

一、主被动监控

P1 主动监控

创建新主机，给 web2 安装 zabbix_agent 软件

```
[root@zabbixserver ~]# scp -r zabbix-3.4.4 root@192.168.2.200:/root/
```

```
[root@web2 ~]# yum -y install gcc pcre-devel autoconf
```

```
[root@web2 ~]# cd zabbix-3.4.4/
```

```
[root@web2 zabbix-3.4.4]# ./configure --enable-agent
```

```
[root@web2 zabbix-3.4.4]# make install
```

以上操作如果已经在前面完成，则不需要重复再做

修改配置文件，设置主动监控(自己收集数据并发送给 zabbixserver 服务器)

```
[root@web2 ~]# vim /usr/local/etc/zabbix_agentd.conf
```

```
93 #Server=127.0.0.1          #注释掉此行
```

```
118 StartAgents=0           #取消注释，修改参数为 0，禁止被动监控
```

```
134 ServerActive=192.168.2.5
```

```
145 Hostname=web2
```

```
183 RefreshActiveChecks=120 #指 web2 每隔 120 秒，向监控服务器发送一次监控数据
```

重启服务

```
[root@web2 zabbix-3.4.4]# yum -y install psmisc
```

```
[root@web2 zabbix-3.4.4]# killall zabbix_agentd
```

```
[root@web2 zabbix-3.4.4]# zabbix_agentd
```

查看 zabbix 对应的进程号，确定服务的开启状态

```
[root@web2 ~]# ps aux | grep zabbix
```

克隆模板

1) 选择 “配置”，选择 “模板”，点击 “Template OS Linux”



选择全克隆

描述

更新

克隆

全克隆

删除

删除并清除

取消

3) 修改新模板的名称为“Template OS Linux Active”，选择“添加”

模板

所有模板 / Template OS Linux 应用集 10 监控项 32 触发器 15 图形 5 聚合图形 1 自动发现规则 2 Web 场景

模板 链接的模板 宏

模板名称

可见的名称

群组 在...群组之中 其它群组

聚合图形

2

添加

取消

4) 查看新克隆的监控模板

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

模板 过滤器 ▲

名称

<input type="checkbox"/>	Template OS HP-UX	应用集 10	监控项 17	触发器 8	图形 3	聚合图形 1	自动发现 2	Web监测	Template App Zabbix Agent
<input type="checkbox"/>	Template OS Linux	应用集 10	监控项 32	触发器 15	图形 5	聚合图形 1	自动发现 2	Web监测	Template App Zabbix Agent
<input checked="" type="checkbox"/>	Template OS Linux Active	应用集 10	监控项 32	触发器 15	图形 5	聚合图形 1	自动发现 2	Web监测	Template App Zabbix Agent

步骤三：修改监控项模式

1) 选择“配置”，选择“模板”，点击新克隆模板的“监控项”

ZABBIX 监测中 资产记录 报表 **配置** 管理

主机群组 **模板** 主机 维护 动作 关联项事件 自动发现 服务

模板

过滤器 ▲

名称

<input type="checkbox"/>	Template OS Linux	应用集 10	监控项 32	触发器 15	图形 5	聚合图形 1	自动发现 2	Web监测	Template App Zabbix Agent
<input checked="" type="checkbox"/>	Template OS Linux Active	应用集 10	监控项 32	触发器 15	图形 5	聚合图形 1	自动发现 2	Web监测	Template App Zabbix Agent
<input type="checkbox"/>	Template OS Linux SNMPv2	应用集 7	监控项 11	触发器 5	图形	聚合图形	自动发现 5	Web监测	Template Module EtherLike-MIB SNMPv2, Template Module Generic SNMPv2, Template Module HOST- RESOURCES-MIB SNMPv2, Templ

2) 全选，拉到最后选择“批量更新”

