UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA

"ANÁLISIS DE NAGIOS CORE COMO HERRAMIENTA PARA EL MONITOREO DE REDES DE DATOS"

TESINA

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA:

MISAEL ALPIZAR SANTANA

ASESOR:

DR. MARCELO ROMERO HUERTAS

TOLUCA, MÉXICO; AGOSTO 2017

Universidad Autónoma del Estado de México
Facultad de Ingeniería

DEPTO. DE EVALUACIÓN PROFESIONAL
No. Oficio: 33/2017
Ciudad Universitaria, Toluca, Méx. a 14 de julio del 2017

C. MISAEL ALPIZAR SANTANA
PASANTE DE INGENIERÍA EN COMPUTACIÓN
P R E S E N T E

En respuesta a su solicitud, a continuación transcribo el tema aprobado por esta Dirección, que propuso el ***DR. MARCELO ROMERO HUERTAS,*** con el fin de que lo desarrolle en la modalidad de ***TESINA,*** le informo que se autoriza la **impresión de su trabajo** para presentar su Evaluación Profesional.

"ANÁLISIS DE NAGIOS CORE COMO HERRAMIENTA PARA EL MONITOREO DE REDES DE DATOS".

	RESUMEN
	INTRODUCCIÓN
CAPÍTULO I.	*MONITOREO DE REDES DE DATOS*
CAPÍTULO II.	*NAGIOS CORE*
CAPÍTULO III.	*ANÁLISIS Y DISCUSIÓN*
	CONCLUSIONES
	REFERENCIAS

Ruego a usted tomar nota de que en cumplimiento a lo especificado por la Ley de Profesiones, deberá prestar Servicio Social durante un tiempo mínimo de seis meses, como requisito indispensable para sustentar su Evaluación Profesional.

Asimismo, para la elaboración de la ***TESINA,*** y demás trámites, deberá sujetarse a la reglamentación respectiva de esta Universidad.

A T E N T A M E N T E
PATRIA, CIENCIA Y TRABAJO
"2017, Año Del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos".

ESTADOS UNIDOS MEXICANOS · UNIVERSIDAD AUTÓNOMA · PATRIA, CIENCIA Y TRABAJO · ESTADO DE MÉXICO
FACULTAD DE INGENIERÍA U.A.E.M.

DRA. MARÍA DOLORES DURÁN GARCÍA
DIRECTORA DE LA FACULTAD DE INGENIERÍA

**/Saha.

Cerro de Coatepec S/N
Ciudad Universitaria; Toluca México
Tel. (722) 2-14-08-55 / 2-15-12-51

Agradecimientos

A todos los que están, ayudaron y acompañan…

RESUMEN

Análisis de Nagios Core como herramienta para el monitoreo de redes de datos es un trabajo que muestra de manera general los puntos relevantes sobre el monitoreo en las redes de datos: qué es el monitoreo, en dónde se aplica el monitoreo, qué métricas son relevantes para el monitoreo de las redes de datos, qué es una herramienta para el monitoreo de redes de datos, qué herramientas existen para realizar el monitoreo en las redes de datos. Estos son algunos tópicos que se discuten en el primer capítulo de esta tesina.

El enfoque principal de este trabajo consiste en mostrar los puntos necesarios para poner en marcha Nagios Core en su versión 4.2.4, una herramienta de código abierto con la libertad para la personalización en varios niveles: dispositivos, servicios, alertas, resolución de problemas, entre otros; conocida por su flexibilidad y robustez al momento de monitorear ambientes de red con diversos dispositivos y servicios. El objetivo de instalar dicha herramienta es investigarla en un ambiente de red de área local e identificar las fortalezas y debilidades detectadas en la práctica.

Índice

Introducción

A menudo, en las redes se encuentran estructuras complejas con diversos elementos que se deben administrar, en los que cuando se presenta una falla resulta complicado encontrar la causa raíz, la operatividad de los sistemas resulta crítica para los clientes y se requiere soporte y monitoreo 24/7 (Hernantes, J. *et al*., 2015).

De acuerdo a Zanikolas, S., y Sakellariou, R., el monitoreo es el acto de recopilar información sobre las características y el estado de los recursos de interés (Zanikolas y Sakellariou, 2005); así que el monitoreo puede convertirse en la clave para lidiar con estos problemas. Por su parte, Fatema, K. *et al.* explica que el monitoreo es un proceso que identifica de forma completa y precisa la causa raíz de un evento, capturando la información necesaria en el momento y al menor costo para determinar el estado de un sistema y dar a conocer su estado de manera oportuna y significativa (Fatema, K. *et al* 2014).

El monitoreo en las redes de datos es ampliamente estudiado y aplicado. Por ejemplo, en el cómputo en la nube, para operar y gestionar adecuadamente infraestructuras complejas se necesita un monitoreo eficaz y eficiente (G. Aceto *et al*., 2013). Por su parte, Mansouri-Samani, M. y Sloman, M., (2002), considera que la supervisión, la recopilación dinámica, la interpretación y la presentación de información sobre objetos o procesos de software, es necesaria para la gestión de sistemas distribuidos o redes de comunicaciones. Sistemas donde la cantidad de equipos interconectados requieren una gestión dinámica de sus componentes hardware y una cuidadosa coordinación de las tareas que realizan, la cual no sería posible sin la recopilación de métricas que brinda el monitoreo en tiempo real. La evaluación en línea de esas métricas puede resultar en decisiones locales o globales para mejorar el comportamiento y rendimiento del sistema (Kornaros, G., y Pnevmatikatos, D., 2013). De acuerdo a Zanikolas, S., y Sakellariou, R., (2005), el monitoreo es esencial en una variedad de escenarios, por ejemplo: la programación, la replicación de datos, la contabilidad, el análisis de desempeño y la optimización de sistemas distribuidos o aplicaciones individuales, y aplicaciones de autoajuste.

El monitoreo aplicado a las redes de datos permite la identificación y resolución de problemas en cuanto a disponibilidad, capacidad (Fatema, K. *et al.*, 2014), latencia de comunicación, utilización de recursos para cada aplicación (Kornaros, G., y Pnevmatikatos, D., 2013), detección de fallas y cuellos de botella y en algunos casos su resolución automática (Zanikolas, S., y Sakellariou, R., 2005). La información que se colecta se utiliza también para tomar decisiones de gestión y realizar las acciones de control adecuadas en la red (Mansouri-Samani, M. y Sloman, M., 2002). Estas métricas son relevantes para que el administrador de red tome decisiones informadas sobre la utilización de los recursos (Fatema, K. *et al.*, 2014), para justificar la adquisición de nueva infraestructura o mejoras de infraestructura requeridas para eliminar cuellos de botella crónicos en la red (Lindros, K., y Tittel, E., 2015).

Estas mediciones pueden ser realizadas por las herramientas para monitoreo de redes que notifican y permiten diagnosticar y reparar errores al administrador de red. Con estos

datos, el administrador puede reconfigurar los componentes de red para un mejor servicio (Engel, F. et al., 2000). A pesar de que algunas herramientas cuentan con más funcionalidad que otras, de manera general ofrecen soluciones que permiten mantener la red a punto.

Las opciones que existen en cuanto a herramientas para monitorear la red son diversas, pasando por las comerciales y aquellas de código abierto. Las opciones de código abierto se vuelven atractivas debido a las ventajas que ofrecen. Clancy, H. (2010) explica que la flexibilidad que estas herramientas las ha convertido en una alternativa viable. El nivel de personalización que ofrecen, así como su capacidad de ser adaptadas tanto en redes como aplicaciones específicas son características determinantes que algunas herramientas comerciales no cumplen o implican gastos extras. Por último, explica también que los costos se han convertido en un factor muy importante, ya que los departamentos de TI buscan agilizar los presupuestos operativos.

Es importante que no se confundan los términos código abierto, software libre y software gratuito. De acuerdo con la *Free Software Foundation* software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el software libre es una cuestión de libertad, no de precio (Free Software Foundation, 2017). De hecho, se especifica que el código libre puede estar asociado a un precio, pero es independiente de las libertades de distribución, modificación, copia o mejora.

Por otro lado, se tiene el término código abierto propuesto por la *Open Source Initiative*. El software de código abierto es un software con código fuente que está disponible públicamente bajo una licencia que otorga a los usuarios el derecho de estudiar, cambiar y distribuir el software como deseen. En el código abierto se cambió el énfasis de la libertad a la seguridad, el ahorro de costos, la transparencia y otros beneficios pragmáticos. El término es más aceptable para el mundo corporativo (King, B., 2013).

Aunado a los términos anteriores existe un tercero que es importante discriminar, el de software gratuito. A diferencia del software libre y del software de código abierto, el software gratuito no hace referencia ni enfatiza la libertad de ninguna manera. En general, es un software que está disponible sin costo alguno. Dicho software sigue siendo generalmente de código cerrado o propietario, como Adobe Reader y Skype (King, B., 2013).

La cantidad de herramientas disponibles es abundante, sin embargo, en esta tesina se mostrará un análisis práctico de la herramienta de código abierto Nagios Core, la cual es reconocida por su rendimiento y flexibilidad (Fatema, K. *et al.*, 2014). Este análisis práctico se iniciará con la instalación y configuración de Nagios Core en una red de área local, donde se estudiará su funcionalidad para discutir las fortalezas y debilidades de dicha herramienta.

Los problemas en las redes de datos tales como: a) disponibilidad; b) capacidad (Fatema, K. *et al.*, 2014); c) latencia de comunicación; d) utilización de recursos para cada aplicación (Kornaros, G., y Pnevmatikatos, D., 2013); e) detección de fallas y f) cuellos de botella (Zanikolas, S., y Sakellariou, R., 2005) son problemas que a los administradores de red interesa resolver de una manera efectiva para mantener la red de datos funcionando de manera adecuada para satisfacer las necesidades de sus usuarios. Una red funcionando

adecuadamente produce un mayor índice de satisfacción, lo que a su vez reduce las llamadas a soporte técnico y sus subsecuentes consecuencias, aseguran Lindros, K., y Tittel, E. (2015).

Las herramientas para el monitoreo de redes de datos permiten una administración simple y automatizada, enviando alarmas o notificaciones cuando algún componente falla y emitiendo reportes sobre el comportamiento de los dispositivos, así como de las aplicaciones que se ejecutan (Hernantes, J. *et al.*, 2015).

Con esta tesina se pretende realizar un análisis de la herramienta para monitoreo de redes Nagios Core para identificar de manera objetiva las fortalezas y debilidades que dicha herramienta presenta. Para llegar a este punto será necesario realizar una minuciosa revisión de manuales de usuario, manuales técnicos y literatura relacionada al tema y finalizar con una validación experimental en un ambiente de red de área local para monitorear a través de Nagios Core diversos dispositivos de red de distintos fabricantes. Los hallazgos encontrados en esta experimentación se van a comparar con lo documentado en la literatura, con lo cual se estará en condiciones de discutir las fortalezas y debilidades de Nagios Core como herramienta para monitoreo.

A pesar de que existe una gran cantidad de herramientas de monitoreo tanto comerciales como de código abierto, de acuerdo a Fatema *et al.* (2014), Hyperic obtiene el puntaje mayor en cuanto a capacidades deseables en una herramienta para el monitoreo de redes de datos. Sin embargo, se considera que en la práctica Nagios Core presenta mayores ventajas debido a su flexibilidad y capacidad para ser personalizada, así como su desempeño en entornos en donde la cantidad de elementos a monitorear es considerablemente alta. Con Nagios Core se sacrifica la parte gráfica ya que su interfaz es pobre, pero esto representa una ventaja debido a que las otras herramientas para monitoreo de redes son poco flexibles, menciona Uretsky, M., (2013). Otro factor decisivo es la comunidad activa. Nagios tiene una activa comunidad global de soporte que desarrolla complementos adicionales estos complementos resuelven algunas de las limitaciones de la herramienta (Hernantes, J. *et al.*, 2015).

Este trabajo está dirigido a profesionales de TI que requieran información objetiva sobre herramientas para monitoreo de redes; que les permita contar con una guía sobre las ventajas y el uso de Nagios Core.

Objetivo general de la tesina

Analizar la herramienta para el monitoreo de redes de datos Nagios Core mediante su instalación, configuración y funcionalidad para discutir sus fortalezas y debilidades.

Organización del documento

Este trabajo está dividido en tres capítulos:

El Capítulo I. Monitoreo de redes de datos, documenta las generalidades sobre el monitoreo de redes de datos y herramientas disponibles.

El Capítulo II. Nagios Core, describe la instalación, configuración y funcionalidad de la herramienta para monitoreo de redes de datos de código abierto Nagios Core.

El Capítulo III. Análisis y discusión, presenta las fortalezas y debilidades detectadas basado en la literatura consultada y en un análisis práctico realizado con la herramienta para monitoreo de redes de datos de código abierto Nagios Core.

Finalmente, se presenta la Sección de Conclusiones, que puntualiza los hallazgos relevantes obtenidos en esta tesina y la lista de referencias consultadas.

Capítulo I. Monitoreo de redes de datos

En este capítulo se proporciona una breve descripción de conceptos básicos necesarios para comprender el contexto de las herramientas de monitoreo de red.

1.1 Generalidades

Las redes dentro de una organización juegan un papel fundamental y su objetivo se centra en mantener la comunicación entre las distintas estaciones de trabajo o la comunicación de los nodos dentro de la misma (Stallings, 2004). Con frecuencia los requerimientos de la empresa invitan a la escalabilidad de los componentes -aumento de sucursales, adquisición de equipo de cómputo, contratación de personal, adición de interfaces de red, etc.- y por lo tanto, la red necesita crecer también para brindar servicios a los nuevos dispositivos que se incorporan. Entre más componentes se integren a la red mayor será la complejidad de administración (G. Aceto *et al.*, 2013), es por esta situación que las herramientas para monitoreo de red se vuelven un recurso ineludible si se quiere tener un control adecuado de las métricas de la red, el nivel de servicio y la disponibilidad entre otras. Para Lindros y Tittel (2015) una herramienta de monitoreo de red es una combinación basada en software o software/hardware que observa la red de extremo a extremo, recopilando datos sobre amplia gama de métricas de rendimiento, entre ellas el ancho de banda, la latencia, la capacidad de respuesta y el uso de CPU de los *hosts*.

Generalmente las herramientas de monitoreo de red muestran la información y el estatus de los componentes de red en una interface con gráficos que permiten al administrador identificar rápidamente la salud de la red, también suelen enviar alertas y notificaciones cuando un incidente se presenta (Hernantes, J. *et al.*, 2015). Las herramientas para monitoreo juegan un papel sumamente importante ya que permiten que se hagan decisiones informadas sobre la utilización de los recursos (Fatema *et al.*, 2014). Si las herramientas para monitorear se implementan de manera adecuada pueden ayudar a justificar costo en el hardware o decidir si es necesario mejorar la infraestructura (Lindros, K., & Tittel, E., 2015) para agilizar problemas de congestión o cualquier otro tipo de situaciones identificadas por los administradores de la red, que degraden la calidad en los servicios que ofrecen a sus clientes o usuarios.

La elección de la herramienta adecuada para monitoreo de la red, puede ser una decisión complicada si no se conocen a fondo las necesidades de la red o no se tiene bien claro que utilidad se le quiere dar (Hernantes, J. *et al.*, 2015), lo que es claro es que los administradores quieren mantener una visión amplia y detallada de sus componentes o servicios de red en una interfaz intuitiva y amigable que emita una variedad de reportes y además se pueda personalizar. De acuerdo con Lindros y Tittel (2015) una herramienta para monitorear la red debe ser capaz de detectar, supervisar y analizar la red y sus dispositivos

en tiempo real, así como permitir al administrador responder con base en las advertencias y alertas. Como mínimo debe ser moderadamente fácil de implementar y configurar, debe soportar el monitoreo de dispositivos de múltiples proveedores, autodescubrimiento, es decir, que de manera automática reconozca los dispositivos que operan en la red; inventario de nodos y dispositivos, alertas automáticas de advertencias y problemas configurables y todo ello a través de una gestión centralizada basada en interfaz *web*. La interfaz debe incluir un tablero con gráficos fáciles de leer; la herramienta para monitoreo debe tener también la capacidad para generar un mapa de la topología de la red, así como comandos para modificar configuraciones de la red y solucionador de problemas. Por último, debe ser capaz de detectar y analizar el protocolo IPv4, así como IPv6.

Un factor importante que se debe tomar en cuenta es la inversión que se va a realizar en la adquisición de una herramienta para monitoreo, un rápido retorno de la inversión debe ser el objetivo menciona Hernantes *et al.* (2015). Las herramientas disponibles pueden encontrarse de manera gratuita (comúnmente de código abierto) o comerciales, la opción que se elija depende de los recursos disponibles, así como los requerimientos y funcionalidad deseada, una herramienta de código abierto puede descargarse e instalarse en cualquier momento pero puede estar limitada en cuanto a funcionalidad y usabilidad, es decir, su interfaz puede ser pobre y para obtener la funcionalidad deseada a veces se requiere de la instalación de parches o complementos (Lindros, K., & Tittel, E., 2015). Por otra parte, opciones comerciales a costos asequibles pueden ofrecer funcionalidad robusta e integración completa para monitoreo de la red, así como una interfaz amigable con gráficos fáciles de ser leídos para la emisión de reportes. En cuanto al soporte y mantenimiento las soluciones comerciales proveen paquetes de soporte, acceso bases de datos de conocimiento, llamadas telefónicas y documentación apropiada como manuales técnicos o de usuario en caso de que algún problema se presente. Por su parte, las herramientas de código abierto cuentan con una amplia cantidad de foros activos en la *web* y documentación de las herramientas en las páginas oficiales (Hernantes, J. *et al.*, 2015) disponible para que sea consultada por sus usuarios.

Hernantes *et al*. (2015) recomienda que cualquiera que sea su elección, debe ser tomada con fundamento en los requerimientos de funcionalidad que se alineen con las necesidades de su negocio y con las capacidades técnicas de su personal de TI, es decir que tan capacitados están y en que tecnologías.

1.2 Monitoreo de redes de datos y los modelos de administración de redes TMN e ISO (FCAPS)

El monitoreo de las redes de datos es un proceso implícito en los modelos de administración de redes *TMN* e *ISO*.

En el caso del modelo *TMN* el monitoreo para la administración de las capas: a) gestión de elementos (*EML*), b) gestión de redes (*NML*) y c) gestión de servicios (*SML*) es indispensable. En la capa de gestión de elementos el monitoreo permite la determinación de errores en el equipo, la medición de temperaturas del dispositivo, la recolección de datos estadísticos con fines contables (Rizos, C., 2013) y alarmas y eventos enviados desde ciertos elementos de la red (Claise, B., y Wolter, R., 2007). Hablando de la capa de gestión de redes,

el monitoreo ayuda en la creación de informes de utilización de red de extremo a extremo, análisis de causa raíz e ingeniería de tráfico (Claise, B., y Wolter, R., 2007). Y por último, en la capa de gestión de servicios se tiene el monitoreo de servicios, el manejo de mesa de ayuda y la facturación. Ejemplos incluyen gestión de *QoS* (retraso, pérdida, *jitter*), contabilidad por servicio y supervisión y notificación de *SLA*.

Por otro lado, el modelo de administración de redes *FCAPS* de *ISO* requiere del monitoreo en cada una de sus cinco capas para cumplir con sus objetivos. A continuación, se mencionan las cinco capas del modelo y de que manera se apoyan del monitoreo para realizar sus funciones.

En primer lugar, está la capa de Administración de fallos (*Fault management*), tiene responsabilidades como detectar, aislar, notificar y corregir fallos encontrados en la red (Claise, B., y Wolter, R., 2007). El monitoreo de la red apoya en todas las tareas que se realizan a este nivel sobre todo la detección de fallos y la notificación.

A continuación, esta la capa de Administración de la configuración (*Configuration management*) que se encarga de coordinar los cambios de hardware y programación, incluyendo la adición de nuevos equipos y programas, la modificación de sistemas existentes y la eliminación de sistemas y programas obsoletos. A este nivel el inventario de equipos y programas se mantiene y actualiza regularmente (Rouse, M., 2007). El monitoreo permite detectar los cambios en la red y llevar un control adecuado de los dispositivos que conforman la red.

Administración de la contabilidad (*Accounting management*) es apoyada por el monitoreo ya que este permite recopilar información del uso de los recursos de la red (Claise, B., y Wolter, R., 2007). Midiendo de manera exacta los recursos consumidos por una aplicación en un rango de tiempo determinado, con lo cual se pueden emitir facturas adecuadas.

En la capa de Administración del rendimiento (*Performance management*) el monitoreo permite medir diversos aspectos de desempeño para que éste se mantenga a un nivel definido. Permite también detectar y evitar problemas de congestión.

