

Nagios 安装与配置

网海过客

Blog:www.chinasa.net

目录

Nagios 安装与配置	1
环境	3
说明	3
Nagios 监控架构图	3
软件下载	4
Nagios 服务端安装	4
Nagios 安装	4
Nagios-plugins 安装	4
Nrpe 安装	4
Nagios 与 Apache 整合	5
Nagios 客户端安装	5
Nagios-plugins 安装	5
Nrpe 安装	6
PNP4Nagios 插件安装	6
PNP4Nagios 安装	6
PNP4Nagios 与 Nagios 整合	6
PNP4Nagios 与 Apache 整合	8
Nagios 与 PNP4Nagios 载图	9
Nagios 配置	10
配置文件说明	10
配置 nagios.cfg 文件	10
配置 commands.cfg 文件	11
添加联系人	11

Nagios 配置验证	11
Nagios 启动	12
Nrep 配置	12
nrpe 启动	12
Nagios 配置实例	13
监控主机	13
监控服务-磁盘分区监控	14
监控服务-端口监控	15
监控服务-http 状态监控	16
添加主机组	17
添加服务组	17
添加手机短信报警	18
Mysql 监控	19
Mysql 主从监控	21
在主 mysql 数据库上创建监控账号	21
修改 nrpe.cfg 配置文件	22
在 nagios 服务端创建监控 mysql 主从配置服务文件	22
Nagios 部份功能载图	23
拓扑图	23
主机	23
主机组	24
服务	24
服务组	25
PNP4Nagios 监控图形	25

Nagios 是一个监视系统运行状态和网络信息的监视系统。Nagios 能监视所指定的本地或远程主机以及服务，同时提供异常通知功能等，Nagios 可运行在 Linux/Unix 平台之上，同时提供一个可选的基于浏览器的 WEB 界面以方便系统管理人员查看网络状态，各种系统问题，以及日志等。

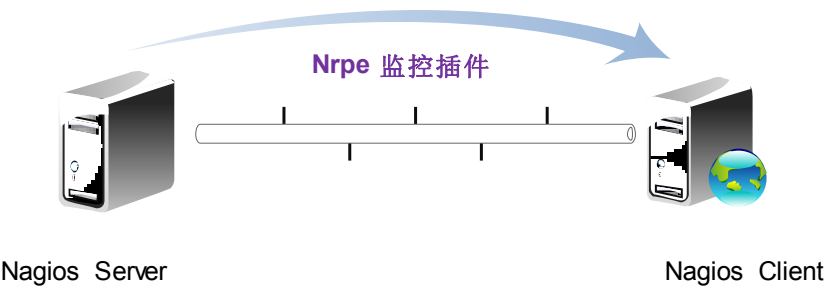
环境

序号	系统	IP	监控类别	软件
1	Centos 5.3 x32	192.168.0.125	服务端	Apache、nagios-cn-3.2.0.tar.bz2、nagios-plugins-1.4.15.tar.gz、nrpe-2.12.tar.gz、pnp4nagios-0.6.4.tar.gz、rrdtool-1.4.4.tar.gz
2	Centos 5.3 x32	192.168.0.126	客户端	nagios-plugins-1.4.14.tar.gz、nrpe-2.12.tar.gz

说明

Apache	实现 nagios 监控通过 web 查看
Nagios	nagios 监控程序
Nagios-plugins	nagios 监控插件
Pnp4nagios	nagios 监控插件
Nrpe	nagios 监控客户端
Rrdtool	pnp4nagios 生成图形工具

Nagios 监控架构图



软件下载

Nagios <http://sourceforge.net/projects/nagios/files/nagios-3.x/nagios-3.2.1/nagios-3.2.1.tar.gz>
Nagios-plugins <http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz>
Pnp4nagios <http://sourceforge.net/projects/pnp4nagios/files/PNP-0.6/pnp4nagios-0.6.6.tar.gz/download>
Nrpe <http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz>

Nagios 服务端安装

Nagios 安装

```
# useradd nagios -s /sbin/nologin
# tar jxvf nagios-cn-3.2.0.tar.bz2
# cd nagios-cn-3.2.0
# ./configure --prefix=/usr/local/nagios \
              --with-nagios-user=nagios \
              --with-nagios-group=nagios
# make all
# make install
# make install-commandmode
# make install-config
```

Nagios-plugins 安装

```
# tar zxvf nagios-plugins-1.4.15.tar.gz
# cd nagios-plugins-1.4.15
# ./configure --prefix=/usr/local/nagios
# make
# make install
```

