


This repository | Search

Pull requests

Issues

Marketplace

Gist



kanghuikai / uplooking

Unwatch 2

Star 1

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki


Settings

Insights

Branch: master

uplooking / UP200\_zabbix-master / 集中化监控工具zabbix.md

Find fileCopy path

 kanghuikai Update 集中化监控工具zabbix.md88fcd59 Aug 6, 2017

1 contributor

Executable File598 lines (384 sloc)28 KB

RawBlameHistory

# 集中化监控工具zabbix

[TOC]

## ##系统监控需求

在互联网的应用中，我们大量的使用网络和服务器来完成项目平台的搭建，在系统运维过程中，我们需要获取很多网络和服务

器运行的信息，比如：网络流量、主机CPU 使用率、主机内存使用率、主机磁盘使用率等。这些信息我们可以通过网络或主机

设备上的命令行指令获取，但这样很不方便，尤其是主机和设备非常多的情况下获取这些信息。想象一下我们管理的是由1000

多台物理服务器和50 多台网络交换或路由设备组成的一个私有云，如果我们需要人工的获取这些服务器和设备的运行状态、网

络流量和服务状态是非常耗时且无法完成的。更不要说在这1000 多台物理服务器上运行着3000 或更多的虚拟服务器，每个服

务器上又运行着相互关联的负责均衡集群、分布式计算集群、分布式存储集群、高性能计算集群等复杂的应用集群方案。我们

必须实时的获取它们的运行状态，并且分析在过去1 小时、24 小时、1 周、1 月、1 年的数据变化趋势。我们需要获取和保持

这些数据并且在需要的时候能够方便的展示出它们的可视化视图。

不仅仅是获取实时的运行数据，我们还需要有一套完整的报警机制。在服务器、网络设备、资源服务、业务逻辑等出现异常的

时候可以通过WEB、邮件、短信、电话等方式自动化的通知管理人员。这种报警机制必须是可配置的，可以根据我们系统应用

的具体情况做出合理的方案，比如设置CPU 使用阈值、报告内存突然的增长、服务器流量异常、应用服务负载异常、集群服务

故障迁移等等情况。

随着项目的推进，我们还需要通过分析记录数据预估出未来可能需要的资源容量，例如服务器数量是否需要增添及何时增

添、网络带宽是否出现瓶颈及何时增添、应用服务器是否需要平行宽展及何时扩展等等。这些都需要大量且长期的数据分析支

撑，也就是长期有效的系统监控数据支撑。

## ##常见的监控解决方案

早期，网络环境中的设备数量尚且不多的时候，人们需要监控的仅仅是几台服务器、一些交换机以及少量的其他设备。首先就

需要有方法将人们关心的各项数据从设备中提取出来，然后再考虑如何集中化的展示出来。

为了满足提取各项数据的需求，互联网工程工作小组（IETF,Internet Engineering Task Force）定义并发布了简单网络管理协

议（SNMP）。该协议能够支持网络管理系统，用以监控连接到网络上的各种设备。Linux系统当然也支持该协议。人们可以使

用SNMP协议获取主机CPU、内存、磁盘、网卡流量等数据，很多商业网络设备如路由器、交换机等基本也支持SNMP协议。

最初，人们用脚本将获取到的SNMP数据存入数据库中，然后再使用一种名为MRTG的软件根据获取的数据绘制图表来分析数

据的变化。MRTG(Multi Router Traffic Grapher), 顾名思义, 这款软件最初是设计用于监控网络链路流量负载的。它可以用过SNMP获取到设备的流量信息, 并根据这些信息绘制成图表并保存为PNG格式的图片, 再将这些PNG图片以HTML页面的方式显示给用户。

不过, MRTG展示的页面和图表曲线相对简陋, 它在一张图片中最多只能绘制两个数据的变化曲线, 并且由于是PNG格式的静态图片, 所以无法针对某一时间进行细化展示。为了解决这个问题, 人们又开发了RRDTOOL工具。

RRDTOOL(Round Robin Database Tool)是一个强大的绘图引擎。RRDTOOL可以使用自己的环状数据库(RRD)来记录绘图数据, 也可以使用关系型数据库, 例如MySQL来纪录数据。利用RRDTOOL我们就可以将获取到的SNMP数据通过RRDTOOL进行实时绘图, 这么一来, 想看什么时候的数据, 只要调取那个时间的监控数据, 让RRDTOOL帮我们画一张图出来就可以直观又清晰的查看了。目前, 几乎所有的开源监控软件都在使用RRDTOOL作为图形绘制引擎。

不过, 直接使用RRDTOOL绘图操作起来很麻烦。同时, 现如今的数据中心动辄成百上千的设备, 一个个的去提取、绘制、监控显然是不现实的事情。况且, 不同的设备监控的对象和方法也不同。有些数据依靠SNMP也提取不到(例如MySQL查询缓存命中率)。我们迫切的需要一个可以支持多种不同的监控手段, 同时具有分类组织图形并且可以按照分类页面的方式展示这些图形的工具。另外, 考虑到企业组织结构越来越复杂, 参与监控的人员往往属于不同的部门或者工作组, 不同的人也应该限定具有不同层级的监控浏览权限。除此之外, 还有很多各种各样杂七杂八的需求不一而足。所以, 现今的监控软件不可以再像MRTG那样仅仅画张图就了事, 还需要考虑更加丰富的功能。

Cacti 是一套基于PHP、MySQL、SNMP 及RRDTool 开发的监测图形分析工具。它提供了非常强大的数据和用户管理功能, 可以指定每一个用户能查看树状结构、主机以及任何一张图, 还可以与轻量级目录服务(LDAP) 结合进行用户验证, 同时也能自己编写模板, 功能非常强大完善界面友好。可是随着设备的不断增多, Cacti 的运行机制就不能满足我们的需求了。Cacti 是使用轮询的方式由主服务器向设备发送数据请求来获取设备上状态数据信息的, 如果设备不断增多, 这个轮询的过程就非常的耗时, 轮询的结果就不能即时的反应设备的状态了。

Cacti 监控关注的是对数据的展示, 却不关注数据异常后的反馈。如果凌晨3 点的时候设备的某个数据出现异常, 除非监控人员在屏幕前发现这个异常变化, 否则是没有任何报警机制能够让我们道出现了异常。这个时候我们就需要一种可以根据我们预设阈值报警的工具。自动化监控报警服务有很多, 其中商业应用比较昂贵且多以硬件为主, 开源自动化监控报警服务中最著名的也是应用最广泛的是Nagios。

Nagios 是一款开源的免费网络监控报警服务, 能有效监控Windows、Linux 和Unix 的主机状态, 交换机、路由器和防火墙等网络设置, 打印机、网络投影、网络摄像等设备。在系统或服务状态异常时发出邮件或短信报警第一时间通知运维人员, 在状态恢复后发出正常的邮件或短信通知。Nagios 有完善的插件功能, 可以方便的根据应用服务扩展功能。Nagios 可以使用服务-代理(Server-Agent) 方式获取主机信息, 也可以使用SNMP 方式获取设备信息。使用服务器-代理(Server-Agent) 方式在每个主机上运行代理软件, 主机上的代理软件主动提交数据信息, 可以避免服务器轮询带来的延迟和非实时性等问题。

Nagios 已经可以支持由数万台服务器或上千台网络设备组成的云技术平台的监控, 它可以充分发挥自动化运维技术特点在设备和人力资源减少成本。只是Nagios 无法将多个相同应用集群的数据集合起来, 也不能监控到集群中特殊节点的迁移和恢复。

## ##zabbix是什么

Zabbix是一个基于WEB界面的提供分布式系统监视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, 保证服务器系统的安全运营; 并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。

zabbix server可以通过SNMP, zabbix agent, ping, 端口监视等方法提供对远程服务器/网络状态的监视, 数据收集等功能, 它可以运行在Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X等平台上。

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 agent可以运行在Linux,Solaris,HP-UX,AIX,Free BSD,Open BSD, OS X, Tru64/OSF1, Windows NT4.0, Windows (2000/2003/XP/Vista)等系统之上。

