Universidad de Alcalá
Escuela Politécnica Superior

INGENIERÍA DE SISTEMAS DE INFORMACIÓN

Trabajo Fin de Grado

Diseño e implementación de sistema de monitorización de máquinas virtuales

Autor: Juan José Anchuelo Rivillo

Tutor/es: Óscar García Población

2021

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Dpto. de Automática

GRADO EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN

Trabajo Fin de Grado

DISEÑO E IMPLEMENTACIÓN DE SISTEMA DE MONITORIZACIÓN DE MÁQUINAS VIRTUALES

AUTOR: Juan José Anchuelo Rivillo

TUTOR: Óscar García Población

Tribunal:

Presidente: Pablo Parra Espada

Vocal 1º: Elena Campo Montalvo

Tutor/Director: Óscar García Población

Suplente: Óscar Rodríguez Polo

Fecha: 16 de enero de 2021

Agradecimientos

Quiero comenzar agradeciendo a Óscar García Población, tutor de este proyecto de Fin de Grado, el tiempo dedicado durante este año a que este documento cogiera forma y plasmara todo el trabajo que he realizado. Sus indicaciones y propuestas han hecho que haya conseguido realizar este documento del cual esto muy satisfecho de haber escrito.

También debo mencionar a mi padre Juan José, mi madre Mª de los Ángeles y mi hermano Ángel Luis, que tenían tantas o más ganas que yo de que finalizara mi carrera universitaria de manera satisfactoria. Siempre me han apoyado y ayudado en todo lo que podían, aportando lo mejor de ellos mismos.

No puedo dejar fuera a mis compañeros, ahora considerados amigos, que me han apoyado desde el primer día de carrera hasta el final, hasta este trabajo. Quiero agradeceros todos los momentos que hemos vivido juntos.

Por último, quiero destacar la gran labor realizada por Jorge Bueno, tutor en Future Space S.A., quien me introdujo en la monitorización desde el Departamentos de Sistemas, consiguiendo experiencia en el sector y dándome la posibilidad de crecer laboralmente en el mundo de las Tecnologías de la Información.

Son muchas las personas que se han interesado por este Proyecto, a cada una de ellas mil gracias y todo mi apoyo en para conseguir vuestras metas.

Resumen

El objetivo principal de este proyecto es el diseño e implementación de un sistema de monitorización de las máquinas virtuales de una empresa.

En un primer momento se expondrán los beneficios que aporta la monitorización a la empresa tanto a nivel económico como a nivel funcional y operativo. Según avance el proyecto se explicarán los elementos que compondrán la arquitectura final a nivel de software además de la justificación de la elección de cada uno de los componentes, centrándose en cubrir las necesidades del caso expuesto en el documento.

El diseño del sistema de monitorización se puede contextualizar sobre cualquier empresa que sustente parte, o todos sus servicios, sobre máquinas virtuales. La evolución de la informática y las nuevas posibilidades que proporciona la tecnología de la virtualización y la monitorización hacen que las empresas cada vez estén más interesadas en estos servicios de control y prevención, en las que la estructura que lleva a cabo la monitorización se encuentra centralizada en una única máquina. Además, se han complementado los servicios que prestan las herramientas seleccionadas con funcionalidades adicionales creadas expresamente para el proyecto expuesto en este documento.

El sistema implementado será validado mediante una serie de pruebas realizadas en la infraestructura tras la implementación del proyecto. Seguidamente se expondrán una serie de mejoras recomendadas para otorgar mayor seguridad y nuevas funcionalidades al sistema de monitorización.

Para finalizar el trabajo, se recogerá en el último apartado conclusiones extraídas tras la puesta en marcha de la solución. En él se expondrán la utilidad del sistema desde el punto de vista operativo y la experiencia del usuario obtenida tras su uso.

Summary

The main objective of this Project is the desing and implementation of a monitoring system for the virtual machines of a company.

In the first place, the benefits that monitoring brings to the company will be presented both at an economic level and at a functional and operational level. As the Project progresses, the elements that will make up the final architecture at the software level will be explained, as well as the justification for the choice of each of the components, focusing on meeting the needs of the case exposed in this project.

The design of the monitoring system can be contextualized on any company which supports part or all of its services, on virtual machines. The evolution of information technology and the new possibilities provided by virtualization and monitoring technology mean that companies are increasingly interested in these control and prevention services, in which the structure that carries out the monitoring is centralized on a single machine. In addition, the services provided by the selected tools have been supplemented with additional functionalities created expressly for the project presented in this work.

The implemented system will be validated through a series of tests carried out on the infrastructure after the implementation of the project. Next, a series of recommended improvements will be presented to provide greater security and new functionalities to the monitoring system.

Finally, the conclusions drawn after the implementation of the solution will be collected in the last section. In it, the usefulness of the system from the operational point of view and the user experience obtained after its use will be exposed.

Índice

..... 0
1. INTRODUCCIÓN 13
1.1. MOTIVACIÓN 13
1.2. OBJETIVOS 14
1.3 FASES 14
1.4. ESTRUCTURA DE LA MEMORIA 15
2. ESTADO DEL ARTE 17
2.1 INTRODUCCIÓN 17
2.2 VIRTUALIZACIÓN 17
2.3 MONITORIZACIÓN 21
2.4 LINUX EN LAS ORGANIZACIONES 22
2.4.1 Sistema operativo: CentOS7 24
2.4.1.1 Requisitos del sistema 25
2.4.1.2 Historial de lanzamientos 26
2.4.1.3. Estructura de CentOS7 27
2.4.1.4. Ventajas 29
2.5. HERRAMIENTAS UTILIZADAS 29
2.5.1. Collectd 29
2.5.2. Graphite 30
2.5.2.1. Carbon 30
2.5.2.2. Whisper 30
2.5.2.3. Graphite-web 31
2.5.3. Grafana 31
2.5.4. Nagios 32
2.5.5. Esquema de distribución de las herramientas 32
3. DISEÑO E IMPLEMENTACIÓN 35
3.1. Creación de máquina virtual 35
3.2. Instalación CentOS7 y configuración de red 37
3.3. Instalación Graphite en servidor 39
3.4. Instalación Collectd en cliente 44
3.5. Instalación Grafana en servidor 47
3.6. Instalación Nagios en servidor 52

3.7. Instalación NRPE en cliente y servidor 56
3.8. Activación e implementación de métricas y alertas 60
3.8.1. Alerta carga CPU 60
3.8.2. Alerta espacio libre disco 60
3.8.3. Alerta http 61
3.9. Activación de notificaciones por correo 61
4. VALIDACIÓN DEL DISEÑO 65
4.1. INTRODUCCIÓN 65
4.2. METODOLOGÍA 65
4.3 Caso de prueba 1 65
4.3.1. Descripción 65
4.3.2. Resultado esperado 66
4.3.3. Resultado obtenido 66
4.3.4. Conclusión del caso 66
4.4. Caso de prueba 2 66
4.4.1. Descripción 66
4.4.2. Resultado esperado 67
4.4.3. Resultado obtenido 67
4.4.4. Conclusión del caso 67
4.5. Caso de prueba 3 67
4.5.1. Descripción 67
4.5.2. Resultado esperado 68
4.5.3. Resultado obtenido 68
4.5.4. Conclusión del caso 69
4.6. Caso de prueba 4 70
4.6.1. Descripción 70
4.6.2. Resultado esperado 70
4.6.3. Resultado obtenido 70
4.6.4. Conclusión del caso 72
4.7. Caso de prueba 5 72
4.7.1. Descripción 72
4.7.2. Resultado esperado 72
4.7.3. Resultado obtenido 72
5. MEJORAS PROPUESTAS 75
5.1. Mejora de seguridad del sistema 75
5.2. Mejora del Sistema Operativo 76

6. CONCLUSIONES 77
7. BIBLIOGRAFÍA 79

Tabla de ilustraciones

Imagen 1: virtualización 18
Imagen 2: virtualización de datos 19
Imagen 3: virtualización de escritorios 19
Imagen 4: virtualización de servidores 20
Imagen 5: virtualización de sistema operativo 20
Imagen 6: virtualización de red 21
Imagen 7: esquema de distribución de las herramientas 33
Imagen 8: panel de inicio de VirtualBox 35
Imagen 9: panel de configuración de la máquina virtual 37
Imagen 10: destino de instalación del sistema operativo 38
Imagen 11: estado servicio httpd 41
Imagen 12: estado servicio carbon-cache (Graphite) 41
Imagen 13: menú de inicio de Graphite-Web 43
Imagen 14: menú métricas Graphite-Web 46
Imagen 15: ejemplo gráfico de métrica Graphite-Web 46
Imagen 16: loggin Grafana 48
Imagen 17: menú de inicio Grafana 48
Imagen 18: menú de selección de base de datos de Grafana 49
Imagen 19: configuración base de datos Grafana 50
Imagen 20: panel de Dashboards de Grafana 51
Imagen 21: add Query de Dashboard en Grafana 51
Imagen 22: query de Dashboard de Grafana 51
Imagen 23: configuración de query en Grafana 52
Imagen 24: proyección de query en Grafana 52
Imagen 25: Dashboard en Grafana 52
Imagen 26: loggin Nagios 55
Imagen 27: menú inicio Nagios 56
Imagen 28: pluggins Nagios 57
Imagen 29: prueba de check_nrpe a cliente 1 57
Imagen 30: contraseña para aplicación mailx (Nagios) 62
Imagen 31: correo aviso cliente 1 fuera de servicio 66
Imagen 32: estados cliente 1 en panel de control de Nagios 66
Imagen 33: correo restablecimiento de servicio del cliente 1 67
Imagen 34: estados de cliente 1 tras restablecer servicio 67
Imagen 35: gráfica en Graphite-Web (cpu-idle) 69
Imagen 36: estado disco / dev/mapper/centos-root 70
Imagen 37: estado almacenamiento cliente 1 en Nagios 70
Imagen 38: mensaje alerta estado disco del cliente 1 71
Imagen 39: estado almacenamiento cliente 1 en Nagios (II) 71
Imagen 40: mensaje alerta estado disco del cliente 1 (II) 71
Imagen 41: mensaje almacenamiento disco cliente 1 72
Imagen 42: loggin GitLab 73
Imagen 43: estado de http de GitLab en cliente 2 73
Imagen 44: correo alerta servicio http en cliente 2 73

1. INTRODUCCIÓN

En este primer apartado del documento se expondrán las motivaciones, objetivos, fases y estructura que caracterizan al proyecto abordado en todo el trabajo.

1.1. MOTIVACIÓN

La tendencia actual en el campo de los servicios de Tecnologías de la Información es que los usuarios, las empresas y las grandes corporaciones puedan acceder a los recursos computacionales a través de la red (Internet o Intranet). Esta tecnología elimina la necesidad de instalación de máquinas locales y permite utilizar otras más potentes y adaptadas a las tareas que deban desempeñar. Esta infraestructura computacional generalmente se sirve de manera virtualizada gracias a máquinas virtuales.

La idea principal de una máquina virtual es permitir ejecutar de manera simultánea varios sistemas operativos en un mismo hardware. Esto se traduce en una significativa reducción de costes de mantenimiento de equipos, mejor aprovechamiento hardware, reducción del uso de memoria física, etc. Además, las máquinas virtuales tienen como ventaja la gran seguridad que ofrecen, ya que se ejecutan de forma independiente del sistema anfitrión. Cuando hay varias máquinas virtuales corriendo en una única máquina física, los distintos usuarios no corren el riesgo de perder información personal o que haya desviación de la información gracias a la independencia con el sistema operativo anfitrión anteriormente mencionada.

Por otro lado, las máquinas virtuales ofrecen la posibilidad de ser trasladadas a otro equipo diferente de forma sencilla y rápida y poder seguir funcionando como lo hacían en el momento anterior a la migración. Esto se traduce en una fácil gestión de los equipos incluso en situaciones comprometidas donde haga falta cambiar la ubicación de los sistemas.

Por último, en todo sistema se debe implantar un instrumento de control para poder asegurar el correcto funcionamiento de toda la infraestructura y poder prevenir errores y situaciones críticas. Esta idea ha ido cobrando cada vez más importancia al aumentar el peso de las tecnologías en las organizaciones y el gran beneficio que reportan en el negocio. Para una empresa conocer el funcionamiento de sus equipos y servicios es de máxima importancia, ya que un mayor conocimiento de su infraestructura IT le ayuda a proteger el negocio y mejorar el funcionamiento de este. Los sistemas de monitorización permiten tener un control absoluto sobre toda la infraestructura IT, asegurando que el funcionamiento de sistemas, aplicaciones, servicios y procesos de negocio sea el correcto.

1.2. OBJETIVOS

El objetivo fundamental de este Trabajo de Fin de Grado ha sido el diseño e implantación de un sistema de monitorización de máquinas virtuales, capaz de controlar desde un servidor centralizado las diferentes máquinas virtuales que se encuentren ejecutándose en la infraestructura IT de una organización. Para ello se pretende:

1) Explicar en qué consiste el sistema que se implantará para realizar la monitorización.
2) Presentar las ventajas que se obtiene al tener monitorizadas las máquinas de una organización.
3) Explicar las fases de implementación de la solución software.
4) Creación de métricas para tener una monitorización personalizada acorde con las necesidades del proyecto.
5) Realizar una serie de pruebas donde se verificará el sistema de monitorización diseñado, comprobando que cumple las necesidades expuestas en el transcurso del documento.
6) Proponer mejoras para dotar al sistema de monitorización de un plus de seguridad y nuevas funcionalidades y mejoras que vienen en versiones recientes de los productos.

Para la implementación de la solución se han utilizado diferentes herramientas software de recolección, análisis, alojamiento y visualización de métricas junto a un sistema de alertas para tener la infraestructura IT totalmente controlada.

1.3 FASES

A continuación, se enumeran y detallan las fases que se han seguido durante la elaboración del Trabajo Fin de Grado:

1) Estudio y análisis del estado del arte en la monitorización de máquinas virtuales. En esta fase del trabajo se han estudiado las soluciones más habituales y comunes a problemas de monitorización.
2) Estudiar y analizar el estado del arte de los sistemas de virtualización empleados comúnmente en las organizaciones.
3) Análisis de la solución. En esta fase se expone el uso que se darán a las diferentes herramientas y servicios que conforman la solución propuesta en el proyecto.
4) Diseño de un sistema de monitorización a partir de los conceptos expuestos anteriormente.
5) Implementación del sistema de monitorización, donde se han realizado las siguientes tareas:
    a. Desarrollar una máquina servidor. Este apartado se centra en la creación de la máquina que aloja el servicio de monitoreo.
    b. Creación de las máquinas clientes que serán controladas y monitorizadas por el servidor.

   c. Desarrollo del sistema de monitorización. En este apartado se ha implementado la estructura que lleva a cabo la acción de monitorear.
   d. Desarrollo del sistema de comunicación entre componentes. En esta fase se ha diseñado la manera de enviar la información desde las máquinas clientes al servidor.
6) Pruebas sobre el sistema completo. Las pruebas han consistido en controlar algunas métricas de las máquinas clientes para verificar el correcto funcionamiento del proyecto. Para la realización de las pruebas ha sido necesario:
   a. Crear y configurar máquinas clientes de prueba.
   b. Generar carga sobre las máquinas clientes.
   c. Registrar los valores monitorizados para analizarlos posteriormente.
   d. Generar gráficas y alertas que posibiliten una clara comprensión de los datos obtenidos.
7) Redacción de la memoria del proyecto. En esta fase se han descrito los pasos realizados durante las fases anteriores, adjuntando los comandos utilizados y resultados de los avances que se iban obteniendo. Finalmente se han redactado unas conclusiones y posibles alternativas para mejorar el sistema final.

1.4. ESTRUCTURA DE LA MEMORIA

Este trabajo está estructurado en 6 partes que abarcan desde lo más teórico del ámbito la virtualización y la monitorización, pasando por la infraestructura en la que se basa el proyecto hasta el caso práctico donde se desarrolla la solución propuesta y las pruebas realizadas en el sistema. Las partes son las siguiente:

- Parte 1: consta de una introducción al trabajo, una exposición de las motivaciones, el objetivo del proyecto y las fases en las que se ha divido el mismo.
- Parte 2: en esta parte se introducen los conceptos fundamentales en el proyecto y el estado del arte de estos. Se empieza por el concepto de la virtualización, exponiendo los diferentes tipos que hay, después la monitorización y la importancia que ha ido adquiriendo desde hace unos años, el uso de Linux en las organizaciones, haciendo hincapié en CentOS7 y por último se detallan las características de los productos software que conforman la solución planteada.
- Parte 3: en esta fase del documento se detallan las nociones prácticas necesarias para configurar y poner en marcha el sistema de monitorización. Se detallan los pasos seguidos acompañados de imágenes, comandos y fragmentos de código para abarcar al completo el proceso de ejecución de la solución software.
- Parte 4: esta parte del documento está formada por las pruebas de evaluación realizadas sobre la solución propuesta. Estas se ejecutan sobre entornos simulados para validar el comportamiento del sistema ante ciertas situaciones y probar, así, la labor de monitorización realizada en el sistema.

- Parte 5: esta fase del Trabajo de Fin de Grado contiene un apartado de mejoras que presenta dos recomendaciones centradas en la seguridad y en el sistema operativo del sistema.
- Parte 6: esta última parte del proyecto recoge una serie de conclusiones extraídas tras el diseño y prueba del sistema de monitorización. Se pretende dar una validez al sistema tras verificar su desempeño con los casos de prueba expuestos en la parte 4.

2. ESTADO DEL ARTE

Este segundo apartado del documento trata de manera más específica el planteamiento de la solución que se llevará a cabo en este proyecto. Se dará una introducción al tema principal del proyecto, explicaciones de los conceptos claves, herramientas que conforman la solución y la integración entre ellas.

2.1 INTRODUCCIÓN

Este capítulo pretende dar una visión general de la situación actual de la virtualización y la monitorización y el punto de unión entre ambas. Además, se estudiará el sistema operativo Linux, y más concretamente CentOS7, las características del sistema operativo y las versiones por las que ha pasado desde su lanzamiento en mayo de 2004.

También se realizará una breve presentación de las herramientas utilizadas en el desarrollo de este proyecto, su funcionalidad y con qué tipo de sistemas se suelen integrar.

2.2 VIRTUALIZACIÓN

A principios de los años 90 la mayoría de las empresas tenían servidores físicos de un solo proveedor, por lo que no se permitía que las aplicaciones heredadas se ejecutarán en un hardware de otro proveedor. A medida que las empresas actualizaron sus entornos TI con servidores básicos, sistemas operativos y aplicaciones menos costosas y pertenecientes a diferentes proveedores, el hardware físico dejo de usarse de una manera ineficiente, ya que cada servidor podía ejecutar solo una tarea específica del proveedor.

A partir de esto la virtualización experimento un crecimiento exponencial, ya que se convirtió en la solución natural para 2 importantes problemas: las empresas podían dividir los servidores y ejecutar aplicaciones heredadas en varios tipos y versiones de sistemas operativos. De esta forma, los servidores comenzaron a ser utilizados de una manera más eficiente, traduciéndose en una reducción de los costes relacionados con las compras, instalación y mantenimiento.

