10. GESTIÓN DE LOGS

10.1.- Introducción

Tanto los procesos como el propio sistema operativo registran los eventos que van sucediendo en unos archivos llamados *logs* que por convención en Linux, se almacenan en */var/log*.

Existe un protocolo llamado Syslog en el que se basa el sistema de registro de *logs* que tiene el sistema, tradicionalmente gestionado por el demonio *rsyslog*. En CentOS7, los mensajes del *syslog* se gestionan con dos servicios s*ystemd-journald* y *rsyslog*.

El servicio de *rsyslog* ordena los mensajes de Syslog por tipo o *facility* y prioridad y los escribe en los archivos de */var/log*. Los archivos más importantes son:

- */var/log/messages*: archivo donde se registran casi todos los eventos.
- */var/log/secure*: mensajes de autenticación y errores. Registro del comando *sudo*.
- */var//log/maillog*: mensajes relativos al servidor de correo.
- */var/log/cron*: mensajes relativos a las tareas periódicas del sistema.
- */var/log/boot.log*: mensajes relativos al arranque del sistema.

El servicio de *systemd-journald* proporciona una gestión de *logs* mejorada escribiendo los mensajes en un *journal* estructurado de eventos.

10.2.- Syslog

Casi todos los programas del sistema utilizan el protocolo Syslog para registrar los eventos. Cada mensaje tiene un tipo de mensaje, llamado *facility* y una prioridad, llamada *severity*. En base a estas cosas, el servicio de *rsyslog* determina que hacer con el mensaje según su configuración.

Los tipos de mensajes o facility del sistema, que se pueden ver en la sección 5 de las páginas del man de *rsyslog.conf*, o en la documentación de */usr/share/doc/rsyslog-*/manual.html*, son: *auth* o *security, authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user, uucp and local0, …, local7*. El programa en concreto generará los mensajes usando una de estas *facility*.

Los mensajes de *logs* escritos por rsyslog se van escribiendo según suceden en los archivos de tal forma que los más antiguos están al principio del archivo y los más recientes al final. Es por eso que se suele utilizar el comando *tail -f <archivo_log>* para ir viendo los últimos mensajes según se producen quedando el terminal a la espera de ir mostrando lo que se va escribiendo en el archivo.

Cada línea de los archivos de log tiene el formato:

<timestamp> <host> <proceso>: <texto_mensaje>

donde:

<timestamp>: cuando se produjo el mensaje con formato:

<mes> <dia> <hora>:<minuto>:<segundo>

<host>: sistema que produjo el mensaje.

<proceso>: que produjo el mensaje.

<texto_mensaje>: el contenido del mensaje.

Se puede enviar un mensaje a Syslog utilizando el comando *logger* con la sintaxis:

logger -p <facility>.<severity> “<texto_mensaje>”

por defecto si no lo indico con *-p* utiliza *user.notice*.

10.2.1.- Prioridades de Syslog

Las prioridades (*severity*) están estandarizadas y son:

- *0, emerg*: el sistema no se puede utilizar, muy grave.
- *1, alert*: se debe realizar una acción de forma inmediata.
- *2, crit*: evento crítico.
- *3, err*: error no crítico.
- *4, warning*: aviso.
- *5, notice*: evento relativamente importante.
- *6, info*: información.
- *7, debug*: información de debug, muy poco relevante salvo en caso de análisis extenso de un problema.

Cuando Syslog tiene en su configuración que debe registrar los mensajes de una determinada *facility*, con una prioridad, registrará todos los mensajes generados de esa *facility* que tengan una prioridad igual o superior a la que aparece en la configuración.

10.2.2.- Configuración de Syslog

En los archivos de /*etc/rsyslog.conf* y los archivos con extensión *.conf* del directorio /etc/rsyslog.d se encuentra la configuración del servicio *rsyslog*.

En la sección de reglas que tienen, están las directivas donde se indica qué mensajes se registran y dónde. En cada línea, la parte izquierda indica el tipo y la importancia con el formato: *facility.severity* con el que debe encajar el evento. Si aparece el carácter *, indica “cualquier cosa”. En la parte de la derecha de la línea, se indica en qué archivo será registrado el mensaje del evento.

Un mismo mensaje puede aparecer en distintos archivos de *log*.

La recomendación a la hora de añadir configuraciones extra a las que ya están establecida por el sistema operativo y los distintos programas, es añadirla en los archivos de */etc/rsyslog.d* ya que en una actualización del sistema operativo, no se perderían al sobrescribirse el archivo */etc/rsyslog.conf*.

Tras cada cambio en la configuración de Syslog, hay que reiniciar el servicio *rsyslog*.

10.2.3.- Rotado de archivos de log

Existe una utilidad llamada *logrotate* que evita que los archivos de log crezcan de forma desmesurada haciendo que roten. Cuando un archivo es rotado, se renombra con una extensión que indique cuando fue rotado y se cree otro archivo nuevo listo para ser usado. De forma automática, el sistema va eliminando y/o comprimiendo los archivos antiguos en base a unas políticas.

Existe en el sistema un trabajo periódico que ejecuta el programa *logrotate* diariamente para ver que archivos deben ser rotados.

