



Tutorial de instalação do *Agente Zabbix*

Por: **Aécio Pires**

JOÃO PESSOA-PB

OUT/2010

Aécio dos Santos Pires

<http://aeciopires.com>
aeciopires@gmail.com

Tecnólogo em Redes de Computadores – IFPB, pós-graduando em Segurança da Informação – iDEZ, administrador de sistemas da Dynavideo e membro da comunidade Zabbix Brasil.

Licença de Uso



Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Uso Não-Comercial 2.5 Brasil. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc/2.5/br/> ou envie uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Sumário

1. Introdução.....	4
2. Instalando o Agente Zabbix no Windows.....	5
2.1. No cliente.....	5
2.2. No servidor Zabbix.....	8
3. Instalando o agente Zabbix no Debian e Ubuntu.....	10
4. Instalando o agente Zabbix no OpenSuse.....	13
5. Instalando o agente Zabbix no Fedora e CentOS.....	19
6. Considerações finais.....	23
7. Referências.....	23

1. Introdução

O Agente Zabbix é uma aplicação cliente, que coleta informações de um equipamento, envia ao Servidor Zabbix para análises posteriores e realiza operações de gerenciamento solicitadas Servidor Zabbix . O Agente é capaz de acompanhar ativamente o uso dos recursos e aplicações locais, tais como: discos rígidos, memória, processador, processos, serviços e aplicativos em execução.

A figura 1 mostra o funcionamento do Agente Zabbix.









Figura 1: Comunicação entre o agente e servidor Zabbix.

No tutorial disponível no link http://zabbixbrasil.org/files/Tutorial_de_instalacao_do_Zabbix_1-8-3.pdf, foi mostrado como instalar o Servidor Zabbix no Ubuntu Server. A instalação do Servidor Zabbix não é pré-requisito à instalação do Agente, mas é necessário que ele esteja instalado em algum computador da rede para processar as

informações enviadas pelo Agente. Neste tutorial haverá interação constante entre estes dois componentes durante a instalação do Agente.

Para a elaboração deste tutorial, foram utilizados máquinas virtuais com as configurações mostradas na tabela 1.

Tabela 1: Configurações dos equipamentos utilizados neste tutorial.

VM	Processador	Memória (MB)	HD (GB)	Interface de rede
	Dual Core E2140 @ 1.60GHz	512	8	100 Mbps
	Dual Core E2140 @ 1.60GHz	512	8	100 Mbps
	Dual Core E2140 @ 1.60GHz	512	8	100 Mbps
 XP	Dual Core E2140 @ 1.60GHz	512	8	100 Mbps
 2003	Dual Core E2140 @ 1.60GHz	512	8	100 Mbps
 2008	Dual Core E2140 @ 1.60GHz	512	16	100 Mbps

OBS.: Depois de instalar o agente, acesse os sites mostrados nas Referências deste tutorial para obter mais informações sobre o sistema.

Na página <http://www.zabbix.com/documentation/1.8/complete> podem ser encontradas as informações sobre a instalação e uso do Zabbix.

2. Instalando o Agente Zabbix no Windows

2.1. No cliente

OBS 1.: Os símbolos “\$”, “#” que precederão os comandos ao longo deste tutorial, representam, respectivamente, o prompt de comando do usuário comum e do root (usuário administrador) do sistema operacional. Quando eu me referir a edição de arquivo, o símbolo “#” representará apenas um comentário, sem importância para o funcionamento do serviço em questão.

OBS 2.: Os procedimentos mostrados nesta seção foram testados com sucesso no Windows XP, Windows 7, Windows 2003 Server e Windows Server 2008.

OBS 3.: Na elaboração deste tutorial é assumido que não existe nenhum firewall filtrando o tráfego entre o Agente e Servidor Zabbix.

Até a data de publicação deste tutorial, a versão do Agente Zabbix para a plataforma Windows é a 1.8.3, disponível para as arquiteturas 32 e 64 bits. Faça o download do Agente em:

http://www.zabbix.com/downloads/1.8.3/zabbix_agents_1.8.3.win.zip .

Descompacte o arquivo **zabbix_agents_1.8.3.win.zip**. Serão extraídas duas pastas: uma com o nome **win32**, para a arquitetura 32 bits, e **win64**, para a arquitetura 64 bits.

