

# ZABBIX

## 使用手册

崔之龙

QQ: 852874966

cuizhilong1215@126.com

---

<b>1. PART1.ZABBIX 简介 .....</b>	<b>1</b>
1.1 ZABBIX 简介 .....	1
1.2 ZABBIX 功能 .....	1
1.3 优劣势 .....	1
<b>2. PART2.安装部署 .....</b>	<b>2</b>
2.1 服务端环境准备 .....	2
2.2 数据库准备 .....	2
2.3 编译安装 .....	3
2.4 配置文件及 WEB 前端文件修改.....	3
2.5 WEB 前端安装配置 .....	3
2.6 启动 SERVER.....	4
2.7 在 HOSTS 上配置 AGENT.....	5
<b>3. PART3.配置使用 .....</b>	<b>6</b>
3.1 添加 HOSTS .....	6
3.2 添加 ITEMS.....	7
3.3 添加 TRIGGERS.....	8
3.4 添加 ACTIONS .....	8
3.5 添加 MEDIAS.....	9
3.6 添加 USERS .....	10
3.7 添加 WEB MONITORINGS .....	10
3.8 添加 GRAPHS.....	13
3.9 添加 SCREENS .....	14
3.10 添加 MAPS .....	15
3.11 添加 MYSQL 监控 .....	15
3.12 添加 ORACLE 监控.....	16
3.13 添加 SNMP 监控 .....	18
3.14 添加通用协议监控 .....	18
3.15 添加自定义监控 .....	18
3.16 添加 TEMPLATES .....	19
3.17 添加 REPORTS（定制报表） .....	19
3.18 添加 MACROS .....	20
3.19 添加自动发现设备 .....	21
3.20 添加 INVENTORY.....	21
3.21 EXPORT/IMPORT XML.....	22
3.22 MAINTENANCE（维护时间） .....	22
<b>4. PART4.FAQ.....</b>	<b>24</b>

# 图表目录

图表 1.....	4
图表 2.....	6
图表 3.....	7
图表 4.....	7
图表 5.....	8
图表 6.....	9
图表 7.....	9
图表 8.....	10
图表 9.....	10
图表 10.....	11
图表 11.....	12
图表 12.....	12
图表 13.....	13
图表 14.....	13
图表 15.....	14
图表 16.....	14
图表 17.....	15
图表 18.....	15
图表 19.....	17
图表 20.....	18
图表 21.....	18
图表 22.....	19
图表 23.....	19
图表 24.....	20
图表 25.....	20
图表 26.....	21
图表 27.....	22
图表 28.....	23
图表 29.....	23
图表 30.....	24

# 1. Part1.Zabbix 简介

## 1.1 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 1.8 官方文档：<http://www.zabbix.com/documentation/1.8/start>

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 MySQL 数据库，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，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

## 1.2 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

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，可以自己开发完善各类监控

## 1.3 优劣势

优点：

开源，无软件成本投入

Server 对设备性能要求低（实际测试环境：虚拟机 Redhat EL AS5，2G CPU 1G 内存，监控 5 台设备，CPU 使用率基本保持在 10% 以下，内存剩余 400M 以上）

支持设备多

支持分布式集中管理

开放式接口，扩展性强

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:

全英文, 界面不友好

无厂家支持, 出现问题解决比较麻烦

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. Part2.安装部署

### 2.1 服务端环境准备

Zabbix Server 需要运行在 CentOS、RedHat Linux、Debian 等 Linux 系统上, 这里以 RHEL AS5 作为部署环境。

Root 用户安装必须的包, 建议配置好 yum, 通过 yum 安装下列包, 解决包的依赖关系。

LAMP 环境

```
#yum install mysql-server httpd php
```

其他需要用到的包:

```
#yum install mysql-devel gcc net-snmp-devel curl-devel perl-DBI php-gd php-mysql php-bcmath php-mbstring php-xml
```

下载最新的 Zabbix 安装包(官网:<http://www.zabbix.com>)到本地, 解压

```
#tar zxvf zabbix-1.8.1.tar.gz
```

增加 zabbix 用户和组

```
#groupadd zabbix
#useradd -g zabbix -m zabbix
```

### 2.2 数据库准备

启动 MySQL 数据库:

```
#service mysqld start
```

修改 MySQL root 用户密码(默认密码为空)

```
#mysqladmin -uroot password root
```

测试能否正常登陆数据库

```
#mysql -uroot -proot
```

创建 Zabbix 数据库

```
mysql> create database zabbix character set utf8;
mysql> grant all on zabbix.* TO zabbix@'localhost' identified by 'zabbixpass';
mysql> flush privileges;
```

导入数据库 sql 脚本

```
#cd zabbix-1.8.1
# cat ./create/schema/mysql.sql |mysql -uroot -proot zabbix
```

```
# cat ./create/data/data.sql |mysql -uroot -proot zabbix
# cat ./create/data/images_mysql.sql |mysql -uroot -proot zabbix
```

## 2.3 编译安装

配置编译, prefix 是安装后程序目录

```
# ./configure --with-mysql --with-net-snmp --with-libcurl --enable-server --enable-agent --enable-proxy --prefix=/usr/local/zabbix
# ./configure --with-mysql=/usr/local/mysql/bin/mysql_config --with-net-snmp --with-libcurl --enable-server --enable-agent --enable-proxy --prefix=/usr/local/zabbix //mysql编译安装
# make
# make install
```

## 2.4 配置文件及 web 前端文件修改

添加服务端口,添加后如下

```
# grep zabbix /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
```

添加配置文件

```
# mkdir -p /etc/zabbix
# cp ./misc/conf/* /etc/zabbix
# chown -R zabbix:zabbix /etc/zabbix
```

修改 Server 配置文件

基本不用修改,用默认配置即可,只需修改一项 DBPassword=密码

```
# vi /etc/zabbix/zabbix_server.conf
```

修改 Agentd 配置文件,更改 HOSTNAME 为本机的 hostname 就好了

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

添加 web 前端 php 文件

```
# cd frontends/
# cp -rf php /var/www/html/
# cd /var/www/html
# mv php zabbix
# chown -R zabbix:zabbix zabbix
```

