Installazione del server MySQL/MariaDB.....	30
Guida all'espansione della memoria interna e di RAM del dispositivo sfruttando la capacità di una periferica di archiviazione USB	32
Partizionamento e formattazione del supporto di archiviazione	32
Riconoscimento della periferica USB connessa al router	37
Mounting e abilitazione del file-system della periferica di archiviazione USB.....	38
Creazione cartelle dati MariaDB e modifica di <i>my.cnf</i>	43
Server MariaDB: inizializzazione e abilitazione in fase di boot di OpenWRT.....	45
Client MariaDB.....	45
Accesso al DBMS	49
DIFFICOLTÀ RISCONTRATE	50
PROCEDURA DI RIPRISTINO ROUTER IN SEGUITO AD AGGIORNAMENTO DI FIRMWARE OpenWRT NON ANDATO A BUON FINE	51
CONCLUSIONI	57
APPROFONDIMENTO: Parametri essenziali di configurazione in <i>my.cnf</i>	58
Moduli di funzionamento allocati in memoria centrale da MariaDB.....	61
Performance_Schema.....	62
Inno_DB	63
RINGRAZIAMENTI	65
BIBLIOGRAFIA	66

INTRODUZIONE

Il presente lavoro di tesi consiste nella realizzazione, con hardware Mikrotik, si potesse implementare un server web con annesso il servizio di gestione database MySQL/MariaDB, dopo il flashing del sistema operativo open source per apparati di rete “OpenWRT”^[1].

La prima parte del lavoro è consistita nel visionare la lista ufficiale dei dispositivi per i quali è stata realizzata una build di OpenWRT^[2] e, tra questi, sceglierne uno con specifiche tecniche minime tali da poter garantire efficacemente la funzionalità richiesta.

In merito alla scelta, si è optato per il router **Mikrotik RB2011UiAS-2HnD-IN**^[3].

La presente trattazione vuole dunque essere proposta come una guida all’implementazione di un sistema client-server MySQL/MariaDB all’interno del dispositivo citato, con la finalità di renderlo utilizzabile per un qualsiasi ambito applicativo che preveda la gestione di basi di dati.

(Nota: Inseriti all’interno del corpo del presente lavoro vi sono degli indici numerici di reindirizzamento alle fonti originali, raggiungibili tramite la corrispondente voce nella sezione “[Bibliografia](#)”)

Passaggio da RouterOS a OpenWRT

Verifica dell'integrità hardware e software del dispositivo

Come primo passo si è verificato che il dispositivo non presentasse difetti o malfunzionamenti hardware o software. A tal fine si è proceduto come segue:

- Si è creato un collegamento LAN point-to-point tra il router e un PC^[4], con IP di rete 192.168.88.0/24, quella di default per tutte le porte di questo specifico router. (Immagine 1.1 e Immagine 1.2)
- Si è poi digitato da un browser l'indirizzo predefinito delle interfacce ethernet del router, 192.168.88.1 per accedere alla Web-Console del sistema operativo originale del router, denominata "RouterOS"^[5]. (Immagine 1.3)



Immagine 1.1 – La rete point-to-point formata da router e PC



Immagine 1.2 – Il router segnala gli indirizzi impostati sulle proprie interfacce

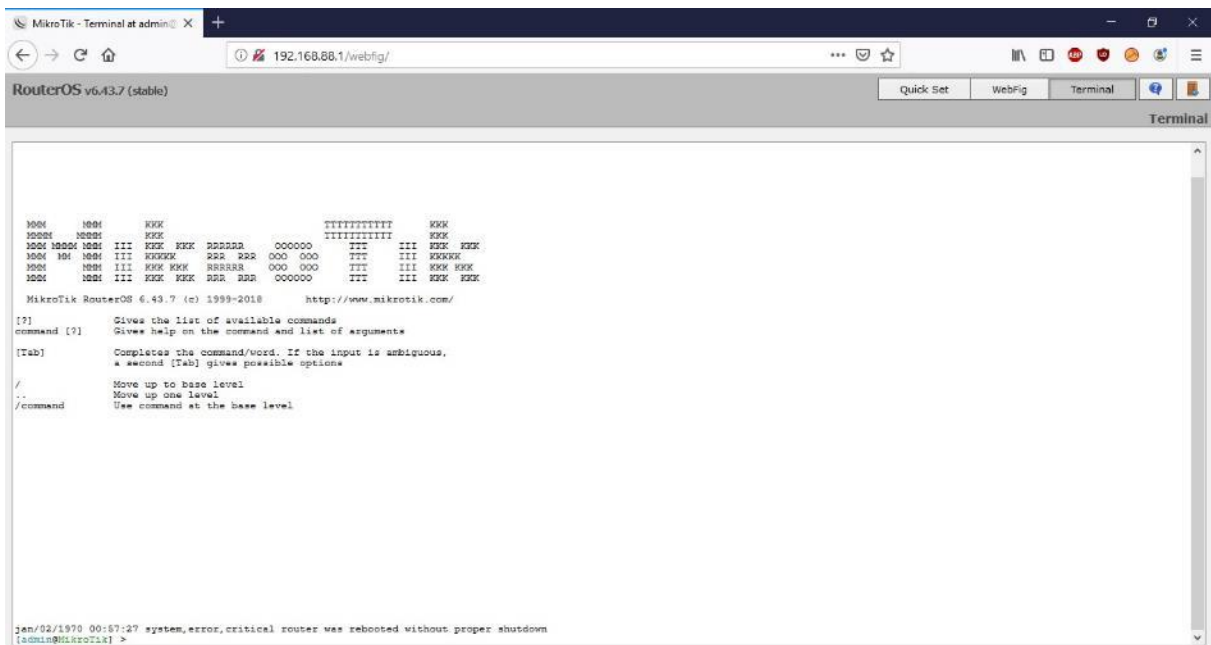


Immagine 1.3 – Accesso alla Web Console di RouterOS da browser

Salvataggio della licenza RouterOS

Per poter in un secondo momento tornare a installare il SO proprietario di default Mikrotik (“RouterOS”) da OpenWRT è necessario esportare il file contenente la chiave di licenza dello stesso, che ha estensione .key .

A questo scopo, Mikrotik mette a disposizione sul proprio portale di supporto online il software WinBox^[6] . Clickando sul link come da Immagine 2.1, si scaricherà il relativo file eseguibile .exe .

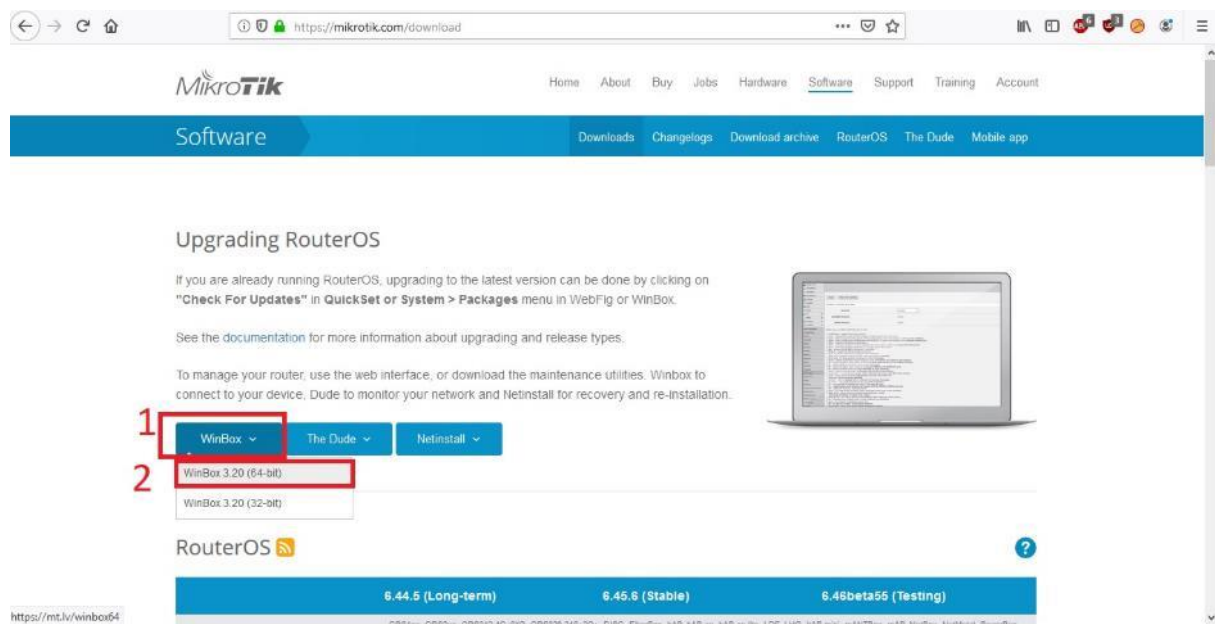


Immagine 2.1 – Download del software WinBox

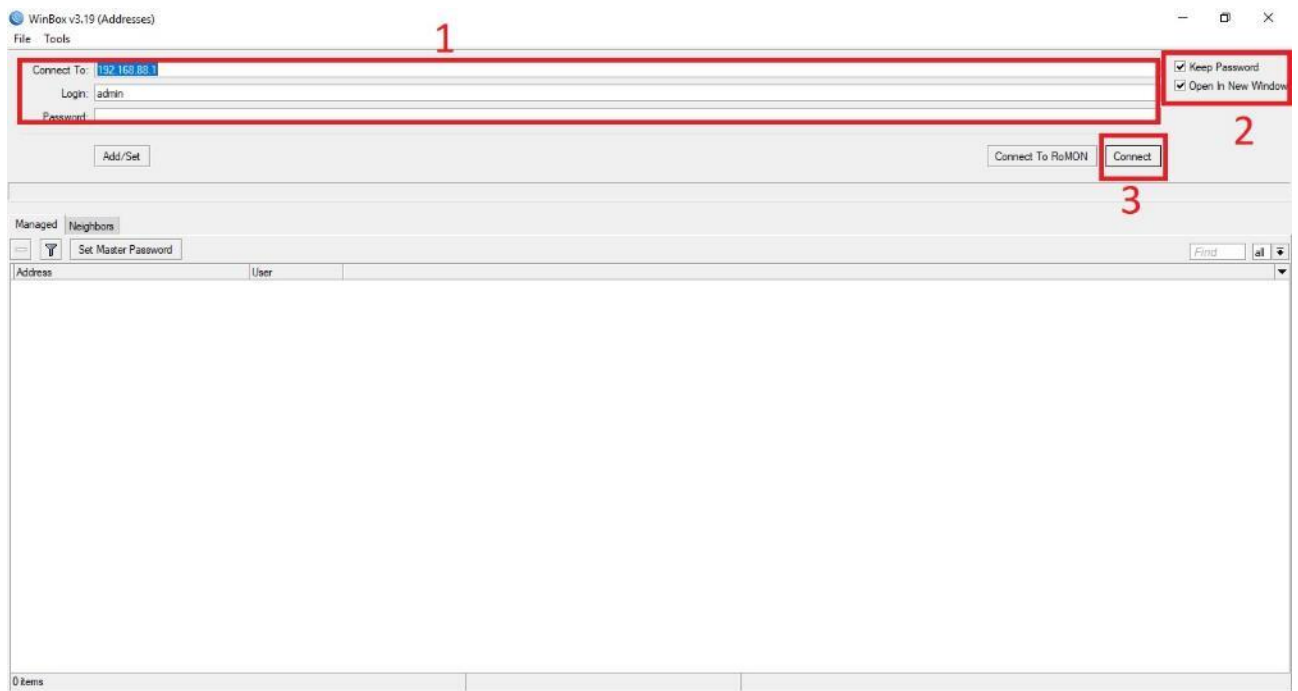


Immagine 2.2 – Impostazione preliminare di WinBox per l’accesso alle impostazioni del router tramite pannello di controllo a interfaccia grafica

Una volta aperto WinBox e inserito le impostazioni come si può vedere nell’Immagine 2.2 (IP assegnato alla porta ETH1 del router, username *admin* e nessuna password), si aprirà una nuova finestra contenente la GUI per il controllo e la configurazione del router, dalla quale si potrà esportare il file .key in una locazione arbitrariamente scelta dall’utente all’interno del File-system del PC (Immagine 2.3 e 2.4).



Immagine 2.3 – WinBox e le opzioni da cliccare per l’esportazione del file di licenza di RouterOS

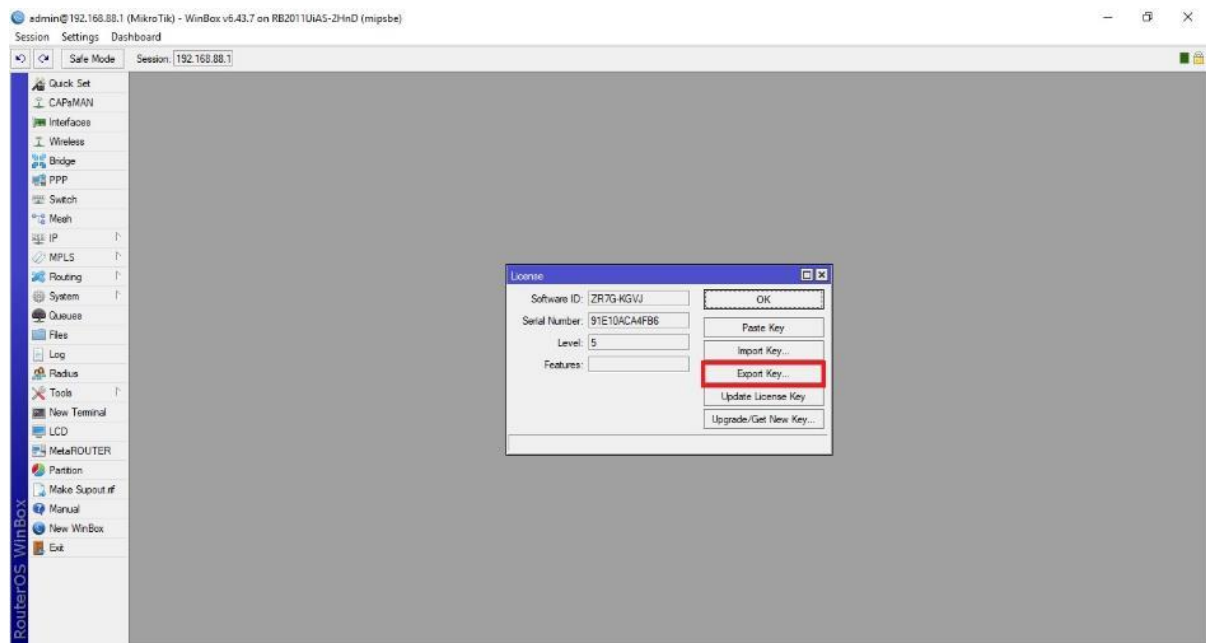


Immagine 2.4 – Finestra di WinBox con le opzioni relative al file di licenza di RouterOS

Installazione di OpenWRT


Prima parte: test “live” della build del sistema operativo indicata per l’hardware specifico tramite caricamento in RAM del file *initramfs.elf*

OpenWRT permette di testare l’immagine del sistema operativo tramite apposita procedura e relativi file, caricandola nella RAM del dispositivo e senza che avvenga alcuna installazione sulla memoria di massa, lanciando così quella che viene definita un’istanza “live”, con la possibilità di ritornare al SO correntemente installato nella memoria NAND-FLASH con un semplice reboot.

Si potrà così procedere alla verifica dell’integrità dell’immagine scaricata e dell’effettiva compatibilità del SO con l’ hardware utilizzato prima di passare al flashing (scrittura su memoria di massa) definitivo.

Per la procedura di verifica della compatibilità bisogna innanzitutto scaricare sul PC un software che faccia da server virtuale con supporto ai protocolli DHCP^[7], BOOTP^[8] e TFTP^[9]. Nella presente trattazione si illustra la procedura seguita tramite il protocollo TFTP. Il software raccomandato nella guida d’installazione di OpenWRT^[10] è “Tiny PXE”^[11].

Nel file “config.ini” contenuto nella cartella che si ottiene dall’estrazione dell’archivio di Tiny PXE, deve essere presente il contenuto evidenziato nell’Immagine 3.1 per fare in modo che si possa utilizzare il protocollo DHCP e BOOTP utilizzando i parametri di default del router.



```
config.ini - Blocco note di Windows
File Modifica Formato Visualizza ?
[arch]
;will over rule the bootp filename or opt67 if the client arch matches one of the below
;00006=bootia32.efi
;00007=bootx64.efi
;00009=bootx64.efi
[dhcp]
;below is applicable only if proxydhcp=0
+rfc951=1
;needed to tell TFTPd where is the root folder
root=files
;bootp filename as in http://tools.ietf.org/html/rfc951
;filename=ipxe-undionly.kpxe
filename=ipxe.pxe
;alternative bootp filename if request comes from ipxe or gpxe
altfilename=menu.ipxe
;start HTTPd
httpd=1
binl=0
start=0
dnssd=0
;if you have a dhcp server on your lan, set proxydhcp=1
proxydhcp=0
;default=1
bind=1
```

Immagine 3.1 – parametro DHCP nel file config.ini di Tiny PXE

Nota: TinyPXE ha il compito molteplice di:

- rendere il PC un server DHCP che offre un pool di indirizzi host pari a 192.168.1.1-192.168.1.254;
- utilizzare il protocollo TFTP per il trasferimento di file (in questo caso dapprima l'immagine temporanea iniramfs di OpenWRT e poi quella che verrà installata effettivamente nella memoria flash in maniera permanente dopo la verifica della funzionalità del SO come accennato poc'anzi).

Così facendo, di fatto si implementa un boot del SO da un dispositivo remoto collegato in rete con il router (appartenente, in questo caso, alla stessa rete del router), ovvero si mette in atto la tecnica del "netbooting", con utilizzo del protocollo TFTP per il trasferimento dell'immagine temporanea del SO.

A questo punto è consigliato^[12] scollegare il PC da qualsiasi rete (che sia LAN e/o WLAN) e collegare l'interfaccia ethernet1 (tipicamente sia porta WAN sia CONSOLE) del router con quella del PC, con indirizzo IP di quest'ultima settato a 192.168.1.10/24.

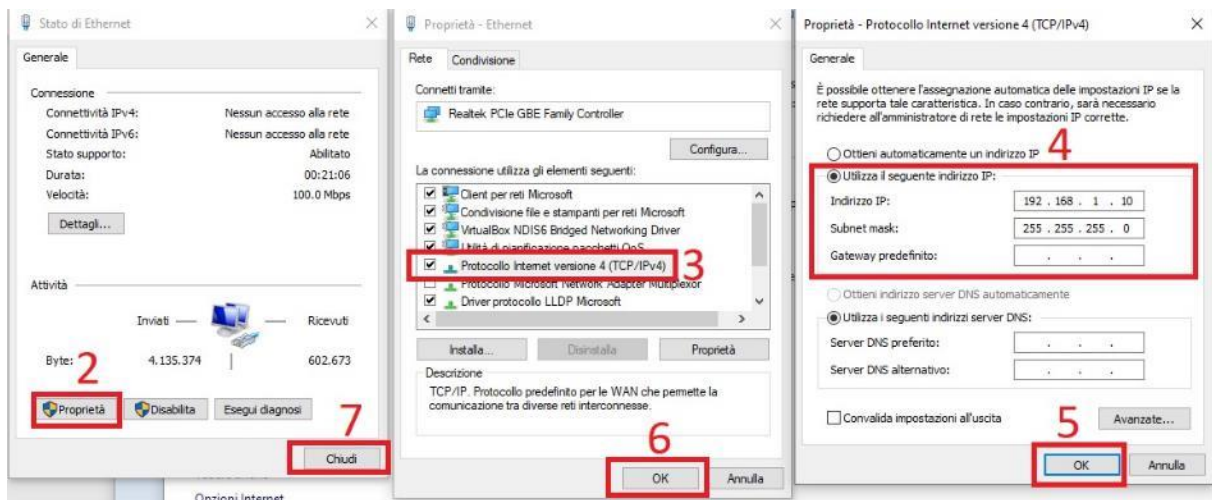
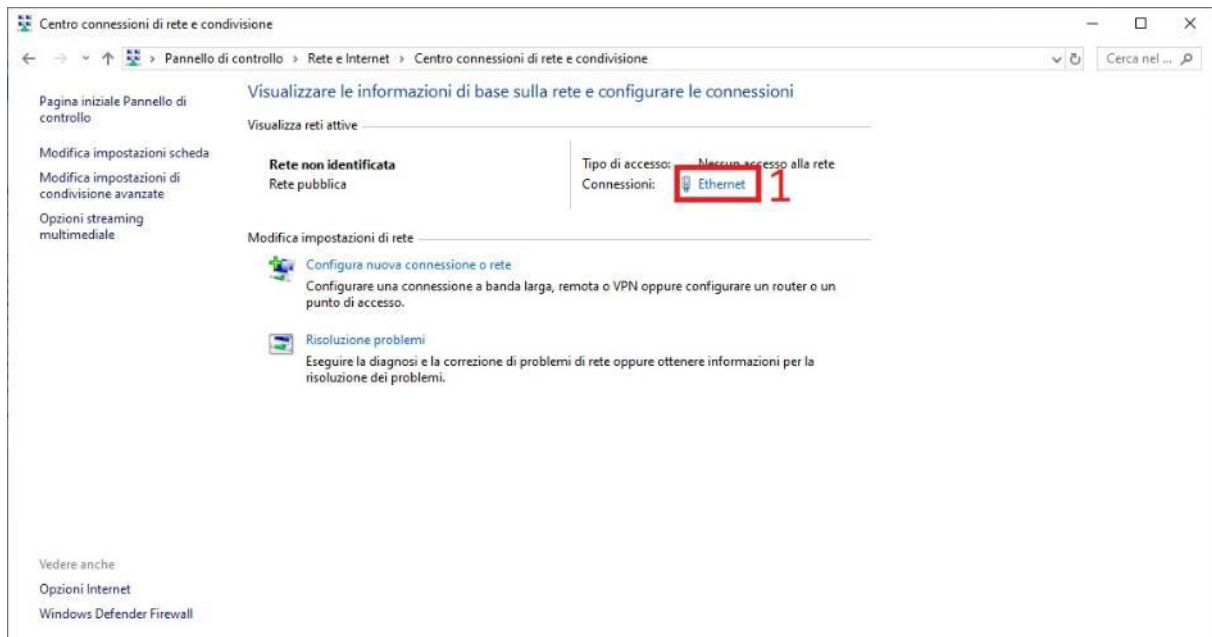


Immagine 3.2 – Impostare un indirizzo IP statico su un'interfaccia di rete in Windows 10

Fatto questo, avviare TinyPXE il quale, avendo scollegato/disattivato tutte le connessioni su PC, eccetto quella con il router, apparirà come nell'Immagine 3.3.

