企业级Zabbix应用监控实践

# 

# 目录

[1.1 Zabbix监控TCP 4](#_Toc20053)

[1.1.1 编写Shell脚本 5](#_Toc20194)

[1.1.2 监控项linux\_tcp.conf的配置文件如下 6](#_Toc12484)

[1.1.3 重启zabbix-agent修改配置文件必须重启 6](#_Toc6835)

[1.1.4 Server测试Agent是否能获取到值，通过Zabbix\_get(不要直接执行脚本) 6](#_Toc16278)

[1.1.5添加所有监控项(记得将模板关联主机) 6](#_Toc13455)

[1.1.6 查看图形(图形是自定义创建) 7](#_Toc18348)

[1.2 Zabbix监控Nginx 7](#_Toc8554)

[1.2.1 实验环境 7](#_Toc24415)

[1.2.2 在nginx.conf的Server标签下添加如下内容 8](#_Toc30376)

[1.2.3 本地访问Nginx Status 8](#_Toc25921)

[1.2.4 编写Nginx的Shell脚本(如果端口不一致,只需要修改脚本端口即可) 8](#_Toc17481)

[1.2.5 给脚本添加执行权限 10](#_Toc3624)

[1.2.6 监控项nginx\_status.conf的配置文件如下： 10](#_Toc1368)

[1.2.7 重启zabbix-agent 11](#_Toc17490)

[1.2.8 使用Zabbix\_get来获取值 11](#_Toc28912)

[1.2.9 添加所有监控项, 如图4-3, 记得关联到指定的主机 11](#_Toc26850)

[1.3 Zabbix监控PHP-FPM 11](#_Toc6232)

[1.3.1实践环境 11](#_Toc20997)

[1.3.2 PHP-FPM工作模式通常与Nginx结合使用,修改php-fpm.conf 12](#_Toc25343)

[1.3.3 访问测试phpfpm\_status 12](#_Toc29568)

[1.3.4 编写php-fpm的Shell脚本(如果端口不一致,只需要修改脚本端口即可) 13](#_Toc16227)

[1.3.5 给脚本添加执行权限 17](#_Toc19577)

[1.3.6 监控项的phpfpm\_status.conf配置文件如下： 17](#_Toc22568)

[1.3.7 重启zabbix-agent 17](#_Toc19638)

[1.3.8 Server使用zabbix\_get命令来获取Agent端的值 17](#_Toc22250)

[1.3.9 添加所有监控项, 如下图4-5, 最后记得关联至对应主机 17](#_Toc13481)

[1.3.10 查看图形，如图4-4(图形自定义) 18](#_Toc19081)

[1.4 Zabbix监控Tomcat 18](#_Toc20224)

[1.4.1 JMX在Zabbix中的运行流程: 19](#_Toc13576)

[1.4.2 配置JMX监控的步骤: 19](#_Toc6036)

[1.4.3 实践环境 19](#_Toc9528)

[1.4.4.安装java以及zabbix-java-gateway (如果源码安装加上--enable-java参数) 19](#_Toc17127)

[1.4.7.修改zabbix-server 配置文件 20](#_Toc8888)

[1.4.8.重启zabbix-server 20](#_Toc29339)

[1.4.9.安装tomcat服务 20](#_Toc10015)

[1.4.10 开启tomcat的远程jvm配置文件 21](#_Toc1664)

[1.4.11 重启tomcat服务 21](#_Toc30022)

[1.4.12.zabbix添加tomcat主机,并添加Zabbix自带java监控模板，如图4-10、图4-11、图4-12 22](#_Toc14122)

[1.5 Zabbix监控MySQL 24](#_Toc26634)

[1.5.1 实践环境 24](#_Toc20021)

[1.5.2在Zabbix-Agent端安装percona Monitoring Plugins 24](#_Toc23555)

[1.5.3查看percona安装后的目录结构 24](#_Toc12698)

[1.5.4将自定义监控项配置文件复制至/etc/zabbix\_agentd.conf.d目录下 25](#_Toc13310)

[1.5.5 重启zabbix-agent 25](#_Toc25287)

[1.5.6 修改脚本中的MySQL用户名和密码 25](#_Toc13885)

[1.5.7在Zabbix-Server端上使用Zabbix\_get获取值(否则会失败) 26](#_Toc22358)

[1.5.8.在Zabbix页面模板选项中导入Percona模板, 模板存放在/var/lib/zabbix/percona/templates， 最后关联主机即可。 26](#_Toc12493)

[1.6 Zabbix监控Redis 26](#_Toc148)

[1.6.1实践环境 26](#_Toc1390)

[1.6.2.编写Shell脚本 27](#_Toc16445)

[1.6.3.添加脚本执行权限 32](#_Toc487)

[1.6.4.Zabbix权限不足处理办法 32](#_Toc15382)

[1.6.5.key的redis\_status.conf的配置文件如下： 32](#_Toc20404)

[1.6.6.重启zabbix-agent 32](#_Toc20296)

[1.6.7.在Zabbix-Server使用Zabbix\_get获取值 33](#_Toc24179)

# 1.1 Zabbix监控TCP

Tcp的连接状态对于我们web服务器来说是至关重要的，尤其是并发量ESTAB；或者是syn\_recv值，假如这个值比较大的话我们可以认为是不是受到了攻击，或是是time\_wait值比较高的话，我们要考虑看我们内核是否需要调优，太高的time\_wait值的话会占用太多端口，要是端口少的话后果不堪设想：所以今天我们来学学如何使用Zabbix监控tcp状态

## 编写Shell脚本

|  |
| --- |
| [root@linux-node1 ~]# cd /etc/zabbix/scripts  [root@linux-node1 scripts]# vim tcp\_status.sh  #!/bin/bash  ############################################################  # $Name: tcp\_status.sh  # $Version: v1.0  # $Function: TCP Status  # $Author: xuliangwei  # $organization: www.xuliangwei.com  # $Create Date: 2016-06-23  # $Description: Monitor TCP Service Status  ############################################################  [ $# -ne 1 ] && echo "Usage:CLOSE-WAIT|CLOSED|CLOSING|ESTAB|FIN-WAIT-1|FIN-WAIT-2|LAST-ACK|LISTEN|SYN-RECV SYN-SENT|TIME-WAIT" && exit 1  tcp\_status\_fun(){  TCP\_STAT=$1  ss -ant | awk 'NR>1 {++s[$1]} END {for(k in s) print k,s[k]}' > /tmp/ss.txt  TCP\_STAT\_VALUE=$(grep "$TCP\_STAT" /tmp/ss.txt | cut -d ' ' -f2)  if [ -z "$TCP\_STAT\_VALUE" ];then  TCP\_STAT\_VALUE=0  fi  echo $TCP\_STAT\_VALUE  }  tcp\_status\_fun $1;  添加执行权限  [root@linux-node1 scripts]# chmod +x tcp\_status.sh |

## 1.1.2 监控项linux\_tcp.conf的配置文件如下

[root@linux-node1 ~]# cat /etc/zabbix/zabbix\_agentd.d/tcp.conf

UserParameter=tcp\_status[\*],/bin/bash /etc/zabbix/scripts/tcp\_status.sh "$1"