Acesse a partição **C:** e crie uma pasta com o nome **zabbix**. Acesse a nova pasta e copie para dentro dela os arquivos que estão dentro da pasta win32 ou win64 (escolha de acordo com a arquitetura do Windows que está instalado).

Acesse a partição **C:** e crie o arquivo **zabbix_agentd.conf** com o seguinte conteúdo.

```
Server=IP do Servidor do Zabbix
```

```
Hostname=Nome do computador cliente
```

```
StartAgents=5
```

```
DebugLevel=3
```

```
PidFile=c:\zabbix\zabbix_agentd.pid
```

```
LogFile=c:\zabbix\zabbix_agentd.log
```

```
Timeout=3
```

Se você criar o arquivo com o Notepad ou outro editor de texto simples. Pode ser adicionada automaticamente ao arquivo a extensão **.txt**. Para verificar isso (depois de salvar o arquivo em C:\), acesse o prompt do Windows em **Iniciar > Executar**. Digite **cmd** para acessar o prompt de comandos do Windows. Digite os comandos abaixo para listar o nome real do arquivo.

```
> cd C:\
> dir
```

Se o arquivo estiver com a extensão **.txt** no final do nome, use o comando abaixo para renomeá-lo:

```
> ren zabbix_agentd.conf.txt zabbix_agentd.conf
```

Agora instale o Agente Zabbix como um serviço do Windows executando os comandos abaixo (ainda no prompt do Windows).

```
> cd c:\zabbix
> zabbix_agentd.exe -i
```

Opcionalmente você pode usar a opção -c informando outro caminho para o arquivo de configuração do agente, como mostra o comando abaixo.

```
> cd c:\zabbix  
> zabbix_agentd.exe -i -c c:\zabbix\zabbix_agentd.conf
```

Verifique o status do Agente Zabbix acessando **Iniciar> Painel de Controle> Ferramentas administrativas> Serviços**. Será mostrada uma tela semelhante a figura 2.



Figura 2: Status do agente Zabbix.

Verifique se o Agente está parado. Dê um dois cliques e será mostrada a tela semelhante a figura 3.

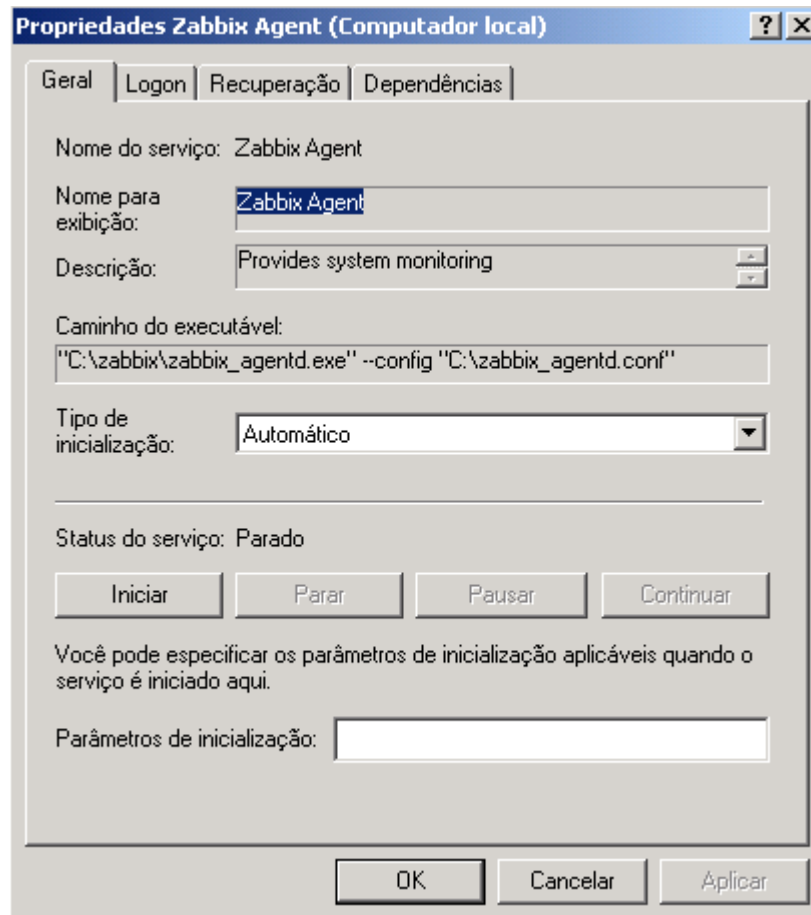


Figura 3: Iniciando o agente Zabbix.