Da notare è l'indirizzo indicato nella prima riga del form di configurazione al di sotto della finestra di log, che è pari a quello impostato per l'interfaccia ethernet del PC secondo quanto indicato in precedenza).

Una volta verificato che tutti i parametri di configurazione corrispondano a quelli illustrati nell'Immagine 3.3, si potrà procedere al caricamento del file *initramfs.elf*.

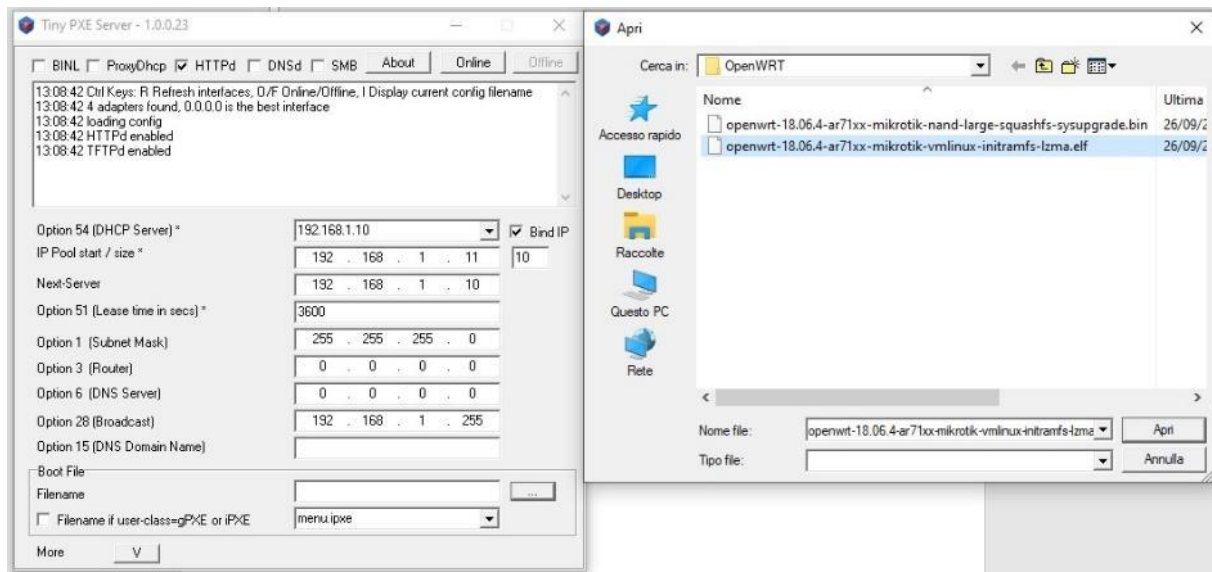


Immagine 3.3 – Apertura di TinyPXE e scelta del file *initramfs.elf* di OpenWRT per il netbooting dell'immagine temporanea da caricare in RAM del nuovo sistema Operativo.

Sarà quindi possibile passare alla fase di netbooting per mezzo di un'apposita procedura tramite combinazione di tasti fisici direttamente sul router.

La procedura consiste, mantenendo la connessione cablata tra router e PC, in:

- staccare l'alimentazione del router;
- tenere premuto il pulsante Reset situato sul lato posteriore del router e ridare alimentazione al router, senza mai rilasciare il pulsante Reset;
- continuare a premere il pulsante Reset finché non si vede, nel log di TinyPXE, un insieme di messaggi che indicano lo scambio di messaggi DHCP tra PC e router, come appare nell'Immagine 3.4 .

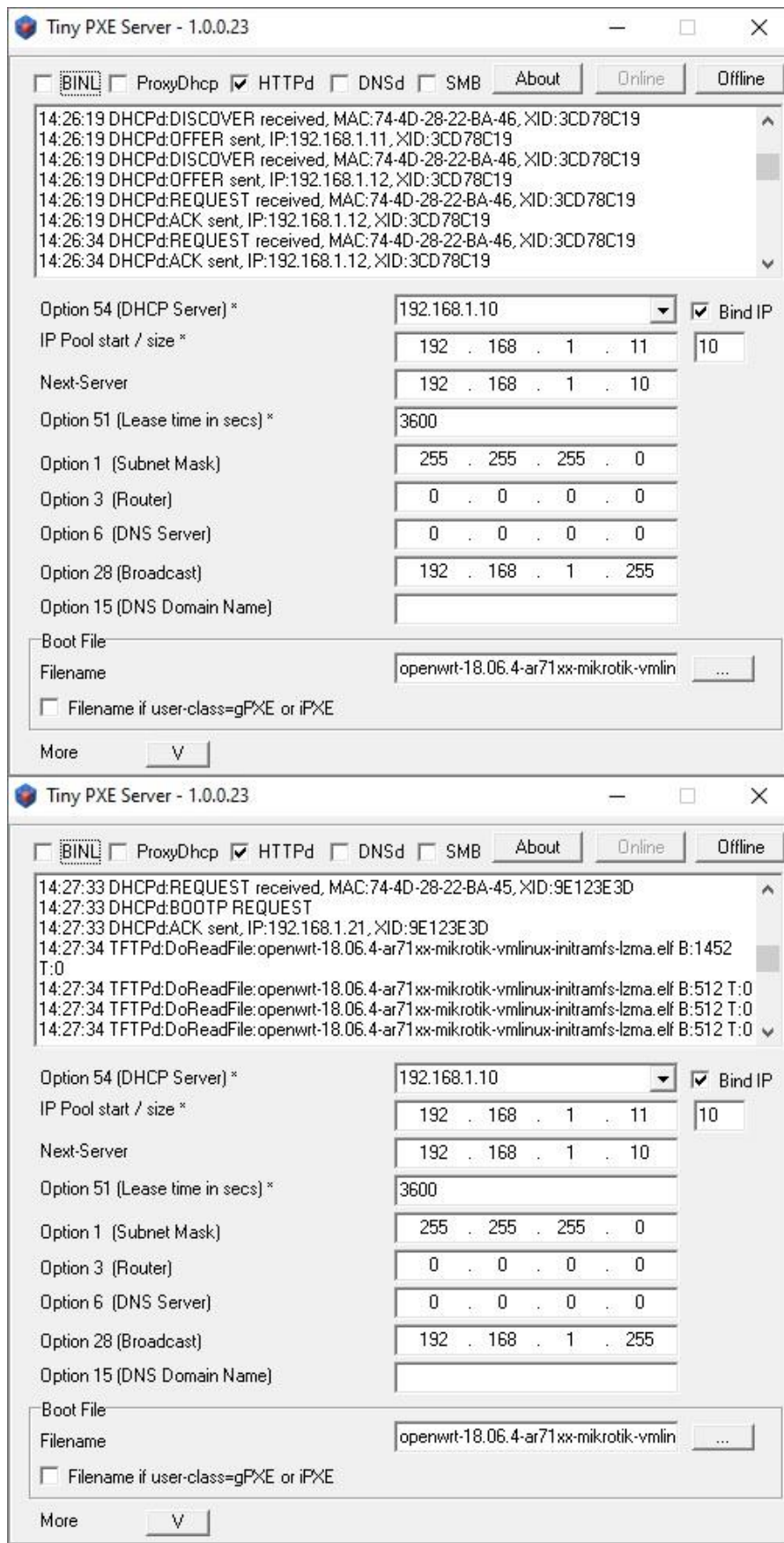


Immagine 3.4 – Scambio di messaggi DHCP tra il router e TinyPXE

- In corrispondenza della seconda schermata dell'Immagine 3.4, in cui si vedono le righe che iniziano con "DoReadFile", sul display LCD del router appare una scritta riportante "ether boot", a indicare che il router è stato forzato con successo all'avvio dalla rete a cui è collegato, ossia che il sistema è in attesa dell'avvio del netbooting (Immagine 3.5)



Immagine 3.5 – Display LCD del router riportante la scritta “ether boot”, a indicare che il sistema è in attesa del netbooting

- Per finalizzare il netbooting tenere premuto il pulsante Reset del router finché questo non emette un breve segnale acustico (del tutto uguale a quello che si sente quando si avvia il dispositivo con RouterOS), a conferma che l'immagine *initramfs.elf* è stata correttamente caricata nella RAM del dispositivo e che l'avvio di OpenWRT come SO non permanente è avvenuto con successo.
- Dopo aver verificato la compatibilità di OpenWRT con il dispositivo, per installare quest'ultimo in maniera definitiva si dovrà cambiare l'interfaccia ethernet del router, non utilizzando più quindi la ethernet1, ma un'altra qualsiasi tra quelle comprese dalla 2 alla 9.

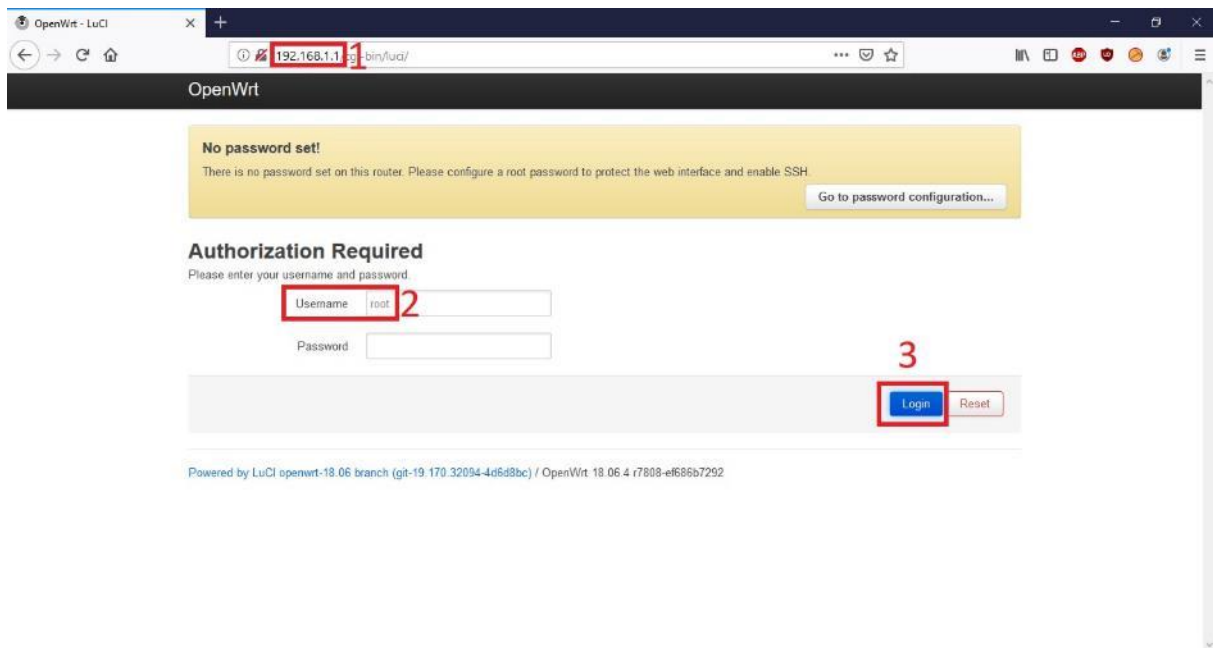


Immagine 3.6 – Schermata della form di login di OpenWRT in netbooting

Seconda parte: flash in memoria di massa dell'immagine di OpenWRT come SO permanente

Dopo essersi connessi tramite un browser da PC all'indirizzo assegnato automaticamente da TinyPXE all'interfaccia ether1 del router, <http://192.168.1.1>, e aver eseguito l'accesso al pannello di configurazione di OpenWRT ("OpenWRT LuCI WebUI") come indicato nell'Immagine 3.6 di cui sopra, per poter installare in maniera definitiva il nuovo SO, bisogna selezionare il relativo file *sysupgrade.bin* seguendo quanto illustrato nell'Immagine 3.7 .

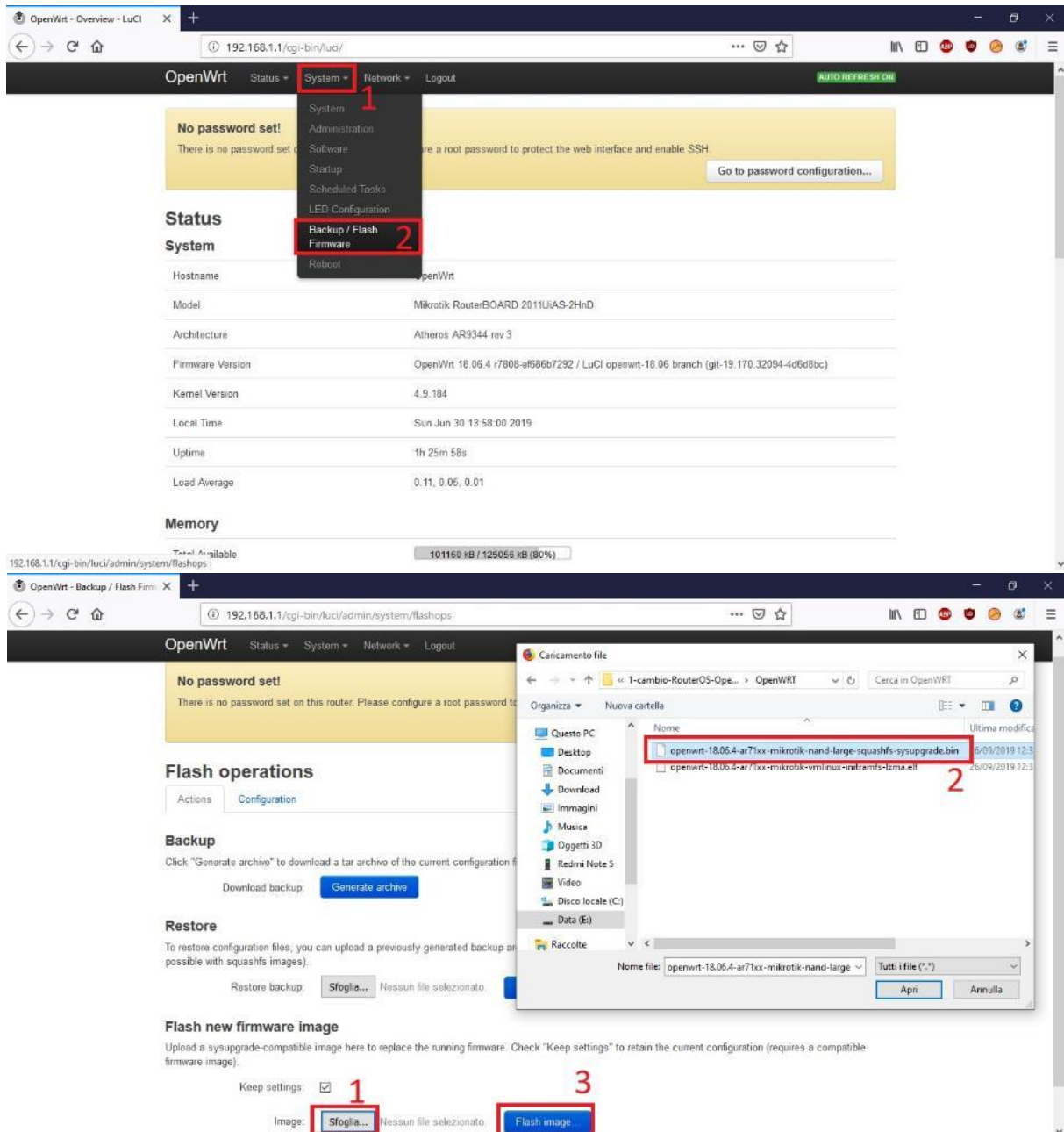


Immagine 3.7 – Selezione dell'immagine sysupgrade.bin per il flash permanente di OpenWRT

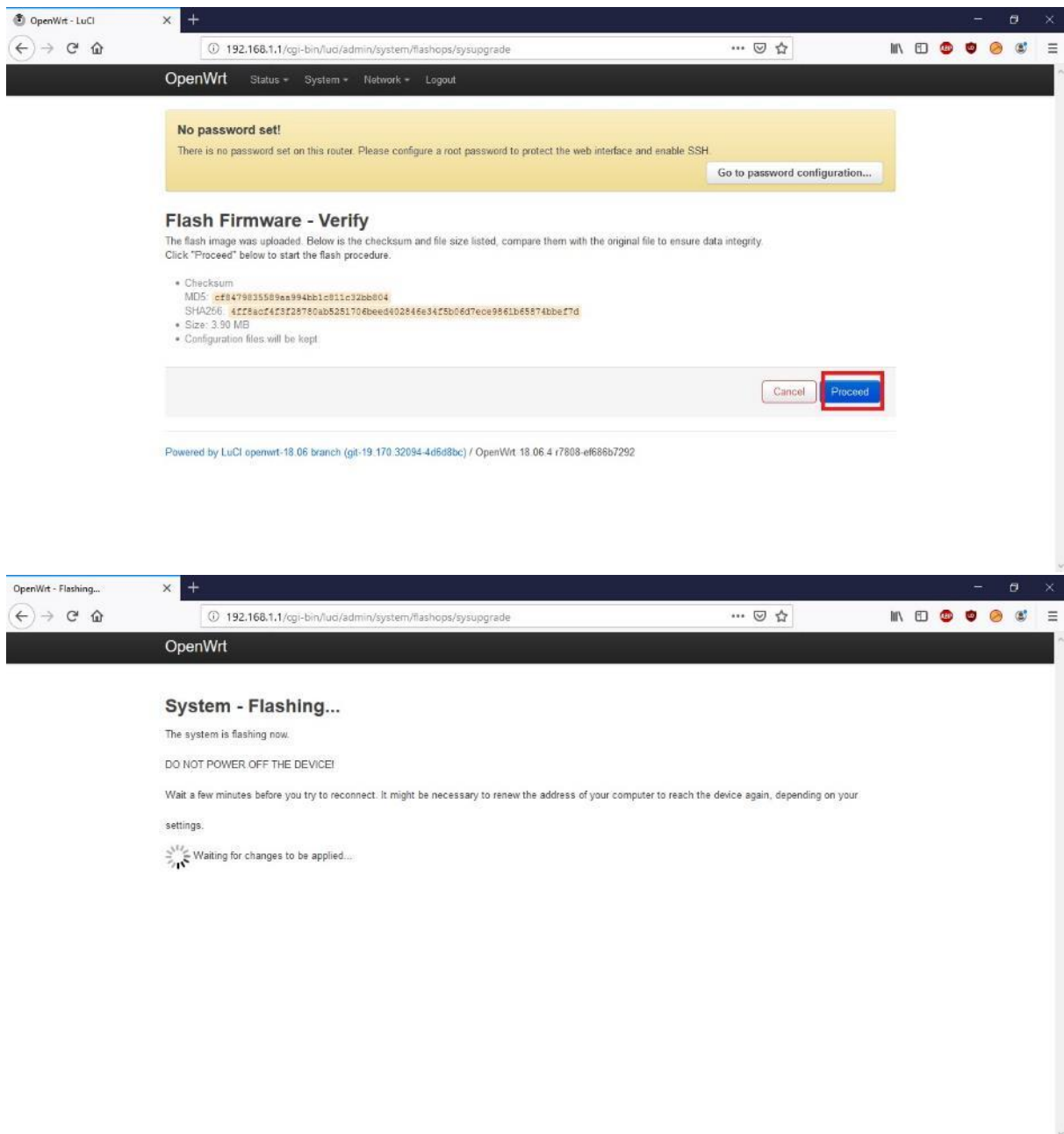


Immagine 3.8 – Flashing permanente del sistema

Se tutto è andato a buon fine, dopo aver visualizzato le schermate da browser come nell'Immagine 3.8, il router si riavvierà caricando dapprima il kernel, che rimarrà l'originale Mikrotik, con ora installato il SO OpenWRT in maniera permanente. Nello specifico, dopo aver percepito il segnale acustico di avvio in seguito al flash dell'immagine di OpenWRT, sul display LCD del router dovranno essere visualizzate, in sequenza, la scritta "routerboot" e "starting kernel", la quale rimarrà costantemente per tutta la durata di funzionamento del router.

Nota: si può ricaricare la pagina di LuCI (all'indirizzo <http://192.168.1.1>) e continuare a utilizzare OpenWRT immediatamente dopo il flash definitivo dell'immagine del SO, senza necessità di ulteriori azioni.

INSTALLAZIONE E AVVIO DI WEBSERVER (uHTTPd) E INTERPRETE PHP (PHP7)

Lo stack standard raccomandato dagli sviluppatori di OpenWRT per l'implementazione di servizi web sui dispositivi supportati è "LAMP"^[13], di cui se ne mostra di seguito l'installazione, nello specifico di uHTTPd come webserver, l'interprete PHP7 come interprete di scripting e, nel successivo capitolo, di MySQL/MariaDB come DBMS. Segue la procedura d'installazione.

[Collegamento SSH con il router, connessione alla rete WAN e aggiornamento dell'elenco dei pacchetti d'installazione](#)

Una volta connessi in SSH via ethernet con il router (operazione per la quale si raccomanda l'utilizzo del software "PuTTY"^[14]) come da Immagine 4.1 e 4.2, si deve cominciare con l'update dell'elenco dei pacchetti d'installazione disponibili dalle repositories di OpenWRT.

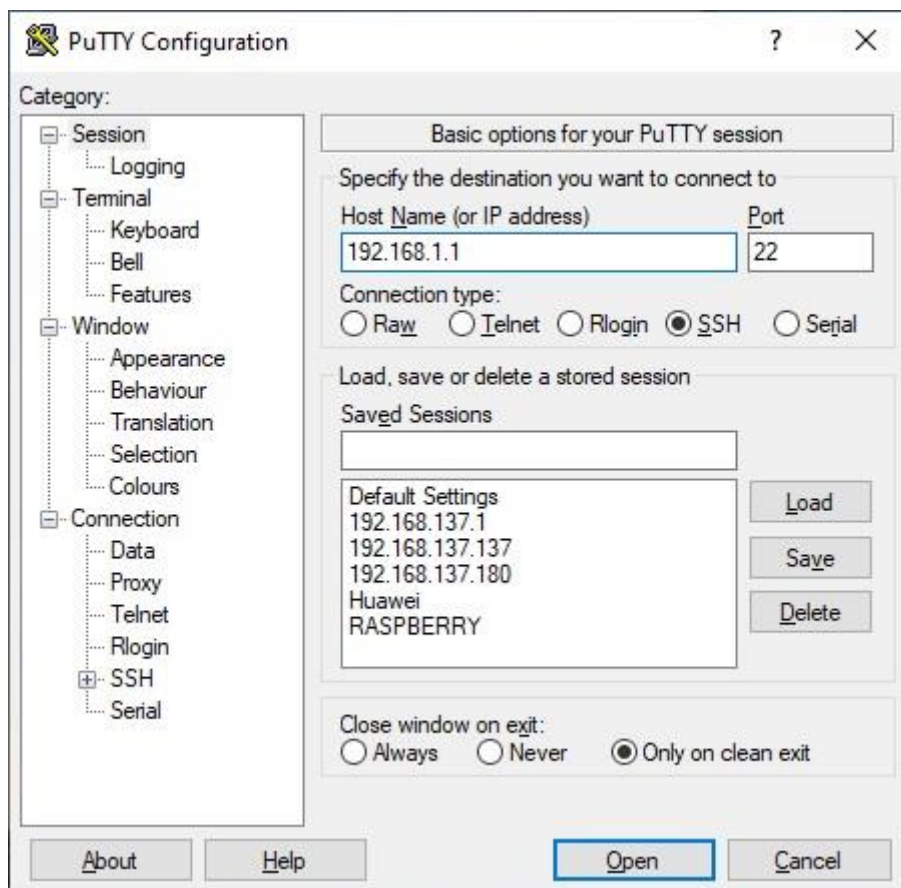


Immagine 4.1 – Parametri per la connessione in SSH al router tramite il software PuTTY

Assicurarsi di connettere il router a internet tramite wireless o, tramite altra porta ethernet, a un punto di accesso cablato.

