

ZABBIX '25

CONFERENCE

LATIN AMERICA

Mejores practicas, configuración y seguridad en **Zabbix**

Facundo Vilarnovo

Trainer

Don't you have any NEW stories?



Mejores Prácticas en **Zabbix**



Principio de Menor Privilegio

(Usuarios y Agentes) [PoLP]

¿Porque es crucial?

- ▶ Minimiza el daño potencial en caso de compromiso
- ▶ Reduce la superficie de ataque
- ▶ Mejora la auditoria y cumplimiento (compliance)



Principio de Menor Privilegio (Usuarios y Agentes) [PoLP]

Aplicación en Zabbix

- Usuarios:
 - Roles de usuario (Admin, Super Admin, User)

☐ Name ▲	#	Users
☐ Admin role		Users
☐ Guest role		Users 1
☐ Super admin role		Users 2
☐ User role		Users

Access to API

Enabled ☒

API methods

Access to actions

- ☒ Create and edit dashboards
- ☒ Create and edit maps
- ☒ Create and edit maintenance
- ☒ Add problem comments
- ☒ Change severity
- ☒ Acknowledge problems
- ☒ Suppress problems
- ☒ Close problems
- ☒ Execute scripts
- ☒ Manage API tokens
- ☒ Manage scheduled reports
- ☒ Manage SLA
- ☒ Invoke "Execute now" on read-only hosts
- ☒ Change problem ranking

Default access to new actions ☒

User roles

* Name

User type

Access to UI elements

Dashboards ☐

Monitoring ☐ Problems ☐ Latest data ☐ Discovery

☐ Hosts ☐ Maps

Services ☒ Services ☐ SLA ☐ SLA report

Inventory ☒ Overview ☐ Hosts

Reports ☐ System information ☒ Top 100 triggers ☒ Notifications

☒ Scheduled reports ☐ Audit log

☒ Availability report ☐ Action log

Data collection ☒ Template groups ☒ Hosts ☒ Discovery

☒ Host groups ☒ Maintenance

☒ Templates ☐ Event correlation

Alerts ☒ Trigger actions ☒ Autoregistration actions ☐ Scripts

☒ Service actions ☒ Internal actions

☒ Discovery actions ☐ Media types

Users ☐ User groups ☐ Users ☐ Authentication

☐ User roles ☐ API tokens

Administration ☐ General ☐ Proxy groups ☐ Queue

☐ Audit log ☐ Proxies

☐ Housekeeping ☐ Macros

* At least one UI element must be checked.

It access to new UI elements ☒

Principio de Menor Privilegio

(Usuarios y Agentes) [PoLP]

Aplicación en Zabbix

- Permisos granulares por grupos de host y templates

User groups

☐	Name ▲		
☐	Disabled	Users 1	guest
☐	Enabled debug mode	Users	
☐	Guests	Users 1	guest
☐	Internal	Users 2	Admin (Zab
☐	No access to the frontend	Users	
☐	Training users	Users 1	student-03 (
☐	Zabbix administrators	Users 1	Admin (Zab

User groupTemplate permissionsHost permissions ●Problem tag filter

Permissions

Host groups

Applications × Databases × demo ×
type here to search

Linux servers ×
type here to search

Zabbix servers ×
type here to search

Add

Permissions

SelectRead-writeReadDenyRemove

SelectRead-writeReadDenyRemove

SelectRead-writeReadDenyRemove

Principio de Menor Privilegio (Usuarios y Agentes) [PoLP]

Aplicación en Zabbix

► Agente Zabbix:

- Ejecutar el agente con usuario no privilegiado (ej zabbix)
- Limitar el acceso del agente solo a recursos que necesitamos monitorear
- Uso de **UserParameter** con precaución y validación de entrada

```
### Option: AllowRoot
#   Allow the agent to run as 'root'. If disabled, the agent
#   will try to switch to the user specified in the User option.
#   Has no effect if started under a regular user.
#   0 - do not allow
#   1 - allow
#
# Mandatory: no
# Default:
# AllowRoot=0

### Option: User
#   Drop privileges to a specific, existing user on the system.
#   Only has effect if run as 'root' and AllowRoot is disabled.
#
# Mandatory: no
# Default:
# User=zabbix
# NOTE: This option is overridden by settings in systemd service file!
```

zabbix	450260	1	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
zabbix	450261	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: collector [idle 1 sec]
zabbix	450262	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix	450263	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix	450264	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix	450265	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #4 [waiting for connection]
zabbix	450266	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #5 [waiting for connection]
zabbix	450267	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #6 [waiting for connection]
zabbix	450268	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #7 [waiting for connection]
zabbix	450269	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #8 [waiting for connection]
zabbix	450270	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #9 [waiting for connection]
zabbix	450271	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: listener #10 [waiting for connection]
zabbix	450272	450260	0	17:44	?	00:00:00	/usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

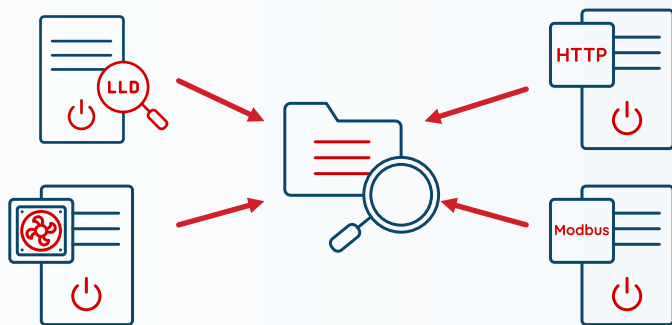
Plantillas Eficientes - Principios Clave

(Templates)

¿Que necesita un buen template?

