



*Solutis*  
t e c n o l o g i a s

# Monitorando ativos com Zabbix

Rafael Gomes  
rafael.gomes@solutis.com.br

Abril 2011

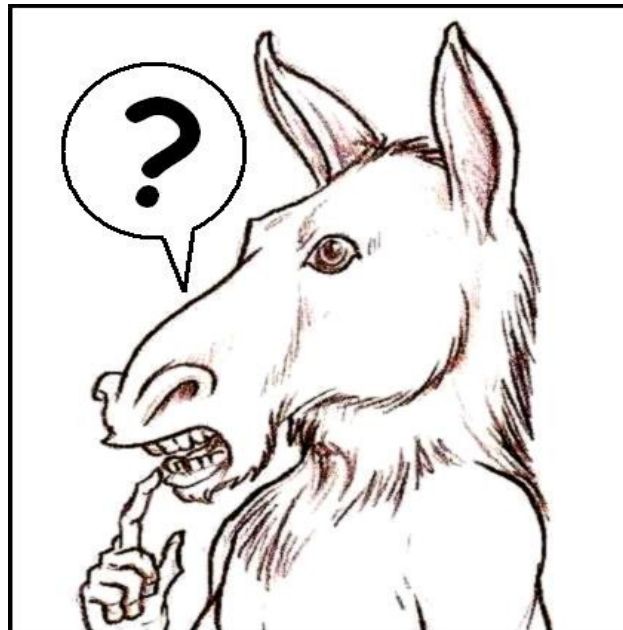
# Motivações

- Agir proativamente
- Detectar e corrigir falhas antes que usuários percebam
- Indicar antecipadamente a necessidade de troca de equipamentos
- Analisar qualidade da prestação de serviço
- Solução de baixo custo
- Solução com interface amigável
- Solução escalável