Per la procedura di connessione a una rete wireless dalla WebUI di OpenWRT, seguire le Immagini 4.3, 4.4, 4.5:

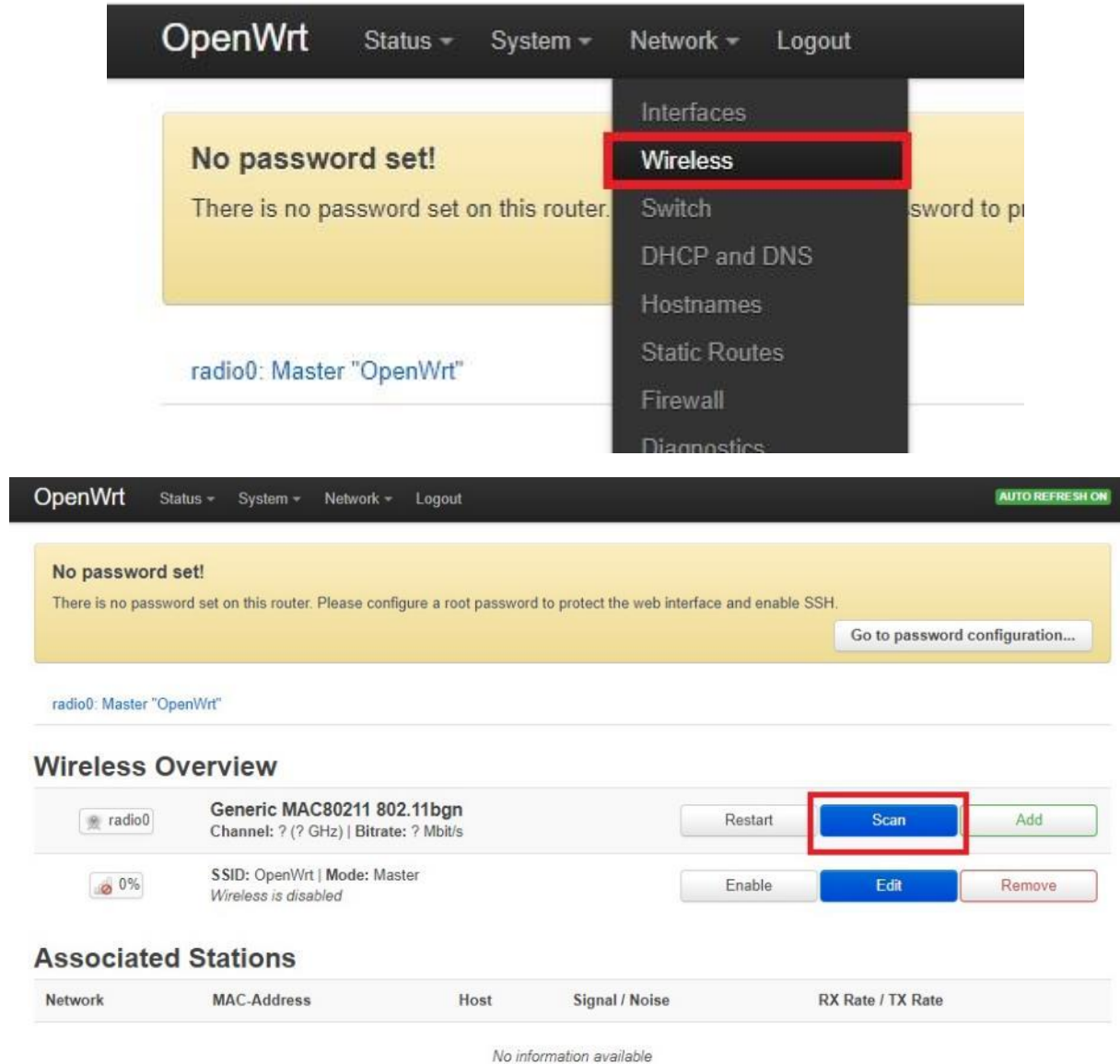


Immagine 4.3 – Connessione a rete Wireless dalla WebUI di OpenWRT (scansione reti)

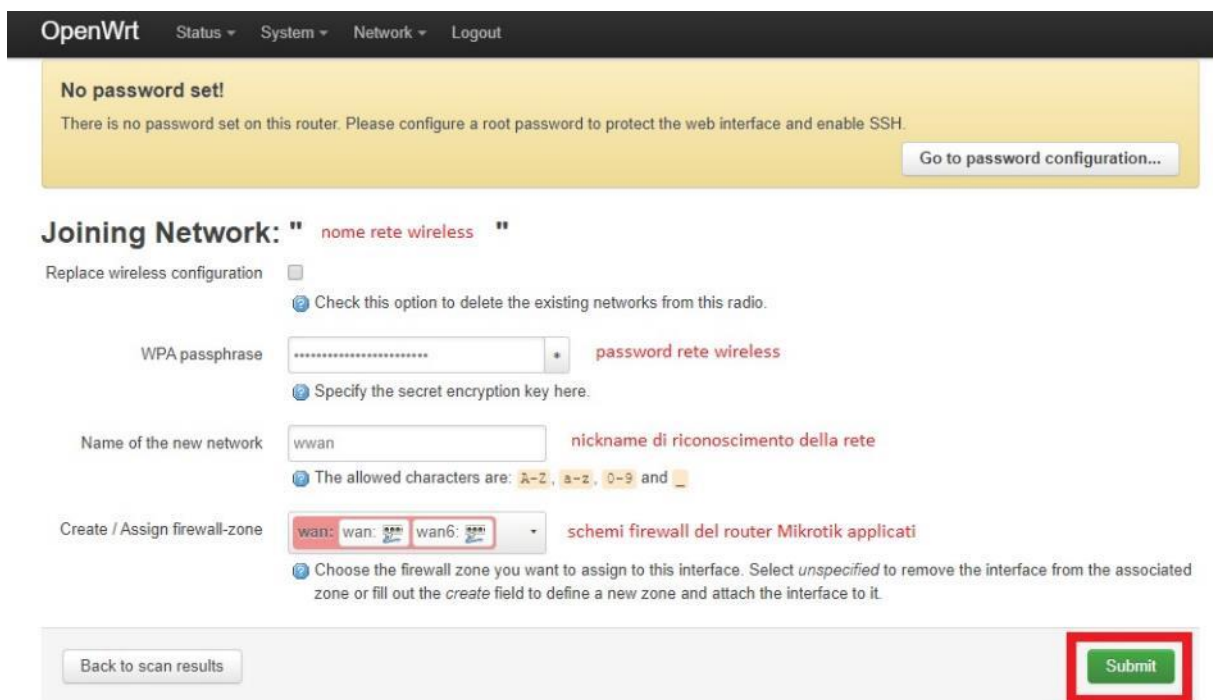
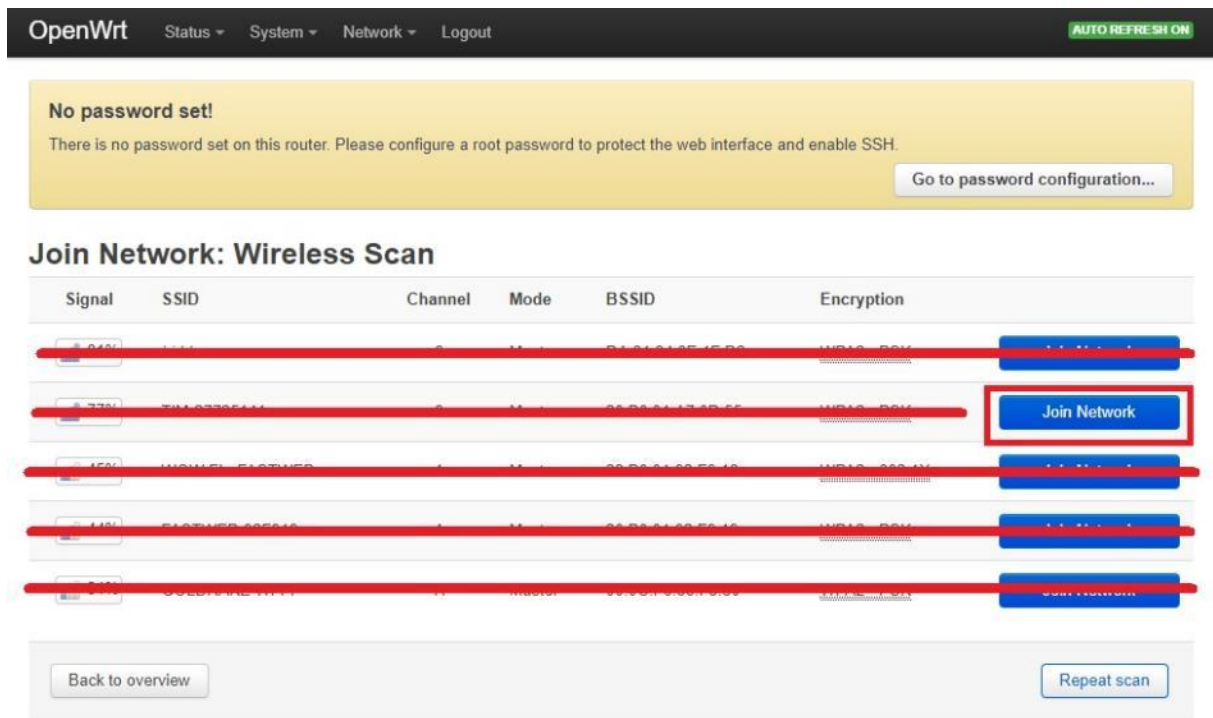


Immagine 4.4 – Connessione a rete Wireless dalla WebUI di OpenWRT (configurazione rete)

Alla schermata mostrata nell’Immagine 4.4 seguirà un’ulteriore pagina di riepilogo che riporta anche alla finestra di configurazione avanzata, utile in particolare per poter selezionare lo standard wireless (per nazione) per cui il punto di accesso wireless cui ci si conatterà è configurato. Il router Mikrotik deve essere configurato come “Client” (combo box “Mode” della scheda “General Setup” sotto la form “Interface Configuration”. Si vedano le schermate delle opzioni avanzate dell’Immagine 4.5).

OpenWrt Status System Network Logout AUTO REFRESH ON

Device Configuration

General Setup **Advanced Settings**

Status Mode: Client | SSID:
 58% BSSID:
 Encryption:
 Channel:
 Tx-Power:
 Signal:
 Bitrate:

Informazioni sulla rete wireless a cui ci si sta connettendo

Wireless network is enabled Disable

Operating frequency: Mode: N Channel: 8 (2437 MHz) Width: 20 MHz

Transmit Power: auto dBm

Interface Configuration

General Setup **Wireless Security** Advanced Settings

Mode: Client

ESSID:

BSSID:

Network: wlan:

Choose the network(s) you want to attach to this wireless interface or fill out the create field to define a new network.

Back to Overview
Save & Apply
Save
Reset

OpenWrt Status System Network Logout AUTO REFRESH ON

Device Configuration

General Setup **Advanced Settings**

Country Code: IT - Italy

Use ISO/IEC 3166 alpha2 country codes.

Allow legacy 802.11b rates:

Distance Optimization:
Distance to farthest network member in meters.

Fragmentation Threshold:

RTS/CTS Threshold:

Force 40MHz mode:
Always use 40MHz channels even if the secondary channel overlaps. Using this option does not comply with IEEE 802.11n-2009!

Beacon Interval:

Immagine 4.5 – Connessione a rete Wireless dalla WebUI di OpenWRT (opzioni avanzate)

Webserver: uHTTPd

Il webserver scelto per l'utilizzo che dovrà avere il router in questione è uHTTPd, già installato e abilitato di default su OpenWRT^{[15][16]}(se lo si vuole aggiornare all'ultima versione, eseguire ***opkg update && opkg install uhttpd***).

L'immagine di default di uHTTPd esegue un'interfaccia web per OpenWRT sulla porta 80 (HTTP) e sulla porta 443 (HTTPS). Si dovrà allora avviare una nuova istanza di uHTTPd su una porta differente (in questo caso viene scelta la 81) tramite i comandi: ^[17]

```
uci set uhttpd.llmp=uhttpd
```

```
// comando per settare il servizio LLMP sul file uhttpd
```

```
uci set uhttpd.llmp.listen_http=81
```

```
// comando per mettere in ascolto LLMP sulla porta 81 per il protocollo HTTP
```

```
uci set uhttpd.llmp.home=/srv/www
```

```
// comando per settare il path della root directory per i file dello stack LLMP su /srv/www[18]
```

```
uci commit uhttpd
```

```
// salva le modifiche apportate nel file di configurazione relativo a uHTTPd in /etc/config
```

```
mkdir -p $(uci get uhttpd.llmp.home) [19]
```

```
// crea una cartella per il contenuto del webserver leggendo il parametro restituito dalla  
funzione uhttpd.llmp.home (il '$' serve per far utilizzare al comando mkdir -p la stringa  
all'interno delle parentesi tonde come parametro di input. Nello specifico, verrà creata una  
cartella il cui path completo è l'output del comando uci get uhttpd.llmp.home)
```

Se uHTTPd era già stato installato e in stato di running, lo si dovrà riavviare con
/etc/init.d/uhttpd restart

Se lo si ha invece installato e/o aggiornato tramite i comandi di cui sopra, bisogna
inizializzarlo manualmente e abilitarlo in fase di boot di OpenWRT tramite
/etc/init.d/uhttpd start && /etc/init.d/uhttpd enable

Testare il webservice:

echo "<P>Hello, this web server runs on OpenWrt!!</P>" > /srv/www/index.html

Con il comando di cui sopra si crea il file "index.html" al path `/srv/www/index.html` e gli si dà il contenuto testuale "`<P>Hello, this web server runs on OpenWrt!!</P>`" (senza apici).

Detto file sarà raggiungibile collegandosi da una finestra del browser aperta dal PC attraverso l'indirizzo <http://192.168.1.1:81/index.html>

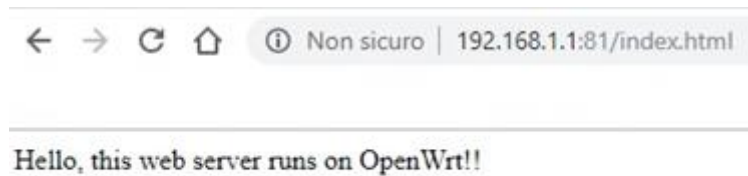


Immagine 4.7 – Visualizzazione del file index.html da browser

Installazione e configurazione PHP per uHTTPd ^[20]

opkg update && opkg install php7 php7-cgi ^[21]

/etc/init.d/uhttpd restart

installa le API utili al funzionamento di PHP7 e riavvia il webserver uHTTPd

uci add_list uhttpd.llmp.interpreter=".php=/usr/bin/php-cgi"

aggiunge alla lista degli interpreti quello relativo al PHP ("*php-cgi*") secondo il file al path specificato

uci set uhttpd.llmp.index_page="index.html index.htm default.html default.htm index.php"

imposta la pagina principale dello stack LLMP

uci commit uhttpd

salvataggio dei cambiamenti apportati nel file di configurazione di uHTTPd

sed -i 's,doc_root.*,doc_root = "",g' /etc/php.ini

modifica la stringa *doc_root.** in *doc_root = ""* all'interno del file */etc/php.ini*

sed -i 's,;short_open_tag = Off,short_open_tag = On,g' /etc/php.ini

modifica il valore della stringa *short_open_tag* da *Off* a *On* all'interno del file */etc/php.ini* ^[22]

/etc/init.d/uhttpd restart

riavvia il webserver uHTTPd

Testare PHP:

Si deve utilizzare la funzione `phpinfo()` per un test iniziale del funzionamento di PHP.

```
echo "<?php phpinfo(); ?>" > /srv/www/index.php
```

//scrive lo script `<?php phpinfo(); ?>` all'interno del file `/srv/www/index.php`

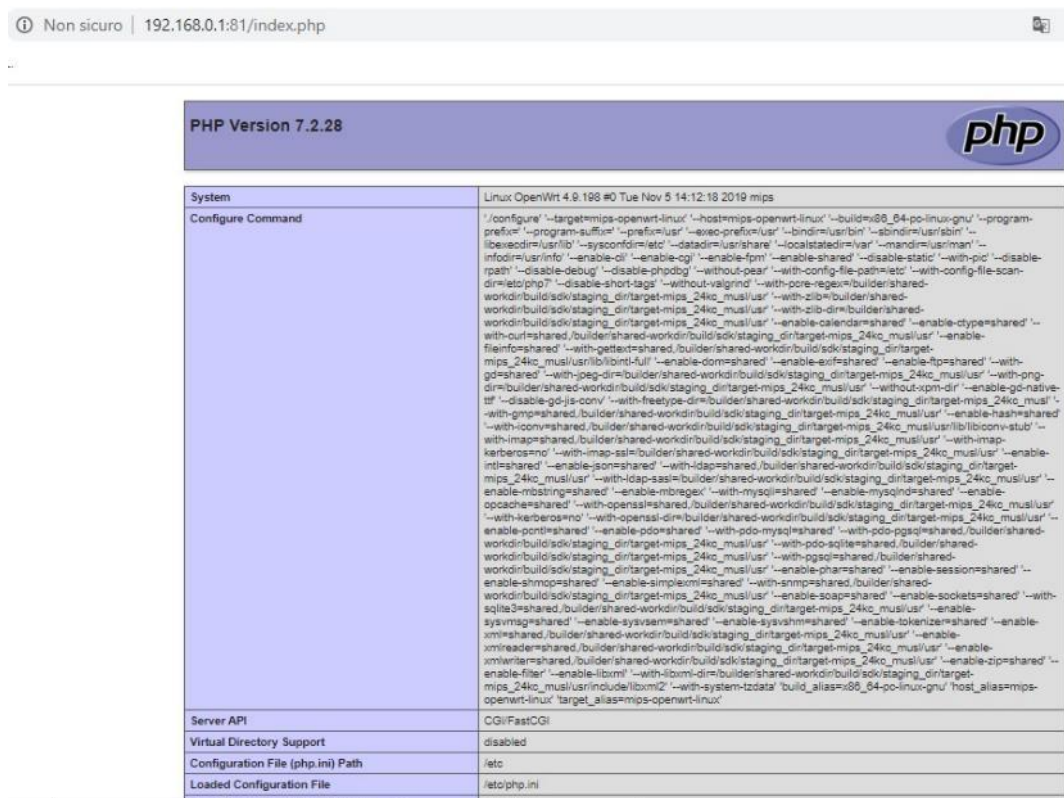
Provare quindi a visualizzare la pagina `index.php` puntando il browser alla pagina

<http://192.168.1.1:81/index.php> :

se si ottiene una pagina vuota o non raggiungibile si può eseguire lo script `php-cgi` dalla shell del router per visualizzare il contenuto del file `index.php` tramite il comando

```
php-cgi /srv/www/index.php
```

altrimenti si otterrà la pagina introduttiva di PHP come illustrato nell'Immagine 4.8



System	Linux OpenWrt 4.9.198 #0 Tue Nov 5 14:12:18 2019 mips
Configure Command	./configure '--target=mips-openwrt-linux' '--host=mips-openwrt-linux' '--build=x86_64-pc-linux-gnu' '--program-prefix=' '--program-suffix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib' '--sysconfdir=/etc' '--datadir=/usr/share' '--localstatedir=/var' '--mandir=/usr/man' '--infodir=/usr/info' '--enable-openssl' '--enable-curl' '--enable-fpm' '--enable-shared' '--disable-static' '--with-pic' '--disable-rpath' '--disable-debug' '--disable-phodog' '--without-pear' '--with-config-file-path=etc' '--with-config-file-scan-dir=/etc/php7' '--disable-short-tags' '--without-valgrind' '--with-pcre-regex=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-zlib=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-zlib-dir=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-calendar=shared' '--enable-ctype=shared' '--with-curl=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-fileinfo=shared' '--with-gettext=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-gmp=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-ldap=shared' '--with-ldap=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-ldap-sasl=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-mbstring=shared' '--enable-mbregex' '--with-mysql=shared' '--enable-mysqli=shared' '--enable-mysqlnd=shared' '--enable-openssl=shared' '--with-openssl=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-kerberos=no' '--with-openssl-dir=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-pcntl=shared' '--enable-pdo=shared' '--with-pdo-mysql=shared' '--with-pdo-pgsql=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-pdo-sqlite=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-pdo-sqlite=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-pgsql=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-phar=shared' '--enable-session=shared' '--enable-shmop=shared' '--enable-simplexml=shared' '--with-snmp=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-soap=shared' '--enable-sockets=shared' '--with-sqlite3=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-sybase=shared' '--enable-sybase-ctlib=shared' '--enable-sysvshm=shared' '--enable-tokenizer=shared' '--enable-xml=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-xmlreader=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-xmlwriter=shared,builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--enable-zip=shared' '--enable-filter' '--enable-libxml' '--with-libxml-dir=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-libxml-dir=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-system-locale=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-system-locale=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' '--with-system-locale=builder/shared-workdir/build/sdk/staging_dir/target-mips_24kc_musl/usr' --target-alias=mips-openwrt-linux
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	etc
Loaded Configuration File	etc/php.ini

Immagine 4.8 – Visualizzazione del file `index.php` da browser

Configurazione di MySQLi in PHP

Per abilitare MySQLi nell'installazione PHP eseguire
opkg update && opkg install php7-mod-mysqli

