> 《Zabbix企业级分布式监控系统第2版》随书代码
>
> 代码仓库地址 https://github.com/zabbix-book/zabbix_v2
>
> 书籍购买地址 https://item.jd.com/12653708.html

434页

各状态代表的意思如下：
00，ERROR_STATUS。
01，TCP_ESTABLISHED。
02，TCP_SYN_SENT。
03，TCP_SYN_RECV。
04，TCP_FIN_WAIT1。
05，TCP_FIN_WAIT2。
06，TCP_TIME_WAIT。
07，TCP_CLOSE。
08，TCP_CLOSE_WAIT。
09，TCP_LAST_ACK。
0A，TCP_LISTEN。
0B，TCP_CLOSING。
10.1.2 TCP连接状态监控脚本的实现
下面通过命令来获取TCP连接状态，并自定义key。

```
shell# vim /etc/zabbix/zabbix_agentd.conf.d/tcp_connect.conf
#LISTEN, SYN-SENT, SYN-RECEIVED, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-
WAIT, CLOSING, LAST-ACK, TIME-WAIT和 CLOSED
UserParameter=socket.tcp_listen,      grep -c  "\b 0A \b" /proc/net/tcp
UserParameter=socket.tcp_synsent,     grep -c  "\b 02 \b" /proc/net/tcp
UserParameter=socket.tcp_synrecv,     grep -c  "\b 03 \b" /proc/net/tcp
UserParameter=socket.tcp_established,grep -c "\b 01 \b"  /proc/net/tcp
UserParameter=socket.tcp_finwait1,    grep -c "\b 04 \b"  /proc/net/tcp
UserParameter=socket.tcp_finwait2,    grep -c "\b 05 \b"  /proc/net/tcp
UserParameter=socket.tcp_closewait,   grep -c "\b 08 \b" /proc/net/tcp
UserParameter=socket.tcp_closing,     grep -c "\b 0B \b"  /proc/net/tcp
UserParameter=socket.tcp_lastack,     grep —c "\b 09 \b"  /proc/net/tcp
UserParameter=socket.tcp_timewait,    grep -c "\b 06 \b"  /proc/net/tcp
UserParameter=socket.tcp_closed,      grep -c "\b 07 \b"   /proc/net/tcp
```

435页

```
shell# vim /etc/nginx/conf.d/monitor.conf #/etc/nginx是配置目录
server {
        listen 127.0.0.1:80;
        server_name 127.0.0.1;
        location /nginxstatus {
```

```
        stub_status on;
        access_log off;
        allow 127.0.0.1;            #这里是允许的IP地址
        deny all;
        }
}
shell# nginx -t  #如果输出错误，请检查配置文件
nginx: the configuration file etc/nginx/nginx.conf syntax is ok
nginx: configuration file/etc/nginx/nginx.conf test is successful
shell# nginx -s reload  #测试配置是否正确
```

436页

Nginx监控脚本如下：

```
shell# vim /etc/zabbix/scripts/nginx_status
#!/bin/bash

# function:monitor nginx  from zabbix
# License: GPL
# mail: itnihao@qq.com
# version 1.0 date:2012-12-09
# version 1.0 date:2013-01-15
# version 1.1 date:2014-09-05

source /etc/bashrc >/dev/null 2>&1
source /etc/profile  >/dev/null 2>&1
nginxstatus=http://127.0.0.1/nginxstatus

# Functions to return nginx stats
function checkavailable {
    code=$(curl -o /dev/null -s -w %{http_code} ${nginxstatus})
    if [ "${code}" == "200" ];then
        return 1
    else
        echo  0
    fi
}
function active {
    checkavailable|| curl -s "${nginxstatus}" | grep 'Active' | awk '{print
$3}'
}
function reading {
    checkavailable|| curl -s "${nginxstatus}" | grep 'Reading' | awk '{print
$2}'
}
function writing {
```

```
    checkavailable|| curl -s "${nginxstatus}" | grep 'Writing' | awk '{print
$4}'
}
function waiting {
    checkavailable|| curl -s "${nginxstatus}" | grep 'Waiting' | awk '{print
$6}'
}
function accepts {
    checkavailable|| curl -s "${nginxstatus}" | awk NR==3 | awk '{print $1}'
}
function handled {
    checkavailable|| curl -s "${nginxstatus}" | awk NR==3 | awk '{print $2}'
}
function requests {
    checkavailable|| curl -s "${nginxstatus}" | awk NR==3 | awk '{print $3}'
}

case "$1" in
    active)
        active
        ;;
    reading)
        reading
        ;;
    writing)
        writing
        ;;
    waiting)
        waiting
        ;;
    accepts)
        accepts
       ;;
    handled)
        handled
        ;;
    requests)
        requests
        ;;
    *)
        echo "Usage: $0 {active |reading |writing |waiting |accepts |handled
|requests }"
esac
```

