

zabbix 4.4.7 监控 redis

作者: [Leif160519](#)

原文链接: <https://ld246.com/article/1585880545206>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1.创建redis监控脚本

1.1 创建脚本

在/etc/zabbix/zabbix_agentd.d/下新建文件redis_status.sh,内容如下:

```
#!/bin/bash
REDISCLI="/usr/bin/redis-cli"
HOST="127.0.0.1"
PORT=6379
PASS=""

if [[ $# == 1 ]];then
    case $1 in
        version)
            result=`$REDISCLI -h $HOST -p $PORT info server | grep -w "redis_version" | awk -F:' '
print $2}`
            echo $result
            ;;
        uptime)
            result=`$REDISCLI -h $HOST -p $PORT info server | grep -w "uptime_in_seconds" | aw
-F:' '{print $2}`
            echo $result
            ;;
        connected_clients)
            result=`$REDISCLI -h $HOST -p $PORT info clients | grep -w "connected_clients" | awk
-F:' '{print $2}`
            echo $result
            ;;
        blocked_clients)
```

```

        result=`$REDISCLI -h $HOST -p $PORT info clients | grep -w "blocked_clients" | awk -F'
' '{print $2}'`
        echo $result
        ;;
        used_memory)
        result=`$REDISCLI -h $HOST -p $PORT info memory | grep -w "used_memory" | awk -
:' '{print $2}'`
        echo $result
        ;;
        used_memory_rss)
        result=`$REDISCLI -h $HOST -p $PORT info memory | grep -w "used_memory_rss" | a
k -F:' '{print $2}'`
        echo $result
        ;;
        used_memory_peak)
        result=`$REDISCLI -h $HOST -p $PORT info memory | grep -w "used_memory_peak" |
awk -F:' '{print $2}'`
        echo $result
        ;;
        used_memory_lua)
        result=`$REDISCLI -h $HOST -p $PORT info memory | grep -w "used_memory_lua" | a
k -F:' '{print $2}'`
        echo $result
        ;;
        used_cpu_sys)
        result=`$REDISCLI -h $HOST -p $PORT info cpu | grep -w "used_cpu_sys" | awk -F:' '{p
int $2}'`
        echo $result
        ;;
        used_cpu_user)
        result=`$REDISCLI -h $HOST -p $PORT info cpu | grep -w "used_cpu_user" | awk -F:' '
print $2}'`
        echo $result
        ;;
        used_cpu_sys_children)
        result=`$REDISCLI -h $HOST -p $PORT info cpu | grep -w "used_cpu_sys_children" | a
k -F:' '{print $2}'`
        echo $result
        ;;
        used_cpu_user_children)
        result=`$REDISCLI -h $HOST -p $PORT info cpu | grep -w "used_cpu_user_children" |
wk -F:' '{print $2}'`
        echo $result
        ;;
        rdb_last_bgsave_status)
        result=`$REDISCLI -h $HOST -p $PORT info Persistence | grep -w "rdb_last_bgsave_sta
us" | awk -F:' '{print $2}' | grep -c ok`
        echo $result
        ;;
        aof_last_bgrewrite_status)
        result=`$REDISCLI -h $HOST -p $PORT info Persistence | grep -w "aof_last_bgrewrite_
tatus" | awk -F:' '{print $2}' | grep -c ok`
        echo $result
        ;;

```

```

aof_last_write_status)
    result=`$REDISCLI -h $HOST -p $PORT info Persistence | grep -w "aof_last_write_statu
" | awk -F'|' '{print $2}' | grep -c ok`
    echo $result
    ;;
*)
    echo -e "\033[33mUsage: $0 {connected_clients|blocked_clients|used_memory|used_m
emory_rss|used_memory_peak|used_memory_lua|used_cpu_sys|used_cpu_user|used_cpu_sys_ch
ldren|used_cpu_user_children|rdb_last_bgsave_status|aof_last_bgrewrite_status|aof_last_write_
tatus}\033[0m"
    ;;
esac
elif [[ $# == 2 ]];then
    case $2 in
        keys)
            result=`$REDISCLI -h $HOST -p $PORT info | grep -w "$1" | grep -w "keys" | awk -F=|,'
'{print $2}'`
            echo $result
            ;;
        expires)
            result=`$REDISCLI -h $HOST -p $PORT info | grep -w "$1" | grep -w "keys" | awk -F=|,'
'{print $4}'`
            echo $result
            ;;
        avg_ttl)
            result=`$REDISCLI -h $HOST -p $PORT info | grep -w "$1" | grep -w "avg_ttl" | awk -F'
|,' '{print $6}'`
            echo $result
            ;;
    *)
        echo -e "\033[33mUsage: $0 {db0 keys|db0 expires|db0 avg_ttl}\033[0m"
        ;;
    esac
fi

