Implementación de un Sistema de Monitoreo en Tiempo Real para Servidores en Diversos Entornos con TIG, Integración y Alertas en Telegram

Andrade Santiago
Berrones Danilo
Calva Andy
Ortíz Adriana
Ortíz Javier

27 de noviembre de 2023

Resumen

Presentamos una propuesta metodológica para implementar un sistema de monitoreo en tiempo real utilizando la combinación de las herramientas de código abierto TIG (Telegraf, InfluxDB y Grafana). La implementación abarca entornos en la nube de Amazon y Digital Ocean, contenedores Docker, así como un despliegue en sitio (on-premise). Además, se integra la plataforma de mensajería Telegram para recibir alertas de manera eficiente y oportuna.

1. Introducción

1.1. Contexto y Justificación

Con la creciente digitalización, las organizaciones enfrentan desafíos cada vez mayores en la protección de sus activos digitales.

Es crucial detectar y responder a amenazas de manera proactiva. Se reconoce la importancia de dotar a las organizaciones con herramientas avanzadas para el monitoreo efectivo de infraestructuras digitales.

En un contexto donde la seguridad de la información es esencial, la necesidad de detectar actividades inusuales se vuelve crítica para cumplir con normativas y estándares como ISO 27001 y garantizar la continuidad del negocio.

Este proyecto busca desarrollar habilidades para implementar un sistema de monitoreo en tiempo real utilizando TIG (Telegraf, InfluxDB, Grafana). Esto permitirá anticipar amenazas y cumplir con estándares internacionales, contribuyendo a la seguridad de la información.

1.2. Objetivos

- Profundizar la comprensión en la importancia del monitoreo en tiempo real.
- Capacitar en la implementación del sistema TIG para monitorear servidores en diversos entornos.
- Desarrollar habilidades para la detección proactiva de actividades inusuales, cumpliendo normativas y estándares de seguridad.
- Fortalecer la comprensión teórica y proporcionar experiencia práctica en soluciones avanzadas de seguridad informática.
- Aplicar buenas prácticas para garantizar seguridad de la información en organizaciones.

1.3. Relevancia y Contribución al Campo de la Ciberseguridad

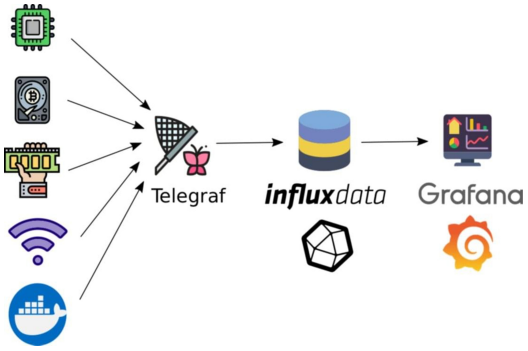
La implementación de la solución TIG, junto a la integración de alertas en Telegram, fortalece la detección proactiva de amenazas y contribuye al cumplimiento de normativas, elevando los estándares de seguridad en la gestión de infraestructuras digitales.

2. Marco Teórico

2.1. TIG Stack (Telegraf, InfluxDB, Grafana)

El TIG Stack es una solución integral para el monitoreo y visualización de datos. Telegraf, co-

mo agente de recopilación, captura información de servidores. InfluxDB, la base de datos de series temporales, almacena datos de manera eficiente. Grafana, la interfaz de visualización, proporciona tableros de dashboards personalizables para analizar y entender los datos en tiempo real. Esta combinación robusta, versátil y escalable garantiza un monitoreo efectivo y una respuesta proactiva ante eventos críticos.



2.2. Monitoreo en Tiempo Real

El monitoreo en tiempo real implica la observación constante de eventos y datos a medida que ocurren. Proporciona la capacidad de identificar anomalías de manera inmediata, permitiendo dar una respuesta ante posibles amenazas. Esta buena práctica es esencial para mantener la integridad, confidencialidad y disponibilidad de los sistemas, alineándose con los estándares de seguridad y asegurando la continuidad operativa de las organizaciones.

