**Zabbix**

1. **Zabbix简介**
2. **Zabbix形成**

**系统监控命令:**

**ps: 进程数量 df-h:查看硬盘**

**uptime:CPU负载 netstat或ss:端口,状态,有多少人连到本机电脑**

**awapon -s:查看交换分区信息 ifconfig:网卡 ping:主机是否为开启状态 iostat:硬盘每秒读写多少k tracerout:监控路由多少跳**

**free:内存------------------>以后工作中不要看free,要看available还有多少可用的**

**由于系统的命令行的监控命令比较多,输入起来比较麻烦,所以有人将这些命令整合成一套软件**

**Cacti:基于SNMP协议的控制然见,强大的绘图功能**

**Nagios:基于Agent监控,强大的状态检查与报警机制,插件多,可自己写监控脚本,非常方便**

**Zabbix:基于多种监控机值,支持分布式监控--------------------->结合了上两种的优点**

**2.Zabbix定义**

**是一个高度集成的监控解决方案**

**可以实现企业级的开元分布式监控**

**是通过C/S模式采集监控数据-------------------->Client/Server客户机和服务器结构**

**是通过B/S模式实现web管理------------------->Browser/Server，浏览器/服务器结构**

**3.Zabbix的特点**

**相对于其他的监控软件比较简单,而且功能比较强大------------------------>比较典型**

**适用于主机较多,不再适合用简单的命令监控的时候**

**是自带报警机制,自动化的监控------------->超过或未达到某个值的时候,就会自动报警**

**最后监控的结果会显示在一个web网页里**

**动态刷新30s一次**

**4.监控的目的**

**报告系统运行状况,提前发现问题,准备解决方案**

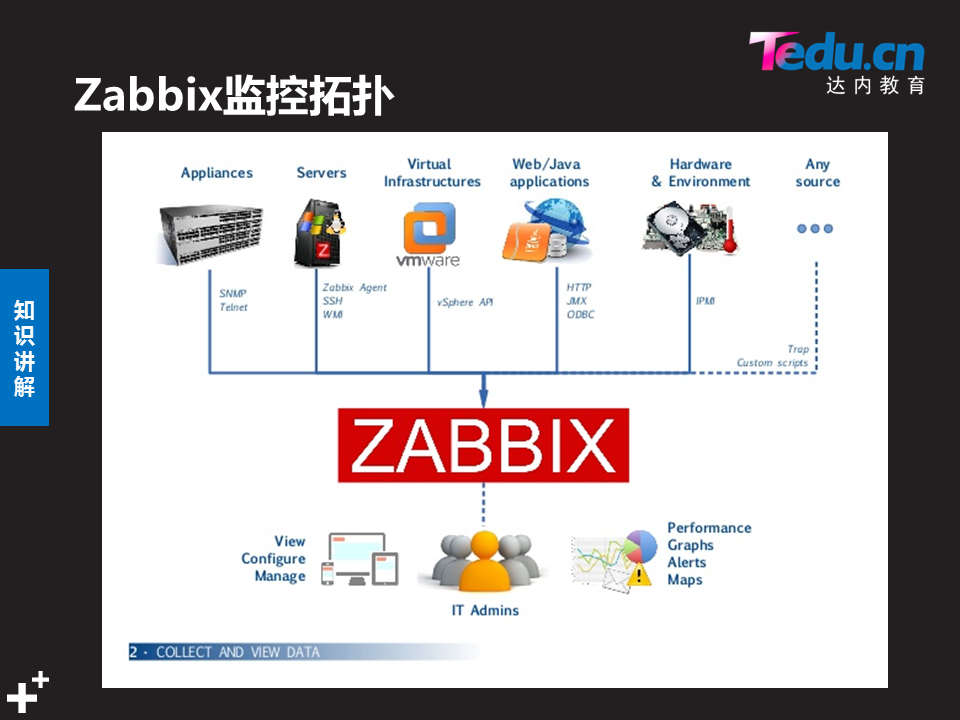
**5.监控的资源:**

**公开数据:web FTP SSH 数据库应用服务 TCP或UDP端口**

**私有数据:CPU 内存 磁盘 网卡流量等使用信息 用户进程运行信息**

**如果监控私有数据一定要在被监控的机器(有多少装多少)上要安装Agent**

1. **监控服务器:**
2. **被监控的主机**



**zabbix即是一个监控机制,同时也是一个web页面,所以要搭建zabbix首先就要搭建LNMP**

1. **部署Zabbix监控平台**
2. **部署监控服务器**
3. **安装LNMP环境**

**(二)部署监控服务器Zabbix Server**

**1.源码包安装**

**[root@zabbixserver lnmp\_soft]# yum -y install net-snmp-devel curl-devel**

**[root@zabbixserver lnmp\_soft]# yum -y install \ libevent-devel-2.0.21-4.el7.x86\_64.rpm**

**注意libevent-devel这个软件包在lnmp\_soft目录下有提供**

**[root@zabbixserver lnmp\_soft]# tar -xf zabbix-3.4.4.tar.gz**

**[root@zabbixserver lnmp\_soft]# cd zabbix-3.4.4/**

**[root@zabbixserver zabbix-3.4.4]# ./configure --enable-server \**

**> --enable-proxy --enable-agent --with-mysql=/usr/bin/mysql\_config \**

**> --with-net-snmp --with-libcurl**

**--enable-server安装部署zabbix服务器端软件**

**--enable-agent安装部署zabbix被监控端软件**

**--enable-proxy安装部署zabbix分布式监控----------->一般都是大框架才启用的模块**

**--with-mysql配置mysql\_config路径----------->可以知道mysql的所有主要信息信息**

**--with-net-snmp允许zabbix通过snmp协议监控其他设备---------->监控交换机路由器**

**--with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务测试被监控主机服务的状态**

**[root@zabbixserver zabbix-3.4.4]# make && make install**

**[root@zabbixserver zabbix-3.4.4]#zabbix\_table------->可以先检查一下是否安装成功**

**zabbix\_agented zabbix\_proxy zabbix\_server**

**zabbix\_get zabbxi\_sender-------->除非做自定义监控,其余情况都是自动启动**

**2.初始化Zabbix**

**因为zbbix后期会用到数据库,所以要让zbbix有数据库的操作权限.在数据库上建立用户,给予权限,让用户对数据库进行所有操作.**

**[root@zabbixserver ~]# mysql**

**mysql> create database zabbix character set utf8;--------------------->支持中文**

**mysql> grant all on zabbix.\* to zabbix@’localhost' identified by 'zabbix’;**

**创建用户和密码:所有权限 只对这个数据库中的所有表 给这个用户只能从本地连 密码是**

**因为是新建的数据库,所以里面什么都没有,所以,官网已经提前做好了表,就放在安装包中,只要倒入就可以用的.但是请注意:一定要按照先库再表最后数据的顺序,不能乱**

**[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/database/mysql/**

**[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < schema.sql**

**[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < images.sql**

**[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < data.sql**

**如果数据导入成功,如果不考虑图形界面,zbbix就可以基于命令行使用了**

**3.