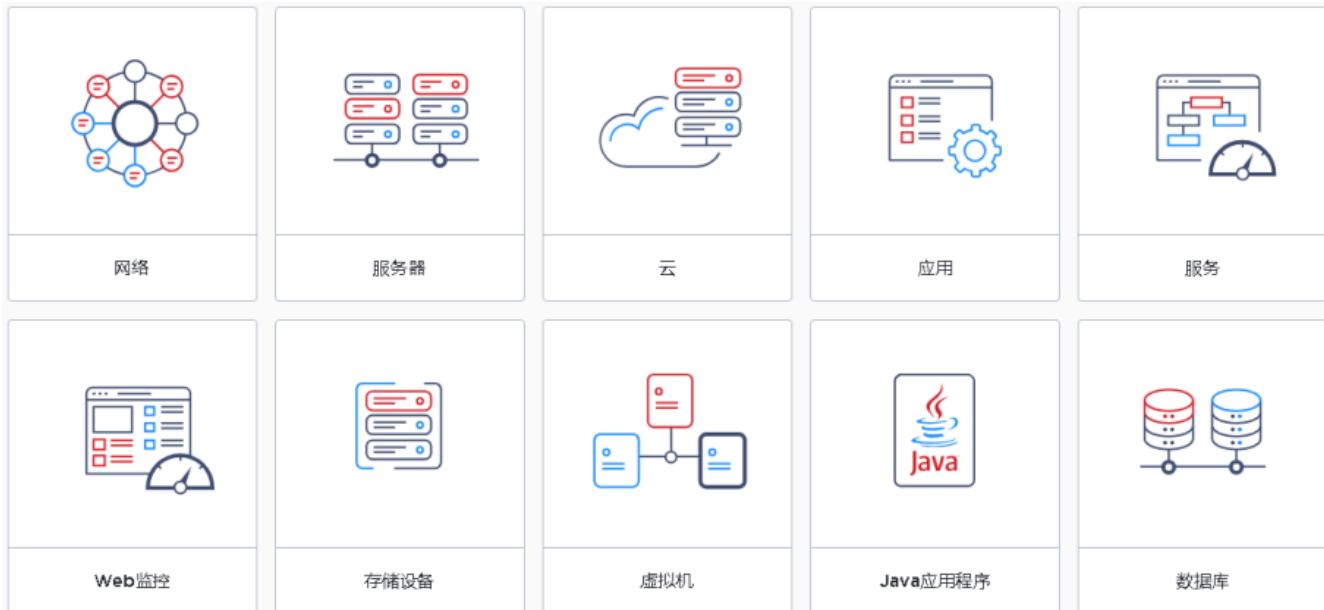


Zabbix

一、Zabbix监控概述

官网介绍：<https://www.zabbix.com/cn/features>

适用于任何IT基础架构、服务、应用程序和资源的解决方案



1.1 : zabbix系统概述

- 数据采集：周期性时序数据
 - 主机/对象：服务器、路由器、交换机、存储、防火墙、IP、PORT、URL、自定义监控对象...
 - 采集目标：监控项，指标数据
- 数据存储：
 - 存储系统：
SQL: MySQL/MariaDB(Zabbix) NoSQL : Redis(Open-falcon) rrd: Round Robin Database(Cacti)
 - 数据：
历史数据: 每个监控项采集到的每个监控值 趋势数据: 趋势表里主要保留某个监控项一个小时内的历史数据的最大值、最小值和平均值以及该监控项一个小时所采集到的数据个数
 - 阈值：severity，可按照等级实现层级报警
 - 告警：email，短信，微信，语音，故障自愈
- 四大核心任务：
 - 采集：zabbix-server, zabbix-proxy,zabbix-agent
 - Agentless : SNMP,Telnet,ssh, IPMI, JMX,
 - Agent : zabbix agent
 - 存储：zabbix database

- 展示：zzabbix web
graph -> screen -> slideshow(将多个screen以幻灯片的方式进行轮流展示)
- 告警：
host (host groups) <- templates host -> items -> triggers -> action (条件-conditions, 操作-operations)

二、zabbix安装

参考官方文档安装：[_](#)

2.1：使用apt命令安装

2.1.1：选择服务器系统版本

The screenshot shows the Zabbix download page at https://www.zabbix.com/cn/download?zabbix=4.2&os_distribution=ubuntu&os_version=18.04_bionic&db=mysql. A large red callout box highlights the '从安装包安装Zabbix' (Install from package) option under '产品用途' (Product Use). Below this, a table allows selecting specific versions for Zabbix, OS distribution, OS version, and database.

ZABBIX版本	OS分布	OS版本	数据库
4.2	Red Hat Enterprise Linux	18.04 (Bionic)	MySQL
4.0 LTS	CentOS	16.04 (Xenial)	PostgreSQL
3.0 LTS	Oracle Linux	14.04 (Trusty)	
2.2 LTS	Ubuntu	Debian	SUSE Linux Enterprise Server

2.1.2：安装数据库

```
root@node1:~# wget https://repo.zabbix.com/zabbix/4.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.0-2+bionic_all.deb
root@node1:~# dpkg -i zabbix-release_4.0-2+bionic_all.deb
root@node1:~# apt update

#修改数据库监听地址
root@node1:~# vim /etc/mysql/mariadb.conf.d/50-server.cnf
bind-address          = 192.168.172.91
root@node1:~# systemctl restart mariadb

#创建zabbix数据库
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@192.168.172.91 identified by 'centos';

#导入初始架构和数据，系统将提示您输入新创建的密码
```

```
root@node1:~# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

2.1.3 : 安装zabbix server , web前端 , agent

```
root@node1:~# apt -y install zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

```
#为zabbix配置数据库
```

```
root@node1:~# vim /etc/zabbix/zabbix_server.conf
```

```
DBHost=192.168.172.91
```

```
DBPassword=centos
```

```
#为zabbix前端配置php
```

```
root@node1:~# vim /etc/zabbix/apache.conf
```

```
<IfModule mod_php7.c>
```

```
...
```

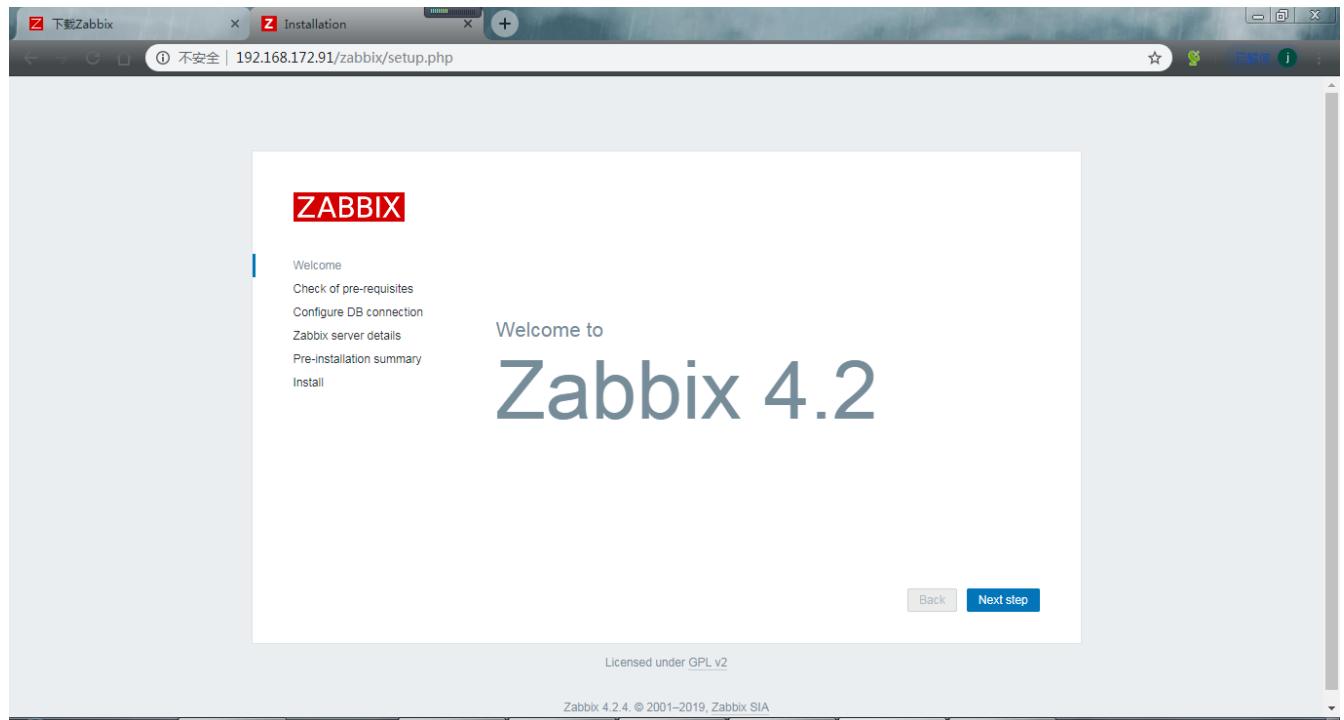
```
    php_value date.timezone Asia/Shanghai
```

```
#启动zabbix server和agent进程
```

```
root@node1:~# systemctl restart zabbix-server zabbix-agent apache2
```

```
root@node1:~# systemctl enable zabbix-server zabbix-agent apache2
```

2.1.4 : 配置zabbix前端



此处检查必须全部为ok才能下一步

ZABBIX

Check of pre-requisites

	Current value	Required
PHP version	7.2.19-0ubuntu0.18.04.1	5.4.0
PHP option "memory_limit"	128M	128M
PHP option "post_max_size"	16M	16M
PHP option "upload_max_filesize"	2M	2M
PHP option "max_execution_time"	300	300
PHP option "max_input_time"	300	300
PHP option "date.timezone"	Asia/Shanghai	
PHP databases support	MySQL	
PHP bcmath	on	
PHP mbstring	on	

Back Next step

Licensed under GPL v2

Zabbix 4.2.4. © 2001–2019, Zabbix SIA

此处应该如实填写

ZABBIX

Configure DB connection

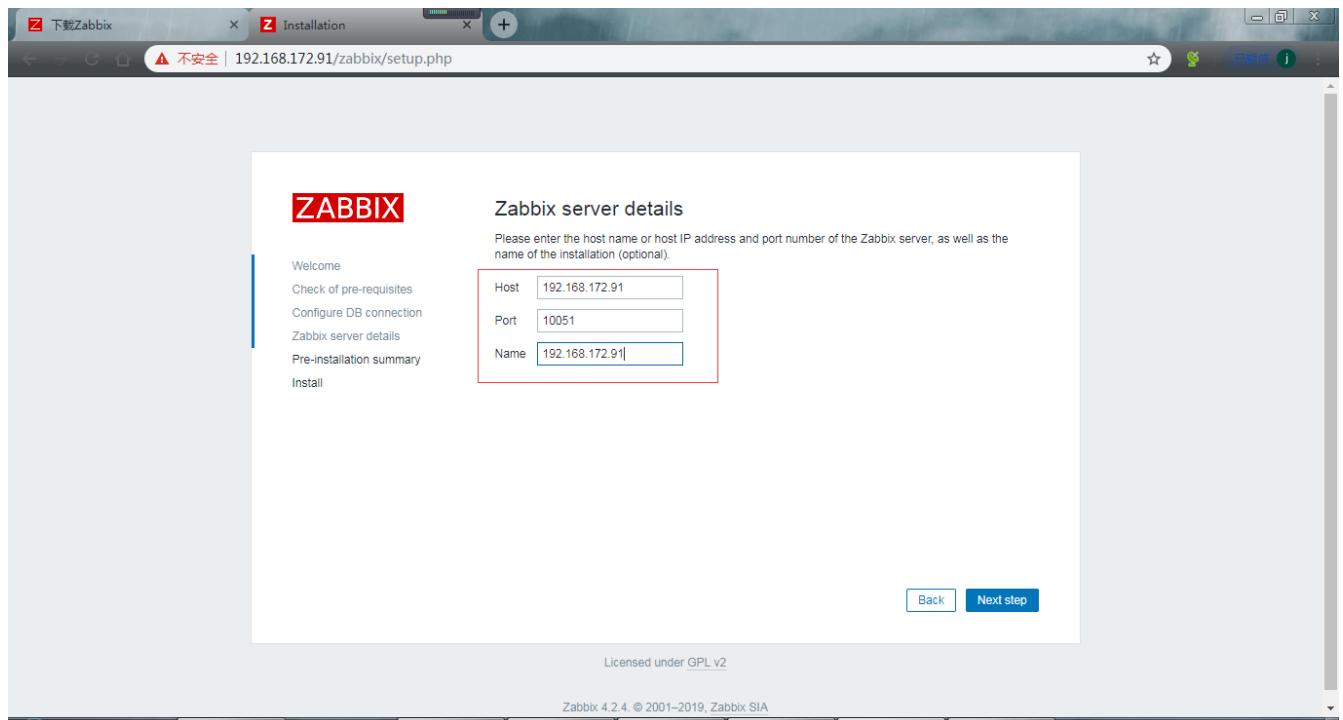
Please create database manually, and set the configuration parameters for connection to this database.
Press "Next step" button when done.

Database type	MySQL	
Database host	192.168.172.91	
Database port	3306	0 - use default port
Database name	zabbix	
User	zabbix	
Password	*****	

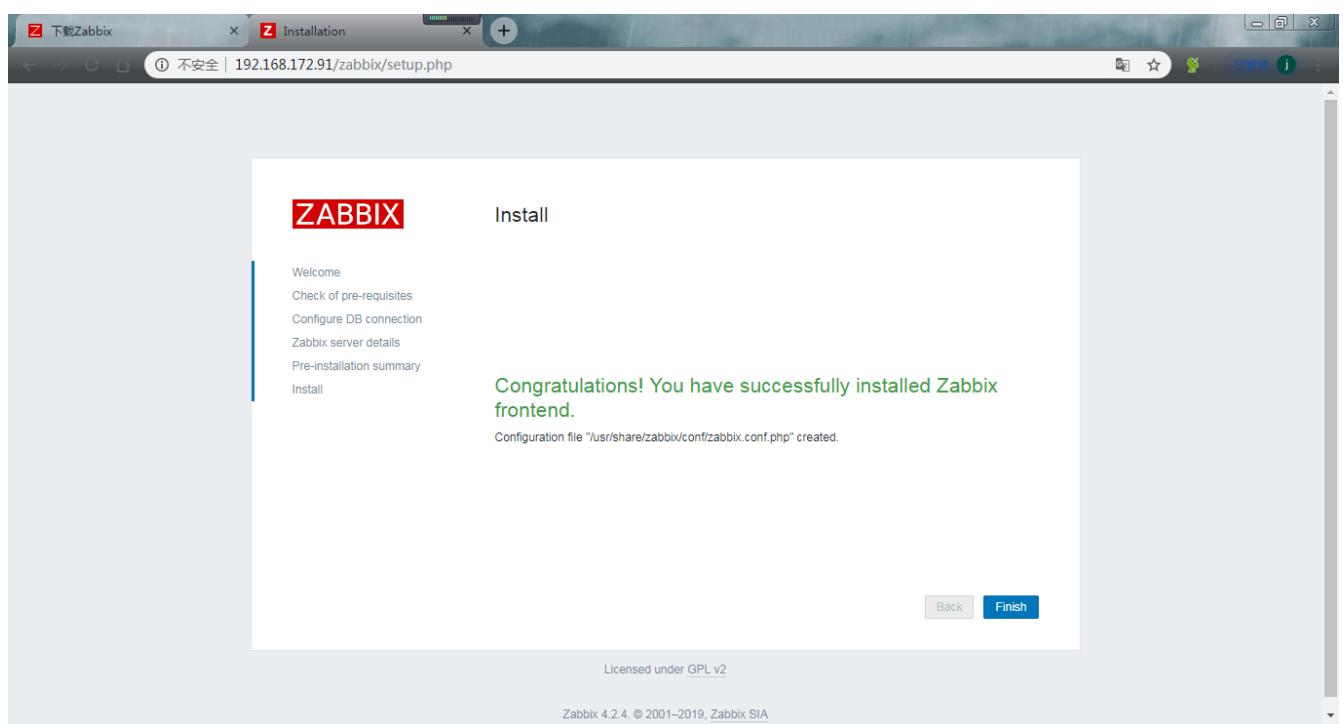
Back Next step

Licensed under GPL v2

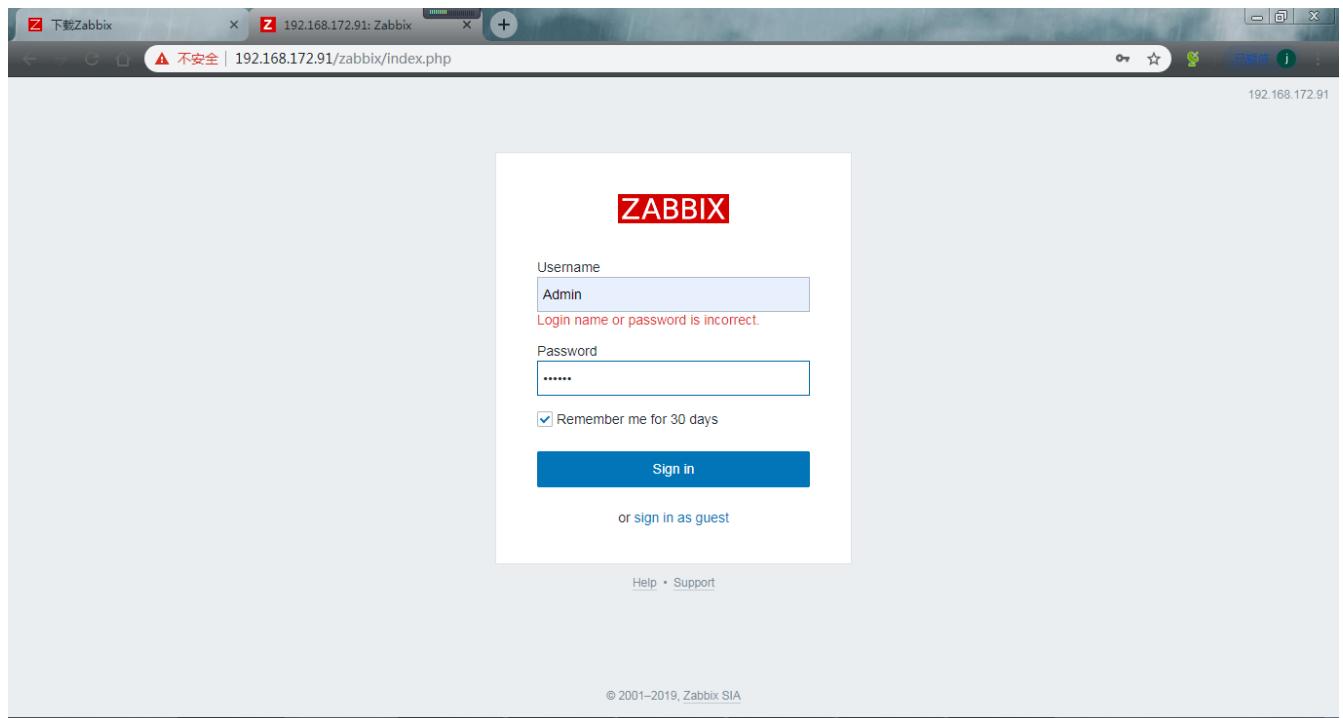
Zabbix 4.2.4. © 2001–2019, Zabbix SIA



确认无误后点击下一步



默认用户名Admin，密码zabbix



登陆之后的页面

2.1.5 : zabbix安装中文语言包

原始的zabbix并没有支持中文

- 安装中文包：

```
root@node1:~# apt-get install language-pack-zh* -y
```

- 配置相关环境变量

```
root@node1:~# vim /etc/environment #在文件中添加语言和编码的设置
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"
LANG="zh_CN.UTF-8"
UNZIP="-O CP936"
ZIPINFO="-O CP936"
```

- 重新设置本地配置

```
root@node1:~# dpkg-reconfigure locales #选择正确的语言即可
```

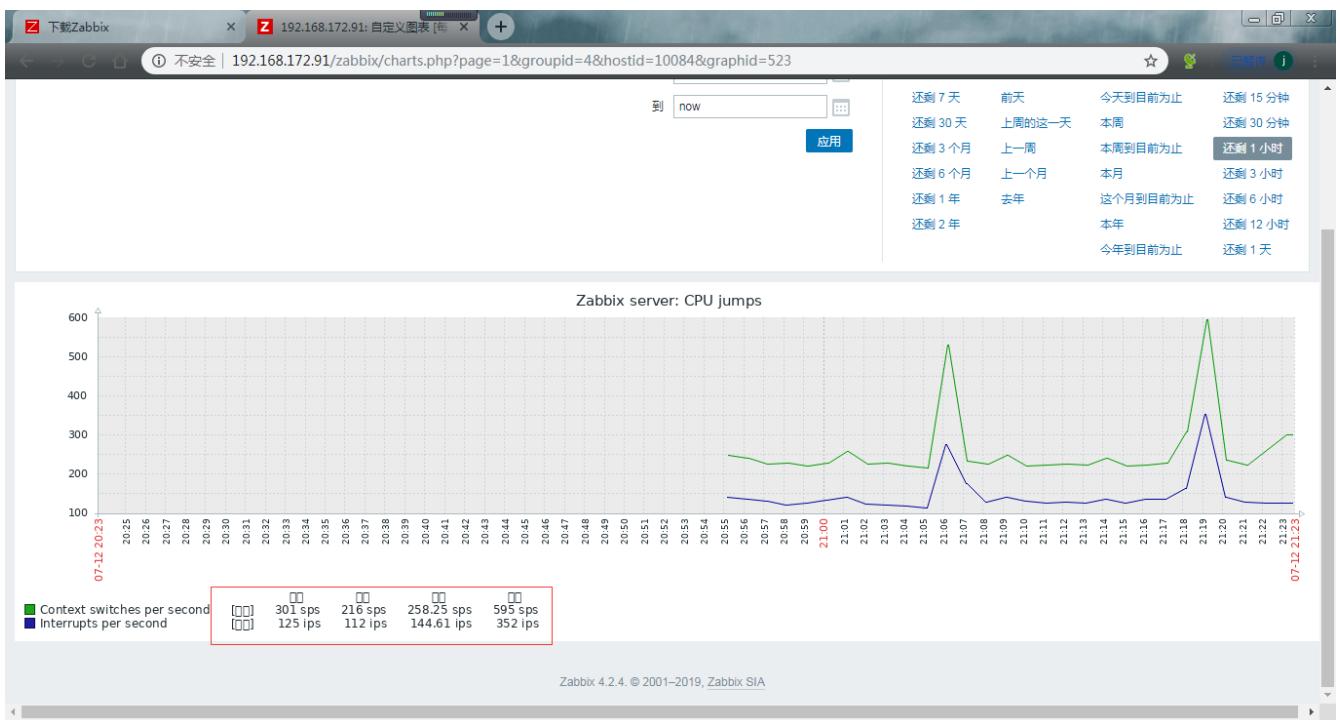
- 重启服务验证

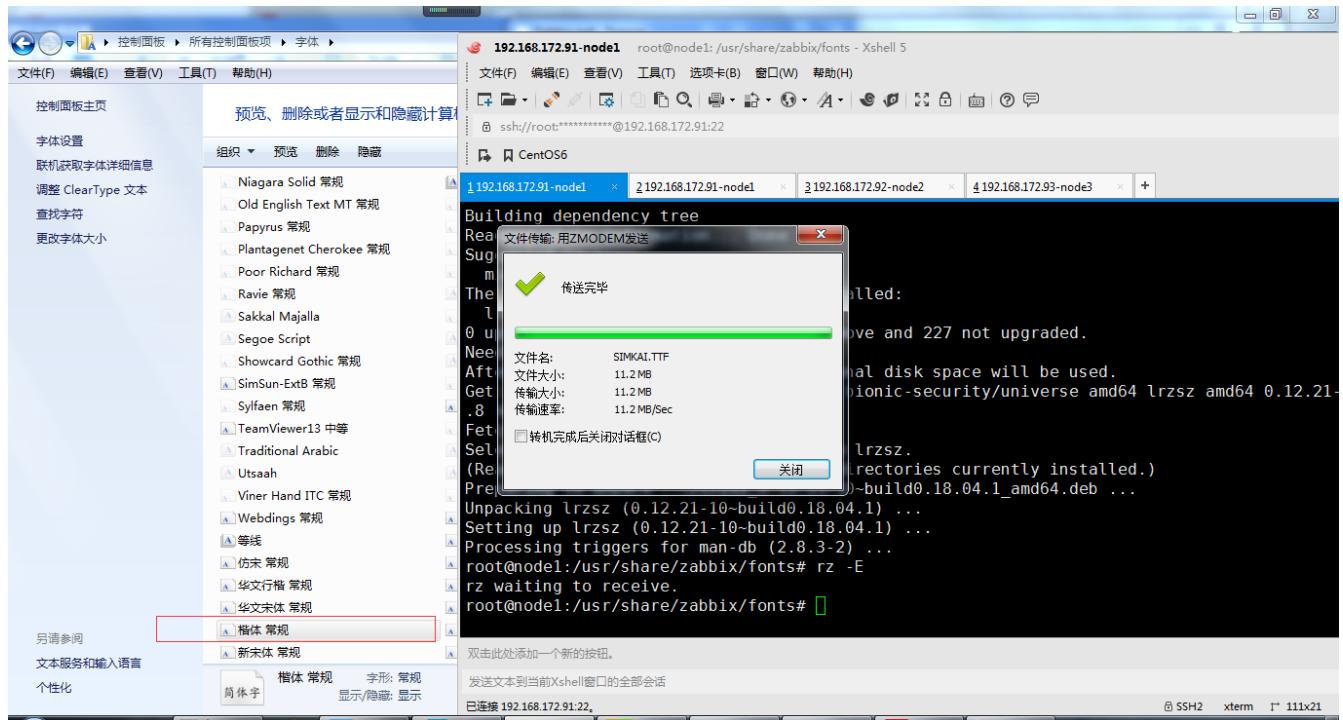
```
root@node1:~# systemctl restart zabbix-server zabbix-agent apache2
```

此时已有中文可供选择

The screenshot shows the 'User profile: Zabbix Administrator' page. In the top-left corner, there is a dropdown menu for 'Language' with 'Chinese (zh_CN)' selected. A red arrow points to this selection. Below the language dropdown, a message states: 'You are not able to choose some of the languages, because locales for them are not installed on the web server.' The page also includes sections for 'Change password', 'Theme' (set to 'ZABBIX'), 'Auto-login', 'Auto-logout', 'Refresh', 'Rows per page', and 'URL (after login)'. At the bottom right, it says 'Zabbix 4.2.4. © 2001–2019, Zabbix SIA'.

