

1. 分布式监控:

zabbix proxy 可以代替 zabbix server 收集性能和可用性数据,然后把数据汇报给 zabbix server,并且在一定程度上分担了zabbix server 的压力.

此外, 当所有agents和proxies报告给一个Zabbix server并且所有数据都集中收集时, 使用proxy是实现集中式和分布式监控的最简单方法。

1.1 应用场景:

- 监控远程区域设备
- 监控本地网络不稳定区域
- 当 zabbix 监控上千设备时,使用它来减轻 server 的压力
- 简化分布式监控的维护

注意: zabbix_proxy的**数据库**必须与zabbix_server的数据库分开。不过可以同时`在server端创建zabbix数据库与zabbix_proxy数据库`, 记得授权即可。

1.2 实战部署:

1.2.1 实验环境:

```
监控端: 192.168.75.121
代理端: 192.168.75.123
被监控端: 192.168.75.122
```

1.2.2 server配置:

```
LogFile=/var/log/zabbixsrv/zabbix_server.log
LogFileSize=0
PidFile=/run/zabbixsrv/zabbix_server.pid
# 指定数据库服务器地址，如果是单机部署，这里也可以直接指定localhost
DBHost=localhost
# 下面三行，与数据库授权信息保持一致
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
Timeout=4
ExternalScripts=/var/lib/zabbixsrv/externalscripts
LogSlowQueries=3000
TmpDir=/var/lib/zabbixsrv/tmp
```

1.2.3 proxy配置主动模式:

proxy端创建数据库:

```
yum install zabbix40-proxy zabbix40-proxy-mysql zabbix40-
dbfiles-mysql -y

create database zabbix_proxy charset=utf8;
grant all on zabbix_proxy.* to "zabbix"@"localhost"
identified by "zabbix_proxy";
flush privileges;

mysql -uzabbix -pzabbix_proxy zabbix_proxy <
/usr/share/zabbix-mysql/schema.sql
```

配置文件:

```
# proxy默认为主动模式，也就是Proxy收集数据，然后主动把数据推送到
Server。在主动模式中，配置检索的时间、心跳、数据发送频率是由每个
Proxy在配置文件中独立定义的。
ProxyMode=0
# 监控端IP地址
Server=192.168.75.121
# 与web中配置的proxy代理程序保持一致
Hostname=proxy-123
ListenPort=10051
LogFile=/var/log/zabbixsrv/zabbix_proxy.log
```

```
LogFileSize=0
PidFile=/run/zabbixsrv/zabbix_proxy.pid
DBHost=localhost
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=zabbix_proxy
#多久进行一次心跳检测
HeartbeatFrequency=60
#代理多久从Server获取一次配置变化
ConfigFrequency=60
#代理收集到数据后，多久向Server发送一次
DataSenderFrequency=3
ExternalScripts=/var/lib/zabbixsrv/externalscripts
LogSlowQueries=3000
TmpDir=/var/lib/zabbixsrv/tmp
```

1.2.4 agent配置:

```
PidFile=/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=192.168.75.123
Hostname=BJ_IDC_122
```

启动服务:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
```

1.2.5 web端配置:

创建proxy:

agent代理程序

agent代理程序 加密

* agent代理程序名称 与proxy配置文件保持一致

系统代理程序模式 主动式 被动式

代理地址

描述

proxy配置文件选择主动模式，这里也选择主动模式

添加监控主机：

主机

主机 模板 IPMI 宏 主机资产记录 加密

* 主机名称

可见的名称

* 群组 选择

* 至少存在一个接口。

agent代理程序的接口	IP地址	DNS名称	连接到	端口	默认
	<input type="text" value="192.168.75.122"/>	<input type="text"/>	<input checked="" type="radio"/> IP地址 <input type="radio"/> DNS	<input type="text" value="10050"/>	<input checked="" type="radio"/> 移除

描述

由agent代理程序监测

已启用

主机 模板 IPMI 宏 主机资产记录 加密

名称	动作
Template OS Linux	取消链接

链接指示器 选择

proxy

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	
<input checked="" type="checkbox"/> proxy-123	BJ_IDC_122	应用集 10	监控项 34	触发器 15	图形 5	自动发现 2	Web监测 192.168.75.122:10050	Template OS Linux (Template App Zabbix Agent)	已启用	ZBX	SNMP JMX IPMI	无

1.2.6 proxy配置被动模式:

被动模式: 是Server启用代理轮询进程, 周期性的从被动代理获取数据, 在被动模式中, 配置检索的时间、数据发送频率是由Server来定义的。

创建数据库等都不变, 主要是需要修改server和proxy的配置文件。

修改server配置文件:

```
[root@node1 ~]# grep "[a-z]" /etc/zabbix_server.conf
```

```
LogFile=/var/log/zabbixsrv/zabbix_server.log
```

```
LogFileSize=0
```

```
PidFile=/run/zabbixsrv/zabbix_server.pid
```

```
SocketDir=/var/lib/zabbixsrv/tmp
```

```
DBHost=192.168.75.121
```

```
DBName=zabbix
```

```
DBUser=zabbix
```

```
DBPassword=zabbix
```

```
Timeout=4
```

```
UnreachablePeriod=600
```

```
AlertScriptsPath=/var/lib/zabbixsrv/alertscripts
```

```
ExternalScripts=/var/lib/zabbixsrv/externalscripts
```

```
LogSlowQueries=3000
```

```
TmpDir=/var/lib/zabbixsrv/tmp
```

开启代理轮询进程

```
StartProxyPollers=5
```

配置发送给代理的刷新时间

```
ProxyConfigFrequency=60
```

从代理收集数据的间隔时间

```
ProxyDataFrequency=3
```

修改proxy的配置文件:

```
ProxyMode=1
```

监控端IP地址

```
Server=192.168.75.121
```

与web中配置的proxy代理程序保持一致

```
Hostname=proxy-123
```

```
ListenPort=10051
```

```
LogFile=/var/log/zabbixsrv/zabbix_proxy.log
```

```
LogFileSize=0
```

```
PidFile=/run/zabbixsrv/zabbix_proxy.pid
```

```
DBHost=192.168.75.123
```

```
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=zabbix_proxy
ExternalScripts=/var/lib/zabbixsrv/externalscripts
LogSlowQueries=3000
TmpDir=/var/lib/zabbixsrv/tmp
```

1.2.7 web端配置:

配置文件中设置为被动模式，这里也选择被动模式

指定代理服务器的IP地址

添加主机与主动模式一样，正常选择代理程序即可。（proxy的主动和被动模式中，agent的配置无需修改！）

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
proxy-123: BJ_IDC_122	应用集 10	监控项 34	触发器 15	图形 5	自动发现 2	Web监测 192.168.75.122: 10050	Template OS Linux (Template App Zabbix Agent)		已启用	ZBX SNMP JMX IPMI	无

2. 微信报警:

Zabbix除了可以使用邮件报警之外，还可以通过多种方式把告警信息发送到指定人，例如短信报警方式，越来越多的企业开始使用Zabbix结合微信作为主要的告警方式，因为每个人每天都在使用微信，这样可以及时有效的把告警信息推送到接收人，方便告警的及时处理。

2.1 微信企业号注册:

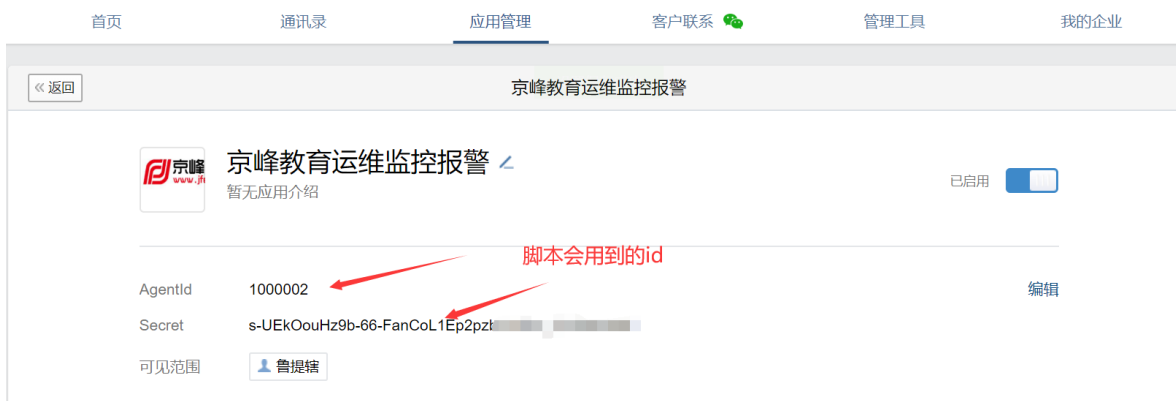
企业号注册地址: <https://qy.weixin.qq.com/>填写企业注册信息，个人也可以申请，不进行认证即可进行告警测试。

企业微信申请好后，可以在通讯录添加部门，以及运维人员，如果是测试，直接给管理员发也是可以的。



2.2 创建应用:

点击应用管理-点击下方创建应用，上传应用logo、输入应用名字、选择可见范围（可以是整个部门，也可以是指定人员），创建好后如下：



2.3 微信接口调试:

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

一、接口类型

二、接口列表 方法: GET

三、参数列表

*corpid

企业id，点击我的企业，在最下方获取。如果刚注册没有看到，多刷新几次，或者稍等片刻就有了。

公司的id

*corpsecret

应用secret，点击刚创建的监控应用可以获取到

企业应用secret

2.4 配置微信脚本:

```
cd /usr/src/
wget http://download.zhsir.org/Zabbix/weixin_linux_amd64
```

```
mv weixin_linux_amd64
/var/lib/zabbixsrv/alertscripts/wechat
chmod +x /var/lib/zabbixsrv/alertscripts/wechat
```

测试脚本:

```
/var/lib/zabbixsrv/alertscripts/wechat --
corpid=ww5922d1828182eb06 --corpsecret=s-UEkOouHz9b-66-
FanCoL1Ep2pzb1mDcv_34JGSCM4 --msg="告警测试" --user=lutixia
--agentid=1000002
```

提示:

--corpid= 企业里面的id(点击我的企业, 最下面可以看到)
--corpsecret= 企业应用Secret(点击报警应用可以看到)
-msg= 内容
-user=应用可见用户的账号(是唯一标识账号, 不是姓名)
--agentid=应用id(点击报警应用可以看到)

成员详情



鲁提辖

帐号: lutixia

2.5 web端创建报警媒介:

报警媒介类型 选项

* 名称 微信报警

类型 脚本

* 脚本名称 wechat

脚本参数

参数	动作
--corpid=ww5922d18	移除
--corpsecret=s-UEkOouHz9b-66-FanCoL1Ep2pzb1r	移除
--agentid=1000002	移除
--user={ALERT.SENDTO}	移除
--msg={ALERT.MESSAGE}	移除

[添加](#)

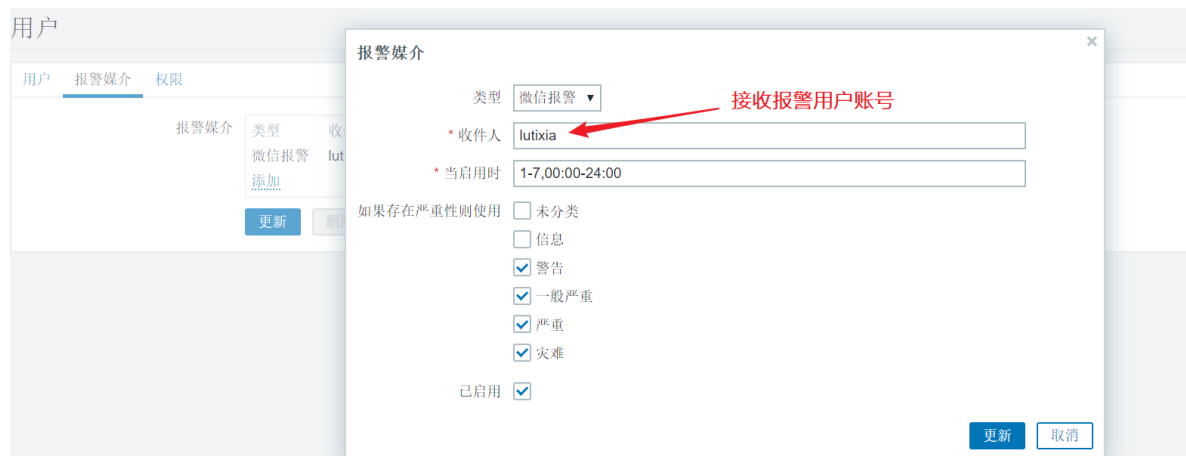
已启用

[更新](#) [克隆](#) [删除](#) [取消](#)

- 脚本参数{参考自己的企业ID和应用ID}

参数
--corpid=ww5922d182818
--corpsecret=s-UEkOouHz9b-66-FanCoL1Ep2pzb1mDcv_34
--agentid=1000002
--user={ALERT.SENDTO}
--msg={ALERT.MESSAGE}

2.6 用户绑定媒介:



2.7 告警测试:

触发器，动作与邮件告警一致，只需要将告警方式由邮件改为微信。