

Check_mk. Chequeos. Primeros pasos.

Para iniciarnos con los chequeos de Check_mk, vamos a ver como se instalan los clientes y como se realiza una mínima configuración de equipos. Esto nos servirá de punto de partida para ir avanzando posteriormente. Este artículo **asume** que tenemos realizada una [instalación de OMD Distro](#), aunque te puede ser igualmente útil si no es así.

Recordamos que la configuración mediante check_mk generará posteriormente sus propios ficheros de configuración en formato Nagios. ¿Hace esto que no necesitemos conocer y usar el formato de Nagios? Desgraciadamente no. Hay algunos objetos que deben estar creados en los ficheros de configuración de Nagios (p.e. grupos de hosts y de servicios). ¿Crearemos en Nagios entonces solo los objetos necesarios? Como quieras, yo personalmente prefiero hacer una mezcla y configurar una parte en un lado y otra parte en otro. No quiero renunciar a lo mejor de cada. Lo que sí es necesario desde luego es conocer los objetos de Nagios y como se relacionan entre ellos y definir lo necesario en los ficheros de Nagios.

Check_mk incorpora un complemento de plugins con un agente propio (Windows, Linux, Unix,...) para realizar chequeos pasivos en sistemas. Al ser chequeos pasivos hace que el sistema de Nagios necesite menos recursos para realizarlos. ¿Que diferencia hay entre chequeos activos o pasivos?

- Activos: Los ejecuta el servidor de Nagios en intervalos regulares.
- Pasivos: Los ejecuta el propio cliente en intervalos regulares o bajo demanda.

Algo habitual, como en el caso que nos ocupa es que se ejecute un chequeo activo contra el host destino para recoger todos los datos que este tiene de chequeos pasivos.

Check_mk usa un fichero especial de configuración a partir del cual genera los ficheros de configuración con la sintaxis de Nagios. Incorpora funcionalidades interesantes como:

- Chequeos pasivos. Con un solo chequeo (activo) al sistema obtiene todos los valores a monitorizar y los “coloca” en sus respectivos servicios de forma pasiva.
- Autodetección de posibles chequeos. Una vez instalado el agente en la máquina y añadido en el fichero de configuración detecta automáticamente los servicios más importantes a chequear de este.
- Plantillas [PNP4Nagios](#). Incorpora plantillas de PNP4Nagios ya definidas para todos sus chequeos.
- Visor de eventos. Configura servicios para los eventos de Windows e incorpora su visualización en Nagios (Siempre que esté dispuesto a pegarte con ellos y tener luces en rojo a menudo).
- Capacidad de ejecutar plugins estándar de Nagios en el propio Host cliente de los que se usan con [NRPE](#) (otro tema que se verá más adelante).

Partimos de que tenemos ya instalado check_mk mediante la instalación de OMD. Si lo hemos

instalado de forma independiente de éste en un servidor de Nagios, cambiarán las rutas pero los procedimientos serán los mismos

Instalación de agentes.

Primero, antes de ponernos a configurar objetos de Nagios con la sintaxis especial de los ficheros de configuración de check_mk, debemos instalar el cliente en los servidores a monitorizar. A modo de ejemplo instalaremos un cliente Windows y otro Linux (el propio servidor de Nagios p.e.).

En una instalación realizada con OMD, los agentes específicos de check_mk para cliente estarán en: /opt/omd/versions/default/share/check_mk/agents

Instalación del agente Windows.

Se copia el directorio del agente check_mk para Windows a nuestro servidor desde /opt/omd/versions/default/share/check_mk/agents/windows

Se ejecuta la instalación “install_agent.exe”. Nos sugiere directorio por defecto y los dos componentes necesarios a instalar. ¿Sencillo no?

Verificaciones.