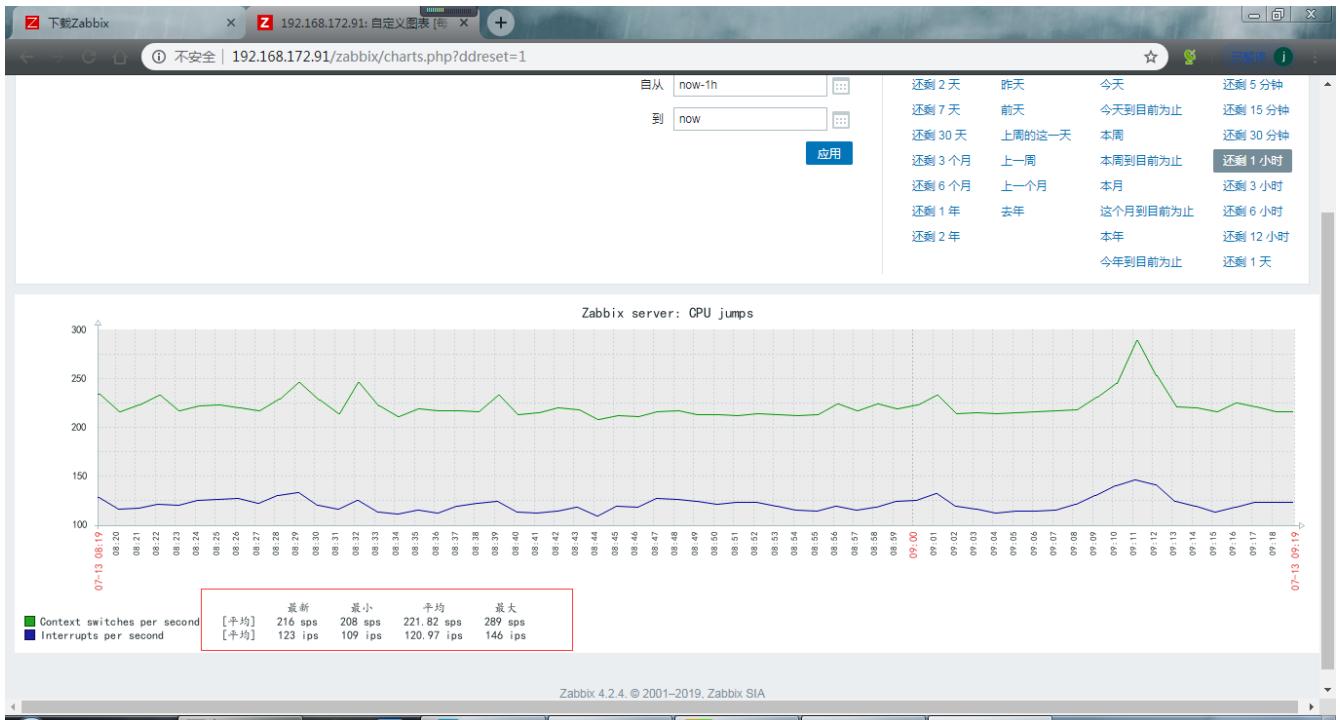
虽然有中文，但还是有些字符显示乱码，需要进一步配置





```
#在windows控制面板中的字体中选择字体放到桌面，再传至此目录
root@node1:/usr/share/zabbix/assets/fonts# pwd
/usr/share/zabbix/assets/fonts
root@node1:/usr/share/zabbix/assets/fonts# ll
total 11520
drwxr-xr-x 2 root root      4096 Jul 12 22:18 .
drwxr-xr-x 5 root root      4096 Jul 12 20:41 ..
lrwxrwxrwx 1 root root      38 Jul 12 20:43 graphfont.ttf -> /etc/alternatives/zabbix-
frontend-font
-rw-r--r-- 1 root root 11785184 Jun 11 2009 simkai.ttf
root@node1:/usr/share/zabbix/assets/fonts# vim /usr/share/zabbix/include/defines.inc.php
define('ZBX_GRAPH_FONT_NAME',           'simkai'); // font file name
define('ZBX_FONT_NAME', 'simkai');
```

重启服务之后再次访问



2.2：使用源码编译安装

IP	ROLE	SERVER	hostname
192.168.172.92	zabbix server	server	node2
192.168.172.93	mysql	mysql , nginx	node3
192.168.172.94	zabbix agent	linux	node4
192.168.172.94	zabbix agent	tomcat	node5

2.2.1：数据库准备

```

root@node3:~# apt-get install mariadb-server -y
root@node3:~# mysql
MariaDB [(none)]> create database zabbix_server character set utf8 collate utf8_bin;
MariaDB [(none)]> grant all privileges on zabbix_server.* to zabbix@"192.168.172.%"
identified by 'centos';
root@node3:~# vim /etc/mysql/mariadb.conf.d/50-server.cnf #修改监听地址
bind-address      = 0.0.0.0
root@node3:~# systemctl restart mariadb

```

node2测试登陆

```
root@node2:~# mysql -uzabbix -pcentos -h192.168.172.93
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.1.40-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

2.2.2：编译安装

安装包下载地址：<https://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/4.0.10/zabbix-4.0.10.tar.gz/download>

- 编译准备

```
#安装依赖包
root@node2:~# apt-get install apache2 apache2-bin apache2-data apache2-utils fontconfig-config fonts-dejavu-core fping libapache2-mod-php libapache2-mod-php7.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libfontconfig1 libgd3 libiksemel3 libjbig0 libjpeg-turbo8 libjpeg8 liblua5.2-0 libodbc1 libopenipmi0 libsensors4 libsnmp-base libsnmp30 libodium23 libssh2-1 libtiff5 libwebp6 libxpm4 php-bcmath php-common php-gd php-ldap php-mbstring php-mysql php-xml php7.2-bcmath php7.2-cli php7.2-common php7.2-gd php7.2-json php7.2-ldap php7.2-mbstring php7.2-mysql php7.2-opcache php7.2-readline php7.2-xml snmpd ssl-cert ttf-dejavu-core libmysqlclient-dev libxml2-dev libxml2 snmp libsnmp-dev libevent-dev openjdk-8-jdk curl libcurl4-openssl-dev -y

root@node2:/usr/local/src# ll
total 16748
drwxr-xr-x 2 root root 4096 Jul 13 09:58 .
drwxr-xr-x 10 root root 4096 Jul 4 09:04 ..
-rw-r--r-- 1 root root 17140053 Jul 13 09:57 zabbix-4.0.10.tar.gz
root@node2:/usr/local/src# useradd -s /sbin/nologin zabbix #创建zabbix用户
root@node2:/usr/local/src# id zabbix
uid=1001(zabbix) gid=1001(zabbix) groups=1001(zabbix)
```

- 开始编译

```
root@node2:/usr/local/src# tar xvf zabbix-4.0.10.tar.gz
root@node2:/usr/local/src# cd zabbix-4.0.10/
root@node2:/usr/local/src/zabbix-4.0.10# ./configure --prefix=/apps/zabbix_server --enable-server --enable-agent --with-mysql --with-net-snmp --with-libcurl --with-libxml2 --enable-java
root@node2:/usr/local/src/zabbix-4.0.10# make && make install
```

- 导入数据库

```
root@node2:/usr/local/src/zabbix-4.0.10# mysql -uzabbix -pcentos -h192.168.172.93  
zabbix_server < database/mysql/schema.sql  
  
root@node2:/usr/local/src/zabbix-4.0.10# mysql -uzabbix -pcentos -h192.168.172.93  
zabbix_server < database/mysql/images.sql  
  
root@node2:/usr/local/src/zabbix-4.0.10# mysql -uzabbix -pcentos -h192.168.172.93  
zabbix_server < database/mysql/data.sql
```

- 修改配置文件

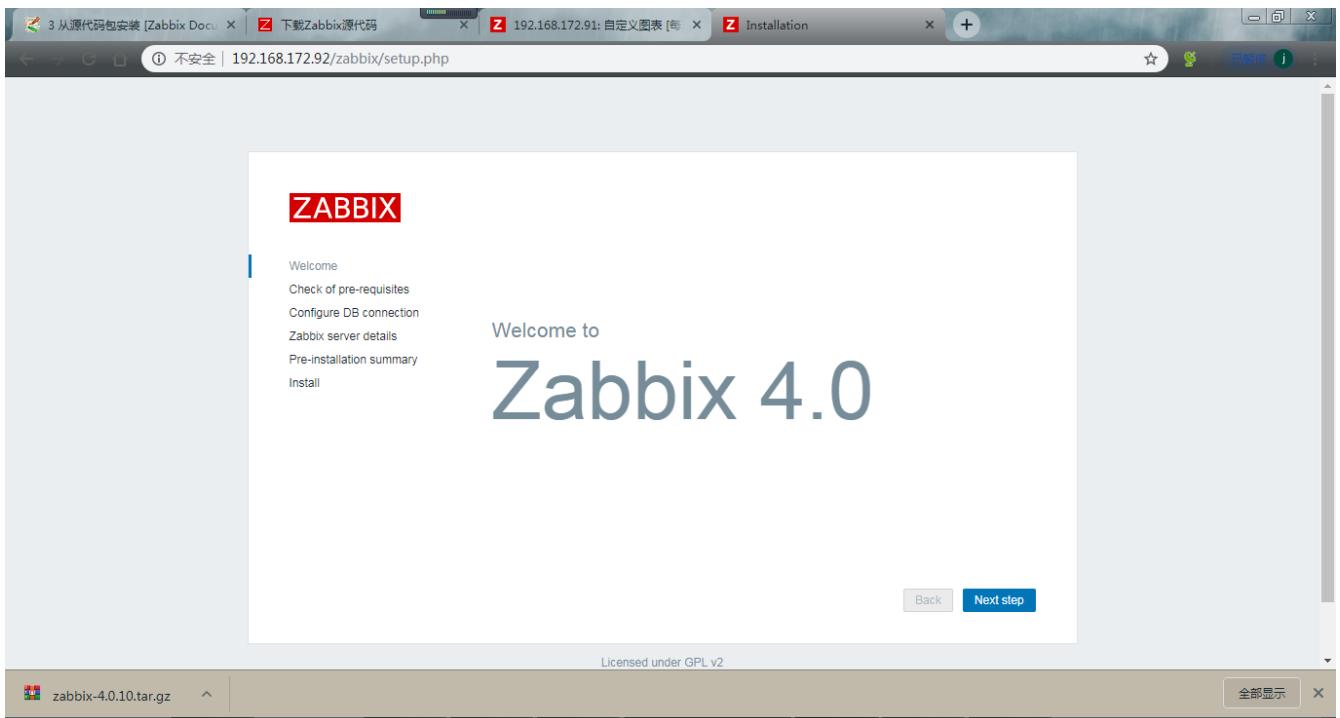
```
root@node2:~# chown -R zabbix.zabbix /apps/zabbix_server  
root@node2:~# mkdir /apps/zabbix_server/run  
root@node2:~# chown -R zabbix.zabbix /apps/zabbix_server/run  
root@node2:~# mkdir /var/log/zabbix  
root@node2:~# chown -R zabbix.zabbix /var/log/zabbix  
  
root@node2:/usr/local/src/zabbix-4.0.10# vim /apps/zabbix_server/etc/zabbix_server.conf  
PidFile=/apps/zabbix_server/run/zabbix_server.pid  
DBHost=192.168.172.93  
DBName=zabbix_server  
DBPassword=centos  
DBPort=3306  
Timeout=30  
LogFile=/var/log/zabbix/zabbix_server.log
```

- 启动zabbix

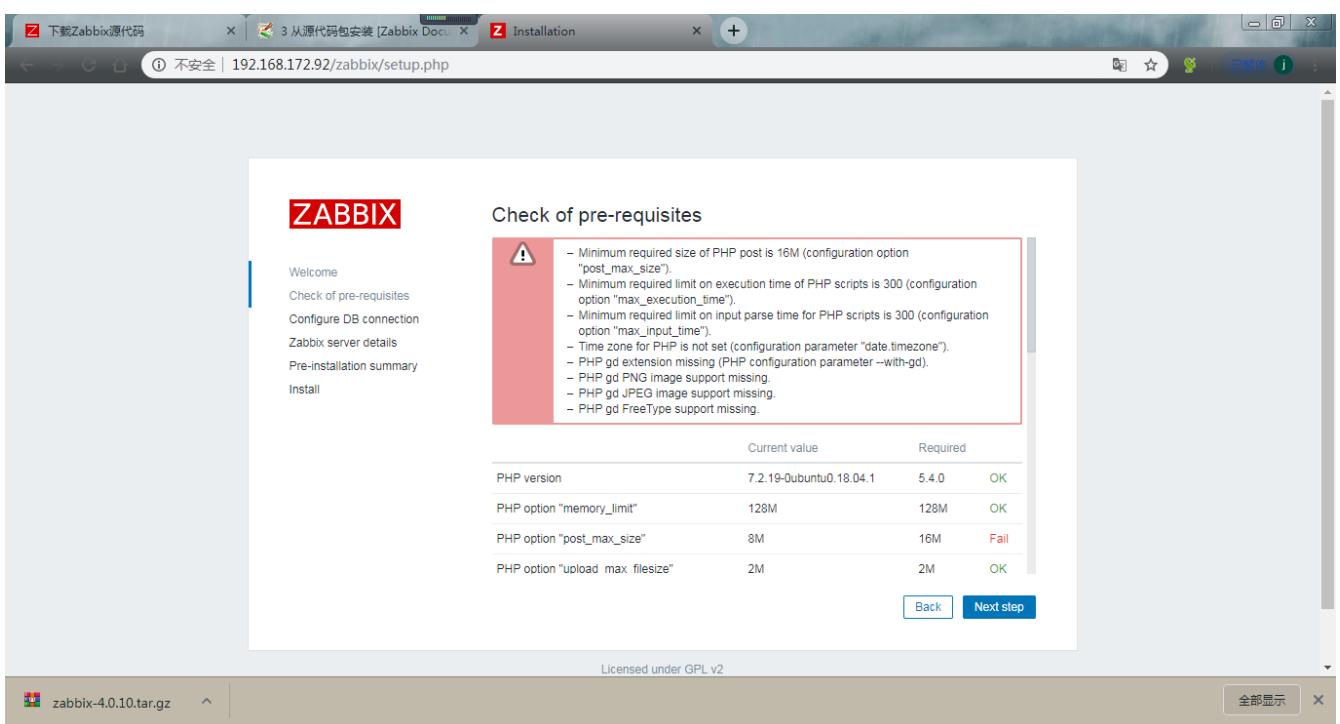
```
root@node2:/usr/local/src/zabbix-4.0.10# /apps/zabbix_server/sbin/zabbix_server -c  
/apps/zabbix_server/etc/zabbix_server.conf  
root@node2:/usr/local/src/zabbix-4.0.10# ss -nlt #查看端口监听  
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port  
LISTEN      0          128         0.0.0.0:10051        0.0.0.0:*
```

2.2.3 : 安装zabbix web界面

```
root@node2:/usr/local/src/zabbix-4.0.10# mkdir /var/www/html/zabbix  
root@node2:/usr/local/src/zabbix-4.0.10# cd frontends/php/  
root@node2:/usr/local/src/zabbix-4.0.10/frontends/php# pwd  
/usr/local/src/zabbix-4.0.10/frontends/php  
root@node2:/usr/local/src/zabbix-4.0.10/frontends/php# cp -a . /var/www/html/zabbix/  
#测试访问web页面
```



排错



```
root@node2:~# vim /etc/php/7.2/apache2/php.ini
post_max_size = 16M
max_execution_time = 300
max_input_time = 300
date.timezone = Asia/Shanghai
root@node2:~# apt-get install php-gettext php-xml php-net-socket php-gd php-mysql -y
root@node2:~# systemctl restart apache2
#修改完后重启apache2服务，再次访问再无报错，继续下一步
```

ZABBIX

Check of pre-requisites

	Current value	Required
PHP version	7.2.19-0ubuntu0.18.04.1	5.4.0 OK
PHP option "memory_limit"	128M	128M OK
PHP option "post_max_size"	16M	16M OK
PHP option "upload_max_filesize"	2M	2M OK
PHP option "max_execution_time"	300	300 OK
PHP option "max_input_time"	300	300 OK
PHP option "date.timezone"	Asia/Shanghai	OK
PHP databases support	MySQL	OK
PHP bcmath	on	OK
PHP mbstring	on	OK

Back Next step



继续安装至此报错，按照提示将文件下载下来传至/var/www/html/zabbix/conf/

ZABBIX

Install

Details ▲ Cannot create the configuration file.
Unable to create the configuration file.

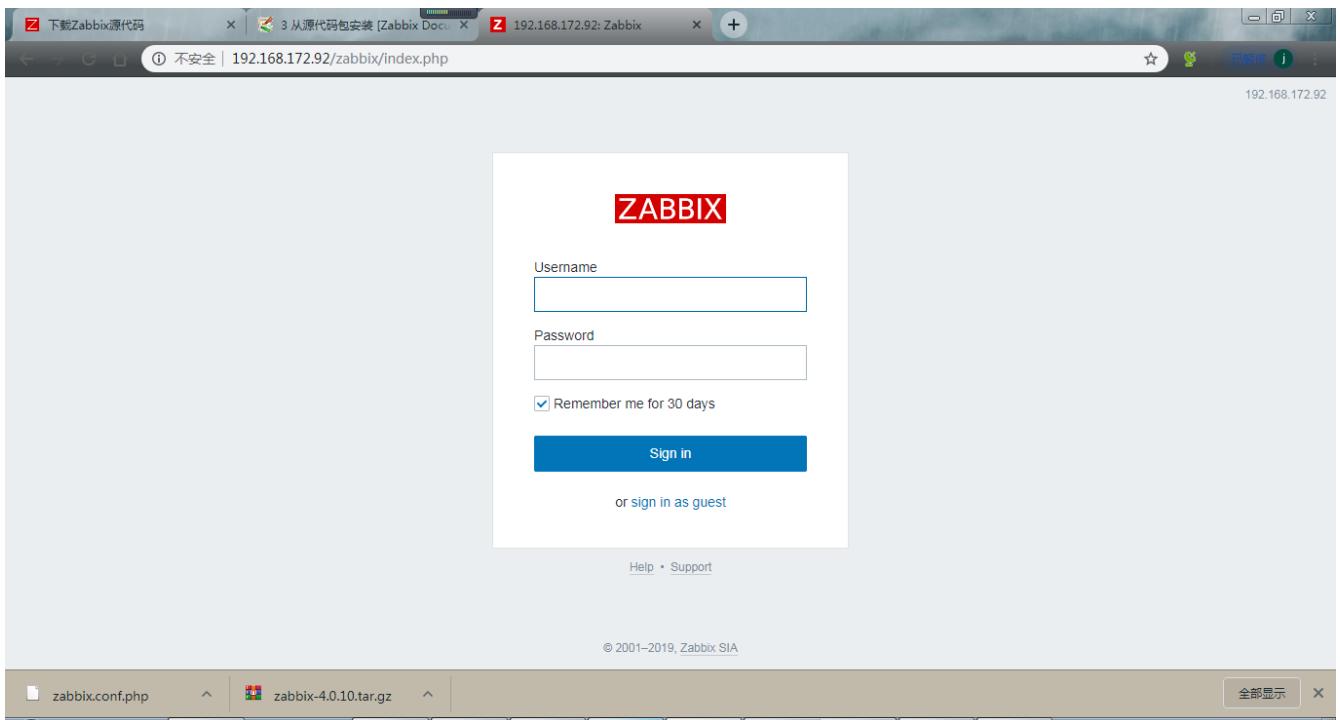
Alternatively, you can install it manually:

1. [Download the configuration file](#)
2. Save it as "/var/www/html/zabbix/conf/zabbix.conf.php"

Back Finish



至此zabbix web安装完成，修改语言已在上一节实现，自行配置即可



```
#编译安装的文件路径
root@node2:~# ll /var/www/html/zabbix/assets/fonts/
total 12260
drwxr-xr-x 2 zabbix zabbix    4096 Jul 13 11:40 .
drwxr-xr-x 5 zabbix zabbix    4096 Jun 26 19:20 ..
-rw-r--r-- 1 zabbix zabbix 756072 Jun 26 19:20 DejaVuSans.ttf
-rw-r--r-- 1 root   root   11785184 Jun 11 2009 simkai.ttf
root@node2:~# vim /var/www/html/zabbix/include/defines.inc.php
```

2.2.4：制作zabbix启动脚本

```
root@node2:/usr/local/src/zabbix-4.0.10# vim /lib/systemd/system/zabbix-server.service
[Unit]
Description=Zabbix Server
After=syslog.target
After=network.target

[Service]
Environment="CONFFILE=/apps/zabbix_server/etc/zabbix_server.conf"
EnvironmentFile=-/etc/default/zabbix-server
Type=forking
Restart=on-failure
PIDFile=/apps/zabbix_server/run/zabbix_server.pid
KillMode=control-group
ExecStart=/apps/zabbix_server/sbin/zabbix_server -c $CONFFILE
ExecStop=/bin/kill -SIGTERM $MAINPID
RestartSec=10s
TimeoutSec=infinity

[Install]
```

```
WantedBy=multi-user.target  
#测试启动停止重启无虞即可
```

三、 zabbix监控

所有主机必须时间同步，否则可能图形显示不出来

3.1 : zabbix agent主机监控

```
#编译安装，编译参数只用--enable-agent  
  
#修改配置文件后启动  
root@node4:~# vim /apps/zabbix_agent/etc/zabbix_agentd.conf  
Server=192.168.172.92 #指向server端ip  
Hostname=192.168.172.94 #将来在仪表盘上的显示名称  
PidFile=/apps/zabbix_agent/run/zabbix_agentd.pid  
LogFile=/apps/zabbix_agent/run/zabbix_agentd.log  
  
root@node4:~# mkdir /apps/zabbix_agent/run  
root@node4:~# chown -R zabbix.zabbix /apps/zabbix_agent  
root@node4:~# /apps/zabbix_agent/sbin/zabbix_agentd -c  
/apps/zabbix_agent/etc/zabbix_agentd.conf  
root@node4:/usr/local/src/zabbix-4.0.10# ss -nlt  
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port  
LISTEN      0          128         0.0.0.0:10050          0.0.0.0:*
```

- zabbix agent启动脚本

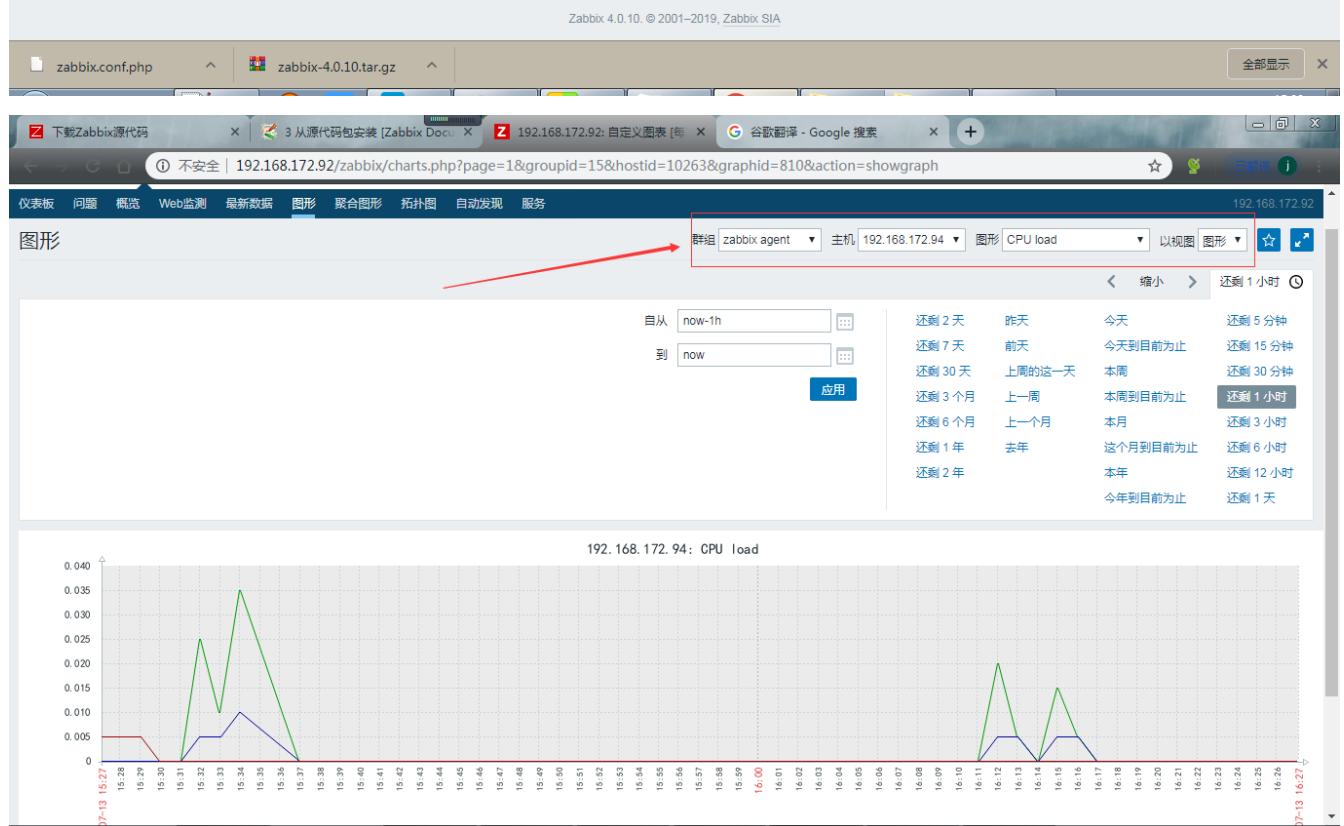
```
root@node4:/usr/local/src/zabbix-4.0.10# vim /lib/systemd/system/zabbix-agent.service  
[Unit]  
Description=Zabbix Agent  
After=syslog.target  
After=network.target  
  
[Service]  
Environment="CONFFILE=/apps/zabbix_agent/etc/zabbix_agentd.conf"  
EnvironmentFile=-/etc/default/zabbix-agent  
Type=forking  
Restart=on-failure  
PIDFile=/apps/zabbix_agent/run/zabbix_agentd.pid  
KillMode=control-group  
ExecStart=/apps/zabbix_agent/sbin/zabbix_agentd -c $CONFFILE  
ExecStop=/bin/kill -SIGTERM $MAINPID  
RestartSec=10s  
  
[Install]  
WantedBy=multi-user.target  
  
#使用pkill命令关闭zabbix agent  
root@node4:~# pkill zabbix_agent  
root@node4:~# systemctl start zabbix-agent
```

```
root@node4:~# ss -nlt
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	0.0.0.0:10050	0.0.0.0:*

创建主机、选择监控模板

The screenshot shows the Zabbix configuration interface for creating a new host. The URL is <http://192.168.172.92/zabbix/hosts.php?ddreset=1>. The main navigation bar includes 'ZABBIX', '监测', '资产记录', '报表', '配置', and '管理'. The '主机' tab is selected. The form fields include '名称' (Name), '模板' (Template) with a search input, 'IP地址' (IP Address), '端口' (Port), and dropdowns for '要监控什么' (What to monitor) and 'agent代理程序' (Agent program). Buttons for '应用' (Apply) and '重设' (Reset) are at the bottom. Below the form is a table listing hosts, with one entry for 'Zabbix server' highlighted. At the bottom are buttons for '选择' (Select), '启用' (Enable), '禁用' (Disable), '导出' (Export), '批量更新' (Batch Update), and '删除' (Delete).



