Ingeniería de Servidores. Memoria práctica 3

Monitorización con Zabbix

Elena Cantero Molina, 3ºA

25 de noviembre de 2018

Índice

1. Introducción

En esta práctica vamos a configurar el monitor Zabbix para realizar una prueba de monitorización de los servicios ssh y apache de nuestros sistemas. Para ello, tenemos que instalar y configurar Zabbix adecuadamente siguiendo su documentación *online*.

Vamos a instalar la versión 3.4, en centOs y Ubuntu(16.04), usando la versión 7 y mysql como base de datos.

2. Instalación

El esquema a seguir será el siguiente: en Ubuntu tendremos instalado el servidor, junto con el *front-end* que permite visualizar los datos que se recojan. Monitorizaremos los servicios ssh y apache tanto en Ubuntu como en CentOs, por lo que habrá que instalar el agente de Zabbix en ambos sistemas.

2.1. Ubuntu

Vamos a instalar el servidor zabbix-server, siguiendo la guía que aparece en la página oficial de zabbix. Lo primero es añadir los repositorios necesarios para nuestra versión de Ubuntu (16.04 xenial), ya que los paquetes que vamos a instalar no se encuentran en los repositorios por defecto:

```
# wget https://repo.zabbix.com/zabbix/3.4/ubuntu/
pool/main/z/zabbix-release/zabbix-release_3.4-1+
xenial_all.deb
# dpkg -i zabbix-release_3.4-1+xenial_all.deb
# apt update
```

```
root@ubuntu-elena:~# wget https://repo.zabbix.com/zabbix/3.4/ubuntu/pool/main/z/zabbix-release/zabbi
x-release_3.4-1+xenial_all.deb
--2018-11-22 10:28:56--  https://repo.zabbix.com/zabbix/3.4/ubuntu/pool/main/z/zabbix-release/zabbix
-release_3.4-1+xenial_all.deb
Resolviendo repo.zabbix.com (repo.zabbix.com)... 162.243.159.138
Conectando con repo.zabbix.com (repo.zabbix.com)[162.243.159.138]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 3884 (3,8K) [application/octet-stream]
Grabando a: “zabbix-release_3.4-1+xenial_all.deb”

zabbix-release_3.4-1+xen 100%[=================================>]   3,79K  --.-KB/s    in 0s

2018-11-22 10:28:57 (131 MB/s) - “zabbix-release_3.4-1+xenial_all.deb” guardado [3884/3884]

root@ubuntu-elena:~# dpkg -i zabbix-release_3.4-1+xenial_all.deb
Seleccionando el paquete zabbix-release previamente no seleccionado.
(Leyendo la base de datos ... 61962 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar zabbix-release_3.4-1+xenial_all.deb ...
Desempaquetando zabbix-release (3.4-1+xenial) ...
Configurando zabbix-release (3.4-1+xenial) ...
root@ubuntu-elena:~# apt update
Obj:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu xenial InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu xenial-backports InRelease
Des:5 http://repo.zabbix.com/zabbix/3.4/ubuntu xenial InRelease [7.093 B]
Des:6 http://repo.zabbix.com/zabbix/3.4/ubuntu xenial/main Sources [900 B]
Des:7 http://repo.zabbix.com/zabbix/3.4/ubuntu xenial/main amd64 Packages [2.493 B]
Des:8 http://repo.zabbix.com/zabbix/3.4/ubuntu xenial/main i386 Packages [2.501 B]
Descargados 13,0 kB en 1s (11,1 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 73 paquetes. Ejecute «apt list --upgradable» para verlos.
root@ubuntu-elena:~#
```

