

**ZABBIX** '24

CONFERENCE

LATIN AMERICA

JUNE, 07-08, 2024. SÃO PAULO, BRAZIL

Presentación:

**Matias Cela Diaz**

**# Tech Lead NOC.**

**# ZCS** - Zabbix Certified Specialist.

**# Lic. en Telecomunicaciones.**



**CTL**  
**OPSERVERS**

Preparativos para Zabbix 7.0

# CONSIDERACIONES DE ARQUITECTURAS COMPLEJAS





## # Negócio:

- Financiadora número 1 de salud.
- Es nexo entre el socio y las prestadoras de salud.
- Brinda servicios de Telemedicina, autorizaciones, gestión de pacientes, cartilla médica, aplicación Móvil.

## # Números:

Múltiples centros de datos distribuidos en todo el país.

- + 200 centros de atención
- + 10.000 transacciones diárias.
- + 2.000.000 sócios.

# # Ecosistema del cliente:

- ONPREM | CLOUD | PRIVADO
- MULTI VENDOR
- STACK OPEN SOURCE | PROPIETÁRIO
- ENTORNOS DE APIS | IOT





“Cualquier dato que podemos obtener programáticamente, se puede monitorear en Zabbix”

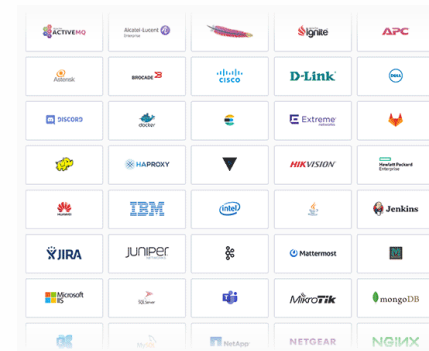
## # Principales desafíos:

- HETEROGENEIDAD DE TECNOLOGÍAS
- ESCALABILIDAD
- AUTOMATIZACIÓN
- UPGRADE A ZABBIX 7.0 🔥

# Heterogeneidad de Tecnologías ❌

## Cómo lo resolvimos:

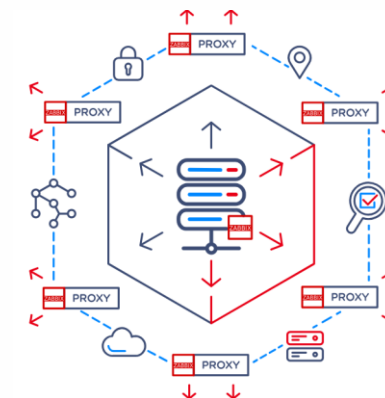
Aprovechar los templates predefinidos de Zabbix para tecnologías comunes y desarrollar templates personalizados para tecnologías específicas. Utilizar agentes Zabbix y SNMP para cubrir una mayor variedad de dispositivos y servicios.



# Escalabilidad ❌

## Cómo lo resolvimos:

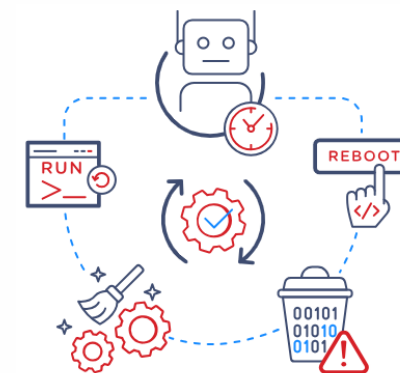
Implementando una arquitectura distribuida con proxies de Zabbix para distribuir la carga. Optimizando la base de datos y considerar el uso de bases de datos más escalables como PostgreSQL o TimescaleDB.



# Automatización ❌

## Cómo lo resolvimos:

Utilizando scripts y la API de Zabbix para automatizar la configuración y gestión. Implementar macros y user parameters para facilitar la personalización.