## 1.1.3 重启zabbix-agent修改配置文件必须重启

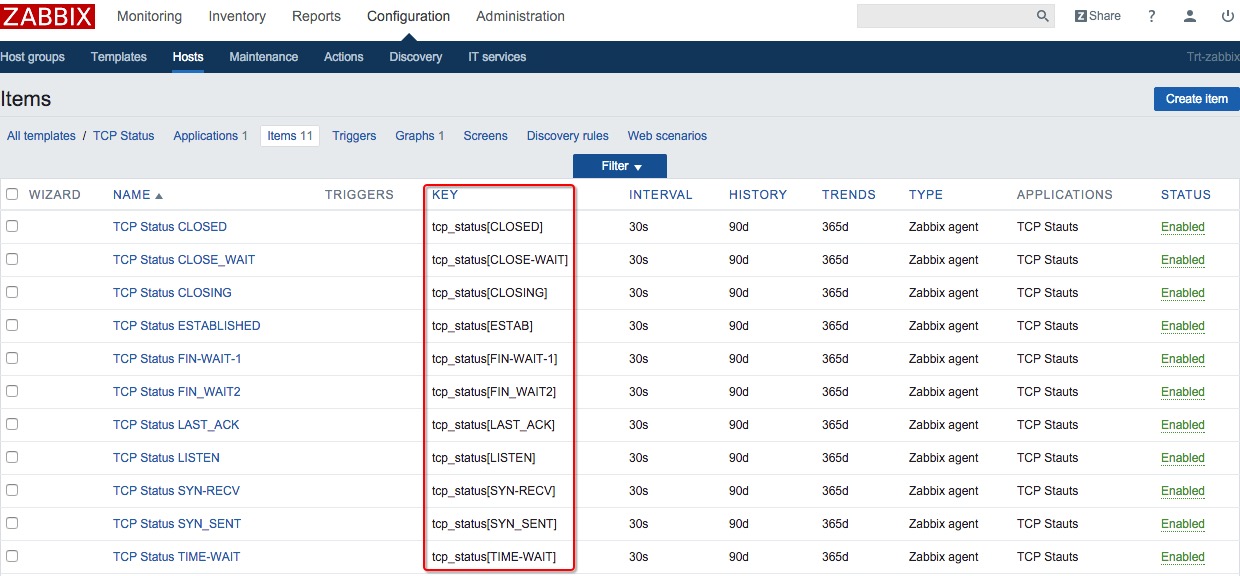
[root@linux-node1 ~]# systemctl restart zabbix-agent

## 1.1.4 Server测试Agent是否能获取到值，通过Zabbix\_get(不要直接执行脚本)

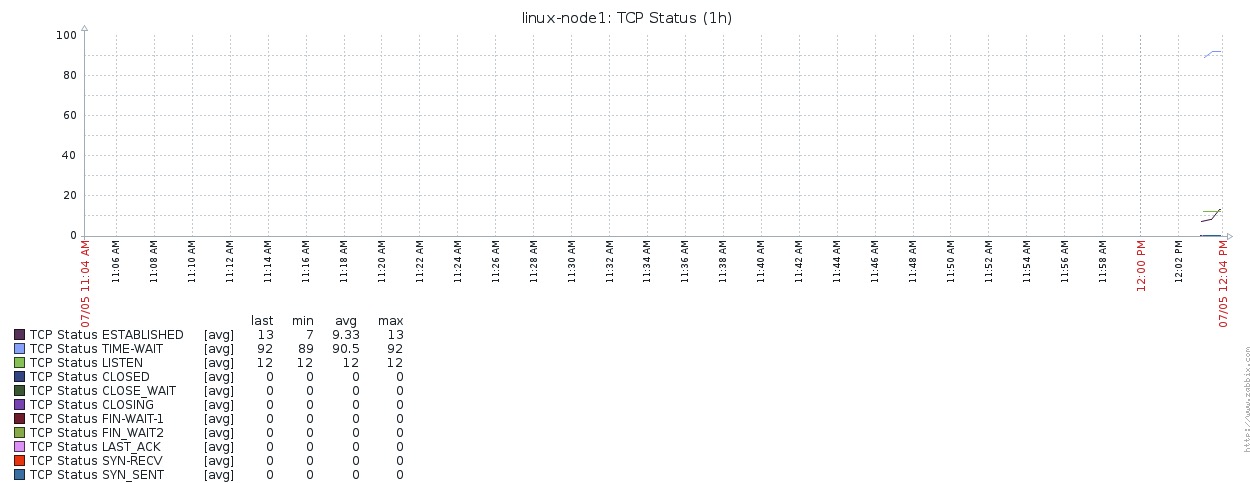
[root@linux-node1 scripts]# zabbix\_get -s 192.168.90.11 -k tcp\_status[ESTAB]

8

## 1.1.5添加所有监控项(记得将模板关联主机)



## 1.1.6 查看图形(图形是自定义创建)



# 1.2 Zabbix监控Nginx

## 1.2.1 实验环境

| **服务器系统** | **角色** | **IP** |
| --- | --- | --- |
| CentOS 7.4 x86\_64 | Zabbix-Server | 192.168.90.10 |
| CentOS 7.4 x86\_64 | Zabbix-Agent | 192.168.90.11 |

## 1.2.2 在nginx.conf的Server标签下添加如下内容

location /nginx\_status {

stub\_status on;

access\_log off;

allow 127.0.0.1;

deny all;

}

## 1.2.3 本地访问Nginx Status

[root@linux-node1 ~]# curl http://127.0.0.1/nginx\_status

Active connections: 1

server accepts handled requests

1 1 1

Reading: 0 Writing: 1 Waiting: 0

Nginx状态解释：

|  |
| --- |
| Active connections Nginx正处理的活动链接数1个  server Nginx启动到现在共处理了1个连接。  accepts Nginx启动到现在共成功创建1次握手。  handled requests Nginx总共处理了1次请求。  Reading Nginx读取到客户端的 Header 信息数。  Writing Nginx返回给客户端的 Header 信息数。  Waiting Nginx已经处理完正在等候下一次请求指令的驻留链接， |

开启。

Keep-alive的情况下，这个值等于active-（reading + writing）。

请求丢失数=(握手数-连接数)可以看出,本次状态显示没有丢失请求。

## 1.2.4 编写Nginx的Shell脚本(如果端口不一致,只需要修改脚本端口即可)

|  |
| --- |
| [root@Agent ~]# mkdir -p /etc/zabbix/scripts  [root@linux-node1 scripts]# vim /etc/zabbix/scripts/nginx\_status.sh  #!/bin/bash  ############################################################  # $Name: nginx\_status.sh  # $Version: v1.0  # $Function: Nginx Status  # $Author: xuliangwei  # $organization: www.xuliangwei.com,www,bjstack.com  # $Create Date: 2016-06-23  # $Description: Monitor Nginx Service Status  ############################################################  NGINX\_PORT=80 #如果端口不同仅需要修改脚本即可，否则修改xml很麻烦  NGINX\_COMMAND=$1  nginx\_active(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk '/Active/ {print $NF}'  }  nginx\_reading(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk '/Reading/ {print $2}'  }  nginx\_writing(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk '/Writing/ {print $4}'  }  nginx\_waiting(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk '/Waiting/ {print $6}'  }  nginx\_accepts(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk 'NR==3 {print $1}'  }  nginx\_handled(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk 'NR==3 {print $2}'  }  nginx\_requests(){  /usr/bin/curl -s "http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/" |awk 'NR==3 {print $3}'  }  case $NGINX\_COMMAND in  active)  nginx\_active;  ;;  reading)  nginx\_reading;  ;;  writing)  nginx\_writing;  ;;  waiting)  nginx\_waiting;  ;;  accepts)  nginx\_accepts;  ;;  handled)  nginx\_handled;  ;;  requests)  nginx\_requests;  ;;  \*)  echo $"USAGE:$0 {active|reading|writing|waiting|accepts|handled|requests}"  esac |