437页

key的配置文件如下：

```
shell# vim /etc/zabbix/zabbix_agentd.conf.d/monitor_ningx.conf
UserParameter=nginx.accepts,/etc/zabbix/scripts/check_nginx_status.sh accepts
UserParameter=nginx.handled,/etc/zabbix/scripts/check_nginx_status.sh handled
UserParameter=nginx.requests,/etc/zabbix/scripts/check_nginx_status.sh requests
UserParameter=nginx.connections.active,/etc/zabbix/scripts/check_nginx_status.s
h active
UserParameter=nginx.connections.reading,/etc/zabbix/scripts/check_nginx_status.
sh reading
UserParameter=nginx.connections.writing,/etc/zabbix/scripts/check_nginx_status.
sh writing
UserParameter=nginx.connections.waiting,/etc/zabbix/scripts/check_nginx_status.
sh waiting
```

438页

```
shell# vim /etc/php-fpm.conf
pm.status_path = /phpfpmstatus
shell# /etc/init.d/php-fpm restart
shell# vim /etc/nginx/conf.d/monitor.conf
server {
        listen 127.0.0.1:80;
        server_name 127.0.0.1;
        location /nginxstatus {
              stub_status on;
              access_log off;
              allow 127.0.0.1;
              deny all;
        }
        location ~ ^/(phpfpmstatus)$ {
                include fastcgi_params;
                fastcgi_pass unix:/tmp/fpm.sock;
                fastcgi_param SCRIPT_FILENAME $fastcgi_script_name;
        }
}
```

439页

```
shell# vim /etc/zabbix/scripts/monitor_phpfpm_status
#!/bin/bash

# function:monitor php-fpm status from zabbix
# License: GPL
# mail:admin@itnihao.com
# date:2016-01-06
```

```
source /etc/bashrc >/dev/null 2>&1
source /etc/profile  >/dev/null 2>&1

PHPFPM_FILE=/var/log/zabbix/phpfpmstatus.tmp
CMD () {
     curl http://127.0.0.1/phpfpmstatus >${PHPFPM_FILE} 2>&1
}

if [ -e ${PHPFPM_FILE} ]; then
    TIMEFROM=`stat -c %Y ${TMP_MYSQL_STATUS}`
    TIMENOW=`date +%s`
    if [ `expr $TIMENOW - $TIMEFROM` -gt 60 ]; then
        rm -f ${PHPFPM_FILE}
        CMD
    fi
else
    CMD
fi

pool(){
     awk '/pool/ {print $NF}' ${PHPFPM_FILE}
}
process_manager() {
     awk '/process manager/ {print $NF}' ${PHPFPM_FILE}
}

start_since(){
    awk '/^start since:/ {print $NF}' ${PHPFPM_FILE}
}
accepted_conn(){
    awk '/^accepted conn:/ {print $NF}' ${PHPFPM_FILE}
}
listen_queue(){
    awk '/^listen queue:/ {print $NF}' ${PHPFPM_FILE}
}
max_listen_queue(){
    awk '/^max listen queue:/ {print $NF}' ${PHPFPM_FILE}
}
listen_queue_len(){
    awk '/^listen queue len:/ {print $NF}' ${PHPFPM_FILE}
}
idle_processes(){
    awk '/^idle processes:/ {print $NF}' ${PHPFPM_FILE}
}
active_processes(){
    awk '/^active processes:/ {print $NF}' ${PHPFPM_FILE}
}
total_processes(){
```

```
    awk '/^total processes:/ {print $NF}' ${PHPFPM_FILE}
}
max_active_processes(){
    awk '/^max active processes:/ {print $NF}' ${PHPFPM_FILE}
}
max_children_reached(){
    awk '/^max children reached:/ {print $NF}' ${PHPFPM_FILE}
}

case "$1" in
pool)
    pool
    ;;
process_manager)
    process_manager
    ;;
start_since)
    start_since
    ;;
accepted_conn)
    accepted_conn
    ;;
listen_queue)
    listen_queue
    ;;
max_listen_queue)
    max_listen_queue
    ;;
listen_queue_len)
    listen_queue_len
    ;;
idle_processes)
    idle_processes
    ;;
active_processes)
    active_processes
    ;;
total_processes)
    total_processes
    ;;
max_active_processes)
    max_active_processes
    ;;
max_children_reached)
    max_children_reached
    ;;
*)
```

```
    echo "Usage: $0 {pool|process_manager|start_since|accepted_conn|
listen_queue|max_listen_queue|listen_queue_len|idle_processes|active_processes|
total_processes|max_active_processes|max_children_reached}"
esac
```

