

ZABBIX 使用手册

1. Zabbix 简介.....	3
1.1 Zabbix 功能.....	4
1.2 优劣势.....	5
2. 安装部署.....	5
2.1 服务端安装 lamp 环境。.....	5
2.2 服务端配置 lamp 使用环境.....	5
2.3 服务端 server 的安装过程.....	6
2.3.1 安装 zabbix 服务端.....	6
2.3.2 添加 zabbix 到系统服务文件.....	6
2.3.3 导入 zabbix 数据库（此处采用 mysql 数据库）.....	7
2.3.4 拷贝 service 启动脚本.....	7
2.3.5 配置 zabbix_server.conf 服务端文件.....	7
2.3.7 拷贝网页文件到 apache 目录.....	8
2.3.8 设置 zabbix 开机启动.....	9
2.4 通过 web 页面配置 zabbix.....	9
2.5 客户端 agentd 的安装过程.....	13
2.5.1 linux 服务器的安装.....	13
2.5.2 windows 服务器的安装.....	14
3. Zabbix 的配置使用.....	15
3.1 添加 Hosts.....	15
3.2 添加 Items.....	18
3.3 添加 Triggers.....	20
3.4 添加 Actions.....	20
3.5 添加 Medias.....	21
3.6 添加 Users.....	22
3.7 添加 WEB Monitorings.....	24
3.8 添加 Graphs.....	27
3.9 添加 Screens.....	29
3.10 添加 Maps.....	30
3.11 添加 MySQL 监控.....	32
3.12 添加 SNMP 监控.....	33
3.13 添加自定义监控.....	33
3.14 添加 Templates.....	35
3.15 添加 Reports（定制报表）.....	37
3.16 添加 Macros.....	38
3.17 添加自动发现设备.....	39
3.18 添加 Inventory.....	39
3.19 Export/Import XML.....	40
3.20 Maintenance（维护时间）.....	40
3.21 Proxy 的使用.....	42
4. Zabbix 的高级使用-之自动化功能.....	44
4.1 自动发现添加主机.....	44
4.2 通过自动发现实现动态监控.....	47
5. 批量更新参考文档.....	54

文档信息

文档名: *zabbix* 使用手册

初创时间: 2012-12-07

当前版本: v1.2

作者: itnihao

Mail: itnihao@qq.com

修订信息

时间	更新内容
2013-01-23	增加自动发现
2013-02-08	增加部分文档说明
2013-02-11	修正部分错误

本文档涉及内容, *zabbix* 的安装配置, *zabbix* 的邮件报警, *zabbix* 自定义脚本
由于 *zabbix* 功能很丰富, 许多功能需要逐步研究整理, 后续本文档将继续完善。

1. Zabbix 简介

Zabbix 是一个企业级的开源分布式监控解决方案, 由一个国外的团队持续维护更新, 软件可以自由下载使用, 运作团队靠提供收费的技术支持赢利。

官方网站: <http://www.zabbix.com>

Zabbix 2.0 官方文档: <http://www.zabbix.com/documentation/2.0/manual>

Zabbix 通过 C/S 模式采集数据, 通过 B/S 模式在 web 端展示和配置。

被监控端: 主机通过安装 *agent* 方式采集数据, 网络设备通过 *SNMP* 方式采集数据

Server 端: 通过收集 *SNMP* 和 *agent* 发送的数据, 写入数据库 (*MySQL*, *ORACLE* 等), 再通过 *php+apache* 在 web 前端展示。

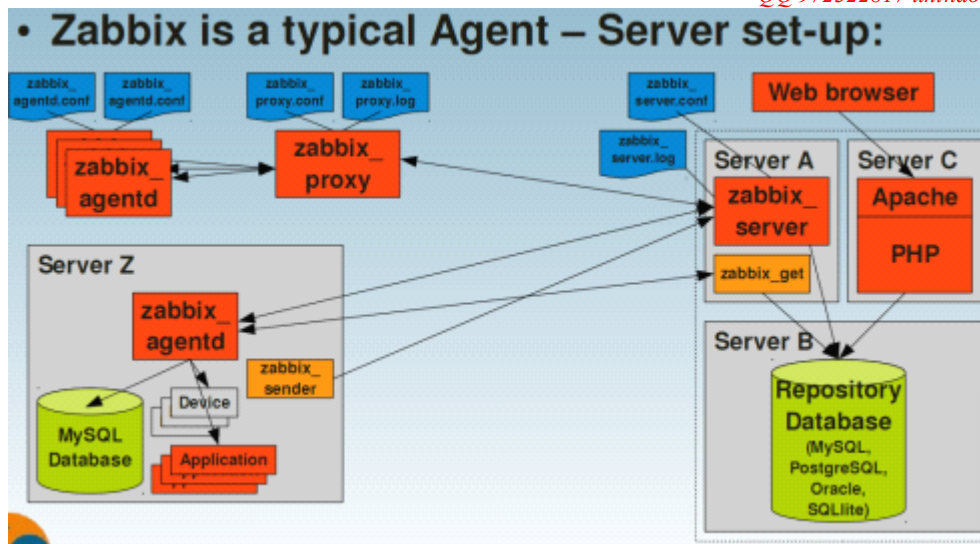
Zabbix 运行条件:

Server: *Zabbix Server* 需运行在 *LAMP* (*Linux+Apache+Mysql+PHP*) 环境下 (或者 *LNMP*), 对硬件要求低

Agent: 目前已有的 *agent* 基本支持市面常见的 OS, 包含 *Linux*、*HPUX*、*Solaris*、*Sun*、*windows*

SNMP: 支持各类常见的网络设备

监控过程逻辑如图示:



Zabbix Server Processes



1.1 Zabbix 功能

具备常见的商业监控软件所具备的功能（主机的性能监控、网络设备性能监控、数据库性能监控、FTP等通用协议监控、多种告警方式、详细的报表图表绘制）

支持自动发现网络设备和服务器（可以通过配置自动发现服务器规则来实现）

支持自动发现（low discovery）key实现动态监控项的批量监控（需写脚本）

支持分布式，能集中展示、管理分布式的监控点

扩展性强，server提供通用接口（api功能），可以自己开发完善各类监控（根据相关接口编写程序实现）

编写插件容易，可以自定义监控项，报警级别的设置。

1.2 优劣势

优点:

开源, 无软件成本投入

Server 对设备性能要求低

支持设备多, 自带多种监控模板

支持分布式集中管理, 有自动发现功能, 可以实现自动化监控

开放式接口, 扩展性强, 插件编写容易

当监控的 *item* 比较多服务器队列比较大时可以采用被动状态, 被监控客户端主动从 *server* 端去下载需要监控的 *item* 然后取数据上传到 *server* 端。这种方式对服务器的负载比较小。

缺点:

需在被监控主机上安装 *agent*, 所有数据都存在数据库里, 产生的数据据很大, 瓶颈主要在数据库。

2. 安装部署

Zabbix Server 可以运行在 CentOS、RedHat Linux、Debian 等 Linux 系统上, 这里以 *centos6.3_X64* 作为部署环境。

预先配置好 *yum*。

2.1 服务端安装 lamp 环境。

```
yum -y install gcc gcc-c++ autoconf httpd php mysql mysql-server php-mysql  
httpd-manual mod_ssl mod_perl mod_auth_mysql php-gd php-xml php-mbstring php-ldap  
php-pear php-xmllrpc php-bcmath mysql-connector-odbc mysql-devel libdbi-dbd-mysql  
net-snmp-devel curl-devel
```

2.2 服务端配置 lamp 使用环境

修改 *php.ini*

shell#vim /etc/php.ini (注意, 这里必须修改, 不然后面安装会提示环境不符)

```
date.timezone = Asia/Shanghai  
max_execution_time = 300  
post_max_size = 32M  
max_input_time=300  
memory_limit = 128M  
mbstring.func_overload = 2
```

如果不想手工修改, 可以使用以下 *sed* 命令操作

```
sed -i "s/;date.timezone =/date.timezone = Asia/Shanghai/g" /etc/php.ini
```

```
sed -i "s#max_execution_time = 30#max_execution_time = 300#g" /etc/php.ini
sed -i "s#post_max_size = 8M#post_max_size = 32M#g" /etc/php.ini
sed -i "s#max_input_time = 60#max_input_time = 300#g" /etc/php.ini
sed -i "s#memory_limit = 128M#memory_limit = 128M#g" /etc/php.ini
sed -i "s#mbstring.func_overload = 0#mbstring.func_overload = 2\n" /etc/php.ini
```

开启 httpd, mysqld 服务,

```
shell#chkconfig mysql on
shell#chkconfig httpd on
shell#service mysql start
shell#service httpd start
```

2.3 服务端 server 的安装过程

zabbix 一键安装脚本，可以实现安装过程全自动化，无需人工干预。

下载 zabbix 安装包，这里以 2.0.3 为例（建议大家下载当前最新版本），

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

Zabbix 2.0 (Stable)

Package	Release	Date	Release Notes	Zabbix Manual	Download
Zabbix Sources Server, Proxy, Agent, GUI	2.0.3	03 October, 2012			Download

<http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/>

2.3.1 安装 zabbix 服务端

```
shell#http://downloads.sourceforge.net/project/zabbix/ZABBIX%20Latest%20Stable/2.0.3/zabbix-2.0.3.tar.gz
增加 zabbix 用户
shell#groupadd zabbix -g 201
shell#useradd -g zabbix -u 201 -m zabbix
shell#tar xvf zabbix-2.0.3.tar.gz
shell#./configure --prefix=/usr/local/zabbix --enable-server --enable-proxy --enable-agent
--with-mysql=/usr/bin/mysql_config --with-net-snmp --with-libcurl
shell#make
shell#make install
```

2.3.2 添加 zabbix 到系统服务文件

```
shell#vim /etc/services
zabbix-agent 10050/tcp #Zabbix Agent
```

```
zabbix-agent    10050/udp          #Zabbix Agent
zabbix-trapper  10051/tcp          #Zabbix Trapper
zabbix-trapper  10051/udp          #Zabbix Trapper
```

```
[root@kx1d ~]# tail -n4 /etc/services
zabbix-agent    10050/tcp          #Zabbix Agent
zabbix-agent    10050/udp          #Zabbix Agent
zabbix-trapper  10051/tcp          #Zabbix Trapper
zabbix-trapper  10051/udp          #Zabbix Trapper
[root@kx1d ~]#
```

2.3.3 导入 zabbix 数据库（此处采用 mysql 数据库）

```
Shell#cd PATH/zabbix-2.0.3 (确保路径在 zabbix 源码下面)
shell#mysqladmin -uroot password 'mysql_pass'; (设置 mysql 的 root 密码)
Shell#mysql -uroot -p (登陆数据库)
mysql>create database zabbix character set utf8;
mysql>grant all privileges on zabbix.* to zabbix@localhost identified by
'zabbix';
mysql>flush privileges;
#导入 zabbix 数据库
shell#mysql -uzabbix -pzabbix zabbix < ./database/mysql/schema.sql
shell#mysql -uzabbix -pzabbix zabbix < ./database/mysql/images.sql
shell#mysql -uzabbix -pzabbix zabbix < ./database/mysql/data.sql
```

确保以上过程无误

#创建链接

```
shell#mkdir /var/log/zabbix
shell#chown zabbix.zabbix /var/log/zabbix
shell#ln -s /usr/local/zabbix/etc /etc/zabbix
shell#ln -s /usr/local/zabbix/bin/* /usr/bin/
shell#ln -s /usr/local/zabbix/sbin/* /usr/sbin/
```

2.3.4 拷贝 service 启动脚本

```
shell#cp misc/init.d/fedora/core/zabbix_* /etc/init.d/
shell#chmod 755 /etc/init.d/zabbix_*
shell#sed -i "s#BASEDIR=/usr/local#BASEDIR=/usr/local/zabbix#g" /etc/init.d/zabbix_server
shell#sed -i "s#BASEDIR=/usr/local#BASEDIR=/usr/local/zabbix#g" /etc/init.d/zabbix_agentd
```

2.3.5 配置 zabbix_server.conf 服务端文件

```
DBName=zabbix    数据库名称
DBUser=zabbix    数据库用户
DBPassword=zabbix 数据库密码
```

```
[root@kx1d zabbix-2.0.3]# cat /etc/zabbix/zabbix_server.conf | grep -v "^#" | grep -v "^$"
```

```
LogFile=/var/log/zabbix_server.log
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
[root@kx1d zabbix-2.0.3]#
```

Sed 命令如下

```
shell#sed -i "s/DBUser=root/DBUser=zabbix/g" /etc/zabbix/zabbix_server.conf
shell#sed -i "/#DBPassword=/aDBPassword=zabbix\n" /etc/zabbix/zabbix_server.conf
shell#sed -i "s#tmp/zabbix_server.log#var/log/zabbix/zabbix_server.log#g" /etc/zabbix/zabbix_server.conf
```

2.3.6 配置 zabbix_agentd.conf 文件（监控 server 本身，如对其他服务器进行监控，配置文件相同）

注意：zabbix_agentd.conf 是客户端的配置文件，这里配置的目的是对自身进行监控

修改 4 处

Server=127.0.0.1 此处添加服务端的 ip，如服务器不为本机，则需要填写远端 zabbix_server 的 ip 地址

ServerActive=127.0.0.1 此处修改为服务端的 ip

/tmp/zabbix_agentd.log 修改日志路径

UnsafeUserParameters=0 默认是不启用自定义脚本功能的，要自定义 key，需开启，设置为 1

```
[root@kx1d zabbix-2.0.3]# cat /etc/zabbix/zabbix_agentd.conf | grep -v "^#" | grep -v "^$"
```

```
LogFile=/var/log/zabbix_agentd.log
Server=127.0.0.1,192.168.1.89
ServerActive=192.168.1.89:20051
Hostname=kx1d.localhost
UnsafeUserParameters=1
```

说明，其中的 192.168.1.89 为 zabbix 服务端的 ip（图中的 20051 为 10051）

使用 sed 命令如下

```
shell#sed -i "s/Server=127.0.0.1/Server=127.0.0.1,192.168.1.89/g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "s/ServerActive=127.0.0.1/ServerActive=192.168.1.89:10051/g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "s#tmp/zabbix_agentd.log#var/log/zabbix/zabbix_agentd.log#g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "#UnsafeUserParameters=0#aUnsafeUserParameters=1\n" /etc/zabbix/zabbix_agentd.conf
```

2.3.7 拷贝网页文件到 apache 目录

```
shell#cp -r ./frontends/php/ /var/www/html/zabbix
shell#chown -R apache.apache /var/www/html/zabbix
```


