

# ZABBIX

## Alertas de ligações via Asterisk



Autor: Fabricio Guimarães  
E-mail: [theguimaraes@gmail.com](mailto:theguimaraes@gmail.com)  
Telegram: @theguima

## Sumário

<b>Introdução</b>	<b>2</b>
<b>Requisitos</b>	<b>3</b>
Instalação Script AGI Googletts	3
Instalação Script de chamadas via Asterisk	4
<b>Configurações Zabbix Server</b>	<b>5</b>
<b>Configurações Asterisk</b>	<b>6</b>
<b>Configuração Front-End</b>	<b>9</b>
<b>Observações Gerais</b>	<b>11</b>
<b>Conclusões</b>	<b>12</b>

# Introdução

A proposta deste projeto foi desenvolver um script, onde em caso de alguns alertas específicos do Zabbix, recebemos uma ligação via celular/ramal.

Este recurso mostrou ser útil devido a facilidade de comunicar certos tipos de incidentes críticos, onde as equipes respnsáveis são comunicadas mais rapidamente, reduzindo o tempo de uma possível indisponibilidade.

Também podemos usar este recurso para escalonar algum assunto para um supervisor/coordenador/gerente.

O script desenvolvido irá executar as seguintes ações para a realização das chamadas:

- Zabbix irá disparar uma trigger
- Com a ação configurada, será executado o script de ligação do Asterisk no Zabbix Server.
- Será construído um arquivo de texto com a extensão .call no diretório /temp
- O arquivo será montado recebendo os valores do número de telefone cadastrado individualmente no usuário do Zabbix, o Canal de origem da ligação, o Ramal de origem válido, o contexto do dialplan configurado, a extensão para o contexto, a mensagem recebida pela Action, e se deseja ou não arquivar.
- Será alterado o permissionamento do arquivo para o usuário "asterisk"
- Será movido o arquivo da pasta /tmp do Zabbix para a pasta /tmp do Asterisk
- O Crontab do Asterisk irá mover o arquivo de ligação para o diretório de spool onde será feito a ligação.
- O Asterisk irá identificar o canal da ligação e a origem.
- A ligação será direcionada para o contexto "zabbix" do dialplan do Asterisk
- Com a extensão passada, irá executar o comando relacionado.
- Será executado o script AGI do Google onde enviaremos o texto que queremos traduzir para voz e a linguagem.
- A ligação será derrubada.

# Requisitos

## Instalação Script AGI Googletts

Para que essa integração funcione, devemos utilizar um script AGI do Google TTS para Asterisk desenvolvido por Lefteris Zafiris, onde pode ser acessado pelo endereço abaixo:

<https://github.com/zaf/asterisk-googletts>

Os detalhes para a instalação estão na documentação do Github, porém segue abaixo.

Instalar as dependências abaixo no servidor do Asterisk para que funcione a tradução de texto em voz.

- Perl The Perl Programming Language
- perl-libwww The World-Wide Web library for Perl
- perl-LWP-Protocol-https For HTTPS support
- sox Sound eXchange, sound processing program
- mpg123 MPEG Audio Player and decoder
- Internet access in order to contact google and get the voice data.

Exemplo no CentOS 7:

```
yum install perl
yum install perl-libwww
yum install perl-LWP-Protocol-https
yum install sox
yum install mpg123
```

Copiar o arquivo googletts.agi do Git acima, para o diretório agi-bin do seu asterisk.

- Normalmente este diretório fica em /var/lib/asterisk/agi-bin, porém caso você tenha uma instalação customizada, você pode conferir o diretório AGI acessando os detalhes em /etc/asterisk/asterisk.conf

O funcionamento é simples, passando o comando no dialplan do asterisk ele irá chamar o Google TTS, passando a frase que você deseja, a língua da pronúncia, uma tecla de interrupção(opcional) e a velocidade da fala(opcional)

```
agi(googletts.agi,"text",[language],[intkey],[speed])
```

## Instalação Script de chamadas via Asterisk

Copiar o arquivo asterisk\_call.sh do script de chamadas para o diretório de alertas do seu Zabbix, usualmente: /usr/lib/zabbix/alertscripts/