## 1.2.5 给脚本添加执行权限

[root@Agent]# chmod +x /etc/zabbix/scripts/nginx\_status.sh

## 1.2.6 监控项nginx\_status.conf的配置文件如下：

[root@Agent ~]# cat /etc/zabbix/zabbix\_agentd.d/nginx\_status.conf

UserParameter=nginx\_status[\*],/bin/bash /etc/zabbix/zabbix\_agentd.d/scripts/nginx/nginx\_status.sh "$1"

## 1.2.7 重启zabbix-agent

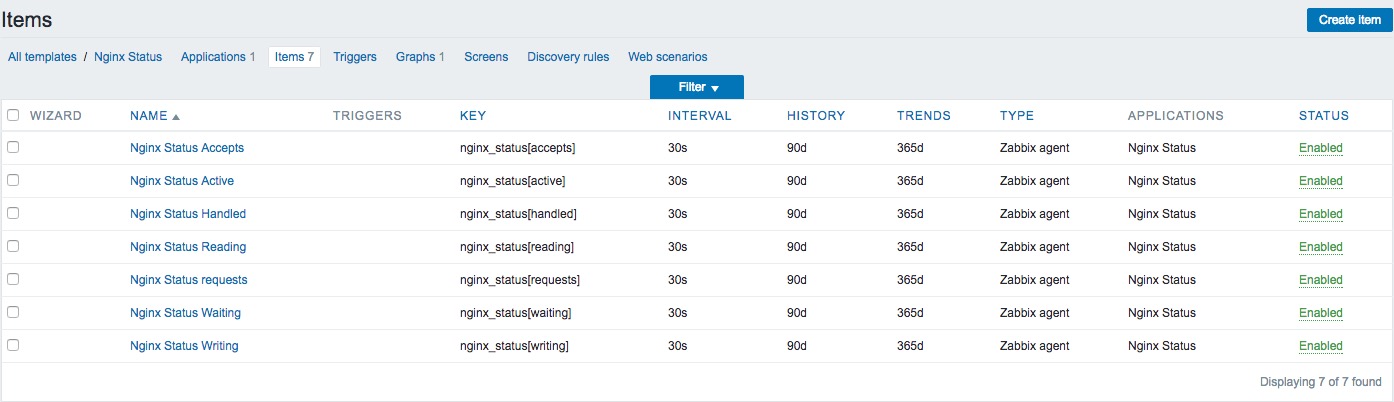
[root@Agent ~]# systemctl restart zabbix-agent

## 1.2.8 使用Zabbix\_get来获取值

[root@linux-node1 ~]# zabbix\_get -s 192.168.90.11 -k nginx\_status[writing]

1

## 1.2.9 添加所有监控项, 如图4-3, 记得关联到指定的主机



# 1.3 Zabbix监控PHP-FPM

## 1.3.1实践环境

| **服务器系统** | **角色** | **IP** |
| --- | --- | --- |
| CentOS 7.4 x86\_64 | Zabbix-Server | 192.168.90.10 |
| CentOS 7.4 x86\_64 | Zabbix-Agent | 192.168.90.11 |

## 1.3.2 PHP-FPM工作模式通常与Nginx结合使用,修改php-fpm.conf

[root@Agent ~]# vim /etc/php-fpm.d/www.conf

pm.status\_path = /phpfpm\_status

## 1.3.3 访问测试phpfpm\_status

[root@Agent ~]# curl http://127.0.0.1/phpfpm\_status

pool: www

process manager: dynamic

start time: 05/Jul/2016:15:30:56 +0800

start since: 409

accepted conn: 22

listen queue: 0

max listen queue: 0

listen queue len: 128

idle processes: 4

active processes: 1

total processes: 5

max active processes: 2

max children reached: 0

#PHP-FPM状态解释：

|  |
| --- |
| pool #fpm池名称,大多数为www  process manager #进程管理方式dynamic或者static  start time #启动日志,如果reload了fpm，时间会更新  start since #运行时间  accepted conn #当前池接受的请求数  listen queue #请求等待队列,如果这个值不为0,那么需要增加FPM的进程数量  max listen queue #请求等待队列最高的数量  listen queue len #socket等待队列长度  idle processes #空闲进程数量  active processes #活跃进程数量  total processes #总进程数量  max active processes #最大的活跃进程数量（FPM启动开始计算）  max children reached #程最大数量限制的次数，如果这个数量不为0，那说明你的最大进程数量过小,可以适当调整。 |

## 1.3.4 编写php-fpm的Shell脚本(如果端口不一致,只需要修改脚本端口即可)

|  |
| --- |
| [root@Agent ~]# cd /etc/zabbix/scripts  [root@Agent scripts]# vim phpfpm\_status.sh  #!/bin/bash  ############################################################  # $Name: phpfpm\_status.sh  # $Version: v1.0  # $Function: Nginx Status  # $Author: xuliangwei  # $organization: www.xuliangwei.com  # $Create Date: 2016-06-23  # $Description: Monitor Nginx Service Status  ############################################################  PHPFPM\_COMMAND=$1  PHPFPM\_PORT=80 #根据监听不同端口进行调整  start\_since(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^start since:/ {print $NF}'  }  accepted\_conn(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^accepted conn:/ {print $NF}'  }  listen\_queue(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^listen queue:/ {print $NF}'  }  max\_listen\_queue(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^max listen queue:/ {print $NF}'  }  listen\_queue\_len(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^listen queue len:/ {print $NF}'  }  idle\_processes(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^idle processes:/ {print $NF}'  }  active\_processes(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^active processes:/ {print $NF}'  }  total\_processes(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^total processes:/ {print $NF}'  }  max\_active\_processes(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^max active processes:/ {print $NF}'  }  max\_children\_reached(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^max children reached:/ {print $NF}'  }  slow\_requests(){  /usr/bin/curl -s "http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status" |awk '/^slow requests:/ {print $NF}'  }  case $PHPFPM\_COMMAND in  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;  ;;  slow\_requests)  slow\_requests;  ;;  \*)  echo $"USAGE:$0 {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 |

## 1.3.5 给脚本添加执行权限

[root@Agent scripts]# chmod +x phpfpm\_status.sh

## 1.3.6 监控项的phpfpm\_status.conf配置文件如下：

[root@Agent ~]# cat /etc/zabbix/zabbix\_agentd.d/phpfpm\_status.conf

UserParameter=phpfpm\_status[\*],/bin/bash /etc/zabbix/scripts/phpfpm\_status.sh "$1"