# Actualización de Zabbix 7.0

# Preparación de entorno



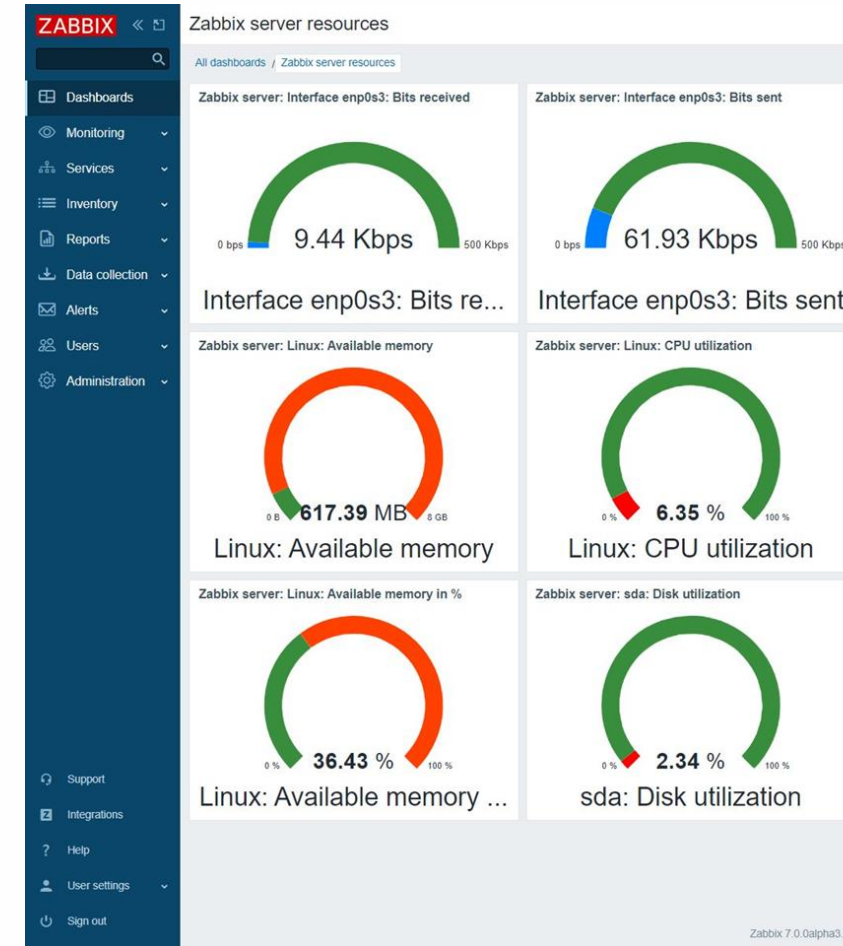
# PLANIFICACIÓN

6 puntos claves!

- 1. Análisis inicial:** Compatibilidad entornos e identificación de dependencias y posibles conflictos.
- 2. Backups:** Base de Datos MySQL, Zabbix Servers, proxies y verificación integridad de backups.
- 3. Pruebas de entornos.**
- 4. Ampliación de Proxies.**
- 5. Actualización** de proxies, Zabbix Server y Base de Datos.
- 6. Verificación** Post Actualización.

# CONFIGURACIONES NUEVAS PARA **ZABBIX 7.0**

- Proxies en Alta Disponibilidad ✓
- Proxy Buffer en Memoria ✓
- API REST mejorada con nuevas funciones ✓
- Ejemplo de Script API REST Zabbix V.6 vs V.7 ✓
- Recolección de Datos Asíncrona ✓
- Timeouts por Ítems ✓
- Actualización de dashboards y widgets ✓
- Seguridad con Autenticación multifactor (MFA) ✓



# Proxies en Alta Disponibilidad

Configurar múltiples proxies y habilitación de failover para garantizar disponibilidad continua.

## Consideraciones

- **Redundancia:** Configurar múltiples proxies para garantizar que si uno falla, otro pueda tomar su lugar sin interrupciones en el monitoreo.
- **Escalabilidad:** Distribuir la carga de monitoreo entre varios proxies para manejar más dispositivos y servicios de manera eficiente.
- **Eficiencia:** Verificar que la configuración optimiza el uso de recursos para reducir los puntos únicos de fallo.
- **Compatibilidad:** Confirmar que los proxies configurados sean compatibles con las nuevas funcionalidades y mejoras en Zabbix 7.
- **Configuración de Failover:** Asegurar que la habilitación de failover esté correctamente configurada para una transición sin problemas entre proxies.

