

zabbix简介

Zabbix 是一个基于 WEB 界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。zabbix 能监视各种网络参数,保证服务器系统的安全运营;并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题。

优点:

开源,无软件成本投入

Server 对设备性能要求低

支持设备多,自带多种监控模板

支持分布式集中管理,有自动发现功能,可以实现自动化监控

开放式接口,扩展性强,插件编写容易

当监控的 item 比较多服务器队列比较大时可以采用被动状态,被监控客户端主动从 server 端去下载需要监控的 item 然后取数据上传到 server 端。这种方式对服务器的负载比较小。

Api 的支持,方便与其他系统结合

缺点:

需在被监控客户端上安装 agent,所有数据都存在数据库里,产生的数据据很大,瓶颈主要在数据库。

区别:

//na gio s: 没有图形,就算用第三方插件,图也不好看。也支持分布式监控,但是没有人去做,可以说不支持分布式监控。所有的配置都是修改配置文件实现的。

1、为什么要监控

在需要的时刻,提前提醒我们服务器出问题了

当出问题之后,可以找到问题的根源

网站/服务器 的可用性

1.1.1、网站可用性

在软件系统的高可靠性(也称为可用性,英文描述为HA, High Available)里有个衡量其可靠性的标准——X个9,这个X是代表数字3~5。X个9表示在软件系统1年时间的使用过程中,系统可以正常使用时间与总时间(1年)之比,我们通过下面的计算来感受下X个9在不同级别的可靠性差异。

1个9: $(1-90\%)*365=36.5$ 天,表示该软件系统在连续运行1年时间里最多可能的业务中断时间是36.5天

2个9: $(1-99\%)*365=3.65$ 天,表示该软件系统在连续运行1年时间里最多可能的业务中断时间是3.65天

3个9: $(1-99.9\%)*365*24=8.76$ 小时,表示该软件系统在连续运行1年时间里最多可能的业务中断时间是8.76小时。

4个9: $(1-99.99\%)*365*24=0.876$ 小时=52.6分钟,表示该软件系统在连续运行1年时间里最多可能的业务中断时间是52.6分钟。

5个9: $(1-99.999\%)*365*24*60=5.26$ 分钟,表示该软件系统在连续运行1年时间里最多可能的业务中断时间是5.26分钟。

6个9: $(1-99.9999\%)*365*24*60*60=31$ 秒,示该软件系统在连续运行1年时间里最多可能的业务中断时间是31秒

1.2、监控什么东西？

监控一切需要监控的东西，只要能够想到，能够用命令实现的都能用来监控

1.2、监控范畴

硬件监控:	(1)通过远程控制卡: Dell的iDRAC, HP的iLO和IBM的IMM等 (2)使用IPMI来完成物理设备的监控工作。通常必须要监控的就是温度、硬盘故障等 (3)路由器, 交换机(端口, 光衰, 日志), 打印机, Windows等
系统监控:	cpu, 内存, 硬盘使用率, 硬盘IO, 系统负载, 进程数
服务监控:	apache, nginx, php-fpm, mysql, memcache, redis, tomcat, JVM, TCP连接数
性能监控:	网站性能, 服务器性能, 数据库性能, 存储性能
日志监控:	系统会产生系统日志, 应用程序会有应用的访问日志、错误日志, 服务有运行日志等, 可以使用ELK来进行日志监控
安全监控	(1)Nginx+Lua编写了一个WAF通过kibana可以图形化的展示不同的攻击类型的统计。 (2)用户登录数, passwd文件变化, 本地所有文件改动
网络监控	端口, web(URL), DB, ping包, IDC带宽网络流量, 网络流出流出速率, 网络入流量, 网络出流量, 网络使用率, SMTP, POP3

1.3.2 监控硬件

查看硬件的温度/风扇转速，电脑有鲁大师，服务器就有ipmitool。

使用ipmitool实现对服务器的命令行远程管理

yum -y install OpenIPMI ipmitool #->IPMI在物理机可以成功，虚拟机不行

```
[root@KVM ~]# ipmitool sdr type Temperature
```

```
Temp      | 01h | ns | 3.1 | Disabled
Temp      | 02h | ns | 3.2 | Disabled
Temp      | 05h | ns | 10.1 | Disabled
Temp      | 06h | ns | 10.2 | Disabled
Ambient Temp  | 0Eh | ok | 7.1 | 22 degrees C
Planar Temp  | 0Fh | ns | 7.1 | Disabled
IOH THERMTRIP | 5Dh | ns | 7.1 | Disabled
CPU Temp Interf | 76h | ns | 7.1 | Disabled
Temp      | 0Ah | ns | 8.1 | Disabled
Temp      | 0Bh | ns | 8.1 | Disabled
Temp      | 0Ch | ns | 8.1 | Disabled
```

1.3.3 查看cpu相关

lscpu、uptime、top、htop、vmstat、mpstat

其中htop需要安装，安装依赖与epel源。

```
[root@alan ~]# lscpu
```

```
Architecture:    x86_64
CPU op-mode(s):  32-bit, 64-bit
Byte Order:      Little Endian
```

CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 2
座: 2
NUMA 节点: 1
厂商 ID: GenuineIntel
CPU 系列: 6
型号: 142
型号名称: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
步进: 10
CPU MHz: 1799.327
BogoMIPS: 3600.00
超管理器厂商: VMware
虚拟化类型: 完全
L1d 缓存: 32K
L1i 缓存: 32K
L2 缓存: 256K
L3 缓存: 6144K
NUMA 节点0 CPU: 0-3

1.3.4 内存够不够用

```
free
[root@alan ~]$free -h
```

	total	used	free	shared	buffers	cached	
Mem:	996M	867M	128M	712K	145M	450M	
-/+ buffers/cache:		271M	725M				
Swap:	1.0G	0B	1.0G				

