# [Zabbix与ELK整合实现对日志数据的实时监控](https://www.cnblogs.com/flytor/p/11440799.html)

# 一、 ELK与zabbix有什么关系？

ELK大家应该比较熟悉了，zabbix应该也不陌生，那么将ELK和zabbix放到一起的话，可能大家就有疑问了？这两个放到一起是什么目的呢，听我细细道来。

ELK是一套日志收集套件，它其实有由Elasticsearch、Logstash和Kibana三个软件组成，通过ELK可以收集系统日志、网站日志、应用系统日志等各种日志数据，并且还可以对日志进行过滤、清洗，然后进行集中存放并可用于实时检索、分析。这是ELK的基础功能。

但是有些时候，我们希望在收集日志的时候，能够将日志中的异常信息（警告、错误、失败等信息）及时的提取出来，因为日志中的异常信息意味着操作系统、应用程序可能存在故障，如果能将日志中的故障信息及时的告知运维人员，那么运维就可以第一时间去进行故障排查和处理，进而也就可以避免很多故障的发生。

那么如何才能做到将ELK收集的日志数据中出现的异常信息及时的告知运维人员呢，这就需要用到zabbix了，ELK（更确切的说应该是logstash）可以实时的读取日志的内容，并且还可以过滤日志信息，通过ELK的读取和过滤功能，就可以将日志中的一些异常关键字（error、failed、OutOff、Warning）过滤出来，然后通过logstash的zabbix插件将这个错误日志信息发送给zabbix，那么zabbix在接收到这个数据后，结合自身的机制，然后发起告警动作，这样就实现了日志异常zabbix实时告警的功能了。

# 二、Logstash与zabbix插件的使用

Logstash支持多种输出介质，比如syslog、HTTP、TCP、elasticsearch、kafka等，而有时候我们想将收集到的日志中一些错误信息输出，并告警时，就用到了logstash-output-zabbix这个插件，此插件可以将Logstash与zabbix进行整合，也就是将Logstash收集到的数据进行过滤，将有错误标识的日志输出到zabbix中，最后通过zabbix的告警机制进行触发、告警。

logstash-output-zabbix是一个社区维护的插件，它默认没有在Logstash中安装,但是安装起来也很容易，直接在logstash中运行如下命令即可：

/usr/local/logstash/bin/logstash-plugin install logstash-output-zabbix

其中，/usr/local/logstash是Logstash的安装目录。

此外，logstash-plugin命令还有多种用法，我们来看一下：

## 2.1、列出目前已经安装的插件

将列出所有已安装的插件

/usr/local/logstash/bin/logstash-plugin list

#将列出已安装的插件及版本信息

/usr/local/logstash/bin/logstash-plugin list --verbose

#将列出包含namefragment的所有已安装插件

/usr/local/logstash/bin/logstash-plugin list "http"

#将列出特定组的所有已安装插件（ input，filter，codec，output）

/usr/local/logstash/bin/logstash-plugin list --group input

## 2.2、安装插件

要安装某个插件，例如安装kafka插件，可执行如下命令：

/usr/local/logstash/bin/logstash-plugin install logstash-output-kafka  
要使用此命令安装插件，需要你的电脑可以访问互联网。此插件安装方法，会检索托管在公共存储库（RubyGems.org）上的插件，然后下载到本地机器并在Logstash安装之上进行自动安装。

## 2.3、更新插件

每个插件有自己的发布周期和版本更新，这些更新通常是独立于Logstash的发布周期的。因此，有时候需要单独更新插件，可以使用update子命令获得最新版本的插件。

将更新所有已安装的插件

/usr/local/logstash/bin/logstash-plugin update

将仅更新指定的插件

/usr/local/logstash/bin/logstash-plugin update logstash-output-kafka

## 2.4、删除插件

如果需要从Logstash插件中删除插件，可执行如下命令：

/usr/local/logstash/bin/logstash-plugin remove logstash-output-kafka

这样就删除了logstash-output-kafka插件。

# 三、logstash-output-zabbix插件的使用

logstash-output-zabbix安装好之后，就可以在logstash配置文件中使用了，  
下面是一个logstash-output-zabbix使用的例子：

zabbix {

zabbix\_host => "[@metadata][zabbix\_host]"

zabbix\_key => "[@metadata][zabbix\_key]"

zabbix\_server\_host => "x.x.x.x"

zabbix\_server\_port => "xxxx"

zabbix\_value => "xxxx"

}

其中：

zabbix\_host：表示Zabbix主机名字段名称， 可以是单独的一个字段， 也可以是 @metadata 字段的子字段， 是必需的设置，没有默认值。

zabbix\_key：表示Zabbix项目键的值，也就是zabbix中的item，此字段可以是单独的一个字段， 也可以是 @metadata 字段的子字段，没有默认值。

zabbix\_server\_host：表示Zabbix服务器的IP或可解析主机名，默认值是 "localhost"，需要设置为zabbix server服务器所在的地址。

zabbix\_server\_port：表示Zabbix服务器开启的监听端口，默认值是10051。

zabbix\_value：表示要发送给zabbix item监控项的值对应的字段名称，默认值是 "message"，也就是将"message"字段的内容发送给上面zabbix\_key定义的zabbix item监控项，当然也可以指定一个具体的字段内容发送给zabbix item监控项。

# 四、将logstash与zabbix进行整合

这里我们以logstash收集日志，然后对日志进行读取，最后选择关键字进行过滤并调用zabbix告警的流程，来看看如何配置logstash实现zabbix告警。

先说明一下我们的应用需求：通过对系统日志文件的监控，然后去过滤日志信息中的一些关键字，例如ERR、error、ERROR、Failed、WARNING等，将日志中这些信息过滤出来，然后发送到zabbix上，最后借助zabbix的报警功能实现对系统日志中有上述关键字的告警。

对于过滤关键字，进行告警，不同的业务系统，可能关键字不尽相同，例如对http系统，可能需要过滤500、403、503等这些错误码，对于java相关的系统，可能需要过滤OutOfMemoryError、PermGen、Java heap等关键字。在某些业务系统的日志输出中，可能还有一些自定义的错误信息，那么这些也需要作为过滤关键字来使用。

## 4.1、配置logstash事件配置文件

接下来就是创建一个logstash事件配置文件，这里将配置文件分成三个部分来介绍，首先是input部分，内容如下：

input {

file {

path => ["/var/log/secure"]

type => "system"

start\_position => "beginning"

}

}

input部分是从/var/log/secure文件中读取数据，start\_position 表示从secure文件开头读取内容。

接着是filter部分，内容如下：

filter {

grok {

match => { "message" => "%{SYSLOGTIMESTAMP:message\_timestamp} %{SYSLOGHOST:hostname} %{DATA:message\_program}(?:\[%{POSINT:messag

e\_pid}\])?: %{GREEDYDATA:message\_content}" } #这里通过grok对message字段的数据进行字段划分，这里将message字段划分了5个子字段。其中，message\_content字段会在output中用到。

}

mutate {

add\_field => [ "[zabbix\_key]", "oslogs" ] #新增的字段，字段名是zabbix\_key，值为oslogs。

add\_field => [ "[zabbix\_host]", "%{host}" ] #新增的字段，字段名是zabbix\_host，值可以在这里直接定义，也可以引用字段变量来获取。这里的%{host}获取的就是日志数据的主机名，这个主机名与zabbix web中“主机名称”需要保持一致。

}

mutate { #这里是删除不需要的字段

remove\_field => "@version"

remove\_field => "message"

}

date { #这里是对日志输出中的日期字段进行转换，其中message\_timestamp字段是默认输出的时间日期字段，将这个字段的值传给 @timestamp字段。

match => [ "message\_timestamp","MMM d HH:mm:ss", "MMM dd HH:mm:ss", "ISO8601"]

}

}

filter部分是个重点，在这个部分中，重点关注的是message\_timestamp字段、message\_content字段。

最后是output部分，内容如下：

output {

if [message\_content] =~ /(ERR|error|ERROR|Failed)/ { #定义在message\_content字段中，需要过滤的关键字信息，也就是在message\_content字段中出现给出的这些关键字，那么就将这些信息发送给zabbix。

zabbix {

zabbix\_host => "[zabbix\_host]" #这个zabbix\_host将获取上面filter部分定义的字段变量%{host}的值

zabbix\_key => "[zabbix\_key]" #这个zabbix\_key将获取上面filter部分中给出的值

zabbix\_server\_host => "172.16.213.140" #这是指定zabbix server的IP地址

zabbix\_server\_port => "10051" #这是指定zabbix server的监听端口

zabbix\_value => "message\_content" #这个很重要，指定要传给zabbix监控项item（oslogs）的值， zabbix\_value默认的值是"message"字段，因为上面我们已经删除了"message"字段，因此，这里需要重新指定，根据上面filter部分对"message"字段的内容划分，这里指定为"message\_content"字段，其实，"message\_content"字段输出的就是服务器上具体的日志内容。

}

}

#stdout { codec => rubydebug } #这里是开启调试模式，当第一次配置的时候，建议开启，这样过滤后的日志信息直接输出的屏幕，方便进行调试，调试成功后，即可关闭。

}

将上面三部分内容合并到一个文件file\_to\_zabbix.conf中，然后启动logstash服务：

[root@logstashserver ~]#cd /usr/local/logstash

[root@logstashserver logstash]#nohup bin/logstash -f config/file\_to\_zabbix.conf --path.data /data/osdata &

这里的–path.data是指定此logstash进程的数据存储目录，用于在一个服务器上启动多个logstash进程的环境中。

## 4.2、zabbix平台配置日志告警

登录zabbix web平台，选择配置—>模板—>创建模板，名称定为logstash-output-zabbix，如下图所示：



接着，在此模块下创建一个应用集，点击应用集—–>创建应用集，如下图所示：



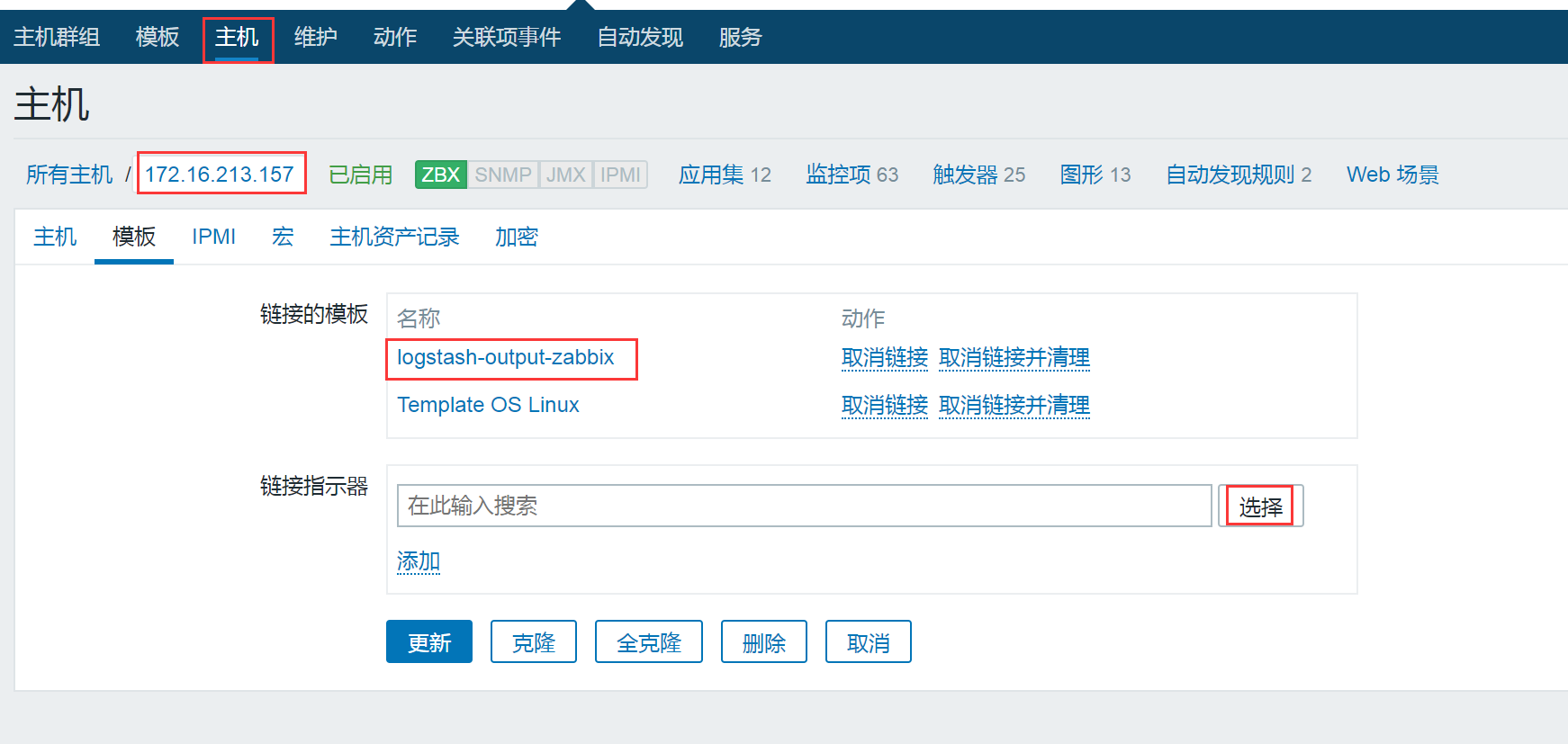
然后，在此模块下创建一个监控项，点击监控项—–>创建监控项，如下图所示：



到此为止，zabbix监控logstash的日志数据配置完成。

这里我们以客户端172.16.213.157主机为例，也就是监控172.16.213.157主机上的系统日志数据，当发现日志异常就进行告警。

在上面创建了一个模板和监控项，接着还需要将此模板链接到172.16.213.157主机上，选择“配置”—“主机”，然后选中172.16.213.157主机，选择“模板”标签，将上面创建的模板加入到172.16.213.157主机上，如下图所示：



这样，上面创建的监控项在172.16.213.157主机上就自动生效了。

下面我们模拟一个故障，在任意主机通过ssh登录172.16.213.157主机，然后输入一个错误密码，让系统的/var/log/secure文件产生错误日志，然后看看logstash是否能够过滤到，是否能够发送到zabbix中。

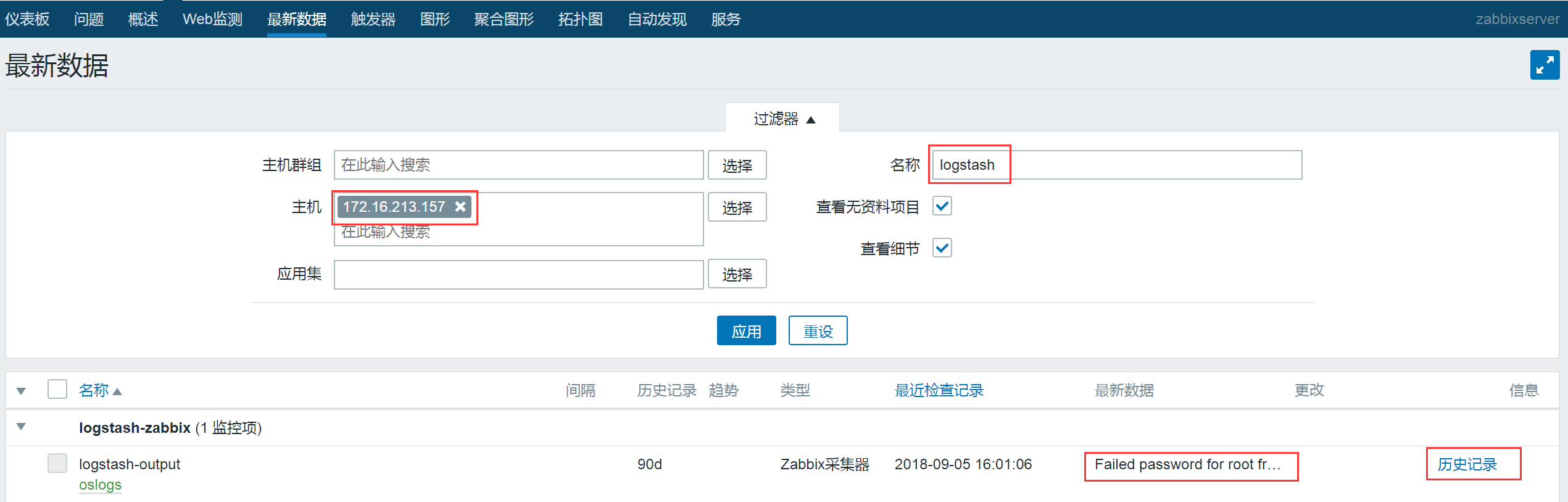
首先让系统文件/var/log/secure产生类似如下内容：

Sep 5 16:01:04 localhost sshd[27159]: pam\_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.16.213.127 user=root

