

# 1 Il server log

Lo scopo di un server di log è quello di raccogliere log da altre macchine e raggrupparli in un unico posto.

## 1.1 Dockerfile

Il server log rsyslog verrà implementato senza usare un immagine docker pre-compilata, installando le componenti manualmente.

```
1 FROM ubuntu:21.10
2
3 RUN echo -e "\t\tUpdating system and installing rsyslog" \
4 && apt-get update \
5 && apt-get install --no-install-recommends -y rsyslog \
6 && apt-get clean \
7 && rm -rf /var/lib/apt/lists/*
8
9 RUN echo -e "\t\tCopying Config"
10 COPY Contents/rsyslog.conf /etc/rsyslog.conf
11
12 ENTRYPOINT ["rsyslogd", "-n"]
```

Listing 1: Dockerfile Rsyslog

### 1.1.1 Analisi Dockerfile

Partiamo caricando un immagine di *ubuntu:21.10* da *Docker Hub*, su questa immagine, dopo aver aggiornato le sorgenti, installiamo il server *rsyslog*.

Una volta installato il server facciamo pulizia del garbage creato dall'installazione, carichiamo il config di *rsyslog* e impostiamo come punto di partenza il comando *rsyslogd -n*.

## 1.2 Servizio docker compose

```
1 Syslogserver:
2   build: Dockerfiles/rsyslog/.
3   image: syslogserver
4   container_name: Syslog
5   volumes:
6     - "[PERCORSO COMPLETO CARTELLA LOG LOCALE]:/var/log"
7   ports:
8     - 514:514
9     - 514:514/udp
10   cap_add:
11     - SYSLOG
```

Listing 2: Rsyslog Docker Compose

La prima riga indica il nome univoco del servizio.

Riga 2 è opzionale e indica il percorso in cui effettuare la build dell'immagine, se questa non è

presente.

Riga 3 indica il nome dell'immagine. Se non è presente in locale verrà o presa dalla repo remota o buildata (se è presente l'istruzione build).

Riga 4 indica un nickname per il servizio.

Riga 6 mappa una directory locale in cui salvare i log alla directory remota */var/log*. Su questa cartella locale saranno salvati i log ricevuti dalle macchine

Riga 8 e 9 Aprono la port 514 in TCP e UDP per consentire al server di ricevere i log.

Se si intende usare solo uno dei protocolli (TCP o UDP), la porta relativa all'altro protocollo va eliminata.

Riga 11 specifica che il server ha bisogno di permessi aggiuntivi di tipo *SYSLOG*, per info su questi permessi consultare *man 7 capabilities*.

### 1.3 Configurazione

```
1 # Commentare per disabilitare UDP logging
2 #module(load="imudp")
3 #input(type="imudp" port="514")
4
5 # Commentare per disabilitare TCP logging
6 module(load="imtcp")
7 input(type="imtcp" port="514")
8
9 # Template nome file log remoto
10 template(name="RemoteDirTemplate" type="string" string="/var/log/remote/%$year
    %/$$Month%/$$Day%/$$Hour%-%APP-NAME%.log")
11 # Regole di log
12 if ($source != "localhost") then {
13     action(type="omfile" dynaFile="RemoteDirTemplate")
14 }
```

Listing 3: File di configurazione Rsyslog

In questa configurazione abilitiamo solo la versione TCP del servizio di log, per abilitare anche UDP è necessario rimuovere il commento dalle righe 2 e 3.

Alla riga 3 e 7 definiamo le porte per il servizio di log rispettivamente UDP e TCP, queste porte possono essere modificate ma DEVONO corrispondere a quelle definite alle righe 8 e 9 nella sottosezione 1.1.

La riga 10 definisce il template per il nome dei file su cui salvare i log remoti, verrà analizzata a parte nella sottosottosezione 1.3.1.

Le righe 12 e 13 applicano il template definito alla riga 10 solo ai log provenienti da sorgenti esterne, ovvero con l'attributo *source* diverso da *localhost*.

#### 1.3.1 Template nome file

Il template per il nome di file è il seguente:

*/var/log/remote/%\$year%/\$\$Month%/\$\$Day%/\$\$Hour%-%APP-NAME%.log*

Possiamo suddividere il template in 3 parti:

1. `/var/log/remote/`

- Percorso FISSO della cartella root su cui salvare i log.

2. `%%$year%%/%%$Month%%/%%$Day%%/`

- Percorso VARIABILE della cartella finale su cui salvare i log.
- Dipende da:
  - `$year`
  - `$Month`
  - `$Day`

3. `%%$Hour%%-%%APP-NAME%%.log`

- Nome del file in cui salvare i log
- Dipende da:
  - – `$Hour`
  - `$APP-NAME`
    - \* Identificativo del programma remoto da cui sono originati i log
    - \* Può essere sostituito con `$fromhost`, l'hostname della sorgente (o indirizzo ip se DNS non disponibile).

Se, ad esempio, la macchina con il programma *pippo* generasse un log il 01/01/1970 alle ore 00:05, il percorso finale verrebbe ad essere:

`/var/log/remote/1970/01/01-pippo.log`

È stato scelto questo ordine delle variabili arbitrariamente, raccogliere i log per data e ora e, in seguito per macchina, consente di avere una migliore visione di insieme.

Altre alternative valide sarebbero potute essere:

- `/var/log/remote/%%APP-NAME%%-%%$year%%/%%$Month%%/%%$Day%%/%%$Hour%%.log`
  - Suddivide prima per macchina e, successivamente, per data.
  - Fornisce una migliore visione temporale per le singole macchine ma peggiore visione di insieme sul sistema completo.
- `/var/log/remote/%%$year%%/%%$Month%%/%%$Day%%/%%$Hour%%.log`
  - Ignora l'attributo `APP-NAME`, raccoglie i log di tutte le macchine nello stesso file, suddivisi per data.
  - Visione d'insieme sul sistema completo MA rischio di generare file molto pesanti e di difficile lettura.
- Qualunque altra configurazione con le variabili presenti sopra e altre dalla [documentazione ufficiale rsynclog](#)

## 1.4 Ricerca di un file di log

Usando il template definito sopra, per cercare un file di log si può usare il seguente script bash:

```
1 #!/bin/sh
2
3 HOST="WS1"
4 YEAR=""
5 MONTH=""
6 DAY=""
7
8 LIMIT="5" # Numero massimo di elementi da visualizzare
9 SEPARATOR="/" # / su sistemi base Unix o Darwin, \ su sistemi base MS-DOS
10 BASE_DIR="./remote" # Directory di partenza
11
12 if [ -z "$HOST" ]; then
13     HOST=".*"
14 fi
15
16 if [ -z "$YEAR" ]; then
17     YEAR="[0-9][0-9][0-9][0-9]"
18 fi
19
20 if [ -z "$MONTH" ]; then
21     MONTH="[0-9][0-9]"
22 fi
23
24 if [ -z "$DAY" ]; then
25     DAY="[0-9][0-9]"
26 fi
27
28 if [ -z "$BASE_DIR" ]; then
29     BASE_DIR="."
30 fi
31
32 REGEX=".*$SEPARATOR$YEAR$SEPARATOR$MONTH$SEPARATOR$DAY$SEPARATOR[0-9][0-9]-$HOST
33     .log"
34
35 if [ -z "$LIMIT" ]; then
36     find $BASE_DIR -regex $REGEX
37 else
38     find $BASE_DIR -regex $REGEX | head -$LIMIT
39 fi
```

Listing 4: Script per ricercare log dati specifici parametri