**Nagios监控服务应用指南**

**版权声明：**

**本系列文档为《公司内部资料》内部教学教案，只允许个人使用，禁止私自传播谢谢大家！**

**联系方式：**

**本文作者：fish QQ 244647158** blog: http://blog.chinaunix.net/uid/23435165.html

==========================================

运维交流群：94776000 **（适合全体朋友）**

**文档信息：**

文档版本：Version 1.1

修改记录：2015-8-13

系统环境：CentOS 6.5（测试64bit）

**格式约定：**

灰色底黑色加粗：为输入的系统命令部分

灰色底：输入的系统命令返回的信息或者配置文件文本信息

绿色底：技巧或需要注意的注释信息

粉色底：需特别注意的地方

蓝色字体：内容注释

[nagios前言工具介绍及原理 7](#_Toc427326748)

[nagios监控工具介绍 7](#_Toc427326749)

[nagiso监控工具原理图 8](#_Toc427326750)

[第一部分 Nagios服务端安装部分 8](#_Toc427326751)

[1.1 nagios准备部分 9](#_Toc427326752)

[1.1.1准备3台服务器 9](#_Toc427326753)

[1.1.2设置yum安装源 9](#_Toc427326754)

[1.1.3解决perl编译问题 10](#_Toc427326755)

[1.1.4关闭nagios server iptables防火墙 11](#_Toc427326756)

[1.1.5解决系统时间同步 11](#_Toc427326757)

[1.2安装nagios所需基础软件包 12](#_Toc427326758)

[1.3创建nagios需要的用户及组 12](#_Toc427326759)

[1.4下载nagios和nagios相关组件 13](#_Toc427326760)

[1.5安装nagios 14](#_Toc427326761)

[1.6创建nagios web监控界面登入时需要的用户名（fish）及密码 16](#_Toc427326762)

[1.7添加监控报警的接受email地址 17](#_Toc427326763)

[1.8配置apache服务 18](#_Toc427326764)

[1.9安装nagios插件软件包 18](#_Toc427326765)

[1.10添加nagios自启动 19](#_Toc427326766)

[1.11验证nagios配置文件（检查语法） 19](#_Toc427326767)

[1.12安装nrpe 20](#_Toc427326768)

[第二部分 Nagios客户端安装 21](#_Toc427326769)

[2.1 nagios client端基础安装 21](#_Toc427326770)

[2.1.1设置yum安装源（同server端） 21](#_Toc427326771)

[2.1.2 解决perl编译问题[同server端] 22](#_Toc427326772)

[2.1.3关闭nagios server iptables防火墙 22](#_Toc427326773)

[2.1.4解决系统时间同步 22](#_Toc427326774)

[2.2在nagios客户端安装插件 23](#_Toc427326775)

[2.2.1安装nagios-plugins-1.4.3.tar.gz 23](#_Toc427326776)

[2.2.2安装nrpe 24](#_Toc427326777)

[2.2.3安装其他相关插件[iostat] 25](#_Toc427326778)

[2.3配置开发的几个基础脚本插件 27](#_Toc427326779)

[2.4配置nrpe 27](#_Toc427326780)

[第三部分 配置server 端nagios监控的服务 32](#_Toc427326781)

[3.1 nagios目录结构 33](#_Toc427326782)

[3.2配置主配置文件nagios.cfg 38](#_Toc427326783)

[3.3 hosts.cfg生产库实战配置例子语法 39](#_Toc427326784)

[3.3.1 hosts.cfg中主机定义部分配置参数详解 39](#_Toc427326785)

[3.3.2 host.cfg中主机定义部分批量模版化配置 40](#_Toc427326786)

[3.3.3千台服务器模版化繁为简的host部署配置 41](#_Toc427326787)

[3.4 services.cfg生产库实战配置例子语法 42](#_Toc427326788)

[3.4.1 services.cfg中服务定义部分配置参数详解 42](#_Toc427326789)

[3.4.2磁盘分区监控[被动监控] 43](#_Toc427326790)

[3.4.3 swap监控[ 被动监控] 44](#_Toc427326791)

[3.4.4内存监控[被动监控] 45](#_Toc427326792)

[3.4.5系统负载监控【被动监控】 46](#_Toc427326793)

[3.4.6 IO监控【被动监控】 47](#_Toc427326794)

[3.4.7 ping监控【被动监控】 48](#_Toc427326795)

[3.4.8千台服务器模版化繁为简的service部署配置 49](#_Toc427326796)

[3.4.9 URL监控 50](#_Toc427326797)

[3.4.9.1顶级域名blog.etiantian.org地址监控 50](#_Toc427326798)

[3.4.9.2域名下带目录文件http://blog.etiantian.org/jimo/exam.php地址监控 51](#_Toc427326799)

[3.4.10服务端口监控 52](#_Toc427326800)

[3.4.11多端口监控的写法 53](#_Toc427326801)

[3.5 commands.cfg生产库实战配置例子写法 53](#_Toc427326802)

[3.6 contacts.cfg生产库实战配置例子写法 55](#_Toc427326803)

[3.7 timeperiode.cfg 生产库实战例子写法 57](#_Toc427326804)

[3.8 templates.cfg生产库实战例子写法 59](#_Toc427326805)

[3.9 nagios 实战配置过程细节 60](#_Toc427326806)

[3.9.1监控客户端服务器磁盘分区、load负载、mem内存、swap、磁盘IO 【被动监控】 60](#_Toc427326807)

[3.9.1.1 nagios排错指南 68](#_Toc427326808)

[3.9.2增加从nagios服务器端发起的监控：如URL地址，端口监控【主动监控】 70](#_Toc427326809)

[3.9.2.1对域名url地址http://www.etiantian.org的监控 70](#_Toc427326810)

[3.9.2.2对域名URL地址http://www.etiantian.org/about/ 71](#_Toc427326811)

[3.9.2.3对特殊带传参的URL地址的监控 73](#_Toc427326812)

[3.9.2.4监控任意端口举例：80，110，25，8080，873 74](#_Toc427326813)

[3.10 Nagios的调试 76](#_Toc427326814)

[3.10.1优化配置nagios启动囧啊本检查nagios语法 76](#_Toc427326815)

[3.11 nagios安装部署过程中的错误集合及解决办法【被动监控】 77](#_Toc427326816)

[3.11.1 CHECK\_NRPE: Error - Could not complete SSL handshake. 77](#_Toc427326817)

[3.11.2 NRPE: Unable to read output 77](#_Toc427326818)

[3.11.3 NRPE: Command 'check\_load' not defined 78](#_Toc427326819)

[第四部分 nagios图形监控显示和管理【服务端】 80](#_Toc427326820)

[4.1 Pnp安装图形监控曲线（服务器端） 80](#_Toc427326821)

[4.1.1 yum安装pnp软件需要的基础包 80](#_Toc427326822)

[4.1.2 install libart\_lgpl by rrdtool 80](#_Toc427326823)

[4.1.3 install rrdtool 81](#_Toc427326824)

[4.1.4 install pnp-0.4.14.tar.gz 82](#_Toc427326825)

[4.1.5 nagios出图相关基本配置 83](#_Toc427326826)

[4.2配置各个服务出监控状态图 86](#_Toc427326827)

[4.2.1服务出图记录数据设置 86](#_Toc427326828)

[4.2.2主机出图记录设置 87](#_Toc427326829)

[4.3整合pnp url超链接到nagios图形显示界面里 87](#_Toc427326830)

[4.3.1给主机添加红太阳超链接图标 88](#_Toc427326831)

[4.3.2给指定的服务添加红太阳超链接图标 90](#_Toc427326832)

[第五部分 报警方式及生产报警策略 93](#_Toc427326833)

[5.1 报警的种类及报警方式选用 93](#_Toc427326834)

[5.1.1报警的种类 93](#_Toc427326835)

[5.1.2短信报警的种类 93](#_Toc427326836)

[5.1.3生产环境使用哪种报警？ 94](#_Toc427326837)

[5.1.4故障报警分级 95](#_Toc427326838)

[5.2实战报警实施细节 96](#_Toc427326839)

[5.2.1邮件报警 96](#_Toc427326840)

[5.2.1.1普通邮件报警 96](#_Toc427326841)

[5.2.1.2邮件报警适合的服务 96](#_Toc427326842)

[5.2.1.3邮件报警的基本配置方法 96](#_Toc427326843)

[5.2.1.4生产环境邮件报警配置 98](#_Toc427326844)

[5.2.1.5邮件转短信报警 99](#_Toc427326845)

[5.2.1.6联系人分级报警生产策略 100](#_Toc427326846)

[5.2.1.7正确的报警思想 102](#_Toc427326847)

[5.3飞信软件的报警 103](#_Toc427326848)

[5.3.1飞信软件报警配置过程细节 103](#_Toc427326849)

[5.3.1.1本地上传飞信包（也可以网上下载） 103](#_Toc427326850)

[5.2.1.4 MSN软件报警 103](#_Toc427326851)

[5.2.1.5 MSN软件报警配置过程细节 104](#_Toc427326852)

[5.2.1.6 http短信网关接口报警 104](#_Toc427326853)

[5.2.1.7短信网关设备报警细节（在监控服务器上） 105](#_Toc427326854)

[第六部分 nagios插件开发 108](#_Toc427326855)

[6.1什么是nagios插件 108](#_Toc427326856)

[6.2为什么要开发nagios插件 109](#_Toc427326857)

[6.3编写nagios插件规则 110](#_Toc427326858)

[6.3.1编写nagios规则说明 110](#_Toc427326859)

[6.3.2 nagios插件开发原理 111](#_Toc427326860)

[6.4 nagios插件开发语言 111](#_Toc427326861)

[6.5 使用shell开发nagios插件 112](#_Toc427326862)

[6.5.1编写检查weburl地址的插件 112](#_Toc427326863)

[6.5.1.1编写nagiois插件前的几个变量测试 112](#_Toc427326864)

[6.5.1.2监控weburl的插件脚本 113](#_Toc427326865)

[6.5.2 weburl插件脚本部署过程（主动监控方式） 115](#_Toc427326866)

[6.5.2.1在nagios服务器端的部署 115](#_Toc427326867)

[6.5.2.2修改nagios服务端的配置文件 115](#_Toc427326868)

[6.5.3编写检查mysql服务的插件 117](#_Toc427326869)

[6.5.3.1使用check\_mysql插件被动方式监控MYSQL服务的配置过程 117](#_Toc427326870)

[6.5.4使用chcek\_mysql插件主动方式监控MySQL服务的配置过程 120](#_Toc427326871)

[6.5.5手工开发脚本监控mysql服务的配置方法 122](#_Toc427326872)

[6.5.5.1手工开发脚本监控mysql服务 122](#_Toc427326873)

[第七部分 nagios分布式部署 127](#_Toc427326874)

[7.1 nagios角色划分 127](#_Toc427326875)

[7.2详细部署说明 129](#_Toc427326876)

[7.2.1安装软件说明 129](#_Toc427326877)

[7.2.2安装nagios被监控服务器 129](#_Toc427326878)

[7.2.3安装nagios分布式服务器 132](#_Toc427326879)

[7.2.4安装nagios中心监控服务器 145](#_Toc427326880)

# nagios前言工具介绍及原理

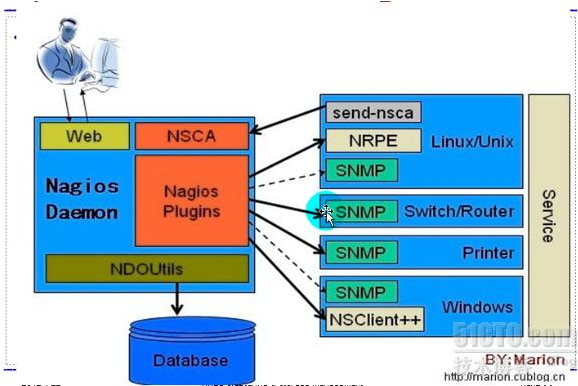
## nagios监控工具介绍

nagios是一款开源的免费网络及服务的监控工具，功能强大，灵活性强，能有效监控windows，linux和unix的主机各种状态信息，交换机，路由器等网络设置。主机端口及URL服务等。根据不同业务故障级别发出告警信息给管理员，当故障恢复时也会发出恢复信息。官网为：<http://www.nagios.org/>

nagios特点：

1. 监控网络服务（STMP, POP3, HTTP, TCP, PING等）；
2. 监控主机资源（CPU负载，IO状况，虚拟及正式内存及磁盘利用率等）；
3. 简单的插件设计模式使得用户可以方便定制符合自己的服务的检测方法
4. 并行服务检查机制
5. 具备定义网络分层结构的能力，用”parent”主机定义来表达网络主机之间的关系，这种关系可以被用来发现和明晰主机宕机或不可达状况
6. 当服务或主机问题产生与解决时将告警发送给联系人（email/im/sms/sound）
7. 具备定义事件句柄功能，它可以在主机或服务的事件发生时获取更多问题定位
8. 可选的web界面用于查看当前的网络状态，通知和故障历史，日志文件等

## nagiso监控工具原理图



# 第一部分 Nagios服务端安装部分

**（本例nagiosserver的IP为192.168.80.161 Centos6.5\_x86）**

## 1.1 nagios准备部分

### 1.1.1准备3台服务器

1）nagios服务器及客户端服务器

|  |  |  |
| --- | --- | --- |
| 管理IP地址 | 角色 | 备注 |
| 192.168.80.161 | Nagios监控服务器 |  |
| 192.168.80.162 | LAMP服务器 | 被监控的客户端服务器 |
| 192.168.80.163 | LNMP服务器 | 被监控的客户端服务器 |

### 1.1.2设置yum安装源

|  |
| --- |
| cd /etc/yum.repos.d  /bin/mv CentOS-Base.repo CentOS-Base.repo.ori  wget http://mirrors.aliyun.com/repo/Centos-6.repo  /bin/mv Centos-6.repo CentOS-Base.repo  yum makecache |

报错如下：

|  |
| --- |
| YumRepo Error: All mirror URLs are not using ftp, http[s] or file.  Eg. $releasever is not a valid release or hasnt been released yet/ removing mirrorlist with no valid mirrors: /var/cache/yum/base/mirrorlist.txt Error: Cannot retrieve repository metadata (repomd.xml) for repository: base. Please verify its path and try again  从字面意思是看是不能正确连接到http或是ftp的真实地址上。  1.有可能是防火墙阻碍了出网http请求或是ftp请求，请检查网关防火墙。  2.有可能是dns不对，无法正常解析，换成8.8.4.4  3.系统取不到Release号  我这里碰到真正的原因是就是3，系统读取不到Release号。  所以需要修改\*.repo源文件中将$releasever变量全替换成6即可正常yum了。  sed -i 's/\$releasever/6/g' Centos-6.repo |

### 1.1.3解决perl编译问题

后面编译的软件有perl程序，这里要提前设置下环境变量

批量执行如下命令

|  |
| --- |
| echo 'export LC\_ALL=C' >> /etc/profile  tail -1 /etc/profile  source /etc/profile  echo $LC\_ALL  cd ~ |

### 1.1.4关闭nagios server iptables防火墙

测试环境下为了调试方便而关掉iptables，生产环境在调试完毕后需要开启防火墙，允许nagios服务通过就可以了，不需要关闭。一般允许的方法是LAN IP端都通过。

|  |
| --- |
| /etc/init.d/iptables stop  /etc/init.d/iptables status  chkconfig iptables off  chkconfig --list iptables |

### 1.1.5解决系统时间同步

如果不解决时间同步，会导致nagios配置异常甚至失败，这个是很多人不注意的地方，从而导致nagios安装部署失败。批量命令如下

|  |
| --- |
| echo "#time sync by fish at 2015-7-28" >> /var/spool/cron/root  echo '\*/10 \* \* \* \* /usr/sbin/ntpdate pool.ntp.org >/dev/null 2>&1' >> /var/spool/cron/root  crontab -l  说明：> /dev/null 表示把正确的输出结果保存到/dev/null中。2>&1：其中&1表示正确的输出通道。2&>1表示把错误的输出结果也保存到正确的输出通道中，也就是/dev/null |

## 1.2安装nagios所需基础软件包

|  |
| --- |
| yum install gcc glbc glibc-comm gd gd-devel httpd php php-gd -y  #设置yum安装本地保留rpm包，以便以后直接rpm安装  vim /etc/yum.conf 把keepcache=0修改为keepcache=1；表示下载的rpm包不清除  通过yum 下载的rpm包默认路径为/var/cache/yum/base/packages/  如果你想用软件包编译apache来做nagios环境的话，那么 yum install httpd php\* -y这条命令不执行。 |

## 1.3创建nagios需要的用户及组

|  |
| --- |
| /usr/sbin/useradd -m nagios  /usr/sbin/useradd apache -M -s /sbin/nologin  /usr/sbin/groupadd nagcmd  /usr/sbin/usermod -a -G nagcmd nagios  /usr/sbin/usermod -a -G nagcmd apache  id -n -G nagios  id -n -G apache  groups nagios  groups apache  说明：useradd -M 表示不创建家目录，-s /sbin/nologin表示不能登录。apache用户可能已经存在，因为使用yum 安装httpd会自动创建apache用户。  /usr/sbin/usermod -a -G nagcmd nagios 表示把nagios加入到nagcmd组中。  id -n -G nagios 检查nagios用户属于那个组，-n等同于--name，显示组名  注意：如果你是yum安装的httpd，那么默认的httpd.conf配置文件中User和Group都是apache。如果你是编译安装httpd，那么是daemon。 |