Se estiver parado, clique em **Iniciar** e depois clique em **OK**.

Pronto! o Agente está sendo executado, coletando informações e enviando-as ao Servidor Zabbix.

2.2. No servidor Zabbix

Agora vamos cadastrar este computador na interface web do Zabbix para iniciar o monitoramento.

OBS.: Para realizar este procedimento será necessário que o Servidor Zabbix esteja instalado. Se você ainda não instalou, pare de ler este tutorial e leia antes o **Tutorial de Instalação do Servidor Zabbix no Ubuntu Server** disponível no link: http://zabbixbrasil.org/files/Tutorial_de_instalacao_do_Zabbix_1-8-3.pdf

Acesse a interface web do Zabbix. Clique na aba **Configuração > Hosts** (**Configuration > Hosts**, se estiver em Inglês). Clique no botão **Criar Host** (ou **Create Host**). Será exibida uma tela semelhante a das figuras 4.1 e 4.2.

Monitoramento | Inventário | Relatórios | **Configuração** | Administração

Grupos de Hosts | Templates | **Hosts** | Manutenção | Web | Ações | Telas | Slides | Mapas | Serviços de TI | Autobusca

Histórico: Hosts » Grupos de Hosts » Hosts » Dados personalizados do usuário » Hosts

CONFIGURAÇÃO DE HOSTS

Host

Nome

Grupos

Nos grupos

- Zabbix servers

Outros grupos

- Cliente Windows
- Discovered hosts
- Linux servers
- Templates
- Windows servers

Novo grupo

Nome DNS

Endereço IP

Connectado a

Porta do agente Zabbix

Monitorado por proxy

Status

Usar IPMI

Salvar Cancelar

Figura 4.1: Adicionando o cliente ao servidor Zabbix (parte 1).

[Ajuda](#) | [Suporte](#) | [Imprimir](#) | [Configurações](#) | [Logout](#)

PROCURAR:

Associado aos templates

Não há templates associados

Adicionar Desassociar Desassociar e

Macros

Macro	Valor
<input type="checkbox"/> <input type="text"/>	⇒ <input type="text"/>

Adicionar Remove selecionado(s)

Configurações

Usar inventário

Perfil estendido do host

Usar perfil estendido

Figura 4.2: Adicionando o cliente ao servidor Zabbix (parte 2).

No campo **Nome** (ou **Name**), como mostrado na figura 4.1, digite o nome do computador cliente. No campo **Grupos** (ou **Groups**) selecione **Zabbix Servers** e clique no botão **>>** para remover este grupo.

No campo **Novo Grupo** (ou **New group**) digite, por exemplo, **Clientes Windows XP** para identificar o novo grupo a ser formado apenas por computadores que tem o Windows XP instalado (isto é muito útil para identificar os computadores da rede).

Em **Nome DNS** (ou **DNS name**) digite o nome que identifica este computador no DNS, deixe em branco se não souber.

Em **Endereço IP** (ou **IP address**) digite o IP do computador cliente. Os campos restantes podem ficar com os valores padrão.

Agora edite os campos mostrados na figura 4.2. No campo **Associado aos templates** (ou **Linked templates**) clique no botão **Adicionar** (ou **Add**). Será exibido uma janela suspensa chamada de **pop up** com uma lista de templates contendo vários parâmetros de monitoramento e gerenciamento pré-configurados para um determinado tipo de equipamento. Escolha a opção **Template_Windows** e clique no botão **Selecionar** (ou **Select**).

Pronto! O cliente foi configurado. Clique no botão **Salvar** (ou **Save**).

Perceba que o Servidor Zabbix tentará estabelecer a comunicação com o cliente e mostrará o status da comunicação com o agente e indicará se o host está ou não sendo monitorado.

Nos próximos tutoriais falarei mais sobre a interface web do Zabbix.

3. Instalando o agente Zabbix no Debian e Ubuntu

Até a data de publicação deste tutorial, a versão do Agente Zabbix para a plataforma Linux é a 1.8.3, disponível para as arquiteturas 32 e 64 bits e para os kernels 2.4.X e 2.6.X. Faça o download do Agente em:

<http://www.zabbix.com/download.php>

OBS 1.: Neste tutorial usei o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** disponível em:

http://www.zabbix.com/downloads/1.8.3/zabbix_agents_1.8.3.linux2_6.i386.tar.gz

OBS 2.: Os procedimentos mostrados nesta seção foram testados com sucesso no Debian 5 e Ubuntu 10.04.