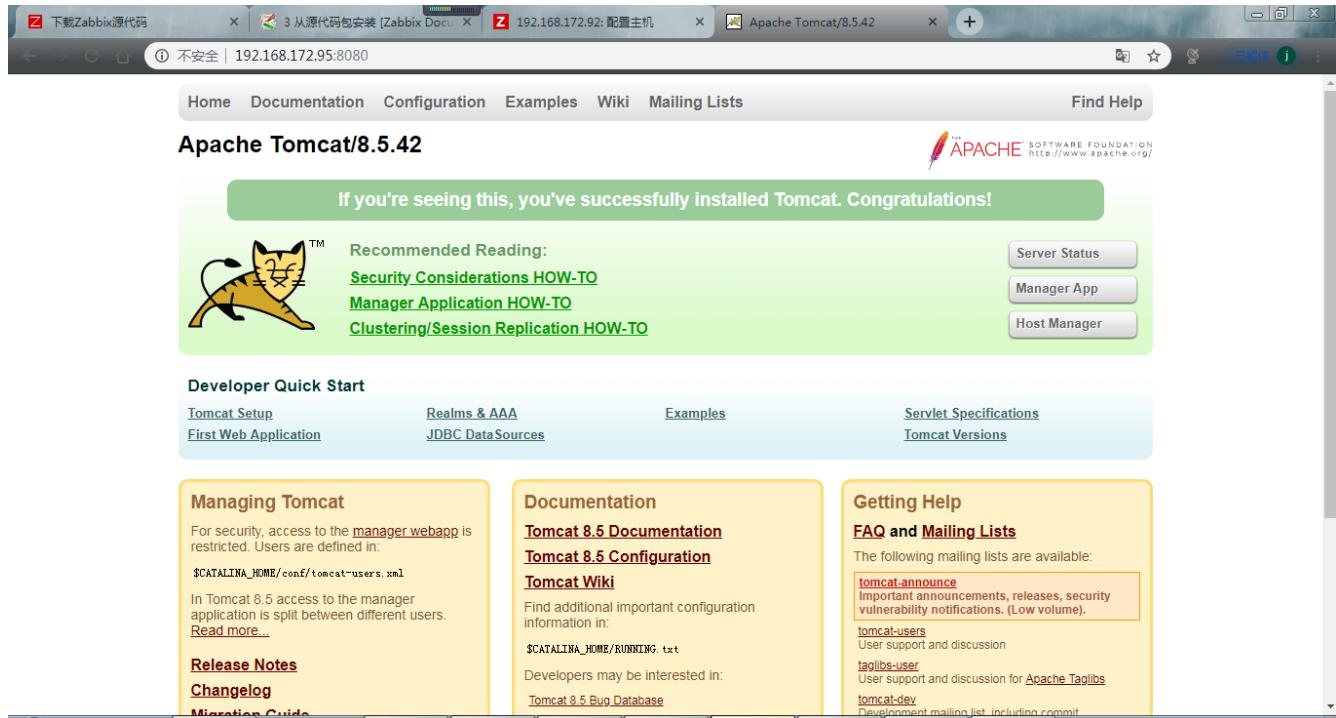
3.2 : tomcat监控

3.2.1 : 安装jdk

```
root@node5:~# apt-get install openjdk-8-jdk #测试环境直接apt，生产环境二进制安装
root@node5:~# java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-8u212-b03-0ubuntu1.18.04.1-b03)
OpenJDK 64-Bit Server VM (build 25.212-b03, mixed mode)
```

3.2.2 : 安装tomcat

```
root@node5:~# tar xvf /apps/apache-tomcat-8.5.42.tar.gz
root@node5:~# ln -sv /apps/apache-tomcat-8.5.42 /apps/tomcat
root@node5:~# /apps/tomcat/bin/catalina.sh start
```



3.2.3 : 配置tomcat监控参数

```
root@node5:~# vim /apps/tomcat/bin/catalina.sh #添加配置
CATALINA_OPTS="$CATALINA_OPTS
-Dcom.sun.management.jmxremote #启用远程监控JMX
-Dcom.sun.management.jmxremote.port=12345 #默认启动的JMX端口号，要和zabbix添加主机时候的端口一致即可
-Dcom.sun.management.jmxremote.authenticate=false #不使用用户名密码
-Dcom.sun.management.jmxremote.ssl=false #不使用ssl认证
-Djava.rmi.server.hostname=192.168.172.95" #tomcat主机自己的IP地址，不要写zabbix服务器的地址
```

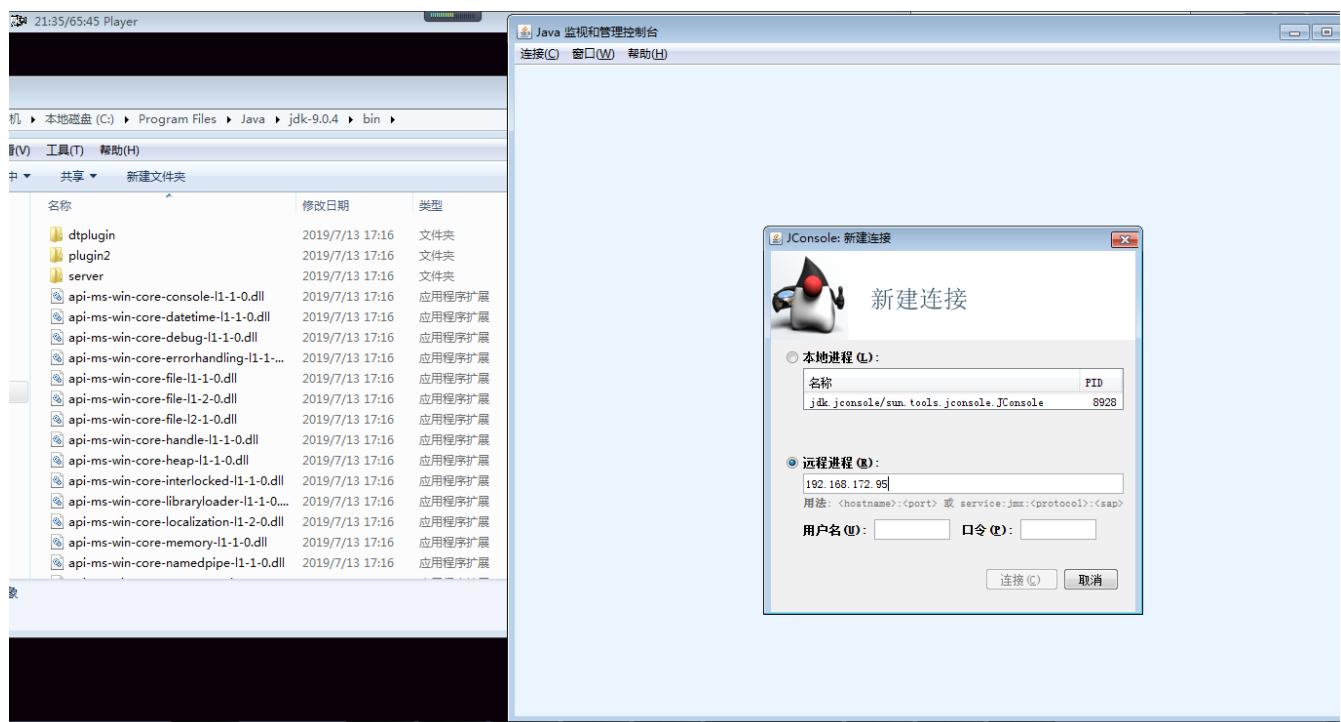
#重启服务后确认端口监听

```
root@node5:~# ss -nlt
State      Recv-Q      Send-Q      Local Address:Port          Peer Address:Port
LISTEN      0           50          *:12345                  *:*
```

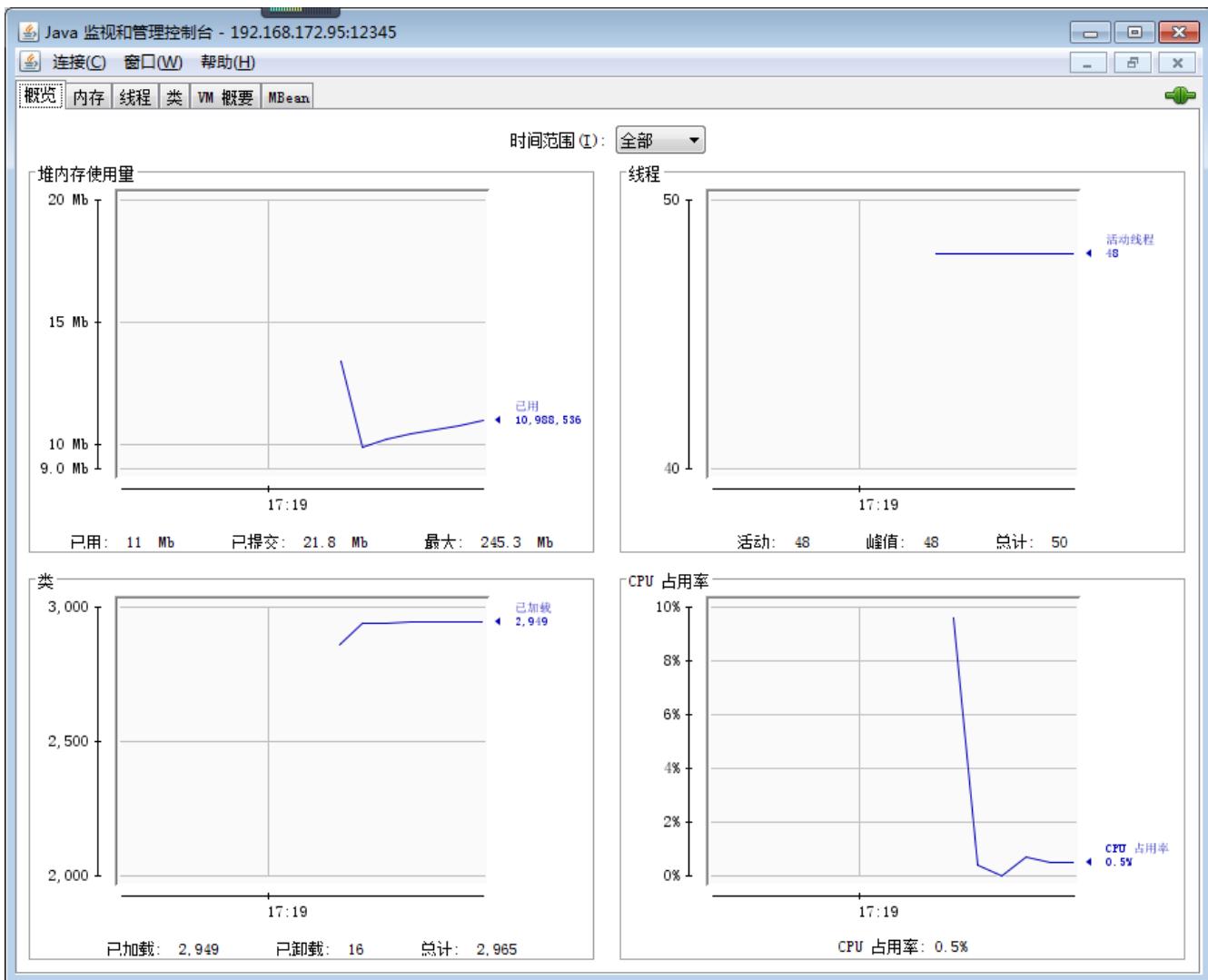
3.2.4 : windows端测试连接



安装完成后测试连接tomcat , tomcat IP : 端口号 (12345)



连接成功页面



3.2.5 : java gateway

```
#在编译安装的时候已经安装有java gateway，生产环境中可不安装在同一台主机
root@node2:~# vim /apps/zabbix_server/sbin/zabbix_java/settings.sh
- LISTEN_IP="0.0.0.0"
- LISTEN_PORT=10052
- PID_FILE="/tmp/zabbix_java.pid"
- START_POLLERS=10 #启动多少个进程轮询java，要和java 应用保持一定关系
- TIMEOUT=30 #此值最长为30s
root@node2:~# /apps/zabbix_server/sbin/zabbix_java/startup.sh
root@node2:~# ss -nlt
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
LISTEN      0          50          *:10052                  *:*
```

- 配置zabbix server调用java gateway

```
root@node2:~# vim /apps/zabbix_server/etc/zabbix_server.conf
- JavaGateway=192.168.172.92 #指定java gateway的地址
- JavaGatewayPort=10052 #指定java gateway的服务器监听端口，如果是默认端口可以不写
- StartJavaPollers=10 #启动多少个进程去轮询java gateway，要和java gateway的配置一致
- Timeout=30
```

3.2.6：将tomcat加入到监控

采集java数据使用JMX接口，不使用agent接口

The screenshot shows the Zabbix 4.0.10 web interface for configuring a host. The top navigation bar includes tabs for 'ZABBIX', '监测' (Monitoring), '资产记录' (Asset Record), '报表' (Report), '配置' (Configuration), and '管理' (Management). The main menu has sections for '主机群组' (Host Groups), '模板' (Templates), '主机' (Hosts), '维护' (Maintenance), '动作' (Actions), '关联项事件' (Associated Events), '自动发现' (Automatic Discovery), and '服务' (Services). The current view is under the '主机' tab.

The host configuration page displays the following details:

- 所有主机 / 192.168.172.95 已启用 [ZBX|SNMP|JMX|IPMI] 应用集 5 监控项 32 触发器 5 图形 4 自动发现规则 Web 场景**
- 主机**: 模板 IPMI 宏 主机资产记录 加密
- 主机名称**: 192.168.172.95
- 可见的名称**: (empty)
- 群组**: zabbix agent (selected) | 选择
- 至少存在一个接口。**
- agent代理程序的接口**:
 - IP地址: 192.168.172.95
 - DNS名称: (empty)
 - 连接到: (empty)
 - 端口: 12345
 - 默认: (radio button selected)
 - 添加**
- SNMP接口**:
 - 添加
- JMX接口**:
 - IP地址: 192.168.172.95
 - DNS: (empty)
 - 端口: 12345
 - 移除** (radio button selected)
 - 添加**
- IPMI接口**:
 - 添加
- 描述**: (empty)

Below the host configuration, there is another section for linking templates:

链接的模板:

- 名称: Template App Apache Tomcat JMX
- 动作:
 - 取消链接
 - 取消链接并清理

链接指示器:

- 在此输入搜索
- 选择

操作按钮:

- 更新
- 克隆
- 全克隆
- 删除
- 取消

The status bar at the bottom indicates the URL: 192.168.172.92/zabbix/hosts.php?form=update&hostid=10265&groupid=0... and the footer: Zabbix 4.0.10. © 2001–2019, Zabbix SIA.

配置完成后JMX为绿色显示成功

名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性	agent 加密	信息
192.168.172.94	应用集 10	监控项 51	触发器 21	图形 9	自动发现 2	Web监测 192.168.172.94:10050		Template OS Linux (Template App Zabbix Agent)	已启用	ZBX, SNMP, JMX, IPMI	无	
192.168.172.95	应用集 5	监控项 32	触发器 5	图形 4	自动发现 12345	Web监测 192.168.172.95:12345		Template App Apache Tomcat JMX	已启用	ZBX, SNMP, JMX, IPMI	无	
Zabbix server	应用集 11	监控项 76	触发器 46	图形 11	自动发现 2	Web监测 127.0.0.1:10050		Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX, SNMP, JMX, IPMI	无	

3.2.7 : 监控java排错方法

```
#测试能否获取到java 当前已经分配的 线程数 , cmdline-jmxclient-0.10.3.jar此包需要自行下载
#java -jar cmdline-jmxclient-0.10.3.jar - 192.168.172.95:12345 'catalina:name="http-nio-8080",type=ThreadPool' currentThreadCount
#java -jar cmdline-jmxclient-0.10.3.jar - 192.168.172.95:12345 'catalina:name="http-nio-8080",type=ThreadPool' maxThreads
```

3.3 : nginx监控

- 编译安装nginx

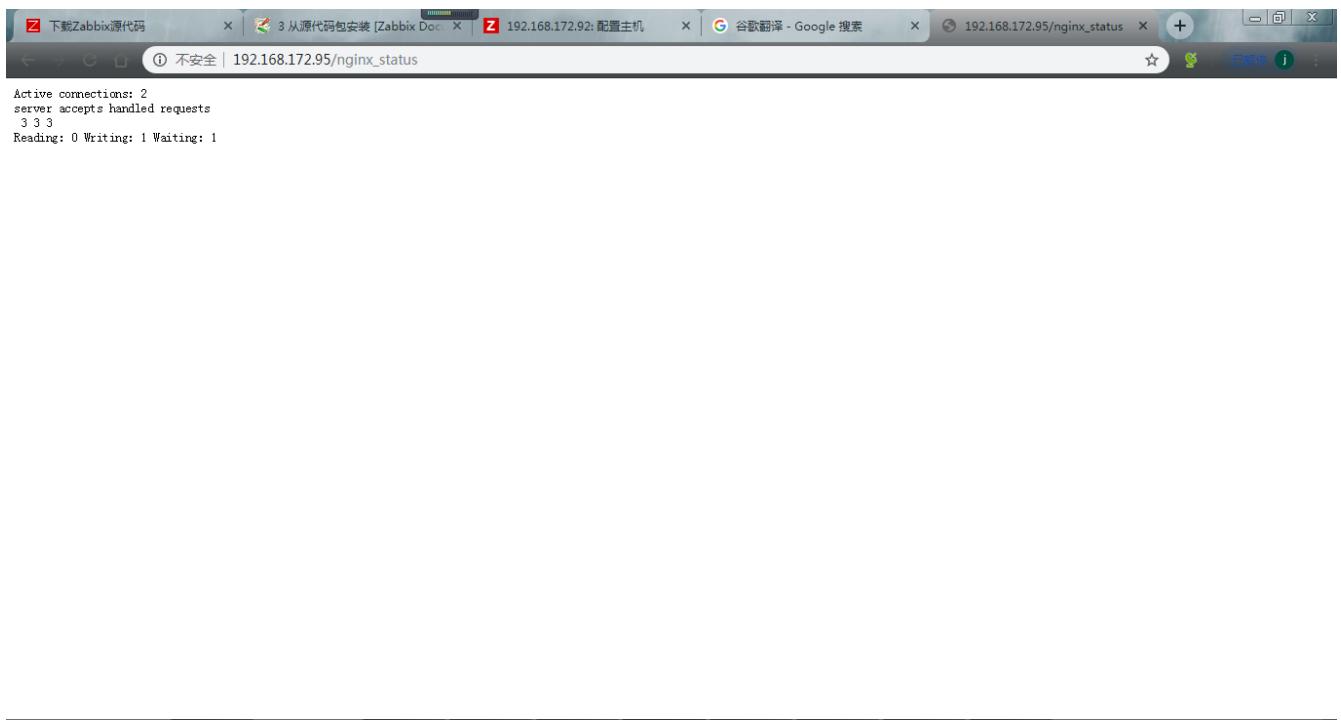
```
root@node5:/usr/local/src# wget http://nginx.org/download/nginx-1.16.0.tar.gz
root@node5:/usr/local/src# tar xvf nginx-1.16.0.tar.gz
root@node5:/usr/local/src# cd nginx-1.16.0/
root@node5:/usr/local/src/nginx-1.16.0# ./configure --prefix=/apps/nginx --with-http_stub_status_module
#编译报错解决
https://blog.csdn.net/hu_feng903/article/details/80297821
```

- 状态页测试

```

root@node5:~# vim /apps/nginx/conf/nginx.conf
location /nginx_status {
    stub_status;
}
root@node5:~# /apps/nginx/sbin/nginx
#启动服务后登陆状态页测试
root@node5:~# curl http://192.168.172.95/nginx_status
Active connections: 1 #当前处于活动状态的客户端连接数，包括连接等待空闲连接数
server accepts handled requests
5 5 5
Reading: 0 Writing: 1 Waiting: 0

```



- 编辑取值脚本

```

root@node5:~# vim /etc/zabbix/zabbix_agentd.d/check_nginx.sh
#!/bin/bash
#
host=${2:-'127.0.0.1'}
port=${3:-'80'}
page=${4:-'nginx_status'}
info=$(curl --connect-timeout 5 -s http://${host}:${port}/${page} 2>/dev/null)
code=$(curl --connect-timeout 5 -o /dev/null -s -w %{http_code}
http://${host}:${port}/${page})
proc=$(pgrep nginx | wc -l)

case "$1" in
status)
    echo "$code $proc" | awk '{code=$1}{proc=$2}END{if(code == "200" && proc != 0)
{printf("%d\n",1)}else{printf("%d\n",0)}}'
;;
active)
    echo "$info" | awk '/^Active/{var=$NF}END{if(var~/^[0-9]+$/)

```

```

{printf("%d\n",var){else{printf("%d\n",0)}}'
;;
reading)
echo "$info" | awk '/Reading/ {print $2}'
;;
writing)
echo "$info" | awk '/writing/ {print $4}'
;;
waiting)
echo "$info" | awk '/waiting/ {print $6}'
;;
accepts)
echo "$info" | awk 'NR==3 {print $1}'
;;
handled)
echo "$info" | awk 'NR==3 {print $2}'
;;
requests)
echo "$info" | awk 'NR==3 {print $3}'
;;
restimes)
echo "$info" | awk 'BEGIN{OFMT=".3f"} NR==3 {print $4/$3}'
;;
*)
echo "ZBX_NOTSUPPORTED"
;;
esac

root@node5:/etc/zabbix/zabbix_agentd.conf.d# chmod +x check_nginx.sh
root@node5:/etc/zabbix/zabbix_agentd.conf.d# chown -R zabbix.zabbix check_nginx.sh
root@node5:/etc/zabbix/zabbix_agentd.conf.d# vim ..../zabbix_agentd.conf
UserParameter=nginx.status[*],/etc/zabbix/zabbix_agentd.d/check_nginx.sh $1 $2

```

3.3.1：创建自定义模板

The screenshot shows the Zabbix 4.0.10 interface for creating a new template. The top navigation bar includes tabs for 'ZABBIX', '监测' (Monitoring), '资产记录' (Assets), '报表' (Reports), '配置' (Configuration), and '管理' (Management). The main menu has '模板' (Template) selected. The page title is '模板'. The form fields are as follows:

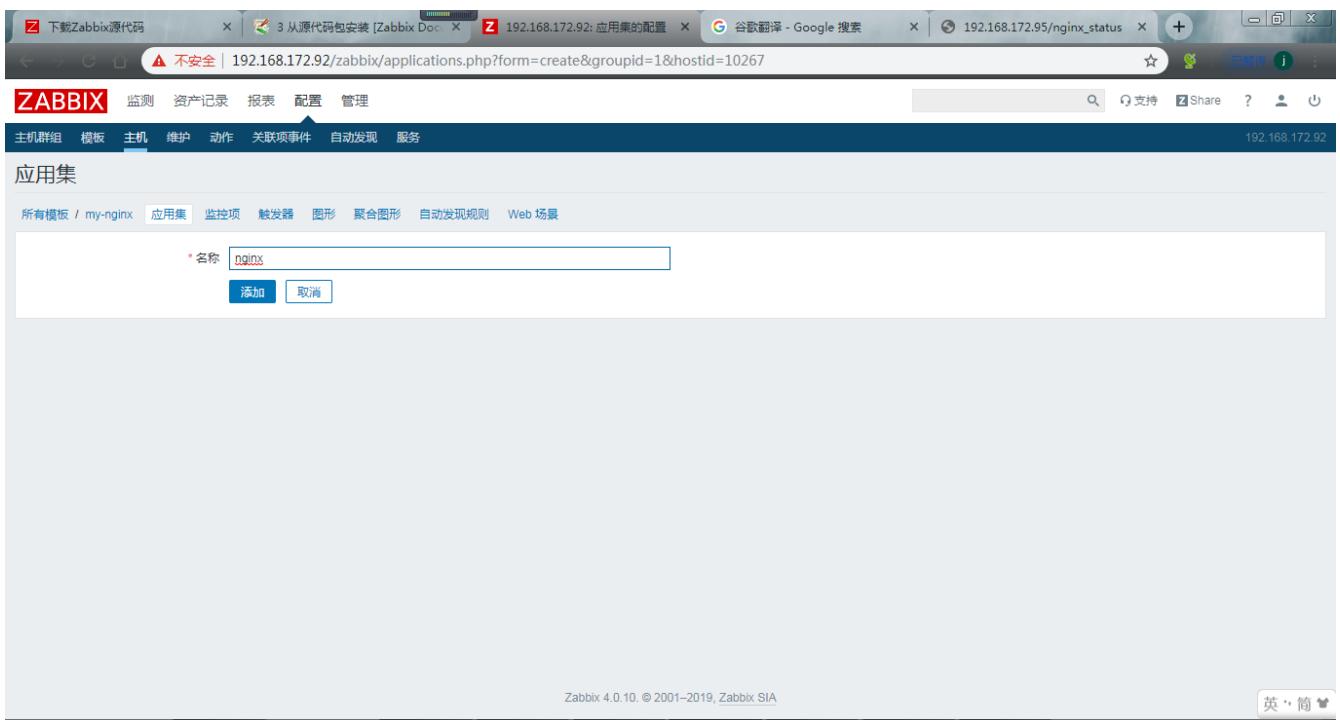
- * 模版名称: my-nginx
- 可见的名称: (empty)
- * 群组: Templates (selected)
- 描述: (empty)

At the bottom are '添加' (Add) and '取消' (Cancel) buttons.

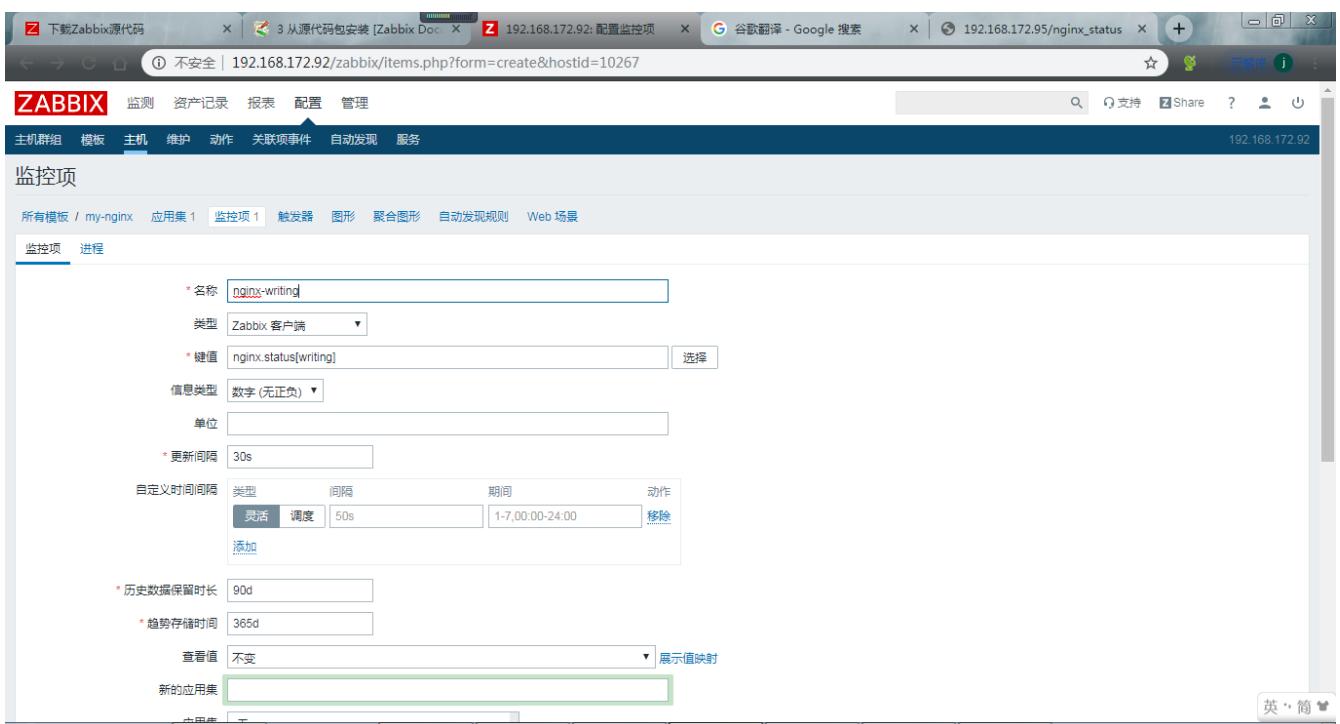
The screenshot shows the Zabbix 4.0.10 interface displaying a list of templates. A red arrow points to the 'my-nginx' entry in the list, which is highlighted with a red border. The list includes the following entries:

名称	应用集	监控项	触发器	图形	聚合图形	自动发现	Web监测	链接的模板	已链接到
my-nginx	应用集 5	监控项 32	触发器 5	图形 4	聚合图形	自动发现	Web监测		192.168.172.95
Template App Apache Tomcat JMX	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		
Template App FTP Service	应用集 3	监控项 55	触发器 26	图形 11	聚合图形	自动发现	Web监测		
Template App Generic Java JMX	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		
Template App HTTP Service	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		192.168.172.95-nginx
Template App HTTPS Service	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		
Template App IMAP Service	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		
Template App LDAP Service	应用集 1	监控项 1	触发器 1	图形	聚合图形	自动发现	Web监测		

