第一章：端口自动发现并监控

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

第二章 TCP 连接数监控

第一步：

增加脚本，脚本内容如下：

```
root@i9Z:/etc/zabbix/zabbix_agentd.d# cat /etc/zabbix/scripts/tcp_status.sh
#!/bin/bash
[ $# -ne 1 ] && echo "Usage $0
(FIN_WAIT2|CLOSE_WAIT|TIME_WAIT|ESTABLISHED|LAST_ACK|FIN_WAIT1)" && exit 7
[ $1 != "ALL" -a $1 != "ESTABLISHED" -a $1 != "CLOSE_WAIT" -a $1 != "TIME_WAIT"
-a $1 != "FIN_WAIT2" -a $1 != "LAST_ACK" -a $1 != "FIN_WAIT1" ] \
&& echo "Usage $0
(ALL|FIN_WAIT2|CLOSE_WAIT|TIME_WAIT|ESTABLISHED|LAST_ACK|FIN_WAIT1)" && exit 7
netstat -n|grep "^tcp\b"| \
awk 'BEGIN
{S["ESTABLISHED"]=0;S["TIME_WAIT"]=0;S["FIN_WAIT1"]=0;S["FIN_WAIT2"]=0;S["CLOSE
_WAIT"]=0;S["LAST_ACK"]=0;} \
{++S[$NF]} \
END
{S["ALL"]=S["ESTABLISHED"]+S["TIME_WAIT"]+S["FIN_WAIT1"]+S["FIN_WAIT2"]+S["CLOS
E_WAIT"]+S["LAST_ACK"];for(a in S) print a, S[a]}' \
|awk '{print }' |grep $1|awk '{print $2}'
```

```
#!/bin/bash
[ $# -ne 1 ] && echo "Usage $0 (FIN_WAIT2|CLOSE_WAIT|TIME_WAIT|ESTABLISHED|LAST_ACK|FIN_
WAIT1)" && exit 7
[ $1 != "ALL" -a $1 != "ESTABLISHED" -a $1 != "CLOSE_WAIT" -a $1 != "TIME_WAIT" -a $1 !=
 "FIN_WAIT2" -a $1 != "LAST_ACK" -a $1 != "FIN_WAIT1" ] \
&& echo "Usage $0 (ALL|FIN_WAIT2|CLOSE_WAIT|TIME_WAIT|ESTABLISHED|LAST_ACK|FIN_WAIT1)" &
& exit 7


#sudo netstat -n|grep "^tcp\b"|awk 'BEGIN {S["ESTABLISHED"]=0;S["TIME_WAIT"]=0;S["FIN_WA
IT1"]=0;S["FIN_WAIT2"]=0;S["CLOSE_WAIT"]=0;S["LAST_ACK"]=0;} {++S[$NF]} END {for(a in S)
 print a, S[a]}'|awk '{print }'|grep $1|awk '{print $2}'
netstat -n|grep "^tcp\b"| \
awk 'BEGIN {S["ESTABLISHED"]=0;S["TIME_WAIT"]=0;S["FIN_WAIT1"]=0;S["FIN_WAIT2"]=0;S["CLO
SE_WAIT"]=0;S["LAST_ACK"]=0;} \
{++S[$NF]} \
END {S["ALL"]=S["ESTABLISHED"]+S["TIME_WAIT"]+S["FIN_WAIT1"]+S["FIN_WAIT2"]+S["CLOSE_WAI
T"]+S["LAST_ACK"];for(a in S) print a, S[a]}' \
|awk '{print }' |grep $1|awk '{print $2}'
~
```

