

# **RELAZIONE LABORATORIO VIRTUALE**

Milani Francesco

5 AI

A.S. 2019/2020

# RELAZIONE DI SISTEMI E RETI

## INDICE

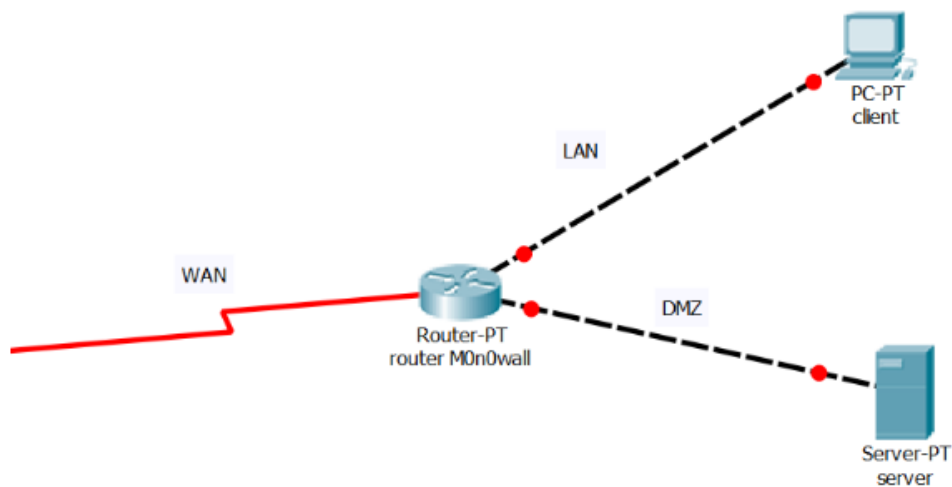
1. Scopo dell'esperienza	4
2. Creazione del client	5
2.1 Note sulla creazione del client	6
3. Installazione Debian	7
3.1 Impostazione utenti e password	
3.2 Configurazione orologio	9
3.3 Partizionamento dei dischi	9
3.4 Installazione sistema base	9
3.5 Driver da includere	9
3.6 Configuratore gestore pacchetti	9
3.7 Selezione e installazione del software	9
3.8 Installazione bootloader GRUB su disco fisso	9
3.9 Note sull'installazione di Debian	10
4. Configurazione del client	11
4.1 Installazione software aggiuntivo	11
4.2 Aggiunta uds al gruppo sudo	11
4.3 Comandi utili	11
4.4 Installazione GUI del client	12
4.5 Note sulla configurazione del client	13
5. Creazione e configurazione del server	14
5.1 Note sulla creazione e configurazione del server	14
6. Creazione e configurazione del router	15
6.1 Impostazioni schede di rete del router	15
6.3 Note sulla creazione e configurazione del router	16
7. Installazione e configurazione di m0n0wall	17
7.1 Ridenominazione schede di rete del router	17
8. Impostazioni di rete del client	18
9. Configurazione m0n0wall lato client	19
10. Configurazione m0n0wall dall'host	20
10.1 Configurazione interfacce	20
10.2 Configurazione regole DMZ	21
11. Configurazione degli Aliases	22

12. Migrazione indirizzi IP	23
12.1 Migrazione indirizzo IP client	23
12.2 Migrazione indirizzo IP server	24
13. Configurazione regole di firewall	25
13.1 Configurazione firewall: LAN	25
13.2 Configurazione firewall: WAN	26
13.3 Configurazione firewall: DMZ	27
14. Configurazione regole del NAT	28
15. Installazione di apache	29
16. Abilitazione certificato SSL	30
16.1 Inserimento regola di NAT	32
16.2 Creazione di un certificato autofirmato	32
17. Installazione e configurazione di una VPN	34
17.1 Configurazione IPsec	35
17.2 Configurazione OpenVPN	36
18. Installazione e configurazione SNMP e MRTG	38
18.1 Configurazione SNMP	38
18.2 Configurazione MRTG	40
19. Installazione e configurazione Cacti	43
19.1 Installazione sul server	43
19.2 Configurazione Cacti	45
20. RAID	47
20.1 Creazione macchina virtuale in RAID 1 con due dischi	48
20.2 Installazione Debian	49
20.3 Partizionamento dei dischi	49
20.4 Configurazione del RAID software	50
20.5 Controllo funzionamento del RAID	52
21. Simulazione della rottura di un disco in RAID 1	53
22. Configurazione di un RAID 5	55
22.1 Partizionamento del nuovo disco	56
22.2 Aggiunta del disco ai device MD	57

## 1. Scopo dell'esperienza

Lo scopo di questa esperienza è quello di riuscire a creare un laboratorio virtuale mediante l'uso di VirtualBox, gestendo 3 diverse reti: LAN, WAN e DMZ, e configurando correttamente tutti i loro componenti.

Per simulare client e server abbiamo dovuto utilizzare le macchine virtuali non per carenza fisica di dispositivi, ma bensì per ragioni di comodità e semplicità.



# VirtualBox

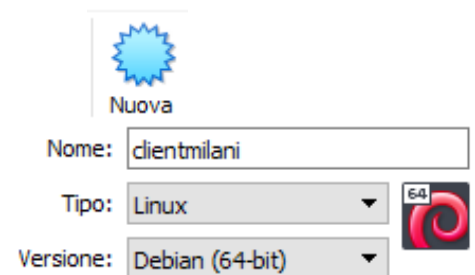


# debian

## 2. Creazione del client

- Aprire VirtualBox, e selezionare l'icona "Nuova"

- Nome: *clientmilani*
- Tipo: *Linux*
- Versione: *Debian 64 Bit*
- Dimensione memoria RAM: *1024 MB*

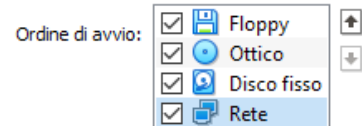


- Creare un nuovo disco fisso
  - Tipo di file: *VDI (VirtualBox Disk Image)*
  - Archiviazione: *Allocato dinamicamente*
  - Nome: *clientmilani.vdi*
  - Dimensione: *4GB*

☒ Crea subito un nuovo disco fisso virtuale

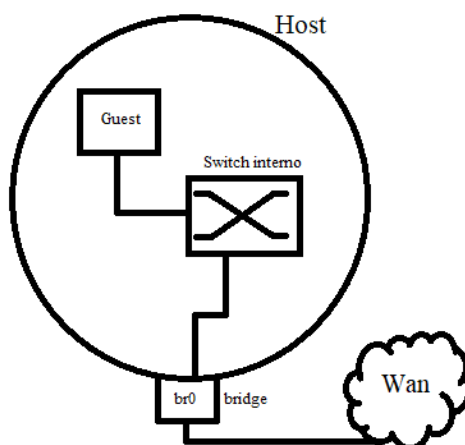
- Impostazioni del client

- Sistema → Ordine di avvio → spuntare *Rete*
- Rete:
  - Connessa a : *Scheda con Bridge*



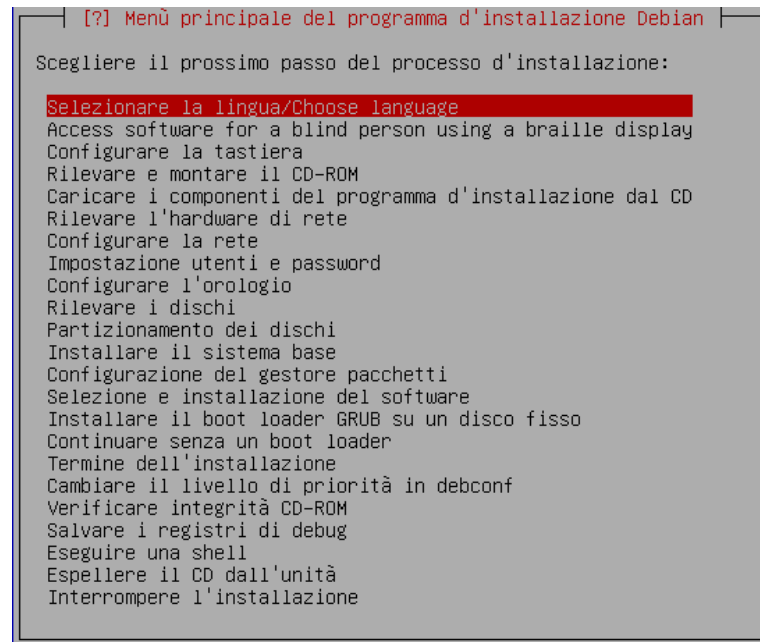
## 2.1 Note sulla creazione del client

- Il disco fisso viene allocato dinamicamente in quanto l'allocazione dinamica è meno prestante rispetto all'allocazione statica, ma è più utile per il nostro utilizzo.
- Connettiamo il computer ad una scheda con bridge, ovvero al posto del router verrà creato uno switch virtuale interno all'host. La scheda di rete quindi accetterà le proprie trame, insieme a quelle di broadcast e multicast, e si potrà istruire per ricevere MAC address specifici.

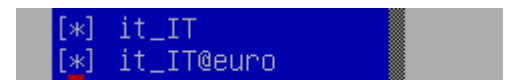


- Come indirizzo MAC verrà utilizzato un indirizzo generico generato automaticamente, che non corrisponde a nessun produttore, in modo da poterlo usare liberamente senza problemi.

