

# Check\_mk. Funcionalidades. La navaja Suiza para Nagios.

Check\_mk es un maravilloso paquete de software que envuelve Nagios. Digo que envuelve porque incluye muchas utilidades para complementarlo y potenciarlo. A continuación vamos a tratar de hacer un resumen de funcionalidades de check\_mk.

En la página oficial ([http://mathias-kettner.com/check\\_mk.html](http://mathias-kettner.com/check_mk.html)) explican muy bien cada uno de los componentes por lo que haremos un somero repaso:

## **Interfaz web Multisite**

Check\_mk añade un interfaz web para Nagios Core. No es como otros que se limitan a cambiar las hojas de estilo y añadir alguna que otra cosilla para que resulte más vistoso. No.

El interfaz web de check\_mk es multisitio, podemos controlar varias instalaciones de Nagios desde él. Además incorpora muchas funcionalidades y filtros muy potentes que ni por asomo están presentes en Nagios Core. Barra de navegación, vistas personalizables, agrupaciones, filtros,...

¿Cómo consigue cambiar completamente el interfaz de Nagios? Reescribiéndolo desde cero en un lenguaje más asequible para las modificaciones e integración de complementos (Python).

## **MK Live Status.**

Hasta la aparición de Live Status la forma habitual de acceder al estado de los objetos de Nagios era o bien a través del fichero status.dat, o bien a través de una base de datos que recibía periódicamente dichos datos mediante un módulo intermedio. Ambas opciones tenían bastantes pegas y hacían engorroso el tema de acceso a los datos de estado. Live Status cambia la aproximación al problema. Crea un socket a través del cual se pueden obtener los datos en tiempo real bajo demanda. Rápido, sencillo, funcional, no consume apenas recursos y se pueden realizar las consultas con un lenguaje propio próximo al SQL. ¿Alguien da más? No. De hecho otras herramientas que hacen uso de acceso a datos como Nagvis, NagiosBP y claro, check\_mk\_multisite, rápidamente pasaron a usar Live Status como el método preferido de acceso a estos.

## **Configuración y motor de chequeos.**

Check\_mk tiene su propia sintaxis de creación de objetos de Nagios. Nos proporciona toda una nueva forma de configurar los objetos, más funcional y más potente que la estándar de Nagios. El truco está en que la definición de objetos se hace con la sintaxis de mk pero luego se “vuelca” a ficheros con el formato estándar de Nagios. Claro, el condicionante es que usemos también el cliente que nos proporciona para los equipos a chequear (si, también un cliente para equipos Unix, Linux y Windows). No es que sea obligatorio, es que pierde sentido si no lo usamos. Dicho cliente se instala en los servidores a chequear y desde el servidor de “Nagios” con las herramientas de check\_mk podemos detectarlo e inventariar de forma automática una serie de chequeos. Parece fácil

¿no?. Lo es. Solo hay que ir introduciéndose en el formato mk de definición de objetos y controlar también el de Nagios ya que lo normal es que nos interese seguir haciendo ciertas configuraciones con el formato de este.

Por cierto, los objetos que configuremos con check\_mk recibirán automáticamente la configuración y visualización de gráficas con PNP4Nagios. Impresionante. Nos ahorramos la configuración de este. Solo tiene que estar correctamente instalado...

## **WATO. Web Administrator tool.**

¿Además poder configurar los objetos a monitorizar de forma gráfica? No... lo siento... Y si... Realmente WATO no es para configurar todo un entorno de objetos de monitorización pero si nos deja realizar pequeñas modificaciones sobre objetos. Suficiente para que alguien que no sea experto en configurar las tripas de los bichos pueda realizar pequeños cambios del día a día a determinados objetos. Suficiente para quitarte cierta carga de trabajo en determinados proyectos.

Para el que se lo pregunte, otras herramientas de configuración visual de objetos de Nagios ([NagiosQL](#), [NConf](#),...) no son compatibles con check\_mk. Si usamos mk no podemos usar estas. En cualquier caso la sintaxis de mk me parece bastante potente como para no echarlas de menos.

## **Business Intelligence.**

Este módulo es para ir un paso más allá en la labor de monitorización. Nos permite agregar chequeos y relacionarlos entre con el objeto de ir más allá de monitorizar servicios sueltos y conseguir monitorizar la lógica de negocio. Es ir un paso más allá. Podemos poner un ejemplo sencillo para entenderlo. Imaginemos que tenemos una APLICACIÓN en clúster formado por varios FrontEnd y varios Backend de BBDD. Bien, con BI podemos monitorizar el funcionamiento de la APLICACIÓN con si fuera un servicio (independientemente de los host y servicios que lo componen). Relacionamos cada uno de los componentes que la forman de tal forma que nos indique si la Aplicación está funcionando en estado óptimo. El hecho de que se caiga un equipo o servicio no significa que nuestra Aplicación no funcione, para eso tenemos los equipos en clúster, para prever esa incidencia. Probablemente querremos una alerta de Warning pero no más. En resumen, monitorizamos el estado de nuestra lógica de negocio mediante la creación de reglas que definen cómo le afectan los servicios y cómo se relacionan, un paso más allá que monitorizar solo equipos y servicios aislados. La idea original realmente es de un addon existente previo de otro desarrollador que también veremos próximamente: [Nagios BP](#).

## **Event Console.**

La monitorización que realizamos con Nagios está basada en Estados (cambios de estado). En ocasiones sería interesante tener disponible algún tipo de monitorización basada en Eventos. Ese es el hueco que viene a cubrir Event Console. Es una aplicación que se integra con Multisite GUI y nos permite entre otras cosas: recibir mensajes de syslog, demonio snmp y otras aplicaciones..., procesarlas mediante reglas predefinidas, clasificarlas, correlacionar eventos, reescribirlas, ejecutar acciones y crear nuevos eventos que muestren un estado en la consola,...

En fin, otro juguete muy interesante que habrá que revisar cuidadosamente ...

## **Artículos relacionados**

- [Check\\_mk. Chequeos. Primeros pasos](#). Instalación de clientes y configuración mínima para el chequeo de estos.
- [Check\\_mk. Chequeos. Primeros pasos \(II\)](#). Configuración de chequeos adicionales básicos.
- [Monitorizar ficheros log con check\\_mk en Linux – Oracle alertlog y otros](#).
- [Instalación de packages de check\\_mk](#). Oracle RMAN package.
- [Monitorizar Oracle con Nagios y check\\_mk](#).
- [Iconos en Nagios \(y check\\_mk\)](#).