第二步：自定义 key：

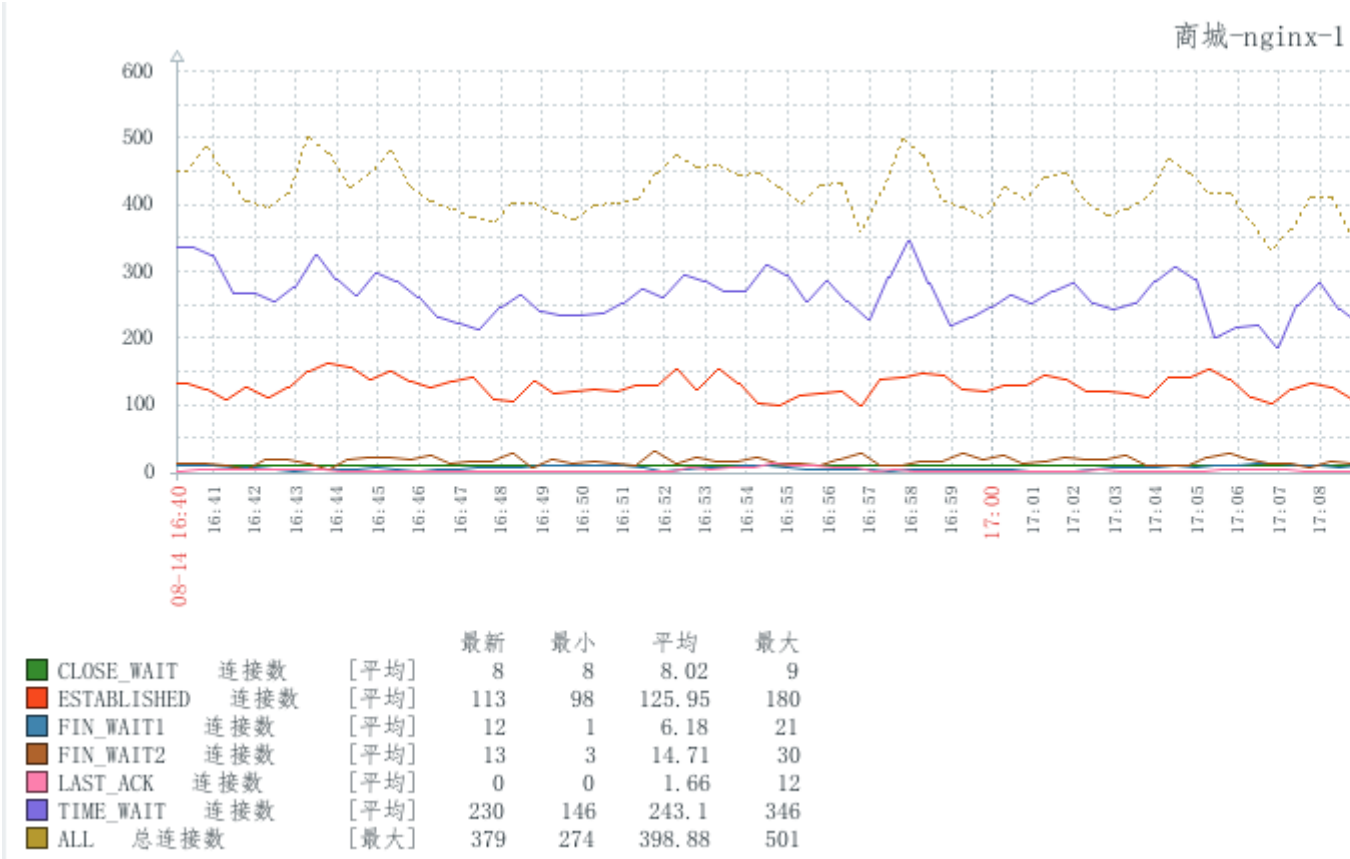
root@iZufvx79Z:/etc/zabbix/zabbix_agentd.d# cat
/etc/zabbix/zabbix_agentd.d/lx2.conf
UserParameter=lx2.tcp.status[*],/etc/zabbix/scripts/tcp_status.sh $1

第三步：添加模板，对应主机添加模板，模板中添加监控项目和图形

过滤器 只影响过滤后的数据

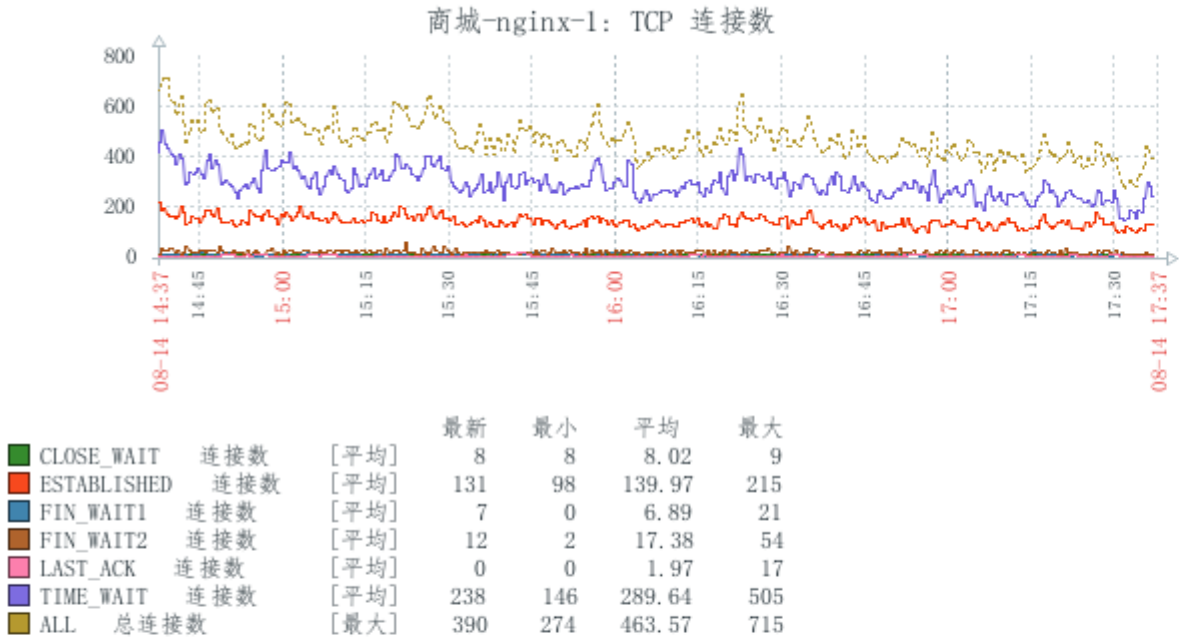
Wizard	名称 ▲	触发器	键值	间隔	历史记录
•••	ALL 总连接数		lx2.tcp.status[ALL]	30s	90d
•••	CLOSE_WAIT 连接数		lx2.tcp.status[CLOSE_WAIT]	30s	90d
•••	ESTABLISHED 连接数		lx2.tcp.status[ESTABLISHED]	30s	90d
•••	FIN_WAIT1 连接数		lx2.tcp.status[FIN_WAIT1]	30s	90d
•••	FIN_WAIT2 连接数		lx2.tcp.status[FIN_WAIT2]	30s	90d
•••	LAST_ACK 连接数		lx2.tcp.status[LAST_ACK]	30s	90d
•••	TIME_WAIT 连接数		lx2.tcp.status[TIME_WAIT]	30s	90d

