Monitorización de redes y servicios en RECETGA

Caso práctico de implementación de Zabbix como servidor de monitorización de red en RECETGA

Una herramienta de monitorización de redes y servicios ayuda a los departamentos de IT de empresas y proveedores de servicios a mejorar su productividad, así como a incrementar el tiempo de disponibilidad o SLA de sus servicios. Todo esto redunda en una mejor experiencia de uso del usuario final de los mismos.

Una infraestructura de monitorización que incluya soporte 24x7 contribuirá a que todos los incidentes catalogados sean detectados, priorizados, escalados y resueltos eficientemente en la menor brevedad de tiempo posible, resultando en un menor tiempo de resolución de incidencias y permitirá reducir los costes de soporte de nivel 1, asociados a la gestión de estos servicios.

Este documento ofrece un análisis en detalle de la implantación del nuevo servicio de gestión y monitorización de la infraestructura de red de comunicaciones de la Red de Ciencia y Tecnología de Galicia (RECETGA).

Frco. Javier Rial Rodríguez, Técnico de Comunicaciones CESGA. 3 años como operador del NOC CESGA y desarrollador/implementador de la nueva infraestructura de monitorización del CESGA basada en Zabbix <fjrial@cesga.es>

Índice de contenidos

Control de cambios

Versión	Autor	Cambios
v 0	Francisco Javier Rial Rodríguez	Redacción inicial del documento.
v 1	Natalia Costas Lago	Revisión del documento.
v 2	Francisco Javier Rial Rodríguez	Revisión del documento.

1 Introducción

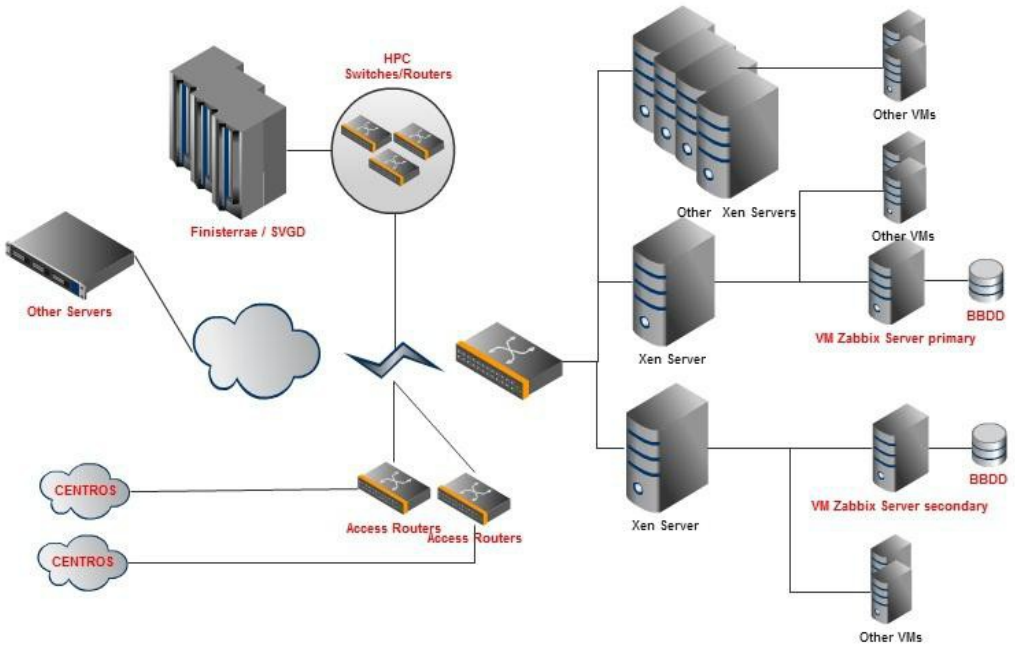
1.1 ¿Qué es RECETGA?

RECETGA[1] es una infraestructura de conectividad de alta capacidad gestionada por el CESGA que dota de servicios de comunicaciones a la comunidad académica y de investigación de Galicia. RECETGA también facilita el acceso a los servicios ofertados por los propios centros de investigación integrantes de RECETGA (servicios de almacenamiento, computación y GIS del propio CESGA, a información meteorológica de Meteogalicia, etc...).

1.2 Esquema de red RECETGA y CESGA

A continuación expondremos, a groso modo, las características de la red a monitorizar: La red regional de investigación (RECETGA) y la red interna del CESGA.

Un esquema de la red y del montaje de la infraestructura sería el siguiente, ya con los servidores de monitorización en su "ubicación" actual:



1 Para más información: http://www.cesga.es → Infraestructuras → Comunicaciones → RECETGA

La red RECETGA se compone de distintos *routers*/*switches* de acceso de centros ubicados en las sedes de los mismos y en puntos intermedios entre la red del centro destino y la red CESGA.

En la parte de red interna del CESGA nos encontramos con equipos de conmutación/*routing* que permiten el acceso a la red *HPC* (*High Performance Computing*) y los diversos equipos de conmutación/*routing* que dan acceso a los restantes servicios CESGA: granja de servidores de virtualización, *cloud*, *housing*, *hosting*, videoconferencia H323, *firewall* de acceso, etc...

1.3 El sistema de monitorización previo de RECETGA

RECETGA disponía desde hace años de un sistema de monitorización de red desarrollado a lo largo de varios años. Dicho sistema tenía como finalidad última la monitorización del estado de la red, de los distintos servicios ofertados y del estado de conectividad de los distintos centros que constituyen RECETGA. Siendo, además, un punto de información centralizada, donde poder generar informes de estado de la red, consultar estadísticas, SLAs, gráficas, alta/gestión de equipos de conmutación/*routing*, etc.