上线Zabbix的Web页面**

**因为工作中不可能用命令行操作,所以要把Zabbix的Web页面上线出来,就是提前写好的一个web页面,页面也是在安装包解析的文件里**

**[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/frontends/php/**

**[root@zabbixserver php]# cp -r \* /usr/local/nginx/html/--------------->拷贝文件**

**[root@zabbixserver php]# chmod -R 777 /usr/local/nginx/html/\*--------->给予权限**

**4.修改配置文件,设置数据库相关参数启动Zabbix\_server服务**

**[root@zabbixserver ~]# vim /usr/local/etc/zabbix\_server.conf**

**DBHost=localhost------------------------------------>数据库主机，默认该行被注释**

**DBName=zabbix-------------------------------------------------->设置数据库名称**

**DBUser=zabbix-------------------------------------------------->设置数据库账户**

**DBPassword=zabbix------------------------------->设置数据库密码，默认该行被注释**

**LogFile=/tmp/zabbix\_server.log------------------------>设置日志，仅查看以下即可**

**[root@zabbixserver ~]# useradd -s /sbin/nologin zabbix**

**一定要创建用户,如果不创建用户就无法启动服务**

**[root@zabbixserver ~]# zabbix\_server---------------------------------->启动服务**

**●查询端口号,查看服务是否启动**

**[root@zabbixserver ~]# ss -ntulp |grep zabbix\_server 确认连接状态，端口10051**

**tcp LISTEN 0 128 \*:10051 \*:\* users:(("zabbix\_server",pid=23275,fd=4),("zabbix\_server",pid=23274,fd=4)**

**起一遍服务,端口开启一个,开启一堆进程,并发量就越大,说明监控的机器就越多**

**如果是因为配置文件不对，导致服务无法启动时,不要重复执行zabbix\_server，因为虽然查不到端口,但是进程其实已经启动了,所以一定要先使用killall zabbix\_server关闭服务后，再重新启动一次。**

**●修改Zabbix\_agent配置文件，启Zabbix\_agent服务(被监控端,有多少台,就做多少次)**

**[root@zabbixserver ~]# vim /usr/local/etc/zabbix\_agentd.conf**

**Server=127.0.0.1,192.168.2.5---------------->允许谁来监控我,要添监控服务器的IP**

**ServerActive=127.0.0.1,192.168.2.5----------->允许哪些主机通过主动模式监控本机**

**Hostname=zabbix\_server------------------------>设置本机主机名,最好跟本机名一样**

**LogFile=/tmp/zabbix\_server.log----------------------------------->设置日志文件**

**UnsafeUserParameters=1----------->是否允许自定义key(是被注释掉的,要将注释去掉)**

**如果等于零,就说明只能用zabbix自己的监控选项,不能使用自定义的,一般企业都会改为1**

**[root@zabbixserver ~]# zabbix\_agentd----------------------------->启动监控agent**

**[root@zabbixserver ~]# ss -ntulp |grep zabbix\_agentd-------------->查看端口信息**

**tcp LISTEN 0 128 \*:10050 \*:\* users:(("zabbix\_agentd",pid=23505,fd=4),("zabbix\_agentd",pid=23504,fd=4))**

**如果是因为配置文件不对，导致服务无法启动时，不要重复执行zabbix\_agentd，一定要先使用killall zabbix\_agentd关闭服务后，再重新启动一次。浏览器访问Zabbix\_server服务器的Web页面**

**5.真机浏览器访问**

**在前面已经部署了LNMP环境 启动了zabbix\_sever zabbix\_agent 设置了web页面**

**[root@zabbixserver ~]# firefox <http://192.168.2.5/index.php>**

****

**第一次访问，初始化PHP页面会检查计算机环境是否满足要求，如果不满足会给出修改建议.默认会提示PHP的配置不满足环境要求，需要修改PHP配置文件**

**(1)页面出现之后点击下一步,因为之前没有修改过配置文件,所以在界面上会出现一些报红的错误信息,所以要更改配置文/etc/php.ini,在此文件中;是起到注释的作用,去掉注释,就将;去掉就可以了**

**按照出错的内容来修改即可.修改之后要重启一下php**

**date.timezone = Asia/Shanghai--------------------------->设置时区 中国时区是上海**

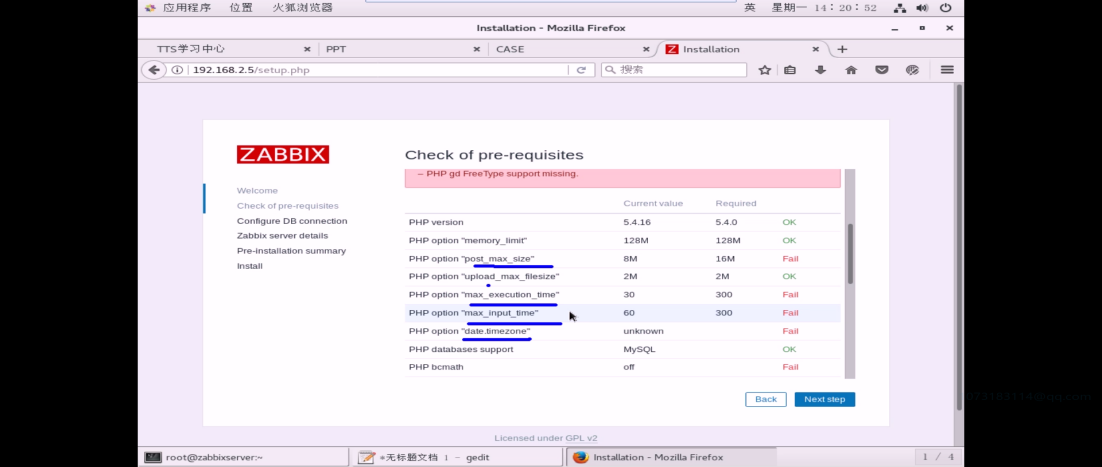
**max\_execution\_time = 300--------------------------------------->最大执行时间，秒**

**post\_max\_size = 32M-------------------------------------------->POST数据最大容量**

**max\_input\_time = 300----------------------------------->服务器接收数据的时间限制**

**memory\_limit = 128M------------------------------------------------>内存容量限制**

**[root@zabbixserver ~]# systemctl restart php-fpm**



**PHP opton:PHP的参数选项**

1. **重启之后,回到页面刷新一下,报红的信息会变少,但是还会有报错,继续向下看,可能会是缺失软件依赖包**

**[root@zabbixserver ~]# yum -y install php-gd php-xml**

**--------->绘图,做图片实别用的**

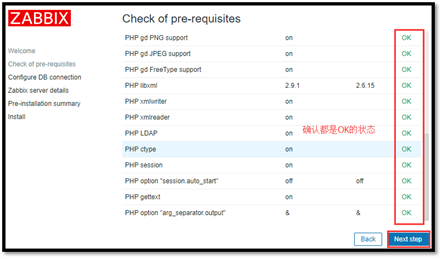
**[root@zabbixserver ~]# yum install php-bcmath-5.4.16-42.el7.x86\_64.rpm**

