**Integración de IA generativas y gestor de Logs (Wazuh) en Zabbix para monitoreo**

**inteligente**

**Contenido**

[Introducción 3](#_bookmark0)

[Objetivos 3](#_bookmark1)

[Objetivo General 3](#_bookmark2)

[Objetivos Específicos 3](#_bookmark3)

[Marco Teórico 4](#_bookmark4)

[IA Generativa 4](#_bookmark5)

[Evolución 5](#_bookmark6)

[Beneficios de Integrar IA Generativa con Zabbix 5](#_bookmark7)

[Wazuh 6](#_bookmark8)

[Características importantes 6](#_bookmark9)

[Integración de Wazuh con zabbix 6](#_bookmark10)

[Desarrollo 7](#_bookmark11)

* [Integración de Zabbix con GLPI 7](#_bookmark12)
* [IA GPT4ALL 18](#_bookmark13)
* [Implementar gestor de logs ‘Wazuh’ 27](#_bookmark14)

[Resultados 32](#_bookmark15)

* [Alertas GLPI 32](#_bookmark16)
* [GPT4ALL 34](#_bookmark17)
* [ORCA MINI 35](#_bookmark18)
* [Gestión de Logs 37](#_bookmark19)

[Conclusiones 39](#_bookmark20)

[Recomendaciones 40](#_bookmark21)

[Referencias 41](#_bookmark22)

# Introducción

En la gestión y monitoreo de infraestructuras tecnológicas, la integración de herramientas basadas en inteligencia artificial (IA) y gestores de logs permite optimizar procesos, fortalecer la seguridad y mejorar la detección y resolución de problemas. Este informe detalla la implementación de GPT4All y Orca Mini, modelos de lenguaje ejecutables localmente que se eligen según las necesidades específicas del escenario, en conjunto con Zabbix y Wazuh, siendo herramientas reconocidas para el monitoreo y la gestión de redes, servidores, aplicaciones y eventos de seguridad.

Por otro lado, en este proyecto es desarrollar un chatbot inteligente que permita a los administradores interactuar con Zabbix utilizando consultas en lenguaje natural para gestionar métricas, alertas y resolver incidencias. Además, Wazuh actúa como una plataforma integral para la gestión de eventos de seguridad y el análisis de logs, complementando la integración al proporcionar información detallada y estructurada sobre eventos críticos y anomalías en los sistemas. Esta implementación busca no solo automatizar tareas administrativas comunes, sino también mejorar la accesibilidad y rapidez en la gestión de sistemas críticos.

# Objetivos:

# Objetivo General:

Integrar un modelo de inteligencia artificial generativa con Zabbix y Wazuh para mejorar la gestión y análisis de alertas relacionadas con eventos de seguridad, proporcionando respuestas automáticas y contextuales que optimicen la resolución de problemas en sistemas monitoreados.

# Objetivos Específicos:

* Configurar Zabbix para ejecutar scripts personalizados que interactúen con modelos de lenguaje generativos como GPT4All.
* Implementar un flujo de trabajo para enviar información de las alertas y eventos de seguridad detectadas en Zabbix y Wazuh al modelo de inteligencia artificial, obteniendo respuestas útiles.
* Desarrollar scripts y configuraciones específicas en Wazuh para identificar eventos críticos en los logs, vinculados con el chatbot para generar diagnósticos automatizados.
* Validar la ejecución de scripts desde el entorno de Zabbix para garantizar su integración del modelo generativo y plataformas de monitoreo.
* Automatizar la generación de recomendaciones o análisis basados en eventos específicos utilizando los resultados del modelo generativo.

# Marco Teórico

# IA Generativa

La Inteligencia Artificial Generativa es una rama de la IA que se enfoca en la creación de contenido nuevo (texto, imágenes, audio, código, etc.) a partir de datos previos. Utiliza modelos de aprendizaje profundo para predecir y generar información de manera autónoma, imitando patrones humanos de creatividad y análisis.



*Figura 1 IA Generativa*

* Utiliza redes neuronales profundas para generar texto, imágenes y otros formatos de datos.
* Modelos populares incluyen GPT-4, LLaMA, DALL-E y Stable Diffusion.
* Permite aplicaciones en automatización, asistencia virtual, seguridad informática y más.
* Su implementación requiere consideraciones éticas y de privacidad, ya que puede generar contenido engañoso o ser utilizado de manera indebida.

# ¿Con que fin se desarrolló?

Automatizar la generación de contenido en múltiples formatos.

Reducir la carga de trabajo repetitiva al generar textos, resúmenes y análisis de datos en tiempo real.

Mejorar la interacción humano-máquina mediante respuestas más naturales.

Optimizar procesos en diversas industrias, como la educación, la medicina, el marketing y el desarrollo de software.

# Evolución

*Tabla 1 Evolución de Inteligencia artificial generativa*

|  |  |  |
| --- | --- | --- |
| **Años** | **Datos** | |
| **1950s-1960s** | Primeros Modelos | * Teorías iniciales sobre inteligencia artificial. * Primeros intentos de   procesamiento de lenguaje natural. |
| **1970s-1980s** | Sistemas Expertos | * Desarrollo de sistemas expertos. * Primeras aplicaciones de   redes neuronales básicas. |
| **1990s** | Redes  Neuronales Profundas | * Avances en aprendizaje profundo y procesamiento   de datos masivos. Expansión  del machine learning en la academia y la industria. |
| **2010s** | Modelos Generativos | * Aparición de las GANs (Redes Generativas   Adversarias).   * Evolución en procesamiento de lenguaje natural (NLP). |
| **2020s** | IA  Generativa Moderna | * GPT-3, GPT-4, LLaMA, GPT4All, DALL-E. * Integración de IA   multimodal en imágenes, audio y texto.   * Optimización para ejecución   en dispositivos con recursos limitados. |

# Beneficios de Integrar IA Generativa con Zabbix

*Tabla 2 Beneficios de la integración de IA Generativa con Zabbix*

|  |  |
| --- | --- |
| **Beneficio** | **Descripción** |
| Análisis inteligente de alertas | La IA puede interpretar el significado de las alertas en Zabbix y ofrecer diagnósticos detallados. |
| Automatización de Respuestas | Se pueden generar acciones automáticas ante eventos específicos, reduciendo intervención humana. |

|  |  |
| --- | --- |
| Predicción de Fallos | Mediante modelos de aprendizaje, la IA puede predecir fallos antes de que ocurran. |
| Interacción en Lenguaje Natural | Se pueden realizar consultas a Zabbix usando chatbots IA, facilitando la administración. |
| Reducción de Falsos Positivos | La IA puede filtrar alertas irrelevantes, evitando ruido en el monitoreo. |

# Wazuh

Es una plataforma de seguridad y monitoreo de logs que proporciona detección de intrusiones, respuesta a incidentes y cumplimiento de normativas en entornos de TI. Se basa en la recopilación, análisis y correlación de eventos de seguridad provenientes de sistemas operativos, servidores y dispositivos en red.

# Características importantes:

* Proteger infraestructuras tecnológicas mediante la detección de amenazas en tiempo real.
* Facilitar la gestión centralizada de la seguridad en redes empresariales y entornos en la nube.
* Automatizar la identificación de vulnerabilidades y posibles ataques informáticos.
* Cumplir con normativas de ciberseguridad como GDPR, HIPAA, PCI-DSS, entre otras.

# Integración de Wazuh con zabbix

La integración de Wazuh con Zabbix combina gestión de seguridad y monitoreo de infraestructura en un solo entorno. Wazuh se encarga de la detección de amenazas, análisis de logs y gestión de seguridad, mientras que Zabbix proporciona monitoreo en tiempo real de recursos, servidores y aplicaciones.

*Tabla 3 Beneficios de integrar Wazuh con Zabbix*

|  |  |
| --- | --- |
| **Beneficio** | **Descripción** |
| Monitoreo Centralizado | Permite ver métricas operacionales y eventos de seguridad desde un solo dashboard. |
| Automatización de Respuestas | Wazuh detecta amenazas y Zabbix ejecuta acciones correctivas automáticamente. |
| Correlación de Eventos | Relaciona problemas de rendimiento en servidores con posibles ataques de seguridad. |
| Análisis en Tiempo Real | Monitorea logs, vulnerabilidades y actividad sospechosa en la red. |

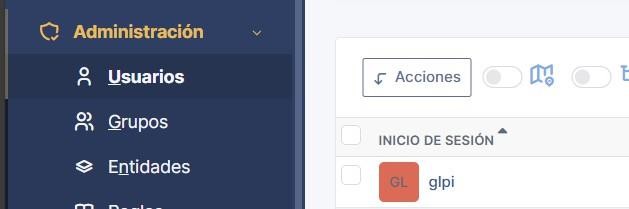
|  |  |
| --- | --- |
| Cumplimiento de Normativas | Facilita auditorías de seguridad bajo estándares como PCI- DSS, GDPR e ISO 27001. |

# Desarrollo

Para el caso de GLPI se instala en una máquina virtual diferente al servidor del zabbix debido al consumo considerado de recursos, de la misma forma se usa otra diferente para el caso de implementación de logs.

# Integración de Zabbix con GLPI

En el panel de administración de GLPI, se accede a la sección de usuarios para gestionar las cuentas registradas en el sistema. Se muestra el usuario predeterminado "glpi", creado automáticamente durante la instalación y generalmente utilizado como administrador inicial. Desde esta interfaz, es posible realizar acciones como la edición, asignación de roles, o eliminación de usuarios, utilizando las herramientas del menú, que también permite gestionar grupos, entidades y configuraciones relacionadas con la administración de acceso al sistema. Este módulo es fundamental para estructurar los permisos y garantizar una gestión eficiente de los usuarios.



*Figura 1 TOKEN LDPI*

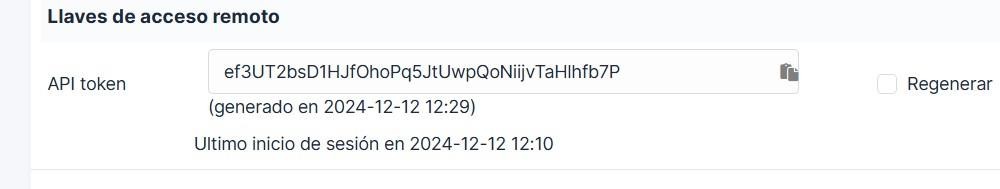
La sección muestra la opción de gestionar las llaves de acceso remoto (API Token) en GLPI, utilizada para integrar el sistema con aplicaciones externas o automatizar tareas a través de su API REST. Se incluye la funcionalidad de regenerar el token mediante la casilla **"Regenerar"**, y se muestra el registro del último inicio de sesión del usuario en el sistema, indicando la fecha y hora exacta. Este apartado es útil para administrar accesos remotos de forma segura y monitorizar actividades.



*Figura 2 Configuración de Llave API en GLPI*

Se muestra un API Token generado automáticamente, que puede ser utilizado para

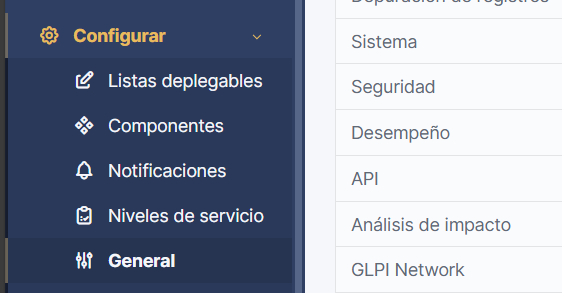
autenticar solicitudes a través de la API REST de GLPI. El token incluye información sobre la fecha y hora en que fue generado (2024-12-12 12:29) y permite regenerarse en cualquier momento seleccionando la opción **"Regenerar"**. También se registra el último inicio de sesión del usuario (2024-12-12 12:10), proporcionando trazabilidad de acceso.



*Figura 3 Generación de API Token en GLPI*

En esta sección del menú de GLPI, se encuentran las opciones principales de configuración del sistema. Entre las opciones destacadas están:

* **Listas desplegables:** Permite gestionar valores predefinidos para facilitar el ingreso de datos.
* **Componentes:** Configura elementos asociados a activos y sistemas.
* **Notificaciones:** Administra y personaliza las alertas enviadas a los usuarios.
* **Niveles de servicio:** Define y gestiona acuerdos de nivel de servicio (SLA) para tickets.
* **General:** Incluye configuraciones globales, como sistema, seguridad, rendimiento y ajustes para la API y redes GLPI.



*Figura 4 Menú de Configuración en GLPIt*

Se configuran los parámetros para habilitar el acceso mediante la API REST de GLPI. Los ajustes incluyen:

* **URL de la API:** Dirección utilizada para interactuar con la API, en este caso

[http://192.168.100.135/apirest.php.](http://192.168.100.135/apirest.php)

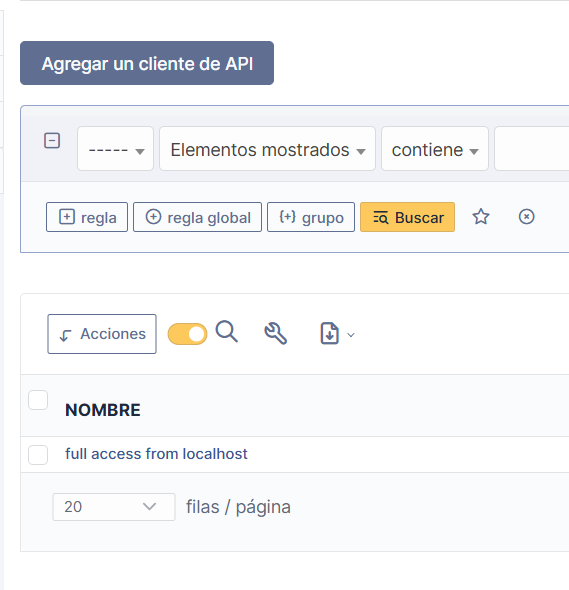
* **Habilitar API REST:** Opción activada para permitir el acceso y uso de la API.
* **Autenticación:** Se habilita el acceso con credenciales para garantizar seguridad en las interacciones con la API.