La idea principal de la virtualización[1] es poder ejecutar varios sistemas operativos sobre el mismo servidor, que contará con su propio sistema operativo y componentes (hardware y software). Mediante este concepto, se consigue la abstracción de recursos físicos tales como hardware, software, memoria o componentes red.

El software llamado hipervisor se conecta directamente con el hardware y permite dividir un sistema en entornos separados, distintos y seguros, conocidos como máquinas

[1] https://www.redhat.com/es/topics/virtualization/what-is-virtualization

virtuales. Estas máquinas virtuales dependen directamente de la capacidad del hipervisor de separar los recursos de la máquina del hardware y distribuirlos de manera adecuada. La virtualización permite un aprovechamiento máximo de los recursos de una máquina física.

El hardware físico original donde se encuentra instalado el hipervisor se llama host, y las máquinas virtuales que utilizan estos recursos se denominan guests o máquina virtual. Estas utilizan los recursos informáticos, como la CPU, memoria y el almacenamiento, como un conjunto de medios que pueden redistribuirse fácilmente. Los operadores tienen la posibilidad de controlar las instancias virtuales de la CPU, memoria, almacenamiento y demás recursos, para que los guests reciban la cantidad de recursos que necesiten para desempeñar su función.

El hipervisor tiene la función de separar los recursos físicos de los entornos virtuales. Estos pueden conformarse como elementos principales de un sistema operativo (como en una computadora portátil) o pueden instalarse directamente en el hardware (como un servidor), que es la forma en la que la mayoría de las empresas virtualizan. Los hipervisores toman los recursos físicos y los dividen de manera tal que los entornos virtuales puedan usarlos.

Imagen 1: virtualización

Los recursos se dividen según las necesidades, desde el entorno físico hasta los numerosos entornos virtuales. Los usuarios interactúan con ellos y los ejecutan dentro del entorno virtual (máquina guest o máquina virtual). Esta máquina virtual funciona como un archivo de datos único. Al igual que con cualquier archivo digital, se puede mover de una computadora a otra, abrir en cualquiera de ellas y esperar que funcione de la misma manera.

Cuando el entorno virtual se está ejecutando y un usuario o programa lanza una instrucción que requiere recursos adicionales del entorno físico, el hipervisor transmite la solicitud al sistema físico y guarda los cambios en la caché. Todo esto sucede prácticamente a la misma velocidad que habría si este proceso se realizará dentro de la máquina física (en especial, si la solicitud se envía a través de un hipervisor OpenSource[2] basado en KVM, la máquina virtual basada en el kernel).

Se pueden diferenciar diferentes tipos de virtualización según las características que ofrecen a los usuarios:

[2] https://opensource.com/

- Virtualización de los datos

Los datos, que se encuentran repartidos por diferentes áreas de la organización, se pueden consolidar en una fuente única. La virtualización de los datos permite que las empresas los traten como si se tratara de una cadena de suministro dinámica. De esta forma, se obtiene la capacidad de procesamiento que permitirá reunir los datos de varias fuentes, integrar otras fuentes nuevas de manera fácil y transformar los datos en función de las necesidades de los usuarios. Las herramientas que se encargan de esta tarea se enfrentan a varias fuentes de datos y permiten tratarlas como una sola. De este modo se consigue dotar a las aplicaciones o usuarios de toda la información que necesiten y en el momento justo.

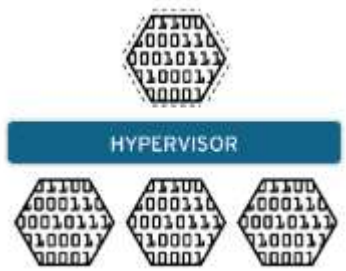


Imagen 2: virtualización de datos

- Virtualización de los escritorios

Con la administración de escritorios un administrador central (o herramienta de administración) pueden implementar entornos simulados de escritorio en un gran número de máquinas físicas al mismo tiempo. A diferencia de los entornos de escritorios tradicionales, que se caracterizan por ser instalados, configurados y actualizados físicamente en cada máquina, la virtualización de los escritorios permite que los administradores puedan realizar estas operaciones de forma masiva en todos los escritorios virtuales.

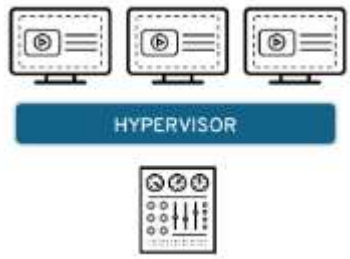


Imagen 3: virtualización de escritorios

- Virtualización de los servidores

Los servidores son computadoras cuya función será procesar un gran volumen de tareas específicas de forma efectiva. Este tipo de virtualización permite ejecutar más funciones específicas e implica dividir el servidor para que los elementos se puedan utilizar al realizar varias funciones.

Imagen 4: virtualización de servidores

- Virtualización del sistema operativo

La virtualización del sistema operativo se realiza en el kernel, es decir, los administradores de tareas centrales de los sistemas operativos. Es una forma útil de ejecutar entornos Linux y Windows de manera paralela. Las empresas también tienen la posibilidad de instalar sistemas operativos virtuales en las computadoras, dando lugar a:

  - Reducción del coste del hardware en masa, ya que las computadoras no requieren capacidades tan inmediatas.
  - Aumenta la seguridad, ya que todas las instancias virtuales se pueden supervisar y aislar.
  - Limita el tiempo destinado a los servicios IT, como puede ser la actualización de software.

Imagen 5: virtualización de sistema operativo

- Virtualización de las funciones de red

La virtualización de las funciones de red (NFV) tienen como función separar las funciones claves de una red, como pueden ser servicio de directorio, uso compartido de archivos, configuración de IP… para distribuirlas en los entornos. Cuando se independencias las funciones de software de las máquinas virtuales en las que se encontraban, las funciones específicas pueden empaquetarse en una nueva red y asignarse a un entorno. La virtualización de redes consigue reducir el número de componentes físicos necesarios para crear redes independientes.

Esta funcionalidad es muy popular en el sector de las telecomunicaciones.

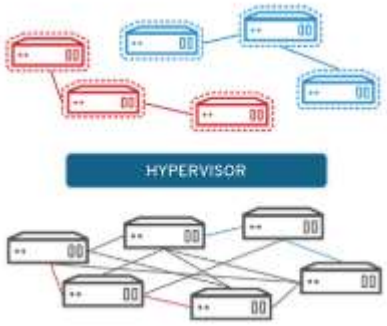


Imagen 6: virtualización de red

2.3 MONITORIZACIÓN

En las empresas, como en la sociedad hoy en día, cada vez hay más tecnología y más conectada. En las organizaciones, los procesos críticos de negocio dependen más de ella, tanto en PYMES como en multinacionales, por lo que el impacto real de los problemas técnicos es cada vez mayor y se necesita un control exhaustivo para asegurar los procesos. La monitorización es relevante de cara a mantener funcionando de manera correcta cualquier entorno, y sobre todo para optimizar mejor los recursos de los que dispone la empresa.

La monitorización[3] se puede definir como el proceso en tiempo real que abarca la recolección, procesamiento y análisis de datos cuantificables de un sistema. Involucra gran cantidad de aristas y dependiendo de cuáles son las necesidades que se tienen y ambiciones que se desean lograr, la monitorización puede ir desde un sistema para conocer el estado de una infraestructura, hasta uno más complejo, resiliente y capaz de anticipar eventos. De esta manera, la monitorización ha dejado de ser algo puramente técnico para pasar a convertirse en la herramienta ideal para obtener gran cantidad de información vital en la toma de decisiones de la empresa, en la disminución de costes y en el control del proceso industrial de la organización.

Las primeras herramientas de monitorización se centraban en obtener datos del funcionamiento de los componentes, y controlar así su disponibilidad y garantizar su correcta actividad. Según fue avanzando la tecnología y aumentando su importancia en las organizaciones, la cantidad de información empezó a aumentar considerablemente y se empiezan a obtener más datos, por lo cual, se fue haciendo necesario medir otros aspectos relacionados con la aplicación y la interacción con los usuarios.

[3] https://www.autentia.com/ Monitorización: métricas de sistema, decisiones de negocio

En el proceso de monitorización, una de las principales tareas es la recolección de métricas. Las métricas son datos, que no han sido tratados, del funcionamiento y uso de las herramientas que se desean controlar. Podemos distinguir 2 tipos de métricas:

- Métricas absolutas: aquellas de las que no existe un valor de referencia previo (usuarios conectados).
- Métricas relativas: aquellas que hacen referencia a un valor previo y que tienen un valor en un momento dado (disponibilidad de memoria).

Otro término importante en la monitorización son los indicadores. Los indicadores se pueden definir como el resultado de manipular las métricas para obtener información sobre el comportamiento basado en diferentes variables. De los indicadores se debe tener presente relacionar y medir aspectos similares.

Tanto los indicadores como las métricas nos pueden dar información para tomar decisiones de negocio relacionadas con las funcionalidades del producto o la plataforma donde se encuentra.

Las métricas de rendimiento nos proporcionan información sobre el funcionamiento de los componentes de nuestro sistema, como servidores, dispositivos o redes. El tipo de información que se puede manejar es muy diverso, desde el uso de CPU, hasta los procesos activos, pasando por memoria usada en el dispositivo o redes disponibles. Estos valores nos dan la posibilidad de tener una imagen clara del uso que se está dando a los recursos, sirven de apoyo de las tareas del equipo de operaciones y ayudan a entender el comportamiento del sistema y su interacción.

Tras todo esto se pueden concretar las siguientes ventajas que ofrece la monitorización:

- Aprovechamiento máximo de los recursos de una empresa
- Prevención de incidencias y detección de problemas
- Notificaciones de posibles problemas y alertas tempranas
- Ahorro de costes en procesos de mantenimiento IT
- Ahorro de tiempo en procesos de mantenimiento de sistemas
- Mejora de la productividad evitando errores
- Mayor de la satisfacción de los usuarios

2.4 LINUX EN LAS ORGANIZACIONES

Hoy en día es comprensible que el uso de aplicaciones, sistemas y soluciones basadas en software libre y abierto facilitan y abaratan procesos de inserción e innovación en el mundo empresarial, así como también la contribución de y apoyo de las comunidades que se generan alrededor de estas soluciones permiten adoptar de una manera más sencilla la transformación digital en las organizaciones.

Cuando la comunidad, de una manera libre y abierta, difunde, comparte y colabora de manera conjunta crea una red de experiencias muy valiosas y productivas que pueden

ser aprovechadas por infinidad de usuarios que encuentran aquí apoyo el uso de estas soluciones. Este concepto es un claro distintivo de Linux, donde la contribución entre usuarios es inmensa gracias a la gran cantidad de comunidades OpenSource que se encuentran distribuidas por todo el planeta, formadas por desarrolladores, codificadores y usuarios que se encargan de probar y mejorar continuamente sus componentes. Este trabajo no ha pasado desapercibido para la empresas y organizaciones, donde aprovechan estos resultados para innovar en áreas estratégicas.

Asimismo, muchas organizaciones de distintos sectores aportan resultados del uso y experiencias a fin de perfeccionar funcionalidades y reforzar posibles puntos débiles. Esto ha derivado en gran cantidad de avances y mejoras en el sistema que hoy en día se consideran parte natural de operar y hacer negocios en el contexto global.

Linux se ha convertido en un referente en el impulso de la innovación empresarial. Este sistema operativo es la arquitectura que soporta los sistemas de IT y las cargas de trabajo más complejas de organizaciones de todo el mundo. Tal es así que dos grandes organizaciones como la bolsa de Nueva York o Wikipedia han escogido este sistema operativo. La primera destaca de Linux su fiabilidad, el alto desempeño y la seguridad. La segunda se inclinó por esta solución debido a la eficiente forma de equilibrar y distribuir las cantidades ingentes de consultas que recibe al día. Su flexibilidad y fortaleza han hecho que Linux se haya convertido en el estándar para ejecutar las cargas de trabajo críticas en los centros de datos y en las implementaciones en la nube. Se prevé que para el año 2025 la totalidad de clientes de SAP realicen la migración a SAP HANA, un sistema de gestión de bases de datos relacionales, que se ejecuta exclusivamente sobre Linux.

Otro de los factores determinantes que han aupado a Linux es la seguridad que el sistema operativo ofrece. Muchas organizaciones han implantado esta solución en áreas críticas de sus negocios. Al ser un sistema modular, la protección del sistema operativo puede gestionarse de una manera fácil y rápida, ya que cada parte que lo conforma puede ser auditada, monitoreada y protegida de manera exclusiva. Además, Linux cuenta con múltiples herramientas enfocadas a aislar, reportar, vigilar y corregir problemas que pudieran aparecer en su sistema.

Este sistema se caracteriza también porque el espacio donde operan los usuarios está separado del kernel, lo que afianza su independencia y protección, algo vital para tecnologías de contenedores y virtualización.

De igual forma, el software creado por Torvalds ha sido la plataforma sobre la que se han basado tendencias globales que marcan el camino de la innovación en una amplia gama de industrias, como IoT, Dockers, BigData, desarrollo de Apps, Nubes privadas, públicas e híbridas, Virtualización, Automatización, Seguridad, Gestión IT…

También se puede encontrar Linux en infinidad de dispositivos y proyectos como drones, sistemas operativos (Android), impresión 3D, buscadores...

En definitiva, el hecho de que Linux sea software libre y la fiabilidad que otorga al usuario ha hecho que este sea elegido en infinidad de proyectos y organizaciones, que comparten resultados y mejoras para perfeccionar el sistema, haciendo que crezca tanto en funcionalidades y aumentando de manera notable su cuota de mercado.

2.4.1 Sistema operativo: CentOS7

Un punto fundamental a la hora de elegir un sistema operativo es considerar los servicios que se van a ofrecer y asegurarse de que existe una solución compatible. En este caso se ha optado por utilizar CentOS como sistema operativo.

CentOS es un proyecto de código abierto gratuito de nivel empresarial con el mismo rendimiento, funcionalidad, estabilidad y seguridad que el sistema operativo de pago RedHat Enterprise Linux (RHEL[4]).

CentOS es uno de los sistemas operativos más elegidos a la hora de montar un servidor Linux, y el dominante entre las empresas que ofrecen servicios de hosting y servidores.

Este sistema operativo fue lanzado en marzo de 2004. Cada versión de CentOS es mantenida por 7 años y se lanzan versiones nuevas y actualizadas cada 2 años.

La elección de este sistema operativo se debe fundamentalmente a que cumple todo tipo de exigencias y al mantenimiento, ya que se necesita un grupo reducido de personas para mantener y actualizar las máquinas. De esta forma, tener todas las soluciones unificadas bajo un mismo sistema operativo facilita considerablemente las tareas de los departamentos de sistemas, ya que están centrados en un solo Sistema Operativo que les permite perfeccionar sus habilidades y disminuir el tiempo de respuesta ante imprevistos e incidencias.

Como solución corporativa, CentOS cuenta con las mismas características que ya hacen a RHEL una solución muy valiosa:

- Estabilidad: CentOS se desarrolla de forma continua con el fin de ofrecer la plataforma perfecta para el software más reciente. Aun así, en este proceso no se pierde de vista el aspecto de seguir ofreciendo la compatibilidad con las aplicaciones antiguas. Cada paso de desarrollo orientado al futuro se da garantizando la estabilidad de los componentes activos.
  Este sistema ofrece un gran rendimiento en virtualización (basada en KVM o máquina virtual basada en el núcleo) y con una alta disponibilidad, siendo esta la razón principal por la que CentOS es muy común en servidores en la nube y virtualización.
- Seguridad: este aspecto es fundamental hoy en día, con lo que CentOS como solución corporativa basada en RHEL representa la mejor elección. Gracias a la detección proactiva de vulnerabilidades por parte del equipo de seguridad de Red Hat, su código fuente cuenta con un gran nivel de seguridad. Además, a la hora de integrar nuevos programas o realizar actualizaciones de CentOS, la

[4] https://www.centos.org/about

comprobación de seguridad y de errores tiene prioridad. Por otro lado, la distribución de Linux ofrece la extensión del kernel Security Enhanced Linux (SELinux[5]), un producto de código abierto que cuenta con la colaboración de NSA y Red Hat. Este programa se encarga de implementar controles de autorización para el uso de los recursos informáticos, protegiendo para accesos no autorizados.

- Ciclos largos de mantenimiento y soporte: desde la primera versión de CentOS, tano los lanzamientos grandes como los más pequeños han estado estrechamente vinculados a publicaciones de RHEL. Para la adaptación de código, el equipo de desarrollo prevé un periodo de 2 a 6 semanas (o de pocas horas si se trata de pequeños cambios). Los números de cada versión se mantienen (RHEL 6.2 y CentOS 6.2), aunque desde la versión 7 se añade una marca temporal (timestamp) que hace referencia a la publicación del código base. Así, por ejemplo, la fuente de la versión 7.0-1406 fue publicada en junio del año 2014. Además del control de versiones, CentOS da gran importancia al periodo de soporte técnico, donde da soporte general de hasta 7 años y un suministro de hasta 10 años de actualizaciones de seguridad.

2.4.1.1 Requisitos del sistema

Como es lógico, dependiendo de la versión de CentOS, los requisitos del hardware pueden ser muy diferentes. Aun así, podemos diferenciar 2 grupos:

1) Sin entorno de escritorio
- memoria RAM: 64 MB (mínimo).
- espacio en Disco Duro: 1024 MB (mínimo) – 2 GB (recomendado).

2) Con entorno de escritorio
- memoria RAM: 1 GB (mínimo).
- espacio en Disco Duro: 20 GB (mínimo) – 40 GB (recomendado).

En cuanto al procesador, CentOS soporta casi las mismas arquitecturas que Red Hat Enterprise Linux. Estas son:

- Intel x86-compatible (32 bit).
- Intel x86-64 (64 bit).
- Estructura directorios CentOS7

[5] https://www.redhat.com/es/topics/linux/what-is-selinux

2.4.1.2 Historial de lanzamientos

A continuación, se muestra una tabla donde se recogen cada uno de los lanzamientos de CentOS junto con la arquitectura, RHEL base y fechas de lanzamientos de ambas.