## 1.3.7 重启zabbix-agent

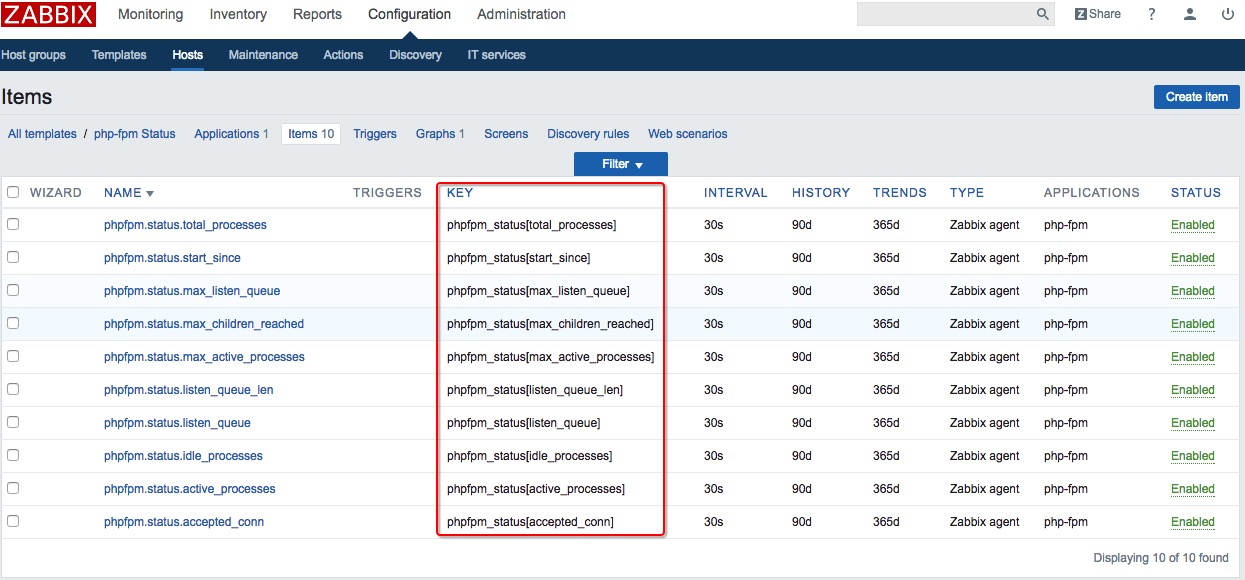
[root@Agent ~]# systemctl restart zabbix-agent

## 1.3.8 Server使用zabbix\_get命令来获取Agent端的值

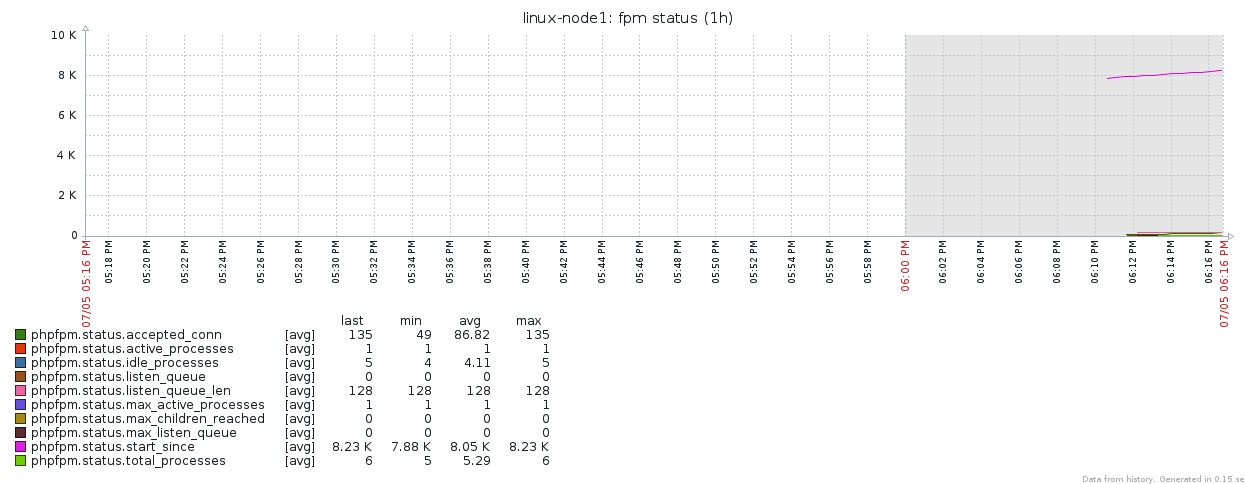
[root@linux-node1 zabbix\_agentd.d]# zabbix\_get -s 192.168.90.11 -k phpfpm\_status[accepted\_conn]

45

## 1.3.9 添加所有监控项, 如下图4-5, 最后记得关联至对应主机



## 1.3.10 查看图形，如图4-4(图形自定义)



## 1.4 Zabbix监控Tomcat

在Zabbix中，JMX监控数据的获取由专门的代理程序来实现,即Zabbix-Java-Gateway来负责数据的采集，Zabbix-Java-Gateway和JMX的Java程序之间通信获取数据

## 1.4.1 JMX在Zabbix中的运行流程:

1.Zabbix-Server找Zabbix-Java-Gateway获取Java数据

2.Zabbix-Java-Gateway找Java程序(zabbix-agent)获取数据

3.Java程序返回数据给Zabbix-Java-Gateway

4.Zabbix-Java-Gateway返回数据给Zabbix-Server

5.Zabbix-Server进行数据展示

## 1.4.2 配置JMX监控的步骤:

1.安装Zabbix-Java-Gateway。

2.配置zabbix\_java\_gateway.conf参数。

3.配置zabbix-server.conf参数。

4.Tomcat应用开启JMX协议。

5.ZabbixWeb配置JMX监控的Java应用。

## 1.4.3 实践环境

| **服务器系统** | **角色** | **IP** |
| --- | --- | --- |
| CentOS 7.4 x86\_64 | Zabbix-Server | 192.168.56.11 |
| CentOS 7.4 x86\_64 | Zabbix-java-gateway | 192.168.56.12 |
| CentOS 7.4 x86\_64 | Zabbix-Agent | 192.168.56.13 |

