

## Zabbix

zabbix server: 中央情报局, 监控程序

zabbix agentd: 代理程序 间谍程序 采集--》os、设备、数据库、应用程序等信息

数据库: MySQL, 存储在这里

展示: web方式

zabbix proxy: 大公司用的多, 中层代理服务器, 最后香中央汇报

### 安装流程

zabbix的web端的操作原理

纯手工打造MySQL和NGINX的监控:

# Zabbix

[https://mirrors.aliyun.com/zabbix/zabbix/4.4/rhel/8/x86\\_64/](https://mirrors.aliyun.com/zabbix/zabbix/4.4/rhel/8/x86_64/)

## zabbix server: 中央情报局, 监控程序

- zabbix\_get: 监控服务器通知代理程序上报采集的信息
- zabbix\_server.conf:
- zabbix\_server.log

## zabbix agentd: 代理程序 间谍程序 采集--》os、设备、数据库、应用程序等信息

- zabbix\_agentd.conf
- zabbix\_agentd.log

## 数据库: MySQL, 存储在这里

### 展示: web方式

## zabbix proxy: 大公司用的多, 中层代理服务器, 最后香中央汇报

### 安装流程

- 服务器
  - 1、安装zabbix相关软件
  - 2、安装mariadb数据库

```
1 # yum install mariadb mariadb-server -y
2
3 # mysql -uroot -p# mysql -uroot -p
4 MariaDB [(none)]> create database zabbix character set utf8
  collate utf8_bin; 建库zabbix, 指定字符集为utf8
5 Query OK, 1 row affected (0.001 sec)
6
7 MariaDB [(none)]> show databases;
8 +-----+
9 | Database |
10 +-----+
```

```

11 | information_schema |
12 | mysql              |
13 | performance_schema |
14 | zabbix              |
15 +-----+
16 4 rows in set (0.001 sec)
17
18 MariaDB [(none)]> grant all privileges on zabbix.* to
    zabbix@localhost identified by 'zabbix123';
19 新建一个用户zabbix@localhost 密码为zabbix123 可以有所有的权限在zabbix
    这个库里的所有的表上
20 Query OK, 0 rows affected (0.001 sec)
21
22 MariaDB [(none)]> exit
23
24 # zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz |mysql
    -uzabbix -pzabbix123 zabbix

```

### ○ 3、关闭防火墙和selinux

```

1 [root@fengdeyong run]# vim /etc/sysconfig/selinux 永久关闭
2 SELINUX=disabled
3 [root@fengdeyong run]# setenforce 0 临时关闭
4 [root@fengdeyong run]# getenforce #查看selinux的状态，建议关闭
5 Permissive
6
7 # iptables -F
8 # systemctl disable firewalld
9 # service firewalld stop

```

### ○ 4、修改zabbix-server配置文件

```

1 在zabbix_server.conf中编辑数据库配置
2 # vi /etc/zabbix/zabbix_server.conf
3 DBHost=localhost
4 DBName=zabbix
5 DBUser=zabbix
6 DBPassword=zabbix123
7
8 启动Zabbix Server进程
9 service zabbix-server start
10

```

### ○ 5、修改php-fpm的配置文件：===》修改时区！（Asia/Shanghai）

```

1 # vim /etc/php-fpm.d/zabbix.conf
2 php_value[max_execution_time] = 300
3 php_value[memory_limit] = 128M
4 php_value[post_max_size] = 16M
5 php_value[upload_max_filesize] = 2M
6 php_value[max_input_time] = 300
7 php_value[max_input_vars] = 10000
8 php_value[date.timezone] = Asia/Shanghai 修改为Asia/Shanghai

```

- 6、启动所有的程序并且设置开机启动

```
1 # systemctl restart zabbix-server zabbix-agent httpd php-fpm
2
3 # systemctl enable zabbix-server zabbix-agent httpd php-fpm
```