El servicio usa el puerto 6556, así que podemos verificar que sea accesible. Desde el propio servidor Windows:

```
#netstat -ano | find "6556"
TCP    0.0.0.0:6556          0.0.0.0:0          LISTENING          7936
```

Desde el servidor de Nagios p.e. (nos tiene que salir un “chorro” de datos):

```
#telnet 192.168.1.33 6556
```

Ojo. SI Windows tiene activado el Firewall debemos permitir el acceso para el ejecutable de check_mk (o para el puerto) al menos para la dirección IP del equipo Nagios.

Alternativamente podemos limitar el acceso también limitando en el fichero de configuración check_mk.ini del directorio de instalación del agente en Windows (tendremos un ejemplo como check_mk.example.ini). Usaremos la directiva “only_from”

Instalación del agente Linux.

Debemos tener instalado previamente xinetd.

Lo podemos verificar en Debian / Ubuntu:

```
# dpkg -list | grep xinetd
```

En Redhat y similar con:

```
# yum list installed | grep xinetd
```

El agente para Linux localizado en “/opt/omd/versions/default/share/check_mk/agents” es para realizar una instalación manual de este. Mejor usar los paquetes para nuestra distribución que

podemos encontrar en la página de downloads de check_mk. Instalaremos el cliente y la extensión de log watch.

Para Debian / Ubuntu sería:

```
dpkg -i check-mk-agent_x.y.z_all.deb
dpkg -i check-mk-agent-logwatch_x.y.z_all.deb
```

Para distribuciones Redhat / Suse / CentOS/..:

```
rpm -i check_mk-agent-x.y.z.noarch.rpm
rpm -i check_mk-agent-logwatch-x.y.z.noarch.rpm
```

Recordad, debemos reiniciar / recargar xinet.d para que habilite el servicio..

```
service xinetd restart
```

Podemos verificar su funcionamiento de la forma ya comentada para el agente Windows. Habrá que asegurarse, si tenemos iptables activo, de dar acceso al menos al servidor de Nagios al puerto 6556. También podemos limitar el acceso mediante el fichero de configuración de xinet.d creado por check_mk. Normalmente el fichero se ubica en /etc/xinetd.d/check_mk y usaremos la directiva “only_from”.

Otros agentes.

En el directorio /opt/omd/versions/default/share/check_mk/agents disponemos de agentes para otros sistemas operativos como Solaris, HPUX, freebsd,... Si queremos instalarlo para estos realizaremos una instalación manual del agente tal como se indica en la documentación de check_mk.

Verificación desde check_mk

Para verificar que check_mk puede acceder a nuestros agentes ejecutaremos desde el servidor de Nagios:

```
check_mk -d host_a_chequear
```

Si tenemos nuestra instalación con OMD debemos realizarlo con el usuario de nuestra instancia (nombre de nuestra instancia). Siempre cambiaremos a dicho usuario previamente a ejecutar tareas relacionadas con el mantenimiento de Nagios, Check_mk,...

```
su - usuario
```

Si queremos más detalles sobre los clientes y su instalación podemos seguir la documentación de check_mk al respecto.

Configuración básica de chequeos.

Antes de ponernos a configurar debemos asegurarnos de que llegamos al host destino por su nombre, bien con DNS o bien porque lo hemos definido en el fichero hosts del servidor de Nagios. Es importante porque en los ficheros de configuración usaremos el nombre de los hosts (corto normalmente) y no definiremos adicionalmente la IP como si se hace en los ficheros de configuración de Nagios.

Definiciones en el fichero de configuración.

El Fichero principal de configuración de objetos de check_mk está en la siguiente ruta en una instalación con OMD:

```
/opt/omd/sites/foo/etc/check_mk/main.mk
```

Existe en dicha ruta un directorio “conf.d” donde podemos crear ficheros con extensión “.mk” que también formarán parte de nuestra configuración.

Para empezar tocaremos lo mínimo posible y en posteriores entradas entraremos más a fondo.