2.3. Entornos de Implementación (Amazon Web Services, Digital Ocean, Docker for Windows, On-Premise)

La implementación que desarrollaremos abarca diversos entornos para adaptarse a las necesidades variadas de las organizaciones. En Amazon Web Services (AWS), aprovechamos la flexibilidad y escalabilidad de la nube. Digital Ocean proporciona una solución eficiente y ágil para entornos en la nube. Con Docker for Windows, facilitamos la portabilidad y despliegue consistente en contenedores. Además, consideramos implementaciones en sitio (On-Premise), permitiendo a las organizaciones tener un control directo sobre su infraestructura y activos digitales. Esta diversidad de entornos asegura que nuestra implementación se pueda aplicar en diferentes contextos, brindando opciones versátiles para el monitoreo de la seguridad en infraestructuras digitales.

2.4. Integración de Telegram para Alertas

La integración de Grafana y Telegram potencia la capacidad de respuesta al permitir la recepción instantánea de alertas. Al vincular nuestro sistema con esta plataforma de mensajería, se facilita la notificación de eventos críticos, permitiendo a los responsables de TICS actuar rápidamente. Esta funcionalidad enriquece la capacidad de nuestro sistema para cumplir con las exigencias de tiempo real y fortalece la eficacia en la gestión de incidentes de seguridad y disponibilidad.

3. Metodología

3.1. Despliegue de InfluxDB para Almacenamiento Eficiente

Para implementar InfluxDB hemos utilizado una instancia de AWS, se siguieron los siguientes pasos:

1. Creación de una máquina virtual con Ubuntu 22.04.
2. Actualización del sistema mediante *sudo apt update* y *sudo apt upgrade*.
3. Instalación de InfluxDB con *sudo apt-get install influxdb* y *sudo apt install influxdb-client*.
4. Inicio del servicio con *sudo service influxdb start*.
5. Adición del servicio para inicio automático en el arranque mediante *sudo systemctl enable influxdb*.
6. Verificación del estado del servicio con *sudo service influxdb status*.
7. Ingreso al entorno InfluxDB con el comando *influx*.
8. Creación de una base de datos con *CREATE DATABASE nombreBaseDatos*.
9. Confirmación de la creación observando las bases de datos con *SHOW DATABASES*.
10. Acceso a la base de datos recién creada con *USE nombreBaseDatos*.
11. Comprobación de la funcionalidad mediante pruebas y exploración de la base de datos.
12. Cierre del entorno InfluxDB con el comando *EXIT*.

13. Verificación de acceso a la base de datos desde uno de los equipos que proporcionará métricas mediante *telnet ipServerInflux 8086*; en caso de error, revisar la configuración del firewall y proporcionar acceso a través del puerto 8086.

Con estos pasos, InfluxDB queda preparado para recibir datos de telemetría desde Telegraf, asegurando un almacenamiento eficiente y estructurado para el monitoreo en tiempo real de la infraestructura.

3.2. Instalación y Configuración de Telegraf para la Recopilación de Datos

Para configurar Telegraf y facilitar la recopilación de datos, se realizaron los siguientes pasos:

1. Instalación de Telegraf con el comando *sudo apt-get install -y telegraf*.
2. Edición del archivo de configuración mediante *sudo nano /etc/telegraf/telegraf.conf*.
3. Desactivación de la configuración de las métricas de Prometheus que vienen activadas por defecto, ya que no se utilizarán, y adición de la conexión de salida a la base de datos InfluxDB configurada según el capítulo anterior.

```
[[outputs.influxdb]]
  urls = ["http://35.78.145.175:8086"]
  database = "nombreBaseDatos"
  username = "nomUsuario"
  password = "passWord"
```

4. Adición de Telegraf al inicio del sistema para asegurar su ejecución constante con *sudo systemctl start telegraf*.
5. Verificación del estado del servicio para asegurar su correcto funcionamiento mediante *sudo systemctl telegraf status*.

Esta instalación y configuración de Telegraf la hemos repetido en nuestros 4 escenarios de prueba:

1. Máquina virtual en AWS con Ubuntu 22.04.
2. Máquina virtual en Digital Ocean con Ubuntu 22.04.
3. Contenedor Telegraf en Docker para Windows sobre mi computador personal.
4. Computador personal en red local con Ubuntu 22.04.

