

## Linux 企业级开源监控软件 Nagios 与 Cacti 整合

### 一、 nagios 简介

Nagios 是一款用于系统和网络监控的应用程序。它可以在你设定的条件下对主机和服务惊醒监控，在状态改变的时候发出告警信息。

Nagios 的主要特征包括：

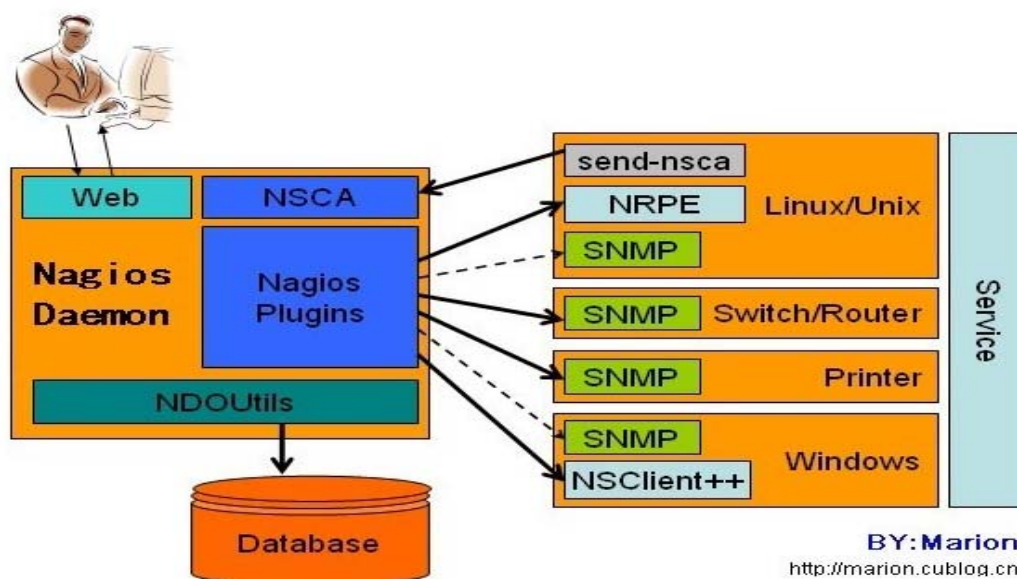
- (1) 监控网络服务 (SMTP、POP3、HTTP、NNTP、PING 等)。
- (2) 监控主机资源 (处理器负荷、磁盘利用率等)。
- (3) 简单的插件设计使用户可以方便地扩展自己服务的检测方法。
- (4) 并行服务检查机制
- (5) 具备定义网络分层结构的能力，用 “parent” 主机定义来表达网络主机的关系、这种关系可以被用来发现和明晰主机宕机或不可达状态。
- (6) 当服务或主机问题产生与解决时将告警发送给联系人 (通过 E-mail、短信、用户定义方式等)，既可以传递给管理员，也可以传递给相关工作人员，可高效的保证服务器的维护。
- (7) 具备定义时间句柄的功能，它可以在主机或服务的事件发生时获取更多问题定位。
- (8) 自动的日志回滚。
- (9) 可以支持并实现对主机的冗余监控。
- (10) 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。

Nagios 通常由一个主程序(nagios)、一个插件程序(Nagios-plugins)和四个可选的 ADDON (NRPE、NSCA、NSClient++和 NDOUTils)组成。Nagios 的监控工作都是通过插件(Nagios-plugins 包含监控所需要的各种命令)来实现的，因此，Nagios 和 Nagios-plugins 是服务器端工作的必须的组件。而四个 ADDON 中。

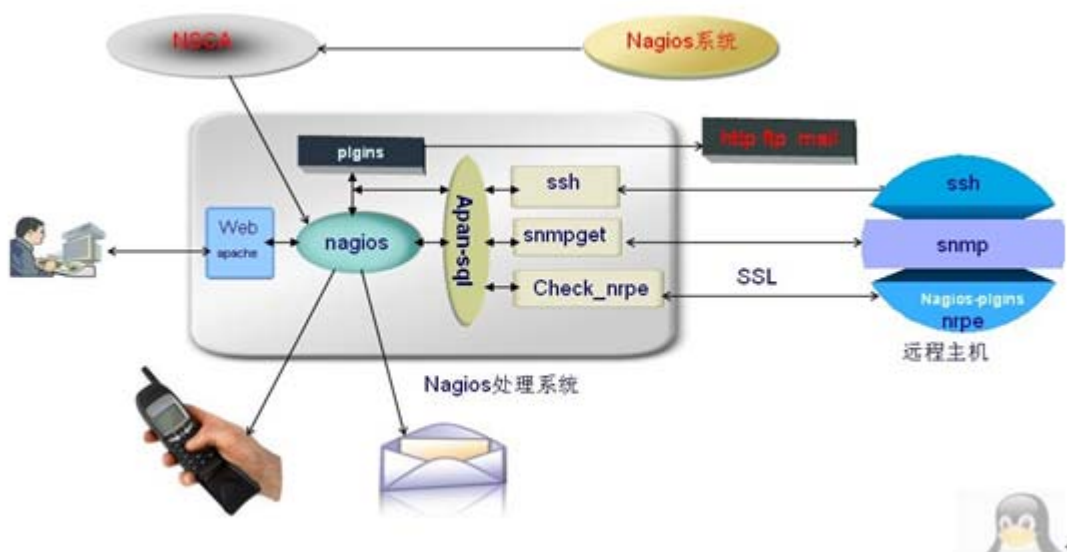
- (1) NRPE: 用来在监控的远程 Linux/Unix 主机上执行脚本插件以实现对这些主机资源的监控。
- (2) NSCA: 用来让被监控的远程 Linux/Unix 主机主动将监控信息发送给 Nagios 服务器 (这在冗余监控模式中特别要用到)。
- (3) NSClient++: 用来监控 Windows 主机时安装在 Windows 主机上的组件 (监控 Windows 的必须组件)。
- (4) NDOUTils: 用来将 Nagios 的配置信息和各 event 产生的数据存入数据库，以实现这些数据的快速检索和处理。

这四个 ADDON(附件)中， NSClient++工作于 Windows 客户端，NDOUTils 工作于服务器端，而 NSCA 和 NRPE 则需要同时安装在服务器端和客户端

用图片展示 nagios 的工作原理:



在用一副图片宏观上展示 Nagios 在企业级应用的位置



## 二、 Cacti 简介

网络管理（如主机负载、网络流量等的监控）是系统管理岗工作人员日常工作的一个重要组成部分，许多厂商为此提供了各自的解决方案。早期开源网管软件中比较著名的其中之一的是 MRTG，是一款使用 perl 开发的，通过 SNMP 协议实现管理工作站与设备代理进程间的通讯，以完成对设备的管理和运行状态的监视。MRTG 安装配置简单、图形界面直观，因而广受当时网络管理人员的喜爱。但是 MRTG 也有很多固有的缺点，如：使用文本式的数据库，数据不能重复使用；只能按日、周、月、年来查看数据；每图只能画两个 DS（一条线、一个块）；每取一次数据即需要绘图一次，浪费系统资源；同时，它也没有提供管理功能。

有鉴于此，MRTG 的作者后来又开发了一个工具，即 rrdtool。rrdtool 是一个性能优良的

数据记录器，同时也是一个功能强大的绘图引擎。它使用 rrd 的数据存储格式，使用的数据可以重复使用；在绘图方面，它可以定义任意时间段进行绘图，能绘出多个 DS。但它也有自己的缺点；虽然提供了强大的数据存储及绘图功能，但 rrdtool 却没有提供类似于 MRTG 中集成的数据采集功能；同时，它提供了过多的参数，以至于在命令行界面使用起来极为不便；此外，rrdtool 也没有提供管理功能。于是 cacti 应运而生。

