



Monitorización de entornos Virtuales



m:

<https://pandorafms.com/manual/!776/>

Permanent link:

https://pandorafms.com/manual/!776/es/documentation/pandorafms/monitoring/05_virtual_environment_monitoring

24/06/10 14:34



Monitorización de entornos Virtuales

Para monitorizar [VMware®](#), [XenServer®](#), [MySQL®](#), [Microsoft SQL Server®](#), [Oracle®](#), [DB2®](#) y [SAP R3®](#) se utiliza [Discovery PFMS®](#).

RHEV

Red Hat® Enterprise Virtualization (RHEV) es una de las tecnologías más usadas por empresas que tienen como base el sistema operativo Red Hat en su Data Center. Pandora FMS Enterprise ofrece la posibilidad de monitorizar las arquitecturas virtuales basadas en RHEV mediante el RHEV Monitoring Plugin.

Arquitectura a monitorizar

- Data Centers.
- Host Clusters.
- Storage Domains.
- Networks.
- Hosts.
- Virtual Machines.

Pandora FMS usa la API oficial que provee el sistema de virtualización RHEV.

Monitorización con RHEV Monitoring Plugin

Para la monitorización del sistema operativo instalado en las máquinas virtuales se recomienda usar un Agente de Pandora FMS en lugar del API RHEV.

La monitorización de entornos virtuales RHEV se basa en los siguientes componentes:

1. Un *plugin* de Agente que realiza las tareas de autodescubrimiento y recogida de datos. El *plugin* de Agente es el encargado de enviar la información a Pandora FMS.
2. Un *script* de reconocimiento que actualiza diversos valores para las entidades descubiertas, necesario para el correcto funcionamiento de las extensiones del *plugin*.
3. RHEV Viewer y RHEV Manager: Son extensiones que aportan el operar el apagando/encendiendo máquinas virtuales, todo ello desde la Consola web de Pandora FMS.

Para poder usar el *script* de reconocimiento es necesario tener activado el [Discovery server](#). Para que ciertas

variables de la API reflejen el valor real de la máquina virtual asociada es necesario instalar el Agente de RHEV; podrá encontrar todo al respecto en la documentación de su versión de RHEV.

Funcionamiento interno del plugin RHEV

El *plugin* RHEV Monitoring Plugin extrae la información mediante la API web que sirve el entorno de virtualización RHEV.

Si solamente necesita la información de monitorización lo único que tiene que configurar es el *plugin* de Agente que realizará esta tarea. La configuración del *plugin* permite escoger qué elementos van a ser monitorizados y la configuración de sus Módulos. Una vez creados los XML, el *plugin* de Agente envía los archivos, bien usando **Tentacle** o bien los copia a un directorio local, según el método de transferencia escogido.

Si además va a usar las extensiones RHEV Viewer y RHEV Manager necesitará usar el *script* de reconocimiento. El guión de reconocimiento es el encargado de actualizar unas variables para cada uno de los Agentes detectados en Pandora FMS según los valores configurados en RHEV. Estas variables son necesarias para poder visualizar las entidades correctamente en la extensión RHEV Viewer y gestionar adecuadamente las máquinas virtuales con la extensión RHEV Manager.

Requisitos previos a la instalación del plugin RHEV

El *plugin* de Agente requiere el siguiente software:

- curl.
- perl-XML-Simple.
- **Agente Software PFMS** y `tentacle_client` de Pandora FMS.

Red Hat

En sistemas basados en Red Hat® puede instalar las dependencias con el comando:

```
yum install perl-XML-Simple curl
```

SLES

En sistemas basados en SUSE puede instalar las dependencias con el comando:

```
zypper install perl-XML-Simple curl
```

Debian/Ubuntu

En sistemas basados en Debian/Ubuntu puede instalar las dependencias con el comando:

```
apt-get install libxml-simple-perl curl
```

Descarga del certificado de RHEV

Antes de usar el *plugin* será necesario descargar el certificado que permite la conexión por HTTPS al API de RHEV. Para ello, ejecute el siguiente comando:

```
curl -o rhvm.cert http://[RHEVM-HOST]:8080/ca.crt
```

Donde [rhvm-host] es el nombre del servidor que sirve el API de RHEV. Un ejemplo concreto podría ser:

```
curl -o rhvm.cert http://rhvm.server:8080/ca.crt
```

Una vez descargado el certificado puede comprobar que la conexión a la API se realiza de forma correcta con el siguiente comando mediante el uso de conectores de línea \:

```
curl -X GET \  
      -H "Accept: application/xml" \  
      -u [USER:PASS] \  
      --cacert [CERT] https://[RHEVM-HOST]:8443/api
```

Con los siguientes valores:

- USER: *user @domain* para conectarse al API.
- PASS: contraseña del usuario con el que se conectará al API.
- CERT: ruta al certificado descargado en el paso anterior.
- RHEVM-HOST: dirección del *host* que sirve el API.

Ejemplo con datos concretos del comando:

```
curl -X GET \  
      -H "Accept: application/xml" \  
      -u [user@testdomain:12345] \  
      --cacert /home/user/ca.crt https://rhvm.server:8443/api
```

Si la ejecución de comando es positiva, devolverá una salida en formato XML con información general sobre la API de RHEV.

Consideraciones previas sobre la configuración de RHEV

En el entorno de virtualización RHEV es posible que varias entidades tengan un mismo nombre. Esto supone un problema, ya que en Pandora FMS esas entidades se transformarán en Agentes en los que no se permite duplicidad en los nombres. Además, también generará problemas a la hora de *parsear* el resultado devuelto por la API en formato XML, mostrando un error parecido al siguiente:

```
Warning: <data_center> element has non-unique value in 'name' key attribute:
Default at ./plugin-rhev.pl line 199
```

Para solucionar el problema lo único necesario es seguir una nomenclatura de nombres para las entidades del entorno de virtualización RHEV en el que no se repitan los nombres.

Instalación del plugin RHEV para Agente

Para instalar el *plugin* de Agente solo tiene que copiar el *script* `rhev-plugin.pl` y el archivo de configuración `rhev-plugin.conf` en un directorio de la máquina donde está instalado el Agente de Pandora FMS que ejecutará el *plugin*. El *plugin* se puede ejecutar en un Agente instalado en las mismas máquina que el servidor de Pandora FMS o en otra máquina distinta.