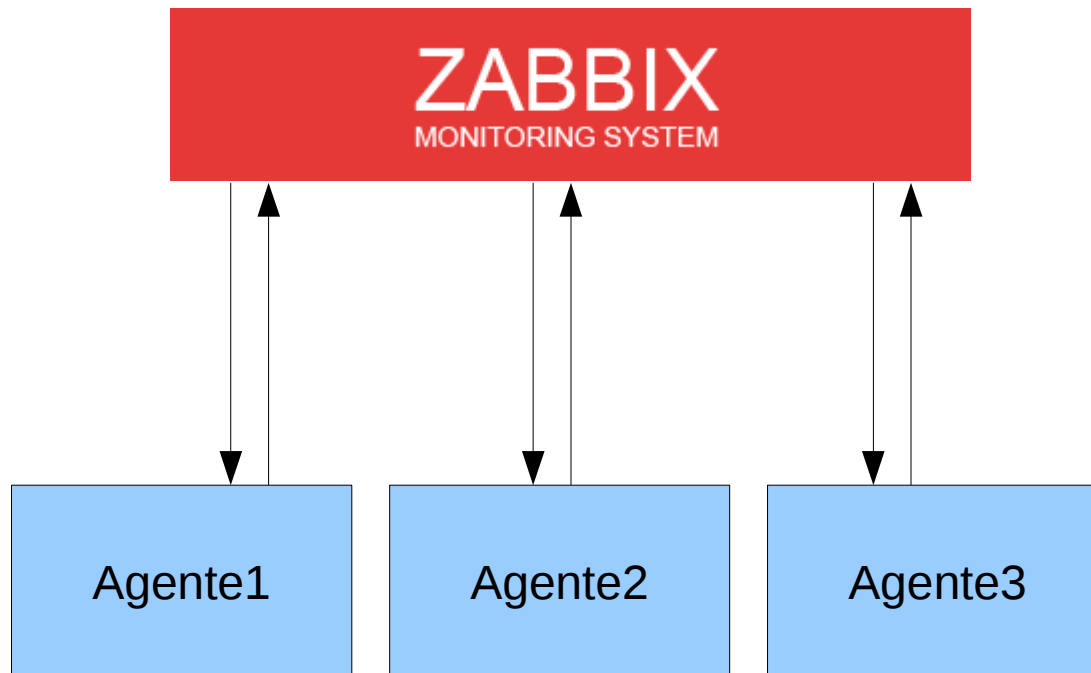
# Zabbix

- Criado em 1998, por Alexei Vladishev
- Licenciado em código aberto
- Backend escrito em C++ e Interface em PHP
- Suporte a SGDB (MySQL, PostgreSQL, SQLite, Oracle or IBM DB2)
- Agentes (AIX, HP-UX, FreeBSD, Gnu/Linux, Windows e outros)
- Monitoria sem agentes
- Suporte a IPV6 e IPV4
- Escalável com utilização de proxys