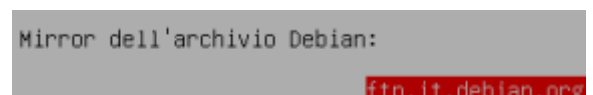
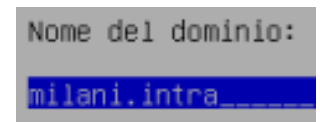
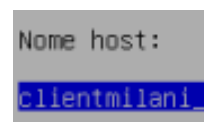
### 3. Installazione Debian



- Lingua: *Italiano*
  - Configurare i locale: *it\_IT*, *it\_IT@euro*
  - Locale prefinito: *UTF-8*



- Tastiera: *Italiano*
- Rilevare l'hardware di rete
  - Configurazione automatica
    - Nome host: *clientmilani*
    - Nome dominio: *milani.intra*
    - Mirror: *http*
    - Nazione del mirror dell'archivio Debian: *Italia*
    - Mirror dell'archivio Debian: *ftp.it.debian.org*



- Informazioni del proxy http: *http://apt-cacher.fermi.intra:3142*
- Versione Debian: *Stable – buster*

### 3.1 Impostazione utenti e password

- Shadow password: *Sì*
- Permettere il root: *Sì*
- Password: *lasolita*
- Account normale:
  - Nome: *utente di servizio*
  - Nome utente: *uds*
  - Password: *lasolita*

Abilitare le «shadow password»?

**<Sì>**    <No>

Permettere l'accesso a root?

**<Sì>**    <No>

### 3.2 Configurazione orologio

- NTP: *Sì*
- Server: *Sì*
- Fuso orario: *Europe/Rome*

### 3.3 Partizionamento dei dischi

- Manuale
- VBOX Hard Disk
- Tabelle: *msdos*
- Spazio libero 1:
  - Nuova partizione: *4.0GB*
  - Primaria
  - Inizio
  - Usare come: *ext4 con journaling*
  - Punto di mount: *“/”*
  - Opzioni:
    - *discard*
    - *noatime*
  - Etichetta: *linuxroot*
  - Flag avviabile: *disattivato*

SCSI3 (0,0,0) (sda) - 4.3 GB ATA VBOX HARDDISK

**pri/log 4.3 GB SPAZIO LIBERO**

Usare come:	File system ext4 con journaling
Punto di mount:	/
Opzioni di mount:	discard,noatime
Etichetta:	linuxroot
Blocchi riservati:	5%
Utilizzo tipico:	standard
Flag avviabile:	disattivato



```
SCSI3 (0,0,0) (sda) - 4.3 GB ATA VBOX HARDDISK
n° 1 primaria 4.0 GB f ext4
pri/log 294.6 MB SPAZIO LIBERO
```

- Spazio libero 2:
  - Nuova partizione: 294.6 MB (*Tutto lo spazio rimanente*)
  - Primaria
  - Usare come: *area di swap*

```
Usare come:      area di swap
Flag avviabile:  disattivato
```

### 3.4 Installazione sistema base

- *linux-image-amd64*

```
linux-image-4.19.0-6-amd64
linux-image-amd64
```

### 3.5 Driver da includere

- *Generico*

```
generico: include tutti i driver disponibili
mirato: solo i driver necessari a questo sistema
```

### 3.6 Configuratore gestore pacchetti

- Software non libero: *No*
- Contrib: *Sì*
- Repository API: *No*

### 3.7 Selezione e installazione del software

- *Nessun aggiornamento*
- *No raccolta statistiche*

### 3.8 Installazione bootloader GRUB su disco fisso

- Installa
- Bootloader nel master: *Sì* → */dev/sda*
- Installazione GRUB: *No*

```
Inserire il device manualmente
/dev/sda (ata-VBOX_HARDDISK_VB13db82bc-98ffef10)
```

### 3.9 Note sull'installazione di Debian

- Il nome Debian nasce dall'unione del nome del suo fondatore Ian Murdock con quello della sua fidanzata Debra
- Viene utilizzato come dominio un sito .intra in quanto questo tipo di dominio non è ancora vendibile quindi non è possibile sia utilizzato da altri.
- Nella rete della scuola è presente un cacher, ovvero uno spazio di memoria dove vengono memorizzati i pacchetti che sono già stati scaricati, in modo da poterli distribuire nella rete in caso di installazioni multiple, senza appesantire il traffico di download.
- UTC sta per Coordinated Universal Time, ed è il fuso orario di riferimento a partire dal quale sono calcolati tutti gli altri fusi orari del mondo. L' UTC+1 è per noi l'ora invernale, mentre L'UTC+2 è l'ora estiva. Per l'estate nel Regno Unito si utilizza il BST, ovvero il British Summer Time.

## 4. Configurazione del client

- Avviare la macchina
- Accedere come utente di servizio (uds)
- Accedere come root
  - *su -*

```
uds@clientmilani:~$ su -  
Password:  
root@clientmilani:~#
```

### 4.1 Installazione software aggiuntivo:

- *apt install less joe tcpdump mtr-tiny cowsay*
- *apt install sudo*
- *apt clean* → Cancella la cache di installazione

### 4.2 Aggiunta uds al gruppo sudo

- *adduser uds sudo*

- Riavviare la macchina

### 4.3 Comandi utili:

- *id* → Per visualizzare in che utente sono
- *id uds* → Per visualizzare chi è uds
- *pwd* → Print Working Directory
- *df -h* → Visualizzare il File System
- *apt upgrade* → Aggiornamento che scarica il software aggiuntivo
- *apt update* → Scansiona e ricarica l'elenco dei software aggiuntivi
- *apt dist-update* → Aggiorna i pacchetti evitando o alleggerendo le intradipendenze che potrebbero portare al blocco dell'aggiornamento
- *Shutdown -h now* → Spegne il computer

## 4.4 Installazione GUI del client

- Accedere come uds
- Accedere come root

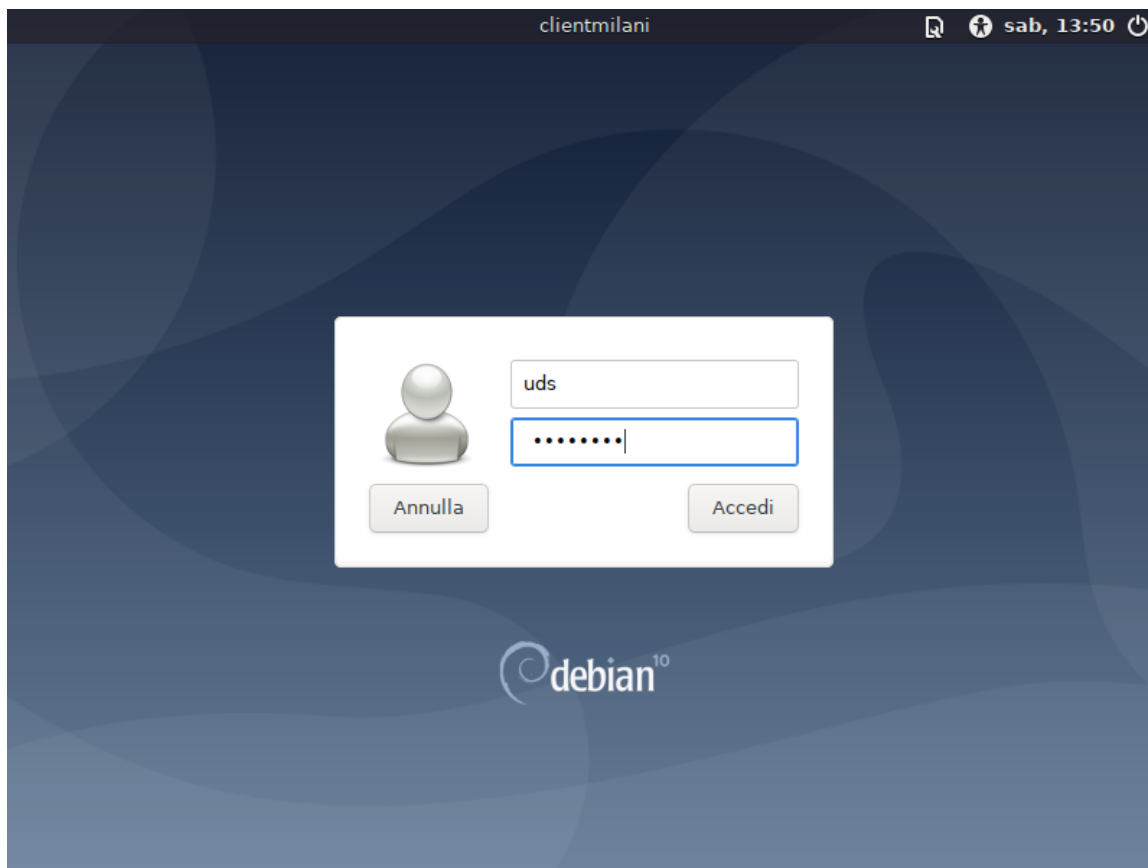
Installazione gestore login, windows manager e firefox

- *apt install light dm mate firefox*
- *apt clean*

Lightdm è il gestore del login grafico, mentre mate è il windows manager

- Riavviare i servizi
  - *cd /etc/init.d/*
  - *./lightdm status*
  - *./lightdm restart*

In questa maniera si riavvierà il gestore grafico, facendo quindi apparire il login grafico.