Lanzamiento de CentOS	Arquitecturas	RHEL base	Fecha lanzamiento de CentOS
2	i386	2.1	2004-05-14
3.1	i386, x86_64, ia64, s390, s390x	3	2004-03-19
3.4 – Server	i386, x86_64, ia64, s390, s390x	3.4	2005-01-23
3.7	i386, x86_64, ia64, s390, s390x	3.7	2006-04-11
3.8	i386, x86_64	3.8	2006-08-25
3.9	i386, x86_64, ia64, s390, s390x	3.9	2007-07-26
4	i386, x86_64, various	4	2005-03-09
4.6	i386, x86_64, ia64, s390, s390x, ppc (beta), sparc (beta)	4.6	2007-12-16
4.7	i386, x86_64	4.7	2008-09-13
4.7 – Server	i386, x86_64	4.7	2008-10-17
4.8	i386, x86_64	4.8	2009-08-21
5	i386, x86_64	5	2007-04-12
5.1	i386, x86_64	5.1	2007-12-02
5.1 – LiveCD	i386	5.1	2008-02-18
5.2	i386, x86_64	5.2	2008-06-24
5.2 – LiveCD	i386	5.2	2008-07-17
5.3	i386, x86_64	5.3	2009-03-31
5.3 – LiveCD	i386	5.3	2009-05-27
5.4	i386, x86_64	5.4	2009-10-21
5.5 – LiveCD	i386, x86_64	5.5	2010-05-16
5.6	i386, x86_64	5.6	2011-04-08
5.6 – LiveCD	i386, x86_64	5.6	2011-04-08
5.7	i386, x86_64	5.7	2012-09-13
5.8	i386, x86_64	5.8	2013-03-07
5.9	i386, x86_64	5.9	2011-01-17
6	i386, x86_64	6	2011-07-10
6.0 – LiveCD	i386, x86_64	6.0	2011-07-25
6.0 – LiveDVD	i386, x86_64	6.0	2011-07-27

6.0 – MinimalCD	i386, x86_64	6.0	2011-07-28
6.1	i386, x86_64	6.1	2011-12-09
6.1 – LiveCD	i386, x86_64	6.1	2011-12-09
6.1 – LiveDVD	i386, x86_64	6.1	2011-12-09
6.1 – MinimalCD	i386, x86_64	6.1	2011-12-09
6.2	i386, x86_64	6.2	2011-12-20
6.2 – LiveCD	i386, x86_64	6.2	2011-12-20
6.2 – LiveDVD	i386, x86_64	6.2	2011-12-20
6.2 – MinimalCD	i386, x86_64	6.2	2011-12-20
6.3	i386, x86_64	6.3	2012-07-10
6.3 – LiveCD	i386, x86_64	6.3	2012-07-15
6.3 – LiveDVD	i386, x86_64	6.3	2012-07-15
6.3 – MinimalCD	i386, x86_64	6.3	2012-07-10
6.4	i386, x86_64	6.4	2013-03-09
6.4 – LiveCD	i386, x86_64	6.4	2013-05-22
6.4 – LiveDVD	i386, x86_64	6.4	2013-05-22
6.4 - MinimalCD	i386, x86_64	6.4	2013-03-09
6.4	i386, x86_64	6.4	2013-03-09
7	x86_64	7.0	2014-07-07
7.1503	x86_64	7.0	2015-03-31
7.1511	x86_64	7.0	2015-12-14
7.1611	x86_64	7.0	2016-12-12

Tabla 1: historial de lanzamientos CentOS. Fuente: https.//www.centos.org/

2.4.1.3. Estructura de CentOS7

Para entender la estructura de CentOS hace falta remontarse a Linux, ya que CentOS es una bifurcación a nivel binario de GNU/Linux.

La estructura de directorios, su contenido y sus funciones viene definida por Filesystem Hiearchy Standard (FHS[6]), el estándar de jerarquía que comparten todos los sistemas Linux y derivados de UNIX. Los sistemas de ficheros de Linux se organizan en una estructura jerárquica de tipo árbol. El nivel mas alto de ficheros es raíz (/), y todos los demás ficheros y directorios se encuentran debajo de este.

En este sistema operativo todo son ficheros, desde los directorios hasta los dispositivos pasando por los propios ficheros.

[6] https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html

Por debajo del directorio raíz (/) se encuentra un conjunto de directorios común a la mayoría de las distribuciones de GNU/Linux. Estos directorios son:

- Directorio /bin: directorio donde se alojan los ficheros ejecutables.
- Directorio /boot: destinado a ficheros y otros directorios de arranque.
- Directorio /dev: contiene ficheros de dispositivos.
- Directorio /etc: ficheros y directorios correspondientes a configuraciones específicas del sistema.
- Directorio /home: directorio utilizado a nivel de usuario, donde se alojan documentos, directorios, etc.
- Directorio /lib: contiene librerías compartidas necesarias para los binarios de /bin/, /sbin/ y el núcleo del sistema.
- Directorio /lost+found: alojamiento de archivos perdidos pertenecientes a cada partición.
- Directorio /media: en él se montan los dispositivos multimedia como las unidades ópticas etc.
- Directorio /mnt: sistemas de ficheros montados temporalmente.
- Directorio /proc: aquí se encuentra información sobre cpu, discos, tiempo uptime, irqs, memoria, etc.
- Directorio /root: directorio del superusuario del sistema.
- Directorio /sbin: contiene archivos ejecutables que suelen ser comandos usados para a administración del sistema.
- Directorio /sys: directorio con parámetros de configuración que se estén ejecutando. Datos del kernel, bus, dispositivos y demás.
- Directorio /tmp: directorios y archivos temporales que se eliminan con el apagado del sistema.
- Directorio /usr: directorio compartido entre todos los usuarios del sistema. Se guardan aplicaciones, librerías, etc.
- Directorio /var: se almacenan datos que están en cambio continuo como ficheros de log del sistema, correo, etc.

Expuesto esto, la FHS distingue entre directorios estáticos, dinámicos, compartidos y restringidos. Los primeros, los estáticos, serian aquellos que contienen binarios, documentación, bibliotecas… y que únicamente pueden ser cambiados por el usuario administrador del sistema, mientras que los dinámicos pueden cambiarse por cualquier usuario. Los directorios compartidos, como su propio nombre indica, pueden compartirse entre varios usuarios mientras que los restringidos son directorios propios de cada máquina, ya que contienen configuraciones, y no pueden ser compartidos con otros usuarios.

2.4.1.4. Ventajas

Algunas de las ventajas que se consiguen usando CentOS7 son:

- CentOS es compatible con la estrategia de redistribución del proveedor y obtiene soporte completo con actualizaciones de seguridad y material de capacitación.
- CentOS7 es un sistema operativo muy estable, caracterizado por tener muy pocos problemas, reduciéndose notablemente el riesgo de caídas y errores, ya que solo ejecuta versiones estables de software empaquetado.
- Mejora el rendimiento y el equilibrio de carga de los recursos configurando los equipos para que estos funcionen de forma colectiva, con un grupo de servidores que comparten un sistema de archivos común y ofrecen aplicaciones de alta disponibilidad.
- Los usuarios de CentOS7 tienen acceso a características de seguridad actualizadas a nivel de empresa, incluyendo un potente firewall y el mecanismo de políticas SELinux.
- Con una instalación de CentOS, los usuarios tienen acceso a un soporte a largo plazo durante 7 años, con actualizaciones de seguridad y parches críticos mantenidos durante una década después del lanzamiento inicial.
- La plataforma CentOS7 goza de un nivel de estabilidad superior a largo plazo que otras distribuciones del mercado, caracterizándose por menos errores y agujeros de seguridad que la competencia, por lo que no necesita nuevas actualizaciones de hardware con tanta frecuencia.

2.5. HERRAMIENTAS UTILIZADAS

Para la implementación del sistema introducido en los apartados anteriores, se ha optado por una serie de herramientas que obtienen de manera sencilla y fiable las métricas necesarias para que se tenga un control veraz de la situación del sistema a monitorizar. La justificación de la elección de estas herramientas radica en que todas ellas se integran perfectamente entre sí y que al ser herramientas de código abierto nos proporciona la libertad para ajustar el funcionamiento de cada una de ellas a las necesidades concretas del proyecto.

2.5.1. Collectd

Collectd es un demonio que recopila periódicamente las métricas de rendimiento del sistema y las aplicaciones y proporciona mecanismos para almacenar los valores de varias maneras, como archivos o tipo RRD[7] (Round Robin Database), un sistema de gráficos y registro de datos de alto rendimiento muy utilizado en la industria OpenSource para datos de series temporales.

[7] https://es.wikipedia.org/wiki/RRDtool

Hay varias diferencias claves entre Collectd y los demás recolectores de métricas que han hecho que esta herramienta sea la elegida. Por un lado, este demonio está escrito en C, ideal para los aspectos de rendimiento y portabilidad, permitiéndole ejecutarse en sistemas sin lenguaje de script o demonio cron. Por otro lado, incluye numerosas optimizaciones y funciones para manejar las numerosas métricas que proporciona. Este demonio viene con más de 100 complementos, activamente desarrollados, respaldados y documentados, haciendo que su utilización sea sencilla.

Este demonio recopila las métricas de varias fuentes, como el sistema operativo, aplicaciones, archivos de registro, dispositivos externos… almacenando la información o poniéndola a disposición en la red. Estas métricas son esenciales para monitorizar sistemas, como en este proyecto, pero también son ideales para la identificación de cuellos de botella de rendimiento y predicción de carga del sistema en situaciones futuras.

2.5.2. Graphite

Es una herramienta gratuita de software de código abierto que monitorea y gráfica datos numéricos de series temporales como el rendimiento de sistemas informáticos. Esta herramienta almacena y muestra datos de series temporales en tiempo real.

La elección de Graphite se basó en que es una herramienta de monitoreo ideal tanto para hardware como para infraestructura en la nube.

Está formada por 3 componentes, cada uno de ellos encargado de tareas específicas que proporcionan al paquete completo la capacidad de representar los datos de series temporales que llegan a Graphite.

2.5.2.1. Carbon

Este componente se encarga de recibir los datos de series temporales. Como se ha especificado anteriormente, Graphite no recopila datos, si no que espera que estos lleguen a él, por lo que Carbon realiza la tarea de escuchar de forma pasiva hasta que le lleguen datos de los equipos monitorizados.

2.5.2.2. Whisper

Es una biblioteca de base de datos cuya función es almacenar datos de series temporales que luego serán utilizados por Graphite-Web para crear los gráficos.

Una base de datos de series temporales es un sistema software optimizado para el almacenamiento y la recuperación de datos de series temporales de una manera rápida y segura. Una de sus características principales es que prioriza las cargas de trabajo. dependiendo de las condiciones involucradas y la configuración aplicada, el kernel debe hacer compromisos para manejar su carga de trabajo de la manera más eficiente posible. Otra característica de este sistema es que realiza un almacenado de las escrituras en una memoria intermedia haciendo las consultas de una manera más fluida y rápida, siempre y cuando tenga suficiente memoria para trabajar.

Desde el primer momento, Whisper fué creado con la función de corregir un error de las bases de datos RRD, que era el no aceptar datos fuera de secuencia; si la métrica A con una marca de tiempo de 09:05:00 llegaba después de la métrica B con una marca de tiempo 09:06:00, la métrica A era descartada por completo por RRD. Whisper abordó esta deficiencia de diseño específicamente y simplificó la configuración y diseño de los periodos de retención de cada archivo de base de datos. Cuando se crea un archivo Whisper, se le da un tamaño fijo que nunca cambiara. Dentro del archivo Whisper hay buckets, que quedan definidos en los archivos de configuración, para puntos de datos con diferentes resoluciones. Por ejemplo:

Bucket A: puntos de datos con resolución 10s.

Bucket B: puntos de datos con resolución 60s.

Cada depósito también tiene un atributo de retención que indica la cantidad de tiempo que se deben conservar los puntos de datos en el depósito, quedando los Buckets de la siguiente forma:

Bucket A: puntos de datos con resolución 10s retenidos 6 horas.

Bucket B: puntos de datos con resolución 60s retenidos 1 día.

Teniendo en cuenta los datos anteriores, Whisper calcula cuántos puntos necesitará para mantener cada bucket.

2.5.2.3. Graphite-web

Graphite-web es la aplicación web de esta herramienta. Es una aplicación web de Django, un framework de desarrollo web de código abierto escrito en Python. Graphite-web se encarga de mostrar los gráficos generados con las métricas gracias a la biblioteca Cairo, una biblioteca de programación de código abierto que proporciona una API independiente del dispositivo basada en gráficos vectoriales.

2.5.3. Grafana

Grafana es un software libre basado en licencia de Apache 2.0 que permite la visualización, análisis, consulta y alertas de métricas independientemente de donde estas estén almacenadas. Proporciona herramientas para convertir los datos de las bases de datos de series temporales en gráficos. A partir de una serie de datos previamente recolectados se obtiene un panorama gráfico de una situación.

Hoy en día hay disponibles diferentes herramientas de visualización de métricas, pero se ha escogido Grafana por sus ventajas frente a las demás.

Una de ellas es el modo TV, que permite que cada cierto tiempo, previamente prefijado, puede mostrar al usuario diferentes paneles de control que hayan sido guardados en listas de reproducción. Esta ventaja es muy valorada ya que busca solucionar dos detalles: dividir la información que se debe de mostrar para apreciar esta de mejor y manera y de forma más clara y otra es captar la atención del supervisor, ya que al

cambiar periódicamente lo que se muestra por pantalla se evita la monotonía de los datos.

Otra ventaja es el uso del teclado. Grafana tiene infinidad de atajos accesibles desde el teclado que permiten operar de manera más ágil y efectiva.

Otra ventaja significativa es la gran cantidad de usuarios con la que cuenta. De esta manera, existe infinidad de información y trabajos realizados para esta herramienta, como tableros, listas de reproducción, guías y tutoriales accesibles gratuitamente.

2.5.4. Nagios

Nagios es un sistema de monitorización de redes muy utilizado. Es una herramienta de código abierto, cuya función es vigilar equipos (hardware) y servicios (software), alertando cuando el comportamiento que se registra sobre ellos es el no deseado.

Una de sus características principales es la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP…), la monitorización de hardware (carga del procesador, uso de discos, memoria, puertos…), independencia de sistemas operativos, monitorización de forma remota mediante SSL cifrados o SSH y programación de plugins específicos.

Por otro lado, permite el chequeo de servicios paralizados, posibilidad de definir la jerarquía de la red, permite definir manejadores de eventos que se ejecuten en ciertos eventos en un servicio o host. Nagios permite la visualización del estado de red en tiempo real a través de su interfaz web, con la posibilidad de generar informes y gráficas de comportamiento de los sistemas que se están monitorizando, además de generar un listado de notificaciones enviadas, historial de problemas, archivos de registros…

Se trata de una herramienta software que proporciona una gran versatilidad y facilidad para consultar cualquier parámetro de interés de un sistema, generando alertas que pueden ser recibidas por correo electrónico o SMS cuando los parámetros exceden de los márgenes establecidos.

Nagios fue originalmente diseñado para ser utilizado en Linux, pero también existen variantes que se ejecutan en UNIX.

Nagios está licenciado bajo General Public License Version 2 (GNU[8]), publicada por la Free Sotware Fundation (FSF), otorgnado a los usuarios finales la libertad de estuiar, compartir y modificar el software.

2.5.5. Esquema de distribución de las herramientas

Tras haber presentado las diferentes herramientas que se van a implantar en la solución software y sus componentes se puede presentar la organización que tendrá el proyecto. En este esquema se pretende aclarar cómo se realizará la comunicación entre los distintos programas y como es el flujo que siguen los datos recolectados desde el primer momento de la monitorización.

[8] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

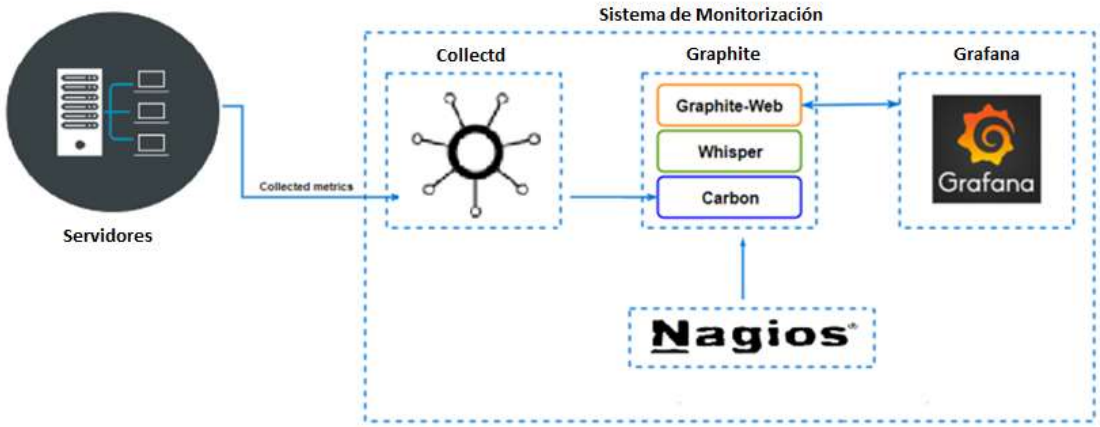


Imagen 7: esquema de distribución de las herramientas

3. DISEÑO E IMPLEMENTACIÓN

Una vez presentadas las necesidades, ventajas y funcionalidades de la monitorización de los servicios IT, se pasa a detallar como se ha implementado la solución software propuesta.

En esta parte del documento se explica cómo se han interconectado todos los elementos descritos a lo largo del documento, desde lo más general, la instalación del sistema operativo en las máquinas, hasta lo más específico, la creación e implementación de las métricas basadas en las necesidades del proyecto.

3.1. Creación de máquina virtual

Antes de comenzar a exponer la instalación de las diferentes herramientas software se va a explicar cómo han sido creadas las máquinas virtuales.

el primer paso de toda la solución es crear las máquinas virtuales en un software de virtualización, en este caso, VirtualBox. Una vez instalado el software en el sistema se ejecuta y aparece la siguiente ventana.

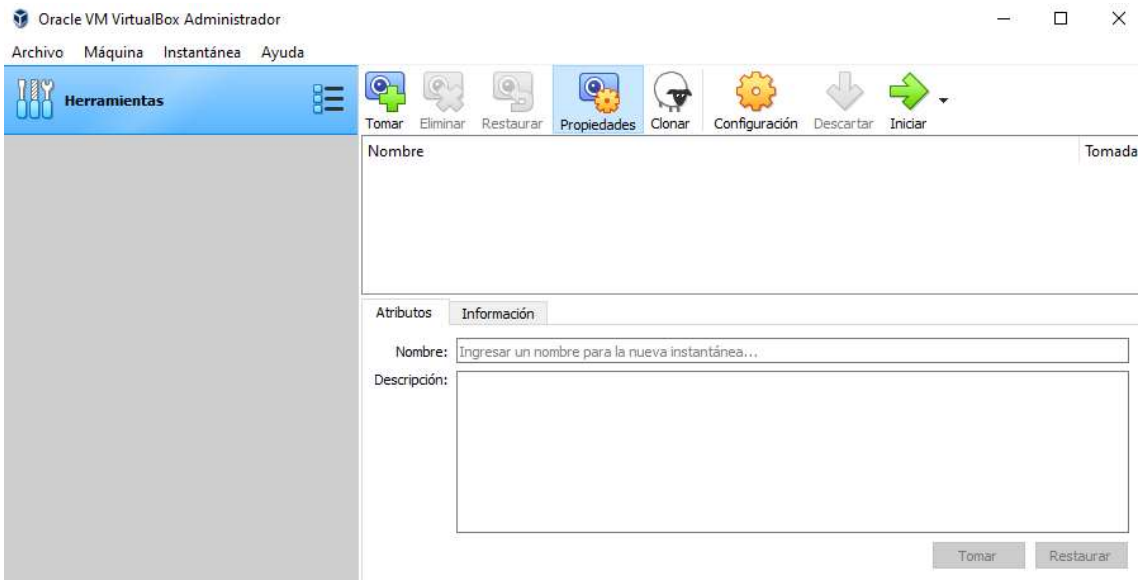


Imagen 8: panel de inicio de VirtualBox

Para crear una máquina virtual se pulsa sobre “Máquina” y nos aparecen diferentes ventanas para configurar algunos de los aspectos básicos de la máquina virtual.