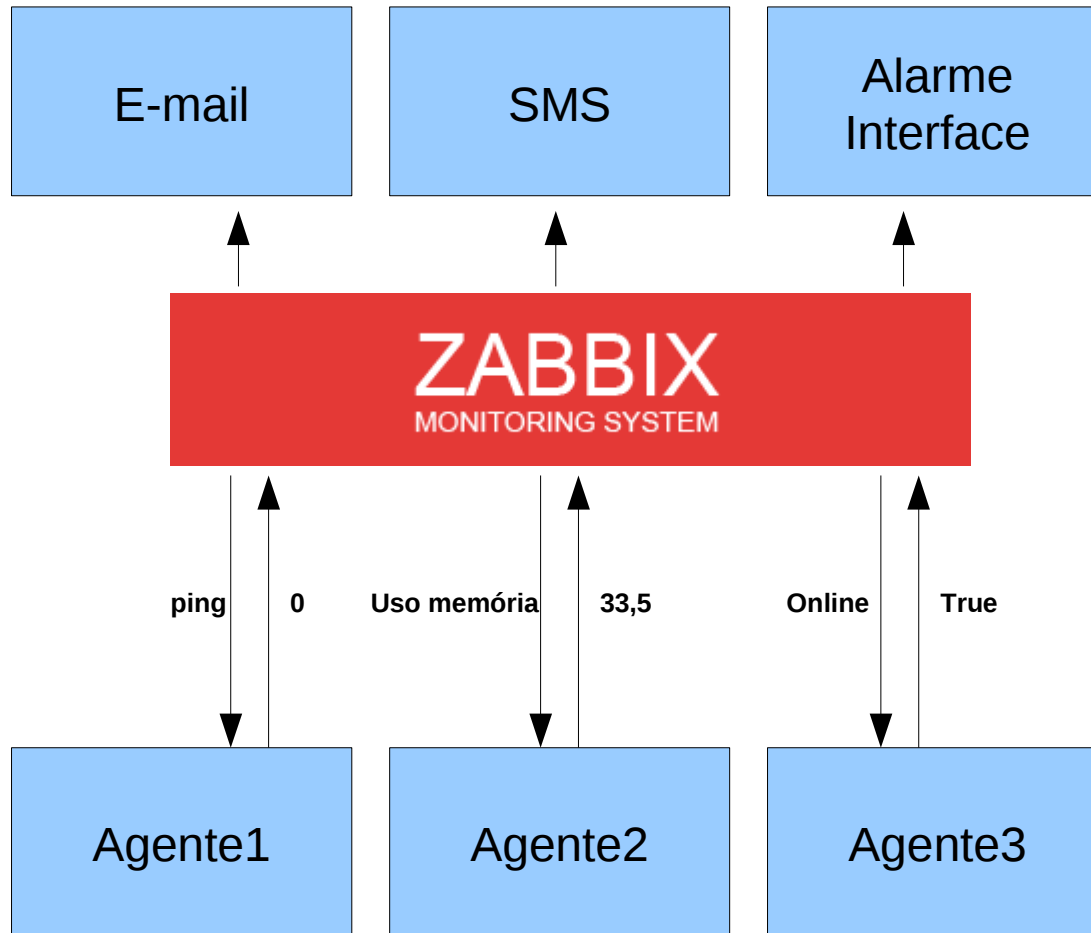
**O Servidor do Zabbix é “burro”!!!**



# Como funciona



# Como funciona



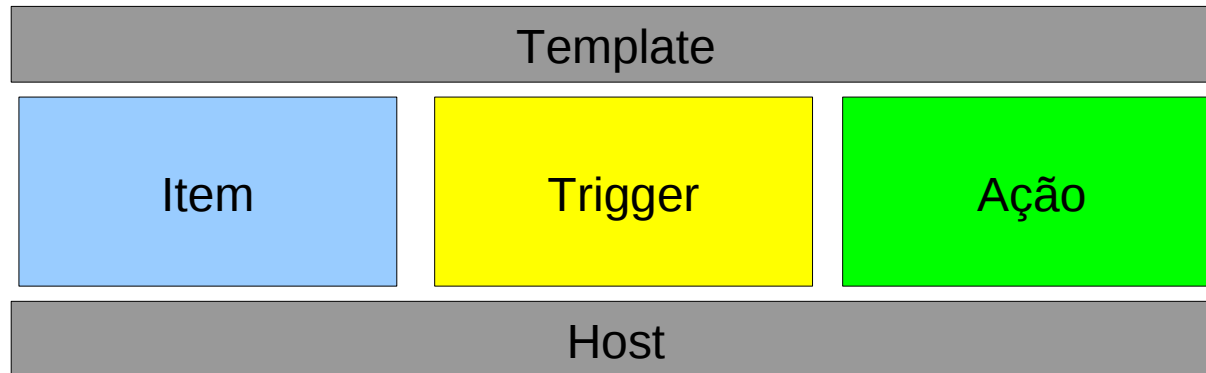


## A inteligência está no agente





# Servidor Zabbix



# Servidor Zabbix

Host

- É a configuração do ativo a ser monitorado
- Contém os seguintes parâmetros:
  - Endereço ip
  - Nome da máquina
  - Template associado
  - Grupo associado
  - Outros...

# Servidor Zabbix

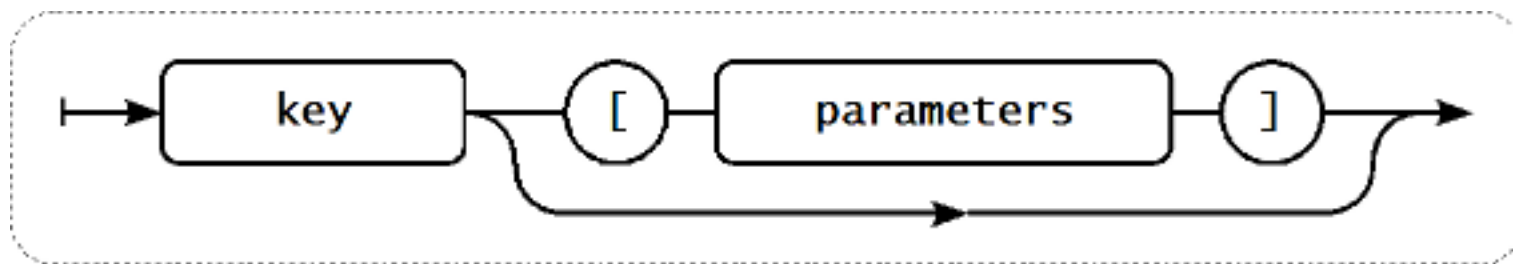
Item

- É o que será monitorado
- Contém os seguintes parâmetros:
  - Chave de monitoramento
  - Frequência da monitoria
  - Tipo do retorno (string, integer ou float)
  - Método de armazenagem (Normal, Delta e Delta/s)
  - Outros...

# Servidor Zabbix

Item

- Chave de monitoramento é a solicitação da informação ao agente
- Formato da chave de monitoramento:



- São permitidos os seguintes caracteres 0-9a-zA-Z\_-.
- Ex. Que horas são? 09:40!