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), 可以与SNMP软件(例如: net-snmp)等配合使用。

相比较Cacti, Zabbix具有不亚于Cacti的图形表现效果, 同时还支持监控报警功能。并且内置的监控模板比Cacti要丰富的多得多。

相比较Nagios, Zabbix安装配置要简单的多, 利于实现。同时学习成本也要低得多。在监控性能方面也不亚于Nagios。因此, 目前越来越多的公司开始使用Zabbix作为监控的最佳解决方案。

## zabbix的优缺点

### 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。

## zabbix基于agent采集数据的方式

### 被动模式

Zabbix服务器端按照监控配置项目制定的时间间隔，定期连接到agent上的守护进程，获取agent组件采集到的数据。

### ###主动模式

Agent端从server端获取采集数据的监控项目列表及其配置信息。而agent端确定所需要的连接zabbix服务器代理的方法是，从agent的配置文件中读取serveractive该配置项的内容。读取频率RefreshActiveChecks指定，默认120s。

主动模式的好处：当被监控设备代理组件所封装的监控项目种类不能满足需要的时候，完全可以使用我们擅长的任何一种高级语言来开发小的工具或者脚本模拟一个组件，从而实现个性化需求。然而通过小工具或者脚本模拟组件时，一般来说使用主动模式会比较方便。原因就是：使用主动模式时，这类小工具不需要以守护进程的方式运行，更不需要在被监控设备上开额外的服务端口，只需这类小工具或脚本能定时将所采集到的数据发送至zabbix服务器或其代理指定的端口上。

## zabbix基本组件的安装配置

zabbix的安装包括四个部分：server服务端、agent数据收集端、web配置管理端、数据保存数据库端。

架构如下：

设备	应用端
servera	agent
serverb	server
serverc	web
serverd	database

先保证我们所有服务器的时区都是Asia/shanghai，分别在每台服务器上执行以下命令

```
[root@servera ~]# timedatectl set-timezone Asia/Shanghai
```

### Server端的安装

软件安装，这里server端通过源码编译的方式，将服务主目录放置/usr/local/zabbix目录下：

```
[root@serverb ~]# setenforce 0
```

```
[root@serverb zabbix]# cp zabbix-2.4.6.tar.gz /tmp
[root@serverb zabbix]# cd /tmp
[root@serverb tmp]# tar -xf zabbix-2.4.6.tar.gz # 这里采用源码编译的方式来安装
[root@serverb tmp]# yum install gcc gcc-c++ mariadb-devel libxml2-devel net-snmp-devel libcurl-devel -y
# 安装源码编译需要的依赖包
[root@serverb tmp]# cd zabbix-2.4.6/
[root@serverb zabbix-2.4.6]# ./configure --prefix=/usr/local/zabbix-server --enable-server --with-mysql --
[root@serverb zabbix-2.4.6]# make
[root@serverb zabbix-2.4.6]# make install
[root@serverb zabbix-2.4.6]# useradd zabbix # 添加zabbix用户用以服务zabbix server端的正常运行
```

服务端的数据库连接配置：

```
[root@serverb zabbix-2.4.6]# cd /usr/local/zabbix/
[root@serverb zabbix]# ls
bin etc lib sbin share
[root@serverb zabbix]# cd etc
[root@serverb etc]# vim zabbix_server.conf # 该文件为服务端的主配置文件，变更如下参数
DBHost=172.25.0.13
DBName=zabbix
DBUser=zabbix
DBPassword=uplooking
```

## Database端的安装

安装mariadb-server软件：

```
[root@serverd ~]# setenforce 0
[root@serverd ~]# yum -y install mariadb-server
[root@serverd ~]# systemctl start mariadb
```

登录上serverb这台服务器，将sql语句远程复制到数据库服务器上：

```
[root@serverb database]# cd /tmp/zabbix-2.4.6/database/mysql/ # 注意：需要导入的数据在server端。
[root@serverb mysql]# scp * 172.25.0.13:/tmp
```

mysql服务器将对应的sql语句进行导入的操作，三个sql文件的导入顺序不能出错：

```
[root@serverd ~]# mysql
MariaDB [(none)]> create database zabbix;
MariaDB [(none)]> Bye
[root@serverd ~]# mysql zabbix < /tmp/schema.sql
[root@serverd ~]# mysql zabbix < /tmp/images.sql
[root@serverd ~]# mysql zabbix < /tmp/data.sql
```

mysql授权，授权给server端及web端：

```
[root@serverd ~]# mysql
MariaDB [(none)]> grant all on zabbix.* to zabbix@'172.25.0.11' identified by 'uplooking';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on zabbix.* to zabbix@'172.25.0.12' identified by 'uplooking';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on zabbix.* to zabbix@'serverb.pod0.example.com' identified by 'uplooking';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on zabbix.* to zabbix@'serverc.pod0.example.com' identified by 'uplooking';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.00 sec)
```

## Web端的安装

安装web端所需要的软件：

```
[root@serverc ~]# setenforce 0  
[root@serverc ~]# yum -y install httpd php php-mysql # 安装lamp环境需要的软件包  
[root@serverc zabbix]# yum -y localinstall zabbix-web-2.4.6-1.el7.noarch.rpm zabbix-web-mysql-2.4.6-1.el7.n  
# 安装依赖包及web包
```

变更web端相关配置文件，指定时区。

```
[root@serverc zabbix]# vim /etc/httpd/conf.d/zabbix.conf  
php_value date.timezone Asia/Shanghai
```

## 启动所有相关服务软件

server端的启动

```
[root@serverb sbin]# cd /usr/local/zabbix/sbin/  
[root@serverb sbin]# ./zabbix_server  
[root@serverb sbin]# ps -ef | grep zabbix  
zabbix    3218      1  0 22:20 ?        00:00:00 ./zabbix_server  
zabbix    3220    3218  0 22:20 ?        00:00:00 ./zabbix_server: configuration syncer [waiting 60 sec for p  
zabbix    3221    3218  0 22:20 ?        00:00:00 ./zabbix_server: db watchdog [syncd alerts config in 0.014  
zabbix    3222    3218  0 22:20 ?        00:00:00 ./zabbix_server: poller #1 [got 0 values in 0.000009 sec, i  
zabbix    3223    3218  0 22:20 ?        00:00:00 ./zabbix_server: poller #2 [got 0 values in 0.000023 sec, i  
zabbix    3224    3218  0 22:20 ?        00:00:00 ./zabbix_server: poller #3 [got 0 values in 0.000011 sec, i  
zabbix    3225    3218  0 22:20 ?        00:00:00 ./zabbix_server: poller #4 [got 0 values in 0.000010 sec, i  
zabbix    3226    3218  0 22:20 ?        00:00:00 ./zabbix_server: poller #5 [got 0 values in 0.000016 sec, i  
zabbix    3227    3218  0 22:20 ?        00:00:00 ./zabbix_server: unreachable poller #1 [got 0 values in 0.0  
zabbix    3228    3218  0 22:20 ?        00:00:00 ./zabbix_server: trapper #1 [processed data in 0.000000 sec  
zabbix    3229    3218  0 22:20 ?        00:00:00 ./zabbix_server: trapper #2 [processed data in 0.000000 sec  
zabbix    3230    3218  0 22:20 ?        00:00:00 ./zabbix_server: trapper #3 [processed data in 0.000000 sec  
zabbix    3231    3218  0 22:20 ?        00:00:00 ./zabbix_server: trapper #4 [processed data in 0.000000 sec  
zabbix    3232    3218  0 22:20 ?        00:00:00 ./zabbix_server: trapper #5 [processed data in 0.000000 sec  
zabbix    3233    3218  0 22:20 ?        00:00:00 ./zabbix_server: icmp pinger #1 [got 0 values in 0.000071 s  
zabbix    3234    3218  0 22:20 ?        00:00:00 ./zabbix_server: alerter [sent alerts: 0 success, 0 fail in  
zabbix    3235    3218  0 22:20 ?        00:00:00 ./zabbix_server: housekeeper [startup idle for 30 minutes]  
zabbix    3236    3218  0 22:20 ?        00:00:00 ./zabbix_server: timer #1 [processed 0 triggers, 0 events i  
zabbix    3237    3218  0 22:20 ?        00:00:00 ./zabbix_server: http poller #1 [got 0 values in 0.001247 s  
zabbix    3238    3218  0 22:20 ?        00:00:00 ./zabbix_server: discoverer #1 [processed 0 rules in 0.0011  
zabbix    3239    3218  0 22:20 ?        00:00:00 ./zabbix_server: history syncer #1 [syncd 0 items in 0.000  
zabbix    3240    3218  0 22:20 ?        00:00:00 ./zabbix_server: history syncer #2 [syncd 0 items in 0.000  
zabbix    3241    3218  0 22:20 ?        00:00:00 ./zabbix_server: history syncer #3 [syncd 0 items in 0.000  
zabbix    3242    3218  0 22:20 ?        00:00:00 ./zabbix_server: history syncer #4 [syncd 0 items in 0.000  
zabbix    3243    3218  0 22:20 ?        00:00:00 ./zabbix_server: escalator [processed 0 escalations in 0.00  
zabbix    3244    3218  0 22:20 ?        00:00:00 ./zabbix_server: proxy poller #1 [exchanged data with 0 pro  
zabbix    3245    3218  0 22:20 ?        00:00:00 ./zabbix_server: self-monitoring [processed data in 0.00001  
root      3247    1100  0 22:20 pts/1  00:00:00 grep --color=auto zabbix
```