☒ Wizard 名称 ▼

	触发器	键值	间隔	历史记	
<input checked="" type="checkbox"/> ... Template App Zabbix Agent: Version of zabbix_agent(d) running	触发器 1	agent.version	1h	1w	
<input checked="" type="checkbox"/> ... Total swap space		system.swap.size[,total]	1h	1w	
<input checked="" type="checkbox"/> ... Total memory		vm.memory.size[total]	1h	1w	
<input checked="" type="checkbox"/> ... System uptime	触发器 1	system.uptime	10m	1w	
<input checked="" type="checkbox"/> ... Checksum of /etc/passwd	触发器 1	vfs.file.cksum[/etc/passwd]	1h	1w	365
<input checked="" type="checkbox"/> ... Available memory	触发器 1	vm.memory.size[available]	1m	1w	365
<input checked="" type="checkbox"/> ... Template App Zabbix Agent: Agent ping	触发器 1	agent.ping	1m	1w	365

32 选择

3) 选择类型“Zabbix 客户端(主动式)”，拉到最后选择“更新”

监控项

所有模板 / Template OS Linux Active 应用集 10 **监控项 32** 触发器 15 图形 5 聚合图形 1 自动发现规则 2 Web 场景

类型 ☒ **Zabbix客户端(主动式)**

JMX endpoint ☐ 原始的

SNMP community ☐ 原始的

上下文名称 ☐ 原始的

安全名称 ☐ 原始的

安全级别 ☐ 原始的

替换应用程序☐ 原始的

添加新的或者已经存在的应用☐ 原始的

主要项☐ 原始的

描述☐ 原始的

3

更新

取消

Wizard	名称	触发器	键值	间隔	历史记录	趋势	类型	应用集	状态
<input type="checkbox"/>	*** Template App Zabbix Agent: Version of zabbix_agent(d) running	触发器 1	agent.version	1h	1w		Zabbix 客户端	Zabbix agent	已启用
<input type="checkbox"/>	*** Total swap space		system.swap.size[total]	1h	1w	365d	Zabbix客户端(主动式)	Memory	已启用
<input type="checkbox"/>	*** Total memory		vm.memory.size[total]	1h	1w	365d	Zabbix客户端(主动式)	Memory	已启用
<input type="checkbox"/>	*** System uptime	+ 触发器 1	system.uptime	10m	1w	365d	Zabbix客户端(主动式)	General, OS	已启用
<input type="checkbox"/>	*** System information	触发器 1	system.uname	1h	1w		Zabbix客户端(主动式)	General, OS	已启用
<input type="checkbox"/>	*** Processor load (5 min average per core)		system.cpu.load[percpu,avg15]	1m	1w	365d	Zabbix客户端(主动式)	CPU, Performance	已启用
<input type="checkbox"/>	*** Processor load (15 min average per core)		system.cpu.load[percpu,avg5]	1m	1w	365d	Zabbix客户端(主动式)	CPU, Performance	已启用
<input type="checkbox"/>	*** Processor load (1 min average per core)	触发器 1	system.cpu.load[percpu,avg1]	1m	1w	365d	Zabbix客户端(主动式)	CPU, Performance	已启用
<input type="checkbox"/>	*** Number of running processes	触发器 1	proc.num[,run]	1m	1w	365d	Zabbix客户端(主动式)	Processes	已启用
<input type="checkbox"/>	*** Number of processes	触发器 1	proc.num[]	1m	1w	365d	Zabbix客户端(主动式)	Processes	已启用
<input type="checkbox"/>	*** Number of logged in users		system.users.num	1m	1w	365d	Zabbix客户端(主动式)	OS, Security	已启用
<input type="checkbox"/>	*** Maximum number of processes	触发器 1	kernel.maxproc	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用
<input type="checkbox"/>	*** Maximum number of opened files	触发器 1	kernel.maxfiles	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用
<input type="checkbox"/>	*** Interrupts per second		system.cpu.intr	1m	1w	365d	Zabbix客户端(主动式)	CPU, Performance	已启用
<input type="checkbox"/>	*** Template App Zabbix Agent: Host name of zabbix_agentd running	触发器 1	agent.hostname	1h	1w		Zabbix 客户端	Zabbix agent	已启用
<input type="checkbox"/>	*** Host name	触发器 1	system.hostname	1h	1w		Zabbix客户端(主动式)	General, OS	已启用
<input type="checkbox"/>	*** Host local time		system.localtime	1m	1w	365d	Zabbix客户端(主动式)	General, OS	已启用

