# Monitorizar ficheros log con check_mk - Oracle alert log.

## *Introducción.*

Siguiendo con la guía de artículos de funcionalidades de check_mk, aplicadas en este caso a Oracle, vamos a analizar este interesante plugin que nos proporciona check_mk para la motorización de ficheros log. La instalación y configuración del plugin es muy sencilla y está bien descrita en la documentación correspondiente de check_mk

## *Instalación.*

Continuando con el procedimiento habitual de uso de plugins de check_mk que no están integrados directamente en el agente, tendremos que copiar el plugin correspondiente en el directorio de plugins del servidor destino. En nuestro caso, una instalación de omddistro, el plugin a copiar está en :

*/opt/omd/versions/VersionOMD/share/check_mk/agents/plugins/mk_logwatch*

En una instalación normal de check_mk debería estar en el directorio de instalación dentro de *./share/check_mk/agents/plugins/*

En nuestro servidor destino que debe tener instalado el agente lo copiaremos (la ruta habitual) en el directorio correspondiente de plugins y lo hacemos **ejecutable.**

*/usr/lib/check_mk_agent/plugins/*

Si no tienes una instalación estándar por paquetes del agente de chech_mk en tu Linux o quieres saber más del tema en este artículo se explicaba más en detalle como localizar las ubicaciones correctas.

## *Configuración.*

La configuración es muy sencilla. Debemos crear  en el agente un fichero en nuestra ubicación por defecto para estos temas de check_mk. El fichero debe llamarse:

*/etc/check_mk/logwatch.cfg*

Y debe tener este formato:

```
/dir/dir/archivo1.log
 C Cadena a buscar e indicar Critical
 W Cadena a buscar e indicar Warning

/dir/dir/archivo1.log
 C Cadena a buscar e indicar Critical
 W Cadena a buscar e indicar Warning
```

**Formato**

- Una entrada por cada fichero log en el que buscar.
- Debajo de esta una línea por cada estado a señalar a buscar (Warning, Crítical,...) seguida de la cadena de texto, teniendo en cuenta que:
    - Hay que dejar un espacio en la línea antes del estado (C, W, …) si no, no funcionará pero tampoco nos dará ningún aviso ni error.
    - Entre el estado y la cadena de Texto a buscar para este hay que dejar también una línea.
    - Los posibles estados son: (C: Critical, W: Warning, O: OK, I: Ignore)
- Una línea de separación para indicar el siguiente archivelog

Repito, antes del estado a señalar (C, W,...) hay que dejar un espacio en blanco en la línea.

**Uso de expresiones regulares.**

Se pueden usar expresiones regulares tanto en las cadenas de texto a buscar como en la definición de archivos a buscar. Ver ejemplos en documentación.

## *Funcionamiento.*

El plugin tiene un funcionamiento muy lógico que hay que conocer para no volvernos locos inicialmente. La primera vez que lo configuramos e inventariamos el host para añadirlo a cmk, ignorará todos los errores aunque los encuentre. Esto tiene la lógica de evitar una avalancha de errores nada más configurarlo además de que necesita mantener un "índice" de líneas ya leídas en los ficheros log para que no alerte del mismo error una y otra vez. La primera ver que lo ejecutemos creará esos índices en un fichero en /etc/check_mk/logwatch.state con entradas del tipo:

```
/var/log/messages|1701|3375624
```

En los chequeos posteriores analizará solo las nuevas líneas del log. Es posible que al chequear encuentre varias coincidencias Warning y Crítical. Indicará siempre la de mayor prioridad (Critical) pero se traerá las líneas del log de todas las coincidencias para verlas desde el propio Nagios como veremos a continuación.

## *Probando el Plugin.*

Antes de ponernos a configurar es mejor como siempre realizar las pruebas correspondientes. Sabiendo además que la primera vez que lo ejecutemos no nos alertará de nada... mejor probar en un fichero propio la detección de errores.

- Crearemos un fichero p.e. /var/log/test.log que iremos editando para nuestras pruebas
- Configuramos en nuestro fichero d `/etc/check_mk/logwatch.cfg algún ejemplo:`

```
/var/log/test.log
 C Error XYZ
 W Error XXX
```

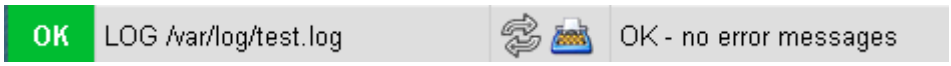
Reinventaríamos nuestro host con cmk y veremos que nos encuentra un nuevo item de “logwatch”

```
# cmk -II servidorX
..
logwatch          1 new checks
…..
```

Si no nos lo encuentra podemos chequear el host en modo debug para intentar determinar cual es el problema. Por ejemplo, si el fichero log que configuramos no existe:

```
cmk -d servidorX
……..
[[[/var/log/test.log:missing]]]
……..
```

Recargamos la configuración de check_mk (cmk -R) y ya tendremos nuestro servicio en nuestro host en el interface gráfico de CMK:



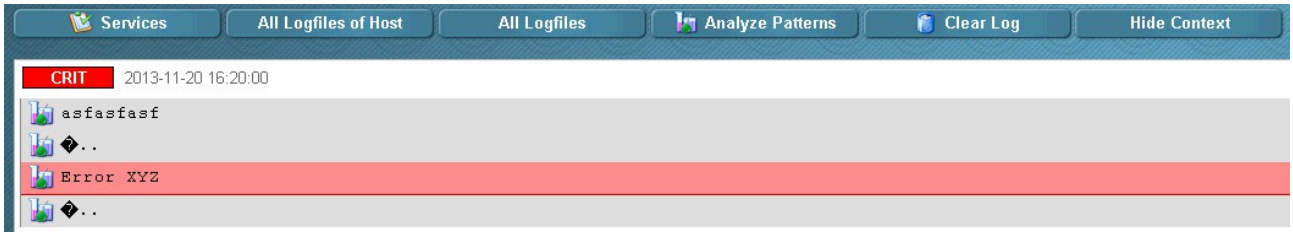

Tenemos el símbolo para volver a chequear directamente que es algo muy cómodo y el símbolo de la máquina de escribir que nos mandará a una página con las líneas de alertas que se trae del fichero log (cuando las tengamos).

Ahora toca probarlo, editamos nuestro fichero /var/log/test.log y añadimos unas líneas con alguna cadena de error a buscar:

```
asfasfasf
asasfasf
Error XYZ
holahola
```

Forzamos el chequeo del servicio y ya vemos resultados:


Si pinchamos en el icono de la máquina de escribir nos abre una página con los errores y opciones varias para estos.



Los símbolos de tubos de ensayo pertenecen a WATO, otro día hablamos de él ya que es una funcionalidad muy interesante pero te obliga más a "casarte" con check_mk.... Vemos que tenemos la opción de Clear Log, es la que usaremos para "limpiar" el log que se ha traído y que vuelva a estado OK hasta que encuentre otro problema.

## *Monitorizar ficheros alertlog de Oracle.*

La configuración de nuestro fichero "/etc/check_mk/logwatch.cfg" será bastarte obvia inicialmente, al menos hasta que vayamos depurándola. Un ejemplo inicial puede ser.

```
/u01/app/oracle/admin/orapro/bdump/alert_orapro.log
 C ORA-
 W Starting up
  I ORA-XXXX
```

**Critical** - > Todos los errores "ORA-"

**Warning** → Cuando inicia Oracle (para enterarnos)

**Ignorar** → Un error ORA concreto.

Podemos monitorizar p.e. los alertlog de todas nuestras instancias y ASM en cada nodo de nuestro RAC:

```
/u01/app/oracle/admin/ora1pro/bdump/alert_ora1pro1.log
 C ORA-
 W Starting up

/u01/app/oracle/admin/ora2pro/bdump/alert_ora2pro1.log
 C ORA-
 W Starting up

/u01/app/oracle/admin/+ASM/bdump/alert_+ASM1.log
 C ORA-
 W Starting up
```

Podriamos resumirlo más si los patrones de busqueda son los mismos para todos los ficheros separando estos en la primera línea con un espacio:

```
/u01/app/oracle/admin/ora1pro/bdump/alert_ora1pro1.log
/u01/app/oracle/admin/ora2pro/bdump/alert_ora2pro1.log
/u01/app/oracle/admin/+ASM/bdump/alert_+ASM1.log
 C ORA-
 W Starting up
```

Según vayamos detectando errores podemos ir filtrando mejor lo que nos interesa. Las posibilidades de ampliación a otros ficheros logs de Oracle (listener, clusterware,...) son también una posibilidad muy interesante.

## *Parámetros.*

En este caso los parámetros que afectan al plugin no se configuran en los ficheros .mk como estamos acostumbrados. Se configuran en los ficheros de configuración logwatch.cfg en cada cliente, inmediatamente después de la línea de cada log en la forma:

```
/var/log/foobar.log maxlines=10000 maxtime=3 overflow=W
```

Los parámetros contemplados básicamente están destinados a evitar timeouts y problemas de rendimiento al analizar logs grandes con crecimiento muy rápido.

Las posibilidades básicamente son:

- **maxlines** → Número máximo de mensajes nuevos en el fichero log que serán analizados en un "turno" de chequeo.
- **maxtime** → El tiempo máximo en segundos que estará analizando nuevas líneas en el fichero log.
- **overflow** → Cuando se sobrepase uno de los valores anteriores se añadirá una línea en el log (página) de check_mk alertando de la situación (crítical). Muy recomendable para enterarnos de que tenemos un log descontrolado.

## *Conclusiones.*

Como siempre check_mk no deja de sorprendernos con los plugins y opciones que tiene. En este caso hemos visto el plugin de monitorizar ficheros logs con ejemplos de Oracle pero en un sistema Linux, con todas las aplicaciones y servicios logeando... las posibilidades son amplias. El plugin además está como siempre muy cuidado y funciona perfectamente. El hecho de que se "traiga" las alertas encontradas para poder verlas desde Nagios es sencillamente genial.