Descompacte o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** com o comando abaixo.

```
$ tar xzvf zabbix_agents_1.8.3.linux2_6.i386.tar.gz
```

Serão extraídos dois diretórios contendo os arquivos executáveis do Agente:

```
bin/  
zabbix_get  
zabbix_sender
```

```
sbin/  
zabbix_agent  
zabbix_agentd
```

Copie os arquivos executáveis do diretório **bin/**, extraído do pacote, para o diretório **/usr/local/bin**. Para isso use os comandos abaixo.

```
$ sudo cp bin/zabbix_get /usr/local/bin/  
$ sudo cp bin/zabbix_sender /usr/local/bin/
```

Copie os arquivos executáveis do diretório **sbin/**, extraído do pacote, para o diretório **/usr/local/sbin**. Para isso use os comandos abaixo.

```
$ sudo cp sbin/zabbix_agent /usr/local/sbin/  
$ sudo cp sbin/zabbix_agentd /usr/local/sbin/
```

Crie o diretório **/etc/zabbix/**.

```
$ sudo mkdir /etc/zabbix/
```

Edite o arquivo **/etc/services** e adicione as linhas abaixo, mantendo a ordenação das portas que já estão no arquivo.

```
zabbix_agent 10050/tcp  
zabbix_trap 10051/tcp  
zabbix_agent 10050/udp  
zabbix_trap 10051/udp
```

Crie o arquivo **/etc/zabbix/zabbix_agent.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix  
Timeout=3
```

Crie o arquivo **/etc/zabbix/zabbix_agentd.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix  
Hostname=Nome do computador cliente  
ServerPort=10051  
ListenPort=10050  
StartAgents=5  
RefreshActiveChecks=120  
  
#DisableActive=1  
#EnableRemoteCommands=1  
# Specifies debug level  
# 0 debug is not created  
# 1 critical information  
# 2 error information
```

```
# 3 warnings
# 4 information (default)
# 5 for debugging (produces lots of information)
DebugLevel=4
LogFile=/var/log/zabbix_agentd.log
PidFile=/tmp/zabbix_agentd.pid
```

Crie o usuário zabbix.

```
$ sudo adduser zabbix
```

Crie o arquivo `/var/log/zabbix_agentd.log`.

```
$ sudo touch /var/log/zabbix_agentd.log
$ sudo chown zabbix:zabbix /var/log/zabbix_agentd.log
```

Crie o arquivo `/etc/init.d/zabbix-agentd` e adicione o conteúdo abaixo.

```
#!/bin/sh
#
# Zabbix agent start/stop script.
#
# Written by Alexei Vladishev <alexei.vladishev@zabbix.com>.

NAME=zabbix_agentd
PATH=/bin:/usr/bin:/sbin:/usr/sbin:/home/zabbix/bin
DAEMON=/usr/local/sbin/${NAME}
DESC="Zabbix agent daemon"
PID=/tmp/${NAME}.pid

test -f $DAEMON || exit 0

set -e

case "$1" in
  start)
    echo "Starting $DESC: $NAME"
    start-stop-daemon --oknodo --start --pidfile $PID \
      --exec $DAEMON
    ;;
  stop)
    echo "Stopping $DESC: $NAME"
    start-stop-daemon --oknodo --stop --pidfile $PID \
      --exec $DAEMON
    ;;
  restart|force-reload)
    #
    # If the "reload" option is implemented, move the "force-reload"
    # option to the "reload" entry above. If not, "force-reload" is
    # just the same as "restart".

