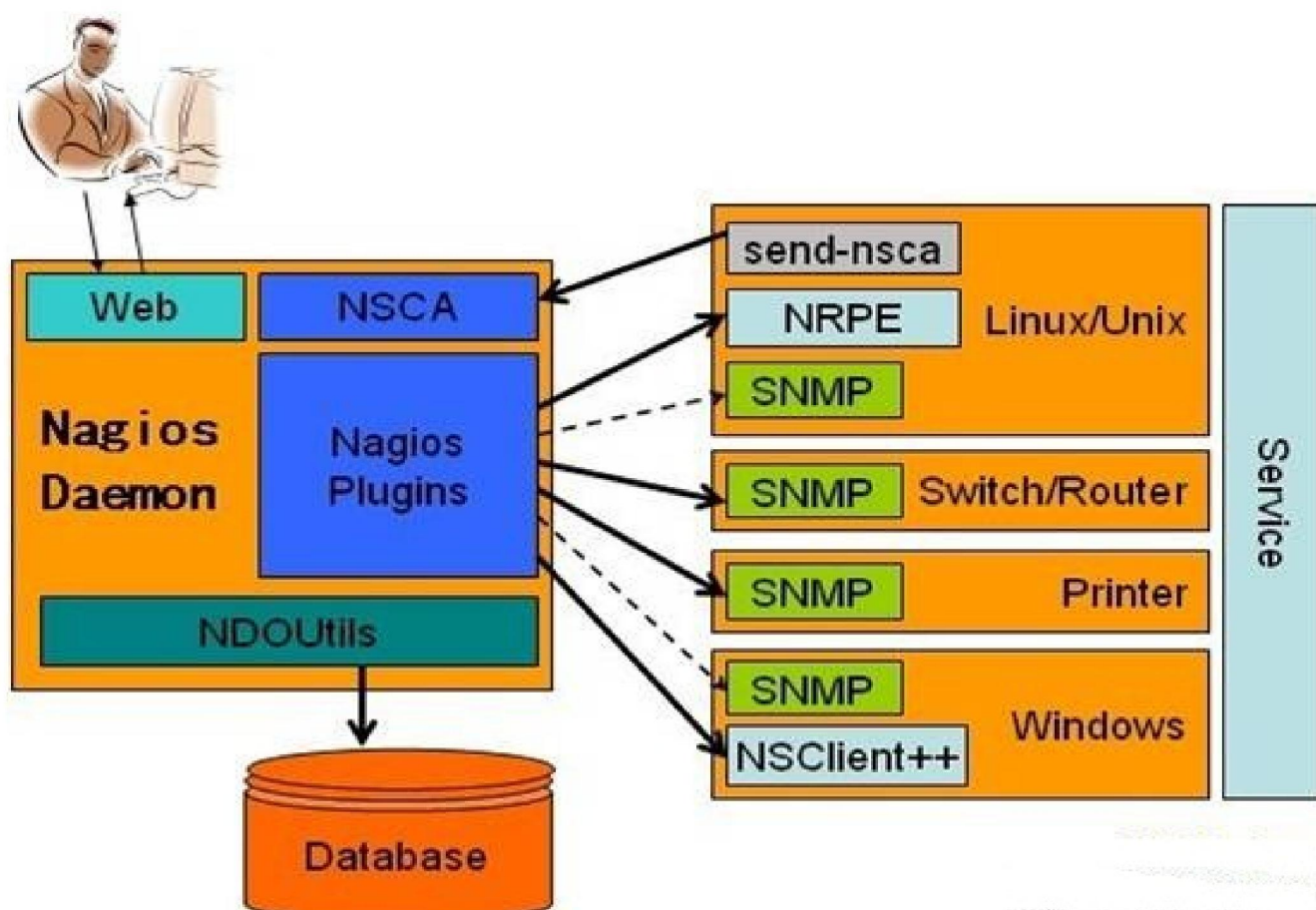


Nagios 安装部署与 Cacti 整合文档

系统环境: CentOS5.4	文档版本: V1.0.6	整理: Treasure
	更新时间: 2011-03-11	备注: 运维专用

Nagios 是一款开源的免费网络监视工具，其功能强大，灵活性强。能有效监控 Windows、Linux 和 Unix 的主机状态，交换机路由器等网络设置，打印机等。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实现的，因此，Nagios 和 Nagios-plugins 是服务器端工作所必须的组件。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
- (4) NDOUtils: 则用来将 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，NRPE 和 NSClient++工作于客户端，NDOUtils 工作于服务器端，而 NSCA 则需要同时安装在服务器端和客户端



目 录

Nagios 安装部署与 Cacti 整合文档	1
1. 文档更新记录	1
2. 文档说明	1
3. Nagios 的安装	1
3.1. 安装基础支持套件和添加用户	1
3.2. 安装 nagios	2
3.3. 安装 nagios-plugins	2
3.4. 安装 nagios-snmp-plugins	3
3.5. 其他设置	4
4. nagios 的配置	5
4.1. nagios 目录结构	5
4.2. nagios 的配置文件	5
5. 安装 cacti	13
6. 整合 cacti 和 nagios	15
6.1. 安装 ndoutils	15
6.2. 安装 php-json	17
6.3. 安装 npc	17
6.4. 测试页面	19
7. Nagios 每日健康检查报警短信	20
7.1. 编写检查脚本	20
7.2. 添加 crond 计划	21
7.3. 配置飞信机器人报警	21
7.3.1. Commands.cfg 配置文件添加如下内容:	21
7.3.2. Contacts.cfg 配置添加:	21
7.3.3. Templates.cfg	22
7.3.4. 修改展示页面监控图片大小: /usr/local/nagios/etc/pnp/config.php	22
8. Troubleshooting	23
8.1. web 界面修改某个服务时报错	23
8.2. 点击 host,service 选项时, 结果无法显示	23
8.3. nagios3.2.0 以后, 安装 nagios 在访问 http://ip/nagios 时出现如下错误提示:	24
8.4. 出现 pnp 小太阳图标, 点击报错如下:	24
8.5. 安装 NAGIOS 时发现有 Status Map、Alert Histogram 打不开链接, 提示找不到 statusmap.cgi 和	

histogram.cgi	25
8.6. 后台 apache 日志报错如下:	25
8.7. 进行编译安装 ndoutils-1.4b7 时, 报错如下:	26
8.8. 安装后, 查看 /usr/local/nagios/var/nagios.log 日志, 报错如下:	26
8.9. 有时开机后, 后台报错如下:	27
8.10. 访问 npc 插件页面时, 主机图标为红色叉号:	27
8.11. 访问点击小太阳后, 报错如下:	28