Finalmente, en la Administración de la seguridad (*Security management*), el monitoreo permite supervisar las acciones de los usuarios, verificar que solo accedan a los recursos que les fueron asignados y detectar accesos no autorizados entre algunas tareas más.

1.3 Herramientas para monitoreo de redes de datos

A continuación, se muestra la Tabla 1 que contiene algunas de las herramientas para monitoreo de infraestructura de red. La Tabla 1 lista herramientas para monitoreo representativas, tanto de código abierto como comerciales. En dicha tabla se muestra información técnica acerca de las herramientas y documenta experiencia de uso, limitaciones reportadas, el tipo de alertas que éstas pueden emitir, la licencia de uso, los sistemas operativos soportados por los agentes de monitoreo, el lenguaje de programación del agente y cuáles son los recursos que la herramienta tiene capacidad de monitorear. Según Fatema *et al*. (2014) las herramientas de monitoreo de infraestructuras de uso general suelen utilizar un modelo cliente-servidor instalando un agente en cada sistema que se va a supervisar. Los agentes son software instalado en las máquinas o dispositivos a ser monitoreados que recopilan información del estado de dicho dispositivo. La información recopilada es comúnmente enviada al servidor en donde corre la herramienta para monitorear para que éste la procese y determine si los umbrales se han sobrepasado o no.

De acuerdo con Jennings, N. R., y Wooldridge, M. (1998) un agente es un sistema de computación situado en algún lugar, que es capaz de una acción autónoma en este lugar con el fin de alcanzar los objetivos para los que fue diseñado. Los agentes se instalan en los dispositivos que se quieren monitorear y crean un canal de comunicación con el servidor de monitoreo.

El agente crea procesos que monitorean ciertos atributos del ente monitoreado ya sean estos privados, como carga de CPU, uso de memoria, etc. o servicios remotos sobre otros *hosts*, como HTTP o FTP (Galstad, E., 2016). Por ejemplo, en la Figura 1 se muestra un agente llamado *NRPE*. El agente corre en la máquina remota recopilando información sobre atributos privados como disco y carga en el CPU y atributos públicos como HTTP y FTP, los datos recopilados son enviados al *host* que realiza el monitoreo para su procesamiento. Este agente es utilizado por la herramienta para monitoreo Nagios. El agente *NRPE* se utiliza para monitorear máquinas *Linux*/*Unix*. En el Capítulo II se describe la instalación de otro agente para monitoreo de esta misma herramienta llamado *NSClient++* pero éste está diseñado para monitorear máquinas *Windows*.

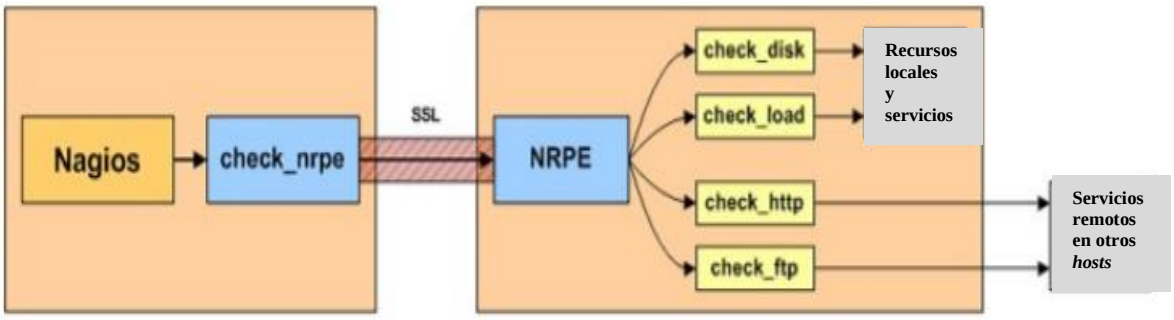

Figura 1. Operación del agente NRPE de Nagios. Galstad, E. (2016).

Los agentes requieren algunos recursos para correr, lo cual puede afectar el rendimiento (Lindros y Tittel, 2015). Es decir, se pueden volver intrusivos y consumir recursos de la máquina en la que operan y generan tráfico en la red debido a la constante comunicación entre cliente y servidor de monitoreo. El agente *NRPE*, por ejemplo, si se ejecuta a través de *SSH* (22) (utilizando el complemento *check_by_ssh*) impone sobrecarga en el CPU tanto de los servidores de monitoreo como en los clientes remotos (Galstad, E., 2016). La afectación se puede convertir en un problema cuando en la red se monitorean cientos de dispositivos.

Una vez que los agentes recogen la información de los dispositivos monitoreados la envían al servidor para que este la procese y dependiendo de si los umbrales son sobrepasados o no se pueden emitir alertas para informar a los administradores sobre la presencia de un evento o problema en la red.

Tabla 1

Herramientas para monitoreo de redes y sus características (Fatema et al., 2014).

Herramienta	Recursos monitoreados	Lenguaje del agente	Sistema operativo para los agentes	Licencia	Alertas	Soporte de mensajería empresarial	Limitaciones reportadas	Experiencia de uso
Nagios (Barth, W. 2008)	Recursos del sistema, red, sensores, aplicaciones y servicios	C	Linux/Unix (Windows a través de un agente proxy)	Código abierto (GLP)	Correo electrónico, SMS, personalizado	No	Dificultad de configuración, Incapacidad para trabajar con alta frecuencia de muestreo, incapacidad de enlazar el servicio debido a la migración a máquina virtual	Monitoreo de la infraestructura de la red, monitoreo de máquina virtual.
Collectd (Cowie, B., 2012)	Recursos del sistema, red, sensores, bases de datos, aplicaciones y servicios	C	Linux/Unix y Mac OS	Código abierto (GPLv2)	No disponible	AMQP (superiores a V5.1)	No hay plataforma de visualización	Recopilación de métricas de despliegues de la nube para fines forenses
Opsview Core (Krizanic, J. et al., 2010)	Recursos del sistema, red, base de datos, aplicaciones y servicios	Perl, C	Linux/Unix, Windows	Código abierto (GLP)	Correo electrónico, SMS, personalizado	No	No disponible	No disponible
Cacti (liang, D. et al., 2009)	Principalmente la red	PHP, C	Sistemas operativos basados en Linux y Windows	Código abierto (GLP)	Alertas audibles, correo electrónico	No	No disponible	Desarrollo integrado de diversas herramientas de monitoreo de terceros
Zabbix (Olups, R. 2010)	Recursos del sistema, red, sensores, bases de datos, aplicaciones y servicios	C, PHP, Java	Linux/Unix, Mac, Windows	Código abierto (GLP)	Correo electrónico, personalizado	XMPP	La función de autodescubrimiento de Zabbix puede ser ineficiente.	En un motor de Cloud Broker para escalar dinámicamente los recursos de la nube, supervisión privada de la nube.

Tabla 1

Herramientas para monitoreo de redes y sus características (Fatema et al., 2014). (Continuación)

Herramienta	Recursos monitoreados	Lenguaje del agente	Sistema operativo para los agentes	Licencia	Alertas	Soporte de mensajería empresarial	Limitaciones reportadas	Experiencia de uso
Open NMS (Pape, C. & Trommer, R., 2012)	Principalmente la red	Java	Linux/Unix, Windows, Mac	Código abierto (GPLv3)	Correo electrónico, SMS	JMS	El servicio de descubrimiento automático tiene una capacidad limitada y su servicio no está generalizado para todos los servicios de red.	No disponible
Ganglia (Massie, M. et al., 2004)	recursos del sistema	C, Perl, PHP, Python	Linux/Unix, Solaris, Windows, Mac	Código abierto (BSD)	No disponible	No	Difícil de personalizar, introduce sobrecarga tanto en los *hosts* y redes debido a las actualizaciones de multidifusión, y la codificación de eventos XML.	Recopilación de datos del lado del servidor para una solución de monitorización basada en la nube y aprovisionamiento de recursos en Rocks Clusters
Hyperic HQ (Hyperic, 2014)	Los recursos del sistema, Red, base de datos, Aplicaciones y servicios	Java	Linux/Unix, Windows, Mac	Código abierto (GPLv2)	Correo electrónico, SMS	No	Alto requerimiento de memoria y Dificultad de personalizar la interfaz gráfica	Para recopilar datos de inventario de software en CloudAlloc, un sistema de monitoreo y reserva para clusters de computación.
IBM Tivoli (IBM Tivoli monitoring, 2014)	Recursos del sistema, red, base de datos, aplicaciones y servicio	Java	Linux/Unix, Windows, Mac	Comercial	Correo electrónico, SMS	No	No disponible	En la monitorización de la nube IBM, en la gestión de servicios empresariales

Tabla 1

Herramientas para monitoreo de redes y sus características (Fatema et al., 2014). (Continuación)

Herramienta	Recursos monitoreados	Lenguaje del agente	Sistema operativo para los agentes	Licencia	Alertas	Soporte de mensajería empresarial	Limitaciones reportadas	Experiencia de uso
Kiwi Application Monitor (Frank, M. et al. 2009)	Procesos de aplicación y actividad del usuario	No disponible	Windows	Código abierto	Personalizado	No	No disponible	En aplicaciones de medición de uso máximo de memoria y el tiempo de CPU en la simulación de un espacio eficiente de cómputo cuántico.
R-OSGi (Rellermeyer, J. et al. 2007)	Aplicaciones distribuidas	Java	No disponible	Código abierto	Correo electrónico, SMS	No	Problema con registro de servicio y configuración estática.	Obtención de información de dependencia para una aplicación distribuida
DAMS (Jiang,H., et al.2010)	Aplicaciones distribuidas	Java	No disponible	Código abierto	No disponible	No	El rendimiento del sistema se ve afectado.	Sistema abierto de la gerencia de la educación de la universidad central de la radio y de la TV.

1.4 Características de las herramientas para monitoreo de redes de datos

En esta sección se discuten las capacidades más importantes de las herramientas asociadas con el proceso del monitoreo, para ello se necesita dejar en claro que es a lo que el concepto se refiere, de acuerdo con Fatema *et al*. (2014), monitoreo es un proceso que identifica de forma completa y precisa la causa raíz de un evento, capturando la información correcta en el momento y al menor costo para determinar el estado de un sistema y dar a conocer el estado de manera oportuna y significativa.

El monitoreo en este sentido es importante si lo que se monitorea son servicios o aplicaciones entregados a los usuarios en donde se especifica en acuerdos de nivel de servicio (SLA´s) las métricas relevantes para un usuario o cliente en particular, ya que la falta o incumplimiento de algún punto especificado en este documento puede conllevar sanciones.

Según Vicente, C. (2005) el monitoreo de las redes de datos tiene al menos dos enfoques: a) El monitoreo activo y b) el monitoreo pasivo. El enfoque activo consiste en el envío de paquetes a determinadas aplicaciones midiendo sus tiempos de respuesta. Este tipo de monitoreo agrega tráfico a la red debido a la constante emisión de paquetes; generalmente se utiliza para medir el rendimiento.

El monitoreo pasivo por su parte se basa en la obtención de datos a partir de recolectar y analizar el tráfico que circula en la red. Este tipo de monitoreo, contrario del enfoque activo, no agrega tráfico a la red y se utiliza para caracterizar el tipo de tráfico en la red y para contabilizar su uso (Vicente, C., 2005).

La Sección 1.4.1 describe las capacidades deseables en una herramienta para monitoreo de redes.

1.4.1 Capacidades deseables en las herramientas para monitoreo de redes de datos

Fatema *et al.* (2014), identifica 29 capacidades deseables para las herramientas de monitoreo: la escalabilidad, la portabilidad, ser no intrusivo, robustez, *multi-tenancy*, interoperabilidad, personalización, extensibilidad, monitoreo compartido de recursos, usabilidad, asequibilidad, capacidad de ser archivado, medición verificable, medición del uso de recursos, medición del uso del servicio, monitoreo de servicio *KPI*, monitoreo de *QoS*, evaluación de riesgos, identificación del estado del componente, monitoreo de la carga del servicio, verificación de la configuración, identificación de la configuración, supervisión del efecto de configuración, supervisión de fallos de seguridad, control de acceso de usuario, actividad del usuario, notificación segura, almacenamiento seguro y dependencia del servicio.

Escalabilidad: El despliegue de las redes puede ser de gran escala, consistente en miles de nodos. Para administrar estos recursos, una herramienta de monitoreo necesita ser

escalable para entregar la información monitoreada de manera oportuna y flexible (Gogouvitis, S. *et al.*, 2013; Aceto, G. *et al.*, 2013; Zanikolas, S., & Sakellariou, R., 2005).

La herramienta debe mantener el mismo desempeño sin importar la cantidad de elementos que se tengan que monitorear además de no perjudicar las operaciones normales de la red.

Portabilidad: Los entornos actuales de las organizaciones incorporan plataformas y servicios heterogéneos. Por lo tanto, la portabilidad de las herramientas de monitoreo, es decir, la capacidad de mover la herramienta de una plataforma a otra es indispensable para facilitar el manejo eficiente (Zanikolas, S., y Sakellariou, R., 2005; Gogouvitis, S. *et al.*, 2013).

Ser no intrusivo: Como se observó en la Sección 1.2 algunas herramientas utilizan agentes para realizar el monitoreo, la potencia computacional consumida por las herramientas para monitorear todos estos recursos puede tener impacto en el desempeño del sistema general. Para atender estos entornos, una herramienta de monitoreo debería consumir la menor capacidad de recursos posible en los sistemas monitoreados para no obstaculizar el desempeño general de los sistemas monitoreados (Gogouvitis, S. *et al*., 2013; Aceto, G. *et al.*, 2013).

Robustez: Las redes de datos representan un entorno dinámicamente cambiante. Es importante que la herramienta de monitoreo detecte cambios en las circunstancias, como la adición o remoción de dispositivos y recursos (Hasselmeyer, P. y Heureuse, N., 2010; Aceto, G. *et al.*, 2013). Una herramienta de monitoreo necesita la capacidad de adaptarse a una nueva situación al continuar su operación en el entorno cambiado, lo cual ayuda a mitigar fallas y proporcionar información monitoreada precisa (Gogouvitis, S. *et al.*, 2013).

Multi-tenancy: Algunas redes pueden ofrecer un entorno *multi-tenant* donde múltiples usuarios comparten los mismos recursos físicos e instancias de aplicación. Esta característica es indispensable para garantizar acuerdos de nivel de servicio y monitoreo de máquina virtual (Cheng, X. *et al.*, 2009; Tovarnak, D. y Pitner, T., 2012). En este tipo de entornos se requiere que la herramienta que monitoree los dispositivos y aplicaciones sea capaz de ofrecer el servicio a niveles lógicos y mantenga la información de los usuarios aislada para que solo aquellos con los privilegios accedan a la parte que les corresponde.

Interoperabilidad: Las redes de las organizaciones pueden incluir estaciones o sucursales independientes y heterogéneas que funcionan como recursos independientes. Una herramienta moderna de monitoreo debe ser capaz de compartir información de monitoreo entre componentes heterogéneos para manejar operaciones colaborativas (Celesti, A. *et al.*, 2010; Hasselmeyer, P. y Heureuse, N., 2010; Tovarnak, D. y Pitner, T., 2012).

Personalización: Existen necesidades particulares de monitoreo para cada uno de los clientes, es decir, algunos están interesados en monitorear servicios o recursos que para otros no son necesariamente relevantes. Lo que se pretende con este requisito es conceder a los clientes la posibilidad de elegir las métricas a supervisar para su servicio. Las herramientas eficientes de monitoreo deben poseer esta capacidad (Fatema, K. *et al.*, 2014).

Extensibilidad: Con el rápido crecimiento de las organizaciones, hay cambios continuos y extensiones a las tecnologías, especialmente en el área de gestión. Dado que las técnicas de monitoreo son fundamentales para la administración de las redes, las herramientas de monitoreo deben ser extensibles y ser capaces de adaptarse a nuevos entornos, como ser capaces de incorporar nuevas métricas de monitoreo. Esto se logra típicamente a través de un diseño modular que permita la definición y adición de nuevos módulos que se encarguen de atender las necesidades que aparezcan en un determinado momento (Zanikolas, S., y Sakellariou, R., 2005).

Monitoreo compartido de recursos: Algunas organizaciones optan por utilizar la virtualización de los recursos físicos del equipo de cómputo para lograr el aislamiento del uso en forma de máquinas virtuales. Las máquinas virtuales comparten los recursos subyacentes mientras que varias aplicaciones comparten los recursos de la máquina virtual. Para evitar la contención de recursos entre las máquinas virtuales o administrar recursos compartidos por aplicaciones en una máquina virtual, se necesita un monitoreo eficiente. Una herramienta de monitoreo necesita la capacidad de supervisar recursos compartidos para manejar tal ambiente (Hasselmeyer, P. y Heureuse, N., 2010).

Usabilidad: Cualquier herramienta de monitoreo debe ser fácil de utilizar, intuitiva, amigable en su instalación y mantenimiento y en la interacción humana (Hasselmeyer, P. y Heureuse, N., 2010).

Asequibilidad: La eficacia en función de los costos de una herramienta de monitoreo (por ejemplo, ser de código abierto) repercute ampliamente en la difusión de aceptación debido a que opciones libres no representan inversión, pero si una amplia capacidad de adaptación a cualquier ambiente de red. Una encuesta de Gartner muestra que más del 85% de las empresas estaban utilizando software de código abierto para reducir el costo y aumentar la *flexibilidad*. (Cirstoiu, C. *et al.*, 2007; Gartner, 2008).

Capacidad de ser archivado: La disponibilidad de datos históricos puede ser útil para analizar e identificar la causa raíz de un problema a largo plazo. Para cumplir con este propósito, una herramienta de monitoreo debe poseer un medio para almacenar datos históricos (Buytaert, J. *et al.*, 2008; Morgan, D. *et al.* 1975).

Medición verificable: La medición verificable significa que los clientes pueden estar seguros de que sus aplicaciones realmente consumen físicamente los recursos que fueron cobrados y que este consumo está justificado basado en una política acordada (Sekar, V., y Maniatis, P., 2011).

Medición del uso de recursos: es la capacidad de capturar el consumo y la información de asignación de los recursos virtuales y físicos, así como la de las aplicaciones. Por ejemplo: cálculo de la hora utilizada y ancho de banda utilizado (Elmroth, E. *et al.*, 2009).

Medición del uso del servicio: En un entorno en donde los servicios se entregan y consumen en base al modelo de suscripción, la medición de los servicios entregados y consumidos es un aspecto esencial del modelo de prestación de servicios (Naik, V. et al., 2014). Esta métrica hace referencia a la capacidad de las herramientas de medir el consumo de un servicio o servicios por parte de los clientes.

Monitoreo de servicio KPI: El monitoreo de los *KPI* es de suma importancia ya que estos contienen información sobre el proceso de medición, el lugar y la unidad. Por ejemplo: El sistema informático debería alcanzar una disponibilidad del 98% durante el período de medición de un mes. La disponibilidad representa, por tanto, la relación del tiempo en el que el servicio trabaja con un tiempo de respuesta inferior a 100% más el tiempo de inactividad previsto para el tiempo de servicio total, medido en el propio servidor. A partir de tal descripción, los valores reales de rendimiento se pueden comparar con los valores de referencia y se calcula el logro. Sobre esta base, pueden llevarse a cabo otras medidas para corregir si es necesario (Frey, S. et al., 2013). Es por esto que llevar un monitoreo adecuado de *KPI* se vuelve fundamental para la administración de los *SLA.*

Monitoreo de QoS: *QoS* denota los niveles de rendimiento, fiabilidad y disponibilidad ofrecidos por una aplicación y por la plataforma o infraestructura que la aloja. El monitoreo de *QoS* es fundamental para los usuarios que esperan que los proveedores ofrezcan las características de calidad anunciadas, y para los proveedores que necesitan encontrar las compensaciones adecuadas entre los niveles de *QoS* y los costos operacionales (Ardagna et al., 2014).

Evaluación de riesgos: Al momento de considerar la utilización de infraestructuras y tecnologías de información resulta imprescindible realizar un análisis adecuado de los riesgos asociados a su implementación para garantizar resultados satisfactorios como consecuencia de su uso y la sostenibilidad de su utilización dentro de una organización en el tiempo (López, M., et al., 2014). Se pueden mitigar los riesgos de forma correcta al aplicar controles de seguridad para disminuir las probabilidades de que puedan explotarse las vulnerabilidades del bien y derivar en la implementación de amenazas. Una herramienta para monitorear la red de manera adecuada debería tener esta capacidad de evaluar los riesgos.

Identificación del estado del componente: La capacidad de detectar el estado de los componentes en la red permite notificar cuando uno de ellos presenta errores o sobrecarga. Esta cualidad ayuda por ejemplo a generar reportes de disponibilidad de nodos, que es necesaria para asegurar el cumplimiento de *SLA* (Fatema *et al.*, 2014).