web端的启动

```
[root@serverc zabbix]# systemctl restart httpd
```

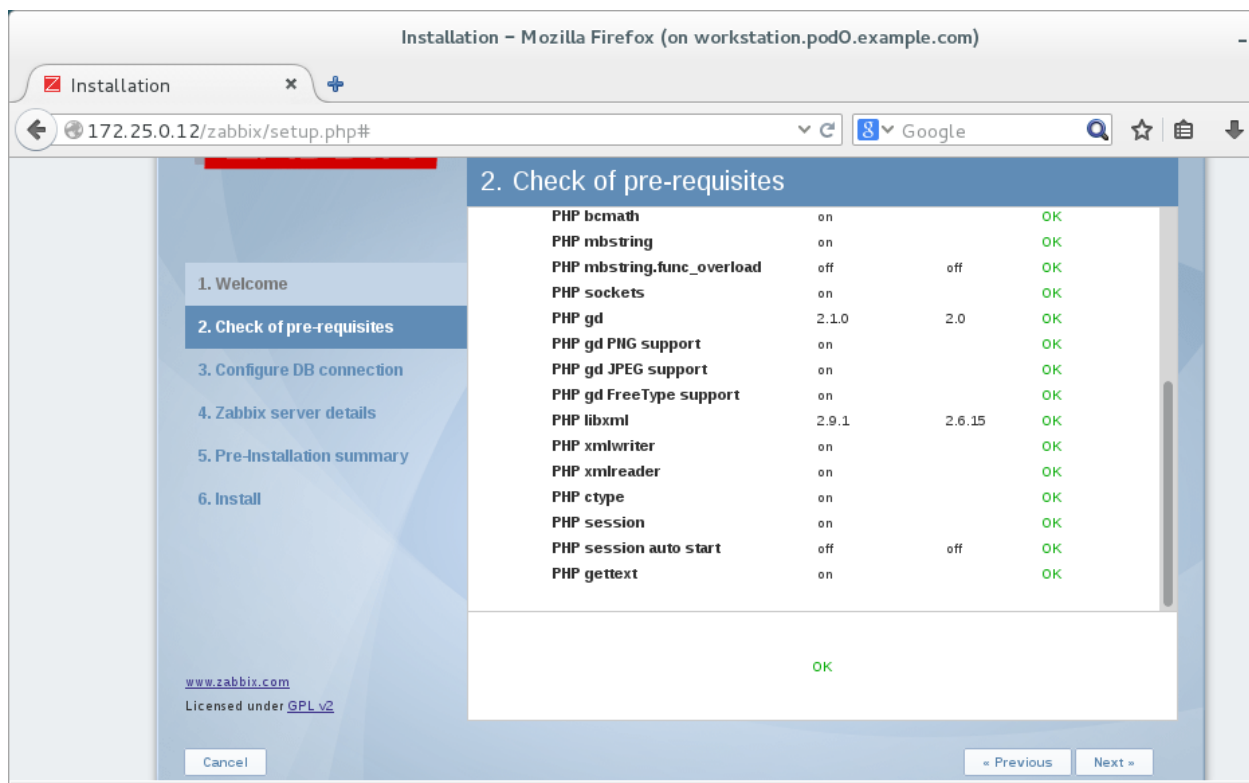
通过图形化界面初始化web端：

登录workstation，打开浏览器访问





保证php环境满足需求



填写数据库相关配置



**ZABBIX**

### 3. Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database.

Press "Test connection" button when done.

Database type	MySQL
Database host	172.25.0.13
Database port	0 0 - use default port
Database name	zabbix
User	zabbix
Password	*****

OK

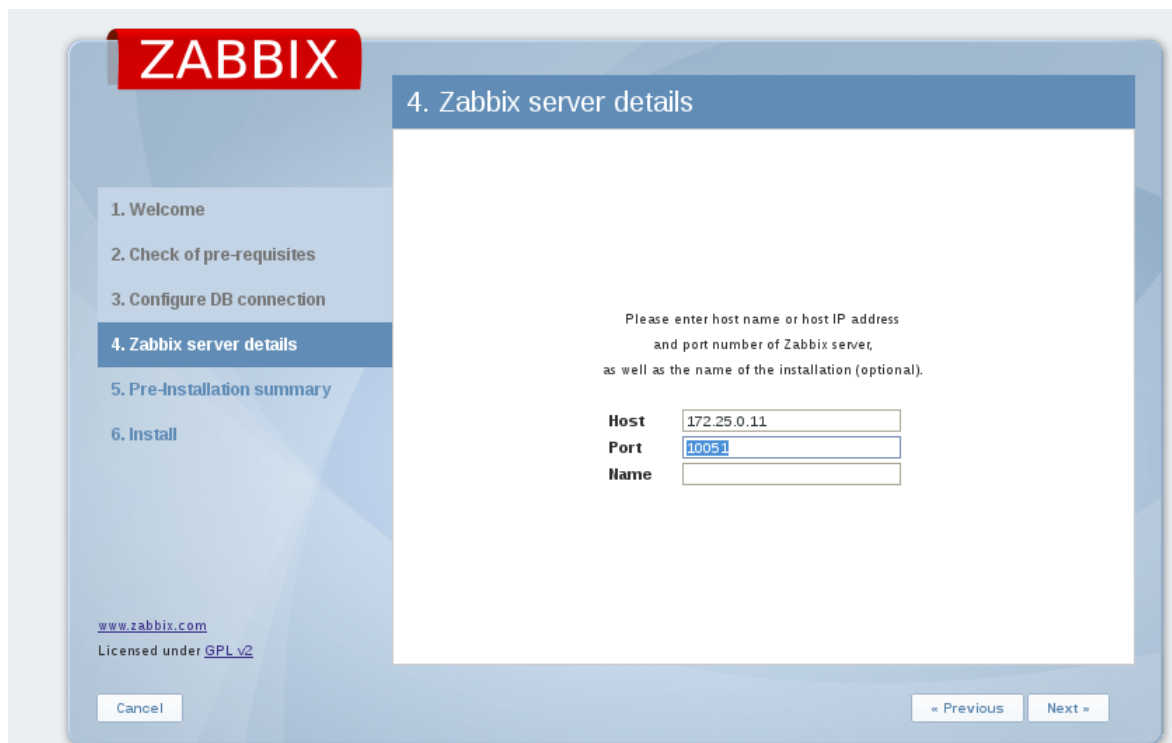
Test connection

Cancel

« Previous Next »