- 客户机

- 1、安装zabbix agent
- 2、关闭防火墙和selinux

```
1 [root@fengdeyong run]# vim /etc/sysconfig/selinux 永久关闭
2 SELINUX=disabled
3 [root@fengdeyong run]# setenforce 0 临时关闭
4 [root@fengdeyong run]# getenforce #查看selinux的状态，建议关闭
5 Permissive
6
7 # iptables -F
8 # systemctl disable firewalld
9 # service firewalld stop
```

- 3、修改/etc/zabbix/zabbix\_agentd.conf配置文件

```
1 Server=192.168.111.132
2 ServerActive=192.168.111.132
```

- 4、启动和设置自启zabbix-agent服务

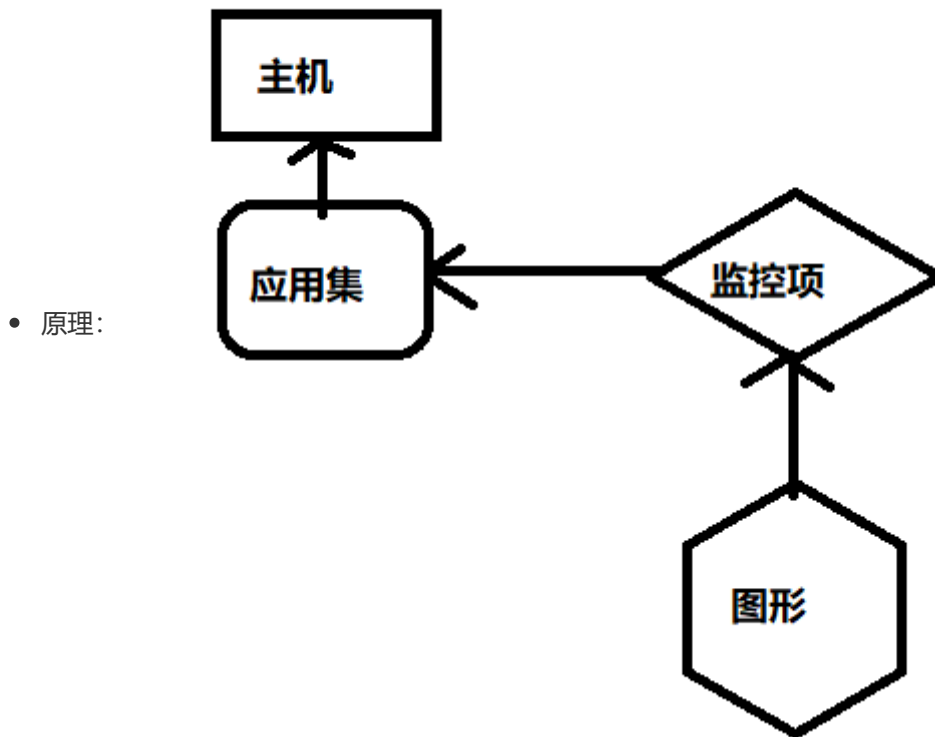
```
1 # service zabbix-agent restart 刷新服务
2 # systemctl enable zabbix-agent
```

- 5、服务器端测试 (zabbix-get)

```
1 # yum install zabbix-get
2
3 # zabbix_get -s 192.168.1.126 -p 10050 -k
  "system.cpu.load[all,avg1]"
```

## zabbix的web端的操作原理

## zabbix的web端原理



## 纯手工打造MySQL和NGINX的监控:

- 打造MySQL监控:

- 在服务器端/etc/zabbix/zabbix\_agentd.d下新建两个文件 (mysql\_status.sh, userparameter\_mysql.conf)

- ```
1 [root@zabbixclient1 zabbix_agentd.d]# cat userparameter_mysql.conf
2 UserParameter=mysql.status[*],/etc/zabbix/zabbix_agentd.d/mysql_status.sh $1
3 UserParameter=mysql.ping,/usr/bin/mysqladmin -utzk -p'tzk123456' -h localhost ping 2>/dev/null | grep -c alive
4 UserParameter=mysql.version,/usr/bin/mysql -V
5 UserParameter=mysql.bytes_sent,/etc/zabbix/zabbix_agentd.d/mysql_status.sh $1
```

- ```
1 [root@zabbixclient1 zabbix_agentd.d]# cat mysql_status.sh
2 #!/bin/bash
3 #Desc: zabbix 监控 MySQL 状态
4 #Date: 2020-3-19
5 #by: tzk
6
7 #主机
8 HOST="localhost"
9 #用户
10 USER="tzk"
11 #密码
12 PASSWORD="tzk123456"
13 #端口
14 PORT="3306"
```

```
15 #MySQL连接
16 CONNECTION="mysqladmin -h ${HOST} -u ${USER} -P ${PORT} -
p${PASSWORD}"
17
18 if [ $# -ne "1" ];then
19     echo "arg error!"
20 fi
21
22 case $1 in
23     Uptime)
24         result=`${CONNECTION} status 2>/dev/null |awk '{print $2}'`
25         echo $result
26         ;;
27     Questions)
28         result=`${CONNECTION} status 2>/dev/null |awk '{print $6}'`
29         echo $result
30         ;;
31     Com_update)
32         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_update" |awk '{print $4}'`
33         echo $result
34         ;;
35     Slow_queries)
36         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Slow_queries" |awk '{print $4}'`
37         echo $result
38         ;;
39     Com_select)
40         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_select" |awk '{print $4}'`
41         echo $result
42         ;;
43     Com_rollback)
44         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_rollback" |awk '{print $4}'`
45         echo $result
46         ;;
47     Com_insert)
48         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_insert" |awk '{print $4}'`
49         echo $result
50         ;;
51     Com_delete)
52         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_delete" |awk '{print $4}'`
53         echo $result
54         ;;
55     Com_commit)
56         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_commit" |awk '{print $4}'`
57         echo $result
58         ;;
59     Bytes_sent)
60         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Bytes_sent" |awk '{print $4}'`
61         echo $result
62         ;;
63     Bytes_received)
```

```

64         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Bytes_received" |awk '{print $4}'`
65         echo $result
66         ;;
67     Com_begin)
68         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Com_begin" |awk '{print $4}'`
69         echo $result
70         ;;
71     Threads_connected)
72         result=`${CONNECTION} extended-status 2>/dev/null |grep -w
"Threads_connected" |awk '{print $4}'`
73         echo $result
74         ;;
75     *)
76         echo
"Usage:$0(Uptime|Questions|Com_update|Slow_queries|Com_select|Com_r
ollback|Com_insert|Com_delete|Com_commit|Bytes_sent|Bytes_received|
Com_begin|Threads_connected)"
77         ;;
78 esac

```

◦ 给mysql\_status.sh加可执行权限

```
1 | # chmod +x mysql_status.sh
```

◦ 刷新服务

```
1 | # service zabbix-agent restart
```

◦ 服务端测试

```

1 | [root@zabbixserver ~]# zabbix_get -s 192.168.200.140 -p 10050 -k
mysql.status[Threads_connected]
2 | 1

```

• 设置mysql-down 的触发器

```

1 | 修改脚本，添加下面的监控
2 | [root@zabbix-client-1 zabbix_agentd.d]# mysqladmin extended-
status|grep Threads_connected
3 | Threads_connected | 3 |
4 | [root@zabbix-client-1 zabbix_agentd.d]#
5 |
6 | 自己增加一个监控项目的流程：
7 |     1.在zabbix-agent上的userparameter_mysql.conf 里添加参数
8 |     UserParameter=mysql.processlist,/usr/bin/mysqladmin -h
localhost -ucali -p'cali123456' processlist|awk '{print $4}'|uniq
|sort -rn|tail -n +2|wc -w
9 |     2.刷新zabbix-agent服务
10 |     3.zabbix-server上添加监控项--》出图
11 |     4.当mysql.processlist=0,发警报!!!

