1. zabbix简介：

zabbix是完全开源的工具，整合了cacti和nagios等特性

cacti:snmp

nagios:linux主机:nrpe 设备：snmp

附：SNMP(udp 161 udp 162)

众多网络工具都支持此协议，比如常见路由交换，常见OS

其既可以做管理端也可以做被管理端

snmp协议大致有3个版本分别是v1 v2 v3

无论是v1 和 v2 的安全性是比较差的，因为传输是明文的， V3的认证密码用MD5/SHA摘要算法加密

很多工具支持网络管理的功能，而对于非网络设备（操作系统），可以完全抛开snmp这种不安全的架构来实现监控的。所以很多工具都是控制端和agent架构，他们有专属的agent

**Zabbix的主要功能：​**

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、FTP 等通用协议监控、多种告警方式、详细的报表图表绘制）

支持自动发现网络设备和服务器（可以通过配置自动发现服务器规则来实现）

支持分布式，能集中展示、管理分布式的监控点，扩展性强

server 提供通用接口（api 功能），可以自己开发完善各类监控（根据相关接口编写程序实现），编写插件容易，可以自定义监控项，报警级别的设置。

数据收集，支持 snmp(包括 trapping and polling )，IPMI，JMX，SSH，TELNET；

自定义的检测；自定义收集数据的频率；

服务器/代理和客户端模式；

灵活的触发器；可以定义非常灵活的问题阈值，称为触发器；

高可定制的报警；发送通知，可定制的报警升级，收件人，媒体类型。

CPU负荷、内存使用、磁盘使用、网络状况、端口监视、日志监视等等。

硬件监控：Zabbix IPMI Interface

系统监控：Zabbix Agent Interface

Java监控：Zabbix JMX Interface

网络设备监控：Zabbix SNMP Interface

应用服务监控：Zabbix Agent UserParameter

MySQL数据库监控：percona-monitoring-plulgins

URL监控：Zabbix Web 监控

zabbix重要组件说明：

1）zabbix server:负责接收agent发送的报告信息的核心组件，所有配置、统计数据及操作数据都由它组织进行；

2）database storage：专用于存储所有配置信息，以及由zabbix收集的数据；

3）web interface：zabbix的GUI接口；

4）proxy：可选组件，常用于监控节点很多的分布式环境中，代理server收集部分数据转发到server，可以减轻server的压力；

5）agent：部署在被监控的主机上，负责收集主机本地数据如cpu、内存、数据库等数据发往server端或proxy端；

另外，zabbix server、proxy、agent都有自己的配置文件以及log文件，重要的参数需要在这里配置，后面会详细说明。

一个监控系统运行的大概的流程是这样的：

agentd需要安装到被监控的主机上，它负责定期收集各项数据，并发送到zabbix server端，zabbix server将数据存储到数据库中，zabbix web根据数据在前端进行展现和绘图。这里agentd收集数据分为主动和被动两种模式：

主动：agent请求server获取主动的监控项列表，并主动将监控项内需要检测的数据提交给server/proxy

被动：server向agent请求获取监控项的数据，agent返回数据。

zabbix常用的监控架构平台

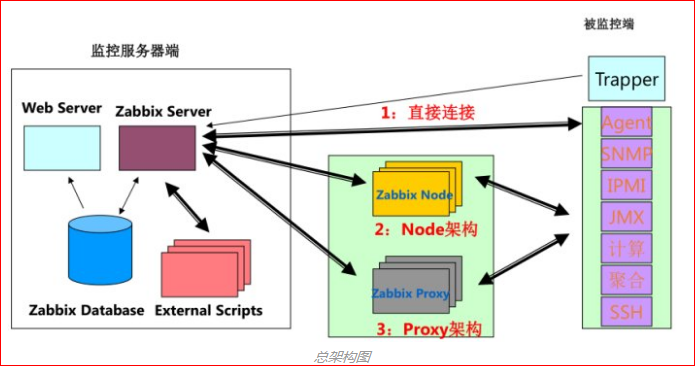
1、server-agentd模式：

这个是最简单的架构了，常用于监控主机比较少的情况下。

2、server-proxy-agentd模式：

这个常用于比较多的机器，使用proxy进行分布式监控，有效的减轻server端的压力。

zabbix的系统架构：



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开源解决方案。借助Zabbix,可以很轻松地减轻运维人员们繁重的服务器管理任务，实现业务系统的持续运行。下面会逐步介绍Zabbix分布式监控系统的部署及使用

配置好IP、DNS 、网关，确保使用远程连接工具能够连接服务器

zabbix监控服务器：192.168.130.76 #zabbix的服务端（若要监控本机，则需要配置本机的zabbix agent)

Zabbix agent被监控主机：192.168.130.78#zabbix的客户端（被监控端，需要配置Zabbix agent

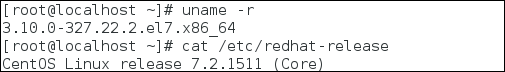
**二、Zabbix部署前的LNMP环境的搭建过程：**

安装编译工具及库文件

[root@mysqla ~]# yum -y install make apr\* autoconf automake curl-devel gcc gcc-c++ openssl openssl-devel gd kernel keyutils patch perl kernel-headers compat\* mpfr cpp glibc libgomp libstdc++-devel keyutils-libs-devel libcom\_err-devel libsepol-devel libselinux-devel krb5-devel zlib-devel libXpm\* freetype libjpeg\* libpng\* libtool\* libxml2 libxml2-devel patch libcurl-devel bzip2-devel freetype-devel

安装mysql5.7.13

1、系统环境：centos7.2 x86\_64



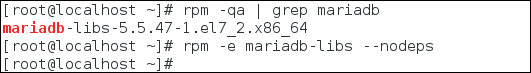
因为centos7.2默认安装了mariadb-libs，所以先要卸载掉

查看是否安装mariadb

#rpm -qa | grep mariadb

卸载mariadb

rpm -e --nodeps mariadb-libs



2、安装依赖包

**注： 相关依赖包的作用**

cmake：由于从MySQL5.5版本开始弃用了常规的configure编译方法，所以需要CMake编译器，用于设置mysql的编译参数。如：安装目录、数据存放目录、字符编码、排序规则等。

Boost #从MySQL 5.7.5开始Boost库是必需的，mysql源码中用到了C++的Boost库，要求必须安装boost1.59.0或以上版本

GCC是Linux下的C语言编译工具，mysql源码编译完全由C和C++编写，要求必须安装GCC

bison：Linux下C/C++语法分析器

ncurses：字符终端处理库

1）安装文件准备

下载cmake-3.5.tar.gz    [http://mysqla.cmake.org/download/](http://www.cmake.org/download/)

下载ncurses-5.9.tar.gzftp://ftp.gnu.org/gnu/ncurses/

下载bison-3.0.4.tar.gzhttp://ftp.gnu.org/gnu/bison/

下载mysql-5.7.13.tar.gz

wget http://cdn.mysql.com/Downloads/MySQL-5.7/mysql-5.7.13.tar.gz

下载Boost\_1\_59\_0.tar.gz

wget http://nchc.dl.sourceforge.net/project/boost/boost/1.59.0/boost\_1\_59\_0.tar.gz

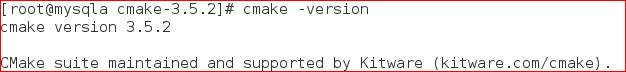
2）安装CMAKE及必要的软件

安装cmake





cmake –version  ---查看cmake版本



**安装ncurses**



安装bison



安装bootst

tar zxf boost\_1\_59\_0.tar.gz

mv boost\_1\_59\_0 /usr/local/boost

3）创建mysql用户和用户组及目录

# groupadd -r mysql && useradd -r -g mysql -s /bin/false -M mysql---新建msyql组和msyql用户禁止登录shell

#mkdir /usr/local/mysql        ---创建目录

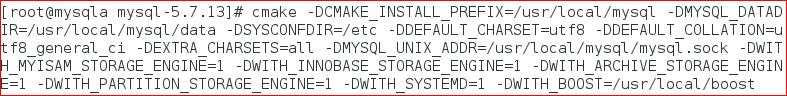
#mkdir /usr/local/mysql/data    ---数据库目录

3、编译安装mysql

解压mysql源码包：



执行cmake命令进行编译前的配置：



开始编译、编译安装：



注1：配置解释：

-DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql[MySQL安装的根目录]-DMYSQL\_DATADIR=/usr/local/mysql /data[MySQL数据库文件存放目录]

-DSYSCONFDIR=/etc [MySQL配置文件所在目录]

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 [添加MYISAM引擎支持]

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1[添加InnoDB引擎支持]

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1 [添加ARCHIVE引擎支持]

-DMYSQL\_UNIX\_ADDR=/usr/local/mysql /mysql.sock[指定mysql.sock位置]

-DWITH\_PARTITION\_STORAGE\_ENGINE=1[安装支持数据库分区]

-DEXTRA\_CHARSETS=all [使MySQL支持所有的扩展字符]

-DDEFAULT\_CHARSET=utf8[设置MySQL的默认字符集为utf8]-DDEFAULT\_COLLATION=utf8\_general\_ci [设置默认字符集校对规则]

-DWITH-SYSTEMD=1 [可以使用systemd控制mysql服务]

-DWITH\_BOOST=/usr/local/boost [指向boost库所在目录]

更多参数执行[root@localhost mysql-5.7.13]**# cmake . –LH**

**注2：为了加快编译速度可以按下面的方式编译安装**



make -j $(grep processor /proc/cpuinfo | wc –l)