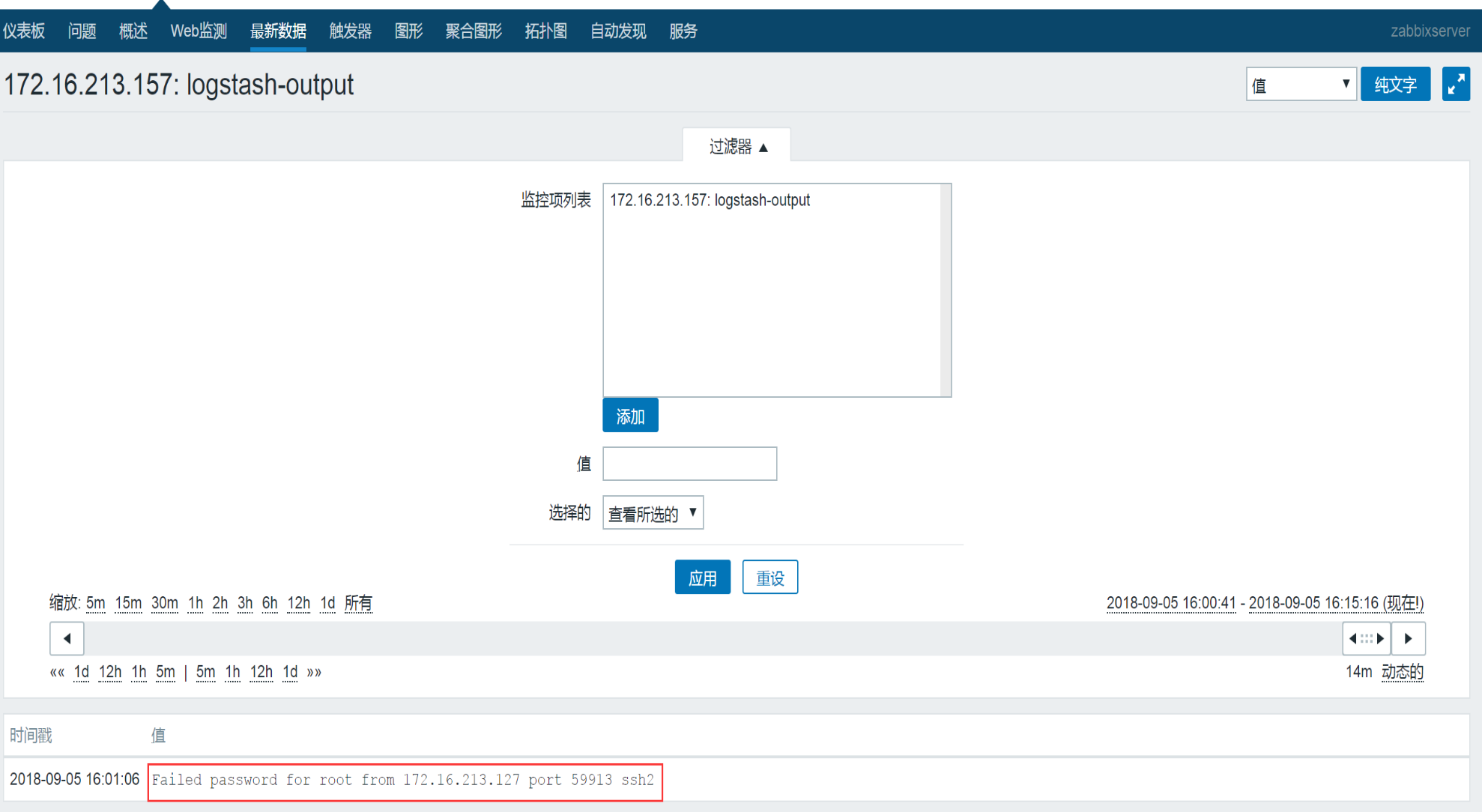
Sep 5 16:01:06 localhost sshd[27159]: Failed password for root from 172.16.213.127 port 59913 ssh2

这里面有我们要过滤的关键字Failed，因此logstash会将此内容过滤出来，发送到zabbix上。

接着，登录zabbix web平台，点击监测中—–>最新数据，如果zabbix能够接收到日志，就可以看到下图的最新数据：



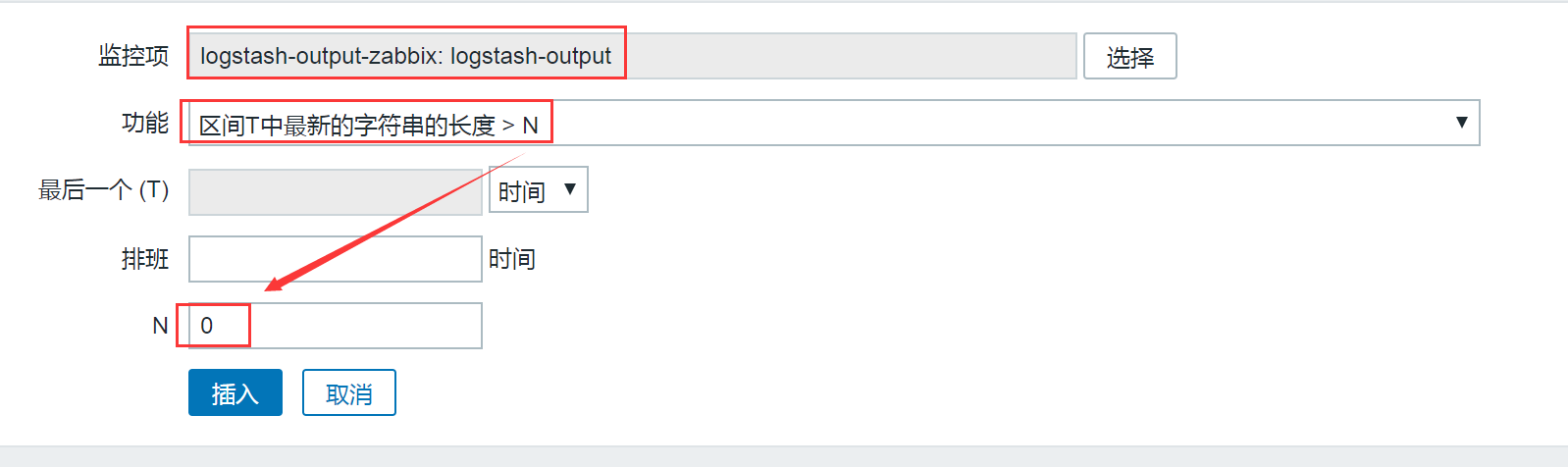
点击历史记录，可以查看详细内容，如下图所示：



可以看到，红框中的内容就是在logstash中定义的message\_content字段的内容。

到这里为止，zabbix已经可以收到logstash的发送过来的数据了，但是要实现报警，还需要在zabbix中创建一个触发器，进入配置—–>模板，选择logstash-output-zabbix这个模板，然后点击上面的触发器，继续点击右上角的创建触发器，如下图所示：





这里注意触发器创建中，表达式的写法，这里触发器的触发条件是：如果接收到logstash发送过来的数据，就进行告警，也就是说接收到的数据，如果长度大于0就告警。

触发器配置完成后，如果配置正常，就会进行告警了，告警内容如下图所示：



[**Zabbix 3.0 从入门到精通(zabbix使用详解)**](https://www.cnblogs.com/clsn/p/7885990.html)

**第1章 zabbix监控**

**1.1 为什么要监控**

   　　在需要的时刻，提前提醒我们服务器出问题了

  　　 当出问题之后，可以找到问题的根源

 　　  网站/服务器 的可用性

**1.1.1 网站可用性**

　　在软件系统的高可靠性（也称为可用性，英文描述为HA，High Available）里有个衡量其可靠性的标准——X个9，这个X是代表数字3~5。X个9表示在软件系统1年时间的使用过程中，系统可以正常使用时间与总时间（1年）之比，我们通过下面的计算来感受下X个9在不同级别的可靠性差异。

1个9：(1-90%)\*365=36.5天，表示该软件系统在连续运行1年时间里最多可能的业务中断时间是36.5天

2个9：(1-99%)\*365=3.65天 ， 表示该软件系统在连续运行1年时间里最多可能的业务中断时间是3.65天

3个9：(1-99.9%)\*365\*24=8.76小时，表示该软件系统在连续运行1年时间里最多可能的业务中断时间是8.76小时。

4个9：(1-99.99%)\*365\*24=0.876小时=52.6分钟，表示该软件系统在连续运行1年时间里最多可能的业务中断时间是52.6分钟。

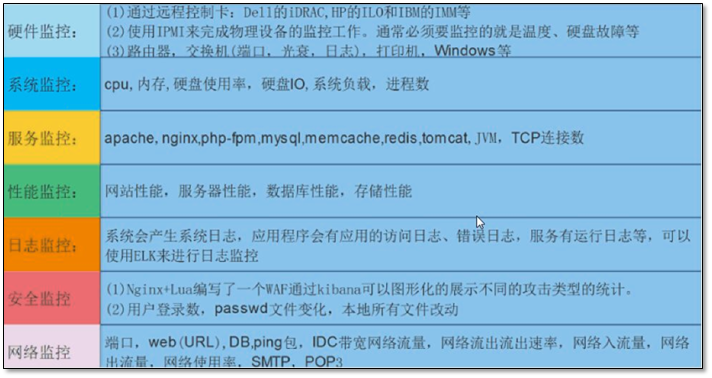
5个9：(1-99.999%)\*365\*24\*60=5.26分钟，表示该软件系统在连续运行1年时间里最多可能的业务中断时间是5.26分钟。

6个9：(1-99.9999%)\*365\*24\*60\*60=31秒， 示该软件系统在连续运行1年时间里最多可能的业务中断时间是31秒

**1.2 监控什么东西**

监控一切需要监控的东西，只要能够想到，能够用命令实现的都能用来监控

**1.2.1 监控范畴**



**1.3 怎么来监控**

**1.3.1 远程管理服务器**

如果想**远程管理服务器**就有远程管理卡，比如Dell idRAC，HP ILO，IBM IMM

**1.3.2 监控硬件**

查看硬件的温度/风扇转速，电脑有鲁大师，服务器就有ipmitool。

使用ipmitool实现对服务器的命令行远程管理

yum -y install OpenIPMI ipmitool #->IPMI在物理机可以成功，虚拟机不行

[root@KVM ~]# ipmitool sdr type Temperature

Temp | 01h | ns | 3.1 | Disabled

Temp | 02h | ns | 3.2 | Disabled

Temp | 05h | ns | 10.1 | Disabled

Temp | 06h | ns | 10.2 | Disabled

Ambient Temp | 0Eh | ok | 7.1 | 22 degrees C

Planar Temp | 0Fh | ns | 7.1 | Disabled

IOH THERMTRIP | 5Dh | ns | 7.1 | Disabled

CPU Temp Interf | 76h | ns | 7.1 | Disabled

Temp | 0Ah | ns | 8.1 | Disabled

Temp | 0Bh | ns | 8.1 | Disabled

Temp | 0Ch | ns | 8.1 | Disabled

**1.3.3 查看cpu相关**

　　lscpu、uptime、top、htop vmstat mpstat

   其中htop需要安装，安装依赖与epel源。

[znix@clsn ~]$lscpu

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Byte Order: Little Endian

CPU(s): 1

On-line CPU(s) list: 0

Thread(s) per core: 1

Core(s) per socket: 1

Socket(s): 1

NUMA node(s): 1

Vendor ID: GenuineIntel

CPU family: 6

Model: 85

Model name: Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz

Stepping: 4

CPU MHz: 2494.150

BogoMIPS: 4988.30

Hypervisor vendor: KVM

Virtualization type: full

L1d cache: 32K

L1i cache: 32K

L2 cache: 1024K

L3 cache: 33792K

NUMA node0 CPU(s): 0

**1.3.4 内存够不够可以用**

　　free

[znix@clsn ~]$free -h

total used free shared buffers cached

Mem: 996M 867M 128M 712K 145M 450M

-/+ buffers/cache: 271M 725M

Swap: 1.0G 0B 1.0G

**1.3.5 磁盘剩多少写的快不快可以用**

　　df、dd、iotop

[znix@clsn ~]$df -h

Filesystem Size Used Avail Use% Mounted on

/dev/vda1 40G 24G 15G 62% /

tmpfs 499M 20K 499M 1% /dev/shm

/dev/vdb1 20G 4.4G 15G 24% /data

**1.3.6 监控网络**

　　iftop nethogs

iftop 监控主机间流量 -i 指定监控网卡

nethogs 监控进程流量

**1.4 监控工具总览**

　　mrtg 流量监控出图

　　nagios 监控

　　cacti  流量监控出图

**zabbix 监控+出图**

**1.5 zabbix介绍**

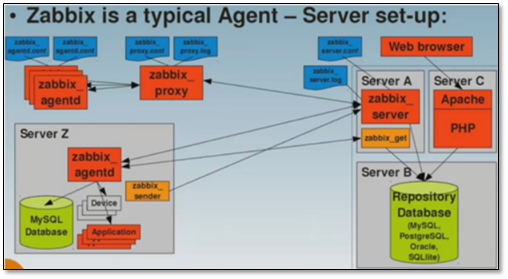
　　Zabbix 是由 Alexei Vladishev 开发的一种网络监视、管理系统，基于 Server-Client 架构。可用于监视各种网络服务、服务器和网络机器等状态。

　　使用各种 Database-end 如 MySQL, PostgreSQL, SQLite, Oracle 或 IBM DB2 储存资料。Server 端基于 C语言、Web 管理端 frontend 则是基于 PHP 所制作的。Zabbix 可以使用多种方式监视。可以只使用 Simple Check 不需要安装 Client 端，亦可基于 SMTP 或 HTTP ... 各种协定做死活监视。

　　在客户端如 UNIX, Windows 中安装 Zabbix Agent 之后，可监视 CPU Load、网络使用状况、硬盘容量等各种状态。而就算没有安装 Agent 在监视对象中，Zabbix 也可以经由 SNMP、TCP、ICMP、利用 IPMI、SSH、telnet 对目标进行监视。

另外，Zabbix 包含 XMPP 等各种 Item 警示功能。

**1.5.1 zabbix的组成**



zabbix官网: https://www.zabbix.com

zabbix 主要由2部分构成 zabbix server和 zabbix agent

zabbix proxy是用来管理其他的agent，作为代理

**1.5.2 zabbix监控范畴**

　　²  硬件监控 ：Zabbix IPMI Interface

　　²  系统监控 ：Zabbix Agent Interface

　　²  Java 监控：ZabbixJMX Interface

　　²  网络设备监抟：Zabbix SNMP Interface

　　²  应用服务监控：Zabbix Agent UserParameter

　　²  MySQL 数据库监控：percona-monitoring-pldlgins

　　²  URL监控：Zabbix Web监控

**第2章 安装zabbix**

**2.1 环境检查**

[root@m01 ~]# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

[root@m01 ~]# uname -r

3.10.0-693.el7.x86\_64

[root@m01 ~]# getenforce

Disabled

[root@m01 ~]# systemctl status firewalld.service

● firewalld.service - firewalld - dynamic firewall daemon

Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)

Active: inactive (dead)

Docs: man:firewalld(1)

**2.2 安装zabbix过程**

**2.2.1 安装方式选择**

　　编译安装 （服务较多，环境复杂）

　　yum安装（干净环境）

　　使用yum 需要镜像yum源 http://www.cnblogs.com/clsn/p/7866643.html

**2.2.2 服务端快速安装脚本**

#!/bin/bash

#clsn

#设置解析 注意：网络条件较好时，可以不用自建yum源

# echo '10.0.0.1 mirrors.aliyuncs.com mirrors.aliyun.com repo.zabbix.com' >> /etc/hosts

#安装zabbix源、aliyun YUM源

curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

curl -o /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

#安装zabbix

yum install -y zabbix-server-mysql zabbix-web-mysql

#安装启动 mariadb数据库

yum install -y mariadb-server

systemctl start mariadb.service

#创建数据库

mysql -e 'create database zabbix character set utf8 collate utf8\_bin;'

mysql -e 'grant all privileges on zabbix.\* to zabbix@localhost identified by "zabbix";'

#导入数据

zcat /usr/share/doc/zabbix-server-mysql-3.0.13/create.sql.gz|mysql -uzabbix -pzabbix zabbix

#配置zabbixserver连接mysql

sed -i.ori '115a DBPassword=zabbix' /etc/zabbix/zabbix\_server.conf

#添加时区

sed -i.ori '18a php\_value date.timezone Asia/Shanghai' /etc/httpd/conf.d/zabbix.conf

#解决中文乱码

yum -y install wqy-microhei-fonts

\cp /usr/share/fonts/wqy-microhei/wqy-microhei.ttc /usr/share/fonts/dejavu/DejaVuSans.ttf

#启动服务

systemctl start zabbix-server

systemctl start httpd

#写入开机自启动

chmod +x /etc/rc.d/rc.local

cat >>/etc/rc.d/rc.local<<EOF

systemctl start mariadb.service

systemctl start httpd

systemctl start zabbix-server

EOF

#输出信息

echo "浏览器访问 http://`hostname -I|awk '{print $1}'`/zabbix"

**2.2.3 客户端快速部署脚本**

#!/bin/bash

#clsn

#设置解析

echo '10.0.0.1 mirrors.aliyuncs.com mirrors.aliyun.com repo.zabbix.com' >> /etc/hosts

