UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Sistema de alertas basado en procesamiento en tiempo real de logs en una plataforma con disponibilidad 24/7

Autor/a:
Adrián Bernárdez Cornes

Titores:
Manuel Lama Penín

Máster universitario en Tecnologías de Análisis de Datos Masivos: Big Data

Marzo 2021

Traballo de Fin de Máster presentado na Escola Técnica Superior de Enxeñaría da Universidade de Santiago de Compostela para a obtención do Máster Interuniversitario en Big Data: Tecnologías de Análisis de Datos Masivos

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introdución

Para un sistema de máxima disponibilidad como es el caso de sistemas de dispensación farmacéutica, los cuales deben estar en funcionamiento las 24 horas del día, los 7 días de la semana se generan una cantidad masiva de logs.

Para cumplir con esta disponibilidad se deberá disponer de un equipo de desarrollo y de soporte que cubrirán todas las horas del día haciendo guardias durante las horas no laborables. Sin embargo, en caso de que un componente falle, el tiempo de respuesta en informar del problema y buscar una solución deberá ser el mínimo posible independientemente de la causa y circunstancia en que se de el problema.

El objetivo del proyecto será aportar una plataforma que se encargue de generar alertas en caso de una incidencia importante, analizando los logs de actividad hasta dar con el componente afectado. De esta forma el sistema deberá responder en tiempo real con el fin de agilizar el proceso y mejorar el servicio. Con esta nueva operativa se podría registrar estas alertas para poder dotar al equipo de un informe de las incidencias y sus componentes afectados, con el fin de elaborar soluciones a problemas comunes y poder formar de una manera más eficiente y exacta a los encargados del correcto funcionamiento del sistema principal.

Se buscará que la alerta generada sea lo más descriptible posible y que consiga dar un indicio de la causa del problema. De esta manera se podrá dar una respuesta con tiempos mínimos tiempos de espera, lo cual supondría tanto una

mejora en el servicio como la facilitación del trabajo de cara al equipo encargado de la supervisión de la aplicación.

A la hora realizar el diseño de la plataforma se deberá tener los siguientes puntos como claves:

- Dada una aplicación que tenga una disponibilidad total y sobre la que opere un número elevado de usuarios, se generan una cantidad de logs masiva, que en caso de que se produzca un error, el proceso de revisión de logs y la diagnosis posterior puede ocupar un tiempo demasiado largo en caso de que no se pueden producir cortes en el servicio.

- La finalidad del proyecto será disponer de una plataforma para detectar en tiempo real posibles problemas e incidencias y clasificarlas según experiencias pasadas reducirá ese tiempo de espera en gran medida. La funcionalidad de la clasificación de la alerta entre problemas conocidos supondría una mejora significativa en la gestión operacional de la aplicación.

- Registrar las incidencias alertadas supone un extra añadido, ya que se dispondrá de un histórico con el que saldrán a la luz posibles bugs o errores frecuentes de cara a poder llevar un control y análisis de la aplicación más exhaustiva.

- El proyecto deberá estar desacoplado de la aplicación principal y estar preparado para adaptarse a cambios y nuevas funcionalidades.

1.1. Objetivos

Una vez planteado el contexto se identifican los siguientes objetivos:

OBJ-01	**Emisión de alertas con información relevante tras la detección de posibles errores o incidencias. Se buscará enviar la posible causa del problema raíz.**
Descripción	El principal objetivo del proyecto es la construcción de una plataforma con las herramientas necesarias para la emisión de alertas tempranas en caso de producirse una incidencia en la aplicación a monitorizar. Se espera que la alertas se generen en near real time con el fin de una recuperación del servicio los más rápida y eficiente posible. La plataforma deberá ser configurable ya que nace de la necesidad de coexistir servicio a un sistema con disponibilidad total que está en continuo desarrollo y evolución.

OBJ-02	**Registro de las incidencias encontradas con la información necesaria para su posterior tratamiento.**
Descripción	Se deberá llevar un registro de las incidencias detectadas y de los logs de la aplicación obtenidos. De esta forma una vez solucionada la incidencia se podrá catalogar y registrar los errores producidos en la aplicación con el fin de un análisis posterior en el que se podrán detectar posibles vulnerabilidades o bugs.

OBJ-03	**Clasificar el error mediante los logs de la aplicación**
Descripción	La plataforma deberá utilizar técnicas de machine learning para la clasificación de la incidencia en base a los logs producidos en la aplicación. De esta forma se podrá plantear el envío de los pasos conocidos para su solución en cuanto se detecte el error.