**---------------->做计算用的**

**[root@zabbixserver ~]# yum install php-mbstring-5.4.16-42.el7.x86\_64.rpm**

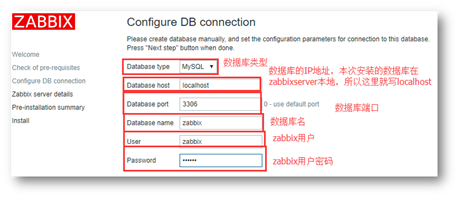
**----------------->支持中文用的**

**[root@zabbixserver ~]# systemctl restart php-fpm**

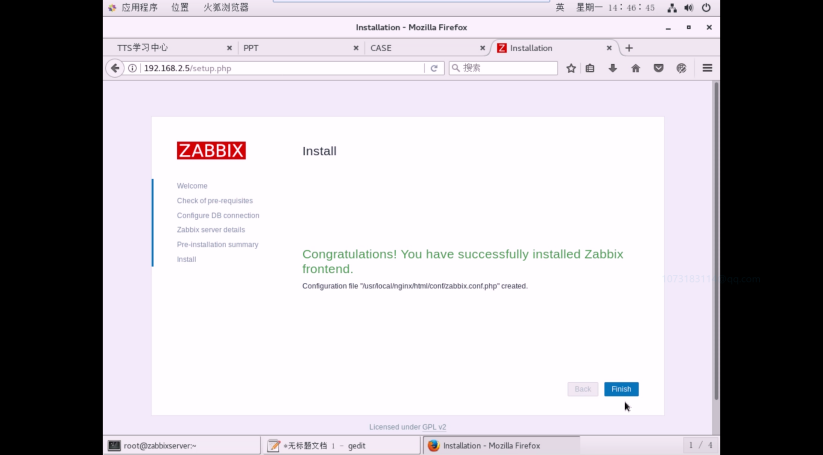
****

**可能刷新之后会有一个PHP LDAP是warning,LDAP意思是存放所有账号的一个服务,有没有无所谓,因为zabbix默认把帐号存放在mysql里.除此之外,所有的都应该是OK的**

**唯一要更改的就是端口号(mysql)和密码(之前配置文件中修改的)**

****

**出现这个界面,为正确,点击结束就可以了**



**如果点击finish之后,如果没有出现登录界面,出现一个提示说没有权限写文件,就说明之前在拷贝web页面的时候,没有给权限;第一次登录不仅会检查环境,而且在你填写配置的时候,会自动生成一个文件zabbix.conf.php文件,此文件会回传到/usr/local/nginx/html/\*如果权限不够,文件传不回来就会出错.**

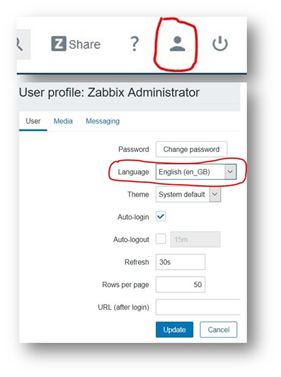
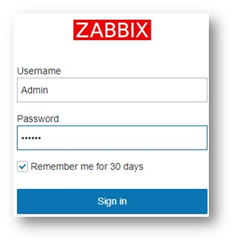
**You can install it manually可以手动去做**

**Download the configuration file-->点击下载,一般会下载到家目录下的下载**

**Save it as /usr/local/nginx/html/conf/zabbix.conf.php**

**将下载的文件存放到路径/usr/local/nginx/html/conf/**

**并且改名为zabbix.conf.php就可以了**

****

**如果出现登录界面,使用默认的用户(admin)和密码(zabbix)登陆,登陆后设置语言环境可以设置为中文,但是在企业里基本都是英文的,因为中文有些地方会出现乱码.**

1. **部署被监控主机Zabbix Agent**

**1源码安装Zabbix agent软件**

**[root@zabbixclient\_web1 ~]# useradd -s /sbin/nologin zabbix---------->建立用户**

**[root@zabbixclient\_web1 ~]# yum -y install gcc pcre-devel----------->安装依赖包**

**[root@zabbixclient\_web1 ~]# tar -xf zabbix-3.4.4.tar.gz**

**[root@zabbixclient\_web1 ~]# cd zabbix-3.4.4/**

**[root@zabbixclient\_web1 zabbix-3.4.4]# ./configure --enable-agent**

**--------------->做为被监控端只安装这一个模块就可以了**

**[root@zabbixclient\_web1 zabbix-3.4.4]# make && make install**

**2修改agent配置文件，启动Agent**

**[root@zabbixclient\_web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf**

**Server=127.0.0.1,192.168.2.5--------------------->谁可以监控本机（被动监控模式）**

**ServerActive=127.0.0.1,192.168.2.5--------------->谁可以监控本机（主动监控模式）**

**Hostname=zabbix\_client\_zabbixclient\_web1----------------->被监控端自己的主机名**

**EnableRemoteCommands=1**

**-------->监控异常后，是否允许服务器远程过来执行命令，如重启某个服务**

**UnsafeUserParameters=1---------------------------------->是否允许自定义key监控**

**[root@zabbixclient\_web1 ~]# zabbix\_agentd----------------------->启动agent服务**

**为了方便以后的管理,可以使用脚本开关服务,如果没有这个脚本,就是要将整个文件杀死才可以停掉服务**

**[root@zabbixclient\_web1 zabbix-3.4.4]# cd misc/init.d/fedora/core**

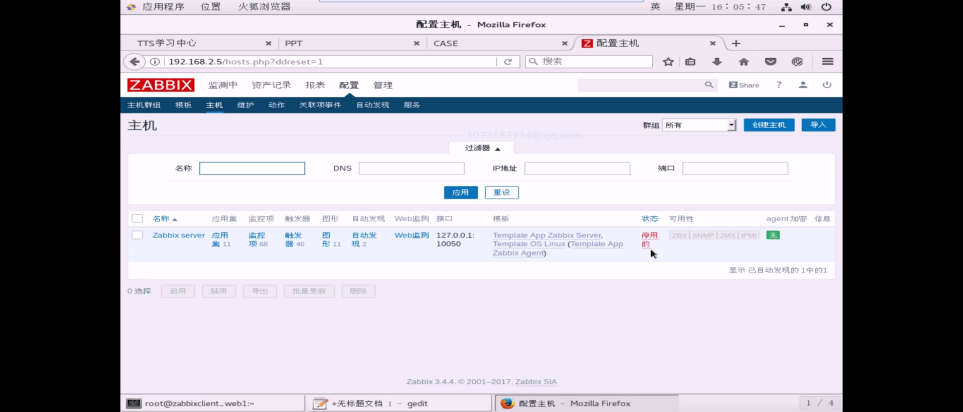
**[root@zabbixclient\_web1 zabbix-3.4.4]# cp zabbix\_agentd /etc/init.d/**