2.3.8 设置 zabbix 开机启动

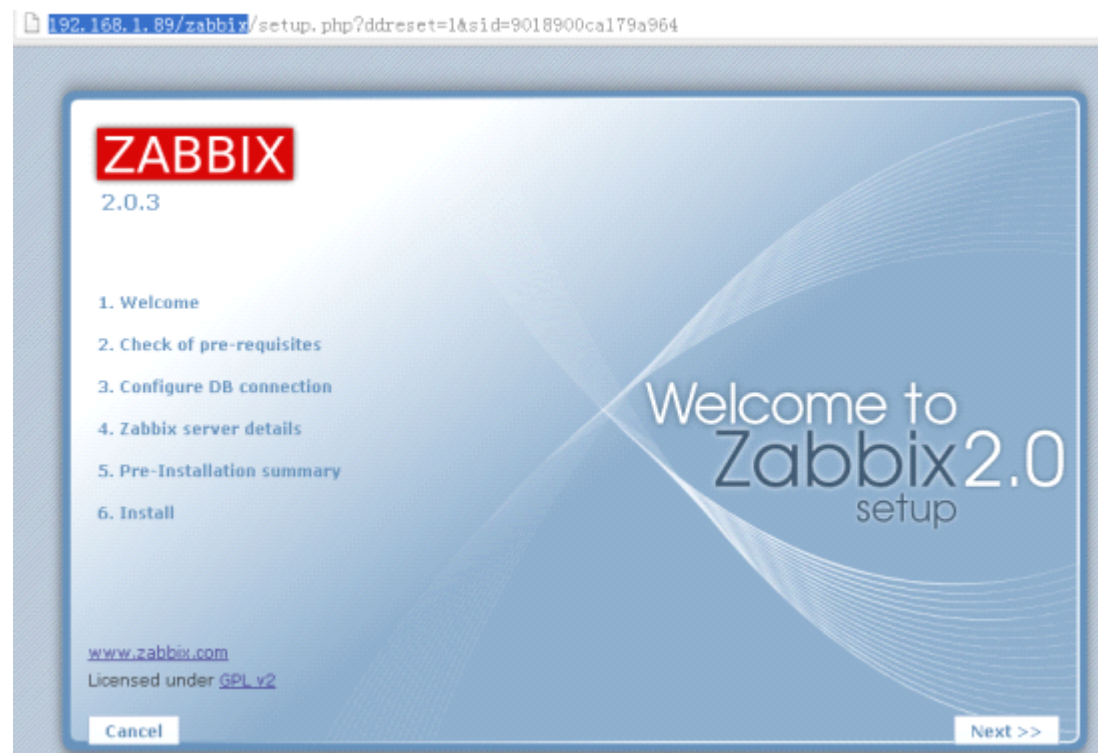
```
shell#start zabbix serivce
shell#chkconfig zabbix_server on
shell#chkconfig zabbix_agentd on
shell#service zabbix_server start
shell#service zabbix_agentd star
```

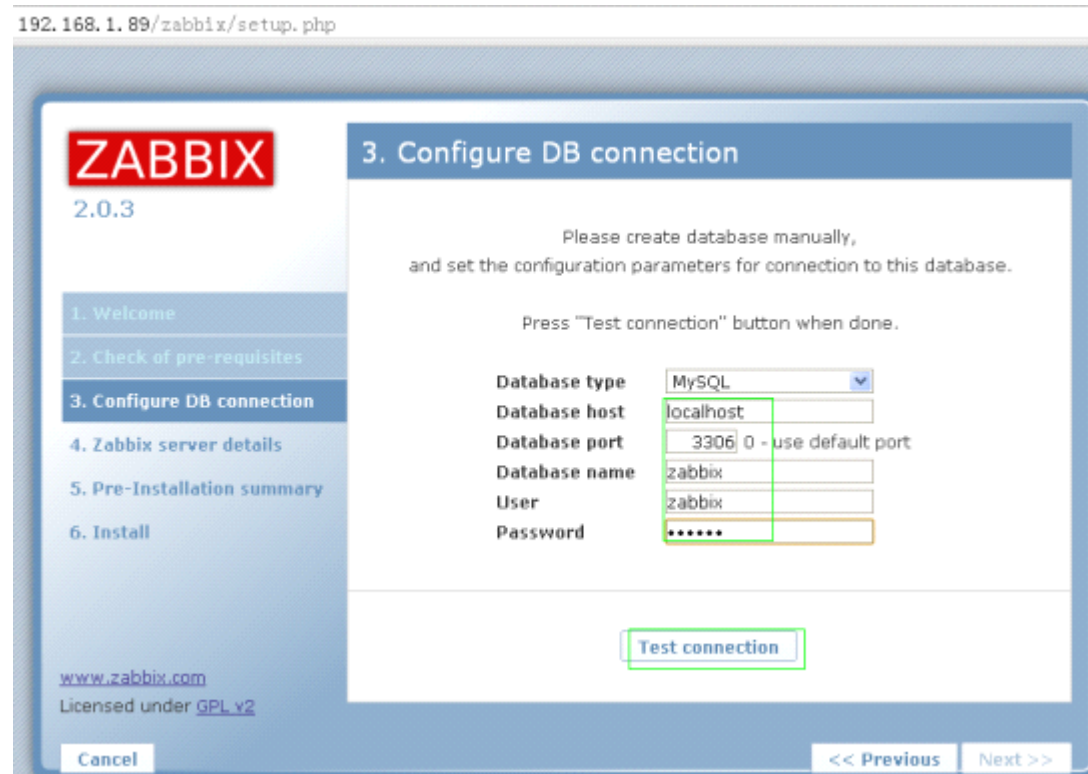
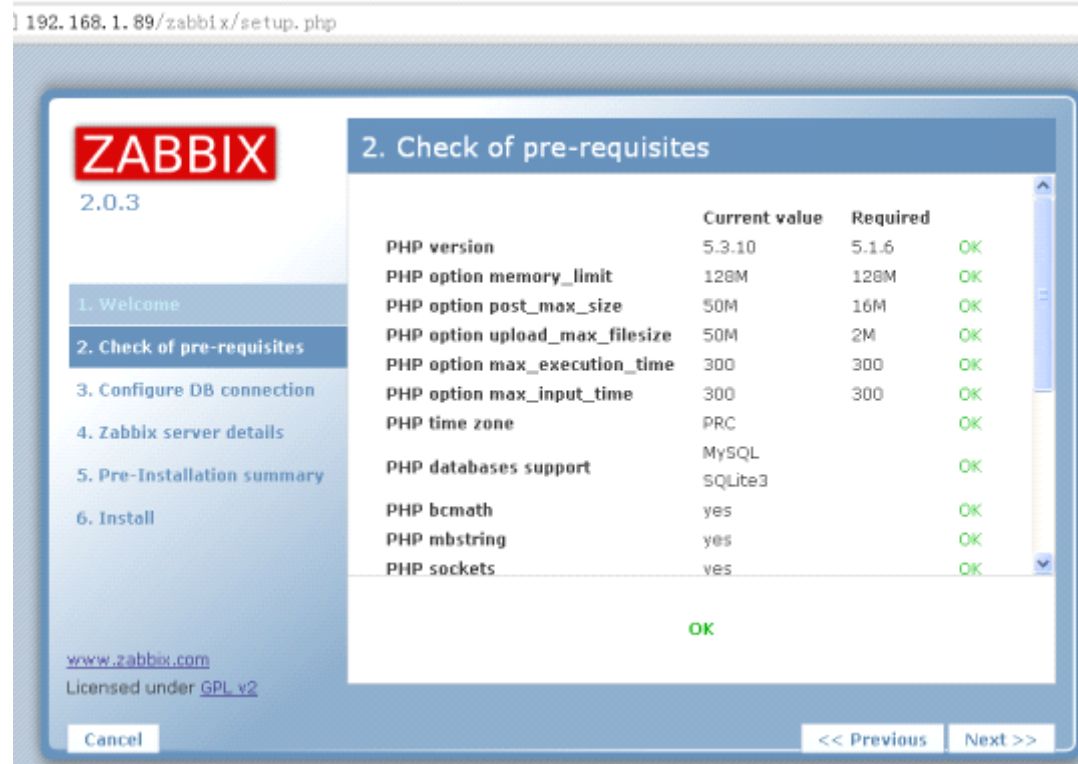
至此，zabbix 的 server 端安装完成。

下面开始通过 web 页面进行配置服务端

2.4 通过 web 页面配置 zabbix

在浏览器输入 <http://192.168.1.89/zabbix>





数据库 ip 为 localhost

端口为 3306

数据库名 zabbix

用户为 zabbix

密码为 zabbix

点击 test connection，如果没有问题，则会提示 OK，否则，需要检查数据库授权是否正确

Press "Test connection" button when done.

Database type	MySQL
Database host	localhost
Database port	3306 0 - use default port
Database name	zabbix
User	zabbix
Password	••••••

OK

Test connection

<< Previous Next >>

192.168.1.89/zabbix/setup.php

ZABBIX
2.0.3

- 1. Welcome
- 2. Check of pre-requisites
- 3. Configure DB connection
- 4. Zabbix server details**
- 5. Pre-Installation summary
- 6. Install

www.zabbix.com
Licensed under [GPL v2](#)

4. Zabbix server details

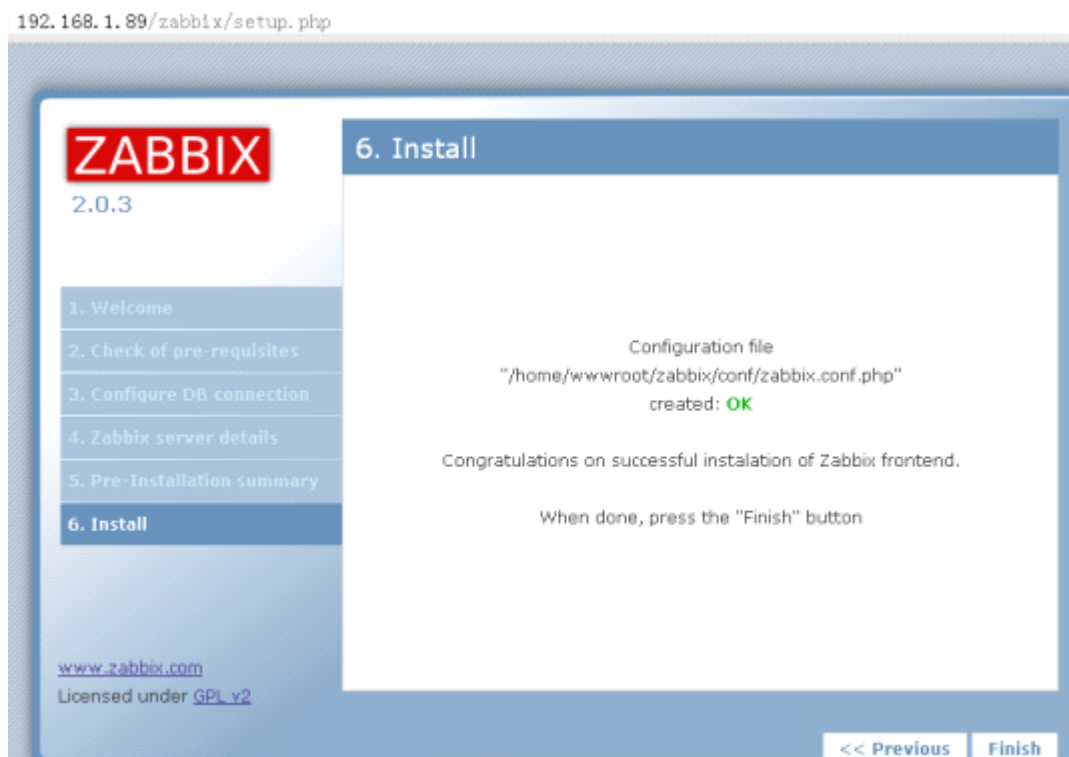
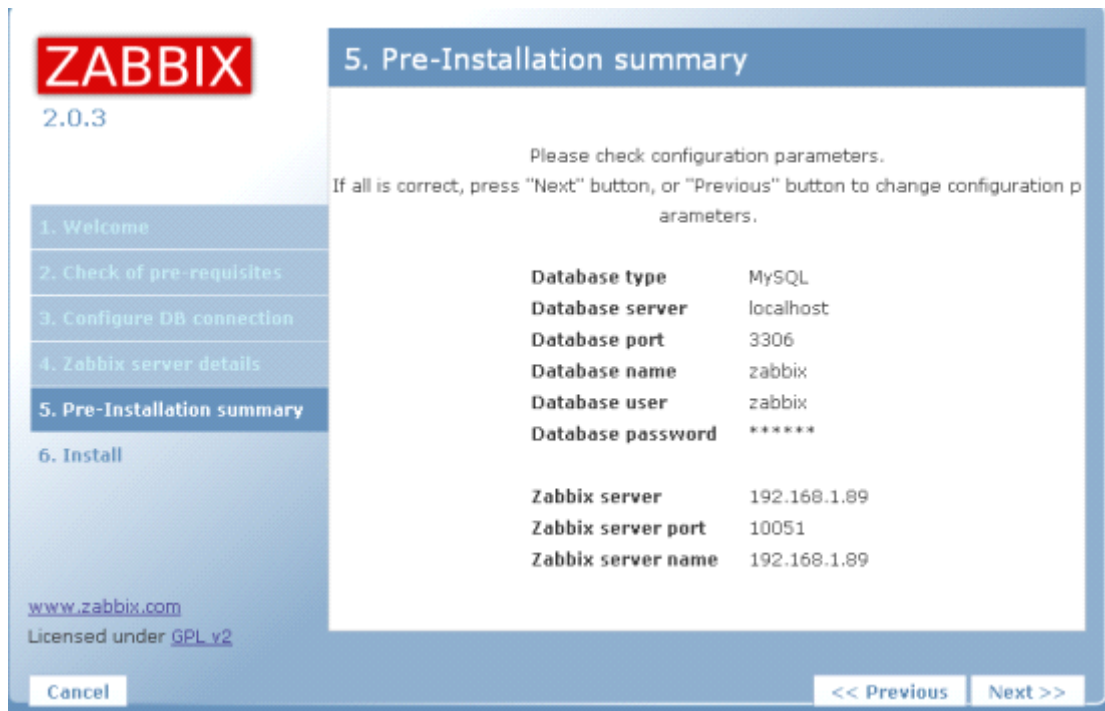
Please enter host name or host IP address and port number of Zabbix server, as well as the name of the installation (optional).