1.2. Alcance

La solución alcanzada deberá cumplir los objetivos definidos en el apartado anterior. Por un lado la plataforma a desarrollar deberá incorporar las herramientas necesarias que permitan de forma simple para un usuario la creación de alertas y modificación de las mismas en base a criterios acordados y a la experiencia obtenida. Por otro deberá utilizar los datos obtenidos con la finalidad de aplicar herramientas de Machine Learning con el fin de clasificar la incidencia ocurrida.

Se deberá tener en cuenta que no se podrán utilizar datos reales de la aplicación actual debido a que se estarían utilizando datos personales de terceros lo que incumpliría la normativa de protección de datos. En su lugar se utilizarán logs de entornos de desarrollo sobre los cuales se implementará una solución que permita crear un escenario equivalente al entorno real sin datos personales y a menor escala.

Una vez finalizado el desarrollo se deberá proporcionar una plataforma que pueda ser agregada a la aplicación actual y que deberá ser validada a posteriori. Para ello deberá proponerse una arquitectura como una extensión de la plataforma actual con el fin de que coexista. Por ello se ha de tener en cuenta que la instalación y despliegue será controlado en un entorno preparado para este fin permitiendo un control de versiones y registro del resultado.

Como entregables del proyecto serán los siguiente:

1. Plataforma encargada de la emisión de alertas con reglas de alertas configurables.
2. Plataforma encargada de la clasificación de las incidencias y su registro.
3. Propuesta de integración en el sistema real

Capítulo 2

Revisión de solucións comerciais

2.1. Gestión de alertas

La monitorización de la actividad en la red puede ser un trabajo tedioso, sin embargo existen buenas razones pare hacerlo. Por un lado permite buscar e investigar inicios de sesión sospechosos en estaciones de trabajo, dispositivos conectados a redes y servidores mientras identifica las posibles deficiencia de seguridad en la adminitración del sistema. También puede reastrar las instalaciones software y las transferencias de datos para identificar problemas potenciales en tiempo real en lugar que después de que el daño esté hecho.

Estos logs trambién contribuyen en gran medida a que la organización cumpla con el reglamento general de protección de datos (GHDPR) que se aplica a cualquier entidad que opere en la unión europea.

El registro, tanto el seguimiento como el análisis debe ser un proceso fundamental en cualquier infraestructura de monitoreo. Es necesario un archivo de registro de transacciones para recuperar una base de datos de un desastre. Además, al rastrear los archivos de registro, los equipo de DevOps y los administradores base de datos (DBA) pueden mantener un rendimiento óptimo de la base de datos o encontrar evidencia de actividad no autorizada en el caso de un ciberataque. Por esta razón, es importante monitorear y analizar regularmente los regitro del sistema. Es una forma confiable de recrear la cadena de eventos que condujo a

cualquier problema que haya surgido.

Debido a esta serie de problemáticas, existen un número considerable de registro de código abierto y herramientas de análisis disponibles en la actualidad, lo que hace que elegir los recursos adecuados para los registros de actividad sea un trabajo considerable. La comunidad de software de código abierto y gratuito ofrece diseños de registros que funcionan con todo tipo de sitios y prácticamente con cualquier sistema operativo.

Para la gestión de los logs y producción de alertas se tuvieron en cuenta las siguientes herramientas:

2.1.1. Graylog

Graylog comenzó en Alemania en 2011 y ahora se ofrece como una herramienta de código abierto o una solución comercial. Está diseñado para ser un sistema de administración de registros centralizado que recibe flujos de datos de varios servidores o puntos finales y le permite navegar o analizar esa información rápidamente.

Graylog se ha ganado una reputación positiva entre los administradores de sistemas debido a su facilidad de escalabilidad. La mayoría de los proyectos web comienzan con poco, pero pueden crecer exponencialmente. Graylog puede equilibrar cargas en una red de servidores backend y manejar varios terabytes de datos de registro cada día.

Los administradores de TI encontrarán que la interfaz frontend de Graylog es fácil de usar y robusta en su funcionalidad. Graylog se basa en el concepto de paneles, que le permite elegir qué métricas o fuentes de datos le parecen más valiosas y ver rápidamente las tendencias a lo largo del tiempo.

Cuando ocurre un incidente de seguridad o rendimiento, los administradores de TI quieren poder rastrear los síntomas hasta una causa raíz lo más rápido posible. La función de búsqueda en Graylog lo hace fácil. Tiene tolerancia a fallas incorporada que puede ejecutar búsquedas de múltiples subprocesos para que pueda analizar varias amenazas potenciales juntas.

2.1.2. Nagios

Nagios comenzó con un solo desarrollador en 1999 y desde entonces se ha convertido en una de las herramientas de código abierto más confiables para administrar datos de registro. La versión actual de Nagios puede integrarse con servidores que ejecutan Microsoft Windows, Linux o Unix.