1.3.5 磁盘剩多少，写的快不快

```
df、dd、iotop
[root@alan ~]#df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vda1	40G	24G	15G	62%	/
tmpfs	499M	20K	499M	1%	/dev/shm
/dev/vdb1	20G	4.4G	15G	24%	/data

1.3.6 监控网络

iftop nethogs
iftop 监控主机间流量 -i 指定监控网卡
nethogs 监控进程流量

1.4、Zabbix 监控系统监控对象:

数据库: MySQL,MariaDB,Oracle,SQL Server
应用软件: Nginx,Apache,PHP,Tomcat agent

集群： LVS,Keepalived,HAproxy,RHCS,F5
虚拟化： VMware,KVM,XEN
操作系统： Linux,Unix,Windows性能参数

硬件： 服务器，存储，网络设备
网络： 网络环境（内网环境，外网环境）

IPMI
SNMP

1.5、Zabbix监控方式：

被动模式
主动模式

面试题：

主动、被动模式都是相对于agent（被监控端-客户端）来说的。agent主动发送数据就是主动模式；agent等待server的请求，再发送数据就是被动模式。因为主动模式可以有效减轻zabbix server压力，需要监控的东西很多时一定要把监控模式更改为主动监控

zabbix--安装

安装zabbix

server（监控端）端安装配置：

永久关闭防火墙和selinux

1.Install and configure Zabbix server

Zabbix服务器的安装和配置

网络安装

2.安装zabbix的yum源：

Zabbix源： # rpm -ivh http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/zabbix-release-3.4-2.el7.noarch.rpm

EPEL源： #wget <https://mirrors.aliyun.com/repo/epel-7.repo>

阿里源： #curl -o /etc/yum.repos.d/Alibaba.repo http://mirrors.aliyun.com/repo/Centos-7.repo

3.Install Zabbix server, frontend, agent

安装Zabbix服务器，前端，代理

```
# yum -y install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

修改服务器端的agent配置文件}
agent（代理） ---被监控的机器

本地安装

安装mariadb： yum install -y mariadb mariadb-server

```
#systemctl restart mariadb
```

```
#mysql_secure_installation
```

1.回车，如果不行，输入原来的数据库root密码然后回车

2.按Y（是否设置root密码）

newpassword： 123

3.

4.

.....全程回车

4.Create initial database

创建初始数据库

登陆mariadb

```
# mysql -u root -p'123'
```

创建一个zabbix数据库并且设置字符集：

```
mysql> create database zabbix character set utf8 collate utf8_bin;
```

创建一个用户zabbix并设置权限允许该用户访问的库和密码：

```
mysql> grant all privileges on zabbix.* to zabbix@'%' identified by 'zabbix';
```

设置密码 -----给用户

```
mysql> flush privileges;
```

-----刷新授权表

```
mysql> quit;
```

-----退出

5.Import initial schema and data. You will be prompted to enter your newly created password.

导入的初始架构和数据。将提示您输入新创建的密码

```
mysql -u zabbix -p zabbix < /usr/share/doc/zabbix-server-mysql*/create.sql.gz
```

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix //
```

导入数据到zabbix数据库

输入密码：zabbix

zcat：看所有包的内容但是不会解压缩。

注意：

vim /etc/zabbix/web/zabbix.conf.php ----里面有默认的zabbix密码，如果设置密码失败，先把这个文件里面的密码改掉

错误：不支持的数据库类型。

解决办法：1.重新导入数据，并且删除原有的授权账户，和数据库。

2./etc/zabbix/web/zabbix.conf.php,是否有该配置文件。如果没有，复制一份对应的zabbix.conf.php文件。

6.Configure the database for Zabbix server

zabbix服务器端配置

```
# vim /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix -----zabbix mysql里面的用户的密码
```

这里的配置文件打开后：如果有的就不用添加了。如果是注释了的也不用取消注释。直接把内容复制粘贴过去就可以了。

7.Start Zabbix server and agent processes

Zabbix服务器和代理服务器进程启动，设置开机启动

```
# systemctl start zabbix-server zabbix-agent httpd
# systemctl enable zabbix-server zabbix-agent httpd
```

8.Configure PHP for Zabbix frontend

配置php zabbix 前端

Apache configuration file for Zabbix frontend is located in /etc/httpd/conf.d/zabbix.conf. Some PHP settings are already configured. But it's necessary to uncomment the "date.timezone" setting and set the right timezone for you.

Apache的配置文件为Zabbix前端位于/etc/ httpd / conf.d/zabbix.conf一些PHP设置已经配置好了。但是有必要取消"日期。时区"设置正确的时区。

```
# vim /etc/httpd/conf.d/zabbix.conf
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
```

```
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value always_populate_raw_post_data -1
# php_value date.timezone Europe/Riga
php_value date.timezone Asia/Shanghai -----时区设置为亚洲上海
```

这个配置文件打开基本不用动，只需要把时区的这一行复制或者修改就可以了。

9.重新启动httpd：

```
# systemctl restart zabbix-server zabbix-agent httpd mariadb (mysqld)
```

```
#systemctl restart zabbix-server zabbix-agent httpd mariadb
```

在浏览器访问ip/zabbix：

并且进行设置：就第一步输入mysql密码。然后一直下一步。

最后登陆：

username: Admin

password: zabbix

agent（被监控端）端安装配置：

永久关闭防火墙和selinux

1.安装zabbix的yum源，epel源，阿里原或者163原：

```
# rpm -ivh http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/zabbix-
release-3.4-2.el7.noarch.rpm
#wget https://mirrors.aliyun.com/repo/epel-7.repo
#curl -o /etc/yum.repos.d/Ali.repo http://mirrors.aliyun.com/repo/Centos-7.repo
```

2.安装zabbix-agent：

```
[root@web1 ~]# yum -y install zabbix-agent
```

3.编辑配置文件修改：

添加服务器端（监控端）的ip地址：

```
[root@web1 ~]# vim /etc/zabbix/zabbix_agentd.conf
```

Server=192.168.122.206,192.168.122.207 被动模式 zabbix-server-ip 可以添加多个ip，可以被多台机器监控

ServerActive=192.168.122.206,192.168.122.207 主动模式 zabbix-server-ip

Hostname=web1 -----被监控机器的主机名称

UnsafeUserParameters=1 ---添加是否限制用户自定义 keys 使用特殊字符

4.启动服务并设置开机启动：

```
[root@web1 ~]#
```



```
[root@web1 ~]# systemctl enable zabbix-agent
```

查看端口：

```
[root@web1 ~]# ss -anlp |grep :10050
```

如果是英文的监测图形不会出现乱码，如果是 `systemctl start zabbix-agent` 中文的会出现乱码，修改如下：

只需要将 `/usr/share/zabbix/fonts/graphfont.ttf` 替换成合适的字体文件即可。可去 windows 下找自己喜欢的字体文件

如果有字体文件：

