

VMWare. Monitorización de servidores vSphere ESX(i)

Introducción.

Para la monitorización de nuestros servidores vSphere ESX / ESXi vamos a usar un estupendo plugin desarrollado por OP5 y disponible a través de [Nagios Exchange](#), [check_vmware_api](#). Podemos localizar más información y descargarlo también en su página de [OP5](#). No debemos confundirnos con el plugin check_esx que es como se llamaba previamente, quedó en NagiosExchange pero desactualizado. Vamos a detallar la instalación tanto para Debian 6 como para Ubuntu 12.04 que es similar.

Si lo que buscas es monitorizar el vCenter... en este artículo encontrarás información.

Vamos a necesitar instalar VMWare vSphere SDK for Perl para que funcione el plugin.

Instalación de dependencias.

Necesitamos instalar previamente algunas dependencias necesarias.

En Debian 6 / Ubuntu 12.04 sería:

```
apt-get install libssl-dev perl-doc liburi-perl libxml-libxml-perl libcrypt-ssleay-perl ia32-libse2fsprogs libnagios-plugin-perl libarchive-zip-perl libclass-methodmaker-perl libclass-methodmaker-perl libsoap-lite-perl libdata-dump-perl
```

En Ubuntu 12.04 además instalamos:

```
apt-get install libuuid-perl
```

Instalación del SDK Perl de VMWare.

Nos bajamos el SDK para nuestra versión de VMWare de: <http://www.vmware.com/support/developer/viperltoolkit/> (Requiere autenticarse con tu cuenta de MyVMware).

Antes de instalarlo debemos tener en cuenta algunos detalles. En el caso de Debian no está oficialmente soportado (aunque funciona) por lo que debemos crear un fichero release para “engañar” al instalador. Creamos el fichero:

“/etc/Debian-release” (Debe acabar en –release, como en Ubuntu el fichero /etc/lsb-release) con el contenido similar al de una instalación de Ubuntu. P.e.:

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04.2 LTS"
```

También debemos configurar provisionalmente el proxy del sistema ya que nos lo exigirá la instalación. Si no tenemos proxy pues lo dejamos vacío (igual a nada) pero parece que debe existir.

```
export http_proxy=http://myproxy.mydomain.com:port
export ftp_proxy=http://myproxy.mydomain.com:port
```

Descomprimos el Software e instalamos.

```
tar xvzf VMware-vSphere-SDK-for-Perl-4.0.0-161974.i386.tar.gz
cd vmware-vmphere-cli-distrib/
./vmware-install.pl
```

Si durante la instalación nos da un error de CPAN del tipo “Install not able to configure CPAN. Please configure CPAN manually and re-run the Installer” deberemos actualizar nuestro CPAN y instalar de nuevo:

```
perl -e shell -MCPAN
install CPAN
reload CPAN
exit
```

Al finalizar la instalación nos dará un aviso de módulos de perl que no son los más actuales. Podemos ignorarlo. Al menos a nuestro plugin no le afecta y no veo manera adecuada ni justificación de tener tan actualizados los módulos perl.

```
The following Perl modules were found on the system but may be too old to work
with vSphere CLI:
Compress::Zlib 2.037 or newer
Compress::Raw::Zlib 2.037 or newer
version 0.78 or newer
IO::Compress::Base 2.037 or newer
IO::Compress::Zlib::Constants 2.037 or newer
UUID 0.03 or newer
LWP::Protocol::https 5.805 or newer
```

Configuración de Nagios.

Copiamos el plugin `check_vmware_api` de Nagios a nuestro directorio de plugins y le hacemos ejecutable como siempre. Nuestro directorio de plugins estará habitualmente en `/usr/lib/nagios/plugins` en nuestra [instalación de Nagios](#) mediante paquetes de la distribución o en una con <http://www.eldespidado.com/nagios-omd-introduccion/> en `/omd/versions/default/lib/Nagios/plugins/`.

Testeamos version del plugin con:

```
#!/check_vmware_api.pl -V
check_vmware_api.pl 0.7.0
```

Si tenemos prisa por ver si funciona podemos ir probándolo:

Probamos a pelo el comando con:

```
#!/check_vmware_api.pl -D IP_vcenter -u User -p password -H NOMBRE_HOST_ESXI -l cpu -s usage  
CHECK_VMWARE_API OK - cpu usage=452.00 MHz (4.86%) | cpu_usagemhz=452.00Mhz;; cpu_usage=4.86%;;
```

Es conveniente evitar usar el usuario root de los ESXi o el admin de vCenter. En la instalación por defecto de vSphere debe haber un profile “read only”. Podemos crear un usuario Nagios miembro del grupo “user” y asignarle ese perfil.

Configuración del fichero resource.cfg de Nagios.

Añadiremos las variables de autenticación al menos de nuestro vCenter si realizaremos los chequeos a través de este y de los servidores ESXi si los realizaremos contra estos directamente.

Si no sabes de que va y para que vale el resource.cfg este aquí tienes más información.

```
# ACCESO A LOS ESX  
$USER11$=username_vcenter  
$USER12$=password_vcenter  
# Si vamos a acceder individualmente a los ESX debemos crear  
# las credenciales para estos si son diferentes  
$USER13$=username_esx  
$USER14$=password_esx
```

commands

Creamos nuestro fichero de commands para usar el plugin. Pongo uno bastante completo aquí.

vmware_commands.cfg

hosts y services

Para usar los command crearemos primero nuestros objetos de Nagios: hostgroups, hosts y services. Algunos ejemplos de definición de estos.

```
define hostgroup {  
    hostgroup_name ESXi-Servers  
    alias ESXi-Servers  
    members virtual1, virtual2  
}  
define host {  
    use generic-host  
    host_name virtual1  
    alias virtual1  
    address 192.168.2.20  
}  
# Servicio - Chequeo de CPU de ESX  
define service {  
    use generic-service
```

```
hostgroup_name ESXi-Servers
service_description VMWare CPU Usage
check_command check_vmware_api_dc_host_cpu_usage!vcenter!90!95
}
```

Puedes bajarte un fichero cfg con ejemplos de un montón de servicios definidos que llaman a los commands del fichero previo:

[vmware-host-services-sample.cfg](#)

SI no sabes donde ponerlo... tienes que leer [este](#) artículo.

Este plugin nos deja acceder a la información de varias maneras:

- Información de servidores ESX / ESXi a través de vCenter (preguntando a este)
- Información de servidores ESX / ESXi a través directamente de cada servidor.
- Información de máquinas virtuales a través de vCenter (preguntando a este)
- Información de máquinas virtuales a través directamente de cada servidor ESXi.

Las formas básicas de uso del plugin, para consultar a través de vCenter (Datacenter) quedarían reflejadas en los siguientes commads:

```
# Commandos para ESX(i) Datacenter/vCenter
define command{
command_name check_vmware_api_dc_vm
command_line $USER1$/check_vmware_api.pl -D $ARG1$ -u $USER11$ -p $USER12$ -l $ARG2$ -s $ARG3$ -N
$HOSTALIASS$ -w $ARG4$ -c $ARG5$
}
define command{
command_name check_vmware_api_dc_host
command_line $USER1$/check_vmware_api.pl -D $ARG1$ -u $USER11$ -p $USER12$ -l $ARG2$ -s $ARG3$ -H
$HOSTADDRESS$ -w $ARG4$ -c $ARG5$
}
```

Donde la diferencia la marca el parámetro -N (máquina virtual) -H (Host ESX(i)) para ambas modalidades posibles al respecto: consulta sobre máquina virtual o sobre ESX(i). Se podría también usar la consulta directa a un host ESX(i) pero no le veo mucho sentido a no se que no tengas vCenter. En cualquier caso en la documentación del plugin y con los ejemplos de los ficheros de configuración de este artículo es sencillo.

Para que sea más sencilla la configuración del check_command en los servicios recomiendo generar nuevos commands con nombres más acordes y más directos al grano. Por ejemplo:

```
define command{
command_name check_vmware_api_dc_host_cpu_usage
command_line $USER1$/check_vmware_api.pl -D $ARG1$ -u $USER11$ -p $USER12$ -H
HOSTALIASS$ -l cpu -s usage -w $ARG2$ -c $ARG3$
```