简单来说，Cacti 就是 rrdtool 的一个 forefront，它内置了快速的获取数据工具、优秀的绘图模板以及许多设计精良的数据获取脚本，从而可以通过结合 rrdtool 强大的数据抓取、数据存储和绘图功能、轻松实现主机负载、网络流量等信息的走势图的绘制。

Cacti 使用 PHP 语言实现的一个软件，它的运行需要网站服务器（如 apache）及 php 环境的支持。同时，cacti 同时还需要 Mysql 配合 PHP 程序存储一些变量数据并对变量数据进行调用，如：主机名、主机 ip、snmp 团体名、端口号、模板信息等变量。

Rrdtool 的对主机负载、网络流量等信息的统计需要通过 snmp 协议实现。经过 snmp 抓取到的数据存储于由 rrdtool 生成的 rrd 文件中，这些文件通常位于 cacti 的 rra 目录中。rrdtool 对数据的更新和存储就是对 rrd 文件的处理，rrd 文件大小固定的归档文件(Round Robin Archive)，它能够存储的数据笔数在创建时就已经定义。

Cacti 的实例应用

- (1) 网络设置
- (2) 主机系统
  - ① 网络接口流量（进与出的带宽）
  - ② 监控 CPU 的负载、内存等等
  - ③ 监控磁盘的空间、进程数等等
- (3) Cacti 常见的监测对象
  - ① 服务器资源：CPU、内存、磁盘、进程、连接数等
  - ② 服务器类型：WEB、Mail、FTP、数据库、中间件
  - ③ 网络接口：流量、转发速度、丢包率
  - ④ 网络设备性能、配置文件（对比与备份）、路由数
  - ⑤ 安全设备性能、连接数、攻击数
  - ⑥ 设备运行状态：风扇、电源、温度
  - ⑦ 机房运行环境：电流、电压、温湿度

### 三、源码实现 Nagios

#### 1. 首先编译安装 nagios

(1)安装前的依赖检查,nagios 需要一些基础支持套件才能运行,如 apache, gcc, glibc, gd 库等。

```
# rpm -q httpd php gcc glibc glibc-common gd gd-devel
```

```
[root@localhost ~]# rpm -q httpd php gcc glibc glibc-common gd gd-devel
httpd-2.2.3-31.el5
php-5.1.6-23.2.el5_3
gcc-4.1.2-46.el5
glibc-2.5-42
glibc-common-2.5-42
gd-2.0.33-9.4.el5_1.1
gd-devel-2.0.33-9.4.el5_1.1
```

```
#/usr/sbin/useradd -m nagios 添加一个名为nagios 的用户用以专门跑nagios
```

#passwd nagios 设置密码

#/usr/sbin/groupadd nagcmd 添加nagcmd 用户组,用以通过web 页面提交外部控制命令

#/usr/sbin/usermod -a -G nagcmd nagios 将nagios 用户加入nagcmd 组

#/usr/sbin/usermod -a -G nagcmd apache 将 apache 用户加入 nagcmd 组

## (2) 安装 nagios

#mkdir ~/downloads

#cd ~/downloads

# wget http://nchc.dl.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz

# wget

http://nchc.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz

#cd ~/downloads



#tar xzf nagios-3.2.1.tar.gz

#cd nagios-3.2.1



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

#make all 编译

#make install 安装

#make install-init 使用 make install-init 在/etc/rc.d/init.d 安装启动脚本

#make install-config

#make install-commandmode 使用 make install-commandmode 来配置目录权限

#make install-webconf 安装nagios 的web 接口

#htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin 设置登陆web 界面时HTTP 验证的账号密码

#service httpd restart 启动 apache

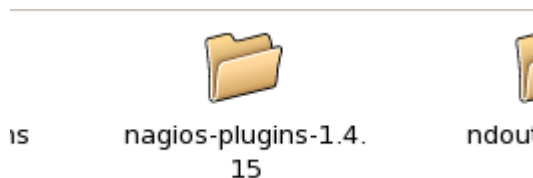
## (3) 安装 nagios-plugins

nagios-plugins 是 nagios 官方提供的一套插件程序, nagios 监控主机的功能其实都是通过执行插件程序来实现的。

#cd ~/downloads

#tar xzf nagios-plugins-1.4.15.tar.gz

#cd nagios-plugins-1.4.15



```
#./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
#make
```

```
#make install
```

安装插件，安装后所有插件命令将被安装到/usr/local/nagios/libexec 目录下。

#### (4) 安装 nagios-snmp-plugins

nagios-snmp-plugins 是一套用 Perl 编写的通过 SNMP 方式监控主机的插件程序。

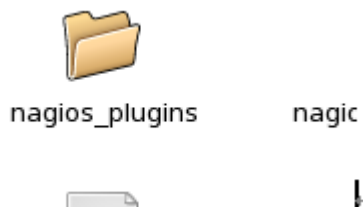
```
#cd ~/downloads
```

```
#wget http://nagios.manubulon.com/nagios-snmp-plugins.1.1.1.tgz
```



```
#tar xzf nagios-snmp-plugins.1.1.1.tgz
```

```
#cd nagios_plugins
```



配置check\_snmp\_int.pl 这些插件的使用时需要配置cpan，CPAN 是Comprehensive Perl Archive

Network 的缩写。它是一个巨大的Perl 软件收藏库，收集了大量有用的Perl 模块(modules)

及其相关的文件。这里主要是使用 Perl-Net-SNMP 模块。有两种方式安装：

A) 通过CPAN 来安装

```
#perl -MCPAN -e shell
```

```
cpan> install Net::SNMP
```

B) 手工安装

首先去官方网站www.cpan.org 下载以下几个模块

```
Crypt::DES
```

```
Digest::MD5
```

```
Digest::SHA1
```

```
Digest::HMAC
```

```
Net::SNMP
```

下载后对于每个模块依次按照下面的方式安装

```
#tar xzf <module>.tar.gz <module>表示模块名，具体请按上面提到的模块替换
#cd <module> <module>表示模块名，具体请按上面提到的模块替换
#perl Makefile.pl
#make test
#make install
```

注意：Net::SNMP 模块必须在最后安装。至此 Net::SNMP 手动安装完毕，我使用的是手动安装

安装包如下：



#./install.sh 执行nagios-snmp-plugins 安装脚本,执行之后会将插件命令安装到/usr/local/nagios/libexec下。

## (5) 进行其他设置

至此nagios 基本已经安装完毕，但这时还不能马上启动nagios，需要以下设置。

#chkconfig --add nagios 将nagios 添加到服务中

#chkconfig nagios on 设置服务为自启动

#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg 检测nagios 的配置是否正确，在后面配置nagios 过程中我们为了检测配置的是否正确需要不断执行该命令来检查配置文件。

#service nagios start 启动nagios

需要注意的是，如果打开了selinux 并且运行于强制安全模式，这将导致在打开nagios的web 界面时会出现Internet Server Error 的错误。

#getenforce 查看是否运行于强制模式，结果为1 表示是

#setenforce 0 更改selinux 运行于宽容模式

但是这个设置重启后就会失效，如需要重启后保持该设置vi /etc/sysconfig/selinux，将其中的SELINUX= enforcing 更改为SELINUX= permissive并重启系统。当然你也可以改成disable禁用selinux。