```
cp 字体文件 /usr/share/zabbix/fonts/graphfont.ttf
```

注意：把新字体文件改名成原来字体文件的名字，不然需要修改如下配置文件，方法如下：

```
# vim /usr/share/zabbix/include/defines.inc.php
```

将字体名称修改为 `font` 目录下的文件名，省去 `ttf` 后缀即可。

```
('ZBX_FONT_NAME', 'DejaVuSans');          #修改前
```

```
('ZBX_GRAPH_FONT_NAME', 'DejaVuSans');    #修改前
```

```
('ZBX_FONT_NAME', 'SIMKAI');              #修改后
```

```
('ZBX_GRAPH_FONT_NAME', 'SIMKAI');        #修改后
```

以上的配置是修改单台主机的图形输出。如果不想一台台机器的去设置,可以直接修改模板的图形输出配置就可以了。

利用命令获取数据：

```
zabbix_get -s 服务（客户）端ip地址 -p 端口 -k 键值名
```

端口：10050

Zabbix--WEB配置

一、Zabbix部署WEB界面

1.URL地址：zabbix-server主机IP/zabbix

例如：192.168.1.129/zabbix



在检测信息时，可查看具体的报错信息进行不同的解决

2.出现下方界面，全为OK则继续。



3. 选择mysql数据库给与对应的数据库端口，然后输入数据库密码即可

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

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

localhost

Database port

3306

0 - use default port

Database name

zabbix

User

zabbix

Password

Back

Next step

4.host与port不需要修改，name自定义