En primer lugar, ha de elegirse el nombre de la máquina, sistema operativo que se va a instalar en ella y la ubicación donde se almacenará. En tipo debe escogerse “Linux” y en versión “Red Hat (64-bit)”.

El siguiente paso del asistente permite configurar la memoria RAM de la que dispondrá la máquina virtual. Se debe de establecer una cantidad de memoria suficiente para que

el sistema funcione sin problemas y pueda ejecutar los diferentes programas de manera fluida, pero permitiendo que nuestro sistema operativo base tenga recursos suficientes para ejecutar sus tareas. En este caso se ha dotado a cada una de las máquinas de 1024MB de memoria RAM.

Siguiendo con el asistente se pasa a la configuración del disco duro virtual. VirtualBox ofrece diferentes opciones en este apartado, crear una máquina virtual sin disco duro, crear un disco duro virtual nuevo o cargar un disco duro ya existente. En este caso se opta por la segunda opción, crear un disco duro virtual nuevo.

La siguiente ventana del asistente determina el tipo de archivo que se quiere usar para el disco duro de la máquina virtual. En este caso de utiliza el tipo VDI (VirtualBox Disc Image).

El siguiente paso del asistente también es referido al disco duro. Esta vez se encarga de determinar cómo se quiere determinar el tamaño de este. Por un lado, ofrece la opción de reservar dinámicamente el espacio, que solo usará espacio a medida que se llena (hasta un máximo de tamaño fijo), y por otro lado un tamaño fijo de disco, que selecciona una cantidad fija de memoria. En este proyecto se ha utilizado la primera opción, reservado dinámicamente.

El último paso del asistente es determinar ese máximo tamaño fijo del disco duro. En este caso se ha marcado el límite en 8,00 GB.

Tras estos pasos ya se tiene la máquina virtual creada y lista para ejecutarse.

Para seguir con la configuración de la máquina virtual se debe seleccionar la máquina y pinchar sobre “Configuración”. Este apartado permite configurar todos los aspectos de la máquina, tanto las configuraciones anteriores como las nuevas. Desde aquí se debe de seleccionar el disco desde el que se cargará el sistema operativo. Esto se realiza desde la ventana “Almacenamiento”, seleccionando el disco que aparece vacío en “Controlador: IDE” y seleccionando en la unidad óptica el disco con el sistema operativo.

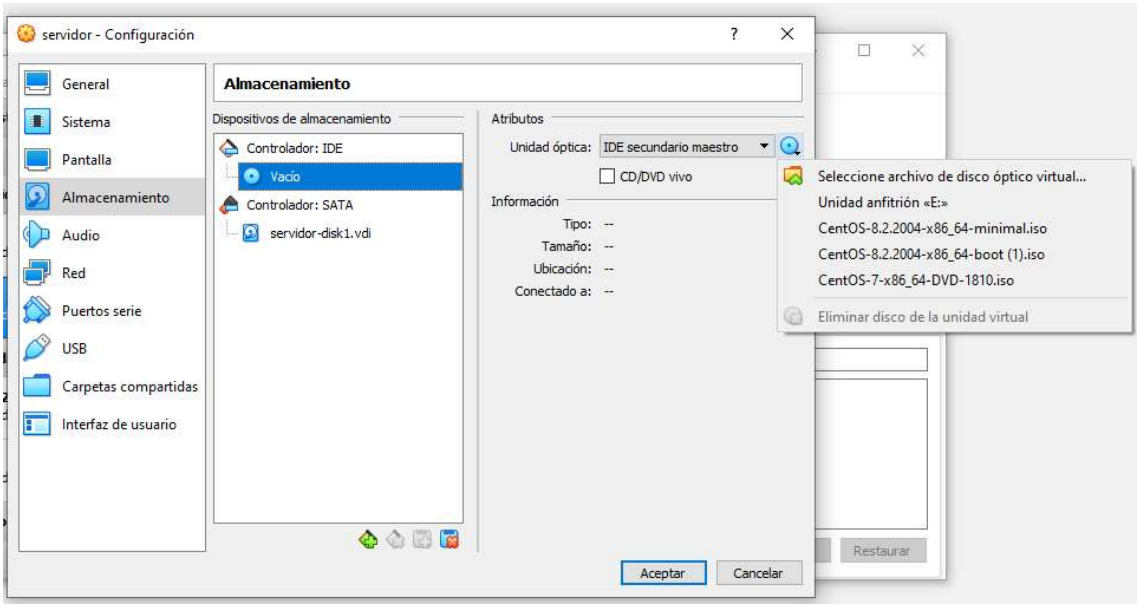


Imagen 9: panel de configuración de la máquina virtual

Otro aspecto que se debe de con figurar es el adaptador red. En este caso se debe de pinchar en el apartado “Red” y el adaptador 1 debe conectarse a “Nat” y se activa el adaptador 2 que estará conectado a “Adaptador puente”.

3.2. Instalación CentOS7 y configuración de red

El primer paso es arrancar la máquina virtual. Cuando ya ha cargado aparece el menú de inicio de la instalación con dos opciones: “Install CentOS 7” y “Test this media & Install CentOS 7”. En este caso se ha pasado directamente a instalarlo, escogiendo la primera opción. El sistema de instalación se arranca y se muestras distintos mensajes de estado.

Tras esto aparece una ventana donde se debe de seleccionar el idioma y la variedad de este.

La siguiente ventana que aparece es un primer menú con un resumen de la instalación, donde la mayoría de las opciones se rellenan automáticamente y solo hay que configurar el destino de la instalación, seleccionando el disco virtual de la máquina.

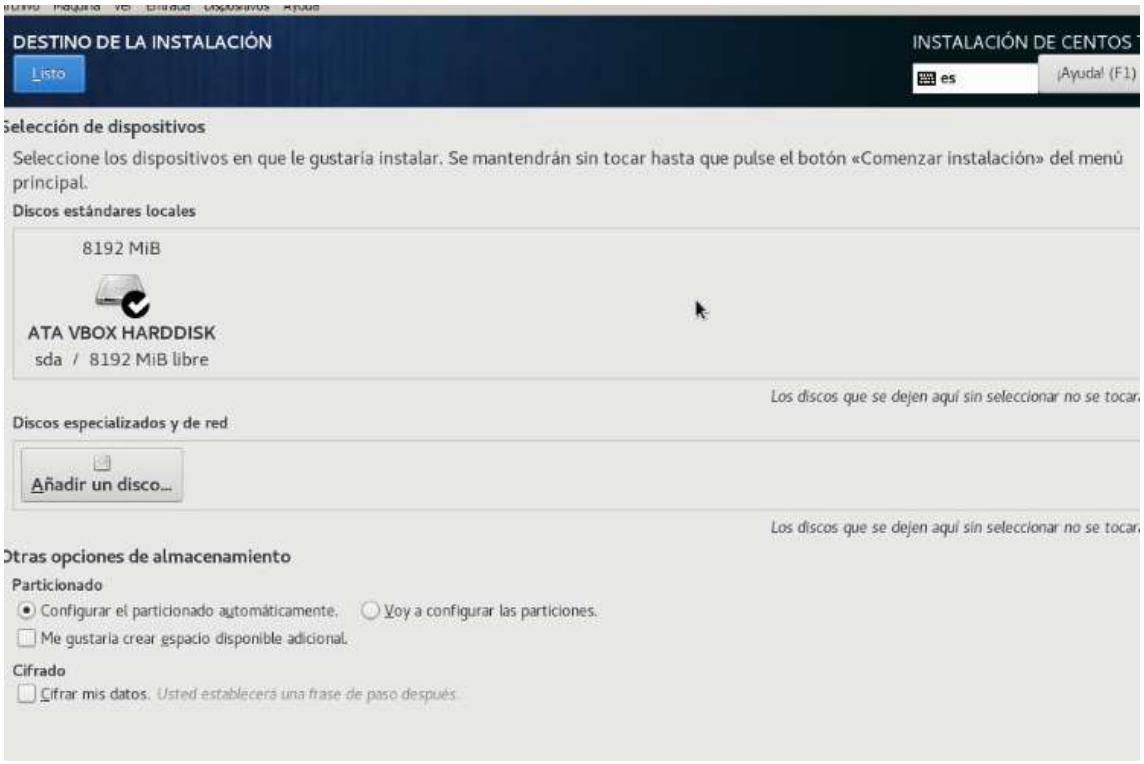


Imagen 10: destino de instalación del sistema operativo

Continuando con el proceso de instalación pide determinar la contraseña de root y la creación de un usuario. Tras esto se instala el sistema operativo y se reinicia la máquina cuando el proceso ha finalizado.

A partir de este momento la máquina ya tiene instalado CentOS7. El siguiente paso es el de configurar la red para que la máquina pueda conectarse a internet y poder descargar los diferentes paquetes software y realizar actualizaciones.

Para ello se debe de editar el archivo de configuración de red. Se requiere que cada una de las máquinas tengan una dirección IP fija. Debido a esto, las diferentes máquinas virtuales cuentan con 2 adaptadores de red, el adaptador 1 de tipo NAT y el adaptador 2 de tipo puente.

El archivo de configuración del adaptador 1 es ifcfg-enp0s3 y contiene la siguiente información:

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s3
DEVICE=enp0s3
ONBOOT=yes
ZONE=public
```

El archivo de configuración del adaptador 2 es ifcfg-enp0s8 y contiene la siguiente información:

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=no
IPV6_DEFROUTE=no
IPV6_FAILURE_FATAL=no
NAME=enp0s8
DEVICE=enp0s8
ONBOOT=yes
IPADDR=192.168.1.41
NETMAASK=255.255.255.0
ZONE=public
```

En el archivo anterior se puede ver como se le asigna a la máquina una dirección IP fija. Para ello se pone BOOTPROTO=static, IPADDR con la dirección IP deseada, NETMASK con la dirección de la máscara red y ZONE=public.

Ambos archivos se encuentran ubicados en la ruta /etc/sysconfig/network-scripts/.

3.3. Instalación Graphite en servidor

A continuación, se detallan los comandos utilizados para descargar e instalar la herramienta Graphite en la máquina servidor.

El primer paso que damos es instalar el paquete para Linux empresarial (EPEL) que crea mantiene y administra un grupo de paquetes de alta calidad para Linux, Red Hat y CentOS.

```
sudo yum -y install epel-release
```

Seguidamente instalamos el paquete Python-pip cuya función es instalar y administrar paquetes de software escritos en Python.

```
sudo yum -y install python-pip
```

Tras esto pasamos a instalar los componentes de Graphite: la parte web y el carbon.

```
sudo yum install -y graphite-web python-carbon
```

Después instalamos algunos paquetes más para que se puedan crear gráficos, se permita alojar paquetes Python y se pueda usar Apache.

```
sudo yum -y install httpd gcc gcc-c++ git pycairo mod_wsgi
```

Seguimos instalando más paquetes.

```
sudo yum -y install python-devel blas-devel lapack-devel libffi-devel
```

Tras esto pasamos a deshabilitar SELinux, un módulo de seguridad del kernel de Linux que controla el acceso para aplicaciones, procesos y archivos dentro del sistema.

Para ello ponemos el estado de SELinux en deshabilitado (/etc/sysconfig/selinux):

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#    minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Tras esto reiniciamos el sistema.

```
sudo reboot
```

Ahora pasamos al firewall. Vamos a activar los servicios http y https y abrir los puertos necesarios.

```
sudo firewall-cmd --permanent --add-service=http
```

```
sudo firewall-cmd --permanent --add-service=https
```

```
firewall-cmd --zone=public --add-port=80/tcp --permanent
```

Para que los cambios tengan efecto debo reiniciar el firewall.

```
firewall-cmd --reload
```

Una vez que tenemos esto, se pasa a comprobar los servicios httpd y carbon, para ver que todo funciona correctamente y no aparece ningún error.

```
[root@localhost plugins]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabl
ed)
   Active: active (running) since sáb 2020-03-28 12:32:36 CET; 24min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 1107 (httpd)
   Status: "Total requests: 36; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─1107 /usr/sbin/httpd -DFOREGROUND
           ├─1361 /usr/sbin/httpd -DFOREGROUND
           ├─1362 /usr/sbin/httpd -DFOREGROUND
           ├─1363 /usr/sbin/httpd -DFOREGROUND
           ├─1364 /usr/sbin/httpd -DFOREGROUND
           ├─1365 /usr/sbin/httpd -DFOREGROUND
           ├─1405 /usr/sbin/httpd -DFOREGROUND
           ├─1409 /usr/sbin/httpd -DFOREGROUND
           ├─1413 /usr/sbin/httpd -DFOREGROUND
           ├─1417 /usr/sbin/httpd -DFOREGROUND
           └─1418 /usr/sbin/httpd -DFOREGROUND

mar 28 12:32:33 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
mar 28 12:32:35 localhost.localdomain httpd[1107]: AH00558: httpd: Could not reliabl...e
mar 28 12:32:36 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

Imagen 11: estado servicio httpd

```
[root@localhost plugins]# systemctl status carbon-cache
● carbon-cache.service - Graphite Carbon Cache
   Loaded: loaded (/usr/lib/systemd/system/carbon-cache.service; enabled; vendor preset: disabled)
   Active: active (running) since sáb 2020-03-28 12:32:37 CET; 26min ago
  Process: 1106 ExecStart=/usr/bin/carbon-cache --config=/etc/carbon/carbon.conf --pidfile=/var/run/carbon-cache.pid --logdir=/var/log/carbon/ start (code
=exited, status=0/SUCCESS)
 Main PID: 1382 (carbon-cache)
   CGroup: /system.slice/carbon-cache.service
           └─1382 /usr/bin/python2 -s /usr/bin/carbon-cache --config=/etc/carbon/carbon.conf --pidfile=/var/run/carbon-cache.pid --logdir=/var/log/carb...

mar 28 12:32:33 localhost.localdomain systemd[1]: Starting Graphite Carbon Cache...
mar 28 12:32:37 localhost.localdomain systemd[1]: Can't open PID file /var/run/carbon-cache.pid (yet?) after start: No such file or directory
mar 28 12:32:37 localhost.localdomain systemd[1]: Started Graphite Carbon Cache.
```

Imagen 12: estado servicio carbon-cache (Graphite)

Como se ve en las imágenes, los dos servicios se encuentran activos.

Seguidamente pasamos a configurar Graphite introduciendo una contraseña e insertando la zona horaria donde va a trabajar el programa.

Para ello abrimos el archivo de configuración que se encuentra en la siguiente ruta:

```
vi /etc/graphite-web/local_settings.py
```

```
# Set this to a long, random unique string to use as a secret key for this
# install. This key is used for salting of hashes used in auth tokens,
# CRSF middleware, cookie storage, etc. This should be set identically among
# instances if used behind a load balancer.
SECRET_KEY = 'hmz$4694'
```

```
# Set your local timezone (Django's default is America/Chicago)
# If your graphs appear to be offset by a couple hours then this probably
# needs to be explicitly set to your local timezone.
TIME_ZONE = 'Europe/Madrid'
```

Pasamos a ejecutar el script de configuración de la base de datos con el siguiente comando:

```
PYTHONPATH=/usr/share/graphite/webapp django-admin syncdb --
settings=graphite.settings
```

Ahora solo queda configurar Apache. Lo primero que hacemos es eliminar la página de índice predeterminada de Apache.

```
echo > /etc/httpd/conf.d/welcome.conf
```

Después editamos el archivo de configuración (/etc/httpd/conf.d/graphite-web.conf) y reemplazo el bloque de Graphite por lo siguiente:

```
<Directory "/usr/share/graphite/">
       Require all granted
       Order allow,deny
       Allow from all
   </Directory>
```

Asigno los permisos adecuados al directorio de Graphite:

```
sudo chown apache:apache /var/lib/graphite-web/graphite.db
```

Evito un error relacionado con la creación de índices:

```
touch /var/lib/graphite-web/index
```

Inicio Apache y habilito el inicio automático:

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

En la configuración de httpd podemos encontrar información más detallada de cómo funciona y está configurado el servicio Apache.

Abrimos el archivo de configuración que se encuentra en la siguiente ruta:

```
vi /etc/httpd/conf/httpd.conf
```

Dentro podemos ver cuál es el puerto que escucha. Por defecto utiliza el puerto 80.

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

También entramos en la configuración de Graphite para conocer más a fondo el funcionamiento y configuración de la herramienta.

Para ello entramos en su archivo de configuración con el siguiente comando:

```
vi /etc/httpd/conf.d/graphite-web.conf
```

Se ha definido la dirección IP de la máquina como nombre del servidor, el puerto que tiene por defecto y los archivos donde almacenará los logs.

```
<VirtualHost *:80>
    ServerName 192.168.1.41
    DocumentRoot "/usr/share/graphite/webapp"
    ErrorLog /var/log/httpd/graphite-web-error.log
    CustomLog /var/log/httpd/graphite-web-access.log common
```

En esto momento tenemos todo listo para poder entrar en la interfaz web de Graphite. Para ello vamos al navegador y se debe introducir la dirección IP que se ha puesto de ServerName y el puerto.

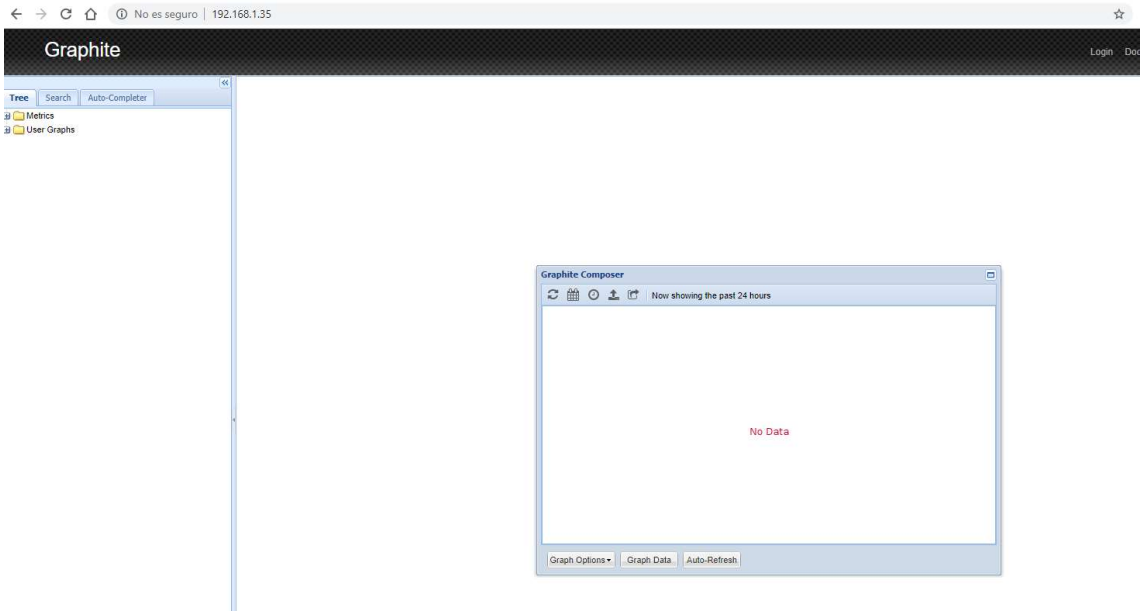


Imagen 13: menú de inicio de Graphite-Web

3.4. Instalación Collectd en cliente

A continuación, se detallan los comandos utilizados para instalar el recolector de métricas Collectd en las máquinas clientes del proyecto.

El primer paso que damos es instalar el paquete para Linux empresarial (EPEL) que crea mantiene y administra un grupo de paquetes de alta calidad para Linux, Red Hat y CentOS.