Editamos el fichero y lo dejamos tal que así para incluir nuestro host local con el cliente check_mk para linux instalado y nuestro cliente Windows de prueba (con su nombre corto, asegúrate que se resuelva correctamente). Podemos eliminar el que no necesitamos.

```
# Put your host names here
all_hosts = [ 'localhost', 'srv_windows' ]
```

Inventariar los equipos en la configuración.

De momento hemos configurado que equipos nos gustaría chequear pero debemos de alguna forma decirle que mire que nos pueden ofrecer para chequear e incluya dichos chequeos. Lo hacemos cambiando al usuario de nuestra instancia OMD (su – usuario) y ejecutando:

```
check_mk -I #(es un i mayúscula)
```

```
OMD[foo]:~$ check_mk -I
cpu.loads          1 new checks
cpu.threads        1 new checks
df                 6 new checks
diskstat           1 new checks
kernel             3 new checks
kernel.util        1 new checks
lnx_if             1 new checks
logwatch           14 new checks
mem.used           1 new checks
mem.win            1 new checks
mounts             1 new checks
omd_status         1 new checks
systemtime         1 new checks
tcp_conn_stats     1 new checks
uptime             2 new checks
winperf_phydisk    1 new checks
winperf_processor.util 1 new checks
```

La orden anterior chequea todos los equipos definidos en el fichero. Si queremos chequear uno solo será:

```
check_mk -I servidor
```

(chequea solo ese equipo)

Reinicio de check_mk

Una vez realizada configuración e inventario debemos reiniciar / recargar la configuración de Nagios. Como estamos bajo entorno OMD lo hacemos con:

- `check_mk -R` (recarga la configuración de ficheros de check_mk a formato Nagios y reinicia Nagios). O bien:
- `check_mk -O` (recarga la configuración de ficheros de check_mk a formato Nagios, sin reiniciar y cortar servicio). Mejor esta última.
- `omd reload` (recarga la configuración de OMD)

Si estuviéramos en un entorno sin OMD (Nagios y check_mk instalados por separado) sería:

- `check_mk -O`

`/ruta/binario/nagios -v /ruta/afichero/config/nagios.cfg` (para verificar previamente configuración)

`service nagios3 reload` (recarga configuración).

Visualización de la información.

Si todo ha ido bien podemos ir a nuestro interface preferido para visualizar la información tal como (recordamos). En nuestro caso la información en el GUI de check_mk se muestra para el servidor Linux.

State	Service	Status detail	Icons	Age	Checked	Perf-O-Meter
OK	Check_MK	OK - Agent version 1.2.0p4, execution time 1.5 sec		3 min	57 sec	1.5s
OK	CPU load	OK - 15min load 0.61 at 1 CPUs		3 min	56 sec	0.1
OK	CPU utilization	OK - user: 1.0%, system: 3.0%, wait: 2.3%		3 min	56 sec	6%
PEND	Disk IO SUMMARY			-	-	
OK	fs_/_	OK - 40.2% used (3.01 of 7.5 GB), (levels at 80.0/90.0%), trend: +9.06MB / 24 hours		3 min	56 sec	40%
OK	Interface 2	OK - [eth0] (up) speed unknown, in: 688.10B/s, out: 649.54B/s		3 min	56 sec	688.1B/s 649.5B/s
OK	Kernel Context Switches	OK - 93/s in last 60 secs		2 min	56 sec	93.4/s
OK	Kernel Major Page Faults	OK - 1/s in last 60 secs		2 min	56 sec	0.5/s
OK	Kernel Process Creations	OK - 6/s in last 60 secs		2 min	56 sec	6.0/s
OK	LOG /var/log/auth.log	OK - no error messages		3 min	56 sec	
OK	LOG /var/log/kern.log	OK - no error messages		3 min	56 sec	
OK	LOG /var/log/messages	OK - no error messages		3 min	56 sec	
OK	LOG /var/log/syslog	OK - no error messages		3 min	56 sec	
OK	Memory used	OK - 0.47 GB used (0.24 GB RAM + 0.23 GB SWAP, this is 130.5% of 0.36 GB RAM)		3 min	56 sec	130%
OK	Mount options of /	OK - mount options exactly as expected		3 min	56 sec	
OK	Number of threads	OK - 149 threads		3 min	56 sec	149
OK	OMD foo status	OK - all services are running		3 min	56 sec	
OK	TCP Connections	OK - ESTABLISHED: 10, TIME_WAIT: 24		3 min	56 sec	
OK	Uptime	OK - up since Wed Mar 27 23:42:16 2013 (0d 13:34:28)		3 min	56 sec	00d 13h 34m