*Figura 5 Configuración de la API REST en GLPI*

Se visualiza el módulo de gestión de clientes de API en GLPI. Desde aquí, es posible:

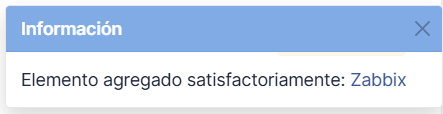
* **Agregar un cliente de API:** Permite crear nuevos accesos específicos con permisos personalizados para aplicaciones o integraciones.
* **Lista de clientes existentes:** En este caso, se muestra un cliente con nombre "full access from localhost", que tiene permisos completos desde la dirección localhost.
* **Herramientas de búsqueda y filtrado:** Facilitan encontrar o gestionar reglas y configuraciones aplicadas a los clientes de API.



*Figura c Gestión de Clientes de API en GLPI*

Se muestra un mensaje de confirmación indicando que un nuevo elemento, denominado

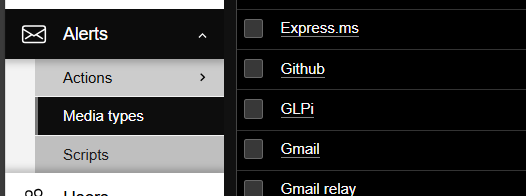
**"Zabbix"**, ha sido agregado exitosamente al sistema. Este mensaje es parte del flujo informativo de GLPI, confirmando que la operación se completó correctamente. Zabbix puede estar vinculado como un componente, integración, o herramienta de monitoreo, dependiendo de su uso en el sistema.



*Figura 7 Confirmación de Elemento Agregado en GLPI*

En el servidor Zabbix, se realiza las siguientes configuraciones:

En la sección **"Media types"** del módulo **"Alerts"** de Zabbix, se configura GLPI como un canal de notificaciones. Esta configuración permite integrar ambos sistemas, de modo que Zabbix pueda generar alertas que se traduzcan automáticamente en tickets en GLPI. Esto facilita la gestión centralizada de incidentes y asegura que las alertas críticas identificadas por Zabbix sean registradas y gestionadas eficientemente en GLPI.



*Figura 8 Configuración de Integración entre Zabbix y GLPI*

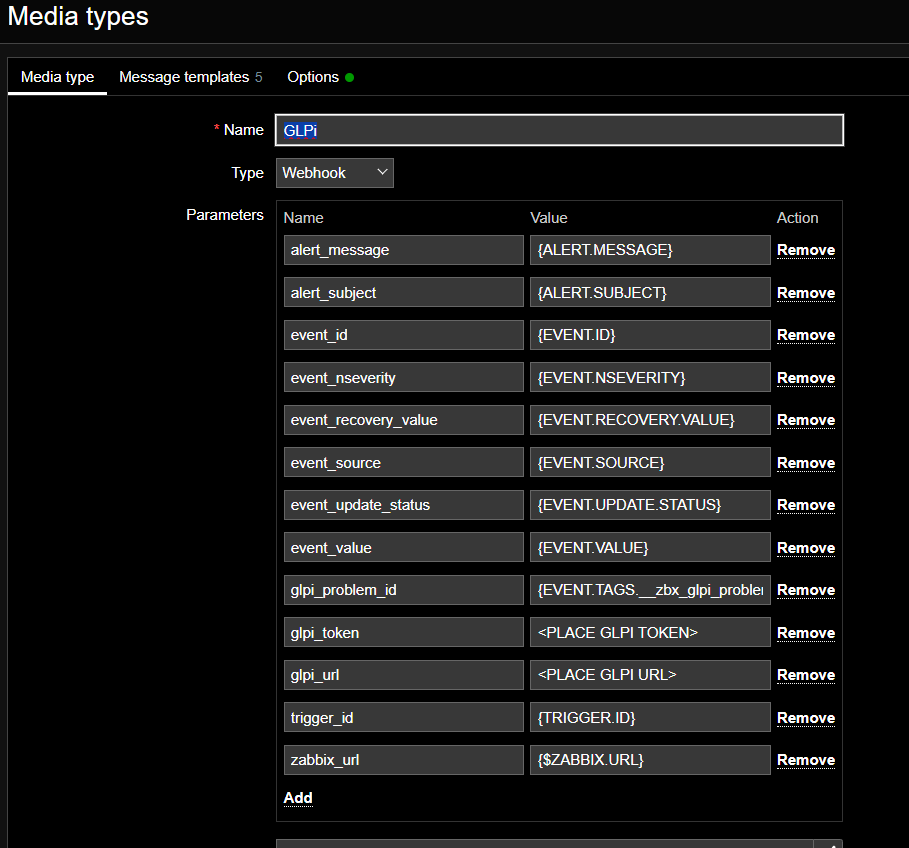
En esta pantalla de configuración de **"Media types"** en Zabbix, se define un Webhook personalizado para integrar las alertas de Zabbix con GLPI. Los parámetros configurados incluyen:

* **alert\_message** y **alert\_subject:** Utilizan los mensajes y asuntos generados por las alertas de Zabbix.

# event\_id, event\_nseverity, event\_source, event\_update\_status:

Transfieren detalles específicos del evento para su registro en GLPI.

* **glpi\_problem\_id, glpi\_token, glpi\_url:** Incluyen identificadores específicos para GLPI, como el token de acceso y la URL del sistema.
* **zabbix\_url:** Especifica la dirección del servidor Zabbix.



*Figura S Configuración del Webhook para la Integración de Zabbix con GLPI*

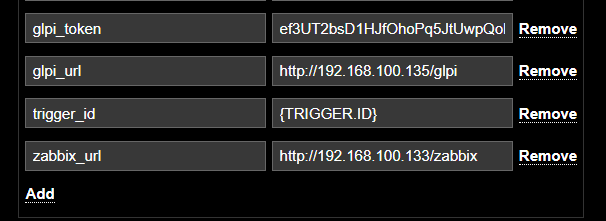
Se completan los detalles específicos para la integración entre Zabbix y GLPI en el Webhook configurado:

* **glpi\_token:** Se utiliza el token generado en GLPI para autenticar las solicitudes de Zabbix.
* **glpi\_url:** Especifica la URL del sistema GLPI

([http://192.168.100.135/glpi),](http://192.168.100.135/glpi)) donde se registrarán los tickets.

* **trigger\_id:** Representa el identificador del evento activado en Zabbix, utilizado para asociar alertas con problemas específicos.
* **zabbix\_url:** Define la URL del servidor Zabbix

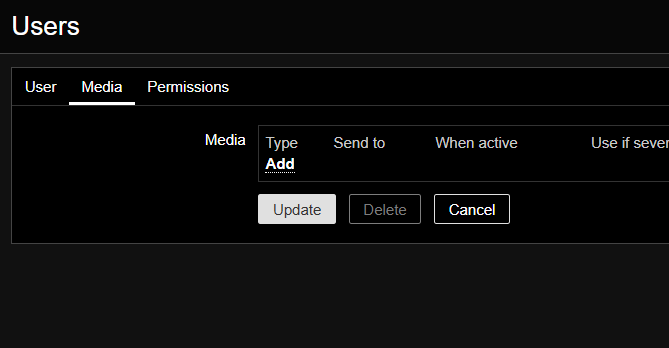
([http://192.168.100.133/zabbix),](http://192.168.100.133/zabbix)) permitiendo el enlace entre ambos sistemas.



*Figura 10 Configuración Detallada del Webhook para GLPI en Zabbix*

En esta sección de Zabbix, se asignan los tipos de medios de notificación a un usuario. Los campos configurables incluyen:

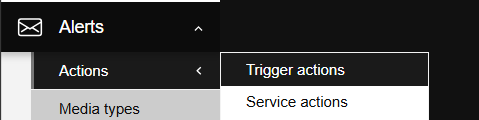
* **Type:** Selección del tipo de medio previamente configurado, como el Webhook para GLPI.
* **Send to:** Dirección o identificador donde se enviarán las notificaciones (por ejemplo, la URL o correo electrónico).
* **When active:** Define los periodos en los que este medio estará activo para enviar alertas.
* **Use if severity:** Establece las severidades de las alertas que activarán este medio de notificación.



*Figura 11 Configuración de Medios de Notificación para Usuarios en Zabbix*

En esta sección del menú **"Alerts"** de Zabbix, se accede a las opciones de **"Trigger actions"** dentro de **"Actions"**. Estas acciones se configuran para automatizar respuestas cuando se activa un disparador (trigger) debido a un evento monitoreado. Las acciones pueden incluir:

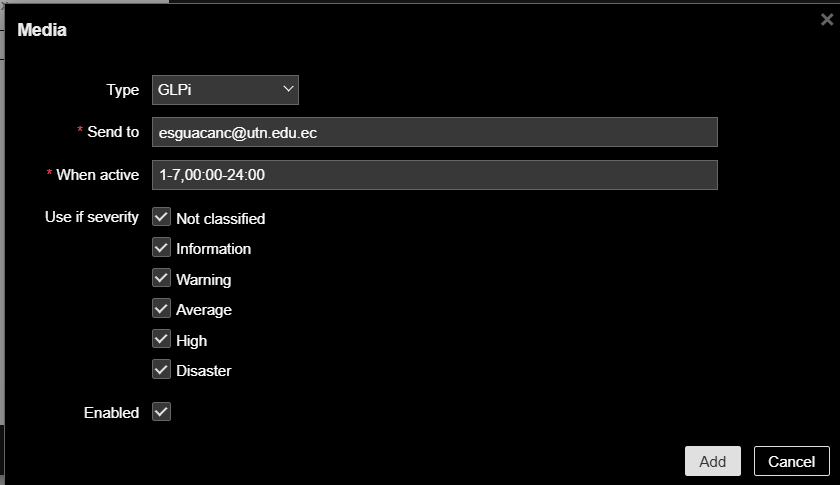
* Notificaciones a través de medios configurados, como el Webhook para integración con GLPI.
* Ejecución de scripts o comandos remotos para abordar el problema detectado.



*Figura 12 Configuración de Acciones Basadas en Triggers en Zabbix*

Se configura el medio de notificación para integrar Zabbix con GLPI. Los parámetros incluyen:

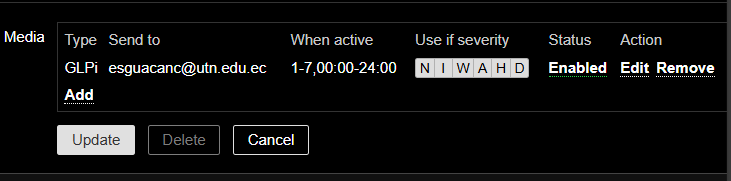
* **Type:** Selecciona el tipo de medio como "GLPI", previamente configurado en Media types.
* **Send to:** Especifica la dirección de correo o identificador que recibirá las alertas (en este caso, [esguacanc@utn.edu.ec).](mailto:esguacanc@utn.edu.ec)
* **When active:** Define el periodo en el que este medio estará activo (de lunes a domingo, 00:00-24:00).
* **Use if severity:** Permite seleccionar las severidades de los eventos que dispararán el medio, como "High" o "Disaster".
* **Enabled:** Indica si el medio está habilitado.



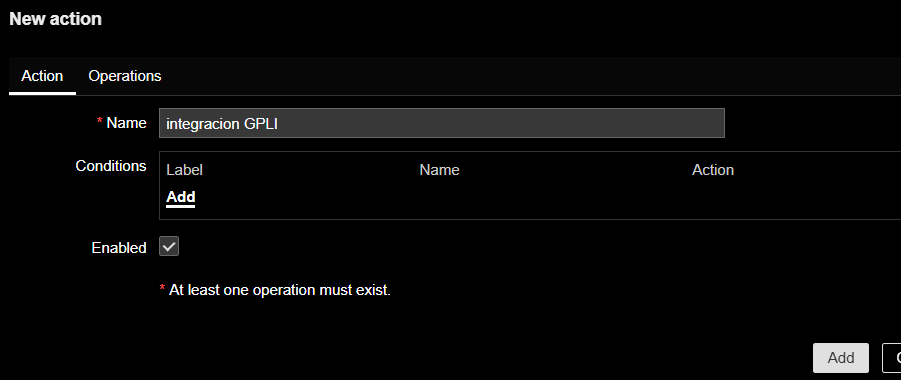
*Figura 13 Configuración del Medio de Notificación GLPI en Zabbix*

La configuración del medio de notificación para GLPI está completa y activada. Los detalles incluyen:

* + **Type:** El tipo de medio es "GLPI".
  + **Send to:** Las alertas se enviarán a [esguacanc@utn.edu.ec.](mailto:esguacanc@utn.edu.ec)
  + **When active:** Estará activo los 7 días de la semana, durante todo el día (00:00- 24:00).
  + **Use if severity:** El medio responde a severidades clasificadas como "Not classified" (N), "Information" (I), "Warning" (W), "Average" (A), "High" (H), y "Disaster" (D).
  + **Status:** Está habilitado (**Enabled**).

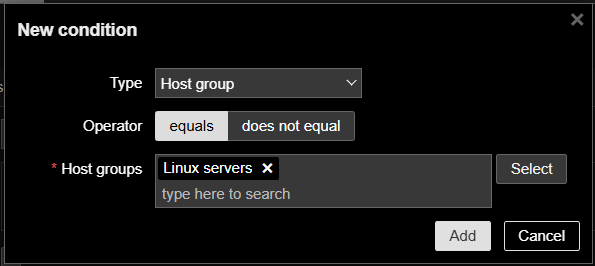


*Figura 14 Medio de Notificación GLPI Configurado en Zabbix*

En la primera imagen, se está configurando una nueva acción en Zabbix llamada **"integracion GLPI"**, con el propósito de gestionar alertas específicas que serán enviadas a GLPI. Se habilita la acción marcando la opción **Enabled**.

