syslog-ng

come gestire i log nel 2007

Mezzora d'amicizia - 07 Marzo 2007

http://www.linuxvar.it/

syslog-ng

come gestire i log nel 2007

- che problema risolve
- che alternative ci sono
- perche` ce n'e` bisogno
- come si usa (brevemente)
- alcune acrobazie con i log

che problema risolve

Abbiamo bisogno di un sistema per gestire i messaggi della macchina e dei vari software che ci girano sopra

che alternative ci sono

- metalog
  - moderno come syslog-ng ma un po` sempliciotto
- sysklogd
  - vecchio (19??)
  - poco flessibile (file di conf classico, sintassi limitata, solo un paio di opzioni)
  - non evolve piu` (il codice e` difficile da mantenere) dal 2001

perche` ce n'e` bisogno

- e` il piu` featureful
- e` il piu` flessibile
- da meno grattacapi degli altri

come si usa (brevemente)

- options
  - permessi di creazione file e directory
- log
  - source (leggi da:)
  - filter (seleziona:) [facoltativo]
  - destination (scrivi in:)

esempio di configurazione

Configurazione semplice per avere i log separati per data

```
source s_all {
    internal();
    unix-stream(“/dev/log”);
    pipe(“/proc/kmsg” log_prefix(“kernel: “));
    fifo(“/var/log/apache/syslog.fifo”);
};

destination df_logsplit {
    file("/var/log/logsplit/$YEAR/$MONTH/syslog.$DAY");
};

destination df_tmp_current {
    file("/tmp/current_syslog");
};

# mettiamo assieme le parti e otteniamo una direttiva “log”
log {
        source(s_all);
        destination(df_logsplit);
        destination(df_tmp_current);
};
```

source

Con questa direttiva, specifichiamo da dove leggere i messaggi di log

```
source nome_sorgente {
    # messaggi “interni” (di syslog-ng stesso)
    internal();

    # socket unix
    unix-stream(“/dev/log”);

    # messaggi da sistemi remoti su udp e tcp
    udp(ip(192.168.1.12) port(514));
    tcp(ip(0.0.0.0) port(514) max-connections(10));    # opzioni
    # ... ma anche tcp6() e udp6()

    # pipe e fifo (stessa cosa) (man mkfifo(1))
    pipe(“/proc/kmsg” log_prefix(“kernel: “));     # opzione log_prefix
    fifo(“/var/log/apache/syslog.fifo”);
};
```

filter

Con filter, definiamo dei filtri da applicare successivamente ai log

```
# i filtri possono usare l'algebra booleana
filter nome_filtro {
    (
        # si puo` filtrare sulla base di facility e level.. bella novita` :)
        facility(auth, mail)
            and level(warn, error)
    ) or (
        # si puo` filtrare sulla base di espressioni regolari..
        match(".+\<250 Ok\>.+")
            # .. sulla provenienza del messaggio (da un certo host o programma)
            and host("^client_[0-9a-f]+.linuxvar.it$")
            and program("postfix")
    )
};

filter un_altro_filtro {
    facility(daemon)
        and not level(debug)
};
```

destination

Con destination gli diciamo dove andare a loggare

```
destination nome_destinazione {
    # scriviamo su file, pipe, socket, ... anche usando le macro
    file("/path/to/$MACRO/dir/$ALTRA_MACRO/..." owner(root);
    pipe("/path/to/pipe");
    tcp6("2ffe::666:40b3:1" port(9999));

    # mandiamo i log in stdin ad un programma
    program("/usr/local/bin/mysql_log");

    # mandiamo un messaggio all'utente loggato (non supporta i template)
    usertty(buccia);

    # template() definisce il formato del log
    template("$HOUR:$MIN:$SEC $TZ $HOST [$LEVEL] $MSG $MSG\n");
};
```

macros

Vengono espanse nelle stringhe di destinazione; si usano cosi`:

```
file("/var/log/$HOST_FROM/$YEAR/$MONTH/syslog_$DAY");
```

$FACILITY	mail, auth, daemon, ...
$LEVEL	debug, info, warn, error, ...
$YEAR	2007
$MONTH	02
$DAY	26
$WEEKDAY	Mon
$HOUR	17
$MIN	38
$SEC	52
$UNIXTIME	1172507930
$HOST	lothlorien
$HOST_FROM	client_02.linuxvar.it
$PROGRAM	postfix
$PID	1178

... e molte altre !
(la data di esempio e` Lun Feb 26 17:38:52 CET 2007)

risultato

- abbiamo i log ben organizzati
- un balzo in avanti rispetto alla sintassi del sysklogd “standard”
- gestiamo facilmente il log via rete
- diciamo addio a logrotate & C. :)

integrazione con apache

facciamo loggare apache su una fifo (che va creata a mano), e facciamo leggere il syslog-ng da questa ..

```
# mkfifo /var/log/apache/syslog.fifo
```

/etc/apache/httpd.conf:

```
    LogFormat "%a:%A/%{Host}i \"%r\" \"%{User-agent}i\" %s %Bb %Ts" syslogfifo
    CustomLog /var/log/apache/syslog.fifo syslogfifo
    ErrorLog syslog
```

/etc/syslog-ng/syslog-ng.conf:

```
    source s_apache {
        fifo("/var/log/apache/syslog.fifo");
    };
    destination d_apache {
        file("/var/log/apache/$YEAR/$MONTH/access.log.$DAY");
        format("$ISODATE $PROGRAM[$PID]: $MSG");
    };

    log { source(s_apache); destination(d_apache); };
```

integrazione con MySQL [1]

C'e` chi fa salvare i log su mysql invece che su file (esistono anche interfacce web per cercare e lavorare con i log salvati in mysql), per fare cio` basta impostare una direttiva del tipo:

```
destination d_mysql {
    pipe("/tmp/mysql.pipe"
        template("INSERT INTO logs (host, facility, priority, level, tag,
            date, time, program, msg) VALUES ( '$HOST', '$FACILITY',
            '$PRIORITY', '$LEVEL', '$TAG', '$YEAR-$MONTH-$DAY',
            '$HOUR:$MIN:$SEC', '$PROGRAM', '$MSG' );\n")
        template-escape(yes)
    );
};

log {
    source(...);
    destination(d_mysql);
};
```

integrazione con MySQL [2]

E poi ovviamente ci vuole qualcosa che mandi la pipe su MySQL :)

```
#!/bin/sh
if [ -e /tmp/mysql.pipe ]; then
    while [ -e /tmp/mysql.pipe ]
    do
        mysql -u root –password=*** syslog </tmp/syslog.pipe
    done
else
    mkfifo /tmp/mysql.pipe
    exec $0 $*
fi
```