## Part1.Zabbix简介

**一、**             **Zabbix简介**

Zabbix Zabbix 是一个提供 Web 管理界面的企业级的开源系统/网络监控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](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

**二、**             **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

**三、**             **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 AS5，2GCPU 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

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

安装前先配置好PHP，要求支持 php-gd、php-bcmath、php-xml、php-mysql、php-net-socket、php-mbstring，即 configure 参数中加上 –with-gd –enable-bcmath –enable-xml –with-mysql –enable-sockets –enable-mbstring

下面开始安装 Zabbix：

下载并解压：<http://nchc.dl.sourceforge.net/project/zabbix/ZABBIX%20Latest%20Stable/1.8.2/zabbix-1.8.2.tar.gz>

tar zxvf zabbix-1.8.2.tar.gz

cd zabbix-1.8.2

\* 创建 zabbix 用户组和用户：

groupadd zabbix  
useradd zabbix -g zabbix

\* 创建 mysql 数据库：

create database zabbix character set utf8;

\* 创建 mysql 用户：

grant all on zabbix.\* to zabbix@localhost identified by 'zabbix';

\* 导入表和数据：

mysql -uroot -p zabbix < create/schema/mysql.sql  
mysql -uroot -p zabbix < create/data/data.sql  
mysql -uroot -p zabbix < create/data/images\_mysql.sql

\* 配置编译：./configure --enable-server --enable-proxy --enable-agent --with-mysql=/usr/local/mysql/bin/mysql\_config --with-net-snmp --with-libcurl  
make && make install

配置参数说明：

–enable-server 安装 Zabbix Server  
–enable-proxy 安装 Zabbix Proxy  
–enable-agent 安装 Zabbix Agent  
–with-mysql 使用 mysql 做数据库服务器  
–with-net-snmp 支持 SNMP  
–with-libcurl 支持 curl，用于 web 监控

\* 服务端口定义：  
编辑 /etc/services，在后面追加：

zabbix-agent 10050/tcp #Zabbix Agent

zabbix-agent 10050/udp #Zabbix Agent

zabbix-trapper 10051/tcp #Zabbix Trapper

zabbix-trapper 10051/udp #Zabbix Trapper

\* 复制配置文件：

mkdir /etc/zabbix  
cp misc/conf/zabbix\_server.conf /etc/zabbix/  
cp misc/conf/zabbix\_proxy.conf /etc/zabbix/  
cp misc/conf/zabbix\_agent.conf /etc/zabbix/  
cp misc/conf/zabbix\_agentd.conf /etc/zabbix/

\* 修改 zabbix server 配置文件 /etc/zabbix/zabbix\_server.conf 中的数据库用户名和密码：

DBUser=zabbix  
DBPassword=zabbix

\* 安装启动脚本

cp misc/init.d/redhat/zabbix\_server\_ctl /etc/init.d/zabbix-server  
cp misc/init.d/redhat/zabbix\_agentd\_ctl /etc/init.d/zabbix-agentd

添加可执行权限：

chmod +x /etc/init.d/zabbix-server  
chmod +x /etc/init.d/zabbix-agentd

修改 zabbix-server 变量定义：

BASEDIR= /usr/local

ZABBIX\_SUCKERD=$BASEDIR/sbin/zabbix\_server

修改 zabbix-agentd 头部变量定义：

BASEDIR= /usr/local

ZABBIX\_AGENTD=$BASEDIR/sbin/zabbix\_agentd

\* 添加到启动服务：

chkconfig --add zabbix-server   
chkconfig --add zabbix-agentd

chkconfig zabbix-server on

chkconfig zabbix-agentd on

|  |
| --- |
| 如何让 Zabbix Server 和 Zabbix agentd 开机自动运行，免得每次都要手动运行，步骤如下：  1. 复制 zabbix 源程序 misc/init.d/redhat 下的启动脚本到 /etc/init.d 目录下 cp /zabbix/zabbix-1.6.6/misc/init.d/redhat/zabbix\_server\_ctl /etc/init.d/zabbix\_server cp /zabbix/zabbix-1.6.6/misc/init.d/redhat/zabbix\_agentd\_ctl /etc/init.d/zabbix\_agentd  2. 修改启动脚本使其支持 redhat 的 chkconfig，分别在两个脚本的 #!/bin/sh 后加入如下两行注释，注意要行前要加“#” 哦 # chkconfig: - 95 95 # description: Zabbix Server   3. 修改启动脚本中 BASEDIR 和 ZABBIX\_SUCKERD ，制定到 zabbix\_server 和 zabbix\_agentd 的安装位置，如： BASEDIR=/usr/local/sbin ZABBIX\_SUCKERD=$BASEDIR/zabbix\_server  4. 使用 chkconfig 将其加入 init 的启动服务 chkconfig --add zabbix\_server chkconfig --add zabbix\_agentd chkconfig --level 345 zabbix\_server on chkconfig --level 345 zabbix\_agentd on  4. 使用 chkconfig --list 检查一下 chkconfig --list | grep zabbix 输出如下： zabbix\_agentd   0:off   1:off   2:off   3:on    4:on    5:on    6:off zabbix\_server    0:off   1:off   2:off   3:on    4:on    5:on    6:off |