## 1.4.4.安装java以及zabbix-java-gateway (如果源码安装加上--enable-java参数)

//安装java-gateway

[root@linux-node1 ~]# yum install zabbix-java-gateway java-1.8.0-openjdk -y

1.4.5.配置zabbix-java-gateway

<保持为默认配置，就可以了>

vim /etc/zabbix/zabbix\_java\_gateway.conf

1.4.6.启动zabbix-java-gateway

[root@linux-node1 ~]# systemctl start zabbix-java-gateway

[root@linux-node1 ~]# netstat -lntup|grep 10052

tcp6 0 0 :::10052 :::\* LISTEN 13042/java

## 1.4.7.修改zabbix-server 配置文件

[root@linux-node1 ~]# vim /etc/zabbix/zabbix\_server.conf

#java gateway地址

JavaGateway=192.168.90.11

#java gateway默认端口10052

JavaGatewayPort=10052

#启动进程轮询java gateway

StartJavaPollers=5

## 1.4.8.重启zabbix-server

[root@linux-node1 ~]# systemctl restart zabbix-server

## 1.4.9.安装tomcat服务

mkdir /soft/package/src -p

wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.2/bin/apache-tomcat-9.0.2.tar.gz

tar xf apache-tomcat-9.0.2.tar.gz -C /soft/

ln -s /soft/apache-tomcat-9.0.2/ /soft/tomcat

## 1.4.10 开启tomcat的远程jvm配置文件

[root@linux-node1 ~]# vim /usr/local/tomcat/bin/catalina.sh

CATALINA\_OPTS="$CATALINA\_OPTS

-Dcom.sun.management.jmxremote

-Dcom.sun.management.jmxremote.port=12345

-Dcom.sun.management.jmxremote.authenticate=false

-Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.90.11"

#jvm配置文件解释

CATALINA\_OPTS="$CATALINA\_OPTS

//启用远程监控JMX

-Dcom.sun.management.jmxremote

//jmx启用远程端口,Zabbix添加时必须一致

-Dcom.sun.management.jmxremote.port=12345

//不开启用户密码认证

-Dcom.sun.management.jmxremote.authenticate=false

//不启用ssl加密传输

-Dcom.sun.management.jmxremote.ssl=false

//运行tomcat主机的IP地址

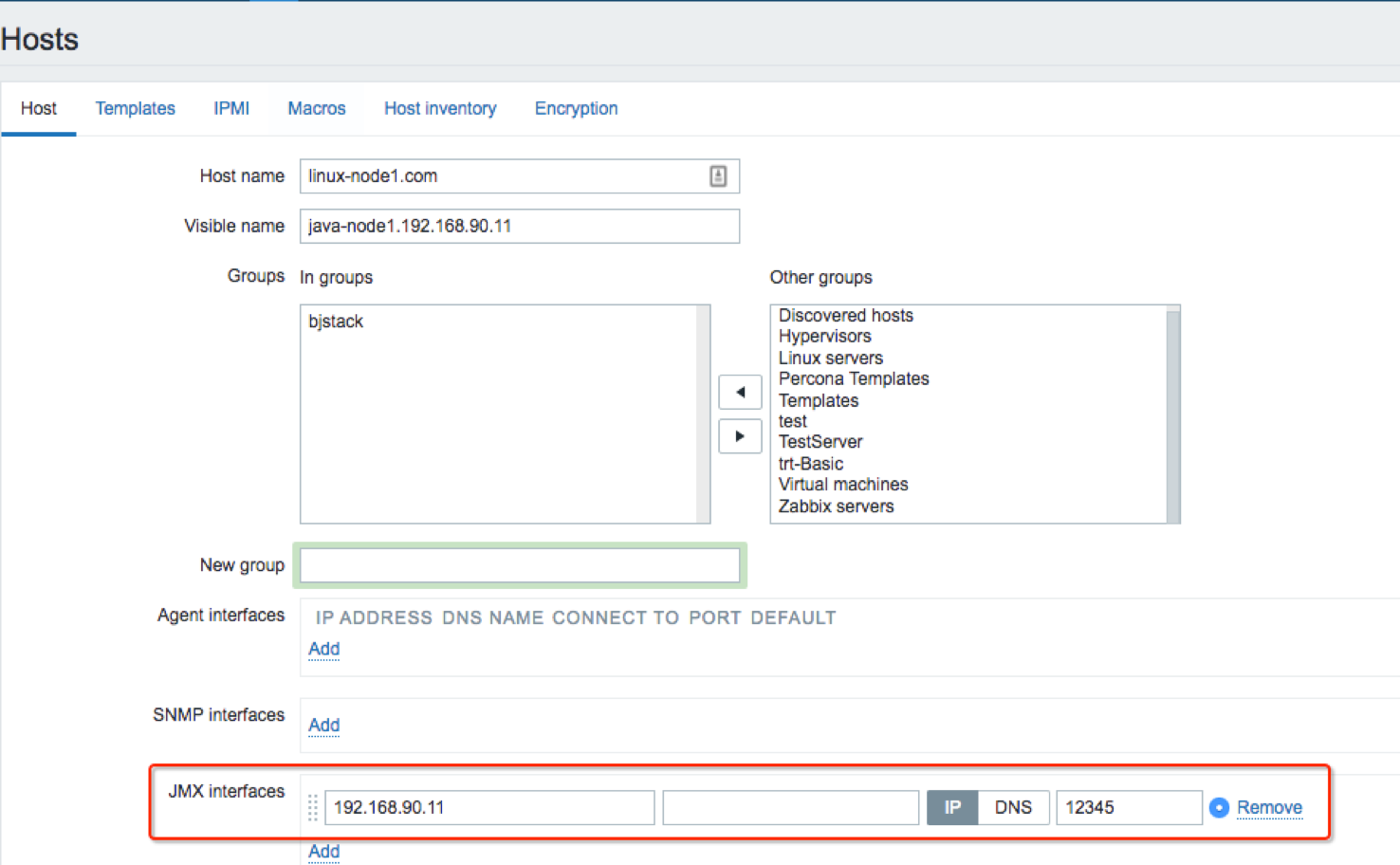
-Djava.rmi.server.hostname=192.168.90.11"