441页

key的php-fpm.conf子配置文件如下：

```
shell# cat /etc/zabbix/zabbix_agentd.conf.d/php-fpm.conf
UserParameter=phpfpm.status.pool,/etc/zabbix/scripts/check_phpfpm.sh pool
UserParameter=phpfpm.status.process.manager,/etc/zabbix/scripts/check_phpfpm.sh
process_manager
UserParameter=phpfpm.status.start.since,/etc/zabbix/scripts/check_phpfpm.sh
start_since
UserParameter=phpfpm.status.accepted.conn,/etc/zabbix/scripts/check_phpfpm.sh
accepted_conn
UserParameter=phpfpm.status.listen.queue,/etc/zabbix/scripts/check_phpfpm.sh
listen_queue
UserParameter=phpfpm.status.max.listen.queue,/etc/zabbix/scripts/check_phpfpm.s
h max_listen_queue
UserParameter=phpfpm.status.listen.queue.len,/etc/zabbix/scripts/check_phpfpm.s
h listen_queue_len
UserParameter=phpfpm.status.idle.processes,/etc/zabbix/scripts/check_phpfpm.sh
idle_processes
UserParameter=phpfpm.status.active.processes,/etc/zabbix/scripts/check_phpfpm.s
h active_processes
UserParameter=phpfpm.status.total.processes,/etc/zabbix/scripts/check_phpfpm.sh
total_processes
UserParameter=phpfpm.status.max.active.processes,/etc/zabbix/scripts/check_phpf
pm.sh max_active_processes
UserParameter=phpfpm.status.max.children.reached,/etc/zabbix/scripts/check_phpf
pm.sh max_children_reached
```

442页

```
mysql> SHOW GLOBAL STATUS;           #查看全局状态
mysql> SHOW GLOBAL VARIABLES;        #查看全局变量
mysql> SHOW ENGINE INNODB STATUS;    #查看InooDB状态
mysql> SHOW SLAVE STATUS;              #查看Slave状态
mysql> SHOW MASTER STATUS;             #查看Master状态
mysql> SHOW  BINARY LOGS;              #查看日志状态
mysql> SHOW PROCESSLIST;               #查看进程状态
mysql> SELECT SUM(compress_time) AS compress_time, SUM(uncompress_time) AS
uncompress_time FROM information_schema.INNODB_CMP;
mysql> SELECT LOWER(REPLACE(trx_state, " ", "_")) AS state, count(*) AS cnt
from information_schema.INNODB_TRX GROUP BY state;
mysql> SELECT SUM(trx_rows_locked) AS rows_locked, SUM(trx_rows_modified) AS
rows_modified, SUM(trx_lock_memory_bytes) AS lock_memory FROM information_
schema.INNODB_TRX;#查看锁
mysql> select table_name,(DATA_LENGTH+INDEX_LENGTH)/1024/1024 as total_mb,
table_rows from information_schema.tables where table_rows is not  NULL; #查看表
名和表记录条数
```

```
shell# cp check_mysql /etc/zabbix/scripts/check_mysql
shell# vim /etc/zabbix/zabbix_agentd.conf.d/userparameter_mysql.conf
UserParameter=mysql.variables[*], /etc/zabbix/scripts/check_mysql -collect
variables -metric "mysql.variables[$1]" --host="$2" --port="$3" --user="$4" --
pass="$5"
UserParameter=mysql.status[*], /etc/zabbix/scripts/check_mysql -collect status
-metric "mysql.status[$1]" --host="$2" --port="$3" --user="$4" --pass= "$5"
UserParameter=mysql.processlist.state[*],/etc/zabbix/scripts/check_mysql -
collect showProcesslist -metric "mysql.processlist.state[$1]" --host="$2" --
port="$3" --user="$4" --pass="$5"
UserParameter=mysql.innodb[*], /etc/zabbix/scripts/check_mysql -collect innodb
-metric "mysql.innodb[$1]" --host="$2" --port="$3" --user="$4" --pass= "$5"
UserParameter=mysql.master[*], /etc/zabbix/scripts/check_mysql -collect master
-metric "mysql.master[$1]" --host="$2" --port="$3" --user="$4" --pass= "$5"
UserParameter=mysql.slave[*], /etc/zabbix/scripts/check_mysql -collect slave -
metric "mysql.slave[$1]" --host="$2" --port="$3" --user="$4" --pass="$5"
UserParameter=mysql.table.discovery[*],  /etc/zabbix/scripts/check_mysql -
collect discoveryTable -metric "mysql.table.discovery[$1]" --host="$2" --
port="$3" --user="$4" --pass="$5"
UserParameter=mysql.table.info.rows[*],  /etc/zabbix/scripts/check_mysql -
collect tableInfoRows -metric "mysql.table.info.rows[$1]" --host="$2" --
port="$3" --user="$4" --pass="$5"
UserParameter=mysql.total.spaces.used[*],/etc/zabbix/scripts/check_mysql -
collect tableInfoUsed -metric "mysql.total.spaces.used[$1]" --host="$2" --
port="$3" --user="$4" --pass="$5"
UserParameter=mysql.ping[*], /etc/zabbix/scripts/check_mysql -collect ping -
metric ping --host="$1" --port="$2" --user="$3" --pass="$4"
```