也可以不更改selinux 的运行模式，解决办法为：

#chcon -R -t httpd\_sys\_content\_t /usr/local/nagios/sbin/

#chcon -R -t httpd\_sys\_content\_t /usr/local/nagios/share/

## 2. nagios 的配置

### (1) nagios 目录结构

nagios 目录结构比较清晰，安装后下面一共有bin etc libexec sbin share var 几个目录，其中配置文件都存放在etc 目录下，bin 下存放的是nagios 的相关命令，sbin 下存放的是通过web 方式外部执行的cgi,libexec 存放的是所有插件，而var 则存放的是log 和pid 文件等。而实现监控都是依靠执行插件来实现的。要实现监控，需要在nagios 中定义



一个service,在这个service 中指定监控对象和监控命令以及报警机制等。

nagios 的配置也比较清晰明了, etc 目录下默认有objects 目录和一些配置文件, 其中objects 里放的是主配置文件nagios.cfg 包含进去的配置文件。在nagios.cfg 中既可以指定单独包含一个cfg, 也可以指定一个包含目录, 即该目录下所有的cfg 文件都会包含进来。



Object文件夹下的文件:



## (2) 主配置文件nagios.cfg的配置

主配置文件的内容很多, 对于这个版本, 我们需要修改和添加的主要是对象配置文件,即:

cfg\_file=<file\_name>

cfg\_file=/usr/local/nagios/etc/objects/commands.cfg (命令定义文件)

cfg\_file=/usr/local/nagios/etc/objects/contacts.cfg (联系人信息定义文件)

cfg\_file=/usr/local/nagios/etc/objects/contactgroups.cfg (添加此行联系人组定义文件)

cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg (添加此行主机定义文件)

cfg\_file=/usr/local/nagios/etc/objects/hostgroups.cfg (添加此行主机组定义文件)

cfg\_file=/usr/local/nagios/etc/objects/services.cfg (添加此行服务定义文件)

cfg\_file=/usr/local/nagios/etc/objects/timeperiods.cfg (时间周期定义文件)

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

# Definitions for monitoring the local (Linux) host

#cfg\_file=/usr/local/nagios/etc/objects/localhost.cfg

#cfg\_file=/usr/local/nagios/etc/objects/switch.cfg (此行可以注释掉)

## (3) 主机定义文件的配置

故名思议把需要监控的主机放在该文件里, 在Nagios的监控界面上就会存在这些主机, 当然, 还有一些其它的设置:

[root@localhost objects]# vi hosts.cfg

```
define host{
```

```
host_name Me          自定义主机名
```

```
alias BenJi          定义主机别名
```

```

address 192.168.11.138    被监控主机的IP地址
check_command check-host-alive  检查的命令（此命令为检查主机是否存活）
check_interval 5          检查的时间间隔
retry_interval 1
max_check_attempts 5
check_period 24x7         检查的时间段一般服务都是7x24
process_perf_data 0
retain_nonstatus_information 0
contact_groups sagroup  设置联系人组：当该主机出现了报警信息，就发信息给这个组
notification_interval 30  提醒的时间间隔，这个单位是分钟。
notification_period 24x7  #提醒的时间段
notification_options d,u,r
}

define host{
host_name Hello
alias Hello_baby
address 192.168.11.137
check_command check-host-alive
check_interval 5
retry_interval 1
max_check_attempts 5
check_period 24x7
process_perf_data 0
retain_nonstatus_information 0
contact_groups sagroup
notification_interval 30
notification_period 24x7
notification_options d,u,r
}

define host{
host_name Windows
alias Windows Server
address 192.168.11.128
check_command check-host-alive
check_interval 5
retry_interval 1
max_check_attempts 5
check_period 24x7
process_perf_data 0
retain_nonstatus_information 0
contact_groups sagroup
notification_interval 30
notification_period 24x7

```



```
notification_options d,u,r
}
```

#### (4) 主机组定义文件的配置

对于一些喜欢分组管理的朋友和一些服务器较多的应用，分组无疑是最好的选择，比如你在一个IDC机房做Nagios监控服务器，分组是很有必要的，可以根据操作系统类型，功能、区域等等很多进行有效的分组，才能达到预期的目的。

```
[root@localhost objects]# vi hostgroups.cfg
define hostgroup {
    hostgroup_name System-Admin    组名字，可以根据应用调整
    alias system Admin            别名
    members Me,Hello,Windows
    #组成员用逗号隔开，这里的组成员是hosts.cfg定义的“host_name”这个的值
}
```

#### (5) 服务定义文件的配置

服务的配置是很重要的，也是配置最多的地方，对与每个主机要监控的服务，都在这个文件中列出，还好的是并不复杂，只需修改“host\_name”、“service\_description”等一些字段。

```
[root@localhost objects]# vi services.cfg
define service {
    host_name Me                主机名字，hosts.cfg中定义的
    service_description check-host-alive  所监控服务的描述，怎么描述合理就怎么定义
    check_period 24x7          检查的时间段
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7    提醒的时间段
    notification_options w,u,c,r 在这四种情况下，通知联系人
    check_command check-host-alive 检查的命令,在command.cfg中定义
}
```

w—报警(warning)

u—未知(unknown)

c—严重(critical)

r—从异常情况恢复正常

```
define service {
    host_name Hello
    service_description check-host-alive
    check_period 24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7
```

```
notification_options w,u,c,r
check_command check-host-alive
}

define service {
host_name Hello
service_description check-users
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_users
}

define service {
host_name Hello
service_description check-load
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_load
}

define service {
host_name Hello
service_description check-total-procs
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_total_procs
}
```

```
define service {
    host_name Windows
    service_description Check-Memory
    check_period 24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_nt!MEMUSE!-w 80 -c 90
}

define service {
    host_name Windows
    service_description Check-Disk-C
    check_period 24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_nt!USEDISKSPACE!-l c -w 80 -c 90
}

define service {
    host_name Windows
    service_description Check-Disk-D
    check_period 24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_nt!USEDISKSPACE!-l d -w 80 -c 90
}
```

(6) 联系人的配置:

联系人, 顾名思义就是管理员了, 根据值班的时间定义了。

```
define contact{
```

```
define contact{
    contact_name      nagiosadmin          ; Short name
    user
        use           generic-contact      ; Inherit
    fault values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name
    user
        email         lee8114@qq.com       ; <<*****
CHANGE THIS TO YOUR EMAIL ADDRESS *****
}
```

将邮箱改为自己的邮箱。

(7) 联系组定义文件配置

[root@localhost objects]# vi contactgroups.cfg

```
define contactgroup {
    contactgroup_name sagroup
    alias system administrator group
    members nagiosadmin
}
~
~
~
~
```

(8) 启动nagios服务

[root@localhost objects]# service httpd restart

[root@localhost objects]# service nagios restart

```
[root@localhost objects]# service httpd restart
Stopping httpd:                                     [ FAILED ]
Starting httpd:                                     [  OK  ]
[root@localhost objects]# service nagios restart
Running configuration check...done.
Stopping nagios: done.
Starting nagios: done.
```

核对配置文件有错误:

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

如果更改配置文件需要反复的进行此不操作, 从中会发现配置出了哪些问题。

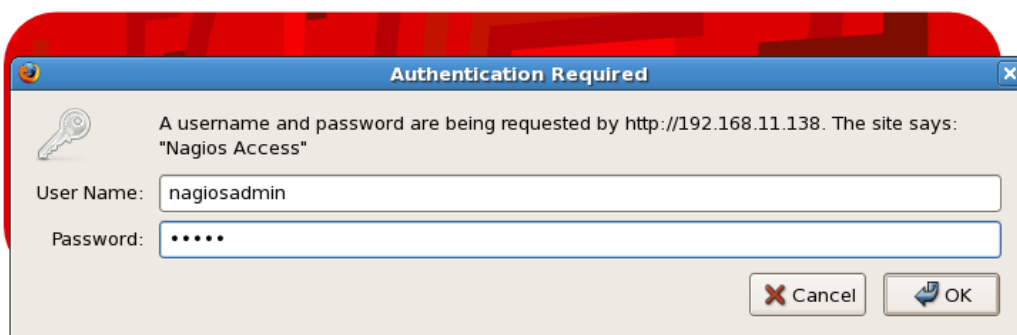
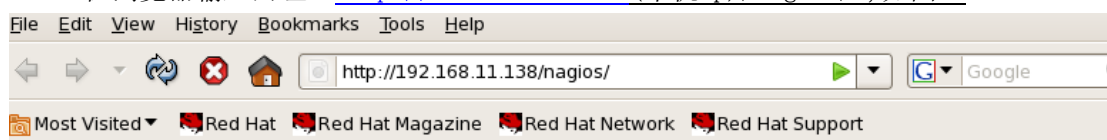
```
Checking misc settings...

Total Warnings: 0
Total Errors:   0

Things look okay - No serious problems were detected during the pre-flight check
[root@localhost ~]#
```

(9) 功能性测试

在浏览器输入网址: <http://192.168.11.138>(本机Ip)/nagios/, 如图:



输入用户名、密码。如图:

Me	check-host-alive	OK	04-18-2012 13:44:15	0d 18h 5m 29s	1/4	PING OK - Packet loss = 0%, RTA = 0.09 ms
Windows	Check-Disk-C	CRITICAL	04-18-2012 13:45:15	0d 21h 11m 33s	1/4	No route to host
	Check-Disk-D	CRITICAL	04-18-2012 13:43:15	0d 21h 13m 25s	1/4	No route to host
	Check-Memory	CRITICAL	04-18-2012 13:44:15	0d 21h 11m 25s	1/4	No route to host
localhost	Current Load	OK	04-18-2012 13:43:30	0d 21h 40m 33s	1/4	OK - load average: 0.45, 0.63, 0.63
	Current Users	OK	04-18-2012 13:44:30	1d 22h 50m 26s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	04-18-2012 13:42:35	0d 0h 3m 26s	1/4	HTTP OK: HTTP/1.1 200 OK - 1668 bytes in 0.163 second response time
						PING OK - Packet

OK, 第一部分的功能已经实现了, 就是简单的监控四台服务器是否存活, 下面进行更深层次的操作。

### 3、使用Nagios监控Linux 客户端

可以把Nagios的服务器监控分为两个部分:

第一部分是主机外监控, 比如: 主机是否存活, WEB服务是否正常, MySQL服务是否正常等内容, 再主机外通过访问其端口即可得知。个人理解这里所说的主机外监控就是指的计算机上的一些服务, 如: smtp、pop3、telnet等等。这些监控命令再安装 nagios-plugins-1.4.13.tar.gz时已经生成了, 再/usr/local/nagios/libexec 目录下。

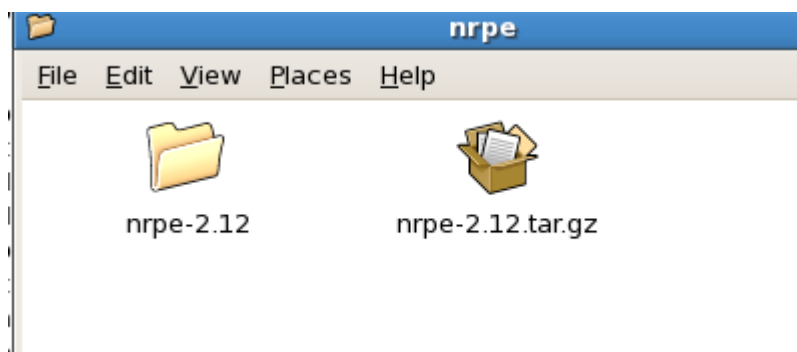
第二部分是主机内监控, 比如: 要监控服务器的进程、磁盘使用等功能。这些功能的实现要依靠nrpe了, nrpe的工作模式是C/S模式, 在被监控主机中, 开启nrpe监听, 当听到监控服务器上所发出的命令, 让它检查该服务器上的硬盘使用信息时, 它就会执行, 并把信息传回, 监控服务器, 用一个不太恰当的比喻, 就是木马的工作模式。

这里有时候可能会弄混, 感觉nagios-plugins-1.4.13.tar.gz和nrpe的功能好像一样, 为什么要安装两个, 刚开始的时候我就弄混了, 弄得头脑好是模糊。nagios-plugins-1.4.13.tar.gz所负责监控的主要是对计算机的服务而言的, 如: smtp、pop3、telnet等等。而nrpe所监控的是计算机本身的特性, 如磁盘、cpu、内存等等。

#### (1) Nagios监控服务端的配置

##### [1] 安装nrpe

```
[root@localhost ~]# cd /usr/local/software/nrpe
```



```
# ./configure && make all
```

```
# make install-plugin
```

```
# make install-daemon
```

```
# make install-daemon-config
```

```
# make install-xinetd
```

[2] 配置nrep

```
[root@localhost nrpe-2.12]# ls
Changelog      docs           install-sh     SECURITY
config.guess   include        LEGAL          src
config.log     init-script    Makefile       subst
config.status  init-script.debian  Makefile.in   subst.in
config.sub     init-script.debian.in  nrpe.spec     update-version
configure      init-script.in    README
configure.in   init-script.suse   README.SSL
contrib        init-script.suse.in sample-config
```

```
[root@localhost nrpe-2.12]# vi /etc/xinetd.d/nrpe
```

```
service nrpe
{
    flags                = REUSE
    socket_type          = stream
    port                 = 5666
    wait                 = no
    user                 = nagios
    group                = nagios
    server                = /usr/local/nagios/bin/nrpe
    server_args           = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure        += USERID
    disable               = no
    only_from             = 127.0.0.1 192.168.11.137 192.168.11.128
    #在only_from 添加要监控的主机的IP地址，中间以空格隔开。
}
```

[3] 添加端口

```
[root@localhost ~]# vi /etc/services
```



```
# Local services
csync      2005/tcp
psync      2005/udp
nrpe       5666/tcp

# Cyrus IMAP Replication Daemon
# Cyrus IMAP Replication Daemon
# nrpe
```

在最后添加: nrpe 5666/tcp #nrpe

[4] 重新启动Xinetd服务

[root@localhost ~]# service xinetd restart

Stopping xinetd: [ OK ]

Starting xinetd: [ OK ]

[5] 修改配置文件commands.cfg加入对nrpe的支持

[root@localhost ~]# vi /usr/local/nagios/etc/objects/commands.cfg

添加如下内容:

```
#nrpe set
define command{
    command_name check_nrpe
    command_line /usr/local/nagios/libexec/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

[6] 查看nrpe的检查命令

[root@localhost ~]# vi /usr/local/nagios/etc/nrpe.cfg

```
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
```

[7] 使用命令监控客户端

[root@localhost ~]# vi /usr/local/nagios/etc/objects/services.cfg

```
define service {
    host_name Hello
    service_description check-users
    check_period 24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups sagroup
    notification_interval 10
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_nrpe!check_users    检查命令必须按照此格式写
}

define service {
```

```

host_name Hello
service_description check-load
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_load
}
define service {
host_name Hello
service_description check-total-procs
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_total_procs
}

```

## (2) Nagios 监控客户端的配置

### [1] 添加nagios用户