Host	192.168.1.89
Port	10051
Name	192.168.1.89

Cancel << Previous Next >>

Host 本机的 ip

Name 本机的 ip



如果此处提示文件 `zabbix.conf.php` 无法创建，则是 `apache` 目录无法写入，
`/var/www/html/zabbix` 的文件权限不为 `apache.apache`
 输入以下命令解决

```
chown -R apache.apache /var/www/html/zabbix
```



默认用户名为 *admin*，密码为 *zabbix*

2.5 客户端 *agentd* 的安装过程

2.5.1 linux 服务器的安装

```
shell#groupadd zabbix -g 201
shell#useradd -g zabbix -u 201 -m zabbix
shell#tar xvf zabbix-2.0.3.tar.gz
shell#./configure --prefix=/usr/local/zabbix --enable-agent
shell#make
shell#make install
shell#mkdir /var/log/zabbix
shell#chown zabbix.zabbix /var/log/zabbix
shell#cp misc/init.d/fedora/core/zabbix_agentd /etc/init.d/
shell#chmod 755 /etc/init.d/zabbix_agentd
shell#sed -i "s#BASEDIR=/usr/local#BASEDIR=/usr/local/zabbix#g" /etc/init.d/zabbix_agentd
shell#vim /etc/services

zabbix-agent    10050/tcp      #Zabbix Agent
zabbix-agent    10050/udp      #Zabbix Agent
zabbix-trapper  10051/tcp      #Zabbix Trapper
```

```
zabbix-trapper 10051/udp #Zabbix Trappe
shell#sed -i "s/Server\=127.0.0.1/Server\=127.0.0.1,192.168.1.89/g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "s/ServerActive\=127.0.0.1/ServerActive\=192.168.1.89:20051/g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "s#tmp/zabbix_agentd.log#var/log/zabbix/zabbix_agentd.log#g" /etc/zabbix/zabbix_agentd.conf
shell#sed -i "#UnsafeUserParameters=0#aUnsafeUserParameters=1\n" /etc/zabbix/zabbix_agentd.conf
```

```
shell#chkconfig zabbix_agentd on
shell#service zabbix_agentd star
shell#yum -y install net-snmp
shell#mv /etc/snmpd/snmpd.conf/etc/snmpd/snmpd.conf.bak
shell#vim /etc/snmpd/snmpd.conf
```

```
com2sec mynetwork 192.168.1.89 public_orshsoft
com2sec mynetwork 127.0.0.1 public
group MyROGroup v2c mynetwork
access MyROGroup "" any noauth prefix all none none
view all included .1 80
```

```
shell#chkconfig snmpd on
shell#service snmpd restar
```

2.5.2 windows 服务器的安装

下载地址 http://www.zabbix.com/downloads/2.0.3/zabbix_agents_2.0.3.win.zip

Windows 下解压客户端包到 c:, 下载修改好的 zabbix_agentd.conf 文件也放到 c:, 打开 cmd 命令行, 执行

```
C:>zabbix_agentd -install
```

安装后会在系统服务里添加一个 zabbix_agentd 服务, 会自动开机运行

如果需要将客户端和配置文件放在其他目录, 请执行

```
C:>DIR/zabbix_agentd -c DIR/zabbix_agentd.conf -install
```

启动 agentd 服务

```
C:>zabbix_agentd -start
```

或是通过管理->服务找到 zabbix_agentd 来启动

注意: zabbix_agentd.conf 配置文件和 linux 语法意思相同, 此处略过。

3. Zabbix 的配置使用

通过本地浏览器访问 <http://ServerIP/zabbix> 来开始配置和使用 zabbix。

使用 zabbix 进行监控之前，要理解 zabbix 监控的流程。

一次完整的监控流程可以简单描述为：

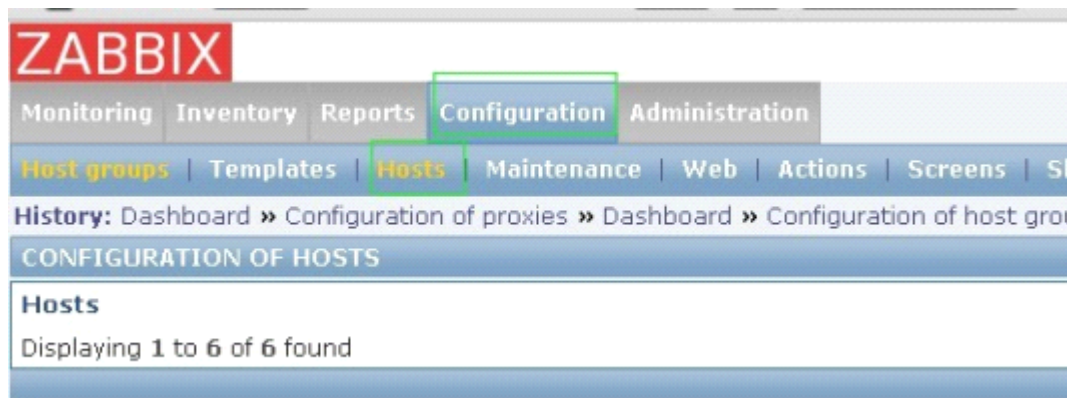
Host Groups (设备组) -> Hosts (设备) -> Applications (监控项组) -> Items (监控项) -> Triggers (触发器) -> Actions (告警动作) -> Medias (告警方式) -> User Groups (用户组) -> Users (用户)

对于实际使用的时候，一般都是采用模板进行监控配置。使用过 cacti 的都知道，可以先添加主机，然后选择对应模板即可，zabbix 中同样存在此功能。

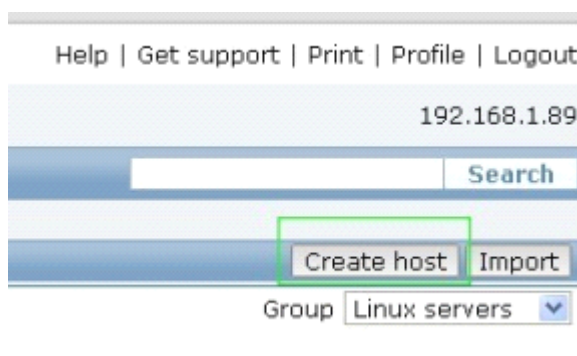
3.1 添加 Hosts

Host 是 Zabbix 监控的基本载体，所有的监控项都是基于 host 的。

通过 Configuration->Hosts->Create Host 来创建监控设备



点击右上角



The screenshot shows the Zabbix web interface for adding a new host. The 'Host' tab is selected. The form includes the following fields and options:

- Host name:** 192.168.1.89
- Visible name:** zabbix_server
- Groups:** Linux servers
- Agent interfaces:** IP address 192.168.1.89, DNS name (empty)
- SNMP interfaces:** Add
- JMX interfaces:** Add
- IPMI interfaces:** Add
- Monitored by proxy:** (no proxy)
- Status:** Monitored

Buttons at the bottom: Save, Cancel.

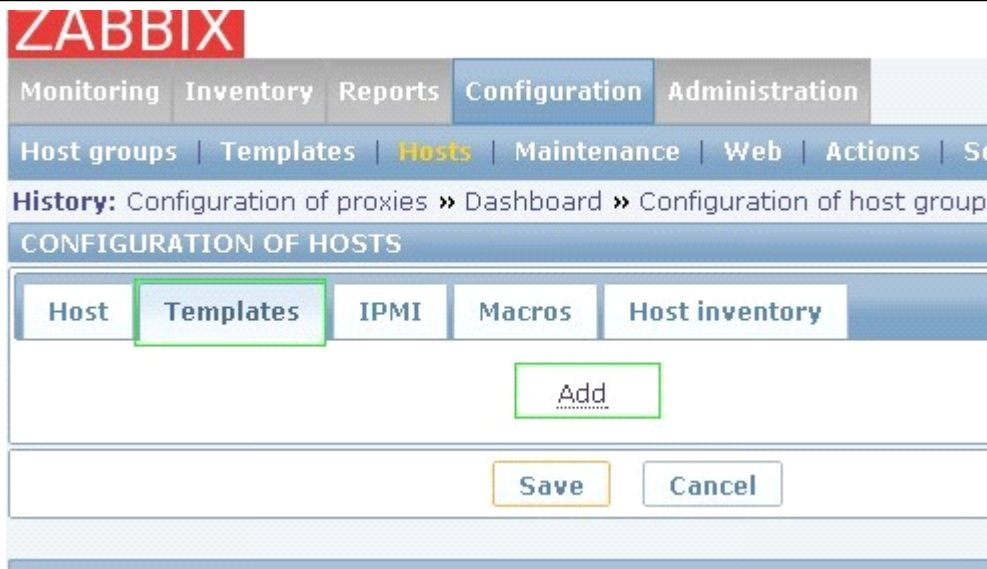
按提示填入 *Name*、*Groups*、*IP*,其他选项默认即可, *Link Templates* 处选择一个模板, *save* 即可成功添加设备。(注: 如果 *host* 上没安装 *agent*, 添加后的状态会是 *unmonitor*, 会采集不到值, *Zabbix* 自带大量的设备监控模板, 我们添加主机时通过 *link* 到这些模板, 就可以快速添加主机的监控项和告警触发条件。一旦采用 *Templates* 模板后, 后面的步骤可以省略)

一类的 *hosts* 可以归属到同一个 *Host Group*, 便于分类管理同一类设备, 在 *Configuration->Host Group->Create Host Group* 可以添加设备组

下面对各项参数进行详解

参数	描述
主机名	输入一个不重复的主机名。只允许大小写字母、数字、标点符号和下线 注意: 你编辑该名称对应客户端配置文件时, 主机名 (<i>hostname</i>) 这一项必须跟你在此处输入的值是一样的。在主机存活检查时需要这个名字。
访问名	你如果设置改名字, 那么它将出现在主机列表,地图等地方。这个属性需要 <i>UTF-8</i> 支持。
群组	选择主机所属的群组。一个主机必须属于至少一个主机组。
新主机组	一个新的群组将被创建然后自动链接到该主机上。如果空的话, 该项将忽略。
接口协议	一个主机支持几种类型的主机接口协议类型: <i>Agent</i> , <i>SNMP</i> , <i>JMX</i> and <i>IPMI</i> 如果想增加一个新的接口协议, 点击 <i>Add</i> 然后输入 <i>IP/DNS</i> , 连接项, 端口等信息
ip 地址	要监控主机的 <i>ip</i> 地址 (可选项)

DNS 名称	要监控主机 DNS 能够解析的名称（可选项）。		
链接	点击各自对应名称的按钮将反馈给 zabbix 服务器用哪个名称（IP 或 DNS）从客户端获得数据。	IP	连接要监控主机的 IP 地址（推荐）
		DNS	链接要监控主机能够正常解析的 DNS 名称
端口	TCP 协议的端口，zabbix 客户端使用的默认值是 10050		
通过代理	主机可以通过 zabbix 服务器或者 zabbix 的一个代理来监控。		
状态	Monitored	主机是活动的，监控就绪	
	Not monitored	主机已停止，因此没被监控	

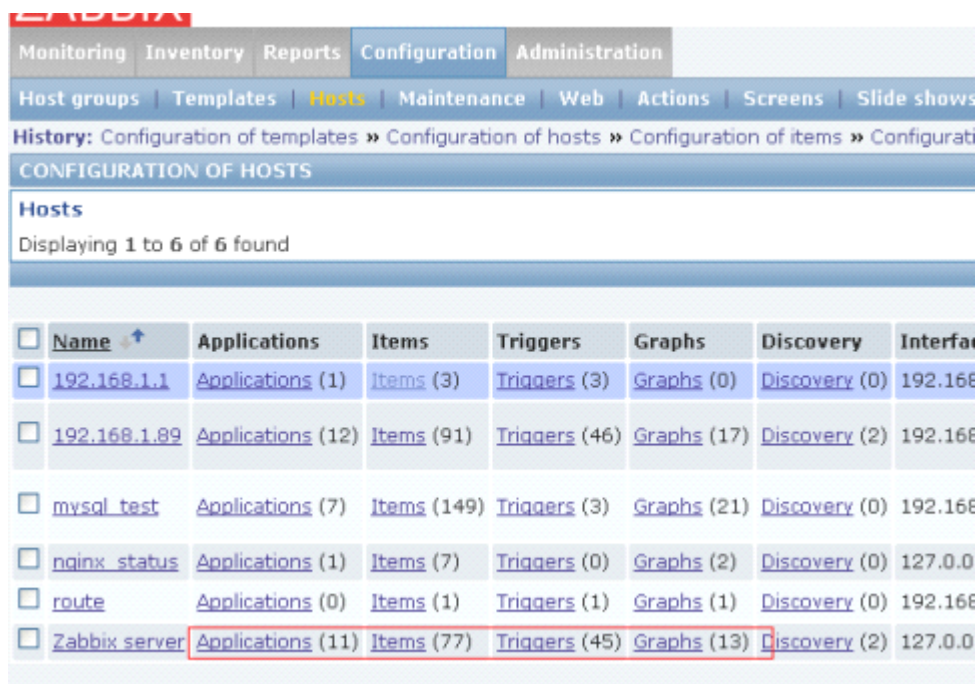


添加模板



Host	Templates	IPMI	Macros	Host inventory
Template App Zabbix Server				Unlink Unlink and clear
Template OS Linux				Unlink Unlink and clear
				Add
<div><div>Save</div><div>Clone</div><div>Full clone</div><div>Delete</div><div>C</div></div>				

注意，此处的模板选择后，会自动创建监控的对象。



Name	Applications	Items	Triggers	Graphs	Discovery	Interface
192.168.1.1	Applications (1)	Items (3)	Triggers (3)	Graphs (0)	Discovery (0)	192.168.1.1
192.168.1.89	Applications (12)	Items (91)	Triggers (46)	Graphs (17)	Discovery (2)	192.168.1.89
mysql_test	Applications (7)	Items (149)	Triggers (3)	Graphs (21)	Discovery (0)	192.168.1.89
nginx_status	Applications (1)	Items (7)	Triggers (0)	Graphs (2)	Discovery (0)	127.0.0.1
route	Applications (0)	Items (1)	Triggers (1)	Graphs (1)	Discovery (0)	192.168.1.89
Zabbix server	Applications (11)	Items (77)	Triggers (45)	Graphs (13)	Discovery (2)	127.0.0.1

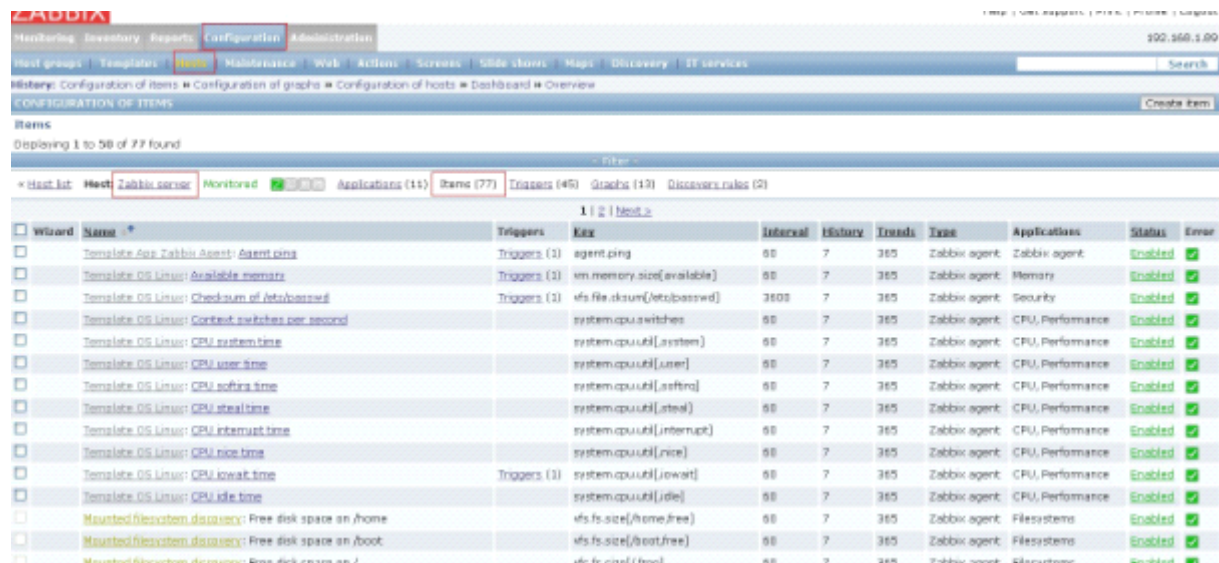
如图示。

3.2 添加 Items

Item 是监控项，是监控的基本元素，每一个监控项对应一个被监控端的采集值。

在 *Configuration->Hosts* 界面，我们能看到每个 *host* 所包含的 *items* 总数，点击对应主机的 *items* 项，可以看到具体的每个 *item* 信息，这些 *items* 可以引用自 *templates*，也可以自己创建。

注意：当我们需要监控的服务器的時候，一般是链接模板，如需自定义 *item*，则需要在此处添加 *Items*。



通过点击具体 *item* 名字可以修改已有监控项的属性，点击 *Status* 的链接可以禁用/启用这个监控项。（注：我们可以通过新建一个 *template*，在 *template* 中禁用掉所有不需要用到的 *items*，然后把同一类 *hosts* link to 这个 *template*，就不用一台台主机去更改 *items*）

新增 *item* 可以通过点击右上角的 *create item* 来创建

此处的 *key* 值来自 *zabbix_agentd.conf* 文件自定义的

`UserParameter=ping.ping1,/etc/zabbix/scripts/ping ping1`

如不需要自定义 *key*，可以点击 *select* 直接选择

关于如何自定义监控脚本，见后文专题叙述。

关于更多 *item*，请参考官方文档

3.3 添加 Triggers

Trigger 是触发器，当 *Items* 采集值满足 *triggers* 的触发条件时，就会产生 *actions*。

每一个 *trigger* 必须对应一个 *item*，但一个 *item* 可以对应多个 *trigger*。通过设置多个 *trigger*，实现触发条件的不同，从而达到不同级别的报警。默认的模板中只有一个 *trigger*。

同样，通过点击 *Configuration->Hosts->Triggers* 中某个 *trigger* 的名字，可以修改 *trigger* 的属性。（注意：引用自 *template* 的 *trigger* 触发值是不能单独修改的，必须在 *template* 中修改，或是复制一个同样的 *trigger* 再修改，然后禁用掉之前的）

新增 *trigger* 可以通过点击右上角的 *create trigger* 来创建

The screenshot shows the Zabbix Trigger configuration form. The 'Name' field is 'ping_checkk'. The 'Expression' field contains '{Template App Zabbix Agent:agent.ping.last(0,)}>0'. The 'Description' field contains 'ping值结果不为0'. The 'Severity' dropdown is set to 'High'. The 'Enabled' checkbox is checked. The 'Save' button is highlighted.