# Servidor Zabbix

Item

- Exemplos reais de chaves de monitoramento:
  - agent.ping
  - system.cpu.load[<cpu>,<mode>]
  - vfs.fs.size[fs,<mode>]
  - proc.num[<name>,<user>,<state>,<cmdline>]

## Trigger

- Condição de alerta com base no que foi monitorado (Item)
- Pode assumir o status de Ok, Problem e Unknown
- A condição é criada com base em uma expressão matemática:
  - $\text{valor\_de\_retorno} = 0$
  - $\text{valor\_de\_retorno} > 2$
- Retorno verdadeiro = Problema
- Sem retorno = Unknown



## Trigger

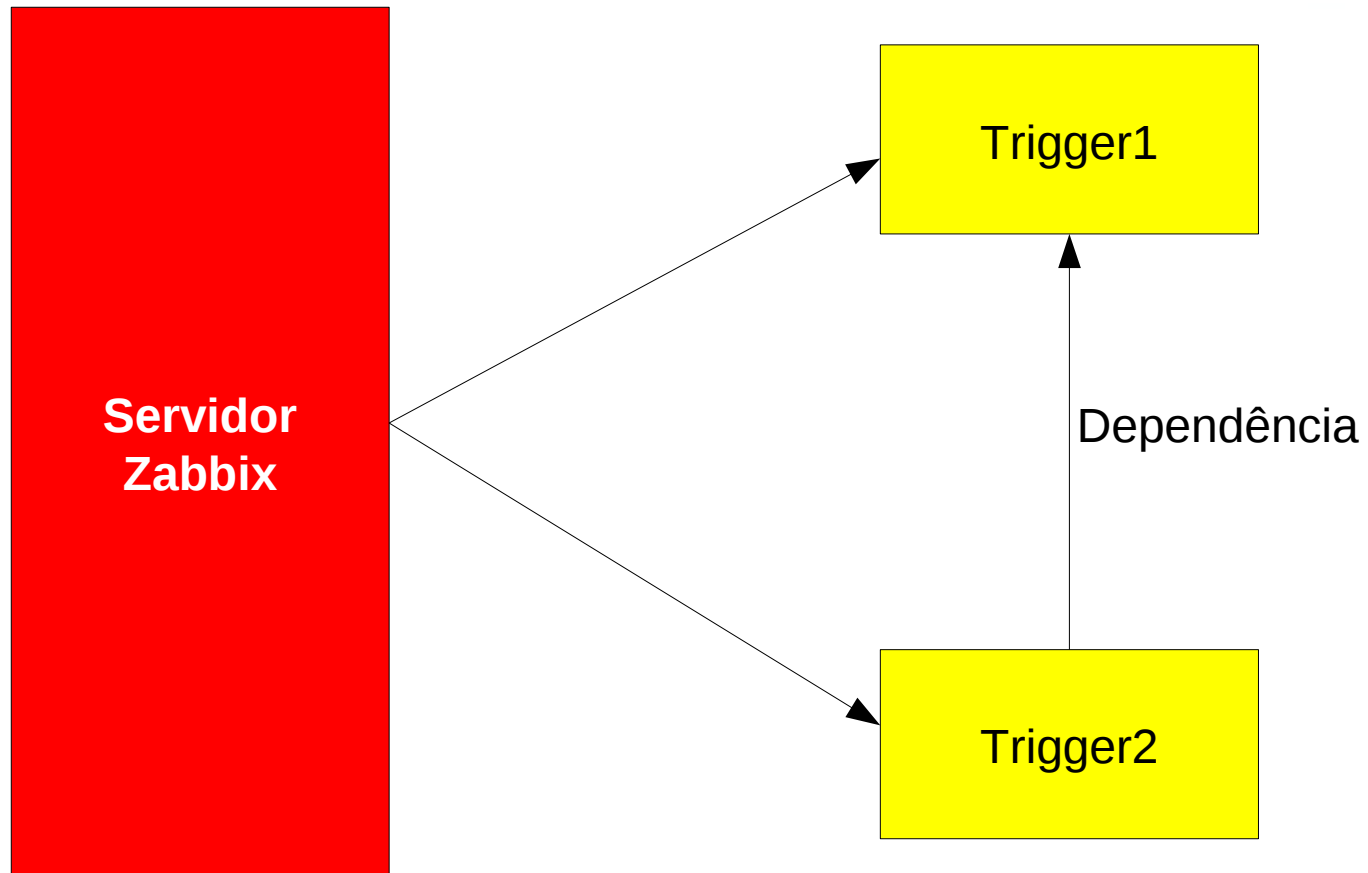
- Segue abaixo o modelo para a expressão real:
  - {<server>:<key>.<function>(<argument>)}<operator><const>
- Alguns exemplos de função:
  - avg[time] - Média de tempo
  - last[time] - Ultimo valor
  - max[time] - Valor máximo
- Ex real: {server1:agent.ping.avg(3)}>0

