**Zabbix3.4 手册**

**目 录**

[1. 文档介绍 3](#_Toc502153756)

[1.1. 文档目的 3](#_Toc502153757)

[1.2. 术语与解释 3](#_Toc502153758)

[1.3. 参考资料 3](#_Toc502153759)

[2. Zabbix简介 4](#_Toc502153760)

[2.1. 介绍 4](#_Toc502153761)

[2.1.1. 什么是zabbix 4](#_Toc502153762)

[2.1.2. zabbix组件 4](#_Toc502153763)

[2.1.3. 安装使用 4](#_Toc502153764)

[2.1.4. 特点和功能 5](#_Toc502153765)

[3. 架构介绍 6](#_Toc502153766)

[3.1. 运行环境 6](#_Toc502153767)

[3.1.1. 硬件环境 6](#_Toc502153768)

[3.1.2. 软件环境 6](#_Toc502153769)

[3.2. 基本设计概念 6](#_Toc502153770)

[3.2.1. 技术架构 6](#_Toc502153771)

[4. 安装部署 8](#_Toc502153772)

[4.1. 自动化安装 8](#_Toc502153773)

[4.1.1. 安装环境 8](#_Toc502153774)

[4.1.2. 安装包说明 8](#_Toc502153775)

[4.1.3. 安装介绍 8](#_Toc502153776)

[4.2. Web界面配置 9](#_Toc502153777)

[4.2.1. 登录zabbix 9](#_Toc502153778)

[4.2.2. 配置模板 12](#_Toc502153779)

[4.2.3. 配置动作 12](#_Toc502153780)

[4.2.4. 配置代理 12](#_Toc502153781)

[4.2.5. 添加主机组 12](#_Toc502153782)

[4.2.6. 添加主机 12](#_Toc502153783)

[4.2.7. 配置资产信息 12](#_Toc502153784)

[4.2.8. 配置正则表达式 12](#_Toc502153785)

[4.3. 尚未解决的问题 13](#_Toc502153786)

# 文档介绍

## 文档目的

此文档的目的是为了规范云监管平台监控系统的实施和部署，包括自动化安装部署，规范模板配置，采集数据测试，以及数据库优化等。提高工作效率，作为采集工作的唯一指引手册。

## 参考资料

https://www.zabbix.com/documentation/3.4/manual

# Zabbix简介

## 介绍

### 什么是zabbix

Zabbix是一个基于WEB界面的提供分布式[系统监视](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E7%9B%91%E8%A7%86" \t "_blank)以及网络监视功能的企业级的开源解决方案。

Zabbix能监视各种网络参数，保证[服务器系统](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8%E7%B3%BB%E7%BB%9F" \t "_blank)的安全运营；并提供灵活的通知机制以让[系统管理员](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E7%AE%A1%E7%90%86%E5%91%98)快速定位/解决存在的各种问题。

### zabbix组件

Zabbix由3部分构成，zabbix server与可选组件zabbix agent 、zabbix proxy。

Zabbix server可以通过[SNMP](https://baike.baidu.com/item/SNMP" \t "_blank)，zabbix agent，[ping](https://baike.baidu.com/item/ping" \t "_blank)，端口监视等方法提供对远程服务器/网络状态的监视，[数据收集](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%94%B6%E9%9B%86" \t "_blank)等功能。

### 安装使用

zabbix agent需要安装在被监视的目标服务器上，它主要完成对硬件信息或与操作系统有关的内存，CPU等信息的收集。

zabbix proxy用来收集zabbix agent采集到的监控数据，然后再上传给zabbix server。可以用来配合zabbix server做分布式监控系统。

zabbix server可以单独监视远程服务器的服务状态；同时也可以与zabbix agent配合，可以轮询zabbix agent主动接收监视数据（agent方式），同时还可被动接收zabbix agent发送的数据（trapping方式）。

另外zabbix server还支持SNMP (v1,v2,v3)，可以与SNMP软件(例如：net-[snmp](https://baike.baidu.com/item/snmp" \t "_blank))等配合使用。

要想搭建一个zabbix的工作环境，需要从服务器入手。与服务器通信，管理员需要使用一个zabbix前端界面，与zabbix服务器和数据库进行通信。三个关键（界面、服务器和数据库）可以安装在同一台服务器上，但是如果你拥有一个更大更复杂的环境，将它们安装在不同的主机上也是一个选项。zabbix服务器能够直接监控到同一网络中的设备，如果其他网络的设备也需要被监控，那还需要一台zabbix proxy代理服务器。

### 特点和功能

1.Zabbix的主要特点：

1. 安装与配置简单，学习成本低
2. 支持多语言（包括中文）
3. 免费开源
4. 自动发现服务器与网络设备
5. 分布式监视以及WEB集中管理功能
6. 可以无agent监视
7. 用户安全认证和柔软的授权方式
8. 通过WEB界面设置或查看监视结果
9. email等通知功能

2.Zabbix主要功能：

1. CPU负荷
2. 内存使用
3. -磁盘使用
4. 网络状况
5. 端口监视
6. 日志监视。

# 架构介绍

## 运行环境

网安云监管项目监控系统主要部署在合作厂商提供的物理服务器上，系统环境是Centos7，各个节点的硬件需求决定于监控规模的大小，关键是cpu和内存以及磁盘的iops大小。

### 硬件环境

|  |  |  |
| --- | --- | --- |
| **硬件设备** | **硬件指标** | **用途** |
| 数据采集服务器 | CPU：Intel Xeon E5-2680 v3  内存：64G+ 可扩展  磁盘：500G 可扩展（ssd）  网卡：GE | 用于数据采集 |

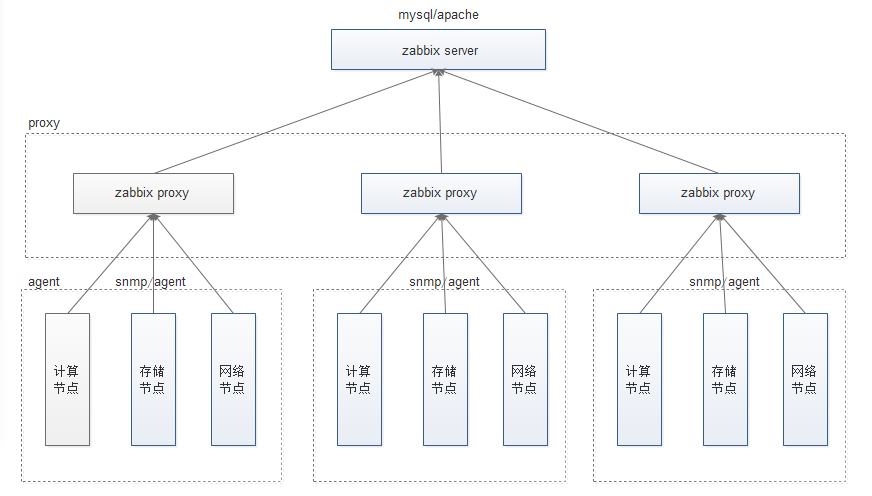
### 软件环境

|  |  |  |
| --- | --- | --- |
| 软件系统 | 软件版本 | 用途 |
| 操作系统 | CentOS 7 | 运行物理机数据采集系统 |
| 数据库管理软件 | MariaDB 5.5 | 持久化存储采集到的数据；  持久化相关检查策略等； |
| 数据采集软件 | Zabbix 3.4 | 采集数据 |
| Web界面 | Apache | Web界面 |

## 基本设计概念

### 技术架构

数据采集服务器部署在合作厂商提供的物理机或虚拟机上，一般分为zabbix server节点和zabbix proxy节点，具体分布式情况视监控规模而定。打通和采集客户端之间的网络通信，通过snmp或者zabbix agent的方式去采集数据。如下图所示：



1.首先需要保证网络可达：

* 数据采集服务器（zabbix 各节点）与监控agent网络可达

2.数据采集服务器主要通过两种方式采集agent端的数据：

* snmp：基于TCP/IP协议族的网络管理标准，是一种在IP网络中管理网络节点（如服务器、工作站、路由器、交换机等）的标准协议。需要在客户端安装并开启snmp服务。
* zabbix agent：zabbix自带的组件，需要安装到被监控的客户端服务器上，默认端口为10050。有两种数据收集方式，一种是主动接收监视数据（agent方式），同时还可被动接收zabbix agent发送的数据（trapping方式）。

# 安装部署

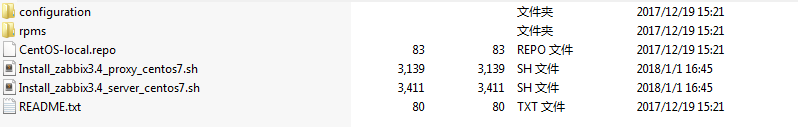
## 自动化安装

### 安装环境

CentOS Linux release 7.2.1511 (Core)