Este servidor de monitorización inicial disponía de unas características propias que hacían **difícil** su **adaptación al *status quo* presente y futuro de la red:** sus funcionalidades se obtenían en base a la implementación de muchas y variadas herramientas, programadas en distintos lenguajes de programación (Perl, Python, C++, PHP, s*hell scripts*) y en las cuales se habían llevado a cabo diversas modificaciones para acomodarlas al entorno de RECETGA en su planteamiento original.

Todo esto complicaba la actualización de cada una de estas herramientas a versiones más modernas y que soportasen **nuevas características**: *IPv6*, autodescubrimiento de servicios, monitorización distribuida, servicios web 2.0, monitorización de otros elementos (*TRAPs*, etc).

Ante esta problemática, surge la **necesidad de desarrollo de un nuevo servidor de monitorización** que tenga las funcionalidades del anterior, solvente sus problemas y dote al mismo de funcionalidades de acuerdo a las nuevas y futuras necesidades de la red.

1.4 Servicios monitorizados

El CESGA, tradicionalmente, ha visto la necesidad de monitorización de sus servicios con el fin de medir la calidad de los mismos, tanto en aquellos destinados al usuario final, como para evaluar el correcto funcionamiento de sus servicios internos.

Este enfoque se ha visto reforzado a raíz de la introducción de diversos cambios en la gestión/evaluación de los procedimientos de calidad, en los cuales se fijan indicadores que permiten evaluar el grado de nivel de servicio proporcionado por el CESGA a sus usuarios, y, en particular, el la concreción de los niveles de *SLA* para el servicio de conectividad de red que se le da a los mismos.

Por todos estos motivos, se monitorizan diversidad de elementos, clasificándose principalmente en los grupos que se indican a continuación: **monitorización de RECETGA** (afectan a las *SLA*), **monitorización de equipamiento de red interno** y **monitorización de servidores/servicios** (de usuario o internos)**.**

Los **servicios de monitorización de RECETGA**:

- Nodos de red troncal
  - Estado de conectividad del nodo: *ping* y latencia
  - Estado de conectividad de los distintos enlaces que dan servicio al nodo
  - % Disponibilidad (***SLA***[1]) del nodo
- Enlaces de red troncal
  - Estado de conectividad del enlace: caído o levantado
  - Tráfico de entrada salida del enlace.
  - % Disponibilidad (***SLA***) del enlace
- Centros conectados a RECETGA
  - Estado de conectividad del centro: *ping* y latencia
  - Estado de conectividad de los distintos *routers*/*switches* y enlaces entre *routers* que conforman la conectividad del centro
  - Tráfico de entrada salida del centro y de los distintos enlaces.
  - % Disponibilidad (***SLA***) de estos servicios

Los **servicios de monitorización de equipamiento de red interno**:

- Estadísticas *SNMP* de tráfico de entrada/salida todos los interfaces de red de los *routers*
- Recepción de *TRAPs* de los equipos de red
- Salud del equipamiento: si está caído o levantado, estado de fuentes, temperatura, memoria,CPU,...

Los **servicios de monitorización de servidores/servicios**:

- Estado general de la máquina (disco, procesos, usuarios conectados, etc.) y de sus servicios (*web*, bases de datos, *email*, CIFS, NFS, etc.)

1 SLA en Wikipedia

2 Estado del arte de sistemas de monitorización

Para el desarrollo e implementación de la nueva herramienta de *NOC*, se procedió a un estudio previo del arte en el ámbito de las soluciones de monitorización de red existentes para encontrar aquella solución que, teniendo licencia *GPL* (requisito obligatorio para su posterior adaptación), nos permitiese cubrir la más amplia gama de funcionalidades existentes en el "antiguo portal de *NOC*" y además nos facilitase la incorporación de otras nuevas.

Durante la elección de la herramienta, se procedió a la evaluación de las siguientes soluciones de monitorización: Nagios, OpenNMS, Pandora FMS, Cacti, Ganglia, Zenoss y Zabbix, resultando elegida como solución más óptima, esta última, **Zabbix**.

La solución implantada consta de un servidor y agentes programados en C++ (instalables en casi cualquiera S.O.), un interfaz web PHP y una base de datos MySQL.

3 Implementación del sistema de monitorización

A raíz del análisis de requisitos preliminar, así como de las diferentes funcionalidades que se han ido añadiendo por las diferentes migraciones acaecidas en la red, de las necesidades de obtención de unos nuevos indicadores de medida y de la mejora de procesos para la detección y resolución de incidencias más eficiente, se han implementado una serie de funcionalidades que indicamos a continuación:

3.1 Funcionalidades básicas

Dado que el nuevo servidor de monitorización reemplazaba a un servicio ya existente, era de esperar que éste contase, al menos, con las funcionalidades del original que fuesen consideradas de interés:

1. Monitorización básica de equipos *hardware*, enlaces (conexiones de red entre puntos), nodos de red troncal y centros conectados a RECETGA.
2. Incorporación de monitorización avanzada de medidas de latencia y obtención de datos de tráfico para equipos, enlaces, nodos y centros mediante ***SNMP***[1].
3. Envío de alertas por correo-e y mensajes *SMS* a los técnicos de soporte tanto del CESGA como de los centros conectados.
4. Incorporación de paradas programadas ante posibles migraciones, actualizaciones, mantenimientos preventivos de los servicios, sin afectación a su *SLA (opcional)*.

1 SNMP en Wikipedia

5. Integración con el sistema de gestión de incidencias del CESGA (*Request Tracker*) configurable para permitir la categorización y asignación automática de las incidencias
6. Generación y consulta de gráficas, estadísticas e informes.