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

填写server端相关信息



**ZABBIX**

### 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	172.25.0.11
Port	10051
Name	

Cancel

« Previous Next »

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

安装完成，登录，默认用户名为admin，密码为zabbix。



web端的初始化完成。

## Agent端的安装

```
[root@servera zabbix]# rpm -ivh zabbix-2.4.6-1.el7.x86_64.rpm zabbix-agent-2.4.6-1.el7.x86_64.rpm
warning: zabbix-2.4.6-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 79ea5ed4: NOKEY
Preparing...                               ##### [100%]
Updating / installing...
 1:zabbix-2.4.6-1.el7                       ##### [ 50%]
 2:zabbix-agent-2.4.6-1.el7                 ##### [100%]
[root@servera zabbix]# yum -y install net-snmp net-snmp-utils
```

配置agent端相关参数

```
[root@servera zabbix]# vim /etc/zabbix/zabbix_agentd.conf
Server=172.25.0.11
ServerActive=172.25.0.11
Hostname=servera.pod0.example.com
UnsafeUserParameters=1
[root@servera zabbix]# systemctl restart zabbix-agent
```

## zabbix变更中文环境###

点击右上角profile参数，可以看到如下界面，Language这里选择chinese (zh\_CN) 即可



但是会有一些问题，并不是所有的语言都被顺利翻译成了中文，比如这里出现的问号。



主要原因是由于mysql数据库的字符集格式，我们默认使用的字符集为utf8，而mysql里的字符集为latin1。

```
MariaDB [(none)]> show create database zabbix;
+-----+-----+
| Database | Create Database |
+-----+-----+
| zabbix   | CREATE DATABASE `zabbix` /*!40100 DEFAULT CHARACTER SET latin1 */ |
+-----+-----+
1 row in set (0.00 sec)

MariaDB [(none)]>
```

这里采用以下操作来变更字符集

```
[root@serverd ~]# mysqldump zabbix > /tmp/zabbix.sql
[root@serverd ~]# sed -i 's/latin1/utf8/' /tmp/zabbix.sql
[root@serverd ~]# mysqladmin drop zabbix
[root@serverd ~]# mysql
MariaDB [(none)]> create database zabbix DEFAULT CHARACTER SET utf8;
Query OK, 1 row affected (0.00 sec)
[root@serverd ~]# mysql zabbix < /tmp/zabbix.sql
```

变更完成后，对应问号部分会显示中文，代表中文环境切换正常。



##zabbix的使用

监控配置流程

使用zabbix配置监控之前，先要了解zabbix的监控流程。一次监控配置的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）

## 配置监控主机组及主机

为了方便管理和易于查看。在监控系统中往往根据被监控的主机角色或其他属性将同类主机划分到同一个主机组中。在Zabbix中可以很方便的添加新的主机组。

添加主机组：点击组态->主机群组->创建主机群组

配置主机群组

172.25.0.12/zabbix/hostgroups.php?ddreset=1&sid=a6e1a98d65045b46

ZABBIX 帮助 | 获取支援 | 打印 | 基本资料 | 登出

监测中 资产记录 报表 组态 管理

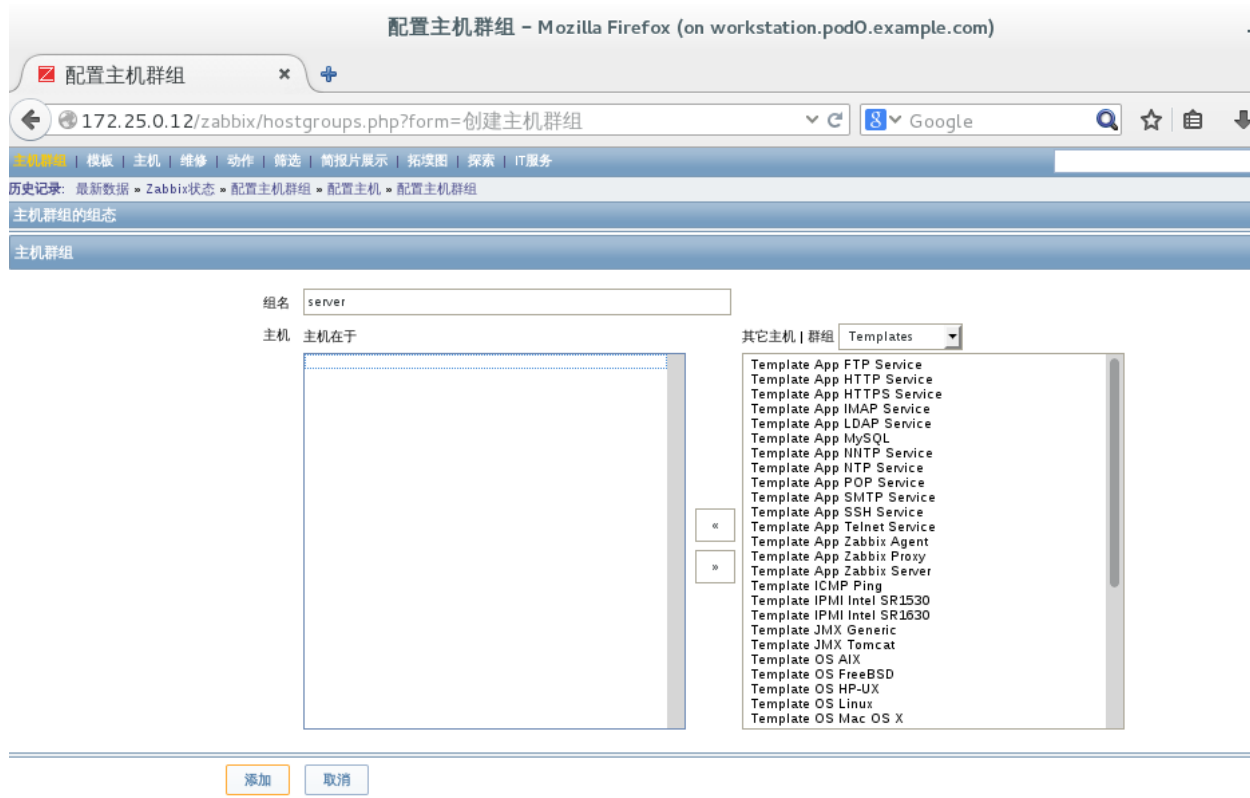
历史记录: Web监控的状态 > 最新数据 > Zabbix状态 > 配置主机群组 > 配置主机

主机群组的组态 创建主机群组

主机群组

显示 1 到 6 共 6 已发现

名称	#	成员	资讯
Discovered hosts	模板 (0) 主机 (0)	-	0 K/s 0 K/s
Hypervisors	模板 (0) 主机 (0)	-	
Linux servers	模板 (0) 主机 (0)	-	
Templates	模板 (38) 主机 (0)	Template OS Linux, Template App Zabbix Server, Template App Zabbix Proxy, Template App Zabbix Agent, Template SNMP Interfaces, Template SNMP Generic, Template SNMP Device, Template SNMP OS Windows, Template SNMP Disks, Template SNMP OS Linux, Template SNMP Processors, Template IPMI Intel SR1530, Template IPMI Intel SR1630, Template App MySQL, Template OS OpenBSD, Template OS FreeBSD, Template OS AIX, Template OS HP-UX, Template OS Solaris, Template OS Mac OS X, Template OS Windows, Template JMX Generic, Template JMX Tomcat, Template Virt VMWare, Template Virt VMWare Guest, Template Virt VMWare Hypervisor, Template App FTP Service, Template App HTTP Service, Template App HTTPS Service, Template App IMAP Service, Template App LDAP Service, Template App NNTP Service, Template App NTP Service, Template App POP Service, Template App SMTP Service, Template App SSH Service, Template App Telnet Service, Template ICMP Ping	
Virtual machines	模板 (0) 主机 (0)	-	