► Flexible y reutilizable

- Usar LLD tanto como sea posible. (previene items y triggers no soportados)
- Utilizar macros de usuario. En triggers e items. Ej. `{ $TEMP.CRIT.MAX }`
- Evitar las dependencias externas. Siempre utilizar preprocesos antes que scripts externos



Plantillas Eficientes - Principios Clave

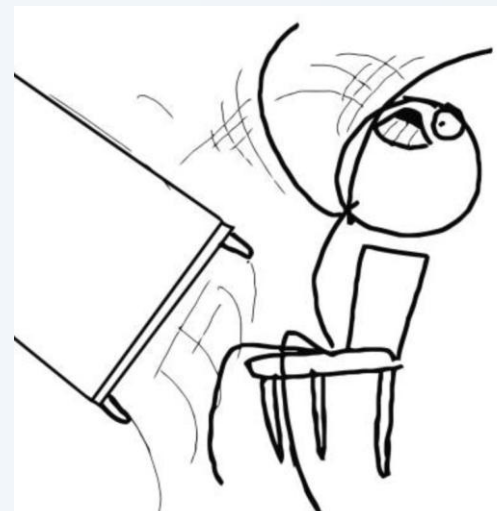
(Templates)

¿Que necesita un buen template?

► Conocimiento y Experiencia (KISS)

- **Evitar demasiadas métricas.** Prevenir el abuso de recolección de datos mejora la performance y mantiene el foco en el problema de raíz.
- Triggers innecesarios **causan fatiga de alertamiento** y desconfianza en la herramienta.

No crear triggers informacionales o anomalías para todo.



Plantillas Eficientes - Principios Clave

(Templates)

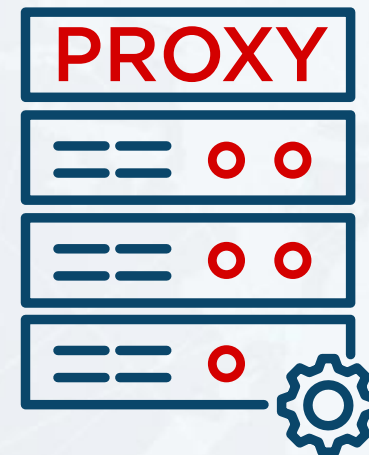
¿Que necesita un buen template?

- ▶ Modularidad y alcance
 - El template adecuado es aquel que tiene un alcance a nivel recurso:
 - ✓ Un servidor específico de un vendor
 - ✓ Un sistema operativo
 - ✓ Sistemas de orquestación
 - ✓ Aplicaciones personalizadas
- ▶ Si mantiene el alcance de una plantilla dentro de un único recurso, será mucho **más fácil compartirlas** y serán útiles para quienes comparten los mismos componentes en su arquitectura.
- ▶ **Evite fusionar recursos de diferentes capas**; no añada métricas para Linux ni PostgreSQL en una única plantilla.

Optimización de Proxies Zabbix

¿Por que usar proxies? Características

- ▶ Monitoreo distribuido
 - Ubicaciones remotas
 - Segmentación de red
 - Ancho de banda reducido
 - Escalamiento horizontal
 - Encriptación nativa
- ▶ Sentido del trafico
 - Activo (Recomendado): El proxy inicia la conexión hacia el server (ProxyMode=0)
 - Pasivo: El server origina la conexión hacia el proxy (ProxyMode=1)



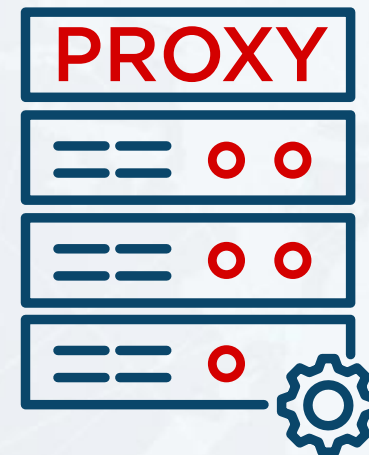
Optimización de Proxies Zabbix

¿Por que usar proxies? **Configuración simplificada**

- Configuración esencial (zabbix_proxy.conf)
 - Server: IP/DNS del Zabbix server
 - Hostname: Nombre exacto del proxy (como fue configurado en el Frontend)
 - ConfigFrequency: Actualizaciones de configuración 60-3600.
 - DataSenderFrequency: Frecuencia de envío de datos al server

☐ Name ▲	Mode	Encryption
☒ Proxy A	Active	None
☐ Proxy B	Active	None
☐ Proxy C	Active	None