1. 文档更新记录

时间	修改人	版本号	修改说明
2010.09.07	Teasure	1.0.0	建立文档
2010.09.24	Teasure	1.0.1	添加报警设置
2011.01.07	Teasure	1.0.2	添加每日健康检查报警机制
2011.02.16	Teasure	1.0.3	更新文档生成 pdf 文档
2011.02.22	Teasure	1.0.4	添加新的 troubleshooting 项
2011.03.10	Teasure	1.0.5	添加 nagios 飞信机器人报警
2011.03.11	Teasure	1.0.6	更新 troubleshooting

2. 文档说明

本文档规范网络监控系统安装流程及安装内容，供运维人员在部署新的运营环境时参考使用。

3. Nagios 的安装

3.1. 安装基础支持套件和添加用户

nagios 需要一些基础支持套件才能运行，如 apache, gcc, glibc, gd 库等。

```
#yum install httpd
#yum install gcc
#yum install glibc glibc-common
#yum install gd gd-devel
#yum install php          nagios3.2.0 以后的版本必须安装 php，nagios 页面访问才正常
#/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
```

3.2. 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
#make install
#make install-init
#make install-config
#make install-commandmode
这时 nagios 基本已经安装完成，默认安装后的配置文件用于启动 nagios 是没有问题的。
#vi usr/local/nagios/etc/objects/contacts.cfg
修改 nagiosadmin 这行其中的邮件地址为你的 email 地址，以将报警邮件发到你的邮箱
#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
```

3.3. 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

3.4. 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


```
#!/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

3.5. 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, Centos 默认打开了 `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需要修改 `/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/
```

4. nagios 的配置

4.1. 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 的配置关系可以按照下图来做说明:

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。

4.2. nagios 的配置文件

首先我们配置下主配置文件, 为了目录的清晰我们自己建立一些目录来存放相关的配置文件, 比如建立 commands 目录存放命令, 建立 services 目录存放服务, 建立 hosts 目录存放主机, 在 nagios.cfg 文件中找到 cfg_file 的部分, 下面是一个示例, 实际部署的时候可以按照你自己的情况去设置:

```
cfg_file=/usr/local/nagios/etc/objects/commands.cfg 包含配置文件, 下同
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
cfg_file=/usr/local/nagios/etc/objects/switch.cfg
cfg_dir=/usr/local/nagios/etc/services 包含配置目录, 目录下所有 cfg 文件将被包含; 下同
cfg_dir=/usr/local/nagios/etc/hosts
cfg_dir=/usr/local/nagios/etc/commands
cfg_dir=/usr/local/nagios/etc/switches
cfg_dir=/usr/local/nagios/etc/routers
```

nagios.cfg 其他地方可以暂不调整, 该文件中具体参数请参考附件中的配置文件中文注释

这里先说明一下, 监控都是依靠插件去完成的, 举个例子说明, 如我们要监控 192.168.1.200 这台服务器的虚拟内存, 当达到 70%的时候状态为警告, 达到 90%的时候为严重。这个依靠 check_snmp_storage.pl 这个插件来完成。

```
[root@localhost etc]# /usr/local/nagios/libexec/check_snmp_storage.pl -H 192.168.1.200 -C mypublic -2 -m "^Virtual Memory$" -w 70 -c 90
Virtual Memory: 21%used(531MB/2472MB) (<70%) : OK
```

其中-H 参数表示主机地址, -C 参数表示 SNMP 团体名称, -2 表示使用 SNMP v2 协议, -w 为 warning(警告)状态的阈值, -c 表示为 critical (严重) 状态的阈值

要用 nagios 实现监控某主机, 就是要实现用 nagios 调用这样的命令即可。

配置 resource.cfg, 文件内容:

```
[root@localhost etc]# cat resource.cfg |grep -v '#' | sed /^$/d 查看 resource.cfg 配置
$USER1$=/usr/local/nagios/libexec
$USER7$=-C mypublic -2
```

这里其实是定义两个变量\$USER1\$和\$USER7\$, 因为在后面定义 command 的时候插件的路径是绝对路径, 为了节省输入直接可以用\$USER1\$代替/usr/local/nagios/libexec。而 SNMP 插件的命令中的参数需要设置 SNMP 相关的信息, 为节省输入这里用\$USER7\$来代替

-C mypublic -2。大家可对照上面检测 192.168.1.200 的虚拟内存的例子结合来加以理解。

下面我们来定义一个 command 用以监控, 仍然以在 nagios 中实现监控 192.168.1.200 这台主机的虚拟内存为例说明。为了目录结构的清晰, 我们将 command 的定义都放到 commands 目录下。在 commands 下建立一个 cfg 后缀的文件, 针对此示例的 command 定义内容为

```
define command{
    command_name    check_snmp_storage
    command_line     $USER1$/check_snmp_storage.pl -H $HOSTADDRESS$ $USER7$ $ARG1$ -w $ARG2$ -c $ARG3$
}
```

其中的\$USER1\$和\$USER7\$即为我们在上面的 resource.cfg 中定义的两个变量, \$HOSTADDRESS\$为我们在下面要定义的主机。

在 hosts 目录下建立一个 cfg 后缀的文件来存放主机配置。

