监控与服务安全

监控的目的

----报告系统运行状况

每一部分必须同时监控

内容包括吞吐量,反应时间,使用率等

---提前发现问题

进行服务器性能调整前,知道调整什么

找出系统的瓶颈在什么地方

监控的资源类别

---公开数据

Web FTP SSH 数据库等应用服务

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 进程等运行信息

自动化监控系统

Cacti 基于SNMP协议的监控软件,强大的绘图能力

Nagios

基于Agent监控,强大的状态检查与报警机制

插件极多,自己写监控脚本潜入到Nagios非常方便

Zabbix 基于多种监控机制,支持分布式监控

Zabbix简介

-Zabbix是一个高度集成的监控解决方案

-可以实现企业级的开源分布式监控

-Zabbix通过C/S模式采集监控数据

-Zabbix通过B/S模式实现Web管理(B:浏览器)

监控拓扑

---监控服务器

监控服务器可以通过SNMP或Agent采集数据

数据可以写入MySQL Oracle等数据库中

服务器使用LNMP实现Web前端的管理

---被监控主机

被监控主机需要安装Agent

常见的网络设备一般支持SNMP

主配置文件 /usr/local/etc/zabbix\_server.conf

默认端口 10051

日志文件 /tmp/zabbix\_server.log

"/usr/local/nginx/html/conf/zabbix.conf.php" zabbix保存配置信息的文件

整体步骤

1部署Nginx (开启php模块 添加fastcgi参数)

2安装mariadb 创建zabbix用户 zabbix库 字符集为utf8 (安装zabbix时涉及到--with-mysql=/usr/bin/mysql\_config)

3安装zabbix 修改 server.conf 配置文件 (将frontends/php/下的文件全部拷贝到html目录下 导入/database/mysql/下的3张表 依次是schema images data )

4.安装php php-fpm php-mysql php-gd php-xml php-ldap php-bcmath php-mbstring 相关软件包,修改 /etc/php.ini相关参数

5.创建zabbix用户 开启服务

部署监控服务

1.安装LNMP环境(Zabbix监控管理控制台需要通过Web页面展示出来，并且还需要使用MySQL来存储数据，因此需要先为Zabbix准备基础LNMP环境。)

[root@zabbixserver ~]# yum -y install gcc pcre-devel openssl-devel

[root@zabbixserver ~]# tar -xf nginx-1.12.2.tar.gz

[root@zabbixserver ~]# cd nginx-1.12.2

[root@zabbixserver nginx-1.12.2]# ./configure --with-http\_ssl\_module

[root@zabbixserver nginx-1.12.2]# make && make install

[root@zabbixserver ~]# yum -y install php php-mysql mariadb mariadb-devel mariadb-server

2.修改Nginx配置文件(配置Nginx支持PHP动态网站，因为有大量PHP脚本需要执行，因此还需要开启Nginx的各种fastcgi缓存，加速PHP脚本的执行速度)

!!fastcgi缓存不能写在server中!!

vim /usr/local/nginx/conf/nginx.conf

… …

http{

… …

fastcgi\_buffers 8 16k; //缓存php生成的页面内容，8个16k

fastcgi\_buffer\_size 32k; //缓存php生产的头部信息

fastcgi\_connect\_timeout 300; //连接PHP的超时时间

fastcgi\_send\_timeout 300; //发送请求的超时时间

fastcgi\_read\_timeout 300; //读取请求的超时时间

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

3）启动服务(启动Nginx、PHP-FPM、MariaDB服务，关闭SELinux与防火墙。)

[root@zabbixserver ~]# systemctl start mariadb

[root@zabbixserver ~]# systemctl start php-fpm

[root@zabbixserver ~]# ln -s /usr/local/nginx/sbin/nginx /sbin/nginx

[root@zabbixserver ~]# nginx

[root@zabbixserver ~]# firewall-cmd --set-default-zone=trusted

[root@zabbixserver ~]# setenforce 0

4)部署监控服务器Zabbix Server(1）源码安装Zabbix Server 多数源码包都是需要依赖包的，zabbix也一样，源码编译前需要先安装相关依赖包。)

[root@zabbixserver lnmp\_soft]# yum -y install net-snmp-devel curl-devel //安装相关依赖包

[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-server安装部署zabbix服务器端软件

--enable-proxy // --enable-proxy安装部署zabbix代理相关软件

--enable-agent // --enable-agent安装部署zabbix被监控端软件

--with-mysql=/usr/bin/mysql\_config // --with-mysql配置mysql\_config路径

--with-net-snmp // --with-net-snmp允许zabbix通过snmp协议监控其他设备

--with-libcurl // --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

5)初始化Zabbix 创建数据库，上线Zabbix的Web页面

[root@zabbixserver ~]# mysql

mysql> create database zabbix character set utf8; //创建数据库，支持中文字符集(必须是支持中文字符集 否则导入不进去)

mysql> grant all on zabbix.\* to zabbix@'localhost' identified by 'zabbix'; //创建可以访问数据库的账户与密码