## 1.4下载nagios和nagios相关组件

|  |
| --- |
| mkdir -p /home/fish/tools/nagios  cd /home/fish/tools/nagios  wget <http://jaist.dl.sourceforge.net/project/nagios/nagios-3.x/nagios-3.2.1/nagios-3.2.1.tar.gz>  wget <https://nagios-plugins.org/download/nagios-plugins-1.4.13.tar.gz>  wget <ftp://ftp.dei.uc.pt/pub/CPAN/authors/id/K/KA/KASEI/Class-Accessor-0.31.tar.gz>  wget <http://search.cpan.org/CPAN/authors/id/A/AD/ADAMK/Config-Tiny-2.12.tar.gz>  wget <ftp://ftp.gnome.org/mirror/CPAN/modules/by-category/06_Data_Type_Utilities/Math/Math-Calc-Units-1.07.tar.gz>  wget <http://search.cpan.org/CPAN/authors/id/T/TO/TONVOON/Nagios-Plugin-0.34.tar.gz>  wget <ftp://ftp.lf.net/pub/Mirrors/CPAN/modules/by-module/Params/Params-Validate-0.91.tar.gz>  wget <https://exchange.nagios.org/components/com_mtree/attachment.php?link_id=3379&cf_id=24>  wget <http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.12/nrpe-2.12.tar.gz/download>  wget <http://sourceforge.net/projects/pnp4nagios/files/PNP/pnp-0.4.14/pnp-0.4.14.tar.gz/download>  wget <ftp://ftp.zedz.net/pub/security/network-monitoring/rrdtool/rrdtool-1.2.14.tar.gz>  wget <ftp://ftp.onenet.net/mirrors/CPAN/modules/by-module/Regexp/Regexp-Common-2010010201.tar.gz>  wget http://ftp.gnome.org/pub/gnome/sources/libart\_lgpl/2.3/libart\_lgpl-2.3.17.tar.gz |

## 1.5安装nagios

|  |
| --- |
| cd /home/fish/tools/nagios  tar zxf nagios-3.2.1.tar.gz  cd nagios-3.2.1  ./configure --with-command-group=nagcmd  make all  make install  make install-init  make install-commandmode  make install-config  make install-webconf  如果是编译安装httpd时：可以执行./configure --with-command-group=nagcmd --with-httpd-conf=/usr/local/apache2/conf/extra 增加一个编译参数即指定编译nagios web配置的生成路径。  ./configure 正确结尾图。  编译安装的apache，在make install-webconf之前需要做如下操作  mkdir -p /etc/httpd/conf.d/ && mkdir -install-webconf  cp /etc/httpd/conf.d/nagios.conf /usr/local/apache2/conf/extra    make all 正确结尾图    make install 正确结尾图    make install-init 正确结尾图    make install-commandmode 正确结尾图    make install-config正确结尾图    make install-webconf正确结尾图 |

## 1.6创建nagios web监控界面登入时需要的用户名（fish）及密码

|  |
| --- |
| [root@nagios-server nagios-3.2.1]# htpasswd -c /usr/local/nagios/etc/htpasswd.users fish  New password:  Re-type new password:  Adding password for user fish  这里我写的密码是fish  [root@nagios-server nagios-3.2.1]# cat /usr/local/nagios/etc/htpasswd.users  fish:637rKTY.BXOok  #提示：如果为单独编译的apache，则操作如下  /usr/local/apache2/bin/htpasswd -c /usr/local/nagios/etc/htpasswd.users fish  为什么要把密码文件指定到/usr/local/nagios/etc/htpasswd.users这个位置呢？因为在make install-webconf生成的文件里面已经定义好了  [root@nagios-server nagios-3.2.1]# cat /etc/httpd/conf.d/nagios.conf | grep AuthUserFile  AuthUserFile /usr/local/nagios/etc/htpasswd.users  AuthUserFile /usr/local/nagios/etc/htpasswd.users |

## 1.7添加监控报警的接受email地址

|  |
| --- |
| sed -i 's#nagios@localhost#linkagetest@126.com#' /usr/local/nagios/etc/objects/contacts.cfg  #1.此功能依赖本机的sendmail服务，可以执行service sendmail start开启  [root@nagios-server ~]# /etc/init.d/sendmail start  Starting sendmail: [ OK ]  Starting sm-client: [ OK ]  [root@nagios-server ~]# netstat -tunlp | grep 25  tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN 1408/master  tcp 0 0 ::1:25 :::\* LISTEN 1408/master  如果sendmail启动很慢，可以修改/etc/hosts 文件  127.0.0.1 nagios-server localhost.localdomain localhost4 localhost4.localdomain4 |

## 1.8配置apache服务

|  |
| --- |
| /etc/init.d/httpd start  netstat -tunlp | grep 80  ps -ef | grep httpd |

## 1.9安装nagios插件软件包

|  |
| --- |
| yum install mysql-devel mysql-server -y  tar zxf nagios-plugins-1.4.13.tar.gz  cd nagios-plugins-1.4.13  ./configure \  --with-nagios-user=nagios --with-nagios-group=nagios --enable-perl-modules  make  make install  注意：在make的时候，如果报错 \*\*\* [all] Errot 2 则./configure的时候需要加上--with-mysql=/usr/local/mysql 解决 |

## 1.10添加nagios自启动

|  |
| --- |
| [root@nagios-server nagios]# chkconfig nagios on  [root@nagios-server nagios]# chkconfig --list nagios  nagios 0:off 1:off 2:on 3:on 4:on 5:on 6:off |

## 1.11验证nagios配置文件（检查语法）

|  |
| --- |
| [root@nagios-server nagios]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  …省略…  Total Warnings: 0  Total Errors: 0  #Total Warnings 警告和Total Errors错误都为0，表示配置正常  [root@nagios-server nagios]# /etc/init.d/nagios checkconfig  Running configuration check... OK.  此命令默认情况只能判断错误或正确，并不能显示错误的详细信息，如果想显示详细错误信息，需要做一下简单的配置。 |

启动nagios服务

|  |
| --- |
| [root@nagios-server nagios]# /etc/init.d/nagios start  Starting nagios: done. |

检查nagios进程

|  |
| --- |
| [root@nagios-server nagios]# ps -ef | grep nagios | grep -v grep  nagios 42706 1 0 16:21 ? 00:00:00 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg |

此时打开浏览器验证访问：[http://192.168.80.161/nagios](http://192.168.10.161/nagios) 会出现如下提示信息：

用户名/密码：fish/fish

## 1.12安装nrpe

|  |
| --- |
| tar zxf nrpe-2.12.tar.gz  cd nrpe-2.12  ./configure  make all  make install-plugin  make install-daemon  make install-daemon-config  cd ..  提示：生成nrpe的配置文件/usr/local/nagios/etc/nrpe.cfg |

服务端也要安装nrpe，不然会缺少相关检查插件，另外服务端也会监控自身的基本健康状态。

# 第二部分 Nagios客户端安装

**[本例客户端IP为192.168.80.162 CentOS6.5\_64]**

**[本例客户端IP为192.168.80.163 CentOS6.5\_64]**

## 2.1 nagios client端基础安装

### 2.1.1设置yum安装源（同server端）

[设置yum源目的]

默认情况下yum会从国外的站点下载，速度慢

换成国内的yum源站点，这样以后yum安装软件时更快。

|  |
| --- |
| cd /etc/yum.repos.d  /bin/mv CentOS-Base.repo CentOS-Base.repo.ori  wget http://mirrors.aliyun.com/repo/Centos-6.repo  /bin/mv Centos-6.repo CentOS-Base.repo  # sed -i 's/\$releasever/6/g' CentOS-Base.repo  yum makecache |

### 2.1.2 解决perl编译问题[同server端]

后面编译的软件有perl程序，这里要提前设置下环境变量

|  |
| --- |
| echo 'export LC\_ALL=C' >> /etc/profile  tail -1 /etc/profile  source /etc/profile  echo $LC\_ALL  cd ~ |

### 2.1.3关闭nagios server iptables防火墙

测试环境下为了调试方便而关掉iptables，生产环境在调试完毕后需要开启防火墙，允许nagios服务通过就可以了，不需要关闭。一般允许的方法是LAN IP端都通过。

|  |
| --- |
| /etc/init.d/iptables stop  /etc/init.d/iptables status  chkconfig iptables off  chkconfig --list iptables |

### 2.1.4解决系统时间同步

如果不解决时间同步，会导致nagios配置异常甚至失败，这个是很多人不注意的地方，从而导致nagios安装部署失败。批量命令如下

|  |
| --- |
| /usr/sbin/ntpdate pool.ntp.org  echo "#time sync by fish at 2015-7-28" >> /var/spool/cron/root  echo '\*/10 \* \* \* \* /usr/sbin/ntpdate pool.ntp.org >/dev/null 2>&1' >> /var/spool/cron/root  crontab -l  说明：> /dev/null 表示把正确的输出结果保存到/dev/null中。2>&1：其中&1表示正确的输出通道。2&>1表示把错误的输出结果也保存到正确的输出通道中，也就是/dev/null |

## 2.2在nagios客户端安装插件

|  |
| --- |
| mkdir -p /home/fish/tools  cd /home/fish/tools/  rz -y #上传nagios下载好的集合包  unzip nagios\_gather.zip  cd nagios\_gather |

### 2.2.1安装nagios-plugins-1.4.3.tar.gz

|  |
| --- |
| yum install mysql-devel -y  useradd nagios -M -s /sbin/nologin  tar zxf nagios-plugins-1.4.13.tar.gz  cd nagios-plugins-1.4.13  ./configure \  --with-nagios-user=nagios --with-nagios-group=nagios --enable-perl-modules  make  make install  cd ..  如果在make时候遇到：\*\*\*[all] Error 2则configure加--with-mysql=/usr/local/mysql解决  其中/usr/local/mysql为mysql的安装路径  如果为redhat linux，则可能会卡到如下位置：  configure: see <http://nagiosplugins.org/fag/compile/configure_appers_to_hang> if this next part takes a long time check for redhat spoen problem  解决办法：   1. 多等待一会就OK了，或者编译增加参数 --enable-redhat-pthread-workaround 重新编译   如果没有check\_mysql需要安装mysql-devel然后./configure && make && make install |

此时检查插件个数：

|  |
| --- |
| ls /usr/local/nagios/libexec/ | wc -l  54 |

### 2.2.2安装nrpe

|  |
| --- |
| tar zxf nrpe-2.12.tar.gz  cd nrpe-2.12  ./configure  make all  make install-plugin  make install-daemon  make install-daemon-config  cd ..  提示：生成nrpe的配置文件/usr/local/nagios/etc/nrpe.cfg |

### 2.2.3安装其他相关插件[iostat]

|  |
| --- |
| #------------------------分割线-----------------------  tar zxf Params-Validate-0.91.tar.gz  cd Params-Validate-0.91  perl Makefile.PL  make && make install  cd..  #------------------------分隔符------------------------------  tar zxf Class-Accessor-0.31.tar.gz  cd Class-Accessor-0.31  perl Makefile.PL  make && make install  cd ..  #----------------------分隔符--------------------------------------  tar zxf Config-Tiny-2.12.tar.gz  cd Config-Tiny-2.12  perl Makefile.PL  make && make install  cd ..  #-------------------------分隔符------------------------------------  tar zxf Math-Calc-Units-1.07.tar.gz  cd Math-Calc-Units-1.07  perl Makefile.PL  make && make install  cd ..  #-------------------------------分隔符---------------------------------  tar zxf Regexp-Common-2010010201.tar.gz  cd Regexp-Common-2010010201  perl Makefile.PL  make && make install  cd ..  #--------------------------------分隔符------------------------------------  tar zxf Nagios-Plugin-0.34.tar.gz  cd Nagios-Plugin-0.34  perl Makefile.PL  make && make install  cd ..  #------------------------------分割符----------------------------------------  #for monitor iostat  yum install sysstat -y  #这个命令是安装系统性能分析工具，监控系统性能时程序或脚本都会调用这些工具 |

## 2.3配置开发的几个基础脚本插件

|  |
| --- |
| #当前路径为软件包的路径  /bin/cp /home/fish/tools/nagios\_gather/check.memory.sh /usr/local/nagios/libexec/  /bin/cp /home/fish/tools/nagios\_gather/check\_iostat /usr/local/nagios/libexec/  chmod 755 /usr/local/nagios/libexec/check.memory.sh  chmod 755 /usr/local/nagios/libexec/check\_iostat  yum install dos2unix -y  dos2unix /usr/local/nagios/libexec/check.memory.sh  dos2unix /usr/local/nagios/libexec/check\_iostat |

## 2.4配置nrpe

|  |
| --- |
| cd /usr/local/nagios/etc  vim nrpe.cfg +79  加入可以监控当前client的nagios server端的IP，如下  allow\_hosts=192.168.80.161  提示：allow\_hosts=后面的IP为nagios服务端的IP，不是客户端的  如果机器大于500台做集群或分布式监控时，可以使用多个nagios\_hosts， ip要用都逗号隔开。  上面的修改可以使用下列命令进行替换：  sed -i 's/allowed\_hosts=127.0.0.1/allowed\_hosts=192.168.80.161/g' /usr/local/nagios/etc/nrpe.cfg  grep allowed\_hosts /usr/local/nagios/etc/nrpe.cfg  小技巧：echo "alias grep='grep --color=auto'" >> ~/.bashrc 。这样以后grep过滤过来的字符会加上颜色 |

然后在命令行模式下执行shilf+g到结尾 进行如下修改：或者注释掉或者删除掉199-203行。即下面几行

|  |
| --- |
| 199 #command[check\_users]=/usr/local/nagios/libexec/check\_users -w 5 -c 10  200 #command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20  201 #command[check\_hda1]=/usr/local/nagios/libexec/check\_disk -w 20% -c 10% -p /dev/hda1  202 #command[check\_zombie\_procs]=/usr/local/nagios/libexec/check\_procs -w 5 -c 10 -s Z  203 #command[check\_total\_procs]=/usr/local/nagios/libexec/check\_procs -w 150 -c 200 |

同时在下面新添加要监控的内容：

|  |
| --- |
| command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20  command[check\_mem]=/usr/local/nagios/libexec/check.memory.sh -w 10% -c 3%  command[check\_disk]=/usr/local/nagios/libexec/check\_disk -w 15% -c 7% -p /  command[check\_swap]=/usr/local/nagios/libexec/check\_swap -w 20% -c 10%  command[check\_iostat]=/usr/local/nagios/libexec/check\_iostat -w 6 -c 10  上面依次是对负载，内存，磁盘，虚拟内存，磁盘IO的监控，这些都是本地的服务（我们这里称之为被动监控），由nagios服务器端通过nrpe插件定时去client的nrpe服务定期获取信息，原理如图。  对于负载，单核CPU的负载不超过5，双核CPU的负载不超过10  中括号里面的名称是需要在nagios服务器端调用的。比较重要 |



启动nagios client

|  |
| --- |
| /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d  echo "# nagios nrpe process cmd by fish 2015-7-30" >> /etc/rc.local  echo "/usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d" >> /etc/rc.local  ps -ef | grep nagios  重启命令：  pkill nrpe && /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d |

提示：客户端的nrpe.cfg配置文件，最好保存到一份到电脑上，这样以后在其他的机器上装nrpe时，直接上传即可，不需要在重新修改了。当然了。客户端nrpe的服务也可以写成一个启动的脚本

下面启动脚本：

|  |
| --- |
| [root@nagios-client1 init.d]# cat nrped  #!/bin/bash  # auth:fish  # version: v1.0  # nrpe start/stop/status service  #  # chkconfig: 35 13 91  # process nagios  # source functions  . /etc/init.d/functions  # base dir  NRPE\_BASE\_DIR=/usr/local/nagios  # bin file  NRPE\_BIN=$NRPE\_BASE\_DIR/bin/nrpe  # config file  NRPE\_CONFIG=$NRPE\_BASE\_DIR/etc/nrpe.cfg  start(){  # start nrpe cmd  $NRPE\_BIN -c $NRPE\_CONFIG -d  if [ $? -eq 0 -a `ps -ef | grep nagios | grep -v grep | wc -l` -gt 0 ];then  action "Starting nrpe..." /bin/true  sleep 1  else  action "Starting nrpe..." /bin/false  fi  }  stop(){  # stop nrpe cmd  pkill nrpe;sleep 1;pkill nrpe  if [ `ps -ef | grep nagios | grep -v grep | wc -l` -lt 1 ];then  action "Stopping nrpe..." /bin/true  sleep 1  else  action "Stopping nrpe..." /bin/true  fi  }  status(){  if [ `ps -ef | grep nagios | grep -v grep | wc -l` -gt 0 ];then  echo "nrpe is running..."  else  echo "nrpe is stopped"  fi  }  case "$1" in  start)  start  ;;  stop)  stop  ;;  restart)  stop  start  ;;  status)  status  ;;  \*)  echo "Usage: $0 {start|stop|restart|status}"  esac |

加入开机自动启动

|  |
| --- |
| [root@nagios-client1 init.d]# chkconfig --add nrped  [root@nagios-client1 init.d]# chkconfig nrped on  [root@nagios-client1 init.d]# chkconfig --list nrped  nrped 0:off 1:off 2:on 3:on 4:on 5:on 6:off |

