Name：曲中岭

Email:zlinggu@126.com

Q Q :441869115

在客户端执行:

cd /etc/zabbix

mkdir scripts

echo "UserParameter=lx2_discover_port,/etc/zabbix/scripts/lx2_discover_port.sh" >> zabbix_agentd.d/lx2.conf

echo "zabbix ALL=NOPASSWD: /bin/netstat,/bin/ps" >> /etc/sudoers

service zabbix-agent restart

cd scripts

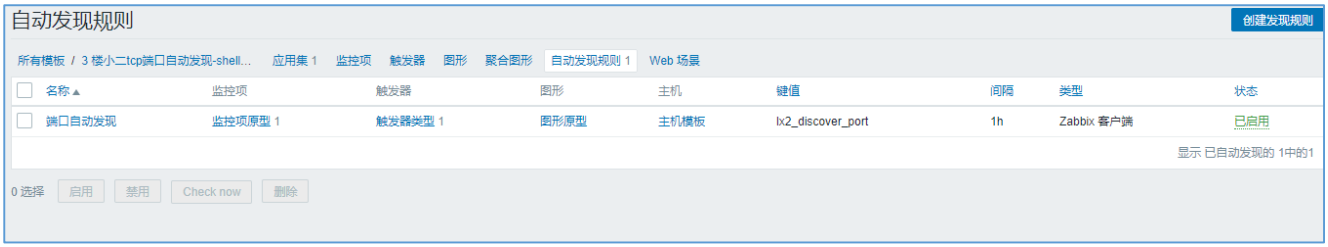
脚本上传到到 /etc/zabbix/scripts

cat lx2_discover_port.sh

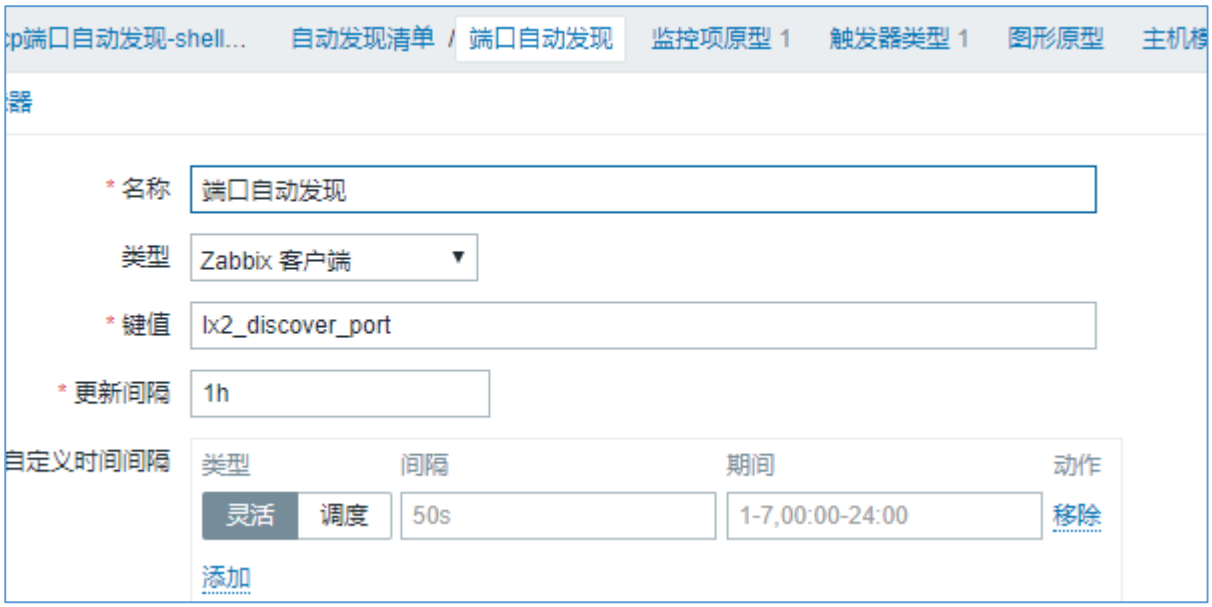
```
#!/bin/bash
all=(`sudo netstat -tnlp|grep -v tcp6|grep tcp |awk '{print $4"/"$7}'|awk -F":" '{print $2}'|awk '/\/[a-zA-Z]/{print}'`)
#awk '/\/[a-zA-Z]/{print},排除无 PID 和进程名的监听端口。例如：tcp 0 0 0.0.0.0:43874 0.0.0.0:* LISTEN -
#$4=IP:prot，$7=pid/name。$4,$7 之间使用/分割,awk '{print $4"/"$7}';awk -F":" '{print $2}',将$4 中冒号前面的排除;all 数组中每一项内容是 port/pid/name
port=(`echo ${all[*]}|sed 's/ /\n/g'|awk -F"/" '{print $1}'`) #all 数组拆分成三个数组,也可使用二维数组，这里未使用
pid=(`echo ${all[*]}|sed 's/ /\n/g'|awk -F"/" '{print $2}'`)
name=(`echo ${all[*]}|sed 's/ /\n/g'|awk -F"/" '{print $3}'`)


length=${#port[@]}
printf "{\n"