Expression 中选择对应的 *item*、触发方式及触发值，*Severity* 是告警级别，根据 *trigger* 的严重性来选择。

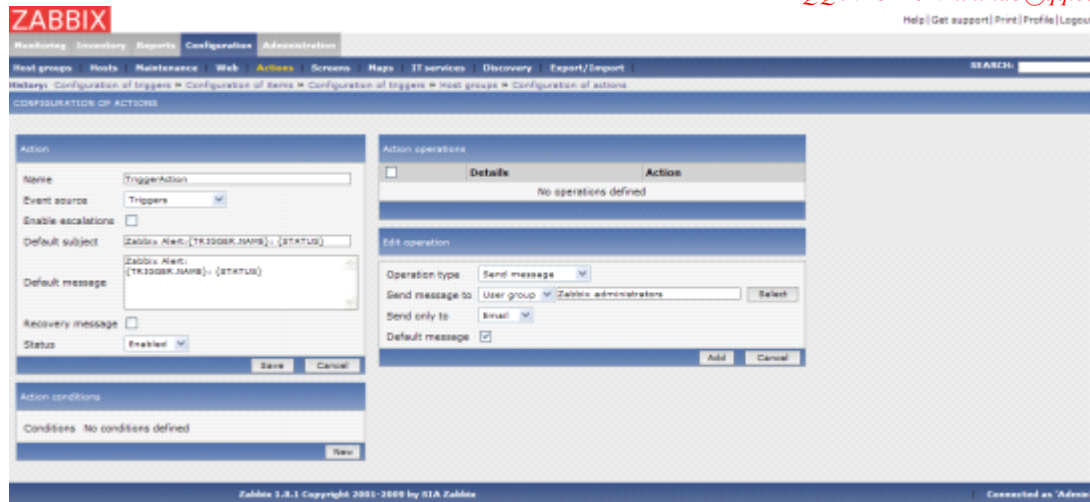
Zabbix 提供多种 *trigger* 触发方式供选择，常用的我们可以选择 *last value* (最近一次采集值),或是选择 *maximal value for period of time* (一段时间内的最大值),等等。可以根据实际需要来设定触发方式。更多的解释请参考：

<http://www.zabbix.com/documentation/2.0/manual/config/triggers>

3.4 添加 Actions

Action 是告警动作，当触发器条件被满足时，就会执行指定的 *action*。

通过 *Configuration->Actions->Create Action* 来创建 *Action*



Event source: 来源

triggers，即所有的 *triggers* 条件满足时都会执行这个 *action*

Discovery: 自动发现模块

Auto registration: 事件产生记录

Escalations: 告警是否升级，及升级时间

Subject、*Message*：告警标题和内容，此处可引用 *zabbix* 的宏变量；例如 `{{HOSTNAME}}:{{TRIGGER.KEY}}.last(0)}` 表示最后一次采集值，更多宏变量参考：

<http://www.zabbix.com/documentation/2.0/manual/config/macros/usermacros>

Recovery Message: 告警恢复信息，不勾选系统会用默认的，勾选后自定义

Conditions: *trigger* 产生的条件，条件可以多选

Operation: 选择 *media* 及 *user*

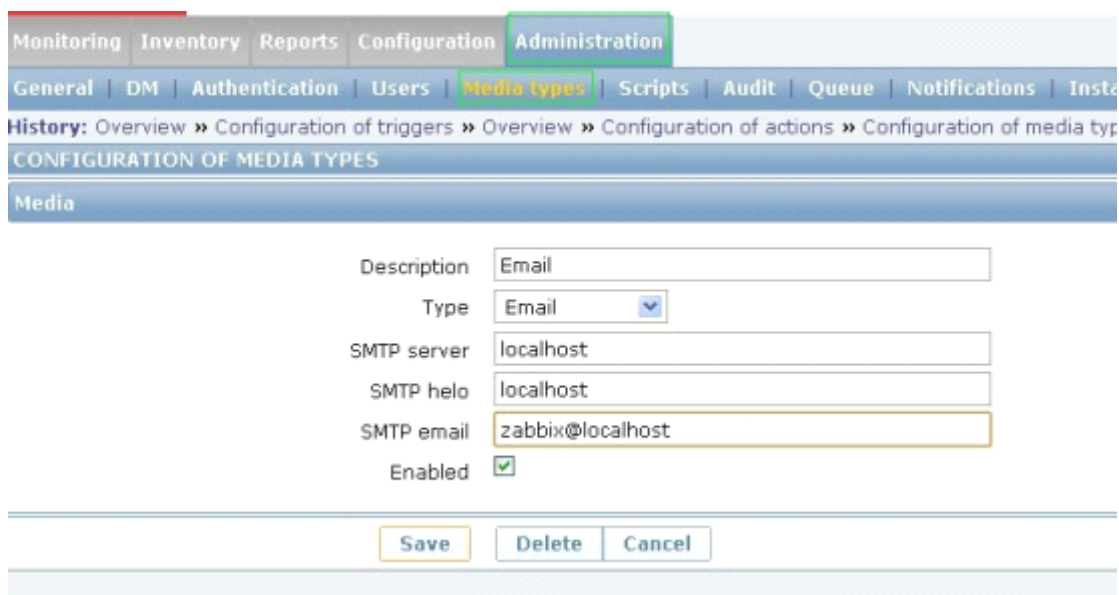
更多资料参考

<http://www.zabbix.com/documentation/2.0/manual/config/notifications/action>

3.5 添加 Medias

Media，即告警方式，*Zabbix* 可以提供四类 *Media*：*Email*/*SMS*/*Jabber*/*Script*，通过 *Administrator*->*Media Type* 来修改或新增告警方式

Email 方式最常用的，填入相关的 *SMTP* 信息，即可通过邮件方式发送告警。



Monitoring | Inventory | Reports | Configuration | Administration

General | DM | Authentication | Users | Media types | Scripts | Audit | Queue | Notifications | Instance

History: Overview » Configuration of triggers » Overview » Configuration of actions » Configuration of media types

CONFIGURATION OF MEDIA TYPES

Media

Description: Email

Type: Email

SMTP server: localhost

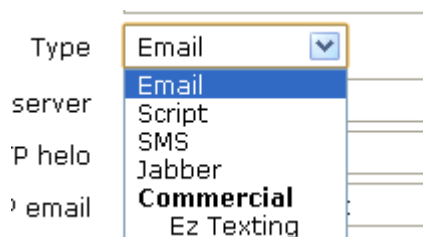
SMTP helo: localhost

SMTP email: zabbix@localhost

Enabled: ☒

Save Delete Cancel

其中 *type* 有以下几种方式



Type: Email

server

P helo

email

Script

SMS

Jabber

Commercial

Ez Texting

Email 方式用邮件

Script 方式可以通过自己编写程序或脚本的方式发送告警信息。

SMS 方式要在 *server* 主机上接入短信 *modem*。

Jabber 方式是一种 *linux* 下的即时通讯工具,通过 *Jabber* 发送即时消息。

3.6 添加 Users

在 *Administrator->Users* 可以添加用户和用户组

通过 *User Group* 可以限制用户的权限, *zabbix* 自带的用户组的权限限制基本能满足我们的要求。

创建用户时可以根据用户的不同作用划分到不同的组, *media* 中填入告警接受地址及告警接受时间等信息。

Monitoring Inventory Reports Configuration Administration

General DM Authentication Users Media types Scripts Audit Queue Notifications Installation

History: Overview » Configuration of actions » Configuration of media types » Overview » Configuration of user groups

CONFIGURATION OF USER

User Media Permissions

Alias Admin

Name Zabbix

Surname Administrator

Password Change password

Groups Zabbix administrators Add

Delete selected

Language English (en_GB)

Theme System default

Auto-login ☒

Auto-logout (min 90 seconds) ☐ 900

Refresh (in seconds) 30

Rows per page 50

URL (after login)

Save Delete Cancel

CONFIGURATION OF USER

User Media Permissions

Media ☐ Email itnihao@163.com

Add Delete selected

192.168.1.89: Media - Google Chrome

192.168.1.89/zabbix/popup_media.php?dstfrm=UserForm

New media

Type Email

Send to

When active 1-7,00:00-24:00

☒ Not classified

☒ Information

Use if severity ☒ Warning

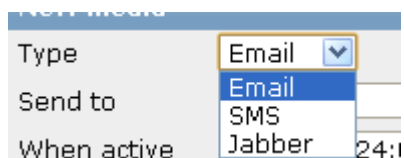
☒ Average

☒ High

☒ Disaster

Status Enabled

Add Cancel

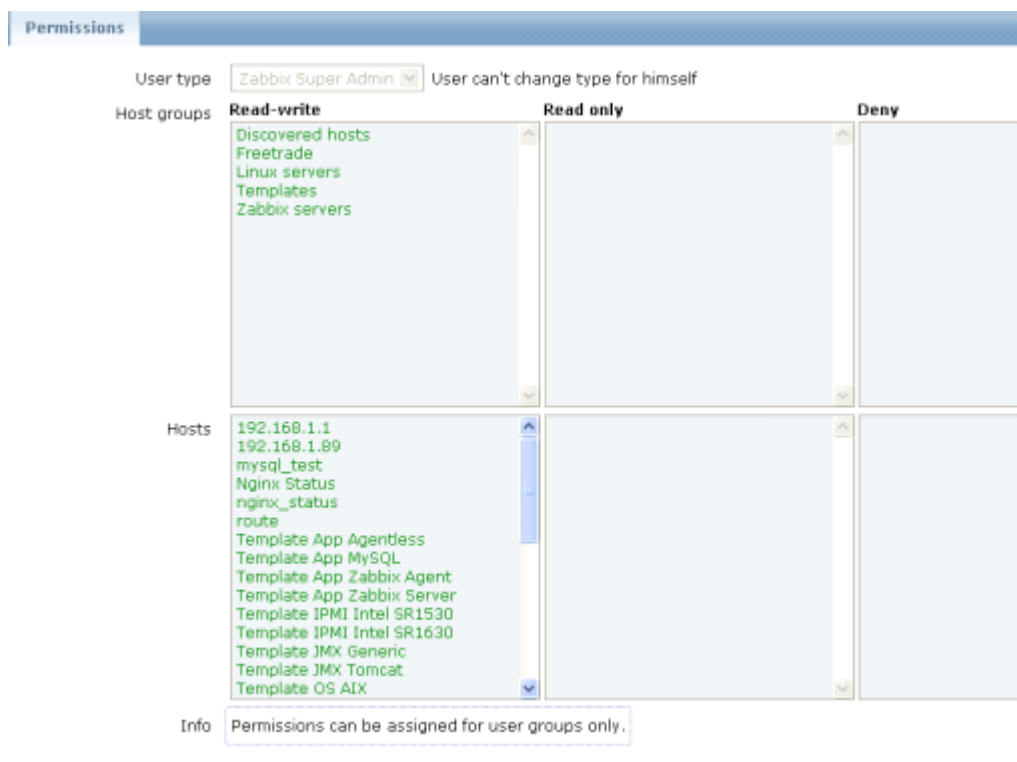


Notification configuration form:

- Type: Email (selected)
- Send to: Email
- SMS
- Jabber
- When active: 24:1

Type: 报警类型

Send to : 收件人地址, 可以增加多个



Permissions configuration page:

- User type: Zabbix Super Admin (User can't change type for himself)
- Host groups: Read-write, Read only, Deny
- Hosts: 192.168.1.1, 192.168.1.89, mysql_test, Nginx Status, nginx_status, route, Template App Agentless, Template App MySQL, Template App Zabbix Agent, Template App Zabbix Server, Template IPMI Intel SR1530, Template IPMI Intel SR1630, Template JMX Generic, Template JMX Tomcat, Template OS AIX
- Info: Permissions can be assigned for user groups only.

用户权限的控制

3.7 添加 WEB Monitorings

Web Monitoring 是用来监控 web 程序的, 可以监控到 web 程序的下载速度、返回码及响应时间, 还支持把一组连续的 web 动作作为一个整体来监控。

下面我们以监控登陆 zabbix 的 web 程序为例, 来展示如何使用 web monitoring。

Configuration->web->Create Scenario 创建一个 Scenario(注: 必须选择 host 后才能创建 scenario, zabbix 的所有 items 都必须创建在 hosts 上)

Name	Timeout	URL	Required	Status	Sort
<input type="checkbox"/> Login	15 sec	http://10.1.50.60/zabbix/index.php			Down
<input type="checkbox"/> ConfigurationPages	15 sec	http://10.1.50.60/zabbix/config.php	CONFIGURATION		Up

图表 1

Application: 选择这个 *scenario* 所在的 *application* 组

Name: *scenario* 的名字

Basic authentication: 鉴权

Update interval: 监控频率, *s* 为单位

Agent: 选择要使用的浏览器客户端, 可能同样的 *web* 程序对不同的客户端展示的内容会不一样

Status: 默认为 *active*

Variables: 变量定义, 这里定义的变量可在后续的 *steps* 中使用, 这里我们定义了用户和密码的变量

Steps: *web* 程序的各个步骤, 选择 *add* 新增一个 *Login* 的 *step*, 来模拟用户登陆, 传递用户和密码给 *index.php* 页面

图表 2

URL: 监控的 *web* 页面(注:必须是全路径带页面名)