#安装zabbix源、aliyu nYUM源

curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

curl -o /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

#安装zabbix客户端

yum install zabbix-agent -y

sed -i.ori 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_agentd.conf

systemctl start zabbix-agent.service

#写入开机自启动

chmod +x /etc/rc.d/rc.local

cat >>/etc/rc.d/rc.local<<EOF

systemctl start zabbix-agent.service

EOF

**2.3 检测连通性**

**2.3.1 服务端安装zabbix-get检测工具**

yum install zabbix-get

**2.3.2 在服务端进行测试**

注意：只能在服务端进行测试

zabbix\_get -s 172.16.1.61 -p 10050 -k "system.cpu.load[all,avg1]"

zabbix\_get -s 172.16.1.21 -p 10050 -k "system.cpu.load[all,avg1]"

***测试结果***

[root@m01 ~]# zabbix\_get -s 172.16.1.61 -p 10050 -k "system.cpu.load[all,avg1]"

0.000000

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "system.cpu.load[all,avg1]"

0.000000

**第3章 web界面操作**

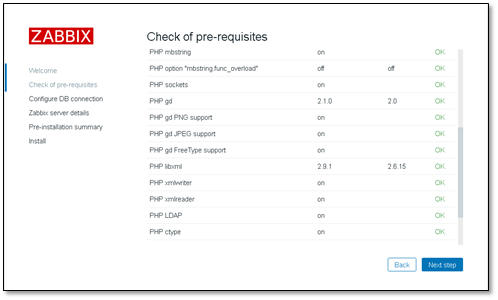
**3.1 zabbix的web安装**

**3.1.1 使用浏览器访问**

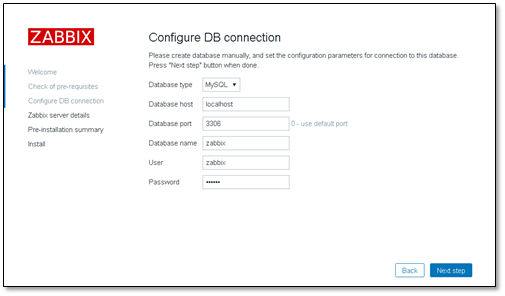
*http://10.0.0.61/zabbix/setup.php*



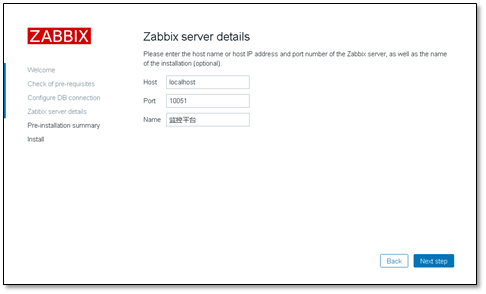
   在检测信息时，可查看具体的报错信息进行不同的解决



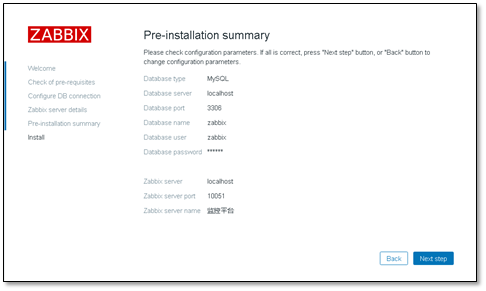
   选择mysql数据库，输入密码即可



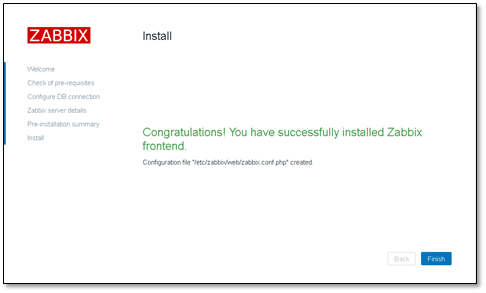
   host与port不需要修改，name自定义



确认信息,正确点击下一步



   安装完成、点击finsh



      进入登陆界面  账号**Admin**密码**zabbix   注意A大写**



**3.2 添加监控信息**

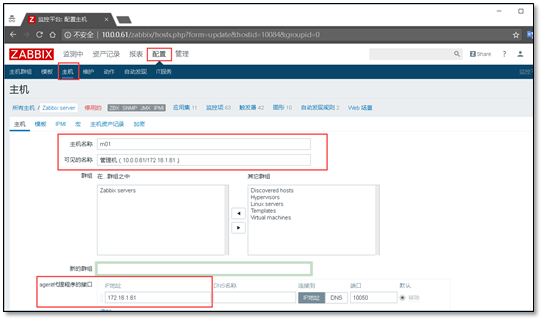
**3.2.1 修改监控管理机zabbix server**

配置 >> 主机



主机名称： 要与主机名相同，这是zabbix server程序用的

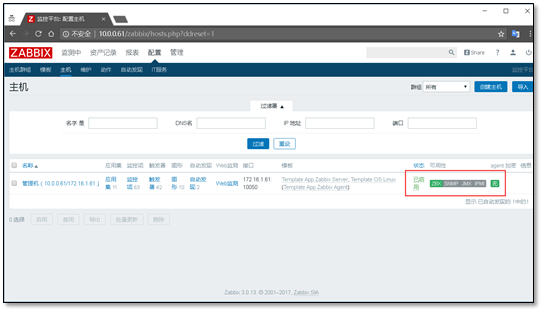
可见名称： 显示在zabbix网页上的，给我们看的



   修改后，要将下面的已启用要勾上

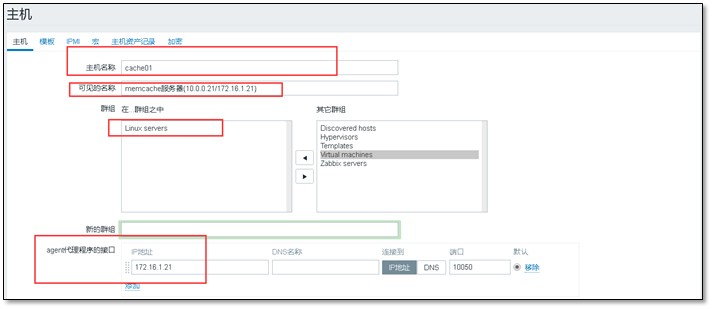


   添加完成就有了管理机的监控主机



**3.2.2 添加新的主机**

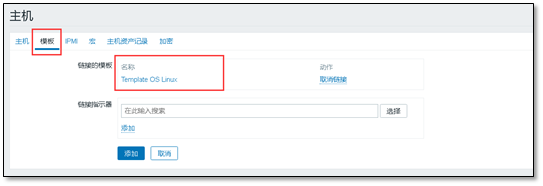
配置 >> 主机 >> 创建主机



注意勾选以启用



   然后添加模板，选择linux OS ，先点小添加，再点大添加。



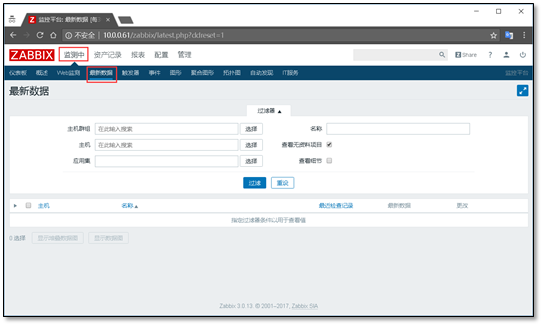
   添加完成，将会又两条监控主机信息



**3.2.3 查看监控内容**

检测中  >> 最新数据

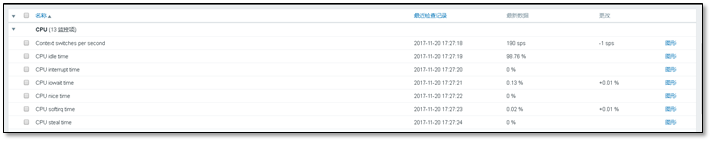
   在最新数据中需要筛选，



   输入ip或者名字都能够搜索出来



在下面就会列出所有的监控项



**3.2.4 查看图像**

检测中 >> 图形

   选择正确的主机。选择要查看的图形即可出图



**第4章 自定义监控与监控报警**

**4.1 自定义监控**

**4.1.1 说明**

zabbix自带模板Template OS Linux (Template App Zabbix Agent)提供CPU、内存、磁盘、网卡等常规监控，只要新加主机关联此模板，就可自动添加这些监控项。

**需求：**服务器登陆人数不能超过三人，超过三人报警

**4.1.2 预备知识**

自定义key能被server和agent认可

# 正确的key

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "system.uname"

Linux cache01 3.10.0-693.el7.x86\_64 #1 SMP Tue Aug 22 21:09:27 UTC 2017 x86\_64

# 没有登记的，自定义的key

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user"

ZBX\_NOTSUPPORTED: Unsupported item key.

# 写错的key

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "system.uname1"

ZBX\_NOTSUPPORTED: Unsupported item key.

**4.2 实现自定义监控**

**4.2.1 自定义语法**

UserParameter=<key>,<shell command>

UserParameter=login-user,who|wc -l

UserParameter=login-user,/bin/sh /server/scripts/login.sh

**4.2.2 agent注册**

[root@cache01 ~]# cd /etc/zabbix/zabbix\_agentd.d/

[root@cache01 zabbix\_agentd.d]# vim userparameter\_login.conf

UserParameter=login-user,who|wc -l

UserParameter=login-user2,who|wc -l

UserParameter=login-user3,who|wc -l

**注意：**key名字要唯一，多个key以行为分割

# 修改完成后重启服务

[root@cache01 zabbix\_agentd.d]# systemctl restart zabbix-agent.service

   在server端进行get测试

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user"

3

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user2"

3

[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user3"

3

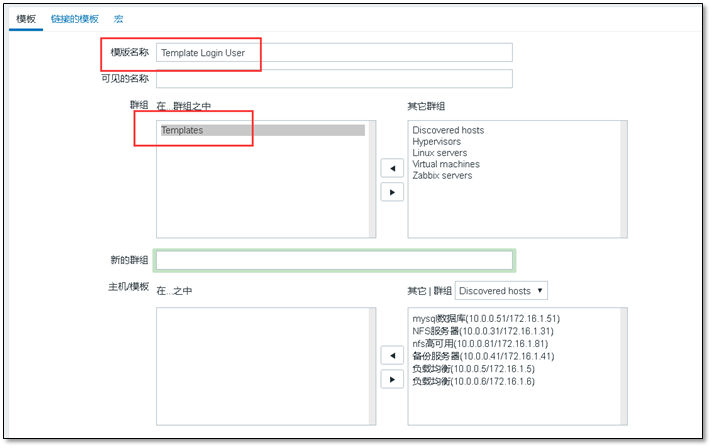
[root@m01 ~]# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user4"

ZBX\_NOTSUPPORTED: Unsupported item key.

**4.2.3 在server端注册(web操作)**

**①   创建模板**

配置 >> 模板 >> 创建模板



点击添加，即可创建出来模板



   查看创建出来的模板。↑

**②   创建应用集**

应用集类似(目录/文件夹)，其作用是给监控项分类。

点击 应用集 >> 创建应用集



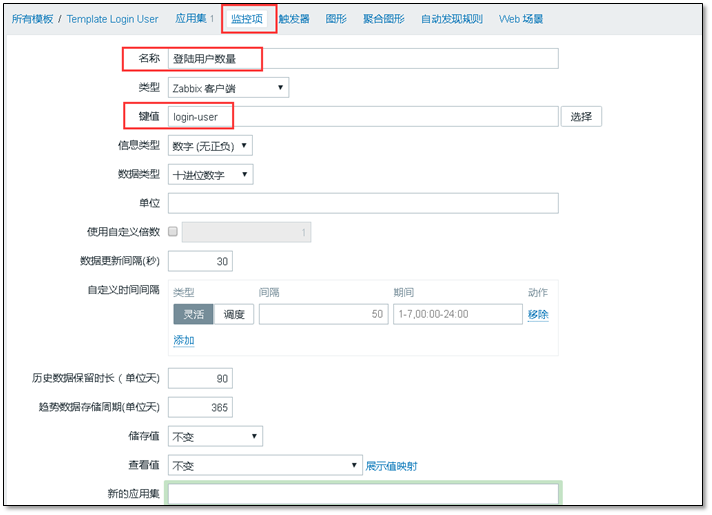
      自定义应用集的名称，然后点击添加

**③   创建监控项**

监控项 >> 创建监控项



**键值** -- key,即前面出创建的login-user。



   注意：创建监控项的时候，注意选择上应用集，即之前创建的安全。



**④   创建触发器**

触发器的作用：当监控项获取到的值达到一定条件时就触发报警

*(根据需求创建)*

触发器 >> 创建触发器

创建触发器，自定义名称，该名称是报警时显示的名称。

   表达式，点击右边的添加，选择**表达式**。

   严重性自定义。

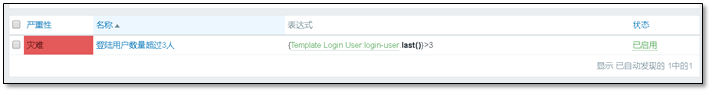


**表达式的定义 ↓ ，选择**之前创建的监控项，

最新的T值为当前获取到的值。



   添加完成，能够在触发器中看到添加的情况



**⑤   创建图形**

以图形的方式展示出来监控信息

图形 >> 创建图形

名称自定义，关联上监控项。



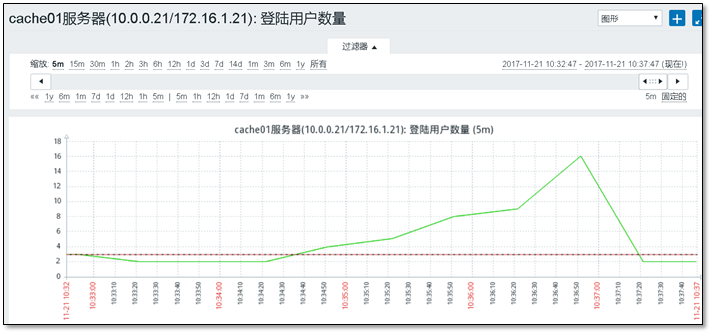
**⑥   主机关联模板**

配置 >> 主机

   一个主机可以关联多个模板



**4.2.4 查看监控的图形**



**4.3 监控报警**

**4.3.1 第三方报警平台**

http://www.**onealert**.com

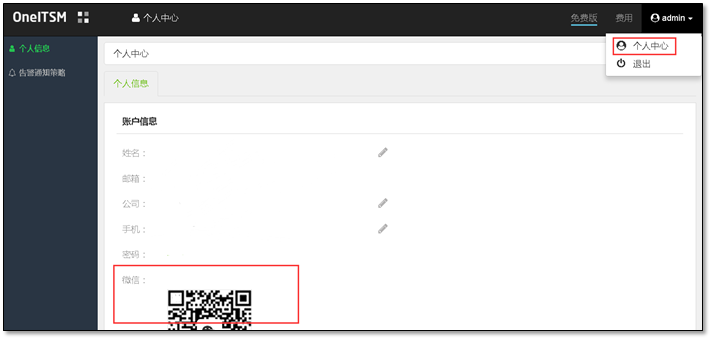
   　 通过 OneAlert 提供的通知分派与排班策略，以及全方位的短信、微信、QQ、电话提醒服务，您可以在最合适的时间，将最重要的信息推送给最合适的人员。

**4.3.2 onealert配置**

添加应用，注意添加的是zabbix



   实现微信报警需要关注微信公众号即可。



**4.3.3 安装 onealert Agent**

1.切换到zabbix脚本目录(如何查看zabbix脚本目录)：

cd /usr/local/zabbix-server/share/zabbix/alertscripts

#查看zabbix脚本目录

vi /etc/zabbix/zabbix\_server.conf

查看AlertScriptsPath

2.获取OneITSM agent包：

wget http://www.onealert.com/agent/release/oneitsm\_zabbix\_release-1.0.1.tar.gz

3.解压、安装。

tar -zxf oneitsm\_zabbix\_release-1.0.1.tar.gz

cd oneitsm/bin

bash install.sh --#个人生成的key

注：在安装过程中根据安装提示，**输入zabbix管理地址、管理员用户名、密码**。

Zabbix管理地址: http://10.0.0.61/zabbix/

Zabbix管理员账号: Admin

Zabbix管理员密码:

4.当提示"安装成功"时表示安装成功!