Para ejecutar el *plugin* debe añadir al archivo de configuración del Agente (por defecto `/etc/pandora/pandora_agent.conf`) la siguiente línea:

```
module_plugin /root/rhev-plugin.pl /root/rhev-plugin.conf
```

Añadiendo esta línea el *plugin* de Agente realizará sus funciones en cada ejecución.

Monitorizando la arquitectura virtual RHEV

Para ver el resultado de la ejecución del *plugin* de Agente vaya a Operation → Monitoring → Views → Agent Detail. Al pulsar el nombre de un Agente podrá ver los Módulos de monitorización creados por el *plugin*, además de otros datos relativos.

El *plugin* crea un Agente en Pandora FMS por cada una de las entidades detectadas en el descubrimiento de la arquitectura RHEV. Para cada tipo de entidad se crean automáticamente una serie de Módulos determinados, monitorizando la información importante de cada una de ellas.

Si el Agente seleccionado correspondiera a un *Host* en lugar de una Máquina Virtual, los módulos de monitorización serían distintos.

El *plugin* RHEV también monitoriza los eventos ocurridos dentro de la arquitectura virtual. El *plugin*

creará un Módulo para cada evento monitorizado dentro de cada entidad afectada. Los datos para los Módulos creados a partir de eventos son datos del evento: hora del suceso, descripción del evento.

Además de los Agentes y los Módulos relativos a la arquitectura RHEV propiamente dicha, se genera un Módulo llamado, por defecto RHEV Plugin, en el Agente que ejecuta el *plugin*.

Módulos de agente de la arquitectura virtual RHEV

Data Center

- *Status*: Estado del Data Center.

Storage Domain

- *Available Space*: Espacio disponible en el Storage Domain.
- *Committed Space*: Espacio comprometido en el Storage Domain.
- *Used Space*: Espacio usado en el Storage Domain.
- *Percent Free Space*: Porcentaje de espacio libre en el Storage Domain.

Network

- *Status*: Estado de la red virtual.
- *STP Status*: Estado de la funcionalidad Spanning Tree Protocol.

Cluster

- *Overcommit Percent*: Porcentaje de sobreasignación del clúster.
- *Transparent HugePages*: Estado de la funcionalidad Transparent HugePages.
- *High threshold*: Límite superior en las políticas de planificación.
- *Low threshold*: Límite inferior en las políticas de planificación.
- *Threshold duration*: Duración de los límites en las políticas de planificación.

Host

- *Status*: Estado del *Host*.
- *Buffers size*: Tamaño de los *buffers*.
- *Cache size*: Tamaño de la *cache*.
- *Cached swap*: Cantidad de memoria Swap *cached* (en bytes).
- *Free memory*: Cantidad de memoria libre (en bytes).
- *Percent free memory*: Porcentaje de memoria libre.
- *Swap cached percent*: Porcentaje de memoria Swap *cached*.
- *Swap free*: Cantidad de memoria Swap libre (en bytes).
- *Swap free percent*: Porcentaje de memoria Swap libre.
- *Total Memory*: Cantidad total de memoria del *Host* (en bytes).

- *Total Swap*: Cantidad total de memoria Swap (en bytes).
- *Used memory*: Cantidad total de memoria usada (en bytes).
- *Used Swap*: Cantidad total de memoria Swap usada (en bytes).
- *Nic [x] TX* y *Nic [x] RX*: Ratio de transferencia de la interfaz de red [x] (en bytes/segundo). Se genera uno por cada interfaz de red detectada.
- *Nic [x] errors TX* y *Nic [x] errors RX*: Número de errores de transmisión de la interfaz de red [x]. Se genera uno por cada interfaz de red detectada.
- *User CPU*: Porcentaje de CPU usada por el usuario.
- *System CPU*: Porcentaje de CPU usada por el sistema.
- *CPU Idle*: Porcentaje de CPU ociosa.
- *CPU Load*: Carga media de CPU de los últimos 5 minutos.
- *KSM CPU*: Porcentaje de CPU usado por KSM.
- *Active VM*: Número de máquinas virtuales activas en el *Host*.
- *Migrating VM*: Número de máquinas virtuales en migración en el *Host*.
- *Total VM*: Número total de máquinas virtuales del *Host*.
- *Fence Status*: Estado del *fencing* del *host*.

Virtual Machine

- *Status*: Estado de la máquina virtual.
- *Disk [x] read* y *Disk [x] write*: Tasa de lectura y escritura del disco x (bytes/segundo). Se genera uno por cada disco (almacenamiento) detectado.
- *Disk [x] size*: Tamaño del disco x (en bytes). Se genera uno por cada disco detectado.
- *Disk [x] status*: Estado del disco x. Se genera uno por cada disco detectado.
- *Nic [x] TX* y *Nic [x] RX*: Ratio de transferencia y de recepción para la interfaz de red [x] (en bytes/segundo). Se genera uno por cada interfaz de red detectada.
- *Nic [x] errors TX* y *Nic [x] errors RX*: Número de errores de transmisión y de recepción para el la interfaz de red [x]. Se genera uno por cada interfaz de red detectada.
- *Installed memory*: Cantidad de memoria instalada (en bytes).
- *Percent free memory*: Porcentaje de memoria libre.
- *Used memory*: Cantidad de memoria usada (en bytes).
- *Stateless*: Estado de la funcionalidad Stateless.
- *HA Status*: Estado de la funcionalidad de HA.
- *Total CPU*: Porcentaje total de CPU usada por la máquina virtual.
- *Hypervisor CPU*: Porcentaje de CPU del Hypervisor usado por la máquina virtual.
- *Guest CPU*: Porcentaje de CPU del *Host* que usa la máquina virtual.

Eventos

- *Event [x]*: Descripción del evento x ocurrido en el sistema. Se creará uno por cada evento detectado en los Agentes afectados.

Gestión y visualización de la arquitectura RHEV

Tareas de reconocimiento

Existe la posibilidad de crear tareas personalizadas de reconocimiento gracias al [Discovery server](#).