# 第三部分 配置server 端nagios监控的服务

**[注意：nagios\_server 的IP为192.168.80.161 CentOS6.5\_64bits]**

**以下为nagios-server端的操作**

## 3.1 nagios目录结构

nagios服务器端安装后的目录结构

|  |
| --- |
| [root@nagios-server ~]# ll /usr/local/nagios/  total 28  drwxrwxr-x 2 nagios nagios 4096 Jul 28 17:01 bin  drwxrwxr-x 3 nagios nagios 4096 Jul 28 17:01 etc  drwxrwxr-x 2 nagios nagios 4096 Jul 29 17:20 libexec  drwxr-xr-x 5 root root 4096 Jul 28 16:14 perl  drwxrwxr-x 2 nagios nagios 4096 Jul 28 14:41 sbin  drwxrwxr-x 10 nagios nagios 4096 Jul 28 16:14 share  drwxrwxr-x 5 nagios nagios 4096 Jul 30 14:31 var |

|  |  |
| --- | --- |
| 目录名称 | 说明 |
| bin | bin 下为nagios相关命令  [root@nagios-server nagios]# tree bin/  bin/  |-- nagios  |-- nagiostats  `-- nrpe  0 directories, 3 files |
| etc | etc下是nagios 的配置文件及目录信息  [root@nagios-server nagios]# tree etc/  etc/  |-- cgi.cfg  |-- htpasswd.users  |-- nagios.cfg #===🡺相当于httpd.conf  |-- nrpe.cfg  |-- objects #====🡺 相当与apache的extra包含目录  | |-- commands.cfg  | |-- contacts.cfg  | |-- localhost.cfg  | |-- printer.cfg  | |-- switch.cfg  | |-- templates.cfg  | |-- timeperiods.cfg  | `-- windows.cfg  `-- resource.cfg  1 directory, 13 files |
| libexec | libexec为所有插件的目录路径  [root@nagios-server nagios]# tree libexec/  libexec/  |-- check\_apt  |-- check\_breeze  |-- check\_by\_ssh  |-- check\_clamd -> check\_tcp  |-- check\_cluster  |-- check\_dhcp  |-- check\_dig  |-- check\_disk  |-- check\_disk\_smb  |-- check\_dns  |-- check\_dummy  |-- check\_file\_age  |-- check\_flexlm  |-- check\_ftp -> check\_tcp  |-- check\_http  |-- check\_icmp  ….省略部分 |
| sbin | [root@nagios-server nagios]# tree sbin/  sbin/  |-- avail.cgi  |-- cmd.cgi  |-- config.cgi  |-- extinfo.cgi  |-- histogram.cgi  |-- history.cgi  |-- notifications.cgi  |-- outages.cgi  |-- showlog.cgi  |-- status.cgi  |-- statusmap.cgi  |-- statuswml.cgi  |-- statuswrl.cgi  |-- summary.cgi  |-- tac.cgi  `-- trends.cgi  0 directories, 16 files |
| share | share为nagios界面的php程序等内容的目录  [root@nagios-server nagios]# tree -L 1 share/  share/  |-- config.inc.php  |-- contexthelp  |-- docs  |-- images  |-- includes  |-- index.php  |-- locale  |-- main.php  |-- media  |-- robots.txt  |-- side.php  |-- ssi  `-- stylesheets  8 directories, 5 files |
| var | var 为nagios数据及日志的目录  [root@nagios-server nagios]# tree var/  var/  |-- archives  |-- nagios.lock  |-- nagios.log  |-- objects.cache  |-- retention.dat  |-- rw  | `-- nagios.cmd  |-- spool  | `-- checkresults  `-- status.dat  4 directories, 6 files |

所有客户端本地服务的监控都是通过执行libexec目录下的插件来实现的

nagios的主配置文件为nagios.cfg，默认在/usr/local/nagios/etc目录下，/usr/local/nagios/etc目录下有个objects目录。objects里面放的是主配置文件nagios.cfg包含的其他的nagios配置文件

|  |
| --- |
| [root@nagios-server nagios]# tree etc/objects/  etc/objects/  |-- commands.cfg  |-- contacts.cfg  |-- localhost.cfg  |-- printer.cfg  |-- switch.cfg  |-- templates.cfg  |-- timeperiods.cfg  `-- windows.cfg  0 directories, 8 files |

在nagios.cfg中既可以指定单独包含一个cfg文件，也可以指定包含一个目录，即该目录下所有的cfg文件都会包含进来。为了目录结构看起来更清晰和批量部署服务的需要。我们把主配置文件包含的配置文件修改如下：

nagios配置文件说明：

|  |  |
| --- | --- |
| 配置文件名称 | 说明 |
| commands.cfg | 存放命令相关配置（也可以指定commands目录） |
| services.cfg | 存放服务相关配置（上百台以上可以指定services目录）（默认不存在） |
| hosts.cfg | 存放主机相关配置（上百台以上可以指定hosts目录）（默认不存在） |
| contacts.cfg | 存放报警联系人相关配置 |
| timeperiods.cfg | 存放报警周期时间等相关配置 |
| templates.cfg | 模版配置文件 |

## 3.2配置主配置文件nagios.cfg

在nagios.cfg文件中找到cfg\_file部分，进行如下设置

|  |
| --- |
| cd /usr/local/nagios/etc/  cp nagios.cfg nagios.cfg.ori  vim /usr/local/nagios/etc/nagios.cfg +34  cfg\_file=/usr/local/nagios/etc/object/hosts.cfg  cfg\_file=/usr/lcoal/nagios/etc/object/services.cfg  cfg\_dir=/usr/local/nagios/etc/object/services  #作为备用增加一个services目录，使用目录的有点很多，在目录下的文件只要符合\*.cfg就可以被nagios加载，使用脚本批量部署时非常方便的随机命名配置文件。 |

并注释掉如下一行

|  |
| --- |
| # Definitons for monitoring the local (linux) host  #cfg\_file=/usr/local/nagios/etc/objects/localhost.cfg  #l localhost.cfg 这个配置为监控nagios服务端本地服务的配置文件，注释掉它，然后统一监控 |

操作完毕，保存nagios.cfg并建立目录 mkdir /usr/local/nagios/etc/object/services

对目录授权 chown -R nagios.nagios /usr/local/etc/object/services

生成host.cfg文件

|  |
| --- |
| head -51 localhost.cfg > hosts.cfg  chown -R nagios.nagios /usr/local/nagios/etc/object/hosts.cfg |

生成services.cfg文件

|  |
| --- |
| touch /usr/local/nagios/etc/object/services.cfg  chown -R nagios.nagios /usr/local/nagios/etc/objects/services.cfg |

## 3.3 hosts.cfg生产库实战配置例子语法

### 3.3.1 hosts.cfg中主机定义部分配置参数详解

hosts.cfg一般用来存放主机的相关配置，设置机器名为 192-etiantian-1-1,格式为（ip-服务-组 ID+组内机器 ID）,这样设置的目的，便于查看。

hosts.cfg中的主机定义部分配置参数详解

|  |
| --- |
| define host { #define host 为关键字，意思是定义主机，主机内容用一对大括号括起来  use linux-server #定义主机使用的模版，集体参见templates.cfg  host\_name 197-etiantian-1-1 #主机名称，根据服务功能可以随意定义  alias 197-etiantian-1-1 #主机别名，同上  address 192.168.80.162 #被监控服务器的IP地址  check\_command check-host-alive #检测主机存活命令  max\_check\_attempts 3 #故障后，最大尝试检测次数  retry\_check\_interval 2 #故障后重试检查间隔，默认单位为分钟  normal\_check\_interval 2 #正常的检查间隔，默认单位为分钟  check\_period 24\*7 #检查周期，具体参见timeperiods.cfg  notification\_interval 300 #故障后，两次报警额通知间隔，默认单位分钟  notification\_period 24\*7 #一天之内通知的周期，比如全天，还是半天等  notification\_option d,u,r #主机状态通知选项，d-down, u-unreacheable, r-recovery  contact\_groups admin,nagios，zabbix #报警到admin用户组，在contaces.cfg里定义  } |

### 3.3.2 host.cfg中主机定义部分批量模版化配置

主机的配置也可以值配置关键选项，多数选项可采取linux-server模版的默认值，有能力的还可以先调整linux-server模版。然后所有机器统一采用这种默认值，看起来更加方便。如下：

|  |
| --- |
| define host {  use linux-server  host\_name 197-etiantian-1-1  alias 197-etiantian-1-1  address 192.168.80.162  }  #以上两种配置都可以，服务多时可以批量部署，当然也可以安装图形化配置界面Centreon，NagiosQL，但是不推荐图形化配置。 |

### 3.3.3千台服务器模版化繁为简的host部署配置

|  |
| --- |
| define host {  use linux-server  host\_name 162-etiantian-1-1  alias 162-etiantian-1-1  address 192.168.80.162  }  define host {  use linux-server  host\_name 163-etiantian-1-2  alias 163-etiantian-1-2  address 192.168.80.163  }  ….省略…  特别提示：以上的配置可以通过脚本来生成 |

## 3.4 services.cfg生产库实战配置例子语法

services.cfg文件是配置监控服务的，是nagios最重要的配置文件之一，对于服务器数量比较少（50台以内），大部分需要监控的服务配置都可以在这里面添加，这个配置文件默认是不存在的，是人为创建的。

### 3.4.1 services.cfg中服务定义部分配置参数详解

services.cfg配置文件的service配置参数详细说明

|  |
| --- |
| define service { #define service为关键字，意思是定义一个服务，服务内容用一对大括号括起来  use generic-service #定义该服务使用的模版，具体参见templates.cfg  host\_name 197-etiantian-1-1 #被监控的主机名，来自hosts.cfg，可以在hosts.cfg中自定义  service\_description Current load #报警服务描述，根据内容取有意义的名称  check\_command check\_nrpe!check\_load #检查服务的命令，这个很关键，注意被动服务的监控均由check\_nrpe调用。这个check\_load是在客户端192.168.80.163中/usr/local/nagios/etc/nrpe.cfg配置文件中command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20，中括号里面的名字[check\_load],这里定义的  max\_check\_attempts 2 #故障后，最大尝试检测次数  retry\_check\_interval 4 #故障后重试检查间隔，默认单位为分钟  normal\_check\_interval 4 #正常的检查间隔，默认单位为分钟  check\_period 24\*7 # 检查周期，具体参见timeperiods.cfg  notification\_interval 1440 #故障后，两次报警的通知间隔，默认单位分钟  notification\_period 24\*7 #一天之内通知的周期，比如全天，还是半天等  notification\_option w,u,c,r #主机状态通知选项，w-worning,u-unknow,c-critical,, r-recovery  contact\_groups admin,nagios，zabbix #报警到admin用户组，在contaces.cfg里定义  process\_perf\_data 1 #PNP出图记录数据相关  } |

### 3.4.2磁盘分区监控[被动监控]

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d #这里可以指定多台机器，通过逗号隔开，调用的是hosts.cfg配置文件中的主机名  service\_description Disk Partition  check\_command check\_nrpe!check\_disk  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups mailusers #磁盘的报警值定的合理一些，可质保邮件不报短信了  process\_perf\_data 1  }  提示：   1. 同样的服务可以定义多个比如磁盘可根据服务器需求不同，在重新定义，报给不同人 2. 可以同时报给多个组，这样写contact\_groups，mailusers,masnusers，用户组的定义见contact.cfg 3. 也可以简写，然后大部分参数在模版中定义。如   define service {  use genetic-disk-services #这就是使用的在templates.cfg定义好的模版的模版名  host\_name 072-wike  } |

### 3.4.3 swap监控[ 被动监控]

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription Swap Usage  check\_command check\_nrpe! check\_swap  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period workhours  contact\_groups admin  process\_perf\_data 1  } |

通过使用模版templates.cfg定义可以把上面的配置简写为如下几行：

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  } |

### 3.4.4内存监控[被动监控]

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription MEM Usage  check\_command check\_nrpe!check\_mem  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period workhours  contact\_groups admin  process\_perf\_data 1  } |

通过使用模版templates.cfg定义可以把上面的配置简写为如下几行：

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  } |

### 3.4.5系统负载监控【被动监控】

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription Current Load  check\_command check\_nrpe!check\_load  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

通过使用模版templates.cfg定义可以把上面的配置简写为如下几行：

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  } |

### 3.4.6 IO监控【被动监控】

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription Disk Iostat  check\_command check\_nrpe!check\_iostat!5!11  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

通过使用模版templates.cfg定义可以把上面的配置简写为如下几行：

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  } |

### 3.4.7 ping监控【被动监控】

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription PING  check\_command check\_ping!100.0,20%!500.0,60%  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

通过使用模版templates.cfg定义可以把上面的配置简写为如下几行：

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  } |

### 3.4.8千台服务器模版化繁为简的service部署配置

|  |
| --- |
| define service { #7 ping  use generic-ping-service  #这就是使用在templates.cfg定义好的模版的模版名  host\_name a,b,c,d  }  define service {  use generic-load-service #这就是使用在templates.cfg定义好的模版的模版名  host\_name a,b,c,d  }  define service {  use generic-disk-service #这就是使用在templates.cfg定义好的模版的模版名  host\_name a,b,c,d  }  …省略… |

其中内存和io插件nagios默认没有提供，但本文在配置时已经拷贝进去了，因此，只需要在commands.cfg中配置即可。

### 3.4.9 URL监控

#### 3.4.9.1顶级域名blog.etiantian.org地址监控

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription blog\_url  check\_command check\_weburl!-H blog.etitantian.org  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

#### 3.4.9.2域名下带目录文件<http://blog.etiantian.org/jimo/exam.php>地址监控

URL监控是有服务器端发送请求主动探测得到的，属于主动监控

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription dati\_url\_ett  check\_command check\_weburl!-H blog.etitantian.org -u /jimo.exam.php  #check\_weburl这是检测命令，在command.cfg里面定义，后面就是真正检查的url地址了，其中-u 后加域名后面的地址。  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

对于监控url的更多选项可以查看check\_httpd命令插件

### 3.4.10服务端口监控

注意写法host\_name 也可以同时写多个，要用逗号分开

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription img\_8150  check\_command check\_tcp!8150  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

### 3.4.11多端口监控的写法

|  |
| --- |
| define service {  use generic-service  host\_name a,b,c,d  service\_desctription img\_8150  check\_command check\_tcp!8150!6915!6916  max\_check\_attempts 8  retry\_check\_interval 2  normal\_check\_interval 3  check\_period 24x7  notification\_interval 360  notification\_options w,u,c,r  notification\_period 24x7  contact\_groups admin  process\_perf\_data 1  } |

## 3.5 commands.cfg生产库实战配置例子写法

commands.cfg这个是定义报警命令的配置文件，例如上文提到的check\_weburl检查命令就是在这里定义的。

|  |
| --- |
| # 'notify-host-by-email' command definition  define command{  command\_name notify-host-by-email  command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ \*\*" $CONTACTEMAIL$  }  # 'notify-service-by-email' command definition  define command{  command\_name notify-service-by-email  command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ \*\*" $CONTACTEMAIL$  }  # ‘check\_nrpe’ command definition #check\_nrpe 命令的定义  define command{  command\_name check\_nrpe  command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$  }  define command{  command\_name check\_weburl  command\_line $USER1$/check\_http $ARG1$ -w 10 -c 30 # $USER1$就是/usr/local/nagios/libexec  }  …省略… |

## 3.6 contacts.cfg生产库实战配置例子写法

|  |
| --- |
| #手机短信用户  define contact{  contact\_name shouji\_jeacen  use generic-contact  alias nagios admin  email 134521902211@139.com  }  define contact{  contact\_name shouji\_wodi  use generic-contact  alias nagios admin  email 13452222211@139.com  }  #邮件用户及msn帐号  define contact{  contact\_name msn\_fish  use generic-contact  alias nagios admin  email [134521211@139.com](mailto:134521211@139.com)  address1 rr@hotmail.com  }  define contact{  contact\_name msn\_fish2  use generic-contact  alias nagios admin  email [1345212231@139.com](mailto:1345212231@139.com)  address1 rsdfds@hotmail.com  }  #仅仅是邮件用户  define contact{  contact\_name youjian\_fish  use generic-contact  alias nagios admin  email [134521211@139.com](mailto:134521211@139.com)，linkageinfo@126.com  }  efine contact{  contact\_name youjian\_fish2  use generic-contact  alias nagios admin  email [134521211@139.com](mailto:134521211@139.com)，linkageitest@126.com  }  #手机组，手机组包含上面的两个手机成员  define contactgroup{  contactgroup\_name shoujiusers  alias nagios administrator  members shouji\_jeacen, shouji\_wodi  }  #邮件及msn帐号组  define contactgroup{  contactgroup\_name msnusers  alias nagios administrator  members youjian\_fish, msn\_fish2  }  #仅邮件组  define contactgroup{  contactgroup\_name msnusers  alias nagios administrator  members youjian\_fish, youjian\_fish2  } |

## 3.7 timeperiode.cfg 生产库实战例子写法

|  |
| --- |
| define timeperiod{  timeperiod\_name 24x7 #周期的名字为24x7  alias 24 Hours A Day, 7 Days A Week #这个是别名，可以详细描述  sunday 00:00-24:00  monday 00:00-24:00  tuesday 00:00-24:00  wednesday 00:00-24:00  thursday 00:00-24:00  friday 00:00-24:00  saturday 00:00-24:00  }  # ‘workhours’ timeperiod definition  define timeperiod{  timeperiod\_name workhours #周期的名字为workhours，工作监控周期  alias Normal Work Hours  monday 07:30-24:00 #和上面定义的区别是从周一到周日 07:30-24:00，这个时间段进行监控，其他时间不监控  tuesday 07:30-24:00  wednesday 07:30-24:000  thursday 07:30-24:00  friday 07:30-24:00  saturday 07:30-24:00  sunday 07:30-24:00  } |