Per caricare l'estensione PHP *mysqli.so* nel file di configurazione di PHP */etc/php.ini*,
eseguire ***sed -i 's,;extension=mysqli.so,extension=mysqli.so,g' /etc/php.ini*** ^{[23][24]}

```
root@OpenWrt:~# cd /usr/lib/php
root@OpenWrt:/usr/lib/php# ls
hash.so      mysqli.so    mysqlnd.so
root@OpenWrt:/usr/lib/php#
```

Immagine 4.9 – Contenuto della cartella con le estensioni .so di PHP

Per rendere operativo MySQLi a lavorare con PHP, si dovrà infine aggiungere in fondo al contenuto del file *php.ini* (*/etc/php.ini*) quanto segue:

```
[MySQLi]
mysqli.allow_local_infile = On
mysqli.allow_persistent = On
mysqli.cache_size = 2000
mysqli.max_persistent = -1
mysqli.max_links = -1
mysqli.default_port = 3306
mysqli.default_socket = /var/run/mysqld/mysqld.sock
mysqli.default_host = 127.0.0.1
mysqli.default_user = root
mysqli.default_password = MySuperSecretPassword
mysqli.connect_timeout = 60
mysqli.trace_mode = Off
```

```
[MySQL]
mysql.allow_local_infile = On
mysql.allow_persistent = On
mysql.cache_size = 2000
mysql.max_persistent = -1
mysql.max_links = -1
mysql.default_port = 3306
mysql.default_socket = /var/run/mysqld/mysqld.sock
mysql.default_host = 127.0.0.1
mysql.default_user = root
mysql.default_password = MySuperSecretPassword
mysql.connect_timeout = 60
mysql.trace_mode = Off
```

La modifica al file `/etc/php.ini` può essere effettuata tramite il comando `vi /etc/php.ini` ^[25], inserendo il testo in fondo al file come da Immagine 4.10

```
root@OpenWrt: ~
max_file_uploads = 20

; Fopen wrappers

allow_url_fopen = On
allow_url_include = Off
;from="john@doe.com"
;user_agent="PHP"
default_socket_timeout = 60
;auto_detect_line_endings = Off

[MySQL]
mysql.allow_local_infile = On
mysql.allow_persistent = On
mysql.cache_size = 2000
mysql.max_persistent = -1
mysql.max_links = -1
mysql.default_port = 3306
mysql.default_socket = /var/run/mysqld/mysqld.sock
mysql.default_host = 127.0.0.1
mysql.default_user = root
mysql.default_password = MySuperSecretPassword
mysql.connect_timeout = 60
mysql.trace_mode = Off

[MySQLi]
mysqli.allow_local_infile = On
mysqli.allow_persistent = On
mysqli.cache_size = 2000
mysqli.max_persistent = -1
mysqli.max_links = -1
mysqli.default_port = 3306
mysqli.default_socket = /var/run/mysqld/mysqld.sock
mysqli.default_host = 127.0.0.1
mysqli.default_user = root
mysqli.default_password = MySuperSecretPassword
mysqli.connect_timeout = 60
mysqli.trace_mode = Off
```

Immagine 4.10 – Inserimento degli attributi di configurazione per rendere PHP compatibile con MySQLi

MySQL/MariaDB

Il package corrispondente all'installazione del DBMS contemplato dalle repository di OpenWRT è il fork^[26] di MySQL denominato "MariaDB"^[27], che attualmente è uno tra i fork più diffusi in molte implementazioni di MySQL per vari sistemi operativi e piattaforme hardware. MariaDB si differenzia dalla versione di prima concezione di MySQL principalmente per gli Storage Engine utilizzati: mentre fino alla versione MySQL 8.0 veniva utilizzato "MyISAM"^[28], MariaDB, nelle sue versioni più recenti, prevede la possibilità di utilizzare "Aria"^[29] (una versione migliorativa "transazionale"^[30] discendente direttamente da MyISAM), "InnoDB"^[31] e "XtraDB"^[32], un fork ottimizzato dello stesso InnoDB.

Gli Storage Engine sono librerie per MySQL e i DBMS da esso derivati che implementano la gestione fisica dei dati. Alcuni dei compiti loro affidati sono:

- scrittura e lettura dei record;
- indicizzazione;
- caching;
- esecuzione delle transazioni.

Come si vedrà in seguito nella presente trattazione in merito alla parte operativa riguardante l'installazione e l'avvio del servizio MySQL sul dispositivo in questione, lo Storage Engine predefinito di MariaDB è InnoDB (nello specifico il fork "XtraDB").

Nota: nell'avviare il server, se InnoDB/XtraDB non riuscisse ad avviarsi, il sistema proverà a implementare un'istanza di Aria, a meno che nel file di configurazione di MySQL/MariaDB, il cui path è */etc/mysql/my.cnf* non sia indicato altrimenti.

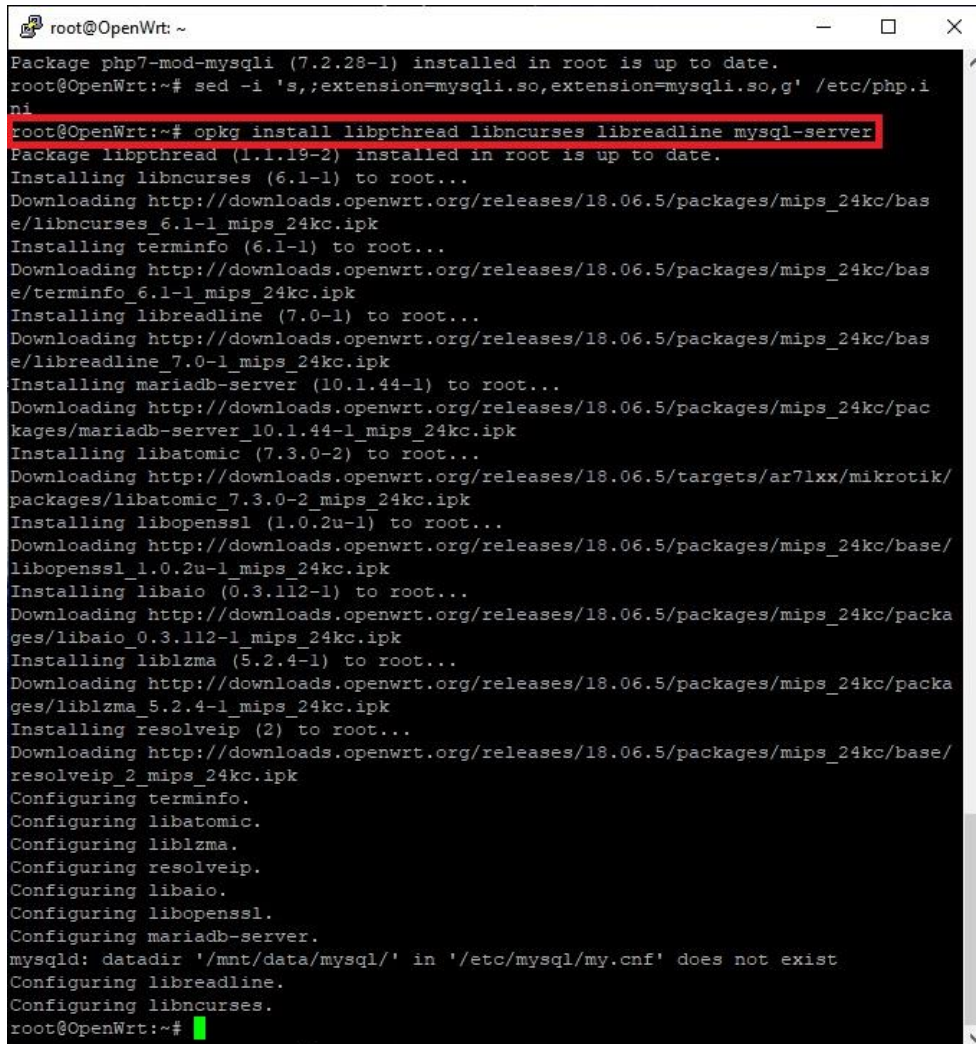
Installazione del server MySQL/MariaDB

opkg update

opkg install libpthread libncurses libreadline mysql-server ^[33]

In particolare il secondo comando installa le librerie e i file binari:

- libpthread^[34], per la gestione dei thread;
- libncurses^[35], per la gestione dell'interazione tra utente e interfaccia a riga di comando;
- libreadline^[36], per la gestione dell'acquisizione dei comandi dalla CLI;
- mysql-server^[37], package del server MariaDB.



```
root@OpenWrt: ~  
Package php7-mod-mysqli (7.2.28-1) installed in root is up to date.  
root@OpenWrt:~# sed -i 's,;extension=mysqli.so,extension=mysqli.so,g' /etc/php.ini  
root@OpenWrt:~# opkg install libpthread libncurses libreadline mysql-server  
Package libpthread (1.1.19-2) installed in root is up to date.  
Installing libncurses (6.1-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/libncurses_6.1-1_mips_24kc.ipk  
Installing terminfo (6.1-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/terminfo_6.1-1_mips_24kc.ipk  
Installing libreadline (7.0-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/libreadline_7.0-1_mips_24kc.ipk  
Installing mariadb-server (10.1.44-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/packages/mariadb-server_10.1.44-1_mips_24kc.ipk  
Installing libatomic (7.3.0-2) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/targets/ar71xx/mikrotik/packages/libatomic_7.3.0-2_mips_24kc.ipk  
Installing libopenssl (1.0.2u-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/libopenssl_1.0.2u-1_mips_24kc.ipk  
Installing libaio (0.3.112-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/packages/libaio_0.3.112-1_mips_24kc.ipk  
Installing liblzma (5.2.4-1) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/packages/liblzma_5.2.4-1_mips_24kc.ipk  
Installing resolvconf (2) to root...  
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/resolvconf_2_mips_24kc.ipk  
Configuring terminfo.  
Configuring libatomic.  
Configuring liblzma.  
Configuring resolvconf.  
Configuring libaio.  
Configuring libopenssl.  
Configuring mariadb-server.  
mysqld: datadir '/mnt/data/mysql/' in '/etc/mysql/my.cnf' does not exist  
Configuring libreadline.  
Configuring libncurses.  
root@OpenWrt:~#
```

Immagine 5.0 – Installazione dei package per il funzionamento del server MariaDB

Tramite il comando Linux `df -h`^[38]

che restituisce la quantità di spazio di archiviazione per ogni partizione rilevata dalla macchina Linux su cui viene eseguito (lo switch `-h` serve per rendere l'output del comando in formato "human-readable"), si può verificare la capacità di memoria di massa dei file-system letti, in gergo "montati", dal sistema operativo del dispositivo

```
root@OpenWrt: /
root@OpenWrt: /# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root        2.5M      2.5M      0 100% /rom
tmpfs            61.1M     84.0K     61.0M   0% /tmp
/dev/ubi0_2     106.2M    22.4M     79.1M  22% /overlay
overlayfs:/overlay 106.2M    22.4M     79.1M  22% /
tmpfs            512.0K      0      512.0K   0% /dev
root@OpenWrt: /#
```

Immagine 5.1 – L'output del comando `df -h`

Nel caso la macchina utilizzata sia dotata di un'esigua capacità di RAM e/o di memoria di massa, come nel caso del router oggetto della presente trattazione, non si potrebbe portare a termine la corretta inizializzazione e susseguente esecuzione del servizio MariaDB, condizione per cui si dovrebbe necessariamente ricorrere alla tecnica dello swap^[39], cioè all'utilizzo di una specifica porzione di memoria di massa come spazio di memoria centrale.

Viene spiegato di seguito il caso di utilizzo di una periferica di archiviazione USB da 4GB con due partizioni appositamente formattate:

- prima partizione da 2GB (1905 MiB) con file-system ext4^[40] come espansione per la memoria di massa;
- seconda partizione da 2GB (1906 MiB) con file-system linux-swap come espansione per la memoria centrale.

Guida all'espansione della memoria interna e di RAM del dispositivo sfruttando la capacità di una periferica di archiviazione USB

Partizionamento e formattazione del supporto di archiviazione^[41]

In Linux questa operazione può essere eseguita tramite utilities da riga di comando come “fdisk”^[42] o da software a interfaccia grafica. Per semplicità è consigliato avvalersi di un software a interfaccia grafica come “GParted”^[43] su un computer con installato Linux. Si è fatto uso di tale programma per la sperimentazione di cui alla presente Tesi per predisporre la periferica di archiviazione in dotazione come illustrato dalle immagini 5.2, 5.3, 5.4, 5.5, 5.6.

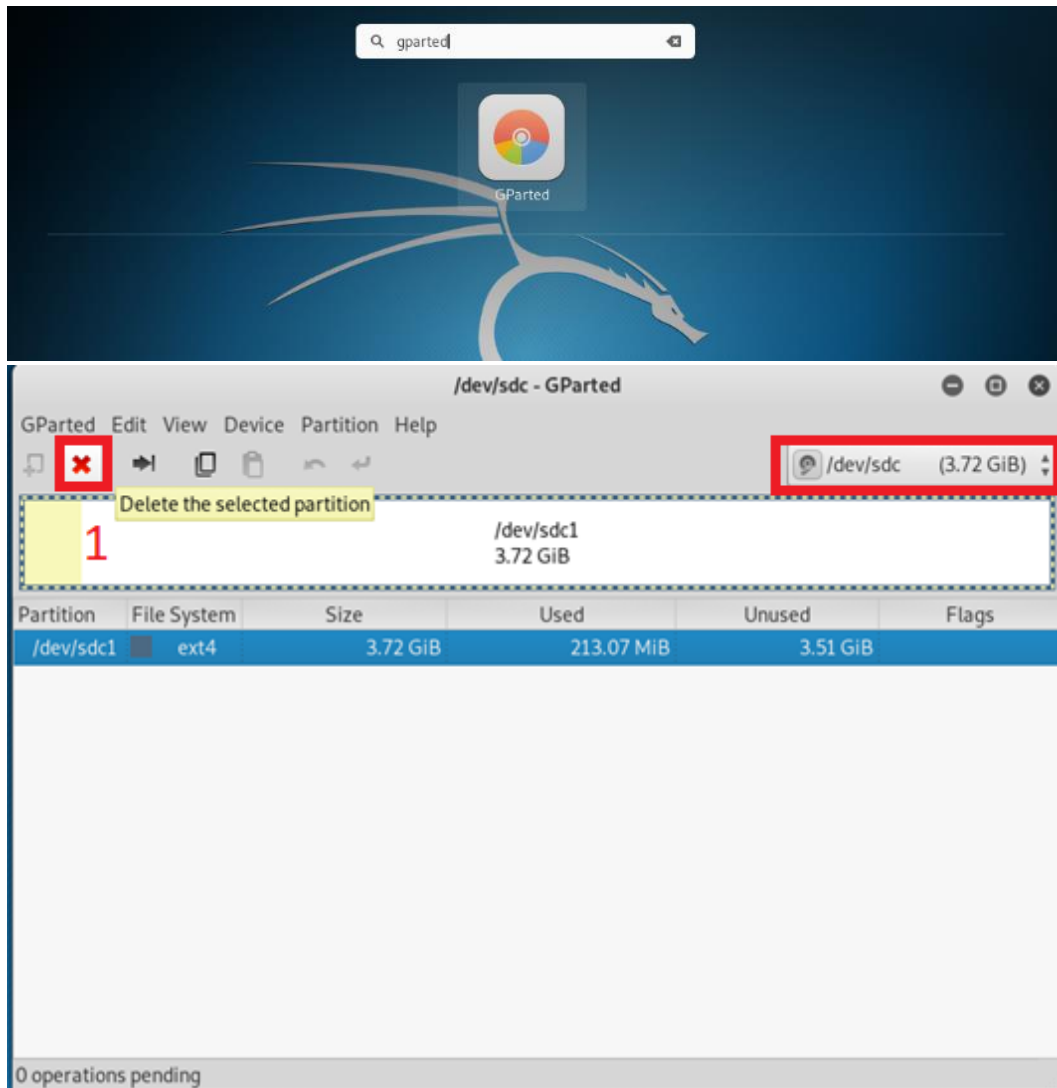


Immagine 5.2 – Da GParted si vede che inizialmente la periferica di archiviazione esterna ha un'unica partizione, individuata come */dev/sdc1*. Il primo passo da compiere è eliminarla.