4) 选择“类型”，点击类型不是主动式的监控项最后的状态“已启用”，变为“停用的”

Wizard	名称	触发器	键值	间隔	历史记录	趋势	类型	应用集	状态
<input type="checkbox"/>	*** Template App Zabbix Agent: Version of zabbix_agent(d) running	触发器 1	agent.version	1h	1w		Zabbix 客户端	Zabbix agent	已启用
<input type="checkbox"/>	*** Template App Zabbix Agent: Host name of zabbix_agentd running	触发器 1	agent.hostname	1h	1w		Zabbix 客户端	Zabbix agent	已启用
<input type="checkbox"/>	*** Template App Zabbix Agent: Agent ping	触发器 1	agent.ping	1m	1w	365d	Zabbix 客户端	Zabbix agent	已启用
<input type="checkbox"/>	*** Maximum number of processes	触发器 1	kernel.maxproc	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用
<input type="checkbox"/>	*** Maximum number of opened files	触发器 1	kernel.maxfiles	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用

Wizard	名称	触发器	键值	间隔	历史记录	趋势	类型	应用集	状态
<input type="checkbox"/>	*** Template App Zabbix Agent: Version of zabbix_agent(d) running	触发器 1	agent.version	1h	1w		Zabbix 客户端	Zabbix agent	停用的
<input type="checkbox"/>	*** Template App Zabbix Agent: Host name of zabbix_agentd running	触发器 1	agent.hostname	1h	1w		Zabbix 客户端	Zabbix agent	停用的
<input type="checkbox"/>	*** Template App Zabbix Agent: Agent ping	触发器 1	agent.ping	1m	1w	365d	Zabbix 客户端	Zabbix agent	停用的
<input type="checkbox"/>	*** Maximum number of processes	触发器 1	kernel.maxproc	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用
<input type="checkbox"/>	*** Maximum number of opened files	触发器 1	kernel.maxfiles	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用

步骤四：添加监控主机

1) 选择“配置”，选择“主机”，点击自动发现的主机 web2

ZABBIX 监测中 资产记录 报表 **配置** | 管理

主机群组 模板 **主机** 维护 动作 关联项事件 自动发现 服务

主机

名称 DNS IP地址 端口

应用 **重设**

<input type="checkbox"/>	名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性
<input type="checkbox"/>	192.168.2.200	应用集	监控项	触发器	图形	自动发现	Web监测	192.168.2.200:10050		已启用	ZBX SNMP JMX
<input type="checkbox"/>	web1	应用集 11	监控项 50	触发器 22	图形 10	自动发现 2	Web监测	192.168.2.100:10050	count_line_passwd, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX
<input type="checkbox"/>	Zabbix server	应用集 11	监控项 85	触发器 52	图形 15	自动发现 2	Web监测	127.0.0.1:10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX

2) 修改主机名为“web2”，移除群组“Discovered hosts”，修改“agent 代理程序的接口”；

所有主机 / 192.168.2.200 已启用 ZBX | SNMP | JMX | IPMI 应用集 监控项 触发器 图形 自动发现规则 Web 场景

主机 模板 IPMI 宏 主机资产记录 加密

主机名称 **web2**

可见的名称

群组 在...群组之中

其它群组

新的群组

agent代理程序的接口

IP地址 **0.0.0.0** DNS名称 连接到 端口 **0** 默认 ☒ 移除

添加

SNMP接口 添加

3) 为 web2 选择新添加的监控模板

主机

所有主机 / 192.168.2.200 已启用 ZBX | SNMP | JMX | IPMI 应用集 监控项 触发器 图形 自动发现规则 Web 场景

主机 **模板** IPMI 宏 主机资产记录 加密

链接的模板 名称 动作

链接指示器 **选择**

添加

模板

名称

Template App Apache Tomcat JMX

Template App FTP Service

☒ Template OS Linux Active 4

☐ Template OS Linux SNMPv2

☐ Template OS Mac OS X

☐ Template OS OpenBSD

☐ Template OS Solaris

☐ Template OS Windows

☐ Template OS Windows SNMPv2

☐ Template Server Intel SR1530 IPMI

☐ Template Server Intel SR1630 IPMI

☐ Template VM VMware

☐ Template VM VMware Guest

☐ Template VM VMware Hypervisor

选择 5

主机

所有主机 / 192.168.2.200 已启用 ZBX SNMP JMX IPMI 应用集 监控项 触发器 图形 自动发现规则 Web 场景

主机 模板 IPMI 宏 主机资产记录 加密

链接的模板 名称 动作

链接指示器 Template OS Linux Active 选择

在此输入搜索

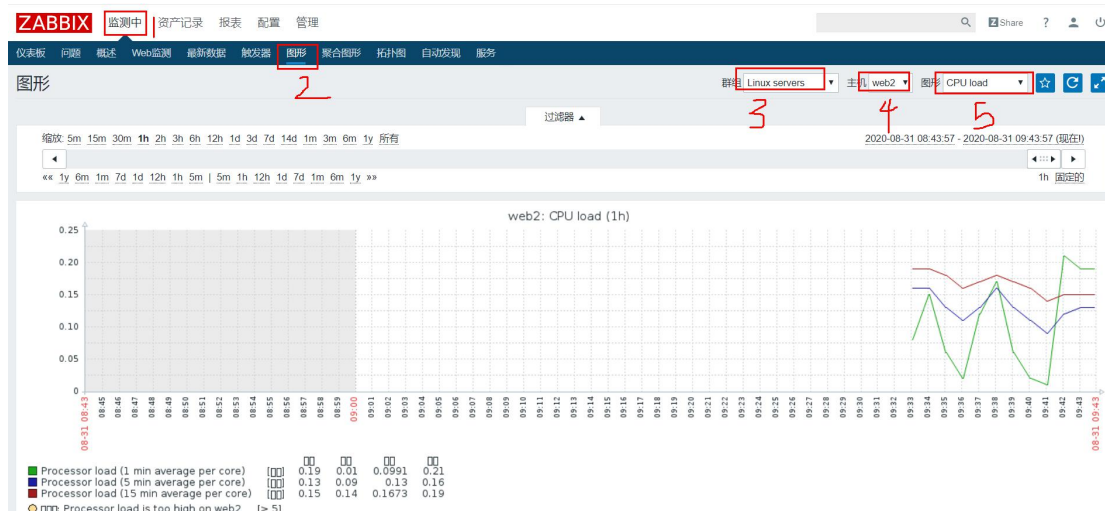
6 添加

7 更新 克隆 全克隆 删除 取消

名称	应用集	监控项	触发器	图形	自动发现	Web监测	接口	模板	状态	可用性	agent
web1	应用集 11	监控项 50	触发器 22	图形 10	自动发现 2	Web监测	192.168.2.100: 10050	count_line_passwd, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX IPMI	无
web2	应用集 10	监控项 32	触发器 15	图形 5	自动发现 2	Web监测	0.0.0.0	Template OS Linux Active (Template App Zabbix Agent)	已启用	ZBX SNMP JMX IPMI	无
Zabbix server	应用集 11	监控项 85	触发器 52	图形 15	自动发现 2	Web监测	127.0.0.1: 10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	已启用	ZBX SNMP JMX IPMI	无

步骤五：验证效果

选择“监测中”，选择“图形”，群组选择“Linux servers”，主机选择“web2”，图形选择“CPU load”



2) 如果显示不明显,可以编写死循环脚本,然后查看 CPU 的负载变化

[root@web2 ~]# while :