Dar permissão de execução para o arquivo (# chmod 755 asterisk\_call.sh)

```
#!/bin/bash

# *****
#
# Script de ligação via asterisk para alertas enviados pelo Zabbix
#
# Ligação realizada pelo canal KHOMP, caso utilize outro canal, necessário
# alterar no Script abaixo no item CANAL.
#
# Autor: Fabricio Guimarães
#
# *****

CANAL='KHOMP/b1/'          #Canal utilizado para fazer as ligações
ORIGEM='r_ZABBIX <1234>'    #Ramal de Origem válido, necessário para alguns canais
                             #funcionarem (como o Khomp)
CONTEXTO='zabbix'          #Cotexto da ligação no Diaplan do Asterisk
EXTENSAO='zabbix'          #Extensão utilizada para a ligação
ARQUIVAR=yes               #Arquivar histórico das ligações (yes or no)

#Salva um arquivo de texto na pasta temporária do Servidor do Zabbix
echo -en "Channel: $CANAL$1\nCallerID: $ORIGEM\nContext: $CONTEXTO\nExtension:
$EXTENSAO\nSetvar: MESSAGE=$2\nArchive: $ARQUIVAR" >>
/tmp/ligacao_zabbix.call

#Da permissão ao arquivo para que o asterisk possa consumir o mesmo
sudo chown asterisk:asterisk /tmp/ligacao_zabbix.call

#Move o arquivo para o servidor de PABX
sudo scp /tmp/ligacao_zabbix.call root@192.168.5.30:/tmp

#Apaga o arquivo temporário do Servidor do Zabbix
sudo rm -Rf /tmp/ligacao_zabbix.call
```

# Configurações Zabbix Server

Primeiramente devemos instalar o SCP caso não tenha instalado. Utilizaremos este sistema para copiar os arquivos para o servidor asterisk com segurança automaticamente. Pegando o exemplo do CentOS, basta executar o comando abaixo:

```
#yum install -y scp
```

Para que o script funcione adequadamente, devemos criar uma chave SSH, para que o servidor do Zabbix consiga enviar o arquivo criado para o servidor do Asterisk sem a necessidade de autenticação e confirmação de usuário/senha. Para isso, acessamos a pasta onde fica a chave SSH, normalmente:

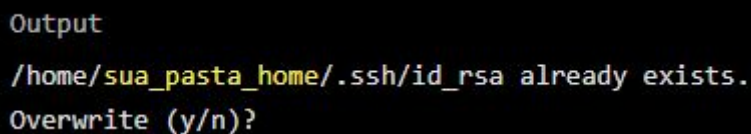
```
#cd /root/.ssh/
```

Executamos o comando para criar a chave SSH que queremos. Este comando cria uma chave RSA de 2048 bits, o que normalmente é segura para a maioria dos casos. Você pode opcionalmente passar o flag (-b 4096) para criar uma chave mais extensa de 4096 bits.

```
#ssh-keygen -t rsa
```

Será solicitado o local onde deseja salvar a chave, basta deixar em branco e dar um Enter,

Se você tiver gerado um par de chaves SSH anteriormente, você pode ver o seguinte prompt:



```
Output
/home/sua_pasta_home/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

Se você escolher sobrepor a chave no disco, você não poderá autenticar usando a chave anterior mais. Seja bastante cuidadoso quando selecionar yes, uma vez que esse é um processo destrutivo e que não pode ser revertido.

Digite uma frase de segurança caso queira. Ou basta dar um Enter e deixar em branco.

Agora você tem uma chave pública e uma privada que você poderá utilizar para autenticar. O próximo passo é colocar a chave pública no seu servidor Asterisk.

Rode o comando abaixo para copiar o arquivo .pub para o servidor asterisk.

```
#scp id_rsa.pub root@IP_DO_ASTERISK:~/.ssh
```

# Configurações Asterisk

Após realizar os ajustes necessários no servidor do Zabbix, que será o responsável por executar o Script de ligação, devemos fazer alguns ajustes no servidor do Asterisk para que as ligações saiam sem problemas.

No início deste tutorial, um dos requisitos para o funcionamento desta funcionalidade, foi a instalação do script AGi de ligações utilizando o Google TTS.

Devemos agora validar se a chave pública gerada no servidor do Zabbix, está no local correto no servidor do Asterisk. Basta acessar o diretório padrão:

```
#cd /root/.ssh/
```

```
[root@01ssrv29 .ssh]# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
```

Vamos acrescentar manualmente o conteúdo do seu arquivo id\_rsa.pub ao arquivo ~/.ssh/authorized\_keys na sua máquina remota.

Para mostrar o conteúdo da sua chave id\_rsa.pub, digite isto em seu computador local:

```
$ cat ~/.ssh/id_rsa.pub
```

Você verá o conteúdo da chave, que deverá se parecer com isto:

Output

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQCqql6MzstZYh1TmWwv11q503pISj2ZF19HgH1JLknLLx
```

Agora você pode criar ou modificar o arquivo authorized\_keys dentro desse diretório. Você pode adicionar o conteúdo do seu arquivo id\_rsa.pub ao final do arquivo authorized\_keys, criando-o se necessário, utilizando este comando:

```
echo string_da_chave_pública >> ~/.ssh/authorized_keys
```

No comando acima, substitua a string da chave pública com a saída do comando `cat ~/.ssh/idrsa.pub` que você executou em seu sistema local. Ele deve iniciar com `ssh-rsa AAAA...`.

Finalmente, vamos garantir que o diretório ~/.ssh e o arquivo authorized\_keys tenham as permissões apropriadas configuradas:

```
chmod -R go= ~/.ssh
```

Isto remove recursivamente todas as permissões “group” e “other” para o diretório ~/.ssh/.

Se você estiver utilizando a conta root para configurar chaves para uma conta de usuário, é importante também que o diretório ~/.ssh pertença ao usuário e não ao root:

```
chown -R asterisk:asterisk ~/.ssh
```

Com essa configuração, o servidor do Zabbix irá conseguir copiar o arquivo de ligação .call para o servidor do Asterisk sem a necessidade de autenticação.

Agora devemos criar no crontab, um job para mover os arquivos da pasta temporária para para a pasta Outgoing do Asterisk, para que o mesmo faça a ligação para nós.

Crie um JOB para rodar a cada 1 minuto

```
# Ligações emergenciais Zabbix
*/1 * * * * root /home/programas/scripts/zabbix_call.sh
```

Detalhes do Script (Bem simples)

```
#!/bin/bash

chown asterisk /tmp/ligacao_zabbix.call
mv /tmp/ligacao_zabbix.call /var/spool/asterisk/outgoing/
```

Temos a necessidade de realizar essa movimentação, pois o Asterisk pode ler o arquivo a qualquer momento na pasta Outgoing (Normalmente é instantâneo). Caso utilizemos o comando CP, corre o risco de o Asterisk ler o arquivo quando ele está 50% escrito, o que fará com que a ligação falhe.

O comando MV, move o inode do arquivo, fazendo com que ele apareça por completo na pasta.

Agora que os scripts estão ajustados, devemos criar um contexto no plano de discagem do Asterisk para que a ligação seja feita da maneira que queremos.

Abaixo podemos ver que há um contexto criado com o nome zabbix (nome este que definimos no script no servidor do Zabbix Server)

Ele irá:

- Atender a ligação
- Converter a mensagem de texto para voz, utilizando o Google TTS
- Perguntar se deseja repetir a mensagem ou não.
- Caso digite 1, a mensagem será repetida e depois encerrada.
- Caso digite 2, irá retornar outra mensagem que pode ser customizada.

Podemos customizar o dialplan da maneira que quisermos, o céu é o limite.



```
[zabbix]
;Simple speech recognition
exten => zabbix,1,Answer()
exten => zabbix,n,agi(googletts.agi,${MESSAGE},pt-br)
exten => zabbix,n,Wait(2)
exten => zabbix,n,agi(googletts.agi,"Para repetir a mensagem, digite 1. Para mandar o
cara do zabbix ir para a esquina, digite 2",pt-br)
exten => zabbix,n,WaitExten(5)
exten => zabbix,n,Hangup()

exten => 1,1,agi(googletts.agi,${MESSAGE},pt-br)
exten => 1,n,Hangup()

exten => 2,1,agi(googletts.agi,"Vai você, seu bobão",pt-br)
exten => 2,n,Hangup()
```

# Configuração Front-End

Com a parte de scripts configuraad, devemos agora configurar nosso Front End para que os alertas sejam enviados para os scripts e feitas as ligações.

Em Media Typer, devemos configurar o script de ligações com os parâmetros abaixo.