验证告警集成

产生新的zabbix告警(problem),动作状态为“已送达”表示集成成功。

**4.3.1 如何删除onealert Agent**

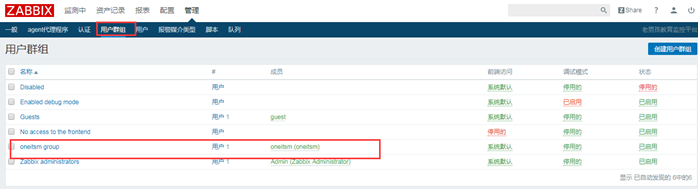
①  删除报警媒介类型中的脚本



②  删除创建的用户



③  删除用户群组



④  删除创建的动作



**4.3.2 触发器响应，发送报警信息**



   在微信和邮件中，均能收到报警信息。



**注意：当状态改变的时候才会发邮件**

   　　 好-->坏

　　    坏-->好

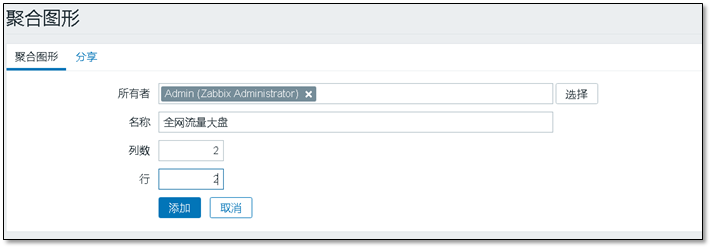
**4.4 监控可视化**

**4.4.1 聚合图形**

最新数据 >> 图形



   自定义名称



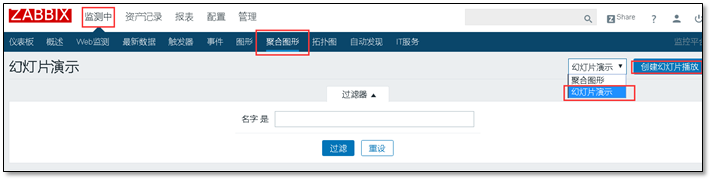
   点击聚合图形的名称，进行更改，添加要显示的图形即可。



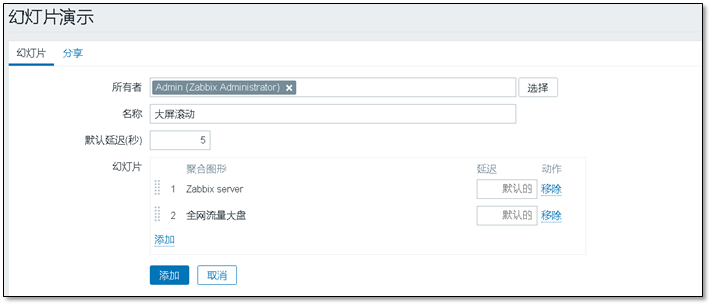
**4.4.2 幻灯片**

添加幻灯片

监测中 >> 复合图形 >> 幻灯片演示



   创建幻灯片，名称自定，选择要显示的



   幻灯片根据设定的时间自动播放

**4.5 模板的共享**

**4.5.1 主机共享**

在主机页打开，全选后点击导出



   导入



**4.5.2 模板共享**

**https://github.com/zhangyao8/zabbix-community-repos**



**第5章 监控全网服务器**

**5.1 需求说明**

实际需求：公司已经有了100台服务器，现在需要使用zabbix全部监控起来。

**5.2 规划方案**

常规监控：cpu，内存，磁盘，网卡  问题：怎样快速添加100台机器

   　　方法1：使用克隆的方式

 　　  方法2：自动注册和自动发现

 　　  方法3：调用zabbix api接口  curl 、python

        　　  开发自己的运维平台兼容zabbix的通道

**服务监控，url监控等特殊监控**：自定义监控

**5.2.1 api接口使用（*curl*）**

curl -i -X POST -H 'Content-Type:application/json' -d'{"jsonrpc": "2.0","method":"user.login","params":{"user":"Admin","password":"zabbix"},"auth": null,"id":0}' "http://10.0.0.61/zabbix/api\_jsonrpc.php"

curl -i -X POST -H 'Content-Type:application/json' -d'

{

"jsonrpc": "2.0",

"method": "host.get",

"params": {

"output": [

"hostid",

"host"

],

"selectInterfaces": [

"interfaceid",

"ip"

]

},

"id": 2,

"auth": "6a450a8fc3dce71fd310cfe338746578"

}' "http://10.0.0.61/zabbix/api\_jsonrpc.php"

**5.3 具体实施规划**

**5.3.1 硬件、系统、网络监控**

　　所有集群节点（所有虚拟机）都监控上

　　交换机，路由器监控（简单方法：换成端口对应服务器网卡流量监控；标准方法：监控交换机的网卡）

　　snmp监控

**5.3.2 应用服务监控**

1. 监控备份服务器，简单方法是监控rsync端口，如果有其他更佳方案可以说明；

方法1：监控873端口net.tcp.port[,873]

方法2：模拟推送拉取文件

2. 监控NFS服务器，使用监控NFS进程来判断NFS服务器正常，如果有其他更佳方案可以说明；

方法1：端口（通过111的rpc端口获取nfs端口） net.tcp.port[,111]

方法2：showmount -e ip|wc -l

3. 监控MySQL服务器，简单方法监控mysql的3306端口，或者使用zabbix提供的Mysql模板，如果有其他更佳方案可以说明；

方法1：端口（通过3306的mysql端口） net.tcp.port[,3306]

方法2：mysql远程登录

方法3：使用zabbix agent自带的模板及key

4. 监控2台web服务器，简单方法监控80端口，如果有其他更佳方案可以说明；

方法1：端口（通过80的web端口） net.tcp.port[,80]

方法2：看网页状态码、返回内容==zabbix 自带WEB检测

5. 监控URL地址来更精确的监控我们的网站运行正常；

使用zabbix自带的监控Web监测 进行监控

6. 监控反向代理服务器，PPTP服务器等你在期中架构部署的服务。

nginx，pptp

ntp 端口udp 123

7. 监控Nginx的7种连接状态。

自定义监控

**5.3.3 监控服务通用方法**

　　1. 监控端口 netstat ss lsof  ==》 wc -l

　　2. 监控进程 ps -ef|grep 进程|wc -l  试运行一下

　　3. 模拟客户端的使用方式监控服务端

  　　    web  ==》 curl

     　　 mysql ==》 select insert

   　　   memcache ==》 set再get

**5.4 实施全网监控**

*安装客户端脚本，for centos6*

#!/bin/bash

#设置解析

# echo '10.0.0.1 mirrors.aliyuncs.com mirrors.aliyun.com repo.zabbix.com' >> /etc/hosts

#安装zabbix源、aliyu nYUM源

curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-6.repo

curl -o /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-6.repo

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/6/x86\_64/zabbix-release-3.0-1.el6.noarch.rpm

yum clean all

yum clean all

#安装zabbix客户端

yum install zabbix-agent -y

sed -i.ori 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_agentd.conf

/etc/init.d/zabbix-agent start

#写入开机自启动

chmod +x /etc/rc.d/rc.local

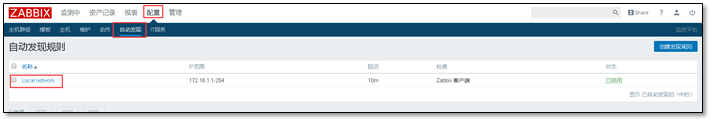
cat >>/etc/rc.d/rc.local<<EOF

/etc/init.d/zabbix-agent start

EOF

**5.4.1 使用自动发现规则**

添加自动发现规则



   创建发现动作



   查看自动发现的机器。



**5.4.2 监控备份服务器**

利用系统自带键值进行监控*net.tcp.listen[port]* 创建新的模板



在服务端进行测试

[root@m01 ~]# zabbix\_get -s 172.16.1.41 -p 10050 -k "net.tcp.listen[873]"

1

# 1为端口在监听 0为端口未监听

将模板添加到主机



**5.4.3 监控NFS服务器**

创建nfs监控模板

使用*proc.num[<name>,<user>,<state>,<cmdline>]*  键值，检测nfs进程的数量



在服务端进行测试

[root@m01 ~]# zabbix\_get -s 172.16.1.31 -p 10050 -k "proc.num[,,,rpc]"

5

[root@m01 ~]# zabbix\_get -s 172.16.1.31 -p 10050 -k "proc.num[nfsd,,,]

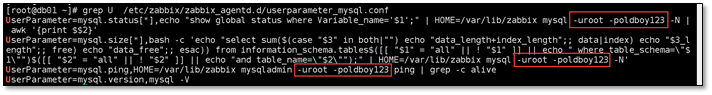
8

将模板绑定到主机



**5.4.4 监控MySQL服务器**

将自带的mysqlkey值加上mysql的账户密码，否则不能获取到数据。



使用系统自带模板  net.tcp.port[<ip>,port] 利用自带的监控端口键值进行监控



添加新的mysql监控项端口



[root@m01 ~]# zabbix\_get -s 172.16.1.51 -p 10050 -k "net.tcp.port[,3306]"

1

#检查是否能建立 TCP 连接到指定端口。返回 0 - 不能连接；1 - 可以连接

将模板关联到主机



**5.4.5 监控web服务器**

创建监控模板 监控 nginx服务与 80 端口

proc.num[<name>,<user>,<state>,<cmdline>] 进程数。返回整数

net.tcp.port[<ip>,port] 检查是否能建立 TCP 连接到指定端口。返回 0 - 不能连接；1 - 可以连接



[root@m01 ~]# zabbix\_get -s 172.16.1.8 -p 10050 -k "proc.num[,,,nginx]"

2

[root@m01 ~]# zabbix\_get -s 172.16.1.8 -p 10050 -k "net.tcp.port[,80]"

1

将模板关联到主机



**5.4.6 监控URL地址**

创建监测页面

echo ok >> /application/nginx/html/www/check.html

测试监控面页

[root@web03 ~]# for ip in 7 8 9 ;do curl 10.0.0.$ip/check.html ;done

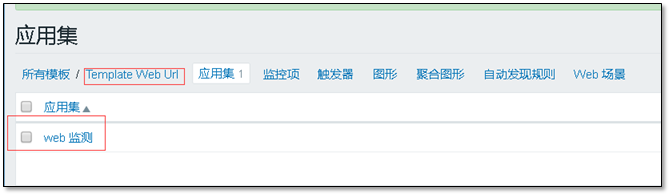
ok

ok

ok

创建web监测模板

*创建应用集*



*创建Web场景*



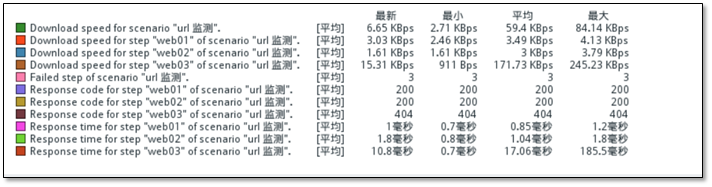
*创建图形*

[](http://10.0.0.61/zabbix/chart2.php?graphid=668&period=60&stime=20191122164643&updateProfile=1&profileIdx=web.screens&profileIdx2=668&width=1052&sid=8a1fe32c9614ae35&screenid=&curtime=1511340464305)

将模板关联到主机



监测结果



**5.4.7 监控反向代理服务器**

创建自定义key

[root@lb01 ~]# cat /etc/zabbix/zabbix\_agentd.d/userparameter\_nk.conf

UserParameter=keep-ip,ip a |grep 10.0.0.3|wc -l

在服务端测试

[root@m01 ~]# zabbix\_get -s 172.16.1.5 -p 10050 -k "keep-ip"

1

[root@m01 ~]# zabbix\_get -s 172.16.1.6 -p 10050 -k "keep-ip"

0

在web界面添加模板



将模板关联到主机



**5.4.8 监控Nginx的7种连接状态**

nginx服务器显示status

……

location /status {

stub\_status on;

access\_log off;

}

……

[root@web01 ~]# for ip in 7 8 9 ;do curl 172.16.1.$ip/status ;done

Active connections: 1

server accepts handled requests

73 73 69

Reading: 0 Writing: 1 Waiting: 0

Active connections: 1

server accepts handled requests

134 134 127

Reading: 0 Writing: 1 Waiting: 0

Active connections: 1

server accepts handled requests

7 7 7

Reading: 0 Writing: 1 Waiting: 0

在nginx服务器上添加key

cat >/etc/zabbix/zabbix\_agentd.d/userparameter\_nginx\_status.conf <<'EOF'

UserParameter=nginx\_active,curl -s 127.0.0.1/status|awk '/Active/ {print $NF}'

UserParameter=nginx\_accepts,curl -s 127.0.0.1/status|awk 'NR==3 {print $1}'

UserParameter=nginx\_handled,curl -s 127.0.0.1/status|awk 'NR==3 {print $2}'

UserParameter=nginx\_requests,curl -s 127.0.0.1/status|awk 'NR==3 {print $3}'

UserParameter=nginx\_reading,curl -s 127.0.0.1/status|awk 'NR==4 {print $2}'

UserParameter=nginx\_writing,curl -s 127.0.0.1/status|awk 'NR==4 {print $4}'

UserParameter=nginx\_waiting,curl -s 127.0.0.1/status|awk 'NR==4 {print $6}'

EOF

服务端测试

[root@m01 ~]# zabbix\_get -s 172.16.1.7 -p 10050 -k "nginx\_waiting"

0

[root@m01 ~]# zabbix\_get -s 172.16.1.8 -p 10050 -k "nginx\_waiting"

0

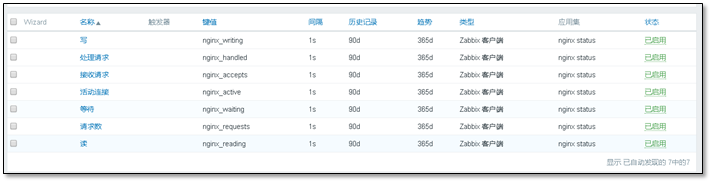
[root@m01 ~]# zabbix\_get -s 172.16.1.9 -p 10050 -k "nginx\_waiting"

0

在zabbix-web上添加



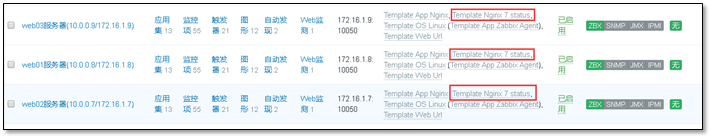
监控项



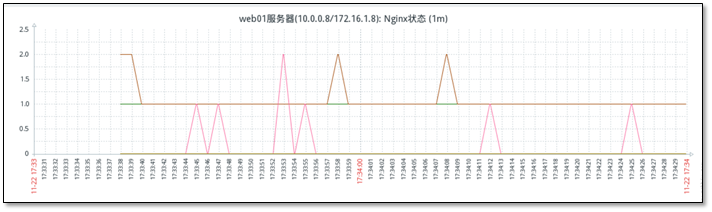
添加图形

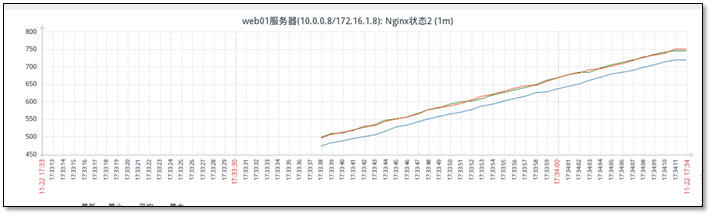


将模板关联到主机



查看添加的图形





**第6章 自动发现与自动注册**

**6.1 自动注册与自动注册**

**6.1.1 简介**

***自动发现：***

zabbix Server主动发现所有客户端，然后将客户端登记自己的小本本上，缺点zabbix server压力山大（网段大，客户端多），时间消耗多。

***自动注册：***

zabbix agent主动到zabbix Server上报到，登记；缺点agent有可能找不到Server（配置出错）

**6.1.2 两种模式**

被动模式：默认 agent被server抓取数据 （都是在agent的立场上说）

主动模式：agent主动将数据发到server端 （都是在agent的立场上说）

**注意： 两种模式都是在agent上进行配置**

**zabbix 的使用要在hosts文件中预先做好主机名的解析**

**6.2 自动发现--被动模式**

　第一个里程碑：完成之前的安装

zabbix Server安装完毕

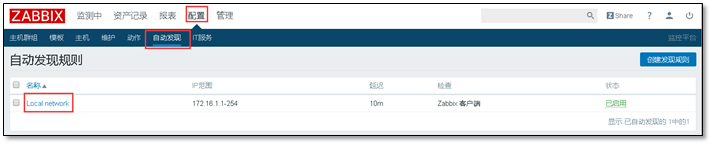
   第二个里程碑：配置agent客户端

zabbix agent安装完毕，注意配置Server=172.16.1.61

   第三个里程碑：在web界面上进行配置

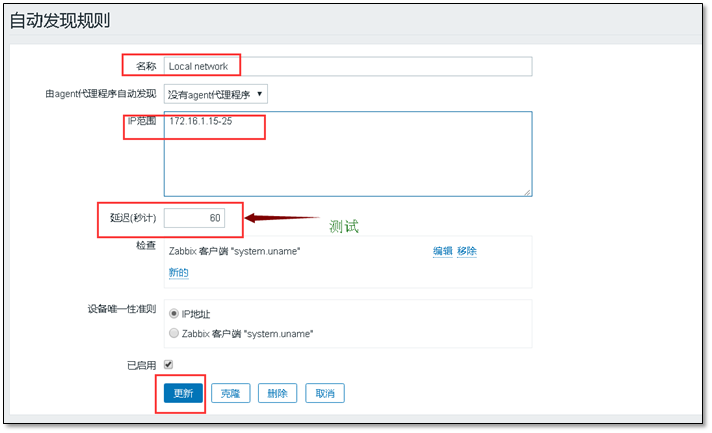
web界面：配置 >> 自动发现 >> Local network

使用自带的自动发现规则（进行修改）即可



在ip范围内输入ip，注意格式；

延迟在实际的生产环境中要大一些，实验环境可以小一些

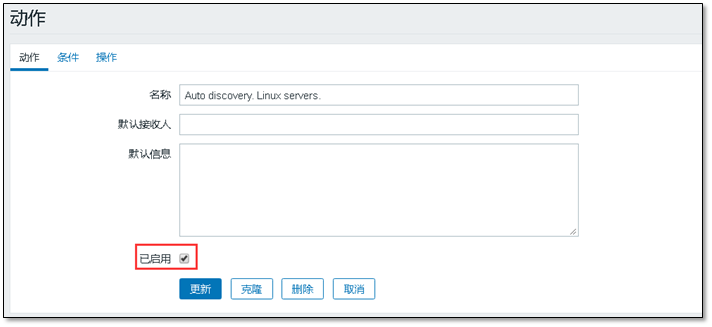


**创建发现动作**

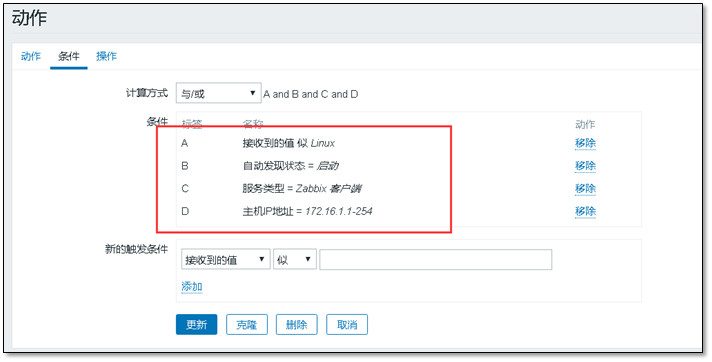
配置 >> 动作 >> Auto discovery. Linux servers.