```
sudo yum -y install epel-release
```

Tras esto pasamos a descargar el paquete de Collectd.

```
sudo yum install collectd
```

Seguidamente iniciamos la herramienta.

```
sudo systemctl start collectd
```

Después habilitamos el inicio automático de Collectd.

```
sudo systemctl enable collectd
```

En este momento ya tenemos la herramienta instalada en la máquina. Ahora se debe configurar para que mande las métricas a la máquina servidor. Para ello entramos en el archivo de configuración de Collectd.

```
vi /etc/collectd.conf
```

El primer paso que realizamos es dar un nombre a la máquina para que a la hora de ver las métricas en Graphite sepamos a qué máquina pertenecen:

```
Hostname    "192.168.1.45"
```

El siguiente paso es descomentar el Plugin write_graphite de la lista de plugins:

```
LoadPlugin write_graphite
```

Por último, editamos el plugin write_graphite dejándolo de la siguiente manera:

En Host ponemos la dirección IP del servidor.

```
<Plugin write_graphite>
  <Node "example">
    Host "192.168.1.20"
    Port "2003"
    Protocol "tcp"
    ReconnectInterval 0
    LogSendErrors true
    Prefix "collectd"
    Postfix "collectd"
    StoreRates true
    AlwaysAppendDS false
    EscapeCharacter "_"
    SeparateInstances false
    PreserveSeparator false
    DropDuplicateFields false
  </Node>
</Plugin>
```

El último paso que tomamos es el de abrir el puerto 2003 en la máquina servidor, que es el que utilizará por defecto para conectarse con el servidor.

```
firewall-cmd --zone=public --add-port=2003/tcp --permanent
```

Para que los cambios tengan efecto debo reiniciar el firewall.

```
firewall-cmd --reloadsu
```

Para que funcione correctamente se debe de deshabilitar SELinux, ya que produce un error al conectar con la máquina servidor y no deja trabajar de manera correcta al plugin write_graphite. Para desablilitar SELinux se debe de ejecutar los siguientes comandos:

```
su
setenforce 0
```

Tras esto se debe de editar el archivo de configuración de SELinux. (/etc/sysconfig/selinux) quedando de la siguiente manera.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#    minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Tras esto se debe de reiniciar el sistema.

Ahora vamos a Graphite y vemos que aparece el cliente dentro de la capeta Carbon:

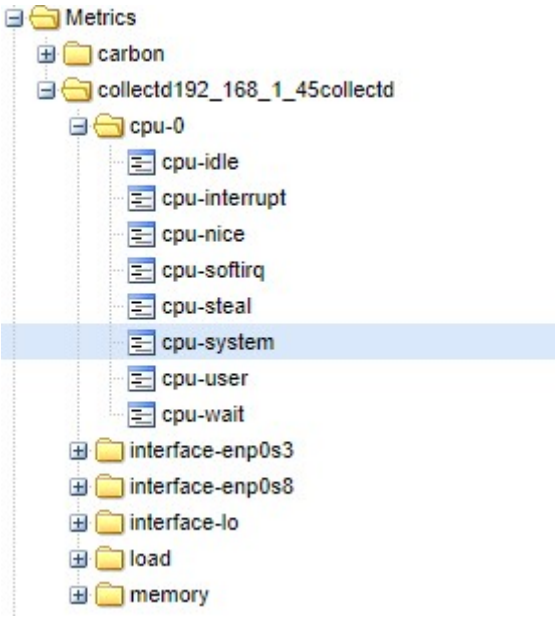


Imagen 14: menú métricas Graphite-Web

Ahora ya se pueden apreciar resultados de la monitorización:

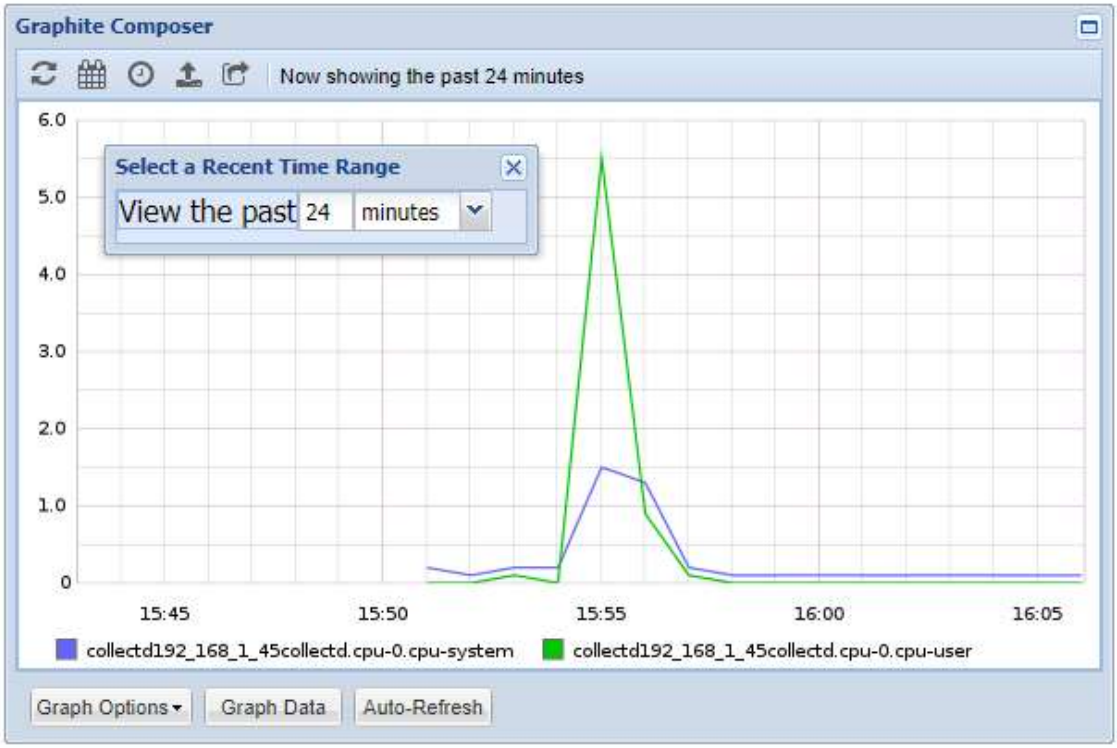


Imagen 15: ejemplo gráfico de métrica Graphite-Web

3.5. Instalación Grafana en servidor

A continuación, se detallan los comandos utilizados para instalar Grafana, una herramienta que nos sirve para visualizar métricas sin importar el almacén de datos backend.

El primer paso que debemos de realizar es añadir un repositorio a nuestra maquina cliente. Este se creará con el siguiente comando:

```
vi /etc/yum.repos.d/grafana.repo
```

Dentro del repositorio debemos incluir la siguiente información:

```
[grafana]
nombre = grafana
baseurl = https://packages.grafana.com/oss/rpm
repo_gpgcheck = 1
habilitado = 1
gpgcheck = 1
gpgkey = https://packages.grafana.com/gpg.key
sslverify = 1
sslcacert = /etc/pki/tls/certs/ca-bundle.crt
```

Tras esto pasamos a instalar Grafana.

```
yum install -y grafana
```

El siguiente paso es iniciar el servicio y habilitar el inicio automático:

```
systemctl start grafana-server
```

```
systemctl enable grafana-server
```

Por último, se debe abrir el puerto 3000 que es el que utiliza Grafana por defecto:

```
firewall-cmd --zone=public --permanent --add-port=3000
```

Reiniciamos el firewall.

```
firewall-cmd --reload
```

Tras esto, en el navegador, ponemos la dirección IP de la máquina y el puerto (IP:3000) y nos aparece la interfaz web de Grafana.

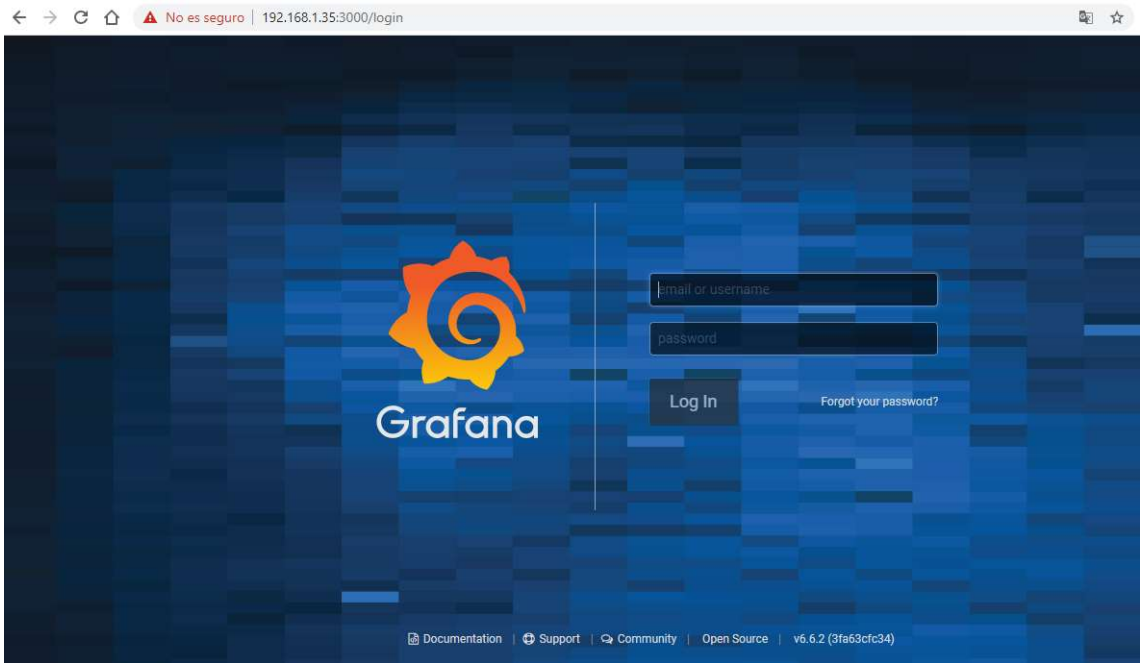


Imagen 16: loggin Grafana

Las credenciales por defecto son admin/admin. Tras entrar, nos obliga a cambiar la contraseña.

Una vez dentro, el panel que se nos muestra es el siguiente:

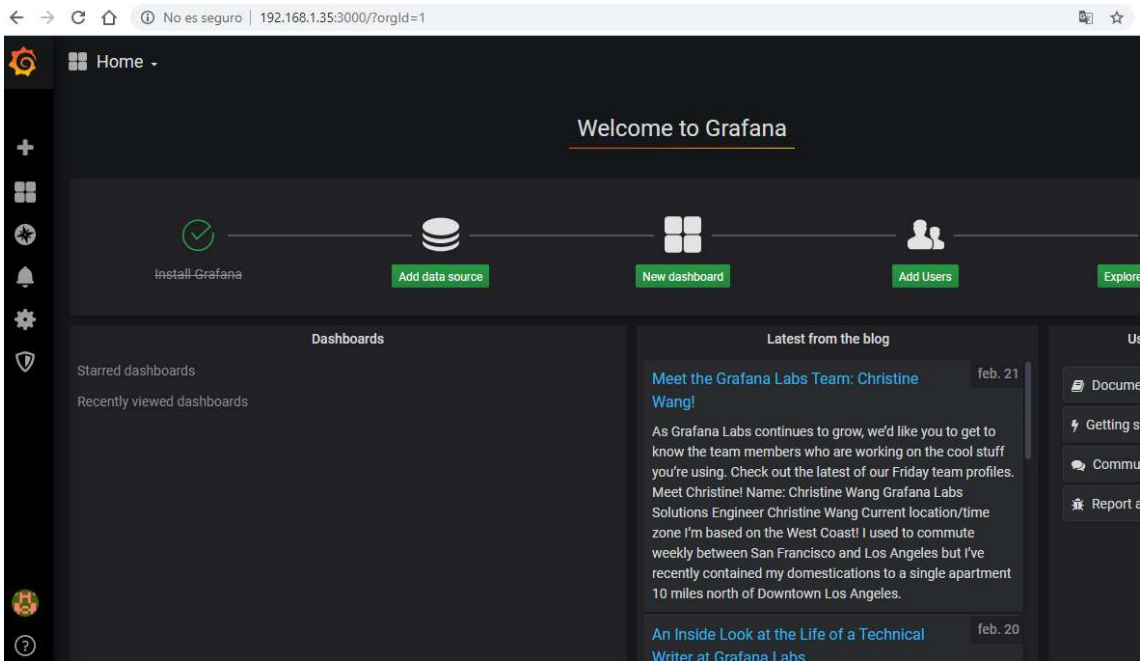


Imagen 17: menú de inicio Grafana

Ahora pasamos a seleccionar la base datos con la que trabajará Grafana, en este caso Graphite.

Para ello se debe pinchar en *Add data source*, y nos aparecerá las siguientes opciones, donde tendremos que seleccionar *Graphite*.

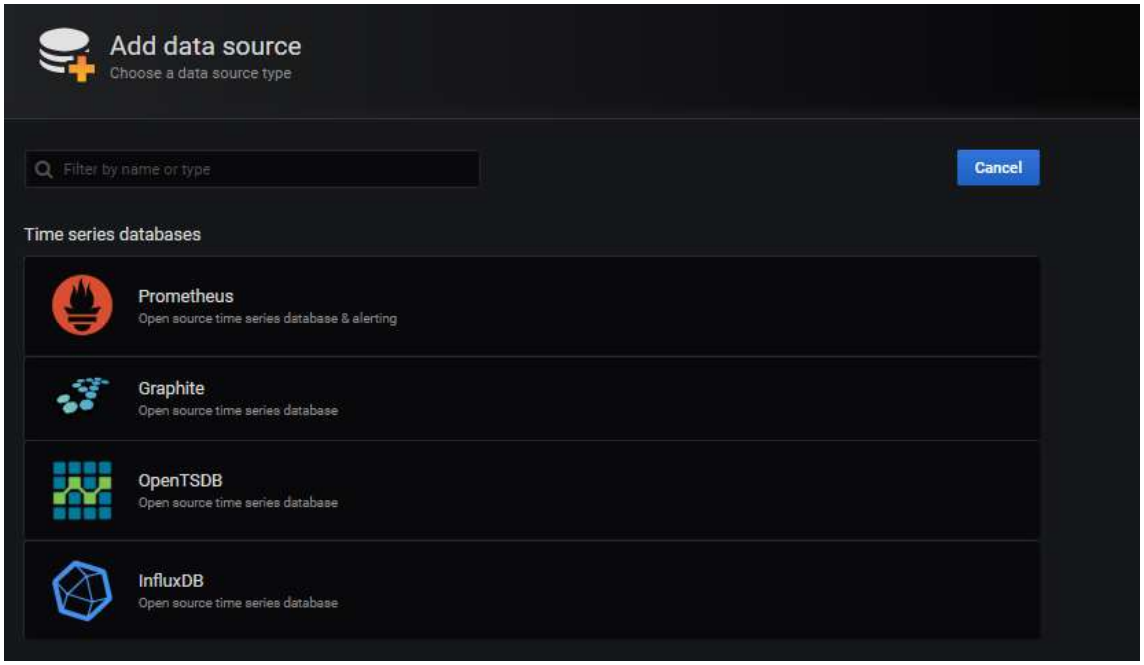


Imagen 18: menú de selección de base de datos de Grafana

Tras esto se debe de completar la información necesaria de la base de datos. En URL se debe de especificar la ruta de Graphite. Tras esto se pulsa en Save & Test y aparece un mensaje de confirmación.

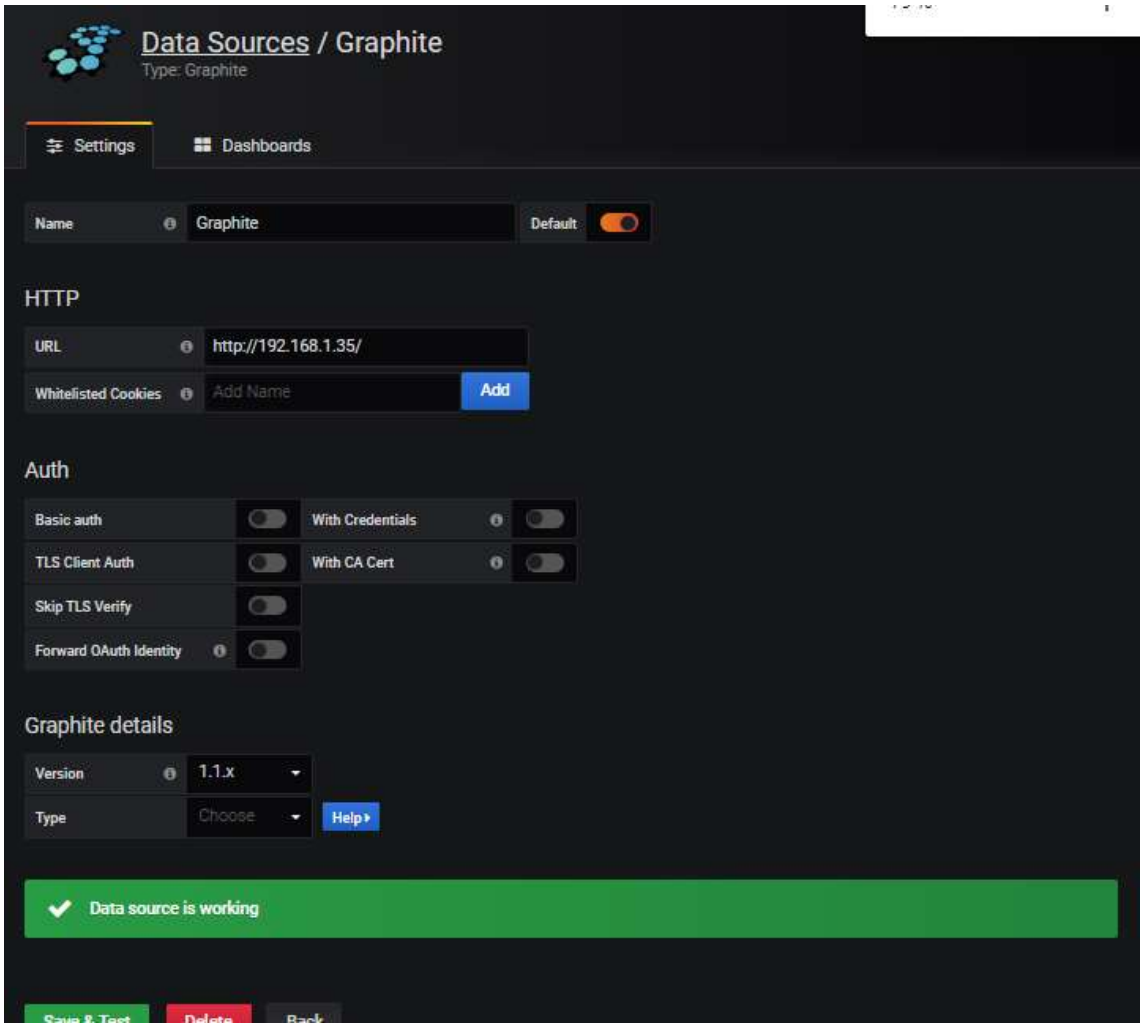


Imagen 19: configuración base de datos Grafana

Una vez que ya está configurada la base de datos y que accede a las métricas, se crea un dashboard.

Un dashboard es donde se visualizan las métricas que llegan a la herramienta. Este mostrará las métricas que estén seleccionadas, dotando al usuario de total control sobre la información mostrada y la manera en la que aparece.

Para ello se debe pinchar en *New Dashboard*. Tras esto nos aparecen las siguientes opciones:

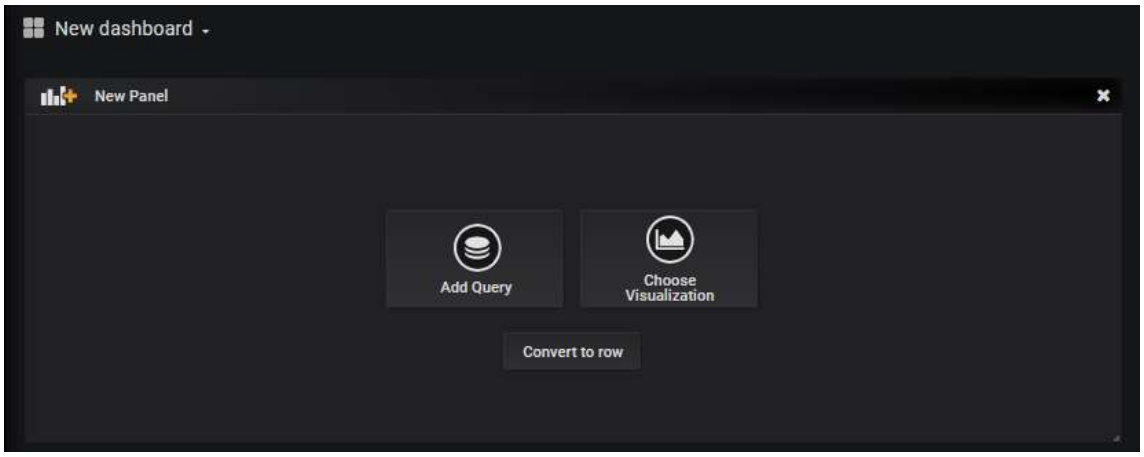


Imagen 20: panel de Dashboards de Grafana

Se debe pinchar en *Add Query*.

Imagen 21: add Query de Dashboard en Grafana

La ventana que nos aparece es la siguiente. En ella se aprecia que en la parte de abajo es donde se deben de configurar las consultas.

En este caso, se debe cambiar la base de datos que esta por defecto (default) a Graphite, para poder manejar las métricas de las que se dispone.

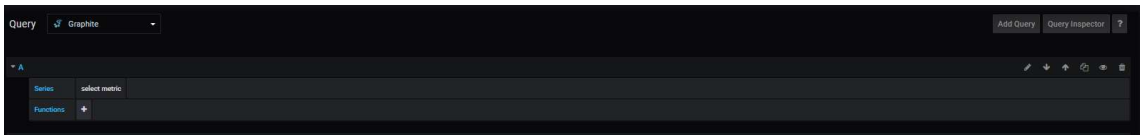


Imagen 22: query de Dashboard de Grafana

Ahora se debe de configurar la consulta A. al pinchar sobre *select metric* aparecen los recolectores que se tienen y seleccionando más opciones la métrica en cuestión que se quiera mostrar. Un ejemplo es el siguiente:

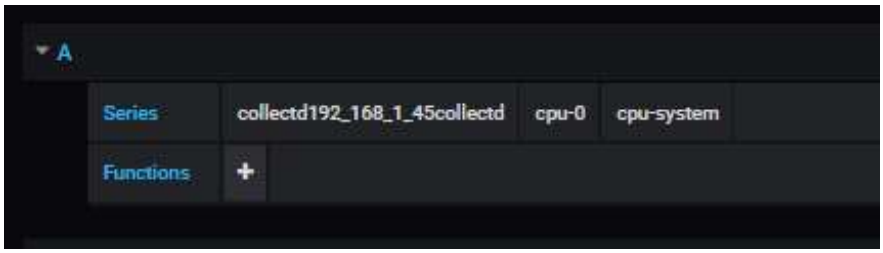


Imagen 23: configuración de query en Grafana

Tras esta consulta, aparecen los valores en el dashboard.

Imagen 24: proyección de query en Grafana

Tras esto guardamos el dashboard. Desde la página de inicio se puede hacer un seguimiento del panel:

Imagen 25: Dashboard en Grafana

3.6. Instalación Nagios en servidor

A continuación, se detallan los comandos utilizados para instalar Nagios, la tercera herramienta de monitorio del proyecto. Con este producto tendremos un control total sobre las maquinas a monitorear, ofreciendo avisos cuando las métricas alcancen valores determinados previamente fijados por el encargado.

El primer comando instala los paquetes necesarios para el software de Nagios Core.

En primer lugar, se debe instalar LAMP, herramienta para el alojamiento web. LAMP debe instalarse con Apache, MariaDB y PHP7:

```
yum install httpd mariadb-server php php-mysql
```

Tras esto aseguramos la instalación de MariaDB.

```
mysql_secure_installation
```

El último paso de relacionado con LAMP es habilitar MariaDB y Apache.

```
systemctl habilita httpd.service
systemctl enable mariadb.service
```

Lo siguiente es instalar otros paquetes necesarios para Nagios Core.

```
yum install gcc glibc glibc-common wget gd gd-devel perl postfix
```

Después se descarga e instala Nagios Core. Primero se cambia al directorio tmp:

```
cd /tmp
```

Dentro del directorio, se descarga el archivo .tar con el siguiente comando:

```
wget -O nagioscore.tar.gz
https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.2.tar.gz
```

Tras haberse completado la descarga del archivo, se extrae el archivo comprimido.

```
tar xzf nagioscore.tar.gz
```

Tras haber extraído el archivo, cambiamos al directorio de Nagios:

```
cd /tmp/nagioscore-nagios-4.4.2
```

Cuando ya se está en el directorio, se ejecuta el siguiente comando para configurar el instalador y preparar el código fuente de Nagios Core para la compilación:

```
./configure
```

Aparece un aviso de que se necesita el paquete unzip en la máquina, por lo que se instala con el siguiente comando:

```
yum install unzip
```

Se vuelve a ejecutar el configurador para ver que se ha resuelto el problema:

```
./configure
```

Tras haber completado la configuración, se compila Nagios Core.

```
make all
```

El siguiente paso es crear el usuario nagios y el grupo y agregarlos a Apache.

```
make install-groups-users
usermod -a -G nagios apache
```

Tras estos pasos se instala Nagios Core:

```
make install
```

Ahora se debe instalar el script de inicialización que se usa para administrar el servicio Nagios.

```
make install-daemoninit
```

Seguidamente se instalan los archivos de configuración de Nagios.

```
make install-config
```

Después se instala y configura el archivo de comando externo para que Nagios Core funcione desde la línea de comandos:

```
make install-commandmode
```

Otro de los archivos de configuración que se deben instalar es el de configuración del servicio web.

```
make install-webconf
```

Tras tenerlos todos instalados, se reinicia el servicio Apache.

```
systemctl restart httpd
```

El siguiente paso es crear una cuenta de usuario de Apache para iniciar sesión en Nagios. Se crea una cuenta de usuario llamada nagiosadmin y se le asigna una contraseña.

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

Tras este paso la instalación principal de Nagios Core está completada.

Para que Nagios Core funcione correctamente se deben instalar una serie de complementos con los comandos que se muestran a continuación:

Antes de descargar e instalar los complementos se debe asegurar que la máquina cuenta con una serie de paquetes.

```
yum install gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-snmp net-snmp-utils epel-release perl-Net-SNMP
```

Ahora se pasa a descargar los complementos. Para ello se cambia al directorio tmp.

```
cd /tmp
```

Se descargan los complementos de Nagios Core.

```
wget --no-check-certificate -O nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
```

Una vez descargados se extrae el archivo comprimido.

```
tar zxf nagios-plugins.tar.gz
```

Cuando ya se tiene extraído el archivo de complementos de Nagios Core se para a realizar el último paso de la instalación, compilar los complementos en el servidor.

```
cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
make
make install
```

Tras todo esto Nagios Core está listo para ejecutarse. Para acceder a su interfaz web se debe poner la IP/nagios en el navegador y rellenar con el nombre de usuario (nagiosadmin) y la contraseña.

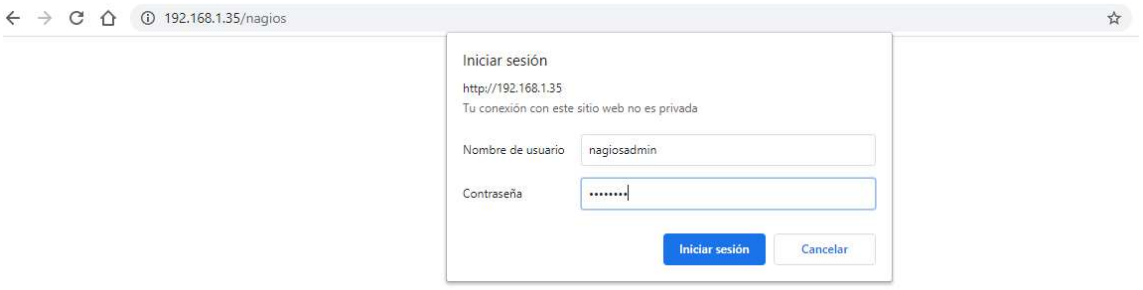


Imagen 26: loggin Nagios

Después de introducir las credenciales, aparece la interfaz web de Nagios.

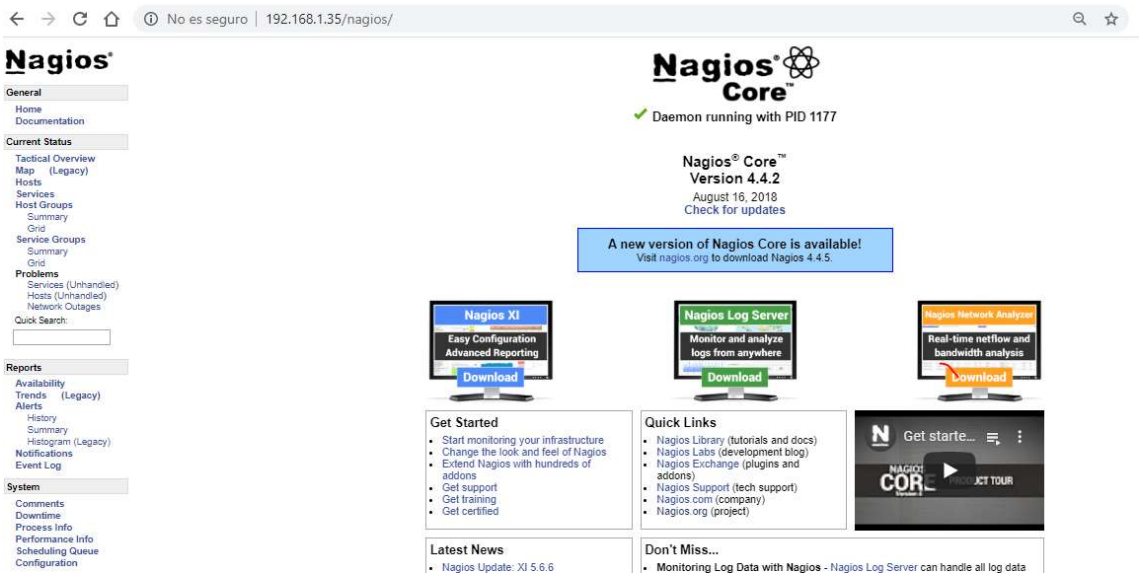


Imagen 27: menú inicio Nagios

3.7. Instalación NRPE en cliente y servidor

Para poder controlar las máquinas clientes desde Nagios es necesario que este programa reciba las métricas de los clientes. Para ello se deben instalar diferentes paquetes, tanto en el servidor como en el cliente.

Por la parte del servidor el sistema debe contar con:

- Los plugins de Nagios
- El plugin NRPE

Por la parte del cliente, la máquina deberá de tener:

- Los plugins de Nagios
- El servicio Nagios NRPE

Nagios Remote Plugin Executor (NRPE) es un agente utilizado por Nagios XI para establecer comunicación con hosts remotos. El funcionamiento es simple: el servidor con Nagios XI envía solicitudes al agente que está en los hosts remotos. Estos agentes siempre están esperando solicitudes por parte del servidor.

En la máquina servidor se deben de seguir los siguientes pasos:

En primer lugar, se debe habilitar el repositorio EPEL.

```
rpm -Uvh https://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release-7-12.noarch.rpm
```

Tras esto se pasa a instalar los plugins de Nagios.

```
yum install -y nagios-plugins-all
```

Una vez descargados e instalado el paquete y sus respectivas dependencias se tendrá un nuevo directorio /usr/lib64/nagios/plugins/ donde se encuentran todos los comandos ejecutables de los plugins.

```
check_breeze    check_game      check_mrtgtraf     check_overcr   check_swap
check_by_ssh    check_hpjd      check_mysql        check_pgsql    check_tcp
check_clamd     check_http      check_mysql_query  check_ping     check_time
check_cluster   check_icmp      check_nagios       check_pop      check_udp
check_dhcp      check_ide_smart check_nntp         check_procs    check_ups
check_dig       check_imap      check_nntps        check_real     check_users
check_disk      check_ircd      check_nrpe         check_rpc      check_wave
check_disk_smb  check_jabber    check_nt           check_sensors  eventhandlers
check_dns       check_ldap      check_ntp          check_simap    negate
check_dummy     check_ldaps     check_ntp_peer     check_smtp     urlize
check_file_age  check_load      check_ntp.pl       check_snmp     utils.pm
check_flexlm    check_log       check_ntp_time     check_spop     utils.sh
check_fping     check_mailq     check_nwstat       check_ssh
check_ftp       check_mrtg      check_oracle       check_ssmtp
```

Imagen 28: pluggins Nagios

El siguiente paso es instalar el paquete nagios-plugins-nrpe, que contiene el plugin NRPE que permite conectar el servidor Nagios con el servicio NRPE de las máquinas remotas.

```
sudo yum install -y nagios-plugins-nrpe
```

Tras esto se comprueba que la conectividad con la máquina remota con el siguiente comando:

```
/usr/lib64/nagios/plugins/check_nrpe -H 192.168.1.45
```

El resultado que aparece es satisfactorio si aparece la versión NRPE del cliente.

```
[root@localhost ~]# /usr/lib64/nagios/plugins/check_nrpe -H 192.168.1.4
5
NRPE v3.2.1
```

Imagen 29: prueba de check_nrpe a cliente 1

Tras esto se debe crear un comando para que Nagios pueda usar este plugin, que le permitirá hacer peticiones a los clientes.

```
vi /etc/nagios/objects/commands.cfg
```

Dentro del archivo debe introducirse la siguiente información:

```
define command {
        command_name check_nrpe
        command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Ahora se debe configurar Nagios para que cargue las configuraciones de las distintas máquinas, para lo que se habilitará un directorio donde se guardará un archivo de configuración para cada máquina remota.

Se edita el archivo principal de configuración de Nagios. Para ello se debe entrar en el archivo:

```
vi /etc/nagios/nagios.cfg
```

Una vez dentro, se busca la línea cfg_dir y se descomenta:

```
cfg_dir=/etc/nagios/servers
```

Esta ruta será donde se deben guardar las configuraciones de las máquinas remotas. Inicialmente esta ruta no existe, por lo que debe crearse.

```
mkdir / etc/nagios/servers/
```

Dentro de este directorio se debe de crear el archivo de configuración de la máquina remota.

```
vi /etc/nagios/servers/192.168.1.45.cfg
```

En este archivo se debe de introducir la siguiente información para definir el host:

```
define host {
        use linux-server
        host_name 192.168.1.45
        alias CentOS 7 - 2
        address 192.168.1.45
        max_check_attempts 5
        check_period 24x7
        notification_interval 30
        notification_period 24x7
}
```

- ➔ max_check_attempts 5: comprobar el host 5 veces como máximo.
- ➔ check_preriod 24x7: de forma predeterminada, los hosts se comprueban durante todo el día.
- ➔ notification_interval 30: reenviar notificaciones cada 30 minutos.
- ➔ notification_period 24x7: periodo de notificación, durante todo el día.

Aquí se detalla información sobre la maquina cliente, como el nombre con el que va a aparecer en Nagios, la dirección IP y los periodos e intervalos de chequeo.

Por la parte del cliente se deben seguirse los siguientes pasos:

En primer lugar, al igual que en la máquina servidor, se debe habilitar el repositorio EPEL.

```
rpm -Uvh https://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release-7-12.noarch.rpm
```

Después de habilitar el repositorio se instala NRPE y complementos en el sistema.

```
yum --enablerepo=epel -y install nrpe nagios-plugins
```

El siguiente paso es instalar comandos que NRPE ejecutará para monitorear las máquinas clientes.

```
yum --enablerepo=epel -y list nagios-plugins*
```

Tras esto se pasa a configurar el agente NRPE. Para ello se debe editar el archivo de configuración de NRPE.

```
vi /etc/nagios/nrpe.cfg
```

El cambio que se debe realizar en este archivo es para permitir que la máquina servidor pueda conectar con el NRPE del cliente. Para ello debemos introducir la IP del servidor en la parte de hosts permitidos.

```
allowed_hosts=127.0.0.1,::1,192.168.1.41
```

Tras esto recargamos los nuevos ajustes para que se ejecute de manera correcta.

```
systemctl reload nrpe
```

Normalmente el firewall de CentOS 7 está activado por defecto, por lo que bloqueará las conexiones entrantes al servicio NRPE, por lo que se deben permitir estas conexiones.

```
sudo firewall-cmd --permanent --add-service=nrpe
```

Se aplican los cambios recargado la configuración del firewall:

```
sudo firewall-cmd --reload
```

Llegado a este punto ya está todo listo para que al activar comandos en las máquinas clientes se produzca la monitorización, apareciendo las alertas en el panel de control de Nagios. En los siguientes apartados se detallará como se deben de implementar los comandos y activar las alertas.

3.8. Activación e implementación de métricas y alertas

3.8.1. Alerta carga CPU

Se ha implementado una métrica para controlar la carga de CPU del cliente. Para ello se utiliza el comando check_load.

Este comando está acompañado de 6 valores. Los 3 primeros argumentos son el umbral de advertencia para cargas superiores a 1, 5 y 15 minutos respectivamente, y los 3 últimos son los umbrales críticos para 1, 5 y 15 minutos.