Monitoreo de la carga del servicio: La habilidad de medir la carga del servicio permite operaciones tales como predecir la necesidad de más recursos o determinar el derroche de recursos. Garantiza la disponibilidad adecuada de recursos para satisfacer la demanda de capacidad, necesaria para asegurar un nivel de calidad de servicio y para atender diversas actividades de gestión operacional (Fatema *et al.*, 2014).

Verificación de la configuración y *Supervisión del efecto de configuración*: Las configuraciones iniciales pueden contener el conjunto mínimo de recursos necesarios para un determinado servicio. Los recursos se pueden agregar o liberar dependiendo de la carga variable que resulta en la reconfiguración en tiempo de ejecución. Un sistema de gestión de la configuración debe ser capaz de verificar configuraciones especificas e identificar el efecto de otras. La habilidad de detectar estos cambios en la configuración permite crear escenarios para observar el comportamiento de la red luego de realizar ciertas configuraciones (Ferretti, S. *et al.*, 2010; Fatema *et al.*, 2014).

Identificación de la configuración: La variabilidad de la configuración es causada por elementos de configuración inconsistentes en equipos o dispositivos. Para evitar la desviación de la configuración, se requiere de una herramienta capaz de mantener información detallada sobre las direcciones de red de los dispositivos de hardware, así como las versiones de software que se ejecutan en ellas y las actualizaciones que se han aplicado (Rouse, M., 2016).

Supervisión de fallos de seguridad: El monitoreo de seguridad supervisa a los servidores virtuales y físicos para evaluar y medir continuamente los comportamientos de datos, aplicaciones o infraestructura para posibles amenazas de seguridad. Esto asegura que la infraestructura y la plataforma funcionen óptimamente mientras se minimiza el riesgo de fugas de datos o accesos no autorizados (Lord N., 2016).

Control de acceso de usuario: El control de acceso es de vital importancia ya que se trata de permitir que un usuario acceda a ciertos recursos en la red. En los entornos actuales con cientos de usuarios compartiendo los mismos recursos se debe tener un mecanismo adecuado que permita tener control sobre quienes acceden a que recursos o a que información. Por esto esta capacidad debe estar incluida en una herramienta para monitoreo de la red (Muddana, L y Aluvalu, R. 2015).

Actividad del usuario: El monitoreo de la actividad de los usuarios es una característica fundamental para llevar un control adecuado sobre que usuarios acceden a que aplicaciones. Esta característica permite introducir seguridad a la red ya que se está atento de que es lo que los usuarios están realizando y asegura que no ejecutan aplicaciones que puedan introducir alguna vulnerabilidad (Fatema *et al.*, 2014).

Notificación segura: La información de registro de red puede ser de naturaleza sensible y puede revelar información sobre otros clientes, por lo que un proveedor no puede permitir el acceso directo a esta información. Es por esto que las herramientas deben definir mecanismos de notificación segura. Se debe cuidar y asegurar de que un cliente es notificado de manera oportuna si sus máquinas son atacadas o si están comprometidas (Fatema *et al.*, 2014).

Almacenamiento seguro: El almacenamiento seguro es fundamental en los entornos actuales en donde la información se puede tener almacenada en múltiples servidores. El acceso a la información es una preocupación general y los usuarios suelen ser desconfiados, se debe tener un mecanismo que asegure que los datos se encuentran resguardados y que nadie excepto el dueño puede acceder a ellos, ni siquiera el administrador de la red; y en caso de que la seguridad sea comprometida y se acceda la información se encuentre cifrada (Kumar, A. et al., 2012). El monitoreo del almacenamiento seguro de la información de los usuarios, así como el almacenamiento seguro de los archivos generados por la herramienta en si son características deseables en las herramientas para el monitoreo de la red.

Dependencia del servicio: Debido a la constitución compleja de los componentes de las redes, las fallas pueden ocurrir de maneras diferentes, por ejemplo, sobrecarga de servidor o fallo de red / hardware / servicio. Se debe tener en cuenta el monitoreo de la dependencia

del servicio por parte de las herramientas para el monitoreo de red para saber cuáles son las consecuencias de un fallo en un momento determinado (Fatema *et al.*, 2014).

1.4.2 Áreas operacionales

Por otro lado, Fatema *et al.* (2014) indican que el monitoreo realizado de manera activa y con una gestión adecuada facilita la operación de siete áreas relevantes: contabilidad y facturación, administración de SLA, aprovisionamiento de servicios/recursos, planificación de la capacidad, gestión de la configuración, garantía de seguridad y privacidad y por último gestión de fallos.

Contabilidad y facturación: La noción de proporcionar computación como un servicio de utilidad depende en gran medida de la capacidad de registrar y contabilizar la información de uso en la que se basan los esquemas de facturación. La contabilidad y facturación precisas dependen de la capacidad de capturar el consumo y la información de asignación de los recursos virtuales, así como la de las aplicaciones (por ejemplo, cálculo de la hora utilizada, ancho de banda utilizado) (Elmroth, E. *et al.*, 2009). Información relacionada a esta área se discute por los siguientes autores: Park, K. *et al.*, 2013 y Sekar, V. & Maniatis, P., 2011.

Administración de SLA: Un acuerdo de nivel de servicio (*SLA*) representa un contrato firmado entre un proveedor de servicios y un cliente especificando los términos de una oferta de servicio incluyendo calidad de servicio (*QoS*), precios y sanciones en caso de violar los términos acordados (Emeakaroha, V. *et al.*, 2010; Haiteng, Z. *et al.* 2012). La gestión de *SLA* es un área importante para los proveedores de servicios, ya que la garantía de cumplimiento de *SLA* es inevitable para la satisfacción del cliente. Se espera que los proveedores cumplan con los requisitos de *QoS*, así como con los indicadores de desempeño clave (*KPI*, por sus siglas en inglés) para los servicios con el fin de hacer cumplir sus términos de *SLA* acordados. El monitoreo es esencial para lograr estos objetivos. Las capacidades de monitoreo necesarias para apoyar las operaciones en esta área incluyen la capacidad de medir los parámetros de *QoS*, almacenar y analizar los datos, la medición del consumo de recursos y la evaluación de parámetros *SLA*. Estas capacidades se esperan de una herramienta de monitoreo con el propósito de la gestión de *SLA* (Haiteng, Z. *et al.* 2012; Comuzzi, M. *et al.*, 2009; Palacios, M. *et al.*, 2012; Emeakaroha. V. *et al.*, 2010).

Provisionamiento de servicios/recursos: La provisión de servicios/recursos implica la asignación de recursos de forma óptima para que coincida con la carga de trabajo (Ferrer A. *et al.*, 2012). El provisionamiento se puede implementar de dos maneras: a) provisionamiento estático en el que las máquinas virtuales se crean con un tamaño especificado y luego se consolidan en un conjunto de servidores físicos. La capacidad de la máquina virtual no cambia; y b) provisión dinámica: la capacidad de la máquina virtual se ajusta dinámicamente para adaptarse a las fluctuaciones de la carga de trabajo (Meng, X. *et al.*, 2010). La capacidad de medir el consumo general de recursos de un sistema, junto con la capacidad de medir el consumo de recursos por servicio (que identifica la cantidad de recursos que cada servicio necesita) es esencial para un provisionamiento eficiente. Además, la capacidad de evaluar el riesgo y la calidad de servicio es necesaria para tomar decisiones efectivas sobre

provisionamiento, tales como asignar o liberar recursos para asegurar que la calidad no se vea comprometida o no se desperdicien recursos (Ferrer A. *et al.*, 2012; Zhang, Q. *et al.*, 2007).

Planificación de la capacidad: La planificación de la capacidad garantiza la disponibilidad adecuada de recursos para satisfacer la demanda de capacidad, necesaria para asegurar un nivel de calidad de servicio. Por ejemplo, la recuperación de desastres y el mantenimiento de copias de seguridad (Pueschel, T. & Neumann, D., 2009). La habilidad de medir el uso de capacidad permite operaciones tales como predecir la necesidad de más recursos o determinar el derroche de recursos (Meng, X. *et al.*, 2010).

Gestión de la configuración: La configuración es un conjunto de parámetros y valores que determinan el comportamiento de los dispositivos y del software (Sekiguchi, A. et al., 2012). Las configuraciones iniciales pueden contener el conjunto mínimo de recursos necesarios para un determinado servicio. Los recursos se pueden agregar o liberar dependiendo de la carga variable que resulta en la reconfiguración en tiempo de ejecución. Un sistema de gestión de la configuración debe ser capaz de verificar configuraciones especificas e identificar posibles cambios (Ferretti, S. *et al.*, 2010).

Garantía de seguridad y privacidad: La capacidad de detectar brechas o ataques de seguridad es esencial y el monitoreo puede ayudar respecto a esto (Krutz, R., & Dean, R., 2010), por ejemplo, identificando un proceso malicioso que consume recursos del sistema desaprobados. Para garantizar la seguridad de los servicios, es importante asegurarse de que la herramienta que se utiliza para el monitoreo no debe introducir ninguna vulnerabilidad. Las capacidades de monitoreo tales como el control de acceso basado en el usuario, la notificación segura y el almacenamiento son esenciales para apoyar esta área operativa (Vaquero, L. *et al.*, 2011; Subashini, S., y Kavitha, V., 2011).

Gestión de fallos: Esta es quizá una de las áreas que el monitoreo más apoya, si se hace de manera continua permite pronosticar y detectar oportunamente los fallos, que pueden ser manejados proactivamente reemplazando los componentes sospechosos (Bala, A., y Chana, I., 2012). Debido a la constitución generalmente compleja de las redes, las fallas pueden ocurrir de diversas maneras. Por ejemplo, sobrecarga de servidor o fallo de red/hardware/servicio (Jhawar, R. *et al.*, 2013).

1.5 Análisis de las herramientas para monitoreo de redes de datos

La Tabla 2 muestra el análisis de las herramientas listadas en la Tabla 1, este análisis fue realizado por Fatema *et al.* (2014) y el cálculo se realizó de la siguiente manera:

La segunda columna de la tabla muestra el porcentaje implementado por las herramientas. En el cálculo, se asignó *1* si la herramienta tiene una capacidad particular y *0* si no. También está la asignación de *0.5* si la herramienta implementa parcialmente tal capacidad. La suma de esos valores es usada para calcular el porcentaje asignado. De acuerdo con la Tabla 2 un *1* es equivalente a *si*, *0* implica *no* y *0.5* representa *limitado*. Las capacidades que han anotado *0* para todas las herramientas son excluidas del cálculo del porcentaje promedio ponderado de capacidades cubiertas por cada herramienta que se presentan en la última fila de las tablas. Para las herramientas con múltiples versiones – por

ejemplo, Nagios, Opsview e Hyperic, las capacidades se diferencian basados en el superconjunto de características de todas las versiones.

A continuación, se muestra un ejemplo de cómo realizar el cálculo del porcentaje implementado para la *Escalabilidad* y para la *Usabilidad*:

Para el cálculo de la *Escalabilidad*:

Se tienen doce herramientas en total (Nagios, Collected, Opsview, Cacti, Zabbix, Open NMS, Ganglia, Hyperic, IBM Tivoli, Kiwi Monitor, DAMS y RTD) de las cuales 5 implementan la característica (Zabbix, Ganglia, Hyperic, IBM Tivoli y DAMS), una la implementa limitado (Opsview) y 6 no la implementan (Nagios, Collected, Cacti, Open NMS, Kiwi Monitor y RDT). Las que implementan la característica anotan un 1, la que implementa limitado anota 0.5 y las que no implementan anotan 0.

Entonces la suma total es 1+1+1+1+1+0.5+0+0+0+0+0+0= 5.5

Se realiza una regla de tres:

(5.5*100) /12

550/12=45.83%

Se aplica redondeo y se obtiene el porcentaje implementado de la *Escalabilidad* de un 46%.

Para la *Usabilidad*:

Once herramientas implementan la *Usabilidad* (Collected, Opsview, Cacti, Zabbix, Open NMS, Ganglia, Hyperic, IBM Tivoli, Kiwi Monitor, DAMS y RTD) estas anotan 1 y una herramienta no la implementa (Nagios) la cual anota 0.

Entonces la suma es 0+1+1+1+1+1+1+1+1+1+1+1=11

Regla de tres:

(11*100) /12

1100/12=91.66%

Se aplica redondeo y se obtiene el porcentaje implementado de la *Usabilidad* de un 92%.

La Tabla 2 muestra un grupo de herramientas con una amplia implementación de la *portabilidad, personalización, extensibilidad, usabilidad, asequibilidad, medición del uso de recursos, identificación del estado del componente, capacidad de verificación de la configuración, capacidad de identificar desviaciones en la configuración* y *la capacidad de monitoreo del efecto de la configuración*.

Estas herramientas son débiles en la *escalabilidad, no intrusividad, robustez, multi-tenancy, capacidad de consumo de recursos por servicio, monitoreo de la calidad del*

servicio, *evaluación de riesgos* y *capacidades de monitoreo de la carga de servicio*. La *interoperabilidad* y otras capacidades relacionadas con la seguridad y privacidad son menos implementadas por estas herramientas como lo demuestran sus porcentajes calculados (ver la columna *Porcentaje implementado* en la Tabla 2). Algunas capacidades deseables como la *medición verificable* y *servicio de monitoreo KPI* no son implementadas por ninguna de las herramientas de este grupo.

La última fila de la Tabla 2 muestra el porcentaje cubierto por cada herramienta de monitoreo. Esta columna proporciona una comparación relativa del número de capacidades implementadas por cada herramienta. Como se muestra en la Tabla 2, la herramienta de monitoreo Hyperic posee el porcentaje más alto, implementa 85% de todas las capacidades. Todas las herramientas de monitoreo encuestadas anotaron más del 50% excepto Ganglia (50%), Kiwi Monitor (33%), DAMS (30%) y RDT (30%).

A continuación, se muestra un ejemplo de cómo se realiza el cálculo del porcentaje cubierto por las herramientas.

Cálculo para la herramienta Hyperic:

Se tiene un total de 29 características (*Escalabilidad, Portabilidad, Ser no intrusivo, Robustez, Multi-tenancy, Interoperabilidad, Personalización, Extensibilidad, Monitoreo compartido de recursos, Usabilidad, Asequibilidad, Capacidad de ser archivado, Medición verificable, Medición del uso de recursos, Medición del uso del servicio, Monitoreo de servicio KPI, Monitoreo de QoS, Evaluación de riesgos, Identificación del estado del componente, Monitoreo de la carga del servicio, Verificación de la configuración, Identificación de la configuración, Supervisión del efecto de configuración, Supervisión de fallos de seguridad, Control de acceso de usuario, Actividad del usuario, Notificación segura, Almacenamiento seguro y Dependencia del servicio*), pero las que anotaron cero en todas las herramientas (*Medición verificable* y *Monitoreo de servicio KPI*) son excluidas del cálculo. Por lo tanto, quedan 27 características en total.

Hyperic tiene *si* en 22 capacidades (*Escalabilidad, Portabilidad, Robustez, Multi-tenancy, Personalización, Extensibilidad, Monitoreo compartido de recursos, Usabilidad, Capacidad de ser archivado, Medición del uso de recursos, Medición del uso del servicio, Monitoreo de QoS, Evaluación de riesgos, Identificación del estado del componente, Monitoreo de la carga del servicio, Verificación de la configuración, Identificación de la configuración, Supervisión del efecto de configuración, Supervisión de fallos de seguridad, Control de acceso de usuario, Almacenamiento seguro y Dependencia del servicio*) estas anotan 1, tiene *limitado* en 2 (*Ser no intrusivo* y *Asequibilidad*) que anotan 0.5 y tiene *no* en 3 (*Interoperabilidad*, *Actividad del usuario* y *Notificación segura*).

Por lo tanto, la suma es:

22+0.5+0.5=23

Regla de tres:

(23*100) /27

2300 /27=85.18%

Se aplica redondeo y se obtiene el porcentaje cubierto por Hyperic del total de las capacidades mencionadas de un 85%.

Tabla 2

Análisis de las herramientas para monitoreo (Fatema et al., 2014).

No.	Capacidad/ características	Porcentaje implementado	Nagios	Collectd	Opsview	Cacti	Zabbix	Open NMS	Ganglia	Hyperic	IBM Tivoli	Kiwi Monitor	DAMS	RDT
1	Escalabilidad	46%	No	No	Limitado	No	Si	No	Si	Si	Si	No	Si	No
2	Portabilidad	79%	Limitado	Limitado	Si	Limitado	Si	Si	Si	Si	Si	No	Si	Si
3	Ser no intrusivo	50%	Limitado	Limitado	Si	No	Si	Si	Limitado	Limitado	Si	No	No	No
4	Robustez	33%	No	No	No	No	No	No	Si	Si	Si	No	No	Si
5	Multi-tenancy	33%	Si	No	Si	No	No	No	No	Si	Si	No	No	No
6	Interoperabilidad	25%	No	Si	No	No	Si	Si	No	No	No	No	No	No
7	Personalización	100%	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si
8	Extensibilidad	100%	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si
9	Monitoreo compartido de recursos	42%	Si	Si	Si	No	Si	No	No	Si	No	No	No	No
10	Usabilidad	92%	No	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si
11	Asequibilidad	79%	Limitado	Si	Limitado	Si	Si	Si	Si	Limitado	No	Si	Si	Si
12	Capacidad de ser archivado	67%	Si	No	Si	Si	Si	Si	Si	Si	Si	No	No	No
13	Medición verificable	0%	No	No	No	No	No	No	No	No	No	No	No	No
14	Medición del uso de recursos	75%	Si	Si	Si	Si	Si	Si	Si	Si	Si	No	No	No

Tabla 2

Análisis de las herramientas para monitoreo (Fatema et al., 2014) (Continuación).

	Capacidad/ características	Porcentaje implementado	Nagios	Collectd	Opsview	Cacti	Zabbix	Open NMS	Ganglia	Hyperic	IBM Tivoli	Kiwi Monitor	DAMS	RDT
15	Medición del uso del servicio	50%	Si	Si	Si	No	Si	No	No	Si	No	Si	No	No
16	Monitoreo de servicio KPI	0%	No	No	No	No	No	No	No	No	No	No	No	No
17	Monitoreo de QoS	50%	Si	No	Si	No	Si	No	No	Si	Si	Si	No	No
18	Evaluación de riesgos	58%	Si	No	Si	Si	Si	Si	No	Si	Si	No	No	No
19	Identificación del estado del componente	100%	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si
20	Monitoreo de la carga del servicio	50%	Si	Si	Si	No	Si	No	No	Si	No	Si	No	No
21	Verificación de la configuración	75%	Si	Si	Si	Si	Si	Si	Si	Si	Si	No	No	No
22	Identificación de deriva de la configuración	75%	Si	Si	Si	Si	Si	Si	Si	Si	Si	No	No	No
23	Supervisión del efecto de configuración	75%	Si	Si	Si	Si	Si	Si	Si	Si	Si	No	No	No
24	Supervisión de fallos de seguridad	33%	Si	No	No	No	Si	No	No	Si	Si	No	No	No

Tabla 2

Análisis de las herramientas para monitoreo (Fatema et al., 2014) (Continuación).

	Capacidad/ características	Porcentaje implementado	Nagios	Collectd	Opsview	Cacti	Zabbix	Open NMS	Ganglia	Hyperic	IBM Tivoli	Kiwi Monitor	DAMS	RDT
25	Control de acceso de usuario	50%	No	No	Si	Si	Si	Si	No	Si	Si	No	No	No
26	Actividad del usuario	17%	No	No	No	No	No	No	No	No	Si	Si	No	No
27	Notificación segura	17%	No	No	No	No	No	Si	No	No	Si	No	No	No
28	Almacenamiento seguro	17%	No	No	No	No	No	No	No	Si	Si	No	No	No
29	Dependencia del servicio	21%	No	No	No	No	No	No	No	Si	No	No	Si	Si
	Porcentaje cubierto por las herramientas		61%	52%	70%	46%	78%	59%	50%	85%	78%	33%	30%	30%

Capítulo II. Nagios Core

Este Capítulo documenta la instalación, configuración y funcionalidad de la herramienta para el monitoreo de redes de datos de código abierto Nagios Core versión 4.2.4.

2.1 Escenario de experimentación

Para realizar el análisis práctico y mostrar la instalación, configuración y funcionalidad de la herramienta Nagios Core, se utilizó el escenario de experimentación mostrado en la Tabla 3 y Figura 2:

Tabla 3
Escenario de experimentación para despliegue de Nagios Core.