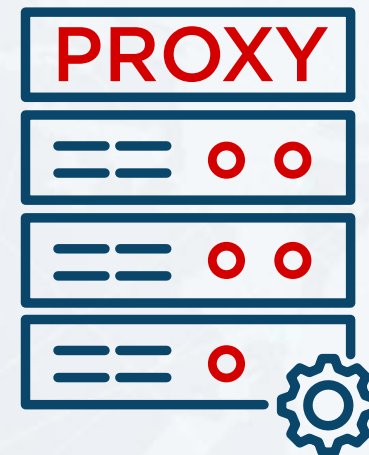
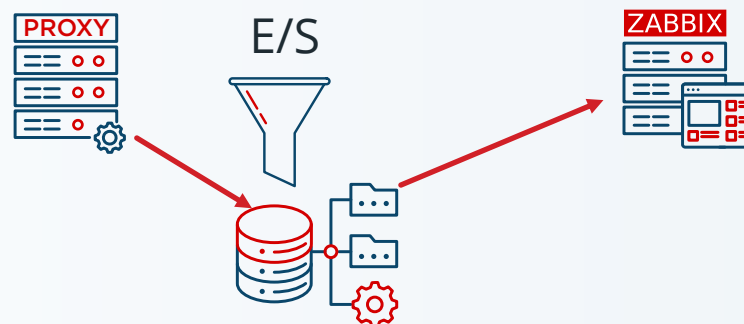
1 selected Refresh configuration Enable hosts Disable hosts



Optimización de Proxies Zabbix

Recursos del proxy

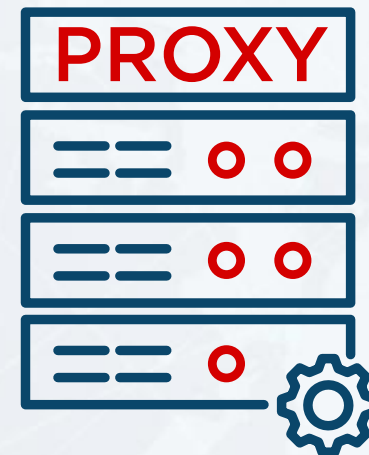
- ▶ Monitorear el proxy mediante **zabbix_proxy_health**
 - Esto nos permite tener una evolución histórica y evaluar la demanda
 - Ajustar los pollers a un **~60% de utilización**
- ▶ Elegir correctamente la DB en función del crecimiento planificado
 - SQLite (por defecto) escala hasta **~1000 NVPS**
 - Utilizar Mysql/PostgreSQL si superamos esta métrica
- ▶ Seleccionar el modo correcto de operación para ProxyBufferMode
 - **Disk (legacy)**



Optimización de Proxies Zabbix

Recursos del proxy

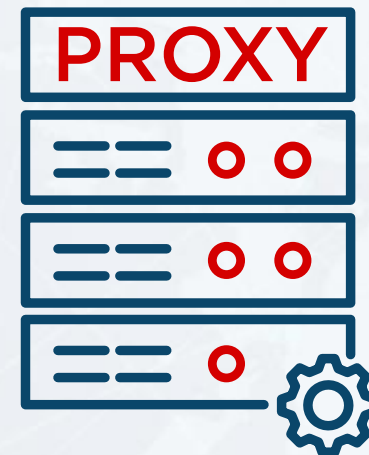
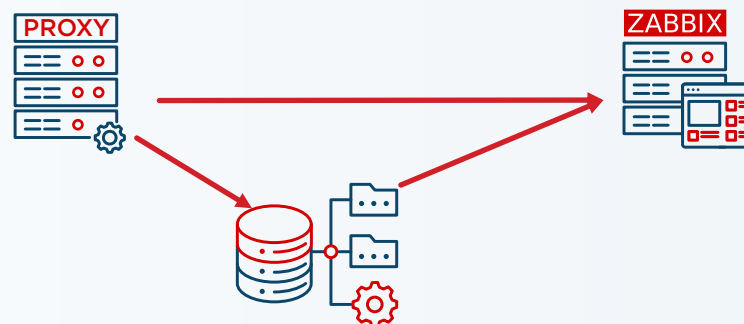
- ▶ Monitorear el proxy mediante **zabbix_proxy_health**
 - Esto nos permite tener una evolución histórica y evaluar la demanda
 - Ajustar los pollers a un **~60% de utilización**
- ▶ Elegir correctamente la DB en función del crecimiento planificado
 - SQLite (por defecto) escala hasta **~1000 NVPS**
 - Utilizar Mysql/PostgreSQL si superamos esta métrica
- ▶ Seleccionar el modo correcto de operación para ProxyBufferMode
 - Disk (legacy)
 - **Memory**