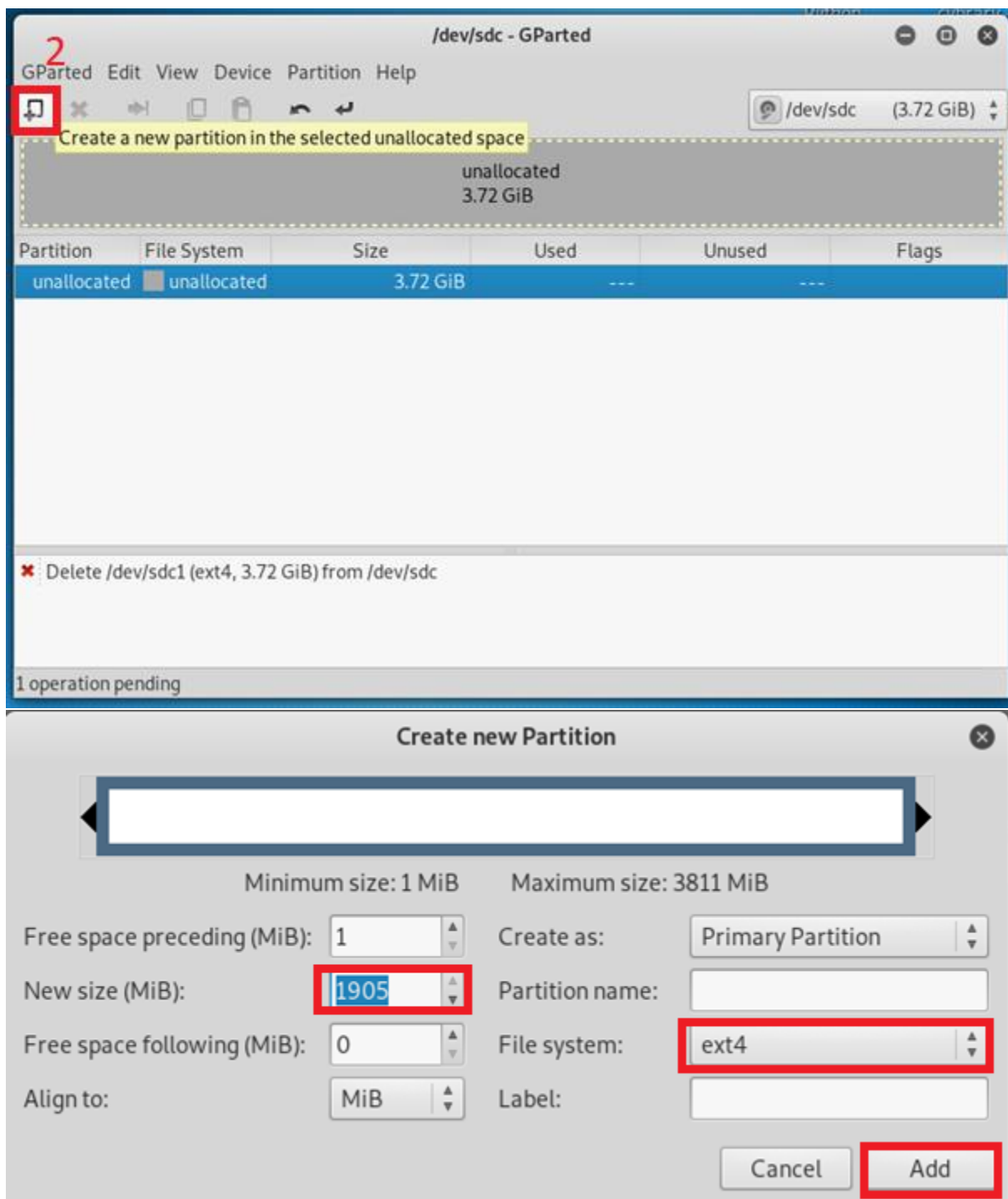


Immagine 5.3 – Creazione della partizione per l'espansione di memoria di massa

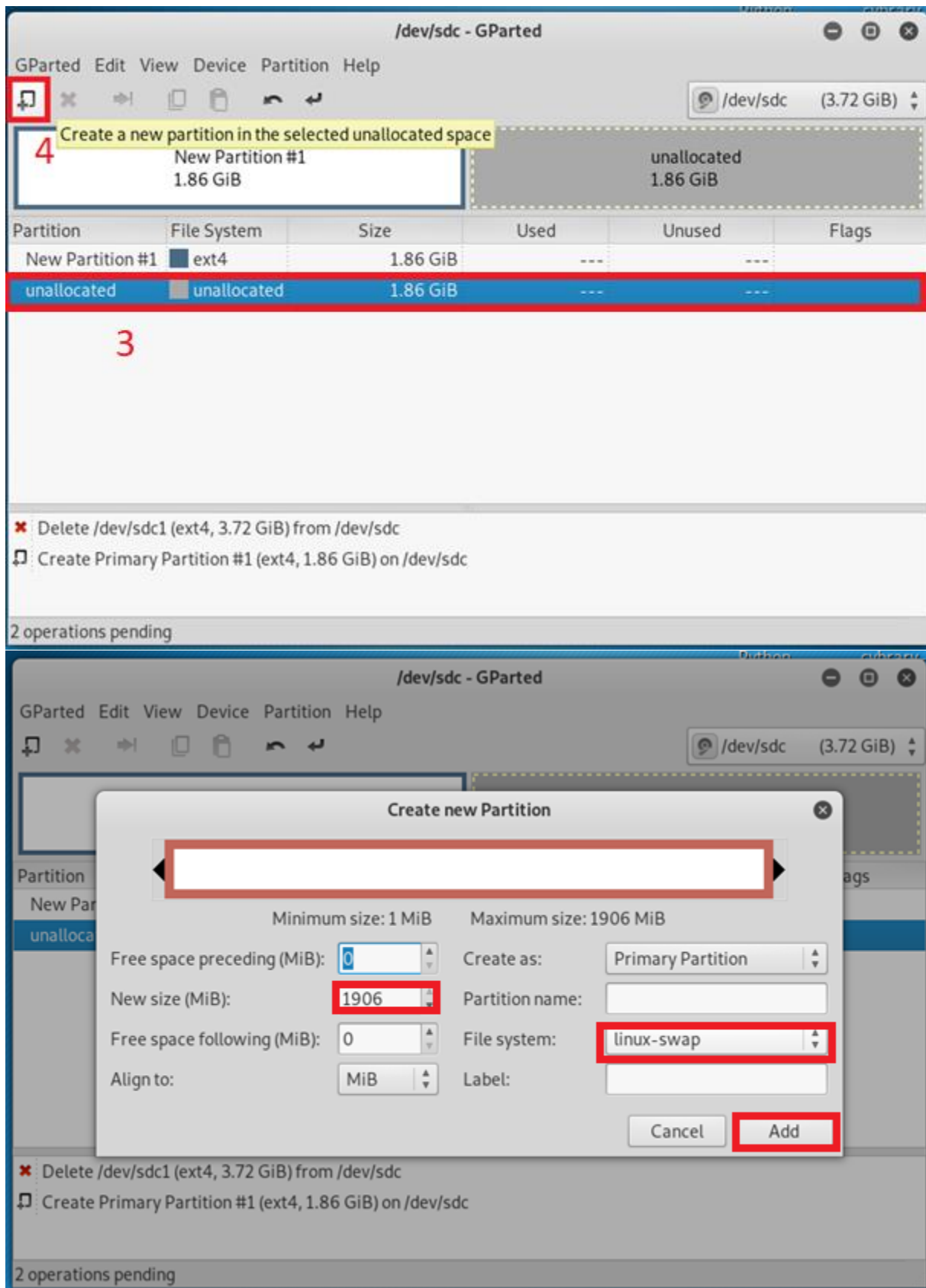


Immagine 5.4 - Creazione della partizione per l'espansione di memoria centrale

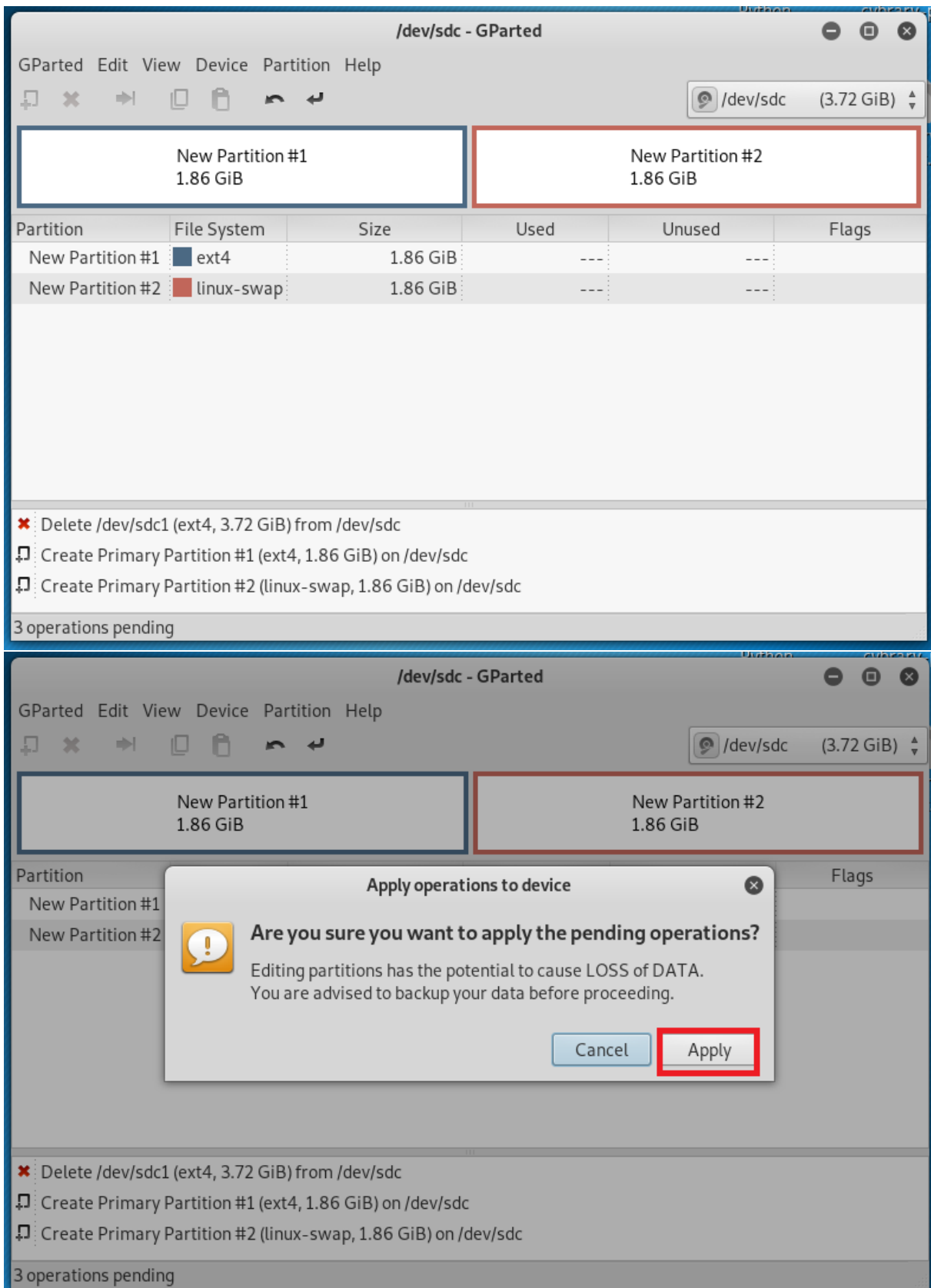


Immagine 5.5 – Prompt di conferma delle operazioni da effettuare sulle partizioni del supporto di archiviazione

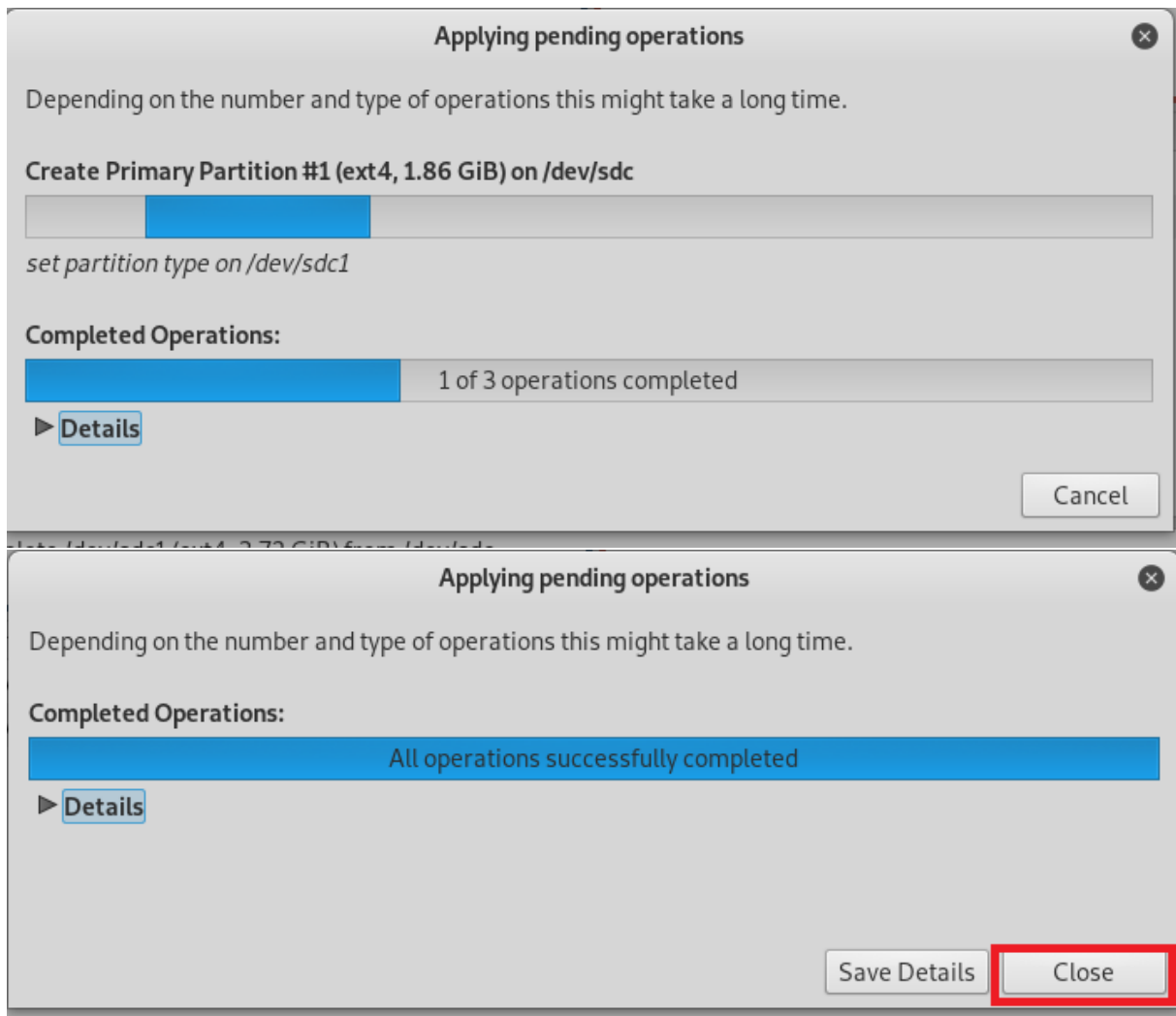


Immagine 5.6 – Completamento delle operazioni di partizionamento

Riconoscimento della periferica USB connessa al router^[44]

Dopo aver installato i package per la lettura dei blocchi di memoria del dispositivo e dei driver per il riconoscimento delle periferiche USB connesse tramite

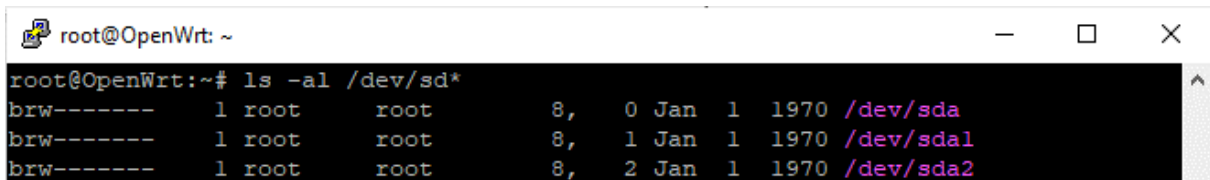
opkg update

opkg install block-mount e2fsprogs kmod-fs-ext4 kmod-usb-storage kmod-usb2 kmod-usb3

connettere il supporto USB nell'apposita porta del router ed eseguire

ls -al /dev/sd*

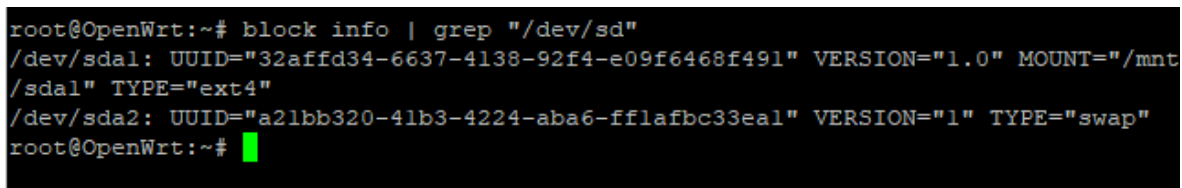
il cui output (dove */dev/sda* rappresenta il dispositivo esterno stesso, */dev/sda1* la partizione destinata ai dati con file-system ext4 e */dev/sda2* la partizione destinata a essere utilizzata come spazio di swap), corrispondente a quello dell'Immagine 5.7, indicherà che il router ne ha riconosciuto la connessione e conseguentemente ne avrà letto le partizioni.



```
root@OpenWrt: ~  
root@OpenWrt:~# ls -al /dev/sd*  
brw----- 1 root root      8,  0 Jan  1 1970 /dev/sda  
brw----- 1 root root      8,  1 Jan  1 1970 /dev/sda1  
brw----- 1 root root      8,  2 Jan  1 1970 /dev/sda2
```

Immagine 5.7 – L'output del comando *ls -al /dev/sd**

Nota: con ***block info | grep "/dev/sd"*** possono essere visualizzate le informazioni relative a tutte le partizioni dei file-system dei dispositivi di archiviazione di massa rilevati dal sistema. Con la seconda parte del comando viene specificato che essi devono rispondere al path *"/dev/sd"*, con cui generalmente le distribuzioni Linux fanno riferimento ai dispositivi di archiviazione di massa periferici quali hard disk e pendrive USB. In questo caso dovranno essere rilevate le sole due partizioni della periferica USB connessa, riconosciute rispettivamente come *sda1*(dati/ext4) e *sda2*(swap/linux-swap).



```
root@OpenWrt:~# block info | grep "/dev/sd"  
/dev/sda1: UUID="32affd34-6637-4138-92f4-e09f6468f491" VERSION="1.0" MOUNT="/mnt  
/sda1" TYPE="ext4"  
/dev/sda2: UUID="a21bb320-41b3-4224-aba6-ff1afbc33ea1" VERSION="1" TYPE="swap"  
root@OpenWrt:~#
```

Immagine 5.8 - L'output del comando *block info | grep "/dev/sd"*

Mounting e abilitazione del file-system della periferica di archiviazione USB

mkdir -p /mnt/sda1

crea la cartella */mnt/sda1* nel file-system del router.

mount /dev/sda1 /mnt/sda1

crea il punto di montaggio^[45] sulla cartella creata al passo precedente. In questo modo, per */mnt/sda1* si accede al file-system della chiavetta USB, in particolare alla partizione *sda1*, cioè quella destinata ai dati

block detect | uci import fstab

importa le informazioni dei blocchi rilevati dal sistema all'interno del file */etc/config/fstab*. In questo modo le due partizioni vengono importate e messe a disposizione del file-system del router. Ciò può essere verificato anche dalla WebUI di OpenWRT: puntando a 192.168.0.1 da browser, si potrà notare che la homepage è stata aggiornata con le informazioni riguardanti la memoria di Swap e che contestualmente il menù alla voce "System" si sarà arricchito della voce "Mount Points", accedendo alla quale si potranno visualizzare maggiori informazioni a riguardo(vedasi l'immagine 5.9).

192.168.0.1/cgi-bin/luci/

OpenWrt Status System Network Logout AUTO REFRESH ON

Status

System

Hostname	openWrt
Model	Mikrotik RouterBOARD 2011UiAS-2HnD
Architecture	atheros AR9344 rev 3
Firmware Version	openWrt 18.06.5 r7897-9d401013fc / LuCI openwrt-18.06 branch (git-19.309.48729-bc17ef6)
Kernel Version	4.19.108
Local Time	Tue Jun 23 12:04:13 2020
Uptime	1h 19m 48s
Load Average	0.01, 0.17, 0.55

Memory

Total Available	85992 kB / 125040 kB (62%)
Free	62288 kB / 125040 kB (49%)
Buffered	3704 kB / 125040 kB (2%)

Swap

Total Available	1951740 kB / 1951740 kB (100%)
Free	1951740 kB / 1951740 kB (100%)

- System
- Administration
- Software
- Startup
- Scheduled Tasks
- Mount Points
- LED Configuration
- Backup / Flash
- Firmware
- Reboot

Active Connections 87 / 16384 (0%)

192.168.0.1/cgi-bin/luci/admin/system/fstab

OpenWrt Status System Network Logout

Mount Points

Global Settings

Generate Config Generate Config

Find all currently attached filesystems and swap and replace configuration with defaults based on what was detected

Anonymous Swap Mount swap not specifically configured

Anonymous Mount Mount filesystems not specifically configured

Automount Swap Automatically mount swap on hotplug

Automount Filesystem Automatically mount filesystems on hotplug

Check filesystems before mount Automatically check filesystem for errors before mounting

Mounted file systems

Filesystem	Mount Point	Available	Used	Unmount
/dev/root	/rom	0.00 B / 2.50 MB	100% (2.50 MB)	
tmpfs	/tmp	60.55 MB / 61.05 MB	1% (516.00 KB)	
/dev/ubi0_2	/overlay	91.48 MB / 106.24 MB	10% (10.04 MB)	
overlayfs:overlay	/	91.48 MB / 106.24 MB	10% (10.04 MB)	
tmpfs	/dev	512.00 KB / 512.00 KB	0% (0.00 B)	
/dev/sda1	/mnt/sda1	1.58 GB / 1.80 GB	6% (111.97 MB)	Unmount

Immagine 5.9 – WebUI di OpenWRT con le informazioni sulle partizioni riconosciute dal router

Una volta verificato che siano presenti le voci relative alle due partizioni del supporto USB, bisogna abilitare queste ultime spuntando l'apposita casella di ciascuna e premendo sul tasto "Save and Apply" che si presenta in fondo (Immagine 5.10).

Le azioni effettuate sugli elementi di questa pagina corrispondono alla modifica dei parametri presenti nel file da cui Linux legge il comportamento che deve tenere su ogni punto di montaggio, swap e altri, denominato "fstab", ubicato al path `/etc/config/fstab`^[46].

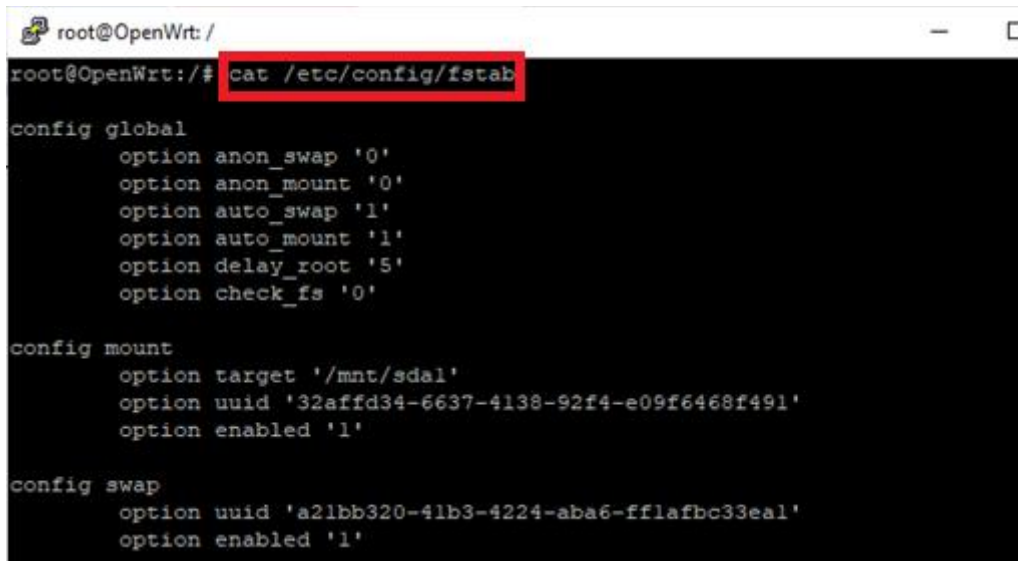
The screenshot shows the OpenWrt web interface for configuring the `fstab` file. The top navigation bar includes 'OpenWrt', 'Status', 'System', 'Network', and 'Logout'. The main content area is divided into several sections:

- Filesystem Table:** A table listing mounted filesystems with columns for Filesystem, Mount Point, Available space, Used space, and an Unmount button. The entries include `/dev/root`, `tmpfs`, `/dev/lubi0_2`, `overlayfs/overlay`, `tmpfs`, and `/dev/sda1`.
- Mount Points:** A section titled 'Mount Points' with a sub-header 'Mount Points define at which point a memory device will be attached to the filesystem'. It contains a table with columns: Enabled, Device, Mount Point, Filesystem, Options, Root, Check, Edit, and Delete. The checkbox in the 'Enabled' column for the USB partition is checked.
- SWAP:** A section titled 'SWAP' with a sub-header 'If your physical memory is insufficient unused data can be temporarily swapped to a swap-device resulting in a higher amount of usable RAM. Be aware that swapping data is a very slow process as the swap-device cannot be accessed with the high datarates of the RAM.' It contains a table with columns: Enabled, Device, Edit, and Delete. The checkbox in the 'Enabled' column is checked.
- Buttons:** At the bottom right, there are three buttons: 'Save & Apply' (highlighted with a red box), 'Save', and 'Reset'.