Instalación de extensiones RHEV View y RHEV Manager

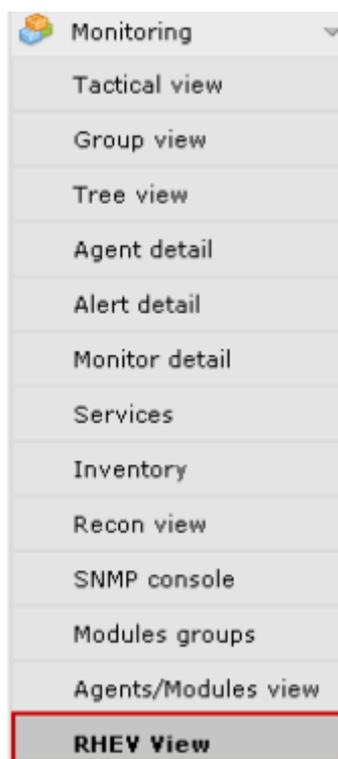
Para instalar las extensiones simplemente debe copiar el contenido de la carpeta `extensions`, que encontrará al descomprimir el *plugin*, en la correspondiente carpeta `extensions` de la parte Enterprise de la Consola de Pandora FMS. El comando a ejecutar es el siguiente:

```
cp -R extensions/* <pandora_console_dir>/enterprise/extensions/
```

Desde ese momento estarán disponibles las extensiones de monitorización RHEV.

Uso de la extensión RHEV View

Para usar la extensión RHEV View solo tiene que pulsar en la opción RHEV View dentro del submenú Monitoring.



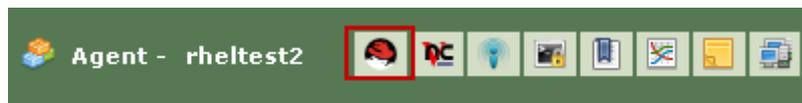
La extensión mostrará un mapa con todos los componentes de la arquitectura RHEV descubiertos por el *plugin*.

Uso de la extensión RHEV Manager

La extensión RHEV Manager está disponible en la vista de operación de los agentes de Pandora FMS que se correspondan con máquinas virtuales dentro de la arquitectura de virtualización RHEV.

Esta extensión usa el comando curl, por lo que será necesario que esté instalado y sea accesible para el servidor web que soporte la Consola de Pandora FMS.

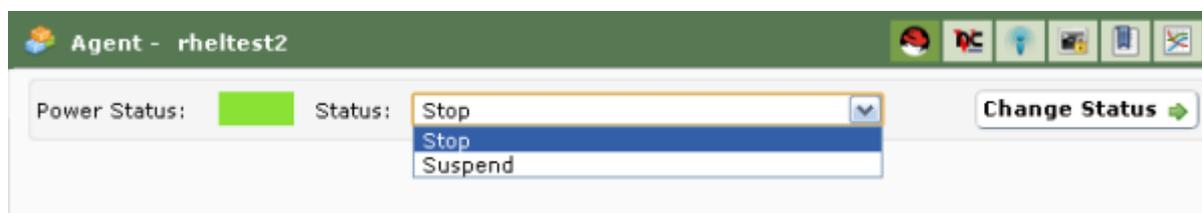
Para acceder a la extensión pulse en el botón con el logo de Red Hat que encontrará junto con los demás pestañas del agente.



La extensión permite gestionar máquinas virtuales (encender, apagar y suspender) sin necesidad de abrir la consola de gestión de RHEV. En la extensión se muestra el estado actual de la máquina virtual con un código de colores:

- Verde = Encendida.
- Naranja = Suspendida.
- Gris = Detenida.

Con un combo con los estados disponibles a los estados que se puede llevar la máquina virtual al pulsar el botón Change Status.



Si escoge el estado Stop para detener la máquina virtual, la extensión conectará con la API RHEV y enviará la orden. El resultado será el cambio de estado en la máquina virtual y las opciones del combo.

El paso entre algunos estados no es automático, como por ejemplo del estado Stop a Start. En este caso, la extensión irá mostrando el estado de la máquina virtual según vaya cambiando en la arquitectura de virtualización.



Configuración del plugin RHEV para Agente

La configuración del *plugin* de Agente se realiza mediante un archivo de configuración cuyo

nombre por defecto es `rhev-plugin.conf`.

Por defecto, el *plugin* de Agente selecciona todas las entidades y crea todos los Módulos correspondientes con unos valores predeterminados para el nombre y la descripción. Todos estos aspectos, así como variables generales del *plugin*, se pueden configurar a través del archivo de configuración.

Archivo de configuración del plugin RHEV

El archivo de configuración tiene dos áreas bien diferenciadas: las variables globales y la configuración de la monitorización.

La sección de variables globales comienza con el *token* Configuration y contiene la información de configuración del *plugin*. Los parámetros permitidos en esta sección son:

- *module_name* : Nombre del Módulo del Agente con el estado de la ejecución del *plugin*.
- *server* : Nombre del *host* que sirve la API de RHEV.
- *user* : Usuario en formato *user@domain* para conectarse a la API.
- *pass* : Contraseña para conectarse a la API.
- *cert* : Ruta del certificado para conectarse a la API.
- *temporal* : Directorio temporal.
- *logfile* : Archivo de registro o *log*.
- *transfer_mode* : Modo de transferencia. Puede tomar los valores: `local` o `tentacle`.
- *tentacle_ip* : Dirección IP del servidor Tentacle al cual enviar la información. Típicamente estará situado en la misma máquina que el servidor de Pandora FMS. Esta opción solo se usa si `transfer_mode` tiene el valor `tentacle`.
- *tentacle_port* : Puerto del servidor Tentacle. Esta opción solo se usa si `transfer_mode` tiene el valor `tentacle`.
- *tentacle_opts* : Opciones de envío de datos para Tentacle. Esta opción solo se usa si `transfer_mode` tiene el valor `tentacle`.