```

```

#
#   echo -n "Restarting $DESC: zabbix_agent"
#   $0 stop
#   $0 start
#   start-stop-daemon --stop --quiet --pidfile \
#       /tmp/$NAME.pid --user zabbix --exec $DAEMON
#   sleep 1
#   start-stop-daemon --start --quiet --pidfile \
#       /tmp/$NAME.pid --user zabbix --exec $DAEMON
#   echo "$NAME."
#   ;;
*)
N=/etc/init.d/$NAME
# echo "Usage: $N {start|stop|restart|force-reload}" >&2
echo "Usage: $N {start|stop|restart|force-reload}" >&2
exit 1
#   ;;
esac

exit 0

```

Torne o arquivo executável com o comando abaixo.

```
$ sudo chmod +x /etc/init.d/zabbix-agentd
```

Habilite o script para ser executado no boot do sistema operacional.

```
$ sudo update-rc.d -f zabbix-agentd defaults
```

Inicie o Agente com o comando abaixo:

```
$ sudo /etc/init.d/zabbix-agentd start
```

Pronto! o Agente está sendo executado, coletando informações e enviando-as ao Servidor Zabbix. Agora siga os procedimentos mostrados na **seção 2.2** deste tutorial para cadastrar o cliente no Servidor Zabbix (adicionando o template Linux ao invés do template Windows).

4. Instalando o agente Zabbix no OpenSuse

Faça o download do Agente em: <http://www.zabbix.com/download.php>

OBS 1.: Neste tutorial usei o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** disponível em: http://www.zabbix.com/downloads/1.8.3/zabbix_agents_1.8.3.linux2_6.i386.tar.gz

OBS 2.: Os procedimentos mostrados nesta seção foram testados com sucesso no OpenSuse 11.3.

Descompacte o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** com o comando abaixo.

```
$ tar xzvf zabbix_agents_1.8.3.linux2_6.i386.tar.gz
```

Serão extraídos dois diretórios contendo os arquivos executáveis do Agente:

```
bin/  
zabbix_get  
zabbix_sender  
  
sbin/  
zabbix_agent  
zabbix_agentd
```

Copie os arquivos executáveis do diretório **bin/**, extraído do pacote, para o diretório **/usr/local/bin**. Para isso use os comandos abaixo.

```
$ sudo cp bin/zabbix_get /usr/local/bin/  
$ sudo cp bin/zabbix_sender /usr/local/bin/
```

Copie os arquivos executáveis do diretório **sbin/**, extraído do pacote, para o diretório **/usr/local/sbin**. Para isso use os comandos abaixo.

```
$ sudo cp sbin/zabbix_agent /usr/local/sbin/  
$ sudo cp sbin/zabbix_agentd /usr/local/sbin/
```

Crie o diretório **/etc/zabbix/**.

```
$ sudo mkdir /etc/zabbix/
```

Edite o arquivo **/etc/services** e adicione as linhas abaixo, mantendo a ordenação das portas que já estão no arquivo.

```
zabbix_agent 10050/tcp  
zabbix_trap 10051/tcp  
zabbix_agent 10050/udp  
zabbix_trap 10051/udp
```

Crie o arquivo **/etc/zabbix/zabbix_agent.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix  
Timeout=3
```

Crie o arquivo **/etc/zabbix/zabbix_agentd.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix
```

```
Hostname=Nome do computador cliente
ServerPort=10051
ListenPort=10050
StartAgents=5
RefreshActiveChecks=120

#DisableActive=1
#EnableRemoteCommands=1
# Specifies debug level
# 0 debug is not created
# 1 critical information
# 2 error information
# 3 warnings
# 4 information (default)
# 5 for debugging (produces lots of information)
DebugLevel=4
LogFile=/var/log/zabbix_agentd.log
PidFile=/tmp/zabbix_agentd.pid
```

Crie o usuário zabbix.

```
$ sudo useradd zabbix
$ sudo groupadd zabbix
```

Crie o arquivo `/var/log/zabbix_agentd.log`.

```
$ sudo touch /var/log/zabbix_agentd.log
$ sudo chown zabbix:zabbix /var/log/zabbix_agentd.log
```