3.2 Funcionalidades mejoradas

3.2.1 Propias de Zabbix

Adicionalmente se incorporaron funcionalidades propias de la herramienta Zabbix que se determinaron como interesantes para el CESGA.

1. Monitorización de servicios:
   - Análisis y reemplazo del software *Nagios*, utilizado para la monitorización de la plataforma de *hosting* del departamento, basada en servidores de máquinas virtuales Xen/KVM.
   - Incorporación a la monitorización de los servidores de *housing* interno/externo así como los servicios que estos servidores puedan tener.
2. Des/habilitar el envío de alertas para incidencias/elementos concretos mediante interfaz *web*.
3. Gestión de alertas basadas en los *TRAPs* recibidos de los equipos *hardware*.
4. Escalado de incidencias en caso de no respuesta del técnico de soporte pasado un determinado tiempo.
5. Gestión de usuarios:
   - Gestión de usuarios y permisos de acceso (lectura/escritura) en los distintos elementos monitorizados, tanto para técnicos CESGA como para los técnicos de los centros conectados a Recetga.
6. Personalización independiente del tiempo de monitorización de cualquier *item*. Anteriormente, el período de monitorización era de 5 minutos. Actualmente, la monitorización se lleva a cabo cada 2 minutos en general.
7. Posibilidad de monitorización de cualquier métrica o elemento de la red.

3.2.2 Desarrolladas por CESGA

Finalmente, con el fin de optimizar los procesos de detección de incidencias, de ofrecer mayor visibilidad al usuario final de los servicios prestados y de mejorar la productividad del *NOC* se añadieron otras funcionalidades, indicadas a continuación:

1. Optimización para una incorporación sencilla y rápida de nuevos equipos que incluye autodescubrimiento de interfaces de red en *routers*, generación automática de pantallas de visualización, auto-actualización diaria de todos los elementos monitorizados de los *routers* (interfaces, *VLANs*, etc...)
2. Generación automática de informes de indicadores de calidad del CESGA:
   - Incidencias de red mayores de 24 horas,
   - *SLA* de los centros conectados, tráfico transferido (medias, máximos de tráfico de entrada/salida por mes/año/período concreto).
   - Generación de los informes en distintos formatos (html, odt, doc y pdf) y archivado automático de los mismos en gestor documental Alfresco.
3. Otras:
   - Integración del *NOC* dentro del nuevo portal web del CESGA gracias a la potente API de *Zabbix*.
   - Auditoría/registro de actividades de usuario del *NOC*
   - Importación/exportación de datos en *XML*
   - Generación de mapas de red mediante la integración de la herramienta "*weathermap*".

3.3 Funcionalidades de Zabbix como herramienta de soporte al NOC

En esta sección explicaremos en detalle, todas las características del servidor de monitorización Zabbix instalado en el CESGA así como sus principales funciones y características.

3.3.1 Funcionamiento básico

El funcionamiento básico de Zabbix es el siguiente:

- El servidor monitoriza por *SNMP* todo *router/switch* dado de alta, y realiza comprobaciones de *ping* y latencia.
- El servidor también procesa todos los "*TRAPs*" generados por los mismos.
- Los agentes monitorizan las máquinas físicas o virtuales y sus servicios y envían los datos al servidor de monitorización que se encarga de evaluarlos.
- Los distintos servicios tiene configurados "monitores" (o *triggers*) que evalúan los datos recogidos (*items* o monitores). Si se cumplen determinados parámetros éstos ejecutan (o no) acciones:
  - Enviar *e-mails* de aviso

- Enviar *SMS* de aviso
- Escalar las acciones en caso de que el técnico de soporte de primer nivel no deje constancia de recibir el mensaje de aviso.
- Otras

- El servidor *web* genera todas las pantallas de visualización/informes/etc. automáticamente en base a tareas programadas o bajo demanda.

3.3.2 Funcionamiento avanzado

En esta sección explicaremos el funcionamiento más avanzado de Zabbix como *NOC*; pero desde un punto de vista más técnico.

3.3.2.1 Gestión SNMP

SNMP (*Simple Network Management Protocol*) es un protocolo estándar para la gestión de dispositivos de red. Algunos de los dispositivos que son compatibles con *SNMP* son *routers*, *switches*, servidores, etc..

Cada fabricante *hardware* implementa aquellas variables *SNMP* que considera de interés en su *hardware*. La información que se puede obtener está definida en un archivo denominado *MIB*, que describe la estructura de los datos *SNMP* de un determinado dispositivo. Cada dato tiene una representación numérica u *OID* en el *MIB* del dispositivo en concreto.

Para monitorizar los *OIDs* en Zabbix, hay que darlos de alta manualmente uno a uno, según la información que interese obtener.

El trabajo de alta de elementos SNMP a monitorizar puede ser tedioso, según la cantidad de elementos que deseemos monitorizar.

Para solventar estos problemas, se desarrollaron una serie de *scripts* que exploran el *hardware* a monitorizar y que crean automáticamente el dispositivo y sus elementos en Zabbix, mediante la importación de un archivo XML con su definición.

Todos los *routers* y *switches* desplegados en el CESGA están monitorizados por *SNMP*, en concreto, se monitorizan los siguientes parámetros:

- Nombre y descripción de todos los interfaces de red
- Tráfico de entrada/salida de los interfaces de red
- Estado del puerto
- Direccionamiento IP asignado (si tiene)
- Tiempo desde el último reinicio.