> do

> echo a

> done

一、监控 nginx

P1 环境要求

环境要求: 一台 Nginx 服务器, 部署 Nginx 时要加载 status 模块

[root@web1 ~]# cd lnmp_soft/nginx-1.16.1/

[root@web1 nginx-1.16.1]# yum -y install gcc pcre-devel openssl-devel

[root@web1 nginx-1.16.1]# ./configure --with-http_stub_status_module

[root@web1 nginx-1.16.1]# make && make install

[root@web1 nginx-1.16.1]# vim /usr/local/nginx/conf/nginx.conf

server {

...

```
location /status {
    stub_status on;
}
```

启动 nginx 服务

[root@web1 nginx-1.16.1]# /usr/local/nginx/sbin/nginx

查看 nginx 的服务器状态

[root@web1 nginx-1.16.1]# curl http://192.168.2.100/status

Active connections: 1 #实时并发量

server accepts handled requests

1 1 1

Reading: 0 Writing: 1 Waiting: 0

创建自定义监控项脚本

1) 被监控端修改配置文件, web1 已经开启了自定义监控项的功能, 不再重复

```
[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/nginx.status
```

UserParameter=nginx.status[*],/usr/local/bin/nginx_status.sh \$1 #自定义监控项的 key 可以传递参数,例如: nginx.status[5] 可以将数值 5 传入到脚本中

重启 zabbix 客户端服务器

```
[root@web1 ~]# killall zabbix_agentd
```

```
[root@web1 ~]# zabbix_agentd
```

编写脚本, 监控 nginx 服务器的并发量, 等待请求的数量, 服务器实际连接的数量

#curl -s 不显示下载信息

```
[root@web1 ~]# vim /usr/local/bin/nginx_status.sh
```

```
#!/bin/bash
```

```
case $1 in
```

```
active)
```

```
curl -s http://127.0.0.1/status | awk '/Active/{print $NF}';; #监控 web 服务器的并发量
```

```
waiting)
```

```
curl -s http://127.0.0.1/status | awk '/Waiting/{print $NF}';; #监控等待处理的请求数量
```

```
accepts)
```

```
curl -s http://127.0.0.1/status | awk 'NR==3{print $2}';; #监控服务器的连接数量 handled
```

```
esac
```

```
[root@web1 ~]# chmod +x /usr/local/bin/nginx_status.sh
```

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'nginx.status[accepts]'
```

```
5
```

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'nginx.status[waiting]'
```