特别说明：

这几节是增加被监控主机时用的，介绍了各个配置的语法。

## 3.8 templates.cfg生产库实战例子写法

define service{

name generic-disktest-service

use generic-service

service\_description Disk Partition

check\_command check\_nrpe!check\_disk

max\_check\_attempts 3

normal\_check\_interval 3

retry\_check\_interval 2

check\_period 24x7

notification\_interval 1440

notification\_period diskhours

notification\_options c,r

contact\_groups test\_admins

process\_perf\_data 1

register 0

}

## 3.9 nagios 实战配置过程细节

### 3.9.1监控客户端服务器磁盘分区、load负载、mem内存、swap、磁盘IO 【被动监控】

被动监控主要使用的是服务器端的check\_nrpe命令连接客户端nrpe服务，使用客户端的插件命令来搜集主机信息。返回给服务器端

1. 首先需要在hosts.cfg里添加客户端主机和主机组（HOST GROUP）

[root@nagios-server ~]# cd /usr/local/nagios/etc/

[root@nagios-server etc]# vim nagios.cfg cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg

cfg\_file=/usr/local/nagios/etc/objects/services.cfg

[root@nagios-server etc]# cp -a objects objects.ori

[root@nagios-server etc]# chown -R nagios.nagios objects

[root@nagios-server etc]# cd objects

[root@nagios-server objects]# vim hosts.cfg #默认是没有这个文件的，需要重新创建

############## define hosts #######################

define host {

use linux-server

host\_name 162-lamp-1

alias 162-lamp-1

address 192.168.80.162

}

define host {

use linux-server

host\_name 163-lamp-1

alias 163-lamp-1

address 192.168.80.163

check\_command check-host-alive

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 2

check\_period 24x7

notification\_interval 300

notification\_period 24x7

notification\_options d,u,r

contact\_groups admins

process\_perf\_data 1

}

################## define hostgroup ########################

define hostgroup {

hostgroup\_name linux-server

alias Linux Servers

members 162-lamp-1,163-lamp-1

}

1. 接下来添加要监控的服务配置到services.cfg

############## define hosts #######################

define host {

use linux-server

host\_name 162-lamp-1

alias 162-lamp-1

address 192.168.80.162

}

define host {

use linux-server

host\_name 163-lamp-1

alias 163-lamp-1

address 192.168.80.163

check\_command check-host-alive

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 2

check\_period 24x7

notification\_interval 300

notification\_period 24x7

notification\_options d,u,r

contact\_groups admins

process\_perf\_data 1

}

################## define hostgroup ########################

define hostgroup {

hostgroup\_name linux-server

alias Linux Servers

members 162-lamp-1,163-lamp-1

}

[root@nagios-server objects]# cat services

cat: services: Is a directory

[root@nagios-server objects]# cat services.cfg

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description Current Load

check\_command check\_nrpe!check\_load

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description MEM Usage

check\_command check\_nrpe!check\_mem

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description Swap Usage

check\_command check\_nrpe!check\_swap

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description Disk Partition

check\_command check\_nrpe!check\_disk

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description Disk Iostat

check\_command check\_nrpe!check\_iostat!5!11

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

提示：

以上service.cfg中添加了对 Load，Mem，Swap，磁盘分区，磁盘io的监控

若此时执行如下检查nagios语句命令：

[root@nagios-server objects]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

#此处检查语法的命令

1. 需要在commands.cfg中加入check\_nrpe的插件配置，不然会报错

#vim command.cfg 进入以后切换到结尾加入下面内容

#check\_nrpe command definition

define command{

command\_name check\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

此时重新执行检查语法命令

检查语法：

[root@nagios-server objects]# /etc/init.d/nagios checkconfig

Running configuration check... OK.

#这是nagios启动脚本里检查语法的命令，默认情况只能检查是否有错，但不显示错误的详细信息

#如果想显示详细的错误信息请使用下面的命令：

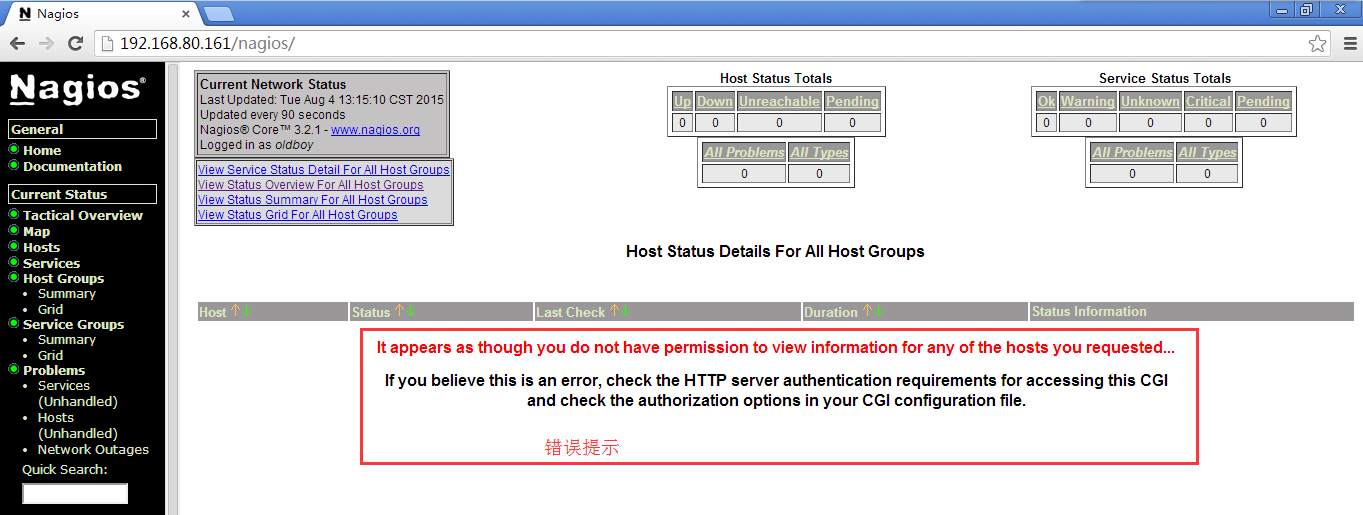
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

接下来使用web登录，打开浏览其如无服务内容，出现下面英文错误提示：

**It appears as though you do not have permission to view information for any of the services you requested...**

**If you believe this is an error, check the HTTP server authentication requirements for accessing this CGI  
and check the authorization options in your CGI configuration file.**

下面截图是hosts按钮上的。



上面报错解决：

[root@nagios-server etc]# cd /usr/local/nagios/etc/

[root@nagios-server etc]# vim cgi.cfg +119

authorized\_for\_system\_information=nagiosadmin,fish

authorized\_for\_configuration\_information=nagiosadmin,fish

authorized\_for\_system\_commands=nagiosadmin,fish

authorized\_for\_all\_services=nagiosadmin,fish

authorized\_for\_all\_hosts=nagiosadmin,fish

authorized\_for\_read\_only=user1,user2,fish

authorized\_for\_all\_service\_commands=nagiosadmin,fish

authorized\_for\_all\_host\_commands=nagiosadmin,fish

提示：

1.理想情况最好干掉默认管理员用户，nagiosadmin或替换自己的fish等，然后分居需求加适合自己的权限，比如给别人看只给浏览权限即可。具体见cgi.cfg

2.authorized\_for\_read\_only=user1,user2,fish这个表示仅仅是以read only的方式查看

3.遇到调试问题注意查看/usr/local/nagios/var/nagios.log

4.可以使用sed 替换命令 sed -i‘s#nagiosadmin#lodboy#g’ cgi.cfg

5.reload nagios命令为 /etc/init.d/nagios reload 此处cgi.cfg的修改可以不执行reload

#### 3.9.1.1 nagios排错指南

排错步骤：

思路：1.在服务器端运行nrpe命令查看是否能联通。2.如果不行就转到客户端运行本地命令例如check\_load查看是否可行。3.查看客户端nrpe.cfg配置文件。在本地客户端运行nrpe是否可行。

1.在服务端运行

[root@nagios-server objects]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.163 -p 5666 -c check\_load

其中：

-H 是客户端主机地址，

-p是客户端nrpe服务端口，

-c 是客户端nrpe.cfg配置文件中大括号的命令，是调用客户端命令

2.在客户端运行

1）在客户端运行本地命令

/usr/local/nagios/libexec/check\_load

查看是否有错误，如果没有则继续

2）在客户端上，配置nrpe.cfg文件中为allowed\_host=服务器端IP，客户端IP

在客户端上/usr/local/nagios/libexec/check\_nrpe -H 192.168.80.163 -p 5666 -c check\_load

查看是否有错误，如果没有错误，说明客户端没有问题。

思路：nrpe的监控流程：

1.在服务器端中的commands.cfg文件中定义check\_nrpe命令

2.服务器端使用check\_nrpe命令连接客户端的nrpe服务，客户端使用本地命令，例如check\_load(这个命令是定义在客户端nrpe.cfg中的)搜集客户端信息，并返回给服务器端

3.服务器端使用check\_nrpe得到客户端数据展示在web上

2)模拟nagios配置错误

1.取消check\_memory.pl执行权限

出错提示：

NRPE:Unable to read output

2.nrpe.cfg中命令名字写错：

command[check\_disk12]=/usr/local/nagios/libexec/check\_disk -w 15% -c %7 -p /

出错提示：

NRPE:Command ‘check\_disk’ not defined

### 3.9.2增加从nagios服务器端发起的监控：如URL地址，端口监控【主动监控】

此类服务一般都是客户端开启了对外提供业务服务，这样的业务，我们一般采用主动监控的方法，当然了，我们可以在客户端写脚本，通过服务器端的nrpe进行被动监控。但是一般我们不那么做。

#### 3.9.2.1对域名url地址<http://www.etiantian.org>的监控

1.在command.cfg中添加要使用的命令插件

[root@nagios-server ~]# cd /usr/local/nagios/etc/objects

在command.cfg中定义check\_weburl 命令

define command{

command\_name check\_weburl

command\_line $USER1$/check\_http $ARG1$ -w 10 -c 20

}

2.添加要监控的服务配置到serivice.cfg

define service {

use generic-service

host\_name 162-lamp-1

service\_description www\_url

check\_command check\_weburl!-H www.etiantian.org

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

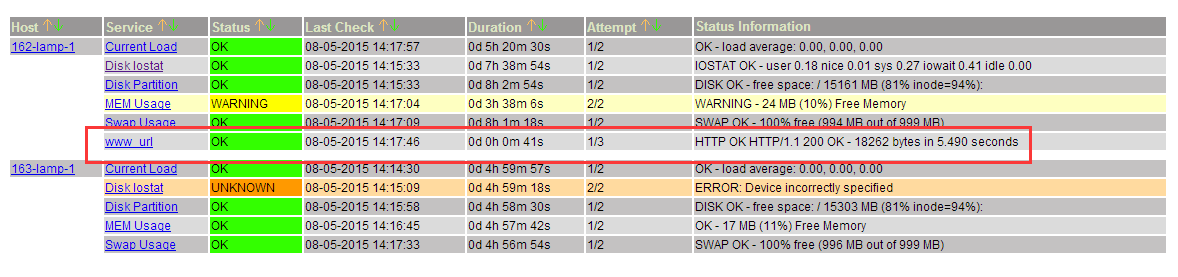
[root@nagios-server objects]# /etc/init.d/nagios checkconfig

[root@nagios-server objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done

登录nagios的web界面查看：www\_url监控项正常



#### 3.9.2.2对域名URL地址<http://www.etiantian.org/about/>

[root@nagios-server ~]# cd /usr/local/nagios/etc/objects

添加要监控的服务配置到serivice.cfg

define service {

use generic-service

host\_name 162-lamp-1

service\_description www\_url\_about

#check\_weburl 这是检查命令，在command.cfg里定义的，后面就是真正检查url地址了，其中-u 为域名后面的地址

check\_command check\_weburl!-H www.etiantian.org -u /about/

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

#检查语法

[root@nagios-server objects]# /etc/init.d/nagios checkconfig

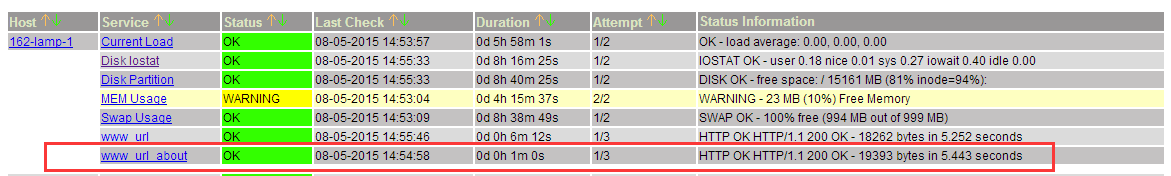
#重新加载配置文件

[root@nagios-server objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done

登录到nagios的web界面查看，监控正常。



#### 3.9.2.3对特殊带传参的URL地址的监控

例如：<http://blog.etiantian.org/article/index.php?m=article&a=list&id=670>这种有特殊字符的监控

define service {

use generic-service

host\_name 162-lamp-1

service\_description www\_url\_about

#check\_weburl 这是检查命令，在command.cfg里定义的，后面就是真正检查url地址了，其中-u 为域名后面的地址。对于特殊带传参的URL地址的监控 -u后面需要加上双引号

check\_command check\_weburl!-H www.etiantian.org -u “article/index.php?m=article&a=list&id=670”

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

#### 3.9.2.4监控任意端口举例：80，110，25，8080，873

在服务器端定义command

修改command.cfg配置文件

# 'check\_tcp' command definition

define command{

command\_name check\_tcp

command\_line $USER1$/check\_tcp -H $HOSTADDRESS$ -p $ARG1$ $ARG2$

}

定义services.cfg

define service {

use generic-service

host\_name 162-lamp-1

service\_description ssh\_22

check\_command check\_tcp!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

norification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

define service {

use generic-service

host\_name 162-lamp-1

service\_description httpd\_80

check\_command check\_tcp!80

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

define service {

use generic-service

host\_name 162-lamp-1

service\_description httpd\_80\_ssh\_22

check\_command check\_tcp!80!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

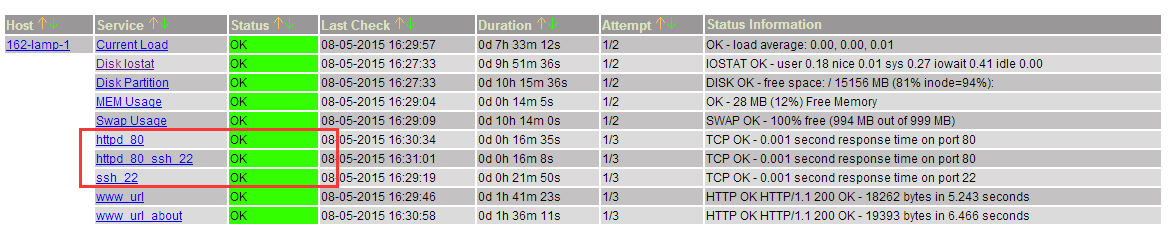
}

[root@nagios-server libexec]# /etc/init.d/nagios checkconfig

[root@nagios-server libexec]# /etc/init.d/nagios reload

#最后一个服务是同时监控80和22端口

注意：端口检查是一个很不多的辅助监控方式，对于业务要求搞的一定要模拟用户监控才好



## 3.10 Nagios的调试

### 3.10.1优化配置nagios启动囧啊本检查nagios语法

[root@nagios-server libexec]# vim /etc/init.d/nagios +178

把#$NagiosBin -v $NagiosCfgFile > /dev/null 2>&1;

修改成：$NagiosBin -v $NagiosCfgFile;

然后使用[root@nagios-server libexec]# /etc/init.d/nagios checkconfig 就可以打印出详细信息。

## 3.11 nagios安装部署过程中的错误集合及解决办法【被动监控】

### 3.11.1 CHECK\_NRPE: Error - Could not complete SSL handshake.

模拟故障：

修改客户端nrpe.cfg配置文件中的

allowed\_hosts=192.168.80.160其中160不是服务器端，然后重启nrpe

然后报错如下：

CHECK\_NRPE: Error - Could not complete SSL handshake.

解决办法：

修改客户端nrpe.cfg配置文件：修改allowed\_host=192.168.80.161

重启客户端nrpe服务

### 3.11.2 NRPE: Unable to read output

模拟故障：

在客户端执行：

[root@nagios-client1 libexec]# chmod -x check\_load

解决办法：

在服务端执行check\_nrpe

[root@nagios-server ~]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.162 -c check\_load

NRPE: Unable to read output

修改客户端nrpe.cfg配置文件，添加allowed\_host 主机地址，具体如下：

[root@nagios-client1 libexec]# vim /usr/local/nagios/etc/nrpe.cfg

allowed\_hosts=192.168.80.161,192.168.80.162

[root@nagios-client1 libexec]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.162 -c check\_load

CHECK\_NRPE: Error - Could not complete SSL handshake.