Posteriormente, una tarea programada 2 veces al día, ejecuta los *scripts* de exploración del *hardware* ya monitorizado, creando un nuevo XML con su definición y, si éste difiere del anterior, actualiza el equipo monitorizado añadiendo los nuevos elementos disponibles. Esta tarea también se puede ejecutar bajo demanda desde el interfaz de Zabbix.

De esta manera, todos los interfaces de red de los *routers* y *switches* están siempre monitorizados, independientemente de cambios en los mismos.

Con estos elementos de monitorización incorporados en Zabbix, posteriormente se definen los enlaces de red y los centros. El tráfico de cada uno de estos elementos será el tráfico del interfaz de red que corresponda.

Datos numéricos:

- **Total elementos monitorizados por Zabbix: 16552**
- **Del total, elementos *SNMP*: 13757**
- **Del total de elementos *SNMP*, cada 2 minutos: 9186**

Con estos datos en mano, el **rendimiento mínimo que necesita el servidor**, es decir, el **número de elementos a monitorizar por segundo** asciende a un total de **76,55**.

3.3.2.2 Alta disponibilidad (HA)

La infraestructura de monitorización se compone de dos máquinas virtuales Zabbix (primario, secundario) que monitorizan constantemente todos los servicios. Cada una de estas máquinas dispone de los servicios necesarios para poder funcionar de forma autónoma. Es decir, cada una cuenta con su propia base de datos, su servidor web, etc.

Se puede considerar una infraestructura en **alta disponibilidad activo-activo**: los dos monitorizan al mismo tiempo los mismos elementos. Ambos Zabbix tienen la misma copia de la base de datos de elementos a monitorizar, eventos y acciones.

Problemas y su solución

- **Duplicidad de alertas recibidas**: como ambos Zabbix tienen configuradas las mismas acciones, ambos podrían enviar las alertas, por lo que se recibirían alertas duplicadas. Para solucionar esta problemática, se configuró lo siguiente:
  - El servidor primario tiene las acciones habilitadas para el envío

- El servidor secundario tiene las acciones deshabilitadas para evitar el envío
- Entre ellos se configuró un *heart-beat* que monitoriza el estado de cada uno de ellos, para activar o desactivar las acciones en uno u otro según corresponda.

- **2 Servidores y bases de datos independientes**: la alta de nuevos elementos a monitorizar solo se produce en el servidor primario, es decir, solo se usa el interfaz *web* del servidor primario, que a su vez inserta los datos en su propia base de datos. Para solucionarlo, cada vez que se insertas nuevos equipos *hardware* a monitorizar (*routers*, *switches*, servidores, etc...) procedemos a realizar una sincronización de los archivos de la base de datos del servidor primario al secundario. Esto implica una parada de los servicios de monitorización de la máquina secundaria durante el tiempo de la copia, pero mientras el servidor primario sigue funcionando correctamente.

  El tiempo medio de esta parada es, actualmente, de unos pocos minutos para el copiado de los datos de un año (10GB aprox.). Posteriormente, se arranca el servidor de MySQL del servidor secundario que hace un chequeo de integridad de los datos y una vez finalizado, se arrancan de nuevo los servicios.

  Esto sólo es factible de realizar, si el motor de la base de datos es MyISAM, ya que este motor no es transaccional, al contrario que InnoDB. Una base de datos InnoDB no se puede copiar con la base de datos en ejecución, porque la mayoría de las veces, la comprobación de integridad de InnoDB fallará provocando que no se pueda acceder a datos y/o tablas de la base de datos.

  Otra posible solución: configurar Zabbix en HA pero con el servidor de MySQL en modo Máster-Máster.

3.3.2.3 Generación de informes

En el CESGA disponemos de indicadores de calidad que miden el cumplimiento de nuestros SLAs. Estos indicadores miden, por ejemplo, la disponibilidad de los nodos de la red troncal y de los enlaces que la constituyen, así como la disponibilidad de la conexión de los centros conectados a RECETGA (categorizados según el número de usuarios del centro) y de cada uno de sus enlaces de acceso.

Por ello, es necesario la obtención de informes que nos permitan evaluar el grado de cumplimiento de los mismos.

Zabbix en su configuración por defecto también dispone de un módulo para la generación y visualización de informes pero que se limita básicamente a medir la disponibilidad de un determinado servicio. En el CESGA necesitábamos una herramienta más completa que nos permitiese visualizar rápidamente el estado de cumplimiento de nuestros objetivos e indicadores.

Con esto en mente, se elaboraron diversos desarrollos en PHP que permiten simular la agregación física y lógica de los distintos sub-servicios que componen el servicio principal: la conectividad a RECETGA de un centro.

Estos desarrollos permiten medir la disponibilidad de cada uno de los servicios básicos de forma independiente con el fin de agregarlos y calcular la disponibilidad del servicio final para su posterior visualización.

INDICADORES CALIDADE CESGA

RECETGA
Rede de Ciencia e Tecnoloxía de Galicia

Tráfico por Instituciones
NOC CESGA

Tráfico comprendido entre o 01-09-2012 as 00:00 e o 02-09-2012 as 23:59

Universidad de Vigo	Organización	MEDIA in (Mb/s)	MAX in (Mb/s)	Total in (MB)	MEDIA out (Mb/s)	MAX out (Mb/s)	Total out (MB)	Dispoñib.
UVIGOB	Universidades	54,808	127,050	1.183.432,256	32,212	114,097	695.548,044	100,000%
UVIGO-PO	Universidades	0,022	0,036	469,008	0,001	0,001	14,943	100,000%
UVIGOO	Universidades	0,021	0,031	463,305	0,001	0,026	16,300	100,000%
Totales				**1.184.364,570**			**695.579,286**	**100,000%**