Optimización de Proxies Zabbix

Recursos del proxy

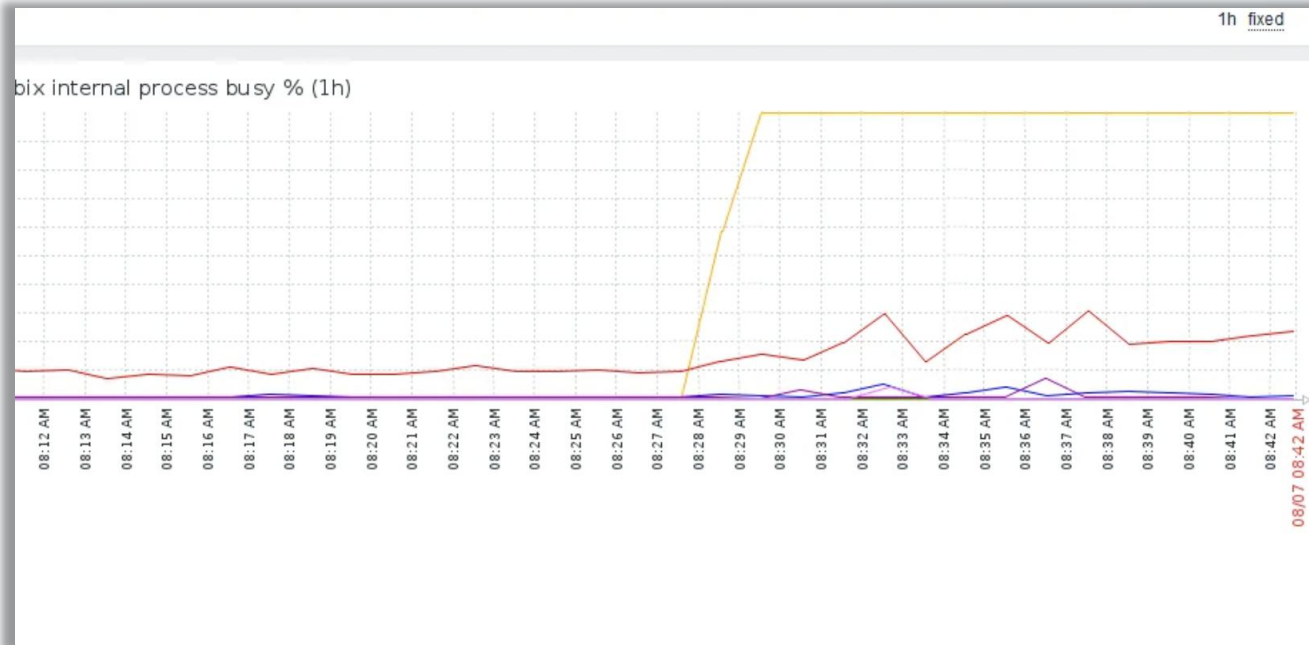
- ▶ Monitorear el proxy mediante **zabbix_proxy_health**
 - Esto nos permite tener una evolución histórica y evaluar la demanda
 - Ajustar los pollers a un **~60% de utilización**
- ▶ Elegir correctamente la DB en función del crecimiento planificado
 - SQLite (por defecto) escala hasta **~1000 NVPS**
 - Utilizar Mysql/PostgreSQL si superamos esta métrica
- ▶ Seleccionar el modo correcto de operación para ProxyBufferMode
 - Disk (legacy)
 - Memory
 - **Hybrid** (Recomendado)



Mantenimiento de la Base de Datos

¿Por que es vital?

- Una DB saludable asegura un Zabbix rápido, confiable y con datos accesibles.



Housekeeping

Events and alerts

Enable internal housekeeping ☒

* Trigger data storage period 365d

* Service data storage period 1d

* Internal data storage period 1d

* Network discovery data storage period 1d

* Autoregistration data storage period 1d

Services

Enable internal housekeeping ☒

* Data storage period 365d

User sessions

Enable internal housekeeping ☒

* Data storage period 365d

History

Housekeeping ☒

History period ☐

Storage period 31d

Trends

Housekeeping ☒

Trend period ☐

Storage period 365d

Audit log

[Audit settings](#)



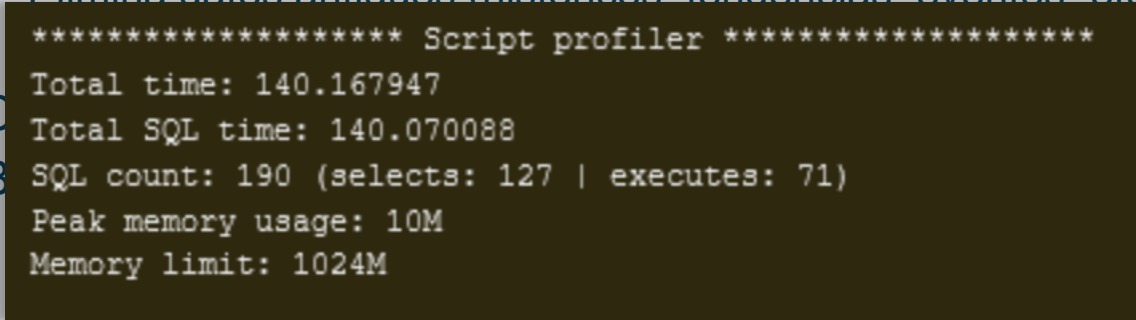
Mantenimiento de la Base de Datos

¿Por que es vital?

- Una DB saludable asegura un Zabbix rápido, confiable y con datos accesibles.