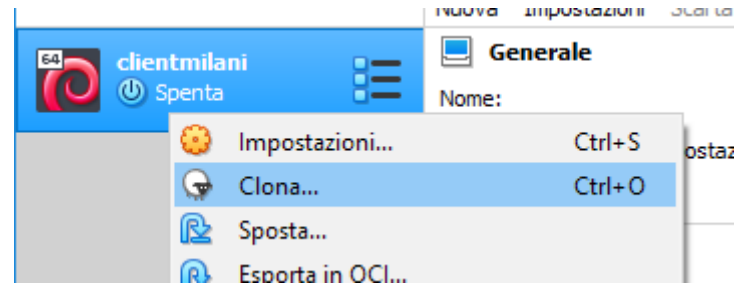


## 4.5 Note sulla configurazione del client

- E' opportuno lasciare inserito il CD di Debian anche dopo l'installazione, in quanto ha al suo interno altri eseguibili per l'installazione di programmi aggiuntivi. Per non avere problemi all'avvio è necessario spostare l'Hard Disk sopra al CD nella sequenza di avvio.
- Le directory sono determinate dal FHS, che sta per Filesystem Hierarchy Standard, ed è lo standard che definisce le directory principali ed il loro contenuto nel file system dei sistemi operativi Unix, tra cui i sistemi Linux.
- La storia tra Debian e Mozilla è controversa, infatti ci sono state vicende legali a causa del fatto che Debian utilizza solamente software libero. Il problema di Firefox stava nel fatto che l'applicazione in sé è libera, ma il logo è registrato, quindi andava contro le politiche di Debian.  
Per anni Debian ha quindi dovuto utilizzare IceWeasel, semplicemente Firefox con nome e logo diverso.  
La controversia si è risolta con la creazione di Firefox esr, che prevede l'assorbimento delle patch di sicurezza.

## 5. Creazione e configurazione del server

- Clonare il client
  - Nome: *servermilani*
  - Inizializzare nuovamente l'indirizzo MAC
  - Tipo: *Completa*



- Avviare il client
- Accedere come utente di servizio (uds)
- Accedere come root
  - `su -`
- Modificare il nome della macchina
  - `joe /etc/hosts` `127.0.1.1 servermilani.milani.intra servermilani`
    - `client.milani.intra` → `server.milani.intra`
    - `clientmilani` → `servermilani`
- Spegnerla la macchina

### 5.1 Note sulla configurazione del server

- Come icona della clonazione è raffigurata una pecora, in onore di Dolly, il primo mammifero ad essere stato clonato con successo da una cellula somatica.

## 6. Creazione e configurazione del router

- Nuova macchina
  - Nome: *routermilani*
  - Tipo: *BSD*
  - Versione: *FreeBSD (32bit)*
  - Memoria RAM: *128 MB*
  - Disco Fisso
    - VDI
    - Statico
    - Memoria: *64 MB*
- Avviare la macchina
  - File ISO di m0n0wall → */home/itis/Internetfiles/monowall.iso*

### 6.1 Impostazioni schede di rete del router

- Scheda 1:
  - Connessa a : *Scheda con bridge*

Scheda 1 Scheda 2 Scheda 3 Scheda 4

☒ Abilita scheda di rete

Connessa a: Scheda con bridge ▼

Nome: Intel(R) Dual Band Wireless-AC 3165

- Scheda 2:
  - Connessa a: *Rete Interna*
  - Nome: *LAN*

Scheda 1 Scheda 2 Scheda 3 Scheda 4

☒ Abilita scheda di rete

Connessa a: Rete interna ▼

Nome: LAN

- Scheda 3:
  - Connessa a: *Rete Interna*
  - Nome: *DMZ*

Scheda 1 Scheda 2 Scheda 3 Scheda 4

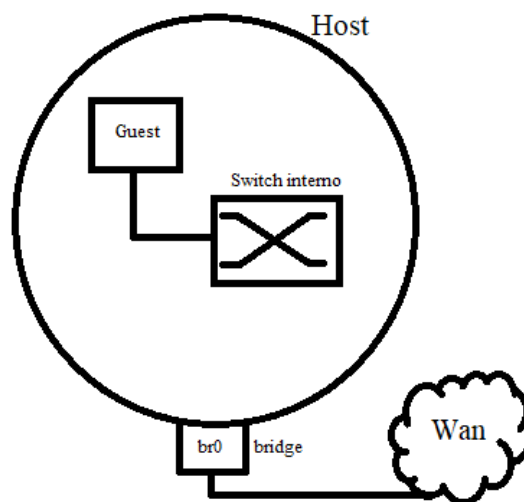
☒ Abilita scheda di rete

Connessa a: Rete interna ▼

Nome: DMZ

## 6.3 Note sulla creazione e configurazione del router

- Nella rete interna viene creato uno switch virtuale, che permetterà la connessione tra i diversi guest, ma non la connessione verso l'esterno, in quanto non connesso.



- m0n0wall è un progetto ormai abbandonato, infatti l'ultima release risale al 15 gennaio 2014.

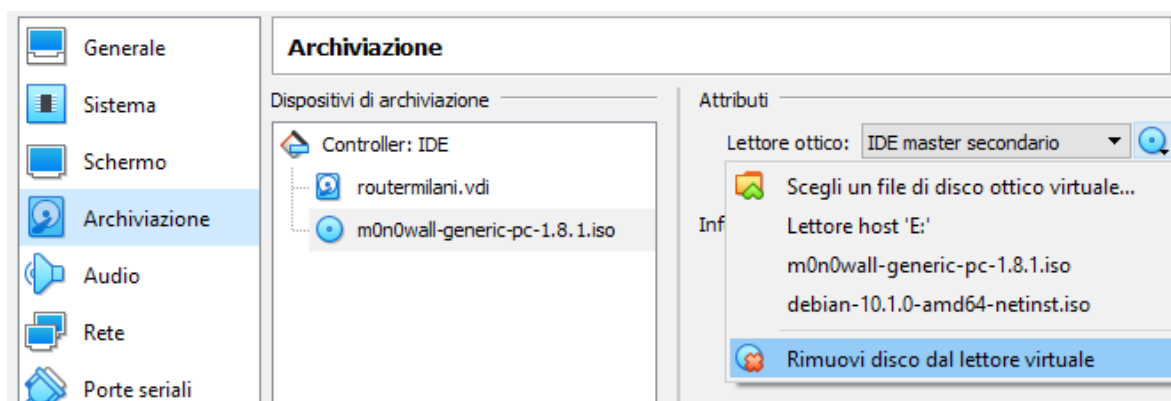


## 7. Installazione e configurazione di m0n0wall

```
m0n0wall console setup
*****
1) Interfaces: assign network ports
2) Set up LAN IP address
3) Reset webGUI password
4) Reset to factory defaults
5) Reboot system
6) Ping host
7) Install on Hard Drive
```

- Selezionare l'opzione 7 → *Install on Hard Drive*
- Spazio di memoria : *ad0*
- Attendere il riavvio

Dalle impostazioni rimuovere il disco di m0n0wall dal lettore virtuale



### 7.1 Ridenominazione schede di rete del router

- Avviare il router
- Selezionare l'opzione 1
  - Set up VLANs: *No*
  - Rinominare le schede nel seguente modo:
    - em0: *WAN*
    - em1: *LAN*
    - em2: *DMZ*

```
LAN -> em1
WAN -> em0
OPT1 -> em2
```

- Riavviare la macchina

## 8. Impostazioni di rete del client

- Scheda 1:
  - Connessa a: *Rete Interna*
  - Nome: *LAN*
- Accedere come uds
- Accedere come root
- Inserire il seguente comando
  - *ip addr* → Per controllare la propria connessione
- Lancio del DHCP manualmente
  - *dhclient enp0s3*

All'avvio di m0n0wall l'interfaccia WAN non presenta inizialmente un indirizzo IP, in quanto la WAN invia una richiesta DHCP a cui risponderà la mia infrastruttura. Basta aggiornare premendo il tasto invio e verrà così visualizzato l'indirizzo IP.

```
LAN IP address: 192.168.1.1  
WAN IP address: (unknown)
```

