

1. zabbix自定义监控项:

在上一节课中，我们通过添加模板，zabbix自动为我们创建了很多监控项，但是往往模板中的监控项并不能满足我们业务需要，我们时常需要自己根据业务需要创建一系列的监控项，也成为自定义监控项。

监控项： zabbix进行监控的一个指标，zabbix中称为item，它的值由独立的key进行识别。

1.1 监控redis服务:

利用zabbix自带的功能创建监控项:

The screenshot shows the Zabbix web interface. At the top, there is a navigation bar with 'ZABBIX' and menu items: '监测', '资产记录', '报表', '配置', '管理'. Below this is a sub-menu: '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 '主机' and contains a form for configuring a monitoring item. The form includes fields for '名称' (Name), 'DNS', '模板' (Template) with a search box and '选择' button, 'IP地址' (IP Address), '要监控什么' (What to monitor) with buttons for '任何' (Any), '服务器' (Server), and 'agent代理程序' (Agent proxy), '端口' (Port), and 'agent代理程序' (Agent proxy) with a '选择' button. There are '应用' (Apply) and '重设' (Reset) buttons. Below the form is a table with columns: '名称', '应用集', '监控项', '触发器', '图形', '自动发现', 'Web监测', '接口', '模板'. The table has one row: '192.168.75.135', '应用集 12', '监控项 46', '触发器 19', '图形 8', '自动发现 2', 'Web监测', '192.168.75.135: 10050', 'Template OS Linux (Template App Zabbix Agent)'. At the bottom, there is another navigation bar with '监测', '资产记录', '报表', '配置', '管理' and a search bar. Below this is another sub-menu: '主机', '维护', '动作', '关联项事件', '自动发现', '服务'. A red arrow points from the '配置' menu in the top navigation bar to the '创建监控项' button in the bottom sub-menu. Below the sub-menu is a summary bar: '168.75.135 已启用 ZBX SNMP JMX IPMI 应用集 12 监控项 46 触发器 19 图形 8 自动发现规则 2 Web 场景'. Below this is a filter section with search boxes and dropdown menus for '类型', '信息类型', '状态', '更新间隔', '历史记录', and '触发器'.

* 名称

类型

* 键值

* 主机接口

信息类型

单位

* 更新间隔

自定义时间间隔

类型	间隔	期间	动作	
<input type="button" value="灵活"/>	<input type="button" value="调度"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	<input type="button" value="移除"/>

自定义脚本创建监控项:

1.1.1 修改zabbix-agent配置文件:

```
vim /etc/zabbix_agentd.conf

# 指定监控项及获取值的方式:
UserParameter=redis.status,bash /data/sh/redis_status.sh

# redis.status: 键 (监控项)
# bash /data/sh/redis_status.sh 值 (获取监控项的方式)
```

1.1.2 创建脚本:

```
mkdir -p /data/sh
vim /data/sh/redis_status.sh
#!/bin/bash
# by lutixia
#####

killall -0 redis-server &>/dev/null

if [ $? -eq 0 ];then
    echo 1
else
    echo 0
fi
```

1.1.3 测试脚本:

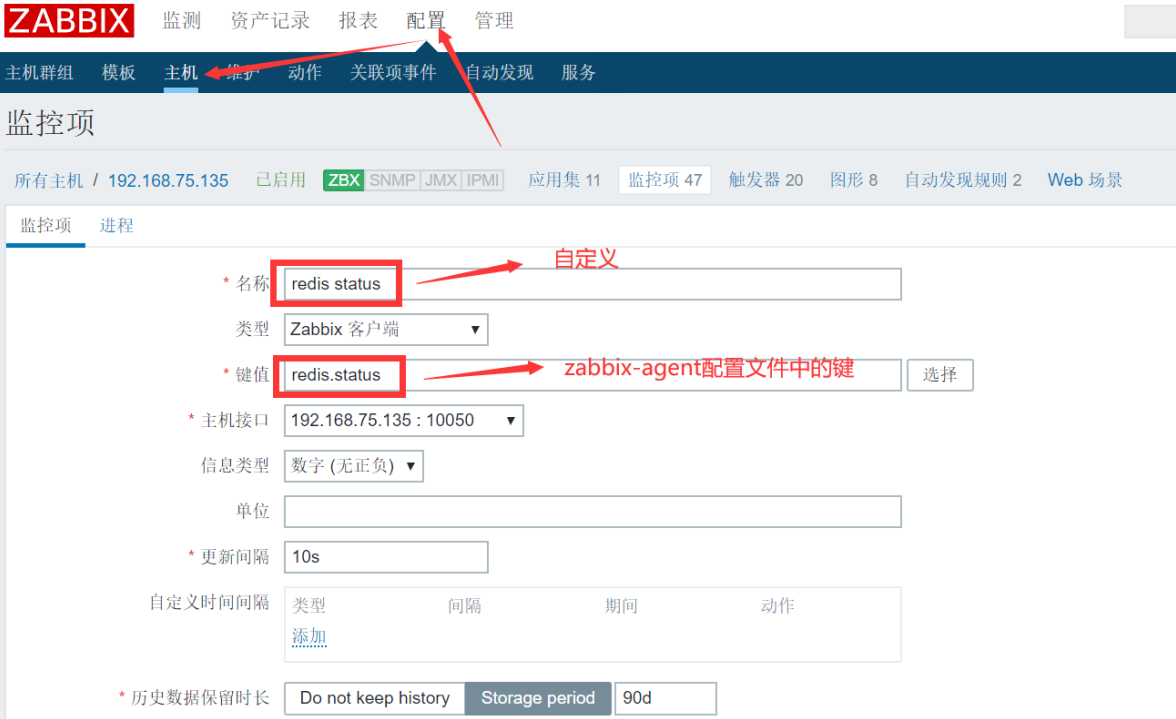
一定要先测试，脚本有没有问题，没有问题才能继续。而且，**测试需要在客户端及服务端都进行测试!**

```
### 客户端测试
# 当redis进程存在时测试:
[root@node5 ~]# ps -ef |grep redis
root      2790      1  0 14:24 ?          00:00:02
/usr/local/redis/bin/redis-server 0.0.0.0:6379
root      3238    1520  0 14:45 pts/0    00:00:00 grep --
color=auto redis
[root@node5 ~]# /data/sh/redis_status.sh
1
# 停止redis测试:
[root@node5 ~]# /etc/init.d/redis_6379 stop
Stopping ...
Redis stopped
[root@node5 ~]# /data/sh/redis_status.sh
0
```

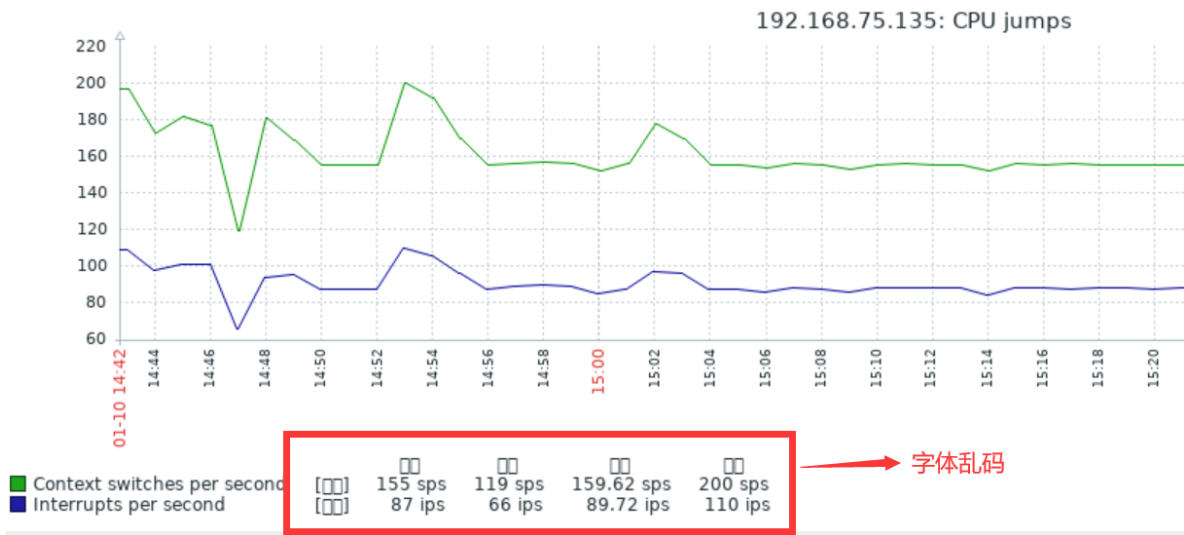
```
### 服务端测试:
# 当redis进程存在时测试:
[root@node7 ~]# zabbix_get -s 192.168.75.135 -k
"redis.status"
1
# 停止redis测试:
[root@node7 ~]# zabbix_get -s 192.168.75.135 -k
"redis.status"
0

# 如果出现，不管redis是启动还是停止，zabbix_get拿到的值都是0，则可以执行以下命令:
chmod u+s /usr/bin/killall
```