443页

```
shell# mysql -uroot -p
mysql> GRANT USAGE,SELECT,PROCESS,REPLICATION CLIENT,REPLICATION SLAVE ON *.*
TO 'zabbix'@'127.0.0.1' IDENTIFIED BY 'zabbix';
mysql> flush privileges;
```

449页

```
shell# ipmitool sensor
Intrusion         | 0x00              | ok
Fan1              | 4440 RPM          | ok
Fan2              | 4440 RPM          | ok
Fan3              | 4440 RPM          | ok
Fan4              | 4440 RPM          | ok
Fan5              | 4440 RPM          | ok
Fan6              | 4440 RPM          | ok
Inlet Temp        | 32 degrees C      | ok
CPU Usage         | 1 percent         | ok
IO Usage          | 0 percent         | ok
MEM Usage         | 0 percent         | ok
SYS Usage         | 2 percent         | ok
Exhaust Temp      | 37 degrees C      | ok
Temp              | 45 degrees C      | ok
Temp              | 44 degrees C      | ok
OS Watchdog       | 0x00              | ok
VCORE PG          | 0x00              | ok
VCORE PG          | 0x00              | ok
3.3V PG           | 0x00              | ok
5V AUX PG         | 0x00              | ok
```

452页

表10-2 常见的磁盘阵列卡工具

工具名称	支持的型号
LSIUtil	LSI1068系列阵列卡
MegaCli	LSI2208
SAS2IRCU	LSI2308
SAS3IRCU	LSI3008
STorCLI	LSI3108
Hpacucli	HP服务器特有的阵列卡
Hpssacli	HP服务器特有的阵列卡

LSI提供的工具名称为MegaRAID，其官方下载地址为：

https://www.broadcom.com/products/storage/raid-controllers/megaraid-sas-9361-4i#downloads

453页

与磁盘监控相关的命令和文件如下：

```
/etc/zabbix/scripts/check_lsimegaraid.sh       #检测脚本
/etc/zabbix/scripts/check_MegaCli64              #检测脚本
/etc/zabbix/zabbix_agentd.conf.d/userparameter_lsimegaraid.conf   #配置文件
```

配置sudo，允许check_MegaCli64命令访问。

```
/etc/sudoers.d/zabbix_sudo
zabbix ALL=(ALL) NOPASSWD: /etc/zabbix/scripts/check_MegaCli64
```

配置UserParameter,定义磁盘监控的key。

```
UserParameter=lsimegaraid.discovery[*],/etc/zabbix/scripts/check_lsimegaraid.sh
"discovery" $1 $2
UserParameter=lsimegaraid.trapper[*],/etc/zabbix/scripts/check_lsimegaraid.sh
"trapper" $1 $2
```

