

Name : 曲中岭

Email: [zlingqu@126.com](mailto:zlingqu@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 /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|grep -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|grep -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[$i]}\",\n\t\t\t{
printf "\t\t{#NAME}\":\"${name[$i]}\",\n\t\t\t{
printf "\t\t{#PID}\":\"${pid[$i]}\",\n\t\t\t{
if [ $i -lt ${length-1} ];then
printf ', '
fi
done
printf "\n\t}\n"
printf "}\n"
配置自动发现:

```

自动发现规则

创建发现规则

所有模板 / 3 楼小二tcp端口自动发现-shell...应用集 1 监控项 触发器 图形 聚合图形 自动发现规则 1 Web 场景

<input type="checkbox"/>	名称 ▲	监控项	触发器	图形	主机	键值	间隔	类型	状态
<input type="checkbox"/>	端口自动发现	监控项原型 1	触发器类型 1	图形原型	主机模板	lx2_discover_port	1h	Zabbix 客户端	已启用

显示 已自动发现的 1 中的 1

0 选择

启用

禁用

Check now

删除

配置端口自动发现：

端口自动发现-shell... 自动发现清单 / 端口自动发现 监控项原型 1 触发器类型 1 图形原型 主机模板

名称 端口自动发现

类型 Zabbix 客户端

键值 ix2\_discover\_port

更新间隔 1h

自定义时间间隔	类型	间隔	期间	动作
灵活	调度	50s	1-7,00:00-24:00	移除

添加

配置监控项原型:

{#NAME}:{#TCP\_PORT}

net.tcp.listen[{#TCP\_PORT}]

监控项原型

所有模板 / 3 楼小二tcp端口自动发现-shell... 自动发现清单 / 端口自动发现 监控项原型 1 触发器类型 1 图形原型 主机模板

监控项原型 进程

名称 {#NAME}:{#TCP\_PORT}

类型 Zabbix 客户端

键值 net.tcp.listen[{#TCP\_PORT}] 选择

信息类型 数字 (无正负)

单位

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

端口自动发现-shell... 自动发现清单 / 端口自动发现 监控项原型 1 触发器类型 1 图形原型 主机模板

名称 {#NAME}:{#TCP\_PORT}端口异常

严重性 未分类 信息 警告 一般严重 严重 灾难

表达式 {lx2\_port\_by\_shell:net.tcp.listen[{#TCP\_PORT}].last()}=0 and {lx2\_port\_by\_shell:net.tcp.listen[{#TCP\_PORT}].max(#3)}=0 添加

监控效果：

<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:60001	触发器 1	net.tcp.listen[60001]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:34988	触发器 1	net.tcp.listen[34988]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:33333	触发器 1	net.tcp.listen[33333]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:17112	触发器 1	net.tcp.listen[17112]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:17111	触发器 1	net.tcp.listen[17111]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:17110	触发器 1	net.tcp.listen[17110]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:8881	触发器 1	net.tcp.listen[8881]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:8880	触发器 1	net.tcp.listen[8880]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:8443	触发器 1	net.tcp.listen[8443]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:8111	触发器 1	net.tcp.listen[8111]
<input type="checkbox"/>	...	端口自动发现: lx2-vms-wss:8110	触发器 1	net.tcp.listen[8110]