- 创建应用集



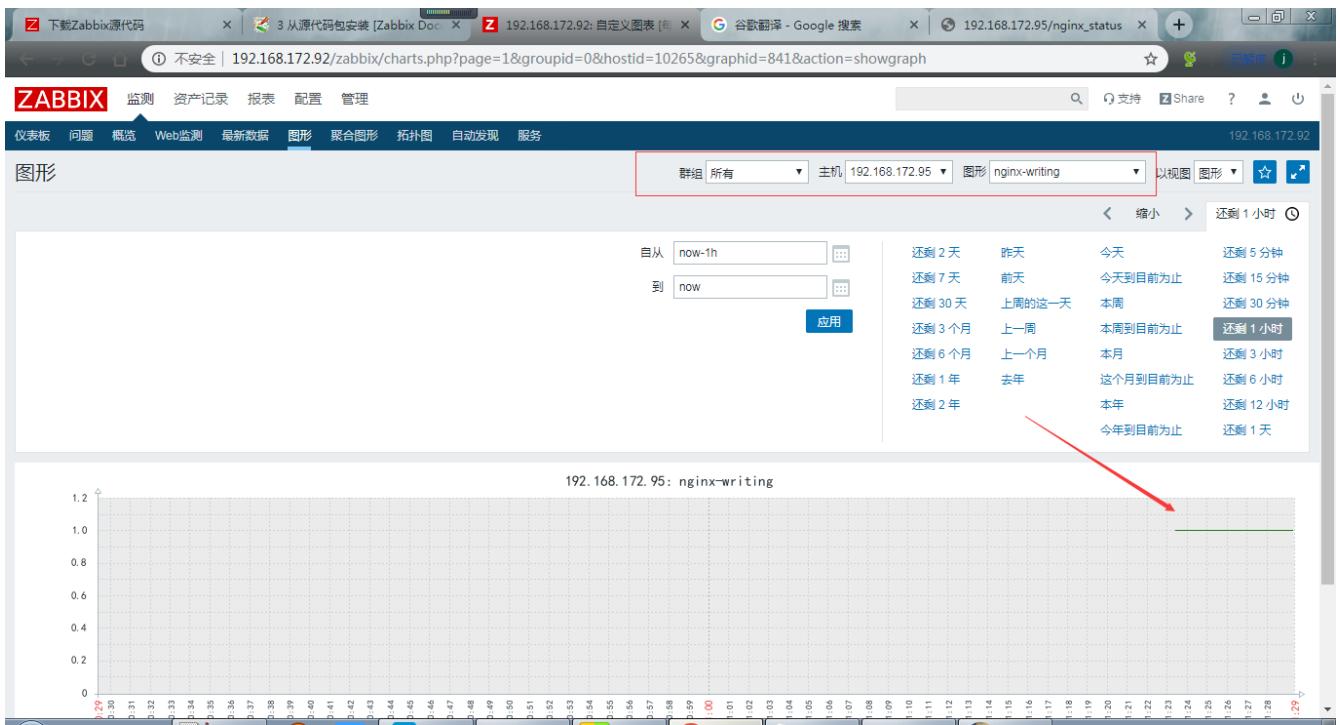
- 创建监控项



- 创建图形



- 将模板添加至192.168.172.95，查看监控图形

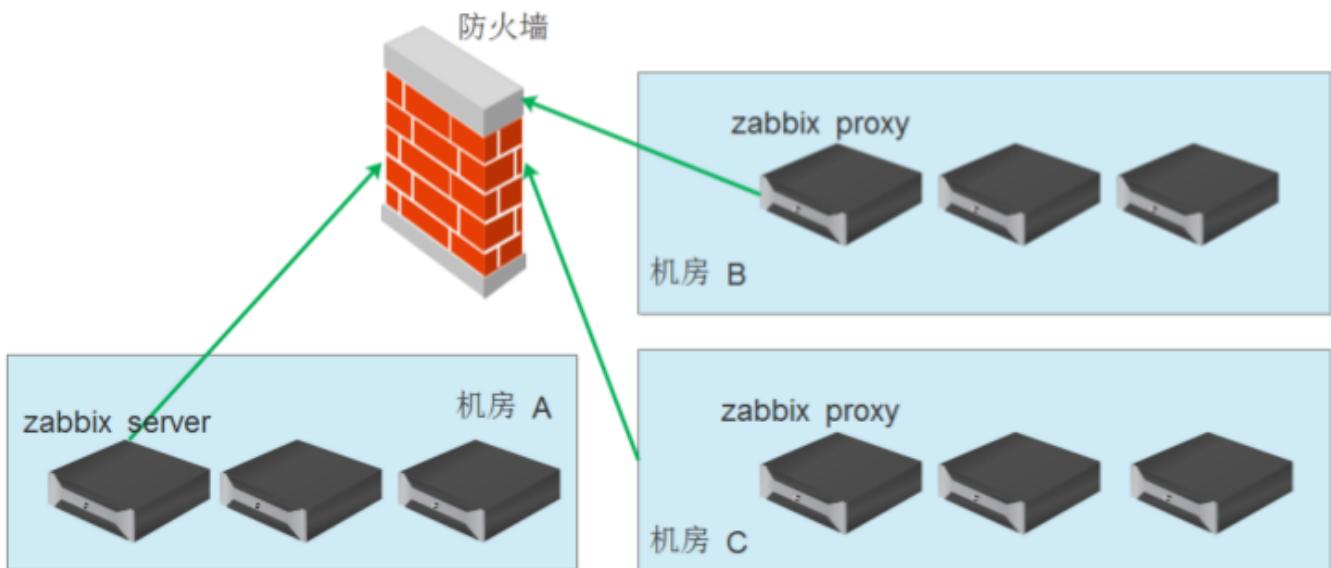


- 测试成功后再将其他监控项添加即可

四、Zabbix Proxy

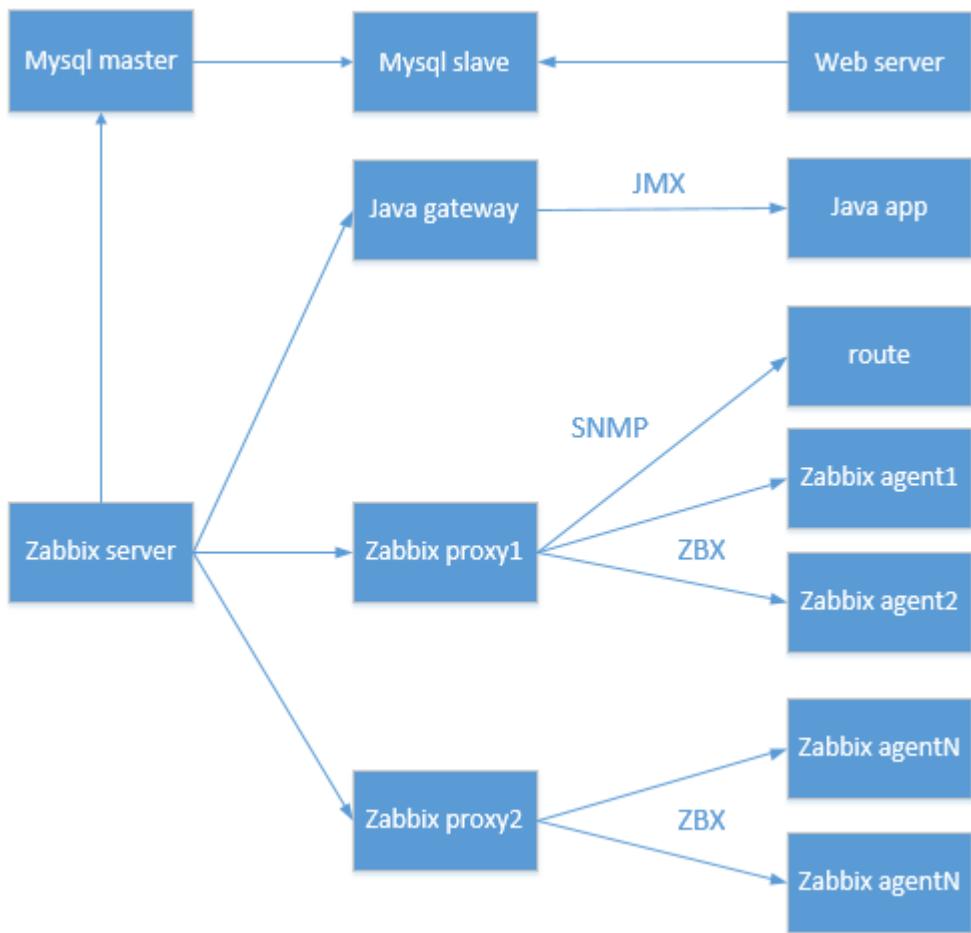
zabbix 是一个分布式的监控系统，支持通过代理服务器zabbix proxy收集zabbix agent的数据，然后把收集的数据保存在本地数据库并发送给zabbix server进行统一存储和展示

- 工作模式



功能	zabbix proxy	zabbix server
轻量级	是	相对重量级
图形	无	带图形控制界面
可以独立工作	是，可以独立采集数据并存储	是，集数据采集，存储，分析，展示于一体
易维护	是，配置完成后基本无需管理	维护不难
独立数据库	保留少量最近数据	保留指定时间内的所有数据
报警通知	否，代理服务器不发送邮件	支持邮件，短信告警机制

- zabbix proxy代理图



4.1 : 主动与被动

主动与被动是对于zabbix agent来说的工作模式

- 被动模式

由zabbix server向zabbix agent发出指令获取数据，即zabbix agent被动的去获取数据并返回给zabbix server，zabbix server周期性的向agent索取数据，这种模式的最大问题就是会加大zabbix server的工作量，在数百台服务器的环境下zabbix server不能及时获取到最新数据，但这也是默认的工作方式

- 主动模式

zabbix agent主动采集数据并返回给zabbix server，不再需要zabbix server进行干预，因此主动模式在一定程度上可减轻zabbix server的压力

4.2 : 编译安装zabbix proxy

IP	server	hostname
192.168.172.91	zabbix server	node1
192.168.172.92	mariadb-server	node2
192.168.172.93	zabbix proxy	node3
192.168.172.94	zabbix agent active	node4

- 安装依赖包

```
root@node3:~# apt-get install fontconfig-config fonts-dejavu-core fping libapache2-mod-php libapache2-mod-php7.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libfontconfig1 libgd3 libiksemel3 libjbig0 libjpeg-turbo8 libjpeg8 liblua5.2-0 libodbc1 libopenipmi0 libsensors4 libsnmp-base libsnmp30 libsodium23 libssh2-1 libtiff5 libwebp6 libxpm4 php-bcmath php-common php-gd php-ldap php-mbstring php-mysql php-xml php7.2-bcmath php7.2-cli php7.2-common php7.2-gd php7.2-json php7.2-ldap php7.2-mbstring php7.2-mysql php7.2-opcache php7.2-readline php7.2-xml snmpd ssl-cert ttf-dejavu-core libmysqlclient-dev libxml2-dev libxml2 snmp libsnmp-dev libevent-dev openjdk-8-jdk curl libcurl4-openssl-dev -y
```

- 准备数据库

```
mysql
create database zabbix_proxy character set utf8 collate utf8_bin;
grant all privileges on zabbix_proxy.* to zabbix_proxy@'192.168.172.%' identified by
'centos';
```

- 编译安装

```
root@node3:/usr/local/src# useradd zabbix -s /sbin/nologin
root@node3:/usr/local/src# tar xvf zabbix-4.0.10.tar.gz
root@node3:/usr/local/src/zabbix-4.0.10# ./configure --prefix=/usr/local/zabbix --enable-proxy --enable-agent --with-mysql --with-net-snmp --with-libcurl --with-libxml2 --enable-java
root@node3:/usr/local/src/zabbix-4.0.10# make && make install
```

- 导入数据库

```
root@node3:/usr/local/src/zabbix-4.0.10# mysql -uzabbix_proxy -pcentos -h192.168.172.92
zabbix_proxy < database/mysql/schema.sql
```

4.3 : proxy主动模式

4.3.1 : proxy主动模式配置文件

```
root@node3:/apps/zabbix_agent/zabbix# grep "^[a-Z]" etc/zabbix_proxy.conf
ProxyMode=0
Server=192.168.172.91
ServerPort=10051
Hostname=proxy-active
ListenPort=10051
SourceIP=192.168.172.93
LogType=file
LogFile=/apps/zabbix_agent/zabbix/logs/zabbix_proxy.log
LogFileSize=0
EnableRemoteCommands=1
LogRemoteCommands=1
PidFile=/apps/zabbix_agent/zabbix/run/zabbix_proxy.pid
```

```

SocketDir=/var/run/zabbix
DBHost=192.168.172.92
DBName=zabbix_proxy_active
DBUser=active
DBPassword=centos
DBPort=3306
ProxyLocalBuffer=3
ProxyOfflineBuffer=24
HeartbeatFrequency=60
ConfigFrequency=60
DataSenderFrequency=60
StartPollers=10
JavaGateway=192.168.172.91
JavaGatewayPort=10052
StartJavaPollers=10
CacheSize=128M
HistoryCacheSize=128M
Timeout=30
LogSlowQueries=3000

root@node3:~# mkdir /apps/zabbix_agent/zabbix/logs
root@node3:~# mkdir /apps/zabbix_agent/zabbix/run
root@node3:~# chown -R zabbix.zabbix /apps/zabbix_agent
root@node3:~# mkdir /var/run/zabbix
root@node3:~# chown -R zabbix.zabbix /var/run/zabbix

```

4.3.2 : 创建数据库

```

#在node2上创建proxy数据库
MariaDB [(none)]> create database zabbix_proxy_active character set utf8 collate utf8_bin;
MariaDB [(none)]> grant all privileges on zabbix_proxy_active.* to active@'192.168.172.%'
identified by 'centos';

#导入数据库
root@node3:/usr/local/src/zabbix-4.0.10/database/mysql# pwd
/usr/local/src/zabbix-4.0.10/database/mysql
root@node3:/usr/local/src/zabbix-4.0.10/database/mysql# mysql -uactive -pcentos -
h192.168.172.92 zabbix_proxy_active < schema.sql

#启动服务
root@node3:~# cd /apps/zabbix_agent/zabbix/
root@node3:/apps/zabbix_agent/zabbix# sbin/zabbix_proxy
root@node3:/apps/zabbix_agent/zabbix# ss -nlt
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
LISTEN      0          128          0.0.0.0:10051          0.0.0.0:*
```

4.3.3 : 添加主动代理

ZABBIX 监测 资产记录 报表 配置 管理

agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列

名称: | 模式: 任何 主动式 被动式

应用 重设

模式 加密 压缩 最近出现(持续) 主机计数 监控项数量 要求的性能(vps) 主机

未发现数据 显示 已自动发现的 0 中的 0

0 选择 启用主机 禁用主机 删除

Zabbix 4.0.10. © 2001–2019, Zabbix SIA

ZABBIX 监测 资产记录 报表 配置 管理

agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列

agent代理程序

* agent代理程序名称: proxy-active (此名称必须等于zabbix proxy配置中的hostname名)

系统代理程序模式: 主动式 被动式

代理地址: 192.168.172.93

描述:

添加 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA

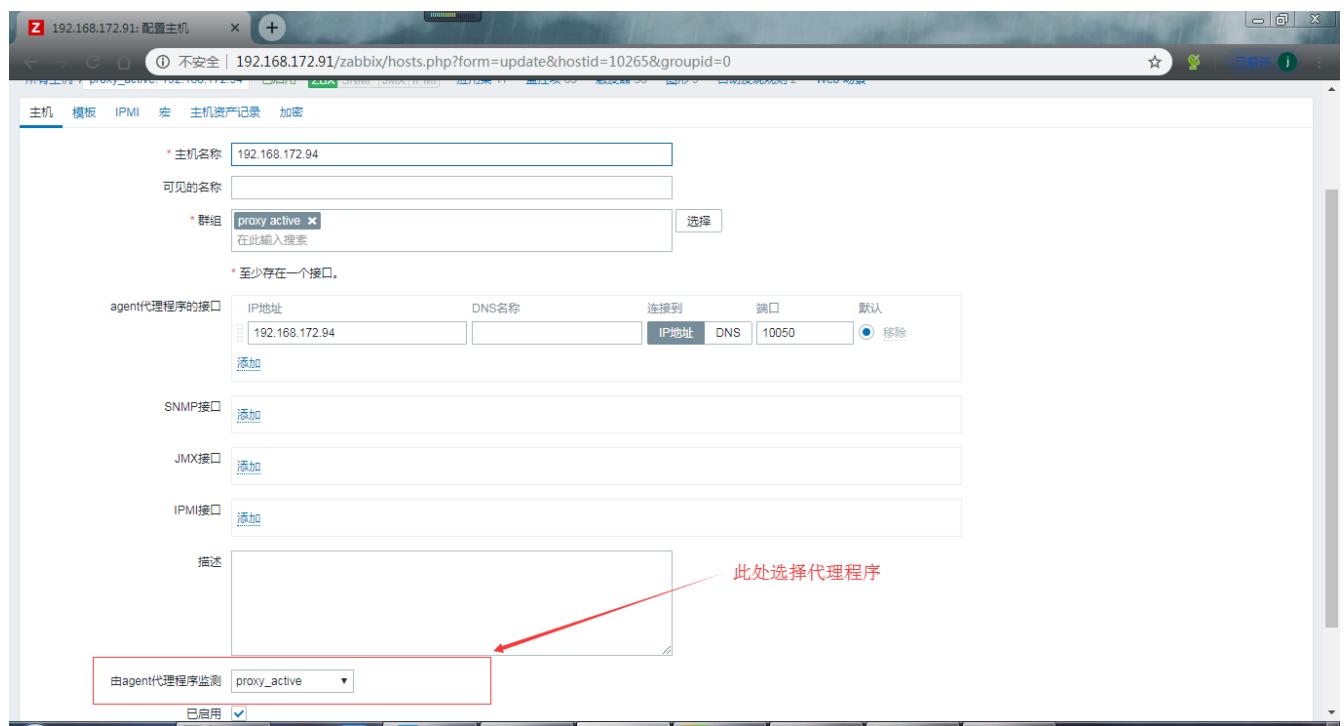
4.3.4 : 安装zabbix agent

```
root@node3:~# wget https://repo.zabbix.com/zabbix/4.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_4.0-2+bionic_all.deb
root@node3:~# dpkg -i zabbix-release_4.0-2+bionic_all.deb
root@node3:~# apt update
root@node3:~# apt-get install zabbix-agent
```

- 修改配置文件

```
root@node4:~# grep "^[a-zA-Z]" /etc/zabbix/zabbix_agentd.conf
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
EnableRemoteCommands=1
LogRemoteCommands=1
Server=192.168.172.93
ListenPort=10050
ServerActive=192.168.172.93
Hostname=192.168.172.94
Include=/etc/zabbix/zabbix_agentd.d/*.conf
UnsafeUserParameters=0
```

4.3.5 : server端添加监控



ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

主机

名称: [] DNS: []
 模板: 在此输入搜索 [选择] IP地址: []
 要监控什么: [任何] 服务器 agent代理程序
 agent代理程序: [选择]

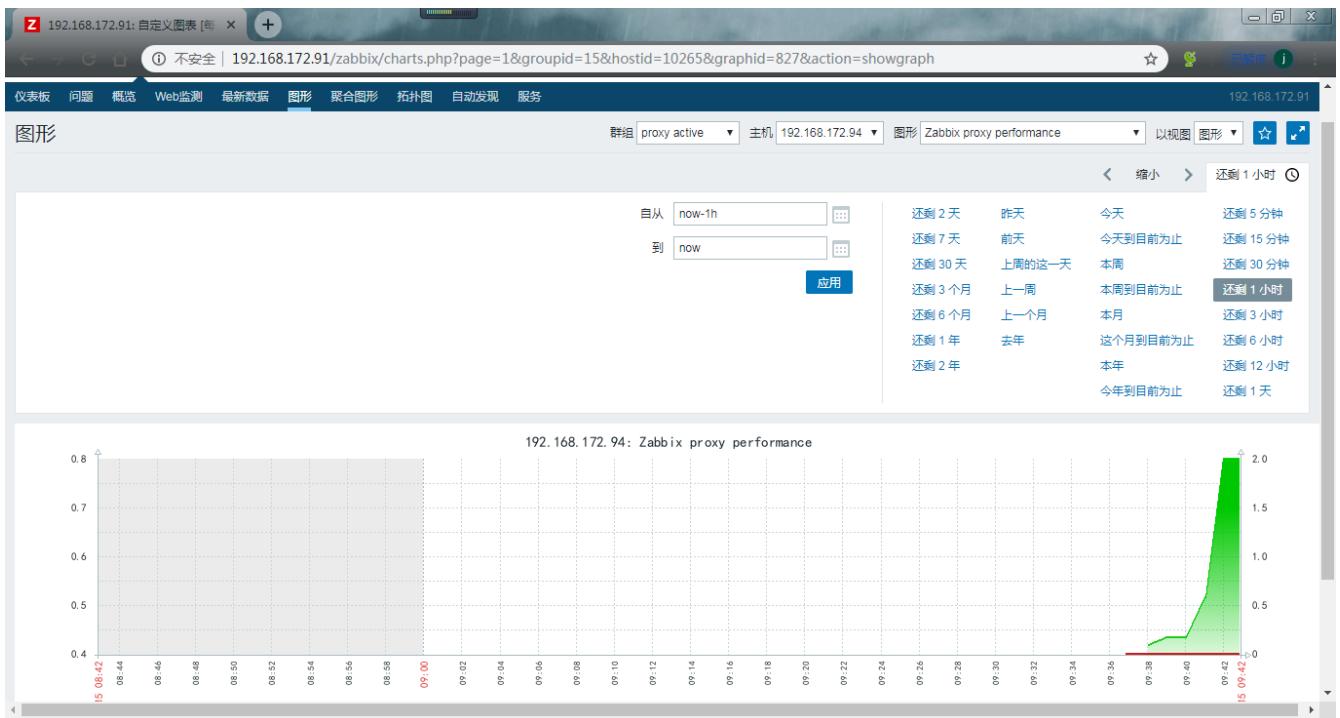
应用 重设

名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性	agent 加密	信息
proxy_active: 192.168.172.94	应用集 11	监控项 65	触发器 38	图形 9	自动发现 2	Web监测 192.168.172.94:10050		Template App Zabbix Proxy, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX IPMI	无	
Zabbix server	应用集 11	监控项 93	触发器 52	图形 15	自动发现 2	Web监测 192.168.172.91:10050		Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX IPMI	无	

显示 已自动发现的 2 中的2

0 选择 启用 禁用 导出 批量更新 删除

Zabbix 4.0.10. © 2001–2019, Zabbix SIA



4.4：监控TCP连接数

4.4.1：创建获取数据的脚本

```
root@node4:~# vim /etc/zabbix/zabbix_agentd.d/tcp_conn.sh
#!/bin/bash
tcp_conn_status(){
    TCP_STAT=$1
    ss -ant | awk 'NR>1 {++s[$1]} END {for(k in s) print k,s[k]}' > /tmp/tcp_conn.txt
    TCP_NUM=$(grep "$TCP_STAT" /tmp/tcp_conn.txt | cut -d ' ' -f2)
    if [ -z $TCP_NUM ];then
        TCP_NUM=0
    fi
}
```

```

        fi
        echo $TCP_NUM
    }

main(){
    case $1 in
        tcp_status)
            tcp_conn_status $2;
            ;;
    esac
}

main $1 $2

root@node4:~# chmod +x /etc/zabbix/zabbix_agentd.d/tcp_conn.sh

```

4.4.2 : 创建conf文件引用脚本

```

root@node4:~# vim /etc/zabbix/zabbix_agentd.d/tcp_conn.conf
UserParameter=linux_status[*],/etc/zabbix/zabbix_agentd.conf.d/tcp_conn.sh "$1" "$2"

root@node4:~# vim /etc/zabbix/zabbix_agentd.conf
include=/etc/zabbix/zabbix_agentd.d/*.conf

#重启agent服务
root@node4:~# systemctl restart zabbix-agent.service

#为zabbix用户授权
root@node4:~# vim /etc/sudoers
root    ALL=(ALL:ALL) ALL
zabbix  ALL=(ALL) NOPASSWD:ALL

```

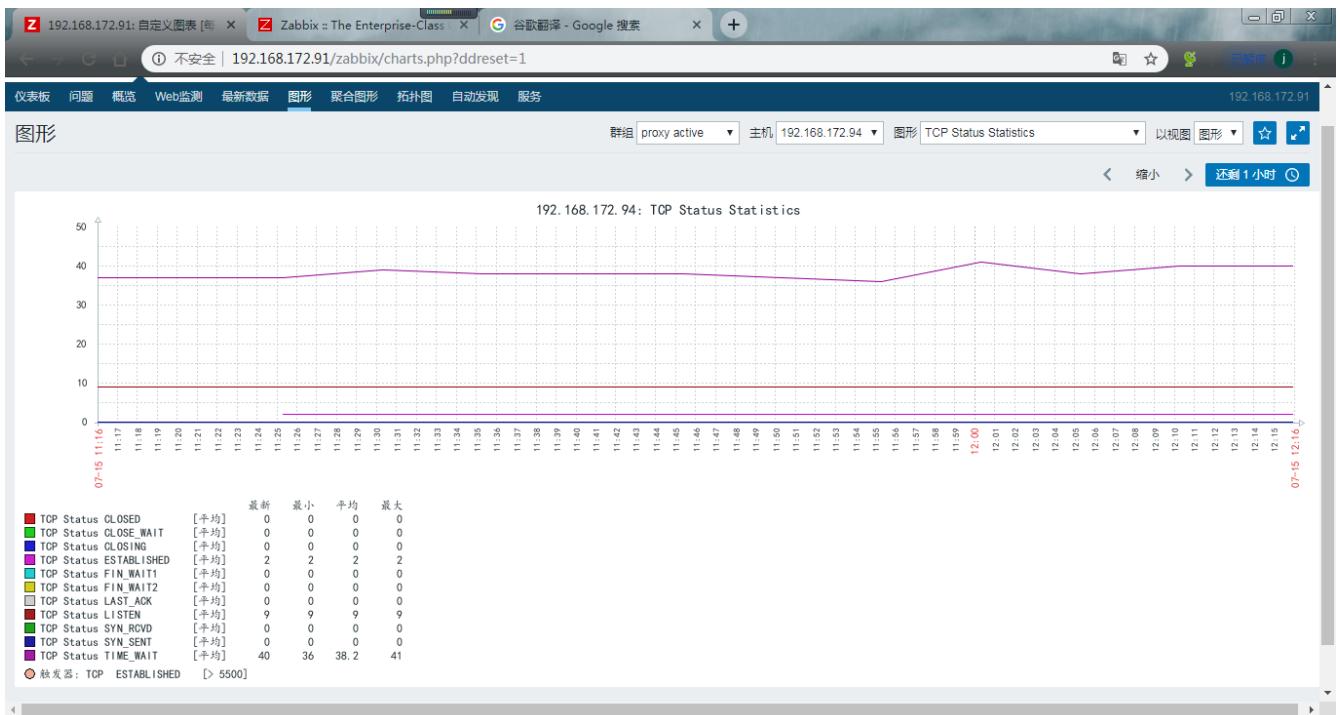
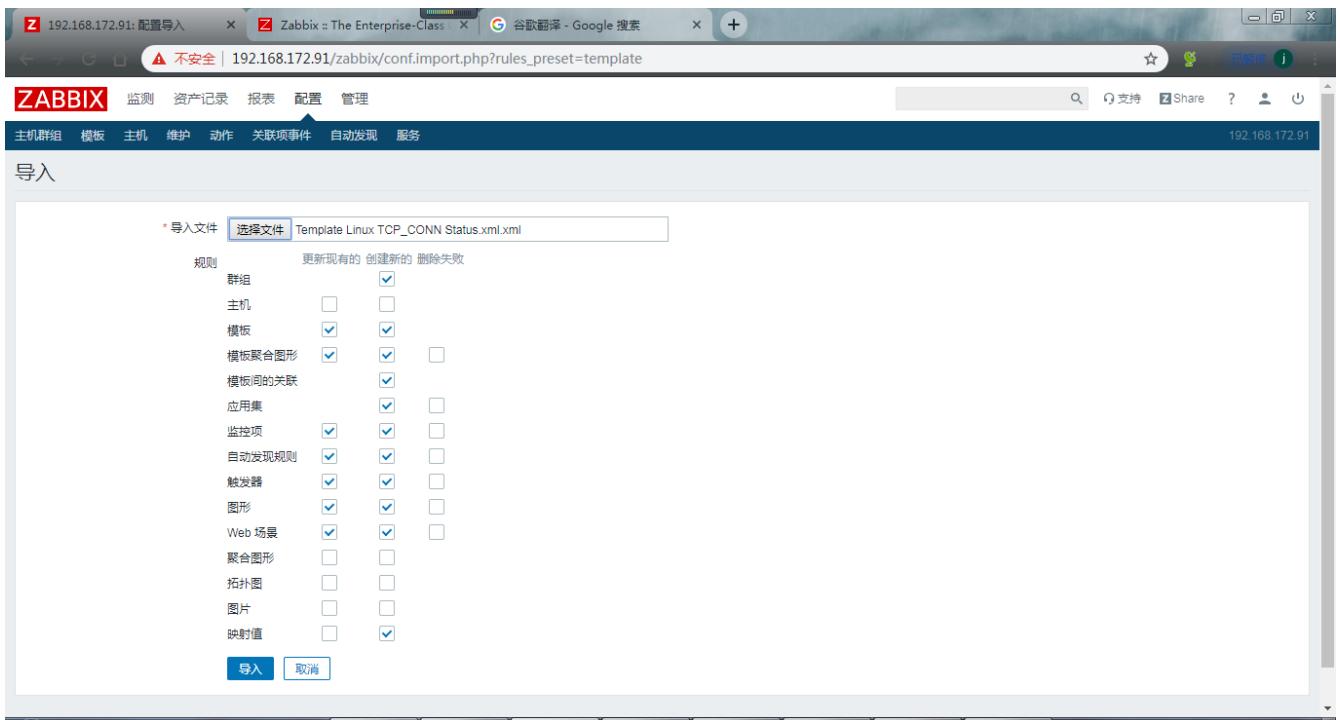
- server端测试连接数据

```

root@node1:~# /apps/zabbix_server/bin/zabbix_get -s 192.168.172.94 -p 10050 -k
"linux_status["tcp_status","LISTEN"]"
9

```

4.4.3 : 导入并关联监控模板测试



4.5 : 监控memcached

4.5.1 : 创建获取数据的脚本

```
root@node4:~# vim /etc/zabbix/zabbix_agentd.d/memcached_status.sh
#!/bin/bash
memcached_status(){
M_PORT=$1
M_COMMAND=$2
echo -e "stats\nclose" | nc 127.0.0.1 "$M_PORT" | grep "STAT $M_COMMAND " | awk '{print
$3}'"
}
```

```
main(){
    case $1 in
        memcached_status)
            memcached_status $2 $3
            ;;
    esac
}

main $1 $2 $3

root@node4:~# chmod +x /etc/zabbix/zabbix_agentd.d/memcached_status.sh
```

4.5.2 : 创建conf文件引用脚本

```
root@node4:~# vim /etc/zabbix/zabbix_agentd.d/memcached_status.conf
UserParameter=memcache_status[*],/etc/zabbix/zabbix_agentd.d/memcached_status.sh "$1" "$2"
"$3"

#测试脚本执行
root@node4:/etc/zabbix/zabbix_agentd.d# ./memcached_status.sh memcached_status 11211
curr_connections
1

#重启zabbix agent服务
root@node4:~# systemctl restart zabbix-agent.service
```

- server端验证

```
root@node1:~# /apps/zabbix_server/bin/zabbix_get -s 192.168.172.94 -p 10050 -k
memcache_status["memcached_status",11211,"curr_connections"]
1
root@node1:~# /apps/zabbix_server/bin/zabbix_get -s 192.168.172.94 -p 10050 -k
memcache_status["memcached_status",11211,"total_connections"]
9
```

4.5.3 : 创建监控模板

The screenshot shows the Zabbix configuration interface for creating a new template. The URL is <http://192.168.172.91/zabbix/templates.php?form=update&templateid=10267&groupid=0>. The page title is "ZABBIX 配置模板". The main navigation bar includes "监测" (Monitoring), "资产记录" (Assets), "报表" (Reports), "配置" (Configuration), and "管理" (Management). The sub-navigation bar under "模板" (Templates) includes "所有模板 / memcache-status", "应用集 1", "监控项 1", "触发器 1", "图形 1", "聚合图形", "自动发现规则", and "Web 场景". The template creation form has the following fields:

- 模板名称:
- 可见的名称:
- 群组:
- 描述:

At the bottom are buttons: **更新**, **克隆**, **全克隆**, **删除**, **删除并清除**, and **取消**.

- 创建监控项

The screenshot shows the Zabbix configuration interface for creating a new monitoring item. The URL is <http://192.168.172.91/zabbix/items.php?form=update&hostid=10267&itemid=28786>. The page title is "ZABBIX 配置监控项". The main navigation bar includes "监测" (Monitoring), "资产记录" (Assets), "报表" (Reports), "配置" (Configuration), and "管理" (Management). The sub-navigation bar under "监控项" (Monitoring Items) includes "所有模板 / memcache-status", "应用集 1", "监控项 1", "触发器 1", "图形 1", "聚合图形", "自动发现规则", and "Web 场景". The monitoring item creation form has the following fields:

- 名称:
- 类型:
- 键值:
- 信息类型:
- 单位:
- 更新间隔:
- 历史数据保留时长:
- 趋势存储时间:
- 查看值:
- 新的应用集:
- 填入主机资产记录栏位:
- 描述:

At the bottom are buttons: **已启用** (checked), **更新**, **克隆**, **删除**, and **取消**.