Universidad de A Coruña	Organización	MEDIA in (Mb/s)	MAX in (Mb/s)	Total in (MB)	MEDIA out (Mb/s)	MAX out (Mb/s)	Total out (MB)	Dispoñib.
UDC	Universidades	48,844	100,970	1.054.670,529	37,668	99,282	813.349,704	100,000%
Totales				**1.054.670,529**			**813.349,704**	**100,000%**

Universidad de Santiago de Compostela	Organización	MEDIA in (Mb/s)	MAX in (Mb/s)	Total in (MB)	MEDIA out (Mb/s)	MAX out (Mb/s)	Total out (MB)	Dispoñib.
USC	Universidades	20,829	138,600	449.745,247	28,042	116,637	605.489,569	100,000%
Totales				**449.745,247**			**605.489,569**	**100,000%**

Creado o 9-10-2012 as 16:52

Ilustración 1: Informe de SLA

Además, se pueden exportar los informes en formato PDF mediante la librería DomPDF[1], integrada en Zabbix a tal efecto.

1 Página oficial de DomPDF

Adicionalmente, estos informes, se generan mensual y/o trimestralmente (o bajo demanda), son almacenados por el NOC y agregados a nuestro gestor documental (Alfresco), informando al coordinador del área de la disponibilidad de los mismos para su consulta.

3.3.2.4 Interfaz de programación de aplicaciones (API)

Definición de API de la Wikipedia[1]:

"*Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción"*

Una de las características más importantes en la decisión de la elección del software de monitorización, fue la existencia de una ***API* en Zabbix**[2] (desde la versión 1.8). La *API* permite la consulta, modificación, alta de casi cualquier elemento monitorizado (*hardware*, servicio, enlace de red, etc..) desde cualquier aplicación externa a Zabbix y programada en un lenguaje que sea compatible con **JSON**[3] (la mayoría).

De esta manera, se programaron distintas páginas PHP integradas en la web del CESGA que ofrecen la posibilidad de consultar gráficas de estado, tráfico de los centros, carga de los servidores, etc. a los clientes finales del CESGA.

En la página siguiente, se pueden ver dos capturas de pantalla de la visualización ofrecida a través de la web del CESGA.

1 API en la Wikipedia
2 Documentación de la Api de Zabbix
3 JSON en la Wikipedia

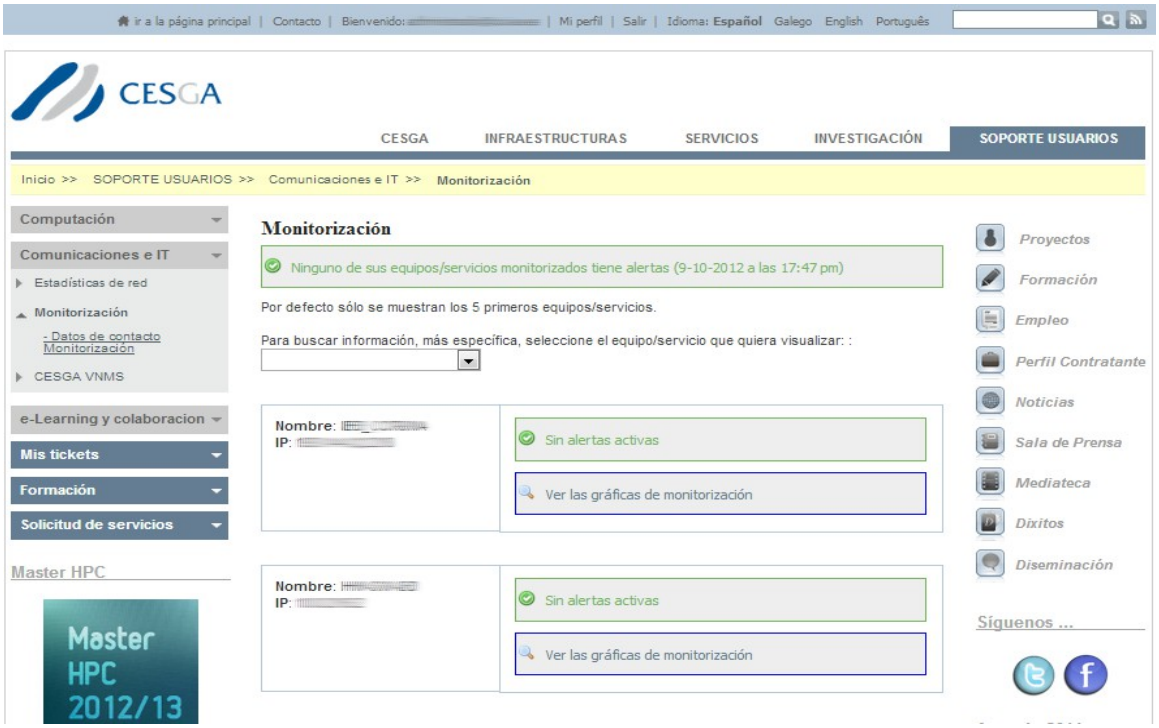


Ilustración 2: Monitorización Web CESGA - Estado

Ilustración 3: Monitorización Web CESGA - Gráficas de monitorización

3.3.2.5 Procesador de TRAPs SNMP

El **procesador de *TRAP* implantado en Zabbix**[1] es un servicio programado en Perl para el procesado de los *TRAP* originados por los diversos *routers* monitorizados. Un *TRAP* es un mensaje generado, normalmente, por un equipo de red (*router/switch*) ante una incidencia/evento detectada por el propio equipo, como puede ser, una nueva des/conexión física al mismo, una subida de temperatura por encima de los umbrales configurados, un cambio en la topología de la red, un cambio en la configuración, un intento de acceso fallido, etc. El *TRAP* es enviado al servidor de procesado de los mismos (en este caso Zabbix).