1.1.4 web界面添加监控项:



1.1.5 查看图：



1.1.6 解决字体乱码问题：

```
# 上传字体，然后替换字体，再次刷新网页即可：
mv -b simkai.ttf
/usr/share/zabbix/assets/fonts/DejaVuSans.ttf
```

1.2 监控nginx服务：

1.2.1 修改zabbix-agent配置文件：

```
vim /etc/zabbix_agentd.conf
```

```
# 指定监控项及获取值的方式:
```

```
UserParameter=nginx.status[*],bash  
/data/sh/nginx_status.sh $1
```

1.2.2 创建脚本 (一) :

```
vim /data/sh/nginx_status.sh
```

```
#!/bin/bash
```

```
# by lutixia
```

```
#####
```

```
function active {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==1{print $NF}'  
}
```

```
function accepts {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==3{print $1}'  
}
```

```
function handled {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==3{print $2}'  
}
```

```
function requests {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==3{print $3}'  
}
```

```
function reading {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==4{print $2}'  
}
```

```
function writing {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==4{print $4}'  
}
```

```
function waiting {
```

```
    curl 192.168.75.135/status 2>/dev/null | awk  
'NR==4{print $NF}'  
}
```

\$1

1.2.3 测试脚本:

```
[root@node7 ~]# zabbix_get -s 192.168.75.135 -k  
nginx.status[requests]  
2079
```

1.2.4 web界面添加监控项:

所有主机 / 192.168.75.135 已启用 ZBX SNMP JMX IPMI 应用集 12 监控项 50 触发器 19 图形 8 自动发现规则 2 Web 场景

监控项 进程

* 名称 → 分别创建监控项

类型

* 键值 →

* 主机接口

信息类型

单位

* 更新间隔

自定义时间间隔

类型	间隔	期间	动作
<input type="button" value="灵活"/>	<input type="button" value="调度"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>

* 历史数据保留时长

1.2.5 创建图像:

所有主机 / 192.168.75.135 已启用 ZBX SNMP JMX IPMI 应用集 12 监控项 50 触发器 19 图形 9 自动发现规则 2 Web 场景

图形 预览

* 名称 → 图像别名

* 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

纵轴最大值

* 监控项

名称	功能	绘图风格	纵轴Y侧	颜色	动作
1: 192.168.75.135: nginx active	<input type="text" value="平均"/>	<input type="text" value="线"/>	<input type="text" value="左侧"/>	<input type="text" value="1A7C11"/>	<input type="button" value="移除"/>
2: 192.168.75.135: nginx accepts	<input type="text" value="平均"/>	<input type="text" value="线"/>	<input type="text" value="左侧"/>	<input type="text" value="F63100"/>	<input type="button" value="移除"/>
3: 192.168.75.135: nginx handled	<input type="text" value="平均"/>	<input type="text" value="线"/>	<input type="text" value="左侧"/>	<input type="text" value="2774A4"/>	<input type="button" value="移除"/>