Cuando un paquete se instala, se añade en el directorio */etc/logrotate.d/* su archivo de configuración *logrotate* encargado de la gestión de los logs.

En el archivo */etc/logrotate.conf*, se configuran las políticas de rotación de logs comunes a todos los logs. En cada archivo de configuración dentro de */etc/logrotate.d*, se pueden sobrescribir las políticas por defecto, redefiniendo el valor para la variable de configuración deseada.

Al no ser *logrotate* un demonio del sistema, cuando se realicen cambios en sus archivos de configuración, no será necesario realizar ningún reinicio. La siguiente vez que se ejecute, ya lo hará con dichos cambios. Si se desea testear *logrotate* tras algún cambio en su configuración, se puede ejecutar el comando: *logrotate -vf /etc/logrotate.conf*.

10.3.- Journald

Journald almacena los datos de log de forma estructurada en un archivo binario indexado. El archivo incluye información extra sobre el evento. La información se almacena por defecto en */run/log/journal* con lo que se pierde al reiniciar el sistema. Es importante cambiar este comportamiento y hacer el *journal* persistente.

El demonio de *systemd-journald* recoge:

- mensajes que provienen del kernel.
- mensajes del arranque del sistema.
- salida estándar y de error de los demonios cuando arrancan y paran.

El comando *journalctl* muestra el *journal* completo del sistema, desde la entrada más antigua a la más nueva.

Opciones comunes:

- *-n <número>:* muestra sólo el número dado de las últimas entradas, si no proporcionamos el número, muestra 10.
- *-p <prioridad>*: filtra por prioridad siendo estas prioridades las mismas que las *severity* de Syslog.
- *-f*: similar a tail -f, deja abierto el terminal y va mostrando lo que se va escribiendo en el journal.
- *-b*: información sólo del último reinicio.
- *--since <año-mes-dia hora:min:seg>, --until <año-mes-dia hora:min:seg>*: filtra los mensajes según el intervalo de tiempo dado. Si se omite la fecha, entiende que es el día actual y si se omite la hora, toma las 00:00:00.
- *-o verbose*: muestra información extensa.

Podemos ver todas las posibles consultas a Journald en la sección 7 del man de *systemd.journald-fields*.

10.3.1.- Hacer persistente el Journald

Si el directorio */var/log/journal* existe con la configuración apropiada, *journald* utilizará este directorio como base para el *journal* en lugar de */run/log/journal*. Aun así, por defecto hay un rotado mensual del archivo y además no se le permite alcanzar más del 10% del sistema de archivos donde reside o dejar al menos un 15% libre. Para modificar estos valores, en el archivo */etc/systemd/journal.conf*.

Los pasos a seguir para hacer persistente el *journal* entre reinicios son:

1. Crear el directorio */var/log/journal* con permisos 2755.
2. Poner de usuario propietario a *root* y grupo propietario a *systemd-journal* en el directorio anterior.
3. Reiniciar el servicio *journald* de la forma: *killall -USR1 systemd-journald*

10.4.- Caso práctico

En **server1**, añadir una configuración en el Rsyslog que almacene los mensajes que se produzcan con prioridad *debug* en el archivo */var/log/mensajes-debug*.

Se va a hacer persistente el Journald tanto en **central** como en **server1**

RESOLUCIÓN

- En **server1** configuramos Rsyslog para que los mensajes con *severity* de *debug* se almacenen en */var/log/mensajes-debug*:

[root@server1 ~]# echo “*.debug /var/log/mensajes-debug” > /etc/rsyslog.d/debug.conf

[root@server1 ~]# systemctl restart rsyslog

- Verificamos la nueva configuración enviando un mensaje al Syslog desde un terminal y desde otro terminal abrimos el archivo */var/log/mensajes-debug*:

[admin@server1 ~]$ logger -p user.debug “Mensaje de prueba de debug”

[root@server1 ~]# tail -f /var/log/mensajes-debug

[root@server1 ~]# tail -f /var/log/mensajes-debug

Sep 17 18:09:42 server1 rsyslogd: [origin software="rsyslogd" swVersion="7.4.7" x-pid="3512" x-info="http://www.rsyslog.com"] start

Sep 17 18:09:42 server1 systemd: Stopping System Logging Service...

Sep 17 18:09:42 server1 systemd: Starting System Logging Service...

Sep 17 18:09:42 server1 systemd: Started System Logging Service.

Sep 17 18:09:42 server1 polkitd[611]: Unregistered Authentication Agent for unix-process:3505:2530773 (system bus name :1.62, object path /org/freedesktop/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8) (disconnected from bus)

Sep 17 18:10:30 server1 root: “Mensaje de prueba de debug”

- Hacemos persistente el *journal* en **central**:

[root@central ~]# mkdir -m 2755 /var/log/journal

[root@central ~]# chown :systemd-journal /var/log/journal

[root@central ~]# killall -USR1 systemd-journald

- Hacemos persistente el *journal* en **server1**:

[root@server1 ~]# mkdir -m 2755 /var/log/journal

[root@server1 ~]# chown :systemd-journal /var/log/journal

[root@server1 ~]# killall -USR1 systemd-journald