## 2.5 web 前端安装配置

修改 php 相关参数

# vi /etc/php.ini 找到如下几项,改成下面的值,前面有;号的要删掉

```
max_execution_time = 300
max_input_time=300
```

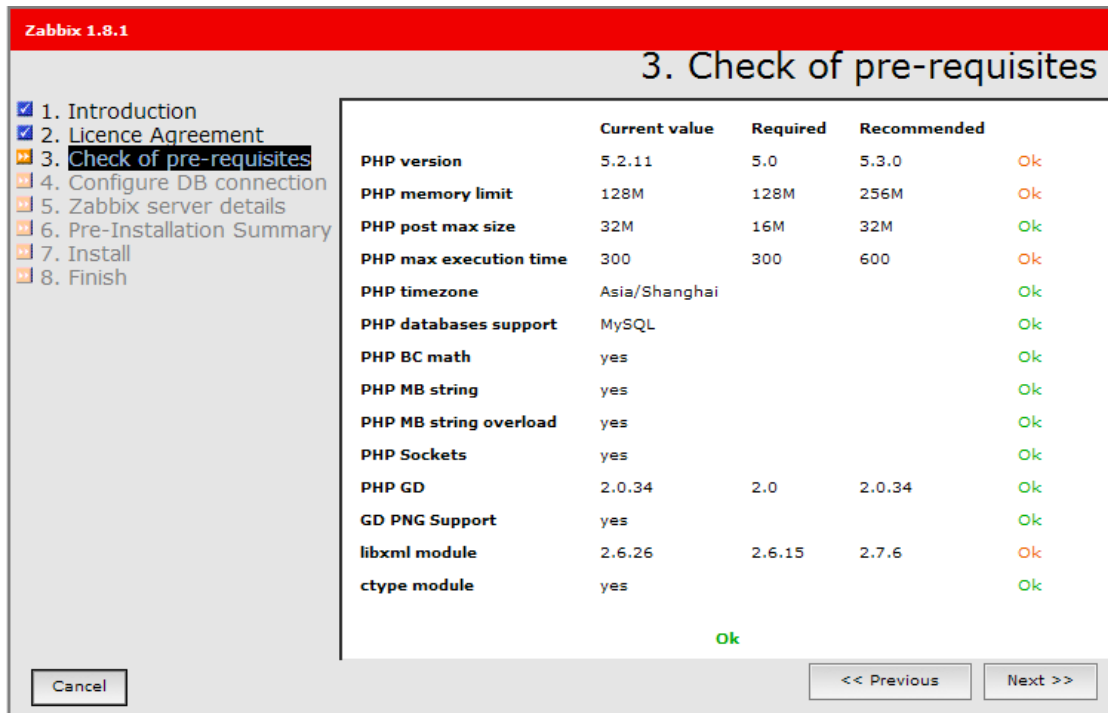
```
memory_limit = 128M
date.timezone = Asia/Shanghai
post_max_size = 32M
mbstring.func_overload = 2
```

重启 apache

```
#service httpd restart
```

在本地浏览器上访问 Zabbix Server 地址开始 web 前端配置,http://ServerIP/zabbix

按提示一步步 next, 其中 Step 3 Check of pre-requisites 必须全部项目 OK 后才能继续配置, 如有提示 fail, 去 server 上检查是否安装这个包或配置是否按上述更改。



图表 1

Step 4 填入 Zabbix Server 登陆 mysql 的用户和密码

Step 7 按提示下载配置文件到 Server 的/var/www/html/zabbix/conf 下, 名字一定要是 zabbix.conf.php

配置完成后, 出现登陆界面, 默认的用户为:admin,密码为:zabbix

## 2.6 启动 Server

安装完成后的 server 程序在/usr/local/zabbix/sbin/目录下, 可以直接启动

```
#/usr/local/zabbix/sbin/zabbix_server
```

为了便于启停 Server, 我们做一些小配置

复制控制程序

```
# cp ./misc/init.d/redhat/zabbix_* /etc/init.d/
```

修改配置目录, 将如下两个文件中的 BASEDIR=改为/usr/local/zabbix, 下面两个配置中的 bin 改为 sbin, ZABBIX\_AGENTD, ZABBIX\_SUCKERD ???

```
# vi /etc/init.d/zabbix_server_ctl
```

```
# vi /etc/init.d/zabbix_agentd_ctl
```

加入开机启动

```
# echo '/etc/init.d/zabbix_server_ctl start &' >>/etc/rc.local
# echo '/etc/init.d/zabbix_agentd_ctl start &' >>/etc/rc.local
```

创建快捷方式

```
# ln -s /etc/init.d/zabbix_server_ctl /usr/bin/zabbix_server_ctl
# ln -s /etc/init.d/zabbix_agentd_ctl /usr/bin/zabbix_agentd_ctl
```

启动 Server 和 Agentd(可以通过 start stop restart 来操作)

```
# zabbix_agentd_ctl start
# zabbix_server_ctl start
```

检查启动是否正常,查看进程是否起来,分 server 和 agentd

```
#ps -ef | grep zabbix
```

如果进程没起来, 可以查看对应的日志错误, 默认在/tmp/zabbix-\*.log  
至此 Zabbix Server 的安装已完成

## 2.7 在 hosts 上配置 agent

Zabbix 需要在被监控的 host 上安装 agent, 在 zabbix 官网下载相应平台的 agent 包到各被监控端, 按如下方式在被监控上运行 agent。(注: 官网下载的 agent 包里没有运行 agentd 所必须的配置文件, 可以从 zabbix server 上拷贝一份 zabbix\_agentd.conf, 这个 conf 文件是所有平台都通用的)

Linux、UX:

Linux、UX 上的 agent 是编译好了的, 不用安装

```
#useradd zabbix
#cd /usr/local/zabbix
#tar zxvf zabbix_agents_1.8.linux2_6.i386.tar.gz
```

取一份 server 上的 zabbix\_agentd.conf 到本地, 然后按说明修改, 一般我们只需修改如下 5 个参数:

```
Server=Zabbix Server IP
Hostname=本机名
LogFile=本机agentd日志保存文件
SourceIP=本机IP
ListenPort=10050
```

修改完成后, 运行 agentd

```
#/usr/local/zabbix/sbin/zabbix_agentd -c /usr/local/zabbix/zabbix_agentd.conf
&
#ps -ef | grep zabbix_agentd
```

如果进程没起来, 可查看 agentd.log 日志来排查问题。(注: 启动时建议用全路径, 否则可能会出错)

加入开机运行

```
# echo '/usr/local/zabbix/sbin/zabbix_agentd -c /usr/local/zabbix/zabbix_agentd.conf &' >> /etc/rc.local
```



Windows:

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

### 3. Part3.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)

#### 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

图表 2

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

### 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。

Log	Description	Triggers	Key	Interval	History	Trends	Type	Status	Applications	Error
<input type="checkbox"/>	Template Linux:Buffers memory	Triggers (0)	vm.memory.size[buffers]	30	7	365	Zabbix agent	Active	Availability, Memory	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Cached memory	Triggers (0)	vm.memory.size[cached]	30	7	365	Zabbix agent	Active	Availability, Memory	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /usr/sbin/sshd	Triggers (1)	vfs.file.cksum[/usr/sbin/sshd]	600	7	365	Zabbix agent	Active	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /usr/bin/ssh	Triggers (1)	vfs.file.cksum[/usr/bin/ssh]	600	7	365	Zabbix agent	Active	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /vmlinuz	Triggers (1)	vfs.file.cksum[/vmlinuz]	600	7	365	Zabbix agent	Not supported	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /etc/services	Triggers (1)	vfs.file.cksum[/etc/services]	600	7	365	Zabbix agent	Active	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /etc/inetd.conf	Triggers (1)	vfs.file.cksum[/etc/inetd.conf]	600	7	365	Zabbix agent	Not supported	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Checksum of /etc/passwd	Triggers (1)	vfs.file.cksum[/etc/passwd]	600	7	365	Zabbix agent	Active	Integrity	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:CPU system time (avg1)	Triggers (0)	system.cpu.util[,system,avg1]	10	90	365	Zabbix agent	Active	CPU, Performance	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:CPU nice time (avg1)	Triggers (0)	system.cpu.util[,nice,avg1]	10	90	365	Zabbix agent	Active	CPU, Performance	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:CPU idle time (avg1)	Triggers (0)	system.cpu.util[,idle,avg1]	10	90	365	Zabbix agent	Active	CPU, Performance	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:CPU iowait time (avg1)	Triggers (0)	system.cpu.util[,iowait,avg1]	10	90	365	Zabbix agent	Active	CPU, Performance	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:CPU user time (avg1)	Triggers (0)	system.cpu.util[,user,avg1]	10	90	365	Zabbix agent	Active	CPU, Performance	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Email (SMTP) server is running	Triggers (1)	net.tcp.service[smtp]	60	7	365	Zabbix agent	Active	Services	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Template Linux:Free disk space on /usr	Triggers (0)	vfs.fs.size[/usr,free]	30	7	365	Zabbix agent	Active	Availability, Filesystem	<input checked="" type="checkbox"/>

图表 3

通过点击具体 item 名字可以修改已有监控项的属性，点击 Satus 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

Item '724AS5'

Host: 724AS5

Description: Check MySQL Status

Type: Zabbix agent

Key: mysql.alive

Type of information: Numeric (unsigned)

Data type: Decimal

Units:

Use multiplier: Do not use

Update interval (in sec): 30

Flexible intervals (sec): No flexible intervals

New flexible interval: Delay 50 Period 1:7:00:00-23:59

Keep history (in days): 90

Keep trends (in days): 365

Status: Active

Store value: As is

Show value throw map: As is

New application:

Applications: Memory, Network, OS, Performance, Processes, Services

Buttons: Save, Cancel

图表 4

按提示逐项填入相关信息即可，其中 key 是 zabbix 已经自带的取值方法，Application 类似于 host groups，是 item 的组。（item key 也可以自定义，后面会讲到如何自定义监控项）

Zabbix 自带非常多的监控采集项及方法，基本能满足当前所有的监控功能，这些都包含在 item 的 item key 中了，更多的 type 和 key 的解释请参考：

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

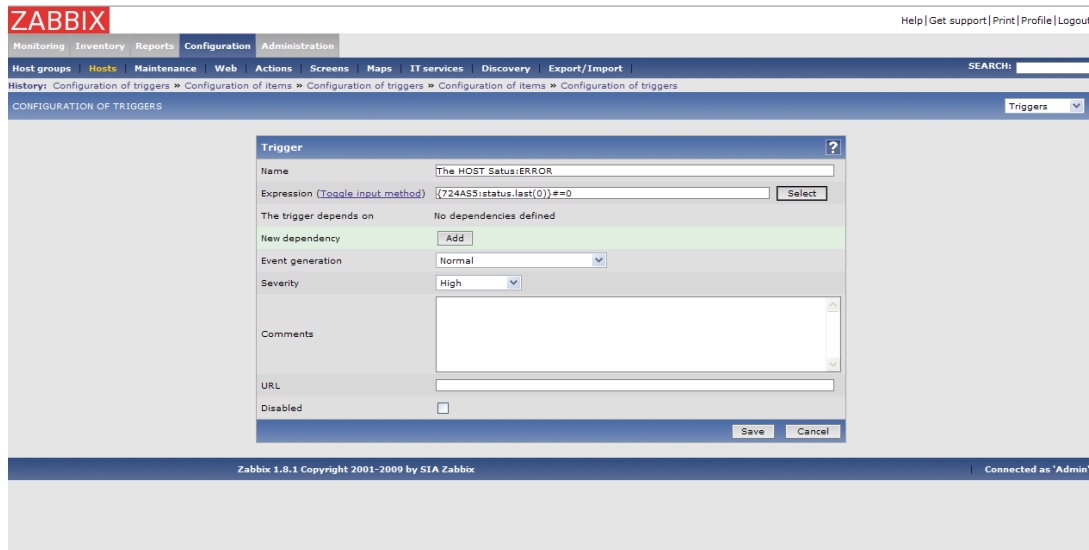
### 3.3 添加 Triggers

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。

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



图表 5

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。

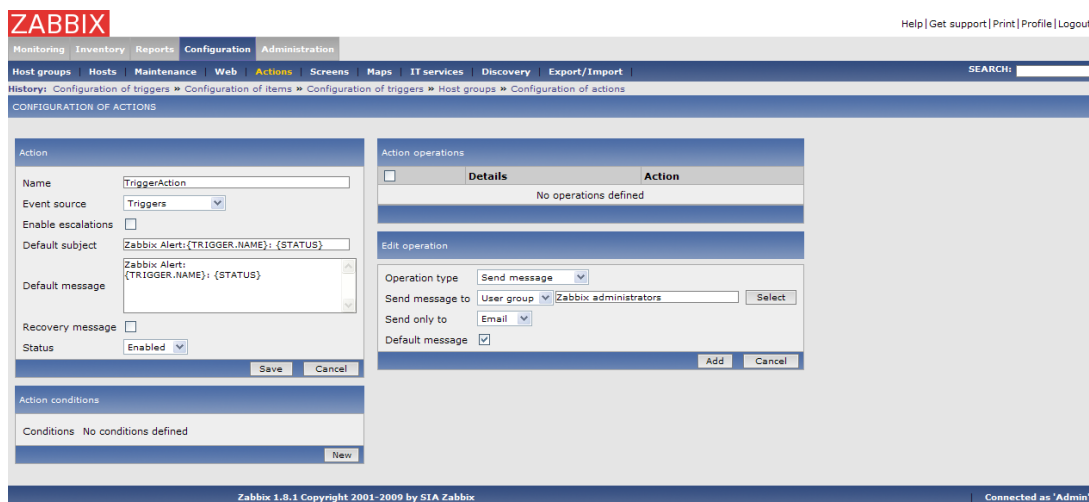
Zabbix 提供多种 trigger 触发方式供选择，常用的我们可以选择 last value (最近一次采集值),或是选择 maximal value for period of time (一段时间内的最大值),等等。可以根据实际需要来设定触发方式。更多的解释请参考：

<http://www.zabbix.com/documentation/1.8/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