La sección de configuración de la monitorización se divide en varias subsecciones. La primera subsección tiene como *token* Reject y sirve para listar las entidades del entorno de virtualización que serán descartadas de la monitorización. Para descartar una entidad será necesario poner el nombre de la misma en esta lista. Por ejemplo:

```
#Dismissed entities
Reject
mv1
mv_Windows10
mv_WebServer1
...
```

Es posible descartar todas las entidades de un mismo tipo, por ejemplo todos los *hosts*, todas las máquinas virtuales, etc. Los *token* para cada entidad son: `all_dc` (Data Center), `all_host` (Hosts), `all_network` (Networks), `all_storage` (Storage Domain), `all_cluster` (Cluster), `all_vm` (Máquinas Virtuales). Ejemplo de uso de estos *tokens*:

```
#Dismissed entities
Reject
all_dc
all_host
all_network
all_storage
all_cluster
all_vm
```

La segunda sección tiene como *token* Rename y sirve para cambiar los nombres de las entidades monitorizadas a través del *plugin*. Esta funcionalidad es muy útil si se quiere combinar la monitorización de agentes software con datos extraídos de la API en el mismo Agente de Pandora FMS. La configuración de esta sección se realiza poniendo primero el nombre antiguo y a continuación el nuevo separado por un espacio; por ejemplo:

```
#Rename entities
Rename
mv_WebServer1 WebServer1
mv_Windows10 Windows10 Test
...
```

Las siguientes subsecciones se corresponden con la configuración de monitorización para cada entidad. Cada entidad tiene su propio *token*, siendo los siguientes: DataCenter, StorageDomain, Network, Cluster, Host y VM. Para cada una de estas entidades es posible definir los Módulos que estarán deshabilitados o definir unos valores personalizados para el nombre, la descripción y los rangos de máximos y mínimos para los estados Warning y Critical. Un ejemplo sería el siguiente:

```
#VM Modules
VM
status disabled
errors_total_tx name = Errores TX Net [%s]; desc = Errores totales TX de red;
limits = 60 70 71 100
memory_used name = Memoria en uso; desc = Memoria usada por la maquina virtual;
limits = 256 1024 1025 2048
...
```

Cada línea de configuración de los módulos de monitorización se corresponde dos opciones disponibles:

- <modulo> disabled: El Módulo NO se creará.
- <modulo> name = <nombre>; desc = <descripcion>; limits = <min_warning> <max_warning> <min_critical> <max_critical> El Módulo se creará con el nombre y la descripción proporcionados y además se definirán los umbrales para los máximos y mínimos de los valores Warning y Critical.

Es muy importante tener en cuenta la estructura de las líneas del archivo de configuración y sobre todo ver que el carácter ; va pegado al nombre y la descripción del módulo. Estas dos líneas *no son equivalentes* (véanse los espacios antes del carácter ;):

```
errors_total_tx name = Errores TX Net [%s]; desc = Errores totales TX de red;
limits = 60 70 71 100 #Correcto
errors_total_tx name = Errores TX Net [%s] ; desc = Errores totales TX de red
; limits = 60 70 71 100 #Incorrecto
```

Los Módulos están referenciados por su nombre corto, un nombre equivalente más sencillo de escribir en la línea de comandos. La tabla de correspondencia entre nombres cortos y expandidos está en la siguiente sección.

Ejemplo de configuración para las máquinas virtuales, sección VM:

Para la monitorización de las máquinas virtuales se han definido una serie de Módulos habilitados o no en la sección VM del archivo de configuración. Más concretamente: el Módulo status ha sido deshabilitado y para los Módulos errors_total_tx y memory_used se han definido unos valores personalizados. Los demás Módulos que no aparecen en el listado se crearán con los valores por defecto. Con esta configuración, el Módulo memory_used tomará los siguientes valores:

- *Nombre:* Memoria en uso.
- *Descripción:* Memoria usada por la máquina virtual.
- *Min Warning:* 256.
- *Max Warning:* 1024.
- *Min Critical:* 1025.
- *Max Critical:* 2048.

Los Módulos se generan dinámicamente; por ejemplo, dos relativos a discos o interfaces de los cuales se crea uno por cada elemento detectado. Tienen una sintaxis especial para el nombre del Módulo, que es la siguiente:

```
errors_total_tx name = Errores TX Net [%s]; desc = Errores totales TX de red;
limits = 60 70 71 100
```

En estos casos, como el nombre tiene parte dinámica, lo que se permite es usar la macro %s que será sustituida por el *plugin* por la parte variable del nombre del Módulo.

Por ejemplo, el Módulo errors_total_tx tiene por nombre por defecto:

```
Nic [nic1] errors TX
```

Pasará a llamarse

```
Errores TX Net [nic1]
```

Siendo nic1 la parte dinámica del nombre del módulo.

Todos los errores relativos al archivo de configuración son presentados en el *log* definido en el archivo de

configuración y además son enviados como un Módulo asíncrono a Pandora FMS que quedará reflejado como un Módulo dentro del Agente que ejecuta el *plugin*.

Además de las secciones propias de cada elemento de la arquitectura, el archivo de configuración tiene una sección común para los Eventos. Esta sección se define con el token EventCodes y en él se listarán los códigos de los eventos a monitorizar; por ejemplo:

```
EventCodes
30
920
980
509
956
```

Si no define esta sección, no se realizará la monitorización de eventos.

Dividir la carga de la monitorización entre varios Agentes Software

Mediante el archivo de configuración del *plugin* de Agente es posible dividir la carga de monitorización de la infraestructura de virtualización RHEV.

Para ello se repartirán las entidades a monitorizar entre los distintos Agentes. Suponga que tiene la siguiente arquitectura:

```
DC1
|
|- Cluster 1.1
   |- c1.1mv1
   |- c1.1mv2
   |- c1.1mv3

|- Cluster 1.2
   |- c1.2mv1
   |- c1.2mv2
   |- c1.2mv3

DC2
|
|- Cluster 2.1
   |- c2.1mv1
   |- c2.1mv2
   |- c2.1mv3

|- Cluster 2.2
   |- c2.2mv1
   |- c2.2mv2
```

```
| - c2.2mv3
```

Una forma de dividir la carga sería asignando un Datacenter a cada uno de los Agentes Software; para ello usaríamos la funcionalidad para descartar entidades a monitorizar (*token Reject*).

