

"mdmon" - Multiple Device Monitoring

Laboratorio di Amministrazione di Sistemi

Gabriele Aloisio [503264]

2023

Università degli studi di Messina

Table of contents

Introduction

Debian e RAID

SAMBA

Creazione e configurazione dell'utenza

Sistema di alert

Installazione e configurazione di Postfix

Conclusioni



Introduction

Descrizione del progetto

Il nostro progetto si focalizza sulla progettazione e implementazione di un'infrastruttura di storage basata su RAID in un ambiente operativo Linux. La configurazione di tale sistema prevede l'utilizzo di **RAID 1** (OS) e **RAID 5** (directory condivisa con **SAMBA**). Viene successivamente configurata la gestione degli accessi, consentendo agli utenti di accedere esclusivamente alle cartelle condivise. In fine, al fine di garantire una pronta risposta a eventuali guasti dei dischi, abbiamo implementato un sistema di alert. Questo sistema avverte gli amministratori di sistema tramite e-mail in caso di malfunzionamenti.



Il RAID, acronimo di *Redundant Array of Independent Disks*, è una tecnologia di archiviazione che combina più dischi rigidi in un'unica unità logica. L'obiettivo principale è migliorare la prestazione e/o fornire ridondanza dei dati per aumentare l'affidabilità e la sicurezza del sistema di storage.



Nel contesto specifico del nostro progetto, implementeremo due livelli di RAID:

- **RAID 1:** In una configurazione RAID 1, due dischi rigidi contenenti gli stessi dati sono utilizzati in parallelo. Ogni dato scritto su un disco viene duplicato sull'altro, creando una copia identica.
- **RAID 5:** Nel caso di RAID 5, la ridondanza dei dati viene ottenuta mediante la distribuzione delle informazioni di parità su tutti i dischi del RAID array. Questo schema permette al sistema di recuperare i dati in caso di guasto di uno dei dischi.



Debian è un sistema operativo open-source basato su Linux, rinomato per la sua stabilità, affidabilità e flessibilità, ed è supportato da una vasta comunità di sviluppatori e utenti appassionati. È una scelta popolare per una varietà di utilizzi, dalle implementazioni di server aziendali alle soluzioni personalizzate. Grazie al suo *package manager* **APT**, gli utenti possono facilmente installare, aggiornare e gestire le applicazioni con pochi comandi. Debian è particolarmente adatto per coloro che cercano un sistema operativo stabile e altamente personalizzabile.



Come ambiente desktop, abbiamo optato per **XFCE** per la sua leggerezza e semplicità. **XFCE** offre un'esperienza utente intuitiva senza compromettere le risorse di sistema, il che è importante per garantire prestazioni ottimali in ambienti server. La combinazione di **Debian** e **XFCE** ci fornisce un sistema stabile, facile da gestire e ottimizzato per le nostre esigenze di progetto.



Cron è un servizio in ambiente Unix e Unix-like che consente agli utenti di programmare l'esecuzione automatica di comandi o script a intervalli specifici di tempo, giorni della settimana o mesi. In merito al progetto, utilizzeremo cron per automatizzare l'esecuzione periodica dello script **mdmon**. Questo script è progettato per monitorare lo stato dei dischi in un array RAID e inviare notifiche agli amministratori in caso di anomalie o malfunzionamenti.



SAMBA è una suite di software che facilita l'integrazione di sistemi basati su Linux e Windows in una rete. Con **SAMBA**, è possibile condividere file e risorse tra piattaforme eterogenee. *SAMBA* permette agli utenti di accedere a cartelle condivise, stampanti e altri servizi, con sicurezza e controllo degli accessi.



Postfix è un server di trasferimento di posta (MTA) utilizzato per la gestione dell'invio e della ricezione delle email in un sistema Linux. Nel contesto del nostro progetto, utilizzeremo Postfix per l'autenticazione e il relay delle email tramite il server SMTP di Google. Ciò significa che Postfix fungerà da intermediario tra il nostro sistema e il server SMTP di Google per inviare le email.