```

• 打造nginx监控

- 1、安装并启动nginx

- ```
1 # yum install epel-release -y
2 # yum install nginx -y
3 # service nginx start
```

- 2、在zabbix-agent端的linux服务器上的nginx的主配置文件里添加下面的配置项，开启nginx的统计功能，**并刷新服务**

- ```
1 # cat /etc/nginx/nginx.conf
2 #开启nginx本身的统计功能
3 location /nginx_status {
4     stub_status on;
5     access_log off;
6 }
7
8
9 # service nginx restart
```

- 3、web方式访问，查看效果

- ```
1 http://192.168.200.140/nginx_status
2 结果如下:
3 Active connections: 1
4 server accepts handled requests
5 1 1 1
6 Reading: 0 Writing: 1 Waiting: 0
7
8 ""Active connections  Nginx正处理的活动链接数个数; 重要
9 server                Nginx启动到现在共处理了多少个连接。
10 accepts               Nginx启动到现在共成功创建几次握手。
11 handled requests      Nginx总共处理了几次请求。
12 Reading               Nginx读取到客户端的 Header 信息数。
13 Writing               Nginx返回给客户端的 Header 信息数。
14 Waiting               Nginx已经处理完正在等候下一次请求指令的驻留链接, 开启。
15 keep-alive的情况下, waiting这个值等于active-(reading + writing)。
16 请求丢失数=(握手数-连接数)可以看出,本次状态显示没有丢失请求。""
```

- 4、编写这两个脚本 (nginx.sh、userparameter\_nginx.conf)

- ```
1 [root@zabbixclient1 zabbix_agentd.d]# cat userparameter_nginx.conf
2 UserParameter=nginx.status[*],/etc/zabbix/zabbix_agentd.d/nginx.sh
3 $1
4
5 [root@zabbixclient1 zabbix_agentd.d]# cat nginx.sh
6 HOST="192.168.200.140"
7 PORT="80"
8 case $1 in
9     active)
10         /usr/bin/curl --insecure
11         "http://$HOST:$PORT/nginx_status/" 2>/dev/null| grep 'Active' | awk
12         '{print $NF}'
13     ;;
14     reading)
15         /usr/bin/curl --insecure
16         "http://$HOST:$PORT/nginx_status/" 2>/dev/null| grep 'Reading' |
17         awk '{print $2}'
```

```

13         ;;
14         writing)
15             /usr/bin/curl --insecure
16             "http://$HOST:$PORT/nginx_status/" 2>/dev/null| grep 'writing' |
17             awk '{print $4}'
18         ;;
19         waiting)
20             /usr/bin/curl --insecure
21             "http://$HOST:$PORT/nginx_status/" 2>/dev/null| grep 'waiting' |
22             awk '{print $6}'
23         ;;
24         accepts)
25             /usr/bin/curl --insecure
26             "http://$HOST:$PORT/nginx_status/" 2>/dev/null| awk NR==3 | awk
27             '{print $1}'
28         ;;
29         ping)
30             /sbin/pidof nginx | wc -w
31         ;;
32         handled)
33             /usr/bin/curl --insecure
34             "http://$HOST:$PORT/nginx_status/" 2>/dev/null| awk NR==3 | awk
35             '{print $2}'
36         ;;
37         requests)
38             /usr/bin/curl --insecure
39             "http://$HOST:$PORT/nginx_status/" 2>/dev/null| awk NR==3 | awk
40             '{print $3}'
41         ;;
42     esac

```

- 5、刷新服务并给予nginx.sh可执行权限

```

1 # service zabbix-agent restart
2 # chmod +x nginx.sh

```

- 6、服务端测试

```

1 [root@zabbixserver ~]# zabbix_get -s 192.168.200.140 -p 10050 -k
2 nginx.status[ping]
3

```