Nombre del dispositivo	Tipo de equipo	Sistema Operativo	Dirección IP	Servicios Monitoreados
Ubuntu	Lap-Top	Ubuntu 14.04	192.168.137.119	Current load, current users, http, ping, pop, root partition, ssh, swap usage, total processes.
Winserver	Lap-Top	Windows 10	192.168.137.1	C:\ Drive Space, CPU load, explorer, Memory Usage, NSClient++ Version, Uptime
PC1	Pc	Windows 7	192.168.137.2	C:\ Drive Space, CPU load, explorer, Memory Usage, NSClient++ Version, Uptime
PC2	Pc	Windows 7	192.168.137.3	C:\ Drive Space, CPU load, explorer, Memory Usage, NSClient++ Version, Uptime
CiscoSystems	Enrutador	Cisco IOS	192.168.137.5	Ping, port fa0/1 link status, Uptime

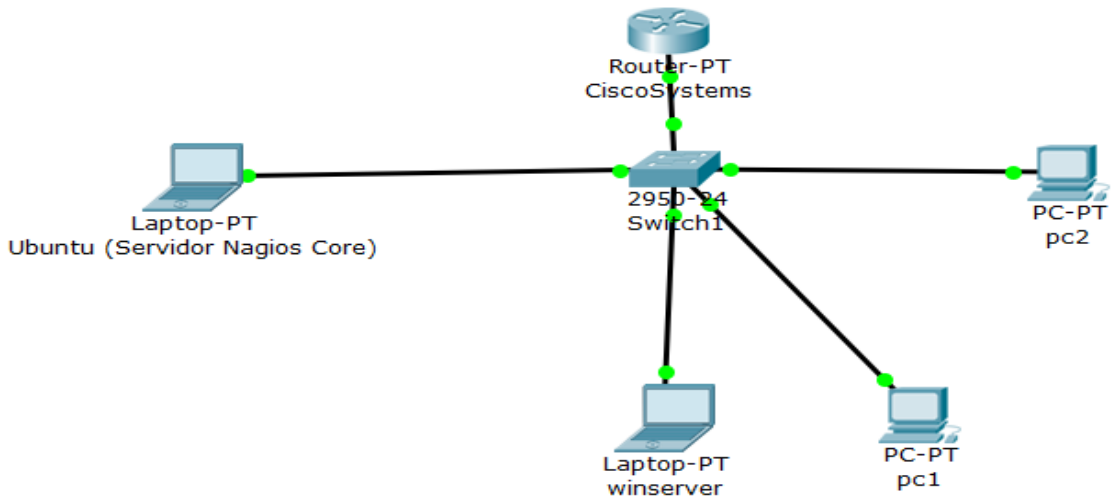

Figura 2. Diagrama de red utilizado para el despliegue de Nagios Core.

2.2 Instalación

Antes de iniciar a describir el proceso de instalación es importante mencionar que Nagios Core es la versión libre y existe una versión comercial llamada Nagios XI. Esta versión incluye más características comparada con la versión libre Nagios Core. Nagios XI tiene dos versiones: *Standard Edition* y *Enterprise Edition*. La versión *Standard Edition* tiene los siguientes precios (Nagios Enterprises., 2017):

- 100 nodos: $1,995 Dólares
- 200 nodos: $2,995 Dólares
- Cantidad ilimitada de nodos: $4,995 Dólares

La versión *Enterprise Edition* tiene los siguientes precios (Nagios Enterprises., 2017):

- 100 nodos: $3,495 Dólares
- 200 nodos: $4,495 Dólares
- Cantidad ilimitada de nodos: $6,495 Dólares

La licencia tiene una duración de doce meses, no hay restricciones en el número de servicios que pueden ser monitoreados, incluye mantenimiento y soporte por correo electrónico. El soporte telefónico tiene un costo adicional de $995 Dólares 5 llamadas o $1,495 Dólares 10 llamadas (Nagios Enterprises., 2017).

Requerimientos mínimos de *software* para la instalación de Nagios Core

1. Una máquina que ejecuta Linux o variante UNIX.
2. Acceso a Internet.
3. Compilador C instalado.
4. Biblioteca de gráficos GD de Thomas Boutell, versión 1.6.3 o superior.
5. Un servidor web (Nagios Enterprises, 2016).

Requerimientos mínimos de *hardware* para la instalación de Nagios Core

La cantidad de recursos de hardware requerida por Nagios Core depende de la cantidad de dispositivos y servicios que se desean monitorear, ver la Tabla 4.

Tabla 4
Requerimientos mínimos de hardware para instalar Nagios Core (Nagios Enterprises., 2014)

Nodos / *hosts* monitoreados	Servicios monitoreados	Espacio en disco duro	Núcleos de CPU	RAM
50	250	40 *GB*	1 - 2	1 – 4 *GB*
100	500	80 *GB*	2 – 4	4 – 8 *GB*
> 500	> 2500	120 *GB*	> 4	> 8 *GB*

Instalación de Nagios Core

Esta sección tiene como objetivo documentar la instalación de Nagios Core versión 4.2.4 desde el código fuente sobre el sistema operativo *Ubuntu 14.04.2* de 32 bits. La razón por la que se decidió utilizar *Ubuntu* como sistema operativo es que en la documentación oficial de Nagios Core se encontró que solo está documentada la guía de instalación rápida para los sistemas operativos Fedora, openSUSE, Ubuntu y CentOS de los cuales el autor está más relacionado con Ubuntu. Otra de las razones es que instalar aplicaciones y descargar paquetes en esta distribución *Linux* es más fácil y existe mayor cantidad de información en la red.

En cuanto a la versión de Nagios Core 4.2.4 esta es la versión estable reciente liberada el 7 de diciembre de 2016. Se instaló sobre *Ubuntu* de 32 bits. Al momento de descargar el paquete no especifica si es de 32 o 64 bits, pero en la documentación se encontró que existen despliegues en plataformas de 32 y 64 bits.

Todos los comandos deben ser ejecutados con permisos de usuario *root*. El siguiente comando se ejecuta para cambiar a un *shell* de *root*:

sudo -i

A continuación, se instalan el compilador de C (*GCC*), el servidor *web* (apache2) y *php* con el siguiente comando:

sudo apt-get install wget build-essential apache2 php5 php5-gd libgd-dev unzip

Descarga de Nagios core y complementos

Los siguientes comandos permiten descargar el código de Nagios Core y sus complementos:

Primero se cambia al directorio *tmp*.

cd /tmp

Se descarga Nagios Core 4.2.4:

wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.2.4.tar.gz

Se descargan los complementos *Nagios-plugins 2.1.2*:

wget http://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz

Agregar el usuario Nagios y el grupo

A continuación, se agrega el usuario apropiado y el grupo para que se ejecute el proceso Nagios.

usseradd nagios

groupadd nagcmd

usermod -a -G nagcmd nagios

Se agrega también el usuario apache *www-data* al grupo *nagcmd*.

usermod -a -G nagios,nagcmd www-data

Instalación de Nagios Core

Extracción del contenido de los paquetes.

tar zxvf nagios-4.2.4.tar.gz

tar zxvf nagios-plugins-2.1.4.tar.gz

Cambio al nuevo directorio e instalación de los paquetes.

cd nagios-4.2.4

Se ejecuta el archivo de configuración Nagios:

./configure --with-command-group=nagcmd --with-mail=/usr/bin/sendmail –with-httpd-conf=/etc/apache2/

La Figura 3 muestra el resumen de la ejecución del comando anterior.

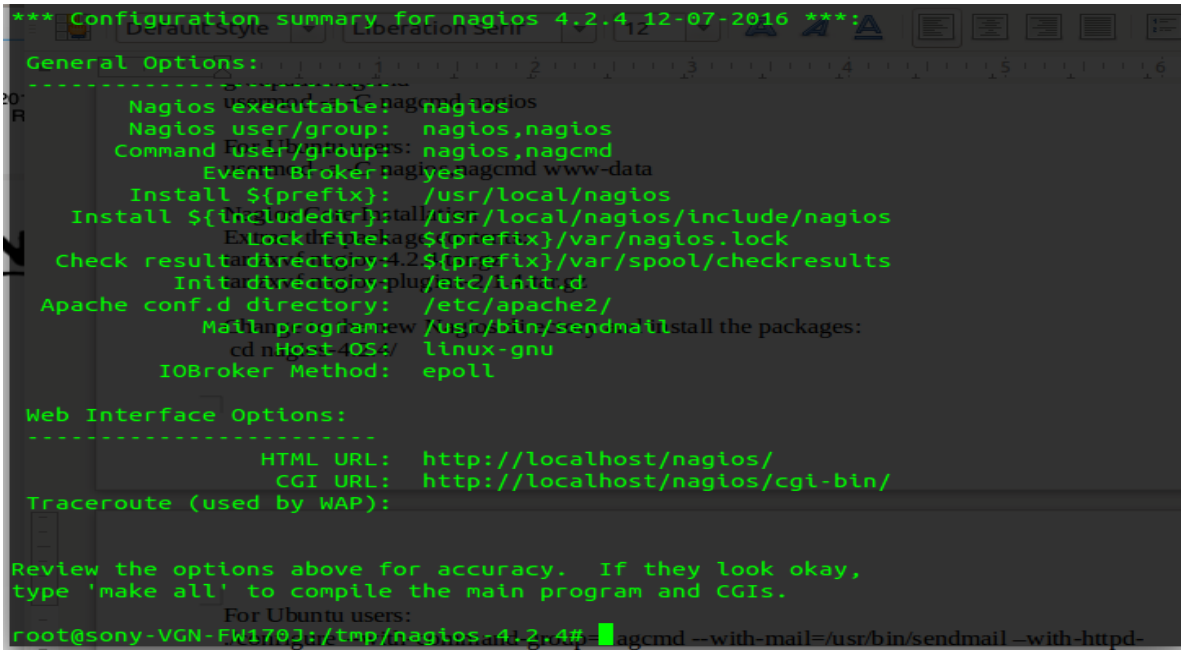

Figura 3. Resumen de la ejecución del comando configure durante la instalación Nagios Core.

Compilación de los archivos binarios de Nagios.

make all

Instalación de los archivos binarios de Nagios.

make install

Instalación del script de inicio.

make install-init

Instalación de los ficheros de configuración.

make install-config

Directorio de comandos externos.

make install-commandmode

Instalación del archivo de configuración de Nagios para *Apache*. La instalación de este archivo permite visualizar la interfaz *web* de Nagios en *Apache*.

make install-webconf

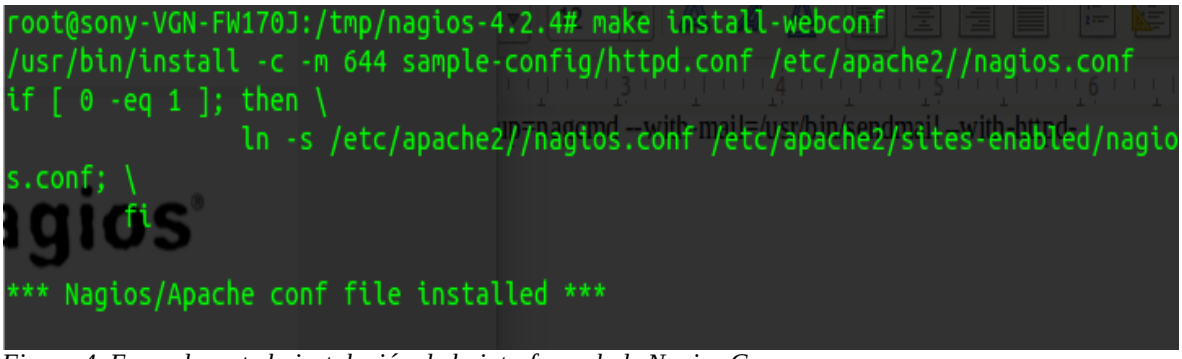

Figura 4. Error durante la instalación de la interfaz web de Nagios Core.

Al ejecutar el comando *make install-webconf* la consola despliega la salida mostrada en la Figura 4, a pesar de que la salida dice que el archivo se instaló, cuando en pasos posteriores se intenta acceder a la interfaz *web*, el servidor responde con el *error 404*. Esto es porque Nagios trató de crear el archivo *nagios.conf* dentro de la ruta */etc/apache2* pero en *Ubuntu* debe colocarse en el directorio */etc/apache2/sites-enabled*. Para corregir este error se debe ejecutar el comando siguiente:

sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf

A continuación se verifica que el archivo *nagios.conf* se encuentra en el directorio */etc/apache2/sites-enabled*:

sudo ls -l /etc/apache2/sites-enabled/

Se copia el archivo *eventhandlers* al directorio *libexec*:

cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/

Se asigna el archivo al usuario nagios:

chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers

Posteriormente se revisa si la instalación y configuración están sin errores:

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

La Figura 5 muestra la salida de la ejecución del comando anterior. Si algo hubiese salido mal es en este punto que se podrían ver los errores o advertencias.

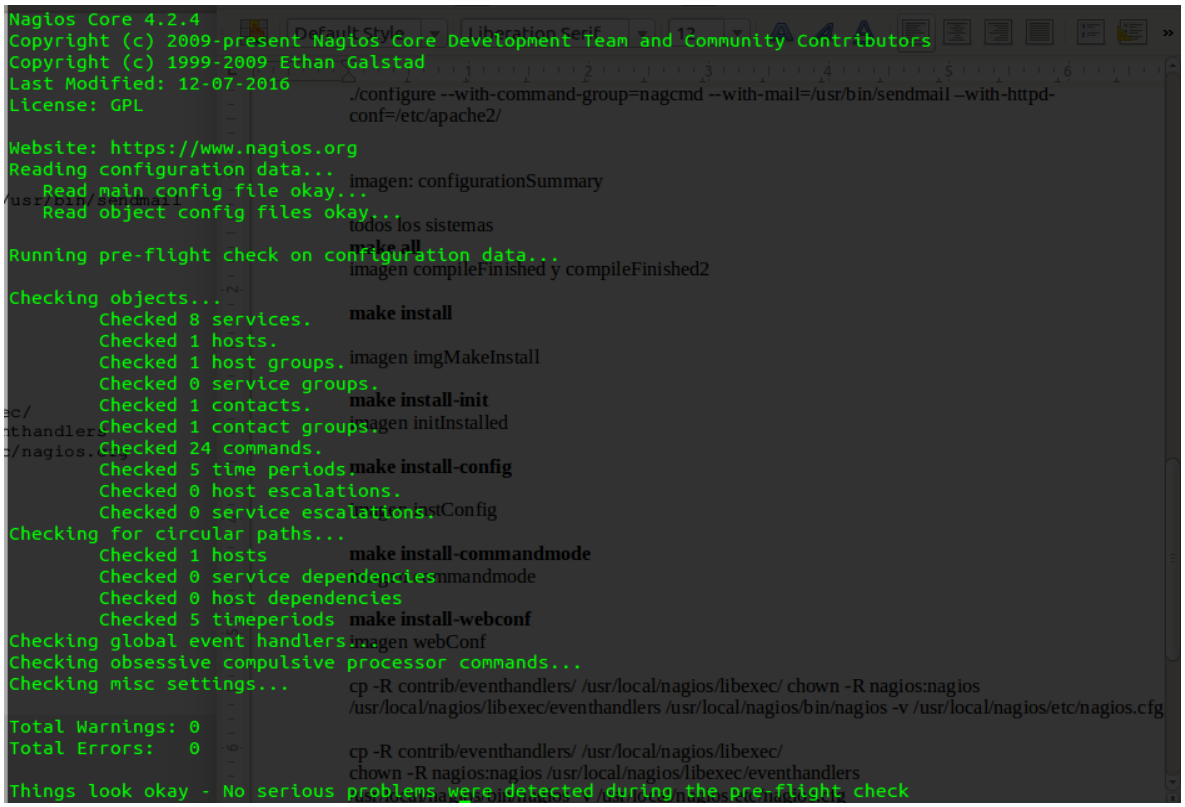

Figura 5. Verificación de errores durante la instalación de Nagios Core.

Se habilita el sitio *web* para Nagios:

sudo a2ensite nagios

Se habilita el módulo CGI de Apache:

sudo a2enmod rewrite cgi

Creación de usuario predeterminado para acceso *web*

Se agrega el usuario predeterminado (*nagiosadmin*):

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

Instalación de los complementos de Nagios Core

Se debe posicionar en el directorio *nagios-plugins-2.1.4:*

cd /tmp/nagios-plugins-2.1.4/

Se ejecuta la configuración de los complementos de Nagios:

./configure --with-nagios-user=nagios --with-nagios-group=nagios

Se realiza la instalación:

make

make install

Configuración del servicio Nagios

El comando siguiente registrará el demonio de Nagios para que se ejecute al iniciar el sistema:

sudo update-rc.d nagios defaults

Interfaz *web* Nagios

Luego de haber realizado los pasos anteriores se está en condición de ingresar a la interfaz *web* de Nagios. Para la autenticación es necesario introducir las credenciales definidas al agregar el usuario *nagiosadmin*.

http://<ip.del.servidor.nagios>/nagios

La Figura 6 muestra como luce la interfaz *web* de Nagios Core.

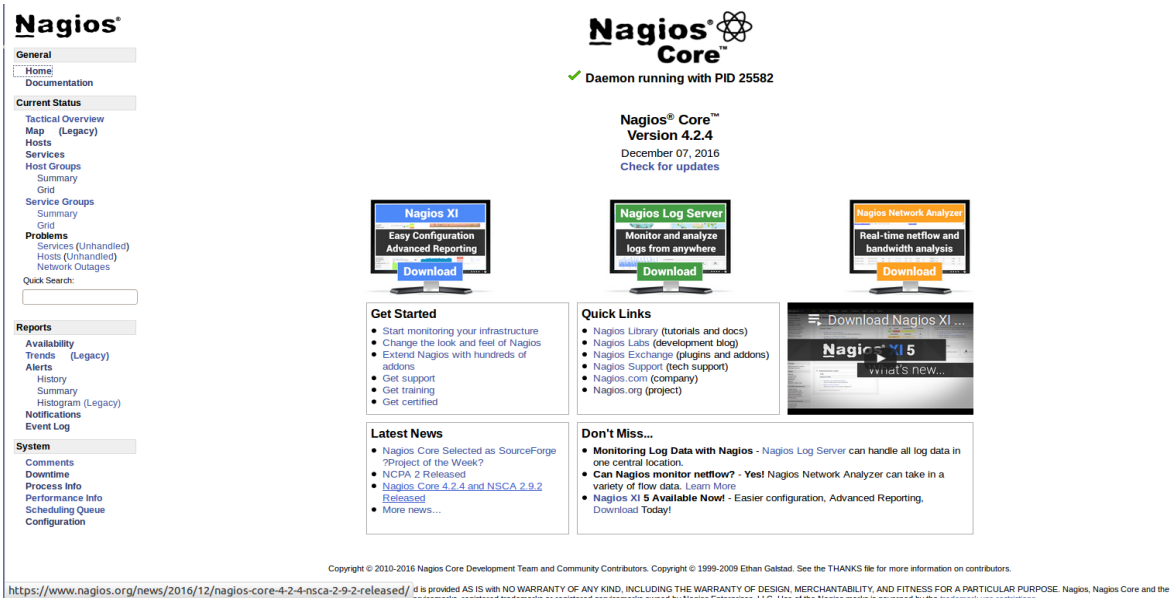

Figura 6. Interfaz web de Nagios Core.

2.3 Configuración

Una vez terminada la instalación de Nagios Core lo único que se tiene en la interfaz es el monitoreo de la máquina local. A fin de monitorear más dispositivos de red o supervisar

procesos o servicios en particular es necesario realizar la configuración que permita personalizar la configuración predeterminada. La configuración de Nagios puede ser un tanto compleja y se debe entender la estructura de los directorios que conforman Nagios, existen foros en la *web* y un foro de soporte Nagios si se desea conocer más información de alguna característica que al lector interese. La configuración que se describe a continuación se limita al procedimiento para definir al contacto de correo electrónico al que se enviaran las notificaciones, la manera de definir y probar servicios que monitorean procesos en particular, los pasos necesarios para instalar *NSClient ++* sobre *Windows* a fin de monitorear una máquina que ejecuta este sistema operativo y a los pasos necesarios para incluir el monitoreo de un *enrutador*.

Configuración de cuenta de correo electrónico

Los archivos de configuración de Nagios se encuentran en el directorio */usr/local/nagios/etc*. El archivo que se debe modificar para definir el contacto al que se enviarán las alertas se llama *contacts.cfg* y se encuentra en */usr/local/nagios/etc/objects/*. Lo único que se debe hacer es abrir dicho archivo con un editor de texto y cambiar el correo predeterminado (*nagios@localhost*) por la dirección de correo que se quiera utilizar. La Figura 7 muestra como luce el archivo y la línea que se debe editar.