**[root@zabbixclient\_web1 zabbix-3.4.4]# /etc/init.d/zabbix\_agentd start**

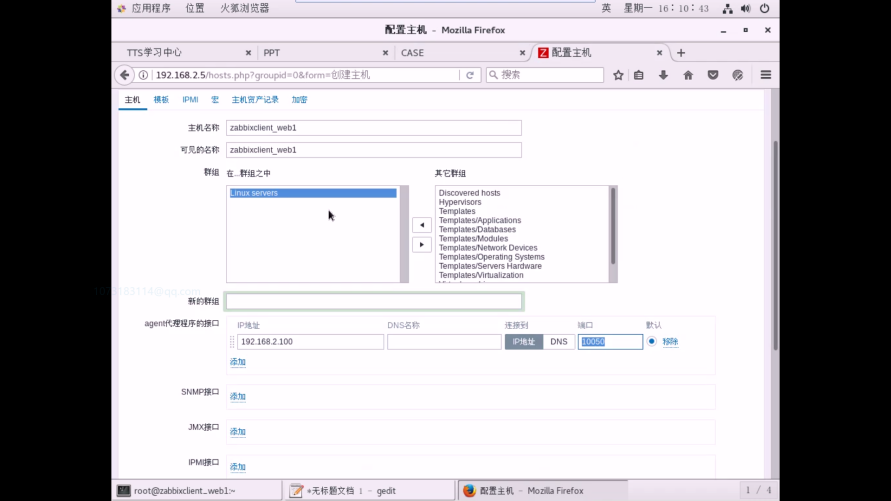
1. **配置及使用Zabbix监控系统**
2. **添加监控主机**

**使用浏览器登录http://192.168.2.5，通过Configuration（配置）-->Hosts（主机）-->Create Host（创建主机）点击停用的,启用主机.添加被监控Linux主机**

****



**添加被监控主机时，需要根据提示输入被监控Linux主机的主机名称（最好与电脑的主机名一致，但也允许不一致）、主机组、IP地址等参数，**



**端口是不用动的,因为启动agent,就会有10050端口启用**

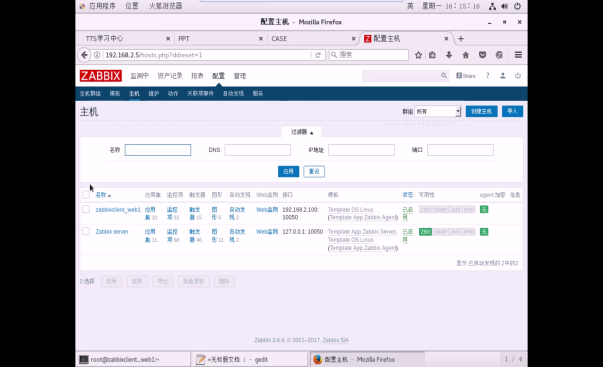
**添加信息之后,不要点击添加,点击主机旁边的模板**

1. **为被监控主机添加监控模板**

****

**点击选择,勾选想要的监控项.然后点击添加,最后点击最下面的添加,点击配置,点击主机,就可以看到所监控的主机以及监控项数量**

**Template OS Linux包含了像CPU 内存 时间 SSH FTP等等**

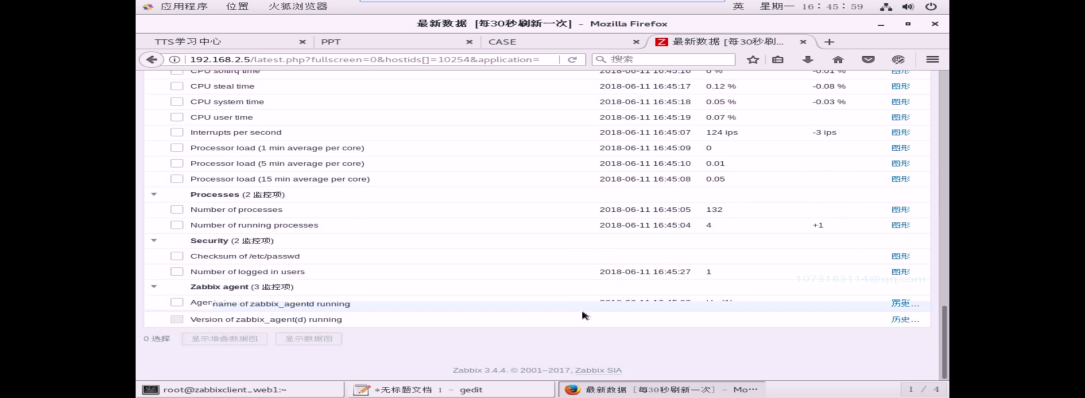


1. **查看监控数据**

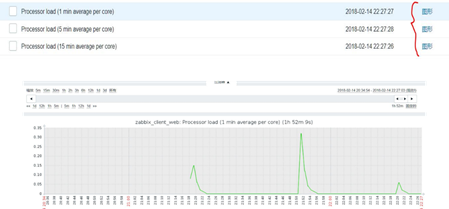
**查看监控数据，登录Zabbix Web控制台，点击Monitoring(监控中)—> Latest data(最新数据)，正过滤器中填写过滤条件，根据监控组和监控主机选择需要查看哪些监控数据**

****

**下面会显示所有监控的数据(是以数字的形式)**



找到需要监控的数据后，可以点击后面的Graph查看监控图形



1. **自定义Zabbix监控项目**

**使用Zabbix自定义key操作步骤:**

**创建自定义key**

**创建监控项目**

**创建监控图形**

**将监控模板关联到主机**

**(1)被监控主机创建自定义key(要监控哪台机器就在哪台上做)**

**●自定义的key文件一般存储在/usr/local/etc/zabbix\_agentd.conf.d/目录，这里还需要修改zabbix\_agentd.conf文件，允许自定义监控key，来读取该目录下的所有文件 。**

**[root@zabbixclient\_web1 ~]# vim /etc/zabbix/zabbix\_agentd.conf**

**UnsafeUserParameters=1。----------->等于1表示允许自定义,等于0表示不允许自定义**

**Include=/usr/local/etc/zabbix\_agentd.conf.d/----------------->监控脚本所在路径**

**大约在264行,解除注释**

**[root@zabbixclient\_web1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/**

**用vim 新建一个文件,名字可以随便起,但是最好起一个不用看文件内容,就知道是干什么的名字**

**[root@zabbixclient\_web1 zabbix\_agentd.conf.d]# vim count.line.passwd**

**UserParameter=count.line.passwd,wc -l /etc/passwd | awk ' {print $1} '**

**自定义key语法格式为：UserParameter=自定义key名称,命令**

**●测试自定义key是否正常工作**

**[root@zabbixclient\_web1 ~]# killall zabbix\_agentd**

**[root@zabbixclient\_web1 ~]# zabbix\_agentd----------------------->重启agent服务**

**因为之前配置文件中,已经设置好谁可以监控自己,所以可以用自己或者是监控主机来测试一下**

**[root@zabbixclient\_web1 ~]# zabbix\_get -s 127.0.0.1 -k count.line.passwd**

**[root@zabbixserver ~]# zabbix\_get -s 192.168.2.5 -k count.line.passwd**

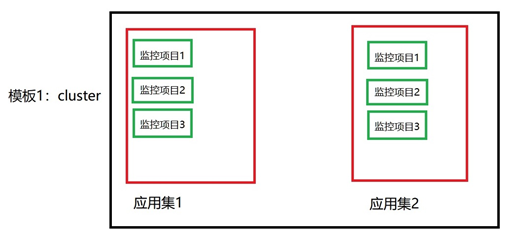
**如zabbix\_get命令执行错误，提示Check access restrictions in Zabbix agent configuration，则需要检查agent配置文件是否正确：**