```
define host{
    use                windows-server    定义使用的模板
    host_name          web83             定义主机名为 web83
    alias              web server on 111.83  主机别名
    address            192.168.1.200      主机 IP 地址
    hostgroups         linuxtoneweb       将该主机归到 linuxtoneweb 这个组, 如
    果要归到多个组里, 用逗号分隔组名
}
```

在定义主机的时候我们将 83 归到了 linuxtoneweb 这个组, 下面来定义这个组名。可以直接在 hosts 下建立一个定义组名的 cfg 文件。

```
define hostgroup{
    hostgroup_name    linuxtoneweb
    alias             linuxtone web servers
}
```

这样就定义了一个 linuxtoneweb 组, 将主机归为一个组有两种方法, 一种是我们这种在主机的定义中去设置, 另外一种是在组的设置中定义 members。如下:

```
define hostgroup{
    hostgroup_name    linuxtoneweb
    alias             linuxtone web servers
    members           web83    设置该组的成员, 需要是在 host 中定义的主机名, 多个
    成员请用逗号分隔
}
```

在定义主机分组里面我们还可以用 hostgroup_members 定义下级分组, 十分方便。

OK, 上面我们完成了命令、主机的定义, 下面我们就要进入 nagios 监控某个对象的关键设置, nagios 的监控都是靠定义一个 service 来实现, 这里我们定义一个 service 来实现监控 83 的虚拟内存。在 services 目录下建立一个 cfg 文件。

```

define service {
    hostgroup_name      linuxtone,linuxtoneweb,database  定义监控对象
    name                memory                          设置服务名
    service_description  check memory                  服务描述
    check_period         24x7                          监控周期设置
    max_check_attempts   4                            最大检测尝试次数
    normal_check_interval 3                            正常检测间隔时间
    retry_check_interval 2                            重试检测间隔时间
    contact_groups       admins                        报警联系组
    notification_interval 10                          通知间隔
    notification_period   24x7                        通知周期设置
    notification_options  w,u,c,r                    定义什么状态时报警
    check_command         check_snmp_storage!-m "^Virtual Memory$"!70!90
}

```

这样我们完成对一个 service 的定义，定义监控对象我们这里是定义了 3 个组，即属于这 3 个组的所有主机都会检测虚拟内存（注意：组必须要在其他文件中事先定义好）。当然你也可以用 host_name 针对主机作为检测对象（多主机用逗号分隔）。监控周期和报警周期我们设置的都是 24x7，这个会在后面去定义。定义报警状态中的 w 表示 warning，u 表示 unknown，c 表示 critical，r 表示 recovery（即恢复后是否发送通知）；报警选项一般生产环境下设置 w,c,r 即可。最后我们看到实现监控的主体部分

```
check_command      check_snmp_storage!-m "^Virtual Memory$"!70!90
```

定义实现这个检测的命令，check_snmp_storage 即为我们上面在 commands 里定义的命令名，!表示带入的参数，这里有三个！表示带入三个参数到 command 里，每个参数之间用！分隔，这里表示第一个参数为-m “^Virtual Memory\$”，第二个参数为 70，第三个参数为 90。下面我们回头去看看 check_snmp_storage 这个 command 的定义：

```
command_line      $USER1$/check_snmp_storage.pl -H $HOSTADDRESS$ $USER7$ $ARG1$ -w $ARG2$ -c $ARG3$
```

这里的\$ARG1\$即表示参数 1，\$ARG2 即表示参数 2，\$ARG3\$即表示参数 3
而\$HOSTADDRESS\$是我们定义的检测对象，它的值就是我们的主机 IP 地址；\$USER1\$和\$USER7\$为在 resource.cfg 里的定义。下面我们将参数值代入这个 command 里，最后实现的效果是：

```
/usr/local/nagios/libexec/check_snmp_storage.pl -H 192.168.1.200 -C mypublic -2 -m "^Virtual Memory$" -w 70 -c 90
```

而这个命令就是我们实现监控 83 的虚拟内存的命令！

设置 3 个参数是为了灵活的针对不同主机的调用，如果你对不同的主机设置的阈值不同，只需要对不同的主机定义不同的 service 即可，command 是可以通用的！而这里的参数 1 也是为了实现不同的功能而设置的，这个脚本还可以检测物理内存，如果我们把参数 1 的值设置为“^Physical Memory\$”就可以针对物理内存进行检测！定义 command 的时候参数的个数的设定是你可以根据需要设置的。只需要在后面 service 中能将参数值代到 command 中最终能实现你要检测的命令即可。到此为止你可以感受到 nagios 的灵活性了吧。

到这里我们基本实现了 nagios 的监控功能，下面我们继续来看看其他设置。在定义 host 的时候我们有个主机模板的设置，这个设置的定义在 object/template.cfg 中定义的，这是一个系统自带的配置文件，关于我们所使用的 Windows-Server 的模板定义内容为：

```
# Windows host definition template - This is NOT a real host, just a template!

define host{
    name                windows-server ; The name of this host template
    use                  generic-host   ; Inherit default values from the
generic-host template
    check_period         24x7           ; By default, Windows servers are
monitored round the clock
    check_interval       5              ; Actively check the server every 5
minutes
    retry_interval       1              ; Schedule host check retries at 1 minute
intervals
    max_check_attempts   10            ; Check each server 10 times (max)
    check_command         check-host-alive ; Default command to check if
servers are "alive"
    notification_period   24x7         ; Send notification out at any time - day or
night
    notification_interval 30           ; Resend notifications every 30 minutes
    notification_options  d,r         ; Only send notifications for specific host
states
    contact_groups        admins       ; Notifications get sent to the admins by
default
    hostgroups            windows-servers ; Host groups that Windows servers
```

```
should be a member of
    register          0          ; DONT REGISTER THIS - ITS JUST A
TEMPLATE
}
```

objects 目录下的 contact.cfg 中定义的是报警联系人。