图表 6

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/1.8/manual/config/macros>

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

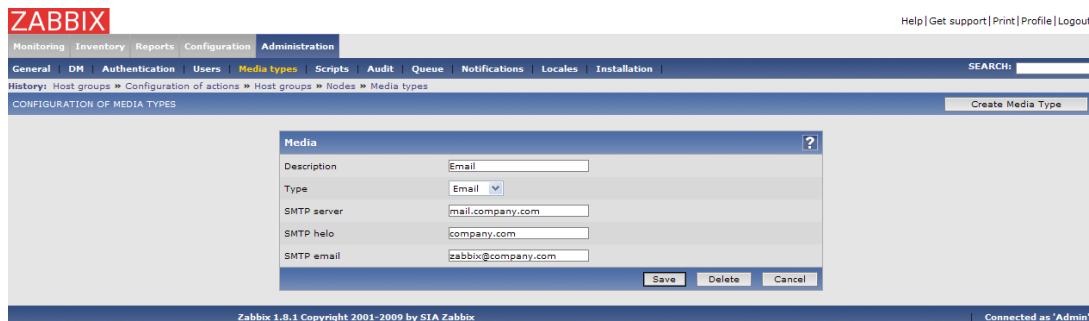
Conditions: trigger 产生的条件, 条件可以多选

Operation: 选择 media 及 user

### 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。



图表 7

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。

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。

### 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。

图表 8

### 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			<a href="#">Down</a>
<input type="checkbox"/> ConfigurationPages	15 sec	http://10.1.50.60/zabbix/config.php	CONFIGURATION		<a href="#">Up</a>

图表 9

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

**Name:** scenario 的名字

**Basic authentication:** 鉴权

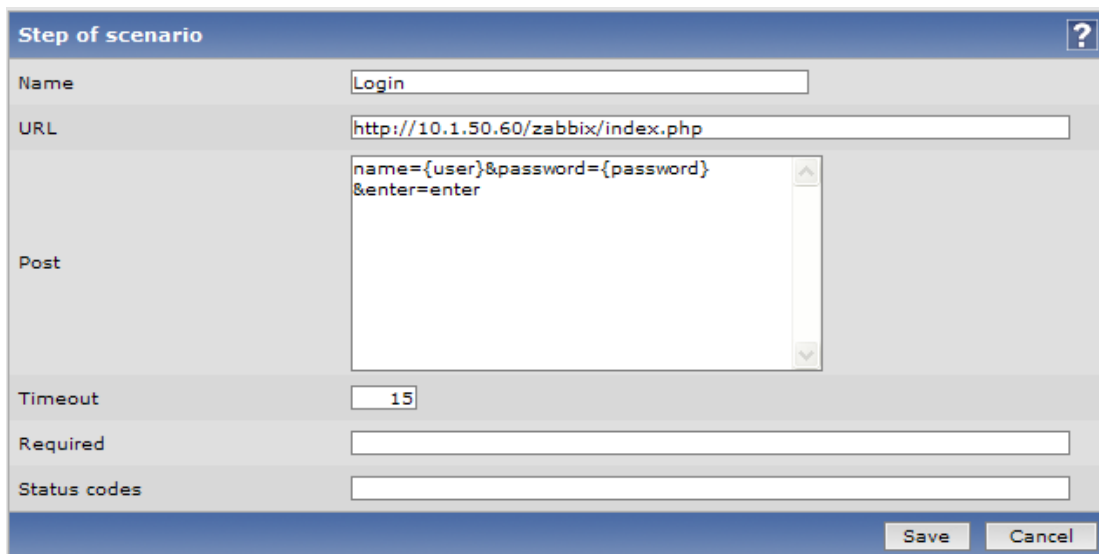
**Update interval:** 监控频率, s 为单位

**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



Name	Login
URL	http://10.1.50.60/zabbix/index.php
Post	name={user}&password={password}&enter=enter
Timeout	15
Required	
Status codes	

图表 10

**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

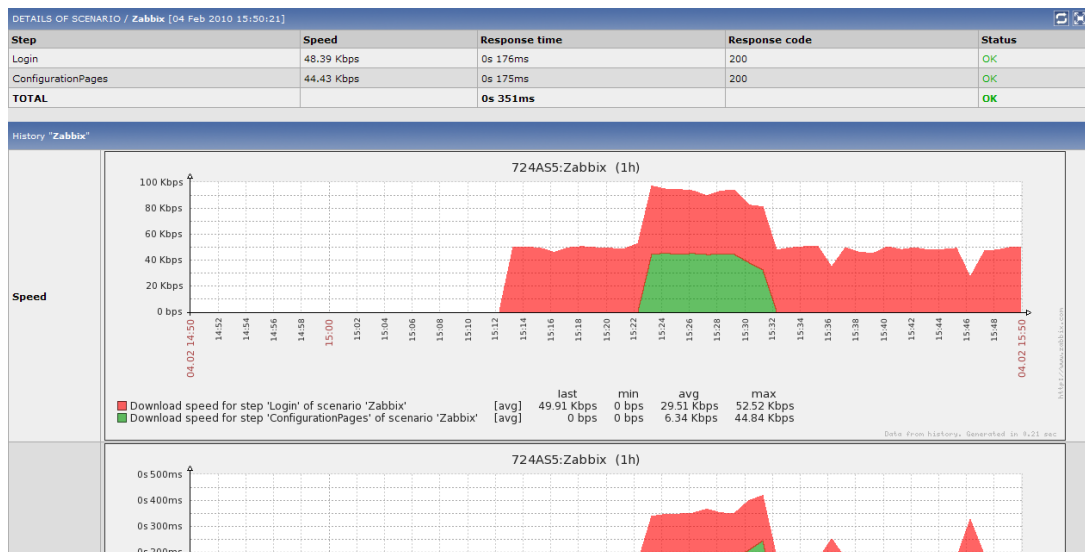


图 表 11

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)

ITEMS	Type	Type of information	Status
Buffers memory	Zabbix agent	Numeric (unsigned)	Active
Cached memory	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/inetd.conf	Zabbix agent	Numeric (unsigned)	Not supported
Checksum of /etc/passwd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /etc/services	Zabbix agent	Numeric (unsigned)	Active
Checksum of /usr/bin/ssh	Zabbix agent	Numeric (unsigned)	Active
Checksum of /usr/sbin/sshd	Zabbix agent	Numeric (unsigned)	Active
Checksum of /ymlinux	Zabbix agent	Numeric (unsigned)	Not supported
CPU idle time (avg1)	Zabbix agent	Numeric (float)	Active
CPU iowait time (avg1)	Zabbix agent	Numeric (float)	Active
CPU nice time (avg1)	Zabbix agent	Numeric (float)	Active
CPU system time (avg1)	Zabbix agent	Numeric (float)	Active
CPU user 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 12

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

The screenshot shows the 'Trigger' configuration window in Zabbix. The 'Name' field is 'ZabbixWeb Login failed'. The 'Expression (Toggle input method)' field contains the Zabbix expression: `{724AS5:web.test.rspcode[Zabbix,Login].last(0)}>399`. The 'The trigger depends on' field is 'No dependencies defined'. The 'Event generation' dropdown is set to 'Normal'. The 'Severity' dropdown is set to 'High'. The 'Comments' field contains the text 'ZabbixWeb Login failed'. The 'URL' field is empty. The 'Disabled' checkbox is unchecked. At the bottom right, there are 'Save' and 'Cancel' buttons.