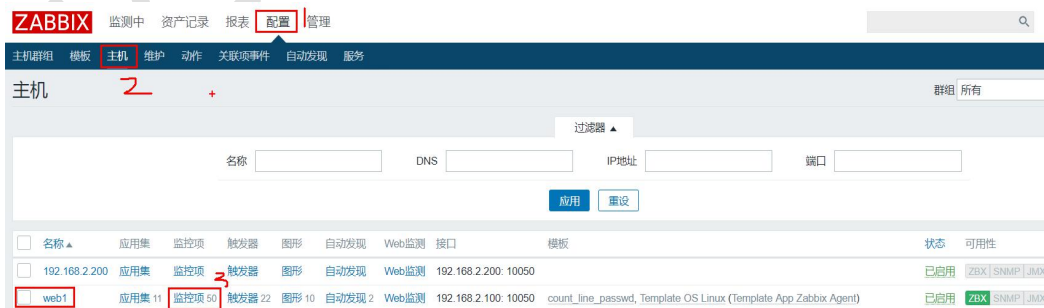
```
0
```

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'nginx.status[active]'
```

```
1
```

步骤二: 添加监控项

选择“配置”, 选择“主机”, 点击 web1 后的“监控项”



创建监控项

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

监控项

所有主机 / web1 已启用 ZBX SNMP JMX IPMI 应用集 11 监控项 50 触发器 22 图形 10 自动发现规则 2 Web 场景

创建监控项

过滤器

主机群组 在此输入搜索 选择 类型 所有 信息类型 所有 状态 所有

主机 web1 x 选择 更新间隔 历史记录 状态 所有

应用集 选择 趋势 触发器 所有

名称 模板 所有

键值

应用 重置

3) 设置名称为“nginx_status_accepts”，设置类型“Zabbix 客户端”，设置键值“nginx_status[accepts]”，主机接口为“192.168.2.100:10050”，类型信息为“数字”，选择“添加”

监控项

所有主机 / web1 已启用 ZBX SNMP JMX IPMI 应用集 11 监控项 50 触发器 22 图形 10 自动发现规则 2 Web 场景

监控项 进程

名称 nginx_status_accepts 1

类型 Zabbix 客户端 2

键值 nginx_status[accepts] 3 选择

主机接口 192.168.2.100:10050 4

信息类型 数字(无正负) 5

单位

更新间隔 30s

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

历史数据保留时长 90d

趋势存储时间 365d

查看值 不变 展示值映射

新的应用集

历史数据保留时长

趋势存储时间

查看值 [展示值映射](#)

新的应用集

应用集

- count_line_passwd
- CPU
- Filesystems
- General
- Memory
- Network interfaces
- OS
- Performance
- Processes

填入主机资产记录栏位

描述

已启用 ☒

2

4) 选择”监测中“，选择”最新数据“，点击”选择“，查看 web1 的数据

ZABBIX **监测中** | 资产记录 报表 配置 管理

仪表盘 问题 概述 Web监测 **最新数据** 触发器 图形 聚合图形 拓扑图 自动发现 服务

最新数据 **2** +

过滤器 ▲

主机群组 选择 名称

主机 **选择 3** 查看无资料项目 ☒

应用集 选择 查看细节 ☐

主机 群组

☐ 名称

☐ 192.168.2.200

☐ **web1 2**

☐ web2

ZABBIX 监测中 资产记录 报表 配置 管理

仪表盘 问题 概述 Web监测 **最新数据** 触发器 图形 聚合图形 拓扑图 自动发现 服务

最新数据

过滤器 ▲

主机群组 + 选择 名称

主机 **选择** 查看无资料项目 ☒

应用集 选择 查看细节 ☐

1

名称	最近检查记录	最新数据	更改
count_line_passwd (1 监控项)			
CPU (13 监控项)			
Filesystems (15 监控项)			
General (5 监控项)			
Memory (5 监控项)			
Network interfaces (2 监控项)			
OS (8 监控项)			
Performance (13 监控项)			
Processes (2 监控项)			
Security (2 监控项)			
Zabbix agent (3 监控项)			
- other - (1 监控项)			
nginx_status_accepts	2020-08-31 13:01:00	23	+1

5) zabbixserver 使用 ab 命令模拟大并发访问，图形查看

```
[root@zabbixserver ~]# yum -y install httpd-tools
```

```
[root@zabbixserver ~]# ab -c 100 -n 10000 http://192.168.2.100/
```

OS (8 监控项)			
Performance (13 监控项)			
Processes (2 监控项)			
Security (2 监控项)			
Zabbix agent (3 监控项)			
- other - (1 监控项)			
nginx_status_accepts	2020-08-31 13:10:00	10056	+1

web1: nginx_status_accepts



三、监控网络连接状态

P1 环境要求

查看网络监控状态

模拟多人并发连接（如果没有命令则需要安装个 httpd-tools）

```
[root@web1 ~]# yum -y install httpd-tools
```

```
[root@web1 ~]# ab -c 100 -n 1000 http://192.168.2.100/
```

```
[root@web1 nginx-1.16.1]# ss -antpu
```

步骤二：自定义监控项 key

1) 注意要允许自定义 key 并设置 Include，web1 已经开启自定义监控项的功能，不再重复

```
[root@web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/net.status
```

```
UserParameter=net.status[*],/usr/local/bin/net_status.sh $1
```

重启 zabbix 客户端服务