```
define contact{
    contact_name      nagiosadmin      ; Short name of
user
    use               generic-contact   ; Inherit default
values from generic-contact template (defined above)
    alias             Nagios Admin     ; Full name of user

    email             netkey@linuxtone.com ; <<***** CHANGE THIS
TO YOUR EMAIL ADDRESS *****

    address1          alarm@hotmail.com ;这里是个自定义的变量
设置，定义了一个 SMS 地址用以接收 SMS 报警信息，联系人的自定义变量只能使用
address1-address6; 通过这个设置你可以实现多种报警方式，如电话，手机短信等，通过在这
里设置你的电话，手机号，然后到报警命令定义里定义一个命令即可。

}
```

其使用了 generic-contact 这个联系模板，关于这个模板 templates.cfg 里的定义内容为：

```
define contact{
    name              generic-contact   ; The name of this
contact template
    service_notification_period 24x7      ; service notifications
can be sent anytime
    host_notification_period 24x7        ; host notifications can
be sent anytime
    service_notification_options w,u,c,r,f,s ; send notifications for all
service states, flapping events, and scheduled downtime events
    host_notification_options d,u,r,f,s   ; send notifications for all
host states, flapping events, and scheduled downtime events
}
```

```
service_notification_commands    notify-service-by-email,notify-service-by-sms ;
send service notifications via email

host_notification_commands      notify-host-by-email,notify-host-by-sms ; send host
notifications via email

register                        0                        ; DONT REGISTER

THIS DEFINITION - ITS NOT A REAL CONTACT, JUST A TEMPLATE!

}
```

上面我在该配置文件中增加了 **notify-service-by-sms** 和 **notify-host-by-sms** 的两种报警命令

而关于报警的周期设置在 `objects` 下的 `timeperiods.cfg` 中, 如我们上面在 `service` 里定义的 7x24 的报警周期设置内容为:

```
define timeperiod{
    timeperiod_name 24x7
    alias           24 Hours A Day, 7 Days A Week
    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
}
```

通过这里我们可以灵活设置报警时间以和报警方式结合, 如周末和休息时间设置报警为发送短信和邮件, 工作时间设置为发送邮件和 SMS 报警方式。

下面我们来看关于报警命令的设置, 在 `objects` 下的 `command.cfg` 里定义的为

```
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 $SERVICESTATE$ ***" $CONTACTEMAIL$
}

# 'notify-host-by-sms' command definition
define command{
    command_name    notify-host-by-sms
    command_line    php /usr/local/nagios/share/sms/smssendmsg.php
$CONTACTADDRESS1$ "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState:      $HOSTSTATE$\nAddress:      $HOSTADDRESS$\nInfo:
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n"
}

# 'notify-service-by-sms' command definition
define command{
    command_name    notify-service-by-sms
    command_line    php /usr/local/nagios/share/sms/smssendmsg.php
$CONTACTADDRESS1$ "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService:      $SERVICEDESC$\nHost:      $HOSTALIAS$\nAddress:
$HOSTADDRESS$\nState:  $SERVICESTATE$\n\nDate/Time:  $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$"
}
```

定义了 4 种报警命令，后面两种 sms 报警是我添加的；邮件报警是使用系统自带的 sendmail 来发送邮件的（注意要启动 sendmail 服务），而 sms 报警是使用我们上面所配置的 sms 机器人发送的报警信息。

至此，整个 nagios 的监控和报警机制了解完毕。注意我们在配置过程中的任何改动都记得要去通过

/usr/local/nagios/bin/nagios -c /usr/local/nagios/etc/nagios.cfg 以检查配置文件是否正确, 如果正确则可以重启 nagios 加载新配置(service nagios restart)

5. 安装 cacti

Cacti 的运行需要 LAMP (Linux+Apache + MySQL + PHP) , RRDTool , net-snmp , AMP 环境我们在上面已经配置。

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
```

安装 net-snmp

```
# yum -y install net-snmp net-snmp-utils net-snmp-libs php-mysql
```

安装 cacti

```
#cd ~/downloads
```

```
#wget http://www.cacti.net/downloads/cacti-0.8.7d.tar.gz
```

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

```
#mkdir -p /usr/local/wwwroot
```

```
#cp -rf cacti-0.8.7.d /usr/local/wwwroot/cacti
```

下面开始导入数据库, mysql 数据库的基本安全配置这里不做说明, 请参考以前的 postfix 邮件安装文档中的 mysql 设置部分。

```
#mysql -u root -p
```

```
mysql>create database cacti; 创建一个数据库供 cacti 使用
```

```
mysql>use cacti;
```

```
Mysql>source /usr/local/cacti/cacti.sql 导入 mysql 数据库
```

```
mysql> grant all privileges on cacti.* to cacti@localhost identified by "cacti";
```

```
Query OK, 0 rows affected (0.03 sec)
```

添加一个数据库账号 **cacti** 密码为 **cacti** 用以访问 **cacti** 库

Mysql>flush privileges; 刷新权限信息

Apache 设置

#vi /etc/http/conf.d/cacti.conf 编辑 cacti 站点的配置文件，文件内容如下：

```
Alias /cacti "/var/www/html/cacti"
```

```
<Directory "/var/www/html/cacti">
```

```
Options FollowSymLinks MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

修改 cacti 的配置文件

#vi /usr/local/wwwroot/cacti/include/config.php