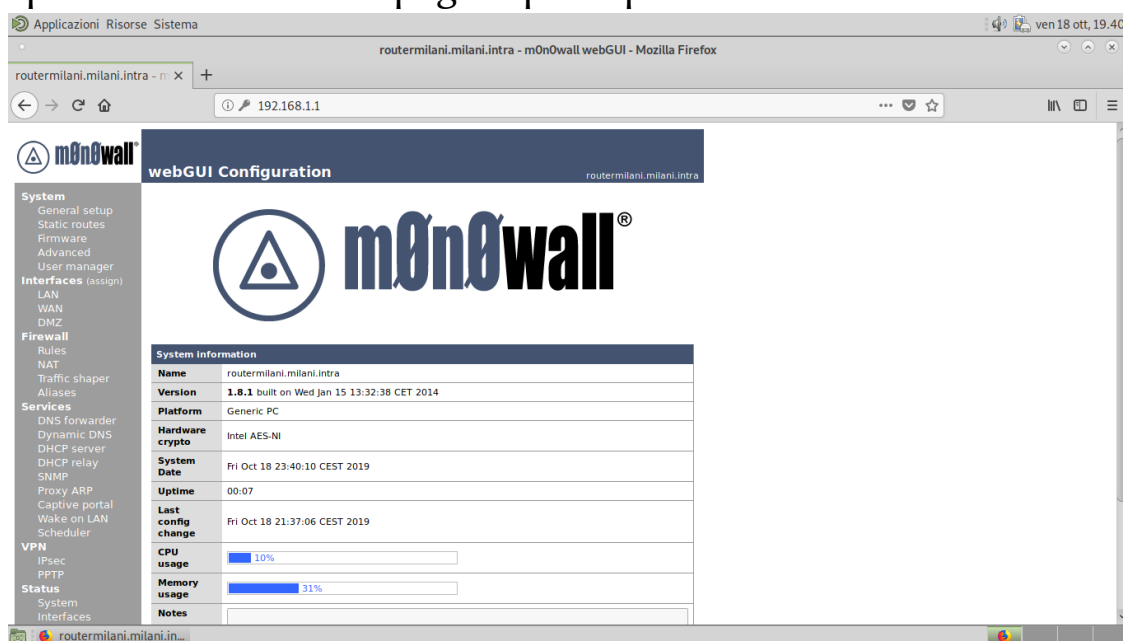


```
LAN IP address: 192.168.1.1  
WAN IP address: 192.168.1.219
```

## 9. Configurazione m0n0wall lato client

- Avviare Firefox
- Connettersi all'indirizzo 192.168.1.1
- Nome utente: *admin*
- Password: *mono*

Sarà quindi visualizzata la pagina principale di m0n0wall



- Firewall
  - Rules
  - Editare la riga presente su WAN
    - Rimuovere la spunta su *Block private networks*
    - Save
  - Creare una nuova regola su WAN
    - Source
      - Type: *Single Host*
      - Address: *172.30.4.1*
    - Destination port range
      - From: *HTTP*
      - To: *WAN*
- Apply changes

## 10. Configurazione m0n0wall dall'host

- Accesso tramite indirizzo WAN
- System
  - General setup
    - Hostname: *routermilani*
    - Domain: *milani.intra*
    - Username: *admin*
    - Password: *lasolita*
    - Time: *Europe/Rome*
- Save

### 10.1 Configurazione interfacce

- Interfaces: *WAN*
  - Hostname: *routermilani*
  - Description: accesso web al m0n0wall dal pc ospitante
- Interfaces: *OPT1*
  - Enable Optional 1 Interface
  - Description: *DMZ*
  - Bridge with: *none*
  - IP address: *192.168.101.1*

## 10.2 Configurazione regole DMZ










- Firewall
  - Rules
    - DMZ
      - Aggiungi nuova regola
      - Action: *Block*
      - Protocol: *Any*
      - Source: *DMZ Subnet*
      - Destination: *LAN Subnet*
      - Description: *Block: DMZ to LAN*
      - Save
      - Aggiungi nuova regola basata su quella appena creata
      - Action: *Pass*
      - Source: *DMZ*
      - Destination: *Any*
      - Protocol: *Any*
      - Description: *Allow: DMZ to ANY*
      - Save

## 11. Configurazione degli Aliases

Gli aliases sono una maniera comoda di ridenominazione degli indirizzi ip, in pratica è possibile sostituire gli ip con nomi a propria scelta. In questo modo, anche in caso di modifica degli indirizzi IP, sarà sufficiente cambiare una sola volta l'indirizzo, e tutti i campi collegati a quell'alias saranno aggiornati automaticamente

- Firewall
  - Aliases

### Firewall: Aliases

	Name	Address	Description	
<input type="checkbox"/>	host-client	192.168.31.100	l'ip del mio host in LAN	
<input type="checkbox"/>	host-pcospitante	172.30.4.1	Il computer da cui opero (windows)	
<input type="checkbox"/>	host-router-dmz	192.168.131.1	router monowall DMZ	
<input type="checkbox"/>	host-router-lan	192.168.31.1	router monowall LAN	
<input type="checkbox"/>	host-server	192.168.131.250	server in DMZ	
<input type="checkbox"/>	lan-labsistemi	172.30.4.0	la rete in cui appoggia la mia WAN	
<input type="checkbox"/>	server-aptcacher	172.30.1.199	Indirizzo del server apt-cacher-fermi.intra	
				 

## 12. Migrazione indirizzi IP

E' possibile che possa emergere la necessità di dover cambiare una serie di IP nella nostra rete contemporaneamente. Il rischio maggiore è quello di perdere l'accesso al router modificando gli IP in maniera errata.

Nel nostro caso, si deve migrare l'IP della rete LAN, da 192.168.1.0 a 192.168.31.0 .

Il numero 31 indica il numero di postazione nel laboratorio.

### 12.1 Migrazione indirizzo IP client

- Agire dalla modalità [root@client](#)
- Inserire il seguente comando:
  - *ifconfig enp0s3 **tempIP** netmask 255.255.255.0*
  - *route add default gw **newGW***
- Entrare nella configurazione web di m0n0wall
- Interfaces: LAN
  - IP: 192.168.31.1
- Services: DHCP server → LAN
  - Range: 192.168.31.100 to 192.168.31.199
- Riavviare il router
- Agire dalla modalità [uds@client](#)
- *sudo dhclient enp0s3* → Richiesta DHCP

In questo momento l'interfaccia di rete avrà due IP assegnati, per risolvere questo problema è necessario:

- Agire dalla modalità [root@client](#)
- *ifdown enp0s3* → disabilita interfaccia
- *ifup enp0s3* → abilita interfaccia

## 12.2 Migrazione indirizzo IP server

- Connettersi in ssh al server dal client
  - Inserire il comando: `ssh uds@192.168.101.1`
- Inserire il seguente comando:
  - `ifconfig enp0s3 tempIP netmask 255.255.255.0`
  - `route add default gw newGW`
- Entrare nella configurazione web di m0n0wall
- Interfaces: DMZ
  - IP: `192.168.131.1`
- Services: DHCP server → DMZ
  - Range: `192.168.131.100` to `192.168.131.199`
- In seguito si assegnerà al server l'indirizzo statico `192.168.131.250`
  - Inserire il comando:
    - `ifconfig enp0s3 192.168.131.250 netmask 255.255.255.0`
    - `route add default gw 192.168.131.1`
- Riavviare il router
- Agire dalla modalità [uds@server](#)
- `sudo dhclient enp0s3` → Richiesta DHCP

In questo momento l'interfaccia di rete avrà due IP assegnati, per risolvere questo problema è necessario:

- Agire dalla modalità [root@server](#)
- `ifdown enp0s3` → disabilita interfaccia
- `ifup enp0s3` → abilita interfaccia



## 13. Configurazione regole di firewall

Si procede ora alla configurazione delle regole di firewall per regolare gli accessi alle varie zone della rete.

### 13.1 Configurazione del firewall: LAN

#### Firewall: Rules

LAN

WAN

DMZ

	Proto	Source	Port	Destination	Port	Description
✗	TCP/UDP	LAN net	*	! host-router-lan	53 (DNS)	Block DNS request not to router
↑	*	LAN net	*	*	*	Default LAN to any

1. Blocco le richieste DNS dalla LAN che non sono indirizzate al router
2. La LAN può andare ovunque, come di Default

## 13.2 Configurazione del firewall: WAN

### Firewall: Rules

LAN

WAN

DMZ

	Proto	Source	Port	Destination	Port	Description
↑	TCP	host-pcospitante	*	WAN address	80 (HTTP)	allow: accesso web al m0n0wall dal pc ospitante
↑	TCP	host-pcospitante	*	host-server	22 (SSH)	NAT Server in SSH
↑	ICMP	*	*	WAN address	*	Allow ping to WAN
↑	TCP	*	*	host-server	80 (HTTP)	Accesso al WebServer dalla porta 80

1. Permette l'accesso al m0n0wall dal pc ospitante in porta 80
2. Permette la connessione dalla WAN al Server in DMZ in SSH
3. Permette di eseguire il ping su tutti i dispositivi della WAN
4. Indirizza la porta 80 al WebServer ( Servirà in seguito con l'installazione di apache )

## 13.3 Configurazione del firewall: DMZ

### Firewall: Rules

LAN

WAN

DMZ

	Proto	Source	Port	Destination	Port	Description
↑	TCP	DMZ net	*	host-client	22 (SSH)	Allow: Server to LAN SSH via port 5022
✗	*	DMZ net	*	LAN net	*	Block: DMZ to LAN
↑	ICMP	DMZ net	*	*	*	Allow: Ping to any
↑	UDP	DMZ net	*	*	53 (DNS)	Allow: DNS to any
↑	UDP	DMZ net	*	*	123	Allow: NTP to any
↑	TCP	DMZ net	*	apt-cacher	3142	Allow: Only update from server apt-cacher
↑	*	DMZ net	*	*	*	Allow: DMZ to any (Normally disabled)

1. Permette l'SSH dal Server al Client in porta 5022
2. Blocca gli accessi dalla DMZ alla LAN
3. Permette il ping verso tutte le reti
4. Permette il DNS verso tutte le reti
5. Permette l'NTP verso tutte le reti
6. Permette l'update dei programmi attraverso l'apt-cacher della scuola
7. Permette gli accessi dalla DMZ a tutte le reti (Disabilitata)

## 14. Configurazione regole del NAT

### Firewall: NAT: Inbound

Inbound	Server NAT	1:1	Outbound
---------	------------	-----	----------

If	Proto	Ext. port range	NAT IP	Int. port range	Description
WAN	TCP	22 (SSH)	host-server	22 (SSH)	Server in SSH
DMZ	TCP	22 (SSH)	host-client	8888	Block: Server SSH in port 22
WAN	TCP	80 (HTTP)	host-server	80 (HTTP)	Reindirizzamento al server "Apache"
DMZ	TCP	5022	host-client	22 (SSH)	DMZ to LAN in SSH via port 22 or 5022
WAN	TCP	8080	host-router	80 (HTTP)	Reindirizzamento al router m0n0wall dalla porta 8080

1. Permette l'SSH verso il Server
2. Blocca l'SSH dal Server in porta 22 e lo indirizza verso una porta vuota per mandarlo in timeout
3. Regola che reindirizza la porta 80 al WebServer di Apache
4. Permette l'SSH dal Server alla LAN attraverso la porta 5022
5. Regola che reindirizza la richiesta alla porta 8080 alla configurazione del router m0n0wall

## 15. Installazione di Apache

Procederemo ora all'installazione di Apache sul server, che ci servirà per la creazione del WebServer.