El primer Agente Software monitoriza el Datacenter DC1 y descarta las entidades de DC2.

```
Reject
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
c2.2mv2
c2.2mv3
```

El segundo Agente Software monitoriza el Datacenter DC2 y descarta las entidades de DC1.

```
Reject
DC1
Cluster 1.1
Cluster 1.2
c1.1mv1
c1.1mv2
c1.1mv3
c1.2mv1
c1.2mv2
c1.2mv3
```

También podríamos dividir la carga basándonos en los clústeres, por ejemplo. Para cada clúster de los dos Datacenters se asignará un agente de los cuatro primeros.

Agente Software 1, monitorizar Cluster 1.1 y descarta las demás entidades.

```
Reject
DC1
Cluster 1.2
c1.2mv1
c1.2mv2
c1.2mv3
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
```

```
c2.2mv2
c2.2mv3
```

Agente Software 2, monitorizar Cluster 1.2 y descarta las demás entidades.

```
Reject
DC1
Cluster 1.1
c1.1mv1
c1.1mv2
c1.1mv3
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
c2.2mv2
c2.2mv3
```

Agente Software 3, monitorizar Cluster 2.1 y descarta las demás entidades.

```
Reject
DC1
Cluster 1.1
Cluster 1.2
c1.1mv1
c1.1mv2
c1.1mv3
c1.2mv1
c1.2mv2
c1.2mv3
DC2
Cluster 2.2
c2.2mv1
c2.2mv2
c2.2mv3
```

Agente Software 4, monitorizar Cluster 2.2 y descarta las demás entidades.

```
Reject
DC1
Cluster 1.1
Cluster 1.2
c1.1mv1
c1.1mv2
c1.1mv3
c1.2mv1
c1.2mv2
c1.2mv3
```

```
DC2
Cluster 2.1
c2.1mv1
c2.1mv2
c2.1mv3
```

La configuración de entidades descartadas es totalmente flexible y se podría llegar a dividir la carga asignando varias entidades a cada Agente Software.

Nutanix

Funcionamiento del plugin Nutanix

El *plugin* Nutanix® es un programa escrito en Perl, que se conectará a la API REST de Nutanix PRISM®, recuperando las métricas necesarias para monitorizar los siguientes elementos:

- Clústeres Nutanix®.
- Dispositivos de almacenamiento.
- Contenedores.
- Máquinas Virtuales.
- *Hosts*.
- Estado de los procesos de replicación.

Requisitos del plugin Nutanix

Para poder recuperar la información de la API REST, necesita:

- La dirección IP/ FQDN del portal.
- Un *usuario* con permisos de lectura sobre la API.
- La *contraseña* de dicho usuario.

En cuanto a la comunicación de los resultados de la monitorización a su Pandora FMS necesita:

- El *modo de transferencia de la información*, sea local o vía Tentacle.
- En caso de ser local, la dirección del directorio donde han de entregarse los ficheros XML con los resultados, así como permisos de escritura en dicho directorio.
- En caso de ser comunicación vía Tentacle, será necesario poder conectar contra la dirección IP o FQDN del servidor Pandora FMS, el puerto utilizado por su instalación Tentacle, la localización del cliente de Tentacle así como cualquier opción extraordinaria que haya definido.

Instalación del plugin Nutanix

Descargue los archivos requeridos por el *plugin* de la [biblioteca de Módulos](#). Transfiera los archivos al equipo remoto desde donde desea realizar la monitorización de su infraestructura Nutanix® y extraiga los archivos del *plugin*:

```
tar xvzf pandora_nutanix.tar.gz
```

Configuración del plugin Nutanix

Se declaran los siguientes campos:

Nutanix API configuration

- `nx_fqdn`: Dirección del servidor principal Prism.
- `nx_port`: Puerto en el que la API REST está publicada (por defecto 9440).
- `nx_user`: Usuario con privilegios de lectura sobre la API REST.
- `nx_pass`: Contraseña de dicho usuario.
- `use_https`: Usar https (1) o no (0)
- `nx_rest_version`: Versión del API Rest (por defecto 'v1').

Nutanix agent configuration

- `agent_interval`: Intervalo de los Agentes generados por el *plugin* (por defecto 300).
- `agent_group`: Grupo al que pertenecerán los Agentes generados (si se comenta 'autocreate_group' en la configuración de su PandoraServer), por defecto Nutanix.
- `module_interval`: Intervalo de los Módulos de los agentes generados (factor de multiplicación, por defecto 1).
- `module_tags`: Etiquetas asociadas a los nuevos módulos de los agentes generados.
- `module_group`: Grupo al que pertenecerán los nuevos Módulos.

Configuración de la comunicación hacia el servidor de Pandora FMS

- `mode`: Modo de transferencia de datos, "local" o "tentacle".
- `tentacle_ip`: dirección IP del servidor Pandora FMS, solo aplica en modo Tentacle.
- `tentacle_port`: Puerto en el que el servicio Tentacle está escuchando.
- `tentacle_opts`: Cualquier opción extra que haya configurado en su servicio Tentacle.
- `tentacle_client`: Ruta completa a su cliente Tentacle.
- `temp`: Directorio de trabajo temporal.
- `local_folder`: Ruta de entrega para el modo de transferencia de datos "local".

Filtros

- `cluster_monitoring`: Habilitar (1) o no (0) la monitorización de clústeres.
- `storage_monitoring`: Habilitar (1) o no (0) la monitorización de dispositivos de almacenamiento.
- `container_monitoring`: Habilitar (1) o no (0) la monitorización de los contenedores de almacenamiento.
- `vm_monitoring`: Habilitar (1) o no (0) la monitorización de máquinas virtuales.
- `host_monitoring`: Habilitar (1) o no (0) la monitorización de servidores de máquinas virtuales (nodos Nutanix).
- `pd_monitoring`: Habilitar (1) o no (0) la monitorización de dominios de protección.

Personalizaciones

- `cluster_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo clúster.
- `storage_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo dispositivo de

almacenamiento.