图表 13

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。

The screenshot shows the 'Trigger' configuration window in Zabbix. The 'Name' field is 'Scenario zabbix run failed'. The 'Expression (Toggle input method)' field contains the Zabbix expression: `{724AS5:web.test.fail[Zabbix].last(0)}#b`. The 'The trigger depends on' field is 'No dependencies defined'. The 'Event generation' dropdown is set to 'Normal'. The 'Severity' dropdown is set to 'High'. The 'Comments' field contains the text 'Scenario zabbix run failed'. The 'URL' field is empty. The 'Disabled' checkbox is unchecked. At the bottom right, there are 'Save' and 'Cancel' buttons.

图表 14

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。



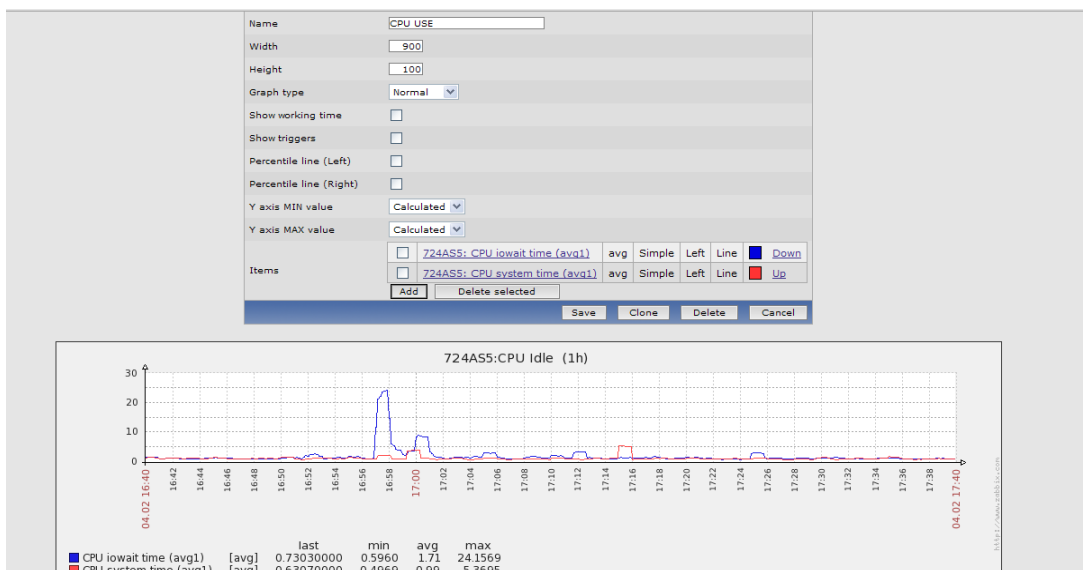


图 表 15

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

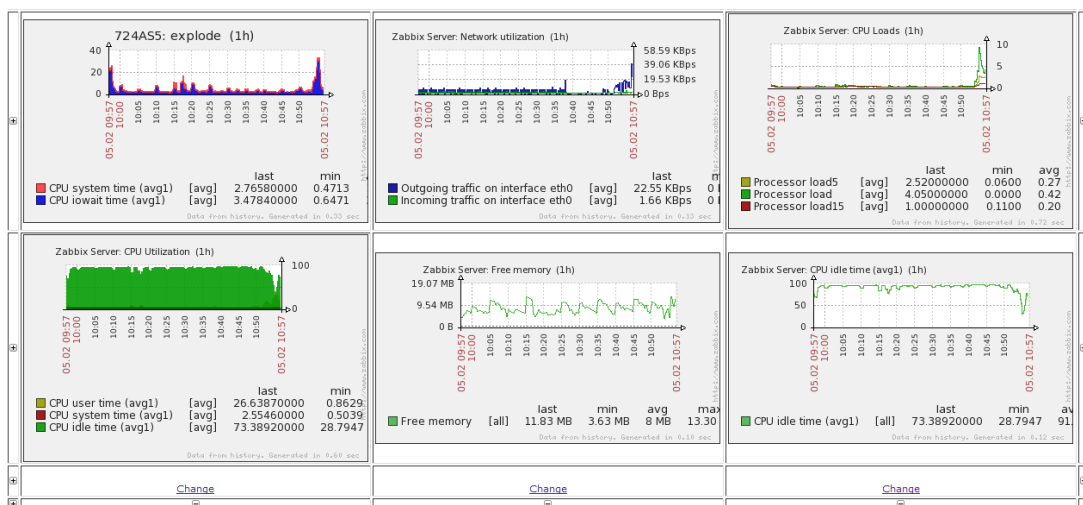


图 表 16

通过 monitoring-screen, 可以查看之前配置好的信息。

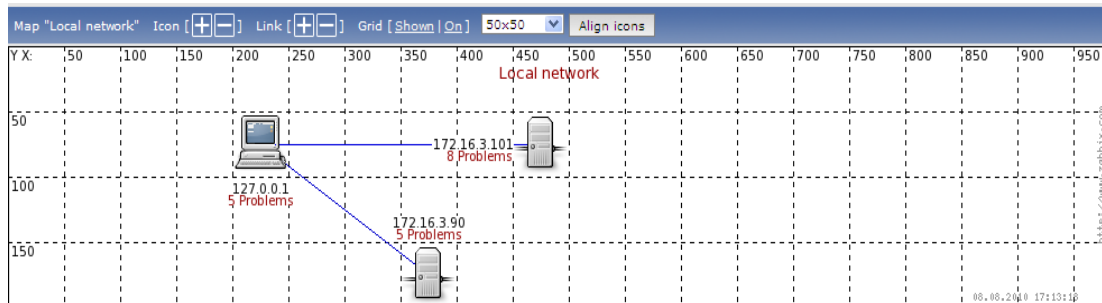
### 3.10 添加 Maps

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 create maps 点 save 保存。

图表 17

然后编辑保存好的 maps



图表 18

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

Agent 的配置文件上默认就有通过 mysqladmin 工具取 MySQL 数据库监控信息的配置, 我们只需更改 MySQL 所在 host 上的 agentd.conf 文件, 将文件最后的所有关于 mysql 的 UserParameter 前的 #号去掉, 更改登陆 mysql 的用户和密码即可。

例如: 修改后其中一条监控数据库状态如下:

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

修改后重启 host 上的 agentd, 使配置文件生效。

## 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\\_replication](http://www.zabbix.com/wiki/howto/monitor/db/mysql/extensive_mysql_monitoring_including_replication)

## 3.12 添加 Oracle 监控

Oracle 监控也参考 zabbix wiki 上如下的方法（调用 zabora 工具）：

<http://www.zabbix.com/wiki/howto/monitor/db/oracle/oracle>

下载 oracle 监控程序 zabora 到 oracle 所在的主机上，修改所在 host 的 agentd.conf，添加监控项，按如下格式：

```
UserParameter=oracle.name,DIR/zabora pattern sid
```

监控 oracle 数据库状态：

```
UserParameter=oracle.checkactive,/usr/local/zabora checkactive OracleSID
```

监控 oracle 数据库连接用户总数

```
UserParameter=oracle.usercount,/usr/local/zabora usercount OracleSID
```