①  配置动作

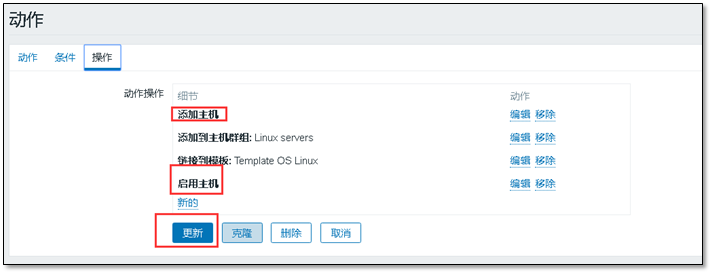


②  在条件中添加条件，让添加更准确



③  在操作中添加

a)  添加主机与启用主机



  　　  然后等待者客户端自动上门就好😏

**6.3 自动注册--主动模式**

　第一个里程碑：zabbix Server安装完毕 （完成）

zabbix Server安装完毕

   第二个里程碑：zabbix agent安装完毕，需要额外增加的配置

vim /etc/zabbix/zabbix\_agentd.conf

ServerActive=172.16.1.61

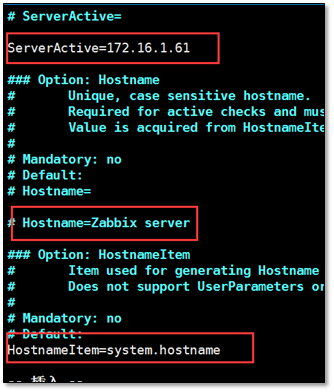
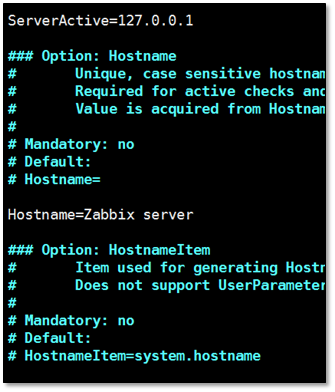
# Hostname=Zabbix server

HostnameItem=system.hostname

systemctl restart zabbix-agent.service

netstat -tunlp|grep zabbix

    源文件与修改后对比

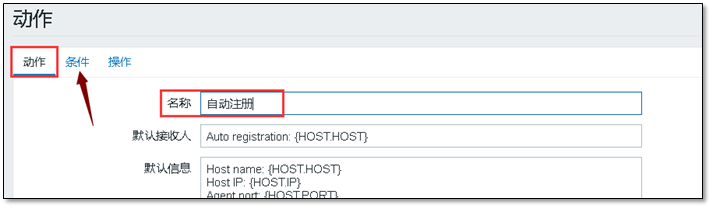


　   第三个里程碑：在web见面上进行配置

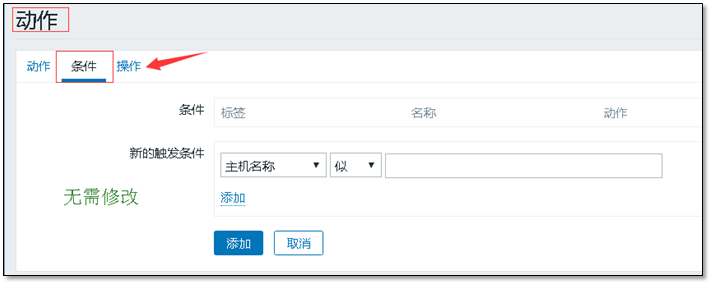
1 配置 >> 动作 >> 事件源(自动注册) >> 创建动作



    创建动作，添加名称即可

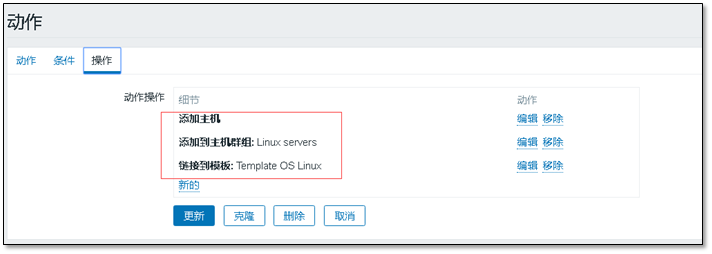


    条件中也无需修改



    在动作中添加动作

（添加主机、添加到主机群组、链接到模板）



    添加完动作后，等待就行了

    注意：重启客户端可以加速发现。但是在生产环境中勿用。



**第7章 分布式监控与SNMP监控**

**7.1 分布式监控**

**7.1.1 作用**

  　　分担压力，减轻负载

  　　多机房监控

　　zabbix Server  ===》  zabbix agent （只能同一个局域网监控）

***分担压力，降低负载***

zabbix Server ===》 zabbix proxy ===》zabbix agent1 agent2 agent3 。。。

172.16.1.61 172.16.1.21 172.16.1.0/24

===》 zabbix proxy ===》zabbix agent4 agent5 agent6 。。。

***多机房监控***

zabbix Server(北京) ==》 zabbix proxy（每个机房搭建） ==》 zabbix agent

122.71.240.233/172.16.1.61 122.71.241.11/172.16.2.21 172.16.2.0/24

**7.1.2 环境说明**

zabbix server m01

zabbix proxy cache01

zabbix agent cache01

**7.1.3 配置zabbix proxy**

**第一个里程碑**：配置zabbix yum源，并安装proxy

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

yum install zabbix-proxy-mysql -y

**第二个里程碑：安装数据库**

    zabbix  proxy也需要数据库，这个数据库不是用于存储监控数据的 只是用于存储配置信息

   #安装数据库

yum -y install mariadb-server

systemctl start mariadb.service

   #建立数据库

mysql

create database zabbix\_proxy character set utf8 collate utf8\_bin;

grant all privileges on zabbix\_proxy.\* to zabbix@'localhost' identified by 'zabbix';

exit

   #导入数据文件

zcat /usr/share/doc/zabbix-proxy-mysql-3.0.13/schema.sql.gz |mysql -uzabbix -pzabbix zabbix\_proxy

   #配置zabbix proxy 连接数据库

sed -i.ori '162a DBPassword=zabbix' /etc/zabbix/zabbix\_proxy.conf

sed -i 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_proxy.conf

sed -i 's#Hostname=Zabbix proxy#Hostname=cache01#' /etc/zabbix/zabbix\_proxy.conf

# Hostname 作为后面添加的代理程序名称，要保持一致

   #启动

systemctl restart zabbix-proxy.service

   #检查端口

[root@cache01 ~]# netstat -lntup |grep zabbix

tcp 0 0 0.0.0.0:10050 0.0.0.0:\* LISTEN 105762/zabbix\_agent

tcp 0 0 0.0.0.0:10051 0.0.0.0:\* LISTEN 85273/zabbix\_proxy

tcp6 0 0 :::10050 :::\* LISTEN 105762/zabbix\_agent

tcp6 0 0 :::10051 :::\* LISTEN 85273/zabbix\_proxy

**第三个里程碑：**修改agent配置指向 proxy

[root@cache01 ~]# grep ^Server /etc/zabbix/zabbix\_agentd.conf

Server=172.16.1.61

ServerActive=172.16.1.61

[root@cache01 ~]# sed -i 's#172.16.1.61#172.16.1.21#g' /etc/zabbix/zabbix\_agentd.conf

[root@cache01 ~]# grep ^Server /etc/zabbix/zabbix\_agentd.conf

Server=172.16.1.21

ServerActive=172.16.1.21

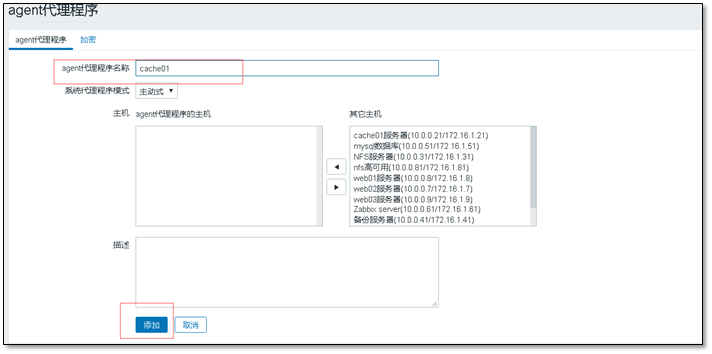
[root@cache01 ~]# systemctl restart zabbix-agent.service

**第四个里程碑：**web界面添加代理

    管理 >> agent代理程序 >> 创建代理



   代理程序名称要填写主机名



   稍等片刻就能在程序中出现代理



   在主机中能发现主机代理



**7.2 SNMP监控**

**7.2.1 使用范围**

　　无法安装agent  很多前辈的监控软件都可以监控各种设备  都是通过snmp监控

　　snmp simple network manager protocol 简单网络管理协议

  　 简单网络管理协议（SNMP），由一组网络管理的标准组成，包含一个应用层协议（application layer protocol）、数据库模型（database schema）和一组资源对象。该协议能够支持网络管理系统，用以监测连接到网络上的设备是否有任何引起管理上关注的情况。

**7.2.2 安装snmp程序**

yum -y install net-snmp net-snmp-utils

**7.2.3 配置snmp程序**

sed -i.ori '57a view systemview included .1' /etc/snmp/snmpd.conf

systemctl start snmpd.service

**7.2.4 测试snmp**

[root@m01 ~]# snmpwalk -v 2c -c public 127.0.0.1 sysname

SNMPv2-MIB::sysName.0 = STRING: m01

**说明：**

 　　   # snmpwalk 类似 zabbix\_get

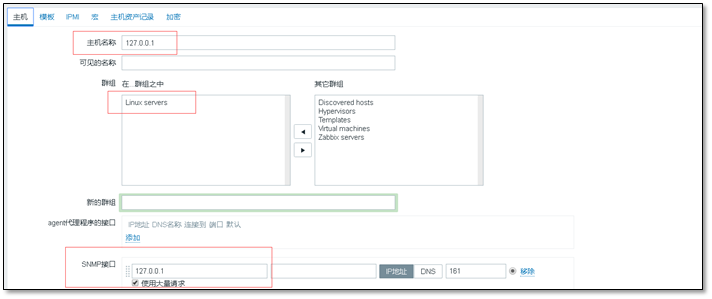
　　　# -v 2c  指定使用snmp协议的版本  snmp分为v1 v2 v3

　　　# -c public  指定暗号

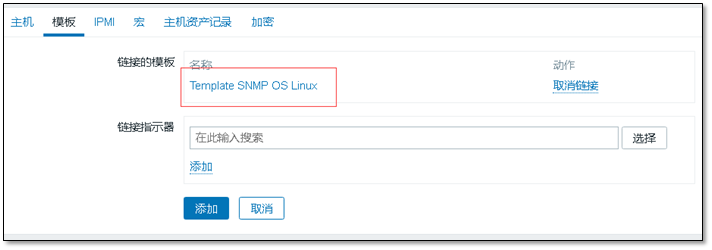
　　　# sysname  类似zabbix的key

**7.2.5 在web界面进行配置**

添加新的主机，注意使用snmp接口



选择模板，注意使用SNMP的模板



    添加完成就能够在主机中看到snmp监控对的主机



**7.2.6 附录**

##SNMP OID列表 监控需要用到的OID

http://www.ttlsa.com/monitor/snmp-oid/

cmdb 资源管理系统

作者：[惨绿少年](https://clsn.cnblogs.com/)  
出处：[http://clsn.io](http://clsn.io/)  
本文版权归作者所有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [运维基本功](https://www.cnblogs.com/clsn/category/1093760.html)

# [Zabbix](https://www.cnblogs.com/bubu99/p/12275447.html)

**为什么要监控**

* 运维的职责  
  1.保障企业数据的安全可靠。  
  2.为客户提供7\*24小时服务。  
  3.不断提升用户的体验。
* 在关键时刻，提前提醒我们服务器要出问题了
* 当出问题之后，可以便于找到问题的根源

拿到公司服务器的管理权之后，

第一件事，备份

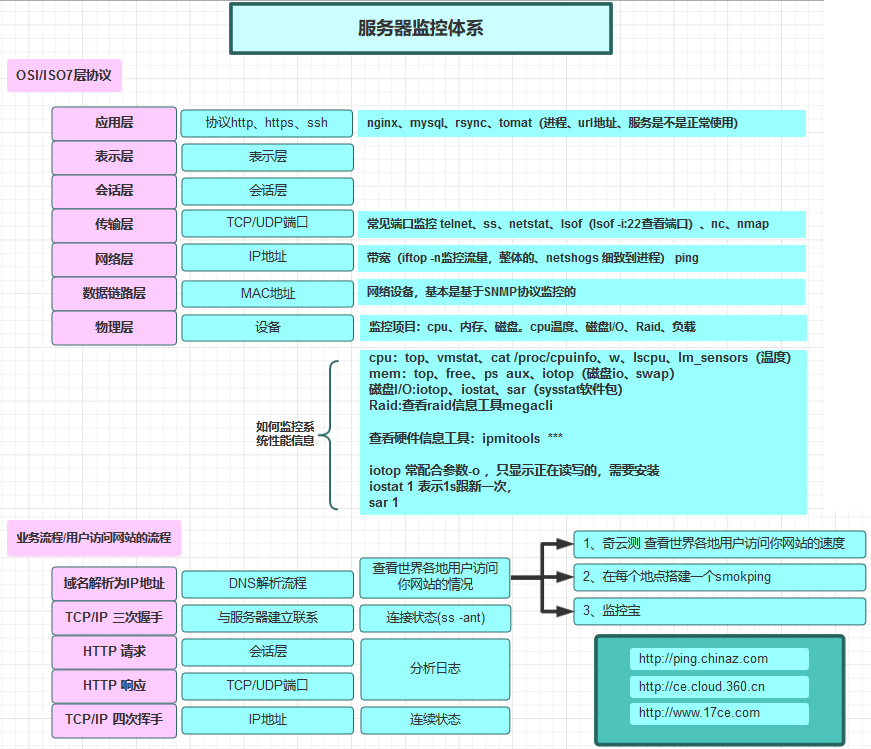
第二件事，监控

数据有故障了，服务有问题了，一些内容挂掉了，比如web服务器，或者是硬件的服务器，raid，磁盘有故障了也会报警，用户访问网站，感觉慢了，也可以获得相应的提示

这是一个非常重要的部分

**你们公司监控是怎么做的-面试题**

* OSI/ISO7层协议进行回答
* 业务流程、用户访问网站的流程进行补充



**监控工具**

mrtg    流量监控出图  
nagios  监控  
cacti   流量监控出图  
zabbix  监控+出图，所有监控范畴，都可以整合到Zabbix中。

* 硬件监控：Zabbix IPMI Interface
* 系统监控：Zabbix Agent Interface
* Java监控：Zabbix JMX Interface
* 网络设备监控：Zabbix SNMP Interface
* 应用服务监控：Zabbix Agent UserParameter
* MySQL数据库监控：percona-monitoring-plulgins
* URL监控：Zabbix Web 监控