Un *TRAP* es generado con IP de origen la del *dispositivo que lo envía*. Zabbix realiza la correlación de esta IP origen con el equipo monitorizado por el mismo y añade la información a un *item* de monitorización.

En base al texto recibido en el *TRAP*, se pueden crear expresiones regulares que cumplan una o varias condiciones para ejecutar diversas acciones.

La configuración actual de Zabbix en el CESGA nos permite notificar aquellos *TRAPs* más relevantes (*linkup*, *linkdown*, cambios en las sesiones de BGP, etc.), así como registrar cualquier otro *TRAP* recibido del *hardware.* Cada tipo de equipamiento envía *TRAPs* en función de sus eventos particulares (cabinas de almacenamiento informan del estado de los discos, los firewall notifican intentos infructuosos de inicio de sesión de *VPN*, etc.), es necesario un análisis preliminar para identificar aquellos *TRAPs* que son de interés en cada despliegue.

El servicio se compone de un *script* principal que contiene la lógica del servicio y dos archivos de configuración:

1. Uno contiene expresiones regulares de *TRAPs* que no son interesantes para el administrador y por tanto se procede a su inmediata exclusión
2. El otro relaciona de forma directa una IP (origen del *TRAP*) con el nombre de un equipo/servicio en la instalación de Zabbix. Este equipo será el que reciba la información del *TRAP* como propia.

El *script* principal recibe el *TRAP* y si cumple las condiciones, lo inserta a Zabbix usando el ejecutable “zabbix_sender” que provee Zabbix.

Posteriormente, si el mensaje del *TRAP* coincide con alguna de las expresiones regulares monitorizadas, se ejecuta la acción correspondiente.

1 Documentación del Procesador de Traps

3.3.2.6 Integración con el sistema de gestión de incidencias: Request Tracker

Una de las características principales de un *NOC* es la gestión de las incidencias. Zabbix tiene un gestor básico de incidencias, donde un usuario puede tomar anotaciones sobre la misma y/o recibir respuestas de la incidencia. El CESGA ya disponía de una herramienta específica de *helpdesk* para la gestión de incidencias: *Request Tracker*[1] (o RT).

No vamos a entrar a explicar el funcionamiento del software RT. Simplemente indicar que permite al CESGA una gestión avanzada de las incidencias siendo, en este caso, la herramienta principal para la gestión de las mismas. Por eso, el Zabbix debía integrarse con el RT, al igual que lo hacía su predecesor.

Para ello, Zabbix nos proporciona una gestión muy eficaz de las alertas y las acciones que conlleva cada una, pudiendo asociar distintas acciones (crear incidencia en el gestor, enviar un aviso por SMS) a cualquier alarma que ocurra en la monitorización.

Como nuestros SLA se basan principalmente en la disponibilidad de conectividad de los centros de RECETGA, **la pérdida de conectividad de unos de estos centros creará un ticket de incidencia en el gestor**. Antes de esto, debe comprobar si, el centro para el cual se necesita crear una incidencia, no tiene otra incidencia abierta anterior. En caso de que exista una incidencia abierta, no se crea otra adicional.

Para llevar a cabo estas acciones, se usa un pequeño script que consulta la base de datos de Request Tracker, verificando la existencia o no de una incidencia de red anteriormente abierta.

Anteriormente, se usaba una pequeña modificación del código fuente del servidor de Zabbix que fue publicada en los foros de la comunidad de Zabbix, en el siguiente enlace: Parche para integrar Zabbix con Request Tracker

Posteriormente, el mencionado parche quedó en desuso para favorecer que las actualizaciónes de las versiones del servidor de Zabbix fueran más “directas”.

3.3.2.7 CESGA VNMS – Cliente para móviles

Una vez puesto en producción el nuevo servidor de monitorización y basándonos en las características de la *API* del Zabbix, se decidió crear una versión móvil del portal de

1 Página oficial de Request Tracker

acceso al mismo que ayudase/permitiese a los técnicos de guardia, conectarse con su *smartphone* para comprobar el estado de la red y servicios.

Para ello se desarrolló una aplicación web en PHP, CESGA *VNMS*[1], basada en *Jquery Mobile* y optimizada para su visualización en pantallas pequeñas que permitiese una rápida consulta de todos los parámetros de los elementos monitorizados.

Así, se puede consultar, por ejemplo, las incidencias abiertas, el tráfico de un centro conectado a RECETGA, la información de los técnicos de guardia del centro, etc..

Ilustración 4: Estado actual de las incidencias - CESGA VNMS

Ilustración 5: Datos de un equipo/centro - CESGA VNMS

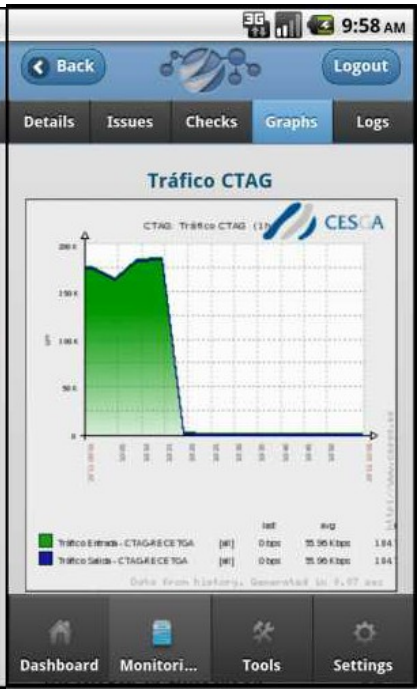


Ilustración 7: Gráfico de un equipo/centro - CESGA VNMS

Puede ver las características (y el código fuente) completas de la herramienta en la página del proyecto: https://github.com/CESGA/CESGA-VNMS-ZABBIX/

1 CESGA VNMS publicado en GITHUB

3.4 Requerimientos hardware/software

La máquina de monitorización de Zabbix es una máquina virtual que se ejecuta en nuestra granja de servidores Xen.