### 安装包说明

如下图所示：



安装包共包含两个文件夹和四个文件：

1. configuration文件夹是配置文件存放的地方，包括zabbix\_server.conf/zabbix\_proxy.conf/my.cnf三个配置文件，在安装之前需要将默认配置文件更改适应实际情况的配置文件。
2. rpms文件夹和Centos-local.repo文件是用来制作需要的yum源
3. Install\_zabbix3.4\_server\_centos7.sh是安装zabbix\_server的脚本，Install\_zabbix3.4\_proxy\_centos7.sh是安装zabbix\_proxy的脚本。
4. README.txt是安装说明。

### 安装介绍

首先需要将安装包Install\_zabbix3.4\_centos7.tar拷贝至系统的/home目录下，然后执行安装脚本进行安装。

## Web界面配置

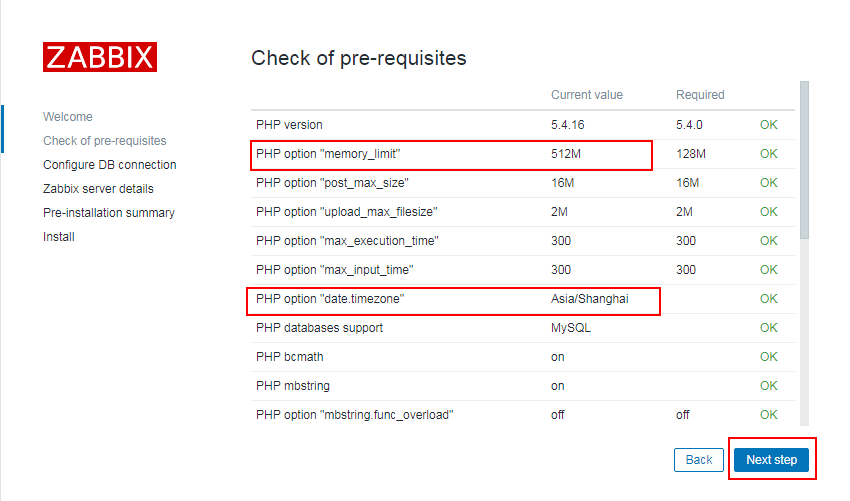
### 登录zabbix

Zabbix server安装完毕后，在浏览器打开<http://server_ip/zabbix/>，进入zabbix初始化配置界面，如下图所示：

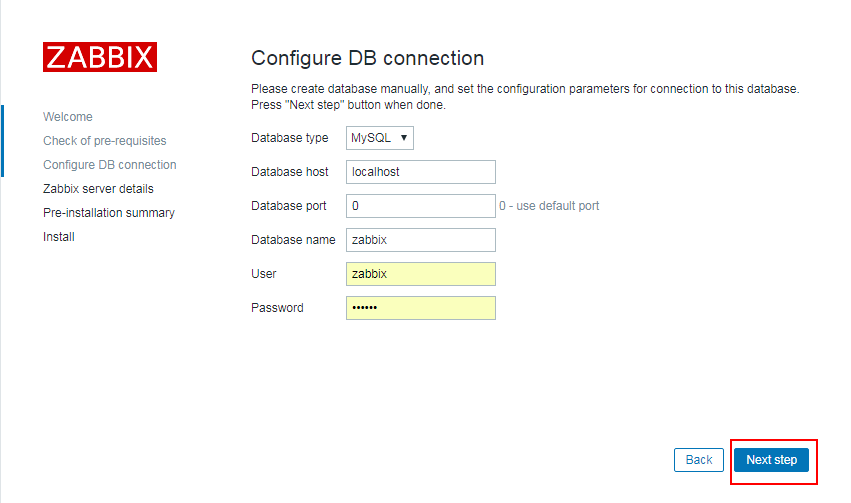
进入初始化界面首先看到如下页面，点击next step



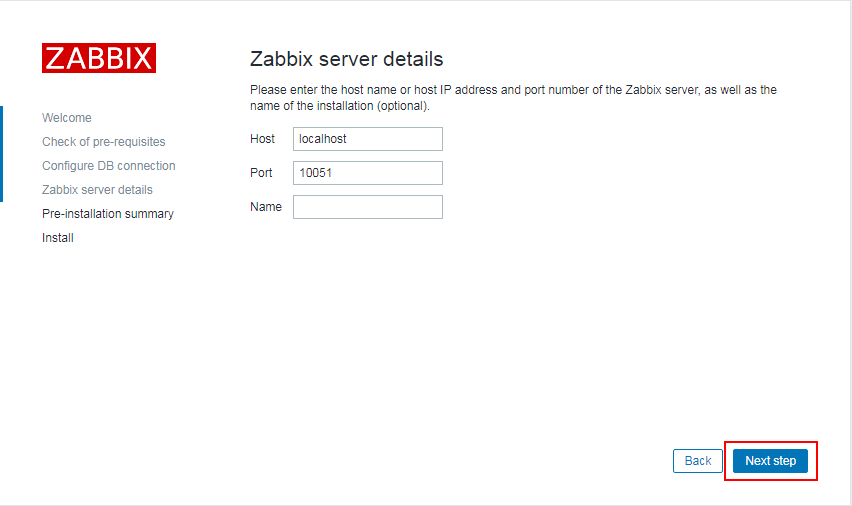
下图检查php参数是否正确，我们主要看memory\_limit以及date.timezone是否正确，确认无误后点next step