Con estos pasos, Telegraf queda configurado para recopilar datos y transmitirlos a InfluxDB, asegurando una integración fluida en el sistema de monitoreo en tiempo real.

3.3. Instalación de Grafana para Monitoreo y Visualización de Datos

La instalación de Grafana Server se llevó a cabo ejecutando los siguientes comandos desde la línea de comandos del terminal Linux:

```
sudo apt-get install -y software-properties-common
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
sudo apt-get update
sudo apt-get install -y grafana
sudo apt-get install -y apt-transport-https software-properties-common wget
sudo mkdir -p /etc/apt/keyrings/
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor
sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
sudo apt-get install grafana
sudo systemctl start grafana-server
sudo systemctl status grafana-server
```

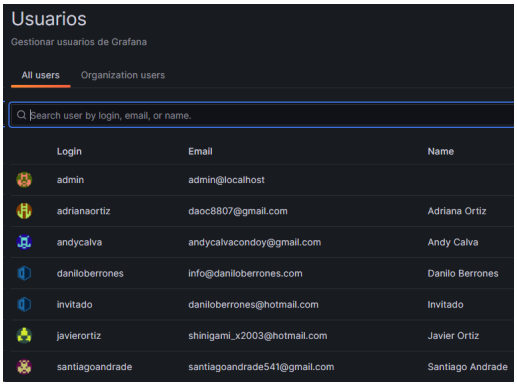
Estos comandos instalaron Grafana y aseguraron su inicio automático con el sistema. El estado del servicio se verificó para confirmar una instalación exitosa. Con Grafana en funcionamiento, se establece la base para la creación de dashboards personalizables que complementarán la visualización de datos provenientes de InfluxDB en nuestro sistema de monitoreo en tiempo real.

4. Implementación

4.1. Inicialización de Grafana: Usuarios y Personalización

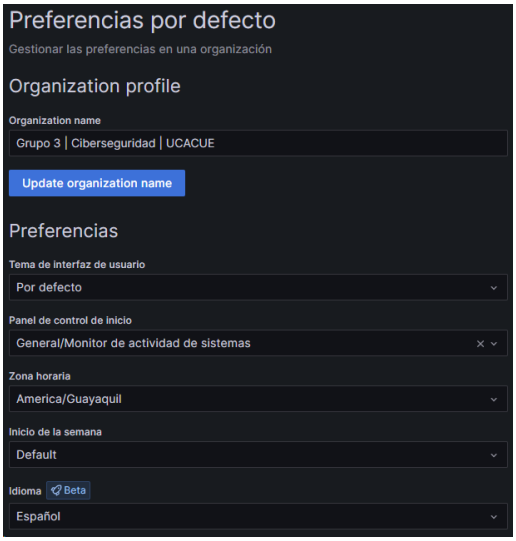
Después de acceder a Grafana con el usuario inicial 'admin' y establecer una contraseña segura, se procedió con lo siguiente:

1. **Gestión de Usuarios**:
   - Se crearon usuarios individuales para los 5 miembros del grupo, asignándoles roles de '*editor*' para facilitar la colaboración.
   - Además, se configuró un usuario invitado con privilegios de '*viewer*' para ofrecer acceso limitado.



2. **Configuración de Preferencias**:

    - Se ajustaron las preferencias por defecto del sistema asignándole un nombre a la organización para una clara identificación.
    - Se estableció la zona horaria según la ubicación geográfica GMT -5.
    - Se modificó el idioma predeterminado a español para una experiencia de usuario más intuitiva.



Con esta configuración inicial del entorno Grafana, buscamos proporcionar una base sólida para el trabajo colaborativo y una buena experiencia de usuario para los visitantes.

4.2. Sincronización con la Fuente de Datos InfluxDB:

Una vez configurado Grafana con las preferencias y usuarios adecuados, se procedió a la sincronización con la fuente de datos InfluxDB para habilitar la visualización de métricas en los dashboards. Los pasos para esta sincronización fueron los siguientes:

1. **Añadir Fuente de Datos**:

    - Desde el menú principal, se seleccionó *Connections* y luego *Data Sources*.
    - Se hizo clic en *Add your first data source* para iniciar la configuración de la fuente de datos.