[root@nagios-client1 libexec]# ./check\_load

-bash: ./check\_load: Permission denied

发现本地插件命令check\_load没有执行权限。

[root@nagios-client1 libexec]# chmod +x check\_load

等待4分钟即可恢复，为什么等待4分钟就好了呢？因为上面services.cfg定义了错误重试时间为4分钟。

### 3.11.3 NRPE: Command 'check\_load' not defined

模拟故障：

在客户端nrpe.cfg中把命令写错。比如我们服务器调用的是check\_load的命令。但是客户端却定义了check\_load1的命令。

[root@nagios-client1 libexec]# vim /usr/local/nagios/etc/nrpe.cfg

command[check\_load1]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20

重启nrpe

[root@nagios-client1 libexec]# /etc/init.d/nrped stop

Terminated

[root@nagios-client1 libexec]# /etc/init.d/nrped start

Starting nrpe... [ OK ]

漫长的等待错误出现，在services.cfg中规定了max\_check\_attempts为2次，retry\_check\_interval 为4分钟。所以得等待8分钟

解决办法：

根据报错说命令没有定义。

在服务端检查：

[root@nagios-server ~]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.162 -c check\_load

NRPE: Command 'check\_load' not defined

在客户端检查：因为上面在客户端的nrpe.cfg中已经定义了allowed\_host允许本机

[root@nagios-client1 libexec]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.162 -c check\_load

NRPE: Command 'check\_load' not defined

[root@nagios-client1 libexec]# ./check\_load

check\_load: Could not parse arguments

Usage:check\_load [-r] -w WLOAD1,WLOAD5,WLOAD15 -c CLOAD1,CLOAD5,CLOAD15

在本地运行命令可以得到结果，我们猜测可能是服务器端中services.cfg中的调用命令写错了。或者是本地命令写错了。所以我们修改任意一处地方即可。我们这里修改客户端命令。

编辑客户端nrpe.cfg。做如下修改：

command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20

然后重启nrpe服务

[root@nagios-client1 libexec]# /etc/init.d/nrped stop

Terminated

[root@nagios-client1 libexec]# /etc/init.d/nrped start

Starting nrpe... [ OK ]

然后，没有然后了。就等吧。等多少分钟呢？上面已经说了。4分钟啊

好了，先写这么些故障吧。例如什么，服务器端check\_nrpe在command.cfg中没有定义。这个依靠服务器端的/etc/init.d/nagios checkconfig 检查命令就可以检查好了。

# 第四部分 nagios图形监控显示和管理【服务端】

## 4.1 Pnp安装图形监控曲线（服务器端）

### 4.1.1 yum安装pnp软件需要的基础包

pnp出图软件官方站点为：<http://www.pnp4nagios.org>

先执行yum安装下面基础pnp软件需要的包，如果有重复的再执行也不会有问题

yum install cairo pango zlib zlib-devel freetype freetype-devel gd gd-devel -y

### 4.1.2 install libart\_lgpl by rrdtool

当然也可以yum install libart\_lgpl libart\_lgpl-devel -y 这里编译方式为例

cd /home/fish/tools/nagios\_gather

wget http://ftp.gnome.org/pub/gnome/sources/libart\_lgpl/2.3/libart\_lgpl-2.3.17.tar.gz

tar xf libart\_lgpl-2.3.17.tar.gz

cd libart\_lgpl-2.3.17

./configure

make && make install

/bin/cp -r /usr/local/include/libart-2.0 /usr/include/

cd ../

### 4.1.3 install rrdtool

wget ftp://ftp.zedz.net/pub/security/network-monitoring/rrdtool/rrdtool-1.2.14.tar.gz

tar xf rrdtool-1.2.14.tar.gz

cd rrdtool-1.2.14

./configure --prefix=/usr/local/rrdtool --disable-python --disable-tcl

make && make install

cd ../

ls -l /usr/local/rrdtool/bin

如果bin下面有下面三行内容，说明rrdtool没有问题

[root@nagios-server nagios\_gather]# ls -l /usr/local/rrdtool/bin

total 116

-rwxr-xr-x 1 root root 55713 Aug 6 10:11 rrdcgi

-rwxr-xr-x 1 root root 6775 Aug 6 10:11 rrdtool

-rwxr-xr-x 1 root root 52707 Aug 6 10:11 rrdupdate

### 4.1.4 install pnp-0.4.14.tar.gz

wget <http://sourceforge.net/projects/pnp4nagios/files/PNP/pnp-0.4.14/pnp-0.4.14.tar.gz/download>

tar xf pnp-0.4.14.tar.gz

cd pnp-0.4.14

./configure \

--with-rrdtool=/usr/local/rrdtool/bin/rrdtool \

--with-perfdata-dir=/usr/local/nagios/share/perfdata/

make all

make install

make install-config

make install-init

cd ..

编译时候遇到错误：

configure: error: Perl Module Time::HiRes not available

解决：

yum install perl-Time-HiRes -y

PNP提供了一个perl脚本，可以用如下命令查到

[root@nagios-server nagios\_gather]# ls /usr/local/nagios/libexec/ | grep process

process\_perfdata.pl

### 4.1.5 nagios出图相关基本配置

1）执行编辑命令 vim /usr/local/nagios/etc/nagios.cfg +835 修改nagios.cfg主配置文件

#-----------------

process\_performance\_data=1 #默认为0，修改为1

#找到如下几项，去掉注释，在此时的848，849行

host\_perfdata\_command=process-host-perfdata

service\_perfdata\_command=process-service-perfdata

2)执行编辑命令 vim /usr/local/nagios/etc/object/commands.cfg +227 修改commands.cfg配置文件。

#修改commands.cfg配置文件，约227-238行

#默认配置为（如果没有直接添加即可）

# 'process-host-perfdata' command definition

define command{

command\_name process-host-perfdata

command\_line /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE$\t$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOSTPERFDATA$\n" >> /usr/local/nagios/var/host-perfdata.out

}

# 'process-service-perfdata' command definition

define command{

command\_name process-service-perfdata

command\_line /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERVICEDESC$\t$SERVICESTATE$\t$SERVICEATTEMPT$\t$SERVICESTATETYPE$\t$SERVICEEXECUTIONTIME$\t$SERVICELATENCY$\t$SERVICEOUTPUT$\t$SERVICEPERFDATA$\n" >> /usr/local/nagios/var/service-perfdata.out

}

删除上面的内容。添加下面的内容

# 'process-host-perfdata' command definition

define command{

command\_name process-host-perfdata

command\_line /usr/local/nagios/libexec/process\_perfdata.pl

}

# 'process-service-perfdata' command definition

define command{

command\_name process-service-perfdata

command\_line /usr/local/nagios/libexec/process\_perfdata.pl

}

也可以用nagios变量$USER1$替代/usr/local/nagios/libexec/路径

特别注意：上面的配置和下面的配置是等价的

# 'process-host-perfdata' command definition

define command{

command\_name process-host-perfdata

command\_line $USER1$/process\_perfdata.pl

}

# 'process-service-perfdata' command definition

define command{

command\_name process-service-perfdata

command\_line $USER1$/process\_perfdata.pl

}

3）检查配置文件

[root@nagios-server objects]# /etc/init.d/nagios checkconfig

Total Warnings: 0

Total Errors: 0

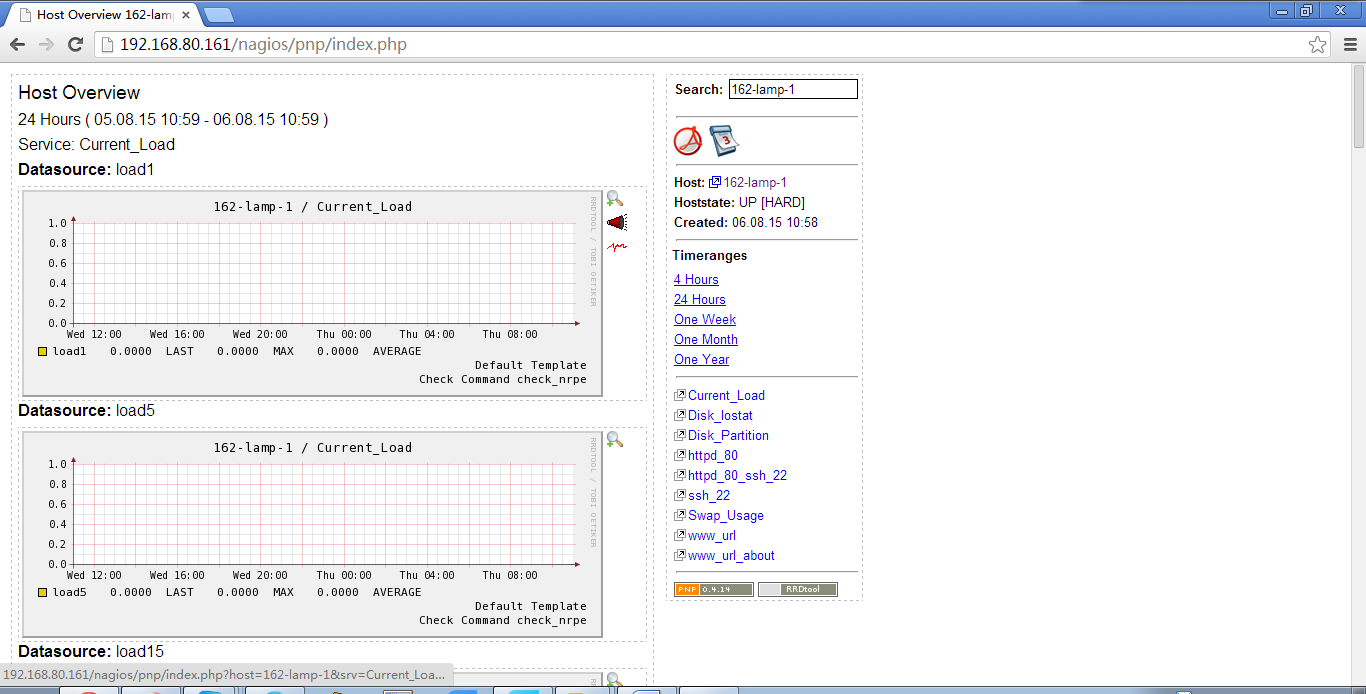
如果配置文件检查没问题。重新加载nagios配置文件，使配置文件生效。

[root@nagios-server objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done

4）在浏览器中输入 <http://192.168.80.161/nagios/pnp/index.php> 打开页面如下



到这里位置，pnp软件的出图就OK了

## 4.2配置各个服务出监控状态图

### 4.2.1服务出图记录数据设置

执行编辑命令 vim services.cfg 在各个服务配置里加process\_perf\_data 1

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description MEM Usage

check\_command check\_nrpe!check\_mem

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

### 4.2.2主机出图记录设置

编辑 vim hosts.cfg配置文件

define host {

use linux-server

host\_name 163-lamp-1

alias 163-lamp-1

address 192.168.80.163

check\_command check-host-alive

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 2

check\_period 24x7

notification\_interval 300

notification\_period 24x7

notification\_options d,u,r

contact\_groups admins

process\_perf\_data 1

}

## 4.3整合pnp url超链接到nagios图形显示界面里

提示：这会在主机或对应服务前面，出现一个红太阳图标，点击就是对应的服务图形监控状态

### 4.3.1给主机添加红太阳超链接图标

默认情况pnp的url为<http://192.168.80.161/nagios/pnp/index.php> 和nagios不再一个界面里。所以查看服务对应的趋势去比较费劲。下面开始整合到一起。

**方法1：推荐**

vim /usr/local/nagios/etc/objects/host.cfg 加入如下内容

#graphic for hosts create by fish!

define host {

use linux-server

host\_name 162-lamp-1

alias 162-lamp-1

address 192.168.80.162

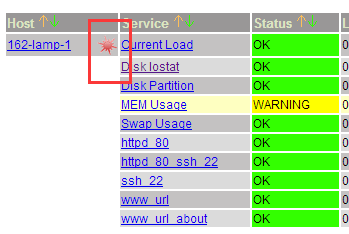
action\_url /nagios/pnp/index.php?host=$HOSTNAME$

}

检查语法并重新加载nagios

/etc/init.d/nagios checkconfig

/etc/init.d/nagios reload



一个大大的太阳看到了不，猛击这个大太阳，就可以超链接到pnp出图地址。

注意：这个的大太阳也可以修改成自己的头像，如果看着不顺眼的话。

**方法2：不推荐**

直接在host.cfg里加参数

define hostextinfo {

host\_name 162-lamp-1,163-lamp-1

action\_url /nagios/pnp/index.php?host=$HOSTNAME$

}

如果host\_name项有多个主机，可以逗号隔开

### 4.3.2给指定的服务添加红太阳超链接图标

**方法1：不推荐**

vim /usr/local/nagios/etc/objects/services.cfg 切到结尾添加如下内容

define serviceextinfo {

host\_name 162-lamp-1

service\_description Swap Usage

action\_url /nagios/pnp/index.php?host=$HOSTNAME$&ser=$SERVICEDESC$

}

define serviceextinfo {

host\_name 162-lamp-1

service\_description http\_80

action\_url /nagios/pnp/index.php?host=$HOSTNAME$&ser=$SERVICEDESC$

}

**方法2：推荐**

提示：也可以直接在define service中加，如

define service {

use generic-service

host\_name 162-lamp-1,163-lamp-1

service\_description Current Load

check\_command check\_nrpe!check\_load

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1 action\_url /nagios/pnp/index.php?host=$HOSTNAME$&ser=$SERVICEDESC$

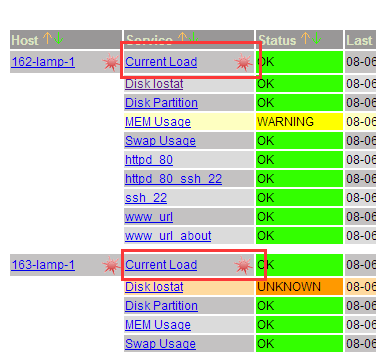
}

提示：重要，当然可以写成服务模版了

#检查语法并重新加载配置nagios配置文件

/etc/init.d/nagios checkconfig

/etc/init.d/nagios reload



看到小太阳了吧

**4.3.4出图的图像数据存放路径**

[root@nagios-server objects]# ll /usr/local/nagios/share/perfdata/

total 8

drwxr-xr-x 2 nagios nagios 4096 Aug 6 13:34 162-lamp-1

drwxr-xr-x 2 nagios nagios 4096 Aug 6 13:35 163-lamp-1

# 第五部分 报警方式及生产报警策略

## 5.1 报警的种类及报警方式选用

### 5.1.1报警的种类

1）邮件报警，在生产环境应尽量使用公司的信箱作为报警信箱，或者建立一个邮件组，[如sa@etiantian.org](mailto:如sa@etiantian.org)。尽量不用非公司信箱作为报警信箱，如126，qq信箱，因为这些信箱都是免费的，对报警的频率都会有限制，很可能会拒收或当垃圾邮件了。

2)飞信：需要在win32上装个飞信客户端，把对方手机加为好友，然后就可以给对方发短信了

3）邮件转短信报警，如139，126，189信箱，信箱里邮件到达后，通过手机通知收件人是邮箱提供商提供的一个功能。

4）http短信网关

5）购买短信猫

6）电话语音报警，在报警时直接电话给报警负责人

7）MSN,QQ及时通信报警，模拟QQ,MSN发消息的功能，网友们开发了程序，从命令行执行程序利用MSN,QQ协议，直接发给MSN和QQ好友。

### 5.1.2短信报警的种类

短信报警是当前最重要的及时报警方式

1. 飞信，装个飞信客户端 把对方手机加为好友（需要对方确认），然后就可以给对方发短信了
2. 邮件转短信报警，如139，126，189信箱，信箱邮件通知手机的一个功能
3. http短信网关（收取发送短信费）（每条短信大概7分钱）
4. 短信猫

建议使用http短信网关

### 5.1.3生产环境使用哪种报警？

在生产环境中，一般会根据业务的紧急程度不同，多个报警策略结合使用，对于不需要紧急处理的业务一般选择邮件报警，如内存，磁盘空间的剩余率，对于重要且紧急的业务，会使用邮件加上短信同时报警，使用邮件报警便于记录故障详细信息，短信报警是及时提醒，短信的有点是及时，而邮件报警如果人不再电脑旁边就没办法知晓了。

短信报警的缺点是报警内容有限，所以在工作中如果接到严重报警时，我们在紧急处理之前也会开启邮件系统先查看邮件细节。

邮件和短信报警是当前监控报警的两大主题。

其中http短信网关是比较好的短信报警方式。原因

1. 简单，易用
2. 稳定，可靠
3. 收费合理

会一定的费用，把业务做到最好，是工作的思维，如果总想着免费，那么如果业务报警报不出来，损失可能会更大，所以选择报警产品，和领导说清楚利弊，让他们去选择，这样即使没选，出问题了，也不都是你的责任，当然，如果业务不是特别重要，或者对你选择的免费产品有信心不会出问题，你也可以尝试，但这不推荐，正规的公司的业务报警应尽量选择可靠的报警。

### 5.1.4故障报警分级

运维工程师值班职责（摘录）

第1条值班报警（故障）分类：

A类：磁盘空间，CPU，内存报警为一般报警，运维内部采取常规处理方式。

B类：网站域名不能打开为严重报警，需要写条技术部相关人员会诊处理

第2条值班配有两部值班手机，遇运维问题将报警

1 若收到A类报警，原则上不限制处理时间，但以不影响服务为前提，进行及时处理。

2 若收到B类报警短信，值班人员需在10分钟内邮件通知运维全体同时及相关技术人员，并解决。

5.1.5 7\*24 值班在线职守

上面我们讲解的是中小公司，无7\*24在线职守的情况下的，报警方式及策略，对于一些游戏公司或者门户网站或者较大公司，如阿里巴巴，蓝汛等，还会有专职的值班人员进行7\*24小时监控职守监控（查看nagios，cacti，mrtg，hyperic等界面）一旦发生问题，由值班人员进行判断并报警给相关负责人，7\*24小时监控，需要额外增加人力成本，一般中小公司采用的不多。