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Host

localhost

Port

10051

Name

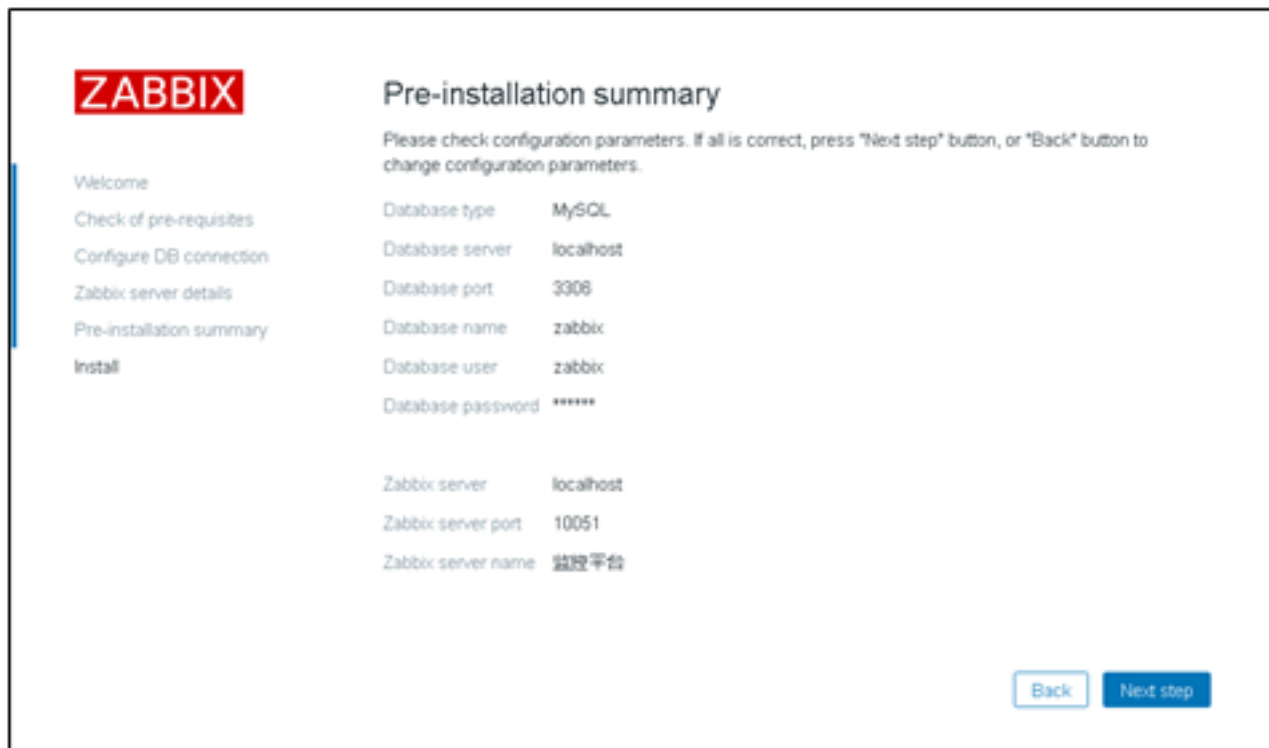
监控平台

Back

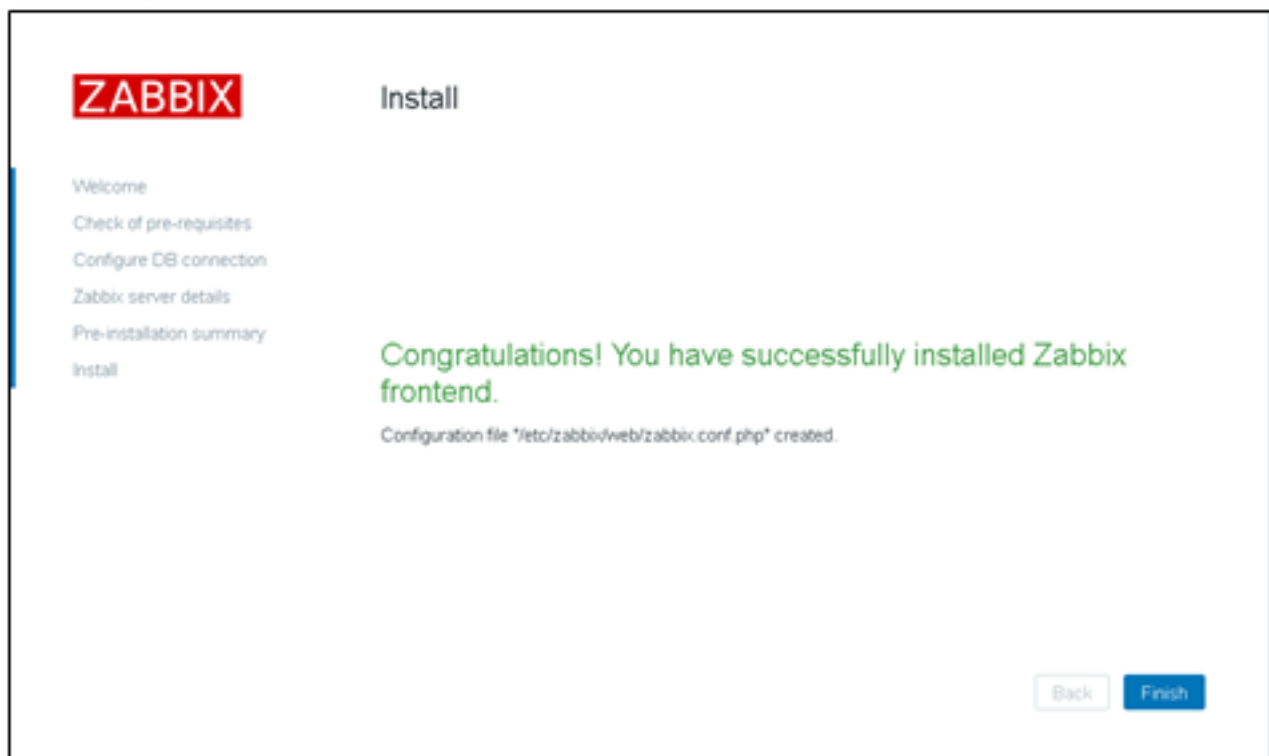
Next step

5.确认信息,正确点击下一步

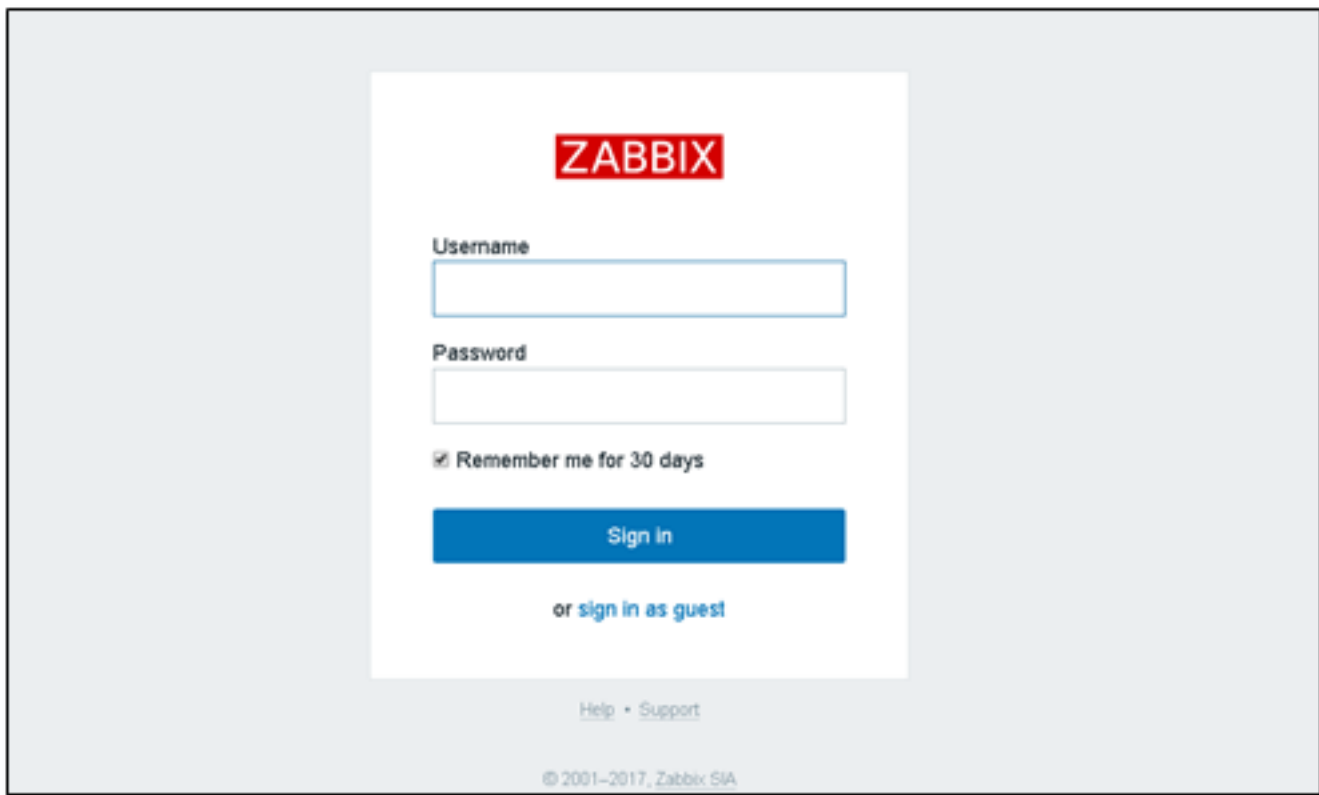
10/30



6.安装完成、点击finsh



7.进入登陆界面 账号Admin密码zabbix 注意A大写



二、Zabbix添加监控信息。

1.监控概念

下面介绍一些**Zabbix**常用术语的含义。

主机 (host)

- 一台你想监控的网络设备，用IP或域名表示

主机组 (host group)

- 主机的逻辑组；它包含主机和模板。一个主机组里的主机和模板之间并没有任何直接的关联。通常在给不同用户组的主机分配权限时候使用主机组。

监控项 (item)

- 你想要接收的主机的特定数据，一个度量数据。

触发器 (trigger)

一个被用于定义问题阈值和“评估”监控项接收到的数据的逻辑表达式。

当接收到的数据高于阈值时，触发器从“OK”变成“Problem”状态。当接收到的数据低于阈值时，触发器保留/返回一个“OK”的状态。

事件 (event)

单次发生的需要注意的事情，例如触发器状态改变或发现有监控代理自动注册

异常 (problem)

一个处在“异常”状态的触发器

动作 (action)

一个对事件做出反应的预定义的操作。

一个动作由操作(例如发出通知)和条件(当时操作正在发生)组成

升级 (escalation)

一个在动作内执行操作的自定义场景；发送通知/执行远程命令的序列

媒介 (media)

发送告警通知的手段；告警通知的途径

通知 (notification)

利用已选择的媒体途径把跟事件相关的信息发送给用户

远程命令 (**remote command**)

一个预定义好的，满足一些条件的情况下，可以在被监控主机上自动执行的命令