四、图形观察，

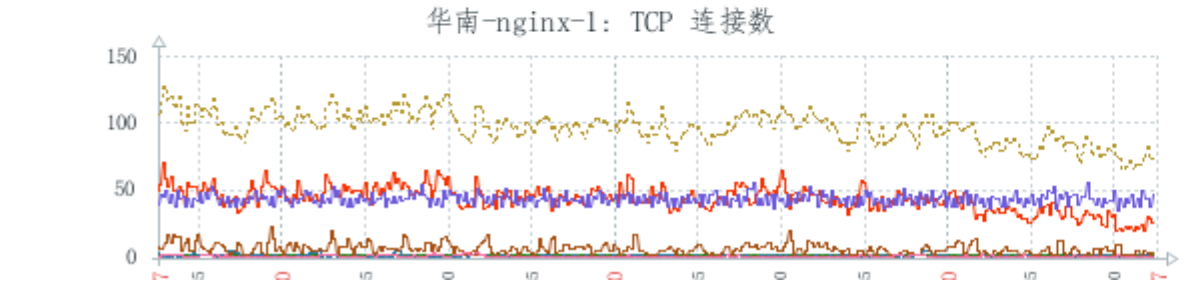


			最新	最小	平均	最大
CLOSE_WAIT	连接数	[平均]	8	8	8.02	9
ESTABLISHED	连接数	[平均]	113	98	125.95	180
FIN_WAIT1	连接数	[平均]	12	1	6.18	21
FIN_WAIT2	连接数	[平均]	13	3	14.71	30
LAST_ACK	连接数	[平均]	0	0	1.66	12
TIME_WAIT	连接数	[平均]	230	146	243.1	346
ALL	总连接数	[最大]	379	274	398.88	501

也可将类似图形添加到聚合图形：



			最新	最小	平均	最大
CLOSE_WAIT	连接数	[平均]	8	8	8.02	9
ESTABLISHED	连接数	[平均]	131	98	139.97	215
FIN_WAIT1	连接数	[平均]	7	0	6.89	21
FIN_WAIT2	连接数	[平均]	12	2	17.38	54
LAST_ACK	连接数	[平均]	0	0	1.97	17
TIME_WAIT	连接数	[平均]	238	146	289.64	505
ALL	总连接数	[最大]	390	274	463.57	715



第三章 zabbix 监控 activemq

```
#!/bin/bash
IP=10.28.93.179
PORT=8161
cd /etc/zabbix/scripts/
curl -uadmin:admin http://10.28.93.179:8161/admin/queues.jsp 2>/dev/null >
queues.jsp
line=(`awk '/<\/a><\/td>/{print NR}' queues.jsp`)
#以下 if 语句，做数据偏移定位使用，参考最后的注释信息进行理解
if [ -z $1 ] || [ -z $2 ];then
echo "Usage: $0 Pending|Consumers|Enqueued|Dequeued queur_name|all"
exit 7
elif [ $1 = Pending ];then
row=1
elif [ $1 = Consumers ];then
row=2
```

```
elif [ $1 = Enqueued ];then
row=3
elif [ $1 = Dequeued ];then
row=4
else
echo "Usage: $0 Pending|Consumers|Enqueued|Dequeued queur_name|all"
exit 7
fi

if [ $2 = all ];then
sum=0
for((i=0;i<${#line[@]};i++));do
#$2=all,获取所有队列的数据，需要将对应行的数据全部相加
rows=$[${line[$i]} + $row ]
line[$i]=`sed -n ${rows}p queues.jsp|sed 's@[<td>|</td>]@@g'`
sum=$[ ${line[$i]} + $sum ]
done
echo $sum
else
#如果$2 不是 all，则使用下面的语句即可获取指定队列的数据
cat queues.jsp | grep -A4 "$2</a></td>"|sed 's@[<td>|</td>]@@g'|tail -4|head -
$row|tail -1
fi


#以下是注释信息
: << !
curl 获取的原始格式如下，分别表示 Pending|Consumers|Enqueued|Dequeued
q.portal.api.employee.decline</a></td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
!
```