## zabbix\_proxy.conf Proxy 1

```
Server=192.168.1.100
Hostname=Proxy1
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=zabbix_password
```

## zabbix\_proxy.conf Proxy 2

```
Server=192.168.1.100
Hostname=Proxy2
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=zabbix_password
```

# Proxy Buffer en Memoria

Mejor manejo del buffer para reducir pérdida de datos.

## Consideraciones:

- **Mejora de Rendimiento:** Configurar el almacenamiento temporal de datos en memoria para reducir la latencia y mejorar la eficiencia en la transferencia de datos.
- **Tolerancia a Fallos:** Verificar que la configuración del buffer permite manejar interrupciones temporales en la comunicación entre el proxy y el servidor central, evitando la pérdida de datos.
- **Flexibilidad:** Ajustar el tamaño del buffer según las necesidades específicas del entorno para optimizar el rendimiento.

```
# Size of configuration cache in bytes.  
# ConfigCacheSize=16M  
  
# Size of history cache, in bytes.  
# HistoryCacheSize=128M  
  
# Size of history text cache, in bytes.  
# HistoryTextCacheSize=16M  
  
# Size of value cache, in bytes.  
# ValueCacheSize=64M  
  
# Proxy local buffer size, in bytes.  
# Increase this value to avoid dropped values in case of temporary communication  
ProxyLocalBuffer=1G  
  
# Number of pre-forked instances of pollers.  
StartPollers=10  
  
# Number of pre-forked instances of data senders.  
StartDataSenders=5
```

# API REST mejorada con nuevas funciones

Zabbix 7 introduce mejoras significativas en su API REST, proporcionando nuevas funcionalidades y mayor flexibilidad para la integración con otras herramientas y la automatización de tareas.

## Consideraciones

- **Mayor Flexibilidad y Personalización:** Asegurar de poder añadir etiquetas y macros directamente a través de la API para gestionar hosts de manera más eficiente.
- **Automatización Avanzada:** Verificar si las nuevas funcionalidades de la API simplifican la automatización de tareas complejas, reduciendo el esfuerzo manual.
- **Compatibilidad:** Confirmar que las integraciones existentes son compatibles con las nuevas funcionalidades de la API REST en Zabbix 7.
- **Seguridad:** Revisar y ajustar las configuraciones de seguridad para asegurar que el uso de la API REST mejorada no introduce vulnerabilidades.

# Ejemplo de Script API REST Zabbix V.6 vs V.7

- **Zabbix 6:** El script crea un host básico y lo asigna a un grupo y a un template específico.
- **Zabbix 7:** Además de las configuraciones básicas, el script añade etiquetas (tags) y macros (macros). Las etiquetas permiten clasificar y buscar hosts más fácilmente, mientras que las macros proporcionan una forma flexible de configurar ítems con valores personalizados.

```
#!/bin/bash

# URL del servidor Zabbix
ZABBIX_SERVER="http://zabbix6-server/api_jsonrpc.php"

# Credenciales de autenticación
USER="Admin"
PASSWORD="zabbix"

# Obtener token de autenticación
AUTH_TOKEN=$(curl -s -X POST -H "Content-Type: application/json" -d '{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "user": "'${USER}'",
    "password": "'${PASSWORD}'"
  },
  "id": 1
}' ${ZABBIX_SERVER} | jq -r .result)

# Crear nuevo host
curl -s -X POST -H "Content-Type: application/json" -d '{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "ExampleHost",
    "interfaces": [
      {
        "type": 1,
        "main": 1,
        "useip": 1,
        "ip": "192.168.0.1",
        "dns": "",
        "port": "10050"
      }
    ],
    "groups": [
      {
        "groupid": "1"
      }
    ],
    "templates": [
      {
        "templateid": "10001"
      }
    ]
  },
  "auth": "'${AUTH_TOKEN}'",
  "id": 1
}' ${ZABBIX_SERVER}
```

```
#!/bin/bash

# URL del servidor Zabbix
ZABBIX_SERVER="http://zabbix7-server/api_jsonrpc.php"

# Credenciales de autenticación
USER="Admin"
PASSWORD="zabbix"

# Obtener token de autenticación
AUTH_TOKEN=$(curl -s -X POST -H "Content-Type: application/json" -d '{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "user": "'${USER}'",
    "password": "'${PASSWORD}'"
  },
  "id": 1
}' ${ZABBIX_SERVER} | jq -r .result)

# Crear nuevo host con etiquetas y macros
curl -s -X POST -H "Content-Type: application/json" -d '{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "ExampleHost",
    "interfaces": [
      {
        "type": 1,
        "main": 1,
        "useip": 1,
        "ip": "192.168.0.1",
        "dns": "",
        "port": "10050"
      }
    ],
    "groups": [
      {
        "groupid": "1"
      }
    ],
    "templates": [
      {
        "templateid": "10001"
      }
    ],
    "tags": [
      {
        "tag": "Environment",
        "value": "Production"
      },
      {
        "tag": "Department",
        "value": "IT"
      }
    ],
    "macros": [
      {
        "macro": "${CPU_LOAD}",
        "value": "5"
      }
    ]
  },
  "auth": "'${AUTH_TOKEN}'",
  "id": 1
}' ${ZABBIX_SERVER}
```