```
useradd -s /sbin/nologin nagios
```

### [2] 安装nagios插件nagios-plugin

```

# tar zxvf nagios-plugins-1.4.13.tar.gz
# cd nagios-plugins-1.4.13 [root@prdora1 nagios-plugins-1.4.13]
# ./configure --enable-redhat-pthread-workaround
# make && make install

```

### [3] 安装nrpe

在客户端安装nrpe与在服务端安装nrpe基本相同，唯一的不同就是修改xinetd.d/nrpe的时候在only\_from 里只加入Nagios服务器的IP地址即可。

```

server          = /usr/local/nagios/bin/nrpe
server_args      = -c /usr/local/nagios/etc/
log_on_failure   += USERID
disable          = no
only_from        = 192.168.11.138
}

```

[4] 修改文件所有者

```
# chown -R nagios:nagios /usr/local/nagios
```

```
# /etc/init.d/xinetd restart
```

[5] 进行测试



Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
Hello	check-host-alive	OK	04-18-2012 15:02:00	0d 0h 44m 34s	1/4	PING OK - Packet loss = 0%, RTA = 0.74 ms
	check-load	OK	04-18-2012 15:03:00	0d 0h 43m 34s	1/4	OK - load average: 0.07, 0.10, 0.09
	check-total-procs	OK	04-18-2012 15:01:00	0d 0h 42m 34s	1/4	PROCS OK: 133 processes
	check-users	OK	04-18-2012 15:02:00	0d 0h 44m 34s	1/4	USERS OK - 2 users currently logged in

#### 4、使用Nagios 监控Windows 客户端

##### (1) Nagios 监控服务器的配置

对与Nagios监控服务器不需要什么其它的配置，只需要编写hosts.cfg和services.cfg即可

```
[root@localhost objects]# vi hosts.cfg
```

```
define host{
host_name Windows
alias Windows Server
address 192.168.11.128
check_command check-host-alive
check_interval 5
retry_interval 1
max_check_attempts 5
check_period 24x7
process_perf_data 0
retain_nonstatus_information 0
contact_groups sagroup
notification_interval 30
notification_period 24x7
notification_options d,u,r
}
```

```
[root@localhost objects]# vi services.cfg
```

```
define service {
host_name Windows
service_description Check-Memory
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
}
```

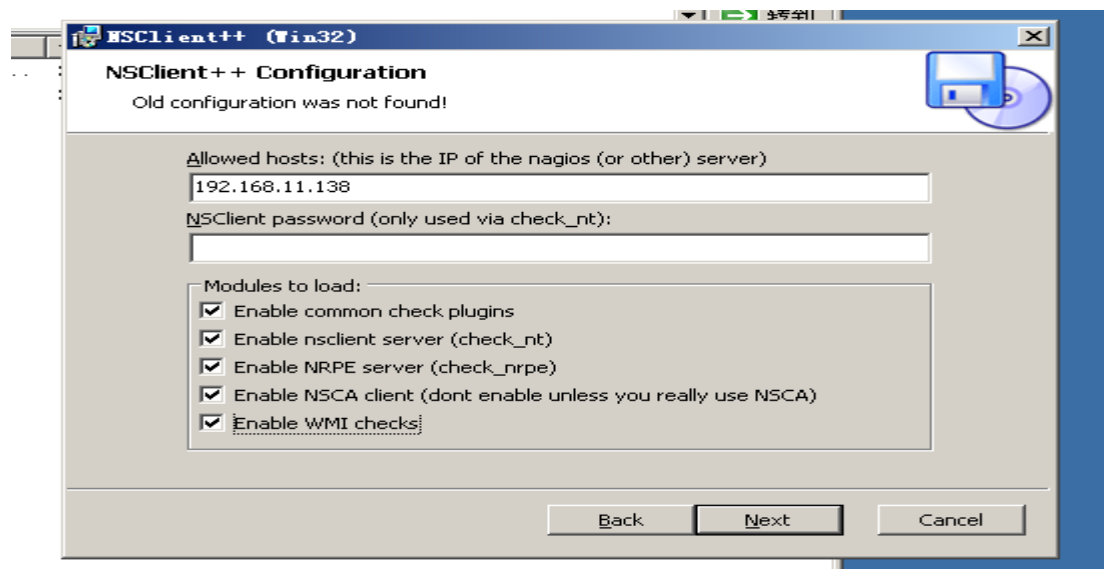
```

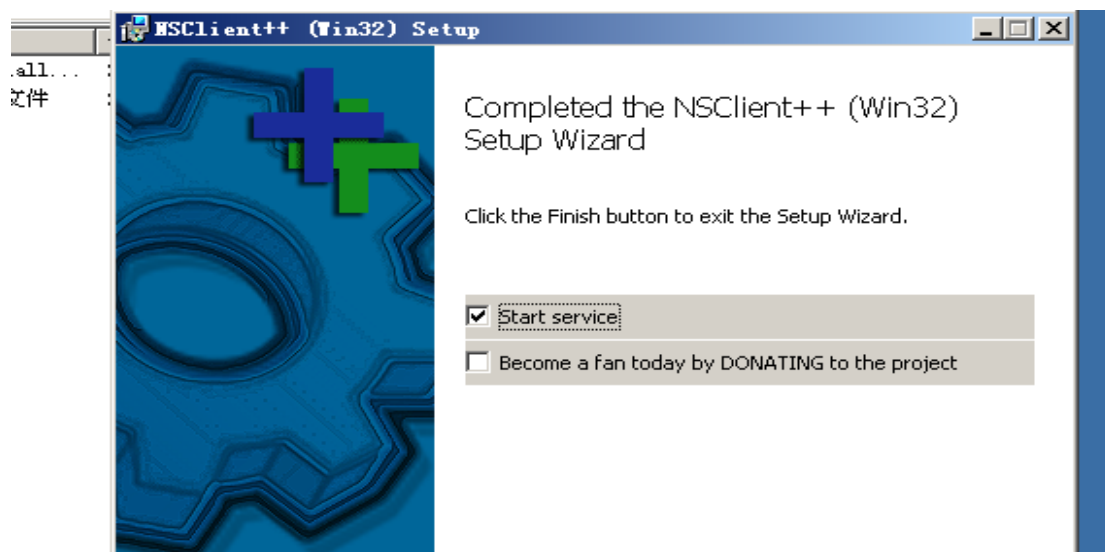
check_command check_nt!MEMUSE!-w 80 -c 90
}
define service {
host_name Windows
service_description Check-Disk-C
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nt!USEDISKSPACE!-l c -w 80 -c 90
}
define service {
host_name Windows
service_description Check-Disk-D
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w,u,c,r
check_command check_nt!USEDISKSPACE!-l d -w 80 -c 90
}

```

## (2) Nagios 监控客户端的配置

安装NSClient++



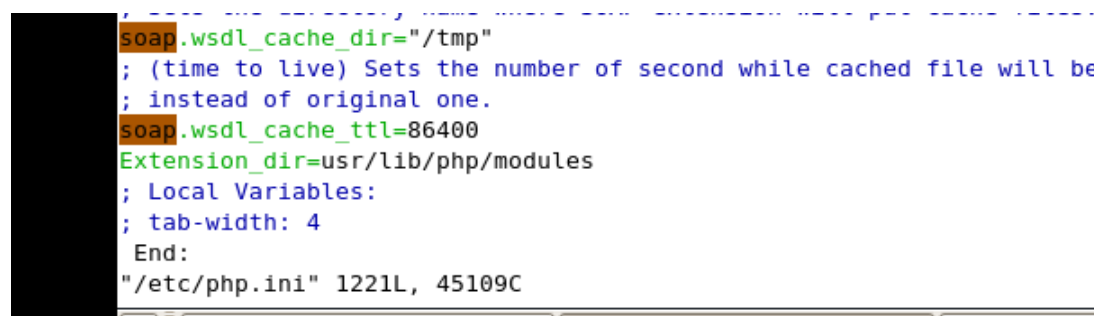


### (3) 进行测试



## 四、 源码编译Cacati

### (1) 配置激活PHP扩展



### (2) 初始化配置mysql建立cacti数据库并授权

Mysqladmin -u root password redhat

Mysql -u root -p

输入之前初始化配置的密码，登录 mysql

```

mysql> create database cacti;
Query OK, 1 row affected (0.01 sec)

mysql> grant all on cacti.* to cacti@localhost identified by 'cacti';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
    
```

(3) 安装rrdtool工具

安装rrdtool, rrdtool 不能直接通过yum 安装, 可以加入Dag RPM Repository 以便让yum 能找到rrdtool.