更多 zabora 的监控项参考如下：

```
zabora version: 1.5
```

usage:

```
zabora checkactive [SID]    -- Check Instance is active and open.
zabora usercount [SID]     -- Count of users connected to Oracle.
zabora activeusercount [SID] -- Count of active users.
zabora dbsize [SID]        -- Size of user data (without temp).
zabora dbfilesize [SID]    -- Size of all datafiles.
zabora version [SID]       -- Oracle version (Banner).
zabora dsksortratio [SID]  -- Disk sorts ratio.
zabora rcachehit [SID]     -- Read Cache hit ratio.
zabora uptime [SID]       -- Instance Uptime (seconds).
zabora commits [SID]      -- User Commits.
zabora rollbacks [SID]    -- User Rollbacks.
zabora deadlocks [SID]    -- Deadlocks.
zabora redowrites [SID]   -- Redo Writes.
zabora tblscans [SID]     -- Table scans (long tables).
zabora tblrowsscans [SID]  -- Table scan rows gotten.
zabora indexffs [SID]     -- Index fast full scans (full).
zabora hparsratio [SID]   -- Hard parse ratio.
```

## ZABBIX 使用手册

```
zabora netsent [SID]      -- Bytes sent via SQL*Net to client.
zabora netresv [SID]     -- Bytes received via SQL*Net from client.
zabora netroundtrips [SID] -- SQL*Net roundtrips to/from client.
zabora logonscurrent [SID] -- Logons current.
zabora lastarclog [SID]  -- Last archived log sequence.
zabora lastapplarclg [SID] -- Last applied archive log (at standby).Next items requires [timed_statistics = true].
zabora freebufwaits [SID] -- free buffer waits.
zabora bufbusywaits [SID] -- buffer busy waits.
zabora logswcompletion [SID] -- log file switch completion.
zabora logfilesync [SID] -- log file sync.
zabora logprllwrite [SID] -- log file parallel write.
zabora enqueue [SID]    -- enqueue waits.
zabora dbseqread [SID]  -- db file sequential read waits.
zabora dbscattread [SID] -- db file scattered read.
zabora dbsnglwrite [SID] -- db file single write.
zabora dbprllwrite [SID] -- db file parallel write.
zabora directread [SID] -- direct path read.
zabora directwrite [SID] -- direct path write.
zabora latchfree [SID]  -- latch free.
zabora zaboraver        -- Version of this script.
```

保存配置，重启 agentd

在 zabbix web 页面上创建对应的 items 和 triggers

创建监控 oracle 数据库状态的 item，item key 不要 select，直接在 key 栏输入在 agent 上的 UserParameter name

例如监控数据库状态的：oracle.checkactive

Item '724AS5:'	
Host	724AS5 <input type="button" value="Select"/>
Description	<input type="text" value="zabora checkactive"/>
Type	<input type="text" value="Zabbix agent"/>
Key	<input type="text" value="oracle.checkactive"/> <input type="button" value="Select"/>
Type of information	<input type="text" value="Numeric (unsigned)"/>
Data type	<input type="text" value="Decimal"/>
Units	<input type="text"/>
Use multiplier	<input type="text" value="Do not use"/>
Update interval (in sec)	<input type="text" value="30"/>
Flexible intervals (sec)	No flexible intervals
New flexible interval	Delay <input type="text" value="50"/> Period <input type="text" value="1-7,00:00-23:59"/>
	<input type="button" value="Add"/>

图表 19

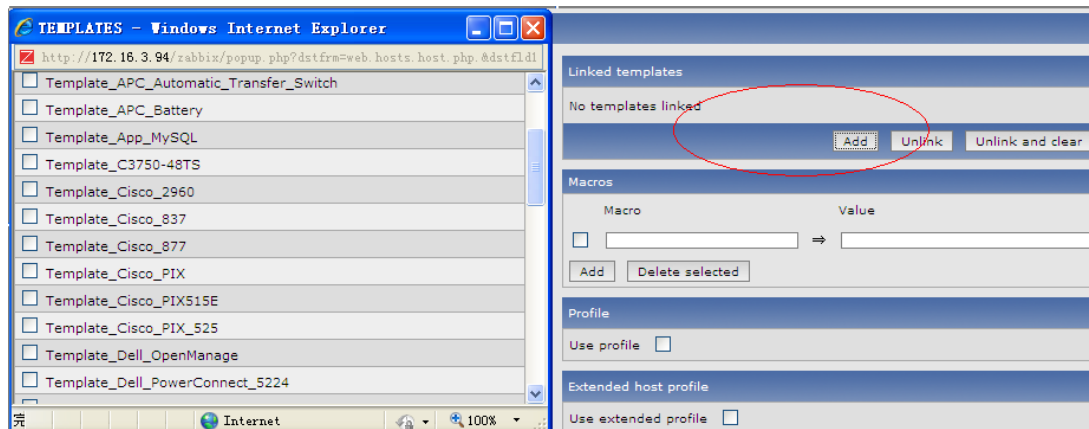
创建 trigger，选择之前创建的 item，最后一次采集值不为 0，即为状态有问题。

(注：不确定每个 zabora 方法的返回值时，可直接在 agent 上运行获取结果)

图表 20

### 3.13 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。在我看来 zabbix 对 server 的监控一般是用 agent 来做的，所以 zabbix 不提供 snmp 对 server 的监控，这一点和 cacti 有本质的区别，因为 cacti 主要是通过 snmp 对主机和网络设备进行监控的。



图表 21

### 3.14 添加通用协议监控

### 3.15 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。

具体的方法请参考上述 oracle 监控。

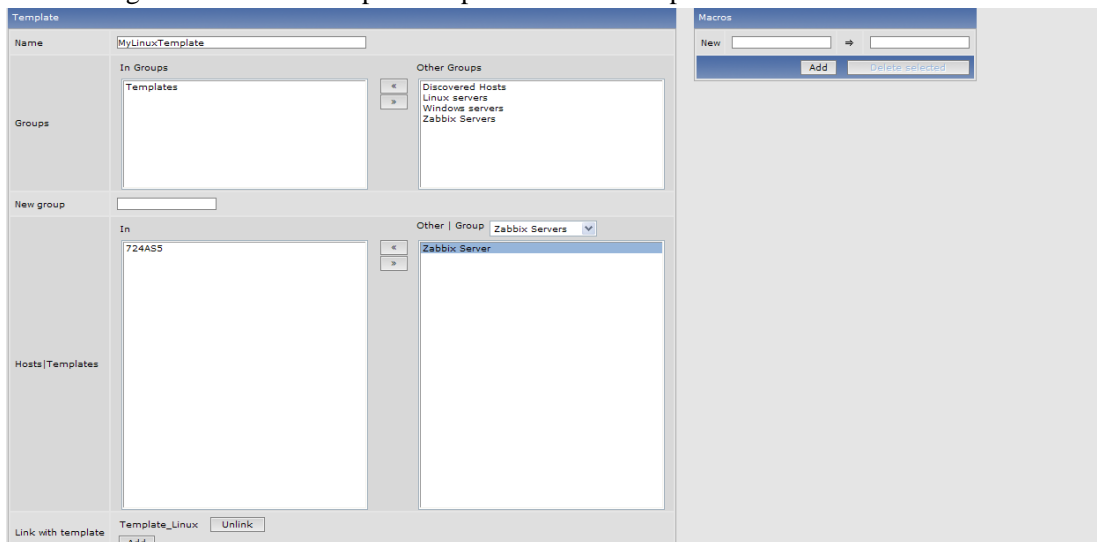
UserParameters 的定义方法，请参考：

[http://www.zabbix.com/documentation/1.8/manual/config/user\\_parameters](http://www.zabbix.com/documentation/1.8/manual/config/user_parameters)