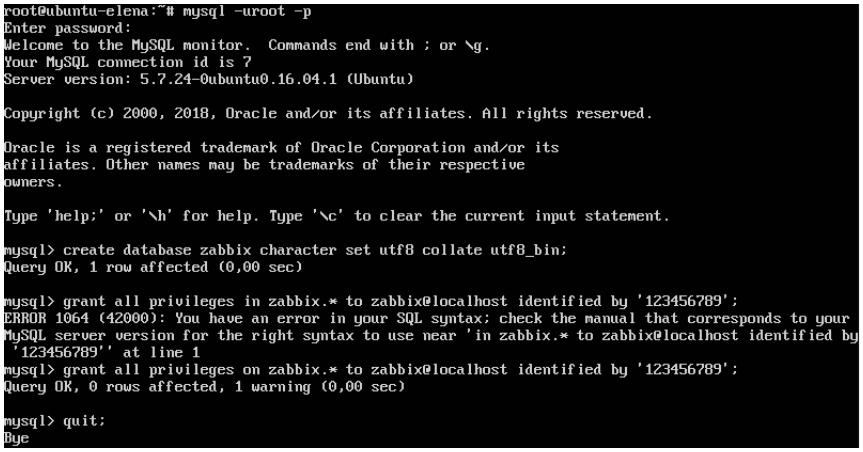
Ahora podemos instalar los paquetes zabbix-server-mysql, zabbix-frontend-php y zabbix-agent con:

```
# apt install zabbix-server-mysql zabbix-frontend
-php zabbix-agent
```

```
root@ubuntu-elena:~# apt install zabbix-server-mysql zabbix-frontend-php zabbix-agent
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  fontconfig-config fonts-dejavu-core fping libcurl3 libfontconfig1 libgd3 libiksemel3 libjbig0
  libjpeg-turbo8 libjpeg8 libltdl7 libmysqlclient20 libodbc1 libopenipmi0 libsensors4 libsnmp-base
  libsnmp30 libssh2-1 libtiff5 libvpx3 libxpm4 php-bcmath php-gd php-ldap php-mbstring php-xml
  php7.0-bcmath php7.0-gd php7.0-ldap php7.0-mbstring php7.0-xml snmpd ttf-dejavu-core
Paquetes sugeridos:
  libgd-tools libmyodbc odbc-postgresql tdsodbc unixodbc-bin lm-sensors snmp-mibs-downloader
  snmptrapd
Se instalarán los siguientes paquetes NUEVOS:
  fontconfig-config fonts-dejavu-core fping libcurl3 libfontconfig1 libgd3 libiksemel3 libjbig0
  libjpeg-turbo8 libjpeg8 libltdl7 libmysqlclient20 libodbc1 libopenipmi0 libsensors4 libsnmp-base
  libsnmp30 libssh2-1 libtiff5 libvpx3 libxpm4 php-bcmath php-gd php-ldap php-mbstring php-xml
  php7.0-bcmath php7.0-gd php7.0-ldap php7.0-mbstring php7.0-xml snmpd ttf-dejavu-core
  zabbix-agent zabbix-frontend-php zabbix-server-mysql
0 actualizados, 36 nuevos se instalarán, 0 para eliminar y 73 no actualizados.
Se necesita descargar 10,0 MB de archivos.
Se utilizarán 42,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

El siguiente paso es crear una base de datos, requerida por el servidor de Zabbix. Seguimos las instrucciones de la documentación para su creación en MySQL:

```
 # mysql -uroot -p
password
mysql> create database zabbix character set utf8
collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@
localhost identified by 'password';
mysql> quit;
```



Una vez creada, debemos cargar en la base de datos la información necesaria para el funcionamiento del servidor, que se ha descargado junto con el paquete. Importamos esta información con la orden:

```
 # zcat /usr/share/doc/zabbix-server-mysql*/create.
 sql.gz  | mysql -uzabbix -p zabbix
```

Ahora que hemos configurado la base de datos, editamos el archivo de configuración del servidor (*/etc/zabbix/zabbix_server.conf*) para permitir la conexión a ella. En concreto, modificamos la siguiente línea:

```
DBPassword=password
```

El front-end de Zabbix nos permite configurar de forma gráfica los hosts que vamos a monitorizar, así como ver la información recogida de dicha información. En primer lugar, escribimos la zona horaria correcta en el archivo de configuración, localizado en */etc/zabbix/apache.conf*. Y descomentamos la siguiente línea:

```
# php_value date.timezone Europe/Riga
```

Y modificamos la zona horaria, que en nuestro caso es Europe/Madrid. Recargamos el el servicio de apache con ”systemctl restart apache2”.

También aprovechamos y abrimos en el firewall el puero por defecto de Zabbix, que es el 10050, con ufw allow 10050. Estamos a disposición de iniciar (y activar) el servicio de zabbix-server y el agente:

```
# systemctl enable zabbix-server zabbix-agent
# systemctl restart zabbix-server zabbix-agent
```

Y ya podemos proceder a la configuración mediante la interfaz gráfica, accediendo desde nuestro navegador a la dirección ”192.168.56.20/zabbix”

El proceso es bastante intuitivo, y está explicado con capturas de pantalla en la propia documentación. Sólo debemos introducir los detalles referentes a la base de datos, y comprobar que todo está correcto.

2.2. CentOs

En CentOs debemos primero añadir los repositorios de forma análoga a como hicimos en Ubuntu, para luego instalar e iniciar el agente.

```
# rpm -ivh https://repo.zabbix.com/zabbix/3.4/rhel/
7/x86_64/zabbix-release -3.4-2.el7.noarch.rpm
# yum install zabbix-agent
# systemctl start zabbix-agent
```

Sin embargo, vemos que el servicio no se inicia correctamente y nos da un error:

```
[root@localhost elena]# systemctl start zabbix-agent
Job for zabbix-agent.service failed because a configured resource limit was exce
eded. See "systemctl status zabbix-agent.service" and "journalctl -xe" for detai
ls.
```

Deducimos que es un problema de SELinux, por lo que debemos crear una regla para permitir iniciar el agente. Utilizamos la herramienta audit2allow, que sirve para ”generar políticas de aceptación de SELinux a partir de los logs de operaciones denegadas”. Para ello, tenemos que instalar los siguientes paquetes, ya que en nuestro sistema operativo no los tenemos instalados:

```
# yum install audit
# yum install policycoreutils-python
```

Tras esto, escribimos la siguiente secuencia de operaciones para crear y activar la regla:

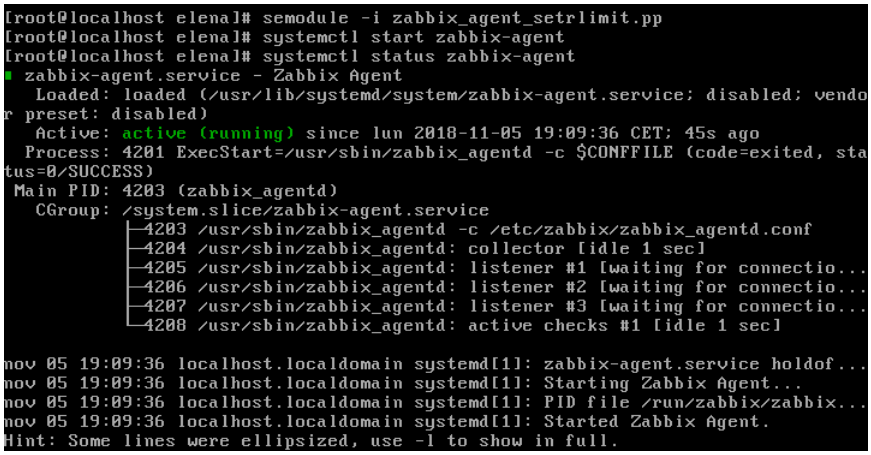
```
# cat /var/log/audit/audit.log|grep zabbix_agentd|
grep denied|audit2allow -M zabbix_agent_setrlimit
# semodule -i zabbix_agent_setrlimit.pp
```

Ya podemos iniciar el agente y comprobar que funciona correctamente.

```
[root@localhost elena]# cat /var/log/audit/audit.log | grep zabbix_agent | grep
denied | audit2allow -M zabbix_agent_setrlimit
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i zabbix_agent_setrlimit.pp

[root@localhost elena]# semodule -i zabbix_agent_setrlimit.pp
[root@localhost elena]# _
```



Aprovechamos y añadimos una regla al firewall para permitir el paso por el puerto 10050:

```
# firewall-cmd --permanent --add-port=10050/tcp
# firewall-cmd --reload
```

Por último, debemos modificar la configuración del agente ubicada en la ruta */etc/zabbix/zabbix_agentd.conf* para permitir que acepte conexiones de la IP del servidor. Buscamos las líneas Server y ServerActive y la modificamos poniendo la IP del servidor:

```
Server=192.168.56.20
ServerActive=192.168.56.20
```

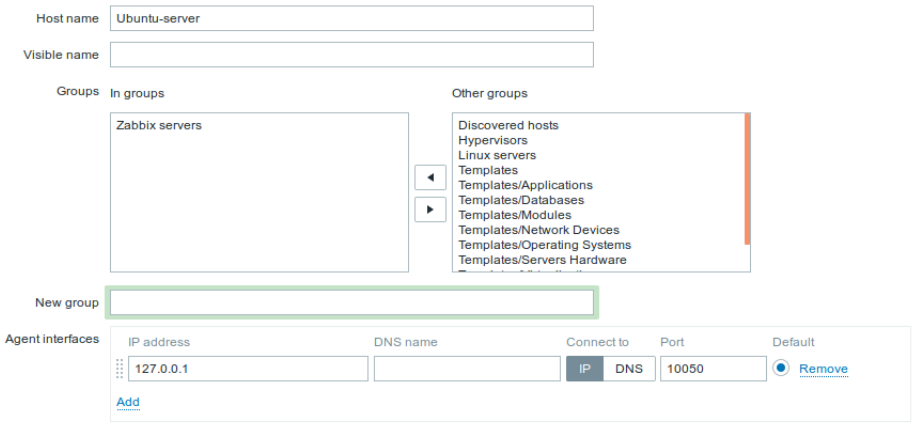
Recargamos el servicio con 'systemctl restart zabbix-agent', y ya podemos pasar a configurar nuestros hosts.

3. Configuración de hosts

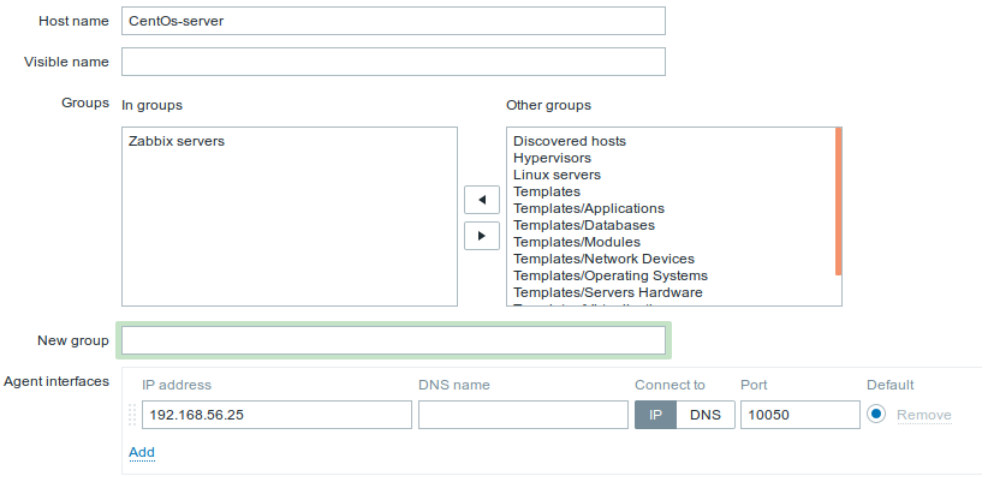
Accedemos al panel de control de Zabbix introduciendo en nuestro navegador la dirección *http://192.168.56.20/zabbix* y metemos como usuario **Admin** y como contraseña **zabbix** para hacer login como superusuario.

Ahora podríamos crear un nuevo usuario con unos permisos concretos para manejar nuestra monitorización, pero en esta prueba simple no es necesario. Pasamos a la pestaña *Configuración* → *Hosts*, donde podremos añadir nuestros *hosts* a monitorizar.

Aunque haya un host por defecto para la máquina local, añadiremos otro. Le damos al host que viene por defecto y hacemos *Full clone* e introducimos los datos necesarios: el nombre del host, el grupo en el que se encuentra y la dirección IP del mismo.



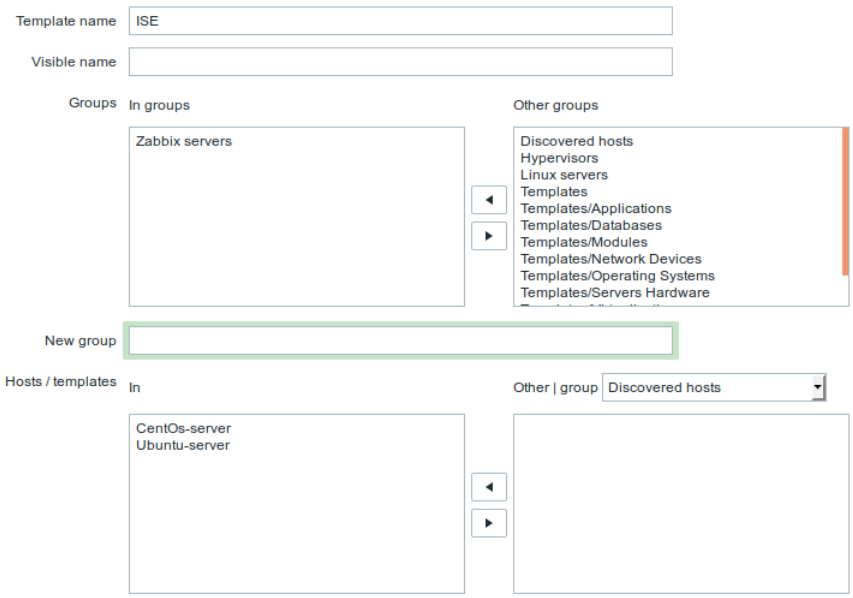
Un host debe pertenecer a al menos un grupo, puesto que los permisos se asignan a estos grupos y no a cada host individualmente. En este caso, la dirección IP es la de la máquina local, pues ahí se encontrará el agente. Para el host de CentOs la configuración es similar, teniendo en cuenta que la dirección IP debe ser la de la máquina en cuestión.