添加监控主机：

点击组态 -> 主机 -> 创建主机，配置一台新的监控。



主机群组 | 模板 | **配置** | 维修 | 动作 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: Zabbix状态 » 配置主机群组 » 配置主机 » 配置主机群组 » 配置主机

主机的组态

主机 | 模板 | IPMI | 巨集 | 主机资产记录

主机名称: servera.pod0.example.com

可见的名称: servera

群组: 在...群组之中

其它群组: Discovered hosts, Hypervisors, Linux servers, Templates, Virtual machines, Zabbix servers

新的群组:

端代理程式的介面

IP地址: 172.25.0.10

DNS名称:

连接到: IP地址 | DNS

埠: 10050

默认: ☒

添加

SNMP介面: 添加

## 配置监控项

应用集的配置：

点击组态--> 主机 --> 应用集-->创建应用集

**ZABBIX** 帮助 | 获取支援 | 打印 | 基本资料 | 登

监测中 | 资产记录 | 报表 | **组态** | 管理

主机群组 | 模板 | **配置** | 维修 | 动作 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: 配置主机群组 » 配置主机 » 应用集的组态 » 配置主机 » 应用集的组态

应用集的组态

« 主机清单 主机: servera 已启用 应用集 (0) 项目 (0) 触发器 (0) 图形 (0) 探索规则 (0) Web 场景 (0)

应用集

名称: CPU

添加 取消

配置监控项： 点击组态--> 主机--> 项目-->创建监控项。

名称 CPU-\$1

类型 Zabbix 代理

键值 system.cpu.util[idle] 选择

主机介面 172.25.0.10 : 10050

数据类型 数字的 (浮点)

单位

使用自订倍数 ☒ 1

数据更新间隔(秒) 30

弹性区间 间隔 期间 动作  
并无定义的弹性区间

新的弹性区间 间隔(秒计) 50 期间 1-7,00:00-24:00 添加

历史数据保留时长 (单位天) 90

趋势数据存储周期(单位天) 365

储存值 不变

展示值 不变 展示值对应

新的应用集

应用集 -无- CPU

这里实际监控的具体内容可以通过两种方式完成

1.通过预设键值的来完成，点击键值右边的选择，则出现如下图示，可自行选择常用监控内容：



2.自定义键值

可通过配置客户端的zabbix\_agent.conf里的Userparameter参数完成

```
[root@servera zabbix]# vim zabbix_agentd.conf
UserParameter=mysqlping, /usr/bin/mysqladmin ping | /usr/bin/grep -c alive
```

该监控项自定义结果可以通过server端的zabbix\_get命令检测

```
[root@serverb bin]# ./zabbix_get -s 172.25.0.10 -k mysqlping
```

## 配置报警

之前我们已经配置了一个cpu-idle的监控项，该监控项是用来监控CPU空闲率的。那么通常情况下，当性能出现瓶颈时，我们都会选择及时报警，以便能够及时解决问题。

报警模式分成三种

- Email 通过邮件报警
- Jabber 通过一个名为jabber的软件实现报警，该软件使用很高效的通信协议，然而国内没有好用的jabber的第三方插件。
- SMS 通过短信报警，需要安装一个GSM猫，此方式需要对于信号质量有一定的保证。

在我们实验环境就通过邮件来完成报警的操作。

配置报警的流程：

1.配置触发器：点击组态--> 点击主机-->点击触发器--> 创建触发器-->选择表达式



The screenshot shows the Zabbix Trigger Configuration interface. The 'Name' field is 'CPU used more than 80%'. The 'Expression' field contains the Zabbix expression: `{server:pod0.example.com:system.cpu.util[idle,load]}<20`. Below the expression field is a link '表达式建构子'. The 'Problem event occurs' checkbox is unchecked. The 'Description' field contains 'CPU used more than 80%'. The 'URL' field is empty. The 'Severity' dropdown is set to '一般严重' (Average Critical). The 'Enabled' checkbox is checked. At the bottom are '添加' (Add) and '取消' (Cancel) buttons.

表达式部分点击添加后如下：



The screenshot shows the 'Expression Builder' (表达式建构子) dialog box in Zabbix. The 'Item' field is 'server: CPU-idle'. The 'Function' dropdown is set to '最近(最近) T值是 < N'. The 'Last (T)' field is empty, and the 'Time' dropdown is set to '时间'. The 'N' field contains '20'. At the bottom are '插入' (Insert) and '取消' (Cancel) buttons.

当配置完触发器之后，若被监控主机满足触发条件，则会在仪表盘看到如下信息



主机群组	无问题	有问题	合计
server	1	1	2
已更新: 07:01:37			
最近20个议题			
主机	议题	上次修改于	历时
server	CPU used more than 80%	2016-12-15 07:01:28	9s
			显示的是 1 of 1 问题
已更新: 07:01:37			
Web 监测			

## 2.配置报警动作相关

具体操作如下

### 1) 配置报警动作：

组态-->动作-->创建动作

动作的组态

动作

条件

操作

名称

CPU used too much

默认接收人

{TRIGGER.STATUS}: {TRIGGER.NAME}

默认信息

Trigger: {TRIGGER.NAME}  
Trigger status: {TRIGGER.STATUS}  
Trigger severity: {TRIGGER.SEVERITY}  
Trigger URL: {TRIGGER.URL}  
  
Item values:  
1 {ITEM.NAME1} ({HOST.NAME1}:{ITEM.KEY1})

恢复信息

☐

已启用

☒

添加

取消

名称：该动作的名称 默认操作步骤停留时间：发现问题后多长时间执行动作 默认标题：报警信息的标题 默认消息：报警信息的内容 恢复消息：故障恢复后的消息内容。如果不开启则不会发送故障恢复通知。 这里的消息内容就是报警邮件的内容。我们可以在内容中使用Zabbix内置宏，邮件发出时会自动将宏替换成对应实际信息。

主机群组 | 模板 | 主机 | 维修 | 报警 | 筛选 | 简报片展示 | 拓坝图 | 探索 | IT服务

历史记录: 最新数据 | 配置主机 | 配置触发器 | 仪表板 | 动作的组态

动作的组态

动作

条件

操作

条件

标示

名称

动作

A 触发器 = server: CPU used more than 80% 移除

新的触发条件

触发器

=

在此输入搜索

选择

添加

添加

取消

Zabbix 2.4.6 版权Copyright 2001-2015 由Zabbix SIA拥有 | 连接

动作

条件

操作

默认操作步骤持续时间  (最少60秒)

动作操作

步骤	细节	开始于	持续时间[秒]	动作
1	送出信息给用户: Admin (Zabbix Administrator) 通过 Email	立即地	默认	<a href="#">编辑</a> <a href="#">移除</a>

操作细节

阶段

自从

到  (0 - 无限地)

步骤持续时间  (最少 60 秒, 0 为预设值)

操作类型

送出信息

送到用户群组

用户群组	动作
<a href="#">添加</a>	

送到用户

用户	动作
Admin (Zabbix Administrator)	<a href="#">移除</a>
<a href="#">添加</a>	

仅送到

Email

默认信息

☒

条件

标示	名称	动作
<a href="#">新的</a>		

[更新](#) [取消](#)

需要特别解释一下的是“步骤”部分的配置。所谓步骤是指报警可以有多个步骤，做不同的报警。例如，自从1到3，就是指报警的步骤有三个。步骤持续时间就是一定时间后如果监控人员仍未响应报警就进入下一个报警步骤。例如，发邮件给你报警，如果60秒后你没响应，那就发jabber信息提醒你。如果60秒后还没响应，那就发短信给你。要是还没响应，就没有然后了。你可以形象的把它理解为Zabbix的一哭二闹三上吊。

## 2) 配置用户送出信息相关：

点击管理-->点击用户-->选择admin用户-->点击用户示警媒介-->添加

ZABBIX

监测中 | 资产记录 | 报表 | 组态 | 管理

一般 | 系统代理程式 | 认证 | 用户 | 示警媒介类型 | 脚本 | 审计 | 队列 | 警报 | 安装

历史记录: 最新数据 | 配置主机 | 配置触发器 | 仪表盘 | 动作的组态

用户组配置 用户群组 ▼ [创建用户群组](#)