Immagine 5.10 – Abilitazione all’utilizzo delle partizioni della periferica USB da parte del router

Nota: se la partizione di swap non viene riconosciuta immediatamente dopo l’abilitazione data da WebUI (tornando alla pagina principale di quest’ultima non viene visualizzata cioè una voce il cui titolo è “SWAP”), riavviare il router con il comando **reboot**

Una volta apportate le modifiche di cui sopra, se ne può vedere il risultato nel file *fstab* con *cat /etc/config/fstab*



```
root@OpenWrt: /
root@OpenWrt: /# cat /etc/config/fstab

config global
    option anon_swap '0'
    option anon_mount '0'
    option auto_swap '1'
    option auto_mount '1'
    option delay_root '5'
    option check_fs '0'

config mount
    option target '/mnt/sdal'
    option uuid '32affd34-6637-4138-92f4-e09f6468f491'
    option enabled '1'

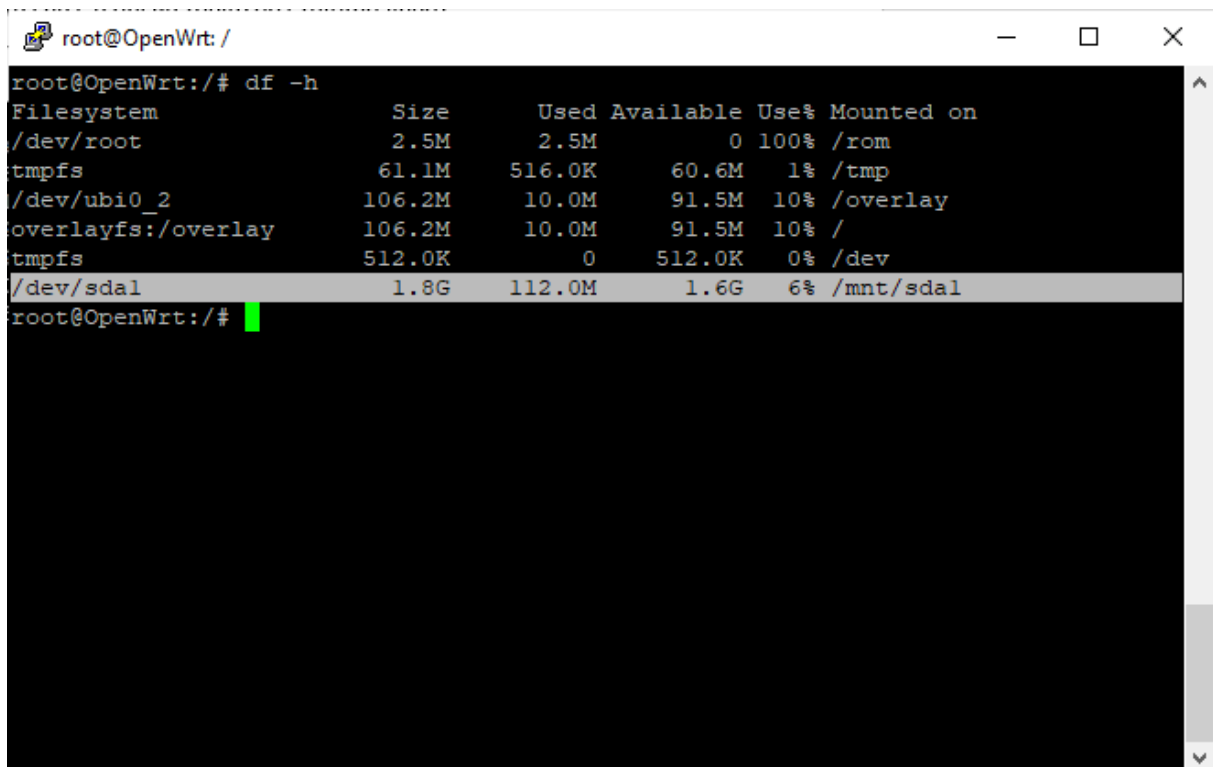
config swap
    option uuid 'a21bb320-41b3-4224-aba6-ff1afbc33eal'
    option enabled '1'
```

Immagine 5.11 – Il contenuto del file *fstab*

Rieseguendo

df -h

si può ora vedere che c'è anche */dev/sdal* (ultima linea dell'Immagine 5.12)



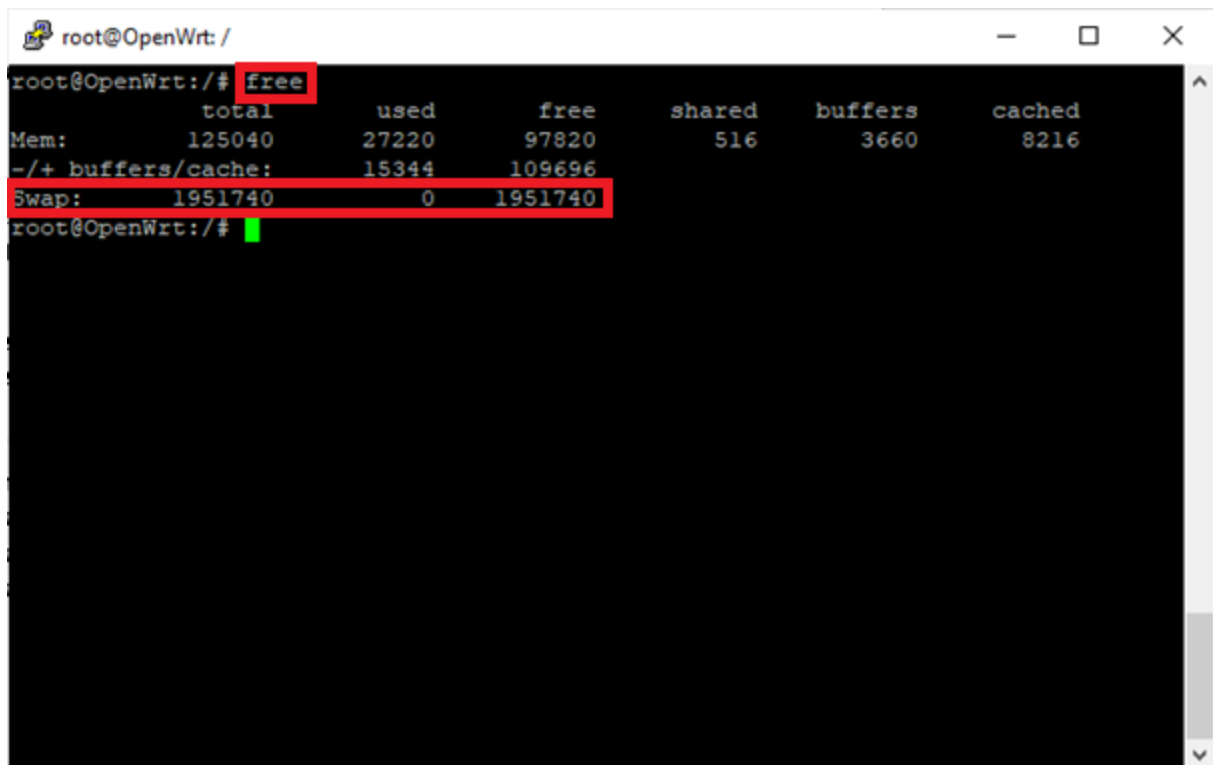
```
root@OpenWrt: /
root@OpenWrt: /# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root        2.5M       2.5M         0 100% /rom
tmpfs           61.1M     516.0K    60.6M    1% /tmp
/dev/ubi0_2     106.2M    10.0M    91.5M   10% /overlay
overlayfs:/overlay 106.2M    10.0M    91.5M   10% /
tmpfs           512.0K         0    512.0K    0% /dev
/dev/sdal       1.8G     112.0M    1.6G    6% /mnt/sdal
root@OpenWrt: /#
```

Immagine 5.12 – Riesecuzione di *df -h* con visualizzazione della partizione */dev/sdal*

Il comando

free

può essere utilizzato per verificare che è stato riconosciuto lo spazio di swap del supporto USB (ultima linea dell'Immagine 5.13)



```
root@OpenWrt: /
root@OpenWrt: /# free
              total        used        free      shared    buffers     cached
Mem:          125040        27220        97820         516       3660        8216
-/+ buffers/cache: 15344        109696
Swap:        1951740           0        1951740
root@OpenWrt: /#
```

Immagine 5.13 – Output del comando *free* in seguito all'abilitazione della partizione di swap

Creazione cartelle dati MariaDB e modifica di *my.cnf*

Si dovrà ora:

- procedere alla creazione delle cartelle^[47] su cui lavorerà MariaDB all'interno della partizione dei dati della chiavetta USB:
mkdir -p /mnt/sda1/mysql/data
mkdir -p /mnt/sda1/mysql/tmp
- dare i permessi di lettura e scrittura a tutti gli utenti registrati in OpenWRT per le cartelle cui si fa riferimento al punto precedente, in modo da non incorrere in eventuali problemi relativi ai permessi^[48]:
chmod 777 /mnt/sda1/mysql
chmod 777 /mnt/sda1/mysql/data
chmod 777 /mnt/sda1/mysql/tmp

In seguito, si modificano i parametri di *my.cnf* tramite l'editor di testo da riga di comando Linux "vi" con
vi /etc/mysql/my.cnf^[49]

- inserendo il path per la cartella *data* e per la cartella *tmp* con valore rispettivamente di */mnt/sda1/mysql/data* alla voce *datadir* e */mnt/sda1/mysql/tmp/* alla voce *tmpdir*;
- cambiando il valore di alcuni dei parametri per gli spazi di RAM contigui dedicati ai vari moduli di funzionamento di MariaDB e aggiungendo (se non già presenti) le stringhe
performance_schema = off
innodb_buffer_pool_size = 25M
innodb_log_buffer_size = 256K
innodb_use_native_aio = 0
key_buffer_size = 8M

[Nota: Alla [pagina 51](#) è presente un approfondimento sul significato di alcune delle stringhe che si possono incontrare all'interno del file *my.cnf* e la relazione che vi è tra esse e la necessità dell'ampliamento della capacità di memoria RAM e di archiviazione di massa per implementare con successo un'istanza del server MySQL/MariaDB].

Il risultato delle modifiche apportate a *my.cnf* dovrà rispecchiare l'Immagine 5.14

```
root@OpenWrt: ~
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

[mysqld]
user                = mariadb
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr

##### Don't put this on the NAND #####
# Figure out where you are going to put the databases
# And run mysql install db --force
datadir              = /mnt/sdal/mysql/data

##### This should also not go on the NAND #####
tmpdir              = /mnt/sdal/mysql/tmp

skip-external-locking

bind-address         = 127.0.0.1

# Fine Tuning
key_buffer_size     = 8M
max_allowed_packet  = 16M
thread_stack        = 192K
thread_cache_size   = 8

performance_schema  = off
innodb_buffer_pool_size = 25M
innodb_log_buffer_size = 256K
innodb_use_native_aio = 0

# Here you can see queries with especially long duration
#log_slow_queries   = /var/log/mysql/mysql-slow.log
#long_query_time    = 2
#log-queries-not-using-indexes

# The following can be used as easy to replay backup logs or for replication.
#server-id          = 1
#log_bin            = /var/log/mysql/mysql-bin.log
#expire_logs_days   = 10
I /etc/mysql/my.cnf [Modified] 32/59 54%
```

Immagine 5.14 – Il contenuto di *my.cnf* in seguito alle modifiche indicate

Server MariaDB: inizializzazione e abilitazione in fase di boot di OpenWRT

Con il comando^[50]

mysql_install_db --force

MariaDB deposita i propri file e alloca in RAM una propria istanza del server, il tutto seguendo i parametri specificati in *my.cnf*;

/etc/init.d/mysqld start (o, alternativamente ***service mysqld start***)

avvia il servizio *mysqld* secondo i parametri di configurazione letti da *my.cnf*

/etc/init.d/mysqld enable (o, alternativamente ***service mysqld enable***)

abilita l'avvio del servizio *mysqld* in fase di boot di OpenWRT

Client MariaDB

Al fine di installare il package contenente i binari del client MariaDB, si esegue ***opkg update && opkg install mariadb-client***

Nelle Immagini 5.15, 5.16, 5.17 viene mostrato il fitto verbose risultante dall'esecuzione dei comandi sopra citati:

```
root@OpenWrt:~# mysql_install_db --force
Installing MariaDB/MySQL system tables in '/mnt/sdal/mysql/data' ...
2020-06-27 10:42:36 2009795852 [Note] /usr/bin/mysqld (mysqld 10.1.45-MariaDB) starting as process 2580 ...
2020-06-27 10:42:37 2009795852 [Note] InnoDB: innodb_empty_free_list_algorithm has been changed to legacy because of small buffer pool size. In order to use backoff, increase buffer pool at least up to 20MB.
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Using mutexes to ref count buffer pool pages
2020-06-27 10:42:37 2009795852 [Note] InnoDB: The InnoDB memory heap is disabled
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2020-06-27 10:42:37 2009795852 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Using generic crc32 instructions
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Initializing buffer pool, size = 25.0M
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Completed initialization of buffer pool
2020-06-27 10:42:37 2009795852 [Note] InnoDB: The first specified data file ./ibdata1 did not exist: a new database to be created!
2020-06-27 10:42:37 2009795852 [Note] InnoDB: Setting file ./ibdata1 size to 12 MB
2020-06-27 10:42:44 2009795852 [Note] InnoDB: Setting log file ./ib_logfile101 size to 48 MB
2020-06-27 10:43:09 2009795852 [Note] InnoDB: Setting log file ./ib_logfile1 size to 48 MB
2020-06-27 10:43:35 2009795852 [Note] InnoDB: Renaming log file ./ib_logfile101 to
```

Immagine 5.15 – Output del comando *mysql_install_db --force* (parte 1)

```
PuTTY (inactive)
2020-06-27 10:43:35 2009795852 [Warning] InnoDB: New log files created, LSN=45781
2020-06-27 10:43:35 2009795852 [Note] InnoDB: Doublewrite buffer not found: creati
ng new
2020-06-27 10:43:37 2009795852 [Note] InnoDB: Doublewrite buffer created
2020-06-27 10:43:39 2009795852 [Note] InnoDB: 128 rollback segment(s) are active.
2020-06-27 10:43:39 2009795852 [Warning] InnoDB: Creating foreign key constraint s
ystem tables.
2020-06-27 10:43:39 2009795852 [Note] InnoDB: Foreign key constraint system table
s created
2020-06-27 10:43:39 2009795852 [Note] InnoDB: Creating tablespace and datafile sys
tem tables.
2020-06-27 10:43:39 2009795852 [Note] InnoDB: Tablespace and datafile system table
s created.
2020-06-27 10:43:39 2009795852 [Note] InnoDB: Waiting for purge to start
2020-06-27 10:43:39 2009795852 [Note] InnoDB: Percona XtraDB (http://www.percona.
com) 5.6.47-87.0 started; log sequence number 0
2020-06-27 10:43:42 1960897828 [Note] InnoDB: Dumping buffer pool(s) not yet start
ed
OK
Filling help tables...
2020-06-27 10:43:51 2004380940 [Note] /usr/bin/mysqld (mysqld 10.1.45-MariaDB) sta
rting as process 2807 ...
2020-06-27 10:43:51 2004380940 [Note] InnoDB: innodb_empty_free_list_algorithm has
been changed to legacy because of small buffer pool size. In order to use backoff
, increase buffer pool at least up to 20MB.

2020-06-27 10:43:51 2004380940 [Note] InnoDB: Using mutexes to ref count buffer po
ol pages
2020-06-27 10:43:51 2004380940 [Note] InnoDB: The InnoDB memory heap is disabled
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Mutexes and rw_locks use GCC atomic
builtins
2020-06-27 10:43:51 2004380940 [Note] InnoDB: GCC builtin __atomic_thread_fence()
is used for memory barrier
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Using generic crc32 instructions
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Initializing buffer pool, size = 25.
0M
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Completed initialization of buffer p
ool
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Highest supported file format is Bar
racuda.
2020-06-27 10:43:51 2004380940 [Note] InnoDB: 128 rollback segment(s) are active.
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Waiting for purge to start
```

Immagine 5.16 – Output del comando `mysql_install_db -force` (parte 2)

```
PuTTY (inactive)
2020-06-27 10:43:51 2004380940 [Note] InnoDB: Percona XtraDB (http://www.percona.com) 5.6.47-87.0 started; log sequence number 1616697
2020-06-27 10:43:52 1960000804 [Note] InnoDB: Dumping buffer pool(s) not yet started
OK
Creating OpenGIS required SP-s...
2020-06-27 10:43:59 2009894156 [Note] /usr/bin/mysqld (mysqld 10.1.45-MariaDB) starting as process 2858 ...
2020-06-27 10:43:59 2009894156 [Note] InnoDB: innodb_empty_free_list_algorithm has been changed to legacy because of small buffer pool size. In order to use backoff, increase buffer pool at least up to 20MB.

2020-06-27 10:43:59 2009894156 [Note] InnoDB: Using mutexes to ref count buffer pool pages
2020-06-27 10:43:59 2009894156 [Note] InnoDB: The InnoDB memory heap is disabled
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2020-06-27 10:43:59 2009894156 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Using generic crc32 instructions
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Initializing buffer pool, size = 25.0M
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Completed initialization of buffer pool
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Highest supported file format is Barracuda.
2020-06-27 10:43:59 2009894156 [Note] InnoDB: 128 rollback segment(s) are active.
2020-06-27 10:43:59 2009894156 [Note] InnoDB: Waiting for purge to start
2020-06-27 10:44:00 2009894156 [Note] InnoDB: Percona XtraDB (http://www.percona.com) 5.6.47-87.0 started; log sequence number 1616707
2020-06-27 10:44:00 1965514020 [Note] InnoDB: Dumping buffer pool(s) not yet started
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
To do so, start the server, then issue the following commands:

'/usr/bin/mysqladmin' -u root password 'new-password'
'/usr/bin/mysqladmin' -u root -h OpenWrt password 'new-password'
```

Immagine 5.17 – Output del comando `mysql_install_db -force` (parte 3)


```
PuTTY (inactive)
Alternatively you can run:
'/usr/bin/mysql_secure_installation'

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the MariaDB Knowledgebase at http://mariadb.com/kb or the
MySQL manual for more instructions.

You can start the MariaDB daemon with:
cd '/usr' ; /usr/bin/mysqld_safe --datadir='/mnt/sdal/mysql/data'

You can test the MariaDB daemon with mysql-test-run.pl
cd '/usr/mysql-test' ; perl mysql-test-run.pl

Please report any problems at http://mariadb.org/jira

The latest information about MariaDB is available at http://mariadb.org/.
You can find additional information about the MySQL part at:
http://dev.mysql.com
Consider joining MariaDB's strong and vibrant community:
https://mariadb.org/get-involved/

root@OpenWrt:~# /etc/init.d/mysqld start
root@OpenWrt:~# /etc/init.d/mysqld enable

root@OpenWrt:~# opkg update && opkg install mariadb-client
Downloading http://downloads.openwrt.org/releases/18.06.5/targets/ar7lxx/mikrotik/
packages/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_core
Downloading http://downloads.openwrt.org/releases/18.06.5/targets/ar7lxx/mikrotik/
packages/Packages.sig
Signature check passed.
Downloading http://downloads.openwrt.org/releases/18.06.5/targets/ar7lxx/mikrotik/
kmods/4.9.198-1-3ba2e8c5db2fc4389b537731cf37120c/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_kmods
Downloading http://downloads.openwrt.org/releases/18.06.5/targets/ar7lxx/mikrotik/
kmods/4.9.198-1-3ba2e8c5db2fc4389b537731cf37120c/Packages.sig
Signature check passed.
Downloading http://downloads.openwrt.org/releases/18.06.5/packages/mips_24kc/base/
Packages.gz
```

Immagine 5.18 – Output del comando `mysql_install_db -force` (parte 4) ed esecuzione dei comandi `/etc/init.d/mysqld start`, `/etc/init.d/mysqld enable`, `opkg update && opkg install mariadb-client`

Accesso al DBMS

Infine, tramite il comando

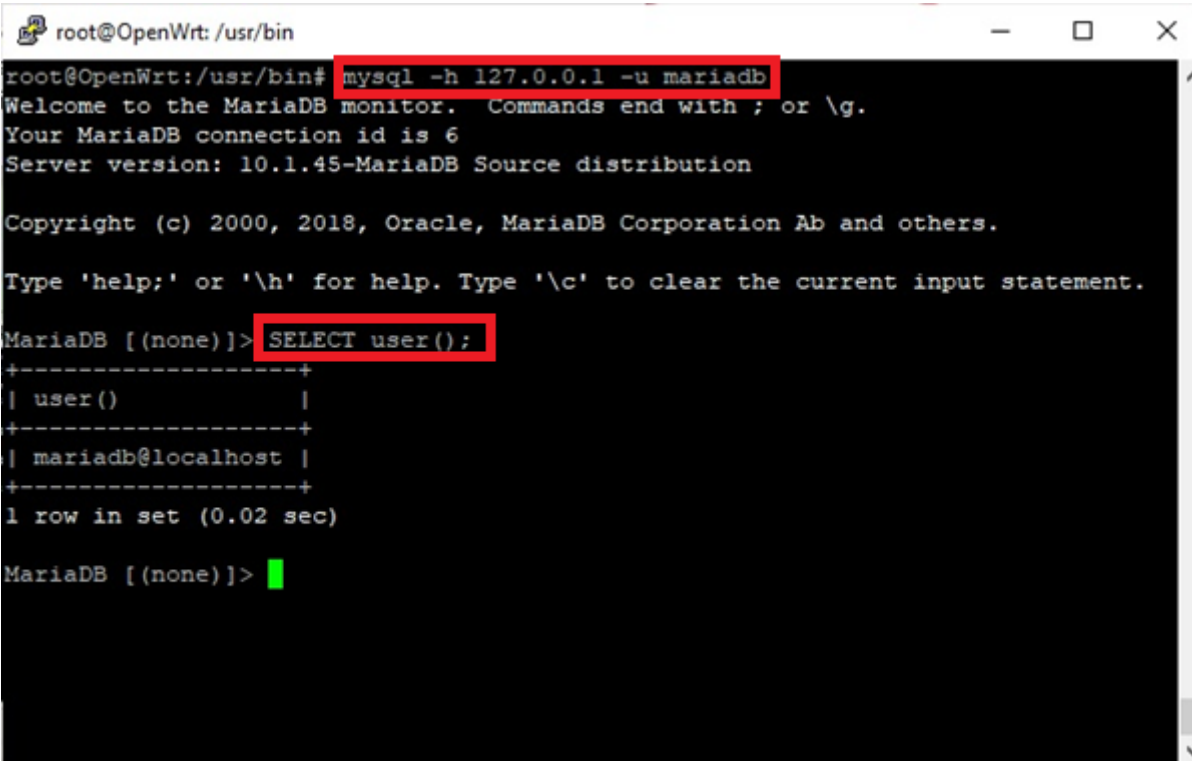
```
mysql -h 127.0.0.1 -u mariadb[51]
```

si lancia il client MariaDB tramite la socket creata in localhost (ossia all'indirizzo 127.0.0.1) con l'utente di default *mariadb*.

A ogni avvio del sistema operativo del router, per collegarsi al server MariaDB basterà eseguire quest'ultimo comando una volta connessi in SSH al router.

Come mostrato nell'Immagine 5.18, si può utilizzare una qualsiasi query di prova, come ad esempio *SELECT user()*; per testare la capacità di elaborazione query da parte dell'istanza del server MariaDB implementato.

Si potranno ora configurare gli utenti autorizzati ad accedere al DBMS^[52] appena ultimato e cominciare a usare il servizio in tutte le sue funzionalità.



```
root@OpenWrt: /usr/bin
root@OpenWrt:/usr/bin# mysql -h 127.0.0.1 -u mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.45-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT user();
+-----+
| user() |
+-----+
| mariadb@localhost |
+-----+
1 row in set (0.02 sec)

MariaDB [(none)]> █
```

Immagine 5.19 – esecuzione del comando *mysql -h 127.0.0.1 -u mariadb* e della query *SELECT user()*;

Da notare è il fatto che il metodo di implementazione dello stack LAMP adoperato prevede un utilizzo del file-system volto a mantenere tutte le librerie, i binari, i file e le configurazioni di più basso livello nella memoria interna del router (nei limiti della capacità di quest'ultima), mentre, immagazzinato nel supporto di archiviazione esterno, tutto ciò che riguarda lo spazio destinato ai dati utente, alle tabelle standard di MariaDB e in generale al contenuto di più alto livello.