## 5.2实战报警实施细节

### 5.2.1邮件报警

#### 5.2.1.1普通邮件报警

普通邮件报警，就是在故障发生或者恢复时，通过邮箱报警到系统管理员或相关维护人员信箱中，一般来说信箱最好为公司邮箱。

#### 5.2.1.2邮件报警适合的服务

一般白天上班时，有i吉安报警还算比较及时，但是如果人员不在电脑胖，邮件报警就不行了，因此，用于保健报警的一般为不是特别重要的业务，或者作为接受大量信息报警内容时的一个辅助的报警方式，如硬盘，内存，及日志相关等不需要及时解决的服务报警，在生产环境中，邮件报警一般会结合其他报警方式同时使用，作为一个故障的记录工具。

#### 5.2.1.3邮件报警的基本配置方法

来自1.7添加监控的接受email地址

sed -i 's#nagios@localhost#linkagetest@126.com#' /usr/local/nagios/etc/objects/contacts.cfg

#提示：

#1.此功能依赖本机的sendmail服务，可执行service sendmail start 开启。关闭postfix服务，因为会和sendmail服务冲突

[root@nagios-server ~]# /etc/init.d/postfix stop

Shutting down postfix: [ OK ]

[root@nagios-server ~]# chkconfig postfix off

[root@nagios-server ~]# /etc/init.d/sendmail start

[root@nagios-server ~]# chkconfig sendmail on

define contact{

contact\_name nagiosadmin #定义联系人名字

use generic-contact #定义使用的模版

alias Nagios Admin #定义别名

email linkagetest@126.com #定义邮箱地址

}

define contactgroup{

contactgroup\_name admins #定义联系组名字，我们在服务中一般调用这个联系组的名字

alias Nagios Administrators #定义别名

members nagiosadmin #定义联系组中的成员

}

默认的报警配置在command.cfg里

# 'notify-host-by-email' command definition

define command{

command\_name notify-host-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ \*\*" $CONTACTEMAIL$

}

# 'notify-service-by-email' command definition

define command{

command\_name notify-service-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ \*\*" $CONTACTEMAIL$

}

templates.cfg中有关邮件报警的联系人模版配置【默认已配置，如果配置飞信，MSN等要追加命令】

define contact{

name generic-contact #在contact.cfg文件中被调用的

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r,f,s

host\_notification\_options d,u,r,f,s

service\_notification\_commands notify-service-by-email #定义了发邮件的命令，调用command.cfg文件中定义发送邮件的命令

host\_notification\_commands notify-host-by-email

register 0

}

#### 5.2.1.4生产环境邮件报警配置

vim commands.cfg

# 'notify-host-by-email' command definition

define command{

command\_name notify-host-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s “Host $HOSTSTATE$ Alert for $HOSTNAME$!" $CONTACTEMAIL$

}

#主机报警内容标题： Host DOWN alter for 162-lamp-1

# 'notify-service-by-email' command definition

define command{

command\_name notify-service-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "$HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ \*\*" $CONTACTEMAIL$

}

#服务报警的邮件的标题：162-lamp-1/http\_80 is CRITICAL

#### 5.2.1.5邮件转短信报警

这里的服务提供商目前很多，如126，139，189等信箱都支持

配置方法和邮件报警一样，只不过在注册信箱后要开启邮箱邮件到达短信提醒的功能（设置白名单）。邮件转短信报警的稳定性一般，且信息的容量有很大限制，因此，建议作为辅助报警方式。

#### 5.2.1.6联系人分级报警生产策略

来自3.6 contacts.cfg生产库实战配置例子写法

#手机短信用户

define contact{

contact\_name shouji\_jeacen

use generic-contact

alias nagios admin

email 134521902211@139.com

}

define contact{

contact\_name shouji\_wodi

use generic-contact

alias nagios admin

email 13452222211@139.com

}

#邮件用户及msn帐号

define contact{

contact\_name msn\_fish

use generic-contact

alias nagios admin

email [134521211@139.com](mailto:134521211@139.com)

address1 rr@hotmail.com

}

define contact{

contact\_name msn\_fish2

use generic-contact

alias nagios admin

email [1345212231@139.com](mailto:1345212231@139.com)

address1 rsdfds@hotmail.com

}

#仅仅是邮件用户

define contact{

contact\_name youjian\_fish

use generic-contact

alias nagios admin

email [134521211@139.com](mailto:134521211@139.com)，linkageinfo@126.com

}

define contact{

contact\_name youjian\_fish2

use generic-contact

alias nagios admin

email [134521211@139.com](mailto:134521211@139.com)，linkageitest@126.com

}

#手机组，手机组包含上面的两个手机成员

define contactgroup{

contactgroup\_name shoujiusers

alias nagios administrator

members shouji\_jeacen, shouji\_wodi

}

#邮件及msn帐号组

define contactgroup{

contactgroup\_name msnusers

alias nagios administrator

members youjian\_fish, msn\_fish2

}

#仅邮件组

define contactgroup{

contactgroup\_name msnusers

alias nagios administrator

members youjian\_fish, youjian\_fish2

}

#### 5.2.1.7正确的报警思想

工作中的报警需要遵循一个原则“该报警的一定要报出来，不该报警的就一定不要报出来”

# 5.3飞信软件的报警

飞信报警的实现原理是：首先在个人电脑上（非报警服务器）装个飞信客户端，然后注册一个用于报警的发送人帐号。需要绑定手机，然后，登录飞信客户端，把需要接受报警的所有手机加为好友（需要对方确认），然后就可以使用发送人帐号给对方发短信了，服务器报警使用其命令行模式的发送方式。

### 5.3.1飞信软件报警配置过程细节

#### 5.3.1.1本地上传飞信包（也可以网上下载）

飞信机器人程序地址：<http://bbs.it-adv.net>

#### 5.2.1.4 MSN软件报警

MSN报警的实现原理是：首先在个人电脑上（非报警服务器）装个MSN客户端，然后注册一个用于报警MSN发送人帐号，然后，登录MSN客户端，把需要接收报警的所有MSN帐号加为好友（需要对方确认），然后就可以使用MSN发送人帐号给所有指定MSN好友发送报警信息了。服务器的MSN报警使用其命令行模式的发送方式，即程序方式发送MSN消息。

#### 5.2.1.5 MSN软件报警配置过程细节

5.2.1.5.1安装phpmsnclass软件

#### 5.2.1.6 http短信网关接口报警

经过多年的对nagios报警的使用，最为推荐的是http短信网关接口，有专门的公司提供直接发给信息到手机的短信网关，常用的报警就是一个url地址携带信息。

推荐原因：

1. 收费合理，报警比较及时服务有保证
2. 配置简单，一个脚本搞定

还有很多公司为了省钱，使用139和飞信等短信报警方式，这个是不推荐的。当购买短信服务后，除了有界面可以发送报警（一般市场人员用）外，还会有一个url地址，地址后面携带帐号，密码，及报警信息。url地址如下面的形式：

[http://s.ccme.cc/qx/send.jsp?circle=fish%pwd-fish123&](http://s.ccme.cc/qx/send.jsp?circle=oldboy%25pwd-oldboy123&)

mobile=$CONTACT&service=abcd546-eee6-gg69-gg40-3gg0524clf88d%msgid=2322&message=$TITLE[${alert\_date} sa]

在这个地址中的信息如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 用户名 | 验证密码 | 目标手机 | 消息内容 |
| fish | fish123 | $CONTACT,脚本变量 | $TITLE[${alert\_date} sa],消息变量及时间变量 |

实际发送命令演示：

#curl 方式

curl -d cdkey-3RTV-EMY-0980-MTUQ2 -d password=189162 -d phone=$CONTACT -d message=” $TITLE[${alert\_date} fishsa]” <http://sdkhttp.eucp.b2m.cn/sdkproxy/sendsms.action>

#wget --quiet “http://s.ccme.cc/qxt/send.jsp?circle=159net\_131&pwd=fish&mobile=1891178221&service=f1fb0546-ebb6-0987-8f20-560524c1f88d&msgid=395672d&message=$TITLE[${alert\_date} fishsa n]”

#### 5.2.1.7短信网关设备报警细节（在监控服务器上）

1）开发报警脚本sms\_send防与libexec下面，授权755

[root@realserver2 libexec]# pwd

/usr/local/nagios/libexec

[root@realserver2 libexec]# chmod 755 sms\_send

[root@realserver2 libexec]# cat sms\_send

#!/bin/bash

PROGNAME=`basename $0`

PROGPATH=`echo $0 | sed -e 's,[\\/][^\\/][^\\/]\*$,,'`

print\_usage(){

echo "Usage:"

echo "/bin/sh $PROGNAME title contact"

exit 1

}

if [ $# -ne 2 ];then

print\_usage

fi

alter\_date=$(date +%y-%m-%d" "%H:%M)

TITLE=$1 #PROMT "Host $HOSTSTATE$ alter for $HOSTNAME$"

CONTACT=$2

#curl mode

curl -d cdkey-3RTV-EMY-0980-MTUQ2 -d password=189162 -d phone=$CONTACT -d message=?$TITLE[${alert\_date} fishsa]" http://sdkhttp.eucp.b2m.cn/sdkproxy/sendsms.action

#wget ----quiet [http://s.ccme.cc/qxt/send.jsp?circle=159net\_131&pwd=fish&mobile=1891178221&service=f1fb0546-ebb6-0987-8f20-560524c1f88d&msgid=395672d&message=$TITLE[${alert\_date} fishsa n]](http://s.ccme.cc/qxt/send.jsp?circle=159net_131&pwd=oldboy&mobile=1891178221&service=f1fb0546-ebb6-0987-8f20-560524c1f88d&msgid=395672d&message=$TITLE%5b$%7balert_date%7d%20oldboysa%20n%5d)

测试脚本：

./sms\_send “this is a test” 13451902202

2）在command.cfg中定义报警命令

#command.cfg

# ‘notify-host-by-pager’ command definition

define command{

command\_name notify-host-by-pager

command\_line $USER1/sms\_send “Host $HOSTSTATE$ alter for $HOSTNAME$” $CONTACTPAGER$

}

# ‘notify-host-by-pager’ command definition

define command{

command\_name notify-service-by-pager

command\_line $USER1/sms\_send “$HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$” $CONTACTPAGER$

}

3)在模版templates.cfg中增加如下配置

define contact{

name generic-contact

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r,f,s

host\_notification\_options d,u,r,f,s

service\_notification\_commands notify-service-by-email,notify-host-by-pager

host\_notification\_commands notify-host-by-email,notify-service-by-pager

register 0

}

~

4）在联系人contact.cfg里做如下联系人定义

define contactgroup{

contactgroup\_name fish-pager

alias generic-contact

members Nagios users

pager 13451902203

}

define contactgroup{

contactgroup\_name admins

alias Nagios Administrators

members nagiosadmin,oldbou-pager

}

这样就可以实现通过http短信网关报警了

# 第六部分 nagios插件开发

## 6.1什么是nagios插件

我们在前文部署nagios服务时安装了nagios-plugins-1.4.3.tar.gz，这个软件包就是nagios的插件安装包，安装后，我们执行 ls -l /usr/local/nagios/libexec可以看到如下插件内容

[root@realserver2 ~]# ll /usr/local/nagios/libexec/

total 4648

-rwxr-xr-x 1 nagios nagios 348425 Aug 3 14:56 check\_apt

-rwxr-xr-x 1 nagios nagios 2265 Aug 3 14:56 check\_breeze

-rwxr-xr-x 1 nagios nagios 92661 Aug 3 14:56 check\_by\_ssh

lrwxrwxrwx 1 root root 9 Aug 3 14:56 check\_clamd -> check\_tcp

-rwxr-xr-x 1 nagios nagios 50881 Aug 3 14:56 check\_cluster

-r-sr-xr-x 1 root nagios 90329 Aug 3 14:56 check\_dhcp

-rwxr-xr-x 1 nagios nagios 87781 Aug 3 14:56 check\_dig

-rwxr-xr-x 1 nagios nagios 390562 Aug 3 14:56 check\_disk

-rwxr-xr-x 1 nagios nagios 8163 Aug 3 14:56 check\_disk\_smb

-rwxr-xr-x 1 nagios nagios 96214 Aug 3 14:56 check\_dns

-rwxr-xr-x 1 nagios nagios 46268 Aug 3 14:56 check\_dummy

-rwxr-xr-x 1 nagios nagios 3143 Aug 3 14:56 check\_file\_age

以上都是nagios插件，其实nagios本身仅仅是一个监控的平台，如果要监控具体的主机及服务的状态和数据信息，还必须要配置或调用插件或程序文件才能完成任务。因此，如果没有nagios插件，nagios就是一个空壳，啥都做不了。

## 6.2为什么要开发nagios插件

既然已经安装了插件软件包，为什么还要开发nagios插件呢。

授权说生产尝尽中常用的大部分服务，都是不需要编写插件就可以完成监控的，如check\_http,check\_tcp,check\_nrpe，这些自带的插件已经很强大了，但是，让然有部分我们想要监控的服务，nagios没有自带的插件，如监控LVS中RS 的 lo网卡的VIP，如监控NFS的状态，再比如监控iostat，mem，sar系统指标及相关app应用（MQ队列），这个时候我们有两个选择，一个是去网上去google，看看别人写过的，哪来使用或修改后使用，另外一个方面是自己开发编写，这里需要学会手工编写插件，如果开始不会写，可以先从网上分享别人写好的。

## 6.3编写nagios插件规则

### 6.3.1编写nagios规则说明

Nagios插件是nagios提供的一种通过可扩展的方式部署的程序组件，该插件可通过Java，C\C++,php等多种语言开发，操作员通过修改配置和响应参数，就能很方便的将插件集成到nagios中，实现对目标系统的监控。

nagios插件程序提供两个返回值，一个是插件的退出状态码，一个是插件在控制台打印的第一行数据，退出状态码可以被nagios主程序作为判断被监控系统服务状态的依据，控制台打印的第一行数据可以被nagios主程序作为被监控系统服务状态的补充说明，会显示在管理页面上面。

为了管理nagios插件，nagios每查询一个服务的状态时，就会产生一个子进程，并且它使用来自该命令的输出和退出代码来确定具体的状态，nagios主程序可识别的状态码和说明如下：

OK --退出代码 0--表示正常的工作

WARNING --退出代码 1--表示服务处于警告状态

CRITICAL --退出代码 2--表示服务处于危险状态

UNKNOW --退出代码 3--表示服务器处于未知状态

最后一种状态通常表示该插件无法确定服务的状态，例如，可能出现内部错误

相关状态我们可以从如下文件中看到

[root@realserver2 libexec]# head -7 utils.sh

#! /bin/sh

STATE\_OK=0

STATE\_WARNING=1

STATE\_CRITICAL=2

STATE\_UNKNOWN=3

STATE\_DEPENDENT=4

### 6.3.2 nagios插件开发原理

nagios插件程序中需要调用监控服务规定的操作序列，并根据预先定义的规则，对返回结果进行分析，判断服务的当前状态，并以指定的状态码退出程序，同时将对该状态的说明不换行输出到控制台。

不同语言的系统退出函数示例如下：

python sys.exit(in status)

bash exit int status

控制台打印函数示例如下：

python print msg

bash echo msg

## 6.4 nagios插件开发语言

nagios的插件开发不限制任何开发语言，只要该插件能被nagios调用获取到响应服务数据就OK，一般来说，如能在命令行执行输出结果也可以，常用的插件语言有，shell，perl，python，php，c

## 6.5 使用shell开发nagios插件

### 6.5.1编写检查weburl地址的插件

#### 6.5.1.1编写nagiois插件前的几个变量测试

a.测试变量的脚本

[root@realserver2 etc]# cat /usr/local/nagios/etc/fish.sh

#!/bin/bash

echo $0

PROGNAME=`basename $0`

#PROGPATH=`echo $0 | sed -e 's,[\\/][^\\/][^\\/]\*$,,'`

PROGPATH=`dirname $0`

echo $PROGNAME

echo $PROGPATH

b.执行fish.sh脚本

[root@realserver2 etc]# bash /usr/local/nagios/etc/fish.sh

/usr/local/nagios/etc/fish.sh #这个是$0

fish.sh #这个是PROGNAME

/usr/local/nagios/etc #这个是PROGPATH,脚本的完整路径，但不包含名称

#### 6.5.1.2监控weburl的插件脚本

[root@realserver2 libexec]# cat check\_url.sh

#!/bin/bash

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

# this script function is: #

# check\_url #

# Create by fish 2015-8-12 #

# mail:45yujing@sina.com qq:244647158 #

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

PROGNAME=`basename $0`

PROGPATH=`dirname $0`

source $PROGPATH/utils.sh

print\_usage() {

echo "Usage:"

echo "/bin/sh $0 url"

echo "For example:"

echo "/bin/sh $0 http://www.baidu.com"

exit 1

}

if [ $# -ne 1 ];then

print\_usage

fi

if wget -T 20 --spider $1 >/dev/null 2>&1 ; then

echo 'HTTP/1.1 OK'

exit $STATE\_OK

else

echo 'HTTP/1.1 NO'

exit $STATE\_CRITICAL

fi

执行check\_url.sh脚本

[root@realserver2 libexec]# chmod 755 check\_url.sh

[root@realserver2 libexec]# sh /usr/local/nagios/libexec/check\_url.sh

Usage:

/bin/sh /usr/local/nagios/libexec/check\_url.sh url

For example:

/bin/sh /usr/local/nagios/libexec/check\_url.sh <http://www.baidu.com>

[root@realserver2 libexec]# /usr/local/nagios/libexec/check\_url.sh http://www.baidu.com