Debian e RAID

L'installer ci permette comodamente di configurare la partizione **RAID 1** per il mountpoint di root (/) prima ancora di installare Debian. Il processo utilizzato è il seguente:



Processo di configurazione RAID1

1. Creare due partizioni vuote per vda e vdb
2. Impostare il tipo delle partizioni come *volume fisico per RAID*
3. Creare un MD (multiple device) di tipo RAID1 (md0)
4. Impostare il numero di device attivi a 2
5. Impostare il numero di spare device a 0
6. Selezionare i device da utilizzare per l'array RAID1 (vda1 e vdb1)
7. Creare Il filesystem di root sulla partizione RAID1 appena creata



Installiamo mdadm con

```
$ sudo apt update
```

```
$ sudo apt install mdadm -y
```



Sucessivamente formattiamo e partizioniamo i dischi vdc, vdd e vde con fdisk

```
aboveskip
1      Command (m for help): n
2      Partition type
3          p   primary (0 primary, 0 extended, 4 free)
4          e   extended (container for logical partitions)
5      Select (default p): p
6      [...]
7      Command (m for help): t
8      Selected partition 1
9      Hex code (type L to list all codes): fd
10     Changed type of partition 'Linux' to 'Linux raid autodetect'.
11
12     Command (m for help): w
13     The partition table has been altered.
14     Calling ioctl() to re-read partition table.
15     Syncing disks.
16
```



Configurazione di RAID5

Creiamo l'array RAID5

```
$ mdadm --create /dev/md1 --level=5  
> --raid-devices=3 /dev/vdc /dev/vdd /dev/vde
```

Ed un filesystem ext4

```
$ mkfs.ext4 /dev/md1
```

Lo montiamo con

```
$ mount /dev/md1 /share
```

E rendiamo la configurazione persistente inserendo questa voce in
/etc/fstab

```
/dev/md1      /share      ext4      defaults      0 0
```



Per default, RAID non dispone di un file di configurazione e quindi dobbiamo salvarlo manualmente. Se questo passaggio non viene seguito, il dispositivo RAID potrebbe essere denominato in un modo diverso da md1. Quindi, è necessario salvare la configurazione affinché persista tra i riavvii; quando avviene un riavvio, viene caricata nel kernel e il RAID viene anch'esso caricato:

```
$ mdadm --detail --scan --verbose >> /etc/mdadm.conf
```



SAMBA

Installazione di SAMBA

Installiamo SAMBA

```
$ sudo apt install samba
```

E aggiungiamo il mountpoint `/share` in `/etc/samba/smb.conf`, specificando che gli utenti facenti parte del gruppo `sambashare` possono accedere alla directory:

```
aboveskip
1 [share]
2     comment = Samba shared directory
3     path = /share
4     read only = no
5     browsable = yes
6     valid users = @sambashare
```



Concludiamo riavviando il servizio

```
$ sudo systemctl restart smb.service
```



Creazione e configurazione dell'utenza

Per dimostrare il funzionamento dei permessi della shared directory, creiamo due utenti *user1* e *user2*, dove solo *user1* farà parte del gruppo *sambashare*:

```
$ sudo useradd -m user1  
$ sudo useradd -m user2  
$ sudo groupadd sambashare  
$ sudo usermod -aG sambashare user1
```



Avremo quindi

```
$ groups user1  
user1 : user1 users sambashare  
$ groups user2  
user2 : user2 users
```



Sistema di alert

Lo script vero e proprio `/etc/cron.hourly/mdmon` viene eseguito ogni ora da *cron*. Questo script legge un altro file di testo `/home/admin/fault_msg` dove scriveremo il messaggio che **mdmon** invierà al destinatario. Un file di log `/var/log/mdmon.log` viene creato, dove si tiene traccia dei *check* che esegue *mdmon*. Il *check* funziona controllando l'output del comando `mdadm --detail <dispositivo md>`, controllando con `awk` se lo stato è **degraded**. In questo caso, viene subito inviata una mail con il comando `mail` di `gnumail`.