```
[root@web1 ~]# killall zabbix_agentd
```

```
[root@web1 ~]# zabbix_agentd
```

编写脚本监控网络连接状态（仅供参考，未检测完整状态）

```
[root@web1 ~]# vim /usr/local/bin/net_status.sh
```

```
#!/bin/bash
```

```
case $1 in
```

```
estab)
```

#统计 ESTAB 连接成功的状态数据；赋初值，当没有此状态时，显示为 0

```
ss -antp | awk 'BEGIN{x=0} /^ESTAB/{x++} END{print x};;
```

```
close_wait)
```

#统计服务端处于 CLOSE-WAIT 等待关闭的状态数量；赋初值，当没有此状态时，显示为 0

```
ss -antp | awk 'BEGIN{x=0} /^CLOSE-WAIT/{x++} END{print x};;
```

```
time_wait)
```

#统计客户端处于 TIME-WAIT 等待状态数量；赋初值，当没有此状态时，显示为 0

```
ss -antp | awk 'BEGIN{x=0} /^TIME-WAIT/{x++} END{print x};;
```

```
esac
```

给脚本赋予执行权限

```
[root@web1 ~]# chmod +x /usr/local/bin/net_status.sh
```

使用 zabbix_get 测试自定义监控项是否可以获取数据

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'net.status[estab]'
```

```
4
```

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'net.status[close_wait]'
```

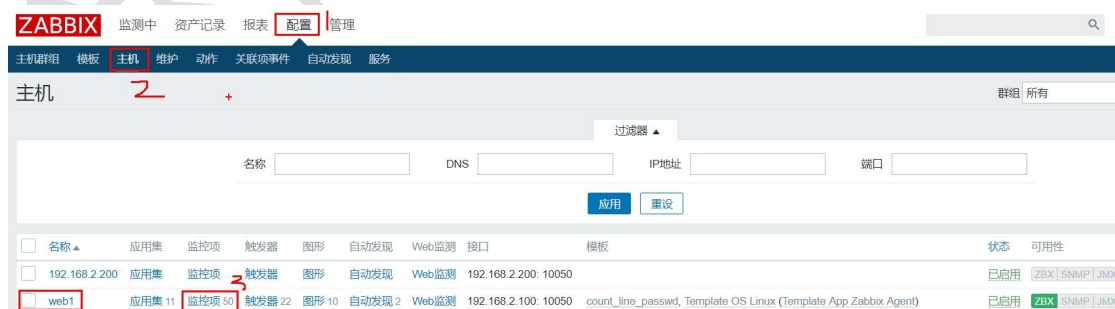
```
0
```