Se ejecuta el comando siguiente para editar el archivo *contacts.cfg*:

sudo nano /usr/local/nagios/etc/objects/contacts.cfg

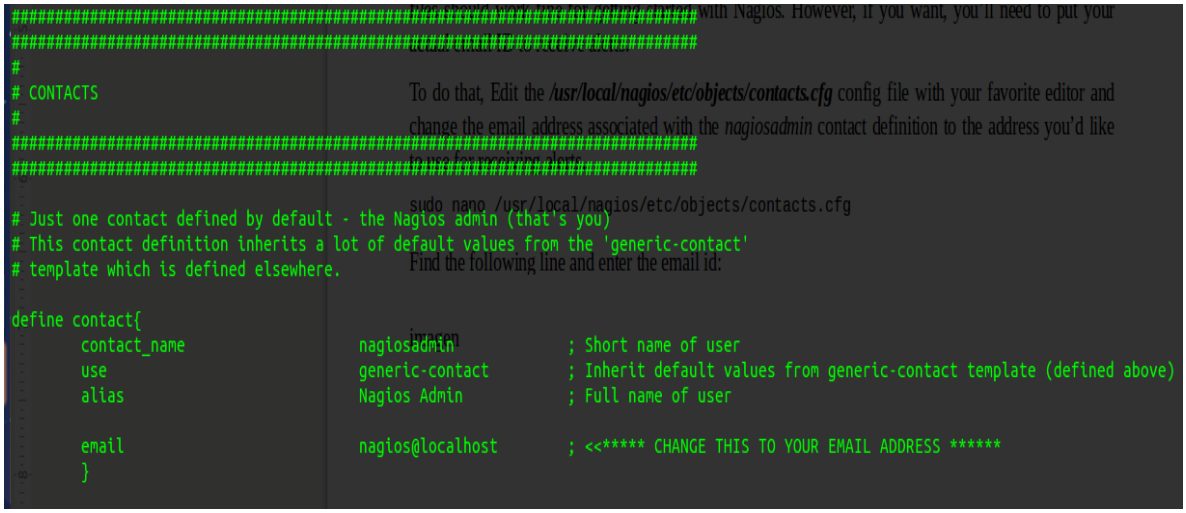

Figura 7. Cambio de correo electrónico en el archivo contacts.cfg.

A continuación, verificar que no existan errores luego de haber editado el archivo, ejecutando el siguiente comando:

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Por último, se reinicia Nagios.

sudo service nagios restart

Definir y probar servicios para monitoreo en Nagios

Los comandos que se pueden ejecutar para monitorear servicios se encuentran en la ruta */usr/local/nagios/libexec* y se pueden listar todos los scripts ejecutando el comando *ls*, para obtener ayuda sobre como probar uno de estos scripts basta con poner el nombre del script seguido de *-h* y para ejecutar algún script se utiliza *./* seguido del nombre del script, por ejemplo:

./check_dns -H www.google.com

Si se quiere definir un nuevo servicio para monitorear algún atributo en particular esta es la sintaxis que se debe seguir:

define service{
use uso
host_name nombreHost
service_description Descripción
check_command comando
}

Los servicios que se definan se incorporan en los archivos ubicados en */usr/local/nagios/etc/objects*, en cada archivo de configuración se define qué es lo que se quiere supervisar de cada uno de los dispositivos, ya sean servidores o dispositivos de interconexión. Dentro de estos archivos, ya están definidos servicios predeterminados, pero se pueden personalizar. A continuación, se muestran ejemplos de servicios para monitorear un servidor *Windows* y cuál es su funcionalidad:

Este servicio supervisa el uso de memoria en un servidor *Windows* y genera una alerta crítica si el uso de memoria es del 90% o más, o una alerta de advertencia si el uso de la memoria es de 80% o mayor.

```
define service {
   use             generic-service
   host_name       winserver
   service_description Memory Usage
   check_command      check_nt!MEMUSE!-w 80 -c 90
}
```

Ahora si se quiere supervisar el tiempo de actividad del servidor *Windows* se agrega la definición de servicio siguiente:

```
define service {
   use             generic-service
   host_name       winserver
   service_description Uptime
   check_command      check_nt!UPTIME
```

}

Proceso de configuración para monitoreo de un servidor *Windows* en Nagios Core

Para poder monitorear servicios de un servidor *Windows* se requiere de un agente instalado en la máquina a supervisar que se ejecute y haga el monitoreo (Nagios Enterprises., 2017). El agente se comunica por la red enviando la información del monitoreo al servidor Nagios y este procesa y emite las alertas o reportes necesarios del monitoreo.

Ahora se describen los pasos para instalar el agente *NSClient++* en el servidor *Windows* y el proceso de configuración en el servidor Nagios para habilitar la comunicación. La Figura 8 representa el esquema de comunicación entre *NSClient++* y Nagios Core.

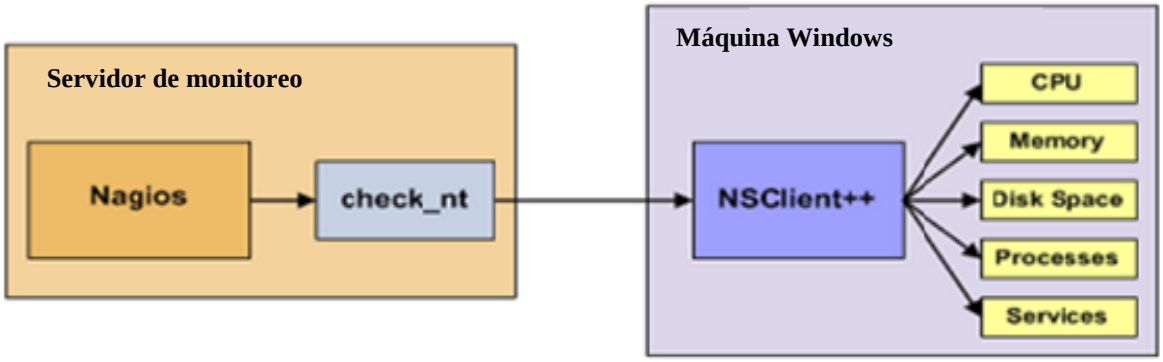

Figura 8. Comunicación del agente NSClient++ con el servidor Nagios, Nagios Enterprises (2017).

Para que sea simple de entender esta sección se subdivide en dos. La primera muestra la instalación del agente *NSClient++* en *Windows* y la segunda parte se centra en la configuración propia del servidor Nagios.

Instalación del agente *NSClient++* en *Windows*

El primer paso es obtener el complemento *NSClient++*, la última versión se puede obtener de la siguiente página *web*: http://sourceforge.net/projects/nscplus. Una vez descargado el agente se procede a instalarlo.

Al hacer doble click sobre el instalador aparece una ventana como la mostrada en la Figura 9.

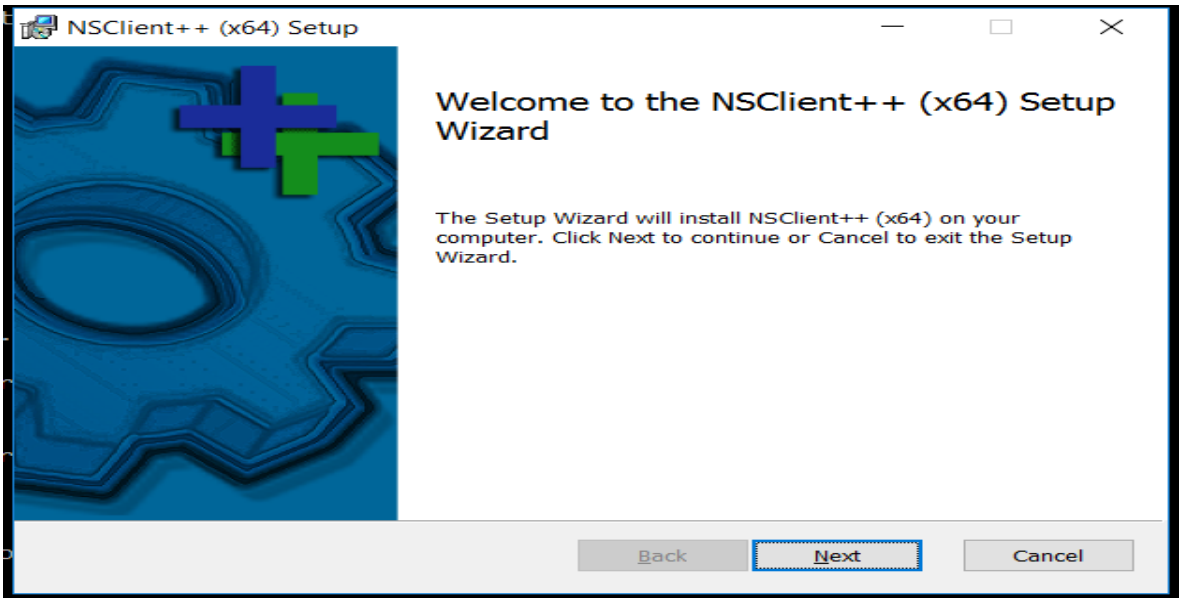

Figura 9. Ventana inicial durante la instalación de NSClient++ en Windows.

Se debe hacer click en el botón siguiente y aparece la ventana de la Figura 10 en la que se elige el tipo de instalación que se desea realizar. En este paso se eligió la instalación típica (*Typical*).

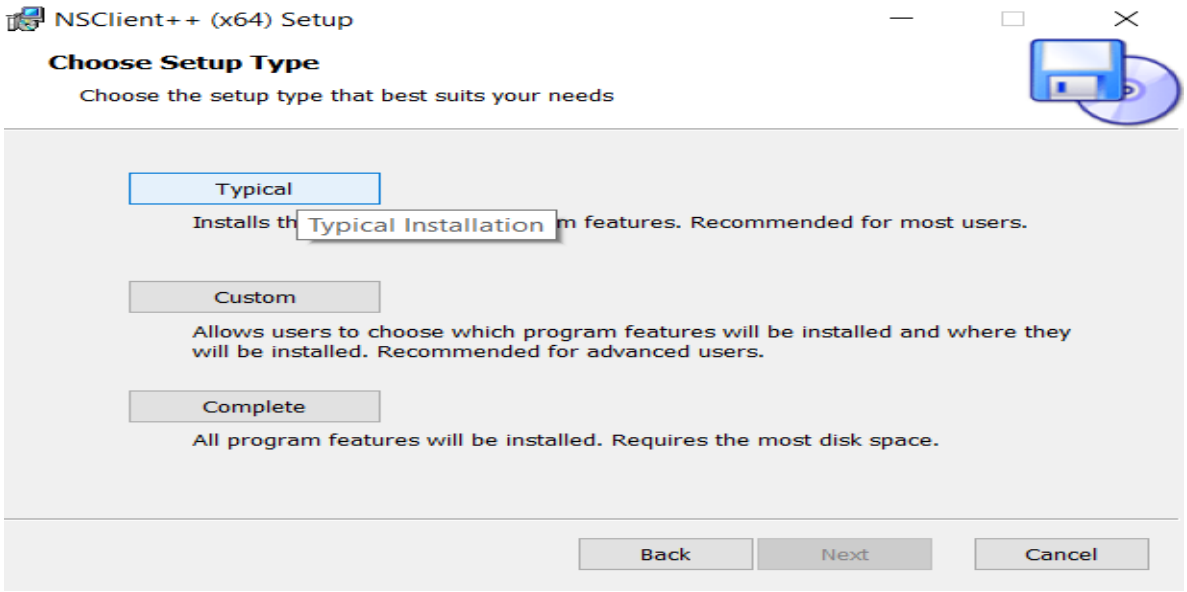

Figura 10. Elección del tipo de instalación de NSClient++.

Posteriormente se pone la ruta en la que se desea realizar la instalación y click en siguiente, ver la Figura 11.

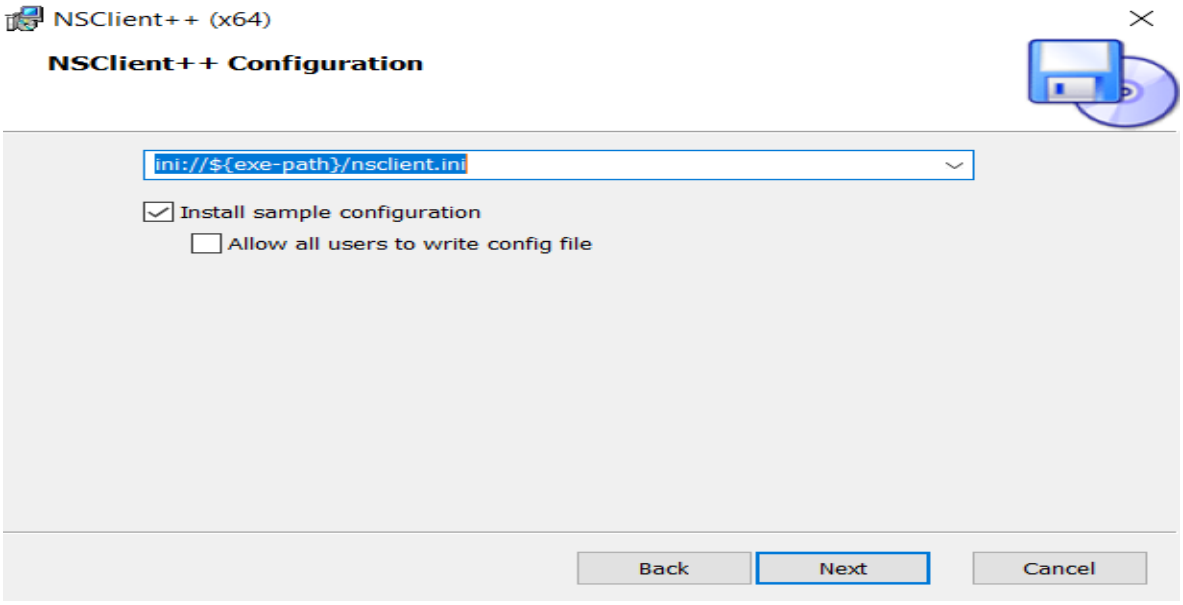

Figura 11. Ruta de instalación de NSClient++.

El siguiente paso es definir la dirección IP del servidor Nagios Core y la contraseña. Es importante recordar la contraseña ya que se requerirá en un paso posterior al momento de configurar los archivos en el servidor Nagios, en la Figura 12 se puede ver un ejemplo de configuración.

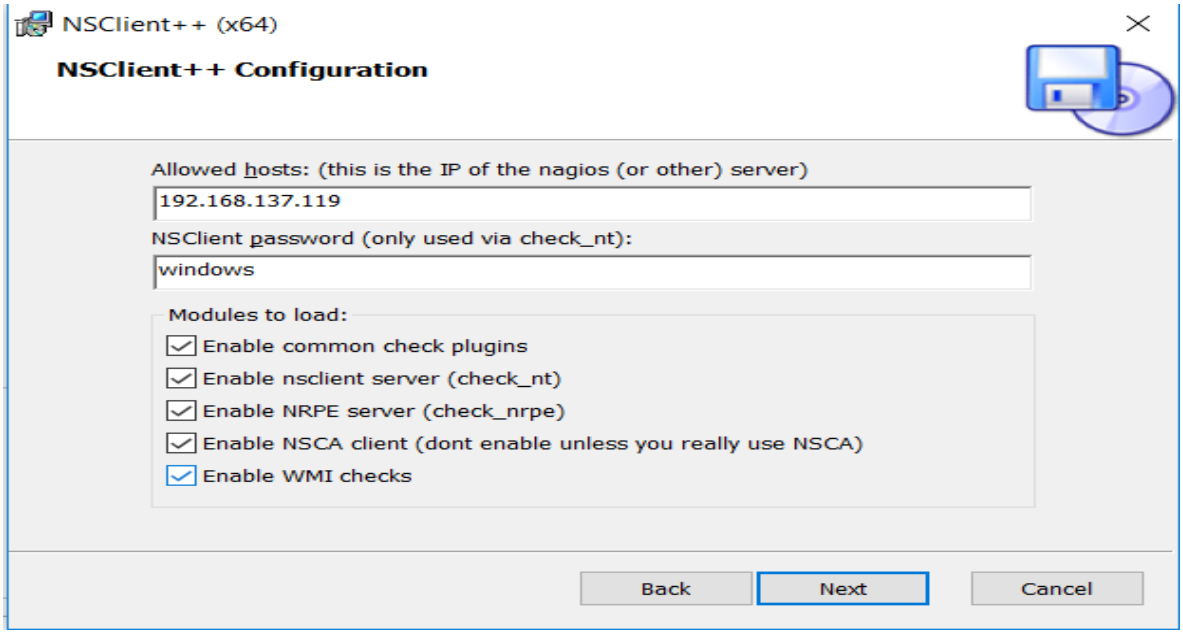

Figura 12. Definición de la dirección IP del servidor Nagios y la contraseña de acceso.

Realizada la configuración de *NSClient++* se puede iniciar la instalación dando click al botón *Install*, como se muestra en la Figura 13.

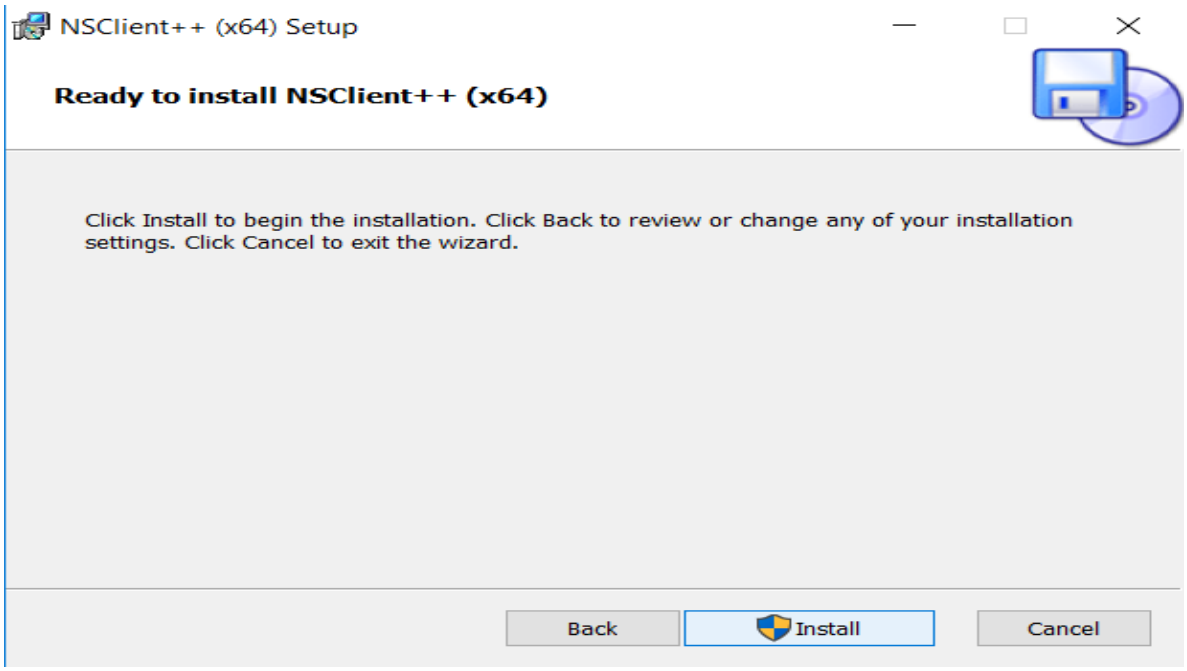

Figura 13. Inicio de la instalación de NSClient++.

Luego de unos minutos se termina la instalación y se da click en el botón Finalizar (ver Figura 14).

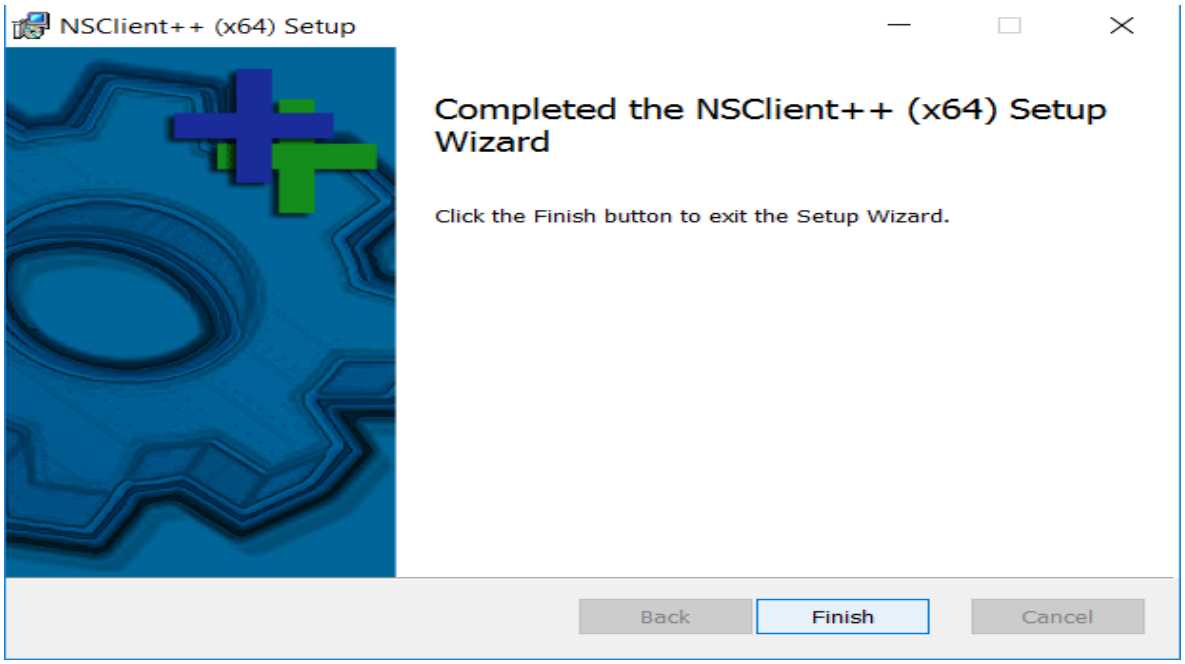

Figura 14. Finalización de la instalación de NSClient++.