- 创建图形项

ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

图形

所有模板 / memcache-status 应用集 1 监控项 1 触发器 1 图形 1 聚合图形 自动发现规则 Web 场景

图形 预览

* 名称 总计连接数

* 宽 900

* 高 200

图形类别 正常

查看图例 ✓

查看工作时间 ✓

查看触发器 ✓

百分比线(左) □

百分比线(右) □

纵轴Y最小值MIN 可计算的

纵轴最大值 可计算的

* 监控项

名称	功能	绘图风格	纵轴Y侧	颜色	动作
1: memcache-status: 总计连接数	平均	线	左侧	1A7C11	移除

添加

更新 克隆 删除 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA

- 创建触发器

ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

触发器

所有模板 / memcache-status 应用集 1 监控项 1 触发器 1 图形 1 聚合图形 自动发现规则 Web 场景

触发器 依赖关系

* 名称 memcached 连接数

严重性 未分类 | 信息 | 警告 | 一般严重 | **严重** | 灾难

* 表达式 {memcache-status:memcache_status["memcached_status",11211,"curr_connections"].last({#3,60})>1000}

表达式构造器

事件成功迭代 表达式 | 恢复表达式 | 无

问题事件生成模式 单个 | 多重

事件成功关闭 所有问题 | 所有问题如果标签值匹配

标记 标记 值 移除

添加

允许手动关闭 ✓

URL

描述

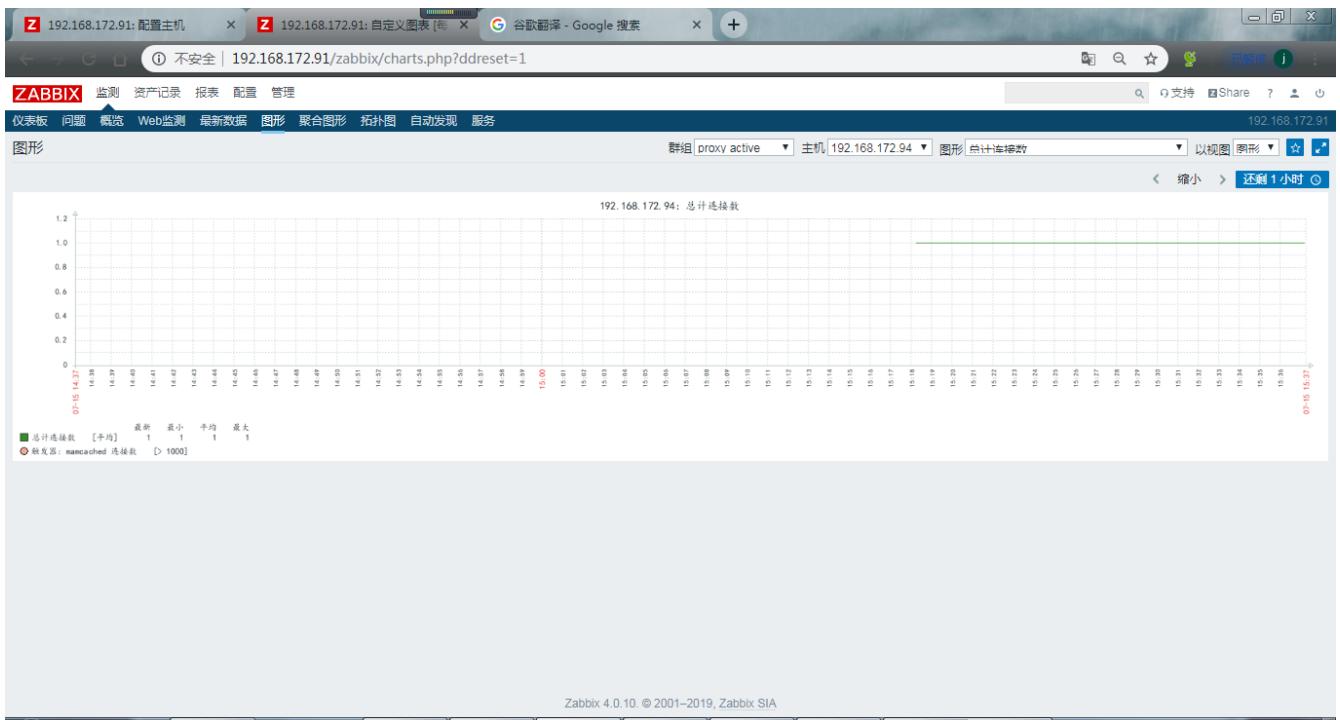
已启用 ✓

更新 克隆 删除 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA

4.5.4 : 关联主机并验证数据

将模板关联至memcache服务器



4.6：监控redis

4.6.1：创建获取数据的脚本

```

root@node4:/etc/zabbix/zabbix_agentd.d# vim redis.sh
#!/bin/bash
redis_status(){
    R_PORT=$1
    R_COMMAND=$2
    (echo -en "INFO \r\n";sleep 1;) | ncat 127.0.0.1 "$R_PORT" >
/tmp/redis_"$R_PORT".tmp #ubuntu使用ncat , centos使用nc , 安装nmap包
    REDIS_STAT_VALUE=$(grep ""$R_COMMAND": " /tmp/redis_"$R_PORT".tmp | cut -d ':' -f2)
    echo $REDIS_STAT_VALUE
}

help(){
    echo "${0} + redis_status + PORT + COMMAND"
}

main(){
    case $1 in
        redis_status)
            redis_status $2 $3
            ;;
        *)
            help
            ;;
        esac
}
main $1 $2 $3

root@node4:/etc/zabbix/zabbix_agentd.d# chmod +x redis.sh

```

```
root@node4:/etc/zabbix/zabbix_agentd.d# apt install nmap
#测试脚本执行
root@node4:/etc/zabbix/zabbix_agentd.d# bash redis.sh redis_status 6379 connected_clients
2
root@node4:/etc/zabbix/zabbix_agentd.d# bash redis.sh redis_status 6379 used_cpu_sys
0.23
```

4.6.2 : 创建conf文件引用脚本

```
root@node4:/etc/zabbix/zabbix_agentd.d# vim redis.conf
UserParameter=redis_status[*],/etc/zabbix/zabbix_agentd.d/redis.sh $1 $2 $3
#重启服务
root@node4:/etc/zabbix/zabbix_agentd.d# systemctl restart zabbix-agent.service
```

- server端测试连接

```
root@node1:/apps/zabbix_server/bin# ./zabbix_get -s 192.168.172.94 -p 10050 -k
"redis_status[redis_status,6379,connected_clients]"
1
root@node1:/apps/zabbix_server/bin# ./zabbix_get -s 192.168.172.94 -p 10050 -k
"redis_status[redis_status,6379,used_cpu_sys]"
1.16
```

4.6.3 : 创建监控模板

The screenshot shows the Zabbix configuration interface for creating a new template. The URL is http://192.168.172.91/zabbix/templates.php?group_id=0&form=create. The template name is set to "redis-status". It is assigned to the "Templates" group. The "Add" button is visible at the bottom left.

- 添加监控项

The screenshot shows the Zabbix configuration interface for creating a new monitoring item. The key settings are:

- 名称 (Name):** connected_clients
- 类型 (Type):** Zabbix客户端(主动式) (Zabbix Client (Active))
- 键值 (Key):** redis_status[redis_status.6379,connected_clients]
- 信息类型 (Information Type):** 数字(无正负) (Integer (No Sign))
- 更新间隔 (Update Interval):** 5s
- 历史数据保留时长 (Historical Data Retention):** 90d
- 趋势存储时间 (Trend Storage Time):** 365d
- 查看值 (Value View):** 不变 (Constant)
- 新的应用集 (New Application Set):** redis

- 数据有小数的需要选择信息类型为浮点型

The screenshot shows the Zabbix configuration interface for creating a new monitoring item. The key settings are:

- 名称 (Name):** used_cpu_sys
- 类型 (Type):** Zabbix客户端(主动式) (Zabbix Client (Active))
- 键值 (Key):** redis_status[redis_status.6379,used_cpu_sys]
- 信息类型 (Information Type):** 浮点数 (Float)
- 更新间隔 (Update Interval):** 5s
- 历史数据保留时长 (Historical Data Retention):** 90d
- 趋势存储时间 (Trend Storage Time):** 365d
- 查看值 (Value View):** 不变 (Constant)
- 新的应用集 (New Application Set):** redis

- 添加图形

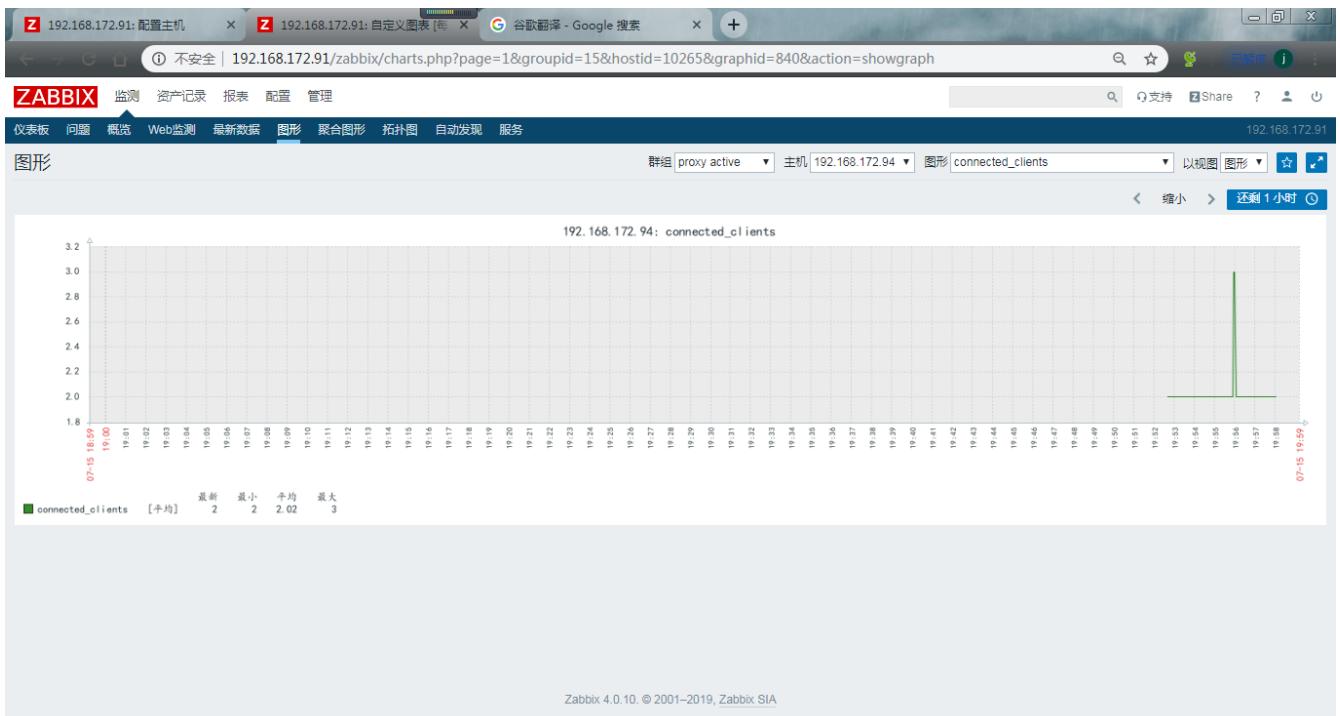
ZABBIX 监测 资产记录 报表 配置 管理

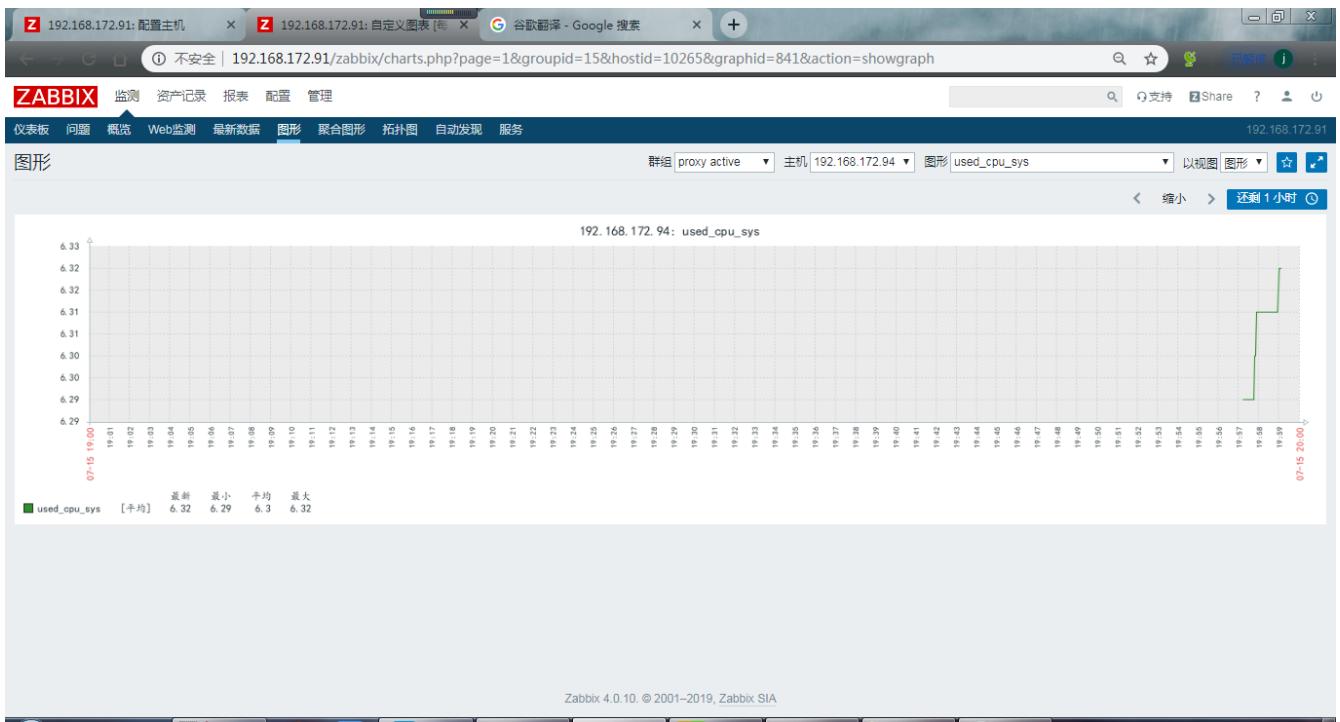
所有模板 / redis-status 应用集 1 监控项 2 触发器 图形 聚合图形 自动发现规则 Web 场景

图形 预览

* 名称 connected_clients
 * 宽 900
 * 高 200
 图形类别 正常
 查看图例
 查看工作时间
 查看触发器
 百分比线(左)
 百分比线(右)
 纵轴Y最小值MIN 可计算的
 纵轴最大值 可计算的
 * 监控项
 名称 功能 绘图风格 纵轴Y侧 颜色 动作
 1: redis-status: connected_clients 平均 线 左侧 1A7C11 移除
 添加 取消

- 关联主机验证数据





4.7 : SNMP

4.7.1 : SNMP协议介绍

- SNMP是英文“Simple Network Management Protocol”的缩写，中文意思是“简单网络管理协议”，SNMP是一种简单网络管理协议，它属于TCP/IP五层协议中的应用层协议，用于网络管理的协议，SNMP主要用于网络设备的管理
- SNMP的基本思想：为不同种类的设备、不同厂家生产的设备、不同型号的设备，定义为一个统一的接口和协议，使得管理员可以使用统一的外观面对这些需要管理的网络设备进行管理。通过网络，管理员可以管理位于不同物理空间的设备，从而大大提高网络管理的效率，简化网络管理员的工作
- SNMP协议是TCP/IP协议簇的一个应用层协议，在1988年被制定，并被Internet体系结构委员会（IAB）采纳作为一个短期的网络管理解决方案，SNMP的协议版本目前有SNMP v1、SNMP v2c和SNMP v3三种版本，其具体差别如下：
 - SNMP v1采用团体名（Community Name）认证，团体名用来定义SNMP NMS和SNMP Agent的关系，如果SNMP报文携带的团体名没有得到设备的认可，该报文将被丢弃，团体名起到了类似于密码的作用，用来限制SNMP NMS对SNMP Agent的访问
 - SNMP v2c也采用团体名认证，它在兼容SNMP v1的同时又扩充了SNMP v1的功能，它提供了更多的操作类型（GetBulk和InformRequest）、支持更多的数据类型（Counter64等）、提供了更丰富的错误代码且能够更细致地区分错误
 - SNMP v3提供了基于用户的安全模型（USM，User-Based Security Model）的认证机制，用户可以设置认证和加密功能，认证用于验证报文发送方的合法性，避免非法用户的访问，加密则是对NMS和Agent之间的传输报文进行加密，以免被窃听。通过有无认证和有无加密等功能组合，可以为SNMP NMS和SNMP Agent之间的通信提供更高的安全性

4.7.2 : SNMP工作机制

SNMP的工作机制通过SNMP网络元素分为NMS和Agent两种：

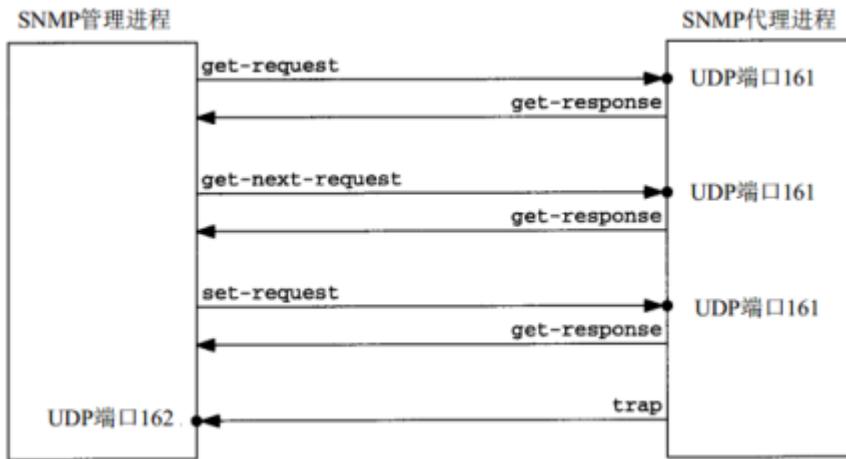
- NMS（网络管理站）：是运行SNMP客户端程序的工作站，能够提供非常友好的人机交互界面，方便网络管理员完成绝大多数的网络管理工作

- Agent是驻留在设备上的一个进程，负责接收、处理来自NMS的请求报文。在一些紧急情况下，如接口状态发生改变等，Agent也会主动通知NMS
- NMS是SNMP网络的管理者，Agent是SNMP网络的被管理者。NMS和Agent之间通过SNMP协议来交互管理信息

4.7.3 : SNMP数据交互

SNMP管理进程与代理进程之间为了交互信息，定义了5种报文：

- get - request操作：从代理进程处提取一个或多个参数值
- get-next-request操作：从代理进程处提取一个或多个参数的下一个参数值
- set-request操作：设置代理进程的一个或多个参数值
- get-response操作：返回的一个或多个参数值。这个操作是由代理进程发出的
- trap操作：代理进程主动发出的报文，通知管理进程有某些事情发生



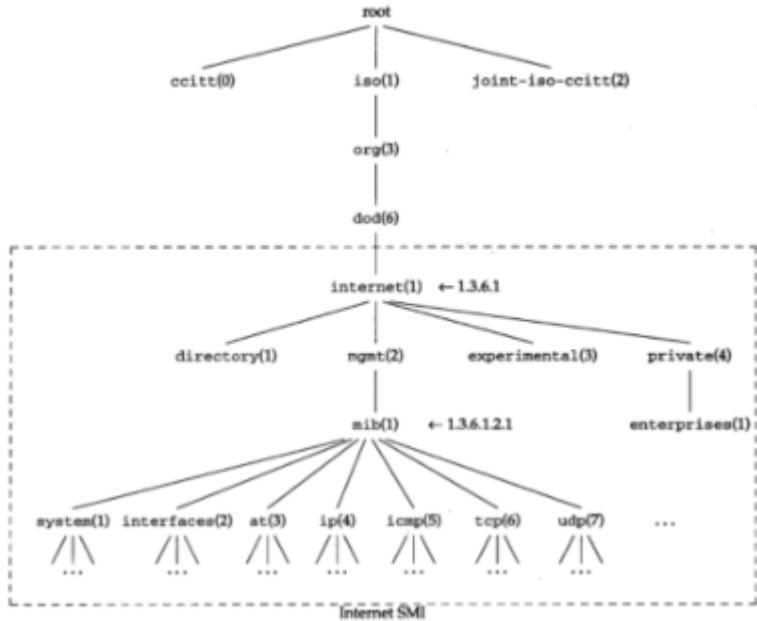
4.7.4 : SNMP组织结构

一套完整的SNMP系统主要包括以下几个方面：

- SNMP报文协议
- 管理信息机构SMI，一套公用的结构和表示符号
- 管理信息库MIB，管理信息库包含所有代理进程的所有可被查询和修改的参数
- OID，一个OID是一个唯一的键值对，用于标识具体某一个设备的某个具体信息(对象标识)，如端口信息、设备名称等

4.7.4.1 : MIB

所谓(MIB)管理信息库，就是所有代理进程包含的、并且能够被管理进程进行查询和设置的信息的集合。MIB是基于对象标识树的，对象标识是一个整数序列，中间以"."分割，这些整数构成一个树型结构，类似于D N S或U n i x的文件系统，M I B被划分为若干个组，如s y s t e m、i n t e r f a c e s、a t (地址转换)和i p组等。i s o . o r g . d o d . i n t e r n e t . p r i v a t e . e n t e r p r i s e s (1 . 3 . 6 . 1 . 4 . 1) 这个标识，是给厂家自定义而预留的，比如华为的为1.3.6.1.4.1.2011，华三的为1.3.6.1.4.1.25506



4.7.4.2 : OID

- Centos部分常用OID

1 minute Load: .1.3.6.1.4.1.2021.10.1.3.1	Total Swap Size: .1.3.6.1.4.1.2021.4.3.0
CPU 负载 5 minute Load: .1.3.6.1.4.1.2021.10.1.3.2	Available Swap Space: .1.3.6.1.4.1.2021.4.4.0
15 minute Load: .1.3.6.1.4.1.2021.10.1.3.3	Total RAM in machine: .1.3.6.1.4.1.2021.4.5.0
percentage of user CPU time: .1.3.6.1.4.1.2021.11.9.0	Total RAM used: .1.3.6.1.4.1.2021.4.6.0
raw user cpu time: .1.3.6.1.4.1.2021.11.50.0	Total RAM Free: .1.3.6.1.4.1.2021.4.11.0
percentages of system CPU time: .1.3.6.1.4.1.2021.11.10.0	Total RAM Shared: .1.3.6.1.4.1.2021.4.13.0
CPU 信息 raw system cpu time: .1.3.6.1.4.1.2021.11.52.0	Total RAM Buffered: .1.3.6.1.4.1.2021.4.14.0
percentages of idle CPU time: .1.3.6.1.4.1.2021.11.11.0	Total Cached Memory: .1.3.6.1.4.1.2021.4.15.0
raw idle cpu time: .1.3.6.1.4.1.2021.11.53.0	Path where the disk is mounted: .1.3.6.1.4.1.2021.9.1.2.1
raw nice cpu time: .1.3.6.1.4.1.2021.11.51.0	Path of the device for the partition: .1.3.6.1.4.1.2021.9.1.3.1
sysDescr 1.3.6.1.2.1.1.1	Total size of the disk/partition (kBytes): .1.3.6.1.4.1.2021.9.1.6.1
sysObjectID 1.3.6.1.2.1.1.2	Available space on the disk: .1.3.6.1.4.1.2021.9.1.7.1
系统信息 sysUpTime 1.3.6.1.2.1.1.3	Used space on the disk: .1.3.6.1.4.1.2021.9.1.8.1
sysContact 1.3.6.1.2.1.1.4	Percentage of space used on disk: .1.3.6.1.4.1.2021.9.1.9.1
sysName 1.3.6.1.2.1.1.5	Percentage of inodes used on disk: .1.3.6.1.4.1.2021.9.1.10.1

- 如何测试OID

snmpwalk是SNMP的一个工具，它使用SNMP的GET请求查询指定OID (SNMP协议中的对象标识) 入口的所有OID树信息，并显示给用户。通过snmpwalk也可以查看支持SNMP协议 (可网管) 的设备的一些其他信息，比如cisco交换机或路由器IP地址、内存使用率等，也可用来协助开发SNMP功能。

```
#要使用snmpwalk需要先安装net-snmp软件包
[root@s1 ~]# yum -y install net-snmp-utils
[root@s1 ~]# snmpwalk -h
USAGE: snmpwalk [OPTIONS] AGENT [OID]
-h : 显示帮助。
-v : 指定snmp的版本， 1或者2c或者3。
-c : 指定连接设备SNMP密码。
-V : 显示当前snmpwalk命令行版本。
```

```
-r : 指定重试次数，默认为0次。  
-t : 指定每次请求的等待超时时间，单为秒，默认为3秒。  
-l : 指定安全级别：noAuthNoPriv|authNoPriv|authPriv。  
-a : 验证协议：MD5|SHA。只有-l指定为authNoPriv或authPriv时才需要。  
-A : 验证字符串。只有-l指定为authNoPriv或authPriv时才需要。  
-x : 加密协议：DES。只有-l指定为authPriv时才需要。  
-X : 加密字符串。只有-l指定为authPriv时才需要
```

4.7.5 : SNMP服务配置

4.7.5.1 : centos系统

```
#安装zabbix agent  
[root@node7 ~]$ rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-  
1.el7.noarch.rpm  
[root@node7 ~]$ yum clean all  
[root@node7 ~]$ yum -y install zabbix-agent
```

```
#安装net-snmp包  
[root@node7 ~]$ yum install net-snmp -y  
  
#修改配置文件  
[root@node7 ~]$ vim /etc/snmp/snmpd.conf  
#定义团体名称community可以修改，作用类似于密码，此处修改后模板内也需要修改  
# sec.name source community  
com2sec notConfigUser default centos  
  
#定义组  
# groupName securityModel securityName  
group notConfigGroup v1 notConfigUser  
group notConfigGroup v2c notConfigUser  
  
#定义视图  
# Make at least snmpwalk -v 1 localhost -c public system fast again.  
# name incl/excl subtree mask(optional)  
view systemview included .1.3.6.1.2.1.1  
view systemview included .1.3.6.1.2.1.25.1.1  
view systemview included .1.3.6.  
  
#定义具体的关联关系  
# group context sec.model sec.level prefix read write notif  
access notConfigGroup "" any noauth exact systemview none none
```

- 启动服务验证端口161

```
[root@node7 ~]$ systemctl start snmpd  
[root@node7 ~]$ systemctl enable snmpd  
[root@node7 ~]$ ss -nlu  
State      Recv-Q Send-Q          Local Address:Port      Peer Address:Port  
UNCONN      0        0              *:161                  *:*
```

- 在zabbix端安装snmpwalk测试

```
root@node1:~# snmpwalk -v 2c -c centos 192.168.172.141 .1.3.6.1.4.1.2021.4.3.0
iso.3.6.1.4.1.2021.4.3.0 = INTEGER: 2097148
root@node1:~# snmpwalk -v 2c -c centos 192.168.172.141 .1.3.6.1.4.1.2021.10.1.3.1
iso.3.6.1.4.1.2021.10.1.3.1 = STRING: "0.00"
```

4.7.5.2 : ubuntu系统

```
#apt-get install snmpd snmp snmp-mibs-downloader
```

4.7.6 : zabbix server添加主机

- 创建主机

The screenshot shows the Zabbix 'hosts.php?form=create' page. The 'Host' tab is selected. The host configuration includes:

- 主机名称:** 192.168.172.141
- 可见的名称:** 192.168.172.141
- 群组:** hard device (新) (new group)
- SNMP接口:** IP地址: 192.168.172.141, 端口: 161, 使用大量请求 (checked)
- agent代理程序的接口:** IP地址: 192.168.172.141, 端口: 161
- JMX接口:** 未配置
- IPMI接口:** 未配置
- 描述:** 未填写
- 由agent代理程序监测:** (无agent代理程序)
- 已启用:**

- 添加模板

The screenshot shows the Zabbix 4.0.10 interface for managing hosts. The top navigation bar includes tabs for 'ZABBIX', '监测' (Monitoring), '资产记录' (Asset Record), '报表' (Report), '配置' (Configuration), and '管理' (Management). The main content area is titled '主机' (Host) and shows a host configuration page for '所有主机 / 192.168.172.141'. The '模板' (Template) tab is selected. The page displays a table of linked templates, with 'Template Module Generic SNMPv2' and 'Template Module Interfaces SNMPv2' listed. Below the table is a search input field and a '添加' (Add) button. At the bottom are buttons for '更新' (Update), '克隆' (Clone), '全克隆' (Full Clone), '删除' (Delete), and '取消' (Cancel).