- Housekeeper: el “Limpiador” de Zabbix

Elimina datos antiguos (históricos, tendencias, eventos, etc)

- 

```
***** Script profiler *****
Total time: 140.167947
Total SQL time: 140.070088
SQL count: 190 (selects: 127 | executes: 71)
Peak memory usage: 10M
Memory limit: 1024M
```

Housekeeping

Events and alerts

Enable internal housekeeping ☒

* Trigger data storage period 365d

* Service data storage period 1d

* Internal data storage period 1d

* Network discovery data storage period 1d

* Autoregistration data storage period 1d

Services

Enable internal housekeeping ☒

* Data storage period 365d

User sessions

Enable internal housekeeping ☒

* Data storage period 365d

History

Housekeeping ☒

History period ☐

Storage period 31d

Trends

Housekeeping ☒

Trend period ☐

Storage period 365d

Audit log

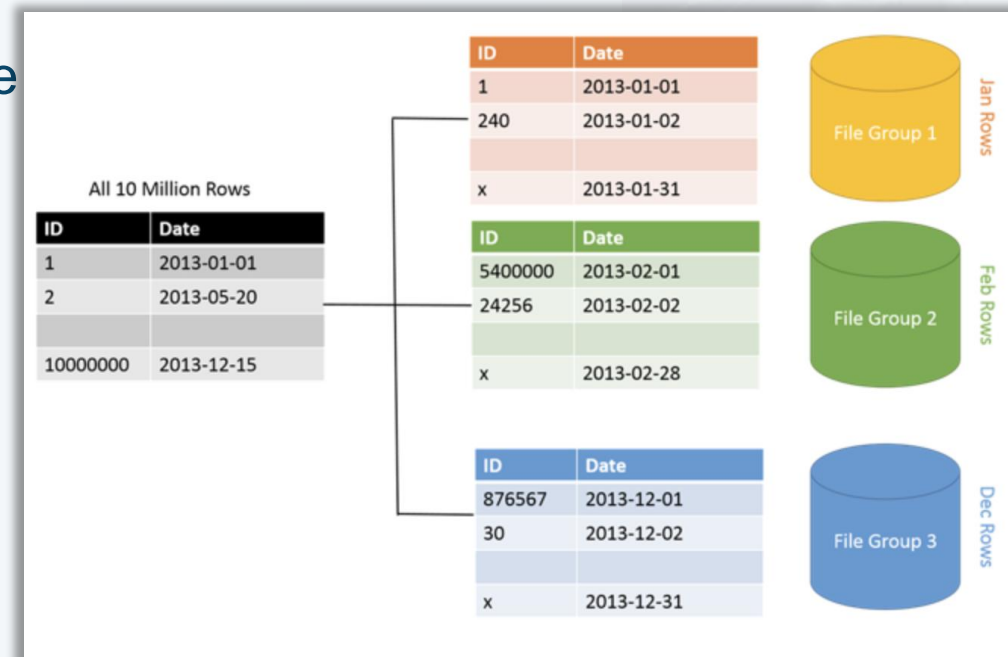
[Audit settings](#)



Mantenimiento de la Base de Datos

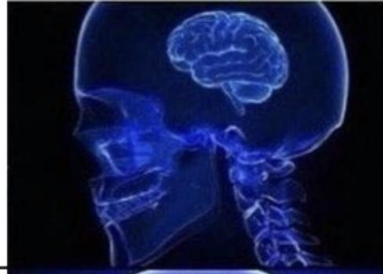
Particionamiento de tablas

- ▶ Recomendado para entornos medianos/grandes
 - Divide las tablas gigantes (history*, trends*, events) en particiones
 - Mejora drásticamente el rendimiento de escritura y especialmente la eliminación de datos.
- ▶ PostgreSQL realiza esto via la extensión de Timescale
- ▶ Mysql/MariaDB requieren scripts adicionales