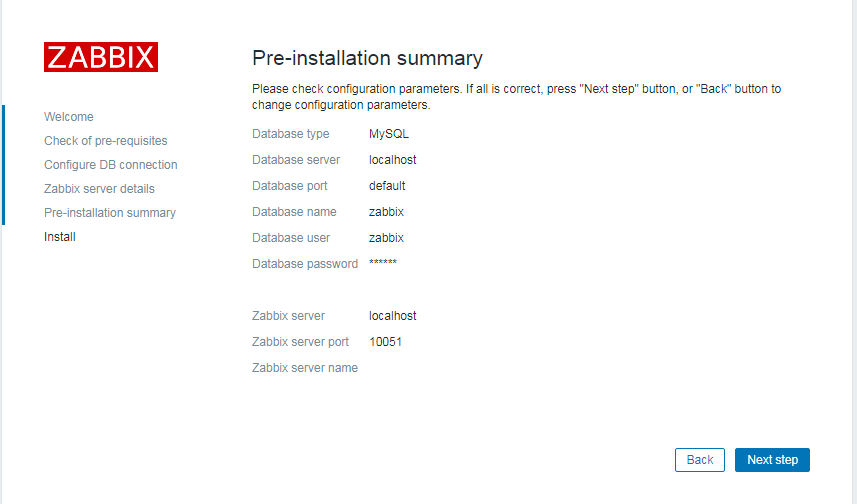
下图检查mysql数据库的用户密码等信息，点击next step。



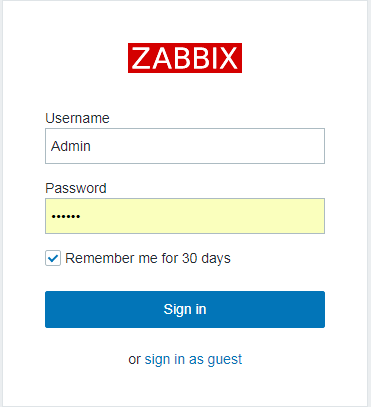
下图是zabbix server的端口ip等信息，直接点next step。



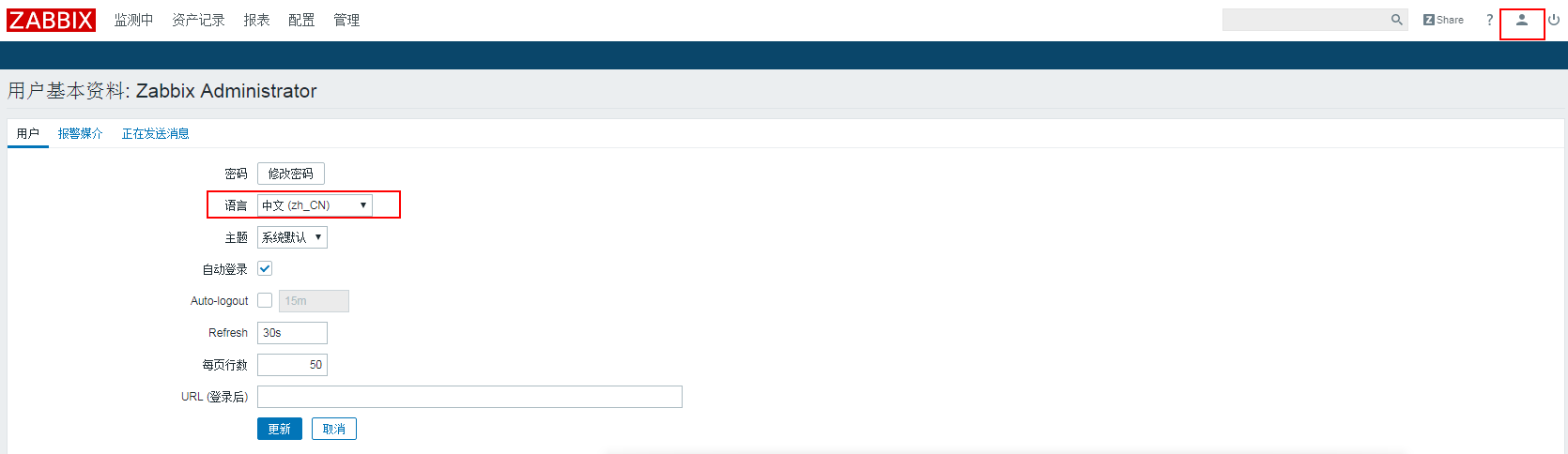
下图列出了所有初始化配置，检查无误后点击next step。



默认登录用户Admin 登录名zabbix



如果想要更改界面的语言为中文，如下图所示，点击界面右上角的，进入用户配置界面：



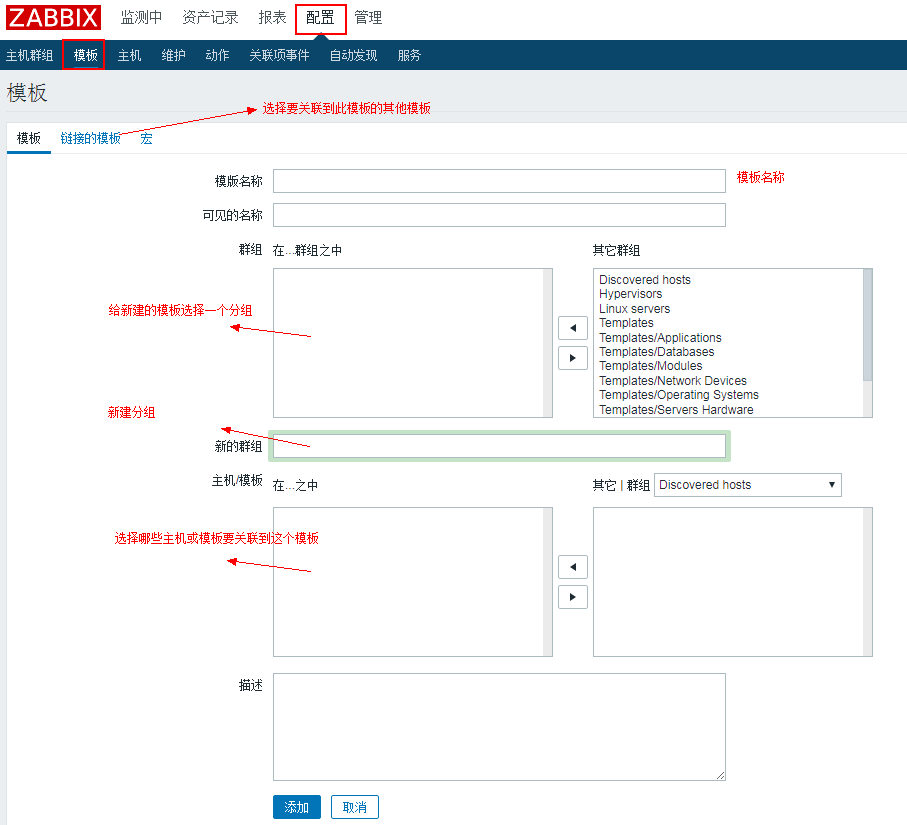
### 配置模板

Zabbix模板包含监控项、触发器、web监控、图表等等项目，创建这些项目之后，在后续的主机只需要套用这个模板，那么主机便可以监控模板里面所配置的监控项目。

1. 添加模板有两种方式

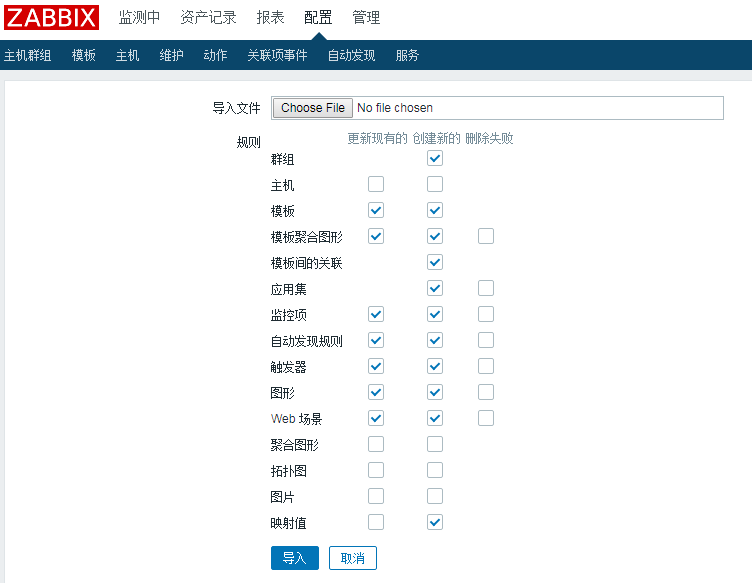


第一种方式如下图所示，配置--模板，点击右上角的创建模板来创建一个新的模板



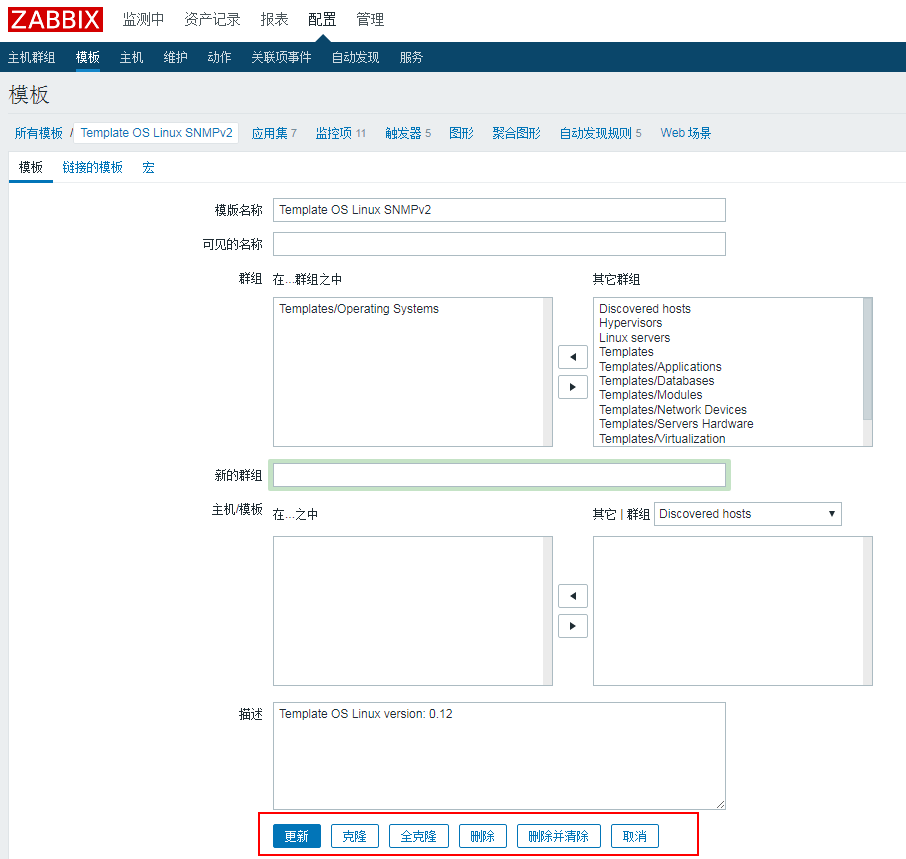
如上图所示，根据提示配置，点击添加创建新的模板。

第二种方式，同样是配置—模板，点击右上角的导入，可以直接选择导入已有的模板，如下图：



1. 对模板进行操作

如下图所示，模板创建好之后，可以对其进行更新、删除、克隆等操作:

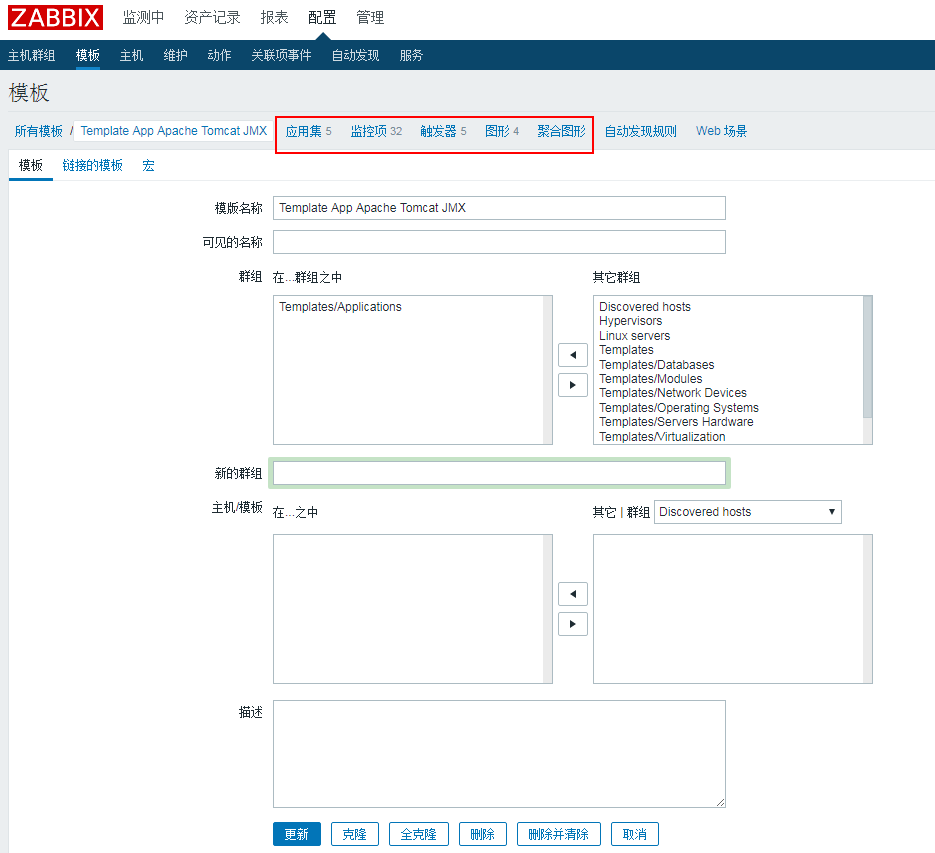


### 创建监控项和触发器

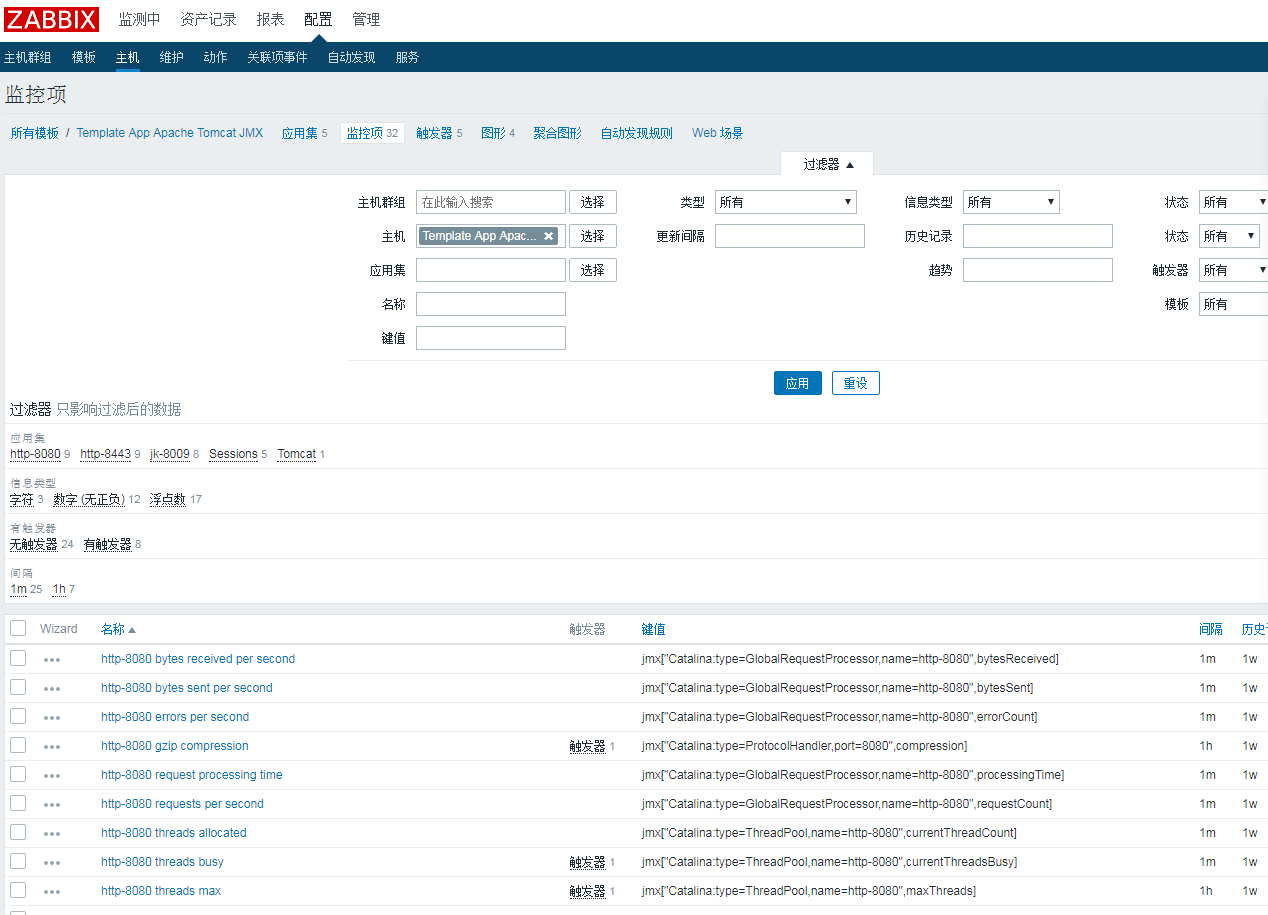
模板创建好之后，在关联到主机之前，需要创建监控主机需要的监控项和触发器。

1. 创建监控项

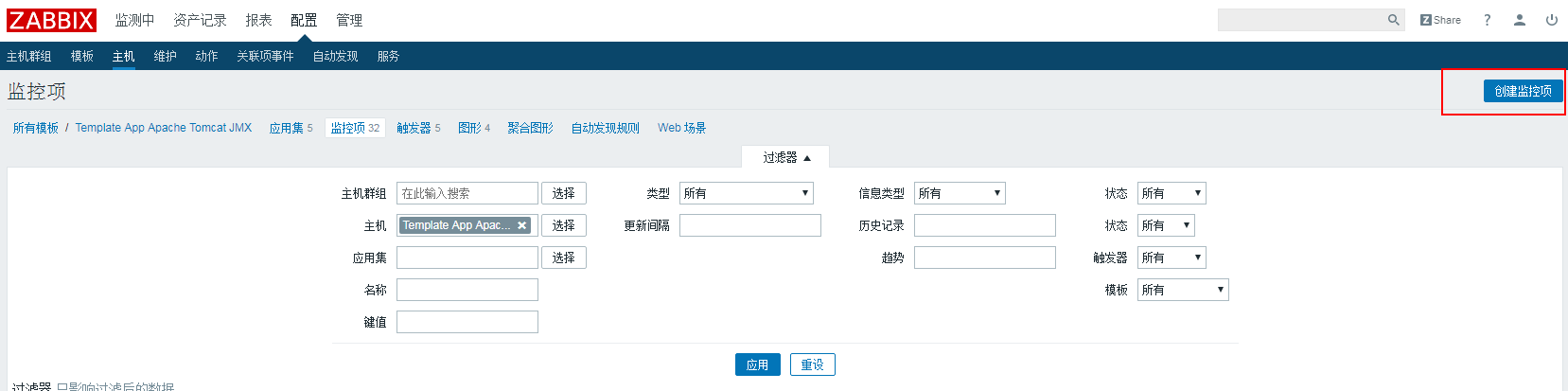
配置—模板，选择需要添加监控项的模板，如下图，选择模板之后，点监控项来查看已有的监控项和创建新的监控项。