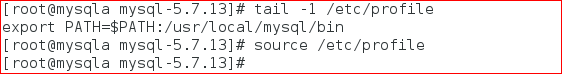
-j参数表示根据CPU核数指定编译时的线程数，可以加快编译速度。默认为1个线程编译。

注3：若要重新运行cmake配置，需要删除CMakeCache.txt文件

# make clean

#rm -f CMakeCache.txt

优化Mysql的执行路径



4、设置权限并初始化MySQL系统授权表

# cd/usr/local/mysql

# **chown -R mysql:mysql .**       ---更改所有者,属组，注意是**mysql .**

**#bin/mysqld --initialize--user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data**

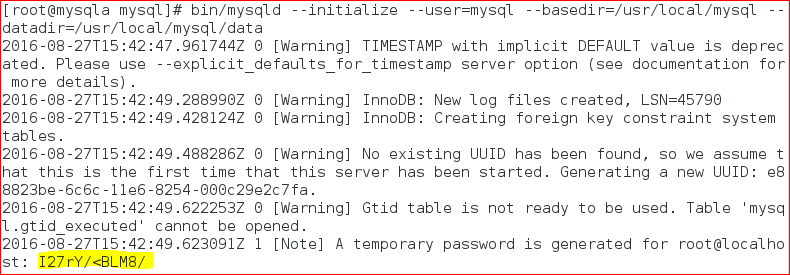
**注1：**以root初始化操作时要加--user=mysql参数，生成一个随机密码（注意保存登录时用）

注2：MySQL 5.7.6之前的版本执行这个脚本初始化系统数据库

/usr/local/mysql/bin/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data

# 5.7.6之后版本初始系统数据库脚本（本文使用此方式初始化）

#/usr/local/mysql/bin/mysqld --initialize-insecure--user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data



注意：如果使用–initialize参数初始化系统[数据库](http://lib.csdn.net/base/14)之后，会生成root用户的一个临时密码，如上图高亮中所示。

# **chown -Rmysql:mysql .**       ---改所有者，注意是**root .**

5、创建配置文件

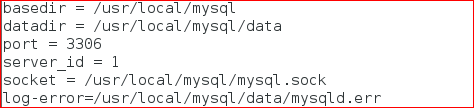
# cd/usr/local/mysql/support-files     ---进入MySQL安装目录支持文件目录

# cp my-default.cnf /etc/my.cnf    ---复制模板为新的配置文件，



修改文件中配置选项，如下图所示，添加如下配置项

#vi /etc/my.cnf

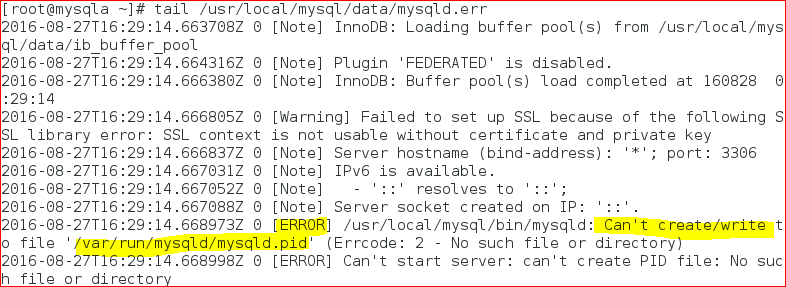


6、配置mysql自动启动





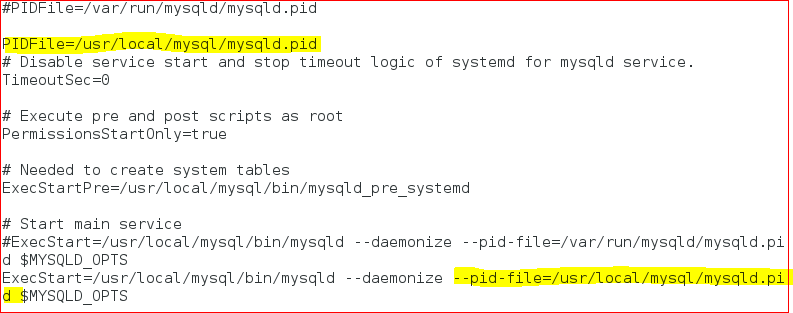
服务启动失败，查看错误日志文件



在mysqld.service，把默认的pid文件指定到了/var/run/mysqld/目录，而并没有事先建立该目录，因此要手动建立该目录并把权限赋给mysql用户。

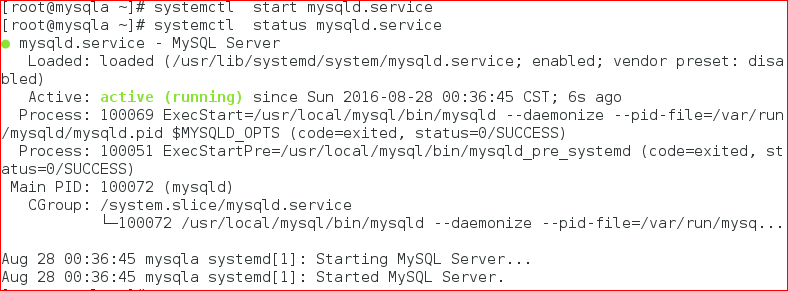


或者修改/usr/lib/system/system/mysqld.service，修改内容如下：



#systemctl daemon-reload

再次启动mysql服务



查看端口号



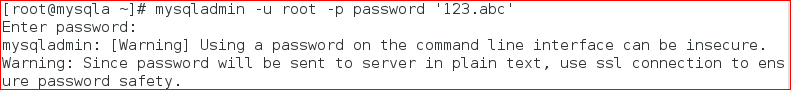
服务启动成功

访问MySQL数据库

# mysql -u root -h 127.0.0.1 -p     ---连接mysql，输入初始化时生成的随机密码



设置数据库管理员用户root的密码



安装nginx：

解压zlib

[root@mysqla ~]# tar zxf zlib-1.2.8.tar.gz

说明：不需要编译，只需要解压就行。

解压pcre

[root@mysqla ~]# tar zxf pcre-8.39.tar.gz

说明：不需要编译，只需要解压就行。

[root@mysqla ~]# groupaddwww

[root@mysqla ~]# useradd -g www www -s /sbin/nologin

下载nginx的源码包：<http://nginx.org/download>

解压源码包：

[root@mysqla ~]# tar zxf nginx-1.10.2.tar.gz

[root@mysqla ~]# cd nginx-1.10.2/

[root@mysqla nginx-1.10.2]# ./configure --prefix=/usr/local/nginx1.10 --with-http\_dav\_module --with-http\_stub\_status\_module --with-http\_addition\_module --with-http\_sub\_module --with-http\_flv\_module --with-http\_mp4\_module --with-pcre=/root/pcre-8.39 --with-zlib=/root/zlib-1.2.8 --with-http\_ssl\_module --with-http\_gzip\_static\_module --user=www --group=www

[root@mysqla nginx-1.10.2]# make&& make install

[root@mysqla ~]# ln -s /usr/local/nginx1.10/sbin/nginx /usr/local/sbin/

[root@mysqla ~]# nginx

[root@mysqla ~]# netstat -anpt | grep nginx

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 11598/nginx: master

[root@mysqla ~]# firewall-cmd --permanent --add-port=80/tcp

success

[root@mysqla ~]# firewall-cmd --reload

success

启动后可以再浏览器中打开页面，会显示nginx默认页面。



安装php

[root@mysqla ~]# tar zxf libmcrypt-2.5.7.tar.gz

[root@mysqla ~]# cd libmcrypt-2.5.7/

[root@mysqla libmcrypt-2.5.7]# ./configure && make && make install

# ln -s /usr/local/mysql/lib/libmysqlclient.so.20.3.0 /usr/local/mysql/lib/libmysqlclient\_r.so

[root@mysqla ~]# tar zxf php-5.6.27.tar.gz

[root@mysqla php-5.6.27]# ./configure --prefix=/usr/local/php5.6 --with-config-file-path=/etc --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-mysql-sock=/usr/local/mysql/mysql.sock --with-gd --with-iconv --with-libxml-dir=/usr --with-mhash --with-mcrypt --with-config-file-scan-dir=/etc/php.d --with-bz2 --with-zlib --with-freetype-dir --with-png-dir --with-jpeg-dir --enable-xml --enable-bcmath --enable-shmop --enable-sysvsem --enable-inline-optimization --enable-mbregex --enable-fpm --enable-mbstring --enable-ftp --enable-gd-native-ttf --with-openssl --enable-pcntl --enable-sockets --with-xmlrpc --enable-zip --enable-soap --without-pear --with-gettext --enable-session --with-mcrypt --with-curl

[root@mysqla php-5.6.27]# make&& make install

[root@mysqla php-5.6.27]# cp php.ini-production /etc/php.ini

编辑配置文件/etc/php.ini ，修改后的内容如下：

找到：  
;date.timezone =   
修改为：  
date.timezone = PRC #设置时区

找到：  
expose\_php = On   
修改为：  
expose\_php = Off #禁止显示php版本的信息

找到：  
short\_open\_tag = Off   
修改为：

short\_open\_tag = On //支持php短标签

找到：

post\_max\_size = 8M

修改为：

post\_max\_size = 16M //上传文件大小

找到：

max\_execution\_time = 30

修改为：

max\_execution\_time = 300 //php脚本最大执行时间

找到：

max\_input\_time = 60

修改为：

max\_input\_time = 300 //以秒为单位对通过POST、GET以及PUT方式接收数据时间进行限制

always\_populate\_raw\_post\_data = -1

mbstring.func\_overload = 0

创建php-fpm服务启动脚本：

[root@mysqla php-5.6.27]# cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

[root@mysqla php-5.6.27]# chmod +x /etc/init.d/php-fpm

[root@mysqla php-5.6.27]# chkconfig --add php-fpm

[root@mysqla php-5.6.27]# chkconfig php-fpm on

提供php-fpm配置文件并编辑：

#cp /usr/local/php5.6/etc/php-fpm.conf.default /usr/local/php5.6/etc/php-fpm.conf