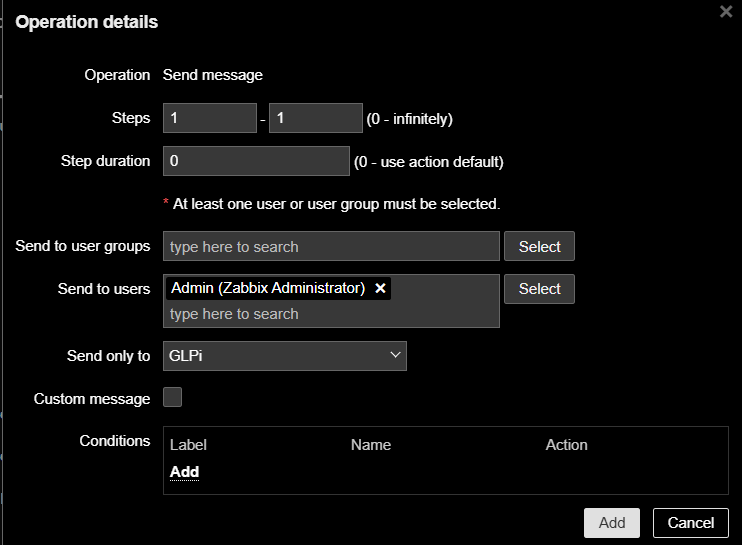
*Figura 15 Configuración de Nueva Acción para la Integración con GLPI*

Se define una condición para la acción previamente creada. La condición seleccionada está basada en el tipo **Host group**, donde el operador **equals** asegura que la acción solo se ejecutará para los grupos de hosts que coincidan, en este caso, **Linux servers**. Esto permite filtrar eventos específicos que serán procesados y enviados a GLPI.



*Figura 1c Configuración de Condición para la Acción en Zabbix*

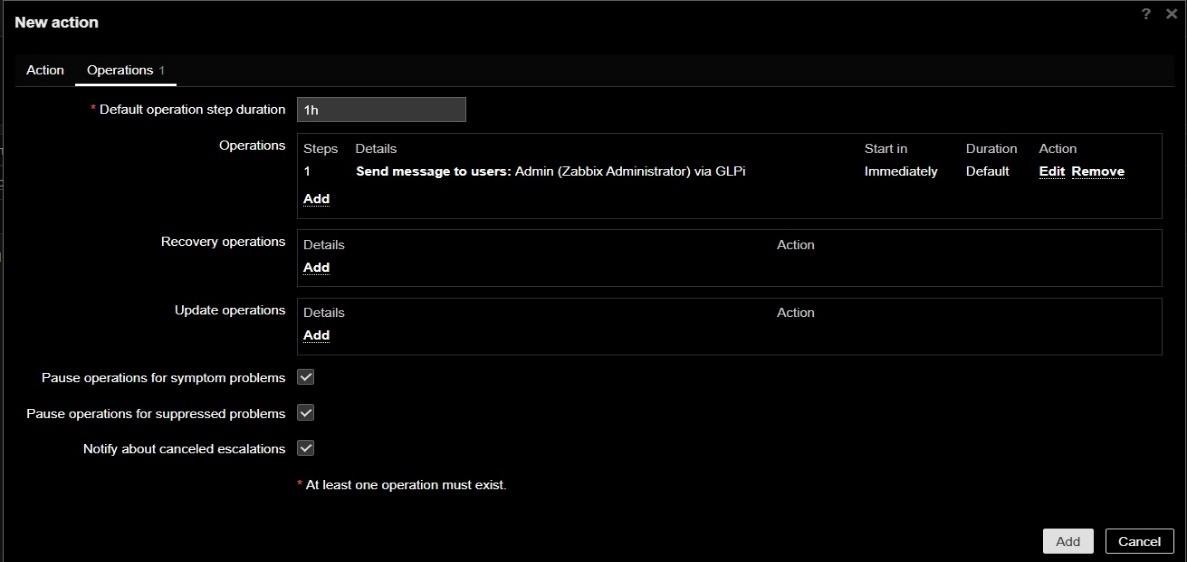
En operaciones, las configuraciones, se define la operación asociada a la acción de integración con GLPI. Los parámetros incluyen:

* + **Operation:** Selecciona "Send message" para enviar una alerta como mensaje.
  + **Steps:** Especifica que la operación se ejecutará en el paso 1.
  + **Send to users:** Se selecciona al usuario "Admin (Zabbix Administrator)" para recibir la notificación.
  + **Send only to:** Indica que el mensaje será enviado exclusivamente al medio configurado como "GLPI".
  + **Custom message:** Permite incluir un mensaje personalizado, aunque no está activado en este caso.

*Figura 17 Detalles de la Operación para la Integración de GLPI en Zabbix*

Se configura una nueva acción en Zabbix para enviar notificaciones a través de GLPI. Los ajustes incluyen:

* + **Default operation step duration:** Se establece la duración de cada paso en 1 hora.
  + **Operations:** En el paso 1, se define el envío de un mensaje al usuario "Admin (Zabbix Administrator)" utilizando el medio de notificación configurado como "GLPI".
  + **Start in:** La operación se inicia de forma inmediata.
  + **Recovery operations y Update operations:** Estas secciones permiten definir acciones adicionales para recuperación o actualizaciones, pero están vacías en este caso.
  + **Notificaciones adicionales:** Se activan opciones como pausar operaciones para problemas sintomáticos o suprimidos y notificar sobre cancelaciones de escaladas.



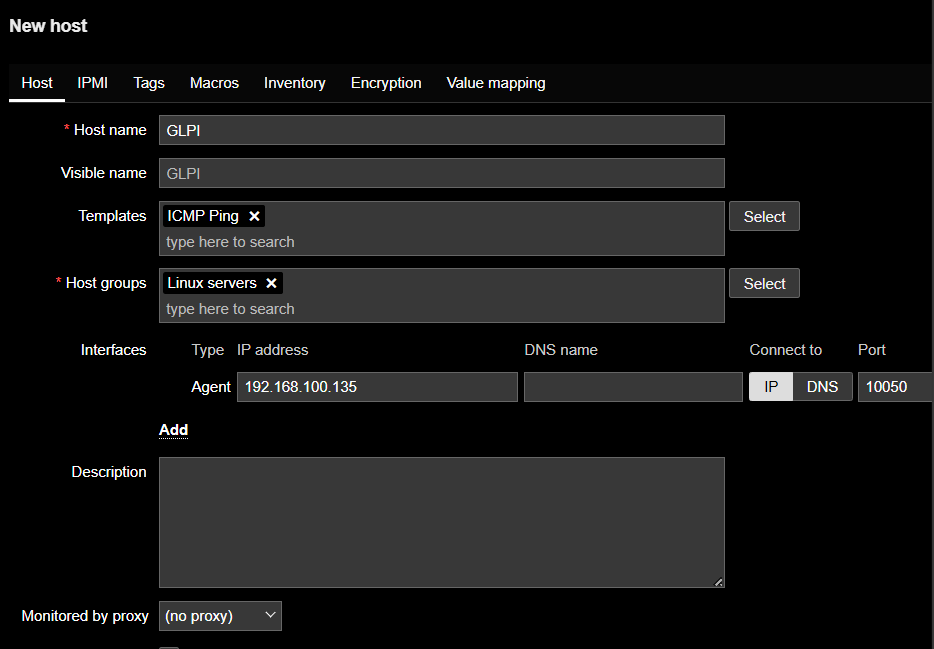
*Figura 18 Configuración Completa de una Acción para GLPI en Zabbix*

Se crea un nuevo host, por tanto, en la pantalla se configura un nuevo host en Zabbix para monitorear el servidor de GLPI. Los parámetros establecidos son:

* + **Host name:** Se define el nombre del host como "GLPI".
  + **Visible name:** También se establece como "GLPI" para facilitar su identificación.
  + **Templates:** Se asigna el template "ICMP Ping" para realizar monitoreos básicos de disponibilidad.
  + **Host groups:** Se agrupa al host en "Linux servers" para clasificarlo dentro de los servidores basados en Linux.
  + **Interfaces:** Se especifica el tipo "Agent" con la dirección IP 192.168.100.135

y el puerto predeterminado 10050.

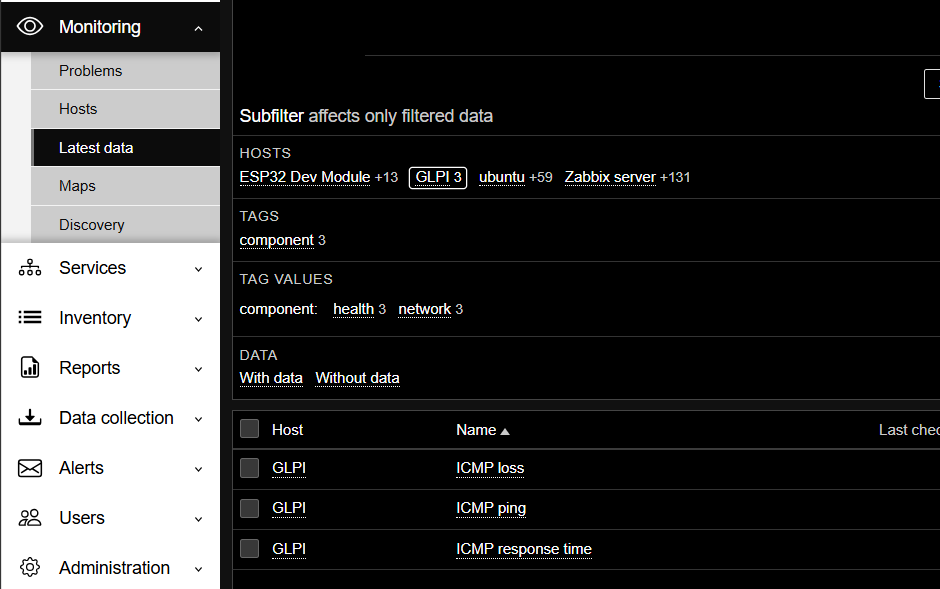
* + **Monitored by proxy:** No se selecciona proxy para el monitoreo.



*Figura 1S Configuración de un Nuevo Host para GLPI en Zabbix*

En esta sección de monitoreo de Zabbix, se muestran los datos más recientes recopilados para el host **GLPI**. Los elementos destacados incluyen:

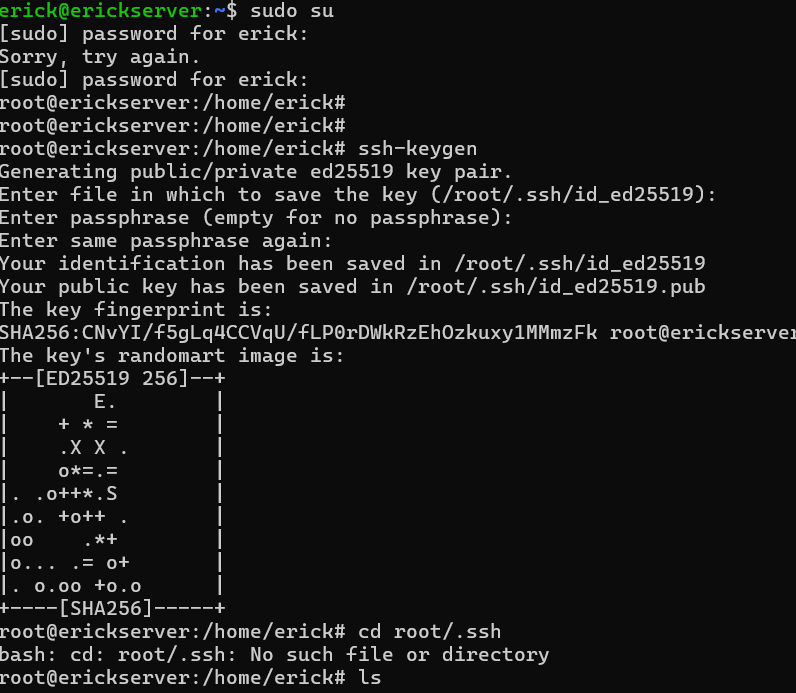
* + **HOSTS:** Lista de hosts monitoreados, con "GLPI" seleccionado para filtrar los datos específicos de este servidor.
  + **Data:** Visualización de métricas relacionadas con el host "GLPI", como:
    - **ICMP loss:** Pérdida de paquetes ICMP.
    - **ICMP ping:** Estado del ping al host.
    - **ICMP response time:** Tiempo de respuesta ICMP



*Figura 20 Visualización de Datos Recientes de GLPI en Zabbix*

# IA GPT4ALL

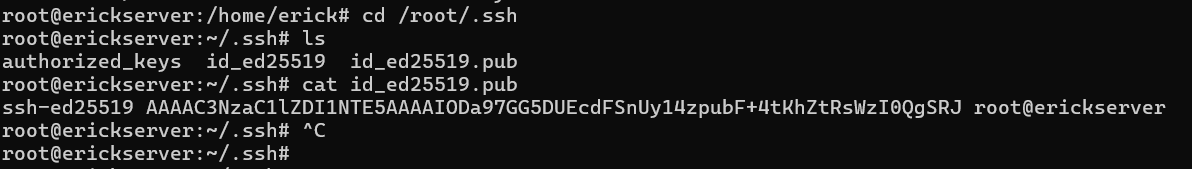
Se inicia con generación de un par de claves SSH en la máquina física (Windows) utilizando el comando ssh-keygen, donde se especifica la clave de tipo **ed25519**. Aunque la generación es exitosa y las claves privadas y públicas se guardan en la ruta predeterminada (/root/.ssh/id\_ed25519), al intentar acceder al directorio /root/.ssh, el sistema reporta que no existe, indicando que es necesario crearlo manualmente para almacenar las claves y continuar con la configuración segura del acceso SSH.



*Figura 2 Generación de claves*

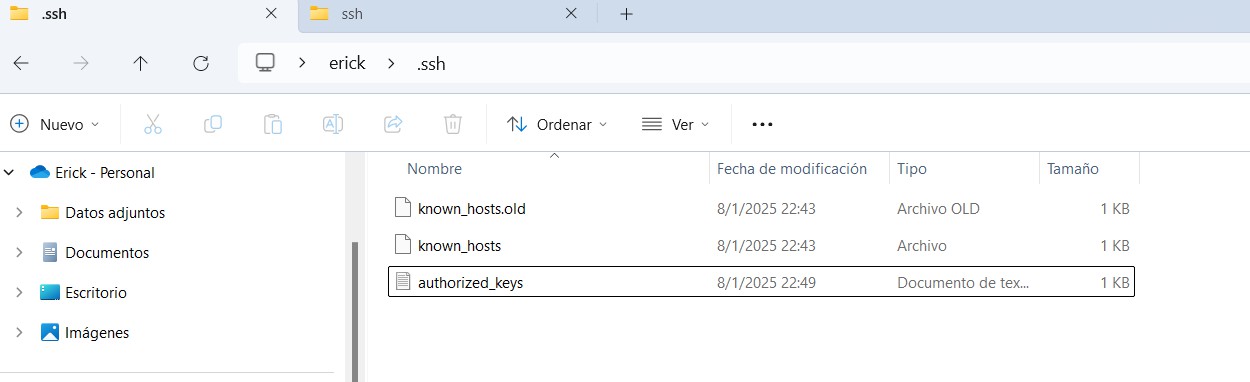
Se verifica el contenido del directorio /root/.ssh, donde están almacenadas las claves SSH generadas: id\_ed25519 (clave privada) y id\_ed25519.pub (clave pública).