El archivo donde se debe de definir el comando es /etc/nagios/nrpe.cfg/ quedando de la siguiente manera:

```
command[check_load]=/usr/lib64/nagios/plugins/check_load -r -w .6,.5,.4 -c .8,.7,.6
```

El argumento -r se utiliza para dividir los promedios de carga entre el número de CPU de la máquina (en este caso solo hay una CPU). Los valores tienen el siguiente significado:

- Advertencia si está al 60% durante 1 minuto
- Advertencia si está al 50% durante 5 minutos
- Advertencia si está al 40% durante 10 minutos
- Warning si está al 80% durante 1 minuto
- Warning si está al 70% durante 5 minuto
- Warning si está al 60% durante 10 minuto

Esta línea está en el archivo de configuración de NRPE del cliente. De esta manera quedan aquí definidos cada uno de los comandos que actúan sobre la máquina.

Por parte del servidor se debe de definir el plugin en el archivo de configuración de la máquina remota (/etc/nagios/servers/192.168.1.45.cfg).

```
define service {
        use generic-service
        host_name 192.168.1.45
        service_description Carga de CPU
        check_command check_nrpe!check_load
}
```

Tras esto ya Nagios puede controlar la carga de CPU del cliente. Tras reiniciar NRPE en la máquina cliente y Nagios en el servidor aparece la alerta de la métrica.

3.8.2. Alerta espacio libre disco

Se ha implementado una métrica para controlar el espacio libre del disco del cliente.

La estructura del comando es la siguiente: 2 valores que determinan los valores de advertencia. Con el primero de ellos, 30%, se remitirá una alerta de advertencia si el espacio libre es menor al valor.

El segundo valor, 10%, indica que si el espacio libre es menor a este porcentaje se creará una alerta crítica.

El archivo donde se debe de definir el comando es /etc/nagios/nrpe.cfg.

```
command[check_sda]=/usr/lib64/nagios/plugins/check_disk -w 30% -c 10% -p
/dev/mapper/centos-root
```

Al final del comando se debe de especificar el disco que se quiere monitorizar, en este caso es /dev/mapper/centos-root.

Se debe proceder de igual forma que la alerta anterior, por lo que se define el servicio en el archivo de configuración del cliente.

```
define service {
        use generic-service
        host_name 192.168.1.45
        service_description Espacio libre en /dev/mapper/centos-root
        check_command check_nrpe!check_sda
}
```

Tras esto ya aparece el servicio en Nagios.

3.8.3. Alerta http

Este comando ha sido utilizado para verificar el estado del servidor HTTP que se está ejecutando en el cliente 2.

La url de la página de GitLab es la IP de la máquina: http://192.168.1.55

Para ello se define el comando en el archivo de configuración de NRPE del cliente.

```
command[check_http]=/usr/lib64/nagios/plugins/check_http -H 192.168.1.55
```

Tras esto, en el archivo de configuración de la máquina cliente, ubicado en el servidor, donde se describen los servicios disponibles se define el servicio:

```
define service {
        use generic-service
        host_name 192.168.1.55
        service_description Estado web GitLab
        check_command check_nrpe!check_http
}
```

Por último, se reinician los servicios tanto Nagios en el servidor como NRPE en el cliente y aparecerá la alerta respectiva en el panel de control de Nagios.

3.9. Activación de notificaciones por correo

Los últimos pasos que quedan por dar para tener listo el sistema de monitorización están referidos al envió de notificaciones de las alertas de Nagios.

De esta manera, el sistema se encarga de enviar correos avisando de incidencias en cualquiera de las máquinas clientes que están siendo monitorizadas.

Para la configuración de esta utilidad se necesita instalar un programa que se encargue de enviar estos correos. Se ha elegido mailx, una versión mejorada del programa mail.

El primer paso para tener operativo es instalar el paquete:

```
yum install -y mailx
```

Tras esto se debe configurar el servidor SMTP para realizar envíos, y usar el de Google. Para ello se edita el archivo de configuración de mailx (etc/mail.rc), añadiendo las siguientes líneas:

```
set smtp=smtps://smtp.gmail.com:465
set smtp-auth=login
set smtp-auth-user=usernagios01@gmail.com
set smtp-auth-password=****************
set ssl-verify=ignore
set nss-config-dir=/etc/pki/nssdb/
```

Para la contraseña de la cuenta se debe crear una específica para la aplicación. Para ello se va “Gestionar tu cuenta de Google” y se busca “Contraseña de aplicaciones”. Ahí se crea una, seleccionando como aplicación correo y de dispositivo se pone otro, en nuestro caso Nagios, tras esto aparece la contraseña.

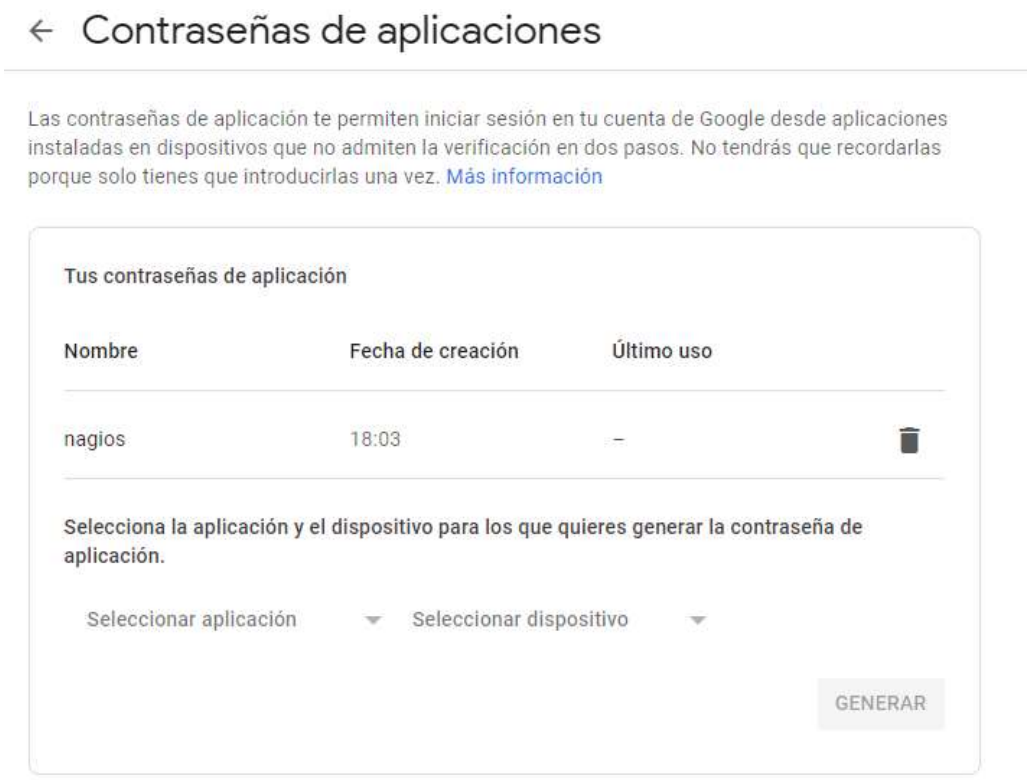


Imagen 30: contraseña para aplicación mailx (Nagios)

Tras esto se debe configurar Nagios para que mande correos cuando haya alertas en los clientes.

El primer paso es editar el archivo de configuración los contactos, ubicado en la ruta: /etc/nagios/objects/contacts.cfg. Aquí se debe de editar en primer lugar el apartado contacts:

```
define contact {

    contact_name            nagiosadmin
    use                     generic-contact
    alias                   Nagios Admin
    email                   usernagios01@gmail.com
    service_notification_period             24x7
    service_notification_options            w,u,c,r,f,s
    service_notification_commands           notify-service-by-email
    host_notification_period                24x7
    host_notification_options               d,u,r,f,s
    host_notification_commands              notify-host-by-email
}
```

Tras esto se debe de editar, en el mismo archivo, el apartado que define el grupo:

```
define contactgroup {

    contactgroup_name       admins
    alias                   Nagios Administrators
    members                 nagiosadmin
}
```

El último paso es editar los comandos que mandan las notificaciones para indicarles el programa que se encargara de mandar estos mensajes. El archivo que se debe de editar es /etc/nagios/objects/commands.cfg.

El primer comando que se debe de configurar es el notify-host-by-email, quedando de la siguiente manera:

```
define command {

    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo:
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mailx -s "**
$NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **"
$CONTACTEMAIL$
}
```

El segundo que se debe editar es el comando notify-service-by-email, quedando de la siguiente manera:

```
define command {

    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService:
$SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$\n" | /usr/bin/mailx -s "**
$NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is
$SERVICESTATE$ **" $CONTACTEMAIL$
}
```

Tras estos pasos el envío de correos referidos a alertas esta activo.

4. VALIDACIÓN DEL DISEÑO

Este apartado tiene la función de otorgar validez a la solución propuesta en este trabajo de Fin de Grado. Para ello, se realizarán pruebas de uso de la herramienta software y se estudiarán los resultados obtenidos.

4.1. INTRODUCCIÓN

El diseño de la solución junto con su posterior implementación da como resultado un sistema real de monitorización que debe ser probado para ver en qué medida quedan controlados los diferentes clientes que componen el entorno diseñado.

No serán objeto de este capítulo ninguna de las pruebas realizadas anteriormente durante el desarrollo de la solución, ya que su finalidad eran comprobar el correcto funcionamiento y no tenían el objetivo de evaluar la validez de la solución propuesta.

Para explicar la validación del diseño mediante las pruebas, se explicará previamente la metodología empleada en cada caso y cuál sería el flujo de acciones que deberían de surgir. Al final del caso se expondrá el resultado real obtenido para ver si se ajusta con lo que se esperaba.

4.2. METODOLOGÍA

Para la realización de las pruebas sobre la solución implementada se han diseñado varios escenarios para cubrir ampliamente las diferentes situaciones que pueden darse en la monitorización de los clientes, y así tener una visión global del sistema. Las pruebas que se ejecutarán se basarán en simular entornos de ejecución en las que ocurran situaciones anómalas y se traduzcan en respuestas por parte del sistema de monitorización, activando alertas y demás mecanismos de control del funcionamiento de los clientes.

Cada prueba vendrá acompañada de una descripción de la situación concreta del sistema. Algunos ejemplos expondrán valores de referencia que harán comprender mejor el funcionamiento de la herramienta, explicando los umbrales con los que se trabaja en cada situación y que se deberán traspasar para comprobar cómo funciona la monitorización.

4.3 Caso de prueba 1

4.3.1. Descripción

En este primer caso se pretende probar la conexión con la máquina cliente. De esta manera vamos a quitar el servicio de esta máquina, haciendo imposible su monitorización.

4.3.2. Resultado esperado

En esta situación Nagios tiene que informar de que la máquina cliente (con IP 192.168.1.45) se encuentra caída. Esto se tendría que enviar al administrador de Nagios en forma de mensaje dando detalles de la alerta.

Además, en el panel de control de Nagios deberían aparecer las métricas de esta máquina en estado crítico, informando de que no se puede realizar la conexión para obtener información las mismas.

4.3.3. Resultado obtenido

Como se ha especificado en el anterior punto, lo primordial es que Nagios informe vía correo de que la máquina cliente se encuentra caída, tal y como ha ocurrido.

Imagen 31: correo aviso cliente 1 fuera de servicio

Otra acción que debe de ocurrir, al estar la máquina cliente apagada, es que no se puedan monitorizar sus alertas, al no llegar métricas de estado. Debido a esto las métricas referidas al cliente aparecen en estado crítico e informan de que no se puede establecer la conexión.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
192.168.1.45	Carga de CPU	CRITICAL	12-01-2020 17:53:11	0d 0h 10m 38s	3/3	(No output on stdout) stderr: connect to address 192.168.1.45 port 5666: No route to host
	Espacio libre en /dev/sda1	CRITICAL	12-01-2020 17:45:13	0d 0h 8m 26s	3/3	(No output on stdout) stderr: connect to address 192.168.1.45 port 5666: No route to host

Imagen 32: estados cliente 1 en panel de control de Nagios

4.3.4. Conclusión del caso

Se ha podido ver que el resultado obtenido es justo lo detallado anteriormente en el resultado esperado, por lo que el sistema de monitorización cumple con las expectativas. Además de cumplir, la información que detalla en el momento de la alerta es bastante completa, informando desde la IP de la máquina hasta la fecha y hora concretas en las que se ha producido en la alerta, por lo que el sistema es bastante completo.

4.4. Caso de prueba 2

4.4.1. Descripción

Este caso de prueba va ligado con la situación del caso de prueba 1. Ya que la máquina se encontraba apagada, en este pasamos a encender la máquina para dar servicio.

4.4.2. Resultado esperado

Se espera que, tras este cambio en el estado de la máquina, el sistema informe por correo que la incidencia ha sido resuelta y la máquina se encuentra otra vez operativa. De esta manera, las monitorizaciones de la máquina ya no informarían de que no se puede establecer conexión en su estado.

4.4.3. Resultado obtenido

Una vez que se ha encendido la máquina, Nagios informa vía correo de que la máquina se encuentra operativa.

Imagen 33: correo restablecimiento de servicio del cliente 1

Tras esto, en el panel de control de Nagios ya empiezan a funcionar las métricas, ya que les llega información del estado de la máquina.

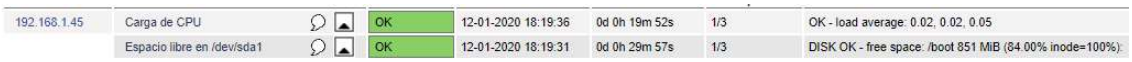


Imagen 34: estados de cliente 1 tras restablecer servicio

4.4.4. Conclusión del caso

El resultado obtenido cumple en su totalidad con lo especificado en el resultado esperado. La acción de avisar por parte de Nagios de que la máquina está de nuevo operativa es de especial interés ya que de esta forma avisa de que las métricas vuelven de nuevo a funcionar, comprobando el estado del equipo al volver a dar servicio.

4.5. Caso de prueba 3

4.5.1. Descripción

En este caso se quiere probar la monitorización que realiza el sistema sobre la carga de CPU de la máquina cliente.

Para ello se va a someter a la máquina a diferentes tareas para variar el uso de CPU, pasando por diferentes estados. En vez de llegar a un estado crítico de la métrica, obtenemos más información visualizando una gráfica que muestre los valores de carga de CPU de la máquina durante un periodo de tiempo. Para ello vamos a la interfaz de Graphite. Esta vez cambiaremos el modo de monitorización realizado anteriormente, en el que prestábamos atención a Nagios, y ampliaremos el control vigilando las gráficas que nos proporciona Graphite.

4.5.2. Resultado esperado

Este caso, al no tener muchos procesos corriendo en la máquina cliente, los avisos por parte de Nagios no son de interés ya que no pretendemos llegar a situaciones tan limites en los equipos, ya que serían situaciones muy raras y difíciles de que ocurrieran.

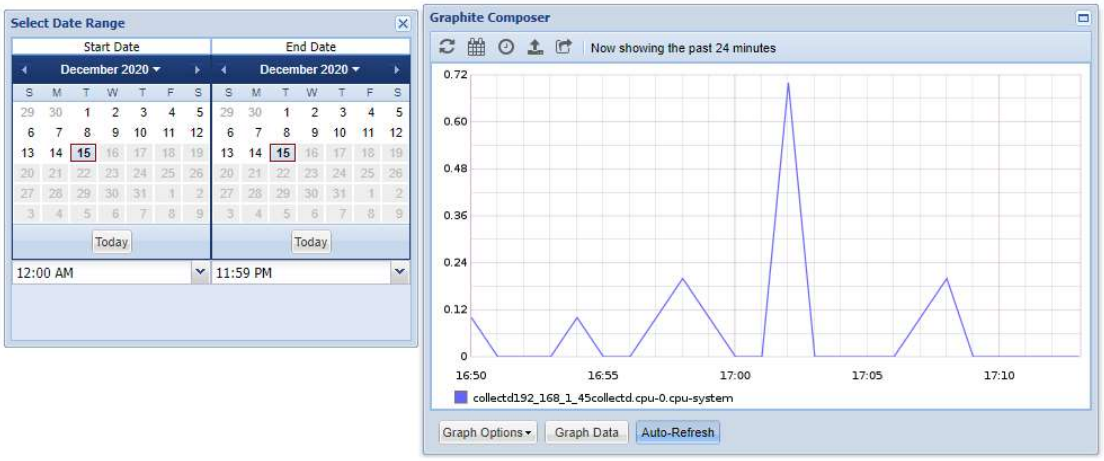
Lo que se pretende es contemplar los diferentes valores por los que pasa la máquina, concretamente el uso de la CPU, a lo largo del tiempo de ejecución del cliente. Se espera que la gráfica varíe sus valores conforme pasa el tiempo. Hay dos métricas de especial interés en este caso: cpu_system y cpu_idle[9], complementarias la una de la otra. La primera, cpu_system, es la carga de la CPU, es decir, la cantidad de CPU que están usando los procesos. La segunda, es la cantidad de carga que está consumiendo el proceso inactivo, es decir, el porcentaje de CPU que está libre. De esta manera, la suma del primero y del segundo prácticamente deben dar el 100% de la CPU. Si esto se da así, significará que el monitoreo se realiza de manera correcta.

4.5.3. Resultado obtenido

Como queda señalado en el apartado anterior, vamos a trabajar con dos métricas para determinar la validez de la monitorización.

Para ello vamos a mostrar una gráfica en Graphite que refleje la métrica cpu_system:

[9] https://www.howtogeek.com/411569/what-is-system-idle-process-and-why-is-it-using-so-much-cpu/



Tras esto se pasa a mostrar la gráfica que recoge los datos de la segunda métrica, cpu_idle:

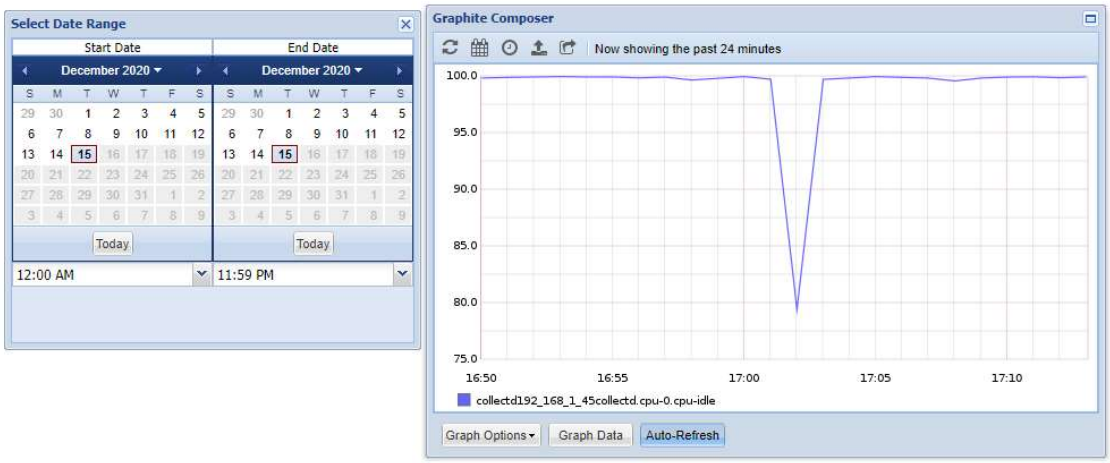


Imagen 35: gráfica en Graphite-Web (cpu-idle)

Como se puede apreciar, la primera se complementa con la segunda, verificando el correcto funcionamiento del sistema de monitorización.