Su producto principal es un servidor de registros, cuyo objetivo es simplificar la recopilación de datos y hacer que la información sea más accesible para los administradores del sistema. El motor del servidor de registro de Nagios capturará datos en tiempo real y los enviará a una poderosa herramienta de búsqueda. La integración con un nuevo punto final o aplicación es fácil gracias al asistente de configuración integrado.

Nagios se usa con mayor frecuencia en organizaciones que necesitan monitorear la seguridad de su red local. Puede auditar una variedad de eventos relacionados con la red y ayudar a automatizar la distribución de alertas. Nagios incluso se puede configurar para ejecutar scripts predefinidos si se cumple una determinada condición, lo que le permite resolver problemas antes de que un humano tenga que involucrarse.

Como parte de la auditoría de red, Nagios filtrará los datos de registro según la ubicación geográfica donde se originan. Eso significa que puede crear paneles de control completos con tecnología de mapeo para comprender cómo fluye su tráfico web.

2.1.3. ELK Stack

1806 / 5000 Resultados de traducción Elastic Stack, a menudo llamado ELK Stack, es una de las herramientas de código abierto más populares entre las organizaciones que necesitan examinar grandes conjuntos de datos y dar sentido a los registros de su sistema.

Su oferta principal se compone de tres productos separados: Elasticsearch, Kibana y Logstash:

- Como sugiere su nombre, Elasticsearch está diseñado para ayudar a los usuarios a encontrar coincidencias dentro de conjuntos de datos utilizando una amplia gama de tipos y lenguajes de consulta. La velocidad es la ventaja número uno de esta herramienta. Se puede expandir en grupos de cientos de nodos de servidor para manejar petabytes de datos con facilidad.
- Kibana es una herramienta de visualización que se ejecuta junto con Elasticsearch para permitir a los usuarios analizar sus datos y crear informes potentes. Cuando instale por primera vez el motor Kibana en su clúster de servidores, obtendrá acceso a una interfaz que muestra estadísticas, gráficos e incluso animaciones de sus datos.
- La última pieza de ELK Stack es Logstash, que actúa como una canalización puramente del lado del servidor hacia la base de datos Elasticsearch. Puede integrar Logstash con una variedad de lenguajes de codificación y API para que la información de sus sitios web y aplicaciones móviles se alimente directamente a su potente motor de búsqueda Elastic Stalk.

Una característica única de ELK Stack es que le permite monitorear aplicaciones creadas en instalaciones de código abierto de WordPress. A diferencia de la mayoría de las herramientas de registro de auditoría de seguridad listas para usar que rastrean los registros de administración y PHP, pero poco más, ELK Stack puede examinar los registros del servidor web y de la base de datos.

El seguimiento de registros y la gestión de bases de datos deficientes son una de las causas más comunes del rendimiento deficiente de un sitio web. No verificar, optimizar y vaciar regularmente los registros de la base de datos no solo puede ralentizar un sitio, sino que también puede provocar un bloqueo completo. Por lo tanto, ELK Stack es una excelente herramienta para el kit de herramientas de todos los desarrolladores de WordPress.

Capítulo 3

Proposta de solución

Los archivos de registro de actividad que sobre los que se obtendrá la información del sistema se dividen en varios archivos, cada uno con un objetivo concreto. En concreto se monitorizarán tres ficheros, los cuales poseen un ciclo de vida que consiste en pasar por un fichero plano, para una vez alcanzado un tamaño determino comprimirse y guardarse como histórico. Si el histórico tiene una antigüedad superior a los días establecidos como máximo se enviarán a un servidor que funcionará como almacenamiento. Estos tres ficheros tienen los siguientes acometidos:

- Registro de interacciones (peticiones y respuestas) con otros sistemas externos e internos.
- Registro de las transformaciones y cálculos que reciben los datos en los distintos servicios de los que se compone la aplicación.
- Registro de la aplicación de entrada que utilizarán los usuarios finales con registro de las operaciones solicitadas y pantallas accedidas.

En los logs de actividad se registrarán el momento en que se registran, la clase en la que se originó el comportamiento y el texto con la información que se está registrando en ese momento.