Crie o arquivo `/etc/init.d/zabbix-agentd` e adicione o conteúdo abaixo.

```
#!/bin/sh
#
# /etc/init.d/zabbix-agentd
# and its symbolic link
# /(usr/)sbin/rczabbix-agentd
#
#### BEGIN INIT INFO
# Provides:          zabbix-agentd
# Required-Start:    $network $remote_fs $syslog
# Should-Start:      $time
# Required-Stop:     $syslog $remote_fs
# Should-Stop:       $time ypbind smtp
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Short-Description: ZABBIX agentd
# Description:       Start ZABBIX agentd
#### END INIT INFO

# Check for missing binaries (stale symlinks should not happen)
```

```

# Note: Special treatment of stop for LSB conformance
ZABBIX_AGENTD_BIN=/usr/local/sbin/zabbix_agentd
test -x $ZABBIX_AGENTD_BIN || { echo "$ZABBIX_AGENTD_BIN not installed";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 5; fi; }

# Check for existence of needed config file and read it
ZABBIX_AGENTD_CONFIG=/etc/zabbix/zabbix_agentd.conf
test -r $ZABBIX_AGENTD_CONFIG || { echo "$ZABBIX_AGENTD_CONFIG not
existing";
    if [ "$1" = "stop" ]; then exit 0;
    else exit 6; fi; }

# Get pid file name from config
PidFile=$(grep ^PidFile $ZABBIX_AGENTD_CONFIG|cut -d "=" -f 2)

# Source LSB init functions
# providing start_daemon, killproc, pidofproc,
# log_success_msg, log_failure_msg and log_warning_msg.
# This is currently not used by UnitedLinux based distributions and
# not needed for init scripts for UnitedLinux only. If it is used,
# the functions from rc.status should not be sourced or used.
#. /lib/lsb/init-functions

# Shell functions sourced from /etc/rc.status:
# rc_check      check and set local and overall rc status
# rc_status     check and set local and overall rc status
# rc_status -v  be verbose in local rc status and clear it afterwards
# rc_status -v -r ditto and clear both the local and overall rc status
# rc_status -s  display "skipped" and exit with status 3
# rc_status -u  display "unused" and exit with status 3
# rc_failed     set local and overall rc status to failed
# rc_failed <num> set local and overall rc status to <num>
# rc_reset      clear both the local and overall rc status
# rc_exit       exit appropriate to overall rc status
# rc_active     checks whether a service is activated by symlinks
. /etc/rc.status

# Reset status of this service
rc_reset

# Return values acc. to LSB for all commands but status:
# 0    - success
# 1    - generic or unspecified error
# 2    - invalid or excess argument(s)
# 3    - unimplemented feature (e.g. "reload")
# 4    - user had insufficient privileges
# 5    - program is not installed
# 6    - program is not configured
# 7    - program is not running

```



```

# 8--199 - reserved (8--99 LSB, 100--149 distrib, 150--199 appl)
#
# Note that starting an already running service, stopping
# or restarting a not-running service as well as the restart
# with force-reload (in case signaling is not supported) are
# considered a success.

case "$1" in
  start)
    echo -n "Starting zabbix agentd "
    ## Start daemon with startproc(8). If this fails
    ## the return value is set appropriately by startproc.
    /sbin/startproc -p $PidFile $ZABBIX_AGENTD_BIN

    # Remember status and be verbose
    rc_status -v
    ;;
  stop)
    echo -n "Shutting down zabbix agentd "
    ## Stop daemon with killproc(8) and if this fails
    ## killproc sets the return value according to LSB.

    /sbin/killproc -p $PidFile -TERM $ZABBIX_AGENTD_BIN

    # Remember status and be verbose
    rc_status -v
    ;;
  try-restart|condrestart)
    ## Do a restart only if the service was active before.
    ## Note: try-restart is now part of LSB (as of 1.9).
    ## RH has a similar command named condrestart.
    if test "$1" = "condrestart"; then
      echo "${attn} Use try-restart ${done}(LSB){attn} rather than
condrestart ${warn}(RH){norm}"
    fi
    $0 status
    if test $? = 0; then
      $0 restart
    else
      rc_reset      # Not running is not a failure.
    fi
    # Remember status and be quiet
    rc_status
    ;;
  restart)
    ## Stop the service and regardless of whether it was
    ## running or not, start it again.
    $0 stop
    $0 start