2. **Configuración de la Fuente InfluxDB**:

    - Se asignó un nombre a la fuente de datos.
    - Se eligió InfluxDB como tipo de base de datos.
    - Se determinó a InfluxQL el lenguaje Query.
    - Se completaron los detalles de conexión, como la URL de la instancia InfluxDB con su respectivo puerto y las credenciales necesarias.

3. **Configuración Avanzada**:

    - Se ajustaron las opciones avanzadas según las necesidades del entorno, incluyendo detalles específicos de la base de datos y la configuración de la conexión.

4. **Verificación y Guardado**:

    - Se realizó una verificación de la conexión utilizando la función *Save & Test* para asegurarse de que Grafana pudiera conectarse correctamente a InfluxDB.

Al completar estos pasos, Grafana quedó totalmente sincronizado con InfluxDB, permitiendo la extracción de datos en tiempo real y facilitando la creación de visualizaciones y dashboards en base a la información almacenada en InfluxDB.

4.3. Despliegue Estratégico de Dashboards en el Panel de Control:

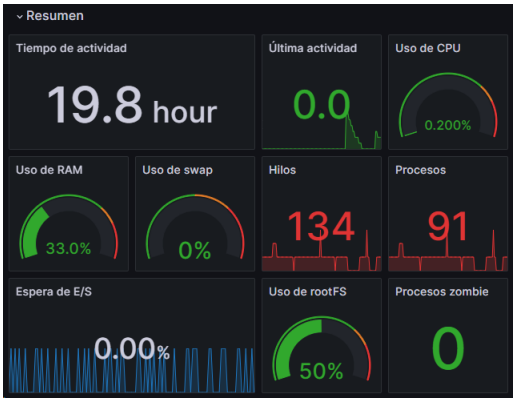
Para lograr un despliegue estratégico de nuestro dashboard, se empleó el siguiente enfoque:

1. **Creación de Dashboard**:

    - Se seleccionó la opción *Nuevo* para iniciar la creación de un nuevo dashboard.
    - Se eligió *Importar* y se optó por *Import via dashboard JSON model*.

2. **Edición del Modelo JSON**:

    - Se editaron las etiquetas del modelo para que se mostraran en el idioma español y se ajustaron las ubicaciones de los elementos del tablero según la preferencia del equipo.
    - Se llevaron a cabo modificaciones en el diseño para una presentación más eficiente y comprensible.



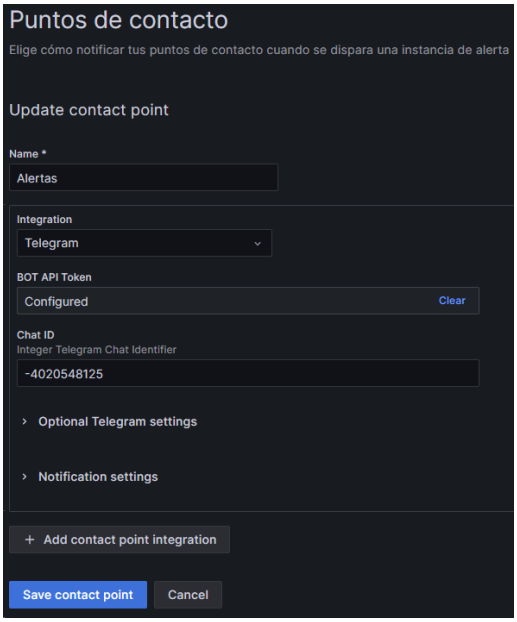
3. **Verificación y Ajustes Finales**:
    - Se realizó una verificación detallada del dashboard importado, asegurándose de que todas las métricas y visualizaciones se mostraran correctamente.
    - Se realizaron ajustes finales según las preferencias del equipo.

Este proceso aseguró que los dashboards estuvieran alineados con nuestros requerimientos, proporcionando una visualización clara y efectiva de los datos provenientes de InfluxDB. La capacidad de importar y personalizar dashboards facilitó un despliegue rápido y estratégico en el panel de control de Grafana.