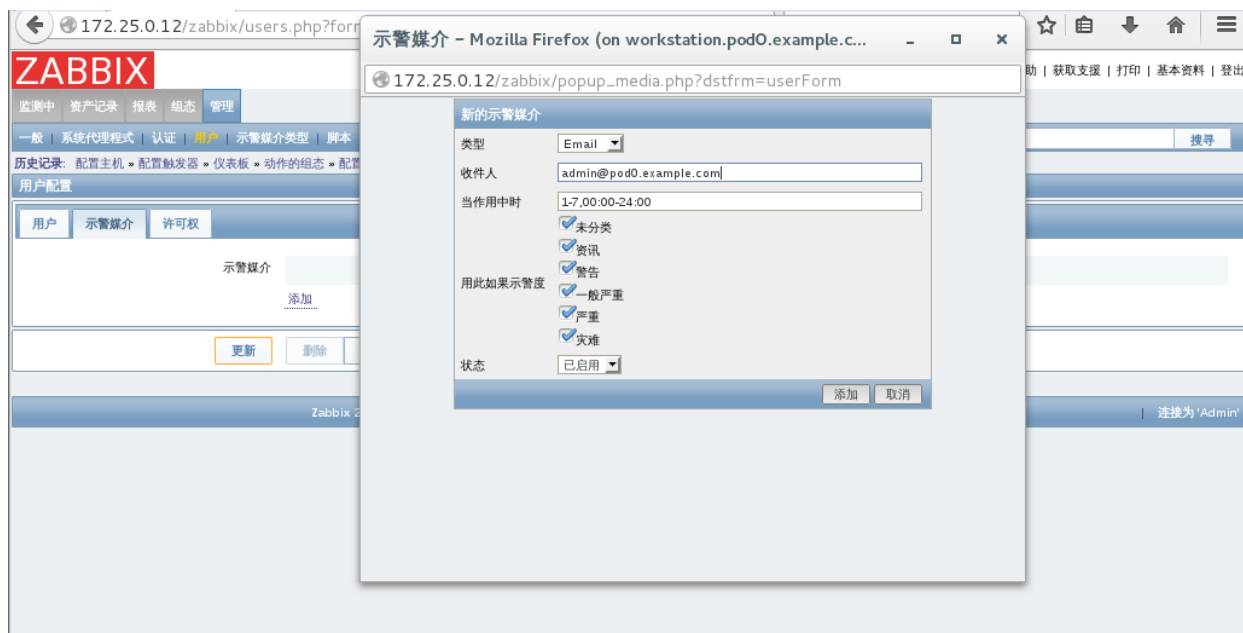
用户群组

显示 1 到 5 共 5 已发现

<input type="checkbox"/> 名称	#	成员	前端存取	调试模式	状态
<input type="checkbox"/> Disabled	用户 (0)		系统预设	停用的	停用的
<input type="checkbox"/> Enabled debug mode	用户 (0)		系统预设	已启用	已启用
<input type="checkbox"/> Guests	用户 (1)	guest	系统预设	停用的	已启用
<input type="checkbox"/> No access to the frontend	用户 (0)		停用的	停用的	已启用
<input type="checkbox"/> Zabbix administrators	用户 (1)	Admin (Zabbix Administrator)	系统预设	停用的	已启用

启用所选

确认 (0)



### 3) 配置smtp服务器指向

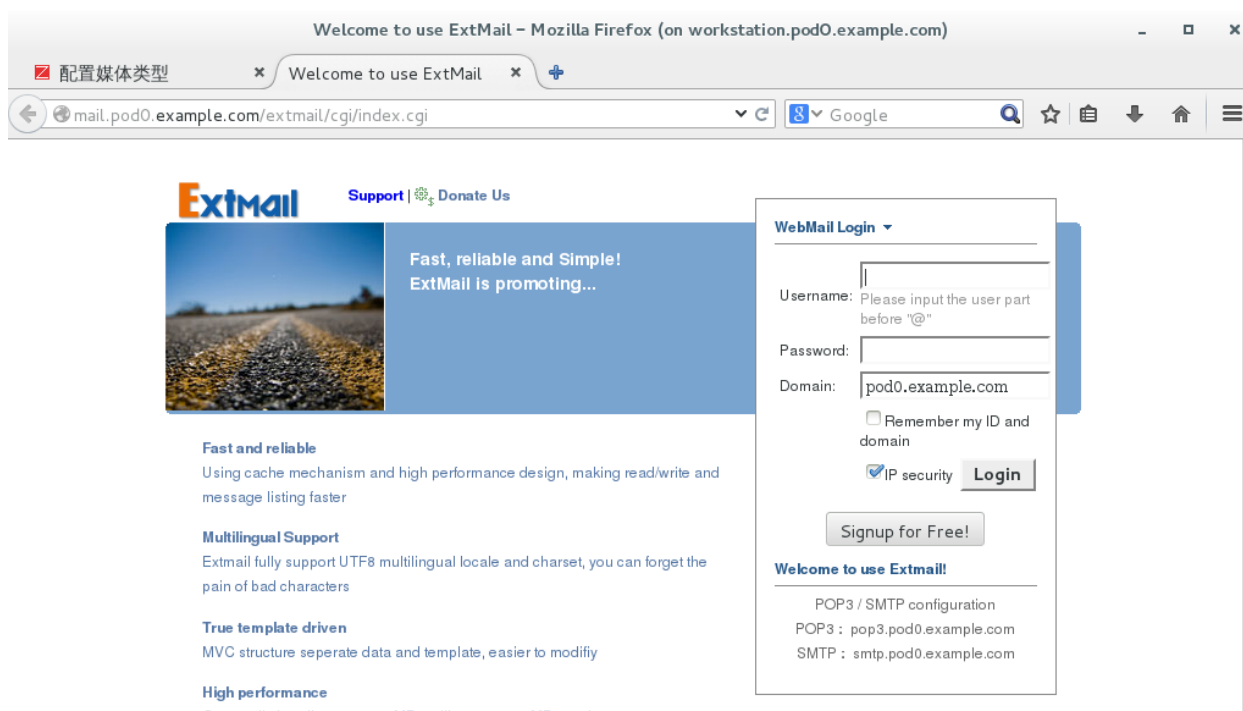
点击管理--> 点击示警媒介类型 -->选择Email-->填写smtp服务器相关参数



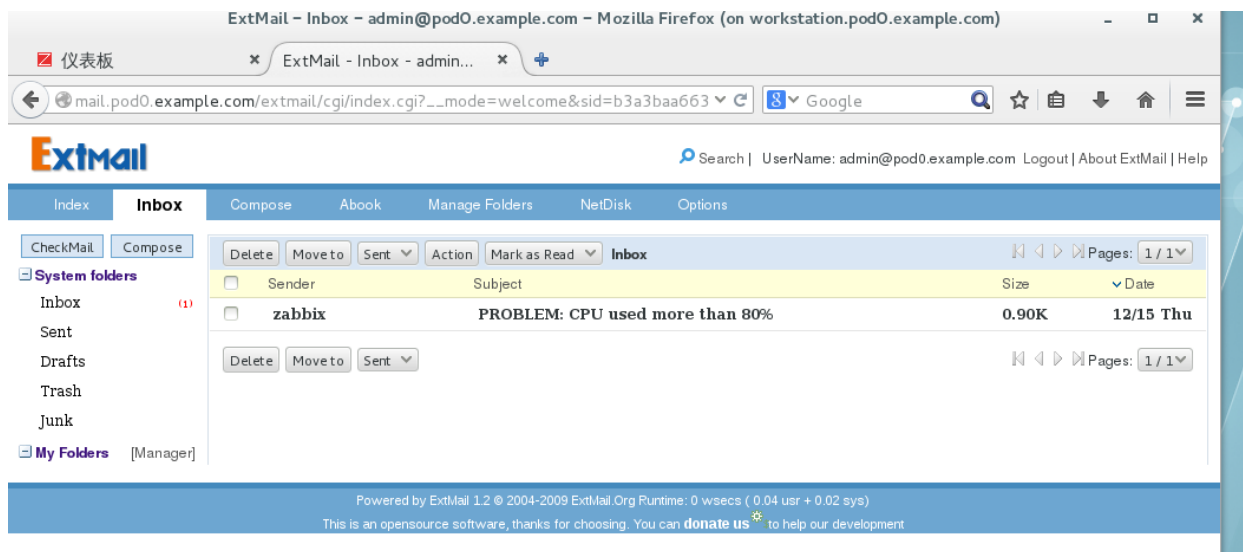
至此报警配置完成，触发一些操作使CPU空闲率低于20%。

登录邮箱 (mail.podx.example.com)

用户名admin 密码uplooking



登录后则会看到相应报警提示：



## 配置图形

点击组态-->主机-->点击图形--> 创建图形

配置图标

ExtMail - Inbox - admin...

