

[zabbix](#), [telegram](#), [notificaciones](#), [alarma](#), [aviso](#)

Notificaciones de Zabbix por Telegram

Con un script bash

<https://github.com/diegosmaia/zabbix-telegram>

Se descarga el script en el directorio AlertScriptsPath directory.



El path a dicho directorio está definido en el fichero zabbix_server.conf. En mi caso AlertScriptsPath=/usr/lib/zabbix/alertscripts

se le da permisos de ejecución

```
chmod +x zabbix-telegram.sh
```

En el script hay que modificarlo para añadir los parámetros correspondientes a nuestra configuración. En mi caso particular he modificado el script que te descargas desde la página de <https://github.com/diegosmaia/zabbix-telegram> y he modificado alguna de las llamadas a curl añadiendo la opción -noproxy '*' para evitar errores de ejecución ya que mi servidor utiliza un proxy para salir a internet.



Para saber el id del grupo creas un grupo nuevo y añades al bot a dicho grupo y desde el navegador accedes a la página <https://api.telegram.org/bot<token>/getUpdates> para obtener el id usuario y el id grupo. El token te llega en un mensaje de telegram



Ojo el bot tiene su propio id, que es distinto del id para el grupo

Para hacer una prueba ejecutamos el siguiente comando desde la consola situandonos en el directorio donde está localizado el script

```
./zabbix-telegram.sh -123456789 "prueba 12:43" "Item Graphic: [37502]"
```

. Al ejecutarlo debería de llegarnos varios mensajes y un gráfico



-123456789 hay que cambiarlo por nuestro id de grupo que debemos obtener anteriormente

Configuración del zabbix para que envíe las alarmas

Creamos un nuevo **Media Types** zabbix→Administration→Media types→Create Media type.

- Como nombre ponemos telegram, tipo script, y en Script name ponemos el nombre de nuestro script **zabbix-telegram.sh** y marcamos la casilla **Enabled**.

En la casilla de Script Parameters pinchamos en **Add** y añadimos los siguientes parámetros

The screenshot shows the Zabbix Administration interface. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. The 'Administration' tab is active, and the 'Media types' sub-tab is selected. The 'Media types' page has two tabs: 'Media type' (active) and 'Options'. The 'Media type' tab contains the following fields and controls:

- Name:** A text input field containing 'telegram'.
- Type:** A dropdown menu set to 'Script'.
- Script name:** A text input field containing 'zabbix-telegram.sh'.
- Script parameters:** A table with two columns: 'Parameter' and 'Action'. It contains three rows:
 - Parameter: '{ALERT.SENDTO}', Action: 'Remove' (link).
 - Parameter: '{ALERT.SUBJECT}', Action: 'Remove' (link).
 - Parameter: '{ALERT.MESSAGE}' (highlighted in yellow), Action: 'Remove' (link).Below the table is an 'Add' link.
- Enabled:** A checkbox that is checked.
- Buttons:** 'Update' (blue), 'Clone' (light blue), 'Delete' (light blue), and 'Cancel' (light blue).

- Ahora vamos a zabbix→Administration→User. Elegimos el usuario con el que enviamos las alarmas y pinchamos sobre el menú **Media** y después en **add** para añadir los parámetros de configuración

Media

Type

Send to

When active

Use if severity ☒ Not classified
☒ Information
☒ Warning
☒ Average
☒ High
☒ Disaster

Enabled ☒

- Por último vamos al menú zabbix→Configuration→Actions En la casilla de Event Source seleccionamos **Triggers** y pulsamos sobre el botón **Create action** o bien modificamos una de las acciones existentes.

En nuestro caso creamos una nueva acción con los siguientes datos

1. Name → Enviar alarmas por telegram
2. Default subject → {HOSTNAME}:{TRIGGER.NAME}-Status-{TRIGGER.STATUS}
3. Default message →

```
Trigger:{TRIGGER.NAME}
Trigger status:{TRIGGER.STATUS}
Trigger severity:{TRIGGER.SEVERITY}
Trigger URL:{TRIGGER.URL}
```

Item values:

```
1. {ITEM.NAME1} ( {HOST.NAME1} : {ITEM.KEY1} ) : {ITEM.VALUE1}
```

```
Item Graphic:[{ITEM.ID1}]
```

ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Discovery IT services

Actions

Action Conditions Operations

Name: Enviar alarmas por telegram

Default subject: {HOSTNAME}:{TRIGGER.NAME}-Status:{TRIGGER.S}

Default message: Trigger: {TRIGGER.NAME}
Trigger status: {TRIGGER.STATUS}
Trigger severity: {TRIGGER.SEVERITY}
Trigger URL: {TRIGGER.URL}

Item names:

Recovery message: ☒

Enabled: ☒

Update Clone Delete Cancel

Zabbix 3.0.3 © 2001-2019, Zabbix SIA

1. Click en el menú **operations** y configuramos

ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Actions Discovery IT services

Actions

Action Conditions Operations

Default operation step duration: 3000 (minimum 60 seconds)