- `host_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo servidor de máquinas virtuales (nodos Nutanix).
- `container_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo contenedores de almacenamiento.
- `vm_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo máquina virtual.
- `pd_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo dominio de protección

Reglas de generación de Módulos

- `vm_stat`: Regla para agregado de Módulos para la monitorización de máquinas virtuales, por defecto `hypervisor_cpu_usage_ppm|hypervisor_memory_usage_ppm|. *avg.*`, esto indica los Módulos extraordinarios que se generarán, cuando el nombre de la métrica coincida con las expresiones regulares indicadas en este campo. Agregue el valor `.*` para monitorizar todas las métricas disponibles.
- `host_stat`: Regla para agregado de Módulos para la monitorización de servidores de máquinas virtuales (nodos Nutanix), por defecto `hypervisor_cpu_usage_ppm|hypervisor_memory_usage_ppm|. *avg.*`, esto indica los Módulos extraordinarios que se generarán, cuando el nombre de la métrica coincida con las expresiones regulares indicadas en este campo. Agregue el valor `.*` para monitorizar todas las métricas disponibles.
- `pd_stat`: Regla para agregado de Módulos para la monitorización de dominios de protección, por defecto `replication_transmitted_bandwidth_kBps|replication_total_transmitted_bytes`, esto indica los Módulos extraordinarios que se generarán, cuando el nombre de la métrica coincida con las expresiones regulares indicadas en este campo. Agregue el valor `.*` para monitorizar todas las métricas disponibles.

Renombrado de entidades

- `RENAME aaa TO bbb`: Regla para el renombrado de entidades, puede definir tantas directivas como elementos necesite renombrar.

Exclusión de entidades

- `REJECT aaa`: Regla para la exclusión de monitorización de entidades, puede definir tantas directivas como elementos necesite excluir.

Ejecución del plugin Nutanix

Se recomienda ejecutar el *plugin* de forma remota desde un equipo con acceso tanto a Pandora FMS server como a su infraestructura Nutanix® a monitorizar.

Ejecución manual:

```
./pandora_nutanix-linux-x64 pandora_nutanix.conf
```

Puede automatizar la ejecución del *plugin* en el cron de sistema agregando la siguiente línea a `/etc/crontab`.

```
/5 * * * * root /path/to/plugin/pandora_nutanix-linux-x64  
/path/to/plugin/pandora_nutanix.conf
```

XenServer

Funcionamiento del plugin XenServer

El plugin Pandora FMS para la monitorización de entornos Xen está escrito en Python. Utiliza XenAPI para recuperar toda la información necesaria. Permite la monitorización de los siguientes tipos de elementos:

- Sistemas virtualizados en Xen.
- Recursos de almacenamiento.
- Propio XenServer 6.5 y 7.2 (host).

Requisitos del plugin XenServer

Es imprescindible que el sistema que ejecute el *plugin* disponga de los siguientes requisitos:

- Python instalado
- Bibliotecas Python instaladas: XenAPI y xmltodict.
- Acceso a la API de su XenServer (web, habilite el tráfico desde el equipo que ejecuta el plugin al puerto 443 o el 80 del XenServer).
- Se recomienda que las máquinas virtuales tengan Xen Server Tools instalada, ya que la información disponible en caso contrario es bastante escasa.

Instalación del plugin XenServer

Descargue una copia del *plugin* de Pandora FMS para XenServer de la [biblioteca de módulos](#). Puede desplegarlo en el equipo que prefiera (MS Windows® o GNU/Linux®), extrayendo el contenido del archivo en un directorio no volátil desde donde pueda ejecutarlo, ya bien sea utilizando el Agente de Pandora FMS o el cron de sistema.

Configuración del plugin XenServer

Configuración disponible para el plugin Pandora FMS para Xen:

Bloque de configuración [CONF]

- xen_server_ip: Dirección IP/FQDN del servidor Xen.
- user: Usuario con permisos de consulta sobre la API de Xen.
- password: Contraseña del usuario
- temporal: Directorio de trabajo temporal

Bloque de configuración [PANDORA]

- `tentacle_client`: Ubicación del binario del cliente de Tentacle.
- `tentacle_ip`: Dirección IP donde el servicio Tentacle está escuchando.
- `tentacle_port`: Puerto donde el servicio Tentacle está escuchando.
- `logfile`: Ruta completa al fichero de *log*.
- `interval`: Intervalo de los agentes generados.
- `group`: Grupo asignado a los agentes generados.

Bloque de configuración [TUNNING]

- `time_adjustment`: Parámetro que permite el ajuste de las posibles diferencias de tiempo entre el equipo que ejecuta el plugin y el servidor Xen. (por defecto=10, medido en segundos)
- `scan_vm_ip`: Parámetro que permite definir si el plugin tratará de obtener las direcciones IP de las VMs del servidor de Xen. Solo se pueden tomar las direcciones IP de aquellas VM con las XenTools instaladas. Se puede habilitar (`scan_vm_ip=true`) o deshabilitar (`scan_vm_ip=false`). Si no se especifica se considera habilitado.

Bloque de configuración [RENAME]

- `xen_element_name = pandora_agent_name`: En este bloque se pueden definir tantas entradas como se quiera con este formato. Permite cambiar los nombres de los elementos del Xen Server a otros distintos para ser usados como nombres de agentes en Pandora FMS. Se pueden renombrar las VMs, los SRs y el propio Xen Server.

Ejecución del plugin XenServer

Puede programar la ejecución del plugin desde cualquier agente de Pandora FMS, agregando lo siguiente a la configuración del mismo:

```
module_plugin python "<ruta>\xen-plugin.py" "<ruta>\xen-plugin.conf"
```

Para programarlo a través del cron de sistema puede añadir la línea siguiente a `/etc/crontab`:

```
/5 * * * * root python "<ruta>\xen-plugin.py" "<ruta>\xen-plugin.conf">
/dev/null 2>&1
```

Si se ejecuta el *plugin* manualmente la salida debe ser similar a la que sigue:

```
python "<ruta>\xen-plugin.py" "<ruta>\xen-plugin.conf"
<module>
<name><![CDATA[XenServer Plugin]]></name>
<type><![CDATA[async_string]]></type>
<description><![CDATA[Result of XenServer Plugin execution]]></description>
<data><![CDATA[OK]]></data>
</module>
```

OpenNebula

Funcionamiento del plugin OpenNebula

El plugin Pandora FMS para la monitorización de entornos OpenNebula está escrito en Perl. Se ejecuta de manera local en el servidor OpenNebula y recuperará toda la información necesaria utilizando los propios comandos de gestión de OpenNebula. Permite la monitorización de los siguientes tipos de elementos:

- Clústeres.
- *Hosts*.
- Máquinas virtuales.
- Recursos de almacenamiento.

