## Zabbix server

Zabbix server是整个Zabbix软件的核心程序。

Server通过轮询和捕获数据，计算是否满足触发器条件，向用户发送通知。它是Zabbix监控代理和Proxy代理报告系统可用性和完整性数据的核心组件。Server自身可以通过简单服务远程检查网络服务(如Web服务器和邮件服务)。

Sever是一个包含了被存储了所有配置，统计方面的和可操作数据的中央仓库,它是监控系统问题升级以至于激活警告管理器的Zabbix中的实体。

基本的Zabbix服务器起作用分三个不同的组件;他们是:Zabbix服务器，Web前端和数据库存储。

Zabbix的所有配置信息都存储在服务器和Web前端进行交互的数据库中。Zabbix的所有配置信息都存储在服务器和Web前端进行交互的数据库中。例如，当你通过Web前端（或者API）新增一个条目时，它会被添加到数据库的item表里。然后，Zabbix服务器以每分钟一次的频率查询item表中的活动列表，接着将它存储在Zabbix服务器中的缓存里。这就是为什么Zabbix前端所做的任何更改最多需要花费两分钟才能显示在最新的数据段的原因。

shell> ./zabbix\_server

你可以使用Zabbix server下的命令行参数：（运行控制台不支持OpenBSD和NetBSD系统）

-c --config <file> 配置文件的绝对路径 (默认路径： /etc/zabbix/zabbix\_server.conf)

-R --runtime-control <选项> 执行管理能力

-h --help 帮助提示

-V --version 显示版本号

命令行参数示例:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf

shell> zabbix\_server --help

shell> zabbix\_server -V

runtime control(运行时的控件)：

Option(选项) Description(说明) Target

config\_cache\_reload 重新加载配置缓存. 忽略当前已加载的缓存.

housekeeper\_execute 启动管家程序. 忽略当前正在进行中的管家程序.

log\_level\_increase[=<目标>] 增加日志的级别，如果没有指定目标则影响所有的进程.

log\_level\_decrease[=<目标>] 降低日志的基本，如果没有指定目标则影响所有的进程.

单一zabbix进程的日志基本改变后,进程的PIDs的值也会改变,允许的范围为1~65535.对大用户<进程类型，N＞目标选项可更改单个进程的日志级别

例如,使用runtime control重新加载server的配置缓存:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R config\_cache\_reload

例如,使用runtime control触发管家服务执行:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R housekeeper\_execute

例如,使用runtime control改变日志的级别:

增加所有进程的日志级别:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R log\_level\_increase

增加第二个poller进程的日志级别:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R log\_level\_increase=poller,2

增加PID为1234的进程日志级别:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R log\_level\_increase=1234

降低http poller进程的日志级别:

shell> zabbix\_server -c /usr/local/etc/zabbix\_server.conf -R log\_level\_decrease="http poller"

进程用户

Zabbix server定义了使用非root用户运行,启动后运行在非root用户的环境下,所以使用非root用户运行server是没有任何问题的.如果您想尝试root用户运行,它会切换到一个硬编码的用户,您可以参考 present,您需要修改配置文件中参数'AllowRoot'的值

如果Zabbix server和agent运行在同一台服务器上,建议您使用不同的用户运行server和agent. 否则, 如果两者都运行相同的用户, 代理可以访问服务器的配置文件, 任何Zabbix管理员级别的用户都可以很容易地检索server的信息, 例如, 数据库密码.

配置文件

请看 配置文件 有关Zabbix\_server的详细配置选项.

启动脚本

当系统启动/关机时启动脚本用来自动启动/停止Zabbix进程,脚本放在目录misc/init.d下.

## Zabbix agent

概况：Zabbix agent部署在监控的目标上，主动监测本地的资源和应用(硬件驱动，内存，处理器统计等)。

Zabbix agent收集本地的操作信息并将数据报告给Zabbix server用于进一步处理。一旦出现异常 (比如硬盘空间已满或者有崩溃的服务进程), Zabbix server会主动警告管理员指定机器上的异常。. Zabbix agents 的极端高效缘于它可以利用本地系统调用来完成统计数据的收集。

被动（passive）和主动（active）检查Zabbix agents可以执行被动和主动两种检查方式

passive check 模式中agent应答数据请求，Zabbix server（或者proxy）询问agent数据,如CPU 的负载情况，然后Zabbix agent回送结果。

Active checks 模式中处理过程将相对复杂。 Agent必须首先从Zabbix sever索取监控项列表以进行独立处理，然后周期性地发送新的值给server。

执行被动或主动检查是通过选择相应的监测项目类型来配置的。item type. Zabbix agent处理监控项类型有’Zabbix agent’和’Zabbix agent (active)’。