La máquina virtual tiene asignados los siguientes recursos:

- 1,5 GB de RAM
- 1 VCPU
- Almacenamiento local del servidor Xen
  - 4GB de S.O.
  - 30GB de almacenamiento de la base de datos.

En la propia máquina virtual reside el servidor de bases de datos (MySQL) de la aplicación, el servidor web (Lighttpd) del frontend de Zabbix, así como el propio servidor de Zabbix, encargado principalmente de ejecutar la monitorización por *SNMP* de todos los elementos de la red, procesar los datos recibidos de los agentes de monitorización instalados en los servidores y procesar todos los *traps* recibidos de los distintos *routers*/*switches*.

Consideraciones sobre el rendimiento

Uno de los problemas principales que nos encontramos durante la puesta en producción de la máquina de monitorización fue el rendimiento[1] en el acceso a disco por el uso intensivo que el servidor Zabbix hace de la base de datos (inserción de registros). A diferencia de muchos sistemas de monitorización que usan RRDs para guardar los datos de monitorización, Zabbix usa una base de datos (PostgreSQL o MySQL) para guardar todos los registros/datos.

Estado de ZABBIX

Parámetro	Estado	Detalles
Servidor ZABBIX operativo	Si	-
Número de equipos/plantillas (monitorizados/no monitorizados/plantillas/eliminados)	269	220 / 20 / 29
Número de monitores (monitorizados/inactivos/no soportados)[trapper]	16165	16036 / 46 / 83
Número de iniciadores (activos/inactivos)[problemas/desconocidos/correctos]	1352	1223 / 129 [10 / 178 / 1035]
Número de usuarios(en línea)	30	3
Rendimiento requerido del servidor, nuevos datos por segundo	93.0568	-

Ilustración 8: Rendimiento: Estado Zabbix

1 Foro de Zabbix sobre rendimiento y optimización de MySQL

En la imagen, vemos una captura de pantalla propia de Zabbix que refleja la cantidad de elementos monitorizados por nuestro Zabbix, así como un valor, denominado "**Rendimiento requerido del servidor, nuevos datos por segundo**", que indica la cantidad de elementos que monitoriza de media por segundo. Este valor se aproxima a la cantidad de consultas SQL de inserción de datos que tiene realizar el servidor de bases de datos.

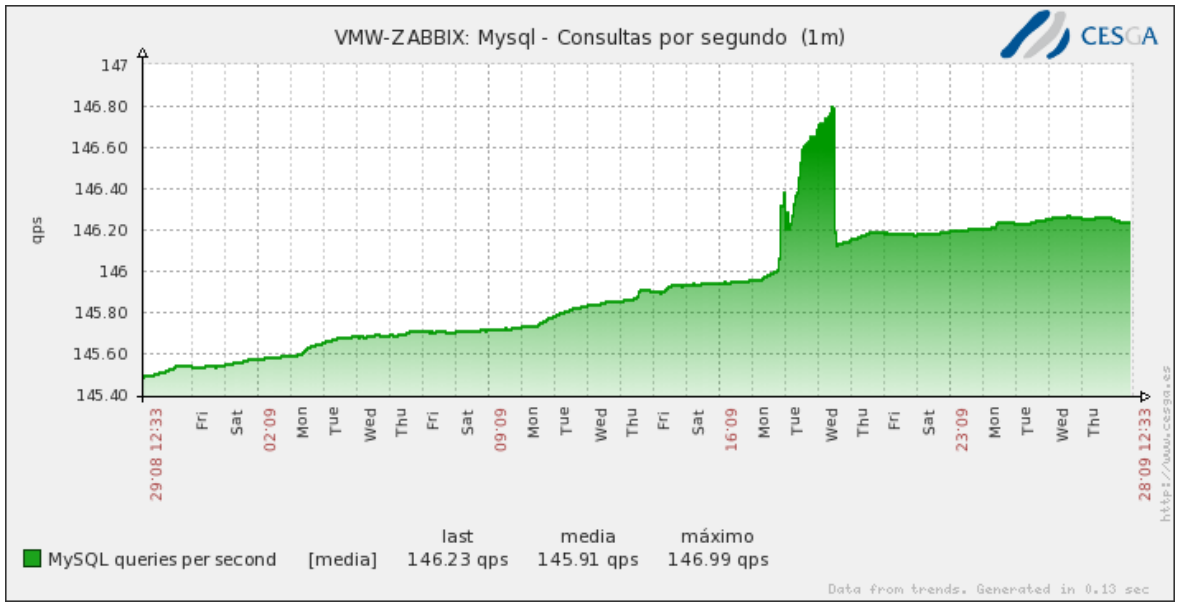


Ilustración 9: Rendimiento: Consultas por segundo del servidor MySQL (período 1 mes)

En la anterior imagen podemos ver la cantidad de consultas MySQL ejecutadas de media por segundo. El número es mayor al mencionado anteriormente ya que incluye las consultas SQL de lectura que realiza tanto el *frontend* web como el propio servidor Zabbix.