Action operations: 1 Send message to user groups: Zabbix administrators via Telegram immediately Default

Operation details:

Steps: 1 - 1 (0 - infinite)

Step duration: 0 (minimum 60 seconds, 0 - use action default)

Operation type: Send message

Send to user groups: Zabbix administrators

Send to Users: (empty)

Send only to: Telegram

Default message: ☒

Conditions: LABEL NAME ACTION

Update Cancel

Update Clone Delete Cancel



para saber si las alarmas se están enviando correctamente ir a: Zabbix→Reports→Action log

```
#!/bin/bash
```

```
#####  
# Zabbix-Telegram envio de alerta por Telegram con graficos dos eventos  
# Filename: zabbix-telegram.sh  
# Revision: 2.1  
# Date: 24/04/2016
```

```
# Author: Diego Maia - diegosmaia@yahoo.com.br Telegram - @diegosmaia
# Aproveitei algumas coisas:
# Script getItemGraph.sh Author: Qicheng
# https://github.com/GabrielRF/Zabbix-Telegram-Notification @GabrielRF
# Obs.: Se añade la opción de usar el script en caso de estar detras de un
proxy (wiki.intrusos.info)
#####

MAIN_DIRECTORY="/usr/lib/zabbix/alertscripts/"

USER=$1
SUBJECT=$2
SUBJECT="${SUBJECT//,/ /}"
MESSAGE="chat_id=${USER}&text=$3"
GRAPHID=$3
GRAPHID=$(echo $GRAPHID | grep -o -E "(Item Graphic: \[[0-9]{7}\])|(Item
Graphic: \[[0-9]{6}\])|(Item Graphic: \[[0-9]{5}\])|(Item Graphic:
\[[0-9]{4}\])|(Item Graphic: \[[0-9]{3}\])")
GRAPHID=$(echo $GRAPHID | grep -o -E
"([0-9]{7})|([0-9]{6})|([0-9]{5})|([0-9]{4})|([0-9]{3})")

ZABBIXMSG="/tmp/zabbix-message-$(date "+%Y.%m.%d-%H.%M.%S").tmp"

#####
# URL donde esta el Zabbix (modificar)
#####
ZBX_URL="http://192.168.0.102/zabbix"

#####
# Usuario y clave para logearse en el Zabbix (modificar)
#####

USERNAME="admin"
PASSWORD="zabbix"

#####
# Bot-Token de ejemplo (modificar)
#####

BOT_TOKEN='161080402:AAGah3HIxM9jUr0NX1WmEKX3cJCv9PyWD58'

#####
# Si enviar o no GRAFICOS / ENVIA_GRAFICO = 0
# Si enviar o no MENSAJES / ENVIA_MESSAGE = 0
#####

ENVIA_GRAFICO=1
ENVIA_MESSAGE=1

# Se não receber o valor de GRAPHID ele seta o valor de ENVIA_GRAFICO para 0
```

```
case $GRAPHID in
    '*'[!0-9]*) ENVIA_GRAFICO=0 ;;
    *) ENVIA_GRAFICO=1 ;;
esac

#####
# Graficos
#####

WIDTH=800
CURL="/usr/bin/curl"
COOKIE="/tmp/telegram_cookie-$(date "+%Y.%m.%d-%H.%M.%S")"
PNG_PATH="/tmp/telegram_graph-$(date "+%Y.%m.%d-%H.%M.%S").png"

#####
# Periodo do grafico em minutos Exp: 10800min/3600min=3h
#####

PERIOD=10800

#####
# Verifica se foi passado os 3 parametros
# para o script
#####

if [ "$#" -lt 3 ]
then
    exit 1
fi

#####
# Envio Mensaje de Alerta usando proxy
#####

echo "$MESSAGE" > $ZABBIXMSG
${CURL} -x http://iproxy:puerto -k -s -c ${COOKIE} -b ${COOKIE} -s -X GET
"https://api.telegram.org/bot${BOT_TOKEN}/sendMessage?chat_id=${USER}&text=\
"${SUBJECT}"\" > /dev/null

if [ "$ENVIA_MESSAGE" -eq 1 ]
then
    ${CURL} -x http://iproxy:puerto -k -s -c ${COOKIE} -b ${COOKIE} --data-
binary @${ZABBIXMSG} -X GET
"https://api.telegram.org/bot${BOT_TOKEN}/sendMessage" > /dev/null
fi
#####
# Envio dos graficos
```

```
#####
```

```
# Se ENVIA_GRAFICO=1 ele envia o gráfico.
```

```
if [ $($ENVIA_GRAFICO) -eq '1' ]; then
```

```
#####
```

```
# Zabbix logando com o usuário no site
```

```
#####
```

```
# Zabbix - Ingles - Verifique no seu Zabbix se na tela de login se o  
botao de login é "Sign in".
```

```
# Obs.: Caso queira mudar, abra a configuração do usuário Guest e mude a  
linguagem para Portugues, se fizer isso comente (#) a linha abaixo e  
descomente a linha Zabbix-Portugues.
```

```
    ${CURL} --noproxy '*' -k -s -c ${COOKIE} -b ${COOKIE} -d  
"name=${USERNAME}&password=${PASSWORD}&autologin=1&enter=Sign%20in"  
${ZBX_URL}/index.php" > /dev/null
```

```
# Zabbix - Portugues - Verifique no seu Zabbix se na tela de login  
se o botao de login é "Conectar-se".
```

```
# ${CURL} -k -s -c ${COOKIE} -b ${COOKIE} -d  
"name=${USERNAME}&password=${PASSWORD}&autologin=1&enter=Conectar-se"  
${ZBX_URL}/index.php" > /dev/null
```

```
# Download do gráfico e envio
```

```
    ${CURL} --noproxy '*' -k -s -c ${COOKIE} -b ${COOKIE} -d  
"itemids=${GRAPHID}&period=${PERIOD}&width=${WIDTH}" ${ZBX_URL}/chart.php"  
-o "${PNG_PATH}"
```

```
    ${CURL} -x http://ipproxy:puerto -k -s -X POST  
"https://api.telegram.org/bot${BOT_TOKEN}/sendPhoto" -F chat_id="${USER}" -F  
photo="@${PNG_PATH}" > /dev/null
```

```
fi
```

```
#####
```

```
# DEBUG
```

```
#####
```

```
# Verificar valores recebidos do Zabbix ou do prompt
```

```
# cat /tmp/telegram-debug.txt
```

```
# echo "User-Telegram=$USER | Subject=$SUBJECT | Menssage=$MESSAGE |  
GraphID=${GRAPHID} | Period=${PERIOD} | Width=${WIDTH}" >/tmp/telegram-  
debug.txt
```

```
# Teste com curl tentando baixar o gráfico
```

```
# Verifique o arquivo /tmp/telegram-graph.png no seu computador para ver se  
o grafico esta sendo gerado corretamente
```

```
##${CURL} --noproxy '*' -k -c ${COOKIE} -b ${COOKIE} -d  
"graphid=51179&itemids=51179&period=10800&width=800"  
172.19.3.202/zabbix/chart.php > /tmp/telegram-graph2.png
```