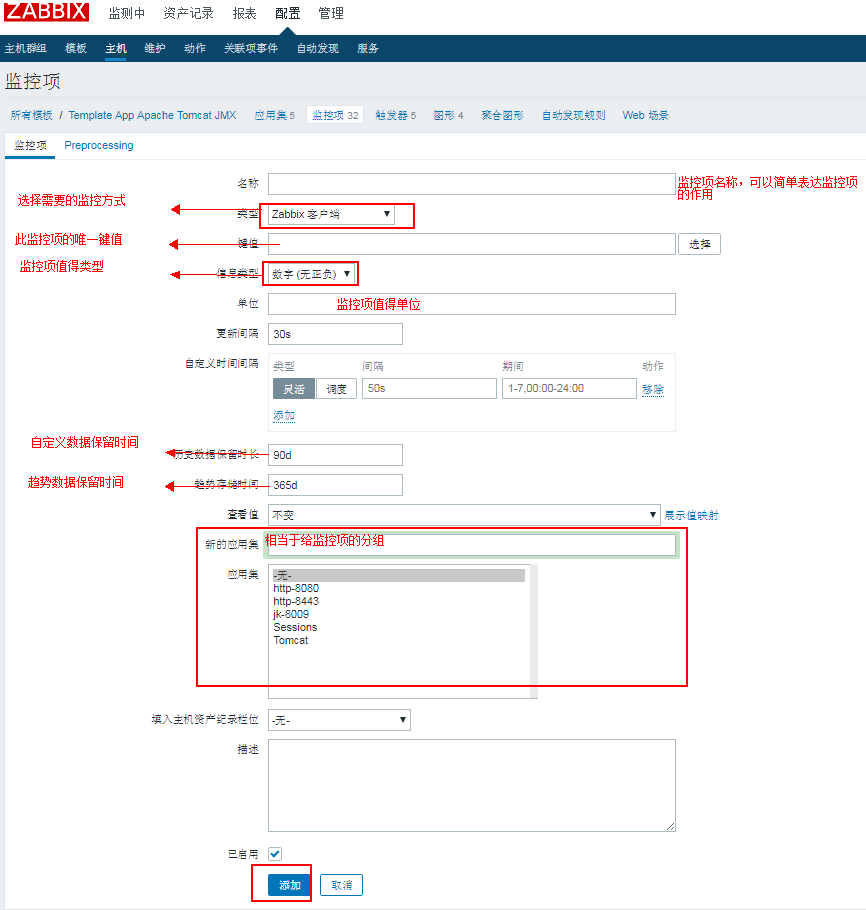


查看监控项，如果监控项太多，可以通过过滤器来过滤要查看的监控项，如下图：



添加新的监控项，如下图所示：





选择不同的监控方式，配置会略有不同，如果现在snmpV2方式，会要求配置团体字和snmp oid。

1. 创建触发器：

触发器是对主机的某一个监控项定义一个阈值，如果监控项的值触发了触发器所定义的告警条件，那么zabbix会通过邮件、短信、电话等方式来通知相关人员。





如上图，触发器的表达式类型比较多，基本可以满足触发器需要的触发条件，详细的表达式可以自己查看zabbix相关文档。

### 配置动作

1. 触发器

触发器动作是为了在监控项触发器触发之后，zabbix做的一系列操作，比如发送消息给用户，发送命令给主机来修复告警。动作里面可以配置动作发生所需要的条件。



如上图所示， 配置—动作，右上角事件源选择触发器，点击创建动作



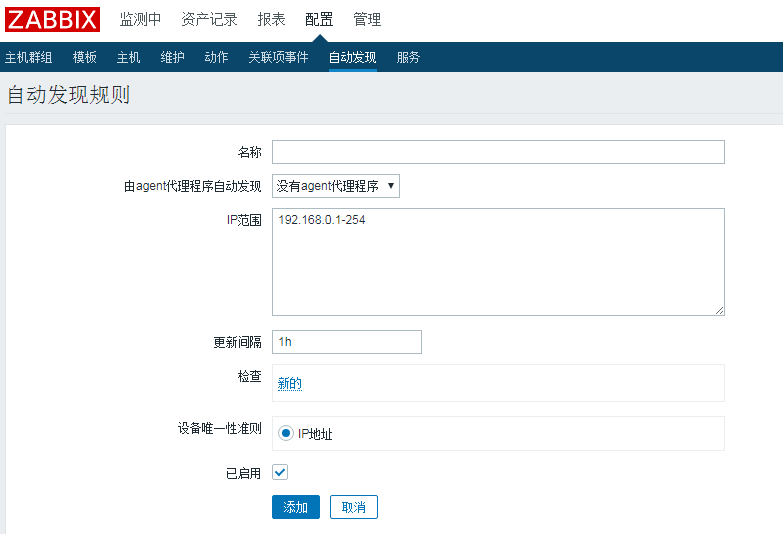
需要配置的包括：触发条件，触发之后的操作，以及恢复操作，根据实际情况来自定义配置。

1. 自动发现

当需要监控的主机很多，不方便一个个添加的时候，可以通过自动发现让zabbix自动完成主机添加。自动发现分为三个步骤：配置自动发现规则、查看自动发现主机、配置自动发现动作。

第一步，配置自动发现规则，配置---自动发现，如下图所示：



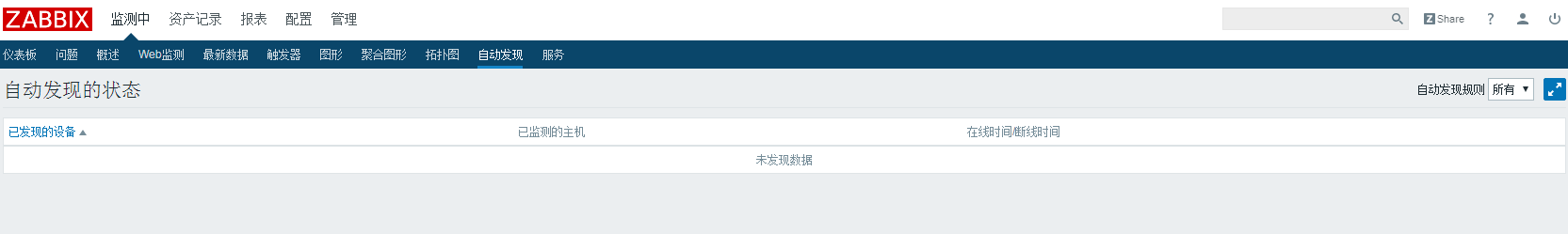


如上配置，zabbix每小时扫描192.168.0.1-254。来判断客户端是否存在，并且以IP地址作为唯一性的标识。

|  |  |
| --- | --- |
| 属性 | 描述 |
| Name | 规则名称，唯一 |
| Discovery by proxy | 谁执行当前发现规则: no proxy - zabbix server <proxy name> - 指定的proxy |
| IP range | 发现规则中的ip范围，格式如下 单IP: 192.168.1.33 一个IP段: 192.168.1.1-255 一个子网: 192.168.4.0/24 支持如下子网掩码: IPV4:/16 - /30 IPV6:/112 - /128 IP列表: 192.168.1.1-255,192.168.2.1-100,192.168.2.200,192.168.4.0/24 备注：1. IP列表中的IP不能重复2. 不同的发现规则里面不要包含相同的IP |
| Delay (in sec) | 规则执行完毕之后，要多久才执行下一次。 |
| Checks | 支持的checks: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping. Port属性如下： 单个端口: 22 端口段: 22-45 端口列表: 22-45,55,60-70 |
| Device uniqueness criteria | 设备唯一标识: IP address - 使用IP地址作为设备唯一性标识 Type of discovery check - 使用SNMP 或者Zabbix agent的check作为唯一标识 |
| Enabled | 是否启用当前规则 |