\* 启动 Zabbix Server：

/etc/init.d/zabbix-server start

\* 启动 Zabbix Agentd

/etc/init.d/zabbix-agentd start

\* 复制 Web Interface 到 web 目录：

cp -r frontends/php /data/wwwroot/nagios.91linux.cn/webroot/zabbix

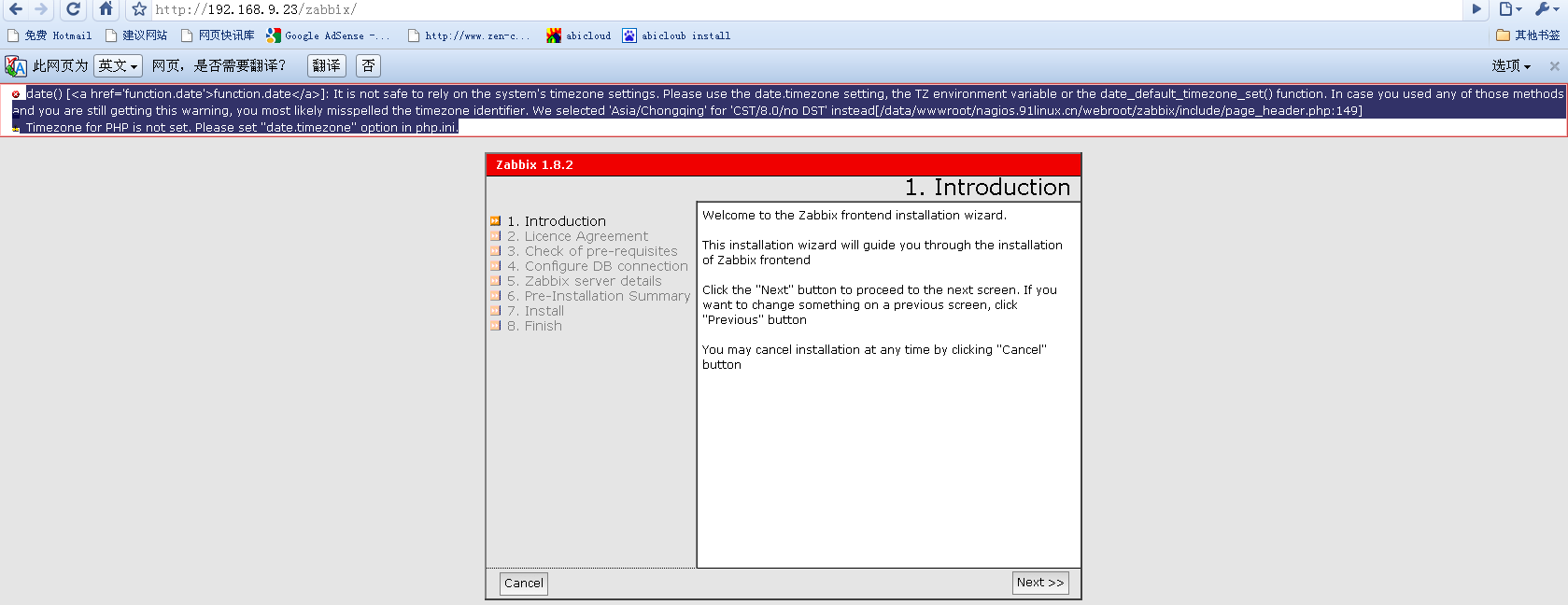
Zabbix会使用到fping，可以使用yum install fping 安装，如果yum无此软件，可使用源码安装

<http://fping.sourceforge.net/>

进入 fping 解压后的源代码目录  
./configure  
make install  
修改/etc/zabbix/zabbix\_server.conf  
FpingLocation=/usr/local/sbin/fping

\* 开始安装 Zabbix Web Interface  
打开 <http://192.168.9.23/zabbix/>，看到提示：

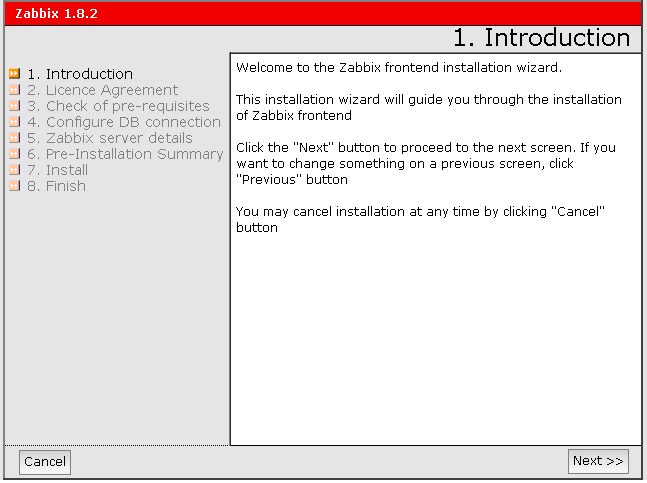
* date() [<a href='function.date'>function.date</a>]: It is not safe to rely on the system's timezone settings. Please use the date.timezone setting, the TZ environment variable or the date\_default\_timezone\_set() function. In case you used any of those methods and you are still getting this warning, you most likely misspelled the timezone identifier. We selected 'Asia/Chongqing' for 'CST/8.0/no DST' instead[/data/wwwroot/nagios.91linux.cn/webroot/zabbix/include/page\_header.php:149]
* Timezone for PHP is not set. Please set "date.timezone" option in php.ini.