模版 (**template**)

一组可以被应用到一个或多个主机上的实体（监控项，触发器，图形，聚合图形，应用，LLD，Web场景）的集合

模版的任务就是加快对主机监控任务的实施；也可以使监控任务的批量修改更简单。模版是直接关联到每台单独的主机上。

应用 (**application**)

一组监控项组成的逻辑分组

web 场景 (**web scenario**)

利用一个或多个HTTP请求来检查网站的可用性

前端 (**frontend**)

Zabbix提供的web界面

Zabbix API

Zabbix API允许你使用JSON RPC协议来创建、更新和获取Zabbix对象（如主机、监控项、图形和其他）信息或者执行任何其他自定义的任务

Zabbix server

Zabbix软件实现监控的核心程序，主要功能是与Zabbix proxies和Agents进行交互、触发器计算、发送告警通知；并将数据集中保存等

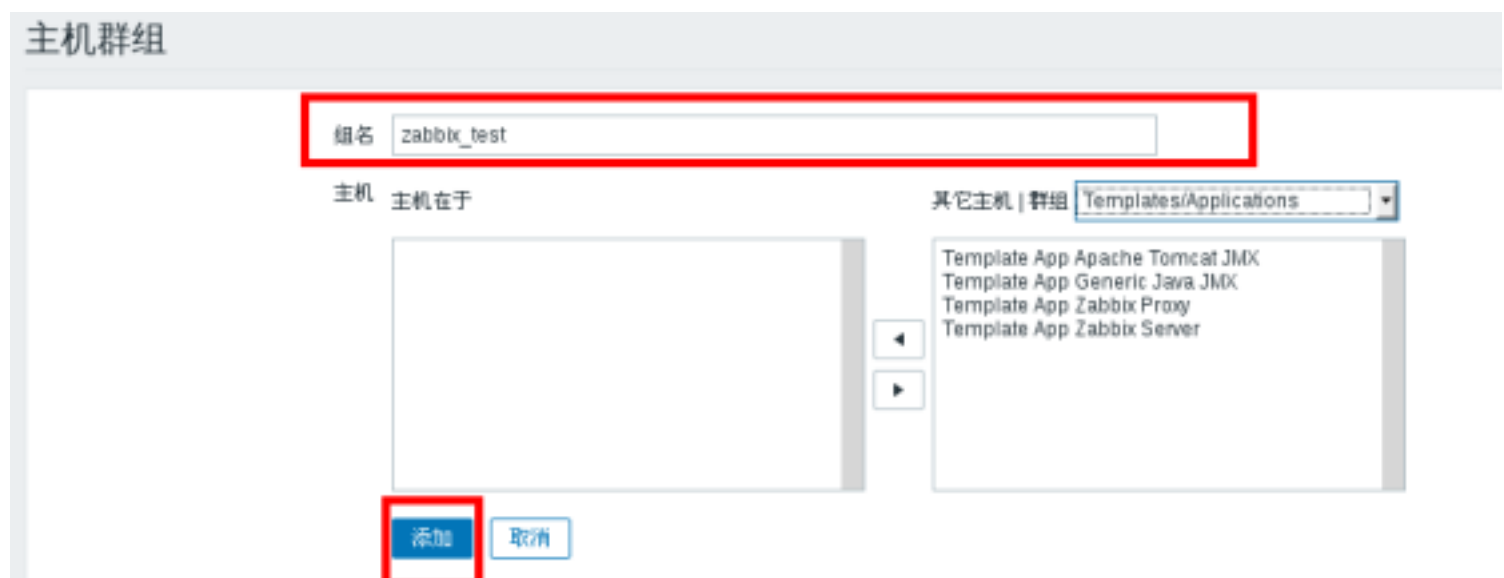
Zabbix agent

一个部署在监控对象上的，能够主动监控本地资源和应用的程序

Zabbix proxy

一个帮助Zabbix Server收集数据，分担Zabbix Server的负载的程序

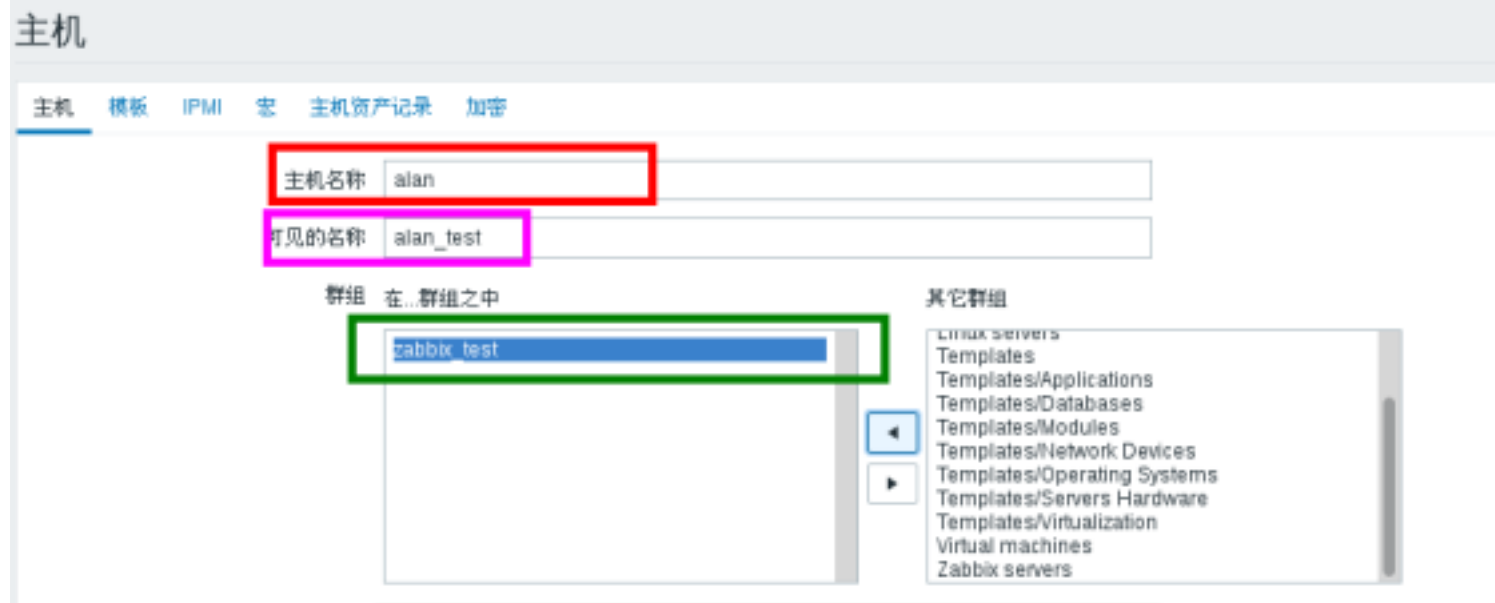
2.1、添加主机群组



2.2、添加主机



点击'创建主机'

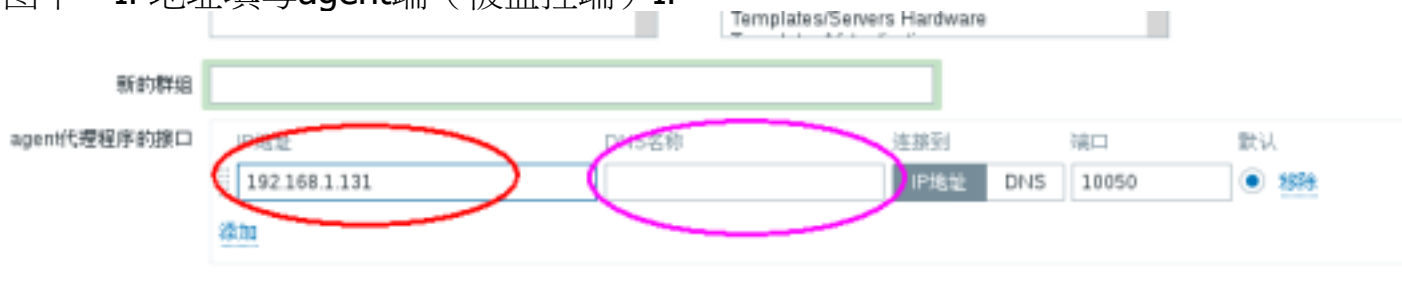


主机名称：要与主机名相同，这是zabbix server程序用的

可见名称：显示在zabbix网页上的，给我们看的

在...群组之中：选择刚刚创建的主机群组。

下图中，IP地址填写agent端（被监控端）IP。



修改后，要将下面的已启用要勾上。



添加完成就有了管理机的监控主机

2.3、添加模板



点击'选择'--->找到Template OS Linux---->添加---->更新。

3、添加自定义监控

3.1.1、监控需求：服务器登陆人数不能超过三人，超过三人则报警

3.1.2、预备知识

(在server端安装测试工具)