Requisitos del plugin OpenNebula

Es imprescindible que el sistema que ejecute el plugin disponga de los siguientes requisitos:

- Perl disponible en el equipo
- Usuario con privilegios para ejecutar los siguientes comandos:
 - `onehost`.
 - `onecluster`.
 - `onedatastore`.

El funcionamiento del *plugin* se ha probado satisfactoriamente en sistemas OpenNebula 5.X.X .

Instalación del plugin OpenNebula

Descargue su copia del plugin de Pandora FMS para OpenNebula de la [biblioteca de módulos](#). Debe extraer el contenido del archivo en un directorio no volátil desde donde pueda ejecutarlo, ya bien sea utilizando el agente de Pandora FMS o el cron de sistema.

```
unzip pandora_OpenNebula.zip
```

Configuración del plugin OpenNebula

Configuración de la comunicación hacia el servidor de Pandora FMS

- `mode`: Modo de transferencia de datos, "local" o "tentacle".
- `tentacle_ip`: Dirección IP del servidor Pandora FMS, solo aplica en modo tentacle.
- `tentacle_port`: Puerto en el que el servicio Tentacle está escuchando.

- `tentacle_opts`: Cualquier opción extra que haya configurado en su servicio Tentacle.
- `tentacle_client`: Ruta completa a su cliente Tentacle.
- `temp`: Directorio de trabajo temporal.
- `local_folder`: Ruta de entrega para el modo de transferencia de datos "local".

Configuración del Agente

- `agent_interval`: Intervalo del agente, por defecto 300.
- `agent_group`: Grupo del Agente, por defecto OpenNebula.

Personalización de los Módulos OpenNebula

- `MODULE_GROUP`: Grupo de los módulos, por defecto OpenNebula.
- `MODULE_INTERVAL`: Intervalo de los módulos (multiplicador), por defecto 1.
- `MODULE_TAGS`: Etiquetas para los Módulos.

Personalización de nombres

- `cluster_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo clúster.
- `host_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo servidor de máquinas virtuales.
- `storage_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo dispositivo de almacenamiento.
- `vm_agent_header`: Cabecera para el nombre del Agente de dispositivos de tipo máquina virtual.

Filtros

- `cluster_monitoring`: Habilitar (1) o no (0) la monitorización de clústeres.
- `host_monitoring`: Habilitar (1) o no (0) la monitorización de servidores de máquinas virtuales.
- `storage_monitoring`: Habilitar (1) o no (0) la monitorización de dispositivos de almacenamiento.
- `vm_monitoring`: Habilitar (1) o no (0) la monitorización de máquinas virtuales.

Renombrado de entidades

`RENAME aaa TO bbb`: Regla para el renombrado de entidades, puede definir tantas directivas como elementos necesite renombrar.

Exclusión de entidades

`REJECT aaa`: Regla para la exclusión de monitorización de entidades, puede definir tantas directivas como elementos necesite excluir.

Ejecución del plugin OpenNebula

Para programarlo a través del cron de sistema puede añadir la línea siguiente a `/etc/crontab`:

```
/5 * * * * root "<ruta>/pandora_opennebula" "<ruta>/pandora_opennebula.conf">
/dev/null 2>&1
```

Si ejecuta el plugin manualmente la salida debe ser similar a la que sigue:

```
[root@valhalla ~]# ./pandora_opennebula pandora_opennebula.conf
[root@valhalla ~]# echo $?
0
```

IBM HMC

Este *plugin* permite monitorizar equipos de virtualización IBM AIX a través de la consola de administración de hardware HMC. Este *plugin* recolectará información de todas las particiones lógicas creadas en un entorno AIX administrado por un sistema HMC, creando un Agente por cada servidor administrado, cada partición lógica y cada servidor IO virtual.

Para recoger la información vía SSH, el plugin puede utilizar tres modos de trabajo:

1. Basado en expect utilizando el script `ssh_launcher.sh`.
2. Basado en `Net::SSH::Perl`
3. Basado en `Net::SSH::Expect`

Para complementar la información capturada también se realizarán consultas contra la API REST, por defecto en:

```
https://fqdn:12443/rest/api/{root_element}
```

Requisitos plugin IBM HMC

Los parámetros necesarios que tiene que proporcionar el área que requiere los servicios de monitorización son:

- Nombre de usuario para autenticarse en el sistema HMC (solo lectura).
- El usuario debe tener permiso para poder conectarse a la API REST y para hacer login en la shell del HMC y ejecutar los siguientes comandos (*como mínimo*):
 - `lssyscfg`
 - `lshwres`
- Contraseña de dicho usuario.
- Ubicación (FQDN/IP) del HMC (p.e. `myhmc.mydomain`)
- URL base de la API rest del HMC, por ejemplo:

https://myhmc.mydomain:12443

Módulos generados por el plugin

Los parámetros que monitoriza el plugin son (agrupados por tipo de elemento):

- Current logical partitions
- Max logical partitions
- Max memory available
- Max memory installed
- Proc pool DefaultPool current proc units
- Proc pool DefaultPool max proc units
- Proc pool DevelopmentPool current proc units
- Proc pool DevelopmentPool max proc units
- Proc pool ProductionPool current proc units
- Proc pool ProductionPool max proc units
- Proc pool TestPool current proc units
- Proc pool TestPool max proc units
- Proc pool VIOPool current proc units
- Proc pool VIOPool max proc units
- Processor pools configured
- Processor units available
- Processor units installed
- State
- UUID
- Virtual proc units max

LPAR:

- Auto start : Configuración de particiones lógicas autoinicio.
- LPAR type : Tipo de partición lógica.
- LPAR UUID : Usado para consultar los HMC API.
- Max memory : Memoria máxima.
- Max memory : Memoria disponible.
- Processor units available : Unidades de procesamiento disponibles.
- Processor units current : Unidades de procesamiento instaladas
- RMC IP address : Dirección IP RMC.
- RMC state : Estado del RMC en LPAR
- State : Estado de la partición lógica.
- Virtual proc units : Unidades de procesamiento virtual asignadas a este LPAR.