Además, se utiliza el comando cat para visualizar el contenido de la clave pública id\_ed25519.pub, que contiene el formato estándar para su uso en autenticación SSH segura. Este archivo es el que debe ser compartido con los servidores o sistemas donde se desea establecer el acceso seguro.



*Figura 3 Verificación de claves*

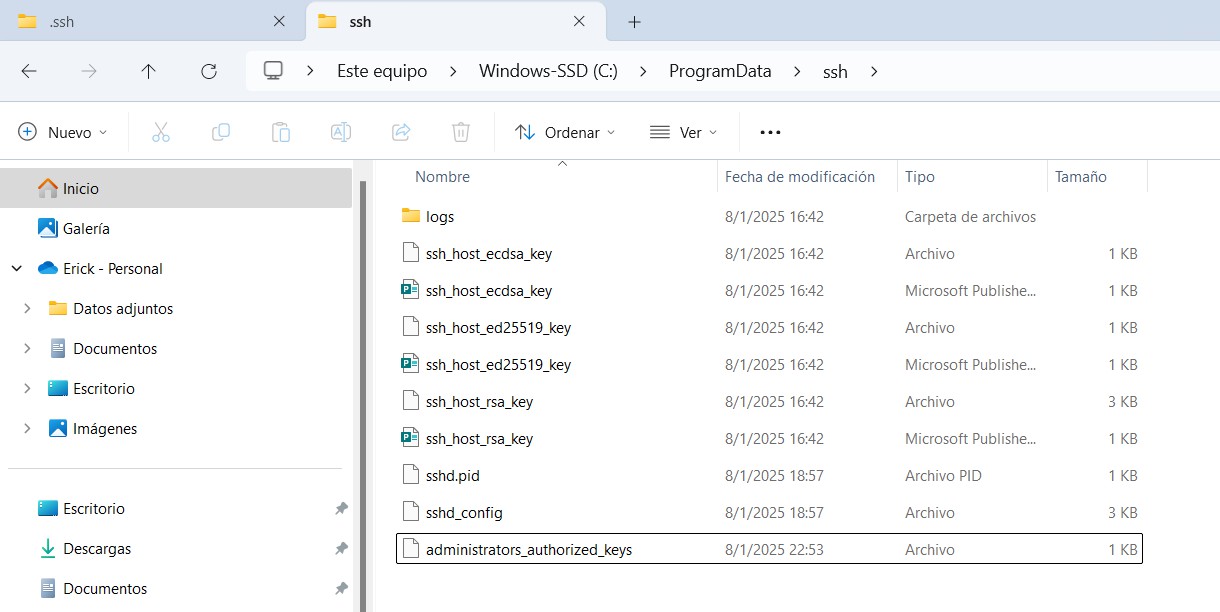
Se verifica contenido del directorio .ssh en el sistema de archivos del usuario. En él se encuentran los archivos known\_hosts y su copia de respaldo known\_hosts.old, que contienen información sobre los servidores conocidos para conexiones SSH. Además, tiene el archivo authorized\_keys, que almacena las claves públicas autorizadas para acceder al sistema de manera segura.



*Figura 4 Verificar contenido del directorio*

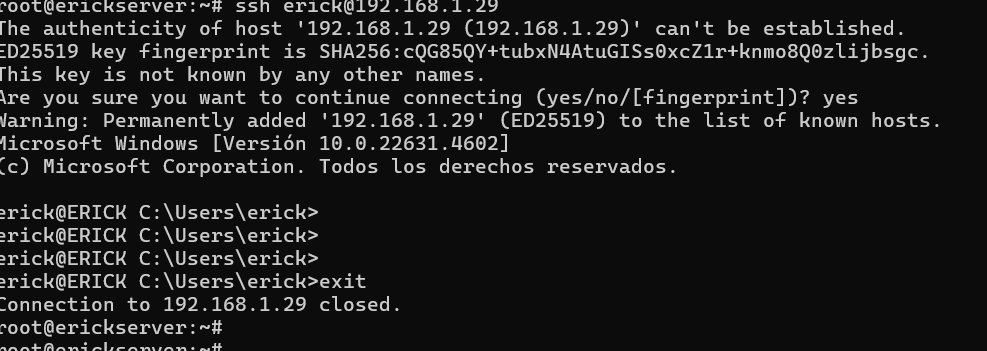
Se muestra el contenido del directorio SSH, que se encuentra:

* **Archivos de claves del host** (ssh\_host\_ecdsa\_key, ssh\_host\_ed25519\_key, ssh\_host\_rsa\_key), que identifican al servidor en conexiones SSH.
* **Configuración del servicio SSH** (sshd\_config), que define parámetros como puertos y métodos de autenticación.
* **Archivo administrators\_authorized\_keys**, que contiene las claves públicas autorizadas para administradores, permitiendo su acceso seguro al sistema.
* El archivo sshd.pid indica que el demonio SSH está activo.



*Figura 21 Servicio SSH en sistemas Windows*

Muestra una conexión SSH exitosa desde un servidor a un host Windows (192.168.1.29), donde se agrega la clave del host a la lista de hosts conocidos (known\_hosts) para futuras conexiones seguras.

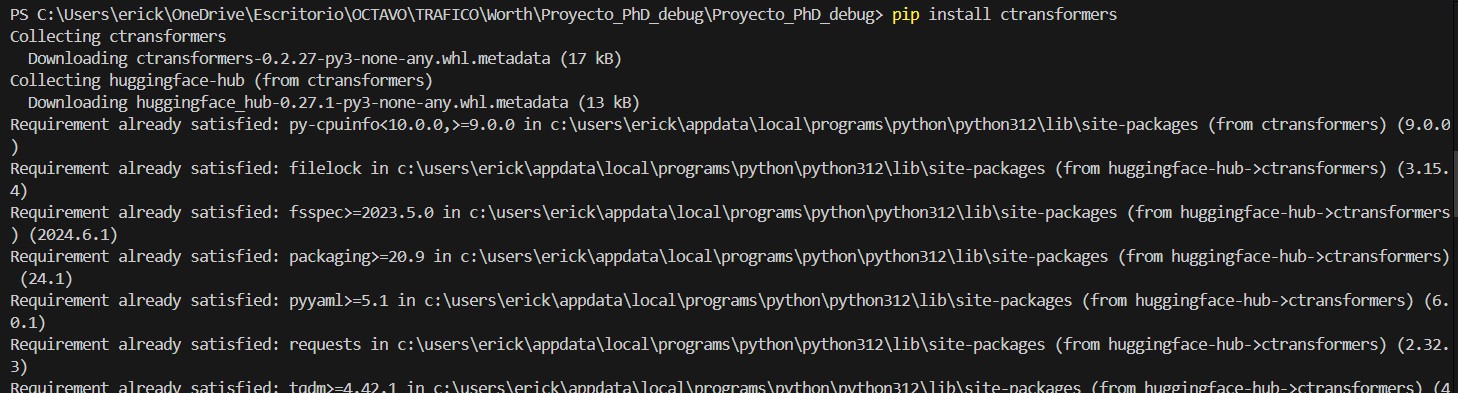


*Figura 22 Establecimiento de conexión*

En este paso se ejecuta el comando `pip install transformers` para instalar el paquete

`transformers`, necesario para trabajar con modelos de lenguaje natural. El instalador descarga el paquete principal desde el repositorio de Python (PyPI) junto con las dependencias requeridas, como `huggingface-hub`, `filelock`, `requests` y otras bibliotecas relacionadas.

El proceso verifica si las dependencias ya están instaladas en el sistema. En este caso, muchas de ellas ya se encuentran disponibles, por lo que no se descargan nuevamente, lo que optimiza el tiempo de instalación. Al finalizar, el paquete `transformers` queda instalado en el entorno Python para su uso posterior.



*Figura 5 Instalación del paquete transformers mediante pip*

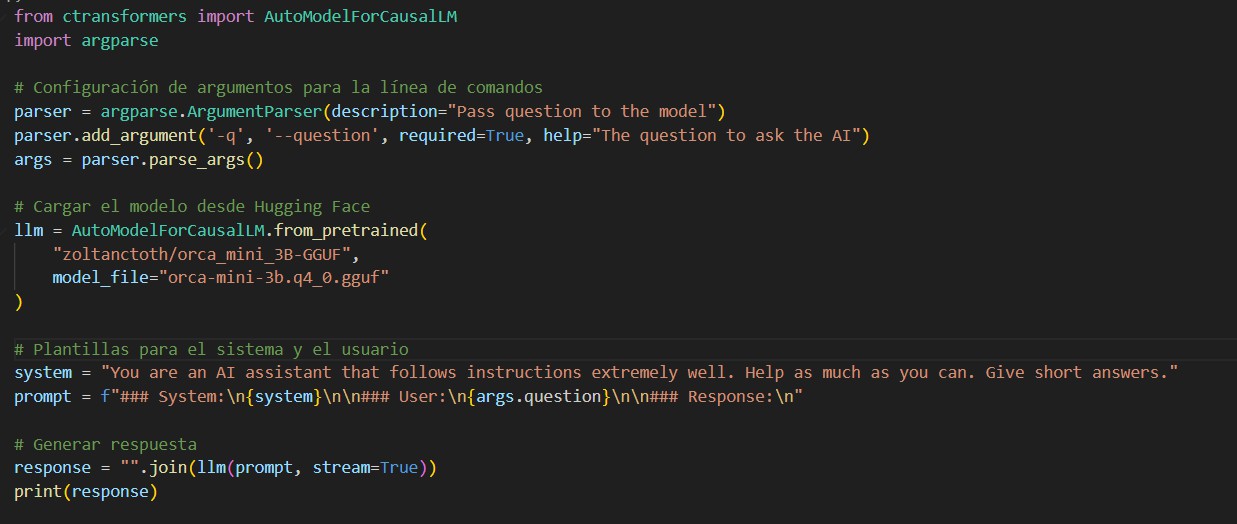
Se importa el módulo AutoModelForCausalLM de la biblioteca ctransformers para trabajar con modelos de lenguaje preentrenados. También se importa el módulo argparse para manejar argumentos desde la línea de comandos.

Se configura un analizador de argumentos mediante argparse.ArgumentParser, donde se define un argumento llamado --question o -q como obligatorio. Este argumento permite al usuario ingresar una pregunta desde la línea de comandos que será procesada por el modelo de lenguaje.

Se carga un modelo preentrenado utilizando la función AutoModelForCausalLM.from\_pretrained, especificando el modelo zoltanctoth/orca\_mini\_3B\_GGUF y el archivo asociado orca-mini-3b.q4\_0.gguf.

Se define una plantilla de texto que incluye una descripción del sistema, especificando que es un asistente de inteligencia artificial. En la misma plantilla se incluye la pregunta proporcionada por el usuario, preparando el texto para ser procesado por el modelo.

Se genera una respuesta del modelo utilizando el texto de la plantilla como entrada. La respuesta generada se une en una cadena y se imprime en la salida estándar.



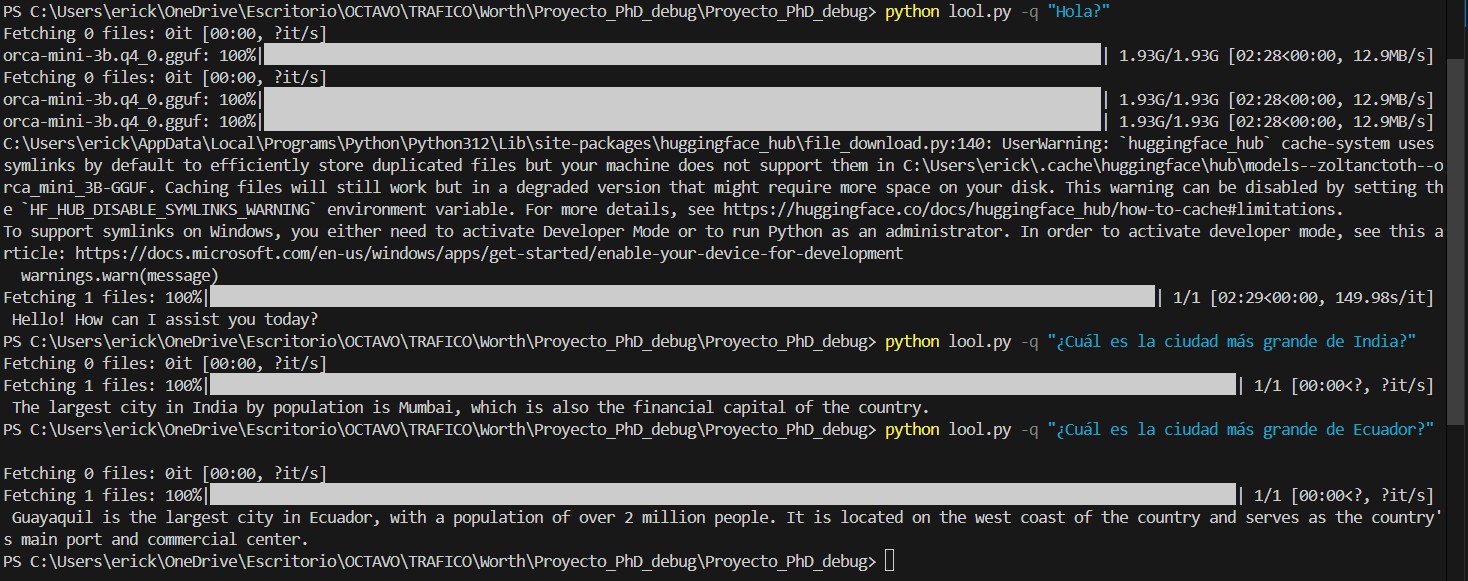
*Figura 6 Carga y ejecución del modelo de lenguaje desde el script en Python*

Se ejecuta el script Python llamado lool.py con el argumento -q para enviar diferentes preguntas al modelo de lenguaje cargado. La primera pregunta ingresada es "Hola", y se envía al modelo para generar una respuesta.