4.4. Configuración de la Sección de Alertas para Notificaciones en Telegram:

La configuración de alertas para notificaciones en Telegram se realizó de la siguiente manera:

1. **Creación del Bot en Telegram**:
    - Se utilizó BotFather para crear un bot en Telegram.
    - Se siguieron los pasos de */start* y */newbot*, asignándole el nombre *dBot*, luego hemos obtenido el token de acceso para la configuración en Grafana.
2. **Canal de Telegram para Alertas**:
    - Se creó un canal en Telegram llamado *AlertasGrafana* para recibir las notificaciones del bot.
    - Se agregó en dicho canal a *dBot* y todos los contactos que recibirán las alertas.
3. **Configuración de las Reglas de Alerta en Grafana**:
    - En Grafana, se ingresó a la opción *Reglas de Alerta* del menú *Alertas*.
    - Se agrega una regla de alerta con *New alert rule*.
    - Asignamos un nombre para la regla, definimos la consulta Query y la condición de alerta.
    - Se realizan ajustes adicionales para trazar la ruta de las notificaciones de alertas.
    - Posteriormente se ingresó a la opción *Puntos de Contacto* del menú *Alertas*.
    - Agregamos a *Telegram* como el canal de notificación dando clic en el botón *Add Contact Point*.
    - Se proporcionó el token del bot creado y se especificó el chat ID del canal *AlertasGrafana*.
    - Finalmente fuimos a *Políticas de notificación*, en donde modificamos la política para que el punto de contacto sea Telegram.



4. **Prueba de Alertas**:
    - Se realizaron pruebas de alertas para asegurarse de que las notificaciones se enviaran correctamente al canal de Telegram configurado.
    - Se realizaron ajustes según sea necesario para garantizar la recepción adecuada de alertas.

Con esta configuración, Grafana está listo para enviar notificaciones instantáneas a través de

Telegram en caso de eventos críticos, brindando al equipo una respuesta proactiva y eficiente ante posibles problemas detectados en tiempo real.

5. Simulación y Resultados:

5.1. Escenarios de Simulación:

En esta fase del estudio, se llevaron a cabo simulaciones en tres escenarios distintos, cada uno representando entornos comunes en la práctica de monitoreo:

1. Máquinas Virtuales (AWS y Digital Ocean):
   - Se crearon instancias en la nube de AWS y Digital Ocean para emular entornos escalables y distribuidos.
   - Para simular y experimentar estos escenarios utilizamos la herramienta stress de Linux (incluído Ubuntu).
   - Se generaron alertas ante un uso elevado de CPU, memoria y disco duro, indicando posibles problemas como saturación del sistema, degradación del rendimiento y riesgo de agotamiento de recursos.
2. Contenedor en Computadora Personal:
   - Se utilizó un contenedor en una computadora personal con Docker para simular un entorno más controlado.
   - Las alertas se activaron en respuesta a altos niveles de CPU, memoria y uso del disco, proporcionando insights sobre la gestión de recursos en entornos de contenedores.
3. Máquina On-Premise en Infraestructura Privada:
   - Se implementó un escenario on-premise para simular una infraestructura privada en un entorno físico.
   - Se monitoreó el rendimiento de CPU, memoria y disco, generando alertas que podrían indicar problemas de capacidad, fallas de hardware o configuraciones no óptimas.

5.2. Métricas evaluadas:

Telegraf nos proporciona 9 categorías de métricas, entenderlas nos va a proporcionar información valiosa sobre el rendimiento y el estado de los sistemas.

1. **CPU**

   Alertas:
   - Leve: superior a 70 %.
   - Moderado: superior a 85 %.
   - Alto: superior a 95 %.

   Causas de Alerta:
   - Cargas de trabajo intensivas.
   - Procesos que consumen muchos recursos.
   - Máquinas virtuales mal configuradas.
2. **DISK** (Espacio en Disco)

   Alertas:
   - Leve: Uso superior al 80 %.
   - Moderado: Uso superior al 90 %.
   - Alto: Uso superior al 95 %.

   Causas de Alerta:
   - Almacenamiento insuficiente.
   - Archivos de registro que crecen rápidamente.
   - Problemas con la rotación de registros.
3. **DISKIO** (Actividad de Entrada/Salida del Disco)

   Alertas:
   - Leve: Altos niveles de actividad de disco.
   - Moderado: Mayor actividad de disco y latencias.
   - Alto: Saturación de disco y latencias inaceptables.