# Recolección de Datos Asíncrona

Introducción de pollers asíncronos para SNMP, agentes Zabbix y verificaciones. Configurar los nuevos pollers en [zabbix\\_server.conf](#), ajustando parámetros como **StartSNMPPollers**, **StartPollersUnreachable** y **StartHTTTPollers**.

## Consideraciones:

- **Eficiencia:** Se deberá asegurar de que los pollers asíncronos estén correctamente configurados para procesar más datos sin esperar respuestas anteriores.
- **Escalabilidad:** Se deben ajustar los parámetros para manejar más dispositivos simultáneamente, asegurando que el sistema pueda escalar según las necesidades.
- **Reducción de Latencia:** Configurar los pollers para reducir el tiempo de espera entre solicitudes y mejorar el rendimiento general.
- **Resiliencia:** Se debe verificar que la configuración permite continuar recolectando datos pese a demoras o fallos en la red.

## zabbix\_server.conf

```
StartSNMPPollers=10
StartPollersUnreachable=5
StartHTTTPollers=10
Timeout=4
SNMPTrapperTimeout=10
HTTPTimeout=15
```



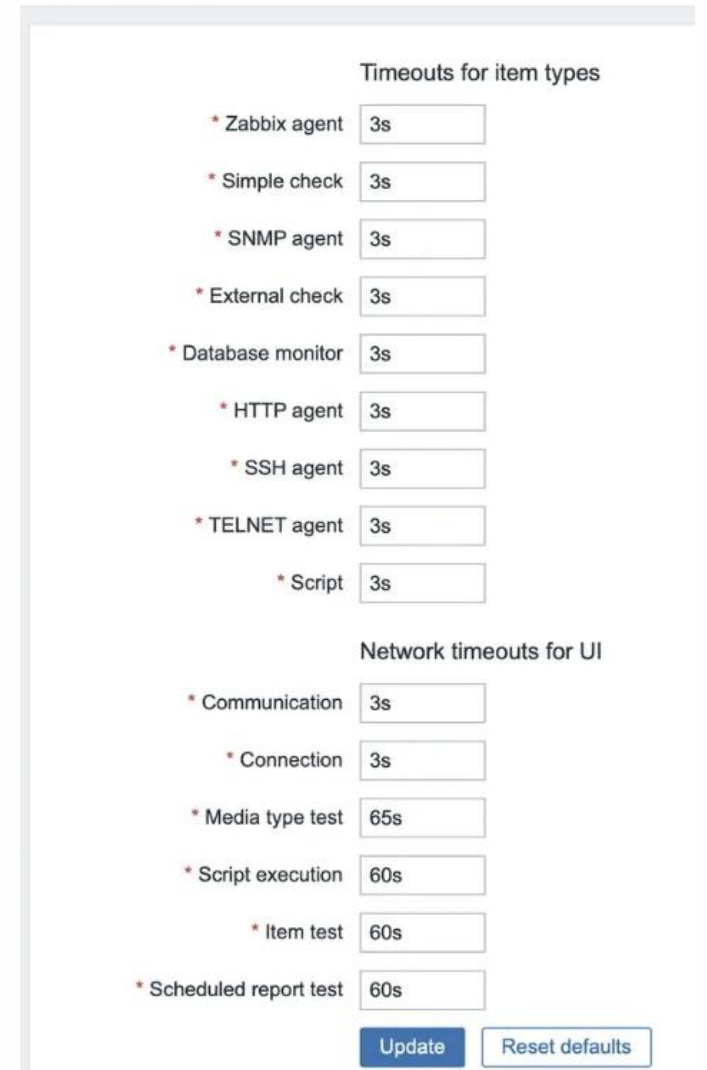
# Timeouts por Ítems

Se establecen timeouts individuales por ítem directamente en su configuración. Zabbix 7 permite ajustar los timeouts a nivel de ítem, ofreciendo mayor precisión en la configuración y mejorando el rendimiento de la recolección de datos.