添加脚本与配置文件[指哪个目录？]后，需要执行重启Zabbix-Agent的操作。

```
shell# chmod 755 /etc/zabbix/scripts/check_lsimegaraid.sh
shell# chmod 755 /etc/zabbix/scripts/check_MegaCli64
shell# chown zabbix /etc/zabbix/scripts/check_lsimegaraid.sh
shell# chown zabbix /etc/zabbix/scripts/check_MegaCli64
shell# systemctl restart zabbix-agent
```

455页

```
Router#configure terminal
Router(config)#ip access-list standard snmp-filter #创建方位列表
Router(config-std-nacl)#permit 192.168.0.240  #允许192.168.0.240访问
Router(config-std-nacl)#deny any log          #拒绝其他用户访问，但记录日志
Router(config-std-nacl)#end
Router#configure terminal
Router(config)#snmp-server   community  public RO snmp-filter #设置团组
（community）名为public
```

Cisco的SNMP配置文档请参考如下地址：

http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf014.html

```
shell# yum  install  net-snmp-utils -y
shell# snmpwalk -v 2c -c public 172.30.31.10 1.3.6.1.4.1.9.2.1.57.0
SNMPv2-SMI::enterprises.9.2.1.57.0 = INTEGER: 8
```

458页

```
shell# systemctl restart zabbix-server
Shutting down zabbix_server:                               [  OK  ]
Starting zabbix_server:  zabbix_server [19571]: ERROR: cannot start vmware
collector because Zabbix server is built without VMware support
                                                           [FAILED]
```

```
shell# vim  /etc/zabbix/zabbix_server.conf
### Option: StartVMwareCollectors
#       Number of pre-forked vmware collector instances.
#
# Mandatory: no
# Range: 0-250
# Default:
# StartVMwareCollectors=0
StartVMwareCollectors=1
```

462页

```
/etc/zabbix/scripts/check_rabbitmq                          #检测脚本
/etc/zabbix/zabbix_agentd.conf.d/userparameter_rabbitmq.conf    #配置文件
添加脚本与配置文件[指哪个目录？]后，需要执行重启Zabbix-Agent的操作。
shell# chmod 755 /etc/zabbix/scripts/check_rabbitmq
shell# chown zabbix:zabbix /etc/zabbix/scripts/check_rabbitmq
shell# systemctl restart zabbix-agent
```

463页

```
# 查看当前所有用户
Shell# rabbitmqctl list_users
# 添加新用户，用户名为admin，密码为pwd
shell# rabbitmqctl add_user admin pwd
# 设置用户tag
shell# rabbitmqctl set_user_tags admin administrator
# 赋予默认用户admin的全部操作权限[这条注释对吗？]
shell# rabbitmqctl set_permissions -p / admin ".*" ".*" ".*"
# 查看用户的权限
shell# rabbitmqctl list_user_permissions admin
测试获取RabbitMQ数据，命令如下：

shell# zabbix_get  -s 127.0.0.1 -k rabbitmq.discovery.node
{"data":[{"{#NODENAME}":"rabbit@localhost","{#NODETYPE}":"disc"}]}
```

465页

```
shell# curl "http://localhost:9200/_cluster/health")
shell# curl "http://localhost:9200/_nodes/_local/stats?all=true"
获取数据之后，我们编写一些代码将数据输出为Zabbix需要的格式。
关于Elasticsearch的监控指标，官方文档地址如下：
```

https://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-nodes-stats.html

466页

```
Kafka监控的JMX配置命令如下：
shell# cd /data/app/kafka_2.12-1.1.0/  #进入Kafka所在目录
shell# bin/kafka-server-stop.sh
shell# JMX_PORT=10053 bin/kafka-server-start.sh  -daemon
config/server.properties  #开启JMX端口
```

```
Kafka服务器指标：CPU 负载、磁盘IO、内存使用情况。[中文意思？]
JVM监控：主要监控Java的垃圾回收时间，Java的垃圾回收机制对性能的影响比较明显。
Kafka Broker监控：Kafka集群中Broker列表，Broker运行状况，包括Node下线、活跃数量、
Broker是否提供服务、数据流量、流入速度、流出速度。
Kafka Controller监控：Controller存活数目。
Kafka Producer监控：Producer数量、排队情况、请求响应时间。
Kafka Consumer监控：Consumer队列中排队请求数、请求响应时间、最近1分钟平均每秒请求数。
Kafka Topic监控：数据量大小、Offset数据流量、流入速度、流出速度。
```