HTTP/1.1 OK

[root@realserver2 libexec]# bash check\_url.sh http://baidu-.com

HTTP/1.1 NO

### 6.5.2 weburl插件脚本部署过程（主动监控方式）

#### 6.5.2.1在nagios服务器端的部署

1）上传监控脚本到指定目录并授权可执行权限

[root@realserver2 libexec]# cd /usr/local/nagios/libexec #山川check\_url.sh到此目录

[root@realserver2 libexec]# chmod +x check\_url.sh

2）手工测试脚本执行结果

[root@realserver2 libexec]# /usr/local/nagios/libexec/check\_url.sh http://192.168.80.162

HTTP/1.1 OK

#### 6.5.2.2修改nagios服务端的配置文件

1)编辑commands.cfg

[root@realserver2 ~]# cd /usr/local/nagios/etc/objects

define command{

command\_name check\_url # 给写好的插件起个名字

command\_line $USER1$/check\_url.sh $ARG1$ #这里是将脚本改为传URL参数的commands.cfg中的部署

}

2）编辑services.cfg

[root@realserver2 ~]# cd /usr/local/nagios/etc/objects

define service {

use generic-service

host\_name 162-lamp-1

service\_description check\_url\_test

check\_command check\_url! <http://192.168.80.162> #脚本改为传参

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

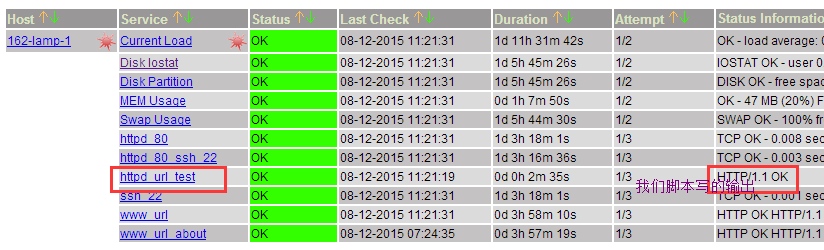
contact\_groups admins

}

检查语法，重新加载配置文件：

[root@realserver2 objects]# /etc/init.d/nagios checkconfig

[root@realserver2 objects]# /etc/init.d/nagios reload



### 6.5.3编写检查mysql服务的插件

#### 6.5.3.1使用check\_mysql插件被动方式监控MYSQL服务的配置过程

1)在nagios客户端进行操作（被监控的mysql服务器）**在客户端操作**

[root@nagios-client1 html]# /etc/init.d/mysqld start

#创建用于监控的授权帐号

mysql> grant select on test.\* to passivemonitor@'localhost' identified by 'fish';

mysql> create database test;

提示：

1.用被动方式监控MySQL服务时，授权可以尽量小，比如只给localhost

2)进入nagios libexec插件目录，测试mysql\_check插件

[root@nagios-client1 ~]# /usr/local/nagios/libexec/check\_mysql -upassivemonitor -p fish -H localhost

Uptime: 78069 Threads: 3 Questions: 14 Slow queries: 0 Opens: 19 Flush tables: 1 Open tables: 12 Queries per second avg: 0.0

3) 编辑nrpe.cfg

[root@nagios-client1 ~]# vim /usr/local/nagios/etc/nrpe.cfg

command[check\_mysql]= /usr/local/nagios/libexec/check\_mysql -upassivemonitor -p fish -H localhost

4)重启nrpe

[root@nagios-client1 ~]# /etc/init.d/nrped stop

Terminated

[root@nagios-client1 ~]# /etc/init.d/nrped start

Starting nrpe... [ OK ]

**6.5.3.2在nagios服务端进行操作**

1）添加hosts.cfg主机，上面已经添加完毕

2) 添加services目录下的check\_mysql.cfg

[root@realserver2 ~]# cd /usr/local/nagios/etc/objects/services

[root@realserver2 services]# vim check\_mysql.cfg

define service {

use generic-service

host\_name 162-lamp-1

service\_description check\_mysql-162

check\_command check\_nrpe!check\_mysql

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

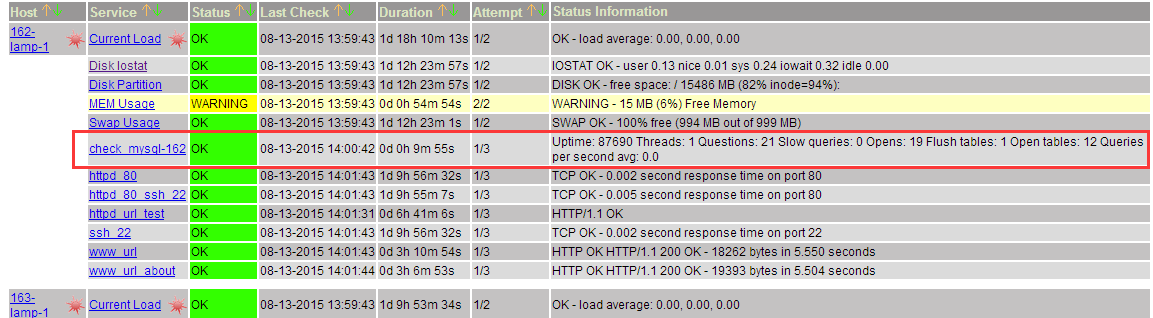
3)检查语法，重新加载配置文件

[root@realserver2 services]# /etc/init.d/nagios checkconfig

[root@realserver2 services]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done



**6.5.3.3被动监控方式遇到问题判断及检查办法**

首先可在 [客户端] 执行如下命令检查

[root@nagios-client1 ~]# /usr/local/nagios/libexec/check\_mysql -upassivemonitor -p fish -H localhost

Uptime: 78069 Threads: 3 Questions: 14 Slow queries: 0 Opens: 19 Flush tables: 1 Open tables: 12 Queries per second avg: 0.0

如果结果没问题，即可以正常获取数据，然后在nagios服务器端通过check\_nrpe检查

[root@realserver2 services]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.162 -c check\_mysql

Uptime: 88167 Threads: 1 Questions: 25 Slow queries: 0 Opens: 19 Flush tables: 1 Open tables: 12 Queries per second avg: 0.0

说明：这里的check\_mysql就是调用客户端的nrpe中的配置。192.168.80.162为客户端的IP地址

如果没有结果，可能服务器端的配置有问题，这样问题就能被最小化的定位了，排查起来就方便了

### 6.5.4使用chcek\_mysql插件主动方式监控MySQL服务的配置过程

提示：因为是主动方式监控，因此以下完全在nagios服务器端进行操作

在客户端授权mysql允许nagios服务器端连接

mysql> grant select on test.\* to passivemonitor@'192.168.80.%' identified by 'fish';

1）首先调试check\_mysql命令

[root@realserver2 libexec]# ./check\_mysql -upassivemonitor -p fish -H 192.168.80.162

Uptime: 91836 Threads: 2 Questions: 60 Slow queries: 0 Opens: 19 Flush tables: 1 Open tables: 12 Queries per second avg: 0.0

提示：请事先安装并启动mysql服务

1. 在命令行调试正确后，编辑command.cfg，定义mysql服务检查命令

[root@realserver2 ~]# cd /usr/local/nagios/etc/objects

[root@realserver2 objects]# vim commands.cfg

define command{

command\_name check\_mysql

#command\_line $USER1$/check\_mysql -upassivemonitor -pfish -H 192.168

.80.162

command\_line $USER1$/check\_mysql $ARG1$ $ARG2$ $ARG3$

}

4)编辑 services目录下的check\_mysql.cfg配置文件

define service {

use generic-service

host\_name 162-lamp-1

service\_description check\_mysql-162-zhudong

# check\_command check\_mysql

check\_command check\_mysql! -upassivemonitor!-pfish!-H 192.168.80.162

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

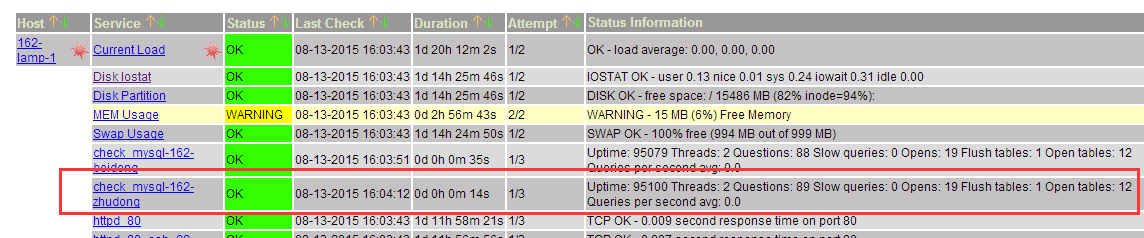
5)检查语法，重新加载配置文件

[root@realserver2 objects]# /etc/init.d/nagios checkconfig

[root@realserver2 objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done



### 6.5.5手工开发脚本监控mysql服务的配置方法

#### 6.5.5.1手工开发脚本监控mysql服务

1）check\_mysql.sh 通用脚本

[root@realserver2 libexec]# cat check\_mysql.sh

#!/bin/bash

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

# this script function is: #

# check\_mysql #

# Create by fish 2015-8-12 #

# mail:45yujing@sina.com qq:244647158 #

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

PROGNAME=`basename $0`

PROGPATH=`dirname $0`

source $PROGPATH/utils.sh

print\_usage() {

echo "Usage:"

echo "/bin/sh $PROGNAME ip"

echo "For example:"

echo "/bin/sh $PROGNAME 192.168.80.162"

exit 1

}

if [ $# -ne 1 ];then

print\_usage

fi

if mysql -upassivemonitor -pfish -h $1 -e "select version();" > /dev/null 2>&1;then

echo 'Mysql service is OK'

exit $STATE\_OK

else

echo 'Mysql service is CRITICAL'

exit $STATE\_CRITICAL

fi

提示：在mysql数据授权时需要授权passivemonitor用户连接nagios客户端mysql数据库

不能落下-e “select version();” ，否则，会进入到mysql登录状态进行交互，导致脚本中断不能执行

[root@realserver2 libexec]# chmod 755 check\_mysql.sh

[root@realserver2 libexec]# bash check\_mysql.sh 192.168.80.162

Mysql service is OK

**6.5.5.2手工开发脚本check\_mysql.sh被动模式监控部署过程**

1）编辑nagios client nrpe.cfg 加入如下内容

command[check\_mysql.sh]=/usr/local/nagios/libexec/check\_mysql.sh 192.168.80.162

2)nagios client重启nrpe服务

[root@nagios-client1 libexec]# /etc/init.d/nrped stop

Terminated

[root@nagios-client1 libexec]# /etc/init.d/nrped start

Starting nrpe... [ OK ]

1. nagios服务端通过check\_nrpe调用check\_mysql.sh

define service {

use generic-service

host\_name 162-lamp-1

service\_description check\_mysql-162-shougong-beidong

check\_command check\_nrpe!check\_mysql

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

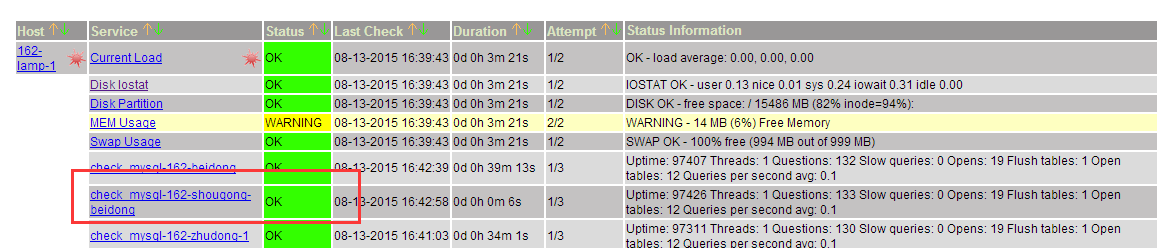
notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}



**6.5.5.3手工开发脚本check\_mysql.sh主动模式监控部署过程**

1）编辑commands.cfg

define command{

command\_name check\_mysql.sh

command\_line $USER1$/check\_mysql.sh $ARG1$

}

2）编辑services目录下的check\_mysql.cfg

define service {

use generic-service

host\_name 162-lamp-1

service\_description check\_mysql-162-shougong-zhudong

check\_command check\_mysql.sh!192.168.80.162

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

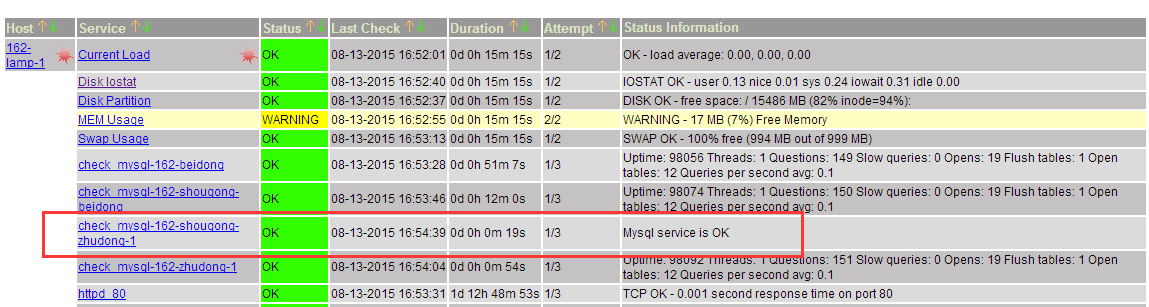
1. 检查配置语法，重新加载配置文件

[root@realserver2 objects]# /etc/init.d/nagios checkconfig

[root@realserver2 objects]# /etc/init.d/nagios reload

Running configuration check...done.

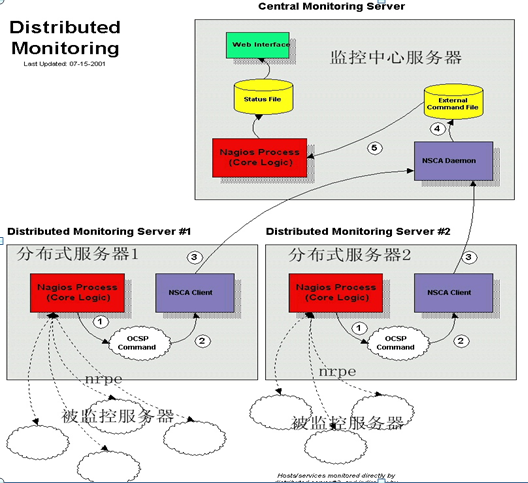
Reloading nagios configuration...done



# 第七部分 nagios分布式部署

## 7.1 nagios角色划分

|  |  |
| --- | --- |
| 角色 | 说明 |
| 监控中心服务器 | 通过NSCA获取分布式监控服务器的相关状态，呈现相关服务器状态和发出报警等 |
| 分布式服务器 | 通过对被监控服务器状态采集并且把被监控服务器的状态通过NSCA\_send发送给监控中心服务器 |
| 被监控服务器 | 被监控服务器就是生产环境服务器 |



## 7.2详细部署说明

### 7.2.1安装软件说明

|  |  |  |
| --- | --- | --- |
| 角色 | 测试服务器IP地址 | 安装的软件包 |
| 监控中心服务器 | 192.168.80.161 | nagios-3.2.1.tar.gz，nagios-plugins-1.4.13.tar.gz，nrpe-2.12.tar.gz，nsca-2.7.2.tar.gz |
| 分布式服务器 | 192.168.80.162 | nagios-3.2.1.tar.gz，nagios-plugins-1.4.13.tar.gz，nrpe-2.12.tar.gz，nsca-2.7.2.tar.gz |
| 被监控服务器 | 192.168.80.163 | nagios-plugins-1.4.13.tar.gz，nrpe-2.12.tar.gz |

### 7.2.2安装nagios被监控服务器

1）安装nagios-plugins

|  |
| --- |
| yum install mysql-devel -y  useradd nagios -M -s /sbin/nologin  tar zxf nagios-plugins-1.4.13.tar.gz  cd nagios-plugins-1.4.13  ./configure \  --with-nagios-user=nagios --with-nagios-group=nagios --enable-perl-modules  make  make install  cd ..  如果在make时候遇到：\*\*\*[all] Error 2则configure加--with-mysql=/usr/local/mysql解决  其中/usr/local/mysql为mysql的安装路径  如果为redhat linux，则可能会卡到如下位置：  configure: see <http://nagiosplugins.org/fag/compile/configure_appers_to_hang> if this next part takes a long time check for redhat spoen problem  解决办法：   1. 多等待一会就OK了，或者编译增加参数 --enable-redhat-pthread-workaround 重新编译   如果没有check\_mysql需要安装mysql-devel然后./configure && make && make install |

此时检查插件个数：

|  |
| --- |
| ls /usr/local/nagios/libexec/ | wc -l  54 |

2）安装nrpe

|  |
| --- |
| tar zxf nrpe-2.12.tar.gz  cd nrpe-2.12  ./configure  make all  make install-plugin  make install-daemon  make install-daemon-config  cd ..  提示：生成nrpe的配置文件/usr/local/nagios/etc/nrpe.cfg |

3）配置nrpe

|  |
| --- |
| cd /usr/local/nagios/etc  vim nrpe.cfg +79  加入可以监控当前client的nagios server端的IP，如下  allow\_hosts=192.168.80.162  提示：allow\_hosts=后面的IP为nagios分布式端的IP，不是nagios监控中心服务器的  如果机器大于500台做集群或分布式监控时，可以使用多个nagios\_hosts， ip要用都逗号隔开。  上面的修改可以使用下列命令进行替换：  sed -i 's/allowed\_hosts=127.0.0.1/allowed\_hosts=192.168.80.162/g' /usr/local/nagios/etc/nrpe.cfg  grep allowed\_hosts /usr/local/nagios/etc/nrpe.cfg  小技巧：echo "alias grep='grep --color=auto'" >> ~/.bashrc 。这样以后grep过滤过来的字符会加上颜色 |