## Consideraciones:

- **Granularidad:** Timeouts a nivel de ítem precisos y adecuados para cada configuración específica.
- **Optimización:** Se ajustan los timeouts según la criticidad y los requisitos específicos de cada ítem.
- **Flexibilidad:** Gestión adecuada de los entornos complejos con múltiples servicios y aplicaciones, según necesidades de tiempo de respuesta.

## Timeouts ▾



The screenshot shows the 'Timeouts for item types' configuration page in Zabbix. It lists various item types with their respective timeout values, each in a text input field. Below this section is the 'Network timeouts for UI' section, also with input fields for different network-related timeouts. At the bottom right, there are two buttons: 'Update' and 'Reset defaults'.

Item Type	Timeout
* Zabbix agent	3s
* Simple check	3s
* SNMP agent	3s
* External check	3s
* Database monitor	3s
* HTTP agent	3s
* SSH agent	3s
* TELNET agent	3s
* Script	3s

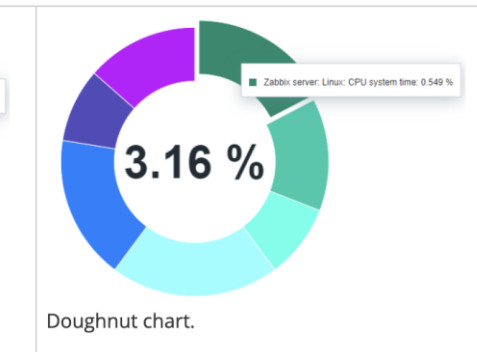
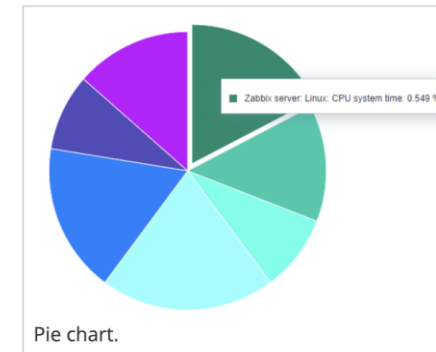
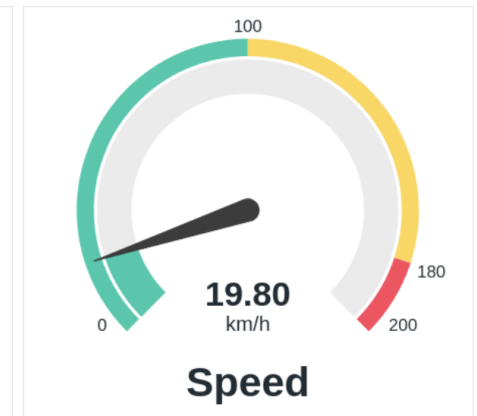
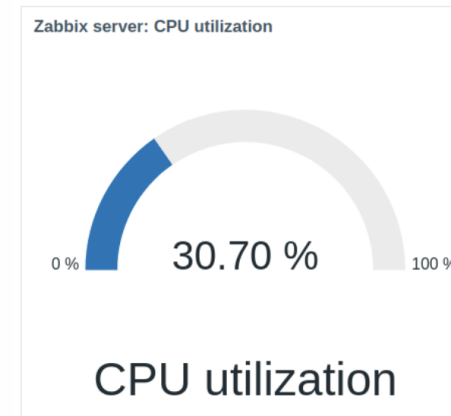
Network timeout for UI	Timeout
* Communication	3s
* Connection	3s
* Media type test	65s
* Script execution	60s
* Item test	60s
* Scheduled report test	60s

# Actualización de dashboards y widgets

Implementación de widgets avanzados para dashboards, que facilitan una monitorización más eficiente y una mejor visualización de los datos.