# Servidor Zabbix

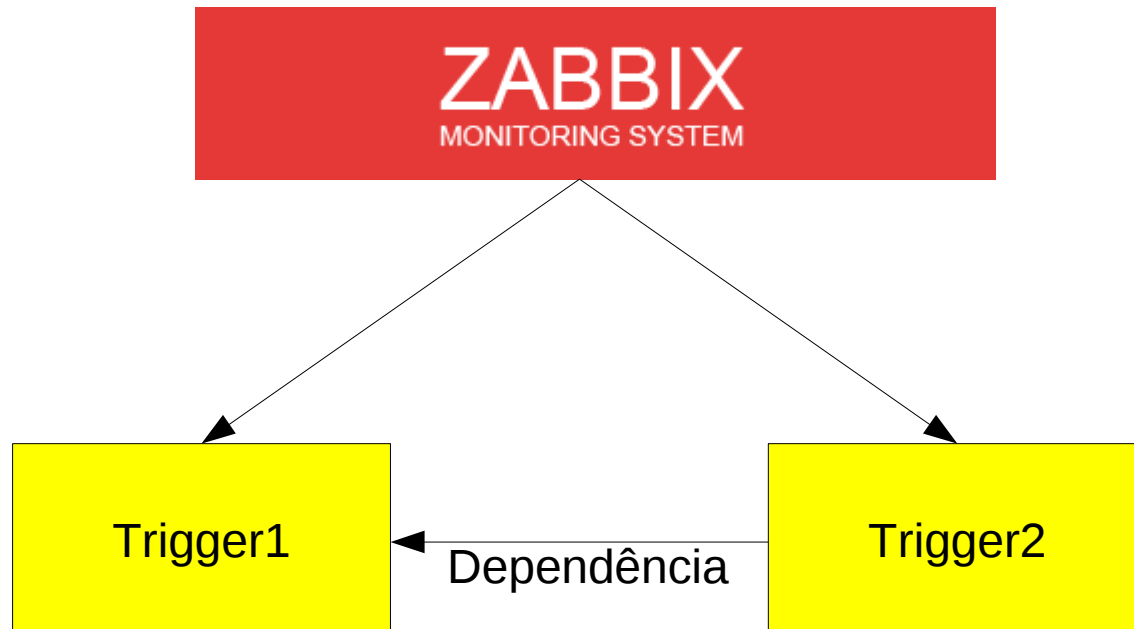
Trigger

- Possibilidade de criar dependências entre triggers distintas
- Caso a trigger que possuir dependentes cair, a dependente não será monitorada até o retorno da dependência