Terminada la instalación del agente se procede a configurar el servidor Nagios Core. El proceso de configuración se explica a continuación.

Configuración en el servidor Nagios

Prerrequisitos

Primero se debe editar el archivo *nagios.cfg*, para esto se usa el comando siguiente:

nano /usr/local/nagios/etc/nagios.cfg

Se debe eliminar la marca de comentario (#) de la línea *#cfg_file=/usr/local/nagios/etc/objects/windows.cfg* como se muestra en la Figura 15.

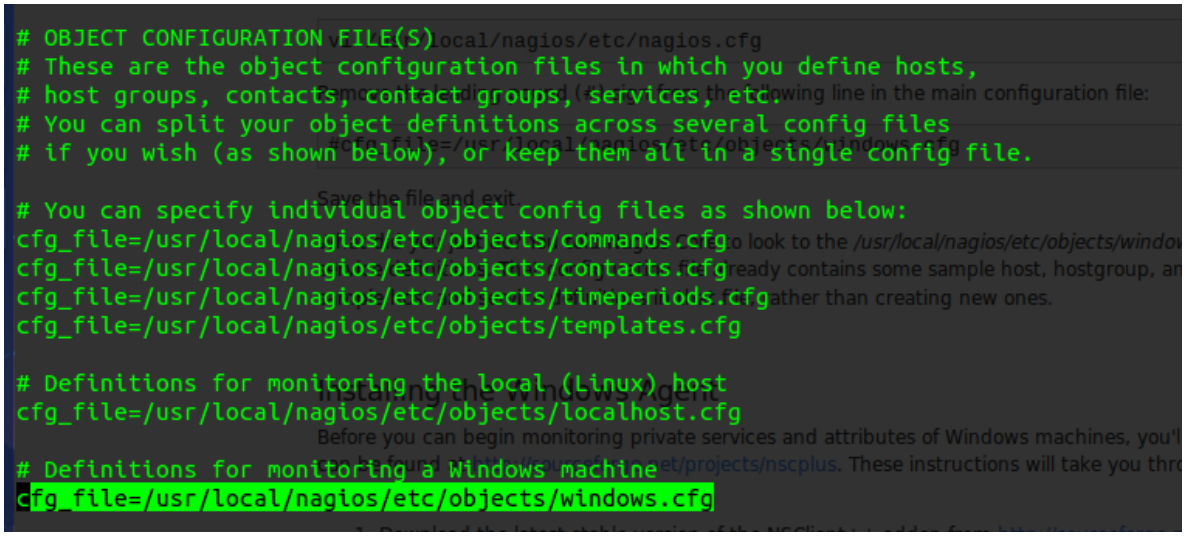

Figura 15. Se habilita el monitoreo de un servidor Windows en Nagios Core.

Por último, se guarda el archivo y se sale del modo de edición.

Configuración de Nagios Core

Para monitorear un servidor Windows hay que modificar algunos archivos a fin de especificar la dirección IP y algunas medidas de seguridad para establecer la comunicación entre el agente de monitoreo y el servidor Nagios.

El primer archivo por modificar es *windows.cfg* en el directorio *objects*, para esto se ejecuta el siguiente comando:

gedit /usr/local/nagios/etc/objects/windows.cfg

Se modifica la definición del *host* y se cambian los valores predeterminados como *hostname*, *alias* y la dirección IP a los valores del servidor que se desea monitorear, para más detalles observe la Figura 16.

```
# Define a host for the Windows machine we'll be monitoring
# Change the host_name, alias, and address to fit your situation

define host{
        use             windows-server  ; Inherit default values from a template
        host_name       winserver       ; The name we're giving to this host
        alias           My Windows Server       ; A longer name associated with the host
        address         192.168.137.1   ; IP address of the host
        }
```

Figura 16. Archivo de configuración windows.cfg.

A continuación, se agregan los servicios para monitorear los atributos que se deseen. Algunos servicios están escritos de manera predeterminada en este archivo, pero son flexibles en cuanto a la personalización y se pueden quitar o agregar servicios a decisión del administrador.

Protección de contraseña

Al momento de instalación de *NSClient++* en el servidor *Windows* se define una contraseña, esta debe ser especificada en el comando *check_nt* que está en el archivo *commands.cfg*. Para esto se ejecuta el siguiente comando:

nano /usr/local/nagios/etc/objects/commands.cfg

En *commands.cfg* se especifica el argumento *-s <contraseña>* especificando en *contraseña* la contraseña que se había establecido en el servidor *Windows* durante la instalación de *NSClient++*. En la Figura 17 se señala y observa con más detalle la línea que se debe cambiar. En este caso la contraseña definida fue ¨*windows*¨.

```
}

# 'check_nt' command definition
define command{
        command_name    check_nt
        command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s windows -v $ARG1$ $ARG2$
        }
```

Figura 17. Se define en check_nt la contraseña definida en el servidor Windows durante la instalación del agente NSClient++.

Una vez realizado el paso anterior se guarda y cierra el archivo y se ejecuta el comando siguiente para verificar si hubo algún error.

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Si no existe error, el paso final es reiniciar el servicio *nagios* con el comando siguiente:

sudo /etc/init.d/nagios restart

Aquí termina la configuración del servidor Nagios, al abrir el navegador *web* en la opción *Hosts* aparecerá ahora el equipo *Windows* que se acaba de agregar.

Proceso de configuración para monitoreo de un enrutador en Nagios Core

Para el monitoreo de un *switch* o de un *enrutador* utilizando Nagios Core es necesario que el dispositivo sea administrable. En este sentido se quiere decir que al menos tenga la posibilidad de configurarse una dirección IP, de otro modo el dispositivo de interconexión será transparente para Nagios Core. Si el dispositivo no soporta el protocolo SNMP (161) solo se podrá obtener información con el comando ping y datos como paquetes perdidos o RTA. Por otro lado, si el dispositivo soporta SNMP (161) entonces se puede obtener información como estatus de las interfaces, tiempo de encendido, entre muchas más, tantas como OID´s en la MIB. Todo el monitoreo del protocolo SNMP (161) se genera con el comando *check_snmp*. Nagios Core cuenta con un comando más llamado *check_mrtgtraf* que permite monitorear el ancho de banda de las interfaces usando la aplicación MRTG. En la Figura 18 se puede observar cómo se hace el monitoreo utilizando los comandos descritos arriba.

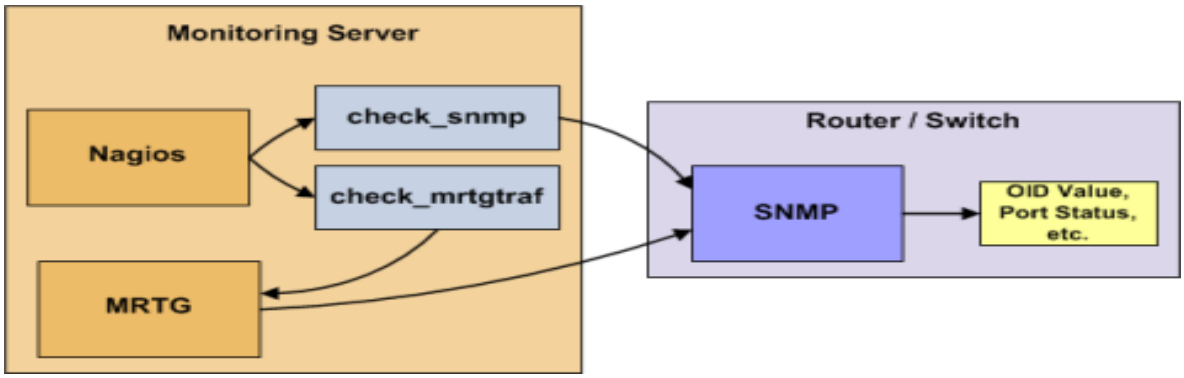

Figura 18. Monitoreo de enrutador o switch, Nagios Enterprises (2017).

Prerrequisitos

Para el monitoreo del primer dispositivo, ya sea un *switch* o un *enrutador* se tienen que editar los siguientes archivos.

Primero el archivo principal de configuración Nagios Core tecleando el siguiente comando:

nano /usr/local/nagios/etc/nagios.cfg

En este archivo se busca la línea *#cfg_file=/usr/local/nagios/etc/objects/switch.cfg* y se quita la marca de comentario (#). Posteriormente se guarda y cierra el archivo.

La Figura 19 muestra como se ve el archivo de configuración y la línea que se debe editar.

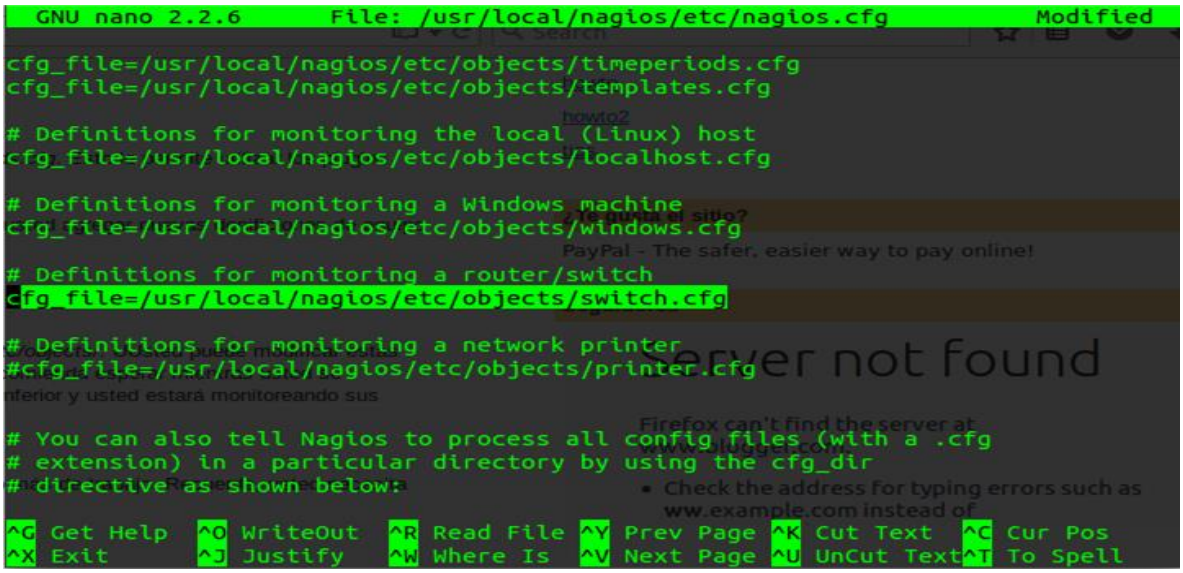

Figura 19. Edición del archivo nagios.cfg para habilitar el monitoreo de un switch o un enrutador con Nagios Core.

Configuración de Nagios Core

La siguiente tarea consiste en editar el archivo switch.cfg para crear la definición del dispositivo que se desea monitorear, una vez en este archivo se tiene que modificar la sección *define host* y especificar los datos del switch o enrutador tales como *host_name*, *alias* y sobre todo *address* en donde se escribe la dirección IP. A fin de lograr esta tarea se usa el siguiente comando:

nano /usr/local/nagios/etc/objects/switch.cfg

La Figura 20 deja ver el archivo y como queda configurado.

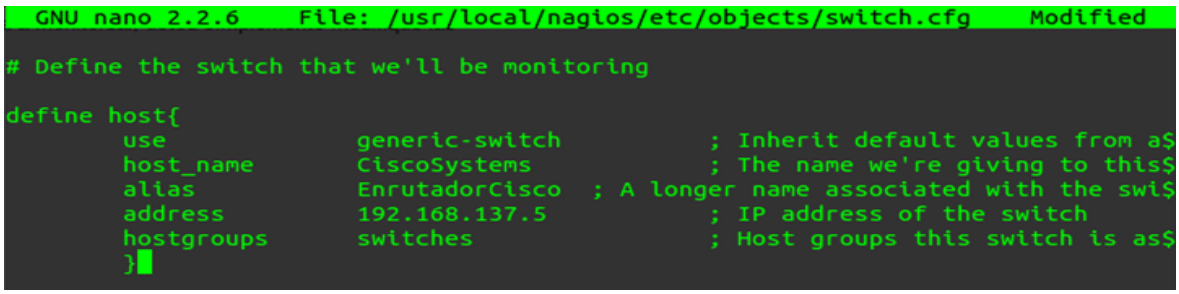

Figura 20. Definición de un dispositivo en el archivo switch.cfg.

Lo siguiente consiste en definir los servicios que se desean monitorear, la sintaxis de los servicios es la misma que para monitorear el resto de los dispositivos (servidores *Windows, Linux*, etc.) y esta configuración se realiza en el mismo archivo *switch.cfg* en la sección *service definitions*. El primer servicio que se define en este trabajo permite monitorear si existe comunicación con el dispositivo a través del comando *ping* que proporciona información de métricas como la cantidad de paquetes perdidos y RTA. La definición completa del servicio se muestra a continuación:

```
define service{
        use                     generic-service   ; Inherit values from a template
        host_name               CiscoSystems      ; The name of the host the service is associated with
        service_description     PING              ; The service description
        check_command           check_ping!200.0,20%!600.0,60% ; The command used to monitor the
service
        normal_check_interval   5                 ; Check the service every 5 minutes under normal
conditions
        retry_check_interval    1                 ; Re-check the service every minute until its final/hard
state is determined
        }
```

Lo que sigue es definir los servicios que permiten monitorear por medio del protocolo SNMP (161). Antes de iniciar con esta parte es importante revisar algunas cosas a fin de que Nagios Core realice sus funciones adecuadamente. Primero asegurarse de que el comando que monitorea SNMP *check_snmp* se encuentra en la dirección */usr/local/nagios/libexec*. Si no se encuentra es debido a que no se tenía instalado SNMP en el servidor y al momento de compilar los *pulgins* de Nagios, al no encontrar SNMP no se incluyó este comando. Para incluir el comando se debe realizar lo siguiente:

Primero se instala SNMP en el servidor mediante el comando:

apt-get install snmp

A continuación, se recompilan los *plugins* de Nagios tal y como se realizó durante el proceso de instalación de Nagios Core descrito en la *Sección 2.2 Instalación de los complementos de Nagios Core.*

Tras realizar los pasos anteriores, se revisa nuevamente la ruta */usr/local/nagios/libexec* y el comando *check_snmp* debe aparecer enlistado.

Por otro lado, se tiene que habilitar el uso de SNMP en el dispositivo a monitorear, en este caso un enrutador cisco, si el dispositivo que desea configurar es de una marca distinta se debe verificar en el manual de usuario correspondiente a dicho dispositivo como habilitar el protocolo SNMP. Los comandos son los siguientes:

En modo de configuración global:

#snmp-server community public RO

#snmp-server community public RW

La palabra *public* que indica el nombre de la comunidad, puede ser sustituida por cualquier cadena de caracteres. Es importante tenerla en mente porque es uno de los parámetros requeridos en la definición de servicio en Nagios. La entrada *RO* y *RW* hacen referencia al tipo de acceso *RO* implica solo lectura (*Read-Only*) y *RW* lectura-escritura (*Read-Write*).

En este punto, si los pasos para habilitar SNMP fueron correctos ya se puede realizar el monitoreo del dispositivo, sin embargo, antes de pasar a la configuración del servicio en Nagios Core se puede verificar que el protocolo SNMP se habilitó de manera adecuada ejecutando desde la consola del servidor Nagios Core el comando *check_snmp*. Para probar se ejecutan los siguientes comandos:

Primero se mueve al direcorio libexec.

cd /usr/local/nagios/libexec

Después se ejecuta *check_snmp.*

./check_snmp -H 192.168.137.5 -C public -o '.1.3.6.1.2.1.1.3.0' -v

Dónde: *-H* especifica la dirección IP del *host*, *-C* la comunidad, *-o* el OID de la MIB que se desea consultar, en este caso instancia de tiempo de inicio del sistema (*sysUpTimeInstance*).

Una vez que se realizaron las configuraciones anteriores se definen los servicios para cualquier OID que se quiera monitorear. A continuación, se muestra el código del servicio que monitorea el estado del puerto *fa0/1* en el archivo *switch.cfg*.

```
define service{
        use                     generic-service    ; Inherit values from a template
        host_name               CiscoSystems
        service_description     Port fa0/1 Link Status
        check_command           check_snmp!-C public -o '.1.3.6.1.2.1.2.2.1.8.2' -r 1 -m RFC1213-MIB
        }
```

Definidos los servicios para realizar el monitoreo se guarda y cierra el archivo *switch.cfg*.

Se ejecuta el comando siguiente para verificar que no existan errores en los archivos de Nagios que fueron editados:

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Por último, se reinicia Nagios.

sudo /etc/init.d/nagios restart

2.4 Funcionalidad

La herramienta Nagios Core permite realizar el monitoreo de numerosos servicios en todos los dispositivos que compongan la red. Todo está reunido en una interfaz gráfica que muestra el estado de los componentes y envía alertas, en este caso por correo electrónico, para avisar que algún servicio se degradó, se perdió por completo la comunicación o bien se restauró de alguna falla.

Cuando se inicia Nagios Core lo primero que muestra es una ventana para el inicio de sesión en la que se requiere un nombre de usuario y una contraseña, la Figura 21 muestra dicha ventana.

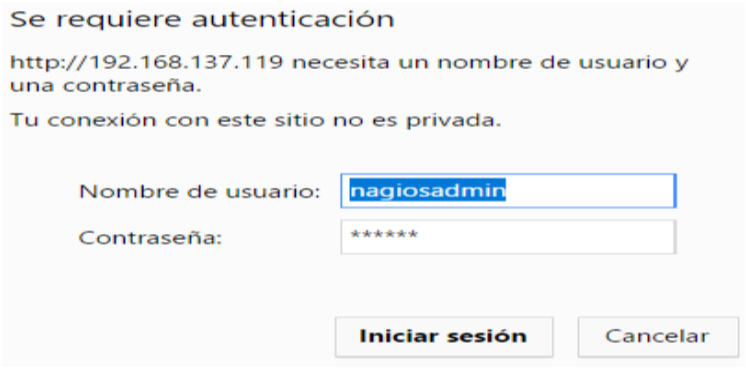

Figura 21. Ventana para autenticación de Nagios Core.

Una vez que se inicia sesión en la parte izquierda aparece el menú principal de la herramienta, el cual contiene cuatro elementos: *General*, *Current Status*, *Reports* y *System*.

General

Este menú contiene las opciones *Home* y *Documentation*. *Home* se utiliza para regresar a la página principal de Nagios, mientras que *Documentation* es una liga a toda la documentación técnica disponible de Nagios Core.

Current Status

Este menú contiene los elementos principales de la herramienta y muestra la información referente a los servicios y dispositivos en la red. Contiene opciones como *Tactical Overview*, en la que se muestra un resumen de la cantidad de dispositivos activos, dispositivos apagados y los inalcanzables. En cuanto a servicios muestra los que se encuentran en estado crítico, en advertencia, desconocido, los que están bien y los pendientes. Muestra también las características del monitoreo tales como detección de *Flap* y notificaciones, entre otros. En este menú se puede saber si dichas características están habilitadas o no, además de clasificar los dispositivos y servicios en una rápida visualización del monitoreo. Por último, como se muestra en la Figura 22, este menú incluye una gráfica que describe el estado de los servicios y dispositivos de la red.