1. Agire dalla modalità [uds@servermilani](mailto:uds@servermilani)

-Inserire i comandi: - *sudo apt update*  
(Per aggiornare i pacchetti contenuti nei repository)

- *sudo apt install apache2*

- *sudo service apache2 start*  
(Per accendere il WebServer di apache)

2. Modifica dell'index del WebServer:

Prima di procedere alla creazione di un nuovo index come pagina principale rinomino il file index.html già presente

- Inserire il comando *sudo mv index.html index.html1*

Ora creare un nuovo file html con la prima pagina del nostro WebServer

- Inserire il comando *sudo touch index.html*

## 16. Abilitazione certificato SSL

Un certificato SSL (Secure Sockets Layer) viene utilizzato per stabilire un collegamento crittografato tra un server ed i vari client che vi si connettono. Ad ogni sessione il collegamento SSL protegge le informazioni per garantirne la non intercettazione da parte di soggetti non autorizzati.

Esistono tre tipi di certificati:

- A pagamento, acquistabile da agenzie che forniscono questo tipo di servizi;

**Informazioni sul certificato**

---

**Scopo certificato:**

- Garantisce l'identità di un computer remoto
- Dimostra la propria identità ad un computer remoto
- 2.16.840.1.114412.1.1
- 2.23.140.1.2.2

\* Per ulteriori dettagli consultare l'informativa dell'Autorità di ce

---

**Rilasciato a:** [www.visasoutheasteurope.com](http://www.visasoutheasteurope.com)

**Rilasciato da:** GeoTrust RSA CA 2018

**Valido dal** 18/02/2020 **al** 19/05/2021

- Gratis, ottenibile attraverso software conosciuti e affidabili.

**Informazioni sul certificato**

---

**Scopo certificato:**

- Garantisce l'identità di un computer remoto
- Dimostra la propria identità ad un computer remoto
- 2.23.140.1.2.1
- 1.3.6.1.4.1.44947.1.1.1

\* Per ulteriori dettagli consultare l'informativa dell'Autorità di ce

---

**Rilasciato a:** [e-fermi.it](http://e-fermi.it)

**Rilasciato da:** Let's Encrypt Authority X3

**Valido dal** 22/01/2020 **al** 21/04/2020

## 16. Abilitazione certificato SSL – continua

- Certificato autofirmato, viene creato nel server ma non assicura totalmente la privacy della comunicazione. Il browser non riconoscerà questo tipo di certificato, bloccando la visione del sito.



Noi procederemo all'abilitazione del terzo tipo di certificato, ovvero di quello autofirmato. Per poter accedere al sito senza incorrere nel blocco da parte del browser sarà necessario inserire una regola di nat.

## 16.1 Inserimento regola di NAT

Come già detto in precedenza, è necessario inserire una regola di NAT per far visualizzare il sito web con il certificato autofirmato al browser.

### 1. Agire da m0n0wall web

- NAT → Inbound

If	Proto	Ext. port range	NAT IP	Int. port range	Description
WAN	TCP	443 (HTTPS)	host-server	443 (HTTPS)	Allow: server access from port 443

## 16.2 Creazione di un certificato autofirmato

### 1. Agire dalla modalità [root@server](#)

Inserire i comandi:

- `a2enmod ssl` per abilitare il modulo ssl
- `systemctl restart apache2` per riavviare il servizio apache

```
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Come scritto, per poter configurare il modulo SSL e creare un certificato autofirmato si può consultare il file README che apache mette a disposizione.

### 2. Inserire i comandi:

- `gzip -d /usr/sharedoc/apache2/README.Debian.gz` per decomprimere il pacchetto contenente il file README
- `less /usr/sharedoc/apache2/README.Debian` per consultare il file



## 16.2 Creazione di un certificato autofirmato – continua

Come scritto nel README il certificato autofirmato di può creare attraverso il pacchetto `ssl-cert`.

3. Inserire il comando:

- `make-ssl-cert generate-default-snakeoil -force-overwrite`

Ora bisognerà modificare il file di configurazione, aggiungendo le righe indicanti ad apache la gestione delle richieste https.

4. Inserire il comando:

- `nano /etc/apache2/sites-available/000-default.conf`

5. Dopo l'ultimo commento inserire:

```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
</VirtualHost>
```

6. Inserire il comando:

- `systemctl restart apache2` per riavviare il servizio apache

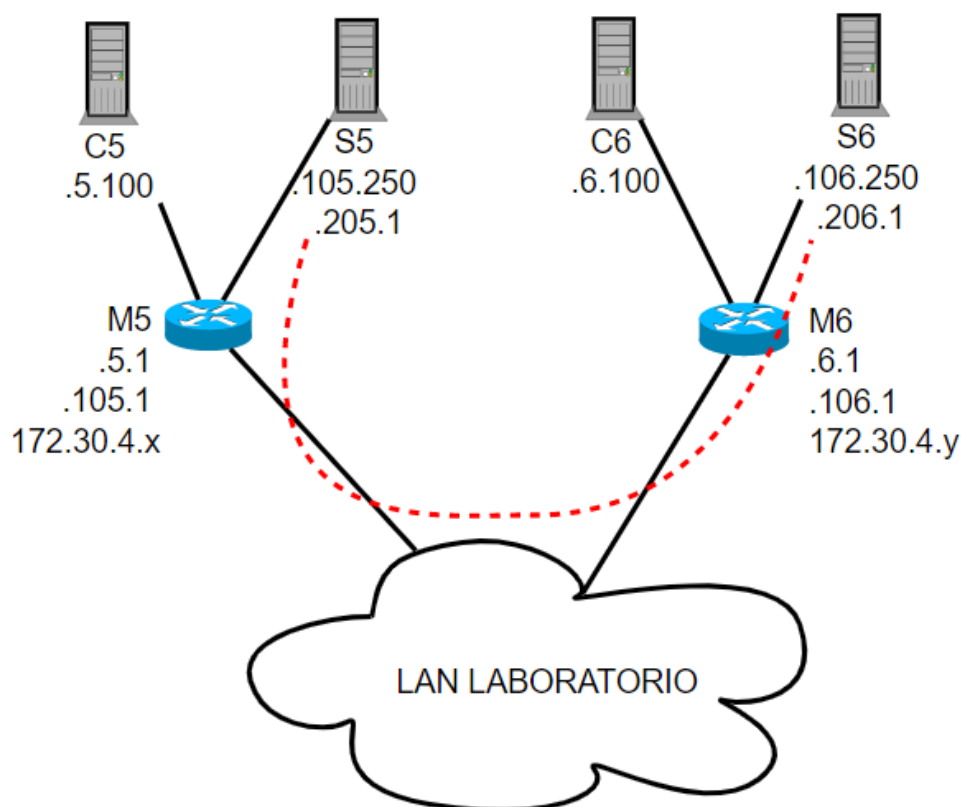
## 17. Installazione e configurazione di una VPN

Una VPN (Virtual Private Network) è una connessione instaurata tra uno o più pc privata e protetta da algoritmi di cifratura.

Il nostro obiettivo è quello di riuscire a pingare dal server1 al server 2 e viceversa. Il server 1 sarà il nostro, mentre il server 2 sarà quello del nostro vicino di banco, nel mio caso Zen.

Le due interfacce dovranno avere un proprio indirizzo IP, che sarà dato dal  $192.168.200 + \text{numerohost}.1$ .

Nel nostro caso il mio indirizzo sarà  $192.168.231.1$ , mentre quello di Zen sarà  $192.168.202.1$ .



## 17.1 Configurazione IPsec

### 1. Agire da m0n0wall web

- VPN → IPsec
  - Abilitare il protocollo →
  - Creare un nuovo tunnel



<b>Interface</b>	WAN ▼
<b>DPD interval</b>	60 seconds
<b>Local subnet</b>	Type: LAN subnet ▼
<b>Remote subnet</b>	192.168.202.0 / 24 ▼
<b>Description</b>	IPsec verso Zen

<b>Negotiation mode</b>	aggressive ▼
<b>My identifier</b>	My IP address ▼
<b>Encryption algorithm</b>	3DES ▼
<b>Hash algorithm</b>	MD5 ▼
<b>DH key group</b>	2 (1024 bit) ▼
<b>Lifetime</b>	28800 seconds
<b>Authentication method</b>	Pre-shared key ▼
<b>Pre-Shared Key</b>	pippo

<b>Protocol</b>	ESP ▼
<b>Encryption algorithms</b>	<input type="checkbox"/> DES <input checked="" type="checkbox"/> 3DES <input checked="" type="checkbox"/> Blowfish
<b>Hash algorithms</b>	<input type="checkbox"/> SHA-1 <input checked="" type="checkbox"/> MD5
<b>PFS key group</b>	2 (1024 bit) ▼
<b>Lifetime</b>	28800 seconds

Questa configurazione va riportata allo stesso modo nel router remoto, cambiando la remote subnet e il remote gateway.

## 17.2 Configurazione OpenVPN

Come prima azione è necessario aggiungere una regola di firewall e di NAT per poter permettere al router di gestire il traffico generato da OpenVPN.

### 1. Agire da m0n0wall web

- Firewall → Rules → DMZ

	Proto	Source	Port	Destination	Port	Description
↑	UDP	host-server	1194	*	1194	Allow: OpenVPN Traffic

- Firewall → NAT → Inbound

If	Proto	Ext. port range	NAT IP	Int. port range	Description
WAN	UDP	1194	host-server	1194	NAT: OpenVPN Traffic

### 2. Agire dal server in modalità [uds@server](#)

- `sudo apt update`
- `sudo apt upgrade`
- `sudo apt install openvpn`

### 2. Generare una chiave che verrà condivisa con l'altro server

- `cd /etc/openvpn →` Cartella di lavoro convenzionale
- `openvpn --genkey --secret nomechiave.key`

## 17.2 Configurazione OpenVPN - continua

### 3. Creazione del file di configurazione

- `nano /etc/openvpn/nome.conf`

### 4. Inserire i seguenti parametri:

```
dev tun10
port 1194
proto udp
ifconfig 192.168.231.1 192.168.202.1
remote 172.30.4.93
secret /etc/openvpn/milanizen.key
log-append /var/log/openvpn-milanizen.log
comp-lzo
float
```

### 5. Inviare la chiave all'host remoto tramite un canale sicuro ( Noi abbiamo usato smtp )

### 6. Nell'altro server andremo a creare un altro file di configurazione identico a quello del nostro server, tranne per la direttiva "remote" che si utilizza solamente per la connessione a remoto.

### 7. Avviare OpenVPN

- `openvpn --config etc/openvpn/nome.conf --verb6`
- `/etc/init.d/openvpn start`
- `systemctl enable openvpn openvpn@serverconfig`

### 7. Test della VPN

Pingare dal server 1 al server 2 (e viceversa nel caso entrambi abbiano la direttiva remote).

## 18. Installazione e configurazione SNMP e MRTG

Ora andremo ad installare e configurare il protocollo SNMP, utile per la gestione ed il monitoraggio dei dispositivi di rete.

### 18.1 Configurazione SNMP

1. Agire da m0n0wall web
  - Services → SNMP
2. Inserire i seguenti parametri
  - System location: Bassano Del Grappa
  - System Contact: ITIS E. Fermi
  - Community : public

Il campo community è una sottospecie di password, che permette di recuperare le statistiche di un router o di altri device di cui si voglia tener traccia.

3. Aggiungere una regola nel firewall WAN per l'interrogazione di SNMP



Proto	Source	Port	Destination	Port	Description
UDP	DMZ net	*	host-server	161	Allow: SNMP connection

4. Agire dalla modalità [uds@server](#)
  - `sudo apt-get install snmp snmpd`
5. Modificare il file di configurazione, decommentando la direttiva:
  - `rocommunity public localhost`

## 18.1 Configurazione SNMP – continua

6. Rimuovere “-V systemonly” dalla direttiva:

- `rocommunity public default -V systemonly`

7. Avviare il servizio:

- `systemctl enable snmpd`
- `sudo systemctl start snmpd`

8. Inserire il comando *snmpconf* e le seguenti opzioni:

- `snmp, snmpd` → all (File da leggere)
- `snmp2.conf` → 2 (File da creare)
- `various` → 1 (Configurazione)
- `disk usage` → 2 (Cosa monitorare)
- `mount point` → / (Partizione)
- `minimum amount` → 100 000 (Spazio minimo)
- `finished`
- `finished`
- `quit`

5. Riavviare il servizio con il comando:

- `service snmpd restart`

## 18.2 Configurazione MRTG

MRTG (Multi Router Traffic Grapher) è utile per avere una visualizzazione, basilare, dei grafici di rete.

1. Agire dalla modalità [uds@server](mailto:uds@server)
  - `sudo apt-get install mrtg`
2. Creare la cartella per mrtg:
  - `sudo mkdir /var/www/mrtg`
3. Assegnare i permessi alla cartella mrtg:
  - `chown -R www-data:www-data /var/www/mrtg`

Per permettere l'accesso alla cartella da parte dell'apache si creerà un link simbolico, che permetterà di lasciare la cartella nella posizione originale, quindi senza creare problemi ad eventuali file di configurazione, ma potrà essere visionata anche da un'altra posizione. Nel nostro caso il link simbolico verrà creato con la cartella `/var/www/mrtg`

- `cd /var/www/html`
- `ln -s /var/www/mrtg`



## 18.2 Configurazione MRTG - continua

Ora andremo a creare un file virtualhost in apache per poter abilitare la visualizzazione dei grafici sul sito.

4. Inserire il comando:

- `nano /etc/apache2/sites-available/mrtg.conf`

5. Inserire i parametri:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot "/var/www/mrtg"
    ServerName servermilani.milani.intra
    <Directory "/var/www/mrtg/">
        Options None
        AllowOverride None
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>
    TransferLog /var/log/apache2/mrtg_access.log
    ErrorLog /var/log/apache2/mrtg_error.log
</VirtualHost>
```

6. Riavviare apache:

- `sudo service apache2 restart`

7. Abilitare mrtg:

- `sudo a2ensite mrtg`

## 18.2 Configurazione MRTG – continua

Ora andremo a creare il file di configurazione di mrtg attraverso il comando *cfgmaker*, per gestire le informazioni ottenute da SNMP.

8. Inserire il comando:

- `cfgmaker public@192.168.131.1 > /etc/mrtg.cfg`

Attraverso questo comando avremo le statistiche di tutte e tre le interfacce del router (LAN, WAN, DMZ).

Poi andremo anche ad aggiungere le statistiche relative al disco del server.

9. Inserire il comando:

- `cfgmaker public@localhost >> /etc/mrtg.cfg`

Per evitare errori, dato che con il doppio maggiore siamo andati ad appendere il risultato del comando al file `mrtg.cfg` già creato, dovremmo andare a commentare due direttive.

10. Inserire il comando:

- `nano /etc/mrtg.cfg`

Commentare le direttive *WorkDir* e *EnableIPv6* quando si ripresentano la seconda volta.

Ora si dovrà creare l'index attraverso il comando *indexmaker*, per poter mostrare i grafici del sito.

11. Eseguire il comando:

- `indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html`

## 19. Installazione e configurazione di Cacti



Cacti è uno strumento di monitoraggio e rappresentazione grafica dei dispositivi di rete.

Useremo Cacti al posto di MRTG in quanto dispone di molte più opzioni e di un'interfaccia grafica migliore rispetto ai grafici MRTG

Con cacti andremo a tracciare:

- m0n0wall
- Server → Dischi, dispositivi di rete
- Apparecchiatura a scelta dell'istituto

### 19.1 Installazione sul server

Prima di procedere all'installazione di cacti, ci sarà l'installazione e configurazione di mariadb.

Agire dalla modalità `uds@server`

1. Inserire i comandi:

- `sudo apt update`
- `sudo apt -y upgrade`
- `sudo apt install mariadb-server mariadb-client`

Agire dalla modalità [`root@server`](#) (`sudo -s`)

2. Inserire il comando:

- `mysql_secure_installation`
- `password lasolita`

## 19.1 Installazione sul server - continua

3. Inserire il comando:

- `sudo nano etc/mysql/mariadb.conf.d/50-server.cnf`

Modificare le righe:

- `character-set-server = utf8mb4`
- `collation-server = utf8mb4_unicode_ci`

4. Inserire i comandi:

- `sudo systemctl restart mariadb`

Ora procederemo all'installazione di cacti

5. Inserire i comandi:

- `sudo apt-get install php php-mysql php-snmp`
- `sudo apt-get install cacti snmp`

6. Seguire la procedura di installazione:

- `apache2`
- `password` → `lasolita`

## 19.2 Configurazione di cacti

Ora possiamo accedere al cacti per configurarlo.

1. Accedere a cacti:

- 172.30.4.112/cacti
- username: admin password: lasolita



2. Selezionare *create devices*

3. Selezionare la + in alto a destra

Ora inseriremo le due configurazioni, una per tracciare m0n0wall, l'altra per tracciare il server in DMZ.