Nrpe 安装

```
# tar zxvf nrpe-2.12.tar.gz
# cd nrpe-2.12
# ./configure --prefix=/usr/local/nrpe
# make
# make install
```

权限设置

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

Nagios 与 Apache 整合

在 apache 的配置 httpd.conf 中加入 include conf/nagios.conf
nagios.conf 内容如下:

```
Alias /nagios "/usr/local/nagios/share"  
<Directory "/usr/local/nagios/share">  
    Options None  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>  
ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"  
<Directory "/usr/local/nagios/sbin">  
    Options ExecCGI  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Nagios 客户端安装

Nagios-plugins 安装

```
# tar zxvf nagios-plugins-1.4.15.tar.gz  
# cd nagios-plugins-1.4.15  
# ./configure --prefix=/usr/local/nagios  
# make  
# make install
```

Nrpe 安装

```
# useradd nagios -s /sbin/nologin
# tar zxvf nrpe-2.12.tar.gz
# cd nrpe-2.12
# ./configure --prefix=/usr/local/nrpe
# make
# make install
```

权限设置

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

PNP4Nagios 插件安装

PNP4Nagios 安装

PNP4Nagios 用到了 RRDTool，需要先安装。

```
# yum install rrdtool
# tar zxvf pnp4nagios-0.6.4.tar.gz
# cd pnp4nagios-0.6.4
# ./configure --prefix=/usr/local/pnp4nagios \
              --with-nagios-user=nagios \
              --with-nagios-group=nagios

# make all
# make install
# make install-webconf
# make install-config
# make install-init
# make fullinstall
```

PNP4Nagios 与 Nagios 整合

修改 nagios 配置文件,vi /usr/local/nagios/etc/nagios.cfg，将相应配置修改为以下内容：

```
process_performance_data=1
host_perfdata_command=process-host-perfdata
service_perfdata_command=process-service-perfdata
```

修改 nagios 配置文件 commands.cfg 文件，
vi /usr/local/nagios/etc/objects/commands.cfg 将 process-host-perfdata 和
process-service-perfdata 修改为以下内容：

```
define command {
    command_name    process-host-perfdata
    command_line    /usr/bin/perl
                   /usr/local/pnp4nagios/libexec/process_perfdata.pl -d HOSTPERFDATA
}

define command {
    command_name    process-service-perfdata
    command_line    /usr/bin/perl
                   /usr/local/pnp4nagios/libexec/process_perfdata.pl
}
```

修改 nagios 配置文件，templates.cfg，
vi /usr/local/nagios/etc/objects/templates.cfg，在末尾添加以下内容：

```
define host {
    name            host-pnp
    register        0
    action_url      /pnp4nagios/index.php?host=$HOSTNAME$
}

define service {
    name            srv-pnp
    register        0
    action_url      /pnp4nagios/index.php?host=$HOSTNAME$&srv=$SERVICEDESC$
}
```

让 PNP4Nagios 显示图形在 nagios 的主机中或服务中显示，加入 host-pnp, srv-pnp 配置实例

```
define host{
    use                linux-server, host-pnp
    host_name          localhost
    alias              localhost
    address            127.0.0.1
}