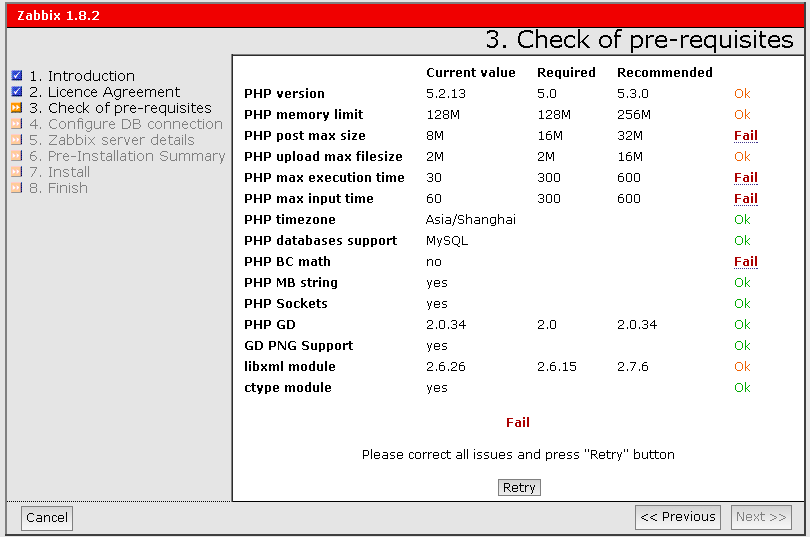


修改 php.ini 中时区设置：

date.timezone = Asia/Shanghai

继续<http://192.168.9.23/zabbix/>





修改php.ini

post\_max\_size = 32M

upload\_max\_filesize = 16M

max\_execution\_time = 600

max\_input\_time = 600

memory\_limit = 256M

默认编译没有加上bcmath，手动加上

cd /usr/local/src/monitor/php-5.2.13/ext/bcmath

/usr/local/php-5.2.13/bin/phpize

./configure --with-php-config=/usr/local/php-5.2.13/bin/php-config

make

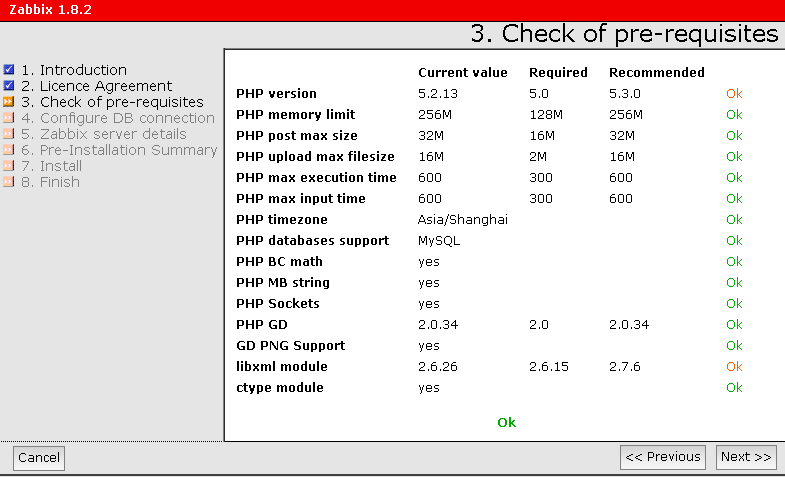
make install

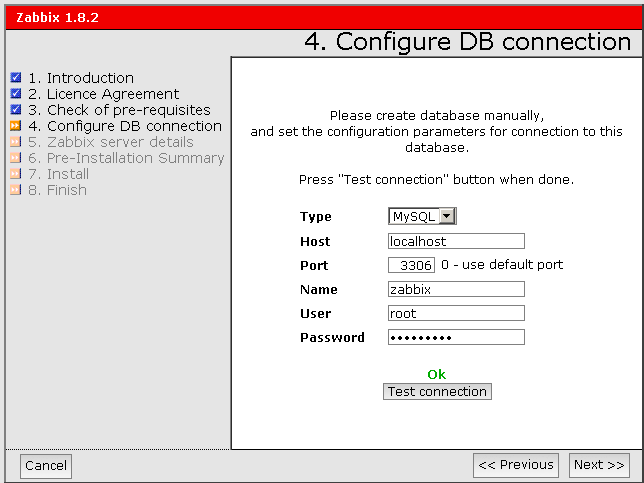
编辑php.ini

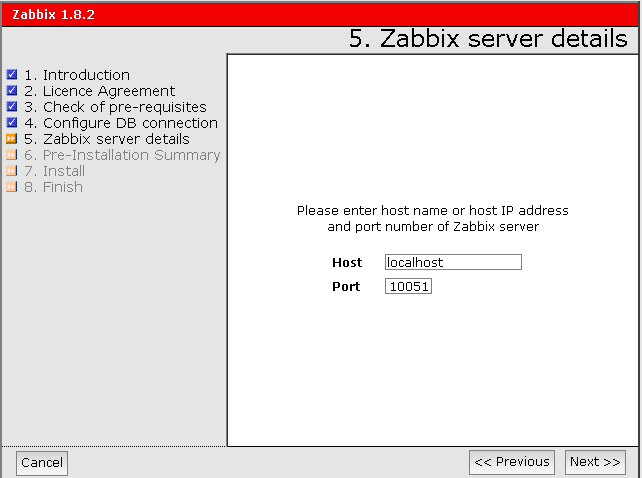
extension\_dir = "/usr/local/php-5.2.13/lib/php/extensions/no-debug-zts-20060613/"