```

注意：若redis有密码，则需要要在\$HOST后加上-a \$PASS参数。

1.2 赋予脚本可执行权限

```
chmod +x /etc/zabbix/zabbix_agentd.d/redis_status.sh
```

1.3 脚本测试

```
[root@test-server zabbix_agentd.d]# bash redis_status.sh version
3.2.12
```

2.创建redis监控配置文件

2.1 创建文件

在 /etc/zabbix/zabbix_agentd.d下新建redis.conf文件，内容如下：

```
UserParameter=Redis.Status,status=`usr/local/bin/redis-cli -h 127.0.0.1 -p 6379 ping|grep -c  
ONG` &&echo $status  
UserParameter=Redis.Info[*],/etc/zabbix/zabbix_agentd.d/redis_status.sh $1 $2
```

2.2 重启zabbix-agent:

systemctl restart zabbix-agent

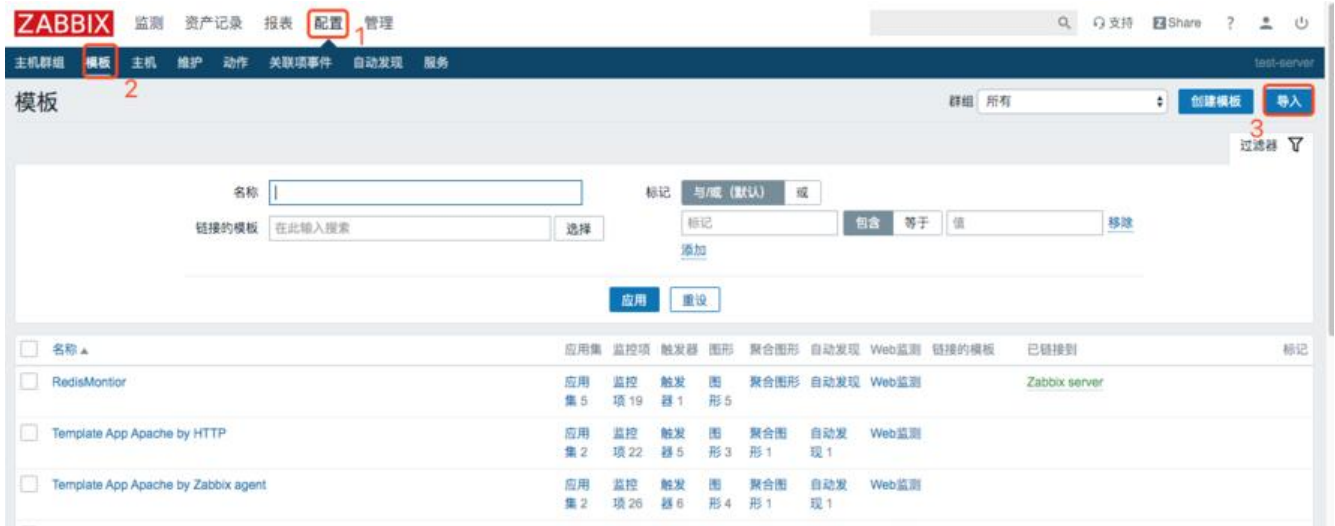
3.创建并导入监控模板

3.1 创建监控模板

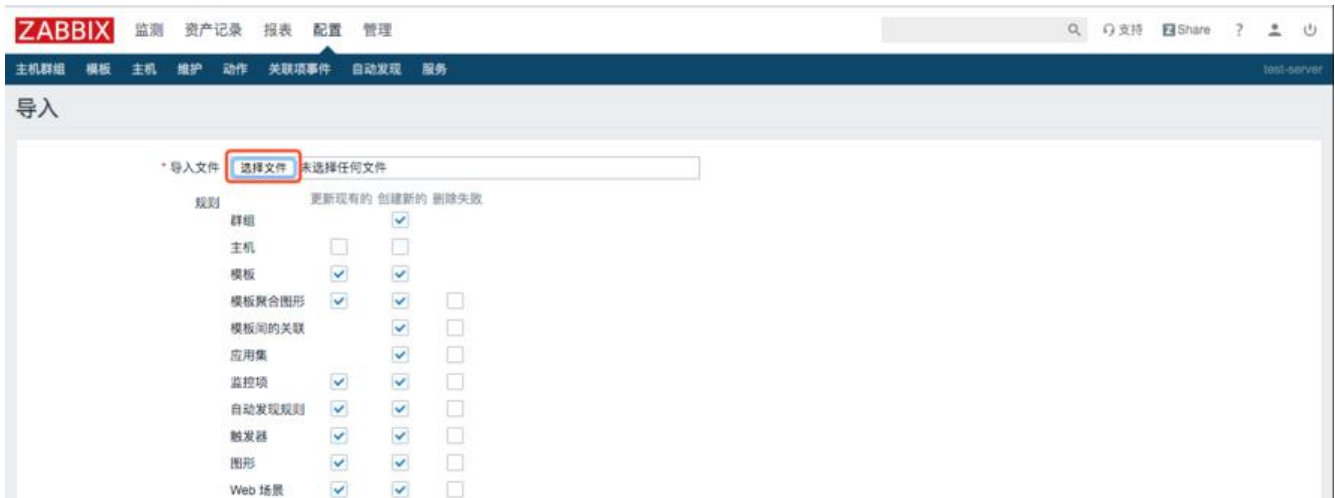
redis-template.xml文件内容参考github:

3.2 导入监控模板（不用zabbix自带的redis模板）

配置-模板-导入

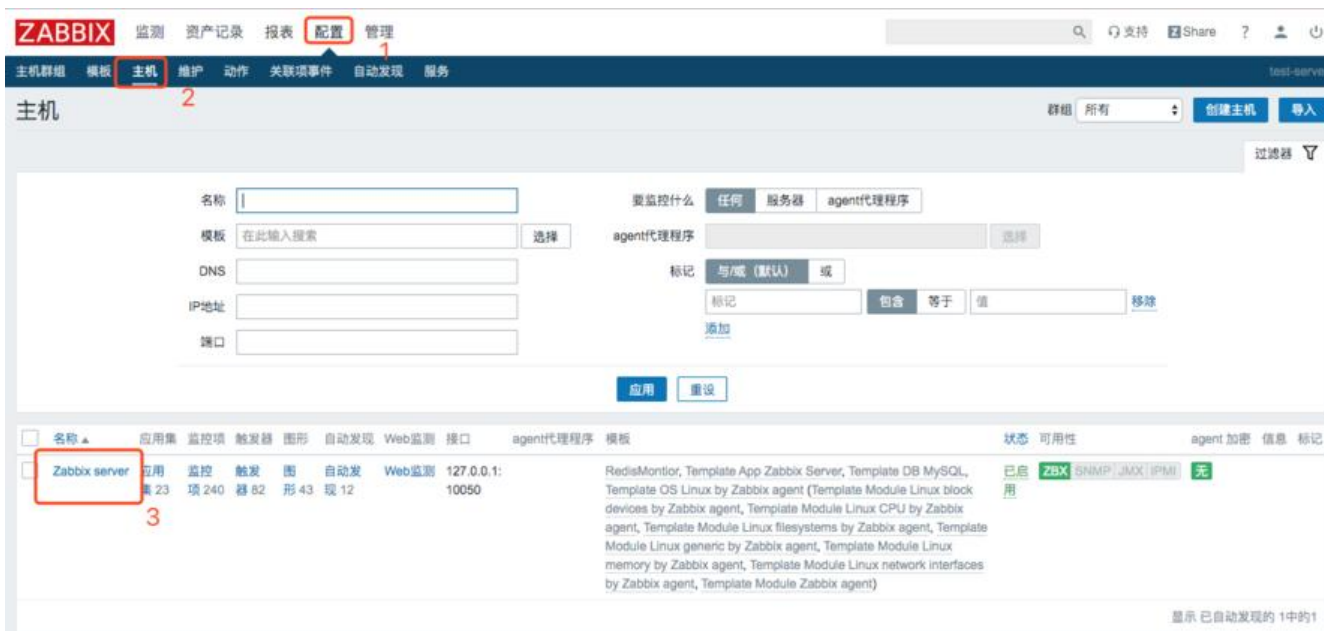


点击选择文件，找到redis-template.xml文件，将其导入





4.给主机添加监控模板



主机

所有主机 / Zabbix server 已启用 ZBX SNMP JMX IPMI 应用集 23 监控项 240 触发器 82 图形 43 自动发现规则 12 Web 场景

主机 模板 IPMI 标记 宏 资产记录 加密

链接的模板

| 名称 | 动作 |
|-----------------------------------|--|
| RedisMontior | 取消链接 取消链接并清理 |
| Template App Zabbix Server | 取消链接 取消链接并清理 |
| Template DB MySQL | 取消链接 取消链接并清理 |
| Template OS Linux by Zabbix agent | 取消链接 取消链接并清理 |

Link new templates

在此输入搜索

选择 1

更新 克隆 全克隆 删除 取消

3

模板 群组 RedisMontior

名称

RedisMontior 选择 取消

2

监控效果如图：



5.参考

[线上zabbix监控redis和redis集群](#)

有兴趣的可以尝试官方提供的redis监控方式, [github地址](#)