## 1.4.11 重启tomcat服务

[root@linux-node1 ~]# /usr/local/tomcat/bin/shutdown.sh

[root@linux-node1 ~]# /usr/local/tomcat/bin/startup.sh

## 1.4.12.zabbix添加tomcat主机,并添加Zabbix自带java监控模板，如图4-10、图4-11、图4-12



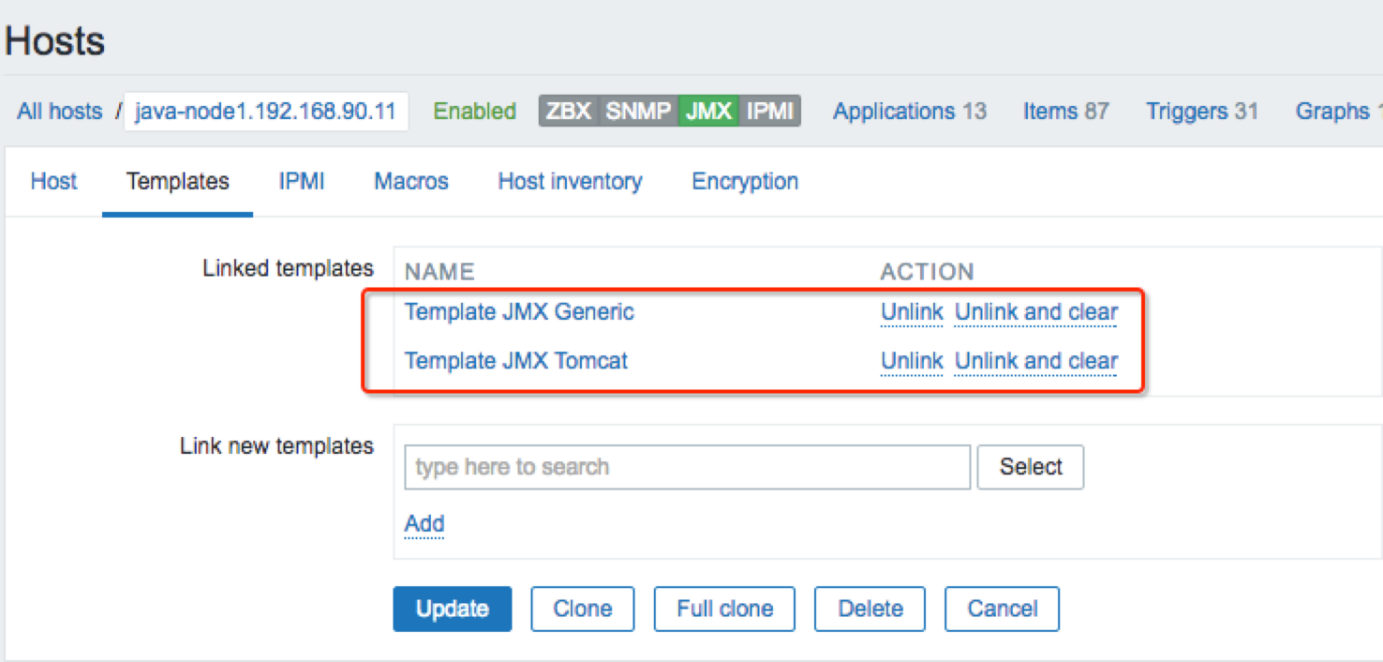


图4-11

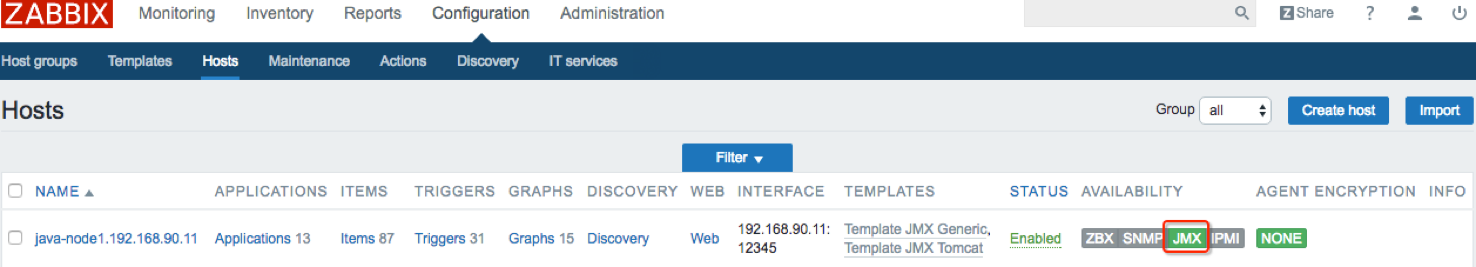
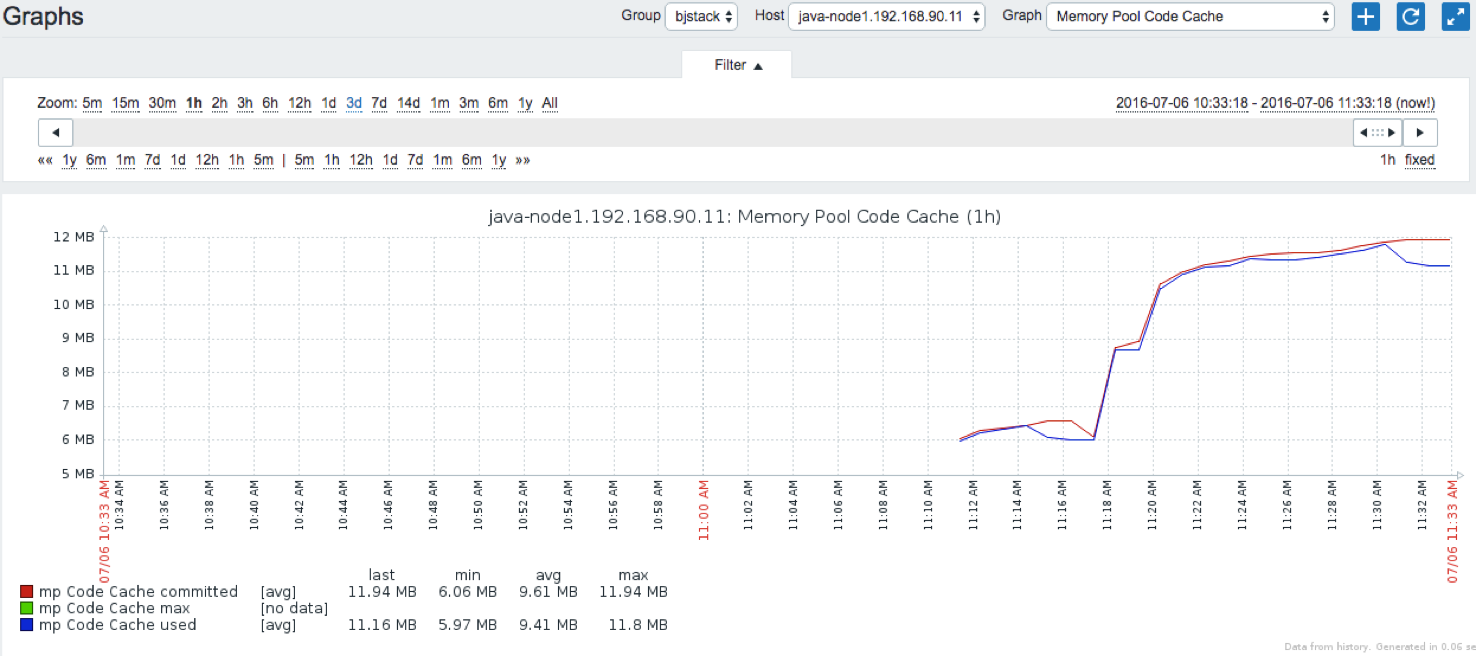


图4-12

1.4.12.查看图形，如图4-13  


自带的监控可能无法满足企业需求,大家可以根据公司的业务定制不同的JVM监控模板。

# 1.5 Zabbix监控MySQL

percona Monitoring Plugins是一个高质量的组件，为MySQL数据库添加企业级的监控和图表功能。但其脚本使用PHP实现，故而Zabbix-Agent需要安装PHP环境。

percona工具集

## 1.5.1 实践环境

| **服务器系统** | **角色** | **IP** |
| --- | --- | --- |
| CentOS 7.4 x86\_64 | Zabbix-Server | 192.168.90.10 |
| CentOS 7.4 x86\_64 | Zabbix-Agent | 192.168.90.11 |

## 1.5.2在Zabbix-Agent端安装percona Monitoring Plugins

[root@Agent ~]# yum install -y http://www.percona.com/downloads/percona-release/redhat/0.1-3/percona-release-0.1-3.noarch.rpm