[root@mysqla php-5.6.27]# vi /usr/local/php5.6/etc/php-fpm.conf

修改内容如下：

pid = run/php-fpm.pid

user = www

group = www

listen =127.0.0.1:9000

pm.max\_children = 300

pm.start\_servers = 10

pm.min\_spare\_servers = 10

pm.max\_spare\_servers =50

启动php-fpm服务：

[root@phpserver ~]# service php-fpm start

Starting php-fpm done

[root@mysqla php-5.6.27]# netstat -anpt | grep php-fpm

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN 10937/php-fpm: mast

配置nginx支持php

[root@mysqla ~]# cat /usr/local/nginx1.10/conf/nginx.conf

user www www;

worker\_processes 4;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

use epoll;

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#gzip on;

server {

listen 80;

server\_name localhost;

charset utf-8;

#access\_log logs/host.access.log main;

location / {

root html;

index index.php index.html index.htm;

}

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

location /status {

stub\_status on;

}

}

}

[root@mysqla conf]# nginx -t

nginx: the configuration file /usr/local/nginx1.10/conf/nginx.conf syntax is ok

nginx: configuration file /usr/local/nginx1.10/conf/nginx.conf test is successful

[root@mysqla conf]# killall -s HUP nginx

测试LNMP

进入nginx默认的网页根目录，创建.php的测试页

[root@mysqla ~]# cat /usr/local/nginx1.10/html/test1.php

<?php

phpinfo()

?>

[root@mysqla ~]# cat /usr/local/nginx1.10/html/test2.php

<?php

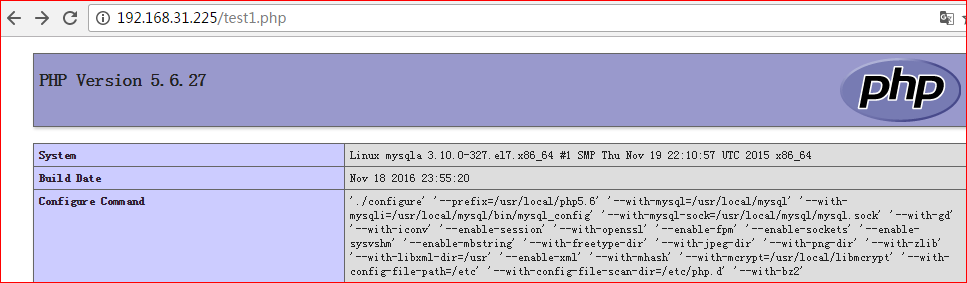
$link=mysql\_connect('localhost','root','123.abc');

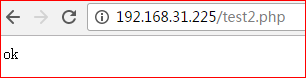
if($link) echo "ok";

mysql\_close();

?>

访问结果：





至此，LNMP部署完毕。

三、监控系统Zabbix-3.2.1的安装

**zabbix-server端的操作**

zabbix服务器端要提前安装好LNMP环境（mysql，nginx，php5的安装目录均是/usr/local）

创建Zabbix运行的用户:

[root@mysqla ~]#groupadd zabbix

[root@mysqla ~]#useradd -g zabbix zabbix

安装libcurl和net-snmp:

[root@mysqla ~]#yum -y install net-snmp net-snmp-develcurl-develjava-1.8.0-openjdk java-1.8.0-openjdk-develOpenIPMI-devel libssh2-devel

注：OpenIPMI-devel 和libssh2-devel软件包使用centos在线yum软件源安装

安装Fping:

[root@mysqla ~]# tar zxf fping-3.12.tar.gz

[root@mysqla ~]# cd fping-3.12/

[root@mysqla fping-3.12]# ./configure && make && make install

[root@mysqla fping-3.12]# chown root:zabbix /usr/local/sbin/fping

[root@mysqla fping-3.12]# chmod 4710 /usr/local/sbin/fping

安装Zabbix Server:

[root@mysqla ~]# tar zxf zabbix-3.2.1.tar.gz

[root@mysqla ~]# cd zabbix-3.2.1/

[root@mysqla zabbix-3.2.1]# ./configure --prefix=/usr/local/zabbix --enable-server --enable-agent --enable-java --with-mysql=/usr/local/mysql/bin/mysql\_config --with-net-snmp --with-libcurl --with-openipmi

注意：编译时最好带上--enable-java这个参数，方便后续监控tomcat程序所用。

--with-ssh2 是不需要在客户端服务器上面安装Zabbix agent，如果需要使用ssh检查，需要在编译的时候加上这项，最低需要libssh2 1.0.0版本，需要安装ssh开发包

--with-openipmi用户可以利用 IPMI 监视服务器的物理特征，如温度、电压、电扇工作状态、电源供应等。

如果添加了--enable-proxy, 那么会生成get和sender两条命令. 如下, 用于接收agent发生过来的信息, 同时发送给server.

[root@mysqla zabbix-3.2.1]#make && make install

添加系统软连接

[root@mysqla ~]# ln -s /usr/local/zabbix/bin/\* /usr/local/bin/

[root@mysqla ~]# ln -s /usr/local/zabbix/sbin/\* /usr/local/sbin/

Zabbix Server配置与启动  
创建Zabbix数据库和MySQL用户:

mysql> create database zabbix character set utf8;

Query OK, 1 row affected (0.01 sec)

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

Query OK, 0 rows affected, 1 warning (0.00 sec)

导入Zabbix初始数据:

切换到zabbix的解压目录下

[root@mysqla zabbix-3.2.1]# cd database/mysql/

[root@mysqla mysql]# pwd

/root/zabbix-3.2.1/database/mysql

[root@mysqla mysql]# ls

data.sql images.sql schema.sql

进行zabbix初始数据导入

[root@mysqla mysql]# mysql -uzabbix -pzabbix -hlocalhost zabbix < schema.sql

[root@mysqla mysql]# mysql -uzabbix -pzabbix -hlocalhost zabbix < images.sql

[root@mysqla mysql]# mysql -uzabbix -pzabbix -hlocalhost zabbix < data.sql

编辑/usr/local/zabbix/etc/zabbix\_server.conf:

root@mysqla ~]# grep -v "^#" /usr/local/zabbix/etc/zabbix\_server.conf | grep -v "^$"

LogFile=/usr/local/zabbix/logs/zabbix\_server.log

PidFile=/usr/local/zabbix/logs/zabbix\_server.pid

DBHost=localhost

DBName=zabbix

DBUser=zabbix

DBPassword=zabbix

DBPort=3306

FpingLocation=/usr/local/sbin/fping

#mkdir -p /usr/local/zabbix/logs  
#chown -R zabbix:zabbix /usr/local/zabbix

启动Zabbix Server:

# /usr/local/zabbix/sbin/zabbix\_server -c /usr/local/zabbix/etc/zabbix\_server.conf

/usr/local/zabbix/sbin/zabbix\_server: error while loading shared libraries: libmysqlclient.so.20: cannot open shared object file: No such file or directory

解决方法：

在ld.so.conf中加入/usr/local/mysql/lib

[root@mysqla zabbix-3.2.1]# cat /etc/ld.so.conf

include ld.so.conf.d/\*.conf

/usr/local/mysql/lib

/usr/local/lib

[root@mysqla zabbix-3.2.1]# ldconfig

再次执行zabbix\_server启动

# /usr/local/zabbix/sbin/zabbix\_server -c /usr/local/zabbix/etc/zabbix\_server.conf

[root@mysqla init.d]# netstat -anpt |grep zabbix\_server

tcp 0 0 0.0.0.0:10051 0.0.0.0:\* LISTEN 27199/zabbix\_server

添加开机启动脚本

[root@mysqla ~]# cd zabbix-3.2.1/

# cp misc/init.d/fedora/core/zabbix\_server /etc/rc.d/init.d/zabbix\_server

# cp misc/init.d/fedora/core/zabbix\_agentd /etc/rc.d/init.d/zabbix\_agentd

[root@mysqla zabbix-3.2.1]# chmod +x /etc/rc.d/init.d/zabbix\_server

[root@mysqla zabbix-3.2.1]# chmod +x /etc/rc.d/init.d/zabbix\_agentd

[root@mysqla zabbix-3.2.1]# chkconfig --add zabbix\_server

[root@mysqla zabbix-3.2.1]# chkconfig --add zabbix\_agentd

[root@mysqla zabbix-3.2.1]# chkconfig zabbix\_server on

[root@mysqla zabbix-3.2.1]# chkconfig zabbix\_agentd on

修改zabbix开机启动脚本中的zabbix安装目录

vi /etc/rc.d/init.d/zabbix\_server #编辑服务端配置文件

BASEDIR=/usr/local/zabbix/ #zabbix安装目录

PIDFILE=/usr/local/zabbix/logs/$BINARY\_NAME.pid #pid文件路径

:wq! #保存退出

vi /etc/rc.d/init.d/zabbix\_agentd #编辑客户端配置文件

BASEDIR=/usr/local/zabbix/ #zabbix安装目录

PIDFILE=/usr/local/zabbix/logs/$BINARY\_NAME.pid #pid文件路径

:wq! #保存退出

[root@mysqla zabbix-3.2.1]# systemctl daemon-reload

[root@mysqla zabbix-3.2.1]# /etc/init.d/zabbix\_server stop

Stopping zabbix\_server (via systemctl): [ OK ]