172.25.0.12/zabbix/graphs.php?hostid=10105&form=创建图形

主机群组 | 模板 | 监控 | 维护 | 动作 | 筛选 | 简报片展示 | 拓展图 | 探索 | IT服务

历史记录: 仪表盘 > 最新数据 > 仪表盘 > 配置主机 > 配置图标

图形的组态

主机清单 主机: servera 已启用 应用集 (1) 项目 (1) 触发器 (1) 图形 (0) 探索规则 (0) Web 场景 (0)

图形 预览

名称

宽 900

高 200

图形类别 正常

展示图标 ☒

展示工作时间 ☒

展示触发器 ☒

百分比线(左) ☐

百分比线(右) ☐

纵轴Y最小值MIN 计算的

纵轴最大值 计算的

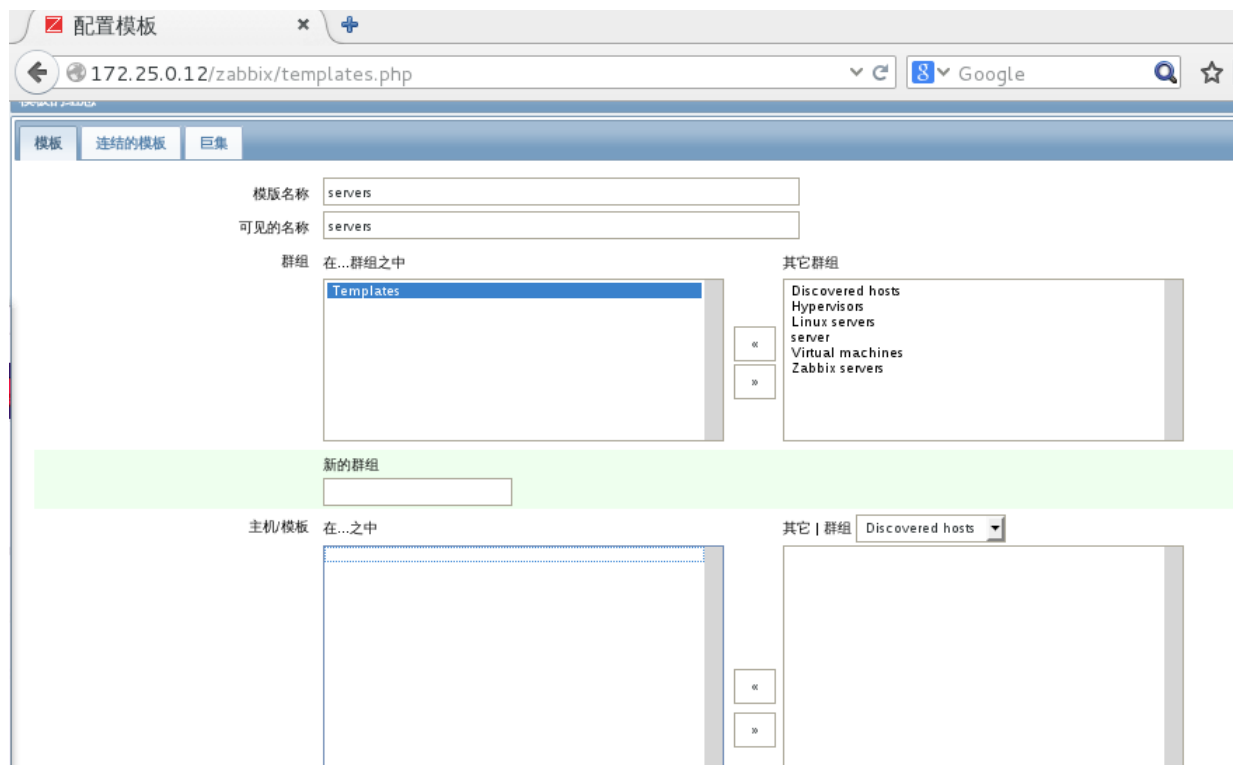
项目	名称	功能	绘图风格	纵轴Y侧	颜色	动作
添加						

项目 - Mozilla Firefox (on workstation.podO.example.com)

172.25.0.12/zabbix/popup.php?writeonly=1&multiselect=1&dstfrm=graphForm&srctbl=items&srcfld1=itemid&srcfld2=...

名称	键值	类型	数据类型	状态
CPU	system.cpu.util[,idle]	Zabbix 代理	数字的 (浮点)	已启用

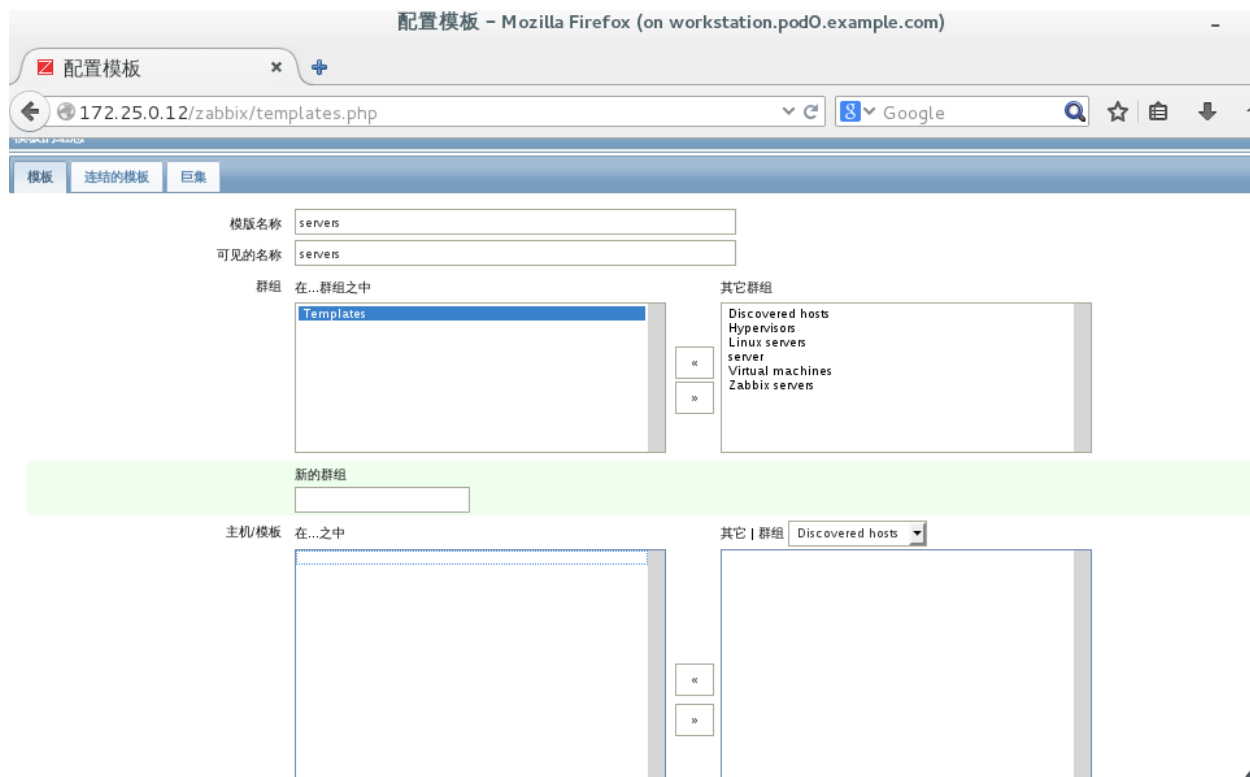
选择



## 配置模板

一般情况下，我们都会以模板的方式来存储多个已配置完成的监控项，以便和添加的主机进行关联操作。

点击组态--> 点击模板--> 创建模板



添加完模板后，可以在模板里面进行应用集、监控项、触发器等配置



主机群组 | 模板 | 主机 | 维修 | 动作 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: 配置图标 » 配置模板 » 仪表板 » 配置主机群组 » 配置模板