第二步，查看自动发现的主机，如下图所示：

监测中--自动发现



如上图所示，自动发现的主机会在上面显示，并且会以ip地址作为主机的唯一标识符。

第三步，配置自动发现动作

如下图所示， 配置—动作，右上角事件源选择自动发现，点击创建动作。配置动作条件和操作。







1. 自动注册

自动注册是zabbix客户端主动注册到zabbix服务端的。





如上图所示，配置—动作，右上角事件源选择自动注册，配置动作条件和操作，服务端会根据条件来来进行相关操作。

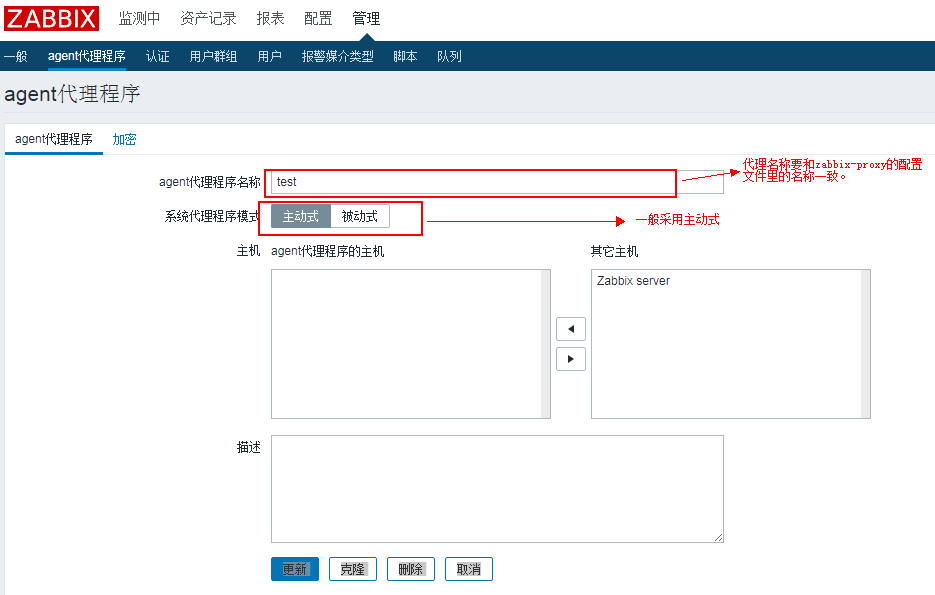
### 配置代理

在分布式监控中，往往一台zabbix server不能满足需求，需要在各个节点增加一台zabbix代理，来分担服务端的压力。

zabbix-proxy安装配置好之后，还需要在web界面添加代理，如下图所示：



点击创建代理



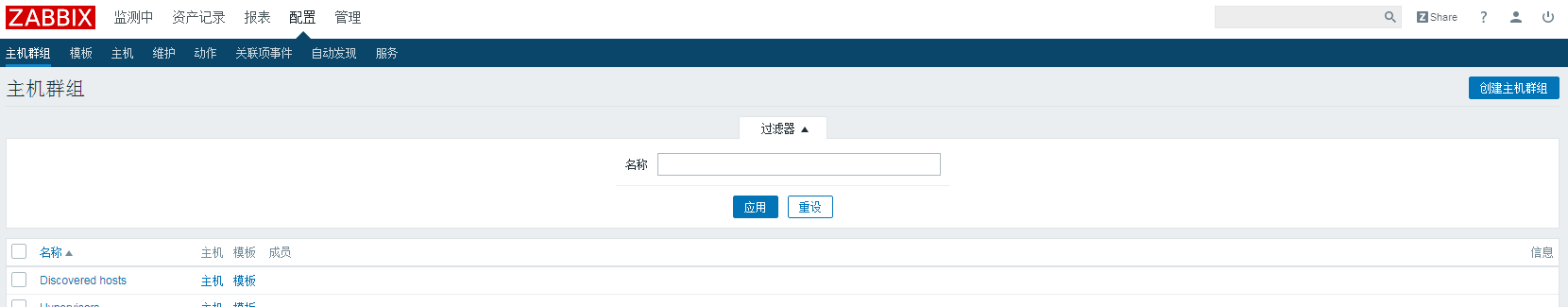
注意代理的名称要和zabbix-proxy的配置文件中的名称一致。

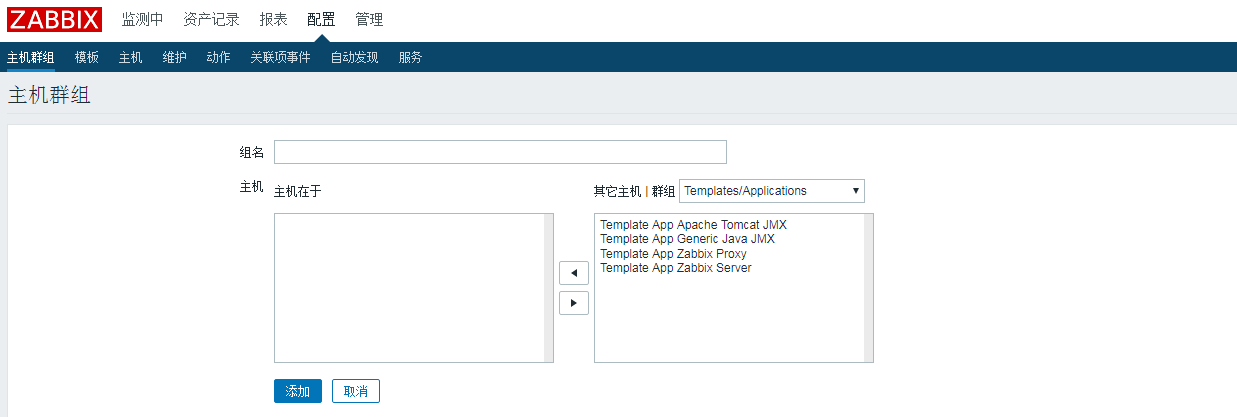
### 添加主机组

主机群组可以给主机和模板分组，在配置动作和用户权限的时候，或者调用zabbix api分批处理主机信息的时候，主机组的存在要方便的多。

如下图所示：

配置—主机群组，选择创建主机群组：



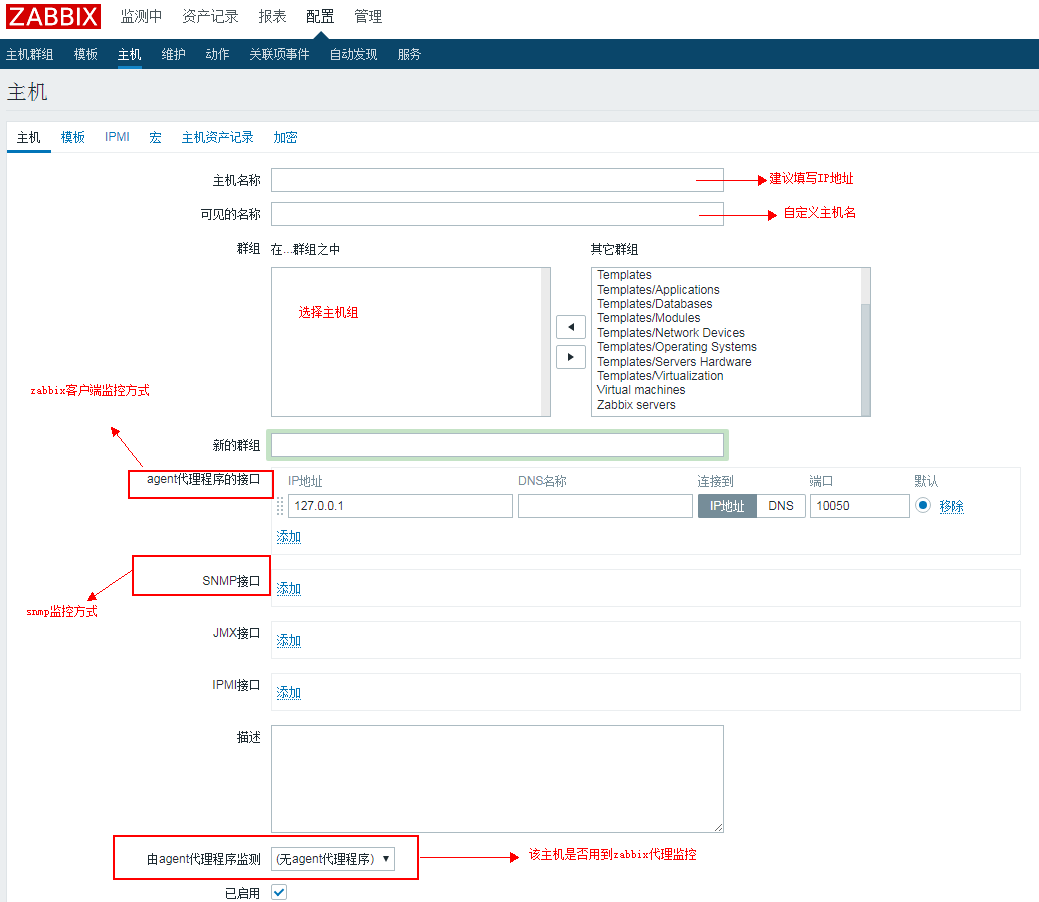


配置主机群组的名称，选择该主机组包含的主机，这里的主机可以是模板也可以是具体的主机。

### 添加主机

1.添加要监控的主机，配置—主机，点击右上角创建主机，如下图所示：





主机名称：建议填写IP地址

可见的主机名：自定义

群组：选择主机所在组

接口：选择监控方式，目前用到的是snmp和agent

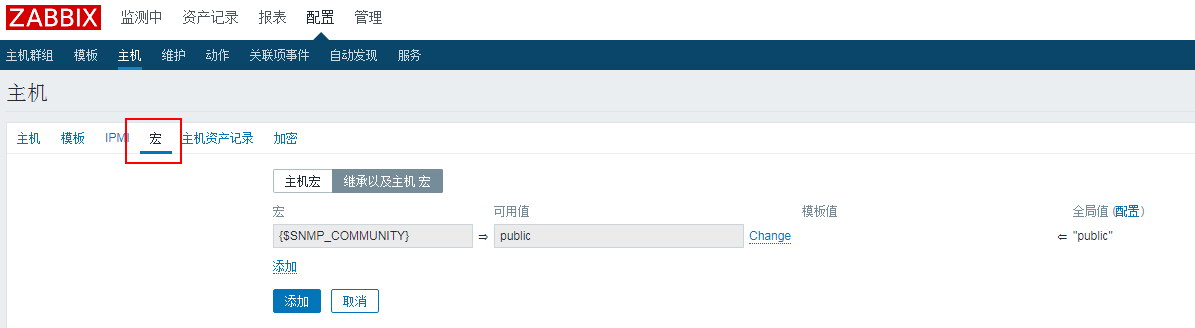
Agent代理程序：如用到代理，需要选择代理

2.添加监控模板



选择监控主机需要的模板。

3.宏



一般用到的是全局宏，此处检查一下用到的宏是否正确。

3.主机资产记录



资产记录配置主机的信息，可以自动获取也可以手动填写，自动获取只能取到部分的信息。

### 配置正则表达式

正则表达式主要用在监控项自动发现里面，过滤自动发现的监控项

管理 >> 一般，选择正在表达式，如下图：



### 配置全局宏

目前用到的全局宏是snmp团体字

管理 >> 一般，选择宏，如图：



## 其他

### 测试服务端和客户端通信

命令如下：

1. 测试目标主机172.31.1.3的snmpv2配置是否可用：

在zabbix proxy执行以下命令:

snmpwalk -v2c -c public 172.31.1.3 .1.3.1

（注意团体字和测试的oid）

2.测试目标主机172.31.1.3的agent是否可用

在zabbix proxy执行以下命令：

zabbix\_get -s 172.31.1.3 -p 10050 -k system.hostname

## Zabbix调优

### zabbix性能比对：

|  |  |  |
| --- | --- | --- |
| 因素 | 慢 | 快 |
| 数据库大小 | 巨大 | 适应内存的大小 |
| 触发器表达式的复杂程度 | max 、min、avg | last、nodata |
| 数据收集的方法 | 轮询 | trapper |
| 数据类型 | 文本，字符串 | 数值 |
| 用户数量 | 多 | 少 |