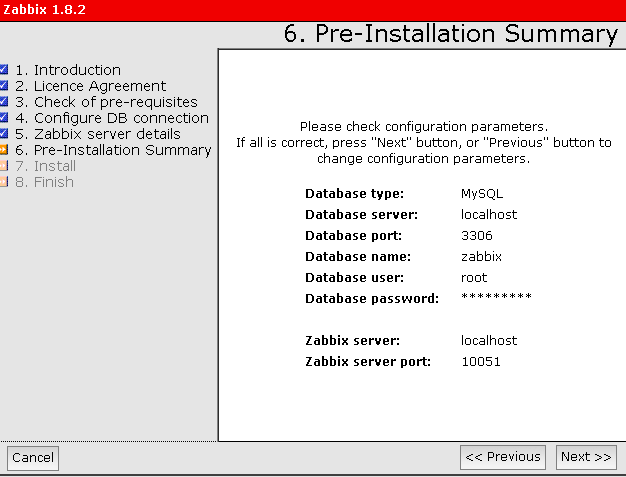
在其下增加一行：

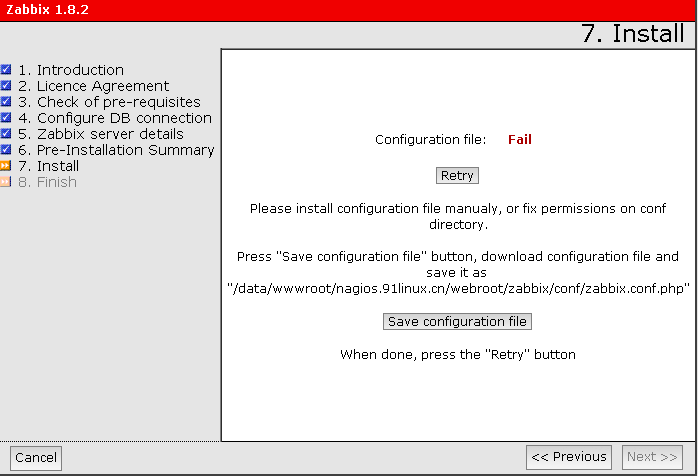
extension = bcmath.so







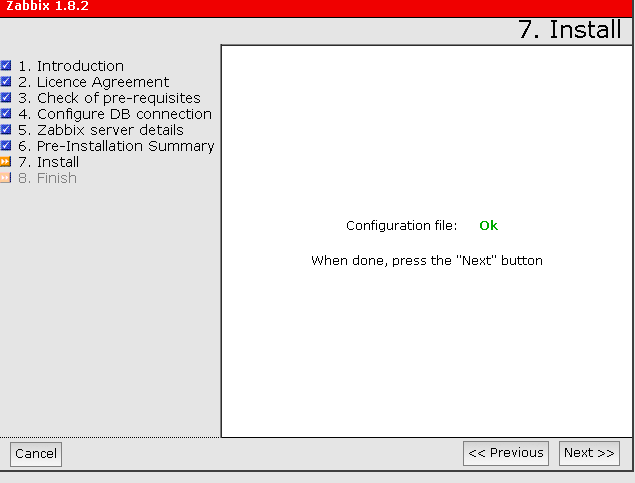


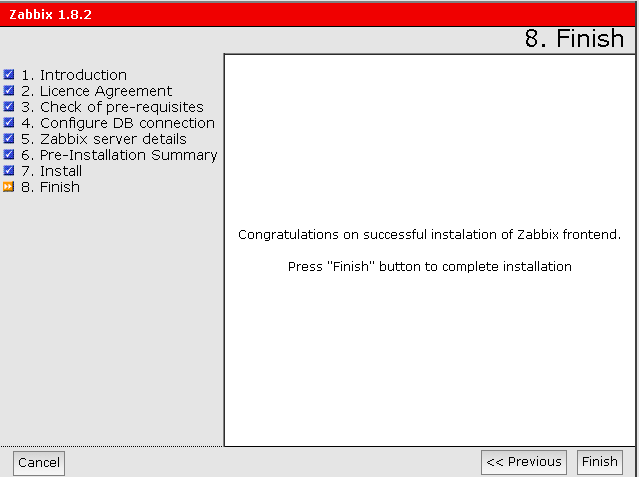


cd /data/wwwroot/nagios.91linux.cn/webroot/zabbix

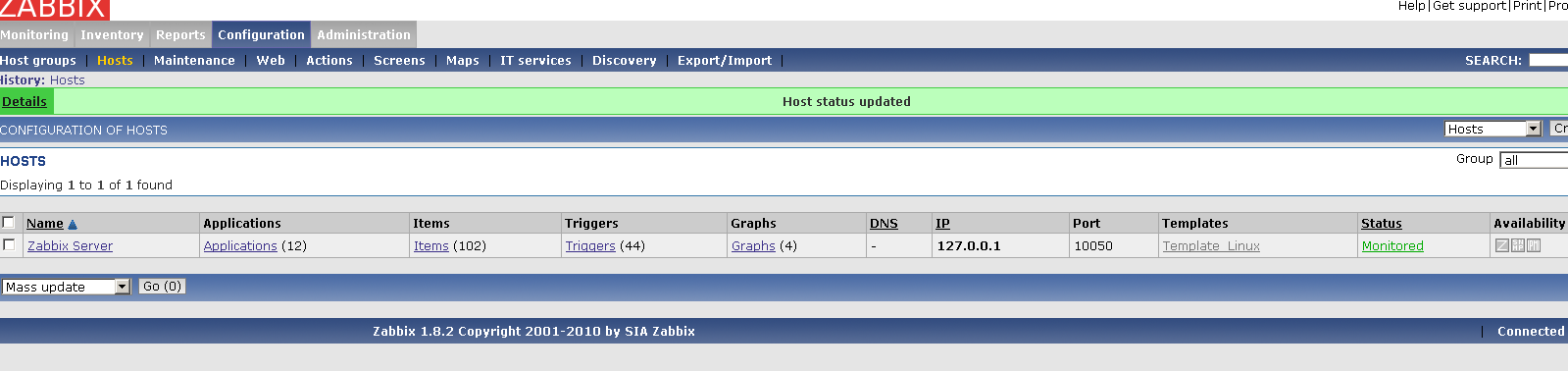
chmod 777 conf

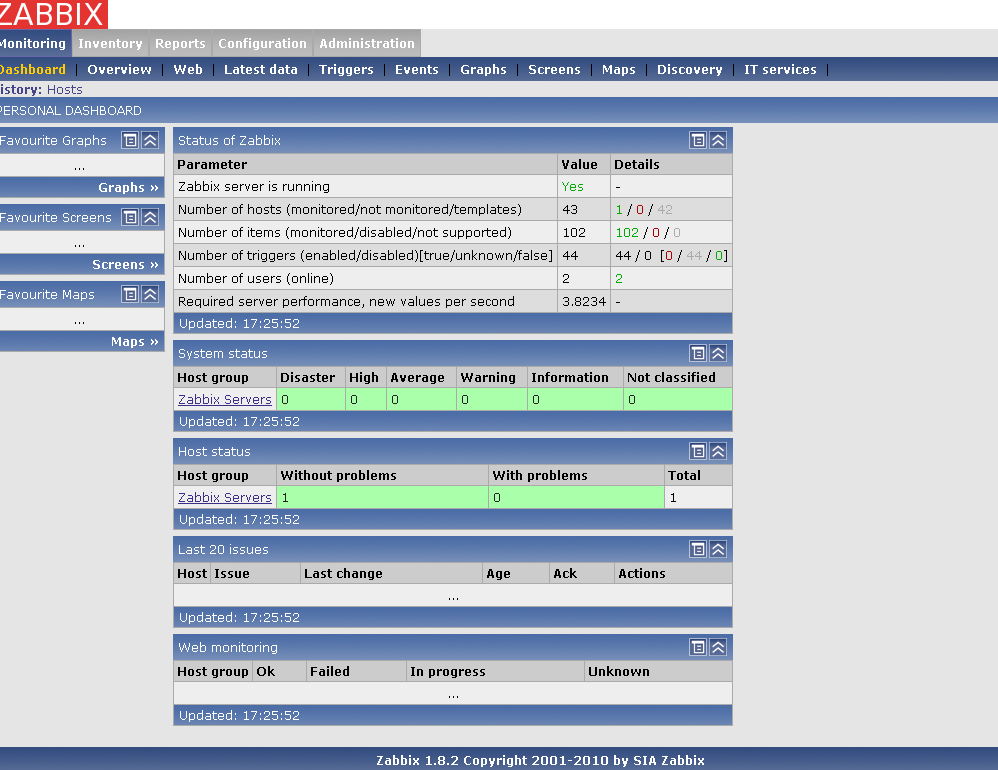
Retry





\* 结束：  
安装完后直接访问：  
http://192.168.9.23  
输入管理用户名：admin 默认口令 zabbix 登陆。进入Configuration -》Hosts 选择 Zabbix Server 然后 Activate Select，Zabbix Server （本机）的监控激活。进入 Monitoring Dashboard，可以看到 Zabbix Server 的监控结果！





汉化：zabbix本身有中文版，不过不够好，可以使用以下汉化文件

下载汉化文件：

<http://www.hiadmin.com/?p=1638>

wget <http://www.hiadmin.com/wp-content/uploads/2010/03/cn_zh.inc.php_.tar.gz>

tar zxvf cn\_zh.inc.php\_.tar.gz

mv /data/wwwroot/nagios.91linux.cn/webroot/zabbix/include/locales/cn\_zh.inc.php /data/wwwroot/nagios.91linux.cn/webroot/zabbix/include/locales/cn\_zh.inc.php.default

cp cn\_zh.inc.php /data/wwwroot/nagios.91linux.cn/webroot/zabbix/include/locales/

**在hosts上配置agent**

Zabbix需要在被监控的host上安装agent，在zabbix官网上下载相应平台的agent包到各被监控端，按如下方式在被监控上运行agent。（注：官网上下载的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

#cd /usr/local/zabbix

#tar zxvf zabbix\_agents\_1.8.linux2\_6.i386.tar.gz

取一份server上的zabbix\_agnetd.conf到本地，然后按说明修改，一般我们只需修改如下3个参数：

Hostname=本机名

Server=Zabbix Server IP

LogFile=本机agentd日志保存文件

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

**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

## Part3.配置使用

通过本地浏览器访问[http://ServerIP/zabbix](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。