**[root@zabbixclient\_web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf**

**Server=127.0.0.1,192.168.2.5**

**ServerActive=127.0.0.1,192.168.2.5**

1. **创建监控模板**

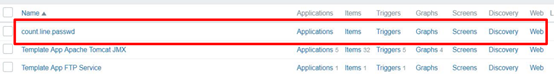
****

**●添加监控模板**

**登录Zabbix Web监控控制台，通过Configuration(配置)-->Template(模板)-->Create template(创建模板)，填写模板名称，新建模板群组**

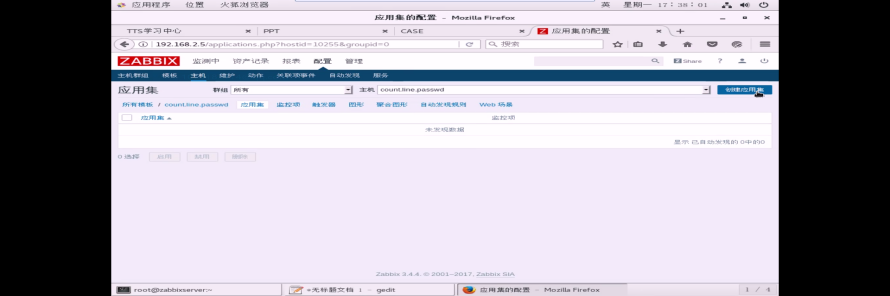
****

**创建模板后，默认模板中没有任何应用、项目、触发器、图形等**

****

**●创建应用**

**点击模板后面的Application（应用集）链接打开创建应用的页面**

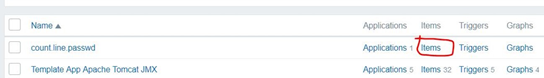




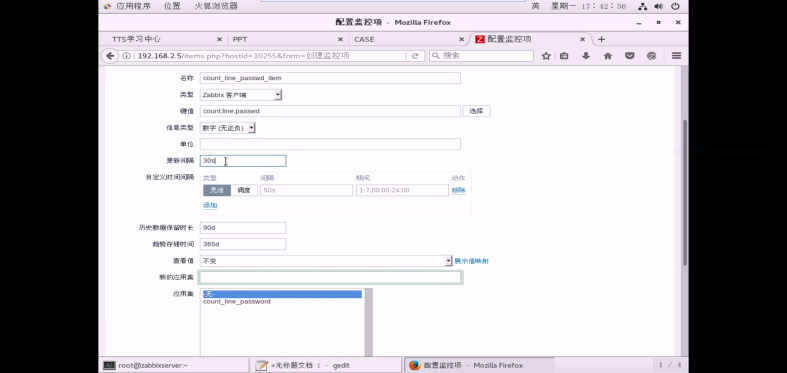
**模板和应用集都是空的概念性的东西,所要监控的东西才是实的**

**●创建监控项目**

**与创建应用一样，在模板中还需要创建监控项目,并在刷新出的新页面中选择Create items（创建监控项）创建项目**

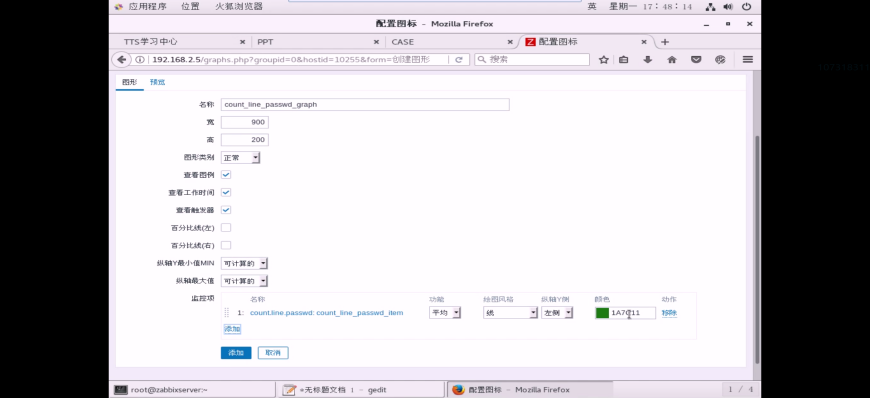
****

**点击之后,选择创建监控项,名字可以随便起,但是key一定要跟之前脚本所写的一致**



**●创建图形**

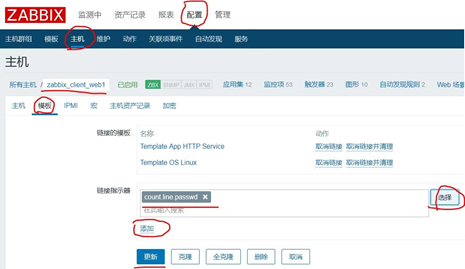
**为了后期可以通过图形的方式展示监控数据，还需要在模板中创建图形，设置方法与前面的步骤一致，在监控模板后面点击Graph（图形）即可创建图形，设置监控图形基于什么监控数据**



**●将模板链接到被监控主机**

**将完整的监控模板制作完成后，就可以将模板链接到主机实现监控功能了。首先找到被监控主机Configuration（配置）-->Hosts（主机）**

**点击配置,选择主机,点击web1 选择模板 点击选择 选中刚才创建的模板 点击添加 点击更新**

****

**●查看监控数据图形**

**点击Monitoring（监控中）-->Craphs（图形），根据需要选择条件，查看监控图形**

****

**验证的时候可以在被监控的主机执行脚本,然后在web页面查询.zabbix查询必须要熟悉脚本.因为zabbix监控是需要自定义监控,而自定义监控最主要的就是脚本.**

1. **zabbix报警机制**

**(一)概念介绍**

**1.自定义的监控项默认不会自动报警,首页也不会提示错误,需要配置触发器与报警动作才可以自定报警**

**2.触发器:当触发条件发生后,会导致一个触发事件,触发事件会执行某个动作**

**3.动作:触发器的条件被触发后的行为**

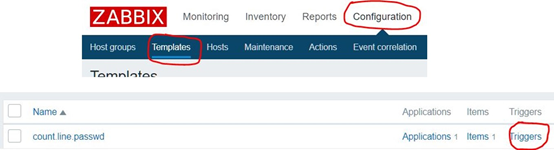
**4.一定要先有触发器,然后才有动作,定义什么动作,就执行什么动作.两者之间是分开做的,其优点是后期非常简单,动作不需要重复定义很多次.两边都做好之后可以随意组合.如果两者在一起去做,一个触发器只能对应一个动作.而动作也只能对应一个触发器.**

**(二)具体步骤**

**1.创建触发器规则**

**1）创建触发器**

**创建触发器时强烈建议使用英文的语言环境，因为触发器是有一些函数和表达式的,如果选择中文,有一些是找不到的.通过Configuration(配置)--> Templates(模板)，点击想要操作的模板**

****

**2）触发器表达式(可以手写,也可以自动写,推荐自动写)**

**创建触发器时需要定义表达式，触发器表达式（Expression）是触发异常的条件，触发器表达式格式如下：**

**{<server>:<key>.<function>(<parameter>)}<operator><constant>**

**{服务器主机名:key(哪个监控数据).函数(参数,zabbix内置函数)}<表达式>常数(自己心里的预期值)**

**案例:**

**{web1:system.cpu.load[all,avg1].last(0)}>5 //0为最新数据**

**如果web1主机最新的CPU平均负载值大于5，则触发器状态Problem**

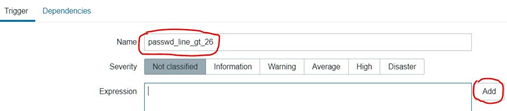
**{vfs.fs.size[/,free].max(5m)}<10G //5m为最近5分钟**