然后在命令行模式下执行shilf+g到结尾 进行如下修改：或者注释掉或者删除掉199-203行。即下面几行

|  |
| --- |
| 199 #command[check\_users]=/usr/local/nagios/libexec/check\_users -w 5 -c 10  200 #command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20  201 #command[check\_hda1]=/usr/local/nagios/libexec/check\_disk -w 20% -c 10% -p /dev/hda1  202 #command[check\_zombie\_procs]=/usr/local/nagios/libexec/check\_procs -w 5 -c 10 -s Z  203 #command[check\_total\_procs]=/usr/local/nagios/libexec/check\_procs -w 150 -c 200 |

同时在下面新添加要监控的内容：

|  |
| --- |
| command[check\_load]=/usr/local/nagios/libexec/check\_load -w 15,10,5 -c 30,25,20  command[check\_mem]=/usr/local/nagios/libexec/check.memory.sh -w 10% -c 3%  command[check\_disk]=/usr/local/nagios/libexec/check\_disk -w 15% -c 7% -p /  command[check\_swap]=/usr/local/nagios/libexec/check\_swap -w 20% -c 10%  command[check\_iostat]=/usr/local/nagios/libexec/check\_iostat -w 6 -c 10  上面依次是对负载，内存，磁盘，虚拟内存，磁盘IO的监控，这些都是本地的服务（我们这里称之为被动监控），由nagios服务器端通过nrpe插件定时去client的nrpe服务定期获取信息，原理如图。  对于负载，单核CPU的负载不超过5，双核CPU的负载不超过10  中括号里面的名称是需要在nagios服务器端调用的。比较重要 |

启动nagios client

|  |
| --- |
| /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d  echo "# nagios nrpe process cmd by fish 2015-7-30" >> /etc/rc.local  echo "/usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d" >> /etc/rc.local  ps -ef | grep nagios  重启命令：  pkill nrpe && /usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d |

提示：客户端的nrpe.cfg配置文件，最好保存到一份到电脑上，这样以后在其他的机器上装nrpe时，直接上传即可，不需要在重新修改了。当然了。客户端nrpe的服务也可以写成一个启动的脚本。脚本见上面2.4章节

### 7.2.3安装nagios分布式服务器

1）安装nagios服务端

|  |
| --- |
| /usr/sbin/useradd -m nagios  /usr/sbin/useradd apache -M -s /sbin/nologin  /usr/sbin/groupadd nagcmd  /usr/sbin/usermod -a -G nagcmd nagios  /usr/sbin/usermod -a -G nagcmd apache  id -n -G nagios  id -n -G apache  groups nagios  groups apache  说明：useradd -M 表示不创建家目录，-s /sbin/nologin表示不能登录。apache用户可能已经存在，因为使用yum 安装httpd会自动创建apache用户。  /usr/sbin/usermod -a -G nagcmd nagios 表示把nagios加入到nagcmd组中。  id -n -G nagios 检查nagios用户属于那个组，-n等同于--name，显示组名  注意：如果你是yum安装的httpd，那么默认的httpd.conf配置文件中User和Group都是apache。如果你是编译安装httpd，那么是daemon。 |
| cd /home/fish/tools/nagios  tar zxf nagios-3.2.1.tar.gz  cd nagios-3.2.1  ./configure --with-command-group=nagcmd  make all  make install  make install-init  make install-commandmode  make install-config  cd..  /etc/init.d/nagios start  chkconfig nagios on |

2）安装nagios-plugins

|  |
| --- |
| yum install mysql-devel -y  tar zxf nagios-plugins-1.4.13.tar.gz  cd nagios-plugins-1.4.13  ./configure \  --with-nagios-user=nagios --with-nagios-group=nagios --enable-perl-modules  make  make install  cd ..  如果在make时候遇到：\*\*\*[all] Error 2则configure加--with-mysql=/usr/local/mysql解决  其中/usr/local/mysql为mysql的安装路径  如果为redhat linux，则可能会卡到如下位置：  configure: see <http://nagiosplugins.org/fag/compile/configure_appers_to_hang> if this next part takes a long time check for redhat spoen problem  解决办法：   1. 多等待一会就OK了，或者编译增加参数 --enable-redhat-pthread-workaround 重新编译   如果没有check\_mysql需要安装mysql-devel然后./configure && make && make install |

此时检查插件个数：

|  |
| --- |
| ls /usr/local/nagios/libexec/ | wc -l  54 |

3）安装nrpe

|  |
| --- |
| tar zxf nrpe-2.12.tar.gz  cd nrpe-2.12  ./configure  make all  make install-plugin  make install-daemon  make install-daemon-config  cd ..  提示：这里不需要在nrpe.cfg配置文件中配置指向nagios监控中心的地址。因为nagios分布式服务和nagios中心服务器的通信是一开NSCA，nagios安装nsca\_send，主动发送给nagios服务器端的nsca进程数据 |

4）安装nsca

wget <http://sourceforge.net/projects/nagios/files/nsca-2.x/nsca-2.9/nsca-2.9.tar.gz/download>

tar zxf nsca-2.9.tar.gz

cd nsca-2.9

./configure

make all

cp sample-config/send\_nsca.cfg /usr/local/nagios/etc/

cd /usr/local/nagios/etc/

chown nagios.nagios send\_nsca.cfg

cd /home/fish/tools/nagios\_gather/nsca-2.9

cp src/send\_nsca /usr/local/nagios/bin/

1. 写一个发送给nagios监控中心服务器数据的脚本

[root@nagios-distribute bin]# cat /usr/local/nagios/libexec/submit\_check\_result

#!/bin/sh

# Arguments:

# $1 = host\_name (Short name of host that the service is

# associated with)

# $2 = svc\_description (Description of the service)

# $3 = state\_string (A string representing the status of

# the given service - "OK", "WARNING", "CRITICAL"

# or "UNKNOWN")

# $4 = plugin\_output (A text string that should be used

# as the plugin output for the service checks)

#

# Convert the state string to the corresponding return code

return\_code=-1

case "$3" in

OK)

return\_code=0

;;

WARNING)

return\_code=1

;;

CRITICAL)

return\_code=2

;;

UNKNOWN)

return\_code=-1

;;

esac

# pipe the service check info into the send\_nsca program, which

# in turn transmits the data to the nsca daemon on the central

# monitoring server

/usr/bin/printf "%s\t%s\t%s\t%s\n" "$1" "$2" "$return\_code" "$4" | /usr/local/nagios/bin/send\_nsca 192.168.80.161 -c /usr/local/nagios/etc/send\_nsca.cfg

#其中192.168.80.161为nagios监控中心服务器

#授权

chmod +x /usr/local/nagios/libexec/submit\_check\_result

chown nagios.nagios /usr/local/nagios/libexec/submit\_check\_result

6）编辑commands.cfg

[root@nagios-distribute bin]# cd /usr/local/nagios/etc/objects/

[root@nagios-distribute objects]# vim commands.cfg

define command{

command\_name submit\_check\_result

command\_line $USER1$/submit\_check\_result $HOSTNAME$ '$SERVICEDESC$' $SERVICESTATE$ '$SERVICEOUTPUT$'

}

7）编辑nagios.cfg配置文件，修改为被动监控

[root@nagios-distribute objects]# vim /usr/local/nagios/etc/nagios.cfg

enable\_notifications=0                 #禁用告警，nagios分布式端禁用报警

obsess\_over\_services=1 #开启被动监控

ocsp\_command=submit\_check\_result #定义每次执行完服务检查后执行的命令

obsess\_over\_hosts=1 #开启主机被动监控

ochp\_command=submit\_check\_result #指定每次执行完主机检查后执行的命令

8）配置nsca\_send发送密码

[root@nagios-distribute objects]# vim /usr/local/nagios/etc/send\_nsca.cfg

#设置密码，此处设置的密码要和监控中心服务器一致

password=nagios

9）测试nagios分布式端和nagios客户端

[root@nagios-distribute bin]# /usr/local/nagios/libexec/check\_nrpe -H 192.168.80.163

NRPE v2.12

#只要出现NRPE版本，说明nagios分布式端和nagios客户端可以正常通信

10）定义nagios分布式监控，nagios客户端的check\_nrpe命令

vim /usr/local/nagios/etc/objects/commands.cfg

#check nrpe

define command{

       command\_name check\_nrpe

       command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

       }

11）在nagios分布式端配置监控客户端

[root@nagios-distribute objects]# cd /usr/local/nagios/etc/objects

[root@nagios-distribute objects]# cat hosts.cfg

############## define hosts #######################

define host {

use linux-server

host\_name 163-lamp-1

alias 163-lamp-1

address 192.168.80.163

check\_command check-host-alive

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 2

check\_period 24x7

notification\_interval 300

notification\_period 24x7

notification\_options d,u,r

contact\_groups admins

process\_perf\_data 1

action\_url /nagios/pnp/index.php?host=$HOSTNAME$

}

################## define hostgroup ########################

define hostgroup {

hostgroup\_name linux-server

alias Linux Servers

members 163-lamp-1

}

编辑services.cfg

[root@nagios-distribute objects]# cat services.cfg

define service {

use generic-service

host\_name 163-lamp-1

service\_description Current Load

check\_command check\_nrpe!check\_load

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

action\_url /nagios/pnp/index.php?host=$HOSTNAME$&ser=$SERVICEDESC$

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description MEM Usage

check\_command check\_nrpe!check\_mem

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Swap Usage

check\_command check\_nrpe!check\_swap

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Disk Partition

check\_command check\_nrpe!check\_disk

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Disk Iostat

check\_command check\_nrpe!check\_iostat

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description ssh\_22

check\_command check\_tcp!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description httpd\_80

check\_command check\_tcp!80

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description httpd\_80\_ssh\_22

check\_command check\_tcp!80!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

}

编辑/usr/local/nagios/etc/nagios.cfg

cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg

cfg\_file=/usr/local/nagios/etc/objects/services.cfg

修改/etc/init.d/nagios脚本。178行

把$NagiosBin -v $NagiosCfgFile > /dev/null 2>&1;

修改成$NagiosBin -v $NagiosCfgFile;

#检查语法，重新加载配置文件

[root@nagios-distribute objects]# /etc/init.d/nagios checkconfig

[root@nagios-distribute objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done

### 7.2.4安装nagios中心监控服务器

首先确认监控中心服务器已经安装了apache且禁用了SElinux

1)安装nagios

cd /home/fish/tools/nagios

tar zxf nagios-3.2.1.tar.gz

cd nagios-3.2.1

./configure --with-command-group=nagcmd

make all

make install

make install-init

make install-commandmode

make install-config

make install-webconf

创建nagios web监控界面登入时需要的用户名（fish）及密码

|  |
| --- |
| [root@nagios-server nagios-3.2.1]# htpasswd -c /usr/local/nagios/etc/htpasswd.users fish  New password:  Re-type new password:  Adding password for user fish  这里我写的密码是fish  [root@nagios-server nagios-3.2.1]# cat /usr/local/nagios/etc/htpasswd.users  fish:637rKTY.BXOok  #提示：如果为单独编译的apache，则操作如下  /usr/local/apache2/bin/htpasswd -c /usr/local/nagios/etc/htpasswd.users fish  为什么要把密码文件指定到/usr/local/nagios/etc/htpasswd.users这个位置呢？因为在make install-webconf生成的文件里面已经定义好了  [root@nagios-server nagios-3.2.1]# cat /etc/httpd/conf.d/nagios.conf | grep AuthUserFile  AuthUserFile /usr/local/nagios/etc/htpasswd.users  AuthUserFile /usr/local/nagios/etc/htpasswd.users |

#编辑nagios.cfg配置文件

[root@nagios-server etc]# cd /usr/local/nagios/etc/

[root@nagios-server etc]# vim cgi.cfg +119

authorized\_for\_system\_information=nagiosadmin,fish

authorized\_for\_configuration\_information=nagiosadmin,fish

authorized\_for\_system\_commands=nagiosadmin,fish

authorized\_for\_all\_services=nagiosadmin,fish

authorized\_for\_all\_hosts=nagiosadmin,fish

authorized\_for\_read\_only=user1,user2,fish

authorized\_for\_all\_service\_commands=nagiosadmin,fish

authorized\_for\_all\_host\_commands=nagiosadmin,fish

提示：

1.理想情况最好干掉默认管理员用户，nagiosadmin或替换自己的fish等，然后分居需求加适合自己的权限，比如给别人看只给浏览权限即可。具体见cgi.cfg

2.authorized\_for\_read\_only=user1,user2,fish这个表示仅仅是以read only的方式查看

3.遇到调试问题注意查看/usr/local/nagios/var/nagios.log

4.可以使用sed 替换命令 sed -i‘s#nagiosadmin#lodboy#g’ cgi.cfg

5.reload nagios命令为 /etc/init.d/nagios reload 此处cgi.cfg的修改可以不执行reload

chkconfig --add nagios

chkconfig nagios on

2)安装nagios-plugins

useradd nagios -M -s /sbin/nologin

tar zxf nagios-plugins-1.4.13.tar.gz

cd nagios-plugins-1.4.13

./configure \

--with-nagios-user=nagios --with-nagios-group=nagios --enable-perl-modules

make

make install

cd ..

3)安装nsca

wget <http://sourceforge.net/projects/nagios/files/nsca-2.x/nsca-2.9/nsca-2.9.tar.gz/download>

tar zxf nsca-2.9.tar.gz

cd nsca-2.9

./configure

make all

cp src/nsca /usr/local/nagios/bin/

chown nagios.nagios /usr/local/nagios/bin/nsca

cp sample-config/nsca.cfg /usr/local/nagios/etc/

chown nagios.nagios /usr/local/nagios/etc/nsca.cfg

#编辑nsca.cfg配置文件

[root@realserver2 nsca-2.9]# vim /usr/local/nagios/etc/nsca.cfg

password=nagios

#编辑nagios.cfg支持接受nagios分布式端被动传送过来的数据

[root@realserver2 nsca-2.9]# vim /usr/local/nagios/etc/nagios.cfg

check\_external\_commands=1 # 配置nagios检查扩展命令

accept\_passive\_service\_checks=1 # 配置接受被动服务检测的结果

accept\_passive\_host\_checks=1 #配置接受被动主机检测的结果

#修改nagios.cfg配置文件

[root@realserver2 etc]# pwd

/usr/local/nagios/etc

[root@realserver2 etc]# vim nagios.cfg

cfg\_file=/usr/local/nagios/etc/objects/services.cfg

cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg

#启动nsca

[root@realserver2 var]# /usr/local/nagios/bin/nsca -d -c /usr/local/nagios/etc/nsca.cfg

4)安装nrpe

|  |
| --- |
| tar zxf nrpe-2.12.tar.gz  cd nrpe-2.12  ./configure  make all  make install-plugin  make install-daemon  make install-daemon-config  cd ..  提示：这里不需要在nrpe.cfg配置文件中配置指向nagios监控中心的地址。因为nagios分布式服务和nagios中心服务器的通信是一开NSCA，nagios安装nsca\_send，主动发送给nagios服务器端的nsca进程数据 |

#修改nagios中心监控服务器端commands.cfg命令。

define command{

ommand\_name check\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c ARG1$

}

1. 配置hosts.cfg和services.cfg

hosts.cfg配置文件

[root@realserver2 objects]# cat hosts.cfg

############## define hosts #######################

define host {

use linux-server

host\_name 163-lamp-1

alias 163-lamp-1

address 192.168.80.163

check\_command check-host-alive

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 2

check\_period 24x7

notification\_interval 300

notification\_period 24x7

notification\_options d,u,r

contact\_groups admins

process\_perf\_data 1

action\_url /nagios/pnp/index.php?host=$HOSTNAME$

}

################## define hostgroup ########################

define hostgroup {

hostgroup\_name linux-server

alias Linux Servers

members 163-lamp-1

}

services.cfg配置文件

[root@realserver2 objects]# cat services.cfg

define service {

use generic-service

host\_name 163-lamp-1

service\_description Current Load

check\_command check\_nrpe!check\_load

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

action\_url /nagios/pnp/index.php?host=$HOSTNAME$&ser=$SERVICEDESC$

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description MEM Usage

check\_command check\_nrpe!check\_mem

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Swap Usage

check\_command check\_nrpe!check\_swap

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Disk Partition

check\_command check\_nrpe!check\_disk

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description Disk Iostat

check\_command check\_nrpe!check\_iostat

max\_check\_attempts 2

normal\_check\_interval 4

retry\_check\_interval 4

check\_period 24x7

notification\_interval 1440

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

process\_perf\_data 1

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description ssh\_22

check\_command check\_tcp!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description httpd\_80

check\_command check\_tcp!80

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

define service {

use generic-service

host\_name 163-lamp-1

service\_description httpd\_80\_ssh\_22

check\_command check\_tcp!80!22

max\_check\_attempts 3

normal\_check\_interval 2

retry\_check\_interval 1

check\_period 24x7

notification\_interval 30

notification\_period 24x7

notification\_options w,u,c,r

contact\_groups admins

passive\_checks\_enabled 1

active\_checks\_enabled 0

}

检查语法，重新加载配置文件

[root@realserver2 objects]# /etc/init.d/nagios checkconfig

[root@realserver2 objects]# /etc/init.d/nagios reload

Running configuration check...done.

Reloading nagios configuration...done