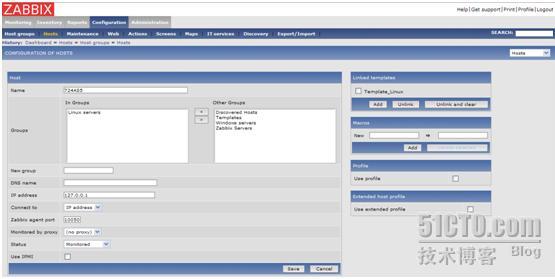
一次完整的监控流程可以简单描述为：

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）

**一、**             **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



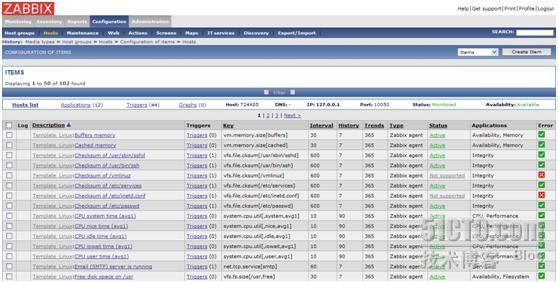
按提示填入Name、Groups、IP ,其他选项默认即可，Link Templates 处选择一个模板，save即可成功添加设备。（注：如果host上没安装agent，添加后的状态会是unmonitor，会采集不到值，Zabbix自带大量的设备监控模板，我们添加主机时通过link到这些模板，就可以快速添加主机的监控项和告警触发条件。）

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

**二、**             **添加 Items**

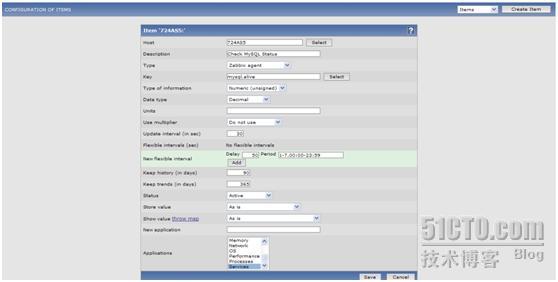
Item是监控项，是监控的基本元素，每一个监控项对应一个被监控端的采集值。

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。



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



  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>

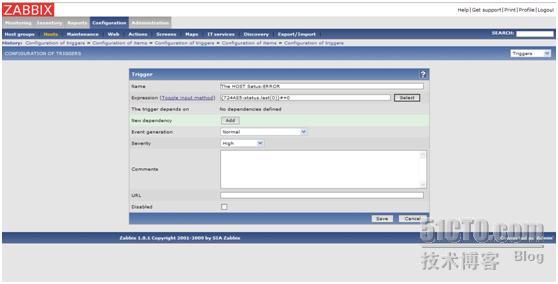
**三、**             **添加 Triggers**

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



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。

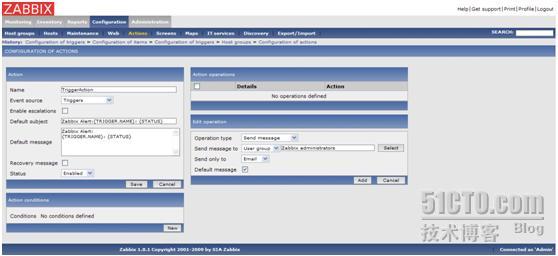
Zabbix 提供多种trigger触发方式供选择，常用的我们可以选择last value </>/=/#/N(最近一次采集值),或是选择maximal value for period of time (一段时间内的最大值),等等。可以根据实际需要来设定触发方式。更多的解释请参考：

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

**四、**             **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，如果选择triggers，即所有的triggers条件满足时都会执行这个action

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

**五、**             **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。



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。

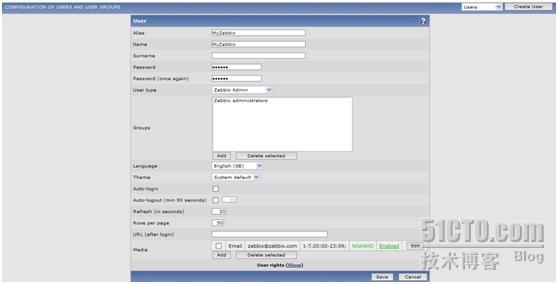
Script方式可以通过自己编写程序或脚本的方式发送告警信息。

**六、**             **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。

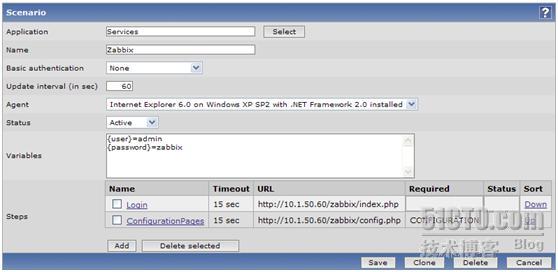


**一、**             **添加 WEB Monitorings**

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)



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



URL：监控的web页面(注:必须是全路径带页面名)