```
/* load up old style plugins here */
```

```
$plugins = array();
```

```
$url_path = "/cacti/";
```

修改其中的数据库连接信息，设置好数据库主机地址，用户，密码信息。

#chmod 777 -R rra log

至此安装完成，访问 <http://ip/cacti/install> 根据提示安装 cacti （ip 换成你对应的主机 ip 地址）

接下来安装 cacti 的插件管理

#cd ~/download/

#wget http://mirror.cactiusers.org/downloads/plugins/cacti-plugin-0.8.7d-PA-v2.4.zip

#unzip cacti-plugin-0.8.7d-PA-v2.4.zip

#mysql -u root -p cacti<pa.sql 导入 sql 到 cacti 库中

#cd files-0.8.7d/

#cp -rf * /usr/local/wwwroot/cacti 复制相关文件到 cacti 下

6. 整合 cacti 和 nagios

整合 cacti 和 nagios 是利用了 cacti 的一个插件 nagios for cacti, 它的原理是将 nagios 的数据通过 ndo2db 导入到 mysql 数据库 (cacti 的库中), 然后 cacti 读取数据库信息将 nagios 的结果展示出来。

6.1. 安装 ndoutils

首先需要安装 ndoutils 以将 nagios 的数据能导入到 mysql 数据库中。

```
#yum -y install mysql-devel    安装 mysql 开发包以编译 ndoutils
#wget http://nchc.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
修改 nagios 主配置文件
#vi /usr/local/nagios/etc/nagios.cfg
添加以下内容
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
添加的内容至此结束

#cd 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 //新添加

#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 downloads]# 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 downloads]# 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=1
```

```
debug_verbosity=1
debug_file=/usr/local/nagios/var/ndo2db.debug
max_debug_file_size=1000000
#/usr/local/nagios/bin/ndo2db -c /usr/local/nagios/etc/ndo2db.cfg    启动 ndo2db
```

6.2. 安装 php-json

npc 展示部分用到 json, 需要在 php 中安装 php-json 的支持, centos 5 下的 php 版本默认是 5.1.6 不带 json 支持, 无法简单地通过 yum install php5-json 方式来安装。

Yum install php-devel 否则, phpsize 命令不好用。

```
#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

为了验证是否成功可使用 phpinfo 查看是否已支持 json
```

6.3. 安装 npc

```
#cd ~/downloads
#wget
http://www.assembla.com/spaces/npc/documents/aUjAwmdW8r3BuPab7jnrAJ/download?filename=npc-2.0.0b.166.tar.gz
wget http://dlwt.csdn.net/fd.php?i=659714146741849&s=796b68562511c6534bfc15d7b04711f4/npc-2.0.3.tar.gz
新链接可用
```

```
#mv npc /usr/local/wwwroot/cacti/plugins/
```

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。

至此安装就完成了

启动 mysql httpd ndo nagios

```
service mysqld start
```

```
service httpd start
```

```
/usr/local/nagios/bin/ndo2db -c /usr/local/nagios/etc/ndo2db.cfg
```

```
service nagios start
```

访问 `[url]http://yourserver/cacti/[url]`

选择 npc 标签就可以看到 nagios 所检测的主机信息, 唯一不足的是我打开这个页面的速度有些慢, 需要进一步寻找原因来进行优化. npc 提供的这一套界面很好看 (个人感觉)。

点击 npc 时, 会出现如下错误提示:



再继续点击即可进入访问页面。

6.4. 测试页面

在 IE 浏览器的地址栏输入: <http://192.168.1.234/cacti>

用户名: admin 密码: admin

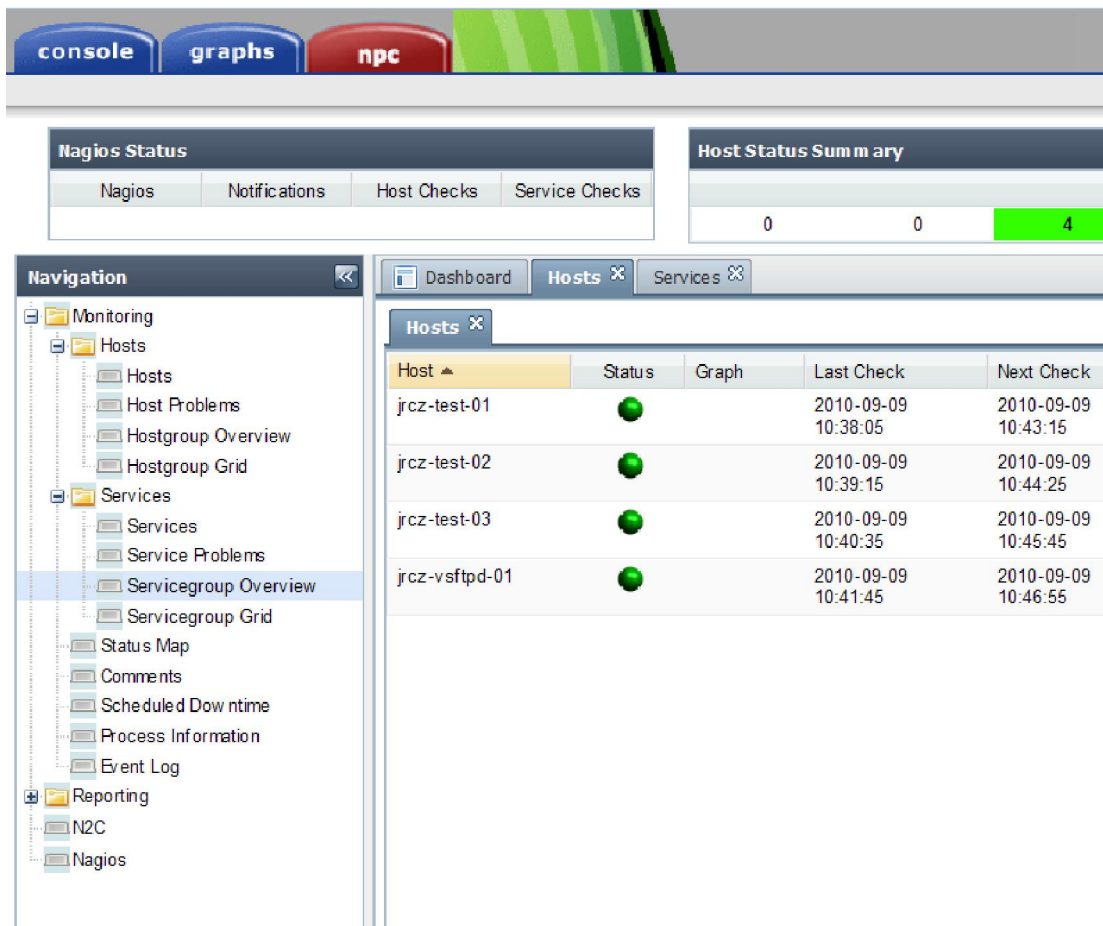
第一次登录的时候, 会提示修改管理员密码, 按照提示修改即可。

The screenshot displays the Nagios web interface. At the top, there are tabs for 'console', 'graphs', and 'npc'. Below these, the 'Nagios Status' section includes links for 'Nagios', 'Notifications', 'Host Checks', and 'Service Checks'. To the right, the 'Host Status Summary' table shows 0 Down, 0 Unreachable, and 4 Up hosts. The left sidebar contains a 'Navigation' menu with categories like Monitoring, Hosts, Services, Reporting, and Nagios. The main content area features two tables: 'Servicegroup: Service Status' and 'Servicegroup: Host Status'. Both tables show status counts for various servicegroups like 'hurge-disk', 'hurge-fms', 'hurge-ftp', 'hurge-load', and 'hurge-memcache'.

Down	Unreachable	Up
0	0	4

Servicegroup	Critical	Warning	Unknown	Ok	Pending
hurge-disk	0	0	0	4	0
hurge-fms	0	0	0	2	0
hurge-ftp	0	0	0	1	0
hurge-load	0	0	0	4	0
hurge-memcache	0	0	0	1	0

Servicegroup	Down	Unreachable	Up	Pending
hurge-disk	0	0	4	0
hurge-fms	0	0	1	0
hurge-ftp	0	0	1	0
hurge-load	0	0	4	0



至此完成 nagios 和 cacti 的整合。

7. Nagios 每日健康检查报警短信

对于没有移动短信网关通道来说，让监控平台每天下午 4:00 发一条短信，不管有没有故障都发，这样以便管理员能够知道短信报警及 nagios 服务是否正常。

检查报警的方法如下：

7.1. 编写检查脚本