Zabbix agent在UNIX上以守护进程运行。Agent（独立的守护进程）可以通过执行代码运行：

shell> cd sbin

shell> ./zabbix\_agentd

Windows上的Zabbix agent是以Windows服务的形式运行的。 您可以在主机上运行Zabbix agent的单个实例或多个实例。 单个实例可以使用默认配置文件或命令行中指定的配置文件。 在多个实例的情况下，每个agent实例必须有自己独立的配置文件（其中一个实例可以使用默认配置文件）。

以下命令参数可以在Zabbix agent中使用：UNIX 和 Windows agent

参数 描述n

-c --config <config-file> 配置文件的绝对路径，您可以使用此选项来制定配置文件，而不是使用默认文件。\\在UNIX中, 默认文件是/usr/local/etc/zabbix\_agentd.conf 要么通过 compile-time 变量 --sysconfdir 或者 --prefix来设置，在Windows中，默认文件是 c:\zabbix\_agentd.conf

-p --print 显示已知监控项并推出

Note: 为了同时返回用户参数user parameter 您必须制定配置文件 (如果不是在指定位置的话).

-t --test <item key> 测试指定监控项并退出。

Note:为了同时返回用户参数 user parameter 您必须制定配置文件 (如果不是在指定位置的话).

-h --help 显示帮助信息

-V --version 显示版本号

仅UNIX agent

-R --runtime-control <option> 执行管理功能。参见 运行时控制. runtime control.

-m --multiple-agents 使用多agent实例 (使用 -i,-d,-s,-x )。仅Windows agent

T为了区分实例的服务名称，每项服务名都会包涵来自配置文件里的主机名值。

仅Windows agent (功能)

-i --install 以服务的形式安装Zabbix Windows agent

-d --uninstall 卸载Zabbix indows agent服务

-s --start 开始Zabbix Windows agent服务

-x --stop 停止bbix Windows agent 服务

有关在Windows上安装和运行Zabbix agent的细节，可以参看 更多细节 more details 。 使用命令行参数的特殊 例子

显示所有内置监控项和它们的值

使用指定的配置文件中定义的“mysql.ping”键来测试用户参数

在Windows下使用默认路径下的配置文件c:\zabbix\_agentd.conf安装一项 Zabbix agent服务

使用位于与agent可执行文件同一文件夹中的配置文件zabbix\_agentd.conf为Windows安装“Zabbix Agent [Hostname]”服务，并通过从配置文件中的唯一Hostname值扩来为命名。

shell> zabbix\_agentd –print

shell> zabbix\_agentd -t "mysql.ping" -c /etc/zabbix/zabbix\_agentd.conf

shell> zabbix\_agentd.exe -i

shell> zabbix\_agentd.exe -i -m -c zabbix\_agentd.conf

运行时控制

使用运行时控制选项，您可以更改agent进程的日志级别。

选项 描述 目标

log\_level\_increase[=<target>] 增加日志级别。如果没有特别指出，将会赋给所有进程。

log\_level\_decrease[=<target>] 降低日志级别。如果没有特别指出，将会赋给所有进程。

用于更改单独Zabbx agent的日志级别的PID可用范围为1到65535。对于具有较大PID的系统，可以使用<process type,N>目标选项来更改单独进程的日志级别。

Examples:

shell> zabbix\_agentd -R log\_level\_increase #给所有进程增加日志级别

shell> zabbix\_agentd -R log\_level\_increase=listener,2 #给第二监听进程增加日志级别

shell> zabbix\_agentd -R log\_level\_increase=1234 #给PID号为1234的进程增加日志级别

shell> zabbix\_agentd -R log\_level\_decrease="active checks" #给所有主动检查进程降低日志级别

运行时控制当前不支持OpenBSD，NetBSD和Windows。

进程用户

Zabbix agent是unix平台上设计在非root账户下的。它会以其他任何非root用户启动的进程一样的方式运行。所以，您可以使用任意非root用户运行agent，且不会产生任何问题。如果您在’root’账户下运行，它将切换到硬编码的“zabbix”用户，该用户必须存在于您的系统上。如果您只想以’root’方式运行proxy，您必须在proxy配额文件里修改‘AllowRoot‘参数。

配置文件

更多配置Zabbix agent的细节，请参阅配置文件选项。 zabbix\_agentd 或者Windows agent.