### zabbix性能不足的表现：

* zabbix队列有太多被延迟的item
* zabbix绘图中经常出现断图，缺失数据
* 前端页面反应慢

### 调优步骤：

* 查看zabbix performance状态
* 确定和修复常见的一般问题
* 优化zabbix配置参数
* 优化数据库参数

1.查看status of zabbix，number of values processed per sencond

一般4核8gb的服务器，zabbix每分钟可以传输一百万个值，每秒15000值，但是实际生产环境下，表现远远不及这么优秀。

2.什么导致zabbix server性能低下？

items的类型，trigger的计算方法，trigger的数量，以及收集数据的类型snmp、agent等。

清理历史数据的参数设置，数据库的大小

zabbix的用户活跃数量

trigger的复杂程度

3.一般的调优初始化手段

用自己自定义的更加简单的模板

优化数据库参数

优化zabbix server的配置参数

历史清理数据的频率和大小

用最新的版本

4.开启zabbix本身的监控

查看本身的性能指标

5.分布式监控

利用zabbix proxy，减轻zabbix server的压力。

6.数据库表拆分

将大型的表按时间拆分，方便查询和清理，history，trends，event

7.zabbix server 、数据库mysql、apache、服务器分开部署

## 数据库分区

数据库分区的目的是为了避免数据库由于数据量过大造成的性能下降，目前支持mysql分区的zabbix版本有2.0 ,2.2, 2.4,3.0以及3.2,3.4，zabbix 2.0+版本的数据库表trends和history没有用外键，因此这些表支持分区。

### 连接到数据库

mysql -h <database\_ip/fqdn> -u<user> -p

mysql> use zabbix;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

### 更改索引（zabbix 3.2版本跳过该步骤）

mysql> Alter table history\_text drop primary key, add index (id), drop index history\_text\_2, add index history\_text\_2 (itemid, id);

Query OK, 0 rows affected (0.49 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> Alter table history\_log drop primary key, add index (id), drop index history\_log\_2, add index history\_log\_2 (itemid, id);

Query OK, 0 rows affected (2.71 sec)

Records: 0 Duplicates: 0 Warnings: 0

### 添加存储过程

DELIMITER $$

CREATE PROCEDURE `partition\_create`(SCHEMANAME varchar(64), TABLENAME varchar(64), PARTITIONNAME varchar(64), CLOCK int)

BEGIN

/\*

SCHEMANAME = The DB schema in which to make changes

TABLENAME = The table with partitions to potentially delete

PARTITIONNAME = The name of the partition to create

\*/

/\*

Verify that the partition does not already exist

\*/

DECLARE RETROWS INT;

SELECT COUNT(1) INTO RETROWS

FROM information\_schema.partitions

WHERE table\_schema = SCHEMANAME AND table\_name = TABLENAME AND partition\_description >= CLOCK;

IF RETROWS = 0 THEN

/\*

1. Print a message indicating that a partition was created.

2. Create the SQL to create the partition.

3. Execute the SQL from #2.

\*/

SELECT CONCAT( "partition\_create(", SCHEMANAME, ",", TABLENAME, ",", PARTITIONNAME, ",", CLOCK, ")" ) AS msg;

SET @sql = CONCAT( 'ALTER TABLE ', SCHEMANAME, '.', TABLENAME, ' ADD PARTITION (PARTITION ', PARTITIONNAME, ' VALUES LESS THAN (', CLOCK, '));' );

PREPARE STMT FROM @sql;

EXECUTE STMT;

DEALLOCATE PREPARE STMT;

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE `partition\_drop`(SCHEMANAME VARCHAR(64), TABLENAME VARCHAR(64), DELETE\_BELOW\_PARTITION\_DATE BIGINT)

BEGIN

/\*

SCHEMANAME = The DB schema in which to make changes

TABLENAME = The table with partitions to potentially delete

DELETE\_BELOW\_PARTITION\_DATE = Delete any partitions with names that are dates older than this one (yyyy-mm-dd)

\*/

DECLARE done INT DEFAULT FALSE;

DECLARE drop\_part\_name VARCHAR(16);

/\*

Get a list of all the partitions that are older than the date

in DELETE\_BELOW\_PARTITION\_DATE. All partitions are prefixed with

a "p", so use SUBSTRING TO get rid of that character.

\*/

DECLARE myCursor CURSOR FOR

SELECT partition\_name

FROM information\_schema.partitions

WHERE table\_schema = SCHEMANAME AND table\_name = TABLENAME AND CAST(SUBSTRING(partition\_name FROM 2) AS UNSIGNED) < DELETE\_BELOW\_PARTITION\_DATE;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

/\*

Create the basics for when we need to drop the partition. Also, create

@drop\_partitions to hold a comma-delimited list of all partitions that

should be deleted.

\*/

SET @alter\_header = CONCAT("ALTER TABLE ", SCHEMANAME, ".", TABLENAME, " DROP PARTITION ");

SET @drop\_partitions = "";

/\*

Start looping through all the partitions that are too old.

\*/

OPEN myCursor;

read\_loop: LOOP

FETCH myCursor INTO drop\_part\_name;

IF done THEN

LEAVE read\_loop;

END IF;

SET @drop\_partitions = IF(@drop\_partitions = "", drop\_part\_name, CONCAT(@drop\_partitions, ",", drop\_part\_name));

END LOOP;

IF @drop\_partitions != "" THEN

/\*

1. Build the SQL to drop all the necessary partitions.

2. Run the SQL to drop the partitions.

3. Print out the table partitions that were deleted.

\*/

SET @full\_sql = CONCAT(@alter\_header, @drop\_partitions, ";");

PREPARE STMT FROM @full\_sql;

EXECUTE STMT;

DEALLOCATE PREPARE STMT;

SELECT CONCAT(SCHEMANAME, ".", TABLENAME) AS `table`, @drop\_partitions AS `partitions\_deleted`;

ELSE

/\*

No partitions are being deleted, so print out "N/A" (Not applicable) to indicate

that no changes were made.

\*/

SELECT CONCAT(SCHEMANAME, ".", TABLENAME) AS `table`, "N/A" AS `partitions\_deleted`;

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE `partition\_maintenance`(SCHEMA\_NAME VARCHAR(32), TABLE\_NAME VARCHAR(32), KEEP\_DATA\_DAYS INT, HOURLY\_INTERVAL INT, CREATE\_NEXT\_INTERVALS INT)

BEGIN

DECLARE OLDER\_THAN\_PARTITION\_DATE VARCHAR(16);

DECLARE PARTITION\_NAME VARCHAR(16);

DECLARE OLD\_PARTITION\_NAME VARCHAR(16);

DECLARE LESS\_THAN\_TIMESTAMP INT;

DECLARE CUR\_TIME INT;

CALL partition\_verify(SCHEMA\_NAME, TABLE\_NAME, HOURLY\_INTERVAL);

SET CUR\_TIME = UNIX\_TIMESTAMP(DATE\_FORMAT(NOW(), '%Y-%m-%d 00:00:00'));

SET @\_\_interval = 1;

create\_loop: LOOP

IF @\_\_interval > CREATE\_NEXT\_INTERVALS THEN

LEAVE create\_loop;

END IF;

SET LESS\_THAN\_TIMESTAMP = CUR\_TIME + (HOURLY\_INTERVAL \* @\_\_interval \* 3600);

SET PARTITION\_NAME = FROM\_UNIXTIME(CUR\_TIME + HOURLY\_INTERVAL \* (@\_\_interval - 1) \* 3600, 'p%Y%m%d%H00');

IF(PARTITION\_NAME != OLD\_PARTITION\_NAME) THEN

CALL partition\_create(SCHEMA\_NAME, TABLE\_NAME, PARTITION\_NAME, LESS\_THAN\_TIMESTAMP);

END IF;

SET @\_\_interval=@\_\_interval+1;

SET OLD\_PARTITION\_NAME = PARTITION\_NAME;

END LOOP;

SET OLDER\_THAN\_PARTITION\_DATE=DATE\_FORMAT(DATE\_SUB(NOW(), INTERVAL KEEP\_DATA\_DAYS DAY), '%Y%m%d0000');

CALL partition\_drop(SCHEMA\_NAME, TABLE\_NAME, OLDER\_THAN\_PARTITION\_DATE);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE `partition\_verify`(SCHEMANAME VARCHAR(64), TABLENAME VARCHAR(64), HOURLYINTERVAL INT(11))

BEGIN

DECLARE PARTITION\_NAME VARCHAR(16);

DECLARE RETROWS INT(11);

DECLARE FUTURE\_TIMESTAMP TIMESTAMP;

/\*

\* Check if any partitions exist for the given SCHEMANAME.TABLENAME.

\*/

SELECT COUNT(1) INTO RETROWS

FROM information\_schema.partitions

WHERE table\_schema = SCHEMANAME AND table\_name = TABLENAME AND partition\_name IS NULL;

/\*

\* If partitions do not exist, go ahead and partition the table

\*/

IF RETROWS = 1 THEN

/\*

\* Take the current date at 00:00:00 and add HOURLYINTERVAL to it. This is the timestamp below which we will store values.