```
# cat /root/sh/nagios_check.sh
#!/bin/bash
#auther by Teasure@cmcc.com.cn
#check nagios service
nid=/usr/local/nagios/var/nagios.lock
if [ -f $nid ]
```

```
then
/usr/local/nagios/libexec/sms/sendsms.sh 13800000000 "Nagios service is OK, Don't worry it!"
echo -e "nagios service is ok"
else
/etc/init.d/nagios start
/usr/local/nagios/libexec/sms/sendsms.sh 13800000000 " nagios service is restart,It's ok "
fi
```

7.2. 添加 crond 计划

crontab -e 添加如下内容:

```
00 16 * * * /root/sh/nagios_check.sh > /root/sh/nagios_check.log /dev/null 2>&1
```

7.3. 配置飞信机器人报警

7.3.1. Commands.cfg 配置文件添加如下内容:

```
#host-notify-by-sms
define command {
command_name host-notify-by-sms
command_line /usr/local/nagios/libexec/sms/sendsms.sh 13800000000 " ** $NOTIFICATIONTYPE$ Host Alert:
$HOSTNAME$ is AT: $DATE$ $HOSTSTATE$ ** "
}

#service-notify-by-sms
define command {
command_name service-notify-by-sms
command_line /usr/local/nagios/libexec/sms/sendsms.sh 13800000000 " ***
$NOTIFICATIONTYPE$ $HOSTNAME$ $DATE$ $TIME$ $SERVICEDESC$ is
$SERVICESTATE$ info:$SERVICEOUTPUT$ *** "
}
```

7.3.2. Contacts.cfg 配置添加:

```
define contact{  
    contact_name sms-members  
    use sms-mail-contact  
    alias Nagios Admin SMS  
    email    755792951@139.com  
    pager    13800000000  
}
```

```
define contactgroup{  
    contactgroup_name  admins  
    alias Nagios Administrators  
    members sms-members  
}
```

7.3.3. Templates.cfg

```
define contact{  
    name                sms-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-sms  
    host_notification_commands   notify-host-by-sms  
    register              0  
}
```

7.3.4. 修改展示页面监控图片大小: </usr/local/nagios/etc/pnp/config.php>

```
# vim /usr/local/nagios/etc/pnp/config.php
```

```
$conf['graph_width'] = "500";
```

```
$conf['graph_height'] = "100";
```

这两行是定义监控页面大小比例的。RRDTool graph Image Size

8. Troubleshooting

8.1. web 界面修改某个服务时报错

例如对某个服务进行临时安排其执行时间,或者不让它发警告,web 页面上都有这样的设置.但是常常会有错误信息如下:

```
Could not open command file '/usr/local/nagios/var/rw/nagios.cmd' for update!
The permissions on the external command file and/or directory may be incorrect. Read the FAQs on how
to setup proper permissions.
An error occurred while attempting to commit your command for processing.
```

关于这部分在 `nagios.cfg` 中有下面的内容

```
# EXTERNAL COMMAND FILE
# This is the file that Nagios checks for external command requests.
# It is also where the command CGI will write commands that are submitted
# by users, so it must be writeable by the user that the web server
# is running as (usually 'nobody'). Permissions should be set at the
# directory level instead of on the file, as the file is deleted every
# time its contents are processed.
这段话的核心意思是 apache 的运行用户要有对文件写的权限.权限应该设置在目录上,因为每次文件的内容被处理后文件就会被删掉

command_file=/usr/local/nagios/var/rw/nagios.cmd
```

本来将 `apache2` 运行的用户 `apache` 加到 `nagios` 组就应该可以了的
但是这个却不行,就将 `rw` 这个目录及其子文件的权限改了 `777`,这样就可以了.

8.2. 点击 host,service 选项时, 结果无法显示

安装 `nagios` 后, 访问页面可以, 点击 `host,service` 选项时, 都是无法显示. 后台日志报错:

```
[Wed Sep 01 17:31:32 2010] [error] [client 222.128.103.52] Premature end of script headers: status.cgi, referer: http://public.ipaddr/nagios/side.php
[Wed Sep 01 17:31:33 2010] [error] [client 222.128.103.52] (13)Permission denied: exec of '/usr/local/nagios/sbin/status.cgi' failed, referer: http://public.ipaddr/nagios/side.php
```

解决方法: 原因是因为开启了 selinux, getenforce

令 SELinux 处于容许模式

setenforce 0

如果要永久性变更它, 需要更改/etc/selinux/config 里的设置并重启系统。

不关闭 SELinux 或是永久性变更它的方法是让 CGI 模块在 SELinux 下指定强制目标模式:

```
chcon -R -t httpd_sys_content_t /usr/local/nagios/sbin/
```

```
chcon -R -t httpd_sys_content_t /usr/local/nagios/share/
```

关闭即可。

8.3. nagios3.2.0 以后, 安装 nagios 在访问 <http://ip/nagios> 时出现如下错误提示:

Forbidden

You don't have permission to access /nagios/ on this server.

Apache/2.2.3 (CentOS) Server at 192.168.1.243 Port 80

解决方法如下: nagios3.2.0 将页面从之前的 html 换成了 php, 首次安装需要先决条件 php
yum install php 即可

8.4. 出现 pnp 小太阳图标, 点击报错如下:

Initialising

Using /usr/local/nagios/share/perfdata/

RRDTool /usr/bin/rrdtool found.

RRDTool /usr/bin/rrdtool is executable

PHP Function proc_open is enabled

PHP Function fpassthru is enabled

PHP Function xml_parser_create is enabled

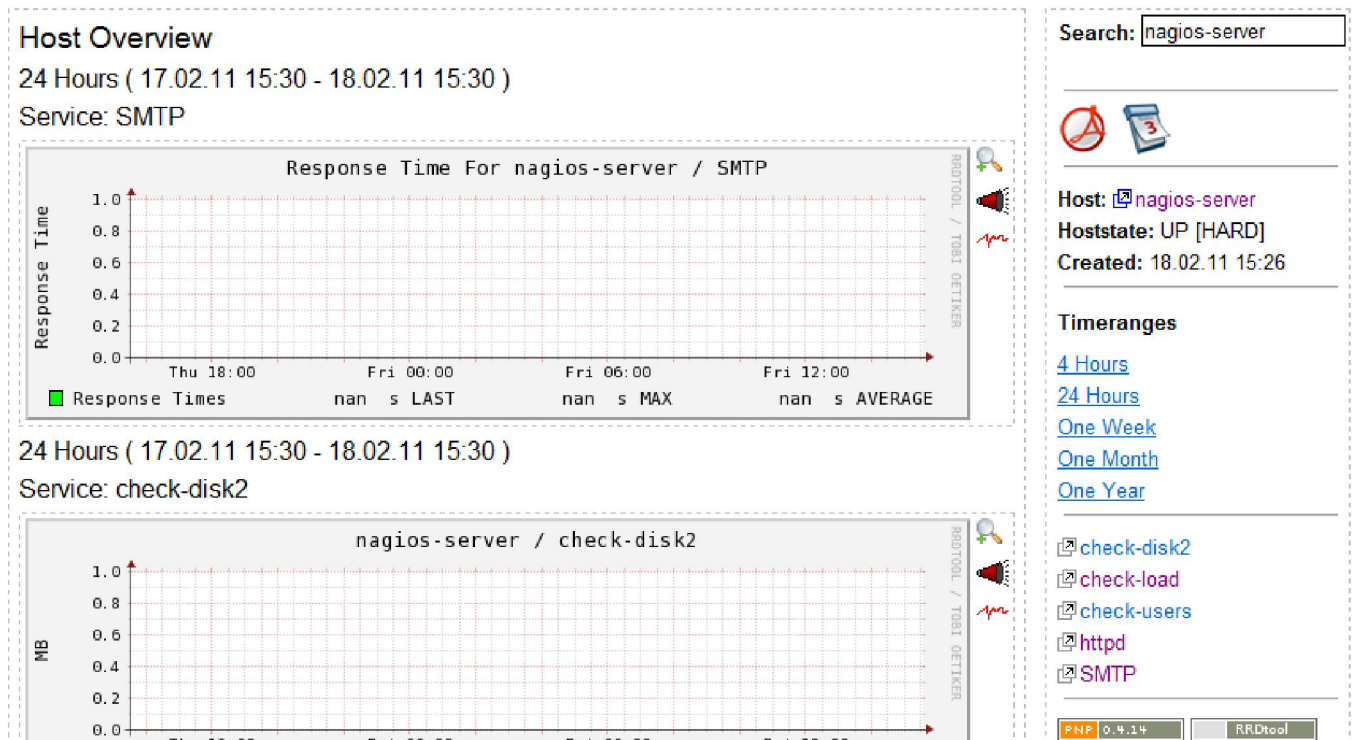
PHP zlib Support found.

PHP GD Support can't found.

解决方法: **yum -y install php-gd**

service httpd restart

再次点击小太阳图标时, 出现如下页面, 则表示正常:



8.5. 安装 NAGIOS 时发现有 Status Map、Alert Histogram 打不开链接，提示找不到 statusmap.cgi 和 histogram.cgi.

解决办法:

原因一: 因为 gd-devel 没有安装的问题, 造成 NAGIOS 在编译时不生成这 statusmap.cgi

原因二: NAGIOS 在编译在前, gd-devel 安装在后, 造成不生成这 statusmap.cgi

8.6. 后台 apache 日志报错如下:

```
# tail -f /etc/httpd/logs/error_log
```

```
[Fri Feb 18 19:07:18 2011] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
```