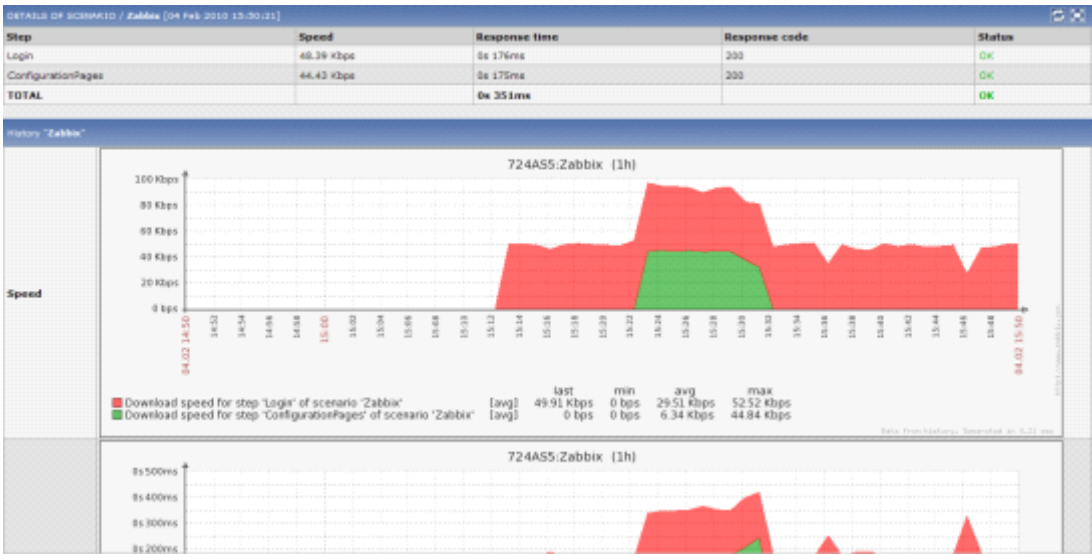
Post:传递给页面的参数, 多个参数之间用&连接, 此处可引用前面定义的变量

Timeout: 超时时间

Required: 页面中能匹配到字符,匹配不到即认为错误

Status codes: 页面返回码

添加完 step 后，我们在 Monitoring->web 页面即能看到监控的状态和图示



图表 3

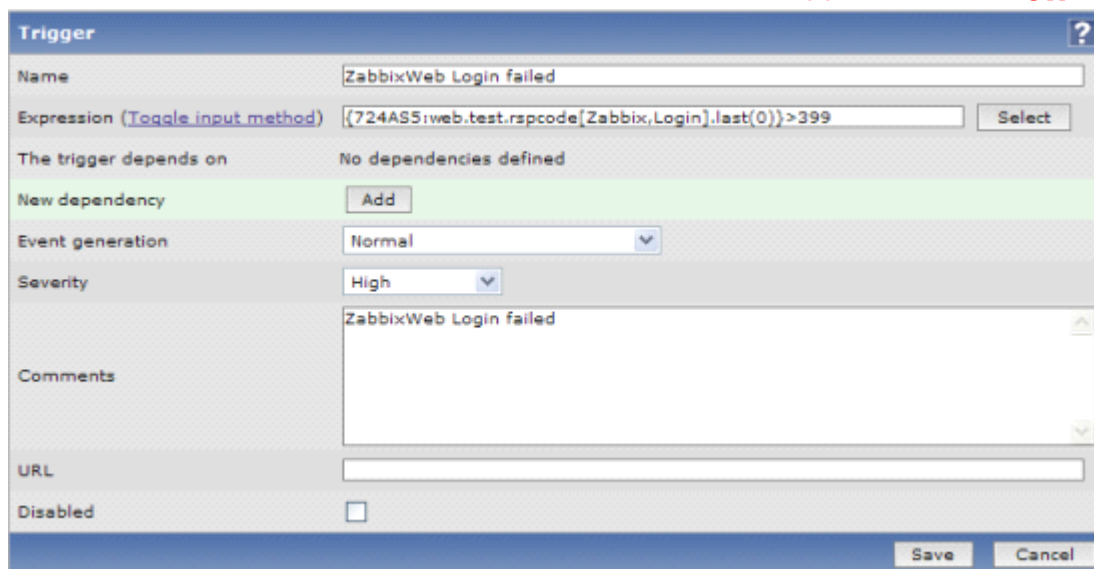
创建完 scenario 后，zabbix server 会自动创建相关的 items，所以我们只需为这些 items 添加 triggers 即可让 web scenario 出错时产生告警

Configuration->hosts->点击 scenario 所在的 host 条目的 trigger，直接 create trigger，在 select items 的时候就可以看到系统自动创建的 items（注：自动创建的 items 在 host 的 items 列表中直接是看不到的，需要在创建 trigger 时选择 items 时才能看到）

Description	Type	Type of information	Status
Buffers memory	Zabbix agent	Numeric (unsigned)	Active
Cached memory	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Not supported
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Not supported
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
Download speed for scenario 'Zabbix'	Web monitoring	Numeric (float)	Active
Download speed for step 'Login' of scenario 'Zabbix'	Web monitoring	Numeric (float)	Active
Download speed for step 'ConfigurationPages' of scenario 'Zabbix'	Web monitoring	Numeric (float)	Active

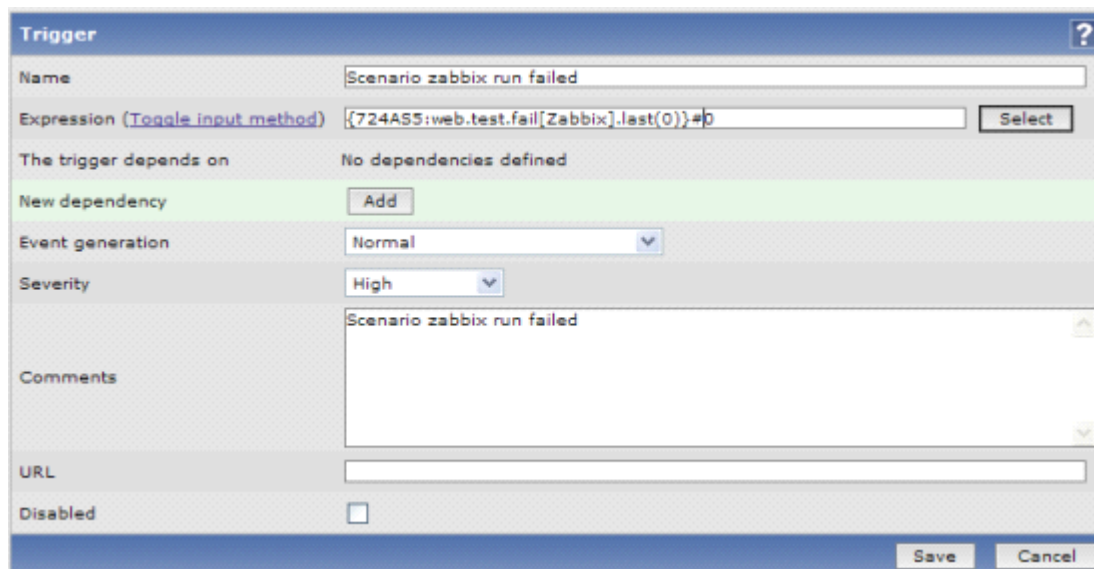
图表 4

可以在 items 列表中看到，系统为每个 step 创建了 3 个 item，Download Speed/Response Code/Response Time,为整个 scenario 创建了一个 test.fail 的 item，可以分别为其创建 trigger 下例我们创建一个 Login 页面返回码的 trigger，大于等于 400 即为错误



图表 5

再创建一个整个 *scenario* 所有 *step* 运行是否成功的 *trigger*, 采集值为 0 表示整个 *scenario* 的所有 *step* 都执行成功了, 第几步的 *step* 执行失败就返回数字几, 且后续的 *step* 都不会继续执行下去。



图表 6

这样, 一个完整的 *web monitoring* 就配置完成了。

Web monitoring 还有更多强大的功能, 未能一一研究了解, 有待挖掘

3.8 添加 Graphs

Zabbix 的 *Graphs* 功能很强大, 可以为每一个 *item* 绘制图表, 也可以把多个 *items* 绘制在一张图表内。

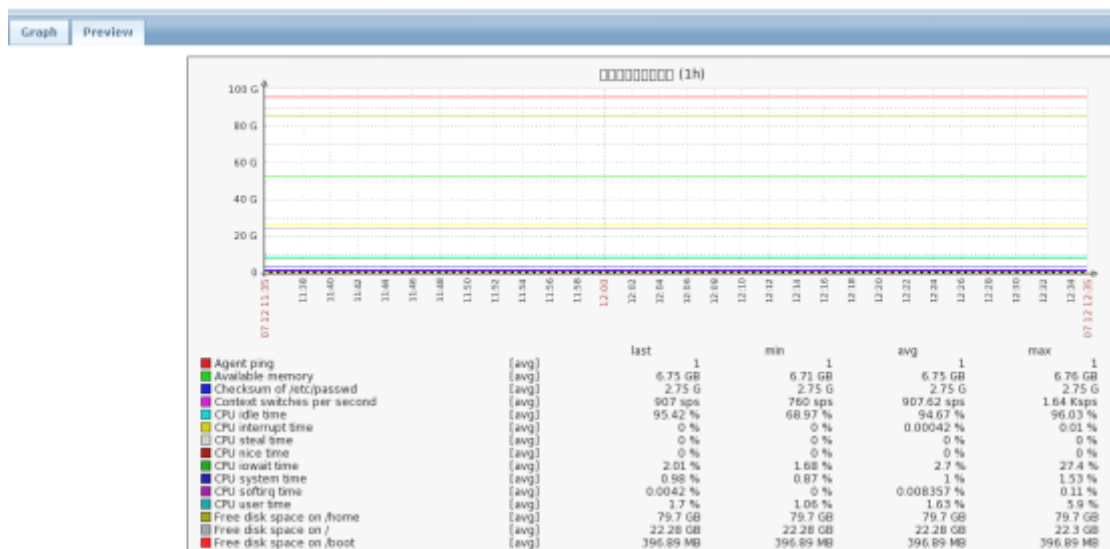
通过 *configuration->hosts* 选择要绘制图表的 *host*, 点击 *graphs*, *create graphs* 即可创建图表。

The image shows the Zabbix Graph configuration interface. The 'Name' field is '服务器各项性能监控'. The 'Width' is 900 and 'Height' is 200. The 'Graph type' is 'Normal'. The 'Show legend', 'Show working time', and 'Show triggers' checkboxes are checked. The 'Percentile line (left)' and 'Percentile line (right)' checkboxes are unchecked. The 'Y axis MIN value' and 'Y axis MAX value' are both set to 'Calculated'. The 'Items' table lists 10 items to be added to the graph:

#	Name	Function	Draw style
1	192.168.1.89: Agent ping	avg	Line
2	192.168.1.89: Available memory	avg	Line
3	192.168.1.89: Checksum of /etc/passwd	avg	Line
4	192.168.1.89: Context switches per second	avg	Line
5	192.168.1.89: CPU idle time	avg	Line
6	192.168.1.89: CPU interrupt time	avg	Line
7	192.168.1.89: CPU steal time	avg	Line
8	192.168.1.89: CPU nice time	avg	Line
9	192.168.1.89: CPU softirq time	avg	Line
10	192.168.1.89: CPU system time	avg	Line

At the bottom, there are 'Save' and 'Cancel' buttons.

注意，此处选择的数据如果为多项，则多项数据呈现在一张图表上面



(此处对中文的支持不好，建议在监控过程中全部使用英文字符)

Graph type: 图表样式，有线状、柱状、饼状

还可以自定义图表大小，及 Y 轴最大最小值

通过 `add items` 可以添加在同一个图表中展示的多个 `items` (注：注意每个 `item` 的颜色及取值范围，范围相差太大图表会显示不全)

配置好的 `graphs` 在 `monitoring->graphs` 中查看

在 `monitoring->last data` 下能快速查看每个 `host` 的每个 `item` 的 `graph`

3.9 添加 Screens

Screen 将多种信息放在一起展示，便于集中展示某个 *host* 的多个信息，或是比较多个 *hosts* 的同一种信息,这些信息可以为 *graphs*、*maps*、*server infos* 等等，几乎涵盖 *zabbix* 所有的监控信息。

通过 *configuration->screen->creat screen* 来创建，创建时定义 *screen* 的行数和列数，点击对应单元格内的 *change*，添加相应的信息

ION OF SCREENS

Name

server1

Columns

2

Rows

3

Save

Cancel

Monitoring | Inventory | Reports | Configuration | Administration

Host groups | Templates | Hosts | Maintenance | Web | Actions | Screens | Slide shows | Maps

History: Configuration of hosts » Configuration of graphs » Overview » Configuration of screens » Overview

CONFIGURATION OF SCREENS

Screens

Displaying 1 to 4 of 4 found

<input type="checkbox"/>	Name	Dimension (cols x rows)
<input type="checkbox"/>	192.168.1.89	2 x 3
<input type="checkbox"/>	mysql test	2 x 6
<input type="checkbox"/>	server1	2 x 3
<input type="checkbox"/>	Zabbix server	5 x 5

History: Overview » Configuration of screens » Overview » Configuration of screens » Overview