- 修改连接密码，不然会提示连接超时错误

The screenshot shows the Zabbix 4.0.10 interface for managing hosts. The top navigation bar includes tabs for 'ZABBIX', '监测' (Monitoring), '资产记录' (Asset Record), '报表' (Report), '配置' (Configuration), and '管理' (Management). The main content area is titled '主机' (Host) and shows a host configuration page for '所有主机 / 192.168.172.141'. The '宏' (Macro) tab is selected. A red arrow points to the '继承以及主机 宏' (Inherit and Host Macro) button. The page lists several macros with their values and global values:

- \$_ICMP_LOSS_WARN => 20
- \$_ICMP_RESPONSE_TIME_W => 0.15
- \$_IFCONTROL => 1
- \$_IF_ERRORS_WARN => 2
- \$_IF_UTIL_MAX => 90
- \$_SNMP_COMMUNITY => centos
- \$_SNMP_TIMEOUT => 3m

At the bottom are buttons for '添加' (Add), '更新' (Update), '克隆' (Clone), '全克隆' (Full Clone), '删除' (Delete), and '取消' (Cancel). A red arrow also points to the value 'centos' in the \$_SNMP_COMMUNITY field.

- 添加图形项

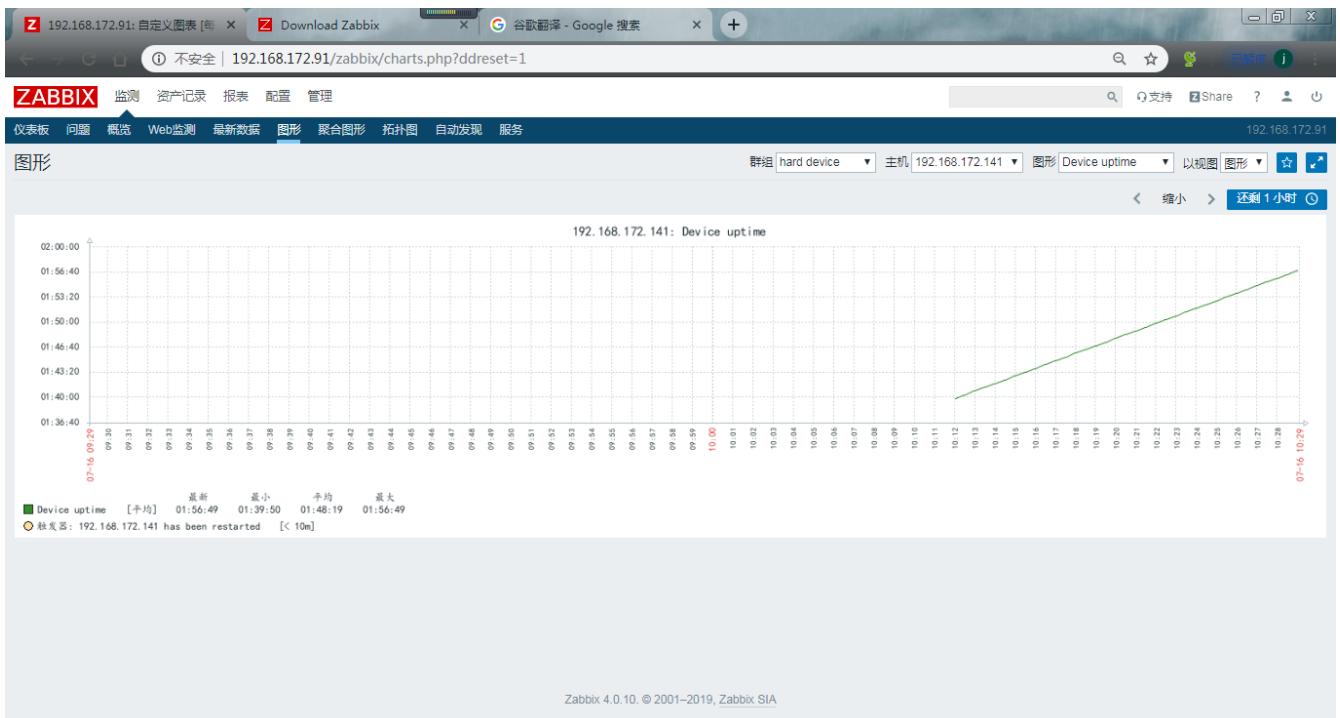
名称	宽	高	图形类别
Device uptime	900	200	正常
SNMP availability	900	200	正常

显示 已自动发现的 2 中的 2

0 选择 复制 删除

Zabbix 4.0.10. © 2001–2019, Zabbix SIA

- 查看检测图形



4.8：监控mysql

4.8.1：部署mysql主从同步

此处不再赘述

4.8.2：安装监控模板文件

- 官方介绍

官方文档及下载地址：插件地址：<https://www.percona.com/doc/percona-monitoring-plugins/LATEST/zabbix/index.html> 下载地址：<https://www.percona.com/downloads/> 安装教程：<https://www.percona.com/doc/percona-monitoring-plugins/LATEST/zabbix/index.html#installation-instructions>

- slave安装模板文件

```
[root@node7 ~]$yum install php php-mysql -y
[root@node7 ~]$rpm -ivh percona-zabbix-templates-1.1.8-1.noarch.rpm
[root@node7 ~]$cp /var/lib/zabbix/percona/templates/userparameter_percona_mysql.conf
/etc/zabbix/zabbix_agentd.d/
[root@node7 ~]$vim /var/lib/zabbix/percona/scripts/ss_get_mysql_stats.php.cnf
<?php
$mysql_user= 'root';
$mysql_pass= '';
```

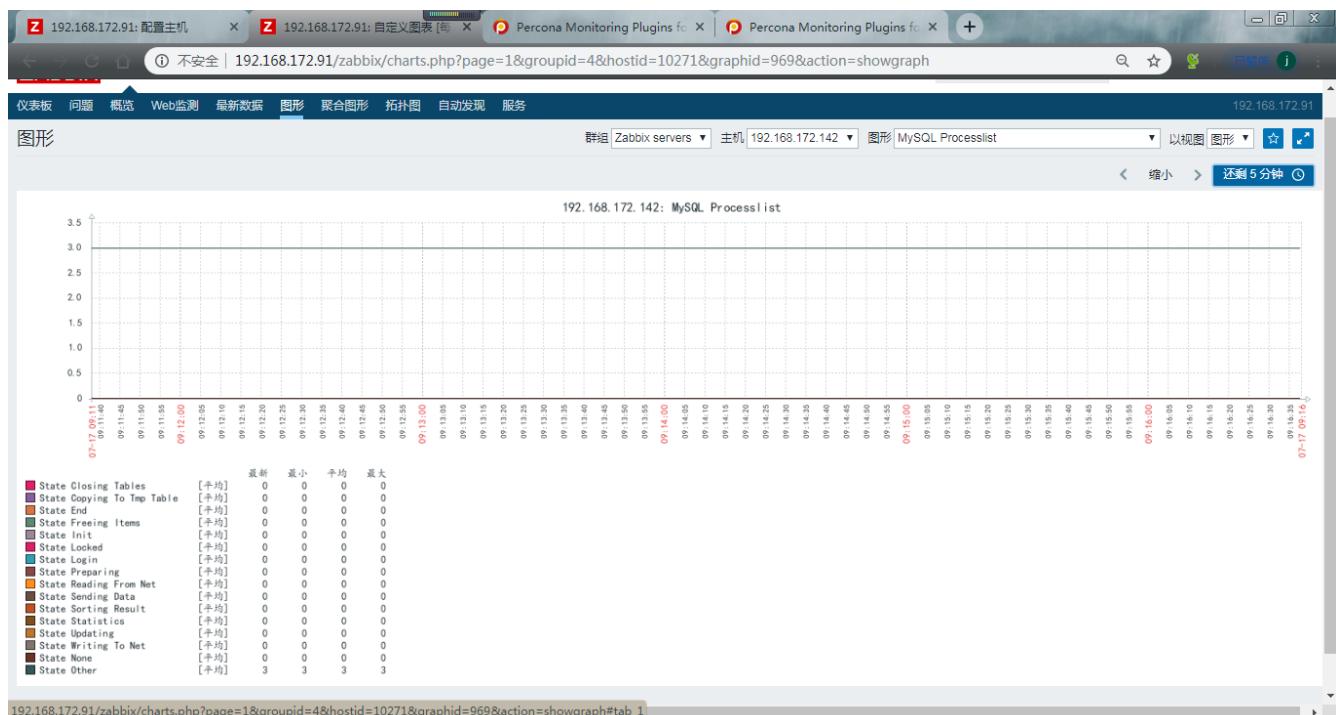
- 测试数据

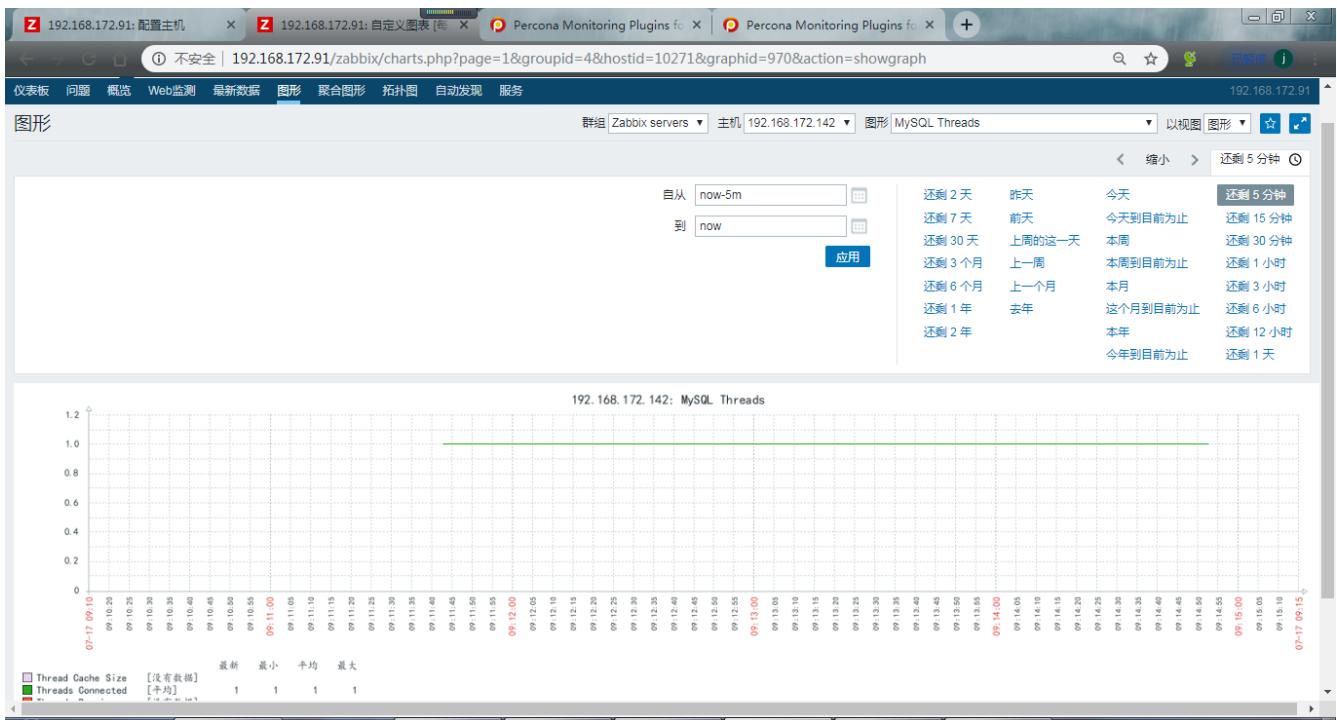
```
[root@node7 ~]$systemctl restart zabbix-agent
[root@node7 ~]$./var/lib/zabbix/percona/scripts/get_mysql_stats_wrapper.sh gg
0
[root@node7 ~]$./var/lib/zabbix/percona/scripts/get_mysql_stats_wrapper.sh running-slave
1
```

- 在web端导入模板zab_mysql_export_templates
- 更改模板模式为主动模式，关联到主机
- web端验证数据：

#如果无法获取数据，就要查看/tmp目录是否有一个localhost-mysql_cacti_stats.txt这样的类似文件，如果说有的话就把权限改为zabbix用户和用户组

```
[root@node7 ~]$chown -R mysql.mysql /tmp/localhost-mysql_cacti_stats.txt
```





4.8.3 : 自定义监控数据库

4.8.3.1 : 创建获取数据脚本

```
[root@node7 zabbix_agentd.d]$/vim mysql_monitor.sh
#!/bin/bash
Seconds_Behind_Master(){
    NUM=`mysql -uroot -hlocalhost -e "show slave status\G;" | grep "Seconds_Behind_Master:" | awk '{print $2}'`^
        echo $NUM
}

master_slave_check(){
NUM1=`mysql -uroot -hlocalhost -e "show slave status\G;" | grep "Slave_IO_Running" | awk -F: '{print $2}' | sed 's/^[\t]*//g'`^
#echo $NUM1
NUM2=`mysql -uroot -hlocalhost -e "show slave status\G;" | grep "Slave_SQL_Running:" | awk -F: '{print $2}' | sed 's/^[\t]*//g'`^
#echo $NUM2
if test $NUM1 == "Yes" && test $NUM2 == "Yes";then
    echo 50
else
    echo 100
fi
}

main(){
    case $1 in
        Seconds_Behind_Master)
            Seconds_Behind_Master;;
        ;;
        master_slave_check)
            master_slave_check
    esac
}
```

```
;;
esac
}
main $1
```

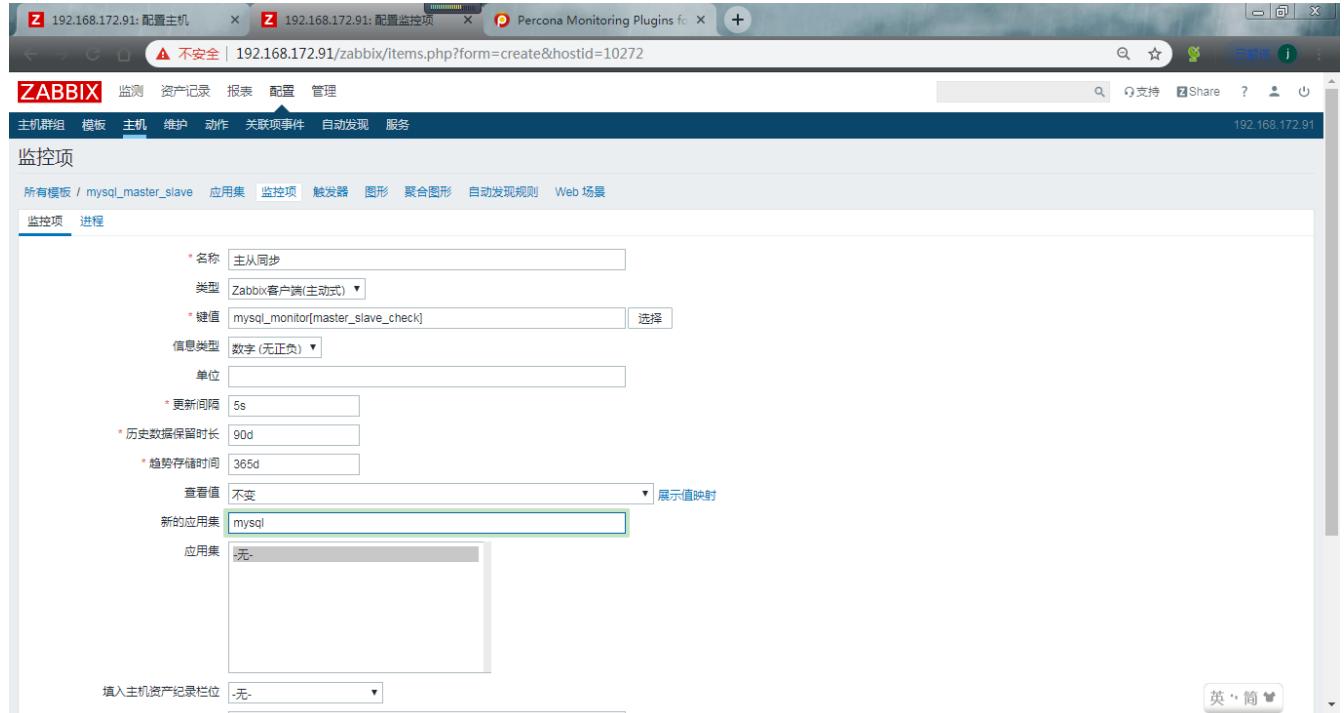
```
[root@node7 zabbix_agentd.d]$chmod +x mysql_monitor.sh
```

4.8.3.2：创建conf引用文件

```
#确保zabbix_agentd.conf文件中有Include=/etc/zabbix/zabbix_agentd.d/*.conf
[root@node7 zabbix_agentd.d]$vim mysql.conf
UserParameter=mysql_monitor[*],/etc/zabbix/zabbix_agentd.d/mysql_monitor.sh "$1"
```

```
[root@node7 zabbix_agentd.d]$systemctl restart zabbix-agent
```

4.8.3.3：创建监控模板



The screenshot shows the Zabbix web interface at <http://192.168.172.91/zabbix/items.php?form=create&hostid=10272>. The page title is "配置监控项". The navigation bar includes "ZABBIX 监测 资产记录 报表 配置 管理" and "主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务". The main content area is titled "监控项" and shows a form for creating a new monitoring item. The form fields include:

- 名称: 同步延时
- 类型: Zabbix客户端(主动式)
- 键值: mysql_monitor[Seconds_Behind_Master]
- 信息类型: 浮点数
- 单位:
- 更新间隔: 5s
- 历史数据保留时长: 90d
- 趋势存储时间: 365d
- 查看值: 不变
- 新的应用集: 无 (mysql selected)

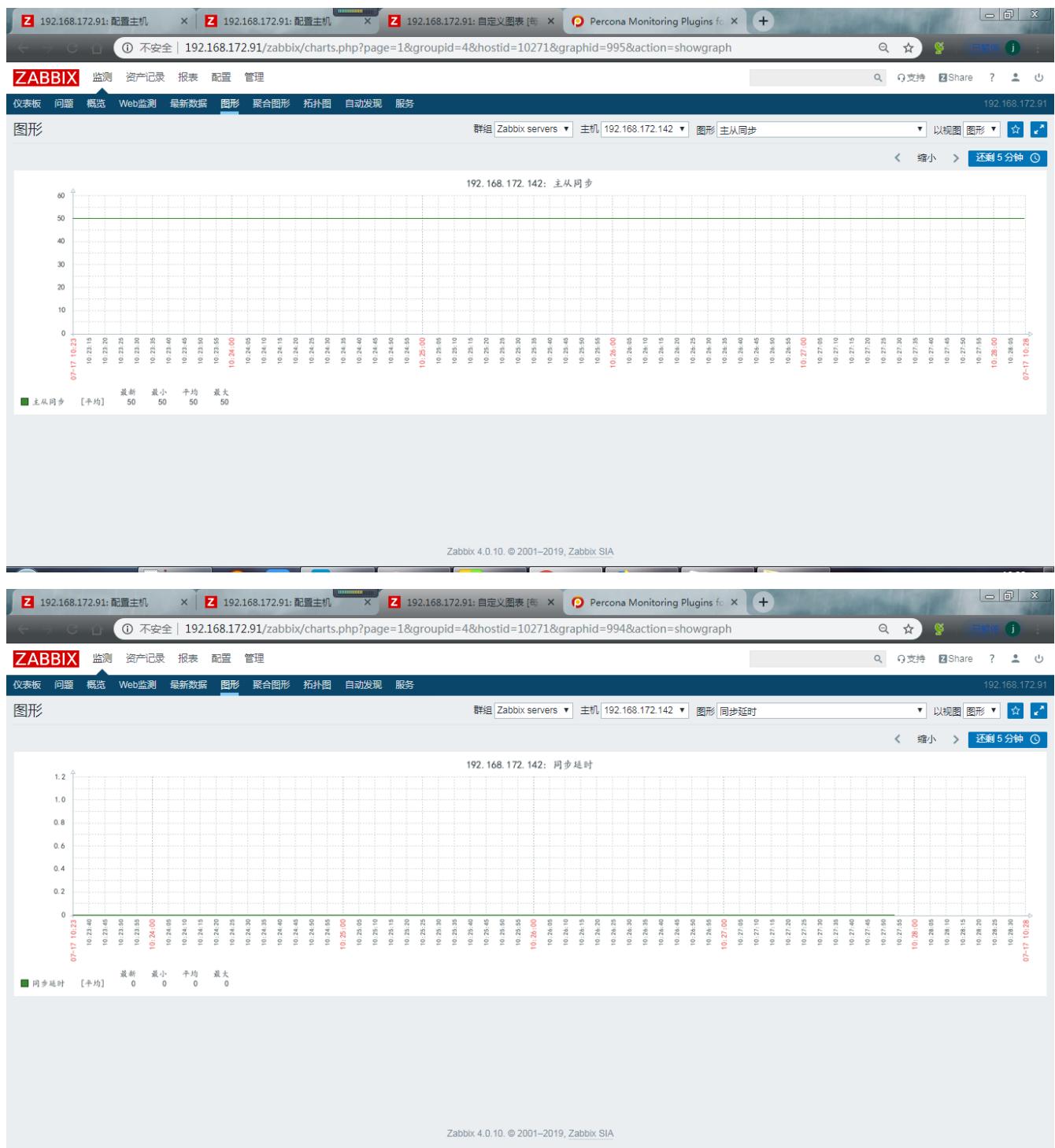
4.8.3.4 : 创建图形

The screenshot shows the Zabbix web interface at <http://192.168.172.91/zabbix/graphs.php?hostid=10272&form=create>. The page title is "配置图形". The navigation bar includes "ZABBIX 监测 资产记录 报表 配置 管理" and "主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务". The main content area is titled "图形" and shows a form for creating a new graph. The form fields include:

- 名称: 同步延时
- 宽: 900
- 高: 200
- 图形类别: 正常
- 查看图例:
- 查看工作时间:
- 查看触发器:
- 百分比线(左):
- 百分比线(右):
- 纵轴Y最小值MIN: 可计算的
- 纵轴最大值: 可计算的
- 监控项:

名称	功能	绘图风格	纵轴Y侧	颜色	动作
1: mysql_master_slave: 同步延时	平均	线	左侧	1A7C11	移除

4.8.3.5 : 验证数据



4.9：自定义端口监控

4.9.1：创建获取数据脚本

```
[root@node7 zabbix_agentd.d]#vim process_port_check.sh
#!/bin/bash
check_process(){
NUM=`ps -ef | grep -v grep | grep -v bash | grep ${NAME} | wc -l`
echo ${NUM}
}

check_port(){

}
```

```

ss -tnl | grep ${PORT} &> /dev/null
if [ $? -eq 0 ];then
    echo 50
else
    echo 100
fi
}

main(){
    case $1 in
        process)
            NAME=$2
            check_process;
            ;;
        port)
            PORT=$2
            check_port;
            ;;
    esac
}
main $1 $2

```

[root@node7 zabbix_agentd.d]\$chmod +x process_port_check.sh

4.9.2：创建conf文件引用脚本

```

[root@node7 zabbix_agentd.d]$vim process_port_check.conf
UserParameter=process_port[*],/etc/zabbix/zabbix_agentd.d/process_port_check.sh $1 $2

```

4.9.3：创建监控模板

The screenshot shows the Zabbix configuration interface. In the top navigation bar, there are three tabs: '配置主机' (Configuration Host), '配置监控项' (Configure Monitoring Item), and 'Percona Monitoring Plugins fo...'. Below the tabs, there's a breadcrumb navigation: '所有模板 / process_port_check 应用集 监控项 触发器 图形 聚合图形 自动发现规则 Web 场景'. The main content area is titled '监控项' (Monitoring Item) and shows a form for creating a new item. The form fields are as follows:

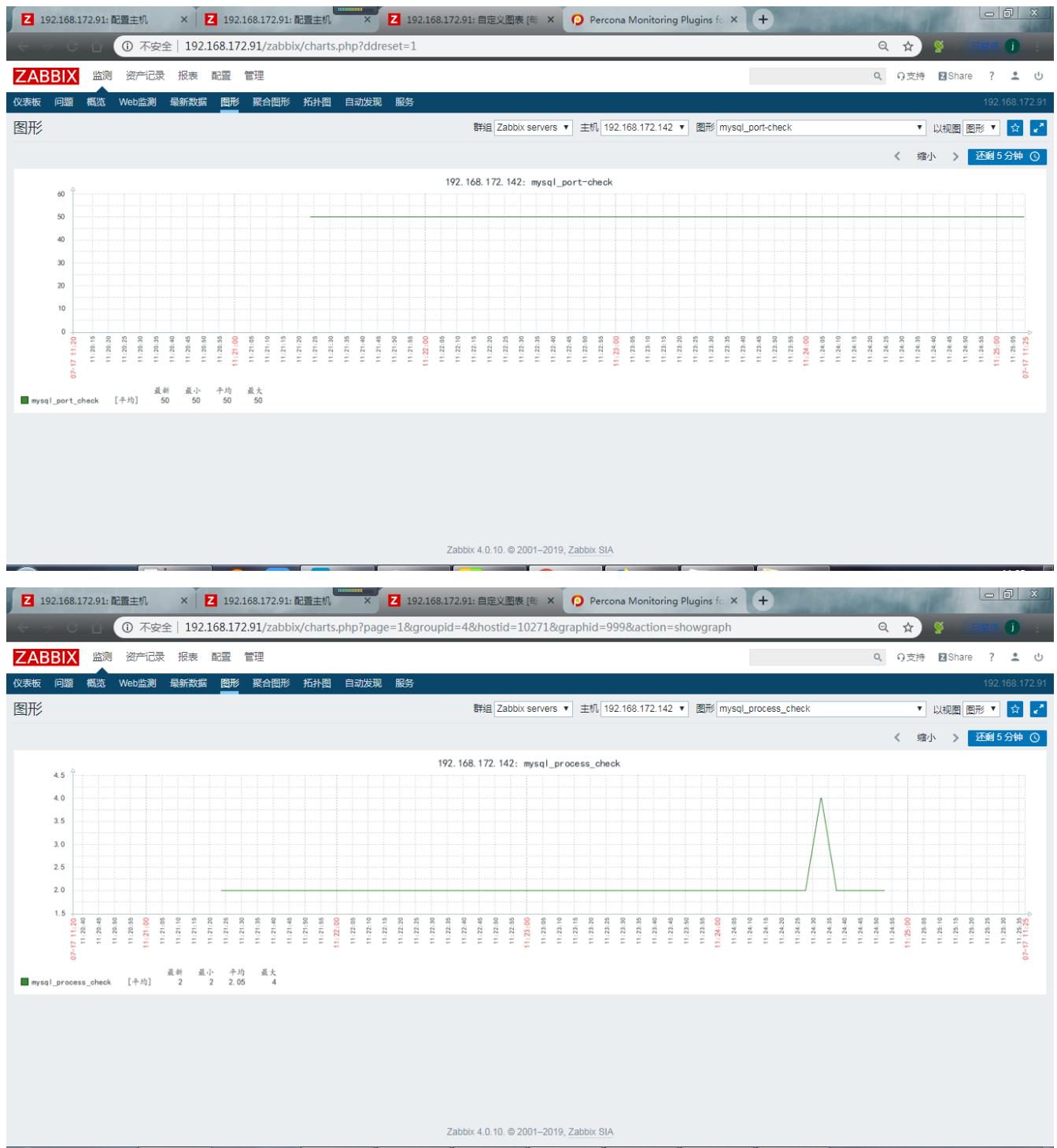
- *名称: mysql_port_check
- 类型: Zabbix客户端(主动式)
- *键值: process_port[port:3306]
- 信息类型: 数字(无正负)
- 单位: (empty)
- *更新间隔: 5s
- *历史数据保留时长: 90d
- *趋势存储时间: 365d
- 查看值: 不变
- 新的应用集: process_port
- 应用集: 无
- 填入主机资产纪录栏位: 无
- 描述: (empty)

The screenshot shows the Zabbix configuration interface for creating a new monitoring item. The item name is set to 'mysql_process_check'. The type is selected as 'Zabbix客户端端(主动式)'. The key value is 'process_port[process.mysql]'. The item type is '数字(无正负)'. The unit is left empty. The update interval is '5s'. The history retention time is '90d'. The trend storage time is '365d'. The value check is set to '不变'. A new application set 'process_port' is selected. The host dropdown is set to '无'.

4.9.4 : 创建图形

The screenshot shows the Zabbix configuration interface for creating a new graph. The graph name is 'mysql_process_check'. The width is '900' and the height is '200'. The chart type is '正常'. The legend is checked. The worktime is checked. The triggers are checked. The left Y-axis has '百分比线(左)' unchecked. The right Y-axis has '百分比线(右)' unchecked. The minimum Y-axis value is '可计算的'. The maximum Y-axis value is '可计算的'. A single monitoring item 'process_port_check: mysql_process_check' is added to the graph. The graph style is '平均' (Average), line color is '1A7C11', and the line style is solid. Buttons at the bottom include '更新' (Update), '克隆' (Clone), '删除' (Delete), and '取消' (Cancel).