[root@mysqla zabbix-3.2.1]# netstat -anpt | grep zabbix

[root@mysqla zabbix-3.2.1]# /etc/init.d/zabbix\_server start

Starting zabbix\_server (via systemctl): [ OK ]

[root@mysqla zabbix-3.2.1]# netstat -anpt | grep zabbix

tcp 0 0 0.0.0.0:10051 0.0.0.0:\* LISTEN 28106/zabbix\_server

设置防火墙规则

[root@mysqla zabbix-3.2.1]# firewall-cmd --permanent --add-port=10051/tcp

success

[root@mysqla zabbix-3.2.1]# firewall-cmd --reload

success

配置zabbix web页面

在安装目录将frontends拷贝到指定的web root:

[root@mysqla ~]# cd /root/zabbix-3.2.1/

[root@mysqla zabbix-3.2.1]# cp -r frontends/php/ /usr/local/nginx1.10/html/zabbix

[root@mysqla zabbix-3.2.1]# chown -R www:www /usr/local/nginx1.10/html/zabbix/

**注：/usr/local/nginx/html为Nginx默认站点目录 www为Nginx运行账户**

注:PHP需要至少开启扩展:

gd,bcmath,ctype,libXML,xmlreader,xmlwriter,session,sockets,mbstring,gettext,mysql  
如下，查看是否包括了上面所提到的扩展模块