[root@Agent ~]# yum install percona-zabbix-templates -y

## 1.5.3查看percona安装后的目录结构

[root@Agent percona]# tree /var/lib/zabbix/percona

/var/lib/zabbix/percona

├── scripts #脚本文件路径

│ ├── get\_mysql\_stats\_wrapper.sh

│ └── ss\_get\_mysql\_stats.php

└── templates

├── userparameter\_percona\_mysql.conf #key文件位置

└── zabbix\_agent\_template\_percona\_mysql\_server\_ht\_2.0.9-sver1.1.6.xml #模板文件位置

## 1.5.4将自定义监控项配置文件复制至/etc/zabbix\_agentd.conf.d目录下

[root@Agent ~]# cp /var/lib/zabbix/percona/templates/userparameter\_percona\_mysql.conf /etc/zabbix/zabbix\_agentd.d/percona\_mysql.conf

## 1.5.5 重启zabbix-agent

[root@Agent ~]# systemctl restart zabbix-agent

## 1.5.6 修改脚本中的MySQL用户名和密码

[root@Agent scripts]# vim /var/lib/zabbix/percona/scripts/ss\_get\_mysql\_stats.php

$mysql\_user = 'root';

$mysql\_pass = 'xuliangwei.com';

$mysql\_port = 3306;

## 1.5.7在Zabbix-Server端上使用Zabbix\_get获取值(否则会失败)

[root@Server ~]# zabbix\_get -s 192.168.90.11 -k MySQL.pool-read-requests

223003813

//如果获取不到值常见问题

1.看是否是MySQL密码错误

2.不要直接执行脚本来获取

3.删除/tmp/localhost-mysql\_cacti\_stats.txt文件

4.权限问题导致

8.在Zabbix页面模板选项中导入Percona模板, 模板存放在/var/lib/zabbix/percona/templates， 最后关联主机即可。

## 1.5.8.在Zabbix页面模板选项中导入Percona模板, 模板存放在/var/lib/zabbix/percona/templates， 最后关联主机即可。

# 1.6 Zabbix监控Redis

Redis使用自带的INFO命令，进行状态监控。以一种易于解释且易于阅读的格式，返回关于Redis服务器的各种信息和统计数值。

## 1.6.1实践环境

| **服务器系统** | **角色** | **IP** |
| --- | --- | --- |
| CentOS 7.4 x86\_64 | Zabbix-Server | 192.168.90.10 |
| CentOS 7.4 x86\_64 | Zabbix-Agent | 192.168.90.11 |

## 1.6.2.编写Shell脚本

|  |
| --- |
| 脚本端口、连接redis服务地址根据具体情况进行修改  AUTH认证没有开启，将PASSWD修改为空即可。  [root@Agent ~]# mkdir -p /etc/zabbix/scripts  [root@Agent ~]# vim /etc/zabbix/scripts/redis\_status.sh  #!/bin/bash  ############################################################  # $Name: redis\_status.sh  # $Version: v1.0  # $Function: Redis Status  # $Author: xuliangwei  # $organization: www.xuliangwei.com  # $Create Date: 2016-06-23  # $Description: Monitor Redis Service Status  ############################################################  R\_COMMAND="$1"  R\_PORT="6379" #根据实际情况调整端口  R\_SERVER="127.0.0.1" #根据具体情况调整IP地址  PASSWD="" #如果没有设置Redis密码,为空即可  redis\_status(){  (echo -en "AUTH $PASSWD\r\nINFO\r\n";sleep 1;) | /usr/bin/nc "$R\_SERVER" "$R\_PORT" > /tmp/redis\_"$R\_PORT".tmp  REDIS\_STAT\_VALUE=$(grep "$R\_COMMAND:" /tmp/redis\_"$R\_PORT".tmp | cut -d ':' -f2)  echo "$REDIS\_STAT\_VALUE"  }  case $R\_COMMAND in  used\_cpu\_user\_children)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  used\_cpu\_sys)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  total\_commands\_processed)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  role)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  lru\_clock)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  latest\_fork\_usec)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  keyspace\_misses)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  keyspace\_hits)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  keys)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  expires)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  expired\_keys)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  evicted\_keys)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  connected\_clients)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  changes\_since\_last\_save)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  blocked\_clients)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  bgsave\_in\_progress)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  bgrewriteaof\_in\_progress)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  used\_memory\_peak)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  used\_memory)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  used\_cpu\_user)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  used\_cpu\_sys\_children)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  total\_connections\_received)  redis\_status "$R\_PORT" "$R\_COMMAND"  ;;  \*)  echo $"USAGE:$0 {used\_cpu\_user\_children|used\_cpu\_sys|total\_commands\_processed|role|lru\_clock|latest\_fork\_usec|keyspace\_misses|keyspace\_hits|keys|expires|expired\_keys|connected\_clients|changes\_since\_last\_save|blocked\_clients|bgrewriteaof\_in\_progress|used\_memory\_peak|used\_memory|used\_cpu\_user|used\_cpu\_sys\_children|total\_connections\_received}"  esac |

Redis状态参数解释：

|  |
| --- |
| server : Redis 服务器信息，包含以下域：  redis\_version : Redis 服务器版本  redis\_git\_sha1 : Git SHA1  redis\_git\_dirty : Git dirty flag  os : Redis 服务器的宿主操作系统  arch\_bits : 架构（32 或 64 位）  multiplexing\_api : Redis 所使用的事件处理机制  gcc\_version : 编译 Redis 时所使用的 GCC 版本  process\_id : 服务器进程的 PID  run\_id : Redis 服务器的随机标识符（用于 Sentinel 和集群）  tcp\_port : TCP/IP 监听端口  uptime\_in\_seconds : 自 Redis 服务器启动以来，经过的秒数  uptime\_in\_days : 自 Redis 服务器启动以来，经过的天数  lru\_clock : 以分钟为单位进行自增的时钟，用于 LRU 管理  clients : 已连接客户端信息，包含以下域：  connected\_clients : 已连接客户端的数量（不包括通过从属服务器连接的客户端）  client\_longest\_output\_list : 当前连接的客户端当中，最长的输出列表  client\_longest\_input\_buf : 当前连接的客户端当中，最大输入缓存  blocked\_clients : 正在等待阻塞命令（BLPOP、BRPOP、BRPOPLPUSH）的客户端的数量  memory : 内存信息，包含以下域：  used\_memory : 由 Redis 分配器分配的内存总量，以字节（byte）为单位  used\_memory\_human : 以人类可读的格式返回 Redis 分配的内存总量  used\_memory\_rss : 从操作系统的角度，返回 Redis 已分配的内存总量（俗称常驻集大小）。这个值和 top 、 ps 等命令的输出一致。  used\_memory\_peak : Redis 的内存消耗峰值（以字节为单位）  used\_memory\_peak\_human : 以人类可读的格式返回 Redis 的内存消耗峰值  used\_memory\_lua : Lua 引擎所使用的内存大小（以字节为单位）  mem\_fragmentation\_ratio : used\_memory\_rss 和 used\_memory 之间的比率  persistence : RDB 和 AOF 的相关信息  stats : 一般统计信息  replication : 主/从复制信息  cpu : CPU 计算量统计信息  commandstats : Redis 命令统计信息  cluster : Redis 集群信息  keyspace : 数据库相关的统计信息  参数还可以是下面这两个：  all : 返回所有信息  default : 返回默认选择的信息 |