更多的监控指标和详细的解释，请参考《Kafka权威指南》的第10章。官方文档见http://kafka.apache.org/documentation/#monitoring。

468页

Redis监控的配置命令如下：

```
shell# vim /etc/zabbix/zabbix_agentd.conf.d/userparameter_redis.conf
UserParameter=redis_info[*],(echo info; sleep 0.01) | telnet 127.0.0.1 $1
2>&1|grep "\b$2\b"|cut -d : -f2
UserParameter=redis.port.discovery,sudo /etc/zabbix/scripts/process_ discovery
redis
```

470页

安装Oracle客户端软件包，执行如下命令：

```
shell# rpm -ivh oracle-instantclient11.2-basic-11.2.0.3.0-1.x86_64.rpm
shell# rpm -ivh oracle-instantclient11.2-devel-11.2.0.3.0-1.x86_64.rpm
shell# rpm -ivh oracle-instantclient11.2-sqlplus-11.2.0.3.0-1.x86_64.rpm
shell# rpm -ivh cx_Oracle-5.1.2-11g-py26-1.x86_64.rpm    #可以使用pip安装
shell# rpm -ivh python-argparse-1.2.1-2.el6.noarch.rpm
```

Oracle客户端软件包的下载地址如下，读者需要根据不同的Oracle版本进行下载，如图10-50所示。

http://www.oracle.com/technetwork/database/database-technologies/instant-client/downloads/index.html

471页

安装成功后，配置环境变量如下：

```
shell# cat > /etc/profile.d/oracle.sh << EOF
#!/bin/bash
LD_LIBRARY_PATH="/usr/lib/oracle/11.2/client64/lib:${LD_LIBRARY_PATH}"
export LD_LIBRARY_PATH
ORACLE_HOME="/usr/lib/oracle/11.2/client64/lib"
export ORACLE_HOME
EOF
```

将代码和配置文件放到相应的目录下：

```
#代码见https://github.com/zabbix-book/Pyora/blob/master/pyora.py
shell# /etc/zabbix/scripts/pyora.py
shell# vim /etc/zabbix/zabbix_agentd.conf.d/oracle.conf
UserParameter=pyora[*],/etc/zabbix/scripts/pyora.py $1 $2 $3 $4 $5 $6 $7 $8
```

配置Oracle的监控账户，账户名称是zabbix，密码为zabbix。

```
shell# su - oracle
```

```
shell# sqlplus / as sysdba
SQL> CREATE USER zabbix IDENTIFIED BY zabbix DEFAULT TABLESPACE SYSTEM
TEMPORARY TABLESPACE TEMP PROFILE DEFAULT ACCOUNT UNLOCK;  #创建用户zabbix，密码为zabbix
SQL> GRANT CONNECT TO zabbix;
SQL> GRANT RESOURCE TO zabbix;
SQL> ALTER USER zabbix DEFAULT ROLE ALL;
SQL> GRANT SELECT ANY TABLE TO zabbix;
SQL> GRANT CREATE SESSION TO zabbix;
SQL> GRANT SELECT ANY DICTIONARY TO zabbix;
SQL> GRANT UNLIMITED TABLESPACE TO zabbix;
SQL> GRANT SELECT ANY DICTIONARY TO zabbix;
SQL> GRANT SELECT ON V_$SESSION TO zabbix;
SQL> GRANT SELECT ON V_$SYSTEM_EVENT TO zabbix;
SQL> GRANT SELECT ON V_$EVENT_NAME TO zabbix;
SQL> GRANT SELECT ON V_$RECOVERY_FILE_DEST TO zabbix;
接下来，我们使用zabbix_get来测试获取数据。

shell# zabbix_get -s 127.0.0.1 -k pyora[uptime]
21623423
```

472页

https://github.com/zabbix-book/Pyora/blob/master/zabbix-template/Pyora.xml

474页

```
开启JMX参数，配置如下：

shell# cd /opt/weblogic/user_projects/domains/webapp/
shell# vim bin/setDomainEnv.sh
export JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.rmi.server.hostname=本机的IP地址"
export JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote"
export JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.port=
10053"
export JAVA_OPTIONS="${JAVA_OPTIONS} -
Dcom.sun.management.jmxremote.authenticate= false"
export JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl= false"
```