4.5.4. Conclusión del caso

Como hemos visto en las gráficas, los valores arrojados por la primera y por la segunda son complementarios, por lo que la monitorización se realiza correctamente. En este caso, es más útil llevar la monitorización de esta forma, controlando de manera continua los datos y no con avisos como en los casos de prueba anteriores.

Esta monitorización aun así esta complementada por Nagios, es decir, en caso de anomalías en el rendimiento de las máquinas, se enviarían alertas como en los casos de prueba anteriores, pero serían situaciones muy excepcionales que no se dan con normalidad.

4.6. Caso de prueba 4

4.6.1. Descripción

En este caso de prueba se pretende trabajar con la métrica que el espacio libre del disco que contiene el directorio raíz de CentOS7. El almacenamiento de la máquina cliente es de 8GB y, el almacenamiento de este directorio es de 6,2GB.

Para ello se utiliza el comando check_sda, que informa con un "warning" si el almacenamiento libre es menor al 30%, y con un "critical" si es menos del 10%.

Se van a realizar la prueba dos veces. La primera solo tendrá un archivo de 3GB, que dejará libre el 27% del almacenamiento del directorio. La segunda prueba tendrá este primer archivo y otro de 1,3GB, que ocuparán el 92% del directorio, dejando libre únicamente el 8%.

4.6.2. Resultado esperado

Se espera que ante las dos situaciones el sistema de monitorización avise en el panel de control de Nagios de la incidencia, (para el primer caso en forma de "warning", y para el segundo "critical"). Además, se deberá de enviar un correo informando del estado del disco en las dos situaciones.

4.6.3. Resultado obtenido

El primer paso va a ser crear el primer archivo pesado en la máquina, el de 3GB, utilizando el siguiente comando:

```
sudo fallocate -l 1G /archivogrande
```

De esta forma el almacenamiento del directorio raíz, que tiene un espacio total de 6,2GB tiene ocupados 4,5GB y 1,8GB libres. Esto supone que tiene el 73% ocupado.

```
/dev/mapper/centos-root   6,2G   4,5G  1,8G  73% /
```

Imagen 36: estado disco / dev/mapper/centos-root

Como se tiene menos de un 30% libre, el panel de control de Nagios informa con un "warning" de la situación de la máquina cliente:

Imagen 37: estado almacenamiento cliente 1 en Nagios

Además, se manda un correo avisando de la incidencia:

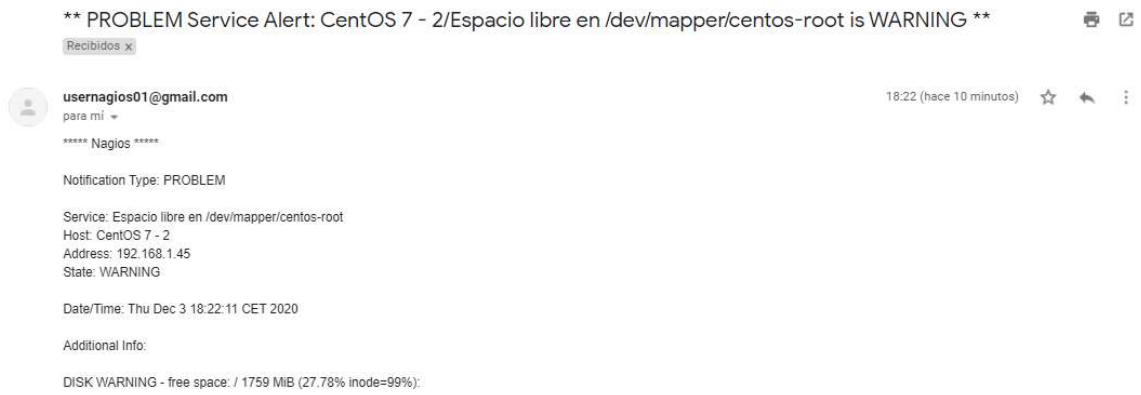


Imagen 38: mensaje alerta estado disco del cliente 1

Para el segundo caso, lo que hacemos es crear otro archivo en la máquina, que sumado con el primero, ocuparan el 92% del disco.

```
sudo fallocate -l 1300MB /archivogrande2
```

En este caso, como en el anterior, además de señalar el estado crítico en el panel de control de Nagios, se manda un correo avisando de la incidencia:

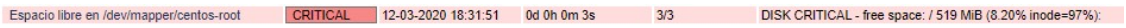


Imagen 39: estado almacenamiento cliente 1 en Nagios (II)

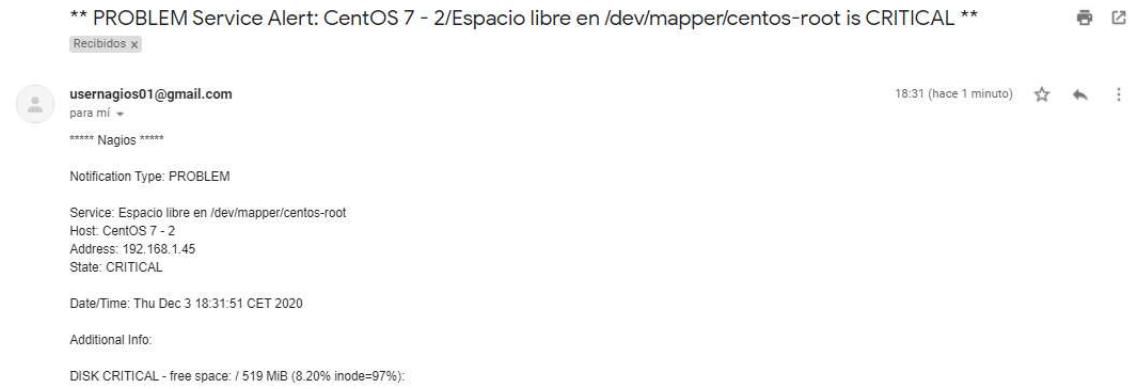


Imagen 40: mensaje alerta estado disco del cliente 1 (II)

Una vez que se eliminan los archivos y el almacenamiento vuelve a liberarse, el sistema manda un mensaje informando del cambio de estado de la alerta:

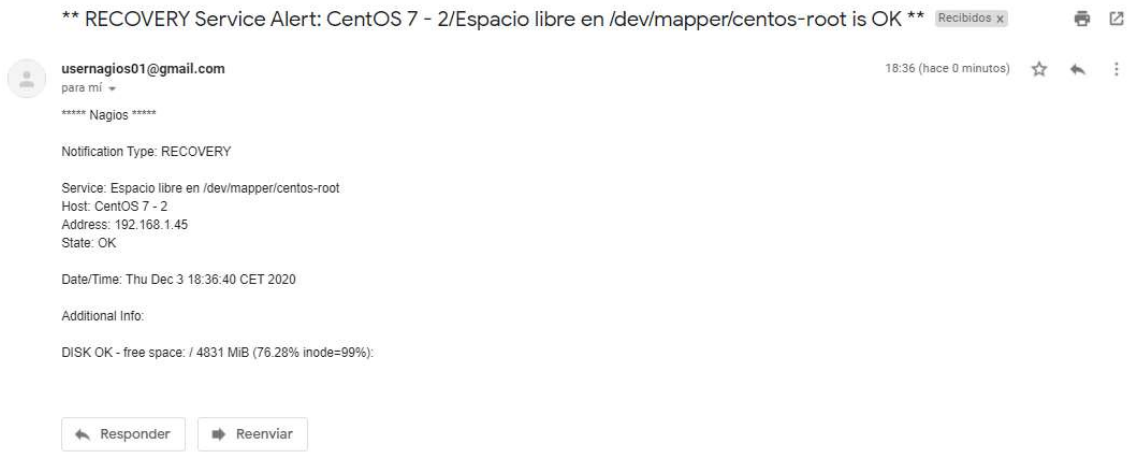


Imagen 41: mensaje almacenamiento disco cliente 1

4.6.4. Conclusión del caso

Como se aprecia en los dos puntos anteriores, las acciones que se detallaban en el resultado esperado se han dado, realizándose la tarea de monitorización de manera correcta.

4.7. Caso de prueba 5

4.7.1. Descripción

Este último caso de prueba se realiza en una máquina que tiene instalado el programa GitLab, un servicio web de control de versiones y desarrollo colaborativo basado en Git, muy utilizado en ámbitos tecnológicos.

Este caso de prueba consiste en chequear el acceso a esta herramienta, a través de una url, para poder trabajar de manera correcta. Para ello se detendrá el servicio en un momento concreto para probar la monitorización del servicio.

4.7.2. Resultado esperado

Mientras la web de la herramienta GitLab este accesible la monitorización no debe avisar de ningún contratiempo.

Cuando se detenga el servicio de GitLab mediante su comando de control el panel de control de Nagios deberá de informar en la alerta que monitoriza este servicio con un “Critical”, ya que el servicio estará caído. Además, deberá de mandarse un correo al administrador informando de la interrupción del servicio GitLab.

4.7.3. Resultado obtenido

En un primer momento, y en un entorno normal de servicio, la herramienta GitLab se encuentra accesible desde la url:

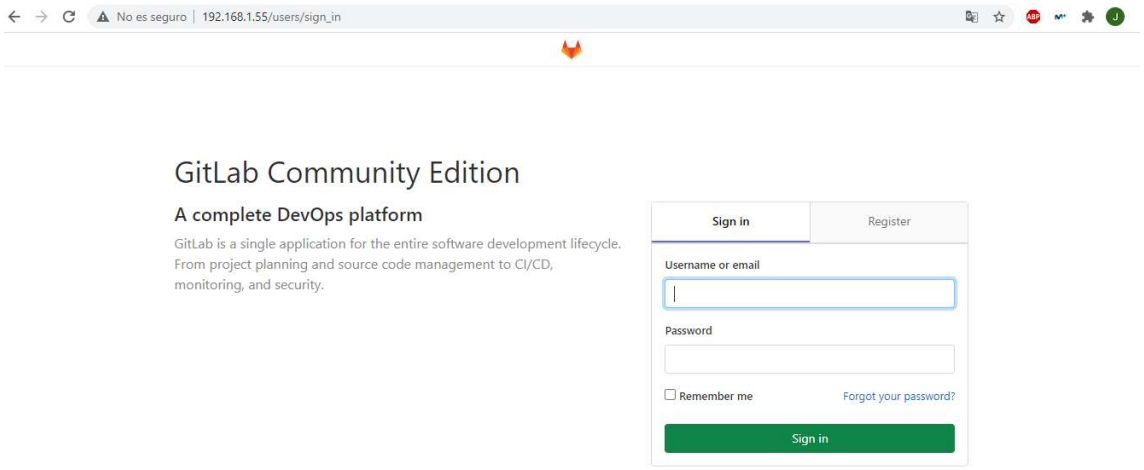


Imagen 42: loggin GitLab

Tras comprobar que todo se encuentra correctamente, se para el servicio de GitLab desde la consola con el siguiente comando:

```
gitlab-ctl stop
```

Tras parar el servicio salta la alerta en el panel de control de Nagios y de manera inmediata se manda el correo avisando de la incidencia:

Imagen 43: estado de http de GitLab en cliente 2

Imagen 44: correo alerta servicio http en cliente 2

5. MEJORAS PROPUESTAS

Después de analizar en profundidad la solución del sistema de monitorización propuesto en los apartados de este Trabajo de Fin de Grado, se han seleccionado una serie de mejoras aplicables al sistema. Algunas de ellas están centradas en la seguridad del sistema y otras, en la versión de productos/sistema operativo.

5.1. Mejora de seguridad del sistema

Por la parte que corresponde a la seguridad del sistema, sería recomendable cifrar el envío de datos de las páginas web de las herramientas implantadas. Desde un punto de vista técnico, la información que se maneja y aparece en las interfaces gráficas de Graphite, Grafana y Nagios es únicamente datos y métricas del comportamiento de las máquinas, y no se puede interferir en el funcionamiento de estas, pero sería recomendable cifrarlo para aportar así un plus de seguridad y robustez frente a ataques o acciones no deseadas por parte de terceros.

Para ello se debe cambiar el protocolo de transferencia de hipertexto de las páginas web que se usan en la solución propuesta, pasando de ser HTTP a HTTPS.

El protocolo https usa una conexión segura mediante un cifrado SSL[10], protocolo de cifrado ampliamente utilizado para garantizar la seguridad de las comunicaciones, haciendo que los datos viajen de un lugar a otro de una forma segura. La manera en la que se trabajaría es la siguiente:

1) El navegador intenta conectarse a un sitio protegido con SSL.
2) El navegador solicita que el servidor web se identifique.
3) El servidor envía una copia de su certificado SSL al navegador.
4) El navegador comprueba si el certificado SSL es de confianza. Si es así, envía un mensaje al servidor.
5) El servidor responde con un acuse de recibo firmado digitalmente para comenzar una sesión SSL.
6) Los datos cifrados se comparten entre navegador y servidor.

Los datos enviados que usan HTTPS están asegurados por el protocolo TLS (Transport Layer Security), que ofrece tres capas protectoras:

- Cifrado: el cifrado de los datos intercambiados los mantiene seguros. Esto se traduce en que mientras el usuario está navegando por un sitio web, nadie puede interceptar el contenido de sus peticiones y comunicaciones.
- Integridad de los datos: los datos no pueden ser modificados o dañados durante la transferencia sin ser detectado.

[10] https://www.redalia.com/ssl/ssl-protocol/

- Autenticación: demuestra que los usuarios están realizando la comunicación con la página web deseada, protegiendo de ataques y aumentando la confianza del usuario.

HTTPS está basado en un sistema que funciona sobre una clave pública y una clave privada. El administrador del servicio web debe crear un certificado de clave publica, el cual debe estar firmado por una autoridad de certificación.

La seguridad de las transacciones se basa en el intercambio de claves entre cliente y servidor. La clave pública y privada están relacionadas entre sí, haciendo imposible el uso de uno sin la otra. Para enviar un mensaje al servidor, se cifra con la clave pública y, mediante la clave privada, el servidor descifra el contenido. Estas claves públicas y privadas pueden obtenerse de manera gratuita, pero deben de renovarse cada cierto tiempo.

5.2. Mejora del Sistema Operativo

La siguiente mejora propuesta para la solución de monitorización es actualizar el sistema operativo sobre el que corre el producto a CentOS8. Una versión actualizada del sistema operativo mejorará tanto la experiencia del usuario como el funcionamiento de las máquinas, aparte de poder contar con las últimas características y optimizaciones.

Las características más destacadas de CentOS8 son:

- Extensión Red Hat Smart Management, administración de almacenamiento hibrido en la nube.
- Application Stream, framework y herramientas de desarrollo.
- Mejoras en la administración de sistemas, Windows y Linux, con Red Hat Enterprise System Roles.
- Soporte para los estándares OpenSSL 1.1.1 y TLS 1.3.
- Actualización de componentes básicos: Kernel más moderno (Linux 4.18).
- Wayland como servidor gráfico predeterminado.
- Gestor de paquetes dnf en vez de yum, proporcionando una mejor gestión del software.
- Arquitecturas soportadas: 64 bites Intel/AMD, 64 bites ARM e IBM Power PC.
- Versión 3 de Python por defecto.

Para obtener CentOS8 se debe de descargar de su página oficial en el apartado de descargas: http://isoredirect.centos.org/centos/8/isos/x86_64/

El fin de vida de CentOS7 es el 30 de junio de 2024. Después de esta fecha esta versión no recibirá mantenimiento de actualizaciones, por lo que es muy conviene tener la última versión y así alargar la vida útil del sistema de monitorización.

6. CONCLUSIONES

Tras haber presentado las herramientas que conformaran la solución software de monitorización y las pruebas que verifican la validez del producto, se puede tener una mejor idea del tipo de monitorización que realiza y las especificaciones y características que tiene finalmente el sistema.

Como se ha expuesto, la monitorización de las máquinas virtuales en las organizaciones, cada vez más dependientes de la tecnología, es cada vez más importante, siendo en muchas de ellas su activo más preciado. Partiendo de este punto, la amplitud de monitorización que ofrece un sistema es vital, ya que la tendencia es abarcar cada vez más métricas para tener un mayor control sobre los sistemas. Es aquí donde el software libre toma especial relevancia, ya que se sustenta sobre una gran comunidad que aporta nuevas funcionalidades y experiencias, todas ellas fundamentadas, ofreciendo de manera continua una mejor experiencia para el usuario y productos cada vez más perfeccionados.

Las herramientas elegidas en este proyecto, al ser OpenSource, permiten ser adaptadas a necesidades muy específicas, dando la posibilidad de crear métricas, gráficos y alertas según el proyecto que se quiera realizar y la demanda que se deba cubrir.

Todo esto expuesto anteriormente es recogido por el sistema que ha sido creado en este Trabajo de Fin de Grado: poder aprovechar la mayor cantidad de métricas, poder adaptarlo a las necesidades del proyecto en cuestión y personalizar cada herramienta aportando funcionalidades creadas por uno mismo.

De esta manera se tiene un producto muy atractivo y de especial valor, ya que la estructura de conexión entre las diferentes herramientas que forman el sistema es siempre la misma, independientemente del tipo de monitorización que se pretenda realizar y del número de máquinas a controlar. De esta forma, se tiene una base común que se puede implantar en cualquier tipo de organización de una forma rápida y sencilla, que después será adaptada a las características de la empresa al crear funcionalidades propias de cada proyecto.

7. BIBLIOGRAFÍA

[1] **Qué es la virtualización:** https://www.redhat.com/es/topics/virtualization/what-is-virtualization

[2] **Bases de datos de series temporales:** https://tienda.digital/2019/08/06/que-son-las-bases-de-datos-de-series-temporales/

[3] **Getting Started with Monitoring using Graphite:** https://www.infoq.com/articles/graphite-intro/

[4] **Instalación Graphite y Grafana CentOS7:** https://www.bujarra.com/instalando-graphite-grafana-visualizar-las-graficas-centreon/

[5] **Instalación Grafana CentOS7:** https://computingforgeeks.com/how-to-install-grafana-on-centos-7/

[6] **Instalación Nagios CentOS7:** https://comoinstalar.me/como-instalar-nagios-en-centos-7/

[7] **Configurar envío correo de alertas Nagios CentOS7:** https://www.youtube.com/watch?v=Cqueqz3W0VU

[8] **Configurar envío correo de alertas Nagios CentOS7:** https://www.youtube.com/watch?v=-NZ_ShGb0vU

[9] **Instalación Mailx CentOS7:** https://eltallerdelbit.com/mailx-gmail-centos/

[10] **Configurar envío correo de alertas Nagios CentOS7:** https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3/html/console_administration_guide/configuring_nagios_to_send_mail_notifications

[11] **Diferencias entre CentOS7 y CentOS8:** https://www.hostwinds.com/guide/what-are-the-differences-between-centos-7-8/

[12] **Monitoring machine metrics with Graphite:** https://ncona.com/2016/08/monitoring-machine-metrics-with-graphite/