### 3.16 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



图表 22

创建 template 后，在 configuration->host->template 下找到刚创建的 MyLinuxTemplate，修改相关的 items、triggers、graphs 等信息，使满足要求后 link 到相关的 host 即可。

### 3.17 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

图表 23

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

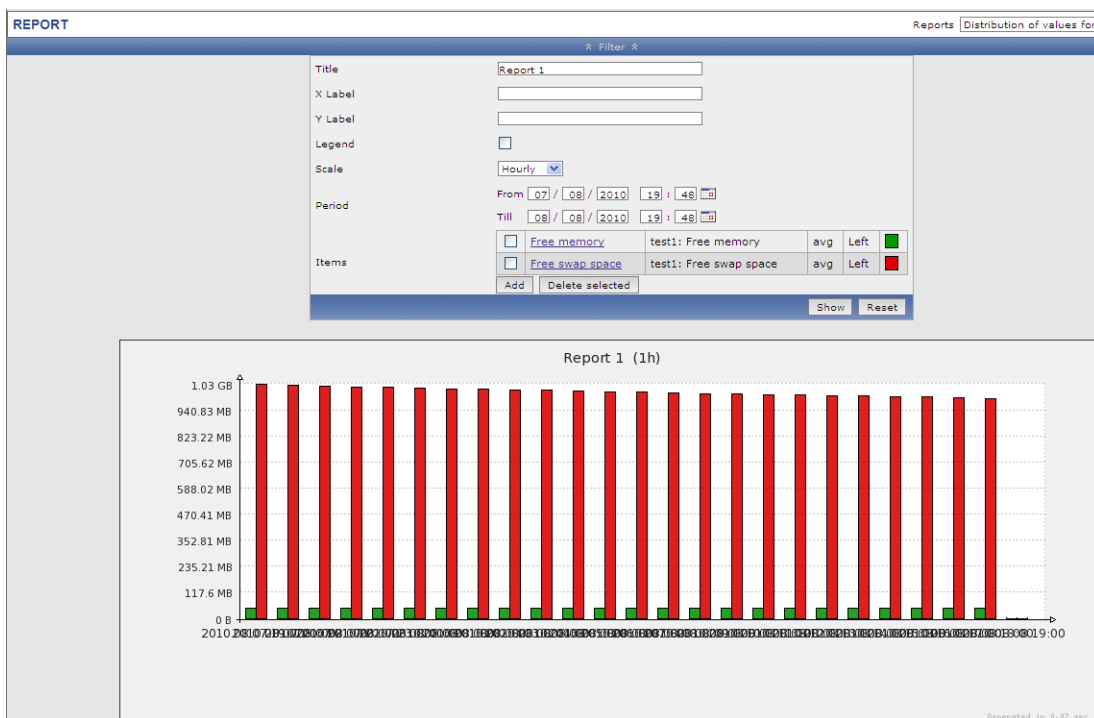
Host	Name	Problems	Ok	Unknown	Graph
test1	/etc/passwd has been changed on server test1	0.0000%	100.0000%	0.0000%	Show
test1	/etc/services has been changed on server test1	0.0000%	100.0000%	0.0000%	Show
test1	/usr/bin/ssh has been changed on server test1	0.0000%	100.0000%	0.0000%	Show
test1	/usr/sbin/sshd has been changed on server test1	0.0000%	100.0000%	0.0000%	Show
test1	Apache is not running on test1	0.0000%	100.0000%	0.0000%	Show
test1	Configured max number of opened files is too low on test1	0.0000%	100.0000%	0.0000%	Show
test1	Configured max number of processes is too low on test1	0.0000%	100.0000%	0.0000%	Show
test1	Email (SMTP) server is down on test1	0.0000%	100.0000%	0.0000%	Show
test1	FTP server is down on test1	0.0000%	100.0000%	0.0000%	Show
test1	Host information was changed on test1	0.0000%	100.0000%	0.0000%	Show
test1	Hostname was changed on test1	0.0000%	100.0000%	0.0000%	Show
test1	IMAP server is down on test1	100.0000%	0.0000%	0.0000%	Show
test1	Inetd is not running on test1	100.0000%	0.0000%	0.0000%	Show
test1	Lack of free memory on server test1	0.0000%	100.0000%	0.0000%	Show

图表 24

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:



图表 25

### 3.18 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/1.8/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。

### 3.19 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/1.8/manual/auto-discovery>

### 3.20 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。

The screenshot shows the Zabbix web interface for configuring a host. The main form is titled 'Host [724AS5]'. It includes fields for Name (724AS5), Groups (Linux servers), DNS name, IP address (127.0.0.1), Connect to (IP address), Zabbix agent port (10050), Monitored by proxy ((no proxy)), Status (Monitored), and Use IPMI (unchecked). The 'Profile' section is expanded, showing 'Use profile' checked, and fields for Device type (HP D408), Name (724AS5), OS (RedHat Linux Enterprise Advance Service 5), SerialNo (123456778), Tag (A4011), MAC Address (00:0F:FE:1B:11:08), Hardware (intel p4 2.0 cpu, 512 M mem), Software (mysql, zabbix), and Contact (admin). The 'Linked templates' section shows 'Template\_Linux' selected. At the bottom, there are buttons for Save, Clone, Full clone, Delete, and Cancel.

图表 26

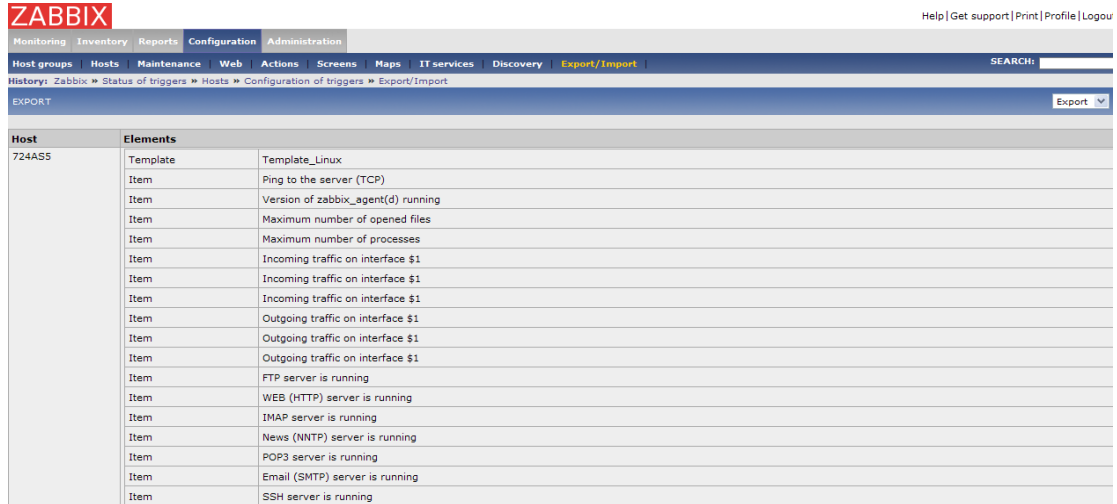
填写的 inventory 信息在 inventory->hosts 下能看到,可以代替 OCS 一些功能。