退出码

在2.2版本前，Zabbix agent 成功退出时退出码为0，异常时退出码为255。从2.2版本开始，Zabbix agent成功退出时退出码为0，异常则退出码为1。

## Zabbix\_proxy

Overview(概述)

Zabbix Proxy是一个可以从一个或多个受监控设备收集监控数据，并将信息发送到Zabbix sever的进程，基本上是代表sever工作的。 所有收集的数据都在本地进行缓存，然后传送到proxy所属的Zabbix sever。

部署Proxy是可选的，，但是可能非常有益于分散单个Zabbix sever的负载。 如果只有proxy收集数据，sever上的进程就会减少CPU消耗和磁盘I / O负载。

Zabbix proxy是完成远程区域、分支机构、没有本地管理员的网络的集中监控的理想解决方案。

Zabbix proxy需要使用独立的数据库。

Zabbix proxy数据库可以使用SQLite, MySQL, PostgreSQL. 使用Oracle或IBM DB2数据库时会有一定风险和限制（例如在低等级发现规则中的 返回值 ） return values of low-level discovery rules.

请参阅： 在分布环境中使用Proxy

Proxy 进程

Zabbix proxy以守护进程的方式运行。Proxy可以通过执行代码运行。

shell> cd sbin

shell> ./zabbix\_proxy

您可以使用Zabbix agent下的命令行参数:

-c --config <file> 配置文件的绝对路径

-R --runtime-control <option> 执行管理功能

-h --help 帮助

-V --version 显示版本号

运行时控制当前不支持OpenBSD和NetBSD。

命令行参数示例：

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf

shell> zabbix\_proxy --help

shell> zabbix\_proxy -V

运行时控制

运行时控制选项：

选项 描述 目标

config\_cache\_reload 重新加载配置缓存。若当前已被加载，则忽略。激活Zabbix proxy将会连接到Zabbix server并查询配置数据。

housekeeper\_execute 开始管理程序。 如果管理程序目前正在进行中，则被忽略.

log\_level\_increase[=<target>] 提升日志级别。 如果没有特别指出，将会赋给所有进程。

log\_level\_decrease[=<target>] 降低日志级别，如果没有特别指出，将会赋给所有进程。

用于更改单独Zabbx进程的日志级别的PID允许范围为1到65535。对于具有较大PID的系统，可以使用<process type,N>目标选项来更改单独进程的日志级别。

使用运行时控制来重新加载配置缓存的例子：

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R config\_cache\_reload

使用运行时控制来触发管家的执行的例子

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R housekeeper\_execute

使用运行时控制更改日志级别的示例：

Increase log level of all processes:

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R log\_level\_increase

Increase log level of second poller process:

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R log\_level\_increase=poller,2

Increase log level of process with PID 1234:

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R log\_level\_increase=1234

Decrease log level of all http poller processes:

shell> zabbix\_proxy -c /usr/local/etc/zabbix\_proxy.conf -R log\_level\_decrease="http poller"

进程用户

Zabbix proxy是设计在非root账户下的。它会以其他任何非root用户启动的进程一样的方式运行。 所以，您可以使用任意非root用户运行agent，且不会产生任何问题。如果您在’root’账户下运行，它将切换到硬编码的“zabbix”用户，该用户必须存在于您的系统上。如果您只想以’root’方式运行proxy，您必须在proxy配额文件里修改‘AllowRoot‘参数。

配置文件

更多Zabbix proxy配置选项细节，可参见 配置文件 .

支持的平台

Zabbix proxy和Zabbix sever一样，在相同的支持平台 下运行.

## Java Gateway

## Zabbix web

## Zabbix安装

### Zabbix server



### Zabbix database

[root@localhost nginx]# vim /etc/zabbix/zabbix\_server.conf #配置文件配置241数据库上.