printf '\t""\"data\":["
for ((i=0;i<$length;i++))
do
[ ${port[$i]} -eq 32000 ] && name[$i]="Aliyun-cloudmonitor" #阿里云监控进程,监听 32000 端口，特殊处理
if [ ${name[$i]} = java ] ;then #如果是 java 项目,netstat 只能查出是 java，若要具体区分进程名，需要做进一步处理
name1=`sudo ps ux|grep jar|grep "\b${pid[$i]}\b"|egrep -o "\:[a-z/.]+?zookeeper-[0-9.]+?\.jar"|awk -F\: '{print $2}'`
#zookeeper 启动的 jar 识别,+? 重复 1 次或更多次，但尽可能少重复 ,简单理解就是最短匹配,下同
```

```
name2=`sudo ps ux|grep "\b${pid[$i]}\b"|grep jar|egrep -o "\:/[a-zA-Z0-9/-]+?\.jar"|awk -F\: '{print $2}'`
#tomcat 启动的项目
name3=`sudo ps ux|grep "\b${pid[$i]}\b"|grep jar|egrep -o "[[:space:]][/.0-9A-Za-z-]+\.jar"|awk '{print $1}'`
#jar 包单独启动的项目，包括 activemq
if [ ! -z $name1 ];then
name[$i]=$name1
elif [ ! -z $name2 ];then
name[$i]=$name2
elif [ ! -z $name3 ];then
name[$i]=$name3
fi
fi
printf '\n\t\t{'
printf "\n\t\t\t\"{#TCP_PORT}\":\"${port[$i]}\",\n\t\t"
printf "\t\"{#NAME}\":\"${name[$i]}\",\n\t\t"
printf "\t\"{#PID}\":\"${pid[$i]}\"\n\t\t}"
if [ $i -lt $[$length-1] ];then
printf ','
fi
done
printf "\n\t]\n"
printf "}\n"
```