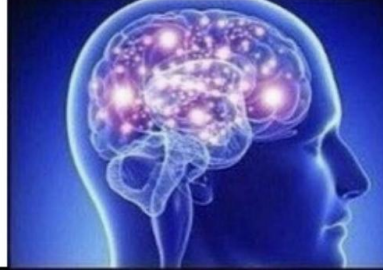


Configuración Esencial y Optimizaciones

**INSTALAR
Y USAR**



**LEER
EL MANUAL**



**APLICAR
CONFIGURACIÓN
ESENCIAL.**



**OPTIMIZAR
PARA MÁXIMO
RENDIMIENTO.**



ZABBIX '25

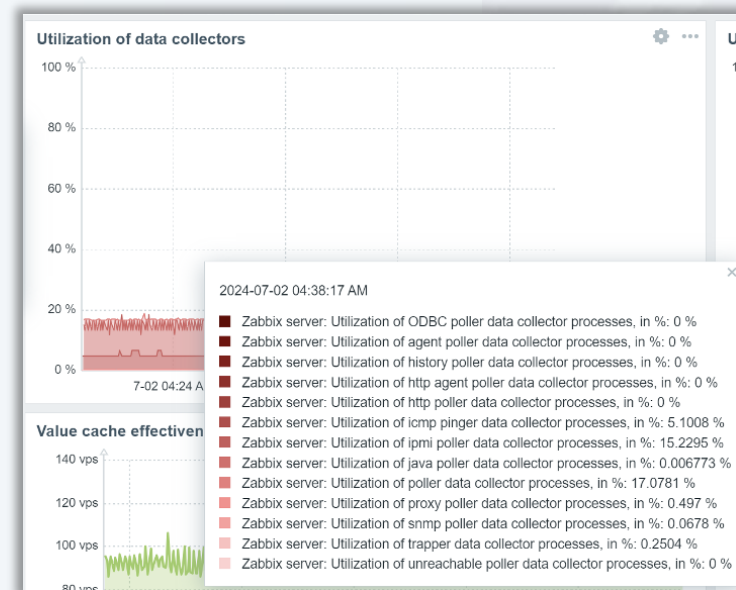
CONFERENCE

LATIN AMERICA

Configuración Inicial de Zabbix

Ajustes Clave en Server/Proxy

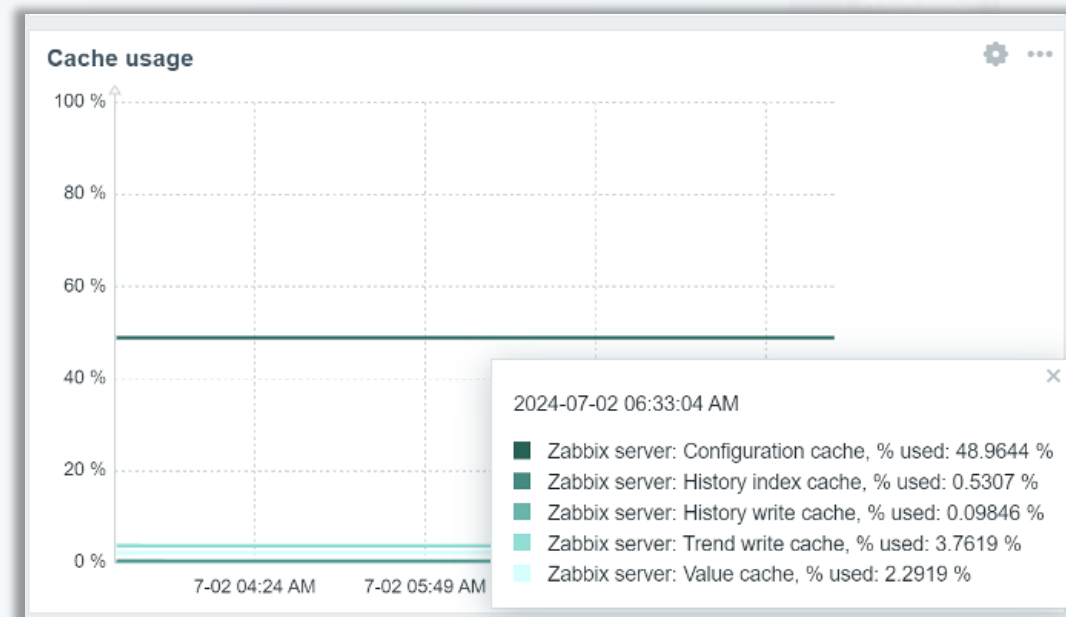
- ▶ Recolectores Asíncronos, todos comparten **MaxConcurrentChecksPerPoller=1000**
 - Cada proceso puede recolectar hasta 1000 métricas en simultaneo
 - Parámetros de kernel deben ser ajustados para permitir cantidad de FileDescriptors
- ▶ Recolectores Sincrónicos
 - Pollers (simplechecks, ssh, external)
 - Trappers (chequeos activos, proxies, etc)
 - UnreachablePollers (metricas que no responden)



Configuración Inicial de Zabbix

Ajustes Clave en zabbix_server.conf

- ▶ Procesos internos
 - Preprocessing workers
 - History Pollers
 - LLD Workers
- ▶ Caches
 - Cachesize (configuration total de monitoreo)
 - HistoryCache (datos antes de ser guardados)
 - TrendCacheSize (datos de tendencia)
- ▶ Los caches deben estar en un ~60% libre

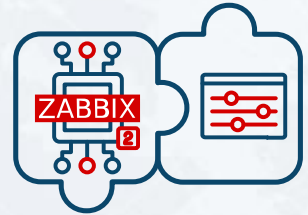


Configuración del Agente Zabbix

(Zabbix Agent/Agent2)

Agente Clásico (Zabbix Agent) vs. Zabbix Agent 2

- ▶ Agent 2 (Recomendado para nuevas implementaciones)
 - Soporte para plugins (Docker, Modbus, MQTT, etc)
 - Chequeos concurrentes (mejor rendimiento)
 - Buffering mejorado
- ▶ Parámetros esenciales (Comunes a ambos)
 - Server: ACL donde permitimos conexiones entrantes
 - ServerActive: chequeos activos (Recomendado para la mayoría de items)
 - Hostname: Debe coincidir con el host configurado en el frontend
 - UserParameter: Extender funcionalidad simplificada
- ▶ Seguridad y control
 - EnableRemoteCommands, deshabilitado por defecto
 - AllowKey/DenyKey, limitar métricas



Listas de Permisos/Denegación en Agentes

(Allow/Deny Keys)

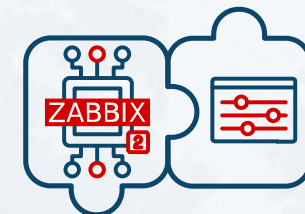
Restringir que ítems keys puede consultar Zabbix Server/Proxy al agente.

► AllowKey=<patrón de key>

- Solo permite keys que hagan coincidencia (puede usar wildcards “*”)
- Es posible definir varias en combinación con DenyKey
- Son procesadas en el orden que fueron definidas
- Todos los system.run[*] son denegados por defecto de forma implícita
- Solo es posible definir un AllowKey con un DenyKey acorde.
- En cuanto se produce coincidencia, la comparación de reglas no continua

► Ejemplo DenyKey

- DenyKey=vfs.file.contents[/etc/password,*]



Monitoreo de Archivos de Log

Item key de agente, mejor **.count**?