En este capítulo se va a describir la plataforma implementada para la emisión de alertas. La arquitectura diseñada para la plataforma es la siguiente:

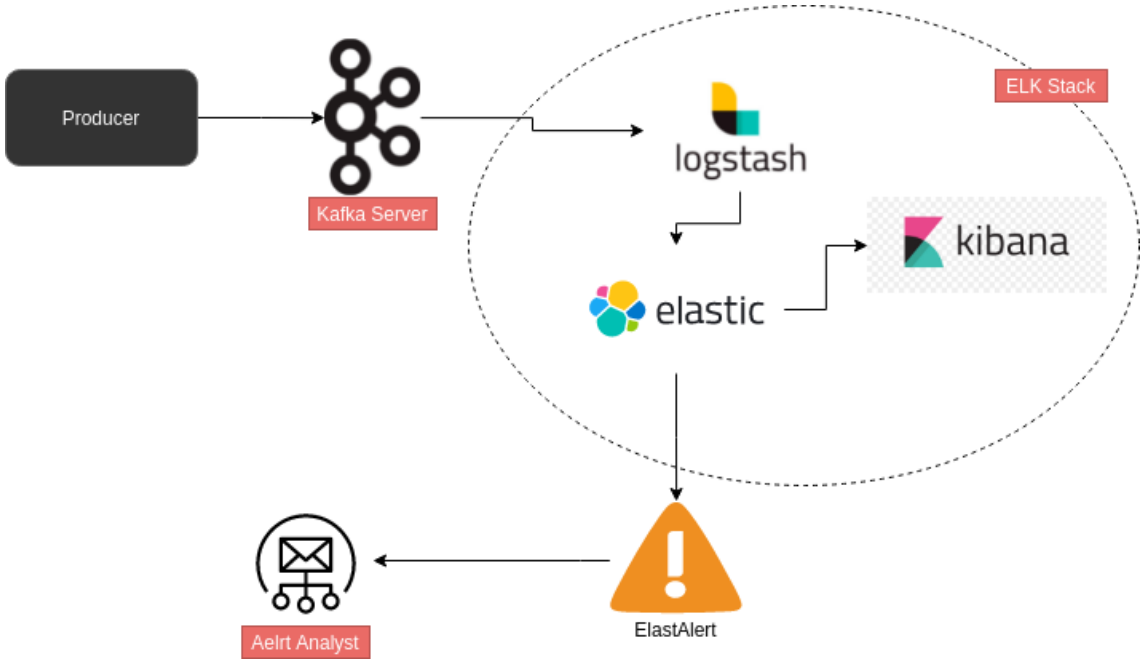

Figura 3.1: Arquitectura de la Plataforma de Gestión de Alertas

La arquitectura estará desplegado en varios contenedores Docker. Se pueden distinguir las siguientes partes:

- **Producer**: como se ha comentado en el capítulo de introducción no es posible utilizar los logs de la aplicación real en el entorno de producción ya que se estaría incumpliendo la ley de protección de datos. Para ello se utilizarán logs ya registrados de los que se tiene certeza que se produjo una incidencia pero procedente de un entorno de desarrollo con datos falsos. Para reproducir el comportamiento de la aplicación se implementará un script python que se encargue de leer cada línea de log y trasladarla a kafka, de esta forma replicará el comportamiento de escribir regitros a lo largo del tiempo.
- **Kafka server**: se utilizará kafka con el fin de trasladar los datos producidos en el productos al ELK Stack. La plataforma Apache Kafka es un sistema de transmisión de datos distribuido con capacidad de escalado y tolerante a fallos. Gracias a su alto rendimiento permitirá transmitir datos en tiempo real utilizando el patrón de mensajería publish/subscribe.
- **ELK Stack**: conjunto de aplicaciones (Elastic Search, Logstash y Kivana). Permite recoger datos de cualquier tpo de fuente y cualquier formato para

realizar búsquedas, análisis y visualización de los datos en tiempo real. En la arquitectura planteada se para la aplicación real se obtendrán los datos a partir de ficheros de logs mediante Logstash y se almacenarán en el motor de búsquedas y análisis de Elasticsearch. Además, se permite la visualización, monitorización y explotación de datos en tiempo real mediante kibana.

- **Elastalert**: herramienta open source que se integra con ELK y permite detectar anomalías e inconsistencias en los datos de Elasticsearch y lanzar alertas. El conjunto de herramientas de **ELK (Elastic Stack)** más ElastAlert permite crear un sistema de monitorización para la extracción, almacenamiento y explotación de los datos de los procesos o sistemas a monitorizar.

Capítulo 4

Conclusións e posibles ampliacións

Resumo das principais aportacións do traballo, explicacións das suposicións e limitacións do traballo, e posibles vías de mellora.

Bibliografía

[1] Nvidia CUDA programming guide. Versión 2.0, 2010. Dispoñible en *http://www.nvidia.com*.

[2] Acceso múltiple por división de código. Artigo da wikipedia (*http://es.wikipedia.org*). Consultado o 2 de xaneiro do 2010.

[3] R.C. Gonzalez e R.E. Woods, *Digital image processing*, 3ª edición, Prentice Hall, New York, 2007.

[4] P. González, J.C. Cartex e T.F. Pelas, "Parallel computation of wavelet transforms using the lifting scheme", *Journal of Supercomputing*, vol. 18, no. 4, pp. 141-152, junio 2001.