```
#yum -y install zabbix-get
```

测试自定义key能被server和agent认可

正确的key

```
[root@alan ~]# zabbix_get -s 192.168.1.131 -p 10050 -k "system.uname"
```

```
Linux web1 3.10.0-514.el7.x86_64 #1 SMP Tue Nov 22 16:42:41 UTC 2016 x86_64
```

没有登记的，自定义的key

```
[root@alan~]# zabbix_get -s 192.168.1.131 -p 10050 -k "login-user"
```

```
ZBX_NOTSUPPORTED: Unsupported item key.
```

写错的key

```
[root@alan ~]# zabbix_get -s 192.168.1.131 -p 10050 -k "system.uname1"
```

```
ZBX_NOTSUPPORTED: Unsupported item key.
```

3.2.1、自定义KEY---登陆用户监测（以下操作在agent端进行）

语法：`UserParameter=<key>,<shell command>`

例子：

```
UserParameter=login-user,who|wc -l
```

```
UserParameter=login-user,/bin/sh /server/scripts/login.sh
```

```
[root@web1 ~]# cd /etc/zabbix/zabbix_agentd.d/
```

```
[root@web1 zabbix_agentd.d]# vim userparameter_login.conf
```

```
UserParameter=login-user,who|wc -l
```

```
UserParameter=login-user2,who|wc -l
```

修改完成后重启zabbix-agent服务

注意：key名字要唯一，多个key以行为分割

3.2.2、自定义KEY---内存使用监测（以下操作在agent端进行）

```
[root@web1 ~]# cd /etc/zabbix/zabbix_agentd.d/
[root@alan zabbix_agentd.d]# vim userparameter_mem.conf
UserParameter=mem.free,/usr/bin/free -m | awk '/^Mem/{print $4}'
UserParameter=mem.all,/usr/bin/free -h
```

修改完成后重启zabbix-agent服务

```
[root@cache01 zabbix_agentd.d]# systemctl restart zabbix-agent.service
```

3.2.3、在server端进行get测试