**[root@mysqla ~]# /usr/local/php5.6/bin/php -m**

**[PHP Modules]**

**bcmath**

**bz2**

**Core**

**ctype**

**curl**

**date**

**dom**

**ereg**

**fileinfo**

**filter**

**ftp**

**gd**

**gettext**

**hash**

**iconv**

**json**

**libxml**

**mbstring**

**mcrypt**

**mhash**

**mysql**

**mysqli**

**openssl**

**pcntl**

**pcre**

**PDO**

**pdo\_sqlite**

**Phar**

**posix**

**Reflection**

**session**

**shmop**

**SimpleXML**

**soap**

**sockets**

**SPL**

**sqlite3**

**standard**

**sysvsem**

**tokenizer**

**xml**

**xmlreader**

**xmlrpc**

**xmlwriter**

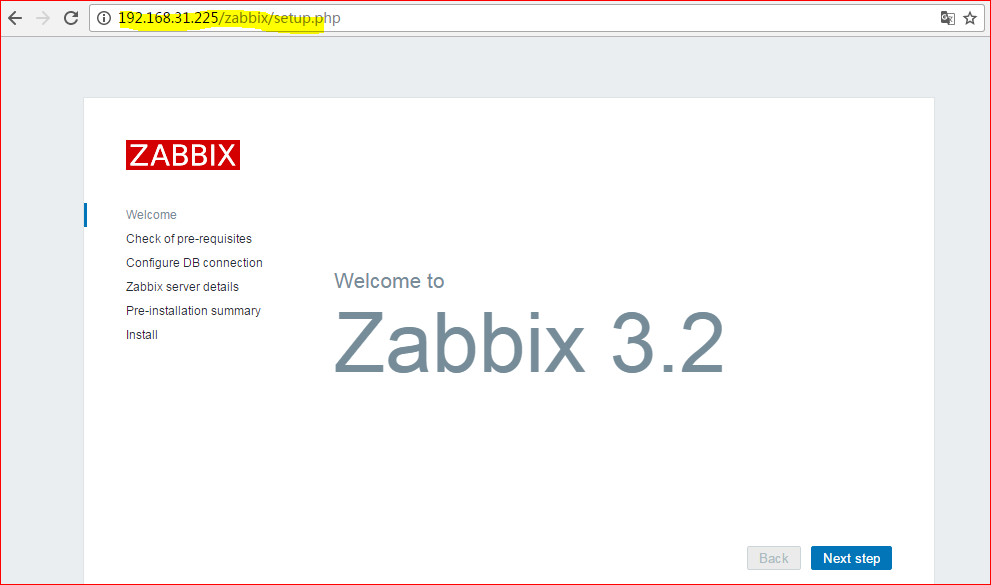
**zip**

**zlib**

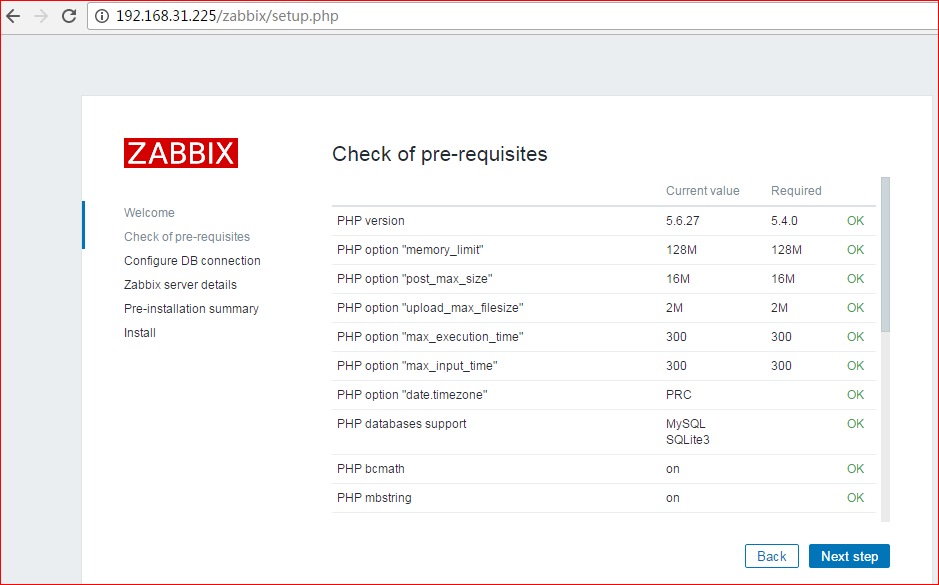
**安装web**

访问Web界面http://192.168.31.225/zabbix,进行相关web配置,配置完成后使用默认用户admin(密码:zabbix)登陆即可

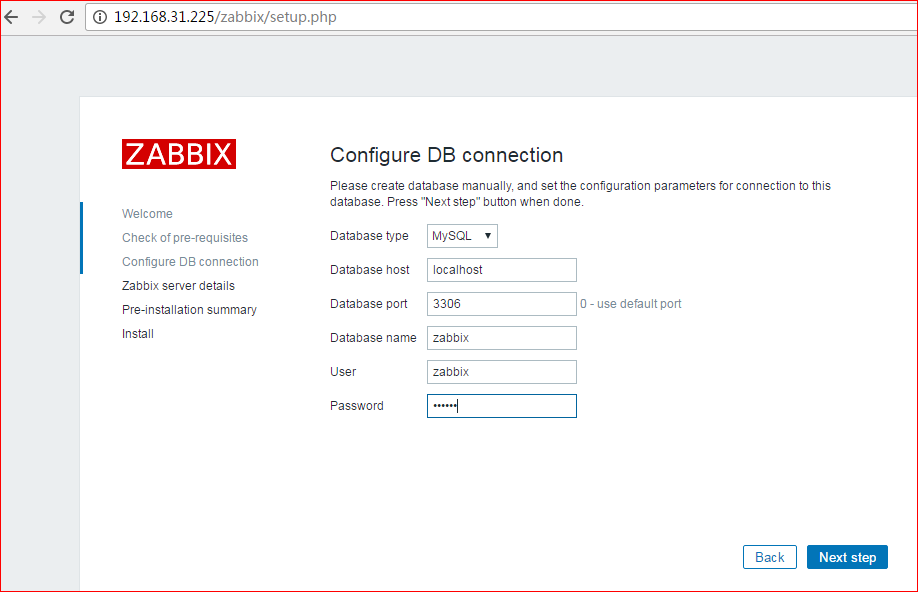
进入ZABBIX WEB安装画面



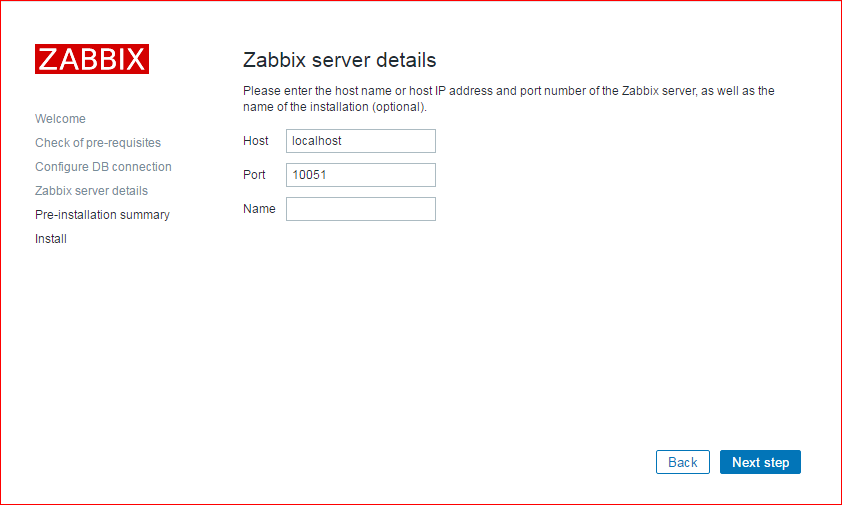
进入ZABBIX检测画面



对数据库进行设置，这里如果数据库在本地的话端口可以使用0

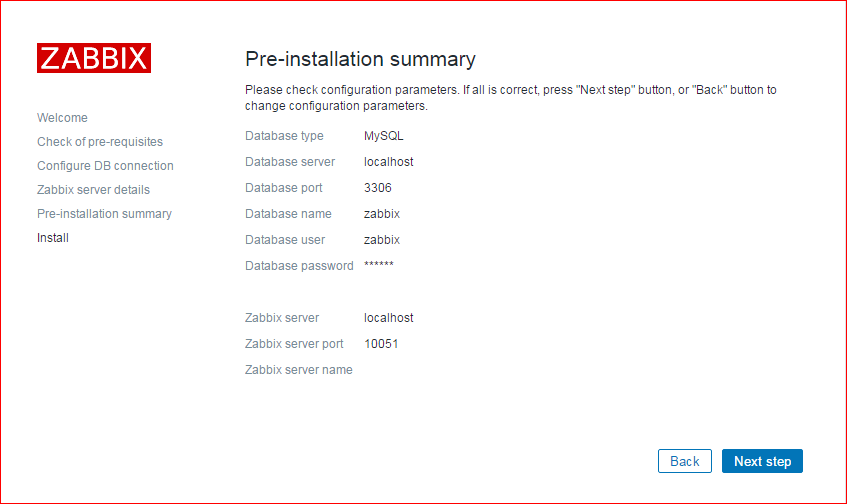


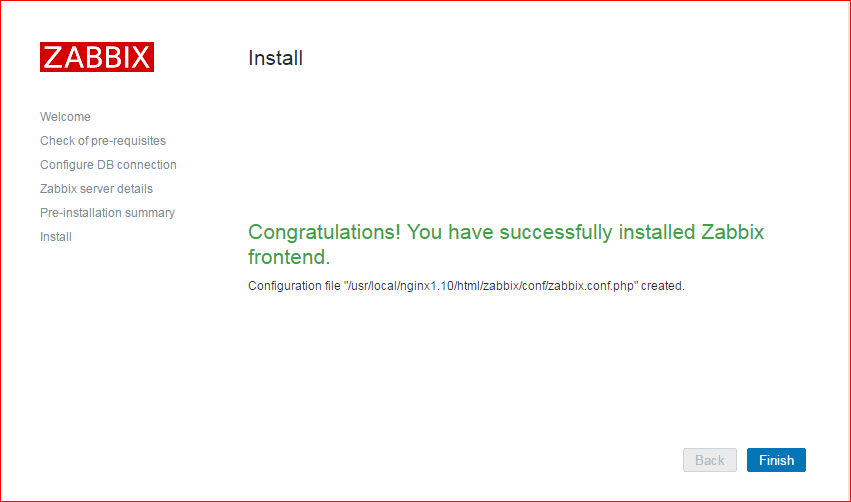
点击Next step 一下步进行zabbix server 细节的设置  这一步可以默认



点击安装，安装完成后  如果没有错误就会进入完成画面

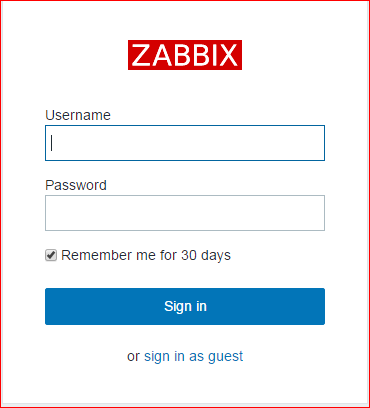
完成画面有设置文件放在服务器的位置，请记下来





点击Finish 完成

使用Admin  用默认密码 zabbix 登录



登录后画面如下



至此 ZABBIX的基础安装完成

注：

1. 显示简体中文界面

在安装数据库时已经将zabbix库设置了utf-8字符

**首先确定zabbix开启了中文支持功能：**  
**登录到zabbix服务器的数据目录下（前面部署的zabbix数据目录是/usr/local/nginx1.10/html/zabbix/），打开 locales.inc.php文件**

[root@mysqla include]# pwd

/usr/local/nginx1.10/html/zabbix/include

[root@mysqla include]# vi locales.inc.php

function getLocales() {

return [

'en\_GB' => ['name' =>\_('English (en\_GB)'), 'display' => true],

'en\_US' => ['name' =>\_('English (en\_US)'), 'display' => true],

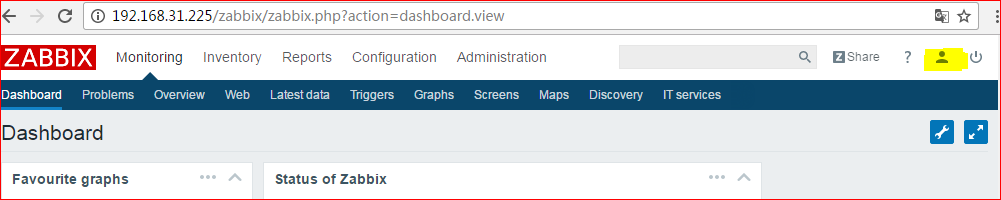
'bg\_BG' => ['name' =>\_('Bulgarian (bg\_BG)'), 'display' => false],

'zh\_CN' => ['name' =>\_('Chinese (zh\_CN)'), 'display' => true],

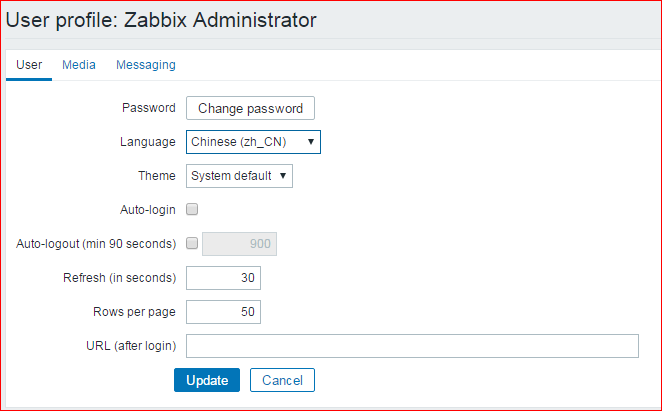
'zh\_TW' => ['name' =>\_('Chinese (zh\_TW)'), 'display' => false],

'cs\_CZ' => ['name' =>\_('Czech (cs\_CZ)'), 'display' => true],

'nl\_NL' => ['name' =>\_('Dutch (nl\_NL)'), 'display' => false],

登陆zabbix后，点击右上角的“用户”图标，将语言设置为“中文”：

按照如图选择Chinese(zh\_CN)，点击update：



解决zabbix绘图中出现中文乱码问题：

a.从windows下控制面板->字体->选择一种中文字库例如“楷体”



b.把它拷贝到zabbix的web端的fonts目录下例如：/usr/local/nginx1.10/html/zabbix/fonts/，并且把TTF后缀改为ttf

[root@mysqla fonts]# pwd

/usr/local/nginx1.10/html/zabbix/fonts

[root@mysqla fonts]# ls

DejaVuSans.ttf simkai.ttf

并且将之前的字体文件DejaVuSans.ttf移动到别处

c.然后，接着修改代码include/defines.inc.php文件中的字体配置，将里面关于字体设置从DejaVuSans替换成simkai

vi替换技巧：%s/DejaVuSans/simkai  
其中：simkai为字库名字,不包含ttf后缀

# cd/usr/local/nginx1.10/html/zabbix

# vi include/defines.inc.php

在vi编辑器的末行模式下输入%s/DejaVuSans/simkai

这样，修改后，zabbix监控图形中的中文字就不会出现乱码了

四、**zabbix客户端的安装**

既然要监控我们就要添加要监控的主机，在添加主机之前我们首先要在被检测主机上面安装agent，安装agent比较简单，我们也是按照安装server的流程，下载软件包，在编译的时候，我们只选择agent

192.168.31.250作为zabbix的被监控端，提供web和mysql应用

安装Zabbix，配置Zabbix agent:

[root@server1 ~]# tar zxf zabbix-3.2.1.tar.gz

[root@server1 ~]# cd zabbix-3.2.1/

[root@server1 zabbix-3.2.1]# ./configure --prefix=/usr/local/zabbix --enable-agent

[root@server1 zabbix-3.2.1]# make&& make install

[root@server1 zabbix-3.2.1]# cp misc/init.d/fedora/core/zabbix\_agentd /etc/init.d/

[root@server1 zabbix-3.2.1]# mkdir -p /usr/local/zabbix/logs

[root@server1 zabbix-3.2.1]# groupadd zabbix

[root@server1 zabbix-3.2.1]# useradd -g zabbix zabbix

[root@server1 zabbix-3.2.1]# chown -R zabbix:zabbix /usr/local/zabbix/

配置开机自动启动:

[root@server1 ~]# chkconfig --add zabbix\_agentd

[root@server1 ~]# chkconfig zabbix\_agentd on

修改zabbix开机启动脚本中的zabbix安装目录

vi /etc/rc.d/init.d/zabbix\_agentd #编辑客户端配置文件

BASEDIR=/usr/local/zabbix/ #zabbix安装目录

PIDFILE=/usr/local/zabbix/logs/$BINARY\_NAME.pid #pid文件路径

:wq! #保存退出

[root@server1 zabbix-3.2.1]# systemctl daemon-reload

编辑zabbix\_agentd.conf

[root@server1 zabbix-3.2.1]# vi /usr/local/zabbix/etc/zabbix\_agentd.conf

内容如下：

[root@server1 zabbix-3.2.1]# grep -v "^#" /usr/local/zabbix/etc/zabbix\_agentd.conf | grep -v "^$"