   Causas de Alerta:
   - Disco duro defectuoso o lento.
   - Cargas intensivas de E/S (entrada/salida).
4. **KERNEL**

   Alertas:
   - Leve: Errores ocasionales del kernel.
   - Moderado: Errores frecuentes del kernel.
   - Alto: Problemas graves del kernel.

   Causas de Alerta:
   - Fallos en el sistema operativo.
   - Problemas de estabilidad del kernel.

5. **MEM** (Uso de Memoria)

   Alertas:

   - Leve: Uso de memoria superior al 80 %.
   - Moderado: Uso de memoria superior al 90 %.
   - Alto: Uso de memoria superior al 95 %.

   Causas de Alerta:

   - Cargas de trabajo que consumen mucha memoria.
   - Fugas de memoria en aplicaciones.
   - Configuración incorrecta de la memoria.

6. **PING** (Latencia y Disponibilidad de Red)

   Alertas:

   - Leve: Aumento en la latencia de ping.
   - Moderado: Pérdida intermitente de ping.
   - Alto: Pérdida constante de ping.

   Causas de Alerta:

   - Problemas de red.
   - Cargas de red inesperadas.

7. **PROCESSES** (Actividad de los Procesos)

   Alertas:

   - Leve: Número elevado de procesos.
   - Moderado: Número de procesos cercano al límite del sistema.
   - Alto: Saturación de procesos.

   Causas de Alerta:

   - Exceso de procesos en ejecución.
   - Problemas con aplicaciones que generan demasiados procesos.

8. **SWAP** (Uso de la Memoria de Intercambio)

   Alertas:

   - Leve: Uso de memoria de intercambio superior al 50 %.
   - Moderado: Uso de memoria de intercambio superior al 75 %.
   - Alto: Uso de memoria de intercambio cercano al 100 %.

   Causas de Alerta:

   - Configuración inadecuada de la memoria.
   - Cargas de trabajo que consumen más memoria de la disponible.

9. **SYSTEM** (Métricas Generales del Sistema)

   Alertas:

   - Leve: Advertencias generales del sistema.
   - Moderado: Problemas de rendimiento sistémico.
   - Alto: Problemas críticos del sistema.

   Causas de Alerta:

   - Errores de sistema operativo.
   - Problemas de estabilidad del sistema.

Esta evaluación es de tipo general y los umbrales pueden variar según el entorno y la aplicación. Es importante ajustar dichos umbrales y las estrategias de alerta según los requisitos y la carga de trabajo específicos del entorno donde vamos a aplicar la monitorización.

5.3. Eficiencia del Sistema en la Detección y Notificación de Alertas:

Destacamos la importancia de la rapidez y precisión en la identificación de eventos críticos, las mismas que pusimos a prueba por medio de la herramienta stress de linux con la que creamos cargas a CPU, memoria, y E/S al disco en diferentes niveles y tiempos de ejecución.

1. *Tiempo de Respuesta*:

   Analizamos el tiempo que transcurre desde la ocurrencia de una anomalía hasta que se detecta y notifica como alerta. La eficiencia en la reducción del tiempo de respuesta es vital para abordar problemas antes de que impacten significativamente en el rendimiento y la seguridad del sistema.

2. *Precisión en la Identificación*:

   Evaluamos la capacidad del sistema para identificar de manera precisa y específica la naturaleza de la anomalía. La precisión en la identificación garantiza que las alertas sean relevantes y proporciona información valiosa para la toma de decisiones.

3. *Manejo de Falsos Positivos y Negativos*:

   Examinamos la capacidad del sistema para minimizar los falsos positivos (alertas incorrectas) y los falsos negativos (no detectar eventos críticos reales). Un manejo efectivo

de estos casos mejora la confianza en las alertas generadas y reduce la carga operativa asociada con eventos no críticos.

4. *Escalabilidad del Sistema*:

   Se consideró la capacidad del sistema para manejar eficientemente un aumento en la carga de trabajo y en el volumen de datos sin comprometer la velocidad y precisión en la detección de alertas, ya que la escalabilidad es fundamental para garantizar que el sistema pueda adaptarse a entornos cambiantes y crecientes demandas.