Se descarga el archivo del modelo orca-mini-3b.q4\_0.gguf desde Hugging Face. Este archivo tiene un tamaño de aproximadamente 1.93 GB, y el progreso de la descarga se muestra en tiempo real. Durante este proceso, se advierte sobre posibles limitaciones en el uso de enlaces simbólicos en el sistema operativo Windows.

Se envía la pregunta "¿Cuál es la ciudad más grande de India?" al modelo de lenguaje, utilizando el mismo script. El modelo genera una respuesta que indica que la ciudad más grande por población es Mumbai, conocida también como la capital financiera del país.

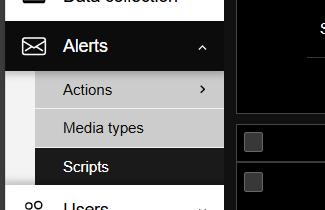
Se realiza otra consulta al modelo preguntando "¿Cuál es la ciudad más grande de Ecuador?". El modelo responde que la ciudad más grande es Guayaquil, con una población de más de dos millones de personas, destacando su importancia como centro portuario y comercial principal del país.



*Figura 7 Pruebas de consulta al modelo con preguntas específicas*

En este paso se accede al menú de configuración de alertas en Zabbix. Dentro de la sección "Alerts", se seleccionan las opciones "Actions" y "Media types".

La opción "Actions" permite configurar las reglas y condiciones bajo las cuales se ejecutan acciones automatizadas en respuesta a eventos o problemas detectados por Zabbix. La opción "Media types" permite gestionar y definir los métodos de notificación disponibles de integraciones con scripts personalizados.



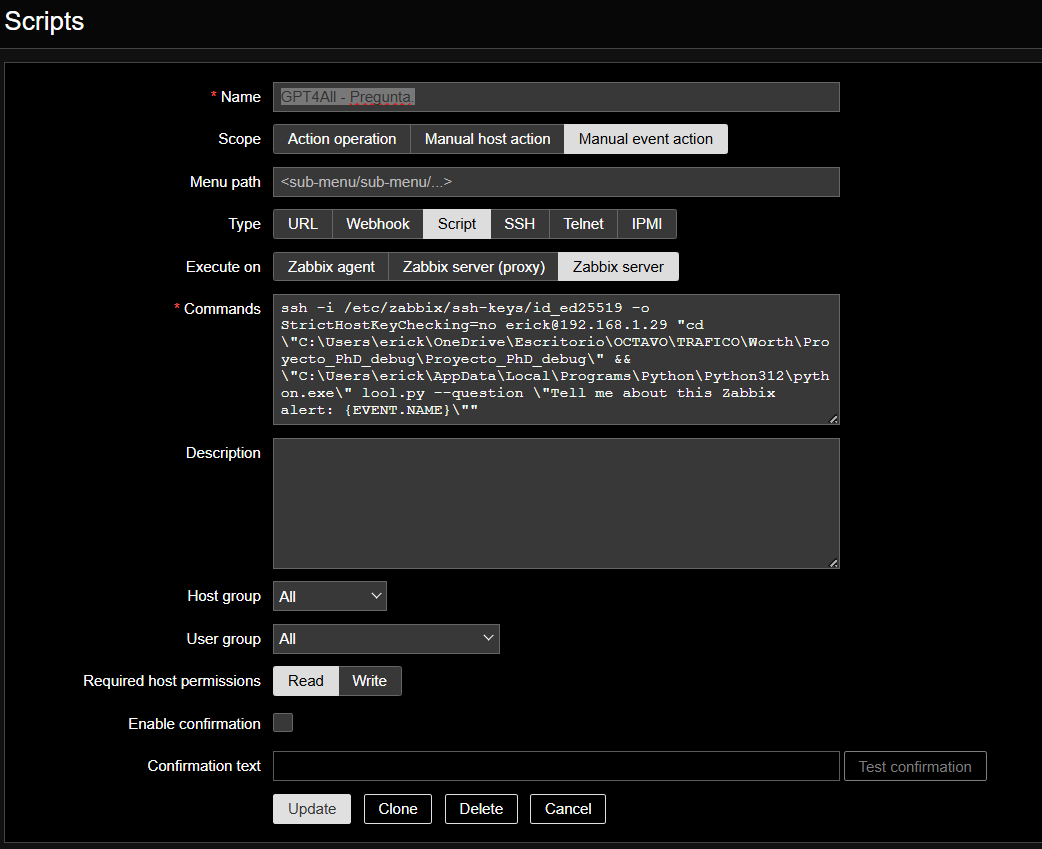
*Figura 8 Acceso al menú de configuración de alertas en Zabbix*

En este paso se configura un script dentro de Zabbix llamado GPT4All - Pregunta. Este script está diseñado para integrarse con un modelo de lenguaje y procesar preguntas relacionadas con eventos en Zabbix.

Se selecciona el ámbito de ejecución para incluir operaciones de acción, acciones manuales en hosts y acciones manuales en eventos. Esto permite que el script pueda ser utilizado tanto de forma manual como automática dentro de las operaciones configuradas en Zabbix.

El tipo de script se define como Script, indicando que ejecutará un comando personalizado. Además, se establece que el script será ejecutado directamente en el servidor de Zabbix, seleccionando la opción Zabbix server.

El comando configurado utiliza una conexión SSH para acceder a un servidor remoto. Dentro de este servidor, se navega al directorio del proyecto y se ejecuta un script en Python llamado lool.py. Este script recibe como argumento el nombre del evento que está siendo procesado, utilizando el parámetro de Zabbix {EVENT.NAME}, para enviar la información al modelo de lenguaje.



*Figura S Configuración de un script personalizado en Zabbix para integración con GPT4All*

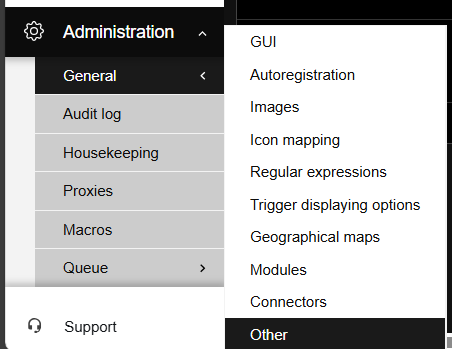
El campo de descripción no se utiliza en esta configuración, y se define que todos los grupos de hosts y usuarios tienen acceso al script, siempre que tengan permisos de lectura y escritura. La opción de confirmación no está habilitada, lo que permite que el script se ejecute directamente sin pasos adicionales de validación.

En este paso se accede al menú de administración de Zabbix, específicamente a la sección General. Dentro de esta sección, se despliega un submenú con diversas opciones para la configuración del sistema.

Las opciones disponibles incluyen GUI, que permite personalizar la interfaz gráfica del usuario. La opción Autoregistration facilita la configuración para el registro automático de hosts en Zabbix. Images e Icon mapping permiten gestionar imágenes e iconos asociados a elementos de monitoreo, respectivamente.

Regular expressions ofrece la posibilidad de crear y gestionar expresiones regulares utilizadas en diferentes configuraciones del sistema. Trigger displaying options permite personalizar cómo se muestran los disparadores en la interfaz. Geographical maps habilita la configuración y visualización de mapas geográficos.

Modules y Connectors ofrecen opciones para gestionar módulos y conectores adicionales que extienden las funcionalidades del sistema. Finalmente, la opción Other agrupa configuraciones adicionales que no se encuentran en las categorías anteriores.



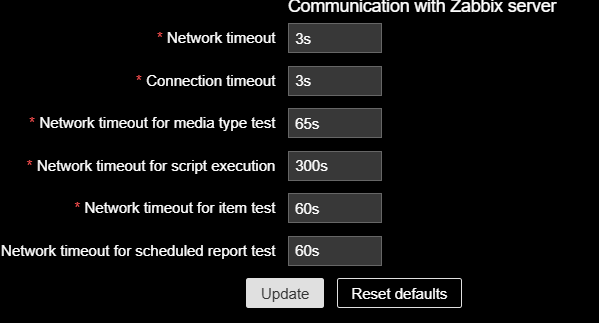
*Figura 10 Acceso a las opciones de administración general en Zabbix*

En este paso se configuran los parámetros de tiempo de espera relacionados con la comunicación con el servidor de Zabbix. Estos valores determinan cuánto tiempo esperará el sistema antes de que una operación de red sea considerada fallida.

El tiempo de espera general de la red se establece en 3 segundos, al igual que el tiempo de espera para la conexión. El tiempo de espera de red para pruebas de tipos de medios se configura en 65 segundos, lo que permite un mayor margen de tiempo para estas operaciones específicas.

El tiempo de espera de red para la ejecución de scripts se define en 300 segundos, dado que las ejecuciones de scripts pueden requerir más tiempo. El tiempo de espera de red para pruebas de ítems se establece en 60 segundos, al igual que el tiempo de espera para pruebas de reportes programados.

Se proporcionan botones para actualizar la configuración o restablecer los valores predeterminados, permitiendo flexibilidad en la personalización o la recuperación de la configuración original según sea necesario.



*Figura 11 Configuración de tiempos de espera en la comunicación con el servidor de Zabbix*

Se inicia accediendo al directorio /etc/zabbix/ y se crea un subdirectorio llamado ssh-keys utilizando el comando mkdir. Este directorio servirá para almacenar las claves SSH necesarias para la autenticación en las operaciones relacionadas con Zabbix.

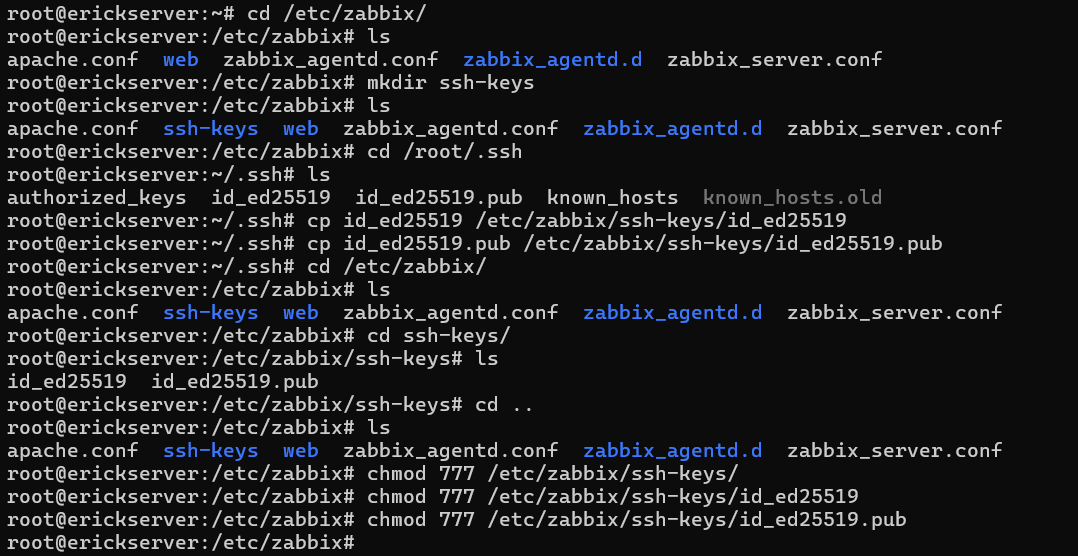
Posteriormente, se navega al directorio .ssh del usuario root donde se encuentran las claves SSH previamente generadas, incluyendo los archivos id\_ed25519 (clave privada) e id\_ed25519.pub (clave pública).

Las claves privadas y públicas se copian desde el directorio .ssh al nuevo directorio

/etc/zabbix/ssh-keys utilizando el comando cp. Esto asegura que las claves estén disponibles en una ubicación específica para Zabbix, facilitando su uso en operaciones relacionadas con el monitoreo.

Se verifica que las claves se hayan copiado correctamente revisando el contenido del directorio ssh-keys con el comando ls. Luego, se ajustan los permisos de los archivos en el directorio ssh-keys a 777 mediante el comando chmod, lo que otorga acceso completo a estos archivos.

Finalmente, se realiza una verificación adicional del contenido del directorio /etc/zabbix/ para confirmar que el subdirectorio ssh-keys y sus contenidos están configurados según lo esperado. Este proceso asegura que las claves SSH estén correctamente configuradas y disponibles para las operaciones de Zabbix.



*Figura 12 Copia y asignación de permisos a las claves SSH en Zabbix*

La imagen muestra el panel de problemas en **Zabbix**, donde se registran alertas de severidad alta relacionadas con la indisponibilidad de hosts detectada mediante ICMP ping. Cada registro incluye información detallada como la hora del evento, la duración del problema y etiquetas clasificatorias (por ejemplo, clase y componente).

Este panel facilita a los administradores la priorización y gestión de incidentes críticos en la infraestructura de red.

En la columna de severidad, todos los problemas están clasificados como "High". El estado de cada problema aparece marcado como "PROBLEM" en color rojo, indicando que aún no se ha resuelto. Los problemas están asociados a diferentes hosts: GLPI, ESP32 Dev Module y Ubuntu.

La descripción del problema indica que los hosts están "Unavailable by ICMP ping", lo que significa que no responden a los paquetes ICMP utilizados para verificar su disponibilidad en la red. Cada problema incluye la duración desde su detección y un enlace para realizar actualizaciones o acciones relacionadas con la resolución.

En la columna de etiquetas, se destacan las clasificaciones "class: network", "component: health" y "component: network", que proporcionan un contexto adicional sobre los problemas detectados. En la parte inferior, se indica que se están mostrando tres problemas detectados.



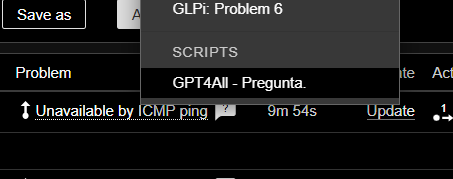
*Figura 13 Visualización de problemas detectados en Zabbix*

En el menú de menú de **Zabbix** se puede seleccionar un script llamado **"GPT4All - Pregunta"**. El cual, se asocia con la integración de **GPT4All** para proporcionar respuestas inteligentes o análisis sobre problemas detectados, como la indisponibilidad de un host debido a fallos de ICMP ping.