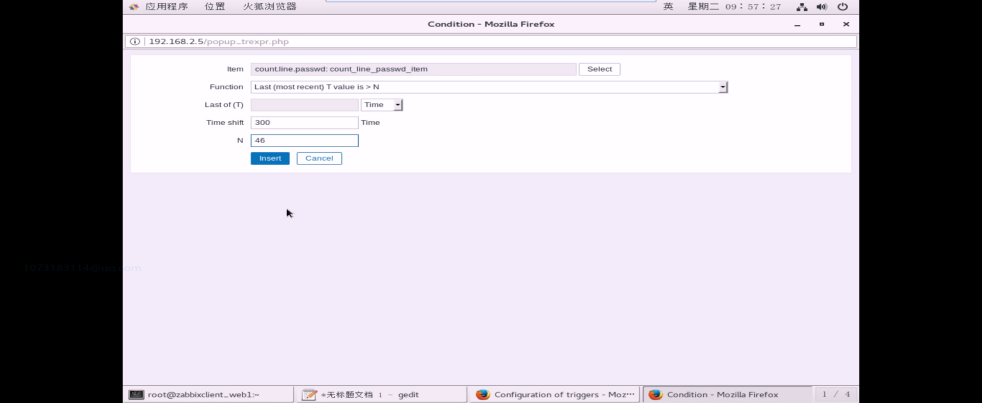
**根分区，最近5分钟的最大容量小于10G，则状态进入Problem**

**{vfs.file.cksum[/etc/passwd].diff(0)}>0 //0为最新数据**

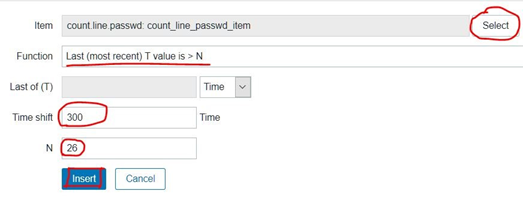
**最新一次校验/etc/passwd如果与上一次有变化，则状态进入Problem**

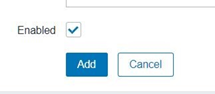
1. **配置触发器**

****



**要选择报警级别:一般信息的报警 警告报警 严重性报警 高的报警 灾难性报警**

****

****

**触发器最难的就是表达式,这些表达式都是需要选择,不需要人来做**

1. **设置邮件**

**1）创建Media**

**通过Administration（管理）-->Media Type（报警媒体类型）-->选择Email（邮件）**

**在触发动作之前,要确定报警的方式,要提前配置好**

****

**Zabbix有三种报警方式:**

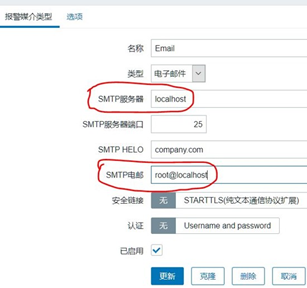
**Email电子邮件方式**

**Jabber:是国外用的,类似于中国的微信,在中国不能用,如果想用可以翻墙**

**SMS:短信,但是必须买一个短信网关,需要一个设备,一调多少钱,会收费的**

**设置邮件服务器信息，设置邮件服务器及邮件账户信息**

**电脑里有一个默认打开的服务:systemctl ststus posstfix.service,只要有这个服务默认开启,就会自动发送邮件,它监听的端口是用来发送邮件的**

****

**如果上面添加的不是本机,添加的是别的邮箱,需要用户名密码,就点击认证,填写用户名和密**

1. **为用户添加Media码**

**在Administration（管理）-->Users（用户）中找到选择admin账户**

****

**点击Admin账户后，在弹出的界面中选择Media（报警媒介）菜单-->点击Add(添加)报警媒介**

****

****

1. **创建Action动作**

**1）Action动作**

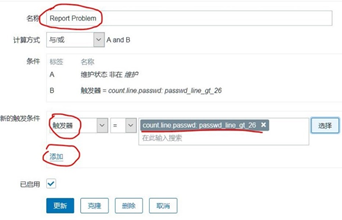
**Action（动作）是定义当触发器被触发时的时候，执行什么行为。**

**通过Configuration（配置）-->Actions（动作）-->Create action（创建动作）**

****

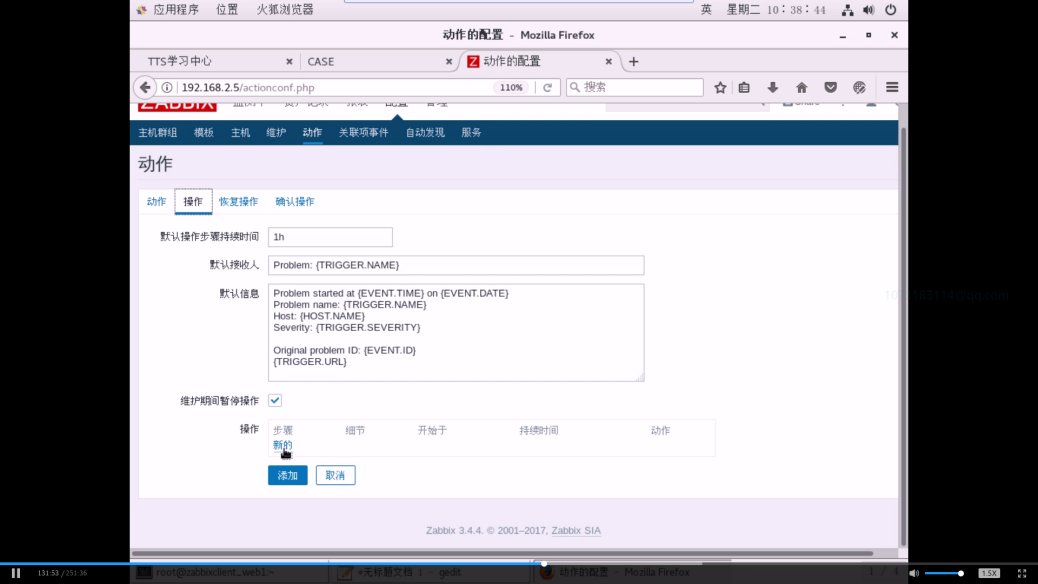
**2）配置Action动作的触发条件**

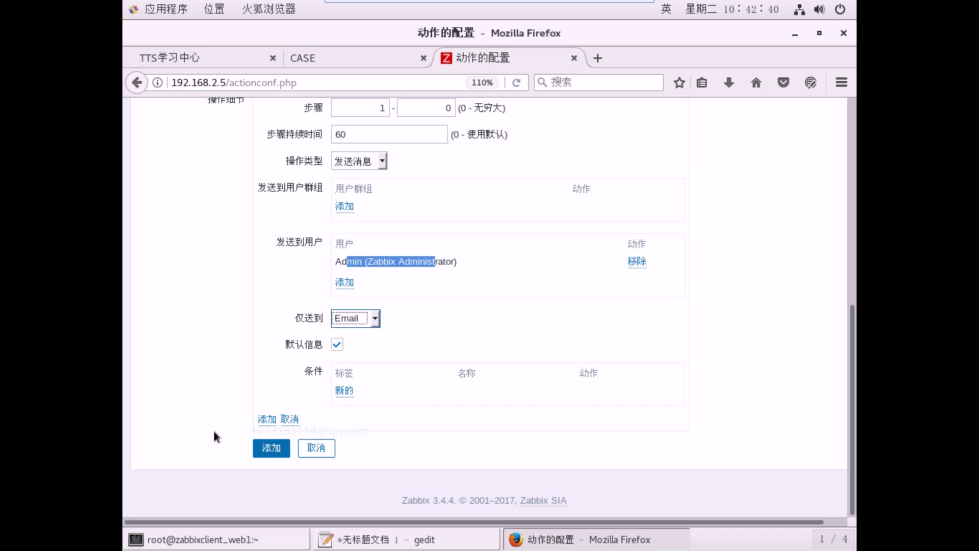
**填写Action动作的名称，配置什么触发器被触发时会执行本Action动作**