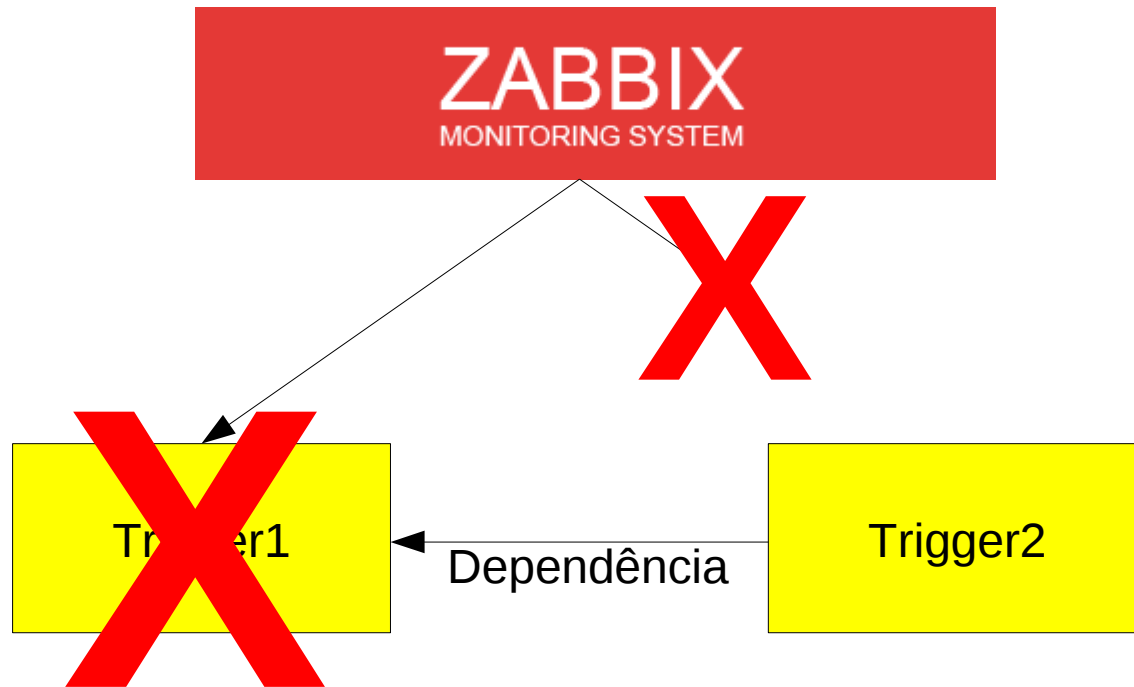
# Dependência



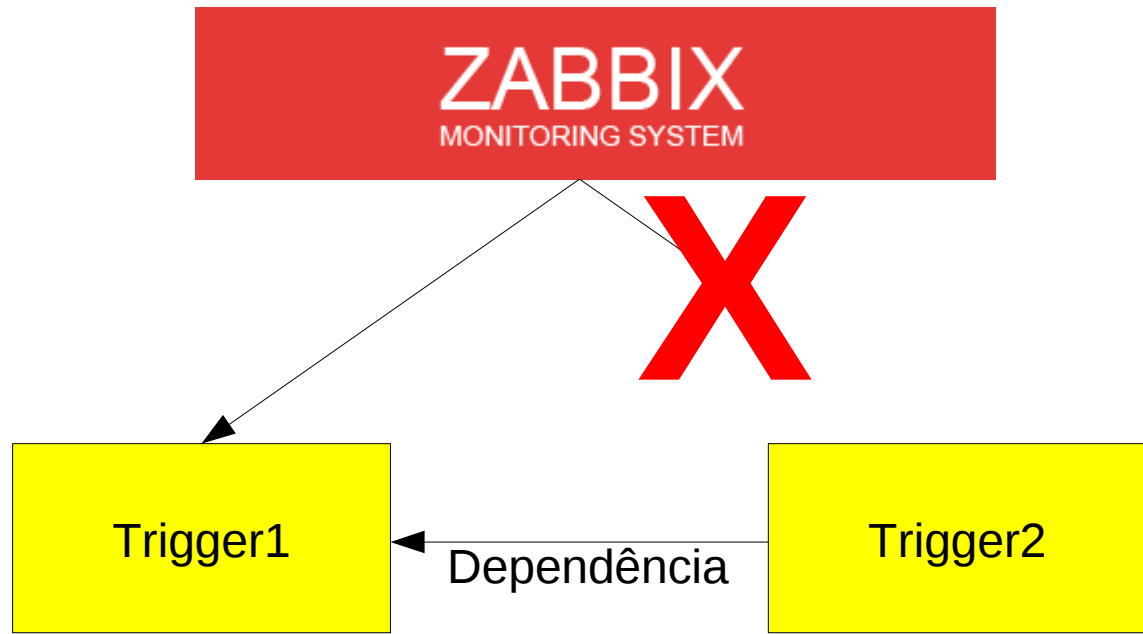
# Dependência



# Dependência

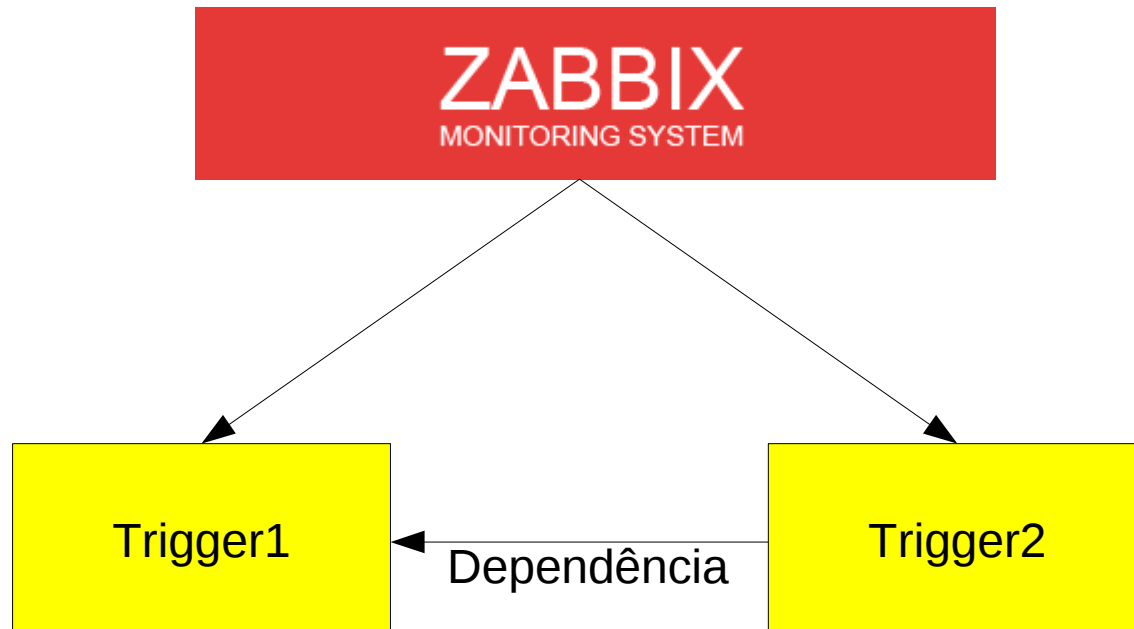


# Dependência





# Dependência



# Servidor Zabbix

## Action

- São as ações executadas com bases nos eventos
- Os eventos podem ser
  - Triggers acionadas
  - Ativos descobertos automaticamente
  - Eventos gerados pelos clientes (Active agent)

# Servidor Zabbix

## Action

- As ações podem ser diversas, mas geralmente são:
  - Envio de e-mail para grupo responsável
  - Envio de SMS para grupo responsável
  - Chat no Jabber para grupo responsável
  - Intervenção no ativo (Ex. reboot remoto)

**SEMPRE** utilize template

# Servidor Zabbix

## Template

- Configuração padrão de itens, triggers e ações
- Configuração automática ao associar o host a um template
- Agilidade na adição de novos hosts

**TUDO** que possa ser obtido via console/scripts ou afins é possível monitorar no Zabbix



# Monitorando tudo

- Possibilidade de configurar QUALQUER checagem no agente
- Modelo da sintaxe:
  - UserParameter=key,command
- Exemplos:
  - UserParameter=check.forward,cat /proc/sys/net/ipv4/ip\_forward
  - UserParameter=mysql.ping,mysqladmin -uroot ping|grep alive|wc -l

# Ambiente real

**Solutis**  
tecnologías



# Perguntas ?



# Fontes

- <http://www.zabbix.com/documentation/1.8/manual>



**Rafael Gomes**  
**[rafael.gomes@solutis.com.br](mailto:rafael.gomes@solutis.com.br)**