En este paso se observa el uso de un menú contextual desplegable en la sección de problemas detectados en Zabbix. Este menú se activa sobre un problema específico que indica "Unavailable by ICMP ping", el cual está asociado a un host.

El menú contextual muestra opciones disponibles para gestionar o interactuar con el problema seleccionado. Una de las opciones pertenece a la categoría "Scripts", donde aparece el script configurado previamente con el nombre "GPT4All - Pregunta". Al seleccionar esta opción, se ejecuta el script personalizado que envía la información del problema al modelo GPT4All para generar una respuesta o análisis relacionado con la alerta.

Este menú proporciona una interfaz directa para invocar scripts personalizados y facilitar la resolución o análisis de los problemas detectados en el sistema.

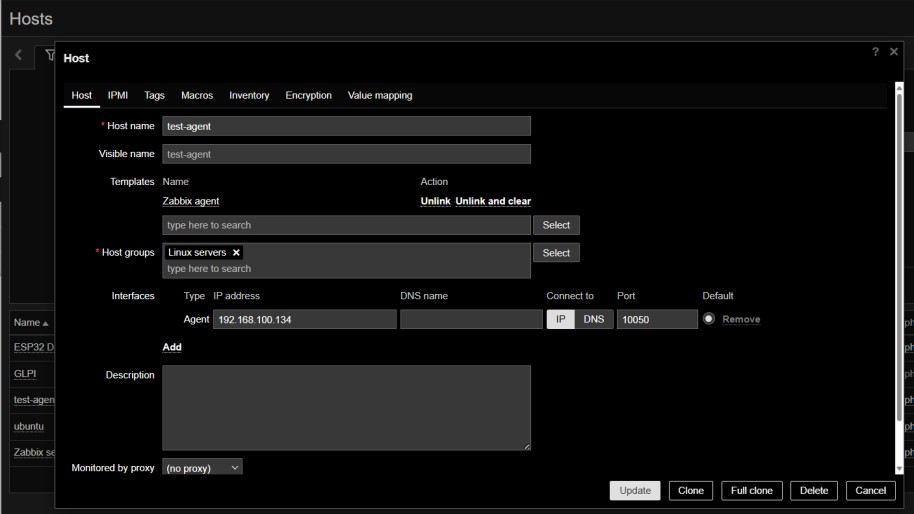


*Figura 23 Seleccionar el scripFigura 14 Ejecución del script GPT4All desde el menú contextual de Zabbix*

# Implementar gestor de logs ‘Wazuh’

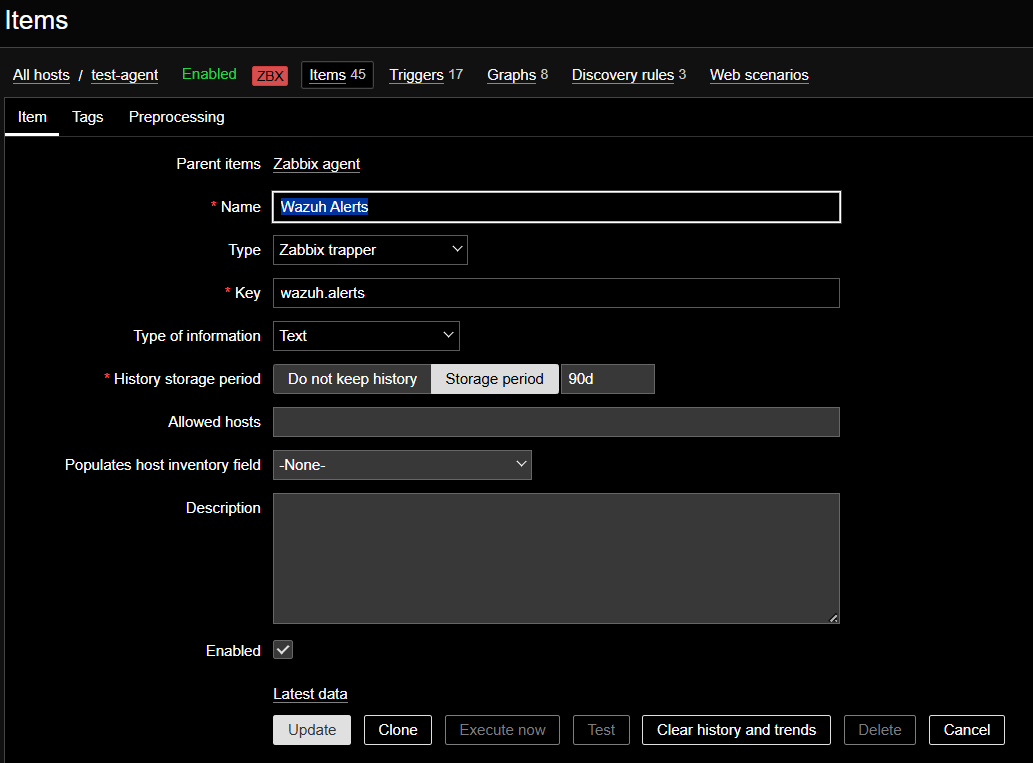
En zabbix se crea un host para la integración con Wazuh, mediante la creación de un agente denominado "test-agent", asociado al grupo "Linux servers". Se especifica la IP del agente (192.168.100.134) como interfaz principal con conexión al puerto 10050.

Esta configuración permite a Zabbix monitorear el agente vinculado a Wazuh para recibir datos de alertas



*Figura 24 Creación de host en Zabbix*

Se define el ítem "Wazuh Alerts" como un **Zabbix trapper**, con la clave wazuh.alerts, para recibir alertas generadas por Wazuh. Este ítem almacena la información como texto y se establece un periodo de retención de 90 días.



*Figura 25 Definición de item en Zabbix*

Verificar del agente e iteam creado en zabbix.



*Figura 26 Item habilitado*

Se procede a instalar el Wazuh en una máquina virtual con direcionamento de 192.168.100.134, y se descarga el script de instalación de Wazuh desde la URL oficial (https://packages.wazuh.com/4.3/wazuh-install.sh). Este script simplifica la instalación

del servidor Wazuh, configurando los componentes necesarios para su funcionamiento, lo que constituye el primer paso en la integración con otras plataformas como Zabbix.



*Figura 27 Descarga de script de instalación de Wazuh*

Se ejecuta el script descargado wazuh-install.sh con permisos de superusuario mediante sudo bash, utilizando la opción -a para realizar una instalación automática de Wazuh.

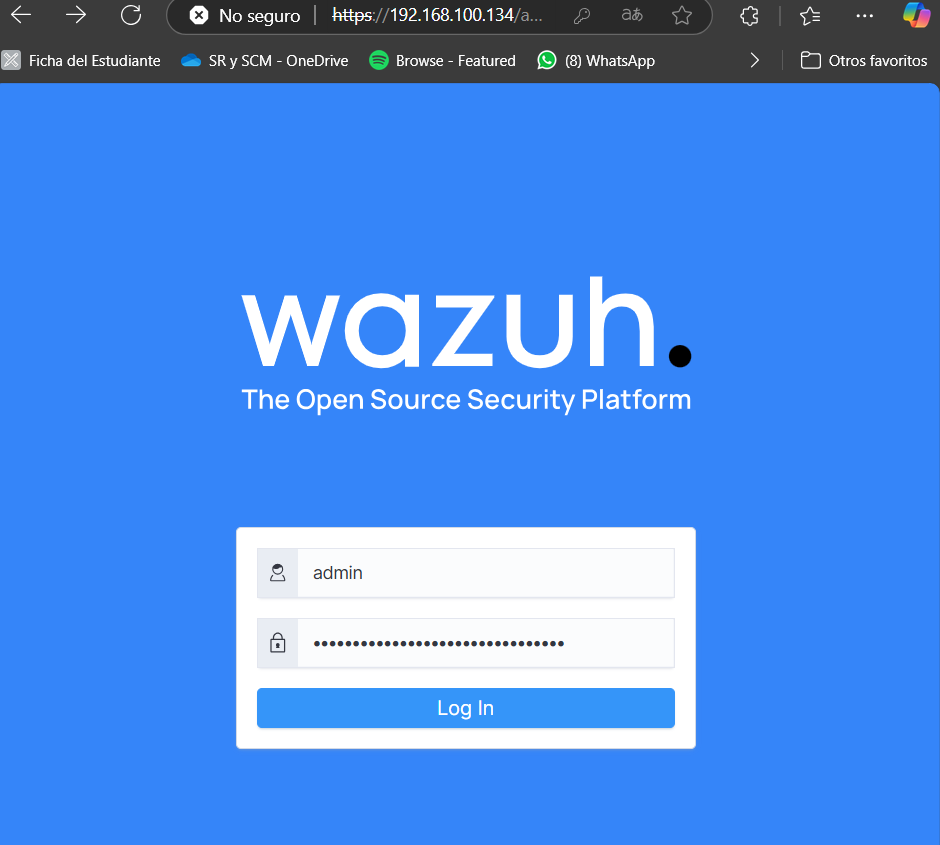


*Figura 28 Ejecución del script*

Al acabar de instalarse se verifica desde el navegador el acceso local, además se crea un usuario y una contraseña y ya se puede verificar en el navegador web:

User: admin

Password: 5\*R8YHby\*Tp47eqCSA2iR2LMRk9kxZsT



*Figura 29 Instalación de Wazuh*

Se modifica el archivo de configuración principal de Wazuh, ossec.conf, ubicado en

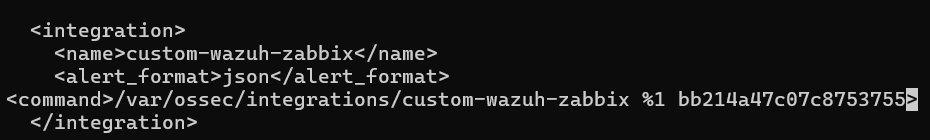
/var/ossec/etc/, para la integración luego se restablece el wazuh-manager.



*Figura 30 Ingresar al archivo principal de configuración*

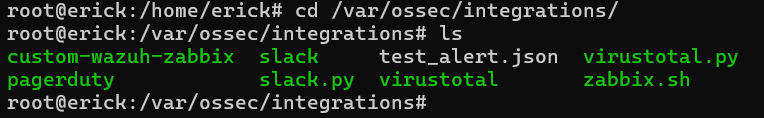
Se realiza una integración personalizada en el archivo **ossec.conf**, donde se define la integración **custom-wazuh-zabbix**. Se especifica el formato de alertas en **JSON** y el comando a ejecutar: **/var/ossec/integrations/custom-wazuh-zabbix %1** junto con una clave o token.

Esto permite a Wazuh enviar alertas procesadas directamente a Zabbix, facilitando la sincronización y el monitoreo centralizado de eventos de seguridad.



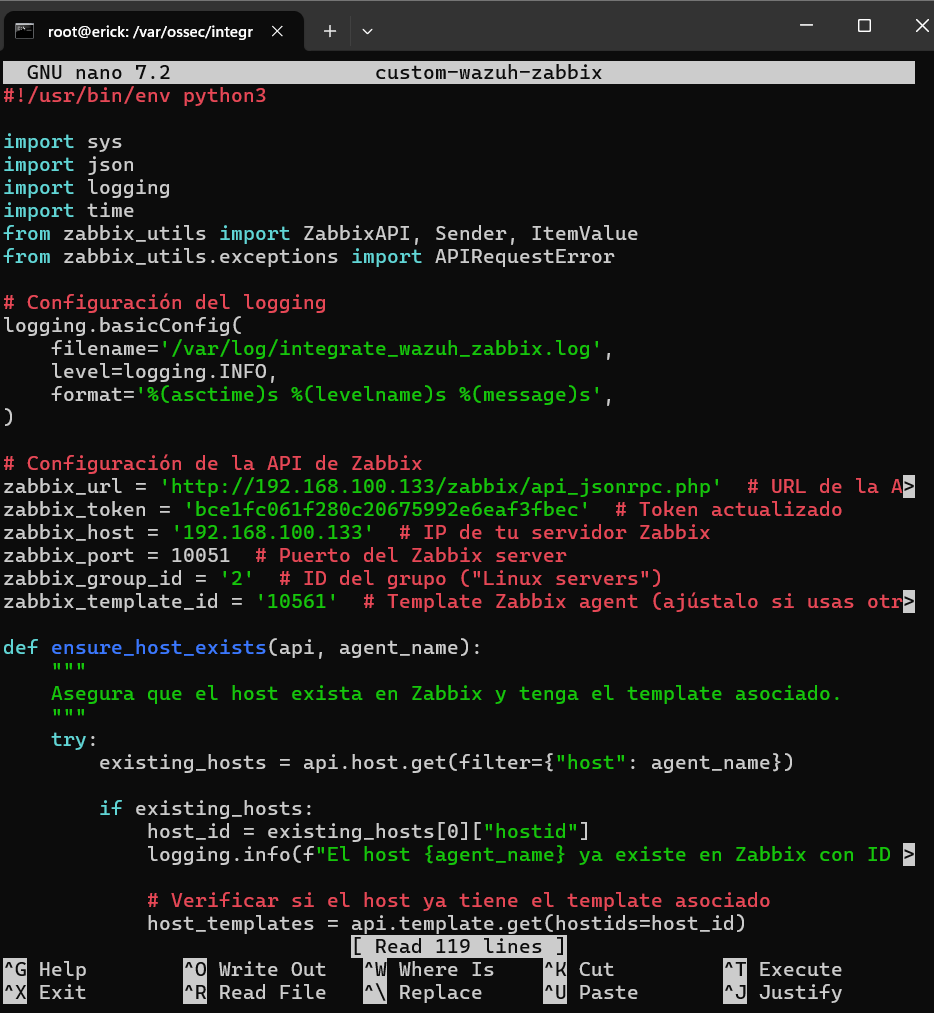
*Figura 31 Configuración del archivo*

Se crea una nueva carpeta y allí se crea un script que son: custom-wazuh-zabbix y el test\_alert.json



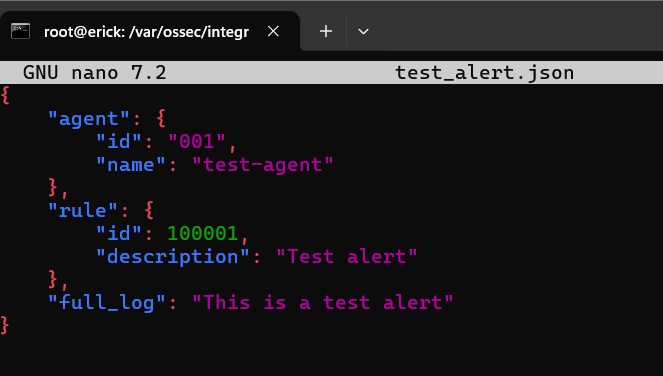
*Figura 32 Creación de scripts*

En el script **custom-wazuh-zabbix** están las configuraciones para la integración de Wazuh con Zabbix, por tanto, se configura parámetros de: URL de la API, token de autenticación, dirección del servidor, puerto, grupo y template en Zabbix.