#Verificando o envio da msg

Envio da msg de texto

Gera uma saída no script com algo parecido com isso

```
{"ok":true,"result":{"message_id":xxx,"from":{"id":xxxx,"first_name":"xxx","username":"xxxx"},"chat":{"id":xxxxx,"first_name":"xxx","last_name":"xxx","username":"xxxxx","type":"private"},"date":xxxx,"text":"teste"}}
```

Se gerar uma saída diferente verifique o seu BOT_TOKEN ou então o UserID ou Group-ID para qual a msg esta sendo enviada

\${CURL} -k -c \${COOKIE} -b \${COOKIE} -X GET

"https://api.telegram.org/bot\${BOT_TOKEN}/sendMessage?chat_id=\${USER}&text=\${SUBJECT}"

Envio do Grafico

`\${CURL} -k -X POST "https://api.telegram.org/bot\${BOT_TOKEN}/sendPhoto" -F chat_id="\${USER}" -F photo="@\${PNG_PATH}"

#####

Apagando os arquivos utilizados no script

#####

rm -f \${COOKIE}

rm -f \${PNG_PATH}

rm -f \${ZABBIXMSG}

exit 0

Mediante curl

Utilizando el siguiente script podemos enviar mensajes directamente con curl

```
#!/bin/bash
TOKEN=tutoken
CHAT_ID=id del chat
MESSAGE="Hola Mundo"
URL="https://api.telegram.org/bot$TOKEN/sendMessage"
curl -s -X POST $URL -d chat_id=$CHAT_ID -d text="$MESSAGE"
```

Con un script en python

<https://github.com/ableev/Zabbix-in-Telegram>

Instalamos en el servidor python

yum install python

yum install python-pip

Instalamos el módulo **request** necesario para ejecutar los scripts

```
pip install requests
```

- Ponemos el script zbxtg.py en el directorio AlertScriptsPath directory. El path a dicho directorio está definido en el fichero zabbix_server.conf. En mi caso AlertScriptsPath=/usr/lib/zabbix/alertscripts
- Copiamos zbxtg_group.py en la misma localización si queremos enviar mensajes a grupos
- Creamos el fichero zbxtg_settings.py con la configuración nuestra en el mismo directorio donde está el script
- Creamos un bot en telegram y apuntamos la API key
- creamos un usuario de sólo lectura en Zabbix (paa obtener los gráficos e imagenes del zabbix)
- Si usamos proxy debe de estar configurado en el fichero de configuración zbxtg_settings.py
- Añadimos en zabbix→Administration_>Media types→Create Media type. Dos nuevos tipos de **media types** para Telegram con las siguiente configuración

y otro para los grupos

From:
<http://wiki.intrusos.info/> - LCWIKI

Permanent link:
<http://wiki.intrusos.info/doku.php?id=seguridad:monitorizacion:zabbix3:telegram&rev=1513683511>

Last update: **2023/01/18 14:39**