## 3.21 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->Export/Import->Export,勾选要导出的 host, 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。



Host	Elements	
724AS5	Template	Template_Linux
	Item	Ping to the server (TCP)
	Item	Version of zabbix_agent(d) running
	Item	Maximum number of opened files
	Item	Maximum number of processes
	Item	Incoming traffic on interface \$1
	Item	Incoming traffic on interface \$1
	Item	Incoming traffic on interface \$1
	Item	Outgoing traffic on interface \$1
	Item	Outgoing traffic on interface \$1
	Item	Outgoing traffic on interface \$1
	Item	FTP server is running
	Item	WEB (HTTP) server is running
	Item	IMAP server is running
	Item	News (NNTP) server is running
	Item	POP3 server is running
	Item	Email (SMTP) server is running
	Item	SSH server is running

图表 27

Import 可导入本地的 XML 文件，注意格式一定要符合标准要求，如果 server 上有同名的配置，会被覆盖掉。

如果有大量的配置需要手动新增，譬如新增 oracle 监控，我们可以手动编写 xml，一次导入所有的 items 和 triggers，但要注意格式，可自己导出一个配置后参考。

## 3.22 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。

You are requesting to schedule downtime for a particular host

**Command Options**

Host Name:

Author (Your Name):

Comment:

Triggered By:

Start Time:

End Time:

Type:

If Flexible, Duration:  Hours  Minutes

Child Hosts:

**Command Description**

This command is used to schedule downtime for a particular host. During the specified downtime, Nagios will not send notifications out about the host. When the scheduled downtime expires, Nagios will send out notifications for this host as it normally would. Scheduled downtimes are preserved across program shutdowns and restarts. Both the start and end times should be specified in the following format: **mm/dd/yyyy hh:mm:ss**. If you select the *fixed* option, the downtime will be in effect between the start and end times you specify. If you do not select the *fixed* option, Nagios will treat this as "flexible" downtime. Flexible downtime starts when the host goes down or becomes unreachable (sometime between the start and end times you specified) and lasts as long as the duration of time you enter. The duration fields do not apply for fixed downtime.

图表 28

上图是 Nagios 的 downtime 设置，可以看出比较简单。

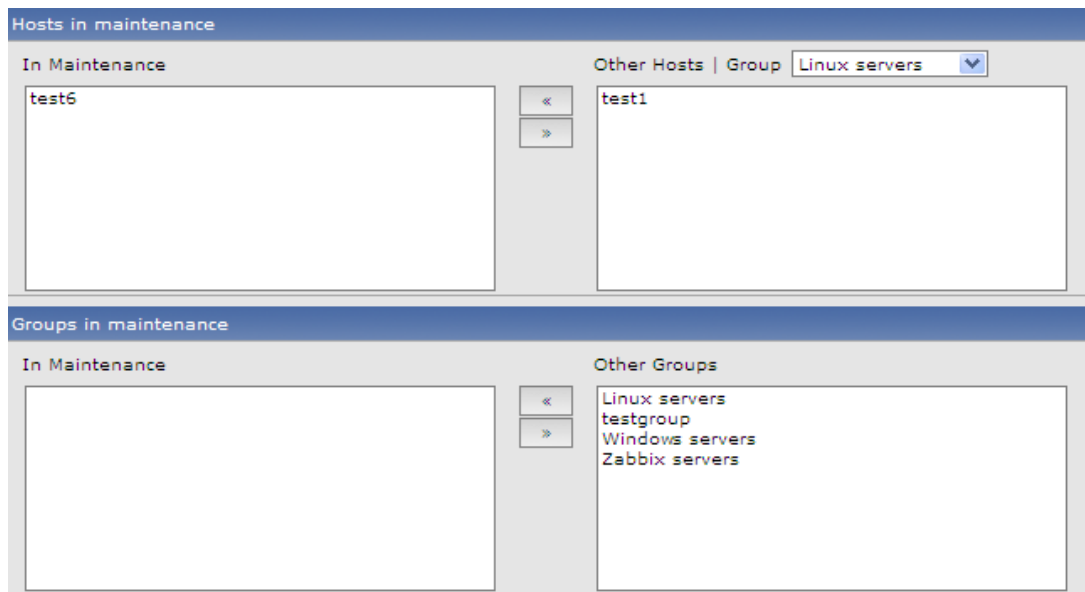
The screenshot shows the 'CONFIGURATION OF MAINTENANCE PERIODS' page in Nagios. It includes a form for creating a maintenance period with fields for Name, Maintenance type (set to 'No data collection'), Active since, Active till, and Description. Below the form is a table with columns for Maintenance type, Schedule, Period, and Action. To the right, there are two panels: 'Hosts in maintenance' and 'Groups in maintenance', each showing a list of items currently in maintenance and a list of other available items to be added.

图表 29

在 Maintenance 中可以添加一个维护的名字，在 Maintenance type 中可以设置 no data collection 选择该选项以后数据不在从 agent 端收集，也就是说不会产生 tigger，这个和 nagios downtime 一样在该阶段都不会有告警信息，还有一个选项是 with time data 收集相关数据,应该也不会产生告警信息(没有试验，验证)。

在下面的 Active since 和 Active till 可以选择开始和结束时间，Description 添加该维护时间的相关描述。

下面的 Maintenance 是更加详细的时间设置，即维护周期时间类型，one time onle daily(每天) 每周，每月等开始日期时间长度等设置。



图表 30

选择维护的 server 或者 groups.

## 4. Part4.FAQ

Export/Import XML 文件时报 http 500 错误

答：安装 php-xml 包后，重启 apache 即可

WEB 前端安装 Step 3 PHP MB Strings Overload 检查 fail

答：修改 php.ini 中的 mbstring.func\_overload = 2，重启 apache

无法修改 trigger 的阈值

答：添加 host 时如果 link 了 template，那 items 和 triggers 都是引用自 template，而不是实际属于这个 host 的，所以如果要修改 trigger 阈值，需要修改 template 中的阈值，修改后所有 link 到这个 template 的 trigger 都会改变。如果只想修改某台 host 的 trigger，可以复制一个 trigger，修改新复制过来的阈值，再禁用掉之前的即可。

运行 agentd 时进程总起不来，查看 agetd.log 又无报错

答：全路径运行后正常