Una vez hecho esto, podemos cambiar la configuración del agente en ambos sistemas para que el Hostname coincida exactamente con el nombre que le hemos puesto a cada host en el front-end. También lo podemos hacer para CentOs, y recargamos ambos servicios. Ahora lo que tenemos que añadir es algún item para monitorizar, que define qué datos recogemos de un host. Lo que haremos será añadir un item para cada uno de los servicios que queremos monitorizar.

Además aprovecharemos y crearemos una plantilla para no repetir la creación

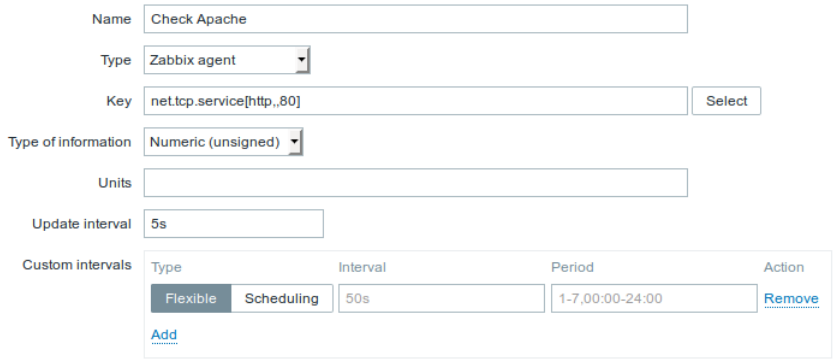
en ambos sistemas. De hecho, existen plantillas predefinidas para monitorizar muchos servicios, pero queremos aprender a crear una nueva. Vamos a la pestaña *Configuration* → *Templates*, y le damos a *Create template*.

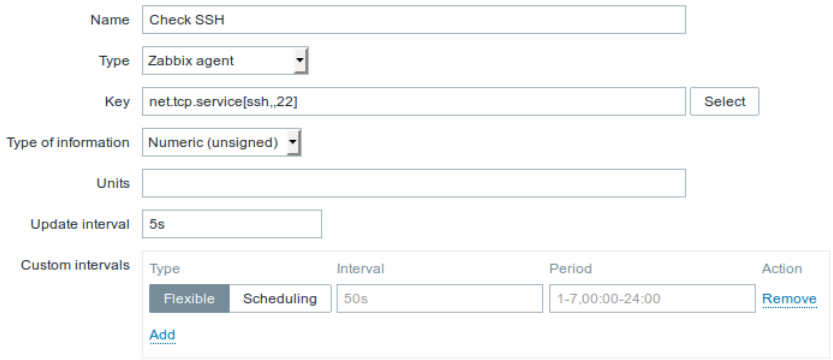


Notamos que hemos añadido ambos hosts a la lista, para que esta se asocie automáticamente a ellos. Ahora, añadimos a esta plantilla nuestros items para monitorizar los servicios ssh y apache. Vamos al menú Items dentro de la configuración de la plantilla, y pinchamos en *Create item*.

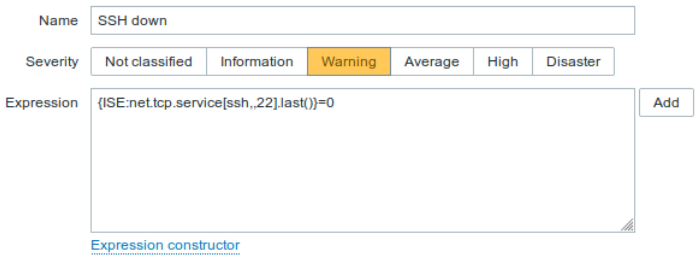
Debemos tener en cuenta los tipos de keys que codifican qué tipo de información queremos obtener del host. En nuestro caso, utilizamos la orden predefinda $net.tcp.service[service, < ip >, < port >]$, que devuelve un 1 si el servicio en cuestión está funcionando y acepta conexiones TCP por la IP y el puerto especificados, o un 0 en otro caso.