DIFFICOLTÀ RISCONTRATE

- La configurazione di PHP7 e MySQL non sono stata immediate
Le repositories dei sorgenti puntavano a MariaDB piuttosto che al pacchetto MySQL originale. Dato che MariaDB utilizza ubicazioni simili ma comunque differenti rispetto a quelle di MySQL per il deposito dei propri file, è stata eseguita una serie di ricerche per portare a termine le dovute correzioni all'interno dei file di configurazione riguardanti il servizio di MySQL/MariaDB, in particolare per quanto concerne alcune stringhe relative ai path che puntavano ai file e/o directories per il corretto avvio ed esecuzione del daemon MySQLi.
- Impossibilità di cambiare l'indirizzo statico delle porte LAN da WebUI
Per fronteggiare questo problema si è dovuto quindi utilizzare la riga di comando da connessione SSH, andando a modificare il file `/etc/config/network` ^[53]. Per necessità sperimentali si è cambiato l'indirizzo della rete point-to-point in 192.168.0.0/24(192.168.0.1 sul router Mikrotik, DHCP client sulla stessa rete per il PC).
Il cambio è stato eseguito tramite editing sul file `/etc/config/network` con il comando `vi /etc/config/network`, come da Immagine 7.1 .

```
root@OpenWrt: ~
config interface 'loopback'
  option ifname 'lo'
  option proto 'static'
  option ipaddr '127.0.0.1'
  option netmask '255.0.0.0'

config globals 'globals'
  option ula_prefix 'fd4b:4f82:a0cc::/48'

config interface 'lan'
  option type 'bridge'
  option ifname 'eth0.1 eth1.1'
  option proto 'static'
  option ipaddr '192.168.0.1' → 192.168.1.1 → 192.168.0.1
  option netmask '255.255.255.0'
  option ipassign '60'

config interface 'wan'
  option ifname 'eth0.2'
  option proto 'dhcp'

config interface 'wan6'
  option ifname 'eth0.2'
  option proto 'dhcpv6'

config interface 'sfp'
  option ifname 'eth0.3'
  option proto 'none'

config switch
  option name 'switch0'
  option reset '1'
  option enable_vlan '1'

config switch_vlan
  option device 'switch0'
  option vlan '1'
  option ports '2 3 4 5 0t'

config switch_vlan
  option device 'switch0'
  option vlan '2'
- /etc/config/network 1/66 1%
```

Immagine 7.1 – Modifica al file `/etc/config/network`

PROCEDURA DI RIPRISTINO ROUTER IN SEGUITO AD AGGIORNAMENTO DI FIRMWARE OpenWRT NON ANDATO A BUON FINE^[54]

Dopo un tentativo di aggiornamento di OpenWRT a una build più recente a quella installata in fase di sperimentazione il router ha presentato il problema di bootloop. La procedura di seguito riportata illustra come ripristinare RouterOS e dunque ricreare le condizioni per rendere nuovamente operativo il router, nonché pronto all'installazione di una build funzionante di OpenWRT come spiegato [all'inizio della presente trattazione](#)

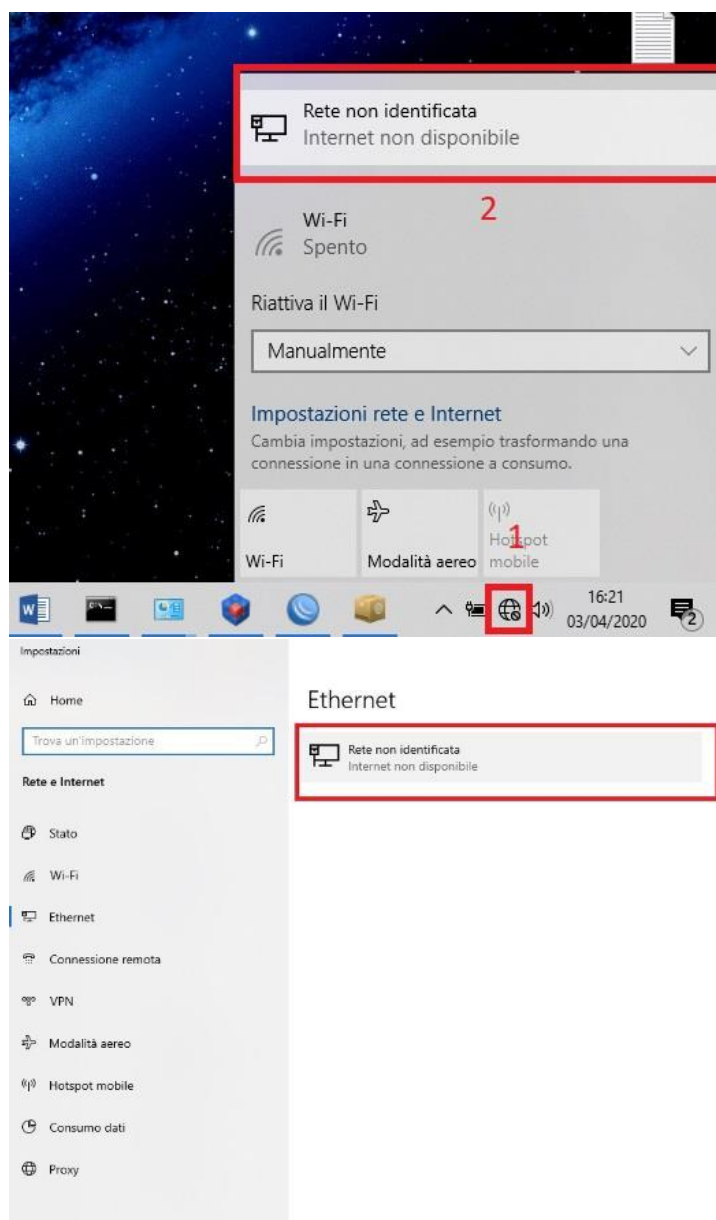


Immagine 7.2 – Ripristino Router: cambio IP della rete point-to-point tra router e computer (1)

🏠 Rete non identificata

Connessione a consumo

Se il tuo piano dati è limitato e vuoi un maggiore controllo sul consumo dei dati, imposta questa connessione come rete a consumo. Alcune app potrebbero funzionare in modo diverso per ridurre il consumo dei dati quando sei connesso a questa rete.

Imposta come connessione a consumo

Disattivato

Se imposti una soglia dei dati, viene configurata automaticamente l'impostazione di connessione a consumo che consente di non superare la soglia.

[Imposta una soglia dei dati per controllare l'utilizzo dei dati nella rete](#)

Impostazioni IP

Assegnazione IP:	Manuale
Indirizzo IPv4:	192.168.1.2
Lunghezza prefisso subnet IPv4:	24
Gateway IPv4:	192.168.1.1

Modifica

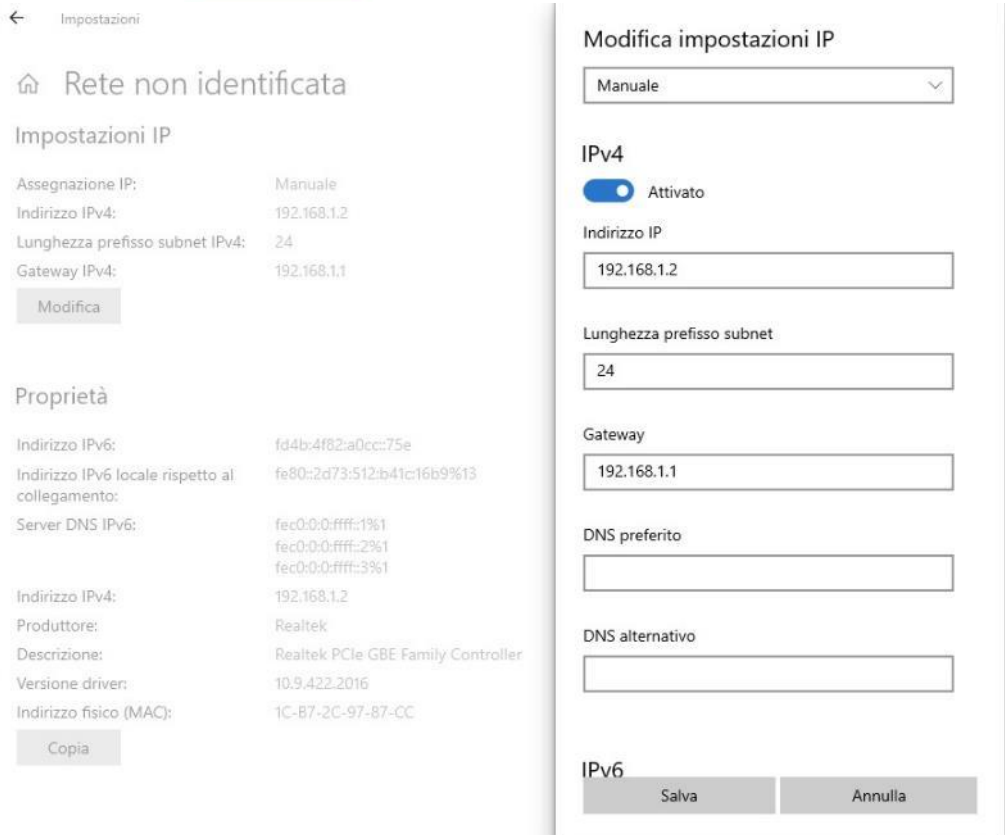


Immagine 7.3 – Ripristino OpenWRT: cambio IP della rete point-to-point tra router e computer (2)

Dopo aver configurato l'indirizzo di rete del PC secondo quanto riportato nelle immagini 7.2 e 7.3, avviare il programma *Netinstall*^[55]

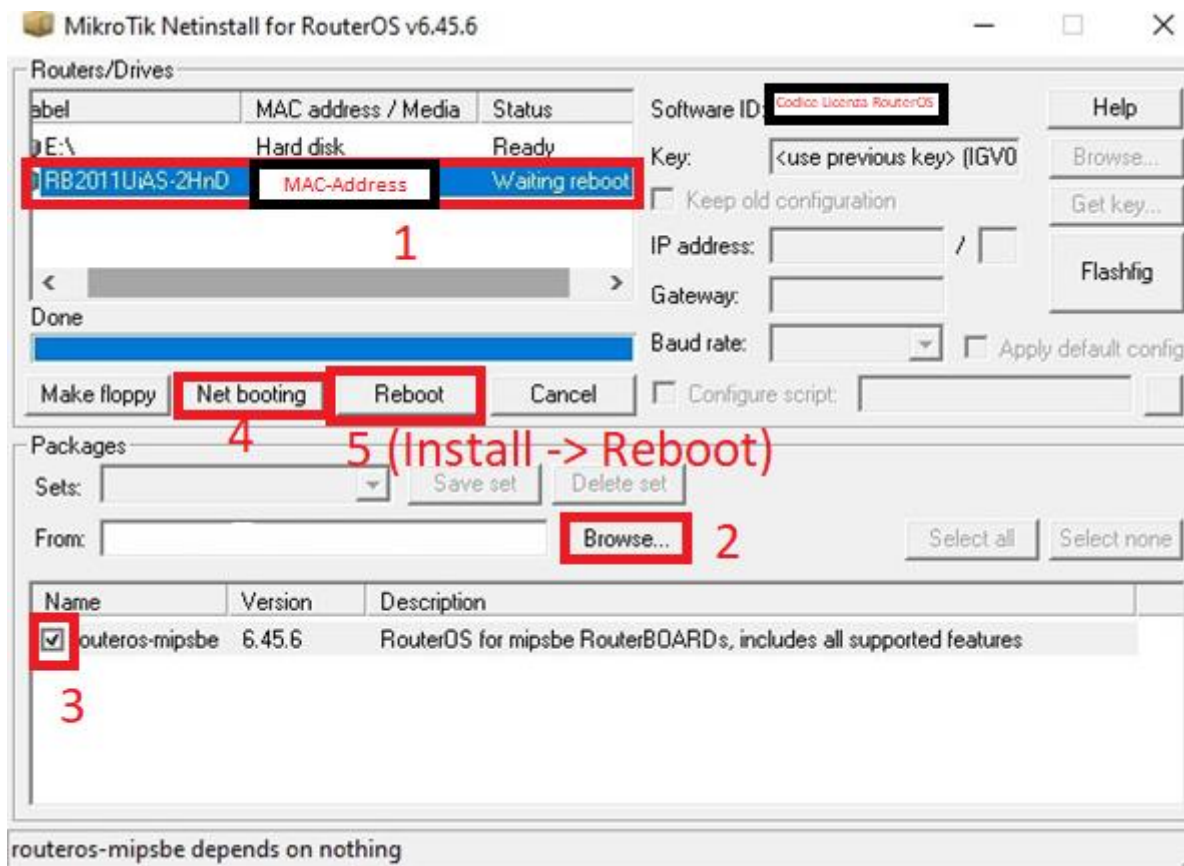


Immagine 7.4 – Ripristino OpenWRT: Netinstall

Una volta aperto, *Netinstall* si presenterà come da Immagine 7.4.

A questo punto bisogna procedere come segue:

- Connettere un cavo ethernet al router sulla porta ETH1;
- Connettere l'altra estremità del cavo ethernet alla scheda di rete del PC;
- Tenere premuto il tasto RESET posto sul lato posteriore del router;
- Dare alimentazione a quest'ultimo **senza** rilasciare il pulsante RESET, al fine di far entrare il dispositivo in "Failsafe mode / Etherboot" (si potrà essere certi di aver fatto avviare con successo il dispositivo in tale modalità una volta che sarà apparsa la scritta "etherboot" sul display dello stesso e sarà possibile rilasciare il tasto RESET nel momento in cui il display riporti tale scritta)
- NOTA BENE: se Netinstall avrà riconosciuto il router, ne verrà visualizzato il nome e il relativo MAC Address, come nell'Immagine 7.4; si vedrà altresì scomparire la scritta "etherboot" dal display del dispositivo;
- Si potrà allora procedere alla scelta del file di RouterOS da flashare tramite la selezione della cartella contenente quest'ultimo, attraverso il pulsante "Browse" ubicato nella parte centrale di Netinstall;
- Selezionata la cartella, vengono visualizzati tutti i file che è possibile flashare;

- Premendo su *NetInstall* il pulsante “Net Booting” si dovrà specificare l’indirizzo del router (mettere la spunta come mostrato nella parte bassa dell’Immagine 7.4);
- Premere Install e attendere finché la barra di progresso che apparirà non si sarà riempita e il pulsante Install non sia cambiato in Reboot;
- Dopo aver cliccato su Reboot, il router emetterà un primo “beep”, seguito da un secondo “beep” se tutto sarà andato a buon fine;
- Si dovrà in conclusione impostare l’IP address della porta ETH1 del router per poter ricreare le condizioni iniziali che consentiranno di procedere all’installazione di OpenWRT. La configurazione dell’IP per l’interfaccia ETH1 va effettuata manualmente attraverso il display touchscreen del dispositivo come da Immagini 7.5, 7.6, 7.7 .

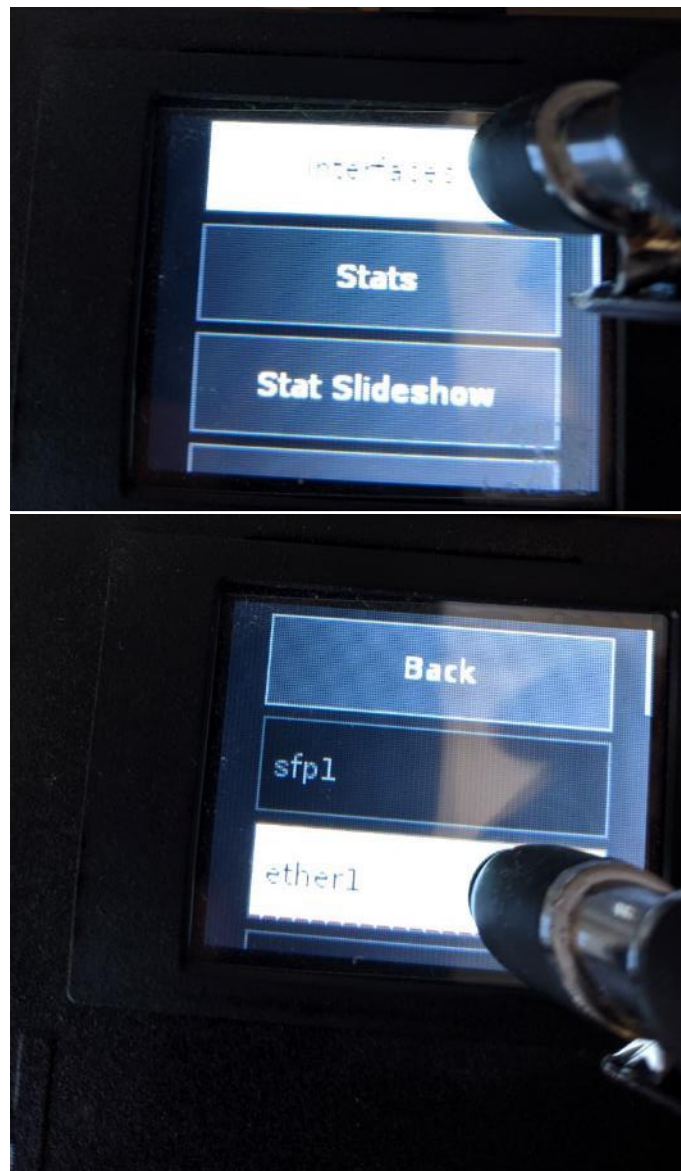


Immagine 7.5– Ripristino Router: assegnazione IP all’interfaccia ETH1 del router (1)



Immagine 7.6– Ripristino Router: assegnazione IP all'interfaccia ETH1 del router (2)



Immagine 7.7– Ripristino Router: assegnazione IP all'interfaccia ETH1 del router (3)

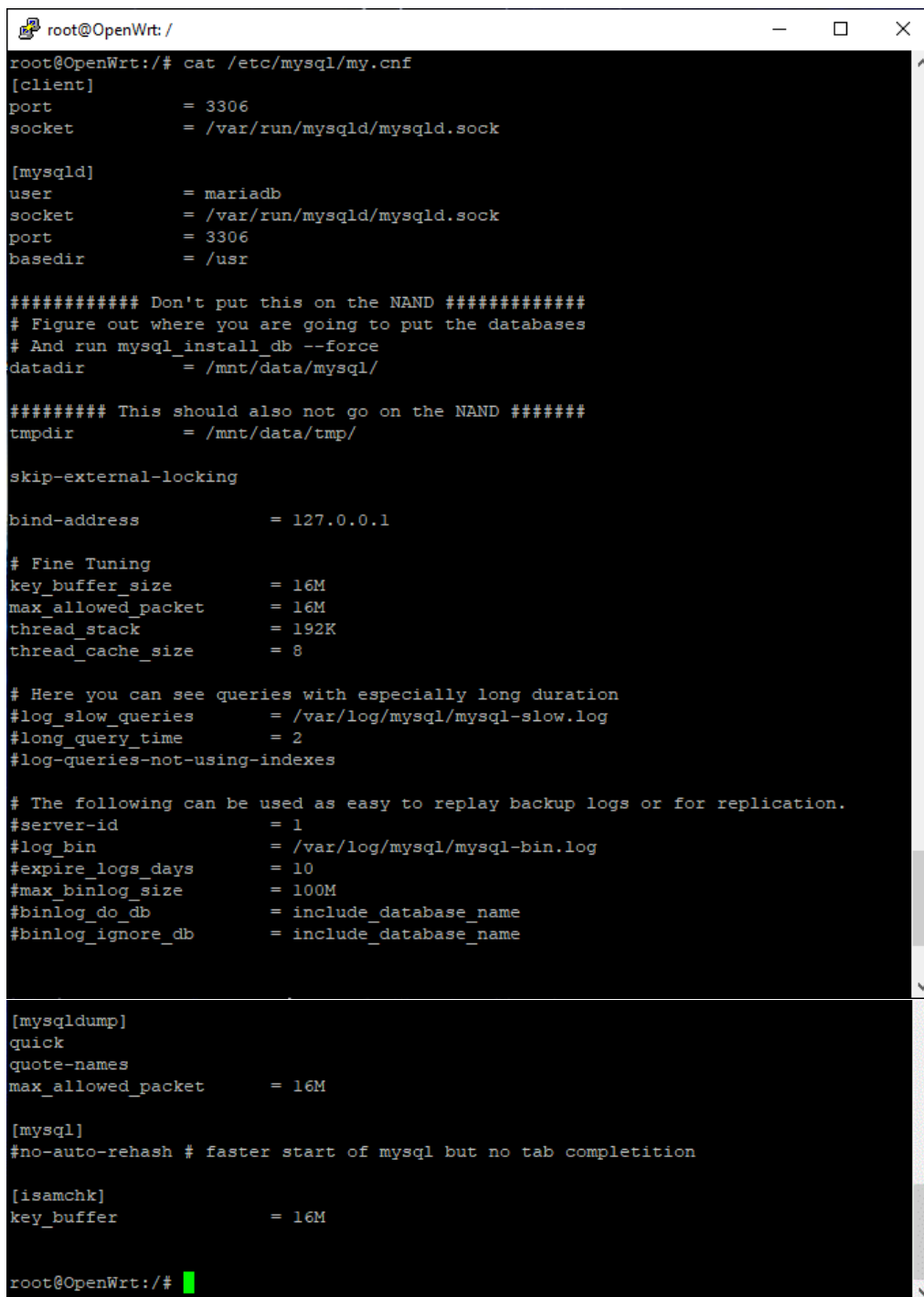
CONCLUSIONI

Il lavoro svolto porta a evidenziare come sia possibile implementare uno stack web completo comprensivo di istanze client e server MySQL/MariaDB su un apparato di rete di medie prestazioni, con il fine di predisporre quest'ultimo alla gestione di funzionalità "web-related" che necessitino di capacità di elaborazione dati tramite DBMS.

Ciò dimostra che nella progettazione e dimensionamento di una rete LAN di elaboratori si possa prendere in considerazione la possibilità di avvalersi di strumenti tecnologici che, seppur di prestazioni non da "top di gamma", garantiscano tanto la fruizione dei servizi per i quali vengono disposti, come il rispetto della non prescindibilità di aspetti chiave quali sostenibilità economica e affidabilità.

APPROFONDIMENTO: Parametri essenziali di configurazione in *my.cnf*

Prima di passare all'inizializzazione e contestuale esecuzione del server MariaDB vera e propria, dopo aver installato il package a esso relativo, bisogna modificare il contenuto del file da cui questo andrà a leggere i parametri di esecuzione, cioè */etc/mysql/my.cnf*, il quale si presenta, di default, come nell'Immagine 6.1:



```
root@OpenWrt: /
root@OpenWrt: /# cat /etc/mysql/my.cnf
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

[mysqld]
user                = mariadb
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr

##### Don't put this on the NAND #####
# Figure out where you are going to put the databases
# And run mysql_install_db --force
datadir             = /mnt/data/mysql/

##### This should also not go on the NAND #####
tmpdir              = /mnt/data/tmp/

skip-external-locking

bind-address        = 127.0.0.1

# Fine Tuning
key_buffer_size     = 16M
max_allowed_packet  = 16M
thread_stack        = 192K
thread_cache_size   = 8

# Here you can see queries with especially long duration
#log_slow_queries   = /var/log/mysql/mysql-slow.log
#long_query_time    = 2
#log-queries-not-using-indexes

# The following can be used as easy to replay backup logs or for replication.
#server-id          = 1
#log_bin            = /var/log/mysql/mysql-bin.log
#expire_logs_days   = 10
#max_binlog_size    = 100M
#binlog_do_db       = include_database_name
#binlog_ignore_db   = include_database_name

[mysqldump]
quick
quote-names
max_allowed_packet  = 16M

[mysql]
#no-auto-rehash # faster start of mysql but no tab completion

[isamchk]
key_buffer          = 16M

root@OpenWrt: /#
```

Immagine 6.1 – Il contenuto originale di *my.cnf*

Tale file è ampiamente configurabile e i parametri in esso già presenti al momento dell'installazione del package del server MariaDB possono variare in dipendenza della piattaforma hardware e software su cui esso viene installato.