****

**3）配置Action动作的具体行为**

**配置动作的具体操作行为（发送信息或执行远程命令），无限次数发送邮件，60秒1次，发送给Admin用户**





**4）测试效果**

**在被监控主机创建账户（让账户数量大于26），然后登录监控端Web页面，在仪表盘中查看问题报警（需要等待一段时间）**

****

## **Zabbix自动发现**

**●自动发现(Discovery)**

**当zabbix需要监控的设备越来越多,手动添加监控设备越来越有挑战,此时,可以考虑使用自动发现功能,需要批量一次性添加一组监控主机,也可以使用自动发现功能**

**●自动发现可以实现:**

**自动发现,添加主机,自动添加主机到组,自动连接模板到主机,自动创建监控项目与图形等**

**●自动发现（Discovery）流程：**

**创建自动发现规则**

**创建Action动作，说明发现主机后自动执行什么动作**

**通过动作，执行添加主机，链接模板到主机等操作**

**1.自动发现规则**

**1）创建自动发现规则**

**通过Configuration（配置）-->Discovery（自动发现）-->Create discovery rule（创建发现规则）**

****

**2）填写规则**

**填写自动发现的IP范围（逗号隔开可以写多个），多久做一次自动发现（默认为1小时，仅实验修改为1m）**

** **

**自动发现的第一个条件是网段,第二个条件是有没有agent,如果有,就自动的加进来,**

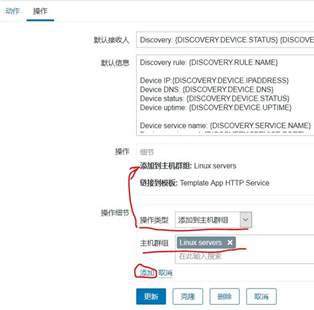
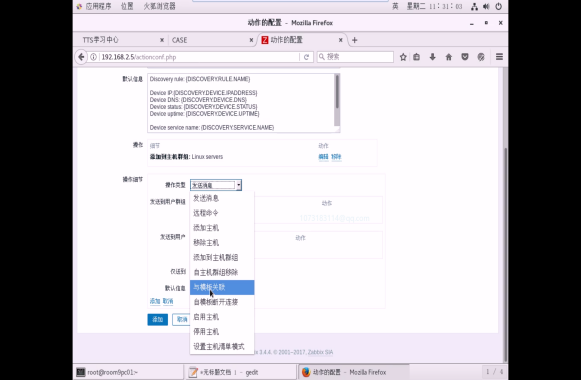
**如果有机器确实没有装agent,但是就是想监控它的web页面,点击检查新的,选择检查类型和端口,点击添加**

1. **创建动作**
2. **创建Action动作 通过Configuration（配置）--> Actions Event source(事件源)：自动发现(Discovery)-->Create action（创建动作）**

****2**）配置Action动作具体行为,配置动作，添加动作名称，添加触发动作的条件**

****

**点击操作（触发动作后要执行的操作指令），操作细节：添加主机到组，与模板链接（HTTP模板）**

**** 

## **Zabbix主动监控**

**●默认zabbix采用的是被动监控，主动和被动都是对被监控端主机而言的！**

**●被动监控：Server向Agent发起连接，发送监控key，Agent接受请求，响应监控数据。**

**●主动监控：Agent向Server发起连接，Agent请求需要检测的监控项目列表，Server响应Agent发送一个items列表，Agent确认收到监控列表，TCP连接完成，会话关闭，Agent开始周期性地收集数据。**

**●区别：Server不用每次需要数据都连接Agent，Agent会自己收集数据并处理数据，Server仅需要保存数据即可。**

**●当监控主机达到一定量级后，Zabbix服务器会越来越慢，此时，可以考虑使用主动监控，释放服务器的压力。**

**●另外，Zabbix也支持分布式监控，也是可以考虑的方案。**

1. **.主要步骤**
2. **添加被监控主机**
3. **为被监控主机安装部署zabbix agent**

**[root@zabbixclient\_web2 ~]# yum -y install gcc pcre-devel**

**[root@zabbixclient\_web2 ~]# tar -xf zabbix-3.4.4.tar.gz**

**[root@zabbixclient\_web2 ~]# cd zabbix-3.4.4/**

**[root@zabbixclient\_web2 ~]#./configure --enable-agent**

**[root@zabbixclient\_web2 ~]# make && make install**

1. **修改agent配置文件,将agent监控模式修改为主动模式**

**[root@zabbixclient\_web2 ~]# vim /usr/local/etc/zabbix\_agentd.conf Server=127.0.0.1,192.168.2.5-------------------------->注释该行，允许谁监控本机StartAgents=0------------------------------------------>被动监控时启动多个进程设置为0，则禁止被动监控，不启动zabbix\_agentd服务**

**ServerActive=192.168.2.5--->允许哪些主机监控本机（主动模式），一定要取消127.0.0.1**

**Hostname=zabbixclient\_web2-------------------->告诉监控服务器，是谁发的数据信**

**一定要和zabbix服务器配置的监控主机名称一致（后面设置）**

**RefreshActiveChecks=120------------------------------------->默认120秒检测一次UnsafeUserParameters=1------------------------------------------>允许自定义key**

**Include=/usr/local/etc/zabbix\_agentd.conf.d/**

**[root@zabbixclient\_web2 ~]#killall zabbix\_agentd---------------------->关闭服务[root@zabbixclient\_web2 ~]# zabbix\_agentd----------------------------->启动服务**

**验证:ps去查进程肯定是有的,ss去查端口号肯定是没有的**

**[root@zabbixclient\_web2 ~]# ps aux |grep zabbix**

**[root@zabbixclient\_web2 ~]# ss -ntulp | grep zabbix**

1. **创建主动监控的监控模板**

**1）克隆Zabbix自动的监控模板**

**为了方便，克隆系统自带模板（在此基础上就该更方便）。通过Configuration（配置）-->Templates（模板）-->选择Template OS Linux-->全克隆，克隆该模板，新建一个新的模板。新模板名称为：Template OS Linux ServerActive(随便起)**

****

**2）修改模板中的监控项目的监控模式**

**将模板中的所有监控项目全部修改为主动监控模式，通过Configuration（配置-->Templates（模板）-->选择新克隆的模板，点击后面的Items（监控项）-->点击全选选择所有监控项目，点击批量更新，将类型修改为：Zabbix Agent（Active主动模式）**

****

**3）禁用部分监控项目**

**批量修改监控项的监控模式后，并非所有监控项目都支持主动模式，批量修改后，会发现有几个没有修改主动模式成功，说明，这些监控项目不支持主动模式，关闭即可。**

**可以点击类型排序，方便操作，点击状态即可关闭。**

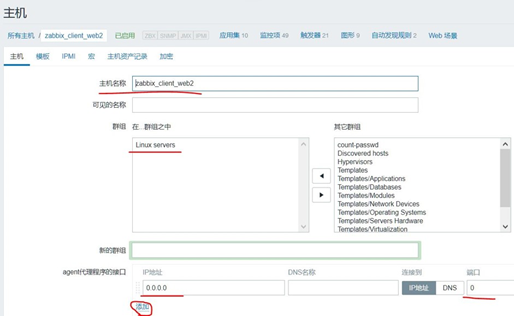
****

**3.添加监控主机**

**1）手动添加监控主机（主动模式监控）**

**在Zabbix监控服务器，添加被监控的主机（主动模式），设置主机名称：zabbix\_client\_zabbixclient\_web2 （必须与被监控端的配置文件Hostname一致），将机添加到Linux servers组，IP地址修改为0.0.0.0，端口设置为0**