```
[root@alan~]# zabbix_get -s 192.168.1.131 -p 10050 -k "mem.free"
[root@alan ~]# zabbix_get -s 192.168.1.131 -p 10050 -k "mem.all"
[root@alan ~]# zabbix_get -s 192.168.1.131 -p 10050 -k "login-user"
[root@alan ~]# zabbix_get -s 192.168.1.131 -p 10050 -k "login-user2"
```

3.3、WEB界面添加监控

① 创建模板

模板必须定义的功能：模板名称、应用集、监控项、触发器、图形(曲线图)。

所有模板 / Template Login User 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

配置 >> 模板 >> 创建模板

模板

所有模板 / 内存检测 (192.1681.131) 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

模板 链接的模板 宏

模板名称 Templates Mem Free

可见的名称 内存检测 (192.1681.131)

群组 在 群组之中

Templates

其它群组

- Discovered hosts
- Hypervisors
- Linux servers
- Templates/Applications
- Templates/Databases
- Templates/Modules

点击添加，即可创建出来模板

② 创建应用集

应用集类似(目录/文件夹)，其作用是给监控项分类。

点击 应用集 >> 创建应用集

名称随便填写，点解更新---创建成功。

③ 创建监控项

[监控项 >> 创建监控项](#)

监控项

键值 -- key：即前面出创建的mem.free。

注意：创建监控项的时候，注意选择上应用集，即之前创建的内存。

④ 创建触发器

触发器的作用：当监控项获取到的值达到一定条件时就触发报警

(根据需求创建)

[触发器 >> 创建触发器](#)

创建触发器：自定义名称，该名称是报警时显示的名称。

表达式：点击右边的添加，选择表达式。

严重性：自定义。

触发器

所有模板 / 内存检测 (192.1681.131) 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

内存小于100报警

严重性

未分类

信息

警告

一般严重

严重

灾难

表达式

{Templates Mem Free:mem.free.last()}<100

添加

表达式的定义↓，选择之前创建的监控项

last() - Last(most recent) T value
结果 < 200

监控项

内存检测 (192.1681.131) : 内存使用情况

选择

功能

最新的T值 < N

最后一个 (T)

时间

间隔(秒)

时间

N

100

插入

取消

添加完成，能够在触发器中看到添加的情况

<input type="checkbox"/>	严重性	名称 ▲	表达式	状态
<input type="checkbox"/>	严重	内存小于100报警	{Templates Mem Free:mem.free.last()}<100	已启用

显示 已自动发现的 1 中的 1

⑤ 创建图形
以图形(曲线、饼图)的方式展示出来监控信息
[图形 >> 创建图形](#)
名称自定义，关联上监控项。

所有模板 / 内存检测 (192.168.1.131) 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

宽

高

图形类别

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

纵轴Y最大值MIN

纵轴最大值

监控项

名称	功能	绘图风格	纵轴Y侧	颜色	动作
1: 内存检测 (192.168.1.131) : 内存使用情况	平均	线	左侧	1A7C11	移除

[添加](#)

点击更新---创建成功。

⑥ 主机关联模板

配置 >> 主机

一个主机可以关联多个模板

主机

名称 DNS IP地址 端口

<input type="checkbox"/>	名称	应用集	监控项	触发器	图形	自动发现	Web检测	接口	模板	状态	可用性	agent加密	信息		
<input type="checkbox"/>	aler_test	应用集 11	监控项 45	触发器 20	图形 9	自动发现 2	Web检测	192.168.1.131:10050	Template OS Linux (Template App Zabbix Agent)	内存检测 (192.168.1.131)	已启用	2000	2000	2000	2000
<input type="checkbox"/>	Zabbix server	应用集 11	监控项 90	触发器 40	图形 11	自动发现 2	Web检测	127.0.0.1:10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	停用的	2000	2000	2000	2000	

如果是英文的监测图形不会出现乱码，如果是中文的会出现乱码，修改如下：

只需要将/usr/share/zabbix/fonts/graphfont.ttf 替换成合适的字体文件即可。可去 windows下找自己喜欢的字体文件
如果有字体文件：

cp 字体文件 /usr/share/zabbix/fonts/graphfont.ttf

注意：把新字体文件改名成原来字体文件的名字，不然需要修改如下配置文件，方法如

下：

```
# vim /usr/share/zabbix/include/defines.inc.php
将字体名称修改为front目录下的文件名，省去ttf后缀即可。
('ZBX_FONT_NAME', 'DejaVuSans');          #修改前
('ZBX_GRAPH_FONT_NAME', 'DejaVuSans');     #修改前

('ZBX_FONT_NAME', 'SIMKAI');               #修改后
('ZBX_GRAPH_FONT_NAME', 'SIMKAI');         #修改后
```

以上的配置是修改单台主机的图形输出。如果不想一台台机器的去设置,可以直接修改模版的图形输出配置就可以了。

利用命令获取数据：

zabbix_get -s 服务（客户）端ip地址 -p 端口 -k 键值名

端口：10050

新微信报警

```
# ping baidu.com
64 bytes from 180.97.33.107 (180.97.33.107): icmp_seq=1 ttl=54 time=12.9 ms
ttl 是linux
第三方平台报警：
lbJ9GgejAki979IVN6puEYImqRUPtXqoHK8cKvxwWHI
1000002
2
wwf503001d9f69a132
ifconfig enp3s0:1 192.168.1.36 //添加网卡
//添加网卡
企业号的申请和配置方法可参考：http://blog.51cto.com/wuhf2015/1688614
```

1.1、创建群组并获取部门ID



1.2.1、创建应用



12.2、根据提示填写信息,获得下方信息。



1.3、点击“我的企业”获取Corpid（企业ID）

企业信息

权限管理

聊天管理

通讯录管理

申请加入设置

安全与保密

设置

企业信息

企业logo

腾帆电子科技

推荐尺寸702*180

验证主体信息

验证后可提高使用人数等

企业简称

腾帆电子科技 修改

企业地址

添加

联系电话

添加

企业域名

添加

企业成员

3 个成员 (未加入 2人)

企业部门

5 个部门

已使用/人数上限

3/200 申请扩容

企业成员

3 个成员 (未加入 2人)

企业部门

5 个部门

已使用/人数上限

3/200 申请扩容

发票抬头

添加 为企业成员配置增值税发票抬头 ⓘ

行业类型

互联网和相关服务 修改

人员规模

1-50人 修改

创建时间

2018年7月24日

企业ID

wwbc27916706540977

1.4.、使用企业微信的API接口测试一下是否可以建立连接。

<http://work.weixin.qq.com/api/devtools/devtool.php>

如下图所示：

一、接口类型

二、接口列表 方法: GET

三、参数列表

*corpid wwbc27916706540977

公司的id

*corpsecret 6cMYoDUUdOiljawS487dLr4SNp1Gku_nQTq22uV9gNM

企业应用secret

[检查问题](#)

建立连接: 获取AccessToken

请求地址: https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=wwbc27916706540977&corpsecret=6cMYoDUUdOiljawS487dLr4SNp1Gku_nQTq22uV9gNM

返回结果: HTTP/1.1 200 OK

Connection: close

Error-Code: 0

Error-Msg: ok

Content-Type: application/json; charset=UTF-8

Content-Length: 277

```
{
  "errcode": 0,
  "errmsg": "ok",
  "access_token": "dHGL2F9sYoUFNPZ4-O6uV694eqwKjSGMHTE-3lo3hMVjepkR_g7gkCfjuxBx6f1lLh9irzZaacy9yfr6021ZykKdppKBqCh0b-_Hx5Oj-MIT1kKaaTFaQP--MaOwFiHj3MPa58nP_H5UgbPBmZY_TcA_mOambmO0b5oQjiBzY1_taTyh9HPwOF7slqcorJ0kQSnirmLNkz7VHLBlqm7qlyA",
  "expires_in": 7200
}
```

二、zabbix端操作。

2.0、在server端创建自定义报警脚本（修改红色字体部分为自己的信息）。

#vim /etc/zabbix/zabbix_server.conf

找到：

AlertScriptsPath=/usr/lib/zabbix/alertscripts //脚本存放位置

#cd /usr/lib/zabbix/alertscripts

#vim wechat.sh

如下作为参考：（给与脚本最高权限）

#!/bin/bash

#set -x

#尚关科技 贺洋伟

CorpID="wwbc27916706540977"

#我的企业下面的CorpID