DBHost=192.168.10.241

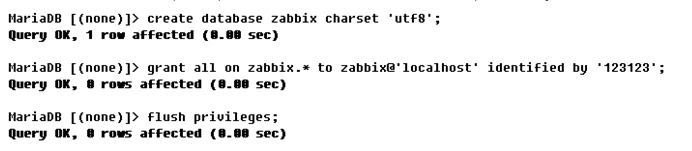
DBName=zabbix

DBUser=root

手动新建zabbix 库，解押安装包内的create.sql.gz包获得zabbix数据库初始化sql脚本，

/usr/share/doc/zabbix-server-mysql-3.4.15/create.sql.gz （数据库初始化脚本包所在路径）

创建库：



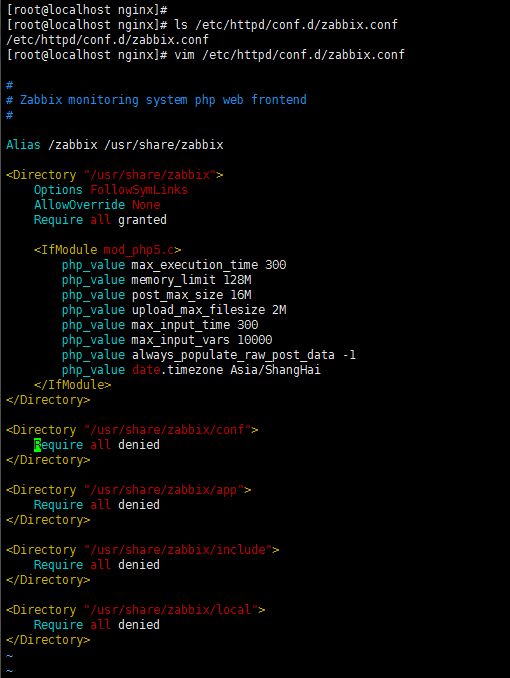
执行zabbix数据库初始化脚本



### Zabbix web

[root@localhost nginx]# ls /etc/httpd/conf.d/zabbix.conf

/etc/httpd/conf.d/zabbix.conf



### Zabbix agent

[root@db1 ~]# systemctl status zabbix-agent

[root@db1 ~]# rpm -ql zabbix-agent

/etc/logrotate.d/zabbix-agent

/etc/zabbix/zabbix\_agentd.conf #配置文件

[root@db1 ~]# vi /etc/zabbix/zabbix\_agentd.conf

############ GENERAL PARAMETERS ############

#一般配置日志等

##### Passive checks related #被动检查相关配置

StartAgents=3 #监听进程的子进程的数量，0为禁用了被动模式，导致两个模式都不能用改为3 重新启动成功

##### Active checks related #主动检查相关配置

## 可用性接口

ZBX：Agent接口

SNMP：协议（简单网络管理协议），

JMX：即java管理扩展，监控JVM虚拟机，

IPMI：我们可以通过标准的IPMI硬件接口，监控物理特征，比如：电压、温度、风扇状态、电源状态等。

## Windows安装zabbix agent

下载：

https://www.zabbix.com/download\_agents

下载的是zip包，解押到指定的C盘zabbix-agent目录，

配置目录中conf目录中的zabbix-agentd.win.conf(写字板或文本文档打开修改即可)

server 、ServerActive、Hostname

cmd中安装zabbix-agent服务

C:\zabbix-agent>C:\zabbix-agent\bin\zabbix\_agentd.exe --config c:\zabbix-agent\conf\zabbix\_agentd.conf --install

zabbix\_agentd.exe [10028]: service [Zabbix Agent] installed successfully

zabbix\_agentd.exe [10028]: event source [Zabbix Agent] installed successfully

启动

C:\zabbix-agent>C:\zabbix-agent\bin\zabbix\_agentd.exe --config c:\zabbix-agent\conf\zabbix\_agentd.conf --start

zabbix\_agentd.exe [5968]: service [Zabbix Agent] started successfully

查看zabbix进程

C:\zabbix-agent>tasklist |findstr zabbix

abbix\_agentd.exe 10180 Service

查看zabbix日志：默认在C盘zabbix\_agentd文件

web端添加主机、关联模板等

注意windows防火墙的影响

卸载

C:\zabbix-agent>C:\zabbix-agent\bin\zabbix\_agentd.exe -c c:\zabbix-agent\conf\za

bbix\_agentd.conf -d

zabbix\_agentd.exe [9592]: service [Zabbix Agent] uninstalled successfully

zabbix\_agentd.exe [9592]: event source [Zabbix Agent] uninstalled successfully

安装卸载二：

cmd /c "C:\zabbix\bin\win64\zabbix\_agentd.exe -c c:\zabbix\conf\zabbix\_agentd.win.conf -i"

cmd /c "C:\zabbix\bin\win64\zabbix\_agentd.exe -c c:\zabbix\conf\zabbix\_agentd.win.conf -s"

cmd /c "C:\zabbix\bin\win64\zabbix\_agentd.exe -c c:\zabbix\conf\zabbix\_agentd.win.conf -x"

cmd /c "C:\zabbix\bin\win64\zabbix\_agentd.exe -c c:\zabbix\conf\zabbix\_agentd.win.conf -d"

-c ：指定配置文件所有位置

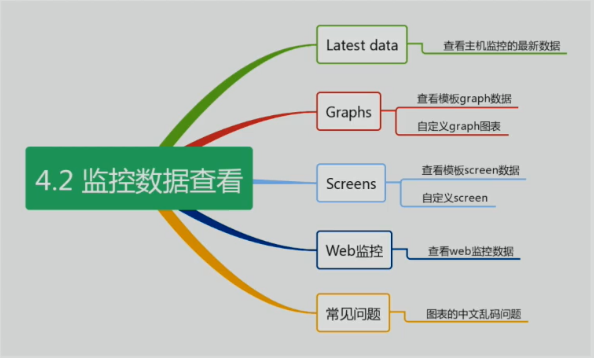
-i ：安装客户端

-s ：启动客户端

-x ：停止客户端

-d ：卸载客户端

Zabbix数据查看



触发器：

告警表达式函数：

diff：对比