```
[root@web1 ~]# zabbix_get -s 127.0.0.1 -k 'net.status[time_wait]'
```

```
43
```

步骤三：监控 netstatus

1) 选择“配置”，选择“主机”，点击 web1 后的“监控项”



2) 选择“创建监控项”

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

监控项

所有主机 / web1 已启用 ZBX SNMP JMX IPMI 应用集 11 监控项 50 触发器 22 图形 10 自动发现规则 2 Web 场景

过滤器

主机群组 在此输入搜索 选择 类型 所有 信息类型 所有 状态 所有

主机 web1 x 选择 更新间隔 历史记录 状态 所有

应用集 选择 趋势 触发器 所有

名称 模板 所有

键值

应用 重设

3) 设置名称为“net_status_time_wait”，设置类型“Zabbix 客户端”，设置键值“net.status[time_wait]”，主机接口为“192.168.2.100:10050”，类型信息为“数字”，选择“添加”

监控项

所有主机 / web1 已启用 ZBX SNMP JMX IPMI 应用集 11 监控项 52 触发器 22 图形 10 自动发现规则 2

监控项 进程

名称 net_status_time_wait 1

类型 Zabbix 客户端 2

键值 net.status[time_wait] 3 选择

主机接口 192.168.2.100 : 10050 4

信息类型 数字 (无正负) 5

单位

更新间隔 30s 6

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

历史数据保留时长 90d 7

趋势存储时间 365d 8

查看值 不变 展示值映射

历史数据保留时长

趋势存储时间

查看值 [展示值映射](#)

新的应用集

应用集

count_line_passwd
CPU
Filesystems
General
Memory
Network interfaces
OS
Performance
Processes

填入主机资产纪录栏位

描述

已启用 ☒

2

4) 选择”监测中“，选择”最新数据“，点击”选择“，查看 web1 的数据

ZABBIX **监测中** | 资产记录 | 报表 | 配置 | 管理

仪表盘 | 问题 | 概述 | Web监测 | **最新数据** | 触发器 | 图形 | 聚合图形 | 拓扑图 | 自动发现 | 服务

最新数据 **1** +

过滤器 ▲

主机群组 名称

主机 **选择 3** ☒ 查看无资料项目

应用集 ☐ 查看细节

主机

☐ 名称

☐ 192.168.2.200 **2**

☐ web1 **2**

☐ web2

3

ZABBIX **监测中** | 资产记录 | 报表 | 配置 | 管理

仪表盘 | 问题 | 概述 | Web监测 | **最新数据** | 触发器 | 图形 | 聚合图形 | 拓扑图 | 自动发现 | 服务

最新数据

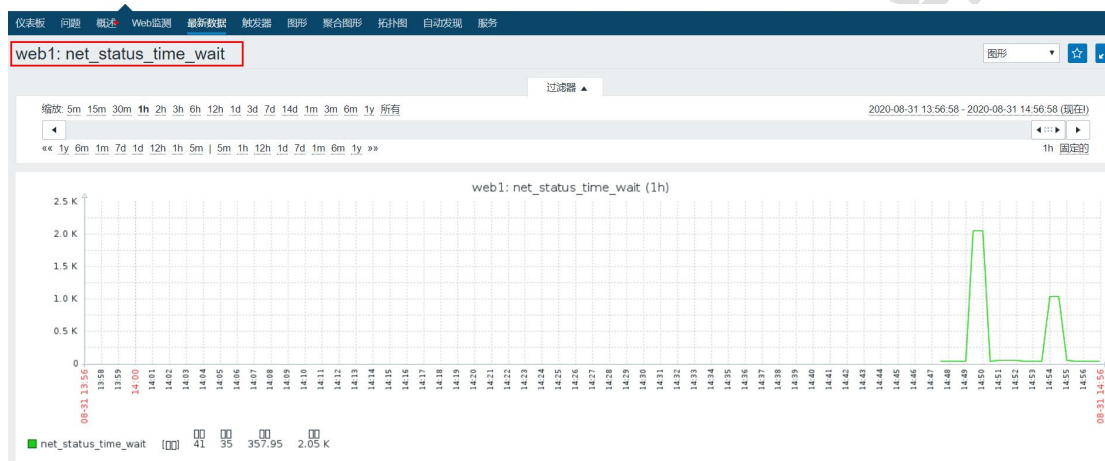
过滤器 ▲

主机群组

主机 ☒ 查看无资料项目

应用集 ☐ 查看细节

名称	最近检查记录	最新数据	更改
count_line_passwd (1 监控项)			
CPU (13 监控项)			
Filesystems (15 监控项)			
General (5 监控项)			
Memory (5 监控项)			
Network interfaces (2 监控项)			
OS (8 监控项)			
Performance (13 监控项)			
Processes (2 监控项)			
Security (2 监控项)			
Zabbix agent (3 监控项)			
- other - (2 监控项)			
<input checked="" type="checkbox"/> net_status_time_wait	2020-08-31 14:56:04	39	-4
<input type="checkbox"/> nginx_status_accepts	2020-08-31 14:56:00	22265	+1



5) zabbixserver 使用 ab 命令模拟大并发访问，图形查看

####ab 命令可以默认并发访问，-c 指 100 人同时访问，-n 指访问的数量为 1000

#ab 在做并发访问测试时，网页路径最后必须加 /

[root@zabbixserver ~]# ab -c 100 -n 1000 http://192.168.2.100/