```

```

# Remember status and be quiet
rc_status
;;
force-reload)
## Signal the daemon to reload its config. Most daemons
## do this on signal 1 (SIGHUP).
## If it does not support it, restart the service if it
## is running.

echo -n "Reload service zabbix agentd "
## if it supports it:
/sbin/killproc -p $PidFile -HUP $ZABBIX_AGENTD_BIN
#touch /var/run/zabbix/zabbix-agentd.pid
rc_status -v

## Otherwise:
##$0 try-restart
#rc_status
;;
reload)
## Like force-reload, but if daemon does not support
## signaling, do nothing (!)

# If it supports signaling:
echo -n "Reload service zabbix agentd "
/sbin/killproc -p $PidFile -HUP $ZABBIX_AGENTD_BIN
#touch /var/run/zabbix/zabbix-agentd.pid
rc_status -v

## Otherwise if it does not support reload:
#rc_failed 3
#rc_status -v
;;
status)
echo -n "Checking for service zabbix agentd "
## Check status with checkproc(8), if process is running
## checkproc will return with exit status 0.

# Return value is slightly different for the status command:
# 0 - service up and running
# 1 - service dead, but /var/run/ pid file exists
# 2 - service dead, but /var/lock/ lock file exists
# 3 - service not running (unused)
# 4 - service status unknown :-(
# 5--199 reserved (5--99 LSB, 100--149 distro, 150--199 appl.)

# NOTE: checkproc returns LSB compliant status values.
/sbin/checkproc -p $PidFile $ZABBIX_AGENTD_BIN
# NOTE: rc_status knows that we called this init script with
# "status" option and adapts its messages accordingly.

```

```

rc_status -v
;;
probe)
## Optional: Probe for the necessity of a reload, print out the
## argument to this init script which is required for a reload.
## Note: probe is not (yet) part of LSB (as of 1.9)

test $ZABBIX_AGENTD_CONFIG -nt $PidFile && echo reload
;;
*)
echo "Usage: $0 {start|stop|status|try-restart|restart|force-reload|reload|
probe}"
exit 1
;;
esac
rc_exit

```

Torne o arquivo executável com o comando abaixo.

```
$ sudo chmod +x /etc/init.d/zabbix-agentd
```

Habilite o script para ser executado no boot do sistema operacional.

```
$ sudo chkconfig zabbix-agentd on
```

Inicie o Agente com o comando abaixo:

```
$ sudo /etc/init.d/zabbix-agentd start
```

Pronto! o Agente está sendo executado, coletando informações e enviando-as ao Servidor Zabbix. Agora siga os procedimentos mostrados na **seção 2.2** deste tutorial para cadastrar o cliente no Servidor Zabbix (adicionando o template Linux ao invés do template Windows).

5. Instalando o agente Zabbix no Fedora e CentOS

Faça o download do Agente em: <http://www.zabbix.com/download.php>

OBS 1.: Neste tutorial usei o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** disponível em:

http://www.zabbix.com/downloads/1.8.3/zabbix_agents_1.8.3.linux2_6.i386.tar.gz

OBS 2.: Os procedimentos mostrados nesta seção foram testados com sucesso no Fedora 13 e CentOS 5.4.

Descompacte o arquivo **zabbix_agents_1.8.3.linux2_6.i386.tar.gz** com o comando abaixo.

```
$ tar xzvf zabbix_agents_1.8.3.linux2_6.i386.tar.gz
```

Serão extraídos dois diretórios contendo os arquivos executáveis do Agente:

```
bin/  
zabbix_get  
zabbix_sender  
  
sbin/  
zabbix_agent  
zabbix_agentd
```

Copie os arquivos executáveis do diretório **bin/**, extraído do pacote, para o diretório **/usr/local/bin**. Para isso use os comandos abaixo.

```
$ sudo cp bin/zabbix_get /usr/local/bin/  
$ sudo cp bin/zabbix_sender /usr/local/bin/
```

Copie os arquivos executáveis do diretório **sbin/**, extraído do pacote, para o diretório **/usr/local/sbin**. Para isso use os comandos abaixo.

```
$ sudo cp sbin/zabbix_agent /usr/local/sbin/  
$ sudo cp sbin/zabbix_agentd /usr/local/sbin/
```

Crie o diretório **/etc/zabbix/**.

```
$ sudo mkdir /etc/zabbix/
```

Edite o arquivo **/etc/services** e adicione as linhas abaixo, mantendo a ordenação das portas que já estão no arquivo.

```
zabbix_agent 10050/tcp  
zabbix_trap 10051/tcp  
zabbix_agent 10050/udp  
zabbix_trap 10051/udp
```

Crie o arquivo **/etc/zabbix/zabbix_agent.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix  
Timeout=3
```

Crie o arquivo **/etc/zabbix/zabbix_agentd.conf** e informe o IP do servidor Zabbix no campo **Server**, mostrado abaixo.