```
# sample check_command: check_vmware_api_dc_host_cpu_usage!vcenter!80!90
# NOTES: Warn and critical in percent
}
```

Hemos creado un command donde ya le definimos el valor en concreto a buscar (-l cpu) y opción de este (-s usage) le hemos denominado check_vmware_api_dc_host_cpu_usage . De esta forma el check_command del servicio quedará bastante más limpio e inteligible como:

```
check_vmware_api_dc_host_cpu_usage!vcenter!80!90
```

pasandole solo los valores de nombre de nuestro vcenter e intervalos de Warning/ Critical.

Este es el resultado:

Datastores	WARNING	2013-04-16 13:13:38	4d 21h 4m 25s	3/3	CHECK_VMWARE_API.PL WARNING - Storages : 'DS_LOCAL (29.08%), 'Volumen3'(free)=253871.00 MB (24.22%), 'Volumen4'
VMFS Volumen1	WARNING	2013-04-16 13:12:48	4d 21h 24m 6s	3/3	CHECK_VMWARE_API.PL WARNING - Storages : 'Volumen1'(
VMware CPU Usage11	OK	2013-04-16 13:09:32	0d 0h 57m 22s	1/3	CHECK_VMWARE_API.PL OK - cpu usage=14.13 %
VMware IO Reads	OK	2013-04-16 13:06:51	4d 22h 20m 59s	1/3	CHECK_VMWARE_API.PL OK - io read latency=0 ms
VMware IO Write	OK	2013-04-16 13:15:14	3d 20h 51m 40s	1/3	CHECK_VMWARE_API.PL OK - io write latency=7 ms
VMware Mem Usage	OK	2013-04-16 13:08:43	0d 1h 28m 31s	1/3	CHECK_VMWARE_API.PL OK - mem usage=62.35 %
VMware NET Usage	OK	2013-04-16 13:07:37	4d 22h 19m 17s	1/3	CHECK_VMWARE_API.PL OK - net usage=629.00 KBps
VMware Runtime Issues	OK	2013-04-16 13:08:30	4d 22h 18m 24s	1/3	CHECK_VMWARE_API.PL OK - No config issues
VMware Runtime Status	OK	2013-04-16 13:11:24	4d 22h 17m 30s	1/3	CHECK_VMWARE_API.PL OK - overall status=green
VMware Swap Usage	OK	2013-04-16 13:10:18	4d 22h 26m 36s	1/3	CHECK_VMWARE_API.PL OK - swap usage=0.00 MB

Aprovecho, ya que está en Warning, para mencionar el tema de chequeo de Datastores. En los ficheros de configuración tenemos ambos ejemplos: monitorizar todos de un plumazo (con lo cual vemos que no está claro cual/cuales están en Warning) o monitorizarlos individualmente (en el ejemplo "Volumen1" que es mucho más claro. Otra duda que nos surgirá... ¿monitorizo los datastores en todos los hosts ESX(i)? (habitualmente son los mismos o comparten muchos de ellos, saltarán alarmas en todos). Lo normal sería monitorizarlos solo en uno de ellos y asignarle los servicios incluso al host vCenter en lugar de a un ESX(i) pero nos puede llegar a interesar tener la vista de cada ESX(i) por si nos enfrentamos a un problema e Multipath p.e.

En los ficheros de configuración están los commands para monitorizar los valores más representativos de nuestros ESX(i) a través de un vCenter así como abundantes ejemplos de la definición de servicios listos para sus uso. También hay ejemplos para monitorizar máquinas virtuales pero... no lo recomiendo para eso. SI no controlas muy bien vSphere y sus estadísticas de máquinas virtuales puedes volverte loco ya que normalmente el valor obtenido no te corresponderá con lo que ves en el S.O. Para esos casos en principio prefiero usar agentes en los servidores, ya sea de check_mk, nsclient++, nrpe,...

Tienes más documentación de las posibilidades del plugin en la página de [documentación](#) de este de OP5 y llamando a la ayuda del plugin como siempre (-h).

Más artículos interesantes disponibles en la [Guía del Visitante](#).