475页

```
C:\> typeperf -qx
省略输出......
\MSRS 2008 R2 Windows Service(sqlexpress)\Report Requests
\MSRS 2008 R2 Windows Service(sqlexpress)\Total Reports Executed
```

```
\MSRS 2008 R2 Windows Service(sqlexpress)\Reports Executed/Sec
\MSRS 2008 R2 Windows Service(sqlexpress)\Total Processing Failures
\MSRS 2008 R2 Windows Service(sqlexpress)\Total Rejected Threads
\MSRS 2008 R2 Windows Service(sqlexpress)\Active Sessions
\MSRS 2008 R2 Windows Service(sqlexpress)\First Session Requests/Sec
\MSRS 2008 R2 Windows Service(sqlexpress)\Next Session Requests/Sec
\MSRS 2008 R2 Windows Service(sqlexpress)\Total Cache Hits
\MSRS 2008 R2 Windows Service(sqlexpress)\Cache Hits/Sec
\MSRS 2008 R2 Windows Service(sqlexpress)\Total Cache Misses
省略输出.....[这段数据缺少说明文字]
```

478

可以从https://github.com/zabbix-book/check-ssl-certificates获取所需的文件。

脚本内容如下：

```
shell# cat /etc/zabbix/scripts/check_ssl_certificates.py
#!/usr/bin/python
import socket
import ssl
import datetime
import argparse

def ssl_expiry_datetime(hostname):
    ssl_date_fmt = r'%b %d %H:%M:%S %Y %Z'

    context = ssl.create_default_context()
    conn = context.wrap_socket(socket.socket(socket.AF_INET),server_
hostname=hostname,)
    # 3 second timeout because Lambda has runtime limitations
    conn.settimeout(3.0)

    conn.connect((hostname, 443))
    ssl_info = conn.getpeercert()
    # parse the string from the certificate into a Python datetime object
    return datetime.datetime.strptime(ssl_info['notAfter'], ssl_date_fmt)

def ssl_valid_time_remaining(hostname):
    """Get the number of days left in a cert's lifetime."""
    expires = ssl_expiry_datetime(hostname)
    return expires - datetime.datetime.utcnow()

parser = argparse.ArgumentParser(description='returns the time before
expiration of certificate')
parser.add_argument("-c","--hostname", help="hostname of the certificate you
want to check")
args = parser.parse_args()
```

```
expire = ssl_valid_time_remaining(args.hostname)
print int(round(datetime.timedelta.total_seconds(expire), 0))
```

在本例中，我们只监控证书过期时间，对其他信息不再监控，比如还可以获取到如下相关信息。

```
{'OCSP': ('http://ocsp.digicert.com',),
 'caIssuers':
(u'http://cacerts.digicert.com/DigiCertSHA2ExtendedValidationServerCA.crt',),
 'crlDistributionPoints': (u'http://crl3.digicert.com/sha2-ev-server-g2.crl',
u'http://crl4.digicert.com/sha2-ev-server-g2.crl'),
 'issuer': ((('countryName', u'US'),),
  (('organizationName', u'DigiCert Inc'),),
  (('organizationalUnitName', u'www.digicert.com'),),
  (('commonName', u'DigiCert SHA2 Extended Validation Server CA'),)),
 'notAfter': 'Jun  3 12:00:00 2020 GMT',
 'notBefore': u'May  8 00:00:00 2018 GMT',
 'serialNumber': u'0A0630427F5BBCED6957396593B6451F',
 'subject': ((('businessCategory', u'Private Organization'),),
  (('1.3.6.1.4.1.311.60.2.1.3', u'US'),),
  (('1.3.6.1.4.1.311.60.2.1.2', u'Delaware'),),
  (('serialNumber', u'5157550'),),
  (('countryName', u'US'),),
  (('stateOrProvinceName', u'California'),),
  (('localityName', u'San Francisco'),),
  (('organizationName', u'GitHub, Inc.'),),
  (('commonName', u'github.com'),)),
 'subjectAltName': (('DNS', 'github.com'), ('DNS', 'www.github.com')),
 'version': 3L}
```

480页

```
{Template App SSL Certificate:ssl.certificates["www.github.com"].last()}
<2592000
```

481页

网络上有很多现成的监控模板，读者可以拿来直接使用，或者参考其实现思路，多去动手实践，举一反三。关于监控模板可参考如下地址：

https://github.com/zabbix-book/

https://www.zabbix.org/wiki/Zabbix_Templates

https://www.zabbix.com/integrations/

http://share.zabbix.com

https://github.com/monitoringartist/zabbix-community-repos