*Figura 33 Configuración del archivo custom*

El archivo **test\_alert.json** contiene una alerta de prueba para validar la integración entre Wazuh y Zabbix, definiendo un agente con ID "001" y nombre "test-agent", una regla con ID "100001" y descripción "Test alert", y un mensaje completo de alerta con el texto "This is a test alert", lo que permite simular el funcionamiento de las alertas en un entorno controlado.



*Figura 34 Configuración del archivo .json*

A continuación, a ambos archivos que fueron configurados se les proporciona permisos de ejecución.



*Figura 35 Dar permisos al archivo custom*

**

*Figura 36 Dar permisos al archivo .json*

En un entorno virtual que se encuentre activado, se ejecuta el comando **pip install zabbix- utils,** para instalar dicho paquete y tener la integración con zabbix.



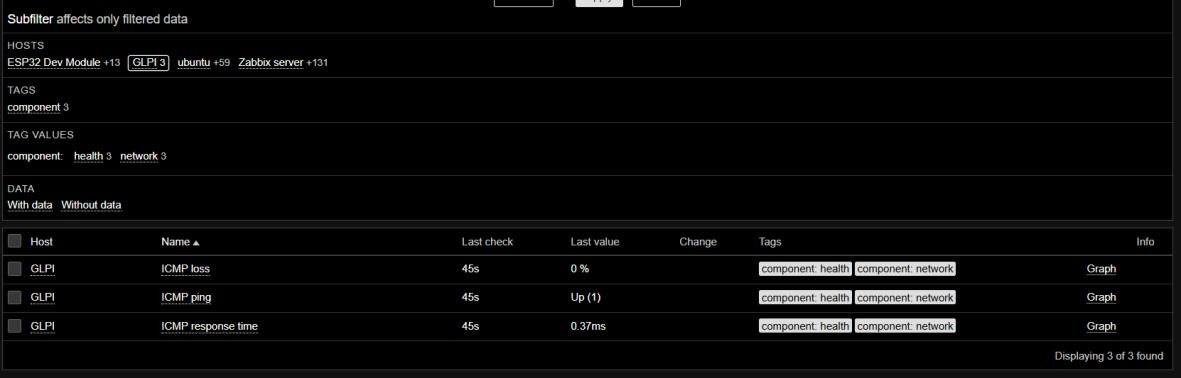
*Figura 37 Instalar paquetes de zabbix*

# Resultados

# Alertas GLPI

El panel muestra las métricas en tiempo real del host **GLPI** monitoreado en Zabbix. Los datos presentados incluyen:

* + **ICMP loss:** Pérdida de paquetes ICMP, con un valor actual de **0%**, indicando que no hay paquetes perdidos.
  + **ICMP ping:** Estado del ping, con un valor de **Up (1)**, confirmando que el host está en línea.
  + **ICMP response time:** Tiempo de respuesta ICMP, actualmente **0.37ms**, lo que indica una buena conectividad.



*Figura 38 Monitoreo de Métricas de GLPI en Zabbix*

Se realiza una comprobación cambiando la IP, además que la alerta ahora tiene un estado de

# down.

La captura muestra un monitoreo reciente del host **GLPI** en Zabbix, con resultados críticos que indican una posible caída del servicio. Las métricas incluyen:

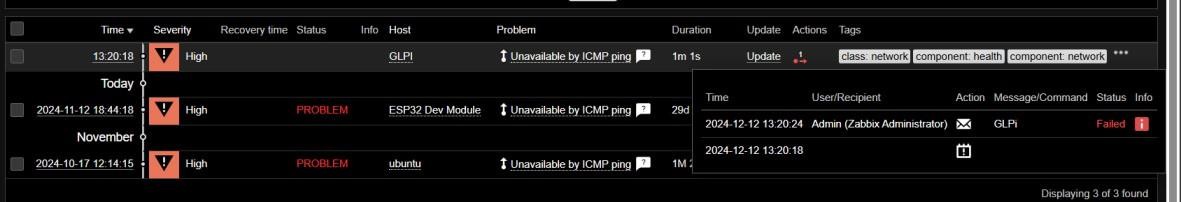
* + **ICMP loss:** Muestra una pérdida de paquetes del **100%**, lo que sugiere una interrupción total en la conectividad.
  + **ICMP ping:** Indica el estado como **Down (0)**, confirmando que el host no responde al ping.
  + **ICMP response time:** El tiempo de respuesta es **0 ms**, lo cual es coherente con la pérdida de conectividad.



*Figura 3S Alerta Crítica en el Monitoreo del Host GLPI en Zabbix*

La captura muestra un registro de problemas detectados por Zabbix con alta severidad. Los detalles relevantes incluyen:

* + **Host:** El problema más reciente afecta al host **GLPI**, con la alerta "Unavailable by ICMP ping".
  + **Severity:** Clasificada como **High**, indicando la criticidad del incidente.
  + **Status:** El estado de la acción configurada muestra un intento de notificación al medio **GLPI**, pero se registra como **Failed**.
  + **Tags:** Asociado con etiquetas como **component: health** y **component: network**, lo que categoriza el problema como relacionado con la red y el estado del sistema.
  + **Duration:** El problema actual tiene una duración de 1 minuto, mientras que los incidentes previos con otros hosts, como "ESP32 Dev Module" y "Ubuntu", también muestran problemas similares.



*Figura 40 Registro de Problemas de Alta Severidad en Zabbix*

En esta captura se observa el registro de un problema sincronizado desde Zabbix hacia GLPI. Los detalles incluyen:

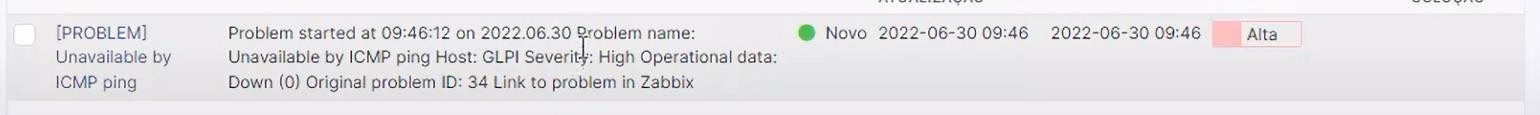
* + **Problem name:** "Unavailable by ICMP ping," indicando que el host **GLPI**

no está disponible debido a una falla en la conectividad ICMP.

* + **Severity:** Clasificado como **Alta**, reflejando la criticidad del problema detectado.
  + **Operational data:** Proporciona información adicional, como el estado **Down**

**(0)** y el enlace al problema original registrado en Zabbix para su trazabilidad.

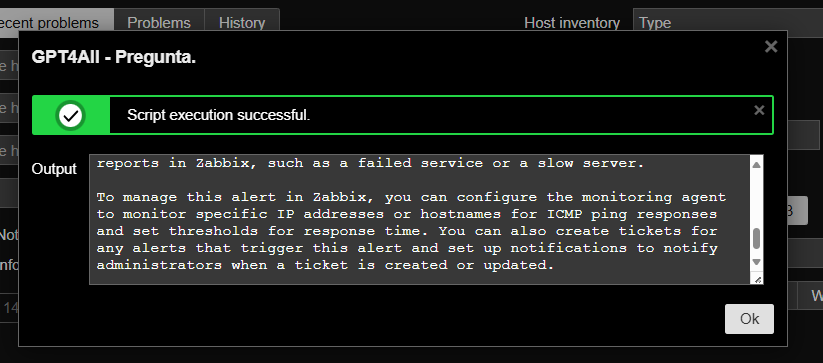
* + **Timestamps:** El problema se detectó el **2022-06-30 a las 09:46**.



*Figura 41 Registro de Problema Detectado en GLPI desde Zabbix*

# GPT4ALL

Se muestra una integración de **GPT4All** en **Zabbix**, donde un script es ejecutado exitosamente, proporcionando una respuesta detallada sobre la gestión de alertas en Zabbix, como fallas de servicios o servidores lentos.



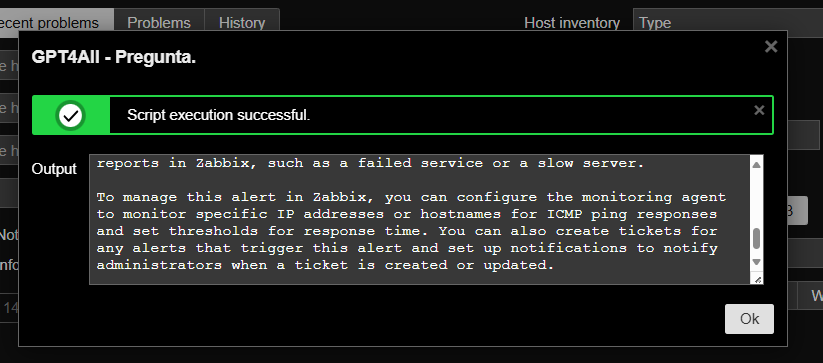
*Figura 42 Interacción entre GPT4ALL y Zabbix*

Se muestra la ejecución exitosa del script "GPT4All - Pregunta" configurado previamente en Zabbix. En la parte superior de la ventana aparece el mensaje "Script execution successful", indicando que el script se procesó sin errores.

El área de salida (Output) despliega el resultado generado por el modelo GPT4All. Este texto incluye un análisis o sugerencias relacionadas con la alerta seleccionada, en este caso, indicando cómo gestionar problemas relacionados con la falta de respuesta ICMP en Zabbix. Se ofrecen recomendaciones como configurar el agente de monitoreo para observar direcciones IP específicas, establecer umbrales de tiempo de respuesta y generar tickets para las alertas asociadas.

La interfaz permite visualizar el resultado detallado del script y cerrar la ventana mediante el botón "Ok" una vez revisada la información proporcionada. Este paso valida la

integración del modelo GPT4All con Zabbix y su funcionalidad para procesar alertas y generar respuestas útiles.



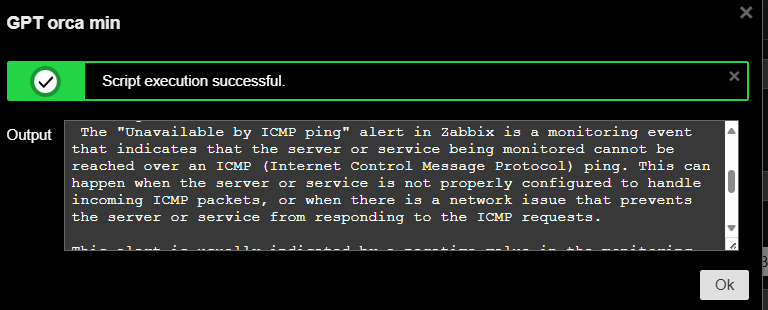
*Figura 15 Resultado de la ejecución del script GPT4All en Zabbix*

# ORCA MINI

Se muestra la ejecución exitosa del script "GPT orca min" configurado previamente en Zabbix. En la parte superior de la ventana aparece el mensaje "Script execution successful", indicando que el script se procesó sin errores.

El área de salida (Output) despliega el resultado generado por el modelo GPT orca min. En este caso, el texto explica el significado de la alerta "Unavailable by ICMP ping", la cual indica que el servidor o servicio monitoreado no puede ser alcanzado a través de un ping ICMP (Internet Control Message Protocol). Se mencionan posibles causas de esta alerta, como una configuración incorrecta que impide el manejo de paquetes ICMP o problemas de red que bloquean la respuesta del servidor o servicio.

La interfaz permite visualizar el resultado detallado del script y cerrar la ventana mediante el botón "Ok" una vez revisada la información proporcionada. Este paso confirma la integración del modelo GPT orca min con Zabbix y su capacidad para analizar eventos de monitoreo y proporcionar respuestas informativas.



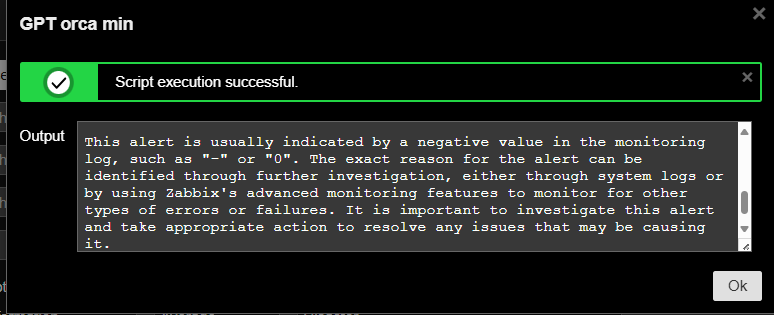
*Figura 43 Resultado de la ejecución del script orcamini en Zabbix*

Se muestra la ejecución exitosa del script "GPT orca min" configurado previamente en Zabbix. En la parte superior de la ventana aparece el mensaje "Script execution successful", indicando que el script se procesó sin errores.

El área de salida (Output) despliega el resultado generado por el modelo GPT orca min. En este caso, el texto proporciona detalles sobre una alerta en el sistema de monitoreo, mencionando que generalmente está indicada por un valor negativo en el registro de monitoreo, como "-" o "0". Se sugiere que la causa exacta de la alerta puede determinarse mediante una investigación más profunda, ya sea a través de los registros del sistema o utilizando las funciones avanzadas de monitoreo de Zabbix para detectar otros tipos de errores o fallas.