CONFIGURATION OF SCREEN

server1

	*	*
*	Change	Change
*	Change	Change
*	Change	Change
*	Change	Change

Zabbix 2.0.3 Copyright 2001-2012 by Zabbix SIA

点击 *change* 添加图像

Screen cell configuration

Resource: Graph

Graph name: 192.168.1.89: CPU jumps Select

Width: 500

Height: 100

Horizontal align: Left Center Right

Vertical align: Top Middle Bottom

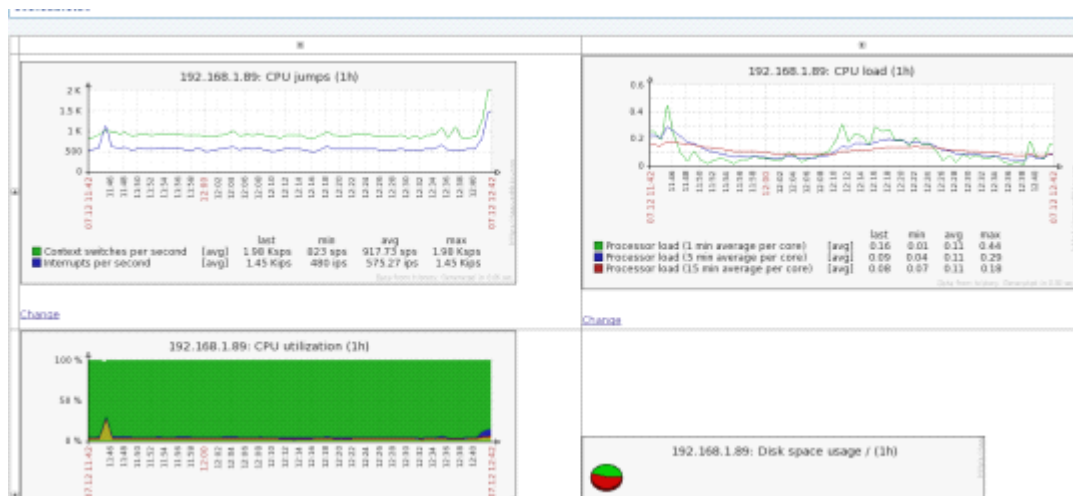
Column span: 1

Row span: 1

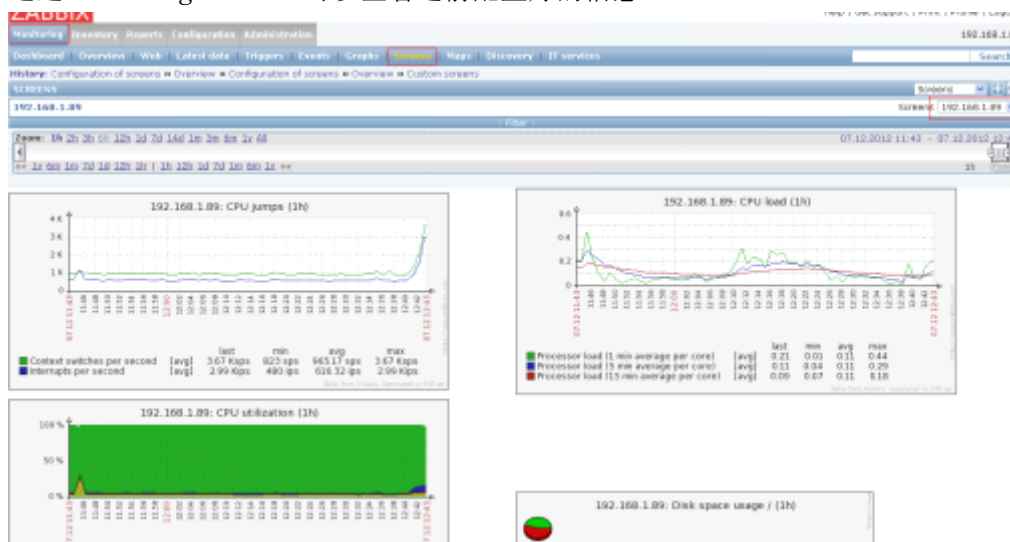
Dynamic item: ☐

Save Cancel

选择图像类型，对齐方式等



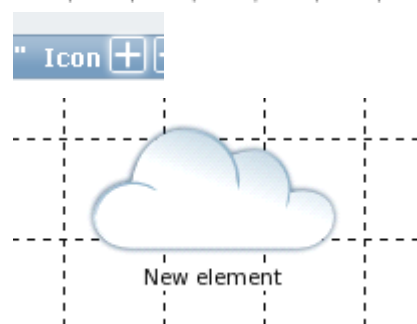
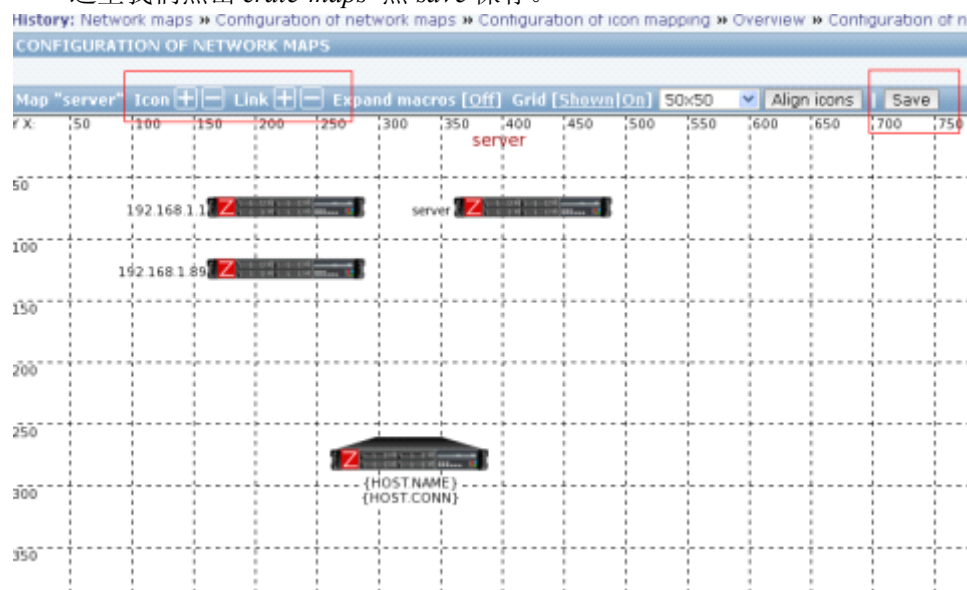
通过 *monitoring-screen*，可以查看之前配置好的信息



3.10 添加 Maps

这里可以添加关于主机的拓扑图：*configuration-maps*，在右上角可以 *create maps* 或者 *import map*

这里我们点击 *crate maps* 点 *save* 保存。



双击

Edit map element

Type: Host

Label: server monitor

Label location: Bottom

Host: 192.168.1.89 [Select](#)

Icons

Automatic icon selection ☐

Default: Zabbix_server_3D_(64) Problem: Default

Maintenance: Default Disabled: Default

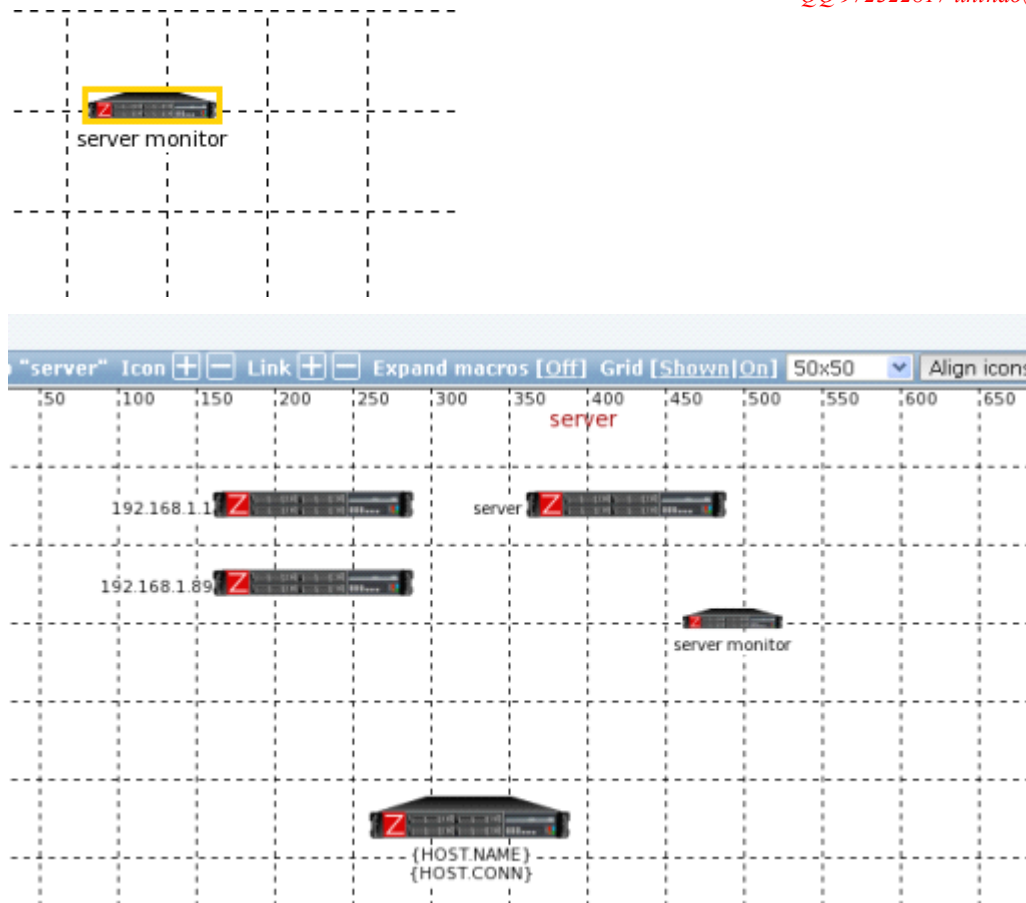
Coordinates X: 461 Y: 141

URLs

Name	URL	
		Remove
		Remove

[Add](#)

[Apply](#) [Remove](#) [Close](#)



上面加号和减号可以增加主机 link 说明主机的连接情况，可以进行自定义。

3.11 添加 MySQL 监控

Zabbix 自带有 MySQL 的监控模板，可以做一些简单的监控。

1、更改 agentd 配置

```
#vim /etc/zabbix/zabbix_agentd.conf
```

```
UnsafeUserParameters=1
```

```
UserParameter=mysql.ping,mysqladmin -uroot -pharry ping|grep alive|wc -l
```

修改后重启 host 上的 agentd，使配置文件生效。

```
#service zabbix_agentd restart
```

2、添加 items

web 端编辑 mysql 所在的 host，使之 link 到 template_APP_MySQL 模板，然后在 host 的 items 里就能看到刚才定义的这些 MySQL 的监控项了，修改相应的 trigger 值即可。

这个 zabbix 自带的 mysql 监控功能比较弱，只是通过 mysqladmin 工具去查询 mysql 的一些状态而已。

我们可以自己编写或是找一些功能更强的 mysql 监控脚本，加到 zabbix 监控里，后面会讲到如何自己添加监控。

另外更详细的方法可参考 zabbix wiki 上的 mysql 监控方法，这个监控的就非常详细：http://www.zabbix.com/wiki/howto/monitor/db/mysql/extensive_mysql_monitoring_including_r

[eplication](#)

3.12 添加 SNMP 监控

Zabbix snmp 的监控要在 *configuration* 中 *hosts* 中添加相关主机的模板和 *snmp* 版本，一般用 *snmp* 监控网络设备就可以了。

3.13 添加自定义监控

对于 zabbix 功能上无法实现的监控，我们可以通过自己编写程序或脚本来辅助完成，并将脚本的结果通过 *agent* 递交给 *zabbix server* 统一管理，一样可以绘制 *graph* 报表等。

UserParameters 的定义方法，请参考：

<http://www.zabbix.com/documentation/2.0/manual/config/items/userparameters>

如何自定义监控

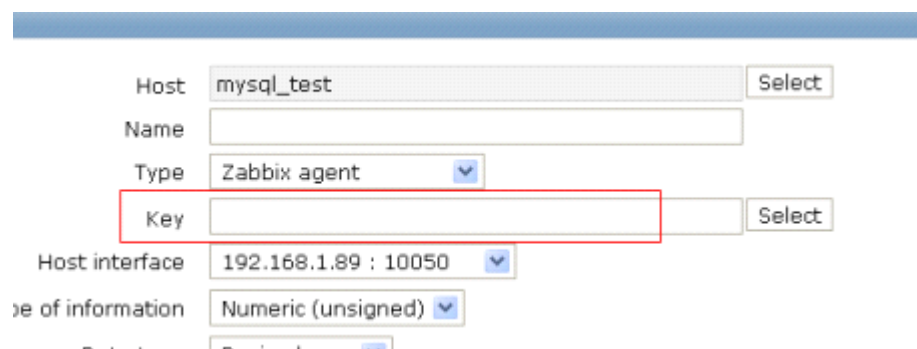
修改 */etc/zabbix/zabbix_agentd.conf*

UnsafeUserParameters=1 此处默认是 0（不允许自定义脚本）

然后在后面添加 *UserParameter=key,command*

key,command 为格式

此处 *key* 是在创建 *item* 的时候填写的



在修改 *zabbix_agentd.conf* 后，重启 *zabbix_agentd* 服务

```
UserParameter=mysql.ping,mysqladmin -uroot -ppharry ping
UserParameter=ping.ping1,/etc/zabbix/scripts/ping ping1
UserParameter=ping.ping2,/etc/zabbix/scripts/ping ping2
```

例如 *ping.ping1*

用 *zabbix_get* 命令获取得到的值

```
[root@kx1d zabbix-2.0.3]# zabbix_get -s 192.168.1.89 -p 10050 -k ping.ping1
0
[root@kx1d zabbix-2.0.3]#
UserParameter=mysql.version,mysql -V
UserParameter=nginx.accepts,/etc/zabbix/scripts/nginx_status accepts
UserParameter=nginx.handled,/etc/zabbix/scripts/nginx_status handled
UserParameter=nginx.requests,/etc/zabbix/scripts/nginx_status requests
UserParameter=nginx.connections.active,/etc/zabbix/scripts/nginx_status active
UserParameter=nginx.connections.reading,/etc/zabbix/scripts/nginx_status reading
UserParameter=nginx.connections.writing,/etc/zabbix/scripts/nginx_status writing
UserParameter=nginx.connections.waiting,/etc/zabbix/scripts/nginx_status waiting
[root@kx1d zabbix-2.0.3]# zabbix_get -s 192.168.1.89 -p 10050 -k nginx.connections.active
8
[root@kx1d zabbix-2.0.3]#
```

下面以一个实例来说明此问题

例如要监控 *mysql*

#vim /etc/zabbix/zabbix_agentd.conf

UnsafeUserParameters=1

UserParameter=mysql.ping_test[],mysqladmin -u\$1 -p\$2 ping|grep alive|wc -l*

#service zabbix_agentd restart

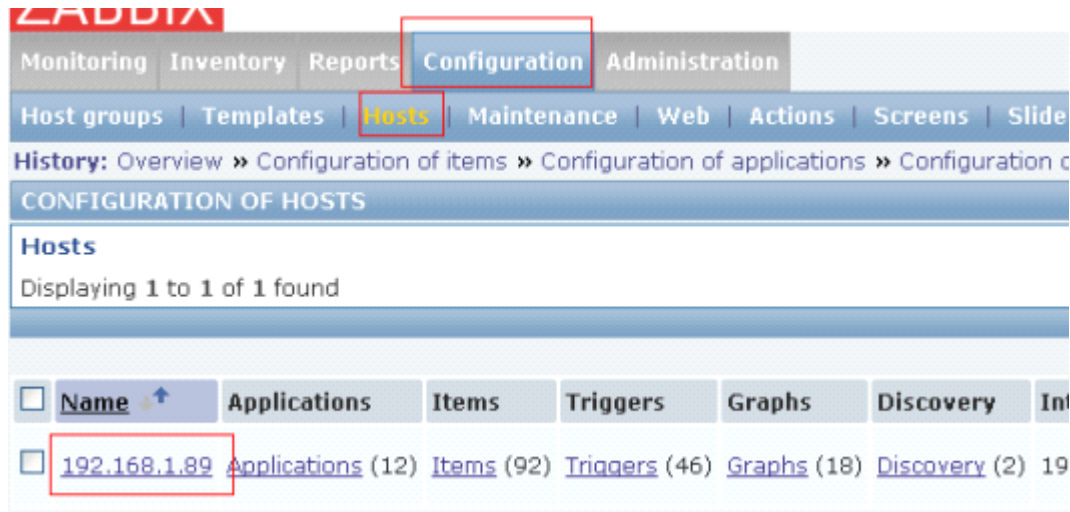
用 zabbix_get 检测设置是否生效

```
[root@kx1d zabbix-2.0.3]# zabbix_get -s 192.168.1.89 -k mysql.ping_test[root,harry]
1
[root@kx1d zabbix-2.0.3]#
```

返回值为 1，说明添加成功

下面添加到主机。

选择一个 host 192.168.1.89



点击 Items-create Itmes

输入 key 值为 mysql.ping_test[root,harry]

The screenshot shows the 'Create item' form in Zabbix. The form fields are filled with the following values:

- Host: 192.168.1.89
- Name: mysql.ping_test
- Type: Zabbix agent
- Key: mysql.ping_test[root,harry]
- Host interface: 192.168.1.89 : 10050
- Type of information: Numeric (unsigned)
- Data type: Decimal
- Units: (empty)
- Use custom multiplier: (checked)
- Update interval (in sec): 30
- Flexible intervals: (empty)
- New flexible interval: Interval (in sec) 50, Period 1-7,00:00-24:00, Add
- Keep history (in days): 90
- Keep trends (in days): 365
- Store value: As is
- Show value: As is, show value mappings
- New application: (empty)
- Applications: (empty)

Status Enabled

Save Clone Clear history and trends Delete Cancel

点击 *graphs---create graph*

Name

Width

Height

Graph type Normal

Show legend ☒

Show working time ☒

Show triggers ☒

Percentile line (left) ☐

Percentile line (right) ☐

Y axis MIN value Calculated

Y axis MAX value Calculated

Items	Name	Function
1:	192.168.1.89: mysql.ping test	avg

[Add](#)

在 ADD 处选择 Items 为 *mysql.ping_test--save*,稍等会儿就出图



当然，此处为一个简单的实现，如果需要监控复杂的应用，需要写脚本来实现，但方法是类似的。

3.14 添加 Templates

如果有大量的同一类设备，需要监控的信息也大致类似，一个个去修改相关参数比较麻烦，我们可以通过创建一个 *template* 来简化操作。

Configuration->Host Groups->Template->Create Template

点击 Items---create Items-

<input type="checkbox"/> Templates	Applications	Items	Triggers	Graphs	Screens
<input type="checkbox"/> mysql_ping_test	Applications (1)	Items (1)	Triggers (0)	Graphs (1)	Screens (0)

create Items

graph--create graph--

CONFIGURATION OF GRAPHS

< Template list Template: mysql.ping_test Applications (1) Items (1) Triggers (0) Graphs (1) Screens (0) Discovery rules (0)