```
#wget
```

```
http://dag.wieers.com/rpm/packages/rpmforge-release/rpmforge-release-0.3.6-1.el5.
```

```
rf.i386.rpm
```

```
#rpm -Uvh rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
#yum install rrdtool
```

(4) 安装 cacti 主程序包

```
[root@localhost ~]# cd ~/downloads
[root@localhost downloads]# ls
cacti-0.8.7d.tar.gz          nagios-snmp-plugins.1.1.1.tgz
cacti-plugin-0.8.7d-PA-v2.4.diff  ndoutils-1.4b7
cacti-plugin-0.8.7d-PA-v2.4.zip  ndoutils-1.4b7.tar.gz
files-0.8.7d                 npc-2.0.3.tar.gz
LICENSE                      pa.sql
nagios-3.2.1                 Perl
nagios-3.2.1.tar.gz          php-json-ext-1.2.0
nagios_plugins               php-json-ext-1.2.0.tar.bz2
nagios-plugins-1.4.15        Readme.txt
nagios-plugins-1.4.15.tar.gz  rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

```
Tar -zxvf cacti-0.8.7d.tar.gz
```

```
Mv cacti-0.8.7d /var/www/html/cacti
```

```
Useradd cacti
```

```
Chown-R cacti.cacti /var/www/html/cacti
```

(5) 配置 apache: 修改网站根目录为/var/www/html/cacti。

```
vi /etc/httpd/conf/httpd.conf
```

```
dev/nullfault, all requests are taken from this directo
# symbolic links and aliases may be used to point to ot
#
DocumentRoot "/var/www/html/cacti"

#
# Each directory to which Apache has access can be conf.
```

修改 apache 首页为 index.php

```
# negotiated documents. The multiviews option can be used to
# same purpose, but it is much slower.
#
DirectoryIndex index.php index.html index.html.var

#
# AccessFileName: The name of the file to look for in each di
```

(6) 将 cacti 主程序包中的 sql 导入数据库

```
Cd /var/www/html/cacti
```

```
mysql-u root -p cacti<cacti.sql
```

(7) 配置 mysql 连接参数

```
vim /var/www/html/cacti/include/config.php
```



```

/* make sure these values reflect your actual database,
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti";
$database_port = "3306";
$plugins = array();
$plugins[]='npc';

/* Default session name - Session name must contain a
#$cacti_session_name = "Cacti";

?>

```

(8) vim /var/www/html/cacti/include/global.php

```

/* Default database settings*/
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "cacti";
$database_port = "3306";

/* Default session name - Session name must contain alpha ch
$cacti_session_name = "Cacti";

$plugins = array();
//$plugins[]='npc';
//$plugins[] = 'thold';

/* Do not edit this line */
$config = array();

```

(9) cacti-plugin 插件及补丁安装

```

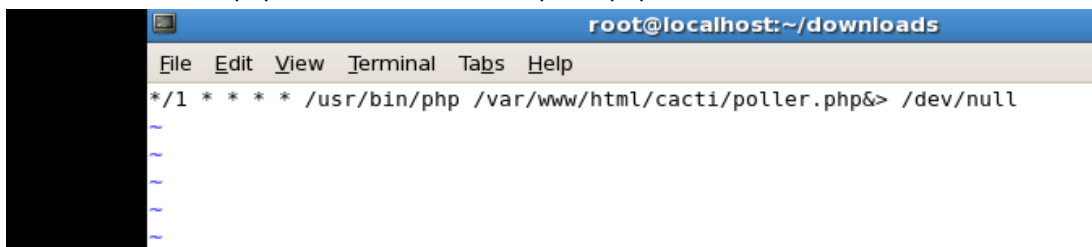
[root@localhost downloads]# ls
cacti-0.8.7d.tar.gz          nagios-snmp-plugins.1.1.1.tgz
cacti-plugin-0.8.7d-PA-v2.4.diff  ndoutils-1.4b7
cacti-plugin-0.8.7d-PA-v2.4.zip  ndoutils-1.4b7.tar.gz
files-0.8.7d                 npc-2.0.3.tar.gz
LICENSE                      pa.sql
nagios-3.2.1                 Perl
nagios-3.2.1.tar.gz          php-json-ext-1.2.0
nagios_plugins              php-json-ext-1.2.0.tar.bz2
nagios-plugins-1.4.15       Readme.txt
nagios-plugins-1.4.15.tar.gz  rpmforge-release-0.3.6-1.el5.rf.i386.rpm
[root@localhost downloads]#

```

```
Tar -zxvf cacti-plugin-0.8.7d-PA-v2.4.zip
cd /usr/local/cacti-plugin-arch
cp pa.sql /var/www/html/cacti/
cd files-0.8.7d
cp -rf * /var/www/html/cacti/
```

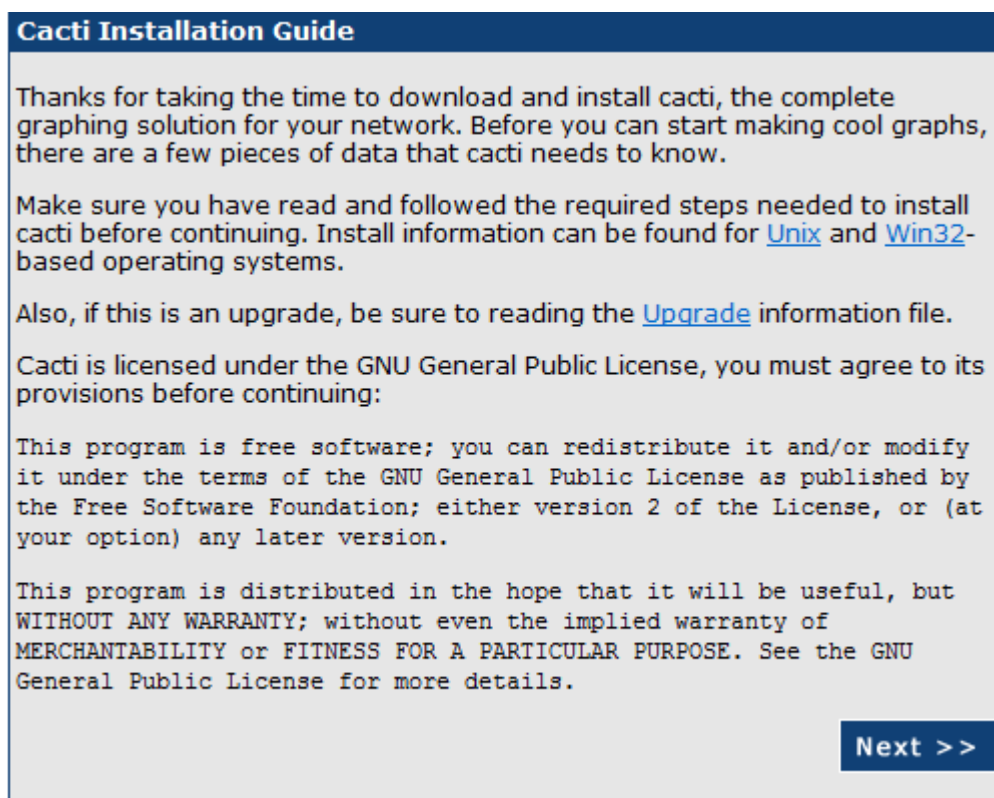
(10) Crontab -e

```
*/1 * * * * /usr/bin/php /var/www/html/cacti/poller.php&> /dev/null
```



(11) Cacti 的 web 初始化

IE浏览器输入<http://yourserverip/cacti>出现如下界面:



点击 Next

### Cacti Installation Guide

Please select the type of installation

New Install

The following information has been determined from Cacti's configuration file. If it is not correct, please edit 'include/config.php' before continuing.