**Zabbix简介**

zabbix是一个基于WEB界面的，提供分布式系统监视以及网络监视功能的企业级的开源解决方案

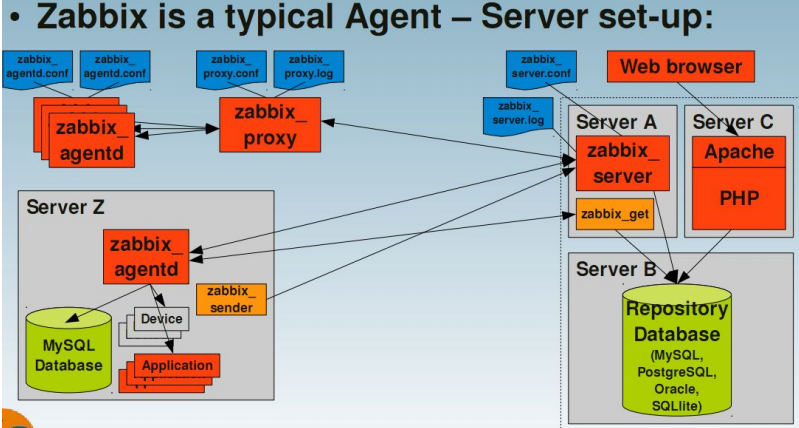
[官网：http://www.zabbix.com](http://www.zabbix.com/)

已经有20年历史，

web界面是基于PHP的，所以需要准备LNMP或者LAMP的环境

软件是c语言的

zabbix的架构图



 zabbix\_proxy 在大型架构中作为代理使用，将分部服务器的情况汇总发送到总部

**zabbix安装以及部署**

**先做环境检查**

从CentOS7模板机全新克隆一台新主机

IP规划为10.0.0.61/172.16.1.61，主机名为m01

注册账号：[http:](http://c.onealert.com/console/ucid/regist.jsp" \t "_blank)[//c.onealert.com/console//ucid/regist.jsp](http://c.onealert.com/console/ucid/regist.jsp)

账号用于接受报警

**安装zabbix服务端**

本次使用yum安装，编译安装留为作业

官网---download

官方安装说明中可以找到下面这些信息和安装步骤

版本号中的LTS指的是长时间维护版本

或者找zabbix官方源repo.zabbix.com----找到有release的，复制链接地址

# 安装Zabbix官方源

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

yum makecahe fast # 快点缓存

# 安装zabbix，httpd，php

yum install zabbix-server-mysql zabbix-web-mysql httpd php -y

**yum安装数据库**

数据库因为用的是centos7，MySQL是正常装不上的，需要配置MySQL源或者二进制安装

所以，偷懒安装mariadb（本质上基本和mysql是一样的）

# 安装mariadb

yum -y install mariadb-server

systemctl start mariadb.service

# 创建数据库，权限

mysql

create database zabbix character set utf8 collate utf8\_bin;

grant all privileges on zabbix.\* to zabbix@localhost identified by 'zabbix';  
创建用户的时候，不用flush跟新信息，跟新只在修改用户信息的时候

exit

# 导入数据库sql文件，官方安装说明中可以找到这些信息和安装步骤

zcat /usr/share/doc/zabbix-server-mysql-3.0.25/create.sql.gz|mysql -uzabbix -pzabbix zabbix

zcat 命令是显示gz压缩包内容的软件

请根据自己的zabbix软件版本 修改zcat的文件目录

**配置zabbix server**

# 配置zabbix Server连接mysql

sed -i.ori '115a DBPassword=zabbix' /etc/zabbix/zabbix\_server.conf

# 修改apache-php配置，因为我们是自动安装，他会自动创建一个虚拟主机  
# 我们修改下配置文件，添加时区，否则后面打开web页面安装zabbix的时候会报错

sed -i.ori '18a php\_value date.timezone Asia/Shanghai' /etc/httpd/conf.d/zabbix.conf

**启动zabbix server**

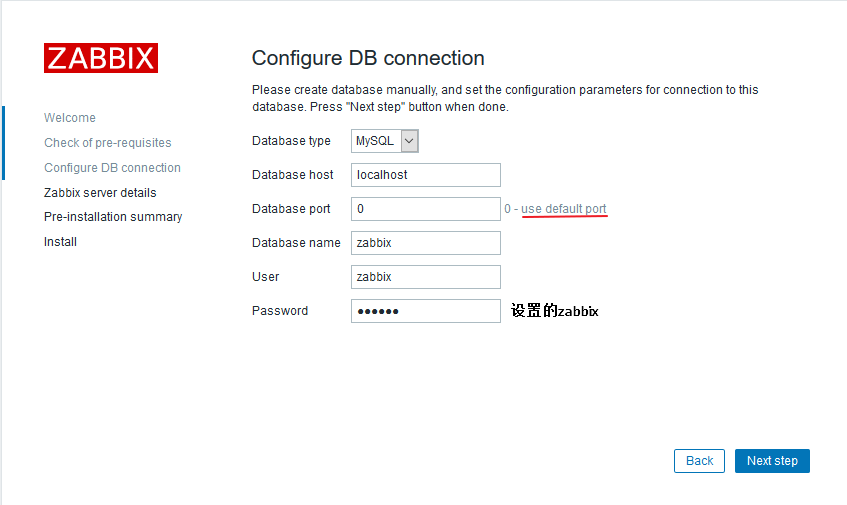
systemctl start zabbix-server # 启动zabbix

systemctl start httpd # 启动apache

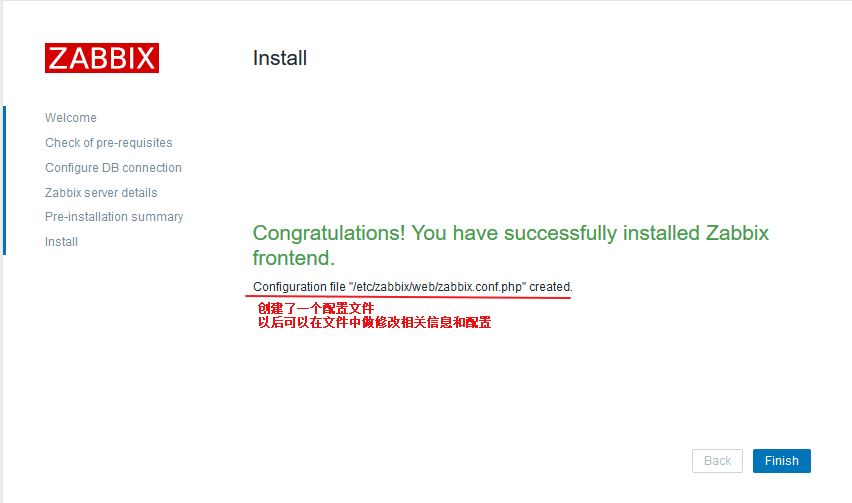
**网页配置**

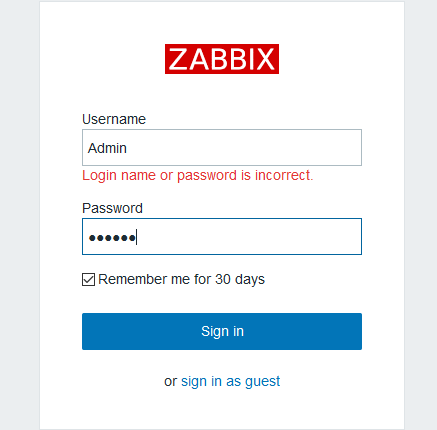
浏览器中访问10.0.0.61/zabbix 。记得用自己的IP

进入页面按照指令操作。









默认的账号Admin密码zabbix

进入主页，单击右上角的小人，修改显示语言和密码

**zabbix开始监控状态-安装配置zabbix-agent**

zabbix客户端就是zabbix agent

国内源：搜索

选择yum源的时候注意版本号，有的是centos6有的是7

可以直接安装agent，但是有一些依赖可能装不上

选择release最为稳妥

所有需要被监控的服务器都要执行下面步骤，下面以在m01和cache01安装客户端为例：

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

yum install zabbix-agent

sed -i.ori 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_agentd.conf

systemctl start zabbix-agent.service

**客户端测试**

此时，还不能在web页面张查看监控状态

需要先在客户端测试确认一下数据，需要用到zabbix-get

# 只能在服务端m01操作，需要使用-s执行相应客户端IP

yum install zabbix-get -y

# agent.ping 表示是否在线 1表示在线  
[root@m01 doc]# zabbix\_get -s 172.16.1.61 -p 10050 -k 'agent.ping'  
1

zabbix\_get -s 172.16.1.61 -p 10050 -k "system.cpu.load[all,avg1]"

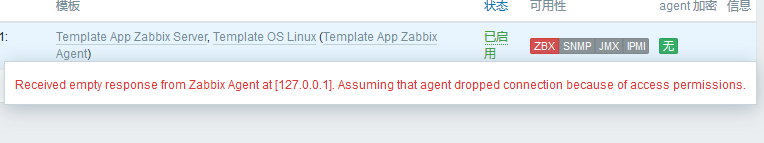
zabbix\_get -s 172.16.1.21 -p 10050 -k "system.cpu.load[all,avg1]"

web页面上怎么看？

配置----主机----将 zabbix server 启用--- 当可用性 都变成绿色，就表名监控上线

等待......如何加快（ping 主机名 可以解析）

等了很久，看一下日志吧...（/var/log/zabbix/zabbix\_server.log）



我们等来了一个错误，意思是，客户端无法访问服务端，是监控IP设置的问题

点击Zabbix server 将 agent代理程序的接口 这一项下面的 ip地址改为172.16.1.61，跟新保存

过一会就变成绿色了，就可以查看监控中---最新数据、图形等

因为中文字体的原因，有些地方会乱码，改成英文就行了

解决乱码问题，单独安装字体就可以了，不断地尝试，发现某个字体可用

yum -y install wqy-microhei-fonts  
将可用字体，复制到zabbix下

\cp /usr/share/fonts/wqy-microhei/wqy-microhei.ttc /usr/share/fonts/dejavu/DejaVuSans.ttf

**添加一台被监控服务器主机**

现在web01服务器上安装zabbix agent 客户端

清华或者阿里软件镜像站------找到zabbix----找到版本-----选择rhel----选择6（web01是centos6的系统）----选择64位----找到release----复制链接地址

在Linux中加入yum源

[root@web01 ~]# rpm -ivh https://mirrors.tuna.tsinghua.edu.cn/zabbix/zabbix/3.4/rhel/6/x86\_64/zabbix-release-3.4-1.el6.noarch.rpm

[root@web01 ~]# yum install zabbix-agent -y

以后这些操作要交给kickstart和cobbler，装完系统就装上zabbix-agent

#修改一下服务端的IP地址

第97行 Server = 172.16.1.61

或者使用sed -i.ori ''

sed -i.ori 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_agentd.conf

#将zabbix 设置为 开机自启动,并重启

[root@web01 ~]# chkconfig zabbix-agent on

[root@web01 ~]# /etc/init.d/zabbix-agent restart

Shutting down Zabbix agent: [FAILED]

Starting Zabbix agent: [ OK ]

在m01 zabbix的服务端上检查，这些服务都是要启动的，而且要开机自启动

[root@m01 doc]# systemctl is-active httpd.service mariadb.service zabbix-server.service zabbix-agent.service

active

active

active

active

[root@m01 doc]# systemctl is-enable httpd.service mariadb.service zabbix-server.service zabbix-agent.service

Unknown operation 'is-enable'.

[root@m01 doc]# systemctl is-enabled httpd.service mariadb.service zabbix-server.service zabbix-agent.service

disabled

disabled

disabled

enabled

[root@m01 doc]# systemctl enable httpd.service mariadb.service zabbix-server.service zabbix-agent.service   
[root@m01 doc]# systemctl is-enabled httpd.service mariadb.service zabbix-server.service zabbix-agent.service   
enabled  
enabled  
enabled  
enabled

在服务端检查一下是不是在线，可以连接

[root@m01 doc]# zabbix\_get -s 172.16.1.7 -p 10050 -k 'agent.ping'  
1

在web界面中  配置----主机----创建主机

主机名称： 主机名

可见名称： 随意设置，知道意思就行，

群组： 最好设置一个群组，方便批量管理

agent代理程序的接口： 172.16.1.7 端口号不用改

下面的SNMP、JMX、IPMI暂时不用管

接下来 点击 模板选项卡

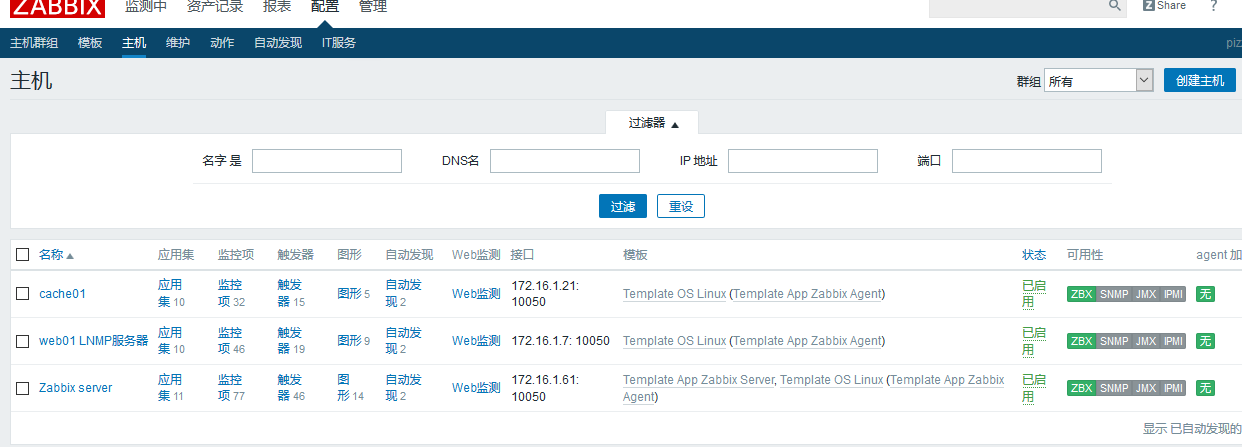
不用点击选择，在搜索框中输入linux ，选择Template OS linux-----点击小的添加----点击添加

保证在m01上能够解析web01 的名字   -----   172.16.1.7 web01

注：实际做的时候，我没有在host中解析web01 也连上了....后面有问题再说吧，

继续添加一台服务器cache01

都连接上了



**如何快速添加主机及键值**

随便选择一个，然后最下面有一个克隆，修改一下主机名和IP

模板中有什么呢？

可以查看监控项中的键值，就是zabbix\_get 命令中的-k参数 中的k值，是获取数据的本质

**自定义创建键值**

也叫作自定义监控，在模板提供的键值中没有我们需要的数据时，我们要自己做键值

zabbix自带模板Template OS Linux (Template App Zabbix Agent)  
提供CPU、内存、磁盘、网卡等常规监控，只要新加主机关联此模板，就可自动添加这些监控项。

需求：服务器登录人数不超过3人，超过3人报警

1、获取登陆服务器人数  
[root@web01 ~]# who  
root     pts/0        2019-03-16 15:49 (10.0.0.253)  
2、统计  
[root@web01 ~]# who|wc -l  
1  
3、将命令放在/etc/zabbix/zabbix\_agentd.d/userparameter\_mysql.conf 中  
这是用来放置用户自定义监控项的，用户可以在这个目录中写自己的.conf文件  
4、在agent注册key

# 语法：key名字要唯一，多个key以行为分割

UserParameter=<key>,<shell command>

UserParameter=login-user,who|wc -l

UserParameter=login-user,/bin/sh /server/scripts/login.sh # 执行脚本

# cd /etc/zabbix/zabbix\_agentd.d/

# vim userparameter\_login.conf

UserParameter=login-user,who|wc -l

# 修改配置文件要重启

systemctl restart zabbix-agent.service   
centos6中重启  
/etc/init.d/zabbix-agent restart

# 在server检测

# zabbix\_get -s 172.16.1.21 -p 10050 -k "login-user"

自定义key能被server和agent认可

**WEB页面中操作**

1. 创建模板：填好名称、群组，就可以保存了
2. 创建应用集：给监控项（item）分类
   1. 添加一个user，表示用户相关的
3. 创建监控项：监控的内容
   1. 点击监控项---创建监控项---填写内容（历史时长短一点，趋势时长可以长到365d）
4. 创建触发器：当监控项获取到的值 达到一定条件时 就触发报警
   1. 名称要表达清楚报警内容　　login.user bigger than 3
   2. 表达式要 选择好 {fist\_model\_web:login.user.last()}>3 这是选择后自动生成的
   3. 点击添加
5. 创建图形
   1. 在选中监控项后，的功能下拉框选最大值
   2. 绘图风格选先
   3. 其他随意
6. 主机关联模板
   1. 配置---主机---web01---模板---添加之前做好的模板

处理报警：在设置的触发器中有一项允许手动关闭，在收到报警后，我们可以在确认事件中关闭报警

**添加报警**

报警也可以在web中设置，就在主页小人处

我们还可以通过网站设置报警，然后等着报警信息，就不用一直盯着页面了

登陆网站：http://www.onealert.com/index.html 注册账号

报警的本质：

什么时候发邮件：当状态改变的时候才会发邮件

好--坏

坏--好

如何查找安装报警脚本的路径？

[root@m01 doc]# grep -i alertscript /etc/zabbix/zabbix\_server.conf

### Option: AlertScriptsPath

# AlertScriptsPath=${datadir}/zabbix/alertscripts

AlertScriptsPath=/usr/lib/zabbix/alertscript

安装报警插件，下载地址和安装命令都是报警网站提供的

[root@m01 alertscripts]# wget http://www.onealert.com/agent/release/oneitsm\_zabbix\_release-1.2.0.tar.gz

...下载过程略...