PidFile=/usr/local/zabbix/logs/zabbix\_agentd.pid

LogFile=/usr/local/zabbix/logs/zabbix\_agentd.log

Server=192.168.31.225

ListenPort=10050

ServerActive=192.168.31.225

Hostname=192.168.31.250

Timeout=15

Include=/usr/local/zabbix/etc/zabbix\_agentd.conf.d/

注：

其中Server和ServerActive都指定zabbixserver的IP地址，不同的是，前者是被动后者是主动。也就是说Server这个配置是用来允许192.168.31.225这个ip来我这取数据。而serverActive的192.168.31.225的意思是，客户端主动提交数据给他。

Hostname=XXX,这个定义的名字必须和web页面里面host的名字一样。

启动zabbix\_agentd:

[root@server1 ~]# /etc/init.d/zabbix\_agentd start

Starting zabbix\_agentd (via systemctl): [ OK ]

[root@server1 ~]# netstat -anpt | grep zabbix\_agentd

tcp 0 0 0.0.0.0:10050 0.0.0.0:\* LISTEN 12926/zabbix\_agentd

[root@server1 ~]# firewall-cmd --permanent --add-port=10050/tcp

success

[root@server1 ~]# firewall-cmd --reload

success

到此，zabbix3.2.1监控系统的基本环境安装完成。

实现zabbix添加监测项

**添加对Linux主机的监控**

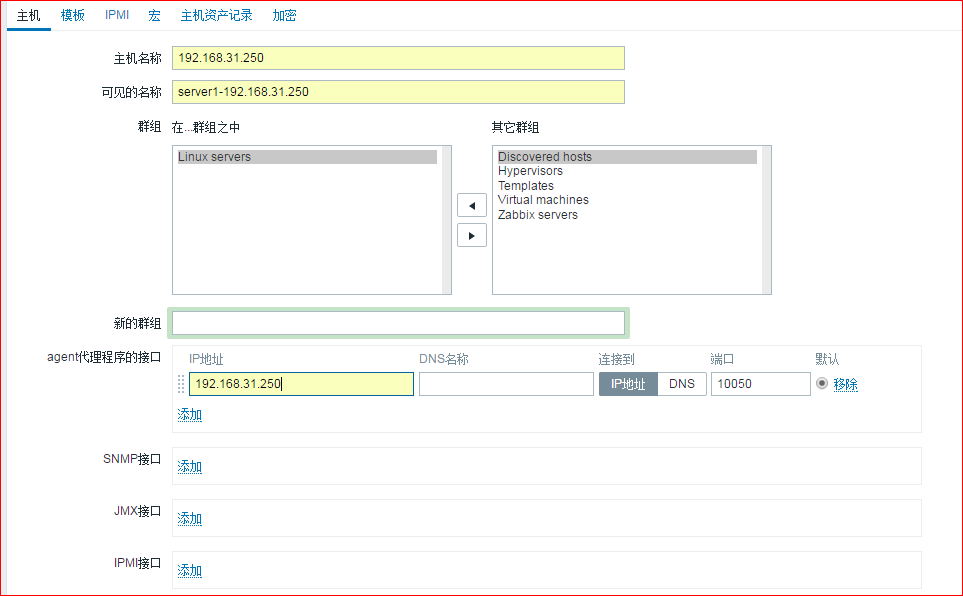
在浏览器中打开：http://192.168.31.225/zabbix

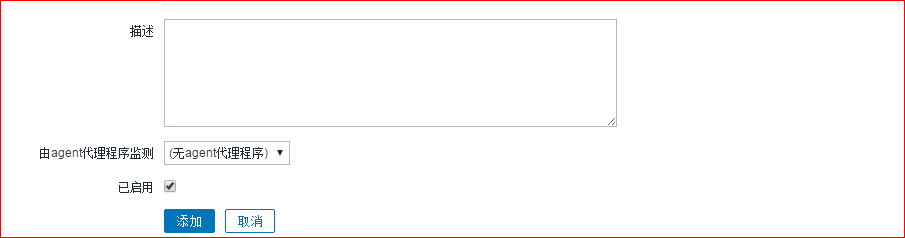
登录zabbix，先在“配置”-“主机”里添加主机监控，监控os资源：内存，cpu，io，负载，带宽等.

登录zabbix，先在“配置”-“主机”里单击”创建主机”



点击”主机”标签，填写相关信息





注意：下图中的“**主机名称**”要和zabbix\_agentd.conf文件中设置的“**Hostname**”后面的名称一致！

主机名称：192.168.31.250

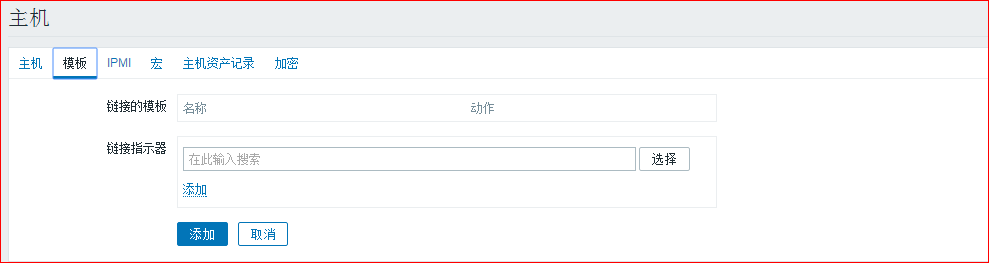
群组：Linux servers

agent代理程序接口ip：192.168.31.250

已启用：勾选

其它选项默认即可

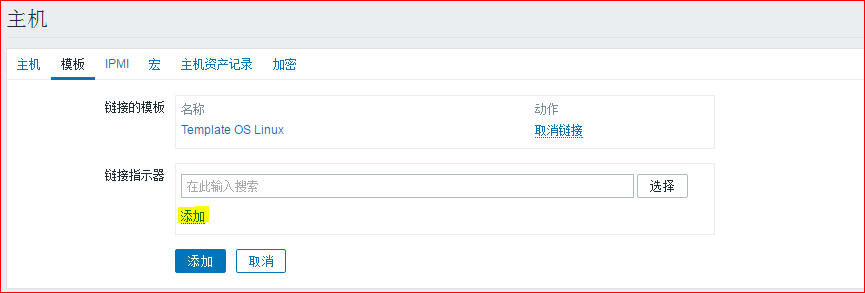
切换到模板



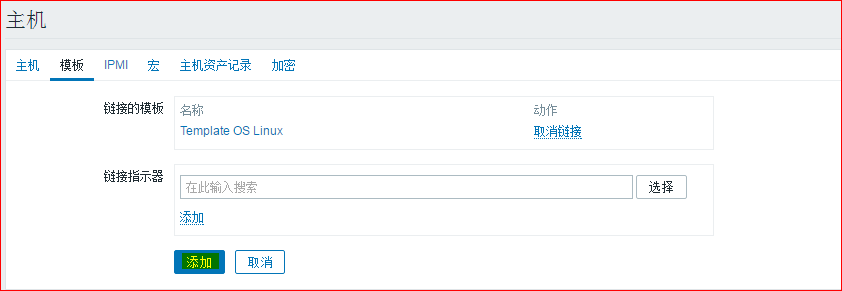
选择

选择：Template OS Linux

点添加



添加



至此，Zabbix监控Linux主机设置完成。



配置过一段时间后，观察下监控图效果出来了没：



zabbix3.0 server已自带[mysql](http://www.07net01.com/tags-mysql-0.html)的模板了，只需配置好agent客户端，然后在web端给主机增加模板就行了。

Zabbix\_agent客户端操作  
（1）首先在客户端的mysql里添加权限，即本机使用zabbix账号连接本地的mysql

mysql> grant all on \*.\* to zabbix@'localhost' identified by "123456”;  
mysql> flush privileges;  
（2）在zabbix\_agent服务目录下创建**.my.cnf** 连接文件  
        zabbix安装目录是/usr/local/zabbix  
[root@server1 ~]# cd /usr/local/zabbix/etc/  
[root@server1 ~]# cat **.my.cnf**   
[client]  
user=zabbix  
password=123456

注意：

如果在数据库grant授权时，针对的是localhost，这个**.my.cnf**里面就不用加host参数了【如上配置】

但如果grant授权时针对的是本机的ip（如192.168.31.250），那么在**.my.cnf**文件里就要加上host参数进行指定了

即在**.my.cnf**文件就要加上：

host=192.168.31.250

socket= /usr/local/mysql/mysql.sock

user=zabbix  
password=123456

(3)配置MySQL的key文件  
    这个可以从zabbix3.2安装时的解压包里拷贝过来

从zabbix\_server服务端安装时解压目录

/root/zabbix-3.2.1/conf/zabbix\_agentd/userparameter\_mysql.conf  拷贝到zabbix\_agent客户端上的/usr/local/zabbix/etc/zabbix\_agentd.conf.d/ 目录下的  
[root@server1 ~]# cd /usr/local/zabbix/etc/zabbix\_agentd.conf.d/

[root@server1 zabbix\_agentd.conf.d]# pwd

/usr/local/zabbix/etc/zabbix\_agentd.conf.d

# cp /root/zabbix-3.2.1/conf/zabbix\_agentd/userparameter\_mysql.conf ./  
[root@server1 zabbix\_agentd.conf.d]# ls userparameter\_mysql.conf

userparameter\_mysql.conf  
然后查看 userparameter\_mysql.conf 文件，看到类似 HOME=/var/lib/zabbix 的路径设置，把路径全都替换为 /usr/local/zabbix/etc/，也就是上面的**.my.cnf**文件所在的目录路径。  
另外，注意userparameter\_mysql.conf 文件里的mysql命令路径（提前做好mysql的系统环境变量，以防mysql命令不被系统识别）

如下：

[root@server1 zabbix\_agentd.conf.d]# cat userparameter\_mysql.conf

# For all the following commands HOME should be set to the directory that has .my.cnf file with password information.

# Flexible parameter to grab global variables. On the frontend side, use keys like mysql.status[Com\_insert].

# Key syntax is mysql.status[variable].

UserParameter=mysql.status[\*],echo "show global status where Variable\_name='$1';" | HOME=/usr/local/zabbix/etc/ mysql -N | awk '{print $$2}'

# Flexible parameter to determine database or table size. On the frontend side, use keys like mysql.size[zabbix,history,data].