```
[Fri Feb 18 19:07:18 2011] [notice] Digest: generating secret for digest authentication ...
```

```
[Fri Feb 18 19:07:18 2011] [notice] Digest: done
```

```
[Fri Feb 18 19:07:18 2011] [notice] Apache/2.2.3 (CentOS) configured -- resuming normal operations
```

```
[Fri Feb 18 19:07:20 2011] [error] [client 127.0.0.1] Directory index forbidden by Options directive: /var/www/html/
```

```
[Fri Feb 18 19:07:42 2011] [error] [client 127.0.0.1] Directory index forbidden by Options directive: /var/www/html/
```

```
[Fri Feb 18 19:07:55 2011] [error] [client 127.0.0.1] Directory index forbidden by Options directive: /var/www/html/
```

监控 http 服务出现响应超时的情况，如下所示：

```
# /usr/local/nagios/libexec/check_http -I localhost -w 15 -c 20 -t 30
```

```
HTTP WARNING: HTTP/1.1 403 Forbidden - 5240 bytes in 0.003 second response time  
|time=0.002991s;15.000000;20.000000;0.000000 size=5240B;;;0
```

解决方法：

```
# echo -n none > /var/www/html/index.html
```

8.7. 进行编译安装 ndoutils-1.4b7 时，报错如下：

```
#./db/installdb -uacti -pacti -d cacti
```

```
DBD::mysql::db do failed: Table 'cacti.nagios_dbversion' doesn't exist at ./db/installdb line 51.
```

命令使用错误，解决方法如下：

```
# ./installdb -uacti -pacti -h localhost -d cacti //加上 -h localhost 参数
```

```
DBD::mysql::db do failed: Table 'cacti.nagios_dbversion' doesn't exist at ./installdb line 51.
```

```
** Creating tables for version 1.4b7
```

```
Using mysql.sql for installation...
```

```
** Updating table nagios_dbversion
```

```
Done!
```

8.8. 安装后，查看 /usr/local/nagios/var/nagios.log 日志，报错如下：

```
#tail -f /usr/local/nagios/var/nagios.log
```

```
[1298198680] Error: Could not safely copy module '/usr/local/nagios/bin/ndomod.o'. The module will not be loaded:  
No such file or directory
```

```
[1298202280] Auto-save of retention data completed successfully.
```

原因为：前面安装 ndoutils-1.4b7，少了一个操作步骤。解决办法如下：

```
# mv /usr/local/nagios/bin/ndomod-3x.o /usr/local/nagios/bin/ndomod.o //新添加
```

正确的日志如下：

```
# tail -f /usr/local/nagios/var/nagios.log
[1298346735] Event broker module '/usr/local/nagios/bin/ndomod.o' deinitialized successfully.
[1298346735] Nagios 3.2.1 starting... (PID=13489)
[1298346735] Local time is Tue Feb 22 11:52:15 CST 2011
[1298346735] LOG VERSION: 2.0
[1298346735] ndomod: NDOMOD 1.4b9 (10-27-2009) Copyright (c) 2009 Nagios Core Development Team and
Community Contributors
[1298346735] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1298346735] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
[1298350335] Auto-save of retention data completed successfully.
[1298353935] Auto-save of retention data completed successfully.
[1298357535] Auto-save of retention data completed successfully.
```

8.9. 有时开机后，后台报错如下：

```
# tail -f /usr/local/nagios/var/nagios.log

[1298439477] ndomod: Still unable to connect to data sink. 23512 items lost, 5000 queued items to flush.
[1298439493] ndomod: Still unable to connect to data sink. 23590 items lost, 5000 queued items to flush.
```

以上报错一般是由于 ndo2db 没有启动，手动启动即可：






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

8.10. 访问 npc 插件页面时，主机图标为红色叉号：

解决办法如下：

```
# cp -r /usr/local/nagios/share/images/logos/logo.gif /var/www/html/cacti/plugins/npc/logo.gif
```

重新刷新页面即可解决问题。正常页面为：

Host ▲	Status	Graph	Last Check	Next Check	Plugin Output
 localhost			2011-02-23 01:23:38	2011-02-23 01:28:48	PING OK - Packet loss = 0%, RTA = 0.04 ms
 nagios-server			2011-02-23 01:25:18	2011-02-23 01:30:28	PING OK - Packet loss = 0%, RTA = 0.04 ms

8.11. 访问点击小太阳后，报错如下：

```
OK Initialising
OK Using /usr/local/nagios/share/perfdata/
OK RRDTool /usr/bin/rrdtool found.
OK RRDTool /usr/bin/rrdtool is executable
OK PHP Function proc_open is enabled
OK PHP Function fpassthru is enabled
OK PHP Function xml_parser_create is enabled
OK PHP zlib Support found.
OK PHP GD Support found.
OK RRD Base Directory /usr/local/nagios/share/perfdata/ found.
OK Hostname nagios.com.cn$ is set.
! Directory /usr/local/nagios/share/perfdata/nagios.com.cn$ not found.
```

Hostname is not set:是 pnp 的提示，pnp 需要以以下方式访问 `index.php?host=$HOSTNAME$&srv=$SERVICEDESC$` 或者 `index.php?host=$HOSTNAME$`；

而通过脚本推送时，变量发生了变化，生成的文件如下：

```
#define_host
    define host {
        name          host-pnp
        register      0
        process_perf_data 1
        action_url /nagios/pnp/index.php?host=nagios.com.cn$    这样不正确的
        action_url /nagios/pnp/index.php?host=$HOSTNAME$        //这是正确格式
    }

#define_service
define service {
    name          srv-pnp
    register      0
    process_perf_data 1
    action_url /nagios/pnp/index.php?host=nagios.com.cn$&srv=$    这样是不正确的
    action_url /nagios/pnp/index.php?host=$HOSTNAME$&srv=$SERVICEDESC$ //正确格式
```

}