Virtual IO:

- Auto start : Configuración de particiones lógicas autoinicio.
- LPAR type : Tipo de partición lógica.
- LPAR UUID : Usado para consultar los HMC API.
- Max memory : Memoria máxima.
- Max memory current : Memoria disponible.
- Processor units available: Unidades de procesamiento disponibles
- Processor units current : Unidades de procesamiento instaladas

- RMC IP address : Dirección IP RMC.
- RMC state RMC : Estado del RMC en LPAR.
- State : Estado de la partición lógica.
- Virtual proc units : Unidades de procesamiento virtual asignadas a este LPAR.

Configuración del plugin IBM HMC

Configuración de comunicación IBM HMC hacia el servidor de Pandora FMS

- mode: Modo de transferencia de datos, "local" o "tentacle".
- tentacle_ip: Dirección IP del servidor Pandora FMS, solo aplica en modo tentacle.
- tentacle_port: Puerto en el que el servicio tentacle está escuchando.
- tentacle_opts: Cualquier opción extra que haya configurado en el servicio tentacle.
- tentacle_client: Ruta completa a su cliente tentacle.
- temp: Directorio de trabajo temporal.
- local_folder: Ruta de entrega para el modo de transferencia de datos "local".

Configuración de acceso a HMC

- hmc_host: IP o FQDN de la HMC.
- hmc_user: Usuario con permiso de lectura.
- hmc_pass: Contraseña.
- as_agent_plugin: La salida del *plugin* se devolverá en formato XML para ejecuciones programadas con el Agente de Pandora FMS (`as_agent_plugin = 1`). O salida estándar (`as_agent_plugin = 0`) para ejecuciones programadas con el cron de sistema o realizadas como *plugin* de servidor.

Configuración del agente IBM HMC

- agent_name: Opcional, indicar un nombre para el Agente padre, por defecto ``hostname``
- agent_interval: Intervalo del Agente, por defecto 300 segundos.
- agent_group: Grupo del Agente, por defecto IBM.

Personalización de los módulos IBM HMC

- module_group: Grupo de los Módulos, por defecto IBM.
- module_interval: Intervalo de los Módulos (multiplicador), por defecto 1.
- module_tags: Etiquetas para los módulos.

Renombrado de entidades

Para el renombrado de entidades se utiliza un renombrado por bloque:

```
rename
MyLPAR_NAME TO my new name
MyLPAR_NAME2 TO my second new name
rename_end
```

Ejecución del plugin IBM HMC

El plugin de Pandora FMS para la monitorización de sistemas IBM AIX a través de HMC se despliega de la siguiente manera:

Configurando el parámetro `as_agent_plugin` a 1 (ejecución como plugin de agente):

```
module_plugin /usr/bin/perl pandora_hmc.pl pandora_hmc.conf
```

Configurando el parámetro `as_agent_plugin` a 0 (ejecución como plugin de servidor):

```
# /etc/crontab
*/5 * * * * root /usr/bin/perl /root/hmc/pandora_hmc.pl
/root/vmware/pandora_hmc .conf
```

HPVM

Funcionamiento del plugin HPVM

Este *plugin* permite monitorizar equipos de virtualización HPVM. Se lanza como un *plugin* de Agente, generando en paralelo un agente más por cada equipo virtualizado alojado en el sistema monitorizado.

Para recoger la información se utilizan comandos locales.

Requisitos del plugin HPVM

Verifique cada uno de los siguientes pasos:

- Desplegar un Agente de Pandora FMS en el equipo que se desee monitorizar.
- Disponer de un usuario con permisos para ejecutar el plugin
- Este usuario debe disponer de permisos para ejecutar el comando `hvvmstatus` para poder interpretar la salida:
 1. `hvvmstatus`.
 2. `hvvmstatus -X`.
 3. `hvvmstatus -r -X`.

Instalación del plugin HPVM

Descargue una copia del *plugin* de Pandora FMS para HPVM HP Virtualization Manager monitoring de la [biblioteca de módulos](#). Puede programar la ejecución utilizando colecciones y el Agente de

Pandora FMS desplegado o bien extraer el contenido del archivo en un directorio no volátil desde donde pueda ejecutarlo a través del cron de su sistema.

```
unzip pandora_HPVM.zip
```

Configuración del plugin HPVM

Configuración de comunicación HPVM hacia el servidor de Pandora FMS

- mode: Modo de transferencia de datos, "local" o "tentacle".
- tentacle_ip: Dirección IP del servidor Pandora FMS, solo aplica en modo tentacle.
- tentacle_port: Puerto en el que el servicio Tentacle está escuchando.
- tentacle_opts: Cualquier opción extra que haya configurado en su servicio Tentacle.
- tentacle_client: Ruta completa a su cliente Tentacle.
- temp: Directorio de trabajo temporal.
- local_folder: Ruta de entrega para el modo de transferencia de datos "local".

Configuración del Agente HPVM

- agent_name: Opcional, indicar un nombre para el Agente padre, por defecto hostname
- agent_interval: Intervalo del Agente, por defecto 300.
- agent_group: Grupo al que pertenecerán los agentes, por defecto HPVM.

Personalización de los módulos HPVM

- module_group: Grupo de los Módulos.
- module_interval: Intervalo de los Módulos (multiplicador), por defecto 1.
- module_tags: Etiquetas para los Módulos.

Ejecución del plugin HPVM

Ejecutando el *plugin* desde el Agente de Pandora FMS, aparecerá en el fichero de configuración del Agente:

```
module_plugin /usr/bin/perl pandora_hpvm.pl pandora_hpvm.conf
```

Para una prueba manual, configure el *plugin* siguiendo los pasos descritos, puede lanzarlo como sigue:

```
perl pandora_hpvm.pl pandora_hpvm.conf
```

[Volver al índice de documentación de Pandora FMS](#)