# Key syntax is mysql.size[<database>,<table>,<type>].

# Database may be a database name or "all". Default is "all".

# Table may be a table name or "all". Default is "all".

# Type may be "data", "index", "free" or "both". Both is a sum of data and index. Default is "both".

# Database is mandatory if a table is specified. Type may be specified always.

# Returns value in bytes.

# 'sum' on data\_length or index\_length alone needed when we are getting this information for whole database instead of a single table

UserParameter=mysql.size[\*],bash -c 'echo "select sum($(case "$3" in both|"") echo "data\_length+index\_length";; data|index) echo "$3\_length";; free) echo "data\_free";; esac)) from information\_schema.tables$([[ "$1" = "all" || ! "$1" ]] || echo " where table\_schema=\"$1\"")$([[ "$2" = "all" || ! "$2" ]] || echo "and table\_name=\"$2\"");" | HOME=/usr/local/zabbix/etc/ mysql -N'

UserParameter=mysql.ping,HOME=/usr/local/zabbix/etc/ mysqladmin ping | grep -c alive

UserParameter=mysql.version,mysql -V

（4）按照上面修改完之后检查一下，然后重启zabbix\_agent

[root@server1 ~]# pkill -9 zabbix\_agent

[root@server1 ~]# netstat -anpt | grep zabbix\_agent

[root@server1 ~]# /usr/local/zabbix/sbin/zabbix\_agentd

[root@server1 ~]# netstat -anpt | grep zabbix\_agent

tcp 0 0 0.0.0.0:10050 0.0.0.0:\* LISTEN 15400/zabbix\_agentd

**接着在zabbix\_server服务端进行命令行测试［下面的192.168.31.250是客户端的ip］**

**若下面报错**

[root@localhost zabbix]# /usr/local/zabbix/bin/zabbix\_get -s 192.168.31.250 -p 10050 -k "mysql.status[Uptime]"

sh: mysql: command not found

解决方法：

在userparameter\_mysql.conf中关于mysql命令前面添加绝对路径

[root@mysqla ~]#/usr/local/zabbix/bin/zabbix\_get -s 192.168.31.250 -p 10050 -k "mysql.status[Uptime]"

12593

[root@mysqla ~]#

如果出现类似这一串key的数字，就说明配置ok，服务端能监控到客户端的mysql数据了！  
成功啦，之后在监控界面增加主机对应的MySQL模板就ok了。

注：zabbix\_get命令介绍

参数说明：

-s --host：指定客户端主机名或者IP

-p --port：客户端端口，默认10050

-I --source-address：指定源IP，写上zabbix server的ip地址即可，一般留空，服务器如果有多ip的时候，你指定一个。

-k --key：你想获取的key

zabbix\_get获取数据

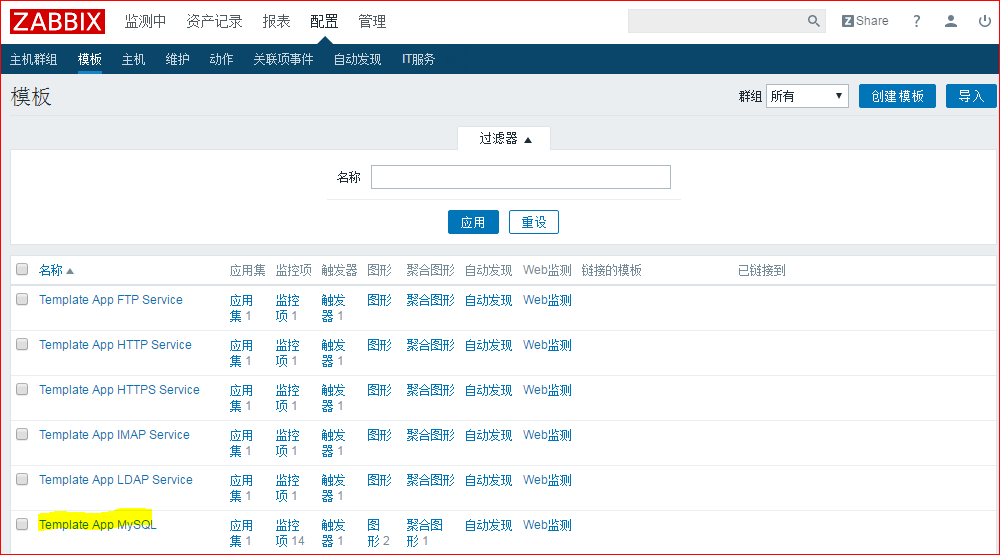
获取负载

# zabbix\_get -s 192.168.31.250 -p 10050 -k "system.cpu.load[all,avg15]"

获取主机名

# zabbix\_get -s 192.168.31.250 -p 10050 -k "system.hostname"

（5）登录zabbix监控界面，在“配置”-“模板”里可以看到自带的mysql监控模板

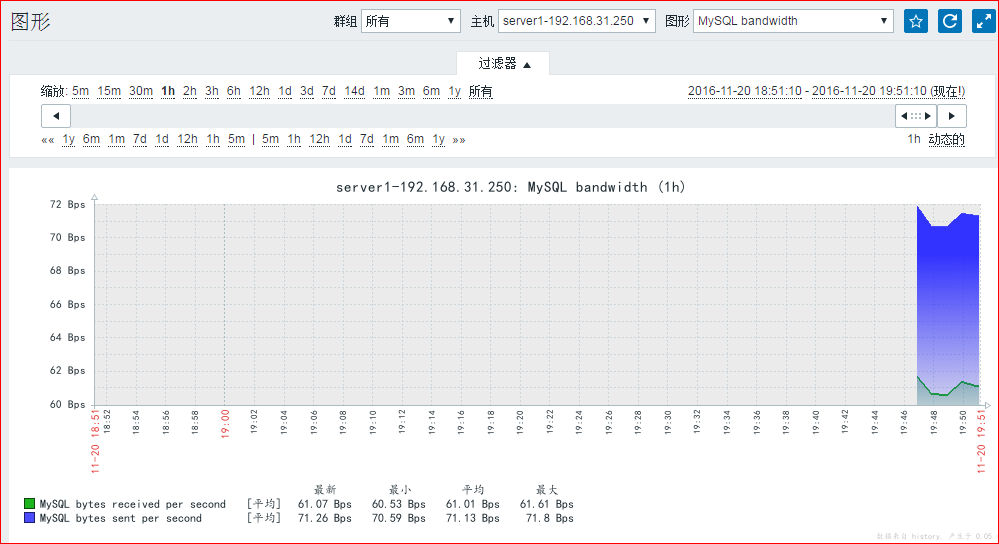


在相应主机监控配置里添加mysql的监控模版（zabbix自带的）即可。

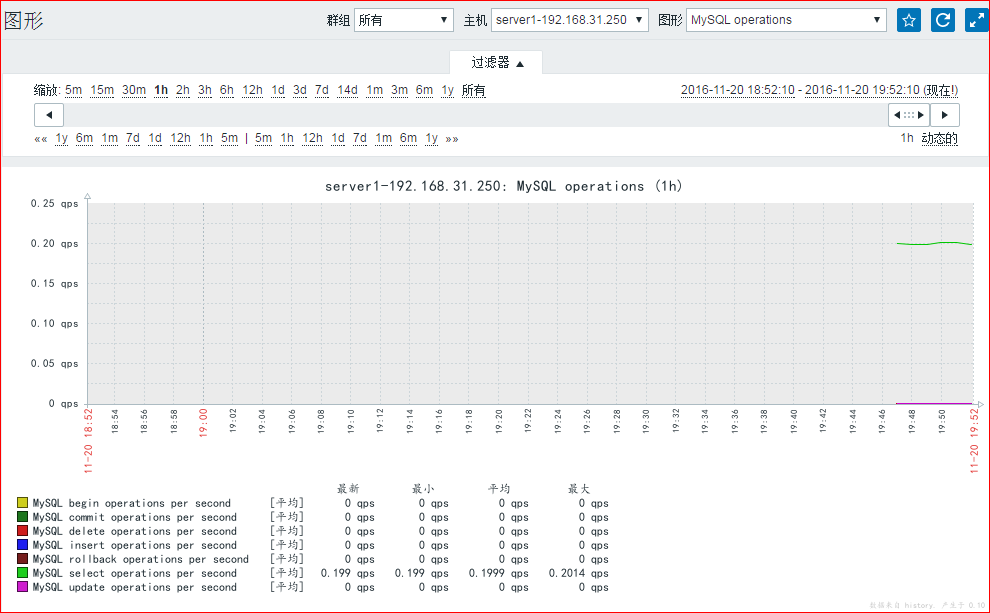


下面是mysql监控项的效果图：

【1】监控mysql的带宽：在zabbix前端可以实时查看mysql发送接收的字节数。其中bytes received表示从所有客户端接收到的字节数，bytes sent表示发送给所有客户端的字节数。