Database User: cacti  
Database Hostname: localhost  
Database: cacti  
Server Operating System Type: unix

Next >>

点击 Next

usr/bin/snmpget

[OK: FILE FOUND]

**[FOUND] snmpbulkwalk Binary Path:** The path to your snmpbulkwalk binary.

/usr/bin/snmpbulkwalk

[OK: FILE FOUND]

**[FOUND] snmpgetnext Binary Path:** The path to your snmpgetnext binary.

/usr/bin/snmpgetnext

[OK: FILE FOUND]

**[FOUND] Cacti Log File Path:** The path to your Cacti log file.

/usr/local/apache/htdocs/log/cacti.log

[OK: FILE FOUND]

**SNMP Utility Version:** The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.

NET-SNMP 5.x

**RRDTool Utility Version:** The version of RRDTool that you have installed.

RRDTool 1.4.x

**NOTE:** Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

Finish

点击 Finish，安装完成。



## User Login

Please enter your Cacti user name and password below:

User Name:

Password:

默认用户名和密码

User : admin password : admin

需要强制修改密码:



## User Login

\*\*\* Forced Password Change \*\*\*

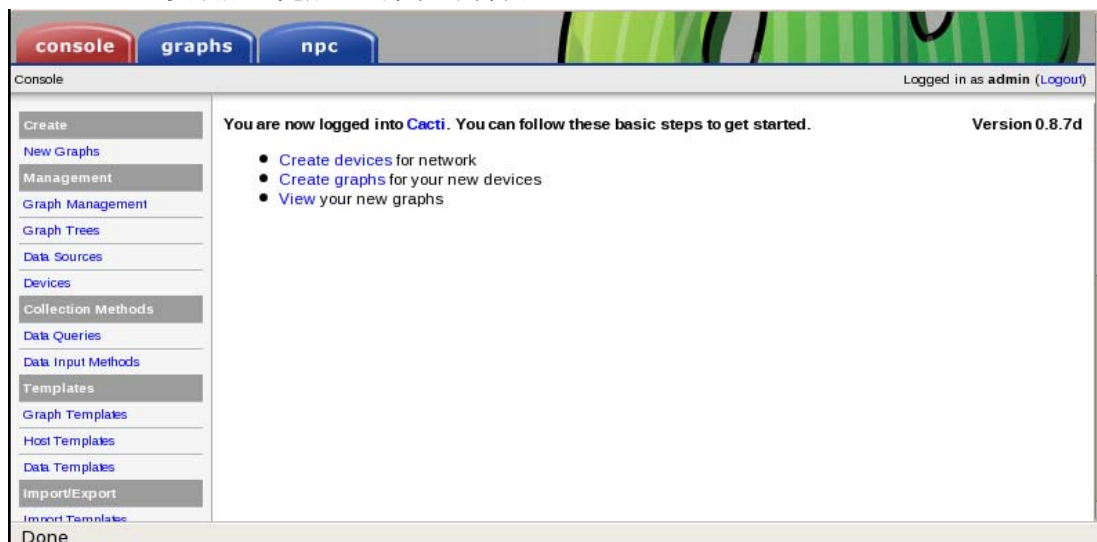
Please enter a new password for cacti:

Password:

Confirm:

输入两次新密码即可。

至此,, cacti 安装配置完成, 出现如下界面



五、 整合 Cacti 和 nagios

### (1) 安装 ndoutils

ndoutils 是将 Nagios 的配置及监控信息存储到数据库里，NPC 通过调用 ndo 所存储的数据来展现 Nagios 的信息。ndoutils 需要用到 mysql 的 mysql-lib 及 mysql-inc，因此需要安装 mysql-devel。

```
#yum -y install mysql-devel 安装mysql 开发包以编译ndoutils
#wget
http://nchc.dl.sourceforge.net/sourceforge/nagios/ndoutils-1.4b7.tar.gz
#tar zxvf ndoutils-1.4b7.tar.gz
#cd ndoutils
#./configure --prefix=/usr/local/nagios LDFLAGS=-L/usr/lib
--with-mysql-inc=/usr/include/mysql --with-mysql-lib=/usr/lib/mysql
--enable-mysql --disable-pgsql --with-ndo2db-user=nagios
--with-ndo2db-group=nagios
#make
#make install //此命令可不用执行
#./db/installldb -ucacti -pcacti -h localhost -d cacti
#cp config/ndomod.cfg /usr/local/nagios/etc
```

### (2) 修改 nagios 主配置文件

```
#vi /usr/local/nagios/etc/nagios.cfg
```

```
check_external_commands=1
command_check_interval=-1
event_broker_options=-1
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
process_performance_data=1

log_file=/usr/local/nagios/var/nagios.log
```

添加以上内容。

```
#cd ~/downloads
#cd ndoutils-1.4b7/src
#cp ndomod-3x.o ndo2db-3x log2ndo file2sock /usr/local/nagios/bin
#cd ..
#cp src/ndo2db-3x /usr/local/nagios/bin/ndo2db
#mv /usr/local/nagios/bin/ndomod-3x.o /usr/local/nagios/bin/ndomod.o
#cp config/ndo2db.cfg /usr/local/nagios/etc
```

修改配置文件 ndocmd.cfg 和 ndo2db.cfg，这里我的配置文件内容为：

```
[root@localhost src]# cat /usr/local/nagios/etc/ndomod.cfg |grep -v '^#' |sed
/^$/d
```

```
[root@localhost src]# cat /usr/local/nagios/etc/ndomod.cfg |grep -v '^#' |sed /^$/d
instance_name=default
output_type=tcpsocket
output=localhost
tcp_port=5668
output_buffer_items=5000
buffer_file=/usr/local/nagios/var/ndomod.tmp
file_rotation_interval=14400
file_rotation_timeout=60
reconnect_interval=15
reconnect_warning_interval=15
data_processing_options=-1
config_output_options=2
```

```
[root@localhost src]# cat /usr/local/nagios/etc/ndo2db.cfg |grep -v '^#' |sed
```

/^\$/d

```
[root@localhost src]# cat /usr/local/nagios/etc/ndo2db.cfg |grep -v '^#' |sed /^$/d
ndo2db_user=nagios
ndo2db_group=nagios
socket_type=tcp
socket_name=/usr/local/nagios/var/ndo.sock
tcp_port=5668
db_servertype=mysql
db_host=localhost
db_port=3306
db_name=cacti
db_user=cacti
db_pass=cacti
db_prefix=np_
max_timedevents_age=1440
max_systemcommands_age=10080
max_servicechecks_age=10080
max_hostchecks_age=10080
max_eventhandlers_age=44640
debug_level=0
debug_verbosity=1
debug_file=@localstatedir@/ndo2db.debug
max_debug_file_size=1000000
[root@localhost src]#
[root@localhost src]#
```