- ▶ Log.*[archivo, patrón,...,....]
 - Usado para archivos que no rotan. Es esencial el uso de patrones para limitar datos
- ▶ Logrt.*[archivo, patrón,...,....]
 - Para archivos con rotación. (Mas intensivo en recursos de agente)
- ▶ Buenas practicas
 - Los patrones buscan coincidencias dentro del archivo y solo envían estos al server
 - El parámetro **maxlines** sirve para limitar el envío por segundo
 - Output nos sirve para dar formato al texto de la coincidencia.
 - Es mejor utilizar *.count para enviar valores numéricos que un string
- ▶ Es importante destacar que Zabbix no reemplaza un concentrador de logs

Autoregistro de Agentes

Como funciona?

- ▶ Un agente active contacta al Zabbix server y pide configuracion
 - Utiliza el valor de ServerActive
 - Envia el valor de Hostname y opcionalmente HostMetadata
- ▶ El server ejecuta una accion de autoregistro
 - Evalua las condiciones con la informacion recibida
 - Ejecuta las operaciones (Agregar a grupo, vincular Plantilla,etc)
- ▶ Es recomendable utilizar hostmetada como valor adicional para mejor toma de decision del lado del server

Macros: Globales, de Plantilla y de Host

Flexibilidad y Reutilización

- ▶ Son variables simbólicas que resuelven un valor en tiempo y contexto
 - Zabbix ofrece 4 tipos: Incluidas, de descubrimiento, de usuario y de expresión
- ▶ De usuario
 - Globales: tienen menor prioridad y pueden ser referenciadas en todos los hosts
 - De template: definidas en la Plantilla y aplican a todos los hosts vinculados
 - De host: Mayor prioridad, sobrescriben valores de plantillas
 - Secretas
 - Son utilizadas para información sensible
 - Pueden ser guardadas en la DB u obtenidas vía bóveda
- ▶ Las macros permiten mayor flexibilidad y granularidad en configuraciones

{MACRO}

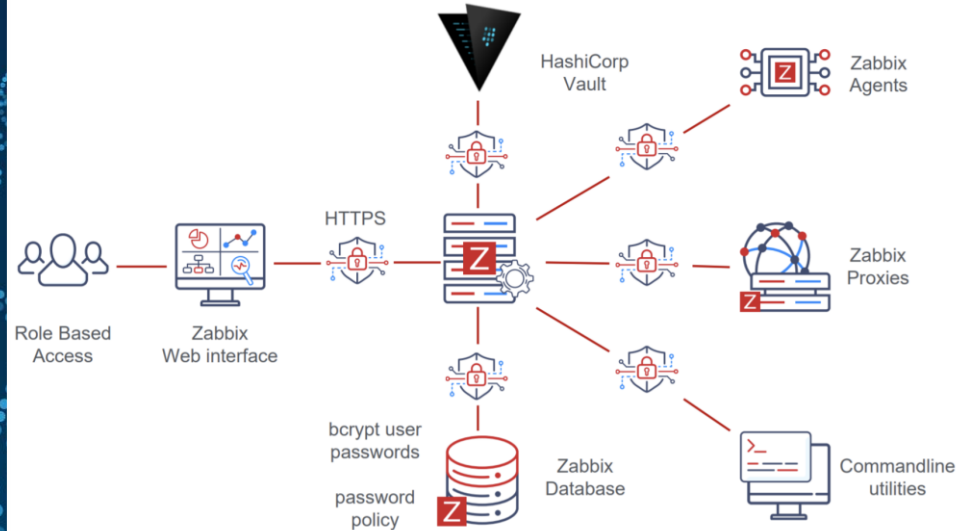
{\${MACRO}}

{#MACRO}

{?MACRO}

Seguridad en Zabbix

ZABBIX '25
CONFERENCE
LATIN AMERICA

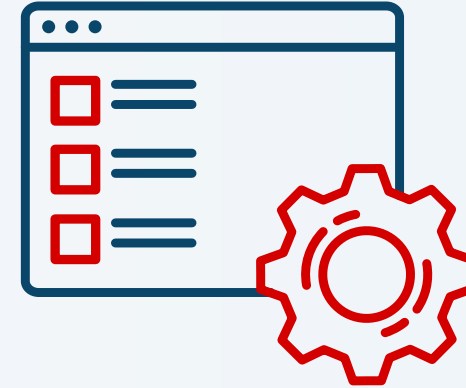


Asegurar la Interfaz Web

(Frontend)

Medidas imprescindibles

- ▶ HTTPS
 - Siempre cifrar la comunicación entre cliente-servidor
 - Utilizar certificados validos (Let's encrypt, comerciales)
- ▶ Hardening del servidor web (apache/nginx)
 - Mantenerlo actualizado
 - Deshabilitar "server signature"
 - Configurar cabeceras de seguridad (X-Content-Type_Options, X-Frame-Options)
 - Limitar el acceso a la URL
- ▶ Aplicar contraseñas robustas y M2FA
- ▶ Archivo zabbix.conf.php
 - Asegurar los permisos de forma restrictiva. Este archivo contiene credenciales o tokens de acceso.