4. Configurazione per m0n0wall:

- Descrizione: m0n0wall
- Nome host: 192.168.131.1
- Device Template: Generic SNMP Device
- SNMP Version: Version 1

5. Salvare.

## 19.2 Configurazione di cacti – continua

### 6. Configurazione per il server in DMZ

- Descrizione: Server in DMZ
- Nome host: 127.0.0.1

### 7. Salvare.

Per poter rendere effettivamente attiva la configurazione del server in DMZ bisogna apportare una modifica al file `/etc/snmp/snmpd.conf`

### 8. Inserire il comando:

- `sudo nano /etc/snmp/snmp.conf`

### 9. Modificare le righe:

- `rocommunity public default -V` (aggiungere `-V`)
- `rocommunity6 public default -V systemonly` (Eliminare `systemonly`)

Ora andremo a creare gli alberi per avere una visualizzazione completa dei grafici

### 10. Selezionare Gestione

### 11. Selezionare Trees

### 12. Selezionare +

- Nome: m0n0wall/Server in DMZ (Due alberi diversi)

### 13. Selezionare Graphs

- Selezionare tutti i grafici “m0n0wall” oppure “Server in DMZ”

### 14. Selezionare la lista Scegli l'azione

- Place on a Tree (m0n0wall o Server in DMZ)

### 15. Tornare su Trees, selezionare il grafico, poi selezionare la lista Scegli l'azione

- Pubblica → Via

Nella sezione Graphs compariranno i nostri grafici.

## 20. RAID

Il RAID , ovvero **Redundand Array of Independent Disks**, è letteralmente un “insieme ridondante di dischi indipendenti”.

E' una tecnica di installazione basata sul raggruppamento di diversi dischi rigidi che operano come un unico volume di memorizzazione.

Si utilizza per migliorare le prestazioni di archiviazione dei dati, garantendo una maggiore disponibilità.

Il RAID è diverso dal backup, in quanto **non** salva un'istanza del disco, ma viene aggiornato a ogni modifica e procede in pari con il resto del sistema, per poter essere utilizzabile immediatamente in caso di rottura di un disco.

Ora analizzo i due diversi sistemi di RAID: hardware e software.

### - RAID **hardware**:

Il sistema basato sull'hardware gestisce il RAID indipendentemente dall'host, presentando a quest'ultimo solo un unico disco per ogni array.

Un dispositivo hardware di RAID si collega ad un controller SCSI, presentandosi come un'unica unità.

Infatti tutto il meccanismo RAID è spostato in un controller che si trova in un sottosistema esterno del disco. Il controller agisce come un controller SCSI gestendo autonomamente tutte le comunicazioni del disco.

E' la tipologia di RAID più costosa, in quanto sarà necessario acquistare dischi prestanti, schede controller e chassis hot-swap per il cambio del disco a caldo.

### - RAID **software**:

Il sistema basato sul software implementa i vari livelli RAID nel codice disco del kernel.

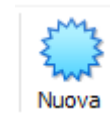
Il driver MD nel kernel Linux è un esempio di RAID indipendente dall'hardware. Le prestazioni dipendono dalle prestazioni e dal carico della CPU del server.

E' la tipologia di RAID più economica, in quanto non sono richieste né schede di controllo o chassis hot-swap. Funziona infatti con dischi IDE o SCSI più economici.

Grazie alle CPU disponibili le prestazioni di RAID software superano quelle di RAID hardware.

## 20.1 Creazione macchina virtuale in RAID 1 con due dischi

- Aprire il gestore di VirtualBox e cliccare su “Nuova”



- Nome: *debianraid*
- Tipo: *Linux*
- Versione: *Debian 64-bit*
- RAM: *1024 MB*

- Nuovo disco fisso

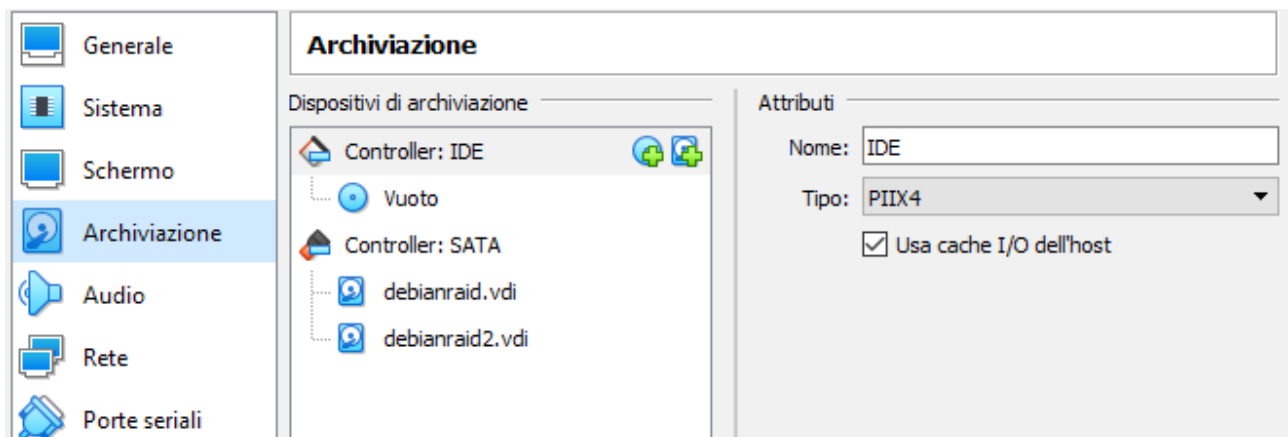
- Tipo di file: *VDI*
- Archiviazione: *Allocato dinamicamente*
- Dimensione: *4 GB*

Ora andremo a creare il secondo disco fisso per poter fare un sistema RAID 1

- Sul gestore di VirtualBox cliccare su “Impostazioni”



- Entrare nella sottocategoria “Archiviazione”



- Nella sezione Controller: SATA aggiungere un nuovo disco fisso:

- Nuovo disco
- Tipo di file: *VDI*
- Nome: *debianraid2*
- Archiviazione: *Allocato dinamicamente*
- Dimensione: *4 GB*

- Aumentare il numero di porte del controller SATA a 4 porte.



## 20.2 Installazione Debian

Per l'installazione di Debian, seguire il manuale dal [capitolo 3](#) fino al paragrafo 3.2, prima del partizionamento dei dischi.

## 20.3 Partizionamento dei dischi

- Metodo di partizionamento: *Manuale*

Spostarsi su “ SCSI3 (0,0,0) (sda) – 4.3 GB ” e iniziare il partizionamento di quel disco.

- Creare una nuova tabella delle partizioni vuota
- Tipo della tabella delle partizioni: *msdos*

Spostarsi su “ pri/log 4.3 GB SPAZIO LIBERO “.

- Creare una nuova partizione
  - Dimensione della partizione: *4 GB*
  - Tipo della partizione: *Primaria*
  - Posizione della nuova partizione: *Inizio*
  - Usare come: *File system ext4 con journaling*
  - Punto di mount: */*
  - Opzioni di mount: *discard, noatime*
  - Etichetta: *linuxroot*
- Impostazione della partizione completata

Spostarsi su “ pri/log 294.6 MB SPAZIO LIBERO “

- Creare una nuova partizione
  - Dimensione della partizione: *294.6 MB*
  - Tipo della nuova partizione: *Primaria*
  - Usare come: *area di swap*
- Impostazione della partizione completata

Eseguire gli stessi passi nel secondo disco.

## 20.4 Configurazione del RAID software

- Cliccare su “Configurare il RAID software”
- Scrivere i cambiamenti sui dispositivi di memorizzazione e configurare il raid
  - Creare un device multidisk (MD)
  - Tipo del device RAID software: *RAID1*
  - Numero di device attivi per l’array RAID1: 2
  - Numero dei device <<spare>> per l’array RAID1: 0

Abbiamo appena creato l’array RAID1 chiamato md0, non aggiungiamo alcun disco in quanto andremo ad eliminarlo per poter avere md1(sda1,sdb1) e md2(sda2,sdb2).

Passiamo ora a creare md1 e md2

- Creare un device multidisk (MD)
  - Tipo del device RAID software: *RAID1*
  - Numero di device attivi per l’array RAID1: 2
  - Numero dei device <<spare>> per l’array RAID1: 0
  - Device attivi per l’array RAID1:
    - */dev/sda1*
    - */dev/sdb1*
  - Scrivere i cambiamenti sui dispositivi di memorizzazione e configurare il RAID? *Sì*
- 
- Creare un device multidisk (MD)
  - Tipo del device RAID software: *RAID1*
  - Numero di device attivi per l’array RAID1: 2
  - Numero dei device <<spare>> per l’array RAID1: 0
  - Device attivi per l’array RAID1:
    - */dev/sda2*
    - */dev/sdb2*
  - Scrivere i cambiamenti sui dispositivi di memorizzazione e configurare il RAID? *Sì*

Ora “Eliminare un device multidisk (MD)” ed eliminare *md0*.

Il risultato dovrebbe essere uguale all'immagine sottostante.

```
RAID1 dispositivo n° 1 - 4.0 GB Device RAID software
n° 1 4.0 GB
RAID1 dispositivo n° 2 - 292.6 MB Device RAID software
n° 1 292.6 MB
SCSI3 (0,0,0) (sda) - 4.3 GB ATA VBOX HARDDISK
n° 1 primaria 4.0 GB K raid
n° 2 primaria 293.6 MB K raid
SCSI4 (0,0,0) (sdb) - 4.3 GB ATA VBOX HARDDISK
n° 1 primaria 4.0 GB K raid
n° 2 primaria 293.6 MB K raid
```

Ora dovremmo definire il file system di root.