Otra medida que nos ofrece el interfaz de Zabbix sobre su propio rendimiento de monitorización, es **la cola (*Zabbix queue*[1]) de elementos (*items*) pendientes de monitorizar**.

1 Zabbix Queue - Documentación

COLA DE MONITORES PARA SER ACTUALIZADOS — Vistazo general

Monitores	5 segundos	10 segundos	30 segundos	1 minuto	5 minutos	More than 10 minutes
Agente ZABBIX	2	0	0	0	0	0
Agente ZABBIX (activo)	0	0	0	0	0	0
Agente SNMPv1	0	0	0	0	0	0
Agente SNMPv2	17	0	0	0	0	0
Agente SNMPv3	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
Comprobación sencilla	7	0	0	0	0	0
ZABBIX interno	0	0	0	0	0	0
ZABBIX añadido	0	0	0	0	0	0
Comprobación externa	0	0	0	0	0	0
Calculado	0	0	0	0	0	0

Ilustración 10: Rendimiento: Zabbix Queue

La cola de Zabbix nos indica cuantos items están pendientes de monitorizar y el retraso acumulado en la obtención de estos elementos. **Si estos elementos se acumulan y retrasan en la cola de Zabbix nos sirven como indicación de que el rendimiento del servidor no es suficiente** para la monitorización que estamos llevando a cabo.

Además, los *items* aparecen categorizados según su tipo, facilitándonos la tarea de identificar el cuello de botella en la monitorización, pudiendo así, cambiar algunos parámetros de configuración de la monitorización:

- *Items* SNMP
- *Items* sencillos: *pings*, latencias
- *Items* de agentes de Zabbix.
- *Items* calculados en base a valores de otros *items*
- etc.

MySQL

En una primera aproximación, configuramos el servidor de base de datos para que hiciese uso del motor transaccional InnoDB que ofrece un rendimiento en escritura mucho mejor que el motor por defecto de MySQL, MyISAM.

Con InnoDB, el rendimiento del servidor era mayor a la hora de realizar el trabajo de monitorización pero perjudicaba el rendimiento en general del servidor Xen donde se ejecuta la máquina, haciendo que las otras máquinas ejecutadas en el mismo servidor no

respondiesen correctamente e incluso haciendo que el propio servidor Xen quedase inestable o, en el peor de los casos, se “colgase”.

Esto último nos generaba un mayor problema, ya que cada vez que pasaba esto último, al reiniciar la máquina, el motor transaccional tenía que reconstruir todos los índices de las tablas, tarea que a veces se demoraba cierto tiempo y que a veces no lograba realizar correctamente, obligándonos a tener que recurrir a restauración de *backups* de la BB.DD que podían llegar a demorarse varias horas (debido al tamaño de la misma), impidiendo la monitorización de los servicios durante todo ese tiempo.

Para subsanar estos problemas, se procedió a realizar las siguientes acciones:

1. Mover la máquina virtual a un servidor Xen con un uso menos intensivo y ajustar distintos parámetros de la virtualización por defecto de Xen.
2. Clonar la máquina de monitorización a un servidor de monitorización secundario ejecutado en otro servidor Xen distinto.
3. Cambiar el motor transaccional por MyISAM y aumentar los recursos hardware asignados a la máquina, en concreto pasamos de 1GB de RAM a 1,5GB.

4 Conclusiones

La solución aportada en este documento no tiene, ni mucho menos, que ser la única, pero en el caso particular que tratamos (monitorización de RECETGA), resultó ser la que mejor se amoldaba a nuestras necesidades con la menor personalización posible.

Actualmente, la solución lleva un año en producción durante el cual se han observado las siguientes ventajas e inconvenientes:

Empezaremos por las ventajas, ya que son bastante más importantes que los problemas encontrados:

1. **Alta disponibilidad en el servicio de monitorización**: dos servidores independientes en HA (*High Availability*) monitorizando al mismo tiempo los mismos servicios.
2. **Recuperación casi inmediata ante caídas de la máquina**: el motor MyISAM no es transaccional lo que implica que en caso de desastre (caída de la máquina), la única tarea que habría que realizar es reparar las tablas afectadas, un proceso "rápido" (pocos minutos) en el cual MyISAM descarta aquellas filas de las tablas que no puede recuperar. Sin embargo, InnoDB al ser un motor transaccional tiene que asegurar la integridad de todas las filas de las tablas por lo que si no puede recuperar algunas claves foráneas u otros datos deja la base de datos en un modo inconsistente que hace imposible su uso, obligándonos a recuperar un *backup* de la misma.
3. **Parametrización mediante interfaz *web*** de todos los elementos monitorizados: el interfaz *web* de gestión nos permite realizar todas las tareas necesarias.
4. **API:** el tener una API facilita enormemente el desarrollo de personalizaciones escritas en cualquier lenguaje de programación así como la integración de la herramienta de monitorización en otras aplicaciones, dotando a la herramienta de monitorización de un "*status*" 2.0 que otras herramientas no poseen.

Los inconvenientes asumidos de esta solución:

1. **Configuración de un *heart-beat* entre el servidor primario y secundario:** este *heart-beat* (o chequeo), comprueba desde el servidor secundario que tiene acceso al servidor primario. Si este no se encuentra operativo, el

secundario comienza a procesar las acciones de las alertas definidas, es decir, a notificar a los usuarios/técnicos de las distintas alertas y viceversa

2. **Mantenimiento y replicación de la base de datos** del servidor primario al servidor secundario: cada cierto tiempo se procede a un copiado (*rsync*) de los archivos de la base de datos del servidor primario al secundario, obligando a parar el secundario durante el tiempo de la copia.