#/usr/local/nagios/bin/ndo2db -c /usr/local/nagios/etc/ndo2db.cfg 启动  
ndo2db

### (3) 安装 php-json

npc 展示部分用到 json, 需要在 php 中安装 php-json 的支持。

#cd ~/downloads

#wget http://www.aurore.net/projects/php-json/php-json-ext-1.2.0.tar.bz2

#tar xvjf php-json-ext-1.2.0.tar.bz2

#cd php-json-ext-1.2.0

#phpize 编译前初始化php 环境

#./configure

#make

#make install

#vi /etc/php.d/json.ini

extension=php\_json.so

#cp /usr/lib/php/modules/json.so /usr/lib/php/modules/php\_json.so 这一步骤  
很关键, 因为, apache 否则日志报错加载不到php\_json.so 文件。

#/usr/sbin/httpd -k graceful 重启apache

### (4) 安装npc

[root@localhost downloads]# tar -zxvf npc-2.0.3.tar.gz

#mv npc /var/www/html/cacti/plugins/

### (5) 将cacti插件管理包中的SQL导入cacti数据库

Cd /var/www/html/cacti

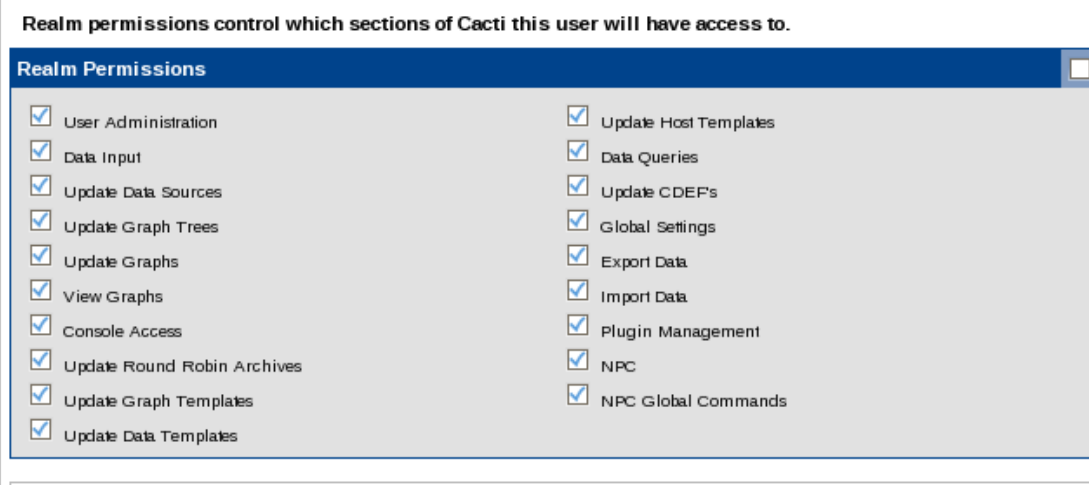
Mysql-u root -p cacti<pa.sql



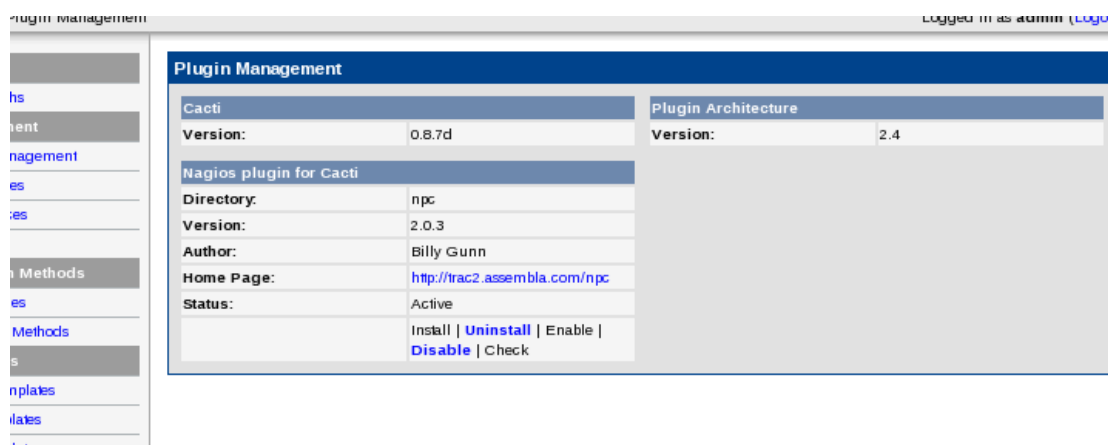


#### (6) 登录 WEB 页面配置 NPC

启用cacti 的插件功能,以admin 用户登陆cacti,在console 中的user management 里对admin 的用户权限进行编辑,勾选上Plugin Management, 然后到插件管理中心安装并启用NPC 即可。



在'User Management'中选择'admin',在下边'Realm Permissions'中勾选'Plugin Management'。这时右侧会出现'Plugin Management'连接, 进去后在'uninstalled'中安装npc, 然后在'intalled'中enable npc。



然后在回去admin 的'Realm Permissions'中会出现'use npc', 如果没有勾选就把它选中。

接着在右侧栏目中选择settings, 点选npc 的标签

勾选Remote Commands

Nagios Command File Path: /usr/local/nagios/var/rw/nagios.cmd

<这个文件启动nagios 后会产生, 根据实际的位置写>

Nagios URL: [url]http://yourserver/nagios/[url]

保存就可以了。/var/www/html/cacti

**Cacti Settings (NPC)**

**General Settings**

**Remote Commands**  
Allow commands to be written to the Nagios command file. ☒ Remote Commands

**Nagios Command File Path**  
The path to the Nagios command file (nagios.cmd).

**Nagios URL**  
The full URL to your Nagios installation (http://nagios.company.com/nagios/)

**Date Format**  
Select the format you want for displaying dates.

**Time Format**  
Select the format you want for displaying times.

**Portlet Refresh Rate**  
The amount of time in seconds to wait before the portlets refresh. The minimum is 30 seconds.

点击npc进行测试:

**Nagios Status**

Nagios	Notification	Host Checks	Service Checks
On	Enabled	Enabled	Enabled

**Host Status Summary**

Down	Unreacha	Up	Pending
2	0	2	0

**Service Status Summary**

Critical	Warning	Unknown	Ok	Pending
7	0	0	9	0

**Service Problems**

Host	Serv	Sta	Plugin Output
Hello host	check	CRITICAL	CRITICAL - Host Unreachable (192.168.11.137)
Hello host	check	CRITICAL	Connection refused or timed out

**Host Problems**

Host	Sta	Plugin Output
Hello	CRITICAL	CRITICAL - Host Unreachable (192.168.11.137)
Window	CRITICAL	CRITICAL - Host Unreachable (192.168.11.128)

至此，整合完成。

总结：经过这段时间的学习，nagios和cacti整合已经基本完成。在这个过程中遇到了很多问题，有的时候一个小小的问题就会阻止我很长时间，这使我感觉自己的linux知识还远远不够，但是在解决这些问题的过程当中使我学到了很多知识，自己的能力也得到了提升，这使我感到很欣慰。对于cacti和nagios整合还有很多监控功能没有实现，现在只做到了一些服务、磁盘、CPU等的监控功能，在接下来的时间我会继续专研，更深层的挖掘nagios和cacti整合之后的所有功能。