5.4. Resultados:

Estos escenarios de simulación proporcionaron resultados valiosos al activar alertas en situaciones que podrían afectar la disponibilidad y rendimiento del sistema. La anticipación de problemas a través de estas alertas es esencial para tomar medidas preventivas y mantener la integridad y disponibilidad de la infraestructura.

También en nuestros escenarios de prueba pudimos demostrar las fortalezas del sistema en la detección y notificación de alertas, ver áreas de mejora y futuras optimizaciones.

6. Discusión

6.1. Interpretación de los Resultados Obtenidos

En la interpretación de los resultados, se observó que las métricas recopiladas a través de Telegraf y almacenadas en InfluxDB proporcionaron una visión detallada del rendimiento y la salud de los sistemas monitoreados. En general, los valores estuvieron dentro de los límites considerados normales para las métricas específicas, lo cual indica un buen estado operativo.

Sin embargo, se identificaron casos puntuales en los que la máquina virtual que aloja Grafana experimentó un aumento en el uso de memoria cuando se enfrentó a múltiples consultas simultáneas. Este comportamiento podría estar relacionado con la carga de trabajo específica del servidor Grafana y podría requerir una revisión más detallada para optimizar el rendimiento.

6.2. Desafíos y Limitaciones Encontradas Durante la Implementación

El desafío más significativo encontrado durante la implementación fue la configuración adecuada de los firewalls para garantizar el funcionamiento seguro y eficiente de la orquestación de los elementos de monitoreo. Configurar reglas de firewall precisas y permitir el tráfico necesario entre los componentes (Telegraf, InfluxDB, Grafana) y los sistemas monitoreados fue esencial para asegurar una comunicación efectiva sin comprometer la seguridad.

Además, se destaca que el monitoreo de Grafana en situaciones de carga intensiva podría requerir una atención especial, ya que se observó un aumento en el uso de memoria en escenarios de múltiples consultas simultáneas. Esto podría considerarse en futuras optimizaciones de la configuración de la máquina virtual y el escenario al que se vaya a enfrentar.

6.3. Implicaciones Prácticas y Teóricas de la Propuesta

La implementación de un sistema de monitoreo basado en Telegraf, InfluxDB y Grafana tiene varias implicaciones prácticas y teóricas:

- *Eficiencia Operativa*: La configuración permite una supervisión continua y eficiente de los sistemas, identificando posibles problemas antes de que afecten el rendimiento general.

- *Toma de Decisiones Informada*: El acceso a métricas detalladas proporciona información valiosa para la toma de decisiones informada sobre la capacidad, el rendimiento y la planificación de recursos.

- *Escalabilidad*: El sistema es escalable y puede adaptarse a entornos con una mayor cantidad de máquinas y complejidad, proporcionando una solución robusta para la monitorización.

- *Desarrollo Continuo*: La identificación de desafíos, como la configuración de firewalls, destaca la importancia del desarrollo continuo y la optimización de la implementación para enfrentar situaciones cambiantes.

En resumen, la propuesta de monitoreo implementada proporciona una base sólida para la gestión eficiente de sistemas, con implicaciones significativas en términos de eficiencia operativa y toma de decisiones. La identificación de desafíos también resalta la necesidad de una gestión proactiva y continua para garantizar la efectividad a largo plazo del sistema de monitoreo.

7. Conclusiones

7.1. Recapitulación de los Hallazgos

La implementación y evaluación del sistema de monitoreo basado en Telegraf, InfluxDB y Grafana proporcionaron una visión detallada del rendimiento de los sistemas, identificando métricas clave en tiempo real. Se destacan los siguientes puntos:

- *Monitoreo Efectivo*: La integración de Telegraf para la recopilación de datos, InfluxDB para el almacenamiento y Grafana para la visualización permitió un monitoreo efectivo y en tiempo real de los sistemas.
- *Identificación de Problemas*: Las alertas configuradas en Grafana permitieron identificar y responder rápidamente a posibles problemas, garantizando un mantenimiento proactivo de los sistemas.
- *Desafíos en la Configuración*: La configuración de firewalls presentó desafíos significativos, destacando la importancia de la seguridad en la implementación de sistemas de monitoreo.
- *Optimización Futura*: La identificación de situaciones específicas, como el aumento del uso de memoria en Grafana bajo cargas intensivas, sugiere áreas para futuras optimizaciones y mejoras.