```

Secret="6cMYoDUUdOiLjawS487dLr4SNp1Gku_nQTq22uV9gNM" #创建的应用那
有Secret
GURL="https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=$CorpID&corpsecret=
$Secret"
Token=$(/usr/bin/curl -s -G $GURL |awk -F": '{print $4}'|awk -F\" '{print $2}')
#echo $Token
PURL="https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token=$Token"

```

```

function body(){
    local int agentid=1000002 #改为AgentId 在创建的应用那里看
    local UserID=$1          #发送的用户位于$1的字符串
    local PartyID=2           #第一步看的通讯录中的部门ID
    local Msg=$(echo "$@" | cut -d" " -f3-)
    printf '{\n'
    printf '\t"touser": ""'$UserID'"",\n"
    printf '\t"toparty": ""'$PartyID'"",\n"
    printf '\t"msgtype": "text",\n'
    printf '\t"agentid": ""'$agentid'"",\n"
    printf '\t"text": {\n'
    printf '\t\t"content": ""'$Msg'"",\n"
    printf '\t},\n'
    printf '\t"safe": "0"\n'
    printf '}\n'
}
/usr/bin/curl --data-ascii "$ (body $1 $2 $3)" $PURL

```

2.1、告警测试。

bash wechat.sh alan 告警测试 告警测试信息
注：

alan：收件人
告警测试：标题
告警测试信息：内容



2.2、测试没问题，开始配置Zabbix--WEB。

2.2.1、配置报警媒介：

>>管理>>报警媒介>>创建报警媒介。



具体配置看下一图：

报警媒介类型 选项

名称

类型

脚本名称

脚本参数

参数	动作
<input data-bbox="432 1771 1449 1816" type="text" value="{ALERT.SENDTO}"/>	移除
<input data-bbox="432 1845 1449 1890" type="text" value="{ALERT.SUBJECT}"/>	移除
<input data-bbox="432 1919 1449 1964" type="text" value="{ALERT.MESSAGE}"/>	移除

{ALERT.SENDTO}
{ALERT.SUBJECT}
{ALERT.MESSAGE}

2.3、配置用户：

>>管理>>用户>>点击系统自带的用户。



<input type="checkbox"/>	别名 ▲	用户名第一部分	姓氏	用户类型	群组	是否在线?	登录	前端访问	调试模式	状
<input type="checkbox"/>	Admin	Zabbix	Administrator	超级管理员	Zabbix administrators	是 (2018-07-25 13:08:31)	正常	系统默认	停用的	已
<input type="checkbox"/>	guest			用户	Guests	不	正常	系统默认	停用的	已

具体配置下图：



Admin>>点击报警媒介>>权限>>更新添加

报警媒介

类型

微信报警

收件人

alan

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

☒

更新

取消

用户

用户

报警媒介

权限

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
	微信报警	alan	1-7,00:00-24:00	未信警一严灾	已启用	编辑 移除
	添加					

2.4、配置动作：
配置>>动作>>创建动作

动作

动作

操作

恢复操作

确认操作

名称 wechat

条件

标签

名称

A

维护状态 非在 维护

动作

移除

新的触发条件

触发器名称

似

添加

已启用



更新

克隆

删除

取消

点击触发器选择触发器
点击“操作”

动作

动作

操作

恢复操作

确认操作

默认操作步骤持续时间

1h

默认标题

故障:{TRIGGER.STATUS};服务器:{HOSTNAME1}发生:{TRIGGER.NAME}故障!

消息内容

告警时间:{EVENT.DATE}{EVENT.TIME}
告警等级:{TRIGGER.SEVERITY}
告警信息:{TRIGGER.NAME}
告警项目:{TRIGGER.KEY1}
问题详情:{ITEM.NAME}{ITEM.VALUE}
当前状态:{TRIGGER.STATUS}{ITEM.VALUE1}
事件标识:{EVENT.ID}

维护期间暂停操作



操作

步骤 细节

开始于 持续时间 动作

1 - 10 发送消息给用户: Admin (Zabbix Administrator) 通过 微信报警 立即地 60
发送消息给用户群组: Zabbix administrators 通过 微信报警

编辑 移除

新的

操作细节

步骤

1 · 10 (0 - 无穷大)

步骤持续时间

60 (0 - 使用默认)

操作类型

发送消息

发送到用户群组

用户群组

Zabbix administrators

添加

动作

移除

发送到用户

用户

Admin (Zabbix Administrator)

添加

动作

移除

仅送到

微信报警

消息内容

☒

条件

标签

A

名称

事件已确认 = 非确认

动作

移除

恢复操作，确认操作可以同上。

{TRIGGER.STATUS}

主机：{HOST.STATUS}

地址：{HOST.IP}

故障发生时间：{EVENT.DETE} {EVENT.TIME}

事件：{TRIGGER.NAME}

故障:{TRIGGER.STATUS},服务器:{HOSTNAME1}发生:{TRIGGER.NAME}故障!

告警等级:{TRIGGER.SEVERITY}

告警信息:{TRIGGER.NAME}

告警项目:{TRIGGER.KEY1}

问题详情:{ITEM.NAME}:{ITEM.VALUE}

当前状态:{TRIGGER.STATUS}:{ITEM.VALUE1}

事件标识:{EVENT.ID}

恢复:{TRIGGER.STATUS}, 服务器:{HOSTNAME1}:{TRIGGER.NAME}已恢复!

告警主机:{HOSTNAME1}

告警时间:{EVENT.DATE} {EVENT.TIME}

告警等级:{TRIGGER.SEVERITY}

告警信息:{TRIGGER.NAME}

告警项目:{TRIGGER.KEY1}

问题详情:{ITEM.NAME}:{ITEM.VALUE}

当前状态:{TRIGGER.STATUS}:{ITEM.VALUE1}
事件ID:{EVENT.ID}