当不带参数直接调用 INFO 命令时，使用 default 作为默认参数。

## 1.6.3.添加脚本执行权限

[root@Agent ~]# chmod +x /etc/zabbix/scripts/redis\_status.sh

## 1.6.4.Zabbix权限不足处理办法

[root@Agent ~]# rm -f /tmp/redis\_6379.tmp

## 1.6.5.key的redis\_status.conf的配置文件如下：

[root@Agent ~]# cat /etc/zabbix/zabbix\_agentd.d/redis\_status.conf

UserParameter=redis\_status[\*],/bin/bash /etc/zabbix/scripts/redis\_status.sh "$1"

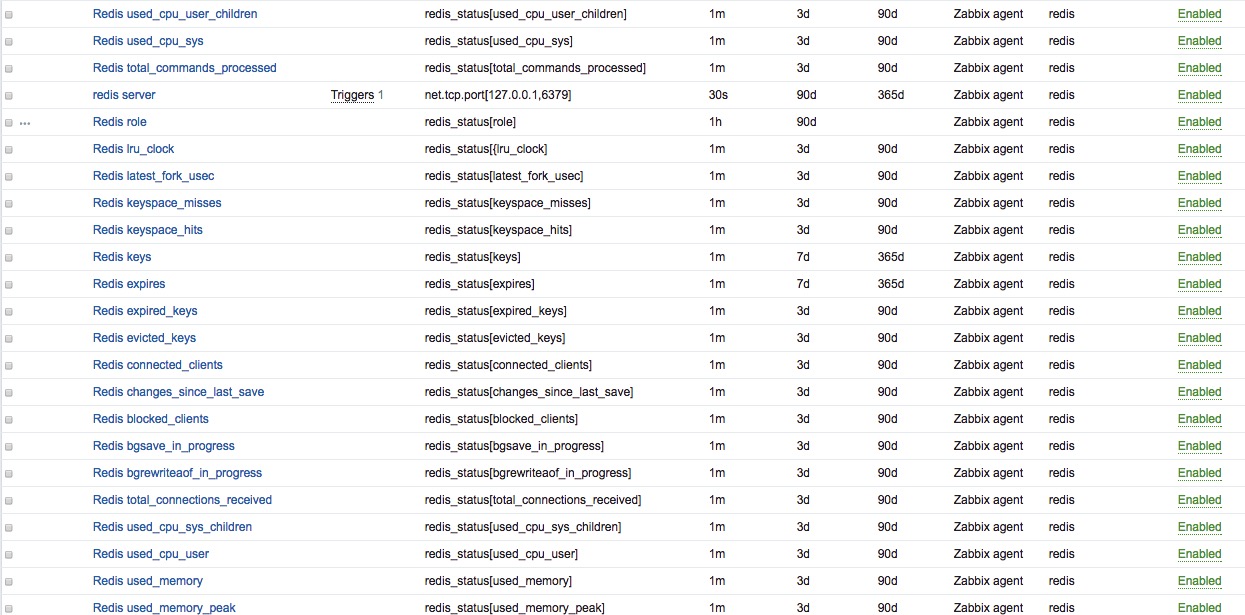
## 1.6.6.重启zabbix-agent

[root@Agent ~]# systemctl restart zabbix-agent

## 1.6.7.在Zabbix-Server使用Zabbix\_get获取值

[root@Server ~]# zabbix\_get -s 192.168.90.11 -k redis\_status[used\_cpu\_sys]

16.81

1.6.8.展示所有Key(记得将模板关联主机)如图4-14  


1.6.9.查看图形，如图4-15、图4-16(图形自定义)

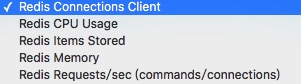


图4-15

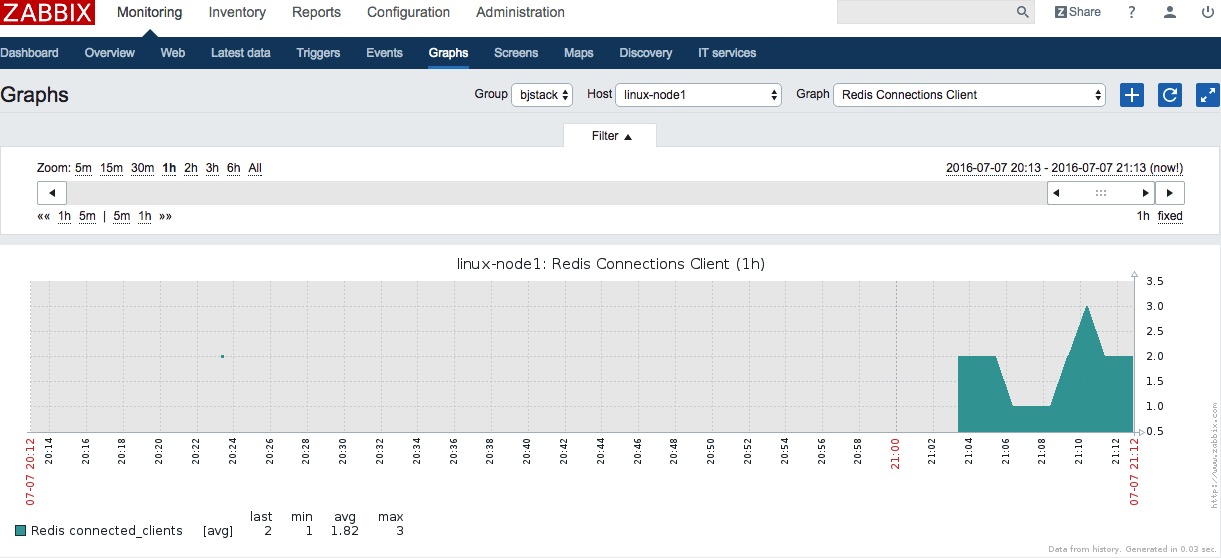


图4-16

# 1.7 zabbix监控系统内存

## 1.7.1 客户端agent，监控项的配置

## ### Option: Include

# You may include individual files or all files in a directory in the configuration file.

# Installing Zabbix will create include directory in /usr/local/etc, unless modified during the compile time.

#

# Mandatory: no

# Default:

# Include=

#启用Include=/etc/zabbix/zabbix\_agentd.d/

Include=/etc/zabbix/zabbix\_agentd.d/

### Option: EnableRemoteCommands

# Whether remote commands from Zabbix server are allowed.

# 0 - not allowed

# 1 - allowed

#

# Mandatory: no

# Default:

#启用EnableRemoteCommands选项，并设置该选项的值为1

EnableRemoteCommands=1

[root@linux-node2 zabbix\_agentd.d]# service zabbix-agent restart

Shutting down Zabbix agent: [FAILED]

Starting Zabbix agent: [ OK ]

[root@linux-node2 zabbix\_agentd.d]# vi /etc/zabbix/zabbix\_agentd.d/meminfo.conf

#monitor mem info

UserParameter=mem.status[\*],/bin/cat /proc/meminfo|awk '/^$1/{print $$2}'

## 1.7.2 服务器端测试

## [root@linux-node1 ~]# zabbix\_get -s 192.168.100.101 -k mem.status[Buffers]

## 129496

## 1.7.3 zabbix服务器配置item项