7.2. Contribuciones al Campo de la Ciberseguridad

Las contribuciones al campo de la ciberseguridad se manifiestan en:

- *Monitoreo Proactivo*: La implementación proporciona una herramienta efectiva para el monitoreo proactivo de sistemas, permitiendo la identificación temprana de posibles amenazas o problemas de rendimiento.
- *Seguridad y Eficiencia*: La configuración segura de firewalls destaca la importancia de la seguridad en el monitoreo de sistemas, contribuyendo a las mejores prácticas de ciberseguridad.
- *Optimización Continua*: La identificación de áreas para futuras optimizaciones contribuye a la evolución continua del monitoreo de sistemas para adaptarse a cambios en el entorno operativo.

7.3. Direcciones Futuras de Investigación

Considerando el deseo de profundizar en el uso práctico y hacer más amigables las alertas para usuarios con menos conocimientos informáticos, se sugieren las siguientes direcciones futuras de investigación:

- *Desarrollo de Interfaz Amigable*: Explorar el desarrollo de interfaces más amigables en Grafana para hacer que las alertas y métricas sean más comprensibles para usuarios no técnicos.
- *Capacitación del Personal*: Investigar estrategias de capacitación efectivas para usuarios de manera que puedan interpretar y actuar sobre las alertas de manera más eficiente.
- *Automatización de Respuestas*: Explorar la posibilidad de integrar respuestas automatizadas a ciertos tipos de alertas, reduciendo la necesidad de intervención manual.
- *Escalabilidad y Flexibilidad*: Investigar opciones para hacer que la solución sea más escalable y adaptable a entornos diversos, considerando diferentes niveles de complejidad en las infraestructuras.

En conclusión, la implementación actual sienta las bases para futuras investigaciones que buscan mejorar la accesibilidad y eficiencia del monitoreo de sistemas, permitiendo su aplicación práctica en una variedad de entornos y perfiles de usuarios.

Si desean revisar con más detalle el desarrollo de esta implementación, los invitamos a ver nuestro tutorial.

Si desean visualizar de forma interactiva pueden ingresar a nuestro monitor utilizando el usuario *invitado*, contraseña *invitado*.

Referencias

[1] Amazon. (2023). *AWS. Documentación. Reglas de AWS WAF - Guía para desarrolladores.* Página web. docs.aws.amazon.com.

[2] Davidochobits. (2018). *Pruebas de estrés en sistemas GNU/Linux.* Página web. www.ochobitshacenunbyte.com.

[3] Devconnected. *How To Setup Telegraf InfluxDB and Grafana on Linux.* Página web. www.devconnected.com.

[4] DigitalOcean. *How To Monitor System Metrics with the TICK Stack on CentOS 7*. Página web. www.digitalocean.com.

[5] Fuentes Washington. (2021). *Monitorización mediante GRAFANA de la seguridad de una red de computadoras para mitigar las vulnerabilidades en el tráfico de datos*. Repositorio Digital de Universidad Ecotec.

[6] Grafana. *Internet Uptime Monitor*. Página web. www.grafana.com.

[7] Grafana Community. *Grafana Labs Community*. Página web. community.grafana.com.

[8] Huntabyte. (2022). *Easily Install InfluxDB, Telegraf, & Grafana with Docker*. Archivo de video. Youtube.

[9] Influxdata. (2023). *Telegraf documentation*. Página web. docs.influxdata.com.

[10] Jorge de la Cruz. (2020). *En busca del Dashboard perfecto: InfluxDB, Telegraf y Grafana (Instalando InfluxDB, Telegraf y Grafana sobre Ubuntu 20.04 LTS)*. Página web. jorgedelacruz.es.

[11] Ionos. (2023). *Instalar Docker en Windows 10*. Página web. www.ionos.es.

[12] OpenWebinars. (2022). *Qué es InfluxDB y primeros pasos*. Página web. openwebinars.net.

[13] Normas APA. (2019). *Normas APA – 7ma (séptima) edición*. Página web. www.normas-apa.org.

[14] NullSafe. (2019). *Monitorización de servidores con Grafana, Telegraf e InfluxDB*. Archivo de video. Youtube.