Graph Preview

Name: mysql.ping_test

Width: 900

Height: 200

Graph type: Normal

Show legend: ☒

Show working time: ☒

Show triggers: ☒

Percentile line (left): ☐

Percentile line (right): ☐

Y axis MIN value: Calculated

Y axis MAX value: Calculated

Name	Function	Draw style	Y axis side	Colour
1: mysql.ping_test:mysql.ping_test	svg	Line	Left	C90000

Add

Save Close Delete Cancel

Zabbix 2.0.3 Copyright 2001-2012 by Zabbix SIA

添加到模板，后面就可以引用改模板了。

3.15 添加 Reports（定制报表）

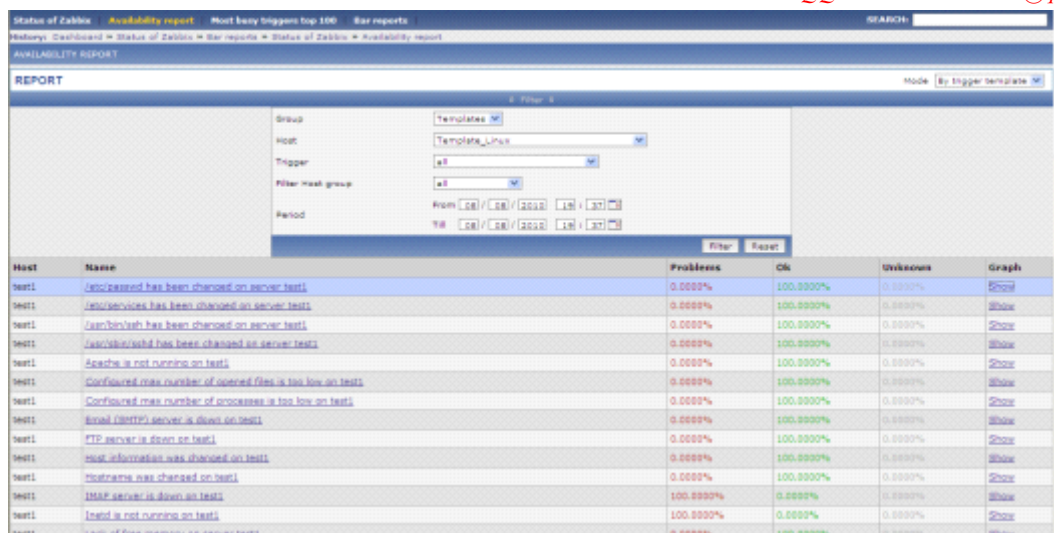
在 zabbix 中关于报表的功能有三项：

Status of zabbix: 这是关于整个 zabbix 监控系统的

Parameter	Value	Details
Zabbix server is running	Yes	-
Number of hosts (monitored/not monitored/templates)	46	3 / 1 / 42
Number of items (monitored/disabled/not supported)	322	294 / 0 / 28
Number of triggers (enabled/disabled)[problem/unknown/ok]	128	127 / 1 [18 / 0 / 109]
Number of users (online)	3	2
Required server performance, new values per second	10.43	-

Updated: 19:43:51

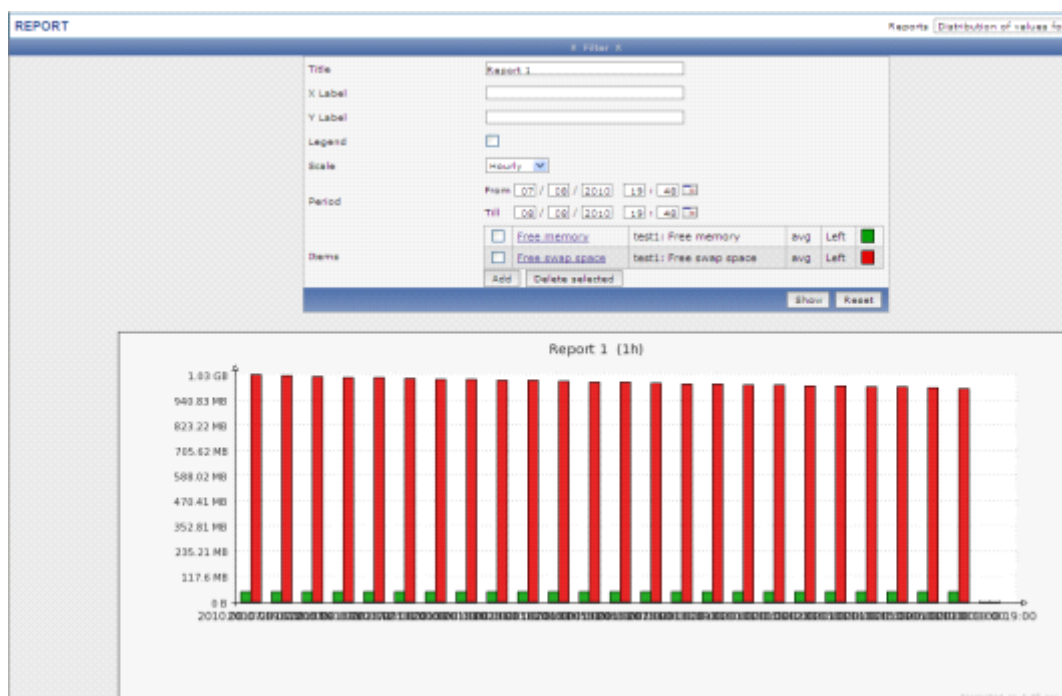
Availiability report: 整个系统可用的系统报表提供过滤功能。



Most busy triggers top 100: 提供最常用的 triggers 预览:

Bar report : 可定制报表可以报多个报表整合到一起。

如下图是对 test1 server 的 free memory 和 swap free 每小时报表:



3.16 添加 Macros

Macros 指宏变量, 定义的宏变量可以在 trigger、actions 等多种场景中引用。

Macros 分系统自带全局宏的及自定义的宏。

系统自带的全局 macros 列表及解释参考:

<http://www.zabbix.com/documentation/2.0/manual/config/macros>

引用 macros 的例子可参考上述 action 中添加 `{{Hostname}}:{{trigger.key}}.last(0)` 的例子。

Zabbix 还支持自定义 macros, 在添加 host 或是 template 时, 我们可以在 macros 项中定义好后续要用到的宏变量, 格式为:

`{macroname}=macrovalue`

自定义的宏变量及系统自带的宏变量都可以在 *zabbix* 场景中引用, *zabbix* 在遇到引用的宏变量时, 会先查找当前场景中定义的宏, 接着查找当前 *host* 的自定义宏, 接着查找 *link* 的 *template* 的宏, 最后查找 *zabbix* 系统自带的全局宏。所以在自定义宏时注意宏引用的顺序。

3.17 添加自动发现设备

通过 *zabbix* 的相关设置, *zabbix* 可以自动添加设备, 可以更友好的维护和添加相关设备。

详细参考:

<http://www.zabbix.com/documentation/2.0/manual/discovery>

3.18 添加 Inventory

Inventory 用来管理设备存档信息的。

在添加 *host* 时, 勾选右侧的 *Use profile*, 我们即可填入该台设备的型号、编码、*MAC* 地址等详细信息, 勾选 *Use extended profile* 则可以填入更详细的信息。

	Disabled	Manual	Automatic
Type	abc		
Type (Full details)	1		
Name			
Alias			
OS			
OS (Full details)			
OS (Short)			
Serial number A	abc		
Serial number B	abc		
Tag	abc		
Asset tag	abc		
MAC address A	abc		
MAC address B	abc		
Hardware	abc		
Hardware (Full details)	abc		

填写的 *inventory* 信息在 *inventory->hosts* 下能看到. 可以代替 *OCS* 一些功能。



3.19 Export/Import XML

Zabbix 提供将所有配置导出为标准 XML 格式的文件, 同样, 也支持导入标准格式的 XML 配置文件。

通过 *configuration->Templates>Export/Import->Export*, 勾选要导出的模板, *Preview* 可以展示要导出的 *host* 的详细配置, 选择 *export* 即可导出 *xml* 文件到本地。

3.20 Maintenance (维护时间)

这一点和 Nagios 的 [Schedule downtime for this host](#) 差不多, 在 Nagios 中可以设置在 *downtime* 不需要告警, 但是 *zabbix* 设置的更加详细和可管理。

CONFIGURATION OF MAINTENANCE PERIODS

Maintenance Periods Hosts & Groups

Name 维护时间

Maintenance type With data collection

Active since 07 / 12 / 2012 16 : 35

Active till 08 / 12 / 2012 16 : 38

Description 维护期间

Save Cancel

Zabbix 2.0.3 Copyright 2001-2012 by Zabbix SIA

选择主机

ZABBIX

Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance Web Actions Screens Slide shows Maps Discovery IT services

History Host inventory overview Configuration of templates Overview Configuration of maintenance Overview

CONFIGURATION OF MAINTENANCE PERIODS

Maintenance Periods Hosts & Groups

Hosts in maintenance In maintenance

192.168.1.99

Other hosts Group Linux servers

Groups in maintenance In maintenance

Other groups

Linux servers

Zabbix servers

Save Cancel

注意，此功能的使用要结合 Action 功能一起使用

ZABBIX

状态统计 资产记录 系统评估 系统配置 管理

本机组 模板 本机 维护 维护 Web 配置 配置图表 Slide shows 拓扑图 自动发现 服务

历史记录: Configuration of maintenance Dashboard Custom graphs Dashboard 配置操作过程

动作配置

操作 触发条件 详细操作

计算方式 (A) 与 (B)

触发条件

Label	名称	操作
(A)	维护状态 不在 "维护"	Remove
(B)	Trigger value = "PROBLEM"	Remove

新的触发条件

Trigger name 触发

添加

保存 复制 删除 取消

3.21 Proxy 的使用

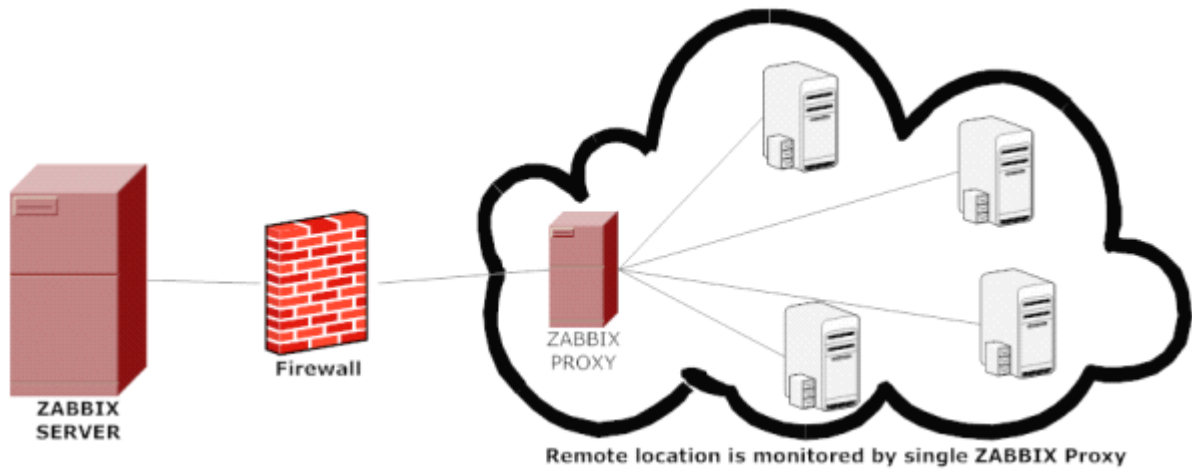
zabbix 中文文档--代理 (Proxies)

概述


一个 zabbix 代理 (Proxies) 可以代表 zabbix 服务器收集性能和可用性数据。这样，代理 (Proxies) 可以负担采集数据的任务并且减轻 zabbix 服务器负载 同时，使用代理 (Proxies) 是实施统一和分布式监控的最简单方式，因为所有的客户端和代理 (Proxies) 向一个 zabbix 服务器报告数据，并且所有数据集中保存在服务器数据库。

一个 zabbix 代理 (Proxies) 可以用在以下：

- 监控远程区域
- 监控拥有不可靠链接的区域
- 当监控数以千计的设备时分担 zabbix 服务器的负载
- 简化分布式监控的维护



代理 (Proxies) 与服务器之间仅需要一个 TCP 连接。这样将更容易避开防火墙因为你仅需要配置一条防火墙规则



zabbix 代理 (Proxies) 必须使用一个单独的数据库。代理 (Proxies) 执行 zabbix 服务器的数据库将打乱配置

所有代理 (Proxies) 采集到的数据在传送给服务器之前都保存在本地。这样，临时与服务器断开连接也不会导致数据丢失。proxy 配置文件中的参数 ProxyLocalBuffer 和 ProxyOfflineBuffer 控制数据在本地保存多久。

zabbix 代理 (Proxies) 是一个数据收集器。它不进行触发器计算，处理事件或发送报警信息。要了解代理 (Proxies) 的全部功能，查看下表

Function	Supported by proxy
----------	--------------------

Items	
Zabbix agent checks	Yes
Zabbix agent checks (active)	Yes ¹
Simple checks	Yes
Trapper items	Yes
SNMP checks	Yes
SNMP traps	Yes
IPMI checks	Yes
JMX checks	Yes
Log file monitoring	Yes
Internal checks	No
SSH checks	Yes
Telnet checks	Yes
External checks	Yes
Built-in web monitoring	Yes
Network discovery	Yes
Low-level discovery	Yes
Calculating triggers	No
Processing events	No
Sending alerts	No
Remote commands	No



为确保客户端代理（agent）实现连接代理服务器（不是 zabbix server）请求主动检测，代理服务器的 ip 地址必须出现在

客户代理（Proxies）的参数 ServerActive 的值中

配置

一旦你安装并配置了一个代理服务器，现在该在 zabbix 前端配置它了。

添加代理（Adding proxies）

要在 zabbix 前端配置代理，按如下步骤：

点击：高级配置 -> DM

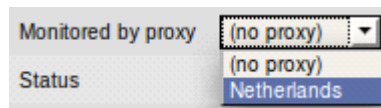
在右上角的下拉列表框中选择 proxies

点击 Create proxy （或已经存在的代理服务器名称）

参数	描述
proxy name	输入代理服务器名称。它必须跟代理服务器配置文件中参数 Hostname 的值一样。
proxy mode	选择代理服务器的模式。 Active - 代理服务器将主动连接 zabbix 服务器并请求配置数据 被动模式 - zabbix 服务器连接代理服务器
hosts	添加被代理服务器监控的主机

主机配置

你可以在主机配置表单中使用 Monitored by proxy 字段指定应该被代理服务器监控的单个主机。



本节参考 http://www.linuxmr.com/2012/zabbix2_0817/265.html

4. Zabbix 的高级使用-之自动化功能

使用 zabbix 的目的是让监控实现自动化,那么究竟如何才能达到自动化功能呢? 首先, zabbix 提供了主机自动发现功能,当客户端安装好之后, zabbix 有自动发现添加主机的功能,其次, zabbix 提供了对多变的监控项目自动发现监控,例如本身有 2 个网卡,新增加 2 个网卡,新增的 2 个网卡会自动监控。基于 zabbix 的这 2 个功能,我们可以做到对服务的自动化监控,从此告别手动添加监控项的痛苦,相信用 cacti 和 nagios 的都有添加监控到手抽筋的经历吧。