## Consideraciones:

- Verificación de compatibilidad.
- Verificación widgets obsoletos.
- Implementación de nuevos Widgets: Gauge, Pie Chart, Honeycomb, entre otros.

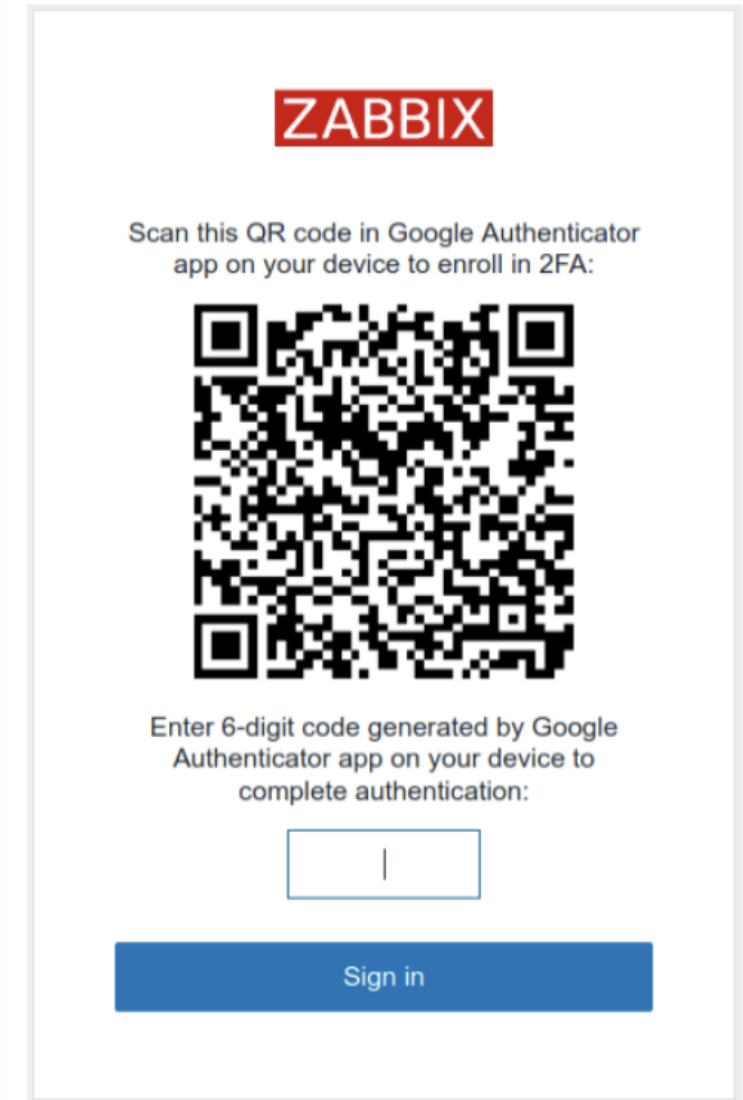


# Seguridad con Autenticación multifactor (MFA)

Implementación de MFA para mayor seguridad.

## Consideraciones:

- Autenticación multifactor obligatorio para todos los usuarios, especialmente aquellos con permisos administrativos.
- Configuración de alertas para intentos de inicio de sesión fallidos y otras actividades sospechosas.
- Utilización de dashboards y widgets mejorados en Zabbix 7.0 para monitorear la actividad de inicio de sesión.



# Conclusiones y Recomendaciones para la Actualización a **Zabbix 7.0** ⚡

Se requiere una planificación cuidadosa y nuevas configuraciones para maximizar las mejoras en rendimiento, seguridad y gestión:

- **Migrar a bases de datos optimizadas: PostgreSQL o TimescaleDB.**
- **Configurar proxies en alta disponibilidad.**
- **Utilizar las nuevas funcionalidades de la API REST.**
- **Implementar la recolección de datos asincrónica.**
- **Actualizar dashboards y widgets para una mejor visualización.**
- **Configurar la autenticación multifactor (MFA) para mejorar la seguridad.**

Estas acciones asegurarán una transición exitosa y aprovecharán al máximo las mejoras de Zabbix 7.0

ZABBIX '24

CONFERENCE

LATIN AMERICA

¡Muchas gracias!



CTL  
OPSERVERS