\* We begin partitioning based on the beginning of a day. This is because we don't want to generate a random partition

\* that won't necessarily fall in line with the desired partition naming (ie: if the hour interval is 24 hours, we could

\* end up creating a partition now named "p201403270600" when all other partitions will be like "p201403280000").

\*/

SET FUTURE\_TIMESTAMP = TIMESTAMPADD(HOUR, HOURLYINTERVAL, CONCAT(CURDATE(), " ", '00:00:00'));

SET PARTITION\_NAME = DATE\_FORMAT(CURDATE(), 'p%Y%m%d%H00');

-- Create the partitioning query

SET @\_\_PARTITION\_SQL = CONCAT("ALTER TABLE ", SCHEMANAME, ".", TABLENAME, " PARTITION BY RANGE(`clock`)");

SET @\_\_PARTITION\_SQL = CONCAT(@\_\_PARTITION\_SQL, "(PARTITION ", PARTITION\_NAME, " VALUES LESS THAN (", UNIX\_TIMESTAMP(FUTURE\_TIMESTAMP), "));");

-- Run the partitioning query

PREPARE STMT FROM @\_\_PARTITION\_SQL;

EXECUTE STMT;

DEALLOCATE PREPARE STMT;

END IF;

END$$

DELIMITER ;

备注：以上代码可以保存成sql文件直接导入数据库：

mysql  -uroot -p'123456' zabbix < mysql\_partition.sql

### 自定义过程

存储过程解释：

mysql>  CALL partition\_maintenance('<zabbix\_db\_name>', '<table\_name>', <days\_to\_keep\_data>, <hourly\_interval>, <num\_future\_intervals\_to\_create>)

zabbix\_db\_name：库名

table\_name：表名

days\_to\_keep\_data：保存多少天的数据

hourly\_interval：每隔多久生成一个分区

num\_future\_intervals\_to\_create：本次一共生成多少个分

自定义分区周期：

DELIMITER $$

CREATE PROCEDURE `partition\_maintenance\_all`(SCHEMA\_NAME VARCHAR(32))

BEGIN

CALL partition\_maintenance(SCHEMA\_NAME, 'history', 28, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'history\_log', 28, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'history\_str', 28, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'history\_text', 28, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'history\_uint', 28, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'trends', 730, 24, 14);

CALL partition\_maintenance(SCHEMA\_NAME, 'trends\_uint', 730, 24, 14);

END$$

DELIMITER ;

备注：以上sql也可以保存成sql文件导入数据库：

mysql  -uroot -p'123456' zabbix < partition\_call.sql

### 设置定时任务

每天执行调用一次存储过程，crontab设置

mysql -uzabbix -pzabbix zabbix -e "CALL partition\_maintenance\_all('zabbix');"

### 关闭housekeeper自动清理

Zabbix控制台 "Administration" -> "General"，选择housekeeper关闭

**注意事项：例子中分区创建了14天，当过了14天没有创建新的分区，数据将会丢失，因此，要定时去调用存储过程创建新的分区。**

### 操作步骤

1. 修改自定义分区周期，可以用默认
2. 执行脚本添加存储过程：bash +x mysql\_partitions.sh
3. Zabbix控制台暂停housekeeper
4. 添加定时任务每日执行一次mysql -uzabbix -pzabbix zabbix -e "CALL partition\_maintenance\_all('zabbix');"

### 参考

https://www.zabbix.org/wiki/Docs/howto/mysql\_partition#Index\_changes\_.282.x.2C\_3.0\_only.29

## Zabbix客户端探针部署

### Snmp

* 安装snmp服务

安装snmp服务需要的rpm包：

libsensors3-2.10.6-55.el5.i386.rpm

lm\_sensors-2.10.6-55.el5.i386.rpm  
以下这三个包有一定的版本要求，必须版本一致，否则安装无法成功

net-snmp-5.7.2-28.el7.x86\_64.rpm

[net-snmp-libs-5.7.2-28.el7.x86\_64.rpm](http://rpmfind.net/linux/centos/7.4.1708/os/x86_64/Packages/net-snmp-libs-5.7.2-28.el7.x86_64.rpm)

[net-snmp-utils-5.7.2-28.el7.x86\_64.rpm](http://rpmfind.net/linux/centos/7.4.1708/os/x86_64/Packages/net-snmp-utils-5.7.2-28.el7.x86_64.rpm)  不是必须安装的包，但安装它会带来不少帮助，它提供了很多工具，例如可以使用snmptranslate命令查看oid，可以使用snmpget、snmpwalk命令等。

* 配置snmp

装好之后修改/etc/snmp/snmpd.conf对其进行配置，备份原来的snmp配置文件，编辑/etc/snmp/snmpd.conf，简单配置如下：

# First, map the community name "public" into a "security name"

#       sec.name  source          community

com2sec notConfigUser  default       public

####

# Second, map the security name into a group name:

#       groupName      securityModel securityName

group   notConfigGroup v1           notConfigUser

group   notConfigGroup v2c           notConfigUser

####

# Third, create a view for us to let the group have rights to:

# Make at least  snmpwalk -v 1 localhost -c public system fast again.

#       name           incl/excl     subtree         mask(optional)

view    systemview    included   .1

view all included .1 80

view mib2 included .iso.org.dod.internet.mgmt.mib-2 fc

####

# Finally, grant the group read-only access to the systemview view.

#       group          context sec.model sec.level prefix read   write  notif

access  notConfigGroup ""      any       noauth    exact  all none none

* 重启snmp服务

### Zabbix agent

* 下载zabbix agent安装包

<http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/>

* 安装zabbix agent

rpm –ivh [zabbix-agent-3.4.0-1.el7.x86\_64.rpm](http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/zabbix-agent-3.4.0-1.el7.x86_64.rpm)

* 配置文件更改

sed -i "s/Server=127.0.0.1/Server=10.0.0.101/" /etc/zabbix/zabbix\_agentd.conf

sed –i "s/ServerActive=127.0.0.1/ServerActive=10.0.0.101/" /etc/zabbix/zabbix\_agentd.conf （主动模式）

sed -i "s/Hostname=Zabbix server/Hostname=10.0.0.105/" /etc/zabbix/zabbix\_agentd.conf

* 启动并设置开机启动

systemctl restart zabbix-agent.service

systemctl enable zabbix-agent.service

### 参考

* 下载rpm包地址：

http://rpmfind.net/linux/rpm2html/search.php?query=&submit=Search+...&system=&arch=

# 云监控部署步骤

### Zabbix服务器硬件需求

由于数据库读写比较频繁，因此最好硬盘选择ssd，阵列选择raid 1或raid 10，不要用raid 5

### 安装zabbix server和zabbix proxy

按照4.1进行安装

### 设备清单

1.《设备清单模板》提供给厂商，厂商需按模板反馈相关信息

2.厂商供监控主机的具体信息，包括主机名、IP地址、硬件信息、监控方式等。

3.厂商配置好snmp或者zabbix agent，需要测试是否能取到数据。

参考4.3.1

### 登录zabbix

参考4.2.1

### 导入模板

1. 导出zabbix3.4默认模板进行备份
2. 删除全部zabbix 3.4模板
3. 导入提供的默认监控模板CETC Templates Default

[CETC Template SNMP OS Linux Default](http://172.31.6.112/zabbix/templates.php?form=update&templateid=10260&groupid=0) 存储节点和计算节点默认模板

[CETC Template SNMP Switch Default](http://172.31.6.112/zabbix/templates.php?form=update&templateid=10262&groupid=0) 网络设备默认模板

网络设备还需要添加CPU和内存使用情况的模板，需要自行查找对应

设备型号的OID

参考4.2.2

### 添加主机组

参考4.2.6

### 添加全局宏

参考4.2.9

### 添加正则表达式

参考4.2.8

如下图所示，添加两个新的正则表达式



### 添加zabbix代理程序

参考4.2.5

### 启用触发器告警动作



### 添加主机

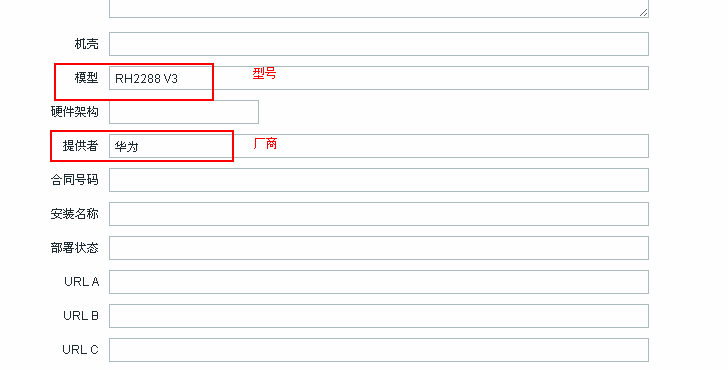
包括：添加主机、关联模板、配置资产记录

参考4.2.7

资产记录主要配置如下：







### 触发器名称可读

要求触发器名称中文并且有良好的可读性。

# 结束语

文档初版，后续需要持续编写优化。