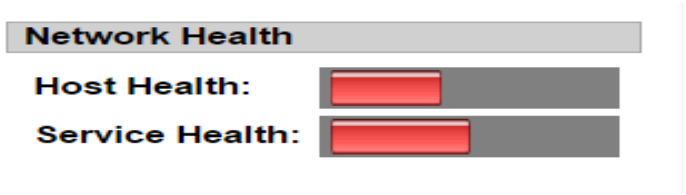

Figura 22. Gráfica del estado general de la red en Nagios Core.

La siguiente opción es *Map*, cuando se accede a ella se muestra un mapa con los dispositivos en la red monitoreados por Nagios, asignando color rojo para los nodos que no están activos y verde para los activos, además de mostrar el nombre de cada nodo. En la Figura 23 se observa dicho mapa.

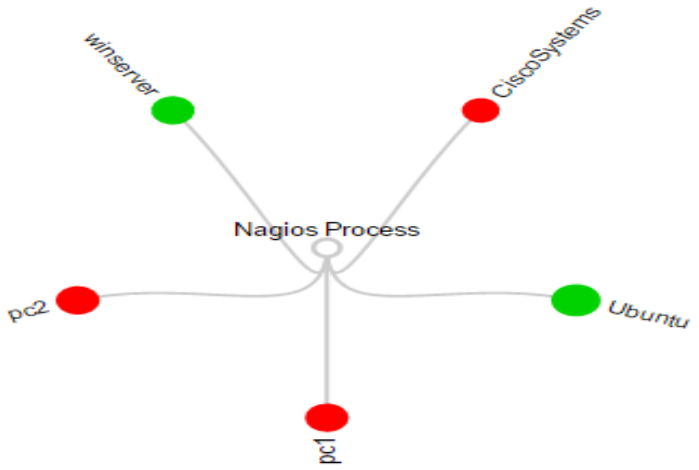

Figura 23. Mapa generado por Nagios Core que muestra los dispositivos monitoreados.

Si se coloca el cursor sobre algún nodo se puede obtener información más detallada como la dirección IP, su alias, los servicios y sus estados, cuando se realizó la última revisión, entre otros. El mapa permite el cambio de nodo raíz y contraer o expandir los nodos que se muestran en el mapa.

A continuación, se encuentra la opción *Hosts*, aquí se visualiza la lista de dispositivos agregados y muestra información como: el nombre del *host*, su estado, cuando se realizó la última revisión, la duración e información del estado. Dicha información puede ser ordenada por cualquiera de los criterios anteriores y se puede paginar de manera dinámica.

En la lista de los dispositivos agregados se puede dar click sobre el nombre del dispositivo para mostrar información detallada del estado del *host* como: *ping*, *RTA*, *packet loss*, tipo de revisión, última revisión, etc. En la parte derecha aparece un menú llamado *Host Commands* con diferentes opciones, en la Figura 24 se detallan estas opciones que van desde localizar el dispositivo en el mapa, enviar notificación personalizada al administrador de la red por correo electrónico hasta programar un chequeo de los servicios para *host*.

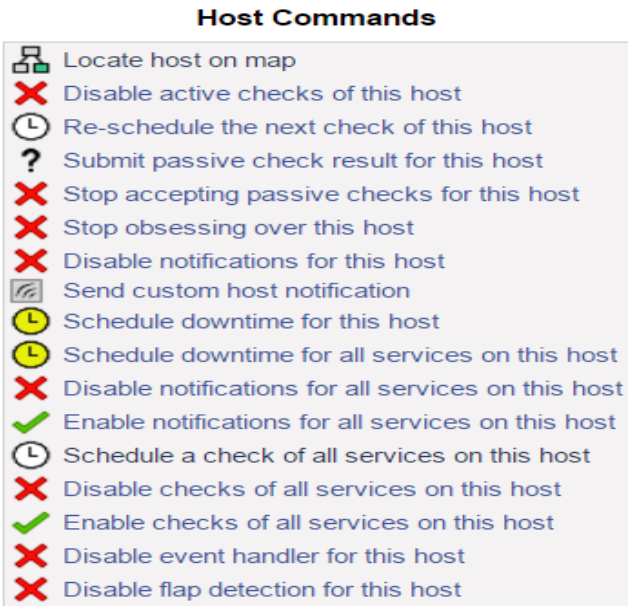

Figura 24. Opciones en el menú Host Commands.

Al lado del nombre del *Host* existe otra opción que permite mostrar detalles sobre los servicios monitoreados. Nuevamente si se da click sobre algún servicio se puede obtener información detallada del servicio y un menú *Service Commands* que muestra opciones similares a las de la Figura 24.

La siguiente opción en el menú *Current Status* es *Services* en donde se enlistan todos los servicios activos por *host*.

A continuación, aparecen las opciones *Host Groups* y *Service Groups*. Cada vez que se agrega un dispositivo o un servicio, se tiene la opción de incluirlo en un grupo para facilitar la administración. Estas opciones en el menú permiten visualizar dichos grupos (si es que se definieron), en los que de manera general se muestran los *hosts* o servicios que pertenecen a cierto grupo, el estado por componente del grupo, los servicios y acciones como información detallada o localización del *host* en el mapa. En la Figura 25 se puede ver un ejemplo con dos grupos. *Host Groups* y *Service Groups* cuentan con dos sub opciones cada uno: *Summary* y *Grid* que básicamente muestra vistas distintas de los grupos creados, por ejemplo, en *Grid* se muestran los servicios de manera individual y se puede obtener información detallada de cada uno eligiendo su nombre.

La última categoría en este menú es *Problems*. Aquí se encuentran opciones como *Services* y *Hosts*, y muestra únicamente los *hosts* o servicios que no están funcionando de manera adecuada.

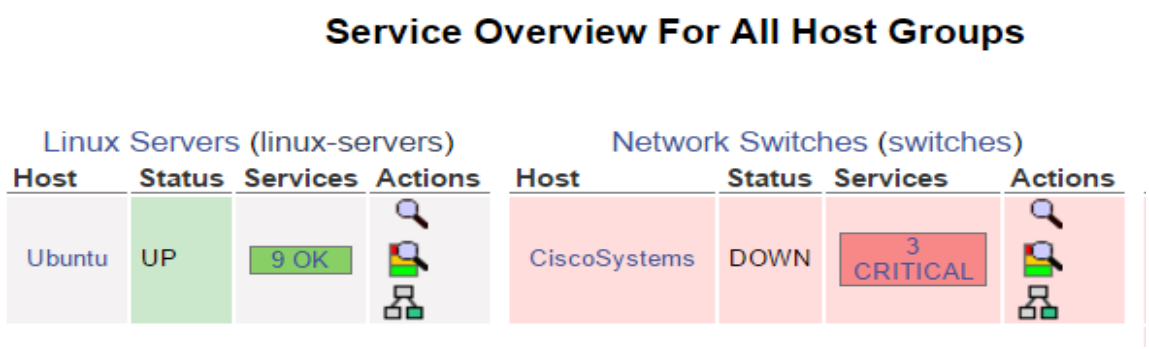

Figura 25. Vista de Host Groups en el menú Current Status.

Reports

El menú *Reports* permite crear una variedad de reportes ya sea por *host*, por servicio, por grupo de *hosts* o por grupo de servicios. Los reportes se pueden realizar de disponibilidad, tendencias y alertas y permiten definir los periodos de tiempo, si se asumen o no estados iniciales, los estados de retención, los estados en los que se programó inactividad entre varias opciones más. Además, una vez que los reportes se crearon se tiene la posibilidad de modificarlos y definir nuevos parámetros. En el caso de reportes de tendencias se muestran gráficos en los que se observan los estados por los que transitó un servicio o dispositivo en un periodo de tiempo determinado (Ver Figura 26). Para los reportes de disponibilidad se muestran tablas con porcentajes del estado del dispositivo y de los servicios (Ver la Figura 27).

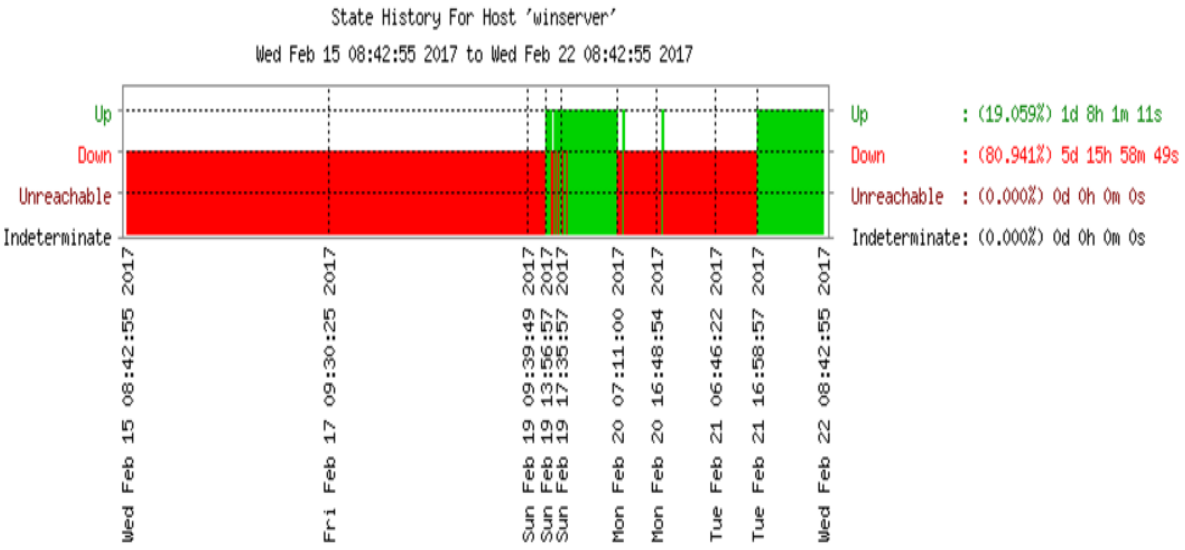

Figura 26. Reporte de tendencias para un host, correspondiente a un periodo de 7 días.

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	1d 23h 59m 12s	28.563%	40.383%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	1d 23h 59m 12s	28.563%	40.383%
DOWN	Unscheduled	2d 22h 50m 34s	42.168%	59.617%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	2d 22h 50m 34s	42.168%	59.617%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	2d 1h 10m 14s	29.268%	
	Total	2d 1h 10m 14s	29.268%	
All	Total	7d 0h 0m 0s	100.000%	100.000%

State Breakdowns For Host Services:

Service	% Time OK	% Time Warning	% Time Unknown	% Time Critical	% Time Undetermined
C:\ Drive Space	28.565% (40.400%)	0.000% (0.000%)	0.000% (0.000%)	42.141% (59.600%)	29.295%
CPU Load	28.465% (40.220%)	0.000% (0.000%)	0.000% (0.000%)	42.309% (59.780%)	29.225%
Explorer	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000% (100.000%)	0.000%
Memory Usage	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	70.746% (100.000%)	29.254%
NSClient++ Version	28.565% (40.385%)	0.000% (0.000%)	0.000% (0.000%)	42.167% (59.615%)	29.269%
Uptime	28.565% (40.393%)	0.000% (0.000%)	0.000% (0.000%)	42.152% (59.607%)	29.283%
Average	19.027% (26.899%)	0.000% (0.000%)	0.000% (0.000%)	56.586% (73.101%)	24.388%

Figura 27. Reporte de disponibilidad para un host correspondiente a un periodo de 7 días.

Finalmente, los reportes de alertas pueden visualizarse históricamente, en un resumen o un histograma. En la Figura 28 se puedo observar el comportamiento para un *host* en un histograma.

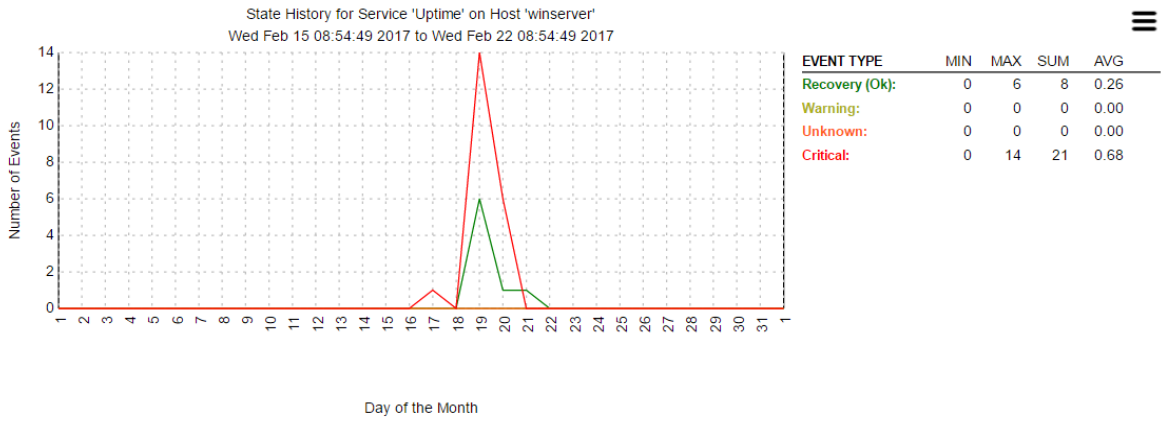

Figura 28. Reporte de alertas para un host en un periodo de 7 días.

Las últimas dos opciones para el menú *Reports* son *Notifications*, que muestra una tabla con el nombre del *host*, el servicio, el tipo de notificación, la hora a la que se generó, quien fue contactado, el comando para la notificación e información sobre el problema que se generó. La opción final, *Event Log* muestra una bitácora por días y horas de todo lo que pasó en Nagios Core: eventos generados, alertas, comandos ejecutados, etc.

System

System contiene opciones como *Downtime* en donde se muestran los *hosts* programados para mantenimiento, *Process Information* con información sobre el proceso Nagios con datos como: versión, cuando se inició el programa, *PID*, si las notificaciones están activadas o no, entre otras y comandos disponibles para el sistema; Por ejemplo, apagar Nagios Core, reiniciar, desactivar notificaciones, detener los chequeos a servicios, entre algunas otras.

Contiene también *Performance Info* que muestra tablas con información de rendimiento de todo el sistema y la última opción, *Configuration* que permite visualizar las configuraciones de los *hosts*, los servicios, los contactos y los comandos entre algunas otras.

Capítulo III. Análisis y Discusión

En este capítulo se analizan las características cubiertas por Nagios Core considerando las especificaciones mínimas propuestas por Lindros y Tittel (2015). De igual forma, se listan las fortalezas y debilidades de Nagios Core.

3.1 Características implementadas en Nagios Core

En el Capítulo 1 de este trabajo se mencionaron las características, que, según Lindros, K., y Tittel, E., (2015) son las mínimas que una herramienta para monitoreo de redes debe implementar. En la Tabla 5 se listan estas características y cuáles de ellas son implementadas o no por Nagios Core.

En la Tabla 5 se puede ver como Nagios Core cumple de manera efectiva con la facilidad para implementarse (implementarse en este sentido se refiere a instalarse), el monitoreo de dispositivos de múltiples proveedores, inventario de nodos y dispositivos, implementa alertas configurables, una interfaz *web* centralizada, fácil lectura de los gráficos, mapa de la topología de la red, comandos para modificar las configuraciones, un solucionador de problemas, monitoreo en tiempo real y detección del protocolo IP en las versiones 4 y 6. Por otro lado, se tiene que no cumple con la facilidad de configuración y tampoco posee autodescubrimiento de los dispositivos. Nagios Core implementa por tanto un 85% de las características mínimas propuestas por Lindros, K., y Tittel, E., (2015).

Tabla 5

Características mínimas implementadas por Nagios Core

Características	Implementadas por Nagios Core
Fácil de implementar	Si
Fácil de Configurar	No
Monitoreo de dispositivos de múltiples proveedores	Si
Autodescubrimiento	No
Inventario de nodos y dispositivos	Si
Alertas de advertencias y problemas configurables	Si
Interfaz *web*	Si
Facilidad de lectura de gráficos	Si
Mapa de la Topología de la red	Si
Comandos para modificar configuraciones	Si
Solucionador de problemas	Si
Detección de protocolo IPv4	Si
Detección de protocolo IPv6	Si
Monitoreo en tiempo real	Si

3.2 Fortalezas

Con base en la literatura consultada y luego de haber realizado la experimentación de Nagios Core se encontraron las siguientes fortalezas en la herramienta:

1. Permite el monitoreo de la red de datos y no solo se limita al monitoreo de los dispositivos, permite también el monitoreo de servicios y protocolos tales como SSH (22), POP (109, 110), HTTP (80), SMTP (25), entre otros (Nagios Enterprises, 2016).

2. Es sumamente flexible en cuanto a las configuraciones y definiciones de servicios, se pueden definir tantos servicios para monitoreo como se quiera desde monitoreo de atributos privados como tamaño de swap, espacio disponible en disco y uso de *CPU* hasta los protocolos que se mencionaron en el punto 1 (SSH (22), POP (109, 110), SMTP (25), HTTP (80), etc.).

3. Permite el monitoreo de servidores con versiones de sistema operativo *Linux*, *Unix* y *Windows*, así como de dispositivos de interconexión como *switches* o *enrutadores* de cualquier fabricante.

4. Monitoreo a través del protocolo SNMP (161) y monitoreo de los archivos de la aplicación MRTG (Nagios Enterprises, 2017) para determinar el tráfico en las interfaces de los dispositivos.

5. Es capaz de monitorear hasta 1000 dispositivos y 5000 servicios (Nagios Enterprises, 2014).

6. Las notificaciones disponibles en correo electrónico son totalmente personalizadas, se puede elegir el intervalo de notificación, el tipo de notificación ya sea por *host*, por servicio o personalizada, si se produjo una degradación que requiere atención, el servicio o *host* no responde o si se recuperó de algún fallo. Los correos electrónicos especifican el tipo de alerta, el nombre del *host* o servicio y la dirección IP de la que proviene la notificación.

7. Nagios Core cuenta con dos tipos de monitoreo: a) monitoreo activo: realizado por el demonio Nagios Core y b) monitoreo pasivo: realizado por una aplicación externa (Nagios Enterprises, 2017) ideal para efectuarse en ambientes con mayor cantidad de dispositivos.

8. Permite la creación de una gran cantidad de reportes con gráficos que facilitan su lectura, almacena una bitácora, por lo que se pueden hacer reportes históricos y además son flexibles y se pueden configurar varias opciones antes de ser emitidos.

9. Nagios Core está equipado con un solucionador de problemas llamado *Event Handler* que permite ejecutar comandos para tomar medidas cuando un problema ocurrió. *Event Handler* puede ejecutar desde tareas simples como reiniciar los servicios, apagarlos o verificarlos de nuevo hasta tareas complejas cuando se monitorean entornos redundantes y distribuidos (Nagios Enterprises, 2017).

10. La herramienta Nagios Core está licenciada bajo los términos de la Licencia Pública General GNU Versión 2 publicada por la Free Software Foundation. Lo cual permite copiar, distribuir y/o modificar a Nagios de manera legal bajo ciertas condiciones (Más información en: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/about.html#licensing).

11. Por último, Nagios Core cuenta con varias comunidades activas, documentación y soporte que incluye manuales de instalación, configuración y descarga de innumerables complementos que extienden la funcionalidad Nagios Core para realizar un monitoreo adecuado de su red.

3.3 Debilidades

Si bien Nagios Core presenta un desempeño satisfactorio y monitoreo adecuado de la red, existen también algunos inconvenientes:

1. En primer lugar, se encuentra que la configuración de Nagios Core es compleja y se requiere un profundo estudio de sus características antes de ponerse en marcha. La curva de aprendizaje una vez que Nagios Core se instaló es pronunciada y el tiempo en dominar la configuración puede variar dependiendo de los conocimientos y capacidades del administrador. En este caso tomó un mes para la puesta en marcha de Nagios Core configurando únicamente características básicas.

2. Nagios Core no puede ser instalada en sistemas operativos que no sean Linux (o variantes UNIX) lo que representa una desventaja en cuanto a su portabilidad.

3. La herramienta no permite agregar dispositivos o servicios en caliente, es decir, para cada nueva entrada o modificación en los archivos de configuración se tiene que reiniciar el servicio Nagios, esto podría resultar en un inconveniente importante en entornos en donde el monitoreo ininterrumpido es crítico.

4. Los archivos de Nagios Core son visibles. La seguridad de los archivos depende de la seguridad del sistema operativo. Por ejemplo, en el archivo *commands.cfg* la contraseña del cliente *NSClient++* esta en texto plano. Si bien la interfaz Nagios Core está protegida por usuario y contraseña, no así los archivos de configuración en donde se puede obtener toda la información como direcciones IP de los dispositivos, servicios que se monitorean, acceso al archivo de bitácora y acceso a la dirección de contacto para las notificaciones.