[root@m01 alertscripts]# tar -zxf oneitsm\_zabbix\_release-1.2.0.tar.gz

[root@m01 alertscripts]# cd oneitsm/bin

[root@m01 bin]# bash install.sh 42dd9ef9-cc1c-8def-e95a-d20526a5341f

./log.sh: line 6: /usr/lib/zabbix/alertscripts/oneitsm/bin/oneitsm.conf: No such file or directory

start to create config file...

Zabbix管理地址: 172.16.1.61

Zabbix访问失败,请重新输入.

Zabbix管理地址: 10.0.0.61

Zabbix访问失败,请重新输入.

Zabbix管理地址: 10.0.0.61/zabbix # 地址 zabbix的WEB页面的地址

Zabbix管理员账号: Admin

Zabbix管理员密码:

start to auth by zabbix admin user and password...

...安装过程略...

create action success!

安装成功.

登陆zabbix的WEB界面配置

管理---报警媒介类型---已经配置好了！！

在onealert的配置下中有一个分配策略---就是报警了分配给谁

**聚合图形**

同时看几个数据的图形

聚合图形中还有幻灯片演示

现有聚合图形，才能创建幻灯片

https://share.zabbix.com/中有许多别人创建好的模板

https://github.com/zhangyao8/zabbix-community-repos 模板共享

**全网监控项目说明**

考试目标：使用zabbix将期中架构所有集群监控

具体内容：

一、硬件、系统、网络监控

所有集群节点（所有虚拟机）都监控上

二、应用服务监控

1. 监控备份服务器，简单方法是监控rsync端口，如果有其他更佳方案可以说明；

2. 监控NFS服务器，使用监控NFS进程来判断NFS服务器正常，如果有其他更佳方案可以说明；

3. 监控MySQL服务器，简单方法监控mysql的3306端口，或者使用zabbix提供的Mysql模板，  
　　如果有其他更佳方案可以说明；

4. 监控2台web服务器，简单方法监控80端口，如果有其他更佳方案可以说明；

5. 监控URL地址来更精确的监控我们的网站运行正常；

6. 监控反向代理服务器，PPTP服务器等你在期中架构部署的服务。

7. 监控Nginx的7种连接状态。

项目一的模板都有

项目二中的端口、进程。可以进入任意模板查看，都有

项目二中的第5个检测URL，自定义web检测

　　创建web场景

项目二中的第6个检测端口+进程

项目二中的第7个Nginx的7中链接状态，用状态模块进行配置，

　　1、写7行（简单粗暴的方法）

　　2、写脚本

**自动发现与自动注册**

自动发现：zabbix Server主动发现所有客户端，然后将客户端登记自己的小本本上，  
　　缺点zabbix server压力山大（网段大，客户端多），时间消耗多。

自动注册：zabbix agent主动到zabbix Server上报到，登记；  
　　缺点agent有可能找不到Server（配置出错）

被动模式：默认，都是站在agent的立场上说话，agent被server抓取数据

主动模式：都是站在agent的立场上说话，agent主动的将数据发送给Server

两种模式都是在agent上的配置文件配置的

**自动发现（被动模式）-服务器少的时候使用**

第一步：zabbix Server安装完毕 （完成）

第二步：zabbix agent安装完毕，Server=172.16.1.61 （完毕）

第三步：网页上配置自动发现规则

配置---自动发现---新建或者使用已经有的

修改IP范围---172.16.1.1-10,172.16.1.45多个用，号分割

默认本质上使用的是system.uname，也可以添加新的

设置唯一性

更新或者保存，启用

------------------------------------------------------------------------------

配置---动作---事件源选择‘自动发现’---创建或者修改原有的，也可以克隆

其中，新的触发条件可以不写，因为在自动发现中已经设置了

在操作选项卡里面，添加新的动作操作（添加主机和启用主机）

完成后，跟新启用

什么时候算是完成自动发现了呢，在自动发现中显示主机名就算成功了，然后会出现在主机列表中

**自动注册（主动模式）**

把之前的自动发现关闭

第一步：zabbix Server安装完毕 （完成）

第二步：zabbix agent安装完毕，需要额外增加的配置

vim /etc/zabbix/zabbix\_agentd.conf

ServerActive=172.16.1.61

# Hostname=Zabbix server 和下面的HostnameItem配置一个就行

HostnameItem=system.hostname # 定义获取服务器主机名的方式  
　 HostMetadataItem=system.uname # 主机的源数据

systemctl restart zabbix-agent.service  
　　 重启centos6的方式/etc/init.d/zabbix-agent restart

netstat -tunlp|grep zabbix

第三步：Server上做hosts解析，解析是在这里发挥作用的

echo "172.16.1.21 cache01" >> /etc/hosts

第四步：网页上配置自动注册

　　配置----动作---自动注册（事件源）  
　　动作-触发条件 设置 成 主机元数据 似 Linux（手打的）  
　　操作-添加主机---添加到群组---链接到模板  
　　添加

主机没有添加上，等候时间

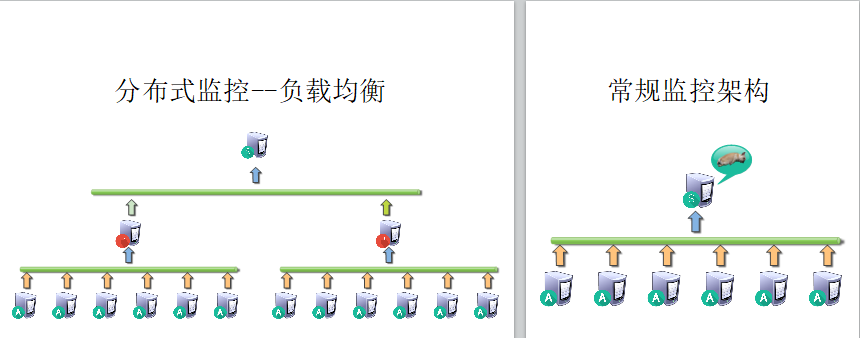
**分布式监控**

Zabbix Server -------->Zabbix agent （只能同一个局域网监控）

分布式监控：

a. 分担压力，降低负载

b. 多机房监控



等于是给各个分区的服务器安排了一个领导，层层汇总

环境准备

做好快照

机器准备：

Zabbix Server m01 监控中心

Zabbix Proxy web01 分中心领导

Zabbix Agent web01 web02 客户端

开始操作，将之前的自动发现和自动注册都关闭

第一步：在代理机web01安装zabbix proxy

rpm -ivh http://repo.zabbix.com/zabbix/3.0/rhel/7/x86\_64/zabbix-release-3.0-1.el7.noarch.rpm

yum -y install zabbix-proxy-mysql

第二步：安装数据库

zabbix proxy也需要数据库，这个数据库不是用于存储监控数据的 只是用于存储配置信息

# 安装数据库

yum -y install mariadb-server

systemctl start mariadb.service

#建库 创建用户授权

mysql

create database zabbix\_proxy character set utf8 collate utf8\_bin;

grant all privileges on zabbix\_proxy.\* to zabbix@'localhost' identified by 'zabbix';

# 将多余的空的用户删除  
drop user “‘@’localhost”；  
drop user “‘@’web01”；  
# 跟新  
flush privileges；  
exit

# 导入数据库文件

zcat /usr/share/doc/zabbix-proxy-mysql-3.0.13/schema.sql.gz |mysql -uzabbix -pzabbix zabbix\_proxy

第三步： 配置zabbix proxy，并在zabbix网页登记

# 配置zabbix proxy连接数据库

sed -i.ori '162a DBPassword=zabbix' /etc/zabbix/zabbix\_proxy.conf

systemctl start zabbix-proxy.service

# 修改Server

sed -i 's#Server=127.0.0.1#Server=172.16.1.61#' /etc/zabbix/zabbix\_proxy.conf

# 修改hostname，确保这个名字是唯一的，不用web01是因为，web01要给下面的客户端使用  
# 这个hostname，一定要能在m01中解析，即最顶端的老大  
sed -i 's#Hostname=Zabbix proxy#Hostname=proxy-web01#' /etc/zabbix/zabbix\_proxy.conf

将172.16.1.7      web01  proxy-web01 添加在m01的 host文件中

systemctl start zabbix-proxy.service

Centos6中的重启方法/etc/init.d/zabbix-proxy start(第一次启动所以使用的start  
### 问题 ###  
# 查看zabbix-proxy日志发现，一只连接不上数据库，它连接的的是/var/lib/mysql/mysql.sock  
　　# 在/etc/zabbix/zabbix\_proxy.conf中的默认的是/tmp/mysql.sock  
　　# 我们需要将它启动,把配置文件中DBSocket前面的# 删掉  
继续重启  
第四步：web页面设置  
管理---agent代理程序---创建代理  
agent代理程序名称：要和 Hostname=proxy-web01 一样  
把客户机加上  
添加

第五步：客户端agent 指向 proxy

[root@web01 ~]# grep ^Server /etc/zabbix/zabbix\_agentd.conf

Server=172.16.1.61

ServerActive=172.16.1.61

[root@web01 ~]# sed -i 's#172.16.1.61#172.16.1.7#g' /etc/zabbix/zabbix\_agentd.conf

[root@web01 ~]# grep ^Server /etc/zabbix/zabbix\_agentd.conf

Server=172.16.1.7

ServerActive=172.16.1.7  
# 文件中的Hostname 要和 页面上的一致，即本服务器的hostname

[root@web01 ~]# systemctl restart zabbix-agent.service

把客户端服务器hosts中的解析本机的去掉？？

主机的名字只需要在proxy和m01上解析就可以了

**SNMP监控**

主要用来监控网络设备，也可以用来监控服务器

只需要在网络设备中开启SNMP监控即可

小结：前面所讲内容全部属于zabbix agent方式监控

当前场景：无法安装agent，通过snmp监控

snmp ：simple network manager protocol 简单网络管理协议

# 在m01下，安装软件

yum -y install net-snmp net-snmp-utils

配置snmp

sed -i.ori '57a view systemview included .1' /etc/snmp/snmpd.conf  
意思是开启被读取的权限

systemctl start snmpd.service

默认的端口是161

使用SNMP

[root@db01 ~]# snmpwalk -v 2c -c public 127.0.0.1 sysname

# snmpwalk 类似 zabbix\_get

# -v 2c 指定使用snmp协议的版本 snmp分为v1 v2 v3

# -c public 指定暗号

# sysname 类似zabbix的key 键值中的键，还有很多其他的 OID

怎么通过snmp协议来监控呢？

在web页面中选择主机。在设置中的SNMP接口处添加

它和agent的区别是不支持自定义监控

模板也要改，把原来的linux os 取消链接。添加Template SNMP OS Linux

还是不行的话，点击原来的先添加再取消并清理，然后保存

补充：web监测的时候，要把SNMP的模板删除，换成之前用的普通的

**最核心的部分是监控的体系部分，一定要作为自己的必杀技，搞定**

分类: [Operations](https://www.cnblogs.com/bubu99/category/1620808.html)

# [Zabbix快速入门](https://www.cnblogs.com/yanshicheng/p/10436954.html)

**模板的管理与使用**

**一 模板**

　　模板是zabbix的核心,因为模板继承了所有要监控的内容,以及所要展示的图形等等,zabbix的安装部署完成之后,自带了很多模板(网络设备模板,操作系统模板,常用应用软件模板),这些模板足够满足日常百分之八十左右的应用需求.

　　模板的组成: 应用集, 监控项, 触发器, 图形, 聚合图形,自动发现, Web检测,

　　模板的重要组成: 监控项, 触发器, 自动发现, 图形

1.1.1 模板的分类

　　1) 监控系统状态的模板

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | # 对 Linux 系统的监控模板  Template OS Linux  # 对 Windows 系统的监控模板  Template OS Windows  # 对 Mac OS X 系统的监控模板  Template OS Mac OS X  # 对 VM VMware 系统的监控模板  Template VM VMware |

　　2) 监控网络和网络设备的模板

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #开启SNMPv1监控的模板  Template Module Generic SNMPv1    #开启SNMPv2监控的模板  Template Module Generic SNMPv2    Template Module Interfaces Simple SNMPv2  emplate Net Cisco IOS SNMPv2  Template Net Juniper SNMPv2  Template Net Huawei VRP SNMPv2</p> |

　　3) 监控应用软件和服务的模板

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | #对HTTPD服务的监控模板  Template App HTTP Service    #对MySQL服务的监控模板  Template DB MySQL    #对SSH服务的监控模板  Template App SSH Service    #对主机Ping的监控模板  Template Module ICMP Ping    #对Java服务的监控模板  Template App Generic Java JMX    #对Zabbix Agent状态的监控模板  Template App Zabbix Agent    #对Zabbix Server状态的监控模板  Template App Zabbix Server |

**二 创建应用集**

**1) 操作步骤**

　　配置 --->> 模板 ---> (新建模板/打开现有模板) --->> 应用集 --->> 创建应用集 --->> 输入要创建的应用集名称

　　创建应用集可以理解为一个群组,部门,列如要对CPU属性监控,那么就可以创建一个CPU的应用集,在这个应用集下面可以创建多个监控CPU的监控项.

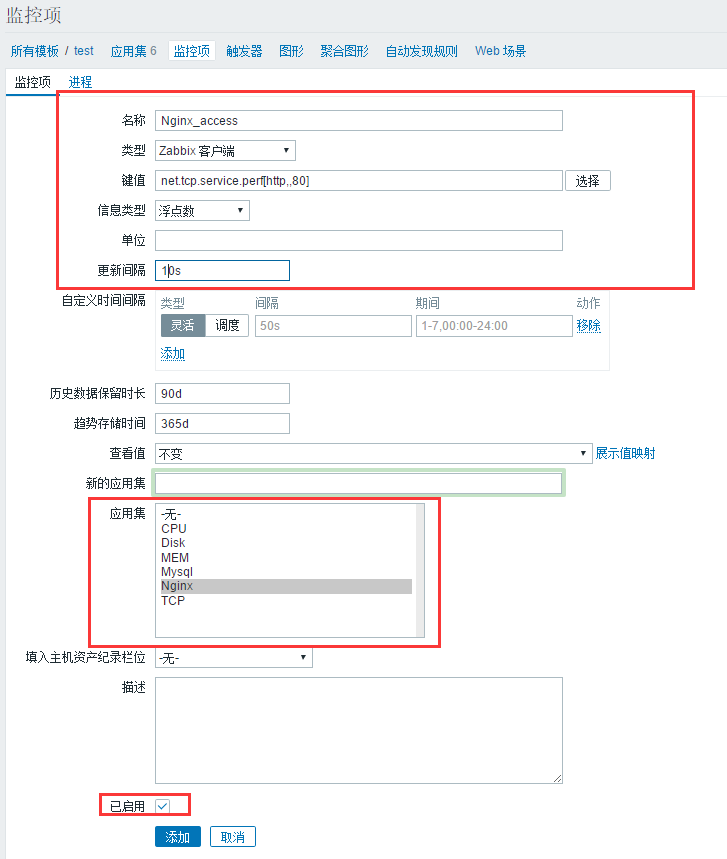
　　应用集的主要作用: 便于对监控项管理,对监控项进行分类



**三 创建监控项**

**1) 操作步骤**

　　　　配置 --->> 模板 ---> (新建模板/打开现有模板) --->> 监控项 --->> 创建监控项 --->> .....



**2) 参数说明**

　　1. 名称

　　　　　名称是创建的监控项的名称,自定义即可,

　　2. 类型

　　　　类型是设置此监控项通过什么方式进行监控,zabbix可选的监控类型有很多,常用的有 zabbix 客户端,zabbix客户端(主动模式),简单检查, SNMP客户端, zabbix采集器等,zabbix 客户端监控,也称为zabbix客户端(被动式)监控,这个需要根据需求进行选择.

　　2. 键值

　　　　键值可以自己定义,也可以使用zabbix 自带的键值,简单来说,键值就是获取数据的方式.

　　　　可以使用: [zabbix\_get -s 192.168.10.53 -k "agent.ping"] 工具进行测试

　　3. 信息类型

　　　　信息类型看具体返回的值,此信息类型为 浮点数,所以选择浮点数.