checkmd.sh

```
aboveskip
1      #!/bin/bash
2
3      log_file="/var/log/mdmon.log"
4      fault_msg="/home/admin/fault_msg"
5      recipient="example@mail.com"
6
7      echo "$(date):" >> $log_file
8
9      for i in /dev/md/*
10     do
11         state=$(sudo mdadm --detail $i | grep State | head -n 1 |
12         awk '{print $NF}')
13         if [[ $state = "degraded" ]]
14         then
15             mail -s "Degraded RAID partition ${i}" $recipient <
16             $fault_msg
17             echo "${i} faulty. Sent notification to administrator."
18             >> $log_file
19         else
20             echo "${i} ok." >> $log_file
21         fi
22     done
23     echo "" >> $log_file
```



aboveskip

1 This is an automated notification.

2

3 Dear Administrator,

4 This message was sent because one of the RAID partitions is in a
degraded state.

5 It is suggested to intervene.

6

7 Sincerely,

8 The automated notification system



Installazione e configurazione di Postfix

Installiamo Postfix con

```
$ sudo apt install mailutils
```



Configuriamo Postfix aggiungendo delle opzioni per l'autenticazione e il nostro relayhost al file `/etc/postfix/main.cf`:

```
aboveskip
1 smtpd_relay_restrictions = permit_mynetworks
    permit_sasl_authenticated defer_unauth_destination
2 myhostname = labbox.org
3 alias_maps = hash:/etc/aliases
4 alias_database = hash:/etc/aliases
5 myorigin = /etc/mailname
6 mydestination = localhost.labbox.org, , localhost
7 relayhost = [smtp.gmail.com]:587
8 mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
9 mailbox_size_limit = 0
10 recipient_delimiter = +
11 inet_interfaces = all
12 inet_protocols = all
13
14 smtp_sasl_auth_enable = yes
15 smtp_sasl_security_options = noanonymous
16 smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
17 smtp_tls_security_level = encrypt
18 smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```



Creiamo il file delle credenziali `/etc/postfix/sasl/sasl_passwd` e scriviamo dentro

```
[smtp.gmail.com]:587 example@email.com:<tua password>
```

Con `postmap` possiamo convertire il file appena creato in un database

```
$ sudo postmap /etc/postfix/sasl/sasl_passwd
```

Questo creerà un file `sasl_passwd.db` nella stessa directory.



Cambiamo i permessi del file per permettere solo all'utente *root* di leggere e scrivere

```
$ sudo chown root:root /etc/postfix/sasl/sasl_passwd
```

```
$ sudo chmod 600 /etc/postfix/sasl/sasl_passwd
```

Infine riavviamo il servizio Postfix

```
$ sudo systemctl restart postfix
```



Bisognerà poi accedere al **pannello di controllo dell'account Google** per permettere ad "app meno sicure" di eseguire l'accesso.



Conclusioni

Lo script funziona come previsto: nel caso di un malfunzionamento dei dischi **md** verrà inviata una mail all'indirizzo specificato nel file di configurazione e scritta una voce nel file di log. In caso contrario, cron continuerà ad eseguire lo script indefinitivamente, continuando a scrivere log sul file apposito.






Posta in arrivo

| | | | | | |
|--------------------------|----------------------|-----------------------|----|--|-------|
| <input type="checkbox"/> | <input type="star"/> | <input type="reply"/> | me | Degraded RAID partition /dev/md/1 - This is an automated notification. ... | 21:34 |
| <input type="checkbox"/> | <input type="star"/> | <input type="reply"/> | me | Degraded RAID partition /dev/md/0 - This is an automated notification. ... | 21:34 |

Degraded RAID partition /dev/md/1

Posta in arrivo x





root
a me ▾

21:34 (0 minuti fa) ☆ ↶ ⋮

This is an automated notification.

Dear Administrator,

This message was sent because one of the RAID partitions is in a degraded state.
It is suggested to intervene.

Sincerely,
The automated notification system



```
Sun Nov 26 09:17:01 PM CET 2023:
```

```
/dev/md/0 ok.
```

```
/dev/md/1 ok.
```

```
Sun Nov 26 09:32:48 PM CET 2023:
```

```
/dev/md/0 ok.
```

```
/dev/md/1 ok.
```

```
Sun Nov 26 09:34:02 PM CET 2023:
```

```
/dev/md/0 faulty. Sent notification to administrator.
```

```
/dev/md/1 faulty. Sent notification to administrator.
```



Grazie mille