Rellenamos los datos según el servicio que queremos monitorizar en cada caso, poniendo siempre que el tipo de item será Zabbix agent, y podemos también especificar el intervalo de actualización de la información.





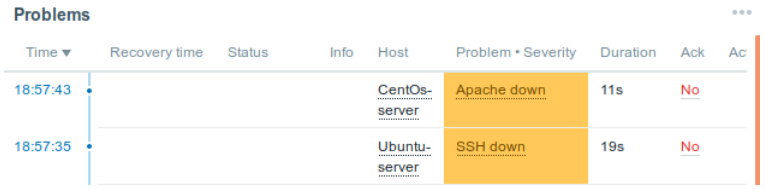
Una vez hemos añadido nuetros items ya podemos monitorizar los servicios. Sólo falta aádir un trigger para que cuando alguno de los servicios no esté operativo obtengamos una notificación. Vamos al menú Triggers dentro de la configuración de la plantilla, y le damos a *Create trigger*.



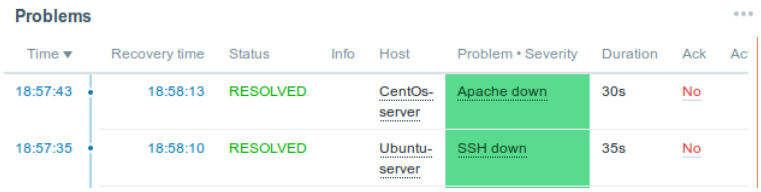
La sintaxis es muy clara: el trigger se activará (notificándonos de un problema) si el último chequeo que se realizó del servicio ssh devolvió un 0. De forma similar se configura el trigger para apache.

4. Prueba de funcionamiento

Para probar nuestro sistema de monitorización, podemos detener los servicios de ssh y/o apache en cualquiera de las dos máquinas, y deberíamos obtener una notificación en el Dashboard (panel de control) de que hay un problema con el servicio.



Esta notificación persistirá hasta que volvamos a encender el servicio, en cuyo caso nos informará de que el problema se ha resuelto.



5. Bibliografía

Referencias

[1] [Zabbix] Instalación y Descarga de Zabbix
`https://www.zabbix.com/download`

[2] [Audit] Instalación del paquete audit2allow
`https://www.centos.org/forums/viewtopic.php?t=5012`

[3] [Zabbix] Documentación de Zabbix
`https://www.zabbix.com/documentation/3.4/start`