È data la possibilità all'utente di modificare i valori dei parametri presenti, rimuoverli e/o aggiungerne di altri se necessario, con il fine non solo di ottenere una corretta inizializzazione del servizio, ma anche di una sua esecuzione ottimizzata in termini di utilizzo di risorse hardware, con particolare riferimento al supporto di archiviazione dati e di memoria centrale.

Come visto alla [pagina 45](#), prima di avviare l'istanza del servizio MariaDB, è necessario utilizzare il comando `mysql_install_db -force`, con cui si inizializza la directory dei dati di MariaDB creando in essa le tabelle di sistema e di help, se non già esistenti, seguendo rispettivamente il path per i dati e quello per i dati temporanei riportati all'interno di `my.cnf`.

Ponendo poi in esecuzione il servizio con `/etc/init.d/mysqld start`, si crea in RAM un'istanza del servizio MySQL/MariaDB rispettando i valori assegnati ai moduli di esecuzione del server riportati all'interno del medesimo file, `my.cnf`.

In quest'ultima fase potrebbe sorgere un problema qualora il sistema (su piattaforma fisica o implementato su macchina virtuale) disponga di poca capacità di memoria RAM (ossia minore di 512MB): se si lasciano immutati i valori relativi ai moduli di funzionamento che dovranno essere allocati nella memoria centrale da MariaDB, il sistema non riuscirà a portare a termine la procedura d'inizializzazione ed esecuzione dell'istanza del server MariaDB, presentando in output, in fase di inizializzazione, l'errore `mysqld: out-of-memory`, come mostrato nell'Immagine 6.2

```
root@OpenWrt: /
root@OpenWrt:~# mysql_install_db --force
Installing MariaDB/MySQL system tables in '/var/data/mysql/' ...
2020-03-17 11:01:16 2010574092 [Note] /usr/bin/mysqld (mysqld 10.1.41-MariaDB) starting as process 4683 ...
2020-03-17 11:01:16 2010574092 [Note] InnoDB: innodb_empty_free_list_algorithm has been changed to legacy because of small buffer pool size. In order to use backoff, increase buffer pool at least up to 20MB.

2020-03-17 11:01:16 2010574092 [Note] InnoDB: Using mutexes to ref count buffer pool pages
2020-03-17 11:01:16 2010574092 [Note] InnoDB: The InnoDB memory heap is disabled
2020-03-17 11:01:16 2010574092 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2020-03-17 11:01:16 2010574092 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
2020-03-17 11:01:16 2010574092 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-03-17 11:01:16 2010574092 [Note] InnoDB: Using Linux native AIO
2020-03-17 11:01:16 2010574092 [Note] InnoDB: Using generic crc32 instructions
2020-03-17 11:01:17 77d6ed0c InnoDB: Error: Linux Native AIO interface is not supported on this platform. Please check your OS documentation and install appropriate binary of InnoDB.
InnoDB: You can disable Linux Native AIO by setting innodb_use_native_aio = 0 in my.cnf
2020-03-17 11:01:17 2010574092 [Warning] InnoDB: Linux Native AIO disabled.
2020-03-17 11:01:17 2010574092 [Note] InnoDB: Initializing buffer pool, size = 128.0M
InnoDB: mmap(138543104 bytes) failed; errno 12
2020-03-17 11:01:17 2010574092 [ERROR] InnoDB: Cannot allocate memory for the buffer pool
2020-03-17 11:01:17 2010574092 [ERROR] Plugin 'InnoDB' init function returned error.
2020-03-17 11:01:17 2010574092 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2020-03-17 11:01:17 2010574092 [ERROR] mysqld: Out of memory (Needed 130760704 bytes)
2020-03-17 11:01:17 2010574092 [ERROR] mysqld: Out of memory (Needed 98058240 bytes)
2020-03-17 11:01:17 2010574092 [ERROR] Unknown/unsupported storage engine: InnoDB
2020-03-17 11:01:17 2010574092 [ERROR] Aborting

Installation of system tables failed! Examine the logs in /var/data/mysql/ for more information.

The problem could be conflicting information in an external my.cnf files. You can ignore these by doing:

    shell> /usr/bin/mysql_install_db --defaults-file=~/.my.cnf

You can also try to start the mysqld daemon with:

    shell> /usr/bin/mysqld --skip-grant-tables --general-log &

and use the command line tool /usr/bin/mysql to connect to the mysql database and look at the grant tables:

    shell> /usr/bin/mysql -u root mysql
mysql> show tables;

Try 'mysqld --help' if you have problems with paths. Using --general-log gives you a log in /var/data/mysql/ that may be helpful.

The latest information about mysql_install_db is available at https://mariadb.com/kb/en/installing-system-tables-mysql_install_db You can find the latest source at https://downloads.mariadb.org and the maria-discuss email list at https://launchpad.net/~maria-discuss

Please check all of the above before submitting a bug report at http://mariadb.org/jira
```

Immagine 6.2 – L'errore *mysqld: out-of-memory*

Per evitare che si presenti l'errore citato (e altri relativi a un'eventuale scarsità nella capacità di archiviazione di massa), la modifica chiave da apportare al file di configurazione *my.cnf* consiste nel cambiare il suo contenuto con il fine di assegnare:

- un valore ai parametri che rappresentano rispettivamente il path per il deposito dei file dati e quello per il deposito dei file temporanei utili alla corretta esecuzione del servizio;
- i valori (espressi in KiloByte e/o MegaByte) per i blocchi contigui di memoria RAM riservati ai moduli di funzionamento di cui il server farà uso per l'elaborazione delle richieste SQL da parte dei client che vi si conetteranno.

In particolare riguardo agli ultimi segue una breve trattazione.

Moduli di funzionamento allocati in memoria centrale da MariaDB

Di questi ne esiste un numero considerevole^[56], per cui si illustreranno solo i moduli toccati dalle modifiche al file *my.cnf* come descritto alla [pagina 43](#) della presente Tesi, per i quali alcuni dei valori assegnati rispecchiano quelli minimi consentiti per la corretta esecuzione di MariaDB^[57].

Nota: per la presente trattazione sono stati interessati da modifica principalmente alcuni dei parametri relativi ai moduli *performance_schema* e *InnoDB*, come da indicazione generale fornita da varie fonti online, tra le quali le pagine ufficiali di aziende specializzate nel settore dell'implementazione, gestione e recovery di database, nonché dagli stessi sviluppatori degli Storage Engine citati.

Performance_Schema^{[58][59]}

Il Performance_Schema è una funzionalità che serve a monitorare le performance del server, introdotta in MariaDB 5.5.3 . È implementato come Storage Engine, pertanto compare nella lista degli Storage Engine disponibili. Tuttavia PERFORMANCE_SCHEMA non è un normale Storage Engine per registrare i dati, ma è un meccanismo per implementare la funzionalità Performance Schema. Lo Storage Engine contiene un database chiamato performance_schema, che a sua volta contiene un certo numero di tabelle che possono essere interrogate con normali istruzioni SQL, le quali restituiscono informazioni specifiche riguardanti le performance.

Di default^[60] performance_schema occupa 400MB di RAM, per cui è raccomandato disabilitarlo in caso si abbia scarsità di memoria RAM, aggiungendo a *my.cnf* la voce ***performance_schema = off*** .

Il monitoraggio delle performance del database realizzato da questo modulo può essere comunque ottenuto adoperando altri strumenti o seguendo una guida in cui venga indicato il giusto settaggio dei parametri per l'occupazione di RAM da parte delle varie componenti di performance_schema.

Inno_DB^[61]

È lo Storage Engine di default di MariaDB. Di default^[62] Inno_DB, se non specificato altrimenti all'interno del file *my.cnf*, occupa 128MB di RAM nella totalità dei suoi moduli. In generale, la memoria utilizzata è legata alle dimensioni dei moduli di *buffer pool*, *read*, *sort* e *join buffers*, a quelle delle *temporary tables* e all'ammontare delle dimensioni delle *max connections*. Inoltre^[63], i parametri che più spesso risultano causare l'errore di "out-of-memory" per MariaDB sono soprattutto alcuni di quelli legati a InnoDB, dei quali principalmente:

innodb_buffer_pool_size
key_buffer_size
read_buffer_size
read_rnd_buffer_size
sort_buffer_size
join_buffer_size

Una formula approssimata basata sui moduli appartenenti all'elenco di cui sopra, il cui risultato restituisce l'occupazione totale di memoria centrale da parte di tali moduli è data da:

$$\begin{aligned} & \text{Maximum MySQL Memory Usage} \\ & = \\ & \text{innodb_buffer_pool_size} + \text{key_buffer_size} + \\ & + (\text{read_buffer_size} + \text{read_rnd_buffer_size} + \text{sort_buffer_size} + \text{join_buffer_size}) \times \\ & \quad \times (\text{max_connections}) \end{aligned}$$

In considerazione di quanto detto e in base ai vari messaggi di Warning e di Errore registrati in output durante la sperimentazione in fase di inizializzazione di MariaDB, i parametri relativi a Inno_DB interessati da modifica in *my.cnf* nella presente Tesi sono stati:

innodb_buffer_pool_size
innodb_buffer_log_size
innodb_use_native_aio = 0
key_buffer_size

Il parametro innodb_buffer_pool_size^[64]

Si fa riferimento al parametro *innodb_buffer_pool_size* come la dimensione di RAM da dedicare per la memorizzazione di indici, cache, strutture dati e tutto ciò che ruota attorno a InnoDB. È uno dei parametri più importanti della configurazione di MySQL e il suo valore va impostato in funzione del quantitativo di memoria RAM disponibile e dei servizi che operano sul server^[65]. Al momento dell'esecuzione del comando *mysql_install_db - -force*, MariaDB potrebbe richiedere di porre un valore minimo per il parametro in questione, per cui in *my.cnf* la modifica da apportare è l'aggiunta della stringa relativa, non già presente.

Per la sperimentazione di cui al corpo della presente Tesi, il valore minimo richiesto è stato di 25MB.

innodb_buffer_pool_size = 25M

Il parametro `innodb_buffer_log_size`^[66]

Indica la lunghezza in Bytes del buffer che InnoDB utilizza per scrivere sui file di log nella periferica di archiviazione al path indicato in *my.cnf*. Se non già presente di default, in *my.cnf* la modifica da apportare è l'aggiunta della stringa a esso relativa con un valore minimo consentito di 256 KByte

```
innodb_log_buffer_size = 256K
```

Il parametro `innodb_use_native_aio`^[67]

Con esso si specifica se utilizzare il “Linux asynchronous I/O subsystem”. Durante la sperimentazione, i messaggi di Warning in fase di inizializzazione del server con il comando *mysql_install_db --force* richiedessero di aggiungere in *my.cnf* la stringa a esso relativo col fine di disabilitarlo, assegnandogli un valore pari a 0 (o alternativamente OFF).

```
innodb_use_native_aio = 0
```

Il parametro `key_buffer_size`^[68]

È una variabile di MyISAM che determina la dimensione degli index buffers che risiedono in memoria centrale, i quali concorrono a influenzare la velocità di lettura degli indici delle tabelle dei database. In *my.cnf* la modifica da apportare è l'assegnazione del valore minimo di 8MB (o 64KB se non si fa uso di MyISAM/Aria, ma di InnoDB come Storage Engine)

```
key_buffer_size=8M (o, se non si fa uso di MyISAM/Aria, key_buffer_size=64K)
```

RINGRAZIAMENTI

È stato possibile raggiungere l'obiettivo di realizzazione descritto nella presente Tesi di Laurea grazie alle risorse reperibili online della comunità di sviluppatori e utilizzatori del sistema operativo open source su base Linux per apparati di rete "OpenWRT" e altri sistemi simili e/o affini.

Il presente lavoro di Tesi è frutto di un intero anno di ricerca e sperimentazione per cui ringrazio il Relatore prof. Ennio Gambi e il Correlatore prof. Adelmo De Santis per il supporto tempestivo e oltre modo accomodante fornitomi, nonché per aver continuato ad alimentare in me la già presente passione per le tecnologie per le telecomunicazioni.

Questo lavoro rappresenta inoltre la conclusione di un ciclo di studi per il quale ringrazio l'UNIVPM, che mi ha dato non solo la possibilità di formarmi culturalmente e personalmente, ma anche di aderire a esperienze degne di nota quali l'Erasmus+ Studio e la competizione europea Huawei, oltre a farmi vivere momenti di grande soddisfazione insieme a tutte le persone che hanno fatto parte di questo percorso e che ringrazio per aver contribuito concretamente a raggiungere i miei obiettivi di studio con i loro consigli e il loro sostegno, quali la mia famiglia, gli amici di sempre, i vari compagni e colleghi che ho avuto la fortuna di conoscere nel corso del tempo.

BIBLIOGRAFIA

- ¹ Portale web di OpenWRT: <https://openwrt.org>
- ² Lista dei dispositivi supportati da OpenWRT: https://openwrt.org/toh/views/toh_fwdownload
- ³ Pagina di supporto del costruttore del router scelto: <https://mikrotik.com/product/RB2011UiAS-2HnD-IN>
- ⁴ Guida a come impostare una rete con indirizzi statici in Windows 10 <https://turbolab.it/reti-1448/guida-windows-10-come-configurare-manualmente-indirizzo-ip-windows-866>
- ⁵ Download di RouterOS <https://mikrotik.com/download>
- ⁶ Pagina di download di software Mikrotik <https://mikrotik.com/download>
- ⁷ DHCP: <https://www.geeksforgeeks.org/dynamic-host-configuration-protocol-dhcp/>
- ⁸ BOOTP: <https://www.ionos.it/digitalguide/server/know-how/protocollo-bootstrap-bootp/>
- ⁹ TFTP: https://it.wikipedia.org/wiki/Trivial_File_Transfer_Protocol
- ¹⁰ Guida per l'installazione di OpenWRT: <https://openwrt.org/toh/mikrotik/common>
- ¹¹ Download di Tiny PXE: <http://reboot.pro/files/file/303-tiny-pxe-server/>
- ¹² <http://alduras.com/wp/category/openwrt/>
- ¹³ Guida ufficiale di OpenWRT per l'implementazione dello stack LAMP: <https://openwrt.org/docs/guide-user/services/webserver/lamp>
- ¹⁴ Download di PuTTY: <https://www.putty.org/>
- ¹⁵ Guida di OpenWRT a uHTTPd: <https://openwrt.org/docs/guide-user/services/webserver/uhttpd>
- ¹⁶ uHTTPd è basato sullo stack LLMP: <http://webuzo.com/llmp>
- ¹⁷ Informazioni sulla direttiva uci: <https://openwrt.org/docs/guide-user/base-system/uci>, https://wiki.teltonika.lt/view/UCI_command_usage
- ¹⁸ Convenzioni sull'utilizzo della cartella /srv in Linux: <https://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/srv.html>

- ¹⁹ Per il comando `mkdir -p` : <https://www.javatpoint.com/linux-mkdir-p>
- ²⁰ Comandi presi da: <https://openwrt.org/docs/guide-user/services/webserver/php>
- ²¹ Per maggiori informazioni riguardo il modulo `php-cgi`:
<https://help.superhosting.bg/en/cgi-common-gateway-interface-fastcgi.html>
- ²² Il file `php.ini` è il file di configurazione dell'interprete PHP. Per maggiori info sul file `php.ini`:
<https://www.ionos.it/aiuto/hosting/utilizzo-di-php-per-i-progetti-web/modificare-le-impostazioni-php-con-phpini/>
- ²³ `/etc/php.ini` è il file di configurazione di PHP
- ²⁴ Dal contenuto di `php.ini` si può evincere che le estensioni PHP sono collocate al path `/usr/lib/php`.
Riguardo le estensioni PHP:
<https://www.sitepoint.com/install-php-extensions-source/>,
<https://www.quora.com/What-are-PHP-extensions>
- ²⁵ Guida all'uso dell'editor di testo Linux `vi`:
<http://www.coresis.com/extra/linuxcorsobase/6-4.htm>
- ²⁶ Fork in informatica: [https://it.wikipedia.org/wiki/Fork_\(sviluppo_software\)](https://it.wikipedia.org/wiki/Fork_(sviluppo_software))
- ²⁷ Un articolo descrittivo su MariaDB: <https://www.html.it/pag/55928/mariadb/>
- ²⁸ MyISAM: <https://it.wikipedia.org/wiki/MyISAM>
- ²⁹ Aria: <https://mariadb.com/kb/en/aria-storage-engine/>
- ³⁰ Maggiori informazioni sul concetto di transazione nei sistemi di gestione di basi di dati
[https://it.wikipedia.org/wiki/Transazione_\(basi_di_dati\)](https://it.wikipedia.org/wiki/Transazione_(basi_di_dati))
- ³¹ InnoDB: <https://dev.mysql.com/doc/refman/5.6/en/innodb-introduction.html>
- ³² XtraDB:
<https://it.wikipedia.org/wiki/XtraDB>, <https://www.percona.com/doc/percona-xtradb-cluster>
- ³³ Link all'elenco aggiornato dei package per OpenWRT utili alla gestione di database:
<https://openwrt.org/packages/index/libraries---database>
- ³⁴ `libpthread`:
<https://openwrt.org/packages/pkgdata/libpthread>,
https://en.wikipedia.org/wiki/POSIX_Threads

³⁵ libncurses:

https://openwrt.org/packages/pkgdata_owrt18_6/libncurses,
<https://it.wikipedia.org/wiki/Ncurses>

³⁶ libreadline:

https://en.wikipedia.org/wiki/GNU_Readline,
https://openwrt.org/packages/pkgdata_owrt18_6/libreadline

³⁷ mysql-server: https://openwrt.org/packages/pkgdata_lede17_1/mysql-server

³⁸ Il comando df:

<https://www.linux.com/training-tutorials/6how-check-disk-space-linux-command-line/>

³⁹ Lo swap in Linux: <https://www.linux.com/news/all-about-linux-swap-space/>

⁴⁰ Il file-system ext4: <https://it.wikipedia.org/wiki/Ext4>

⁴¹ Un'ottima video guida è disponibile al link:

<https://www.youtube.com/watch?v=3GgRKuVaH5o>

⁴² fdisk: <https://www.geeksforgeeks.org/fdisk-command-in-linux-with-examples/>

⁴³ GParted: <https://gparted.org/>

⁴⁴ I comandi riportati sono stati presi da

<https://openwrt.org/docs/guide-user/storage/usb-drives-quickstart>
<https://openwrt.org/docs/guide-user/storage/usb-drives>

⁴⁵ Linux mounting point: http://www.linfo.org/mount_point.html

⁴⁶ Una guida sulle opzioni contenute in fstab per OpenWRT è disponibile al link

<https://openwrt.org/docs/guide-user/storage/fstab>

⁴⁷ Creazione cartelle da riga di comando di cartelle in Linux:

<https://www.javatpoint.com/linux-mkdir-p>

⁴⁸ Il comando chmod in Linux:

<https://www.ionos.it/digitalguide/server/know-how/lassegnazione-dei-permessi-con-il-comando/>

⁴⁹ Guida all'uso dell'editor di testo Linux vi:

<http://www.coresis.com/extra/linuxcorsobase/6-4.htm>

⁵⁰ I comandi sono stati presi da

<http://www.ibuyopenwrt.com/index.php/8-yun-compatible/140-install-mysql-server>

⁵¹ Per ulteriori switch relativi al comando mysql si guardi:

<https://mariadb.com/kb/en/connecting-to-mariadb/>

⁵² Configurazione utenti in MariaDB:

<https://mariadb.com/kb/it/gestione-degli-account/>,

<https://phoenixnap.com/kb/how-to-create-mariadb-user-grant-privileges>

⁵³ Modifica al file */etc/config/network*:

<https://forum.openwrt.org/t/solved-how-to-correctly-change-router-lans-ip-address/21654>

⁵⁴ Fonti per la procedura di ripristino:

http://wiki.villagetelco.org/OpenWrt_Failsafe_Mode_and_Flash_Recovery,

<https://wiki.mikrotik.com/wiki/Manual:Netinstall>,

<https://blog.ligos.net/2016-12-27/Recover-A-Broken-Mikroik-Device.html>

⁵⁵ Link al download di Netinstall:

<https://download.mikrotik.com/routeros/6.47/netinstall-6.47.zip>

⁵⁶ Lista e spiegazione dei vari moduli allocati in RAM da un'istanza del server MariaDB:

<https://dev.mysql.com/doc/refman/8.0/en/memory-use.html>,

<https://www.codepedia.org/ama/optimizing-mysql-server-settings/>

⁵⁷ Guida ai valori minimi consentiti per la corretta esecuzione di MariaDB:

<https://stackoverflow.com/questions/43466124/mysql-out-of-memory-mariadb>

⁵⁸ Guida ufficiale di MariaDB riguardante la voce su `performance_schema`:

<https://mariadb.com/kb/it/performance-schema-overview/>

⁵⁹ Articolo descrittivo e di utilizzo di `performance_schema`, nonché dell'entità del suo impatto sulle performance del server MariaDB:

<https://engineering.linecorp.com/en/blog/mysql-research-performance-schema-instruments/>

⁶⁰ Valore di default di occupazione di memoria centrale di `performance_schema`:

<https://mariadb.com/resources/blog/starting-mysql-on-low-memory-virtual-machines/>

⁶¹ Pagina concernente i moduli relativi a `InnoDB` che vengono implementati in fase di startup sono disponibili al link <https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html>

⁶² Valore di default di occupazione di memoria centrale di `InnoDB`:

<https://mariadb.com/resources/blog/starting-mysql-on-low-memory-virtual-machines/>

⁶³ Articolo sull'errore `mysqld out-of-memory`:

<https://bobcares.com/blog/mysqld-out-of-memory/>

⁶⁴ Pagina di MariaDB sul parametro `innodb_buffer_pool`:

<https://mariadb.com/kb/en/innodb-buffer-pool/>

⁶⁵ Descrizione e calcolo percentuale della dimensione di `innodb_buffer_pool_size` rispetto alla RAM disponibile:
<https://www.speedywp.it/ottimizzazione-mysql-come-scegliere-il-valore-di-innodb-buffer-pool/>

⁶⁶ Pagina di MySQL su `innodb_log_buffer_size`:
https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_log_buffer_size

⁶⁷ Pagina di MySQL su `innodb_log_buffer_size`:
https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_use_native_aio

⁶⁸ Pagina di MariaDB su `key_buffer_size`:
https://mariadb.com/kb/en/optimizing-key_buffer_size/