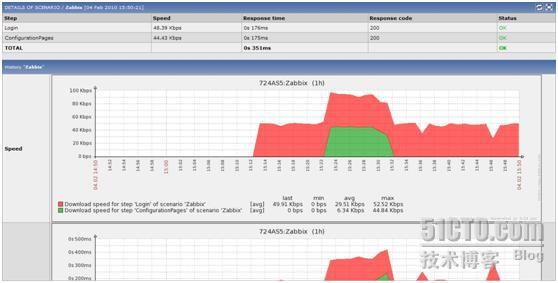
Post:传递给页面的参数，多个参数之间用&连接，此处可引用前面定义的变量

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



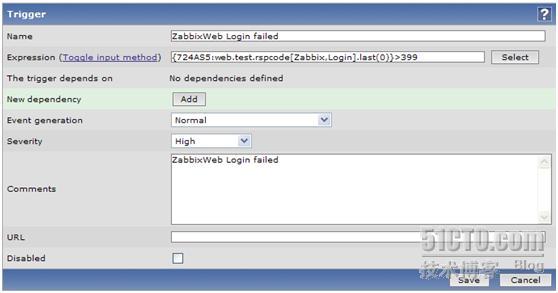
创建完scenario后，zabbix server会自动创建相关的items，所以我们只需为这些items添加triggers即可让web scenario出错时产生告警

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）

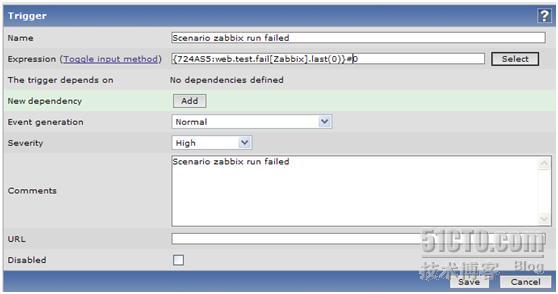


可以在items列表中看到，系统为每个step创建了3个item，Download Speed/Response Code/Response Time,为整个scenario创建了一个test.fail的item，可以分别为其创建trigger

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



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。



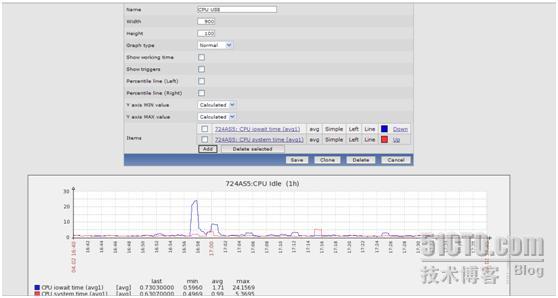
这样，一个完整的web monitoring就配置完成了。

Web monitoring还有更多强大的功能，未能一一研究了解，有待挖掘

**二、**             **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。



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）

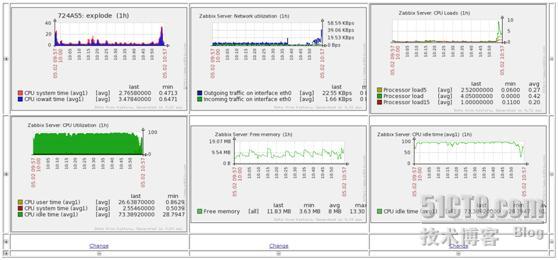
配置好的graphs在monitoring->graphs中查看

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

**三、**             **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



通过monitoring-screen，可以查看之前配置好的信息。

**四、**             **添加 Maps**

**五、**             **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，我们只需更改需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>

**一、**             **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/orcale/oracle>

1、    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oralce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 Intance 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.

   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 lastapplarclog [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.

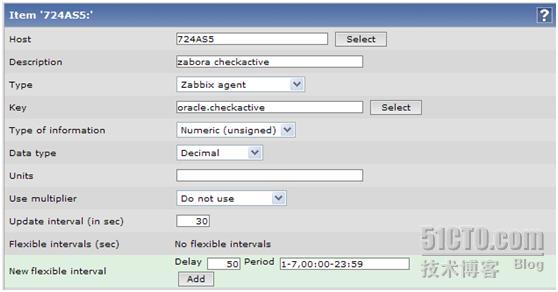
   zabora zaboraver             -- Version of this script.

保存配置，重启agentd

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

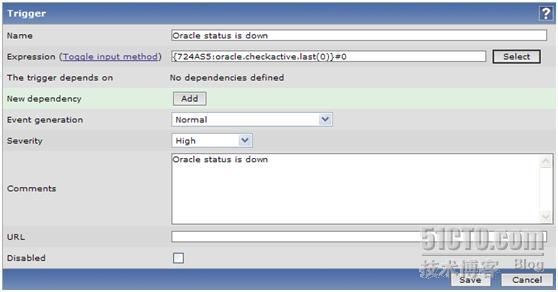
创建监控oracle数据库状态的item，item key不要select，直接在key栏输入在agent上的UserParameter name