Seguridad en la Comunicación

Componentes Zabbix

Métodos de Cifrado en Zabbix

- ▶ **Claves Precompartidas** (Pre-Shared Keys - PSK)
 - Se genera una clave secreta (el PSK) y una identidad para esa clave
 - Ambas partes de la comunicación (ej. Server y Agente) deben tener configurada la misma identidad y PSK
 - Parámetros clave en archivos .conf (server, proxy, agente):
 - TLSConnect: Qué usar para conexiones salientes (ej. psk).
 - TLSAccept: Qué aceptar para conexiones entrantes (ej. psk).
 - TLSPSKFile: Ruta al archivo con identidades y PSKs.
 - TLSPSKIdentity: La identidad PSK a usar.
- ▶ **Certificados** (PKI - Infraestructura de Clave Pública)
 - Más robusto, pero más complejo de implementar y gestionar (requiere una CA).
 - Permite una autenticación más fuerte de ambas partes.
 - Parámetros clave: **TLSCertFile**, **TLSKeyFile**, **TLSCAFile**

Seguridad en la Comunicación

Componentes Zabbix

Métodos de Cifrado en Zabbix

► Recomendaciones Clave:

- **Cifrar todas las comunicaciones entre componentes Zabbix.**
- **PSK es un buen punto de partida:** Ofrece un buen balance entre seguridad y facilidad de uso.
- **Certificados para máxima seguridad:** Si ya se cuenta con una PKI o se requiere un nivel de aseguramiento superior.
- Utilizar PSKs fuertes y proteger los archivos de PSK.
- Configurar TLSConnect y TLSAccept para hacer el cifrado mandatorio.

Seguridad de la Base de Datos

¡La Base de Datos almacena TODO! Su seguridad es **primordial**.

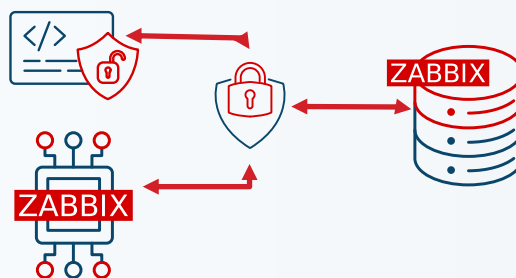
► Medidas de Seguridad Clave

- Principio de Menor Privilegio para el Usuario de Zabbix:

El usuario que Zabbix Server/Frontend usa para conectarse a la BD (MySQL, PostgreSQL) debe tener **SOLO los permisos estrictamente necesarios** (SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, etc., sobre la base de datos de Zabbix, no más)

- Segmentar los usuarios por componente. (frontend/server)
- Consultar la documentación oficial de Zabbix para los GRANT exactos
- Cifrar Conexiones a la Base de Datos (SSL/TLS)

Protege los datos en tránsito, incluyendo credenciales y datos de monitoreo



Macros secretas

Integración de Vault



¿Cómo manejar contraseñas, tokens de API, comunidades SNMP de forma segura en Zabbix?

► Macros Secretas

- Ya las mencionamos: {\$NOMBRE_MACRO_SECRETO}. El valor se oculta (*****) en la interfaz tras guardarlo.
- Si no se usa Vault, se almacenan cifradas en la base de datos de Zabbix (mejor que texto plano).
- Usar para: credenciales de ítems, comunidades SNMP, etc.

► Integración con HashiCorp Vault

- Configuras Zabbix Server/Proxy para conectarse a tu instancia de Vault.
- crear una macro secreta en Zabbix, en lugar del valor, indicas la **ruta al secreto en Vault**
- Zabbix Server/Proxy consulta a Vault para obtener el valor del secreto EN **TIEMPO DE EJECUCIÓN**

► Participa del workshop para configurar Vault y encriptación a la DB!

Auditoría y Monitoreo de Zabbix

Vigilando al Vigilante: Auditoría y Monitoreo de Zabbix

- ▶ Log de Auditoría de Zabbix (Reports-> Audit Log)
- ▶ Registra la mayoría de los eventos (los que producen cambios)
 - Inicio de sesión incluidos, ABM de hosts, items, etc
 - **Práctica Clave:** Revisar periódicamente, especialmente tras cambios importantes o si se sospecha actividad inusual.
- ▶ Monitorear la Salud del Propio Zabbix (Internals)
 - Métricas Críticas a Vigilar:
 - % de procesos ocupados de Zabbix (pollers, trappers, discoverers, icmp pingers, etc.).
 - % de uso de cachés (value cache, configuration cache, history cache).
 - Errores en los logs del servidor/proxy Zabbix.
 - Rendimiento y espacio de la base de datos de Zabbix.

Hardening del Sistema Operativo

Acciones Esenciales de Hardening

► Gestión de Parches Rigurosa:

- El OS y las librerías (requerimientos) se recomienda, estén en la ultima versión soportada
- Automatizar este proceso de ser posible

► Minimizar la superficie de ataque

- Instalar solo lo necesario para el componente (Server/Proxy/etc)
- Eliminar usuarios innecesarios
- Cerrar puertos no utilizados
- Usar sudo para escalar privilegios
- Incorporar y administrar SELinux de ser posible

Gracias!

Facundo Vilarnovo
Trainer