[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

//刚刚创建是空数据库，zabbix源码包目录下，有提前准备好的数据

//使用mysql导入这些数据即可（注意导入顺序）

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

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

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

//将web页面放入Nginx页面目录

6)安装PHP相关依赖 并设置相关配置

[root@zabbixserver ~]# cd /root/

[root@zabbixserver ~]# yum -y install php-gd php-xml php-ldap php-bcmath php-mbstring

[root@zabbixserver ~]# vim /etc/php.ini

date.timezone = Asia/Shanghai //设置时区

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 //服务器接受数据的时间限制

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

7)开启服务并初始化配置

[root@zabbixserver ~]#zabbix\_server

[root@zabbixserver ~]#firefox 192.168.2.5/index.php //web初始化配置

部署被监控主机Zabbix Agent

1)源码安装Zabbix agent软件

[root@zabbixclient\_web1 ~]# yum -y install gcc pcre-devel

[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 //只开启此功能即可

[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）

[root@zabbixclient\_web1 ~]# useradd -s /sbin/nologin zabbix

[root@zabbixclient\_web1 ~]# zabbix\_agentd //启动agent服务

配置及使用Zabbix监控系统(添加被监控的服务器)

1.web登录zabbix 使用管理员admin登录

2.通过Configuration（配置）-->Hosts（主机）-->Create Host（创建主机)

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

3.为被监控主机添加监控模板

Zabbix通过监控模板来对监控对象实施具体的监控功能，根据模板来定义需要监控哪些数据，对于Linux服务器的监控，Zabbix已经内置了相关的模板（Template OS Linux），选择模板并链接到主机即可

4.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

自定义Zabbix监控项目(zabbix支持自己写的监控脚本添加到模板中)

1）被监控主机创建自定义key

自定义key语法格式为：UserParameter=自定义key名称,命令

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

UnsafeUserParameters=1 //开启自定义功能

# Include=/usr/local/etc/zabbix\_agentd.userparams.conf

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

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

//加载配置目录 3选1

[root@zabbixclient\_web1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/ //进入对应目录

[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:UserParameter=自定义key名称,命令

////UserParameter=key[\*],<command> (一般用于脚本)key里的所有参数，都会传递给后面命令的位置变量

////UserParameter=ping[\*],echo $1 $1=[\*]

////严格区分大小写! U P必须大写

2）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

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

///若自定义key语法格式 UserParameter=key[\*],<command> 则 -k 'key[\*]' 'nginx.status[accepts]'

3)创建监控模板

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

创建完成模板后，默认模板中没有任何应用、项目、触发器、图形等资源。这里需要点击模板后面的Application（应用集）链接打开创建应用的页面

依次创建应用集 监控项

!!! 监控项 中的key值 与 被监控的上自定义创建的必须一致!!!

####################################################################################################

Zabbix报警机制

概念介绍

--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动报警

触发器

--表达式,如内存不足300M,用户超过30个等

--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

动作

--触发器的条件被触发后的行为

--可以是发送邮件,也可以是重启某个服务等

1）创建触发器

通过Configuration--> Templates，找到我们之前创建的count.line.passwd模板，点击模板后面的triggers

2）触发器表达式

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>

{主机：key.函数(参数)}<表达式>常数

在如图-2所示的蓝色方框中编写触发器表达式，可以直接手写，也可以通过add选择表达式模板。

{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

大多数函数使用秒作为参数，可以使用#来表示其他含义（具体参考表-1）。

avg, count, last, min and max 等函数支持额外的第二个参数time\_shift（时间偏移量），这个参数允许从过去一段时间内引用数据。

3）配置触发器

设置触发器名称 选择触发器报警级别 点击add添加表达式，填写表达式：监控项为账户数量，最近300秒账户数量大于26

4)设置邮件 创建Media

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

5)为用户添加Media

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

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

6)创建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）

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（账户数量大于26）

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(操作中步骤持续时间使用=60)，发送给Admin用户

概述

自动发现 :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

自动发现可以实现: 发现主机,添加主机,添加主机到组,连接模板等

自动发现流程

--创建自动发现规则

--创建Action动作(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

主动监控

主动和被动都是对 被监控端 主机而言的

默认zabbix采用的是被动监控

--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(简单的来说就是服务端主动与被监控段发送请求连接)

--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开始周期性的收集数据

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

优势:

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

主动模式 只有进程没有端口

添加主机时,名称必须是被监控配置文件中设置的hostname值

被监控端配置

[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，是谁发的数据信息

//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

web监控端配置

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

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）

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。

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：zabbixclient\_web2 （必须与被监控端的配置文件Hostname一致），将主机添加到Linux servers组，IP地址修改为0.0.0.0，端口设置为0

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