5. También debe notarse que la interfaz de Nagios Core es pobre y los gráficos no están detallados.

6. El idioma predeterminado de la herramienta es inglés y no es configurable a español o a alguno otro.

7. Por último, la herramienta carece de autodescubrimiento de los elementos de la red, si se quiere monitorear a un dispositivo este debe estar agregado en los archivos de configuración, de otro modo es invisible para la herramienta. Esta tarea de configuración sería tediosa para monitorear una red compuesta de varios elementos.

Conclusiones

En esta tesina, se ha analizado la herramienta para el monitoreo de redes de datos Nagios Core mediante su instalación y configuración que permitió revisar su funcionalidad y discutir sus fortalezas y debilidades; con lo cual se cumple el objetivo general de este trabajo. Más aún del análisis realizado, se puede decir lo siguiente:

En la elección de una herramienta para el monitoreo de redes de datos se debe considerar principalmente una opción que reúna las necesidades de monitoreo que requiera su red y que proporcione escalabilidad para el futuro. Se debe tomar en consideración que las opciones a las que se puede acceder incluyen herramientas de código abierto, así como opciones comerciales. Si se habla propiamente del monitoreo, en trabajos como el de Fatema et al. (2014), se explica que el monitoreo de la nube es la tendencia futura y aunque las herramientas tradicionales para monitoreo de infraestructura se están adaptando para estos fines no implementan la funcionalidad y monitoreo de todos los recursos importantes para la nube.

En cuanto a la herramienta que se discutió en este trabajo, luego de revisar las fortalezas y debilidades y con base en la Tabla 5 se puede identificar a Nagios Core como una herramienta para el monitoreo de redes de datos que cumple de manera efectiva con un 85% de las características mínimas planteadas por Lindros y Tittel (2015). Se encontró que aparte de estas características, Nagios Core es adecuada para desplegarse en ambientes con un gran número de dispositivos a monitorear y es sumamente flexible en cuanto a la definición de servicios, equipos a monitorear y personalización. Nagios Core es además una herramienta de código abierto por lo que el costo de la adquisición no representa gastos en absoluto, sin embargo, la inversión necesaria esta en capacitar al personal para que opere de manera adecuada la herramienta. Debido a la naturaleza de Nagios Core (al ser esta de código abierto) y tomando en cuenta que existe una versión comercial (Nagios XI), debe considerarse la posibilidad de que en algún momento *Nagios Enterprises* decida quitar Nagios Core, por supuesto esto implica que el mantenimiento y soporte para la herramienta desaparezcan y los usuarios se vean obligados a elegir Nagios XI o bien buscar otra alternativa.

Nagios Core tiene ciertas limitaciones como la falta de autodescubrimiento que sin duda es una característica deseable pero que si se contrasta con el resto de las ventajas ofrecidas al final del día es una opción atractiva, sin contar que en la gran comunidad activa se desarrollan complementos para resolver carencias como esta. Un ejemplo es la comunidad *Nagios Exchange* en donde está disponible un complemento para el autodescubrimiento. Aunque su implementación requiere cambios en el entorno se presume que con las modificaciones adecuados el autodescubrimiento en Nagios Core es una realidad.

Glosario

CGI *Common Gateway Interface* es una interfaz de intercambio de datos estándar en www a través del cual se organiza el envío y recepción de datos entre navegador y programas residentes en servidores www. (González, R., 2010).

FCAPS Es un modelo de administración de redes de datos propuesto por la *ISO*. Cada letra en FCAPS simboliza una capa para la administración: *Fault*, *Configuration*, *Accounting*, *Performance* y *Security*

ISO La Organización Internacional para la Estandarización, es una organización sin fines de lucro que desarrolla y publica estándares de prácticamente todos los tipos posibles, desde estándares para la tecnología de la información hasta la dinámica de fluidos y la energía nuclear (American National Standards Institute., 2017).

Jitter También conocido como fluctuación de fase, se define como una variación en el retardo de los paquetes recibidos (Cisco Systems., 2005)

KPI Los Indicadores de Desempeño Clave (*Key Performance Indicators*), se define como un conjunto de métricas establecidas para cuantificar aspectos específicos de una red en funcionamiento (Brown, D., 2016).

MIB La Base de Información para Gestión (*Management Information Base*) es un tipo de base de datos que contiene información jerárquica, estructurada en forma de árbol, de todos los dispositivos gestionados en una red de comunicaciones. Es un componente fundamental en funcionamiento del protocolo SNMP (Cisco., 2007).

MRTG Por sus siglas *Multi Router Traffic Grapher*, es una herramienta para supervisar la carga de tráfico en los enlaces de red creada por Tobias Oetiker.

NRPE Ejecutor Remoto de Complementos Nagios o *Nagios Remote Plugin Executor*, es un complemento diseñado para permitir ejecutar complementos Nagios en máquinas Linux / Unix remotas (Galstad, E., 2016).

NSClient++ Es un agente que permite el monitoreo para sistemas Windows y funciona con Nagios.

OID En el contexto SNMP es un Identificador de Objeto u *Object Identifier*, es una dirección usada para identificar dispositivos y sus estados en una MIB (Cisco., 2007).

QoS *Quality of service* o Calidad del servicio, se define como la totalidad de las características de un servicio de telecomunicaciones que determinan su capacidad para satisfacer las necesidades explícitas e implícitas del usuario del servicio (International Telecommunication Union., 2009)

RTA	*Round Trip Average* o Tiempo Promedio de Ida y Vuelta, en el informe ping, el tiempo requerido por un paquete para completar un viaje entre el servidor de origen y el servidor de destino (Abbreviations., 2017).
SLA	Acuerdo de Nivel de Servicio (*Service Level Agreement*) es un documento que se firma entre la empresa que ofrece un servicio y su cliente en el que se especifican parámetros de calidad y las sanciones que conlleva el incumplimiento de alguno de estos.
SNMP	Protocolo Simple de Administración de Red (*Simple Network Management Protocol*) es un protocolo de capa de aplicación diseñado para facilitar el intercambio de información de gestión entre dispositivos de red (Cisco., 2007).
TMN	Red de Gestión de las Telecomunicaciones (*Telecommunications Management Network*) es un modelo de referencia que cubre un amplio rango de temas relacionados con los principios de cómo gestionar las redes de telecomunicaciones (Rouse, M., 2007).

Referencias

Abbreviations. (2017). What does RTA mean in Networking?. 02/07/2017, de abbreviations Sitio web: http://www.abbreviations.com/term/1448881

Aceto, G. et al. (2013). Survey Cloud monitoring: A survey. Computer Networks: The International Journal of Computer and Telecommunications Networking, 57, 2093-2115. 28/02/2017, De The ACM Digital Library.

American National Standards Institute. (2017). International Organization for Standardization. 01/07/2017, de American National Standards Institute (ANSI). Sitio web: http://webstore.ansi.org/SdoInfo.aspx?sdoid=39

Ardagna et al. (2014). Quality-of-service in cloud computing: modeling techniques and their applications. 01/05/2017, de Springer Open Sitio web: https://jisajournal.springeropen.com/articles/10.1186/s13174-014-0011-3.

Bala, A., & Chana, I. (2012). Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing. IJCSI International Journal of Computer Science, 9, 288-293.

Brown, D. (2016). The Role of KPIs in Network Management. 02/07/2017, de Techopedia Sitio web: https://www.techopedia.com/2/31633/networks/the-role-of-kpis-in-network-management

Buytaert, J. et al. (2008), Systems monitoring shootout, p. 53 de: Linux Symposium.

Celesti, A. et al. (2010). How to Enhance Cloud Architectures to Enable Cross-Federation. Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, 337-345. De IEEE.

Cheng, X. et al. (2009). A multi-tenant oriented performance monitoring, detecting and scheduling architecture based on SLA. Pervasive Computing (JCPC), 2009 Joint Conferences on, 599-604. De IEEE.

Cirstoiu, C. et al. (2007). Monitoring, accounting and automated decision support for the alice experiment based on the MonALISA framework. 16th International Symposium on High Performance Distributed Computing 2007, 39-44. De Elsevier B.V.

Cisco Systems. (2005). Jitter. 02/07/2017, de Cisco Networking Academy Program Sitio web: http://www.hh.se/download/18.70cf2e49129168da015800094781/1341267715838/7_6_Jitter.pdf

Cisco. (2007). MIB. 02/07/2017, de Cisco Sitio web: http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/pgw/7/mibs/guide/7MIB_Ch1.html

Claise, B., & Wolter, R. (2007). Architectural and Framework Standards: The TMN/FCAPS Model (ITU-T). 23/04/2017, de eTutorials.org Sitio web: http://etutorials.org/Networking/network+management/Part+I+Data+Collection+and+Methodology+Standards/Chapter+3.+Accounting+and+Performance+Standards+and+Definitions/Architectural+and+Framework+Standards+The+TMN+FCAPS+Model+ITU-T/

Clancy, H. (2010). Open source network monitoring: SLAs push network pros to new tools. 28/02/2017, de TechTarget Sitio web: http://searchnetworking.techtarget.com/Open-source-network-monitoring-SLAs-push-network-pros-to-new-tools.

Comuzzi, M. et. al. (2009). Establishing and Monitoring SLAs in Complex Service Based Systems. Web Services, 2009. ICWS 2009. IEEE International Conference on, 2007, 783-790. De IEEE.

Cowie, B. (2012). Building A Better Network Monitoring System.

Elmroth, E. et al. (2009). Accounting and Billing for Federated Cloud Infrastructures. Grid and Cooperative Computing, 2009. GCC '09. Eighth International Conference on, 268-275. De IEEE.

Emeakaroha, V. et al. (2010). Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. High Performance Computing and Simulation (HPCS), 2010 International Conference on, 48-54. IEEE.

Engel, F et al. (2000). Network monitoring. 28/02/2017, de IFI CLAIMS Patent Services Sitio web: https://www.google.com/patents/US6115393.

Fatema, K. et al. / J. Parallel Distrib. Comput. 74 (2014) 2918–2933.

Ferrer, A. et al. (2012). OPTIMIS: A holistic approach to cloud service provisioning. Future Generation Computer Systems, 28, 66-77. De Elsevier B.V.

Ferretti, S. et al. (2010). QoS–Aware Clouds. Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, 321-328. De IEEE.

Frank, M. et al. (2009). A space-efficient quantum computer simulator suitable for high-speed FPGA implementation. Conference, De SPIE Base de datos.

Free Software Foundation. (2017). ¿Qué es el software libre?. 18/06/2017, de Free Software Foundation, Inc Sitio web: https://www.gnu.org/philosophy/free-sw.es.html

Frey, S. et al. (2013). Key Performance Indicators for Cloud Computing SLAs. The Fifth International Conference on Emerging Network Intelligence, 1-5. 25/04/2017, Chinacloud.

Galstad, E. (2016). NRPE Documentation. 01/02/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf

Gartner. (2008). Open source survey. De Gartner, Inc Sitio web: http://www.gartner.com/technology/home.jsp

Gerard, P. (2013). Monitoring network devices with nagios. 20/01/2017, de paulgporter.net Sitio web: https://paulgporter.net/2013/01/30/network-monitoring-nagios/

Gogouvitis, S. et al. (2013). A Monitoring Mechanism for Storage Clouds. Cloud and Green Computing (CGC), 2012 Second International Conference on, 153-159. De IEEE.

Gómez, J. (2010). Instalar Nagios en Ubuntu. 20/01/2017, de Nosolounix.com Sitio web: http://www.nosolounix.com/2010/04/instalar-nagios-en-ubuntu.html

González, R. (2010). Programación Web. 01/07/2017, de Blogger Sitio web: http://ramon-esteban.blogspot.mx/2010/11/cgi-common-gateway-interface.html

Haiteng, Z. et. al. (2012). Establishing Service Level Agreement Requirement Based on Monitoring. Cloud and Green Computing (CGC), 2012 Second International Conference on, 472-476. 2017, De IEEE.

Hasselmeyer, P. & Heureuse, N. (2010). Towards holistic multi-tenant monitoring for virtual data centers. Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, 350-356. De IEEE.

Hernantes, J. et al. (2015). IT Infrastructure- Monitoring Tools. Computing in science and engineering, 33, 94-100. 01/03/2017. IEEE software.

Hyperic (Febrero, 2014). URL http://www.hyperic.com

IBM Tivoli monitoring (Febrero, 2014). URL https://publib.boulder.ibm.com/

Iiang, D. et al. Developing an active mode of network management system with intelligent multi-agent techniques, in: Pervasive Computing (JCPC), 2009 Joint Conferences on, 2009, pp. 77–82.

International Telecommunication Union. (2009). Definitions of terms related to quality of service. Geneva, Switzerland: ITU-T E.800.

Jennings, N. R., & Wooldridge, M. (1998). Applications of intelligent agents. In Agent technology (pp. 3-28). Springer Berlin Heidelberg.

Jhawar, R. et al. (2013). Fault Tolerance Management in Cloud Computing: A System-Level Perspective. IEEE Systems Journal, 7, 288 - 297. De IEEE.

Jiang,H., et al.(2010) A performance monitoring solution for distributed application system based on JMX, in: Grid and Cooperative Computing (GCC), 2010 9th International Conference on, pp. 124 – 127.

King, B. (2013). MTE Explains: What Is The Difference Between Free Software, Open Source Software, and Freeware?. 18/06/2017, de Make Tech Easier Sitio web: https://www.maketecheasier.com/free-software-vs-open-source-vs-freeware/

Kornaros, G., & Pnevmatikatos, D. (2013). A survey and taxonomy of on-chip monitoring of multicore systems-on-chip. ACM Transactions on Design Automation of Electronic Systems, 18. 28/02/2017, The ACM Digital Library.

Krizanic, J. et al. Load testing and performance monitoring tools in use with AJAX based web applications, in: MIPRO, 2010 Proceedings of the 33rd International Convention, IEEE, 2010, pp. 428–434.

Krutz, R., & Dean, R. (2010). Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Cloud Security: A Comprehensive Guide to Secure Cloud Computing. De The ACM Digital Library.

Kumar, A. et al. (2012). Secure storage and access of data in cloud computing. ICT Convergence (ICTC), 2012 International Conference on. 04/05/2017, De IEEE.

Lindros, K., & Tittel, E. (2015). What to Look for in Network Monitoring Tools. 10/20/2016, de TechTarget Sitio web: SearchNetworking.com

López, M., et al. (2014). Gestión de riesgos para la adopción de la computación en nube en entidades financieras de la República Argentina. Contaduría y Administración, 59, 5-10. 01/05/2017, De Science Direct Base de datos.

Lord N. (2016). What is Cloud Security Monitoring?. 01/05/2017, de Digital Guardian Sitio web: https://digitalguardian.com/blog/what-cloud-security-monitoring.

Mansouri-Samani, M. & Sloman, M. (2002). Monitoring distributed systems. IEEE Network, 7, 20 - 30. 28/02/2017, De IEEE Communications Society Base de datos.

Massie, M. et al. The Ganglia distributed monitoring system: design, implementation, and experience, Parallel Comput. 30 (7) (2004) 817–840.

Mauro, D., & Schmidt, K. (2001). Essential SNMP. Estados Unidos de America: O'REILLY.

Meng, X. et al. (2010). Efficient resource provisioning in compute clouds via VM multiplexing. ICAC '10 Proceedings of the 7th international conference on Autonomic computing, 11-20. The ACM Digital Library.

Morgan, D. et al. (1975). A computer network monitoring system. IEEE Transactions on Software Engineering, SE-1, 299- 311. De IEEE.

Muddana, L & Aluvalu, R. (2015). A Survey on Access Control Models in Cloud Computing Authors Authors and affiliations. Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI), 1, 653-664. 01/05/2017, De Springer Link.

Nagios Enterprises. (2014). Nagios XI - Hardware Requirements. 20/02/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagiosxi/docs/Nagios-XI-Hardware-Requirements.pdf

Nagios Enterprises. (2016). Nagios – About Nagios Core. 05/05/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/about.html#requirements

Nagios Enterprises. (2016). Nagios – Installing Nagios Core From Source. 20/01/2016, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/Installing_Nagios_Core_From_Source.pdf

Nagios Enterprises. (2016). Protocol Monitoring With Nagios. 09/03/2017, de Nagios Sitio web: https://www.nagios.com/solutions/protocol-monitoring/

Nagios Enterprises. (2017). Monitoring Routers and Switches. 01/02/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-routers.html

Nagios Enterprises. (2017). Monitoring Windows Machines. 31/01/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/monitoring-windows.html

Nagios Enterprises. (2017). Nagios Core (user manual). 09/03/2017, de Nagios Sitio web: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/toc.html

Nagios Enterprises. (2017). Nagios XI Enterprise Server and Network Monitoring Software. 07/05/2017, de Nagios Sitio web: https://www.nagios.com/products/nagios-xi/#systemreqs

Naik, V. et al. (2014). Service Usage Metering in Hybrid Cloud Environments. Cloud Engineering (IC2E), 2014 IEEE International Conference on. 25/04/2017, De IEEE.

Oetiker, T. (2012). What is MRTG?. 02/07/2017, de MRTG Sitio web: http://oss.oetiker.ch/mrtg/doc/mrtg.en.html

Olups, R. (2010). Zabbix 1.8 Network Monitoring. Birmingham, UK: Packt Publishing.

Palacios, M. et al. (2012). Identifying Test Requirements by Analyzing SLA Guarantee Terms. Web Services (ICWS), 2012 IEEE 19th International Conference on, 351-358. De IEEE.

Pape, C. & Trommer, R. (2012). Monitoring VMware-based virtual infrastructures with OpenNMS.

Park, K. et al. (2013). THEMIS: A Mutually Verifiable Billing System for the Cloud Computing Environment. IEEE Transactions on Services Computing, 6, 300 - 313. De IEEE

Pueschel, T. & Neumann, D. (2009). Management of cloud infrastructures: Policy-based revenue optimization, in: Thirtieth International Conference on Information Systems (ICIS 2009), 1–16.

Recomendación ITU-T E.800. (2008) Series e: overall network operation, telephone service, service operation and human factors.

Rellermeyer, J. et al. (2007). Building, deploying, and monitoring distributed applications with Eclipse and R-OSGI. eclipse '07 Proceedings of the 2007 OOPSLA workshop on eclipse technology eXchange, 50-54. ACM Digital Library.

Rizos, C. (2013). What is the TMN Model?. 23/04/2017, de SNMP Center Sitio web: https://www.snmpcenter.com/what-is-the-tmn-model/

Rouse, M. (2007). FCAPS (fault-management, configuration, accounting, performance, and security). 23/04/2017, de TechTarget Sitio web: http://searchnetworking.techtarget.com/definition/FCAPS

Rouse, M. (2016). configuration drift. 01/05/2017, de Tech Target Sitio web: http://searchwindowsserver.techtarget.com/definition/configuration-drift.

Sekar, V. & Maniatis, P. (2011). Verifiable resource accounting for cloud computing services. CCSW '11 Proceedings of the 3rd ACM workshop on Cloud computing security workshop, 21-26. The ECM Digital Library.

Sekiguchi, A. et al. (2012). Configuration management technology using tree structures of ICT systems. Proceedings of the 15th Communications and Networking Simulation Symposium, Society for Computer Simulation International, p. 4.

Stallings, W. (2004). Comunicaciones y Redes de Computadoras. Madrid: Pearson Educación.

Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34, 1-11. De ScienceDirect.

Tanembaum, A. (2003). Computer Networks. New Jersey: Prentice Hall.

The Committee on Communications Policy Institute of Electrical and Electronics Engineers - United States of America. (2010). network traffic management and the evolving internet. Washington, D.C: IEEE-USA.

Thomas, C. (2007). What network monitoring tools monitor all OSI layers?. 07/01/2017, de TechTarget Sitio web: http://searchnetworking.techtarget.com/answer/What-network-monitoring-tools-monitor-all-OSI-layers

Tovarnak, D, & Pitner, T . (2012). Towards Multi-tenant and Interoperable Monitoring of Virtual Machines in Cloud. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 436-442. De The ACM Digital Library.

Uretsky, M. (2013). Practical differences in between Nagios and Hyperic HQ?. 15/02/2017, de Quora Sitio web: https://www.quora.com/Practical-differences-in-between-Nagios-and-Hyperic-HQ.

Vaquero, L. et al. (2011). Locking the sky: a survey on IaaS cloud security. Computing, 91, 93-118. De Springer Link.

Vicente, C. (2005). Monitoreo de recursos de red. 14/03/2017, de UNAM Sitio web: https://juliorestrepo.files.wordpress.com/2011/04/monitoreo.pdf

Zanikolas, S., & Sakellariou, R. (2005). A taxonomy of grid monitoring systems. Future Generation Computer Systems, 21, 163-188. 28/02/2017, De The ACM Digital Library.

Zhang, Q. et al. (2007). A Regression-Based Analytic Model for Dynamic Resource Provisioning of Multi Tier Applications. Autonomic Computing, 2007. ICAC '07. Fourth International Conference on. De IEEE.