【2】监控mysql的操作状态：在zabbix前端可以实时查看各种SQL语句每秒钟的操作次数。



zabbix3.2.1邮件报警设置：

Zabbix监控服务端、客户端都已经部署完成，被监控主机已经添加，Zabiix监控运行正常。

在Zabbix服务端设置邮件报警，当被监控主机宕机或者达到触发器预设值时，会自动发送报警邮件到指定邮箱。

具体操作：

以下操作在Zabbix监控服务端进行

下载sendemail软件

然后安装那两个依赖包，命令是：

yum install perl-Net-SSLeay perl-IO-Socket-SSL -y

解压软件，然后将sendemail复制到/usr/local/bin/目录下，并加上可执行权限，然后修改用户和群组。

[root@mysqla ~]# tar zxf sendEmail-v1.56.tar.gz

[root@mysqla ~]# cd sendEmail-v1.56/

[root@mysqla sendEmail-v1.56]# cp sendEmail /usr/local/bin/sendEmail

[root@mysqla sendEmail-v1.56]# chown zabbix:zabbix /usr/local/bin/sendEmail

[root@mysqla sendEmail-v1.56]# ll /usr/local/bin/sendEmail

-rwxr-xr-x. 1 zabbix zabbix 80213 Nov 21 21:26 /usr/local/bin/sendEmail

执行# /usr/local/bin/sendEmail -h查看sendEmail的帮助

#/usr/local/bin/sendEmail -f bpcyh1@126.com -t bpcyh1@126.com -s smtp.126.com -u "hello" -o message-content-type=html -o message-charset=utf8 -xu bpcyh1@126.com -xp 123abc -m "abcfdd"

命令主要选项说明：  
/usr/local/bin/sendEmail                     #命令主程序  
-f from@126.com                          #发件人邮箱  
-t to@126.com                              #收件人邮箱  
-s smtp.126.com                           #发件人邮箱的smtp服务器  
-u "我是邮件主题"                               #邮件的标题  
-o message-content-type=html          #邮件内容的格式,html表示它是html格式  
-o message-charset=utf8                   #邮件内容编码  
-xu from@126.com                       #发件人邮箱的用户名  
-xp WEE78@12l$                             #发件人邮箱密码  
-m "我是邮件内容"                             #邮件的具体内容

进入/usr/local/zabbix/share/zabbix/alertscripts/目录

然后在这个目录下创建报警脚本sendmail.sh

[root@mysqla alertscripts]# vi mail.sh

下面是脚本内容（涂颜色部分是你的邮箱和密码，-f和-xu参数后面跟邮箱，-xp后面跟授权码）

#!/bin/bash

to=$1

subject=$2

body=$3

/usr/local/bin/sendEmail -o tls=auto -f bpcyh1@126.com -t "$to" -s smtp.126.com -u "$subject" -o message-content-type=html -o message-charset=utf8 -xu bpcyh1@126.com -xp 123abc -m "$body"

[root@mysqla alertscripts]# chmod +x mail.sh

执行脚本测试发邮件

[root@mysqla alertscripts]# ./mail.sh bpcyh1@126.com test 123

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Using the default of SSL\_verify\_mode of SSL\_VERIFY\_NONE for client

is deprecated! Please set SSL\_verify\_mode to SSL\_VERIFY\_PEER

together with SSL\_ca\_file|SSL\_ca\_path for verification.

If you really don't want to verify the certificate and keep the

connection open to Man-In-The-Middle attacks please set

SSL\_verify\_mode explicitly to SSL\_VERIFY\_NONE in your application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

at /usr/local/bin/sendEmail line 1906.

invalid SSL\_version specified at /usr/share/perl5/vendor\_perl/IO/Socket/SSL.pm line 415

注：进行发送邮件测试时，出现了第一个报错

原因是sendemail软件和perl5里面的ssl版本不兼容导致的，通过修改sendemail里面的内容可以解决这个问题。

解决方法：

在sendemail中1906行的内容，从

if (! IO::Socket::SSL->start\_SSL($SERVER, SSL\_version => 'SSLv3 TLSv1')) {

修改为

if (! IO::Socket::SSL->start\_SSL($SERVER)) {

即不指定SSL的版本，让程序来自己选择

不管你使用的是163/126邮箱还是qq邮箱，你都必须开通pop3服务，而且126/qq邮箱还要求使用第三方客户端发送邮件是必须使用**授权码**登录，使用账号密码验证时，会出现下面图片中的错误。

sendEmail[5102]: ERROR => ERROR => SMTP-AUTH: Authentication to smtp.126.com:25 failed.

至于邮箱的授权码，自己登录126邮箱去设置就好。

经过上面两个问题的处理后就可以正常发送邮件如下所示显示成功提示

[root@mysqla alertscripts]# ./mail.sh bpcyh1@126.com test 123

Nov 21 22:50:47 mysqla sendEmail[5195]: Email was sent successfully!

zabbix web端配置

上面的邮件脚本配置好后，就要在web端添加报警媒介，在web页面的管理，报警媒介类型页面，点击右边的创建媒体类型

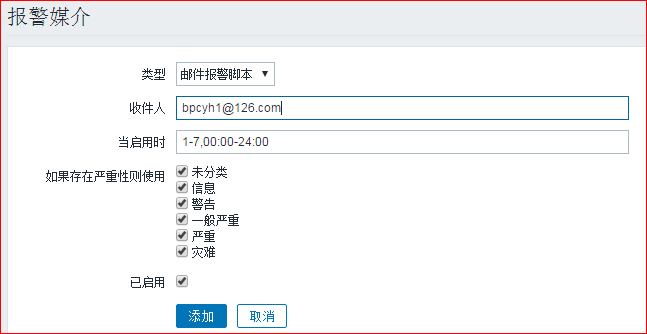


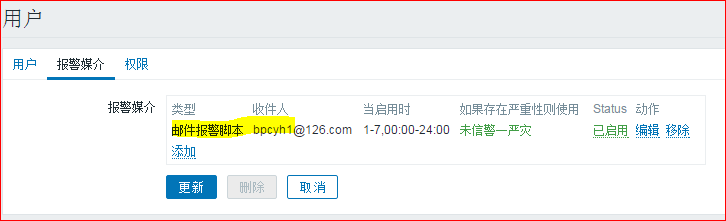
进入创建页面，然后输入名称“邮件报警脚本”，类型选择脚本，名称选择mail.sh，点击添加按钮，添加三个脚本参数，分别是{ALERT.SENDTO},{ALERT.SUBJECT},{ALERT.MESAGE}，分别对于mail.sh中的to、subject和body。添加完成后点下面的添加按钮，我们的报警媒介就创建完成了





创建好报警媒介后，就需要将报警媒介关联到我们的用户。点击管理，用户，然后选择我们的用户admin也可新建一个用户，点击用户名就进入到用户配置界面。点击报警媒介，然后点添加，选择我们上一步创建的邮件报警脚本，输入收件人，选择报警时间和类型，然后点击添加，就将创建好的报警媒介关联到我们的用户上面了。





现在需要做的是，创建一个触发动作，即系统出现故障时，触发这个动作，然后给我们的邮箱发送报警邮件。点击配置，选择动作，然后点击右边的创建动作。

在动作这一页名称中写入我们定义的动作名以及触发条件



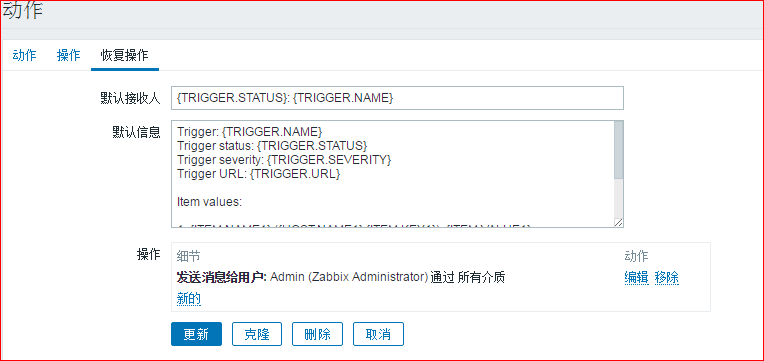
设置“操作”





步骤是1-3,也即是从1开始到3结束。一旦故障发生，就是执行Email.sh脚本发生报警邮件给zabbix用户。   
假如故障持续了1个小时，它也只发送3次，第1-3次（即前3次）邮箱发送给zabbix用户，时间间隔为0秒。  
如果改成1-0，0是表示不限制.无限发送

设置“恢复操作”



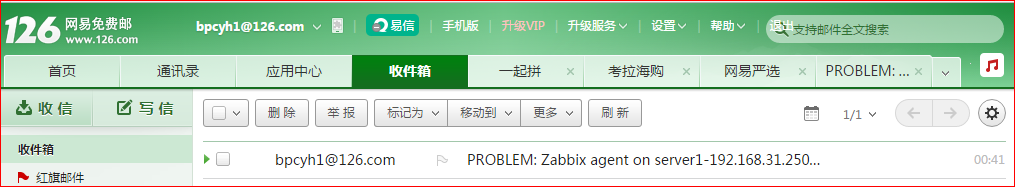
添加好以后就可以看到我们添加的动作名，以及该动作已启用的状态。



报警测试

**将192.16831.250服务器的zabbix\_agent进程关闭。**

**发现可以正常收到故障报警邮件**



**最后，点击”报表“->”动作日志“,可以看到触发动作的次数（右边还看查看到邮件发送状态“已发送”）。**