- Selezionare “ n°1 4.0 GB ”
  - Usare come: *File system ext4 con journaling*
  - Punto di mount: */*
  - Opzioni di mount: *discard, noatime*
  - Etichetta: *linuxroot*
- Impostazione della partizione completata

Nel secondo dispositivo definiremo l'area di swap

- Selezionare “ n°2 292.6 MB ”
  - Usare come: *area di swap*
- Impostazione della partizione completata

Il risultato finale sarà quindi:

```
RAID1 dispositivo n° 1 - 4.0 GB Device RAID software
n° 1 4.0 GB f ext4 /
RAID1 dispositivo n° 2 - 292.6 MB Device RAID software
n° 1 292.6 MB f swap swap
```

- Terminare il partizionamento e scrivere le modifiche sul disco
- Terminare l'installazione di Debian fino dal [paragrafo 3.4](#) al paragrafo 3.8

ATTENZIONE: Installare il GRUB in entrambi i dischi (sda e sdb), per poter permettere al sistema di avviarsi nel caso di rottura del primo disco.

## 20.5 Controllo funzionamento del RAID

Per poter controllare il funzionamento del RAID sarà utile inserire il seguente comando dopo l'avvio della macchina:

- `cat /proc/mdstat`

L'output dovrebbe essere simile all'immagine sottostante.

```
root@debianraid:~# less /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sda1[0] sdb1[1]
      3902464 blocks super 1.2 [2/2] [UU]
      [=====>...]  resync = 89.0% (3474112/3902464) finish=1.2min speed=5509K/sec

md2 : active (auto-read-only) raid1 sdb2[1] sda2[0]
      285696 blocks super 1.2 [2/2] [UU]
      resync=PENDING

unused devices: <none>
```

Come possiamo vedere saranno attivi md1 e md2, rispettivamente con sda1,sdb1 e sda2,sdb2.

## 21. Simulazione della rottura di un disco in RAID 1

Ora andremo a testare l'effettivo funzionamento del RAID, simulando la rottura (oppure la sostituzione) di un disco a caldo, ovvero senza spegnere la macchina, ma agendo durante il suo funzionamento.

Per fare ciò non potremmo agire dal gestore di VirtualBox, in quanto non ci sarà permesso modificare i dischi durante il funzionamento di una macchina. Il gestore ci sarà comunque utile, in quanto dovremmo eseguire un'operazione necessaria per l'esperimento.

- Entrare nelle impostazioni della macchina (spenta)

- Sottocategoria "Archiviazione"
- Selezionare "debianraid.vdi"
- Spuntare "collegabile a caldo"
- Selezionare "debianraid2.vdi"
- Spuntare "collegabile a caldo"

Utilizzeremo quindi lo strumento "Vbox Manager" accessibile attraverso il comando ***vboxmanage*** da riga di comando.

- Entrare nella directory di VirtualBox.

- Da Windows dovrebbe trovarsi in "C:\Program Files\Oracle\VirtualBox"
- Eseguire un terminale ed inserire il comando " *vboxmanage* ".

Per simulare la rottura di un disco o il suo scollegamento, eseguire il comando:

- *vboxmanage storageattach "debianraid" --storagectl "SATA" --port 0 --medium none*

Le porte sono numerate partendo da 0, quindi ora stiamo staccando il primo disco.

Per controllare se effettivamente il disco è stato staccato eseguire sempre il comando *cat /proc/mdstat* .

Per riattaccare il disco sarà necessario eseguire il comando:

- *vboxmanage storageattach "debianraid" --storagectl "SATA" --port 0 --type hdd --medium "debianraid/debianraid.vdi"*

Dopo aver ricollegato il disco sarà necessario andare ad agire sui device MD, in quanto dopo la rottura di un disco verrà tolto dal device MD e non verrà ricollegato automaticamente.

Per modificare il device MD (nel nostro caso md1) inserire il comando:

- `mdadm /dev/md1 --add /dev/sda`

Per controllare l'avvenuta modifica, sarà necessario inserire il solito comando `cat /proc/mdstat` dove potremo vedere come il disco sda compaia nel device md1.

```
md1 : active raid1 sda1[2] sdb1[1]
      3902464 blocks super 1.2 [2/2] [UU]

md2 : active (auto-read-only) raid1 sdb2[1] sda2[0]
      285696 blocks super 1.2 [2/2] [UU]
      resync=PENDING
```

## 22. Configurazione di un RAID 5

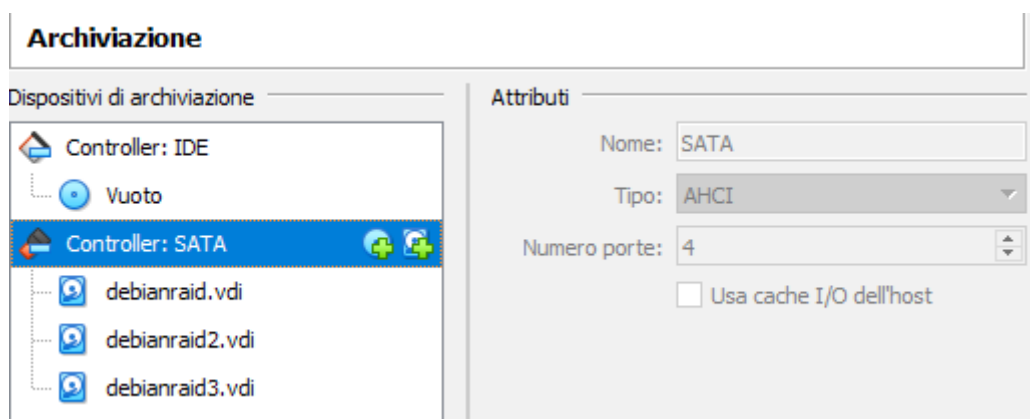
In questo capitolo andremo ad eseguire un upgrade da RAID 1 a RAID 5.

Il RAID 5 è sostanzialmente una miglioria del RAID 4, eliminando il problema del disco “collo di bottiglia”, apportando una parità distribuita.

Le letture e le scritture sono più veloci, in quanto rispetto al RAID 4, con la parità distribuita, ha un disco in più disponibile per le letture parallele.

Per poter configurare un RAID 5 avremo bisogno di un minimo di 3 dischi, quindi andremo ad aggiungerne un altro alla nostra macchina virtuale.

- Dal gestore di VirtualBox entrare su Impostazioni e nella sottocategoria Archiviazione.
- Nella sezione Controller: SATA aggiungere un nuovo disco fisso:
  - Nuovo disco
  - Tipo di file: *VDI*
  - Nome: *debianraid3*
  - Archiviazione: *Allocato dinamicamente*
  - Dimensione: *4 GB*



## 22.1 Partizionamento del nuovo disco

In questo passo andremo a partizionare il disco appena inserito, utilizzando il programma “parted”.

- Installare il programma attraverso il comando: *apt install parted*
  - Controllare il partizionamento attuale: *parted -l | less*

Se è stato eseguito correttamente tutto il processo il risultato dovrebbe essere simile al seguente:

```
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 4295MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
-----  -
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 4295MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
-----  -
1       1049kB  4000MB  3999MB  primary ext4          raid
2       4000MB  4294MB  294MB   primary linux-swap(v1) raid
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 4295MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

Si può notare come sda e sdb siano effettivamente partizionati, mentre sdc no.

- Partizionamento guidato disco sdc
  - Eseguire il partizionamento attraverso il comando: *parted /dev/sdc*
  - Creare una tabella delle partizioni attraverso il comando: *mklabel msdos*
  - Creare la partizione di filesystem attraverso il comando:  
*mkpart primary ext4 0 4000MB*
  - Creare la partizione di swap attraverso il comando:  
*mkpart primary linux-swap 4000MB 4295MB*
  - Attivare la flag in entrambe le partizioni attraverso i comandi:  
*set 1 raid on* e *set 2 raid on*



- Per verificare l'effettiva riuscita del processo eseguire il comando: *print*

```
(parted) mkpart primary ext4 0 4000MB
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? i
(parted) mkpart primary linux-swap 4000MB 4295MB
(parted) set 1 raid on
(parted) set 2 raid on
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 4295MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
  1      512B   4000MB  4000MB  primary  ext4         raid, lba
  2      4000MB 4295MB  295MB   primary  linux-swap(v1) raid, lba
```

- Chiudere parted inserendo “q”

- Eseguire un reboot del sistema per rendere effettive le modifiche

## 22.2 Aggiunta del disco ai device MD

Dopo il partizionamento del terzo disco, per poter configurare il RAID 5 è necessario aggiungere le partizioni nuove ai device MD.

- Aggiungere la partizione ext4 a md1 attraverso il comando:  
*mdadm --add /dev/md1 /dev/sdc1*
- Aggiungere la partizione di swap a md2 attraverso il comando:  
*mdadm --add /dev/md2 /dev/sdc2*

Infine per rendere attivo il RAID 5 sarà necessario inserire i seguenti comandi:

- *mdadm /dev/md1 --grow --raid-devices=3 --level=5*
- *mdadm /dev/md2 --grow --raid-devices=3 --level=5*

Come ultimo passaggio andremo ad installare il GRUB nel terzo disco per renderlo a tutti gli effetti uguale agli altri due.

Basterà inserire il comando:

- *grub-install /dev/sdc*

Al termine delle operazioni andremo a controllare il RAID attraverso il comando: *cat /proc/mdstat*