模板已添加

模板的组态

创建模板

模板

群组 | 所有

显示 1 到 39 其 39 已发现

<input type="checkbox"/> 模板	应用集	项目	触发器	图形	筛选	探索	Web	连结的模板	已连结到
<input type="checkbox"/> <a href="#">servers</a>	应用集 (0)	项目 (0)	触发器 (0)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App FTP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App HTTP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App HTTPS Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App IMAP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App LDAP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App MySQL</a>	应用集 (1)	项目 (14)	触发器 (1)	图形 (2)	筛选 (1)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App NNTP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App NTP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App POP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App SMTP Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App SSH Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-
<input type="checkbox"/> <a href="#">Template App Telnet Service</a>	应用集 (1)	项目 (1)	触发器 (1)	图形 (0)	筛选 (0)	探索 (0)	Web (0)	-	-

配置完监控项等内容之后，只需一步操作，即可使多台监控主机应用模板当中的配置，使我们的模板和主机关联即可。

模板清单 模板: [servers](#) 应用集 (0) 项目 (0) 触发器 (0) 图形 (0) 筛选 (0) 探索规则 (0) Web 场景 (0)

模板 连结的模板 巨集

模版名称

可见的名称

群组 在...群组之中

Templates

其它群组

Discovered hosts  
Hypervisors  
Linux servers  
server  
Virtual machines  
Zabbix servers

新的群组

主机/模板 在...之中

servera  
serveral

其它 | 群组 

server

## 配置zabbix的自动探索

创建探索规则：

点击组态--> 点击探索 --> 点击创建发现规则

21 of 26

2017年08月07日 11:39

ZABBIX

帮助 | 获取支援 | 打印 | 基本资料 | 登出

监测中 | 资产记录 | 报表 | 组态 | 管理

主机群组 | 模板 | 主机 | 维修 | 动作 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: 配置维护时段 > 动作的组态 > 配置主机群组 > 动作的组态 > 配置发现规则

探索规则的组态

探索规则

名称

http-server

由系统代理程式探索

没有系统代理程式

IP范围

172.25.0.10-19

延迟(秒计)

30

检查

HTTP

编辑

移除

新的

设备唯一性准则

☒ IP地址

已启用

☒

添加

取消

配置发现后动作，和http模板相关联：

ZABBIX

帮助 | 获取支援 | 打印 | 基本资料 | 登出

监测中 | 资产记录 | 报表 | 组态 | 管理

主机群组 | 模板 | 主机 | 维修 | 探索 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: 配置维护时段 > 动作的组态 > 配置主机群组 > 动作的组态 > 配置发现规则

动作的组态

动作

事件源 探索

显示 1 到 1 共 1 已发现

名称	条件	操作	状态
<input type="checkbox"/> <a href="#">Auto discovery Linux servers</a>	接收到的值 似 Linux 探索状态 = 上 服务类型 = Zabbix 代理	添加到主机群组: Linux servers 连接到模板: Template OS Linux	停用的

启用所选

确认 (0)

Zabbix 2.4.6 版权Copyright 2001-2015 由Zabbix SIA拥有

连接为 'Admin'

ZABBIX

帮助 | 获取支援 | 打印 | 基本资料 | 登出

监测中 | 资产记录 | 报表 | 组态 | 管理

主机群组 | 模板 | 主机 | 维修 | 探索 | 筛选 | 简报片展示 | 拓模图 | 探索 | IT服务

历史记录: 动作的组态 > 配置主机群组 > 动作的组态 > 配置发现规则 > 动作的组态

动作的组态

动作

条件

操作

名称

http-server

默认接收人

Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.I

默认信息

Discovery rule: {DISCOVERY.RULE.NAME}  
Device IP: {DISCOVERY.DEVICE.IPADDRESS}  
Device DNS: {DISCOVERY.DEVICE.DNS}  
Device status: {DISCOVERY.DEVICE.STATUS}  
Device uptime: {DISCOVERY.DEVICE.uptime}  
Device service name: {DISCOVERY.SERVICE.NAME}

已启用

☒

添加

取消

Zabbix 2.4.6 版权Copyright 2001-2015 由Zabbix SIA拥有

连接为 'Admin'



点击检测中--> 探索，就可以看到多台主机已经被发现了



并且：该主机已经和模板关联。



## zabbix的proxy配置

zabbix可以作为一个分布式监控服务器，而在这环节当中，最重要的就是proxy这个组件。

在这里，我们以serverj作为zabbix的代理服务器。

### 软件安装

```
[root@serverj zabbix]# yum -y localinstall zabbix-proxy-2.4.6-1.el7.x86_64.rpm zabbix-proxy-mysql-2.4.6-1.el7.x86_64.rpm
[root@serverj zabbix]# yum -y install mariadb-server
```

### 配置proxy的相应参数

```
[root@serverj zabbix]# vim /etc/zabbix/zabbix_proxy.conf
Server=172.25.0.11
Hostname=serverj.pod0.example.com
DBHost=localhost
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=uplooking
```

### 配置数据库相关

```
[root@serverj zabbix]# systemctl restart mariadb
[root@serverj zabbix]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.41-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database zabbix_proxy;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on zabbix_proxy.* to zabbix@'localhost' identified by 'uplooking';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> Bye
[root@serverj zabbix]# rpm -ql zabbix-proxy-mysql
/usr/sbin/zabbix_proxy_mysql
/usr/share/doc/zabbix-proxy-mysql-2.4.6
/usr/share/doc/zabbix-proxy-mysql-2.4.6/create
/usr/share/doc/zabbix-proxy-mysql-2.4.6/create/schema.sql
[root@serverj zabbix]# mysql zabbix_proxy < /usr/share/doc/zabbix-proxy-mysql-2.4.6/create/schema.sql
```

## 启动zabbix-proxy服务

```
[root@serverj zabbix]# systemctl restart zabbix-proxy
```

## 配置zabbix-agent端

要求允许zabbix-proxy服务器采集监控数据

```
[root@servera nginx]# vim /etc/zabbix/zabbix_agentd.conf
Server=172.25.0.11, 172.25.0.19
ServerActive=172.25.0.11, 172.25.0.19
[root@servera nginx]# systemctl restart zabbix-agent
```

## 设置Web端

打开浏览器访问zabbix-web端，在web端将该代理节点添加进去，并选择哪些主机由代理监控。

The screenshot shows the Zabbix web interface. The top navigation bar includes links for Monitoring, Asset Inventory, Reports, Configuration, and Management. The main content area is titled 'System Proxy Configuration' and contains the following fields and sections:

- System Proxy Name:** serverj.pod0.example.com
- System Proxy Mode:** Active (下拉菜单)
- Hosts:** A list of hosts associated with the proxy, including servera, serverc.pod0.example.com, serverg.pod0.example.com, and serverh.pod0.example.com.
- Other Hosts:** A list of other hosts, including servera1, serverc.pod0.example.com\_2, serverg.pod0.example.com\_2, serverh.pod0.example.com\_2, and Zabbix server.
- Description:** A text area for additional information.