**Media types**

Media type Options

\* Name

Type

\* Script name

Script parameters

Parameter	Action
<input data-bbox="606 902 1286 931" type="text" value="{ALERT.SENDTO}"/>	<a href="#">Remove</a>
<input data-bbox="606 949 1286 978" type="text" value="{ALERT.MESSAGE}"/>	<a href="#">Remove</a>
<a href="#">Add</a>	

Enabled ☒

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

No User, definimos o número para onde será feita a ligação. (Pode ser que precise adicionar ou não a operadora, depende do seu plano de ligações)

No exemplo abaixo está o número 021 (13) 99999.9999 Utilizamos a operadora Embratel(21)

Media	Type	Send to	When active	Use if severity	Status	Action
Email	Email	test@ingulmar.com.br	1-7,00:00-24:00	N I W A H D	Enabled	Edit Remove
Ligacao Asterisk	Ligacao Asterisk	0211399999999	1-7,00:00-24:00	N I W A H D	Enabled	Edit Remove
telegram.py	telegram.py	60550234				

[Add](#) [Update](#) [Delete](#) [Cancel](#)

**Media**

Type

\* Send to

\* When active

Use if severity

- ☒ Not classified
- ☒ Information
- ☒ Warning
- ☒ Average
- ☒ High
- ☒ Disaster

Enabled ☒

[Update](#) [Cancel](#)

Agora vamos para a criação da Action que irá disparar o envio da ligação.  
Colocamos as condições que foram de sua preferência.

**Actions**

Action Operations Recovery operations Update operations

\* Name Asterisk - Ligar CELL

Conditions

Label	Name	Action
A	Host equals <code>{HOST.NAME}</code>	<a href="#">Remove</a>

New condition

Trigger name contains

[Add](#)

Enabled ☒

\* At least one operation, recovery operation or update operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Em Operations, Configuramos a mensagem que será enviada, passando as variáveis que quisermos, direcionamos para enviar por mensagem, para o usuário X, que configuramos o telefone, utilizando o script de ligações que criamos.

Action Operations Recovery operations Update operations

\* Default operation step duration 1h

Default subject Problem: {EVENT.NAME}

Default message Atenção, alerta Zabbix. Cara, detectei um problema no servidor {HOST.NAME}, ele tem o IP {HOST.CONN}. Ele está me alertando {TRIGGER.NAME}. Da uma olhada por favor?

Pause operations for suppressed problems ☒

Operations

Steps	Details	Start in	Duration	Action
1	Send message to users: {HOST.CONN} (Fabricio Guimaraes) via Ligacao Asterisk	Immediately	60	<a href="#">Edit</a> <a href="#">Remove</a>

Operation details

Steps 1 - 1 (0 - infinitely)

Step duration 60 (0 - use action default)

Operation type Send message

\* At least one user or user group must be selected.

Send to User groups

User group	Action
<a href="#">Add</a>	

Send to Users

User	Action
{HOST.CONN} (Fabricio Guimaraes)	<a href="#">Remove</a>
<a href="#">Add</a>	

Send only to Ligacao Asterisk

Default message ☒

Conditions

Label	Name	Action
<a href="#">New</a>		

[Update](#) [Cancel](#)

\* At least one operation, recovery operation or update operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

## Observações Gerais

- Deverá instalar o script AGI para converter texto para voz utilizando o Google TTS
- Deverá criar o script no servidor do Zabbix
- Deverá criar a chave SSH para a copia de arquivos sem pedir senha
- Deverá aplicar a chave publica no servidor do PABX
- Deverá configurar o Dialplan do Asterisk
- Deverá configurar uma nova Media Type no Front End do Zabbix
- Deverá configurar o telefone em um usuário
- Deverá criar a Action com o alarme

Resultado Final, action executada com sucesso e ligação realizada no Asterisk

Problem	Duration	Ack	Actions	Tags		
Service SNMP Service is unavailable	13m 28s	No	1			
Service SNMP	Time	User/Recipient	Action	Message/Command	Status	Info
Service SNMP	2019-03-13 17:29:48					
Service SNMP	2019-03-13 17:16:22	fabriciog@ecsc.ecorodovias.corp (Fabricio Guimaraes)		Ligacao Asterisk	Sent	
Service SNMP	2019-03-13 17:16:20					
Service SNMP						

**Este Script não deve ser executado em ambiente produtivo sem os devidos testes, não me responsabilizo pela utilização incorreta deste material.**

# Conclusões

Este script foi criado devido a uma necessidade específica na empresa onde trabalho, para comunicar o plantonista caso algum alarme de alta criticidade seja acionado, e não tenhamos tempo de aguardar a abertura de um chamado e que a nossa Central de atendimento N1 faça a primeira análise e posterior acionamento.

Este foi o primeiro modelo do Script, pode ser que seja necessário melhorar para que fique mais genérico, atendendo a todos.

Caso tenha algum problema ou sugestão de melhoria, pode enviar um Telegram para @theguima para conversarmos e melhorar a solução.

Obrigado