```
Server=172.16.14.133 #Coloque o IP do servidor Zabbix  
Hostname=Nome do computador cliente  
ServerPort=10051  
ListenPort=10050  
StartAgents=5
```

```
RefreshActiveChecks=120

#DisableActive=1
#EnableRemoteCommands=1
# Specifies debug level
# 0 debug is not created
# 1 critical information
# 2 error information
# 3 warnings
# 4 information (default)
# 5 for debugging (produces lots of information)
DebugLevel=4
LogFile=/var/log/zabbix_agentd.log
PidFile=/tmp/zabbix_agentd.pid
```

Crie o usuário zabbix.

```
$ sudo useradd zabbix
$ sudo groupadd zabbix
```

Crie o arquivo **/var/log/zabbix_agentd.log**.

```
$ sudo touch /var/log/zabbix_agentd.log
$ sudo chown zabbix:zabbix /var/log/zabbix_agentd.log
```

Crie o arquivo **/etc/init.d/zabbix-agentd** e adicione o conteúdo abaixo.

```
#!/bin/sh
#
# chkconfig: - 85 15
# description: zabbix agent daemon
#

#### BEGIN INIT INFO
# Provides: zabbix-agent
# Required-Start: $local_fs $network
# Required-Stop: $local_fs $network
# Default-Start:
# Default-Stop: 0 1 2 3 4 5 6
# Short-Description: start and stop zabbix agent
# Description: Zabbix Agent
#### END INIT INFO

# zabbix details
ZABBIX_AGENTD=/usr/local/sbin/zabbix_agentd
CONF=/etc/zabbix/zabbix_agentd.conf
PIDFILE=/var/tmp/zabbix_agentd.pid

# Source function library.
./etc/rc.d/init.d/functions
```

```

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -x $ZABBIX_AGENTD ] || exit 5
[ -e $CONF ] || exit 6
RETVAL=0
case "$1" in
start)
echo -n "Starting zabbix agent: "
daemon $ZABBIX_AGENTD -c $CONF
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/zabbix-agent
;;
stop)
echo -n "Shutting down zabbix agent: "
killproc zabbix_agentd
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/zabbix-agent
;;
restart|reload)
$0 stop
$0 start
RETVAL=$?
;;
status)
status zabbix_agentd
RETVAL=$?
;;
*)
echo "Usage: $0 {start|stop|restart|reload|status}"
exit 1
;;
esac
exit $RETVAL

```

Torne o arquivo executável com o comando abaixo.

```
$ sudo chmod +x /etc/init.d/zabbix-agentd
```

Habilite o script para ser executado no boot do sistema operacional.

```
$ sudo chkconfig zabbix-agentd on
```

Inicie o Agente com o comando abaixo:

```
$ sudo /etc/init.d/zabbix-agentd start
```

Pronto! o Agente está sendo executado, coletando informações e enviando-as ao Servidor Zabbix. Agora siga os procedimentos mostrados na **seção 2.2** deste tutorial para cadastrar o cliente no Servidor Zabbix (adicionando o template Linux ao invés do template Windows).

6. Considerações finais

Neste tutorial foi mostrada a instalação do agente Zabbix no Ubuntu, Debian, OpenSuse, Fedora, CentOS, Windows 2003 Server, Windows 2008 Server, Windows 7 e Windows XP.

Os próximos tutoriais mostrarão a instalação do componente Zabbix Proxy e o uso da interface Web do Zabbix para monitorar alguns equipamentos.

7. Referências

DEO, André. **Passo a Passo: Instalando o Zabbix no CentOS 5**. Disponível em:

<http://andredeo.blogspot.com/2008/10/passo-passo-instalando-o-zabbix-no.html>

Acessado em: 05 de outubro de 2010.

DEO, André. **Instalação do Agente Zabbix em Ambiente Linux**. Disponível em:

<http://zabbixbrasil.org/wiki/tiki-index.php?page=Instalacao+de+Agente+Zabbix+em+ambiente+Linux>

Acessado em: 05 de outubro de 2010.

Lista da Comunidade Zabbix Brasil. <http://br.groups.yahoo.com/group/zabbix-brasil> Acessado em: 05 de outubro de 2010.

Manual do Zabbix: Zabbix Agent. Disponível em:

http://www.zabbix.com/documentation/1.8/manual/installation/installation_from_source#zabbix_agent Acessado em: 05 de outubro de 2010.

Zabbix Brasil. <http://www.zabbixbrasil.org> Acessado em: 05 de outubro de 2010.