配置自动发现：



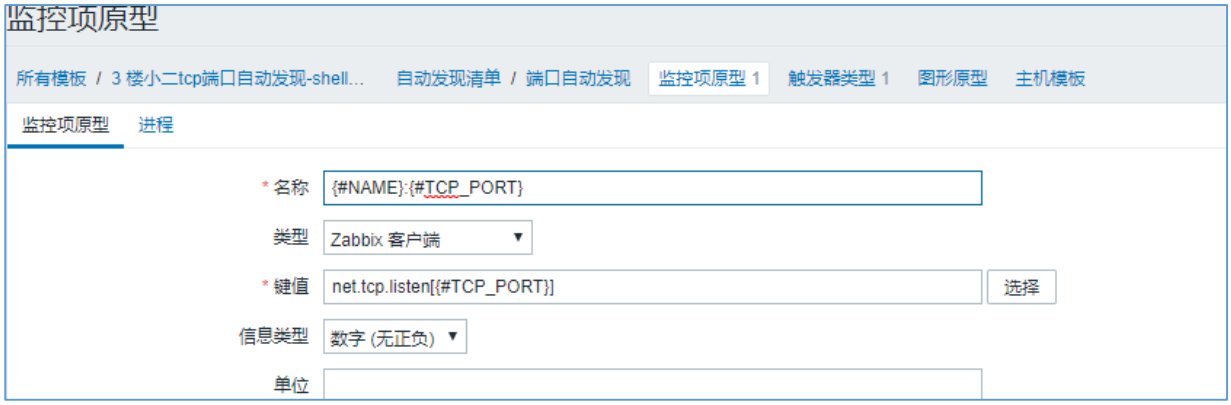
配置端口自动发现：



配置监控项原型:

{#NAME}:{#TCP_PORT}

net.tcp.listen[{#TCP_PORT}]

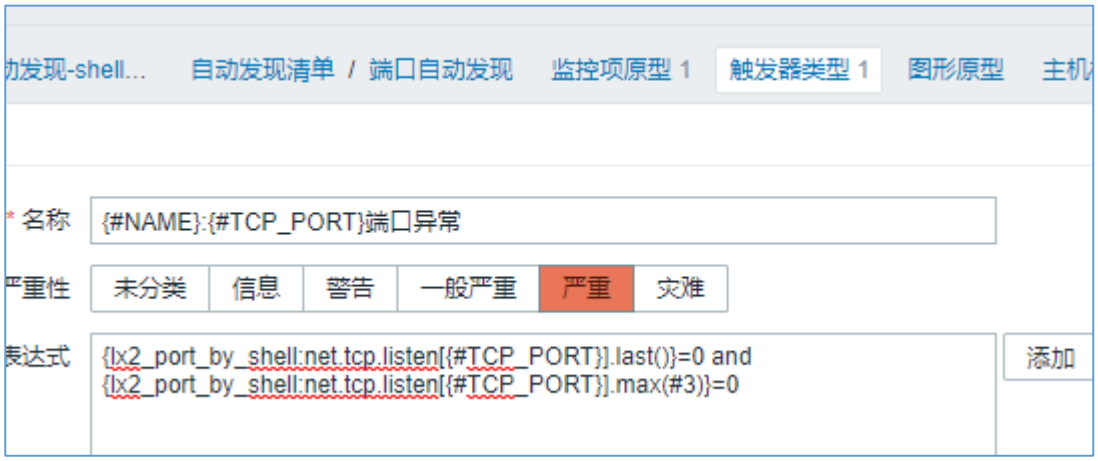


配置触发器原型:

{#NAME}:{#TCP_PORT}端口异常

{lx2_port_by_shell:net.tcp.listen[{#TCP_PORT}].last()}=0 and

{lx2_port_by_shell:net.tcp.listen[{#TCP_PORT}].max(#3)}=0



监控效果:

☐	•••	端口自动发现: lx2-vms-wss:60001	触发器 1	net.tcp.listen[60001]
☐	•••	端口自动发现: lx2-vms-wss:34988	触发器 1	net.tcp.listen[34988]
☐	•••	端口自动发现: lx2-vms-wss:33333	触发器 1	net.tcp.listen[33333]
☐	•••	端口自动发现: lx2-vms-wss:17112	触发器 1	net.tcp.listen[17112]
☐	•••	端口自动发现: lx2-vms-wss:17111	触发器 1	net.tcp.listen[17111]
☐	•••	端口自动发现: lx2-vms-wss:17110	触发器 1	net.tcp.listen[17110]
☐	•••	端口自动发现: lx2-vms-wss:8881	触发器 1	net.tcp.listen[8881]
☐	•••	端口自动发现: lx2-vms-wss:8880	触发器 1	net.tcp.listen[8880]
☐	•••	端口自动发现: lx2-vms-wss:8443	触发器 1	net.tcp.listen[8443]
☐	•••	端口自动发现: lx2-vms-wss:8111	触发器 1	net.tcp.listen[8111]
☐	•••	端口自动发现: lx2-vms-wss:8110	触发器 1	net.tcp.listen[8110]