　　4. 单位

　　　　范围就是数据的返回单位,百分比还是什么

　　5. 更新间隔时间

　　　　数据获取更新的事件

　　6. 自定义时间间隔

　　　　灵活的在指定的时间段进行更新数据

　　7. 历史数据保留时长

　　　　历史数据的保留时间,自动删除

　　8. 趋势储存时间

　　　　趋势图的保留时间

　　9. 应用集

　　　　在那个应用集,

**四 创建触发器**

　　　触发器是用于故障告警的一个设置,将一个监控项,添加触发器后,此监控项如果出现问题,就会激活触发器,然后触发器将自动连接告警动作,最后触发告警.



　　触发器函数(diff,avg,last,nodata)

　　1) diff

|  |  |
| --- | --- |
| 1  2  3 | # 参数：不需要参数  # 支持值类型：float,int,str,text,log  # 作用：返回值为1表示最近的值与之前的值不同，即值发生变化，0表示无变化。 |

　　2) last

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | 参数：#num  支持值类型：float,int,str,text,log  作用：获取最近的值，“#num”表示最近第N个值，请注意当前的#num和其他一些函数的#num的意思是不同的，  例如：last(0)或last()等价于last(#1)，表示获取最新的值，last(#3)表示最近第3个值（并不是最近的三个值），  注意，last函数使用不同的参数将会得到不同的值，#2表示倒数第二新的数据。例入从老到最新值为1,2,3,4,5,6,7,8,9,10，last(#2)得到的值为9，last(#9)得到的值为2。  另外，last函数必须包含参数。 |

　　3) avg

|  |  |
| --- | --- |
| 1  2  3  4  5 | 参数：秒或#num  支持类型：float,int  作用：返回一段时间的平均值  例如，avg(5)表示最后5秒的平均值，avg(#5）表示最近5次得到值的平均值，avg(3600,86400）表示一天前的一个小时的平均值。  如果仅有一个参数，表示指定时间的平均值，从现在开始算起，如果有第二个参数，表示漂移，从第二个参数前开始算时间，#n表示最近n次的值。 |

　　4) change

|  |  |
| --- | --- |
| 1  2  3  4 | 参数：无需参数  支持类型：float,int,str,text,log  作用：返回最近获得值与之前获得值的差值，返回字符串0表示相等，1表示不同。  例如，change(0)>n表示最近得到的值与上一个值的差值大于n，其中，0表示忽略参数。 |

　　5) nodata

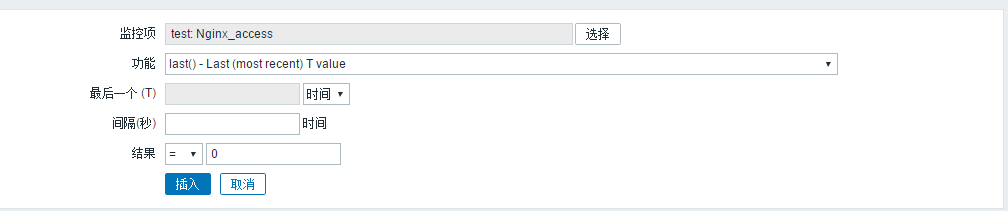
|  |  |
| --- | --- |
| 1  2  3 | 参数：秒  支持值类型：any  作业：探测是否能接收到数据，当返回值为1表示指定的间隔(间隔不应小于30秒)没有接收到数据，0表示其正常接收数据 |

　　6) count

|  |  |
| --- | --- |
| 1  2  3  4  5 | 参数：秒或#num  支持类型：float,int,str,text,log  作用：返回指定时间间隔内数值的统计。  例如：count(600)：表示最近10分钟得到值的个数  count(600,12)：表示最近10分钟得到值的个数等于12其中，第一个参数是指定时间段，第二个参数是样本数据。 |

　　7.) sum

|  |  |
| --- | --- |
| 1  2  3  4 | 参数：秒或#num  支持值类型：float,int  作用：返回指定时间间隔中收集到的值的总和，时间间隔作为第一个参数支持秒或收集值的数目(以#开始)。  例如：sum(600)：表示在600秒之内接收到所有值的和。sum(#5)：表示最后5个值的和。 |



作者：闫世成

出处：[http://cnblogs.com/yanshicheng](http://www.cnblogs.com/yanshicheng/)

联系：[yans121@sina.com](https://www.cnblogs.com/yanshicheng/p/yans121@sina.com)

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，如有问题或建议，请多多赐教，非常感谢。

# ELK学习总结——我们为什么要用ELK

# ****一. ELK是什么？****

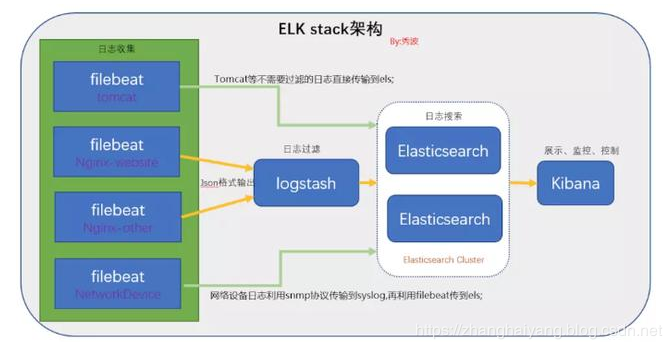
ELK 是elastic公司提供的**一套完整的日志收集以及展示的解决方案**，是三个产品的首字母缩写，分别是ElasticSearch、Logstash 和 Kibana。



**ElasticSearch简称ES**，它是一个实时的分布式搜索和分析引擎，它可以用于全文搜索，结构化搜索以及分析。它是一个建立在全文搜索引擎 Apache Lucene 基础上的搜索引擎，使用 Java 语言编写。  
**Logstash**是一个具有实时传输能力的数据收集引擎，用来进行数据收集（如：读取文本文件）、解析，并将数据发送给ES。  
**Kibana**为 Elasticsearch 提供了分析和可视化的 Web 平台。它可以在 Elasticsearch 的索引中查找，交互数据，并生成各种维度表格、图形。  
**二：ELK的用途**

传统意义上，**ELK是作为替代Splunk的一个开源解决方案**。Splunk 是日志分析领域的领导者。日志分析并不仅仅包括系统产生的错误日志，异常，也包括业务逻辑，或者任何文本类的分析。而基于日志的分析，能够在其上产生非常多的解决方案，譬如：

1.**问题排查**。我们常说，运维和开发这一辈子无非就是和问题在战斗，所以这个说起来很朴实的四个字，其实是沉甸甸的。很多公司其实不缺钱，就要稳定，而要稳定，就要运维和开发能够快速的定位问题，甚至防微杜渐，把问题杀死在摇篮里。日志分析技术显然问题排查的基石。基于日志做问题排查，还有一个很帅的技术，叫全链路追踪，比如阿里的eagleeye 或者Google的dapper，也算是日志分析技术里的一种。  
2.**监控和预警**。 日志，监控，预警是相辅相成的。基于日志的监控，预警使得运维有自己的机械战队，大大节省人力以及延长运维的寿命。  
3.**关联事件**。多个数据源产生的日志进行联动分析，通过某种分析算法，就能够解决生活中各个问题。比如金融里的风险欺诈等。这个可以可以应用到无数领域了，取决于你的想象力。  
4.**数据分析**。 这个对于数据分析师，还有算法工程师都是有所裨益的。



# ****三：ELK火热的原因？****

**ELK之前，有没有类似解决方案呢？**

某大神说是有的，当时应该是基于 Sphinx + Google char的。 Sphinx 对应ES, Google char 对应 Kibana。

**那为啥当时它没有火而现在的ELK火了呢？一种比较玄幻的解释是**：

事实上开源界永远有多种选择，比如基于java的lucene的es，也有基于c的lucy的dezi。但是谁火谁不火，真的是一个很玄妙的事情。

我觉得原因有很多方面，一个简单而较为核心的因数是时机，所谓时势造英雄是也。当然，任何一件事情不可能是一个因子引起的，或者我们说时机是一个较为宽泛抽象的因子。

**下面我从多个因素去阐述为什么ELK突然蓬勃发展？**

**1，数据(日志)的日益增多：**

早年能够产生足够数据的就那么一些站点，**而现在一个初创的企业可能都需要面临海量用户/海量请求/海量分析的压力**，其中产生的日志自然也是非常可观，而随着业务越来越复杂，微服务重新得到重视，无论系统日志，还是业务日志都更进一步了。运维或者开发们发现，我要从这么大规模的系统中(几百个上千个服务)产生的这么多日志(千亿规模)，去排查问题，简直是没有可能了。以前有这么大数据量的公司，都是有实力的公司，他们可能有内部专用的系统去处理。然而**现在突然成为了一个普遍需求，这个时候ELK顺势而上，也就水到渠成**。

**2，开源：**

开源现在已经融入到IT社区的血液里。虽然我们说商业,自研，还有开源三者之间是相辅相成，相濡以沫或者偶尔会相爱相杀，但是如果有开源可以选择，显然大部分开发或者运维还是首选开源的。有位大牛说的好：  
**ELK其开源属性，显然是比Splunk 略胜一筹的。**

**3，行业属性：**

有些行业对日志的依赖是非常大的，比如 CDN 日志除了能排查错误，对其分析还能对CDN调度等很多方面产生影响，这些都是实打实的经济效益。

**4，运维自身发展：**

运维本身也在发展，不可能一直在刀耕火种的年代。而日志对于运维来说，应该算是命根子了。对一个成体系的，标准化的日志分析方案的需求，也是历史发展的必然。**ELK在恰当的时候产生，运维接受他就是自然而然的了。**

**5，ELK自身属性问题：**

引用一位大神的说法：

ELK能解决的核心问题，覆盖面也广，标准化，易扩展集成，开发和运维都对其感冒

**ELK 本身非常易用，现在也有一个非常好的社区，加上需求如此之大，不火都不行。**

**6，大数据导致机器开始变得廉价：**

大数据的一个很好的副作用是让机器在某种意义上变得廉价了。少则几十上百台，多则上万甚至几十万台。服务器数量的急速攀升促进了很多技术的发展，典型的比如现在火的不要不要的深度学习。这就意味着，拿出几十台，上百台服务器做日志分析，一点问题也没有，集中式的日志分析慢慢成为主流。而ELK也是一个典型的集中式日志分析方案。

**7，写入时计算->查询时计算：**

所谓写入时计算是指将数据经过较为复杂的处理，聚合，得到的结果直接面向查询。 写入时计算规则由查询需求决定。随着存储格式的不断进步，譬如列式存储等的普及，以及强大的计算资源(一个ES集群动则上百台)，使得直接存储原汁原味的数据，然后查询的时候做各种计算变得可能。而ELK已经提供较为强大的查询功能。总体而言，写入时计算的大方向是往查询时计算转化。查询时计算最大的优势是支持任意查询，不丢失信息。

# [ELK日志系统介绍](https://www.cnblogs.com/Epir/p/10694694.html)

**ELK介绍**

**需求背景：**

* 业务发展越来越庞大，服务器越来越多
* 各种访问日志、应用日志、错误日志量越来越多，导致运维人员无法很好的去管理日志
* 开发人员排查问题，需要到服务器上查日志，不方便
* 运营人员需要一些数据，需要我们运维到服务器上分析日志

**为什么要用到ELK：**

一般我们需要进行日志分析场景：直接在日志文件中 grep、awk 就可以获得自己想要的信息。但在规模较大也就是日志量多而复杂的场景中，此方法效率低下，面临问题包括日志量太大如何归档、文本搜索太慢怎么办、如何多维度查询。需要集中化的日志管理，所有服务器上的日志收集汇总。常见解决思路是建立集中式日志收集系统，将所有节点上的日志统一收集，管理，访问。

大型系统通常都是一个分布式部署的架构，不同的服务模块部署在不同的服务器上，问题出现时，大部分情况需要根据问题暴露的关键信息，定位到具体的服务器和服务模块，构建一套集中式日志系统，可以提高定位问题的效率。

一个完整的集中式日志系统，需要包含以下几个主要特点：

* 收集－能够采集多种来源的日志数据
* 传输－能够稳定的把日志数据传输到中央系统
* 存储－如何存储日志数据
* 分析－可以支持 UI 分析
* 警告－能够提供错误报告，监控机制

而ELK则提供了一整套解决方案，并且都是开源软件，之间互相配合使用，完美衔接，高效的满足了很多场合的应用。是目前主流的一种日志系统。

**ELK简介：**

ELK是三个开源软件的缩写，分别为：Elasticsearch 、 Logstash以及Kibana , 它们都是开源软件。不过现在还新增了一个Beats，它是一个轻量级的日志收集处理工具(Agent)，Beats占用资源少，适合于在各个服务器上搜集日志后传输给Logstash，官方也推荐此工具，目前由于原本的ELK Stack成员中加入了 Beats 工具所以已改名为Elastic Stack。

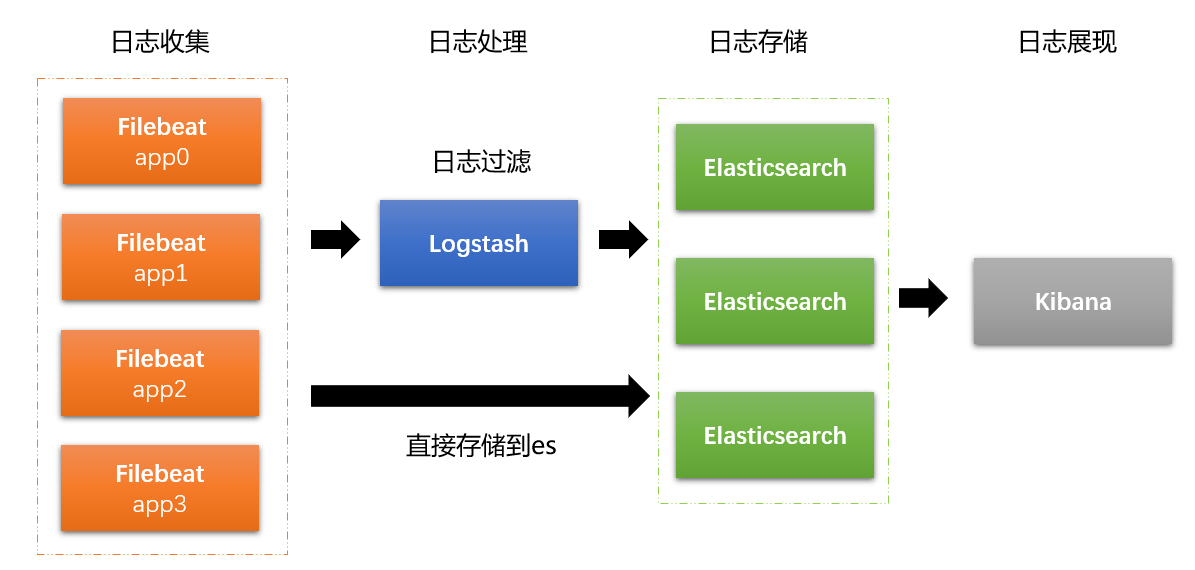
Elastic Stack包含：

* Elasticsearch是个开源分布式搜索引擎，提供搜集、分析、存储数据三大功能。它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，多数据源，自动搜索负载等。详细可参考[Elasticsearch权威指南](https://www.elastic.co/guide/cn/elasticsearch/guide/current/index.html)
* Logstash 主要是用来日志的搜集、分析、过滤日志的工具，支持大量的数据获取方式。一般工作方式为c/s架构，client端安装在需要收集日志的主机上，server端负责将收到的各节点日志进行过滤、修改等操作在一并发往elasticsearch上去。
* Kibana 也是一个开源和免费的工具，Kibana可以为 Logstash 和 ElasticSearch 提供的日志分析友好的 Web 界面，可以帮助汇总、分析和搜索重要数据日志。
* Beats在这里是一个轻量级日志采集器，其实Beats家族有6个成员，早期的ELK架构中使用Logstash收集、解析日志，但是Logstash对内存、cpu、io等资源消耗比较高。相比 Logstash，Beats所占系统的CPU和内存几乎可以忽略不计

ELK Stack （5.0版本之后）--> Elastic Stack == （ELK Stack + Beats）。目前Beats包含六种工具：

* Packetbeat： 网络数据（收集网络流量数据）
* Metricbeat： 指标 （收集系统、进程和文件系统级别的 CPU 和内存使用情况等数据）
* Filebeat： 日志文件（收集文件数据）
* Winlogbeat： windows事件日志（收集 Windows 事件日志数据）
* Auditbeat：审计数据 （收集审计日志）
* Heartbeat：运行时间监控 （收集系统运行时的数据）

上面是一些官方说明，比较清晰明了，下面是一般的ELK架构：



 filebeat是作为客户端的日志收集器，部署在若干个产出日志的应用系统上，将数据传输到es集群和logstash服务，传递给logstash的数据会被进一步处理最终流入es集群，kibana从es中获取索引日志信息进行展现。

之后会依次详细介绍logstash、beats、kibana组件，由于在工作中elasticsearch是作为存储组件使用的，无需特别调研，就不详细介绍了，只介绍些在elk中使用到的操作。

**标签:**[ELK](https://www.cnblogs.com/Epir/tag/ELK/)**,**[日志系统](https://www.cnblogs.com/Epir/tag/%E6%97%A5%E5%BF%97%E7%B3%BB%E7%BB%9F/)