El mensaje enfatiza la importancia de investigar la alerta y tomar acciones apropiadas para resolver cualquier problema subyacente que pueda estar causándola.

La interfaz permite visualizar el resultado detallado del script y cerrar la ventana mediante el botón "Ok" una vez revisada la información proporcionada. Este paso confirma la integración del modelo GPT orca min con Zabbix y su capacidad para proporcionar diagnósticos útiles en la gestión de alertas.



*Figura 44 Resultado de la ejecución del script orcamini en Zabbix*

# Gestión de Logs

Se muestra la ejecución del script de integración en el entorno de monitoreo Wazuh. En la terminal, el usuario root ejecuta el comando desde el directorio /var/ossec/integrations/, utilizando el intérprete de Python ubicado en /home/erick/mi\_entorno/bin/python, seguido del script custom-wazuh-zabbix.

El comando se ejecuta en dos ocasiones sin mostrar errores ni mensajes de salida, lo que sugiere que el proceso se llevó a cabo correctamente o que la salida se maneja internamente.

La terminal permanece en espera dentro del mismo directorio, indicando que la ejecución del script ha finalizado sin interrupciones.

Este paso valida la integración de Wazuh con Zabbix y permite continuar con la verificación en los registros de eventos para confirmar su correcto funcionamiento.



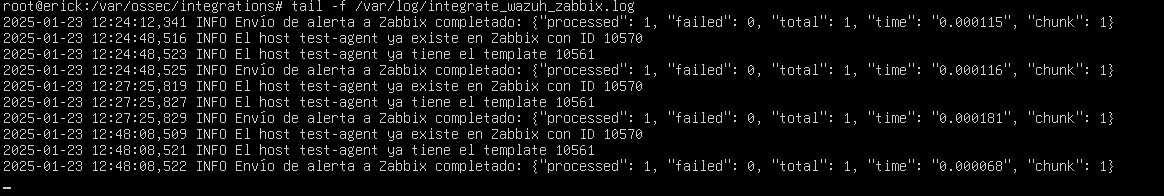
*Figura 45 Integración de Wazuh con Zabbix*

Se muestra la verificación de los logs en el servidor Wazuh tras la ejecución del script de integración con Zabbix. En la terminal, el usuario root ejecuta el comando tail -f

/var/log/integrate\_wazuh\_zabbix.log, lo que permite monitorear en tiempo real las entradas generadas en el archivo de log correspondiente.

El área de salida muestra múltiples registros con marca de tiempo, indicando el procesamiento exitoso de alertas enviadas a Zabbix. Cada entrada confirma que las alertas han sido procesadas correctamente, sin fallos, con un tiempo de ejecución mínimo. Además, se evidencia que el host "test-agent" ya existe en Zabbix, asociado a los identificadores 10570 y 10651, lo que sugiere que la integración está funcionando correctamente y asignando los templates adecuados.

La terminal permanece activa en modo de monitoreo, permitiendo observar en tiempo real cualquier nuevo evento registrado en los logs. Este paso valida la comunicación efectiva entre Wazuh y Zabbix, asegurando que las alertas se están enviando y gestionando sin errores.



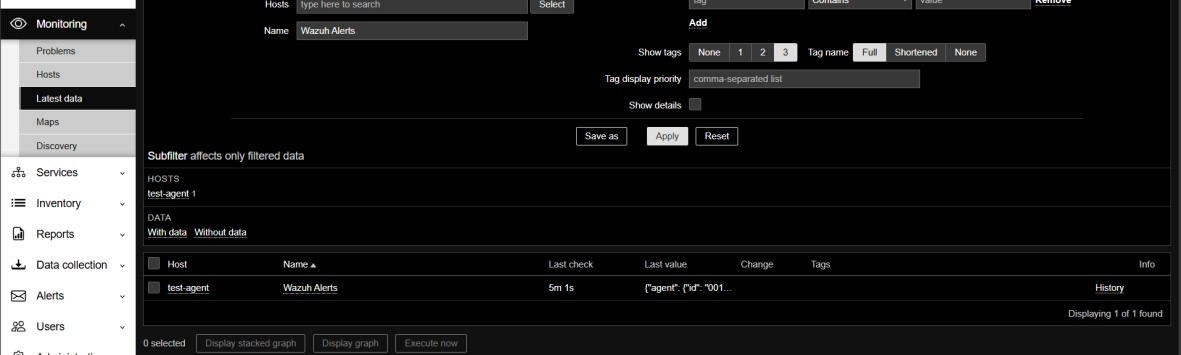
*Figura 46 Monitoreo activo*

Se muestra la verificación en Zabbix, confirmando que el mensaje ha sido procesado con éxito y ahora es visible en la sección "Latest data". En la interfaz de Zabbix, dentro del

módulo de Monitoreo, se ha seleccionado el host "test-agent", el cual está asociado a la categoría "Wazuh Alerts".

El área de datos muestra que el último registro de monitoreo fue realizado hace 5 minutos y 1 segundo, indicando que Zabbix está recibiendo correctamente la información enviada desde Wazuh. En la columna de "Last value", se observa un dato en formato JSON que sugiere que la alerta ha sido almacenada y procesada exitosamente.

En la parte inferior de la interfaz, se presentan opciones para visualizar gráficos o ejecutar acciones adicionales sobre los datos recopilados. Esto confirma que la integración entre Wazuh y Zabbix está funcionando correctamente y que los eventos de seguridad están siendo monitoreados en tiempo real.



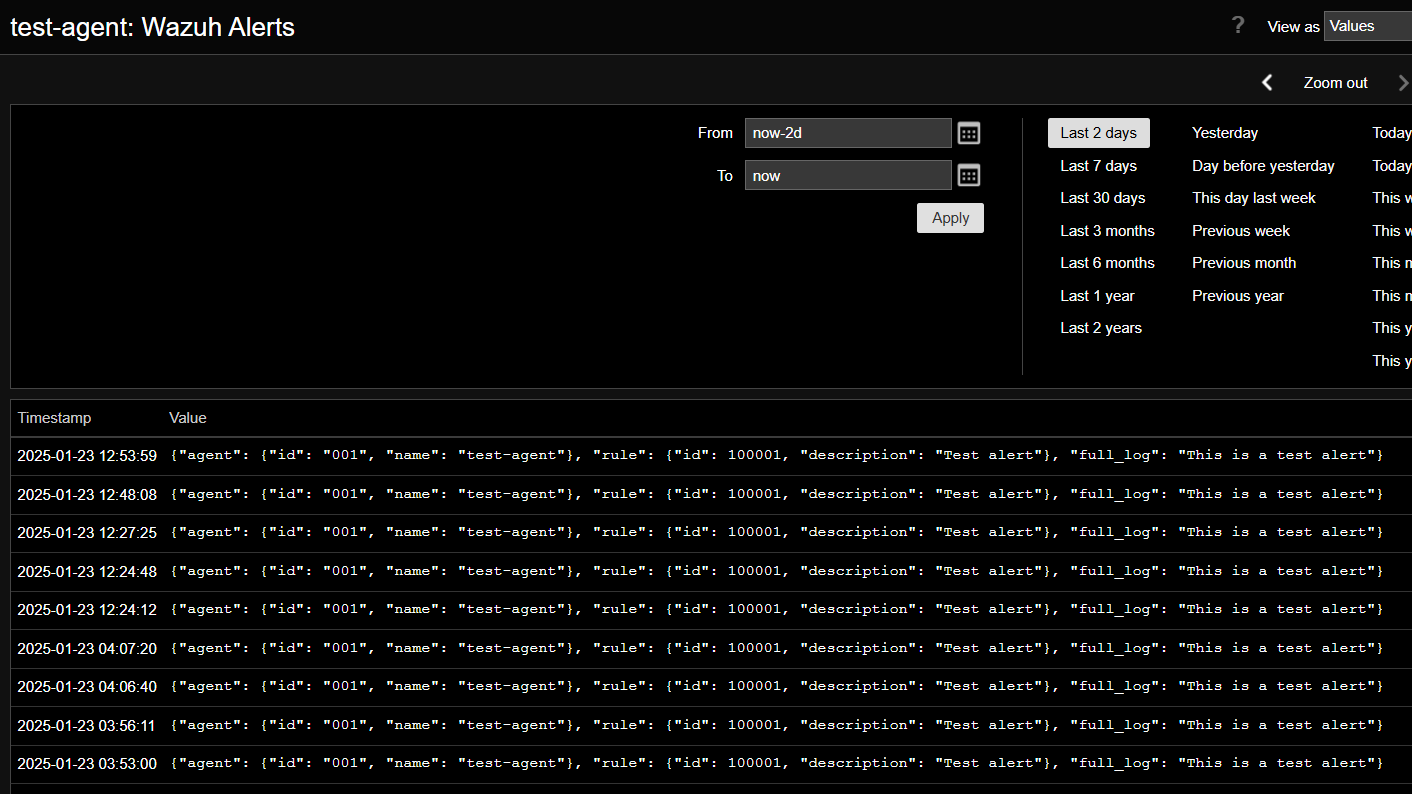
*Figura 47 Mensaje procesado exitosamente en zabbix*

Se muestra la visualización de los logs en Zabbix dentro de la sección "Wazuh Alerts", correspondiente al host "test-agent". En la interfaz, se observa un panel de filtros que permite seleccionar un rango de tiempo específico para la consulta, con opciones como los últimos 2 días, 7 días, 30 días, entre otros.

En la tabla de datos, se presentan múltiples registros con marca de tiempo y valores en formato JSON, lo que confirma que las alertas han sido procesadas y almacenadas correctamente en Zabbix. Cada entrada contiene información relevante sobre la alerta, incluyendo el ID del agente ("id": "001"), el nombre del host ("name": "test-agent"), y el ID de la regla ("id": 100001"), la cual está vinculada a la detección de eventos específicos.

Además, se muestra la descripción de la alerta ("description": "Test alert") y el registro completo del evento ("full\_log": "This is a test alert"), indicando que se trata de una alerta de prueba enviada desde Wazuh.

Los registros aparecen con diferentes marcas de tiempo, lo que sugiere que las alertas están siendo generadas y enviadas de manera continua. Esto confirma que la integración entre Wazuh y Zabbix está funcionando correctamente y que los eventos de seguridad están siendo monitoreados en tiempo real.



*Figura 48 Registros de logs*

# Conclusiones

* La incorporación de un chatbot basado en GPT4All facilita la interacción con Zabbix, reduciendo la necesidad de conocimientos técnicos para mejorar la eficiencia en la toma de decisiones operativas.
* El chatbot implementado dentro de Zabbix está diseñado para interactuar con los usuarios utilizando lenguaje natural, permitiendo realizar consultas sobre el estado de la infraestructura monitoreada, gestionar alertas, y proporcionar respuestas inteligentes basadas en la información que Zabbix recopila.
* GPT4All es ideal para aplicaciones que requieren análisis profundo, precisión en el contexto y generación de texto avanzado, pero a costa de un mayor consumo de recursos y hardware robusto. Por otro lado, Orca Mini, basado en LLaMA, es más eficiente y adecuado para entornos con limitaciones de hardware, sacrificando algo de precisión y profundidad a cambio de velocidad y menor costo.
* Para responder tareas complejas o trabajar con múltiples variables en profundidad como generación detallada de informes o análisis complejos en sistemas como Zabbix. GPT4All es la mejor opción. Sin embargo, si se priorizan rapidez, eficiencia y un uso más ligero del hardware para automatizar tareas básicas o responder alertas, Orca Mini ofrece una solución más práctica y económica.
* La implementación de GPT-J no fue posible debido a sus altos requerimientos de recursos, tanto a nivel de hardware como de infraestructura, como CPU/GPU, memoria RAM y almacenamiento para funcionar de manera eficiente, que exceden las capacidades del sistema actual.
* Con Wazuh en Zabbix amplía significativamente las capacidades de monitoreo de seguridad al proporcionar herramientas visuales para la gestión de alertas y correlación de eventos en tiempo real. Esto mejora al centralizar las alertas críticas en un solo panel de control, como es el dashboard de Zabbix que presenta eventos de Wazuh

# Recomendaciones

* Configurar una dirección IP estática en la máquina física para garantizar una conexión remota estable y confiable con la máquina virtual que aloja el servidor de Zabbix. Esto evita conflictos de red y facilita el acceso continuo a los recursos del servidor sin interrupciones.
* Usar scripts personalizados, como el custom-wazuh-zabbix, para conectar Wazuh con herramientas externas. Complementar este flujo con API o frameworks de IA generativa, asegurando que ambos sistemas puedan comunicarse mediante un formato común como JSON.
* Asegurarse de que Wazuh esté correctamente configurado con todos los componentes funcionales, como agentes, reglas y configuraciones de integración, incluyendo la API, token, URL, puerto, grupo y template asociados.
* Proteger las comunicaciones entre Wazuh, Zabbix y la IA generativa, además, garantizar que el sistema sea escalable y capaz de manejar volúmenes crecientes de datos.
* Asegurarse de que las alertas generadas por Wazuh se transformen a un formato estándar compatible con los modelos de IA generativa, como JSON, XML u otro formato legible para facilitar la interoperabilidad entre sistemas.
* Se puede configurar tanto Wazuh como la IA generativa para que clasifiquen las alertas según niveles de riesgo o impacto, permitiendo un enfoque más efectivo en la mitigación de problemas críticos

# Referencias

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

OpenAI. (2023). GPT-4 technical report. OpenAI. Disponible en https://openai.com/research/gpt-4

Hugging Face. (2023). Transformers library documentation. Hugging Face.

Disponible en https://huggingface.co/docs/transformers/index

Zabbix Documentation Team. (2023). Zabbix 6.0 documentation. Disponible en https://[www.zabbix.com/documentation/current/](http://www.zabbix.com/documentation/current/)

Olups, R. (2016). Zabbix network monitoring (2nd ed.). Packt Publishing.

Wazuh Documentation Team. (2023). Wazuh 4.3 documentation. Disponible en https://documentation.wazuh.com/

Gaikwad, S. (2020). Mastering Wazuh - A practical guide to threat detection and response.

MITRE ATT&CK Framework. (2023). Técnicas de detección de amenazas en ciberseguridad. Disponible en https://attack.mitre.org/