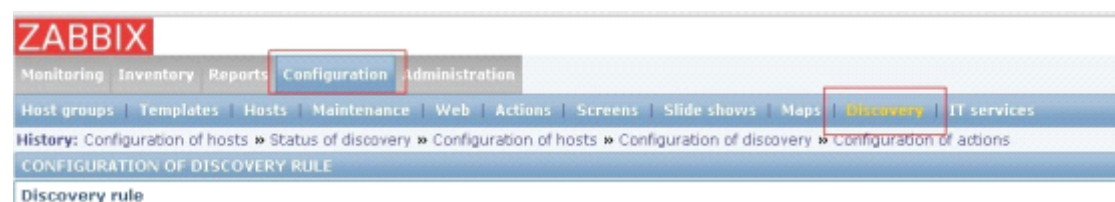
4.1 自动发现添加主机

功能介绍:

官方文档已经给出了很详细的步骤,在此演示一下过程

https://www.zabbix.com/documentation/2.0/manual/discovery/network_discovery/rule

创建自动发现规则



Create discovery rule

Discovery rule

Name: server 10.10.10.0

Discovery by proxy: No proxy

IP range: 10.10.10.1-254

Delay (in sec): 180

Checks: ICMP ping, Zabbix agent "system.uname", New

Device uniqueness criteria: IP address, Zabbix agent "system.uname"

Enabled: ☒

Save, Clone, Delete, Cancel

Names: 名称，可以写自己能代表服务功能的名称，便于识别

Discovery by proxy: 是否通过代理

IP range: ip 地址的范围，可以写一段地址，也可以写多段地址

Delay: 检测时间周期，注意这个值默认是 3600，即一个小时才能发现服务

Checks:检测命令，这里选择 ICMP, zabbix agentd 来检测

Device uniqueness criteria: 设备唯一的名称，此处采用 ip

Enabled: 发现功能是否激活

创建自动添加到相应模板规则

ZABBIX

Monitoring | Inventory | Reports | Configuration | Administration

Host groups | Templates | Hosts | Maintenance | Web | Actions | Screens | Slide shows | Maps | Discovery

History: Status of discovery » Configuration of hosts » Configuration of discovery » Configuration of actions » Configuration of discovery

CONFIGURATION OF ACTIONS

Actions

Displaying 1 to 3 of 3 found

Create action

Event source: Discovery

选择 Discovery,

Name 填写业务相关的 Action 名称

注意：Discovery rule 是添加前面定义的 Discovery rule 项目

此处是添加从自动发现规则里面发现的主机，包含这些规则，会进行下一步的操作

Operations

Action operations

Details

Add to host groups: Linux_servers

Link to templates: check

Link to templates: Template OS Linux

Action

Edit Remove

Edit Remove

Edit Remove

New

Save Clone Delete Cancel

ZABBIX

Help | Get support | Print | Profile | Logout

Monitoring Inventory Reports Configuration Administration

Dashboard Overview Web Latest data Triggers Events Graphs Screens Maps **Discovery** IT services

History: Configuration of discovery Dashboard Status of discovery Latest events Status of discovery

STATUS OF DISCOVERY

Discovery rules

Discovery rule: Server 10.10.10.0

Discovered hosts	Hostname	Uptime/Overtime	Actions
10.10.10.117	-	00:04:23	
10.10.10.118	-	04:12:22	
10.10.10.119	-	00:04:23	
10.10.10.120	-	00:04:23	
10.10.10.121	-	00:04:23	
10.10.10.122	-	00:04:22	
10.10.10.123	-	00:04:22	
10.10.10.124	-	00:04:22	
10.10.10.125	-	00:04:22	
10.10.10.126	-	00:04:22	
10.10.10.127	-	00:04:22	
10.10.10.128	-	00:04:22	
10.10.10.129	-	00:04:22	
10.10.10.130	-	00:04:22	
10.10.10.131	-	00:04:22	
10.10.10.132	-	00:04:22	

这里是对上面规则的进一步添加到应用，包含上面规则的，将会自动添加到分组，并用指定的模板。此处可以实现的功能，当我们发现主机包含某些规则后，可以用相关的分组，相关份额模板添加监控项。

需要注意的是，选择相关选项后，需点击 *add* 后，然后才可以保存。

4.2 通过自动发现实现动态监控

自动化运维之监控篇——利用 zabbix 自动发现功能实现批量 web url 监控

需求：

现在有大量 url 需要监控，形式如 <http://itnihao.blog.51cto.com>，要求 url 状态不为 200 即报警。

需求详细分析：

大量的 url，且 url 经常变化，现在监控用的是 zabbix，如果手动添加模板，会造成大量重复工作，造成人力财力的浪费，造成休息时间的浪费，得不偿失，如果利用脚本+mail，无法图形呈现

解决方案：

zabbix 有 *discovery* 功能，利用此功能，即可轻松解决此问题

zabbix 客户端配置


```

1. #####cat /etc/zabbix/zabbix_agentd.conf|grep -v "^#"|grep -v "^$"#####
##此处省略 N 多信息,
Include=/etc/zabbix/zabbix_agentd.conf.d/    #配置文件路径
UnsafeUserParameters=1                        #自定义 key
#####自动发现脚本编写
#####cat /etc/zabbix/scripts/web_site_code_status#####
#!/bin/bash
|
#function:monitor tcp connect status from zabbix
# License: GPL
# mail:itnihao@qq.com
# version: 1.0 date:2012-12-09
source /etc/bashrc >/dev/null 2>&1
source /etc/profile >/dev/null 2>&1
|
#/usr/bin/curl -o /dev/null -s -w %{http_code} http://$1/
WEB_SITE_discovery () {
WEB_SITE=$(cat WEB1.txt|grep -v "^#")
    printf '\n'
    printf "\t"data":[\n
for((i=0;i<${#WEB_SITE[@]};++i))
{
    num=$(echo ${${#WEB_SITE[@]}-1})
    ij [ "$i" != ${num} ];
        then
            printf "\t\t\t\n"
            printf "\t\t\t\"${SITENAME}\".\"${WEB_SITE[$i]}\"},\n"
        else
            printf "\t\t\t\n"
            printf "\t\t\t\"${SITENAME}\".\"${WEB_SITE[$num]}\"}"}\n"
        fi
    }
}
web_site_code () {
/usr/bin/curl -o /dev/null -s -w %{http_code} http://$1/
}
case "$1" in
web_site_discovery)
WEB_SITE_discovery
;;
web_site_code)
web_site_code $2
|
;;

```

```
*)
echo "Usage:$0 {web_site_discovery|web_site_code [URL]}"
;;
esac
```

客户端配置文件

```
#####cat /etc/zabbix/zabbix_agentd.conf.d/web_site_discovery.conf #####
UserParameter=web.site.discovery,/etc/zabbix/scripts/web_site_code_status web_site_discovery
UserParameter=web.site.code[*],/etc/zabbix/scripts/web_site_code_status web_site_code $1
```

域名如下

```
##### cat /etc/zabbix/scripts/WEB.txt#####
```

www.qq.com

www.baidu.com

www.sina.com.cn

测试:

```
zabbix_get -s 127.0.0.1 -k web.site.discovery
```

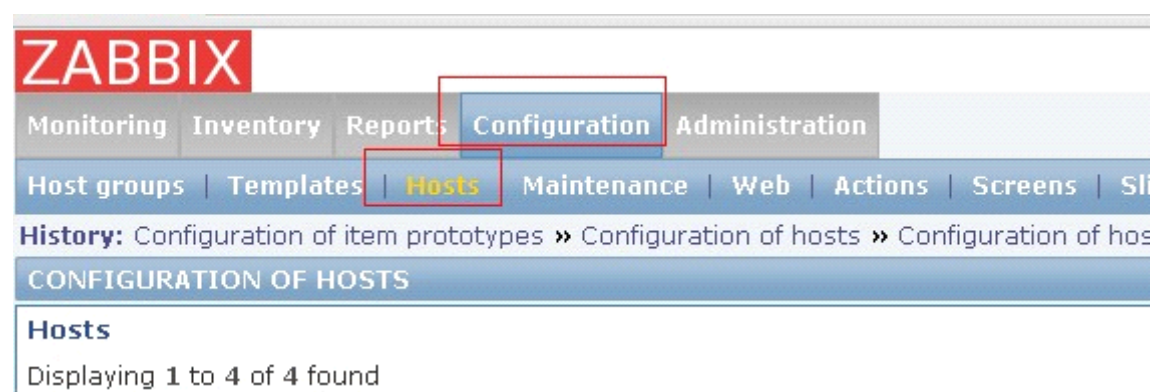
```
{
  "data": [
    {
      "#SITENAME": "www.qq.com",
    },
    {
      "#SITENAME": "www.baidu.com",
    },
    {
      "#SITENAME": "www.sina.com.cn"
    }
  ]
}
```

```
zabbix_get -s 127.0.0.1 -k web.site.code[www.qq.com]
```

此时返回状态为 200

至此，脚本，客户端配置文件 OK

接下来是 web 的添加



Search

Create application

Group Templates Host all

Show

Host: web_monitor Monitored Applications (1) It

Host web_monitor

Name web_status_code_monitor

Save Clone Delete Cancel

Discovery (1)

Create discovery rule

Name	<input type="text" value="web.site.discovery"/>								
Type	<input type="text" value="Zabbix agent"/>								
Key	<input type="text" value="web.site.discovery"/>								
Host interface	<input type="text" value="10.0.0.1 : 10050"/>								
Update interval (in sec)	<input type="text" value="30"/>								
Flexible intervals	<table> <thead> <tr> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td colspan="3">No flexible intervals defined.</td> </tr> </tbody> </table>			Interval	Period	Action	No flexible intervals defined.		
Interval	Period	Action							
No flexible intervals defined.									
New flexible interval	Interval (in sec)	<input type="text" value="50"/>	Period	<input type="text" value="1-7,00:00-24:00"/>	<input type="button" value="Add"/>				
resources period (in days)	<input type="text" value="1"/>								
Filter	Macro	<input type="text" value="{#SITENAME}"/>	Regexp	<input type="text" value=""/>					
Description	<div></div>								
Status	<input type="text" value="Enabled"/>								

Zabbix 2.0.2 Copyright 2001-2012 by Zabbix SIA

Name

Type

Key

Host interface : 10050

Type of information

Data type

Units

Use custom multiplier ☐

Update interval (in sec)

Flexible intervals

Interval	Period	Action
No flexible intervals defined.		

New flexible interval

Interval (in sec)	Period	Action
<input type="text" value="50"/>	<input type="text" value="1-7,00:00-24:00"/>	<input type="button" value="Add"/>

Keep history (in days)

Keep trends (in days)

Store value

Show value [show value mappings](#)

New application

Applications

Displaying 1 to 3 of 3 found

< Host list Host: web_monitor Monitored < Discovery list Discovery: web.site.discovery Item prototypes (1) Trigger prototypes (3) Graph prototypes (1)

Severity	Name	Expression	Status
Disaster	web.site.code[{#SITENAME}] is not 200	(127.0.0.1:web.site.code[{#SITENAME}])last(#{3,15m})#200&(127.0.0.1:web.site.code[{#SITENAME}])last(0)#200	Enabled
High	web.site.code[{#SITENAME}] is not 200	(127.0.0.1:web.site.code[{#SITENAME}])last(#{3,10m})#200&(127.0.0.1:web.site.code[{#SITENAME}])last(0)#200	Enabled
Average	web.site.code[{#SITENAME}] is not 200	(127.0.0.1:web.site.code[{#SITENAME}])last(#{3,2m})#200&(127.0.0.1:web.site.code[{#SITENAME}])last(0)#200	Enabled

Enable selected Go (0)

报警级别的设置：3个报警级别

« Host list Host: web_monitor Monitored « Discovery list Discovery: web_site_discovery Item prototypes (1) Trigger prototypes (3) Graph prototypes (1)

Graph Preview

Name: web.site.code on {#SITENAME}

Width: 900

Height: 200

Graph type: Normal

Show legend: ☒

Show working time: ☒

Show triggers: ☒

Percentile line (left): ☐

Percentile line (right): ☐

Y axis MIN value: Calculated

Y axis MAX value: Calculated

Items	Name	Function	Draw style	Y axis side	Colour	Action
1:	web_monitor: web.site.code_QN {#SITENAME}	avg	Line	Left	C80000	Remove

Add Add prototype

nts | Graphs | Screens | Maps | Discovery | IT services

hs » Configuration of hosts » Dashboard » Custom graphs

Group: Host: web_monitor Graph:

Filter

No graphs defined.

right 2001-2012 by Zabbix SIA

not selected
web.site.code on a
web.site.code on a
web.site.code on a
web.site.code on b
web.site.code on b
web.site.code on c
web.site.code on c
web.site.code on c
web.site.code on c
web.site.code on d
web.site.code on d
web.site.code on f
web.site.code on k
web.site.code on l
web.site.code on n
web.site.code on p
web.site.code on r
web.site.code on r

利用以上方法，哪怕是 1000 个 url 监控，也是分分钟的事情

总结：工欲善其事，必先利其器

参考文档：https://www.zabbix.com/documentation/2.0/manual/discovery/low_level_discovery

此外，还可以参考我写的文档：<http://code.google.com/p/auto-task-pe/downloads/list>

5. 批量更新参考文档

[easy Update on Custom Scripts](#)

<http://zabbixzone.com/zabbix/easy-update-on-custom-scripts/>

Every time that you need to add or change an UserParameter on zabbix_agentd.conf you need to restart the agent. It's easy if you have less than 10 servers, but could be a trouble if you have more than 30 servers.

Automation Tools like [CFEngine](#) and [Puppet](#) do this work well done, but unfortunately they aren't used in many companies.

But it's possible using a **Dynamic UserParameter**:

1) add two lines on **zabbix_agentd.conf** and **restart the agent**:

```
UserParameter=custom.getvalue[*],/etc/zabbix/zabbix_agentd/custom-getvalue $1 $2 $3 $4 $5
UserParameter=custom.updatescript,/etc/zabbix/zabbix_agentd/custom-updatescript
```

2) create the script file **/etc/zabbix/zabbix_agentd/custom_getvalue**

```
#!/bin/bash
ACTION=$1
PARAM1=$2
PARAM2=$3
PARAM3=$4
PARAM4=$5
case ${ACTION} in
# key: custom.getvalue[samplescript1,arg]
samplescript1)
    /bin/echo This is a test - ${PARAM1}
;;
# key: custom.getvalue[samplescript2,arg1,arg2]
```



```
samplescript2)
    /bin/echo This is another test - ${PARAM1} ${PARAM2}
;;
*)
    /bin/echo ZBX_NOTSUPPORTED
;;
esac
```

Don't forget to set execute permission:

```
1 chmod 0755 /etc/zabbix/zabbix-agentd/custom-getvalue
```

Now when you need create a new script, you must update this script and restart is unnecessary.

Remote Updates

Finally, let's create a script to update **custom_getvalue** script remotely. Before make sure to publish your **custom_getvalue** script on a WebServer.

Create the file **/etc/zabbix/zabbix-agentd/custom_updatescript**:

```
1 #!/bin/bash
2 /usr/bin/wget "http://yourdomain.com/zabbix/custom-getvalue" -O
   /etc/zabbix/zabbix-agentd/custom-getvalue -o /dev/null
```

Don't forget to set execute permission:

```
1 chmod 0755 /etc/zabbix/zabbix-agentd/custom-updatescript
```

Remote updates can be done from **Zabbix Server/Proxy** using **zabbix_get** utility:

```
1 zabbix_get -s hostserver.yourdomain.com -k custom.updatescript
```

It's a contribution from **laneovcc**:

in my way i config zabbix-agent.conf to include a UserParameter.conf then use
system.run[wget http://server/UserParameter.conf -O
/path/to/UserParameter.conf] to update the UserParameter.conf and
system.run[services zabbix-agentd restart] to restart the agent