rsyslog : Collecte, traitement et forwarding de logs.

/etc/rsyslog.conf : peut ocntenir des includes vers d'autres fichiers de conf

# commentaire

/\*

Commentaire

\*/

formats utilisés par rsyslog :

\*basic : format sysklogd : format le plus utilisé pour des choses basiques : une ligne

\*advanced : = rainerscript format : plus d'une ligne meilleur et plus précis.

ex : mail.err action()

\*obsolete legacy : format obsolète : tout ce qui commence par $ : permet juste la

rétrocompatibilité

format basic :

Règles : 2 champs : selecteur et action

selecteur :

une facilité :

auth

authpriv

cron

daemon

ftp

kern

lpr

mail

mark

news

security (déprécié remplacé par auth)

syslog

user

uucp

local0 - local7

Chaque service peut logger sur la facilité qu'elle veut, c'est juste une convention.

et une priorité :

debug

info

notice

warning

warn (deprecie remplace par warning)

err

error (deprecie remplace par err)

crit

alert

emerg

panic (deprecie remplace par emerg)

Toute priorité et supérieure seront loggées selon l'action.

Utilisation de \* ou none possible

mail,user.info possible

mail.info; notice.warning... lu de gauche a droite, les règles lues overide celles d'avant.

mail.=info uniquement cette priorité (normalement celle la plus celles au dessus)

mail.\*,mail.!info : tout sauf info et supérieurs //attention ! quand ! ou !=, il faut un selecteur sans ! ou != avant.

mail.\*,mail.!=info : tout sauf info

ligne 1 \ surtout mettre un espace avant \

ligne 2

Action :

Fichier regulier : logs dans un fichier standards

Tubes nommés (FIFOs)

terminal/console

remote machine

List of users

examples :

kern.info;kern.!err : va uniquement logger info, notice et warning

\*.=info;mail.none : tous les logs info, sauf ceux provenant de mail.

\*.emerg \* : ecrit tous les messages d urgences à tous les utilisateurs.

\*.alert root,joey : ecrit tous les messages alertes et superieurs sur le terminal de root et joey.

\*.\* @host : remote all logs messages.

input modules -> rsyslog with rules -> actions if match (via modules output)

ruleset(name="rulesetname"){ //regroupement de règles

action(type="omfile" file "/path")

action(type=...)

...

}

les templates modifient et formattent l'output

templates : list, subtree, string, plugin

type list :

template(name="toto" type="list"){

constant(value="Syslog MSG is: '")

property(name="msg")

property(name="time" dateformat="date-year") //ou date.day ou date.month ou date.inUTC

constant(value="', ")

property(name="timereported" dateFormat="rfc3339" caseConversion="lower")

constant(value="\n" outname="toto") //intégration dans un format structuré

constant(outname="@version" value="1" format="jsonf") //genere : “@version”:”1”

}

type string : %variable% le reste constant text

template(name="tpl3" type="string"

string="%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-lf%\n"

)

Spécial JSON :

template(name="outfmt" type="list" option.jsonf="on") {

property(outname="@timestamp" name="timereported" dateFormat="rfc3339" format="jsonf")

property(outname="host" name="hostname" format="jsonf")

property(outname="severity" name="syslogseverity-text" caseConversion="upper" format="jsonf")

property(outname="facility" name="syslogfacility-text" format="jsonf")

property(outname="syslog-tag" name="syslogtag" format="jsonf")

property(outname="source" name="app-name" format="jsonf")

property(outname="message" name="msg" format="jsonf")

}