****

****

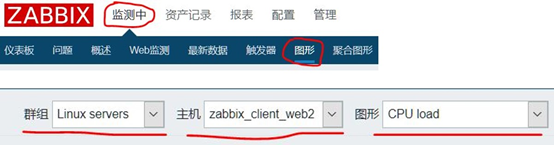
**主机名一定要和被监控端的配置文件中的一样**

**为主机添加监控模板，选择刚刚创建的模板（主动模式），添加链接模板到主机**

****

**2）验证监控效果**

**查看数据图表，通过Monitoring-->Graphs菜单，选择需要查看的主机组、主机以及图形，查看效果**

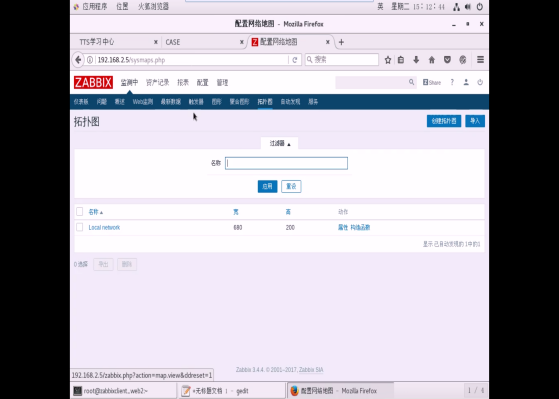
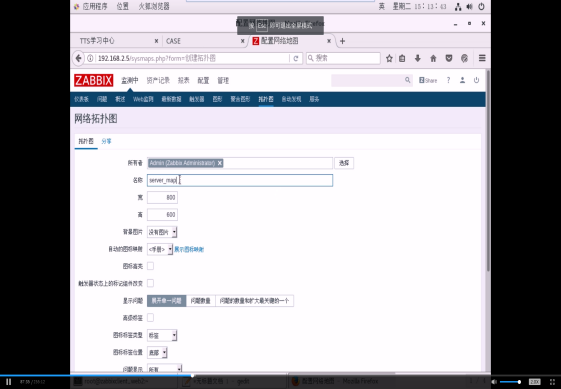
****

**七.拓扑图与聚合图形**

**(一)创建拓扑图**

**1.创建拓扑(方便多台机器的管理)**

**绘制拓扑图可以快速了解服务器架构，通过Monitoring（监控中）-->Maps（拓扑图），选择默认的Local network拓扑图，编辑即可（也可以新建一个拓扑图）点击添加即可.点击新的拓扑图进去,点击编辑拓扑图**

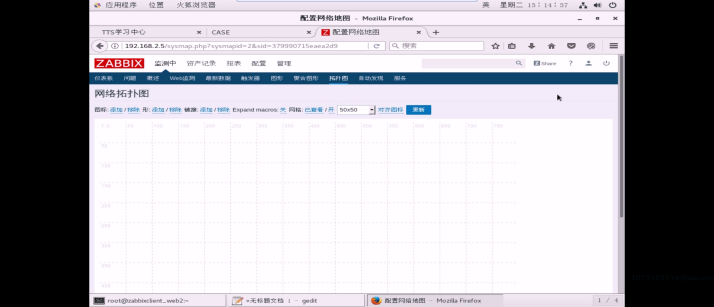
**2拓扑图图表说明**

**Icon（图标），添加新的设备后可以点击图标修改属性**

**Shape（形状）**

**Link（连线），先选择两个图标，再选择连线**

**完成后，点击Update（更新）**



这个拓扑图对公司补太熟悉的人很有用

1. **创建聚合图形**

**1）创建聚合图形**

**聚合图形可以在一个页面显示多个数据图表，方便了解多组数据。通过Monitoring（监控中）-->Screens（聚合图形）-->Create screen(创建聚合图形)即可创建聚合图形**

****

**修改聚合图形参数如下：**

**Owner：使用默认的Admin用户**

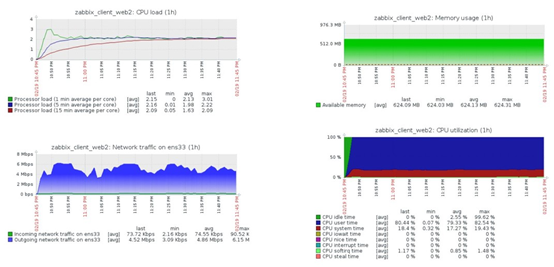
**Name：名称设置为Zabbixclient\_web2\_host**

**Columns：列数设置为2列**

**Rows：行数设置为4行**

**2）为聚合图形中添加监控图形**

**选择刚刚创建的聚合图形（zabbixclient\_web2\_host)，点击后面的构造函（constructor），点击Change(更改)，设置每行每列需要显示的数据图表**

****

1. **自定义监控案例**

**1.监控Nginx服务状态**

**1）准备环境，部署nginx软件**

**安装nginx软件，开启status模块**

**[root@zabbixclient\_web1 nginx-1.12.2]# ./configure - -help |grep status**

**[root@zabbixclient\_web1 nginx-1.12.2]# ./configure \**

**> --with-http\_stub\_status\_module**

**[root@zabbixclient\_web1 nginx-1.12.2]# make && make install**

**[root@zabbixclient\_web1 ~]# cat /usr/local/nginx/conf/nginx.conf**

**… …**

**location /status {**

**stub\_status on;**

**}**

**… …**

**[root@zabbixclient\_web1 ~]# curl [http://192.168.2.100/status](http://192.168.2.100/status-)**

**[-----------------------](http://192.168.2.100/status-)->想监控谁,就打谁的IP**

**Active connections: 1 ------------------------------------------>当前时是并发数**

**server accepts handled requests**

**10 10 3------------------------------------->建立多少个连接, 连接次数(历史页面)**

**Reading: 0 Writing: 1 Waiting: 0**

1. **自定义监控脚本**

**[root@zabbixclient\_web1 ~]# vim /usr/local/bin/nginx\_status.sh**

**#!/bin/bash**

**case $1 in**

**active)**

**curl -s http://192.168.2.100/status |awk '/Active/{print $NF}';;**

**waiting)**

**curl -s http://192.168.2.100/status |awk '/Waiting/{print $NF}';;**

**accepts)**

**curl -s http://192.168.2.100/status |awk 'NR==3{print $2}';;**

**esac**

**[root@zabbixclient\_web1 ~]# chmod +x /usr/local/bin/nginx\_status.sh**

**3)自定义监控key（仅供参考，未检测完整状态）**

**语法格式：**

**UserParameter=key,command**

**UserParameter=key[\*],<command>**

**key里的所有参数，都会传递给后面命令的位置变量**

**[root@zabbixclient\_web1 ~]# cd /usr/local/etc/zabbix\_agent.conf.d/**

**[root@zabbixclient\_web1 zabbix\_agent.conf.d]# vim**

**/usr/local/etc/zabbix\_agentd.conf.d/nginx.status**

**UserParameter=nginx.status[\*],/usr/local/bin/nginx\_status.sh $1**

**[root@zabbixclient\_web1 ~]# killall zabbix\_agentd**

**[root@zabbixclient\_web1 ~]# zabbix\_agentd**