例如监控数据库状态的：oracle.checkactive



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）



**二、**             **添加 SNMP监控**

**三、**             **添加通用协议监控**

**四、**             **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。

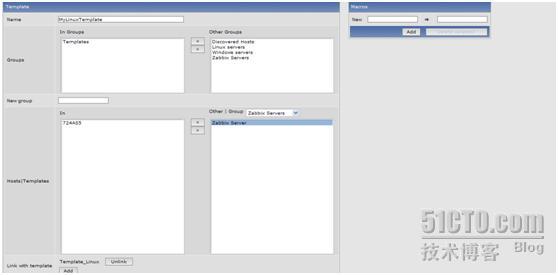
UserParameters的定义方法，请参考：

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

**五、**             **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



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。

**六、**             **添加Log File**

**七、**             **定制报表**

**八、**             **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。

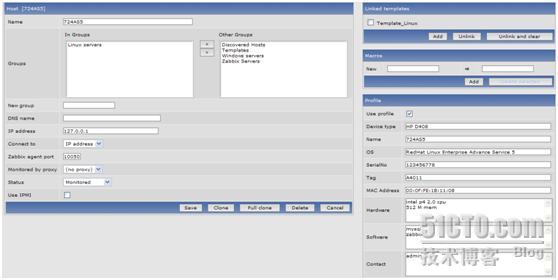
**九、**             **添加自动发现设备**

**十、**             **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。

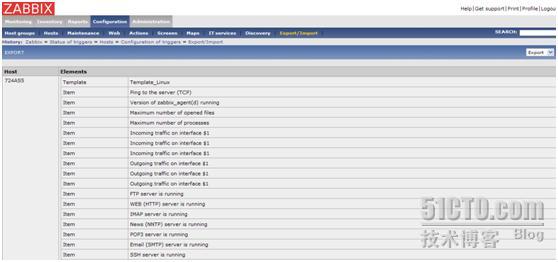
填写的inventory信息在inventory->hosts下能看到.



**十一、**      **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。



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oralce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。

**Part4.FAQ**

**1、 Export/Import XML 文件时报 http 500错误**

答：安装php-xml包后，重启apache即可

**2、 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

**3、 无法修改trigger的阀值**

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阀值，需要修改template中的阀值，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，修改新复制过来的阀值，再禁用掉之前的即可。

**4、 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

# zabbix全面监控mysql安装方法（原创）

一、从网上下载相应脚本与ＸＭＬ定义文件。

下载地址：[http://www.zabbix.com/wiki/doku.php?id=extensive\_mysql\_monitoring\_including\_replication](http://home.ixpub.net/link.php?url=http://www.zabbix.com%2Fwiki%2Fdoku.php%3Fid%3Dextensive_mysql_monitoring_including_replication)

Template\_MySQL\_Server.xml

Template\_MySQL\_Replication\_Master.xml

Template\_MySQL\_Replication\_Slave.xml

和执行脚本：mysql.php

二、把mysql.php上传至配置文件目录/etc/zabbix/

chmod 755 mysql.php

三、修改mysql.php文件

根据本机环境。

第一行加入：#!/usr/bin/php

在最后一行加入：?>

关闭调试：define('DEBUG',true);　为　define('DEBUG',False);

修改日志、数据文件路径：

define('LOG',"/tmp/zabbix\_".SYSTEM.".log");  
define('DAT',"/tmp/zabbix\_".SYSTEM.".dat");  
define('UTIME',"/tmp/.zabbix\_".SYSTEM.".utime");  
define('DTIME',"/tmp/.zabbix\_".SYSTEM.".dtime");

修改：define('SYSTEM','mysql'.(DEBUG ? "-debug" : "")); 为：define('SYSTEM','mysql');

打开系统日志功能：//system("zabbix\_sender -z $server -i ".DAT." >> ".LOG);　为：system("zabbix\_sender -z $server -i ".DAT." >> ".LOG);

在$cmd = "zabbix\_sender -z $server -p 10051 -s $host -k ".SYSTEM.".$var -o $val";  
下面增加一行：exec($cmd);

四、在zabbix\_agentd.conf配置文件中加入

UserParameter=mysql.daily,php /etc/zabbix/mysql.php daily 用户名 密码  
UserParameter=mysql.live,php /etc/zabbix/mysql.php live 用户名　密码

daily:每天执行一次。

live:按指定时间执行一次。

php :执行php文件

/etc/zabbix/mysql.php:mysql.php文件所在的文件路径

用户名　密码：登录mysql数据库的账户与密码

重启zabbix\_agentd

五、在zabbix web管理界面导入xml文件

配置－导入/导出－浏览

六、在主机内增加指定模板

配置－主机－具体主机名－Link with Template－新增－刚导入的模板。

模板分为三种：

template MYSQL\_server:监控项最全

template MYSQL\_Replication\_Master:只监控主要参数，监控功能相对最少

template MYSQL\_Replication\_Slave:主要针对性能方面的监控。

七、测试监控是否成功

删除　定时文件

rm /tmp/.zabbix\_mysql-debug.dtime;

取的每天的监控数据

php /etc/zabbix/mysql.php daily root \*\*\*\*\*

取即时数据

php /etc/zabbix/mysql.php live root \*\*\*\*\*

查看日志

cat /tmp/zabbix\_mysql.log

Info from server: "Processed 1 Failed 0 Total 1 Seconds spent 0.006261"  
sent: 1; skipped: 0; total: 1  
Info from server: "Processed 1 Failed 0 Total 1 Seconds spent 0.004047"  
sent: 1; skipped: 0; total: 1  
Info from server: "Processed 0 Failed 106 Total 106 Seconds spent 0.045790"  
sent: 106; skipped: 0; total: 106

看到上述为成功

查看ＤＡＴ的文件。上传的数据

cat /tmp/zabbix\_mysql.dat

八、最后查看zabbix　的最新数据

这时会看到新增加的监控项。

接下去就是绘制图形。