→ 添加监控项

1.2.6 创建脚本 (二) :

```
#!/bin/bash
```

```

# by lutixia
#####

values=`curl 192.168.75.135/status 2>/dev/null |awk
'NR==1{print $NF};NR==3{print
$1};NR==3{print$2};NR==3{print $3};NR==4{print
$2};NR==4{print $4};NR==4{print $NF}'`

case $1 in
    active)
        echo `echo $values | awk '{print $1}'`
        ;;
    accepts)
        echo `echo $values | awk '{print $2}'`
        ;;
    handled)
        echo `echo $values | awk '{print $3}'`
        ;;
    requests)
        echo `echo $values | awk '{print $4}'`
        ;;
    reading)
        echo `echo $values | awk '{print $5}'`
        ;;
    writing)
        echo `echo $values | awk '{print $6}'`
        ;;
    waiting)
        echo `echo $values | awk '{print $7}'`
        ;;
esac

```

2. 触发器:

触发器是对监控项采集的数据进行评估的逻辑表达式，如果接收的数据超过了可接受的状态，则触发器会被触发。

2.1 创建触发器:

在前面课程中，我们创建了对redis服务进行监控的监控项。现在对该监控项设置一个触发器，如果监控项的值为0，则触发一系列动作。

• 点击配置-->主机-->触发器-->点击创建触发器:



• 设置触发器名字:



• 设置触发器表达式:



确认之后, 可以在主机-触发器中看到刚才创建的触发器。

<input type="checkbox"/>	警告	正常	Mounted filesystem discovery: Free inodes is less than 20% on volume /boot
<input type="checkbox"/>	警告	正常	Template OS Linux: Lack of free swap space on {HOST.NAME}
<input type="checkbox"/>	警告	正常	Template OS Linux: Processor load is too high on {HOST.NAME}
<input type="checkbox"/>	警告	正常	redis status ←
<input type="checkbox"/>	警告	正常	Template OS Linux: Too many processes on {HOST.NAME}
<input type="checkbox"/>	警告	正常	Template OS Linux: Too many processes running on {HOST.NAME}

2.2 验证触发器:

关闭redis服务, 可以在问题中看到事件。


```
# 关闭136上的redis服务:
[root@node6 ~]# /etc/init.d/redis_6379 stop
Stopping ...
Redis stopped
```

点击监测--问题:

时间	<input type="checkbox"/> 严重性	恢复时间	状态	信息	主机	问题
19:41:08	<input type="checkbox"/> 警告		问题		192.168.75.136	redis status

3. 动作:

触发器可以在当接受到某个值超过预设的值时, 直观的显示出问题, 但是也只是仅仅显示在web界面, 监控人员还是需要时刻盯着屏幕, 才能及时看到问题。这样工作效率还是没有明显提升, 我们需要当这个触发器被触发时, 有一个动作帮我们报警或者直接帮我们恢复故障。

3.1 自动邮件报警:

3.1.1 为监控负责人绑定媒介:

- 点击管理-->用户-->报警媒介:

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

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
添加						

[更新](#) [删除](#) [取消](#)

- 点击添加:

类型 ← 选择邮件类型

* 收件人 ← 填写接收报警邮件的邮箱 移除

[添加](#)

* 当启用时 ← 选择接收邮件的时间

如果存在严重性则使用 未分类

信息
 警告
 一般严重 ← 发送什么级别的问题
 严重
 灾难

已启用

3.1.2 设置报警邮箱:

刚才设置的是接受报警信息的邮箱，现在设置报警邮箱（邮件的发送者）。

- 点击管理-->报警媒介类型:

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

报警媒介类型

名称 状态

<input type="checkbox"/>	名称 ▲	类型	状态	用于动作中	细节
<input type="checkbox"/>	Email	电子邮件	已启用	Report problems to Zabbix administrators	SMTP服务器: "s
<input type="checkbox"/>	Jabber	Jabber	已启用		Jabber 标识符: '
<input type="checkbox"/>	SMS	短信	已启用		GSM 调制解调器