Y para el servidor Windows (con una alerta crítica de espacio en disco):

srv_windows							
State	Service	Status detail	Icons	Age	Checked	Perf-O-Meter	
OK	Check_MK	OK - Agent version 1.2.0p2, execution time 0.2 sec		3 min	27 sec	0.2s	
OK	CPU utilization	OK - 8% used / 2 CPUs (in last 60 secs)		2 min	27 sec	8%	
OK	Disk IO SUMMARY	OK - 59.38KB/sec read, 64.99KB/sec write		87 sec	27 sec	0.06M/s 0.06M/s	
OK	fs_C:/	OK - 38.3% used (187.21 of 488.3 GB), (levels at 80.0/90.0%), trend: +83.06KB / 24 hours		3 min	27 sec	38%	
CRIT	fs_D:/	CRIT - 90.0% used (105.50 of 117.2 GB), (levels at 80.0/90.0%), trend: 0.00B / 24 hours		3 min	27 sec	90%	
OK	fs_E:/	OK - 20.4% used (67.20 of 329.0 GB), (levels at 80.0/90.0%), trend: 0.00B / 24 hours		3 min	27 sec	20%	
OK	fs_G:/	OK - 54.2% used (322.97 of 596.2 GB), (levels at 80.0/90.0%), trend: 0.00B / 24 hours		3 min	27 sec	54%	
OK	fs_M:/	OK - 60.7% used (597.21 of 984.1 GB), (levels at 80.0/90.0%), trend: +2.77MB / 24 hours		3 min	27 sec	60%	
OK	LOG ACEEventLog	OK - no error messages		3 min	27 sec		
OK	LOG Application	OK - no error messages		3 min	27 sec		
OK	LOG HardwareEvents	OK - no error messages		3 min	27 sec		
OK	LOG Internet Explorer	OK - no error messages		3 min	27 sec		
OK	LOG Key Management Service	OK - no error messages		3 min	27 sec		
OK	LOG Media Center	OK - no error messages		3 min	27 sec		
OK	LOG OAlerts	OK - no error messages		3 min	27 sec		
OK	LOG Security	OK - no error messages		3 min	27 sec		
OK	LOG System	OK - no error messages		3 min	27 sec		
OK	LOG Windows PowerShell	OK - no error messages		3 min	27 sec		
OK	Memory and pagefile	OK - Memory usage: 70.9% (2.3/3.2 GB), Page file usage: 48.7% (3.2/6.5 GB)		3 min	27 sec	70%	
OK	System Time	OK - Offset is -0.4 sec (levels at 30/60 sec)		3 min	27 sec	-0.4 s	
OK	Uptime	OK - up since Thu Mar 28 08:06:11 2013 (0d 05:11:02)		3 min	27 sec	00d 05h 11m	

Hasta aquí hemos visto una configuración muy básica de los chequeos con check_mk (el 1% quizá :-). Avanzamos en la segunda parte de este artículo: [Check_mk. Chequeos. Primeros pasos \(II\)](#).

Otros artículos de interés relacionados

- [Monitorizar ficheros log con check_mk en Linux – Oracle alertlog y otros.](#)
- [Instalación de packages de check_mk.](#) Oracle RMAN package.
- [Monitorizar Oracle con Nagios y check_mk.](#)
- [Iconos en Nagios \(y check_mk\).](#)