4.9.5 : 验证数据



4.10：自定义系统基础监控模板

4.10.1：创建获取数据脚本

- mem.sh

```
# 在/etc/sudoers修改配置
99 zabbix ALL = NOPASSWD: /sbin/iotop #授权

#!/bin/bash
grep 7 /etc/redhat-release &> /dev/null
if [ $? -eq 0 ];then
```

```

mem_total(){
    TOTAL=`free |grep -i mem |awk '{print $2}'` 
    echo ${TOTAL}
}

mem_use(){
    USE=`free |grep -i mem | awk '{print $3}'` 
    echo ${USE}
}

mem_free(){
    FREE=`free |grep -i mem |awk '{print ($4+$6)}'` 
    echo ${FREE}
}

mem_usage(){
    USAGE=`free |grep -i mem | awk '{print ($3)/$2*100}' | awk -F ":" '{print $1}'` 
    echo ${USAGE}
}

else
mem_total(){
    TOTAL=`free |grep -i mem |awk '{print $2}'` 
    echo ${TOTAL}
}

mem_use(){
    USE=`free |grep -i mem | awk '{print $3}'` 
    echo ${USE}
}

mem_free(){
    #FREE=`free |grep -i mem |awk '{print $3+$5}'` 
    FREE=`free |grep -i mem |awk '{print ($4+$6+$7)}'` 
    echo ${FREE}
}

mem_usage(){
    USAGE=`free |grep -i mem | awk '{print ($3)/$2*100}' | awk -F ":" '{print $1}'` 
    echo ${USAGE}
}

fi

main(){
    case $1 in
        mem_total)
            mem_total;
            ;;
        mem_use)
            mem_use;
            ;;
        mem_free)
            mem_free;
            ;;
        mem_usage)
            mem_usage;
            ;;
    esac
}

```

```
}
```

```
main $1
```

- iotop_total.sh

```
#!/bin/bash
disk_read(){
    NUM=`/usr/bin/sudo iotop -b -n 3 | grep "Total DISK READ" | grep -v grep | awk -F ":" '{print $1}' | awk -F ":" '{print $2}' | tail -n1 | awk -F "/" '{print $1}'`  

    NUM2=`echo $NUM | awk '{print $1}'`  

    UNIT=`echo $NUM | awk '{print $2}'`  

    if [ $UNIT == "B" ];then  

        echo $NUM2  

    elif [ $UNIT == "K" ];then  

        NUM3=$(echo "$NUM2 * 1024" | bc)  

        echo $NUM3  

    elif [ $UNIT == "M" ];then  

        NUM3=$(echo "$NUM2 * 1024 * 1024" | bc)  

        echo $NUM3  

    elif [ $UNIT == "G" ];then  

        NUM3=$(echo "$NUM2 * 1024 * 1024 * 1024" | bc)  

        echo $NUM3
    fi
}
disk_write(){
    NUM=`/usr/bin/sudo iotop -b -n 3 | grep "Total DISK WRITE :" | grep -v grep | awk -F ":" '{print $2}' | awk -F ":" '{print $2}' | tail -n1 | awk -F "/" '{print $1}'`  

    NUM2=`echo $NUM | awk '{print $1}'`  

    UNIT=`echo $NUM | awk '{print $2}'`  

    if [ $UNIT == "B" ];then  

        echo $NUM2  

    elif [ $UNIT == "K" ];then  

        NUM3=$(echo "$NUM2 * 1024" | bc)  

        echo $NUM3  

    elif [ $UNIT == "M" ];then  

        NUM3=$(echo "$NUM2 * 1024 * 1024" | bc)  

        echo $NUM3  

    elif [ $UNIT == "G" ];then  

        NUM3=$(echo "$NUM2 * 1024 * 1024 * 1024" | bc)  

        echo $NUM3
    fi
}
main(){
    case $1 in
        disk_read)
            disk_read;
            ;;
        disk_write)
            disk_write;
            ;;
    esac
```

```
}
```

```
main $1
```

- cpu_use.sh

```
#!/bin/bash
cpu_usage(){
    NUM=`/usr/bin/top -bn 1 | grep Cpu | awk '{print $2}' | awk -F ":" '{print $1}'` 
    echo ${NUM}
}

main(){
    case $1 in
        cpu_usage)
            cpu_usage;
            ;;
        esac
}

main $1
```

```
[root@node7 zabbix_agentd.d]$chmod +x mem.sh
[root@node7 zabbix_agentd.d]$chmod +x iotop_total.sh
[root@node7 zabbix_agentd.d]$chmod +x cpu_use.sh
```

```
[root@node7 zabbix_agentd.d]$systemctl restart zabbix-agent
```

4.10.2 : 创建conf文件引用脚本

```
[root@node7 zabbix_agentd.d]$vim all.conf
UserParameter=mem_status[*],/etc/zabbix/zabbix_agentd.d/mem.sh $1
UserParameter=disk_io[*],/etc/zabbix/zabbix_agentd.d/iotop_total.sh $1
UserParameter=cpu_use[*],/etc/zabbix/zabbix_agentd.d/cpu_use.sh $1
```

4.10.3 : 创建监控模板

ZABBIX 监测 资产记录 报表 配置 管理

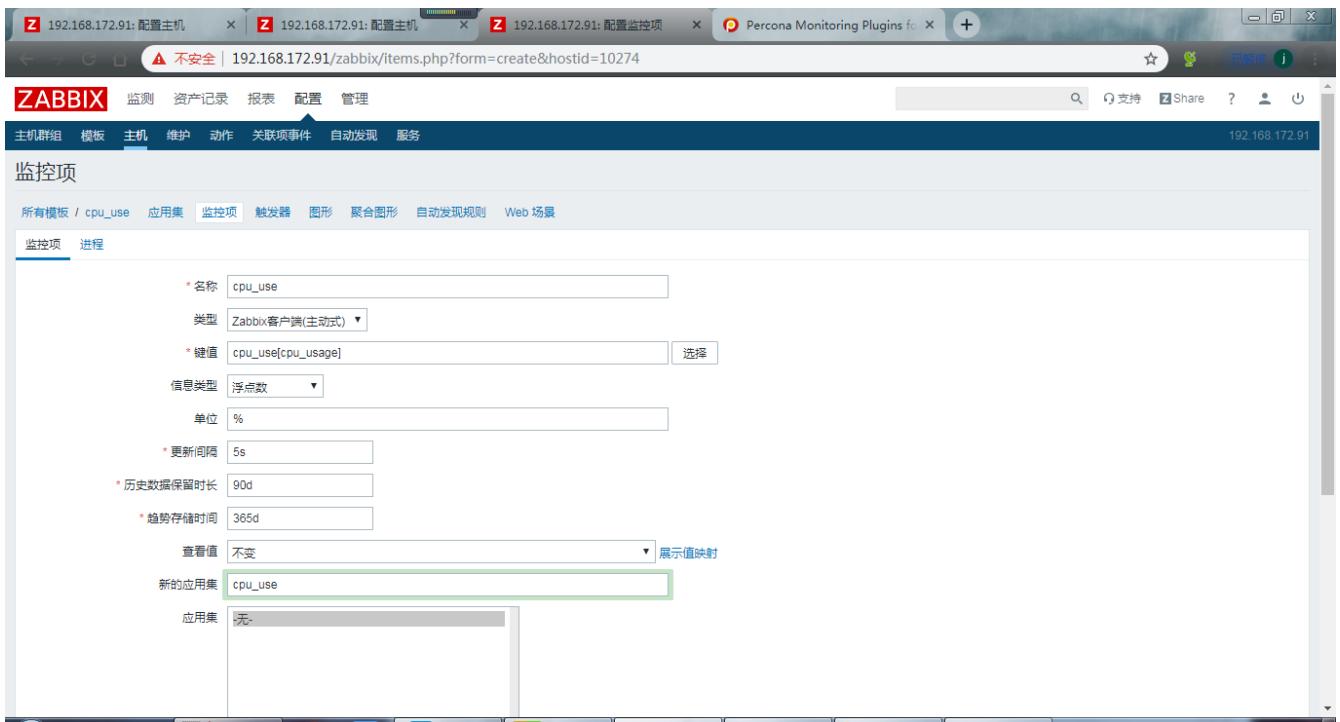
主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 192.168.172.91

监控项

所有模板 / cpu_use 应用集 监控项 触发器 图形 聚合图形 自动发现规则 Web 场景

监控项 进程

* 名称: cpu_use
类型: Zabbix客户端(主动式)
键值: cpu_use[cpu_usage]
信息类型: 浮点数
单位: %
* 更新间隔: 5s
* 历史数据保留时长: 90d
* 趋势存储时间: 365d
查看值: 不变 展示值映射
新的应用集: cpu_use
应用集: -无-



ZABBIX 监测 资产记录 报表 配置 管理

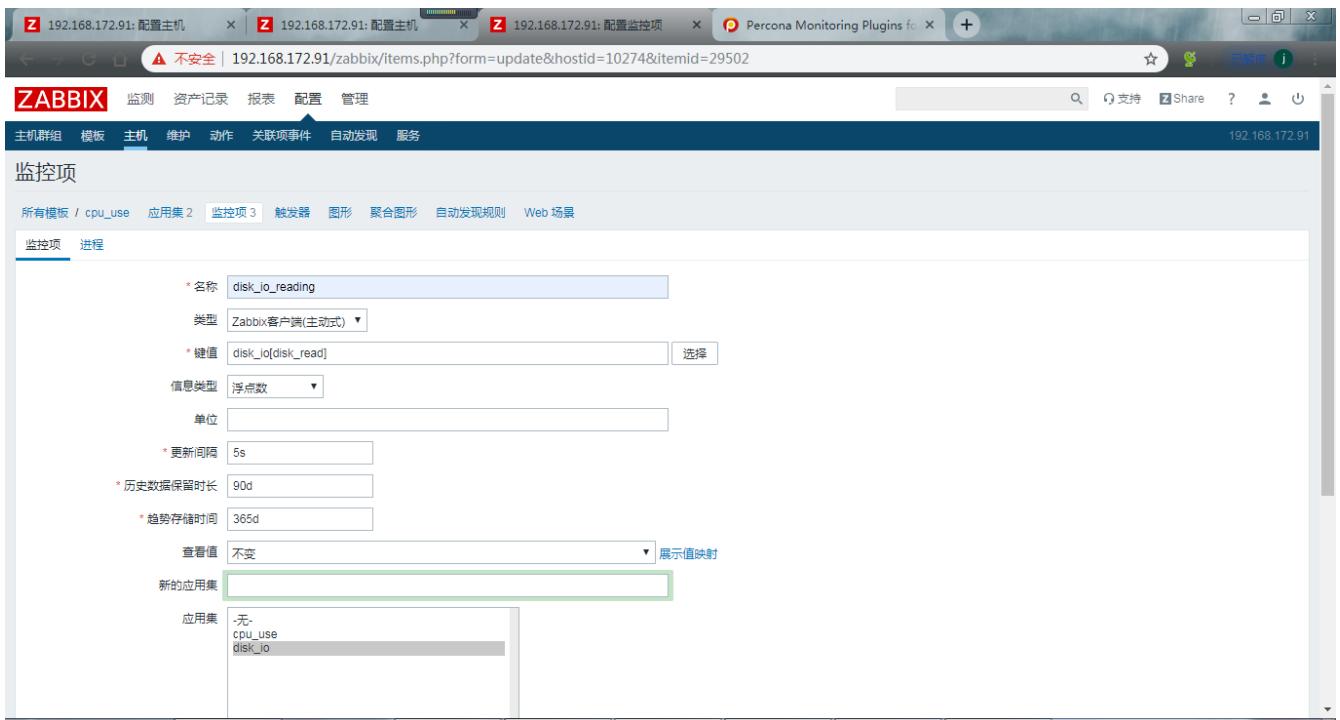
主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 192.168.172.91

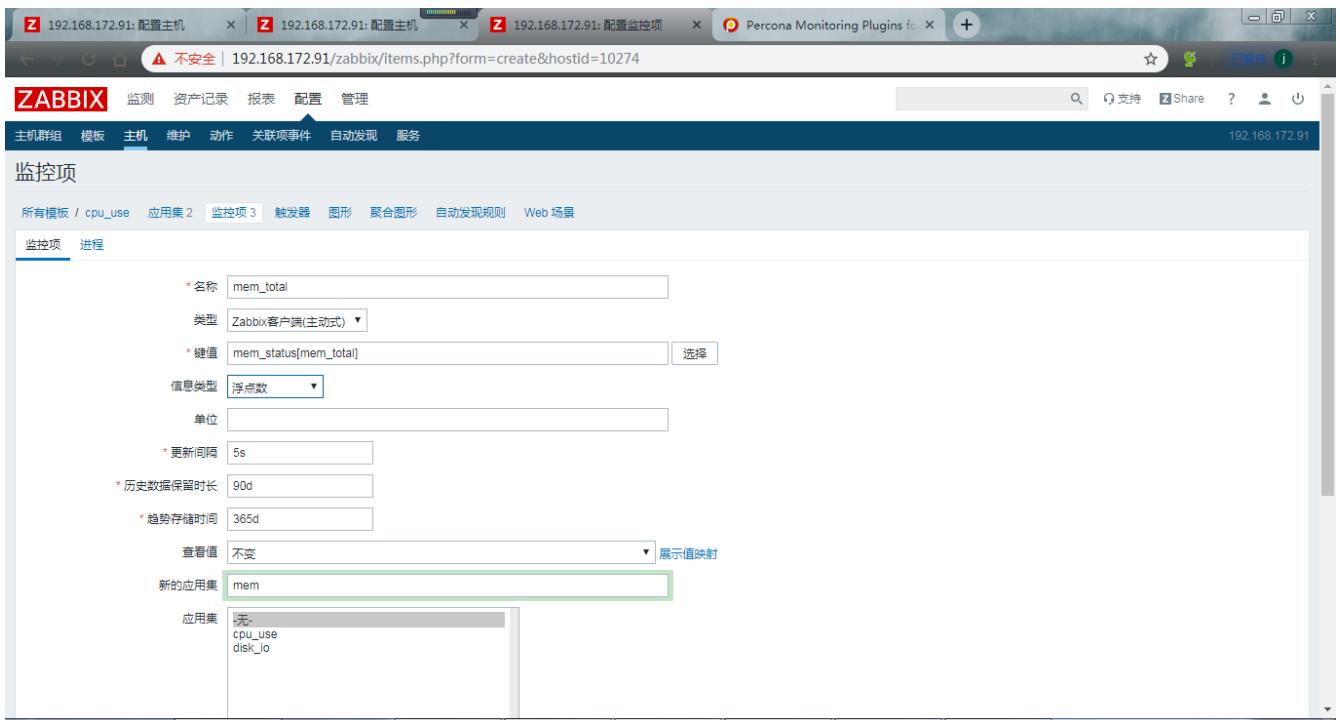
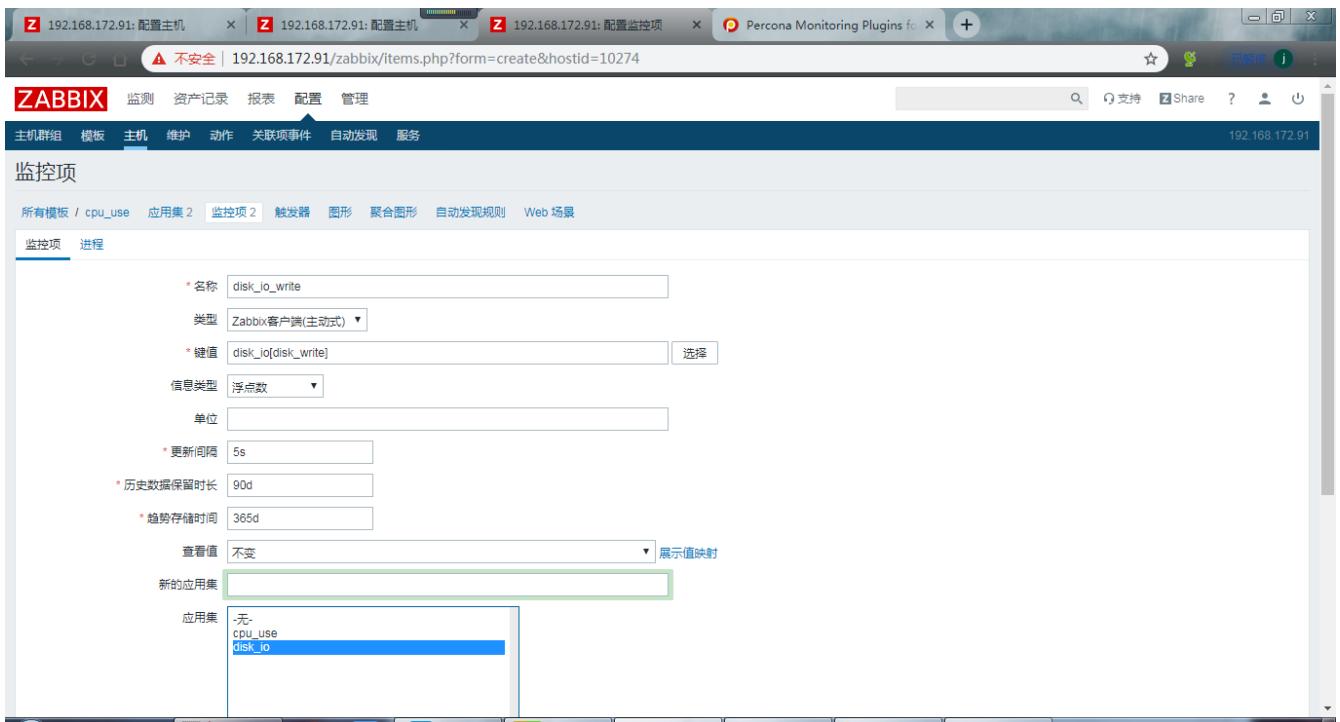
监控项

所有模板 / cpu_use 应用集 2 监控项 3 触发器 图形 聚合图形 自动发现规则 Web 场景

监控项 进程

* 名称: disk_io_reading
类型: Zabbix客户端(主动式)
键值: disk.io[disk.read]
信息类型: 浮点数
单位:
* 更新间隔: 5s
* 历史数据保留时长: 90d
* 趋势存储时间: 365d
查看值: 不变 展示值映射
新的应用集: -无-
应用集: -无-
cpu_use
disk_io





4.10.4 : 验证图形



五、监控报警及自愈

5.1：邮件报警

5.1.1：配置nginx

```
#开启nginx的状态页
location /nginx_status {
    stub_status;
    allow 127.0.0.1;
    deny all;
}

#启动nginx，访问测试
[root@node7 ~]$ curl 127.0.0.1/nginx_status
Active connections: 1
server accepts handled requests
 2 2
Reading: 0 Writing: 1 Waiting: 0
```

5.1.2：安装zabbix agent

```
[root@node7 ~]$ rpm -uvh https://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbix-release-4.0-1.el7.noarch.rpm
[root@node7 ~]$ yum install zabbix-agent -y

#修改zabbix agent配置文件
[root@node7 ~]$ vim /etc/zabbix/zabbix_agentd.conf
Server=192.168.172.93
ServerActive=192.168.2
Hostname=192.168.27.20
```

```

Timeout=30
UserParameter=nginx_status[*],/etc/zabbix/zabbix_agentd.d/nginx_status.sh $1 $2 $3

#在server端测试
root@node1:/apps/zabbix_server/bin# ./zabbix_get -s 192.168.172.142 -p 10050 -k
"nginx_status[nginx_status 80 active]"
1

```

5.1.3：编写监控脚本

```

[root@node7 zabbix_agentd.d]$vim nginx_status.sh
#!/bin/bash

nginx_status_fun() { #函数内容
    NGINX_PORT=$1 #端口，函数的第一个参数是脚本的第二个参数，即脚本的第二个参数是段口号
    NGINX_COMMAND=$2 #命令，函数的第二个参数是脚本的第三个参数，即脚本的第三个参数是命令
    nginx_active(){ #获取nginx_active数量，以下相同，这是开启了nginx状态但是只能从本机看到
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| grep
'Active' | awk '{print $NF}'
    }
    nginx_reading(){ #获取nginx_reading状态的数量
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| grep
'Reading' | awk '{print $2}'
    }
    nginx_writing(){
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| grep
'Writing' | awk '{print $4}'
    }
    nginx_waiting(){
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| grep
'Waiting' | awk '{print $6}'
    }
    nginx_accepts(){
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| awk NR==3
| awk '{print $1}'
    }
    nginx_handled(){
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| awk NR==3
| awk '{print $2}'
    }
    nginx_requests(){
        /usr/bin/curl "http://127.0.0.1:\"$NGINX_PORT\"/nginx_status/" 2>/dev/null| awk NR==3
| awk '{print $3}'
    }
    case $NGINX_COMMAND in
        active)
            nginx_active;
            ;;
        reading)
            nginx_reading;
            ;;
        writing)
            nginx_writing;
    esac
}

```

```

;;
waiting)
    nginx_waiting;
;;
accepts)
    nginx_accepts;
;;
handled)
    nginx_handled;
;;
requests)
    nginx_requests;
esac
}

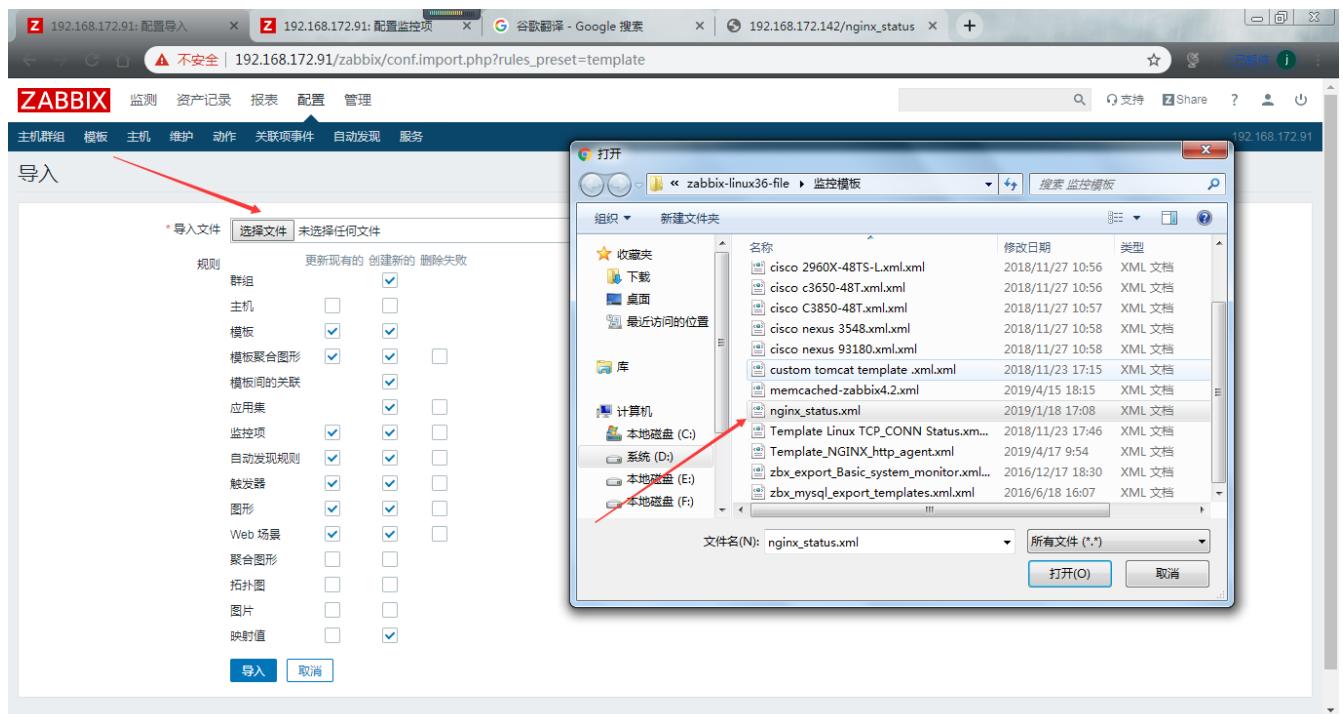
main() { #主函数内容
    case $1 in #分支结构，用于判断用户的输入而进行响应的操作
        nginx_status) #当输入nginx_status就调用nginx_status_fun，并传递第二和第三个参数
            nginx_status_fun $2 $3;
            ;;
        *) #其他的输入打印帮助信息
            echo $"Usage: $0 {nginx_status key}"
    esac #分支结束符
}

main $1 $2 $3

```

[root@node7 zabbix_agentd.d]\$chmod +x nginx_status.sh

5.1.4：导入nginx监控模板并关联至主机



5.1.5：配置邮件通知

ZABBIX 监测 资产记录 报表 配置 管理

一般 agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列 192.168.172.91

报警媒介类型

邮件类型为email

* 名称 Email

类型 电子邮件

* SMTP服务器 smtp.qq.com

SMTP服务器端口 465 qq邮箱端口

* SMTP HELO qq.com

* SMTP电邮 1146992442@qq.com

安全链接 无 STARTTLS(纯文本通信协议扩展) SSL/TLS

SSL验证对端

SSL验证主机

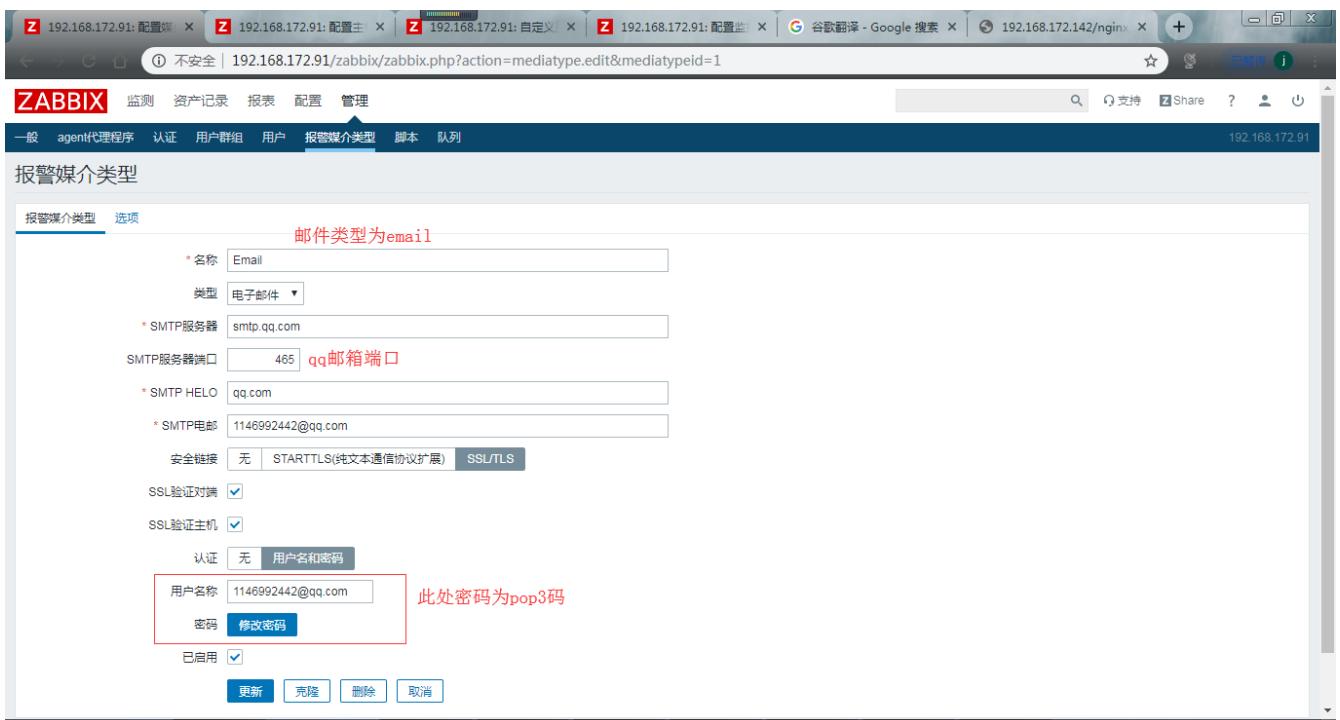
认证 无 用户名和密码

用户名 1146992442@qq.com 此处密码为pop3码

密码 修改密码

已启用

更新 克隆 删除 取消



ZABBIX 监测 资产记录 报表 配置 管理

一般 agent代理程序 认证 用户群组 用户 报警媒介类型 脚本 队列 192.168.172.91

用户

用户 报警媒介 权限

报警媒介

类型 收件人 Email 1146992442@qq.com 移除

添加

更新 删除

* 当启用时 1-7:00:00-24:00

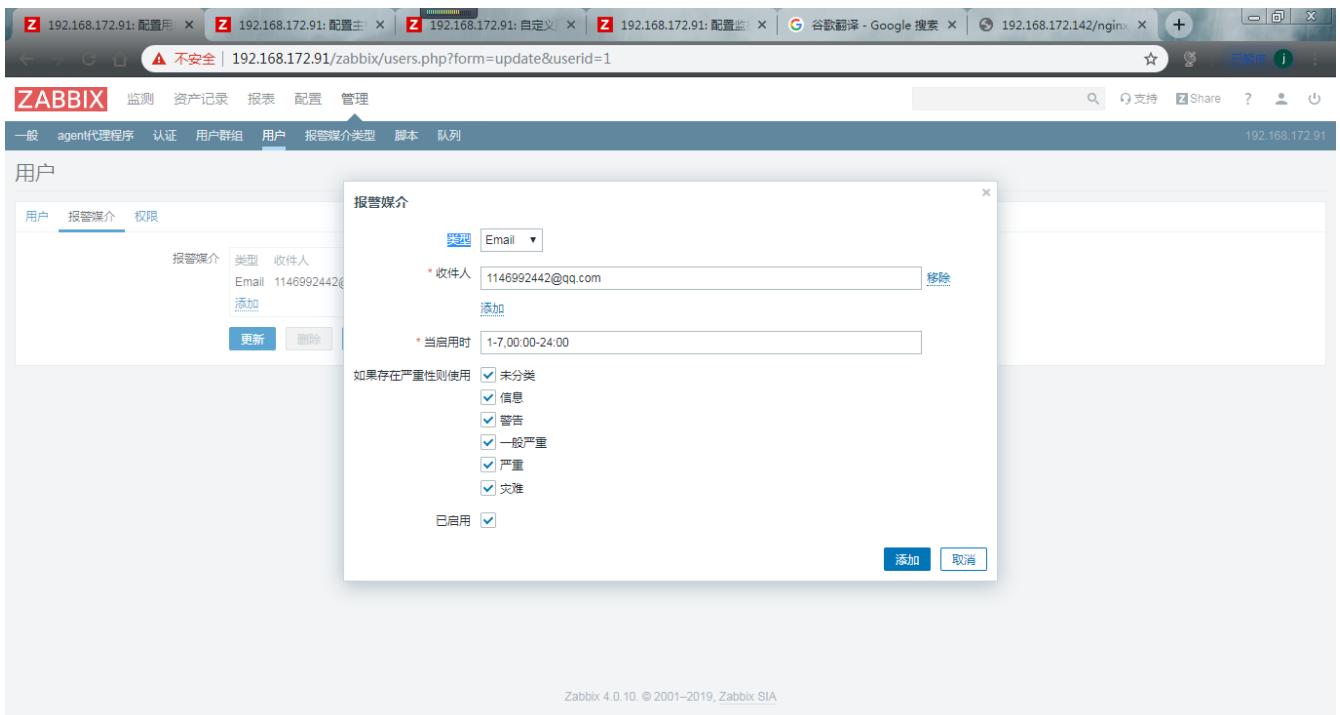
如果存在严重性则使用

未分类
 信息
 警告
 一般严重
 严重
 灾难

已启用

添加 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA



5.1.6：添加报警通知动作

ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 192.168.172.91

动作

动作 操作 恢复操作 更新操作

* 名称: send_mail
计算方式: 或 A or B or C or D or E or F

条件:

标签	名称	动作
A	触发器示警度 等于 <u>灾难</u>	移除
B	触发器示警度 等于 <u>严重</u>	移除
C	触发器示警度 等于 <u>一般严重</u>	移除
D	触发器示警度 等于 <u>警告</u>	移除
E	触发器示警度 等于 <u>信息</u>	移除
F	触发器示警度 等于 <u>未分类</u>	移除

新的触发条件:

触发器示警度 等于 未分类

添加 未分类 信息 警告 一般严重 严重 灾难

已启用

* 必须设置恢复时的至少一个执行内容或更新时的执行内容。

更新 克隆 删除 取消



ZABBIX 监测 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务 192.168.172.91

动作

动作 操作 恢复操作 更新操作

* 默认操作步骤持续时间: 60s
默认标题: zabbix报警_{EVENT.NAME}
消息内容: 报警服务器:Host: {HOST.NAME},
IP:{HOSTNAME!},
详情:{ITEM.NAME}:{ITEM.VALUE}

暂停操作以制止问题

操作 步骤 细节 开始于 持续时间 动作

1 - 3 发送消息给用户: Admin (Zabbix Administrator) 通过 Email 立即地 60 编辑 移除

操作细节:

步骤: 1 - 3 (0 - 无穷大)
步骤持续时间: 60 (0 - 使用默认)
操作类型: 立即消息

发送到用户群组: 用户群组 动作
添加

发送到用户: 用户 Admin (Zabbix Administrator) 动作
添加

仅发送到: Email

消息内容:
条件 标签 名称 动作



- 停止nginx服务后测试

ZABBIX 监测 资产记录 报表 配置 管理

仪表板 问题 概览 Web监测 最新数据 图形 聚合图形 拓扑图 自动发现 服务 192.168.172.91 编辑仪表盘

Global view

添加仪表盘 / Global view

系统信息		
参数	值	细节
Zabbix服务器运行中	是	192.168.172.91:10051
主机数量 (已启用/已禁用/模板)	96	5 / 0 / 91
监控项数量 (已启用/已禁用/不支持)	516	296 / 2 / 218
触发器数量 (已启用/已禁用 [问题/正常])	159	154 / 5 [0 / 154]
用户数(线上)	2	1
要求的主机性能, 每秒新值	11.12	

按严重性的问题

主机群组 ▲	灾难	严重	一般严重	警告	信息	未分类
Zabbix servers						未发现数据

本地

常用的拓扑图

未添加拓扑图

问题

时间 ▾	信息	主机	问题 · 严重性	持续时间	确认	动作	标记
未发现数据							

1146992442@qq.com
zabbix报警: 192.168.172.142 listen_80
1146992442 <1146992442@qq.com>
报告服务器:Host:192.168.172.142,IP:192.168.172.142,详情...
删除邮件 1/5

ZABBIX 监测 资产记录 报表 配置 管理

仪表板 问题 概览 Web监测 最新数据 图形 聚合图形 拓扑图 自动发现 服务 192.168.172.91 编辑仪表盘

Global view

添加仪表盘 / Global view

系统信息		
参数	值	细节
Zabbix服务器运行中	是	192.168.172.91:10051
主机数量 (已启用/已禁用/模板)	96	5 / 0 / 91
监控项数量 (已启用/已禁用/不支持)	516	296 / 2 / 218
触发器数量 (已启用/已禁用 [问题/正常])	159	154 / 5 [1 / 153]
用户数(线上)	2	1
要求的主机性能, 每秒新值	11.12	

按严重性的问题

主机群组 ▲	灾难	严重	一般严重	警告	信息	未分类
Zabbix servers		1				

本地

时间 用户/接受者 动作 信息/命令 状态 信息

2019-07-17 19:39:43	Admin (Zabbix Administrator)	Email	已送达
2019-07-17 19:38:43	Admin (Zabbix Administrator)	Email	已送达
2019-07-17 19:37:43	Admin (Zabbix Administrator)	Email	已送达
2019-07-17 19:37:23			

问题

时间 ▾	信息	主机	问题 · 严重性	持续时间	确认	动作	信息
19:37:23	192.168.172.142	192.168.172.142 listen_80	严重	2m 38s	不	3	

5.1.7 : 添加服务恢复通知动作

默认标题: Resolved: {EVENT.NAME}

消息内容: Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}
Problem name: {EVENT.NAME}
Host: {HOST.NAME}
Severity: {EVENT.SEVERITY}
Original problem ID: {EVENT.ID}

操作: 细节 动作

操作细节: 操作类型: 发送消息
* 您必须至少选择一个用户或用户组。
发送到用户群组: 用户群组 动作
添加

发送到用户: 用户 Admin (Zabbix Administrator) 动作 移除
添加
仅送到: Email
消息内容:

添加 取消 英 简

5.1.8 : 配置自治愈动作

* 名称: nginx_自愈

条件: 标签 名称 动作
新的触发条件: 触发器 等于 192.168.172.142: Nginx_status
添加 选择
已启用

必须设置恢复时的至少一个执行内容或执行内容或更新时的执行内容。
添加 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA 中 简

The screenshot shows the Zabbix configuration interface for a 'Problem' event. The 'Default title' is set to 'Problem: {EVENT.NAME}'. The 'Message content' field contains: 'Problem started at {EVENT.TIME} on {EVENT.DATE}', 'Problem name: {EVENT.NAME}', 'Host: {HOST.NAME}', 'Severity: {EVENT.SEVERITY}', and 'Original problem ID: {EVENT.ID}' with '(TRIGGER.URL)'. Under 'Action Details', there are two steps: 'Step 1' and 'Step 2'. Step 1 has a duration of '0'. The 'Action type' is set to 'Remote command'. The 'Target' is 'Current host' and the 'Action' is 'Delete'. The 'Type' is 'Shell script', 'Run in' is 'Zabbix Client', and the 'Command' is 'sudo /usr/sbin/nginx'. A red box highlights the 'Delete' action.

The screenshot shows the Zabbix configuration interface for a 'Resolved' event. The 'Default title' is 'Resolved: {EVENT.NAME}'. The 'Message content' field contains: 'Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}', 'Problem name: {EVENT.NAME}', 'Host: {HOST.NAME}', 'Severity: {EVENT.SEVERITY}', and 'Original problem ID: {EVENT.ID}'. Under 'Action Details', there is a 'Details' section with a note: '必须至少选择一个用户或用户组.' (At least one user or user group must be selected). It shows a 'User Group' section with '添加' (Add) and a 'User' section with 'Admin (Zabbix Administrator)' and '添加' (Add). The 'Type' is 'Email', 'Recipient' is 'Email', and 'Content' is checked. A red box highlights the 'User' section.

#修改zabbix agent配置文件

#开启远程命令

EnableRemoteCommands=1

#开启特殊字符的使用

UnsafeUserParameters=1

#重启服务

[root@node7 zabbix_agentd.d]\$systemctl restart zabbix-agent

#修改sudo权限

zabbix ALL=(ALL) NOPASSWD:ALL

- 停止nginx服务测试是否能自启动

5.2 : 微信报警

5.2.1 : 首先注册企业邮箱

申请企业微信号：<https://work.weixin.qq.com/>

首页--应用于小程序-创建应用

The screenshot shows the 'WeChat Work' application creation interface. The main title is 'WeChat Alarm'. It includes fields for 'AgentId' (1000002), 'Secret' (oNDcHx7hgNvRtFxe6jTxyyoO5GnVsSDbN4MuvN4jBzY), and 'Visible Range' (朱晓旭). Below these are sections for 'Functions': 'Send Message', 'Receive Message', and 'Automatic Response'.

记录创建过程中的三个数据：

AgentId	1000002
Secret	oNDcHx7hgNvRtFxe6jTxyyoO5GnVsSDbN4MuvN4jBzY
企业ID	ww44a7afecccd875f33

5.2.3 : ubuntu系统环境准备

```
root@node1:~# apt install python2.7
root@node1:~# ln -s /usr/bin/python2.7 /usr/bin/python
root@node1:~# apt install python-pip
root@node1:~# pip install requests
```

5.2.3 : zabbix server设置

```
#查看配置文件中的脚本存放路径
root@node1:~# cat /apps/zabbix_server/etc/zabbix_server.conf |grep AlertScriptsPath
### Option: AlertScriptsPath
# AlertScriptsPath=${datadir}/zabbix/alertscripts
```

- 在alertscripts路径下添加微信报警脚本

```
root@node1:/apps/zabbix_server/share/zabbix/alertscripts# pwd
/apps/zabbix_server/share/zabbix/alertscripts
root@node1:/apps/zabbix_server/share/zabbix/alertscripts# vim weixin.py
#!/usr/bin/env python
#coding:utf-8
#Author:Zhang Shijie
import requests
import sys
import os
import json
import logging

logging.basicConfig(level = logging.DEBUG, format = '%(asctime)s, %(filename)s, %(levelname)s, %(message)s',
datefmt = '%a, %d %b %Y %H:%M:%S',
filename = os.path.join('/tmp','weixin.log'),
filemode = 'a')

corpid='企业ID'
appsecret="SCRET"
agentid="Agent ID"
token_url='https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=' + corpid + '&corpsecret=' + appsecret
req=requests.get(token_url)
accesstoken=req.json()['access_token']

msgsend_url='https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token=' + accesstoken
touser=sys.argv[1]
subject=sys.argv[2]
message=sys.argv[2] + "\n\n" +sys.argv[3]

params={
"touser": touser,
"msgtype": "text",
"agentid": agentid,
"text": {
"content": message
},
"safe":0
}

req=requests.post(msgsend_url, data=json.dumps(params))
logging.info('sendto:' + touser + ';;subject:' + subject + ';;message:' + message)

#脚本添加可执行权限
root@node1:/apps/zabbix_server/share/zabbix/alertscripts# chmod +x weixin.py
root@node1:/apps/zabbix_server/share/zabbix/alertscripts# chown -R zabbix.zabbix
/tmp/weixin.log

#测试脚本，收到消息即为正确
```

```
root@node1:/apps/zabbix_server/share/zabbix/alertscripts# python weixin.py zhuxiaoxu "主题"
"内容"
```

5.2.4：添加报警媒介类型

The screenshot shows the Zabbix 4.0.10 interface with the '报警媒介类型' (Media Types) page open. The 'wechat' media type is selected and highlighted with a red border. The table lists the following media types:

名称	类型	状态	用于动作中	细节
Email	电子邮件	已启用	send_mail	SMTP服务器: "smtp.qq.com", SMTP HELO: "qq.com", SMTP电邮: "1146992442@qq.com"
Jabber	Jabber	已启用		Jabber 标识符: "jabber@example.com"
SMS	短信	已启用		GSM 调制解调器: "/dev/ttyS0"
wechat	脚本	已启用	wechat	脚本名称: "weixin.py"

The screenshot shows the Zabbix 4.0.10 interface with the '报警媒介类型' (Media Types) configuration page for editing the 'wechat' media type. The 'Options' tab is selected. The form fields are as follows:

- 名称: wechat
- 类型: 脚本
- 脚本名称: weixin.py
- 脚本参数:
 - 参数: [ALERT.SENDTO] 动作: 移除
 - 参数: [ALERT.SUBJECT] 动作: 移除
 - 参数: [ALERT.MESSAGE] 动作: 移除
- 已启用:

At the bottom are buttons for '更新' (Update), '克隆' (Clone), '删除' (Delete), and '取消' (Cancel).

5.2.5：添加报警用户

The screenshot shows the Zabbix 4.0.10 interface for managing users. A modal window titled '报警媒介' (Alarm Medium) is open, allowing configuration for a 'wechat' type. The configuration includes:

- 收件人 (Recipient): zhuxiaoxu
- 当启用时 (When Enabled): 1-7:00:00-24:00
- 如果存在严重性则使用 (If severity exists then use):
 - 未分类 (Unclassified) (unchecked)
 - 信息 (Information) (checked)
 - 警告 (Warning) (checked)
 - 一般严重 (General warning) (checked)
 - 严重 (Severe) (unchecked)
 - 灾难 (Disaster) (checked)
- 已启用 (Enabled) (checked)

At the bottom of the modal are '更新' (Update) and '取消' (Cancel) buttons.

The main user list table shows two entries:

报警媒介	类型	收件人	当启用时	如果存在严重性则使用	Status	动作
Email	1146992442@qq.com	1-7:00:00-24:00	未值	警一严灾	已启用	编辑 移除
wechat	zhuxiaoxu	1-7:00:00-24:00	未值	警一严灾	已启用	编辑 移除

At the bottom of the page is the footer: Zabbix 4.0.10. © 2001–2019, Zabbix SIA

5.2.6 : 添加动作

The screenshot shows the Zabbix 4.0.10 configuration interface for creating a new action. The action is named "wechat". The trigger selection dropdown is set to "或" (Or) and includes triggers A through F. Trigger A is selected, which corresponds to a severity of "严重" (Severe). The "已启用" (Enabled) checkbox is checked. The "更新" (Update) button is highlighted in blue.

The screenshot shows the detailed configuration of the "wechat" action. It includes a summary of the default operation parameters (持续时间: 60s, 标题: Problem: {EVENT.NAME}), message content (Problem started at {EVENT.TIME} on {EVENT.DATE}, Problem name: {EVENT.NAME}, Host: {HOST.NAME}, Severity: {EVENT.SEVERITY}), and original problem details (Original problem ID: {EVENT.ID}, [TRIGGER.URL]). The "暂停操作以制止问题" (Pause operation to stop problem) checkbox is checked. The "操作" (Operations) section shows a single step: "1 - 3 发送消息给用户: Admin (Zabbix Administrator) 通过 wechat 立即地 60" (Send message to user: Admin (Zabbix Administrator) via wechat immediately 60). The "操作细节" (Operation details) section shows the step number (1), duration (60), and message type (发送消息). The "发送到用户群组" (Send to user group) section lists "Zabbix administrators" as the recipient group. The "发送到用户" (Send to user) section lists "Admin (Zabbix Administrator)" as the recipient. The "仅送到" (Only send to) dropdown is set to "wechat".

5.2.7 : 测试报警

关闭nginx服务测试报警

9:44

4G+ 72%

← 微信报警



Problem: 192.168.172.142
listen_80

Problem started at 21:22:03 on
2019.07.17

Problem name:
[192.168.172.142](#) listen_80
Host: [192.168.172.142](#)
Severity: High

Original problem ID: 1568



Problem: 192.168.172.142
listen_80

Problem started at 21:22:03 on
2019.07.17

Problem name:
[192.168.172.142](#) listen_80
Host: [192.168.172.142](#)
Severity: High

Original problem ID: 1568



Problem: 192.168.172.142
listen_80

Problem started at 21:22:03 on



5.2.8 : 自治愈

The screenshot shows the Zabbix 4.0.10 interface with the 'Actions' tab selected. The main form is for creating a new action named 'hnginx_自治愈'. The 'Conditions' section contains a single condition: '触发器 等于 192.168.172.142:192.168.172.142/listen_80'. The 'Recovery conditions' section is currently empty. The 'Enabled' checkbox is checked. At the bottom, there are buttons for 'Update', 'Clone', 'Delete', and 'Cancel'.

ZABBIX 监测 资产记录 报表 配置 管理

动作 操作 恢复操作 更新操作

*名称 hnginx_自治愈

条件 标签 名称 A 触发器 等于 192.168.172.142:192.168.172.142/listen_80 动作 移除

新的触发条件 触发器名称 包含 添加

已启用

必须设置恢复时的至少一个执行内容或执行内容或更新时的执行内容。

更新 克隆 删除 取消

Zabbix 4.0.10. © 2001–2019, Zabbix SIA 英 · 简

The screenshot shows two separate configurations for actions in Zabbix:

Action 1 (Top):

- Default title:** Problem: {EVENT.NAME}
- Message content:** Problem started at {EVENT.TIME} on {EVENT.DATE}
Problem name: {EVENT.NAME}
Host: {HOST.NAME}
Severity: {EVENT.SEVERITY}
Original problem ID: {EVENT.ID}
(TRIGGER.URL)
- 暫停操作以制止問題:** checked
- 操作 (Operation):** Step 1-2: 在当前主机上执行远程命令 (Run remote command on current host)
- 操作细节 (Operation details):**
 - 步骤 (Step): 1 - 2 (0 - 无穷大)
 - 步骤持续时间 (Step duration): 0 (0 - 使用默认)
 - 操作类型 (Operation type): 远程命令 (Remote command)
 - 目标列表 (Target list):** 目标 (Target) - 当前主机 (Current host) - 移除 (Remove)
 - 类型 (Type): 自定义脚本 (Custom script)
 - 执行在 (Execute on): Zabbix 客户端 (Zabbix Client), Zabbix server (代理) (Zabbix Server (Proxy)), zabbix服务器 (Zabbix Server)
 - 命令 (Command): sudo /usr/sbin/nginx

Action 2 (Bottom):

- Default title:** Resolved: {EVENT.NAME}
- Message content:** Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}
Problem name: {EVENT.NAME}
Host: {HOST.NAME}
Severity: {EVENT.SEVERITY}
Original problem ID: {EVENT.ID}
- 操作 (Operation):** 发送消息给用户: Admin (Zabbix Administrator) 通过 wechat (Send message to user: Admin (Zabbix Administrator) via wechat)
- 操作细节 (Operation details):**
 - 操作类型 (Operation type): 告警消息 (Alert message)
 - 发送到用户群组 (Send to user group): 用户群组 (User group) - 添加 (Add)
 - 发送到用户 (Send to user): Admin (Zabbix Administrator) - 移除 (Remove) - 添加 (Add)
 - 仅送到 (Only send to): wechat (wechat) - 仅送到 (Only send to) - 添加 (Add)
 - 消息内容 (Message content):

六、API

参考资料：<https://www.zabbix.com/documentation/4.0/zh/manual/api>

6.1：使用API查询

6.1.1：获取token

```
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "user": "Admin", #zabbix登录名
        "password": "zabbix" #zabbix登录密码
    },
    "id": 1 #用户id , id=1为admin用户
}' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool #zabbix serverIP
```

```
root@node1:~# curl -s -X POST -H 'Content-Type:application/json' -d '
> {
>     "jsonrpc": "2.0",
>     "method": "user.login",
>     "params": {
>         "user": "Admin",
>         "password": "zabbix"
>     },
>     "id": 1
> }' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
{
    "id": 1,
    "jsonrpc": "2.0",
    "result": "8847f3273e4e0682331579cf9f4ee8a2" #此为获取到的token值
}
```

6.1.2 : 获取指定主机

```
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "192.168.172.142"
            ]
        }
    },
    "auth": "8847f3273e4e0682331579cf9f4ee8a2",
    "id": 1
}' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
```

6.1.3 : 获取所有主机

```
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["host"]
    },
    "auth": "8847f3273e4e0682331579cf9f4ee8a2",
    "id": 1
}' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
```

6.1.4 : 获取所有用户

```
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "auth": "8847f3273e4e0682331579cf9f4ee8a2",
    "id": 1
}' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
```

6.1.5 : 获取模板信息

```
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "NGINX_Check_Status" #此为模板名称
            ]
        }
    },
    "auth": "8847f3273e4e0682331579cf9f4ee8a2",
    "id": 1
}' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
```

6.1.6 : 基于脚本获取token

```
root@node1:~# cat token.py
#!/usr/bin/env python
# -*- coding:utf-8 -*-

import requests
import json
```

```

url = 'http://192.168.172.91/zabbix/api_jsonrpc.php'
post_data = {
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "user": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
post_header = {'Content-Type': 'application/json'}

ret = requests.post(url, data=json.dumps(post_data), headers=post_header)

zabbix_ret = json.loads(ret.text)
if not zabbix_ret.has_key('result'):
    print 'login error'
else:
    print zabbix_ret.get('result')

root@node1:~# python token.py
329f3c1181d2911f8c3ea4b038c634f7

```

6.2：使用API添加主机

API添加主机为预先知道要添加的主机IP、预先安装并配置好zabbix agent、预先知道要关联的模板ID/组ID等信息，然后同API提交请求添加

6.2.1：使用API添加主机命令格式

```

curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "host.create", #定义方法，N多钟
    "params": {
        "host": "API Add Host Test", #自定义添加后的agent的名称
        "name": " ", #可见的名称选项
        "interfaces": [
            {
                "type": 1, #类型为1表示agent，2是SNMP，3是IMPI，4是JMX
                "main": 1, #more接口
                "useip": 1, #0是使用DNS，1是使用IP地址
                "ip": "192.168.0.24", #添加的zabbix agent的IP地址
                "dns": "",
                "port": "10050" #agent端口
            }
        ],
        "groups": [
            {
                "groupid": "2" #添加到的组的ID
            }
        ],
        "templates": [

```

```
{  
    "templateid": "10001" #关联的模板的ID  
}  
]  
],  
"auth": "977781251d1222ebead6f05da1a9ec4d",  
"id": 1  
}' http://192.168.7.101/zabbix/api_jsonrpc.php | python -m json.tool
```

6.2.2：通过API添加主机-非proxy模式

可以通过一个模板关联多个模板给同一主机同时添加多个模板

```
root@node1:~# curl -s -X POST -H 'Content-Type:application/json' -d '  
> {  
>     "jsonrpc": "2.0",  
>     "method": "host.create",  
>     "params": {  
>         "host": "192.168.172.95",  
>         "name": "192.168.172.95",  
>         "interfaces": [  
>             {  
>                 "type": 1,  
>                 "main": 1,  
>                 "useip": 1,  
>                 "ip": "192.168.172.95",  
>                 "dns": "",  
>                 "port": "10050"  
>             }  
>         ],  
>         "groups": [  
>             {  
>                 "groupid": "2"  
>             }  
>         ],  
>         "templates": [  
>             {  
>                 "templateid": "10001"  
>             }  
>         ]  
>     },  
>     "auth": "8847f3273e4e0682331579cf9f4ee8a2",  
>     "id": 1  
> }' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool  
{  
    "id": 1,  
    "jsonrpc": "2.0",  
    "result": {  
        "hostids": [  
            "10279"  
        ]  
    }  
}
```

```
}
```

#server端查看

- 添加成功

名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性	agent 加密	信息
proxy_active: 192.168.172.94	应用集 14	监控项 96	触发器 46	图形 17	自动发现 2	Web监测 192.168.172.94:	10050	memcache-status, redis-status, Template App Zabbix Proxy, Template Linux TCP_CONN Status, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX	SNMP JMX IPMI	无
192.168.172.95	应用集 10	监控项 34	触发器 15	图形 5	自动发现 2	Web监测 192.168.172.95:	10050	Template OS Linux (Template App Zabbix Agent)	已启用	ZBX	SNMP JMX IPMI	无
192.168.172.141	应用集 3	监控项 20	触发器 9	图形 3	自动发现 1	Web监测 192.168.172.141:	10050	Template Module Generic SNMPv2 (Template Module ICMP Ping), Template Module Interfaces SNMPV2	已启用	ZBX	SNMP JMX IPMI	无
proxy_active: 192.168.172.142	应用集 17	监控项 246	触发器 28	图形 58	自动发现 2	Web监测 192.168.172.142:	10050	linux_basic, mysql_master_slave, NGINX_Check_Status, process_port_check, Template OS Linux (Template App Zabbix Agent), Template Percona MySQL Server	已启用	ZBX	SNMP JMX IPMI	无
Zabbix server	应用集 11	监控项 93	触发器 52	图形 15	自动发现 2	Web监测 192.168.172.91:	10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX	SNMP JMX IPMI	无

6.2.3 : 通过API添加主机-proxy模式

```
root@node1:~# curl -s -X POST -H 'Content-Type:application/json' -d '
> {
>     "jsonrpc": "2.0",
>     "method": "host.create",
>     "params": {
>         "host": "192.168.172.95",
>         "name": "192.168.172.95",
>         "proxy_hostid": "10264",
>         "interfaces": [
>             {
>                 "type": 1,
>                 "main": 1,
>                 "useip": 1,
>                 "ip": "192.168.172.95",
>                 "dns": "",
>                 "port": "10050"
>             }
>         ],
>         "groups": [
>             {
>                 "groupid": "2"
>             }
>         ],
>         "templates": [
>             {
>                 "templateid": "10264"
>             }
>         ]
>     }
> }
```

```

>             "templateid": "10001"
>         }
>     ]
> },
> "auth": "8847f3273e4e0682331579cf9f4ee8a2",
> "id": 1
> }' http://192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
{
    "id": 1,
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10280"
        ]
    }
}
#查询测试

```

名称	应用集	监控项	触发器	图形	自动发现	Web 监测	接口	模板	状态	可用性	agent 加密	信息
proxy_active: 192.168.172.94	应用集 14	监控项 96	触发器 46	图形 17	自动发现 现 2	Web 监测 192.168.172.94: memcache-status, redis-status, Template App Zabbix Proxy, Template Linux TCP_CONN Status, Template OS Linux (Template App Zabbix Agent)	10050		已启用	ZBX	SNMP JMX IPMI	无
proxy_active: 192.168.172.95	应用集 10	监控项 34	触发器 15	图形 5	自动发现 现 2	Web 监测 192.168.172.95: Template OS Linux (Template App Zabbix Agent)	10050		已启用	ZBX	SNMP JMX IPMI	无
192.168.172.141	应用集 3	监控项 20	触发器 9	图形 3	自动发现 现 1	Web 监测 192.168.172.141: Template Module Generic SNMPv2 (Template Module ICMP Ping), Template Module Interfaces SNMPv2	10050		已启用	ZBX	SNMP	无
proxy_active: 192.168.172.142	应用集 17	监控项 246	触发器 28	图形 58	自动发现 现 2	Web 监测 192.168.172.142: linux_basic, mysql_master_slave, NGINX_Check_Status, processes_port_check, Template OS Linux (Template App Zabbix Agent), Template Percona MySQL Server	10050		已启用	ZBX	SNMP JMX IPMI	无
Zabbix server	应用集 11	监控项 93	触发器 52	图形 15	自动发现 现 2	Web 监测 192.168.172.91: Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	10050		已启用	ZBX	SNMP JMX IPMI	无

6.2.4 : 使用脚本调用API快速批量添加

```

#!/bin/bash

IP="

192.168.172.94
192.168.172.95
"

for node_ip in ${IP};do
curl -s -X POST -H 'Content-Type:application/json' -d '
{
    "jsonrpc": "2.0",
    "method": "host.create",

```

```
"params": {
    "host": "'${node_ip}'",
    "name": "'${node_ip}'",
    "proxy_hostid": "10264",
    "interfaces": [
        {
            "type": 1,
            "main": 1,
            "useip": 1,
            "ip": "'${node_ip}'",
            "dns": "",
            "port": "10050"
        }
    ],
    "groups": [
        {
            "groupid": "15"
        }
    ],
    "templates": [
        {
            "templateid": "10268"
        }
    ]
},
"auth": "8847f3273e4e0682331579cf9f4ee8a2",
"id": 1
}' http://192.168.192.168.172.91/zabbix/api_jsonrpc.php | python -m json.tool
done
```