报警媒介类型

报警媒介类型 选项

可以自定义

* 名称 选择电子邮件

类型 设置QQ的smtp服务器

* SMTP服务器 设置邮箱

SMTP服务器端口

* SMTP HELO

* SMTP电邮

安全链接

认证 选择用户和密码

用户名

密码

已启用

设置用户和密码，注意，密码是邮箱的授权码，进入邮箱账户开启smtp后，设置授权码

京峰教育-朱老师 <1550684538@qq.com>

邮箱首页 | 设置 - 换肤

邮箱设置

常规 帐户 换肤 收信规则 反垃圾 文件夹和标签 其他邮箱 我的订阅 信纸 体验室

默认帐户信息

京峰教育-朱老师
1550684538@qq.com

POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务

开启服务:	POP3/SMTP服务 (如何使用 Foxmail 等软件收发邮件?)	已关闭 开启
	IMAP/SMTP服务 (什么是 IMAP, 它又是如何设置?)	已开启 关闭
	Exchange服务 (什么是Exchange, 它又是如何设置?)	已关闭 开启
	CardDAV/CalDAV服务 (什么是CardDAV/CalDAV, 它又是如何设置?)	已关闭 开启
	(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置?)	

温馨提示: 登录第三方客户端时, 密码框请输入“授权码”进行验证。生成授权码

如果已经开启, 可以点击这里重新生成授权码

3.1.3 添加动作:

动作

动作 操作 恢复操作 更新操作

* 名称

自定义，动作名称

条件

标签	名称	动作
A	触发器 等于 192.168.75.136: redis status	移除

触发条件可以是某个具体的触发器，也可以是相同程度的一类触发器

新的触发条件

触发器名称 包含

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

动作

动作 操作 恢复操作 更新操作

* 默认操作步骤持续时间

动作默认持续时间，如果有多个动作，则表示每个动作之间的持续时间。

默认标题

消息内容

```
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}
```

暂停操作以制止问题

操作

步骤	细节	开始于	持续时间	动作
----	----	-----	------	----

操作细节

步骤	<input type="text" value="1"/> - <input type="text" value="1"/> (0 - 无穷大)
----	---

设置动作的优先级以及次数，动作按照1, 2, 3这样的顺序执行

步骤持续时间 (0 - 使用默认)

操作类型

可以自定义时间，如果不设置，则用默认时间

* 您必须至少选择一个用户或用户组。

选择发消息

发送到用户群组

用户群组	动作
Zabbix administrators	移除

发给哪个组

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 添加	动作
仅送到	Email ▼	
消息内容	<input checked="" type="checkbox"/>	
条件	标签 新的	名称 动作

[添加](#) [删除](#) 一定要点击添加或者更新，否则上面配置的信息将不会保存

* 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。

3.2 自动重启服务：

在配置动作中，我们可以设置相应的报警媒介给工作人员报警，也可以尝试先让zabbix为我们重启相应的服务，如果多次重启都失败了，则继续报警，让负责人来处理相关问题。

- 点击动作-->操作-->选择远程命令：

步骤	<input type="text" value="1"/> - <input type="text" value="1"/> (0 - 无穷大)	执行脚本的次数
步骤持续时间	<input type="text" value="0"/> (0 - 使用默认)	每次动作间隔时间，设置为0表示使用上面默认的时间
操作类型	远程命令 ▼	
* 目标列表	目标 当前主机 新的	动作 移除 对当前主机执行命令
类型	SSH ▼	选择使用ssh连接
认证方法	密码 ▼	
* 用户名称	<input type="text" value="root"/>	远程服务器用户名和密码，ssh 端口
密码	<input type="text" value="1"/>	
端口	<input type="text" value="22"/>	
* 命令	<input type="text" value="/etc/init.d/redis_6379 restart"/>	要执行的命令

操作	步骤	细节	开始于	持续时间	动作
	1	于当前主机上执行远程命令	立即地	默认	编辑 移除
	2	发送消息给用户群组: Zabbix administrators 通过 Email	00:01:00	默认	编辑 移除

操作细节

步骤 - (0 - 无穷大)

步骤持续时间 (0 - 使用默认)

报警也可以多次

先尝试用命令
重启服务，
如果重启失败，
则继续报警

操作类型

* 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

Zabbix administrators

[移除](#)

[添加](#)

发送到用户

用户

动作

[添加](#)

仅送到

消息内容