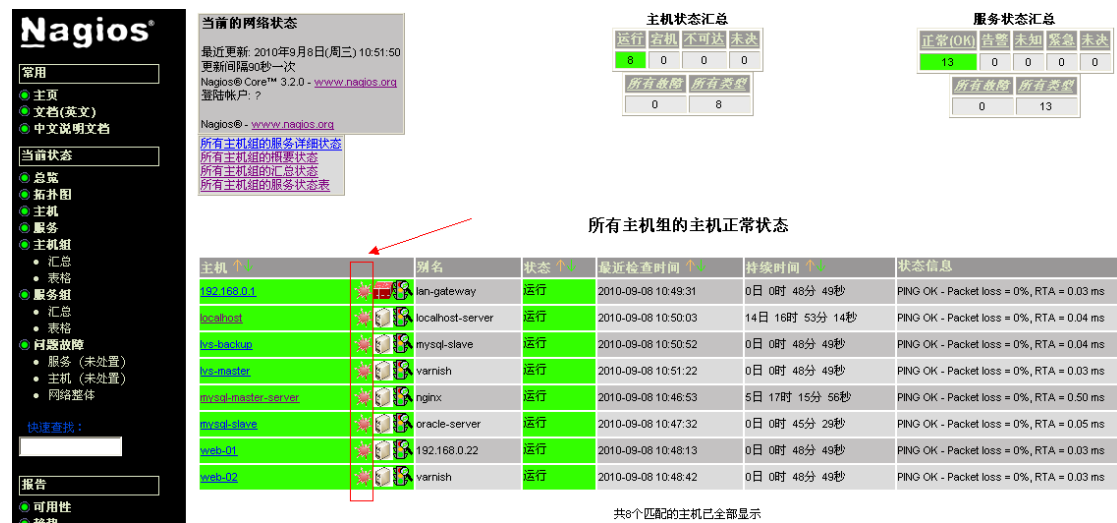
define service{
    use                local-service, srv-pnp
    host_name          localhost
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
}
```

PNP4Nagios 与 Apache 整合

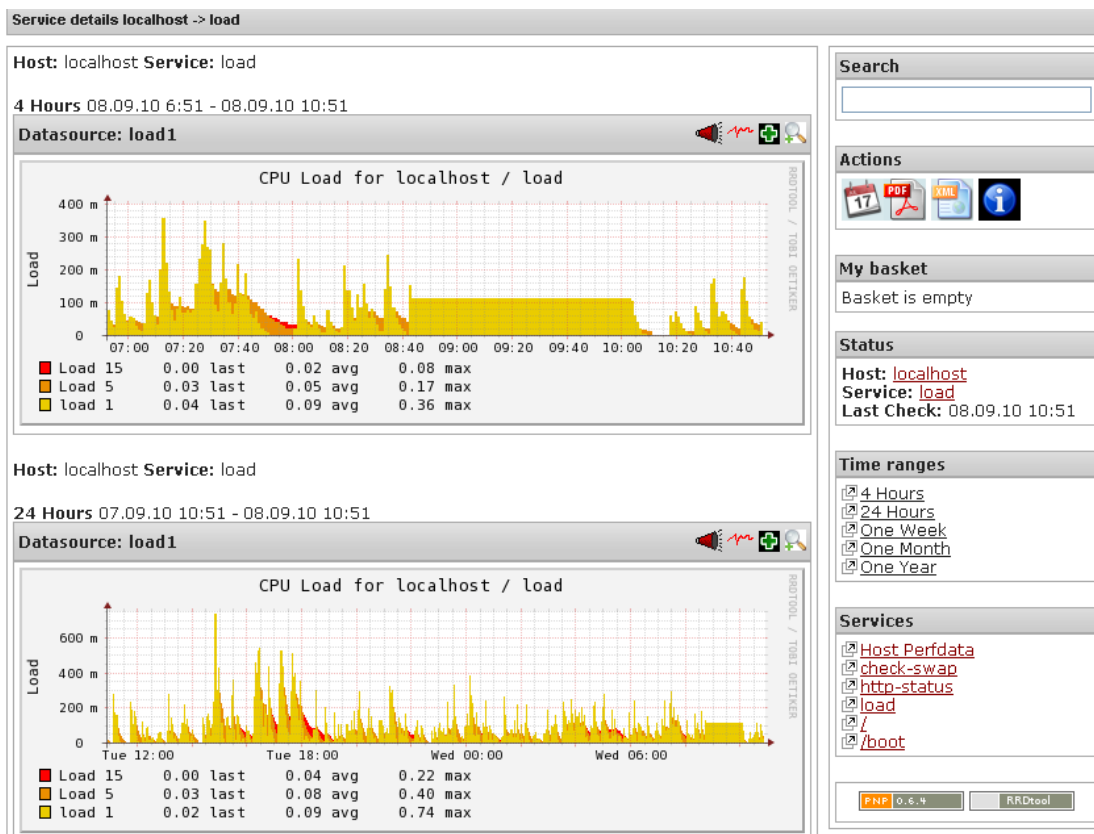
在 apache 的配置 httpd.conf 中加入 include conf/ pnp4nagios.conf
pnp4nagios.conf 内容如下：

```
Alias /pnp4nagios "/usr/local/pnp4nagios/share"
<Directory "/usr/local/pnp4nagios/share">
    AllowOverride None
    Order allow,deny
    Allow from all
    <IfModule mod_rewrite.c>
        RewriteEngine On
        Options FollowSymLinks
        RewriteBase /pnp4nagios/
        RewriteRule ^(application|modules|system) - [F,L]
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule .* index.php/$0 [PT,L]
    </IfModule>
</Directory>
```


Nagios 与 PNP4Nagios 载图



上图中如出现红色箭头所指的图标，表示 PNP4Nagios 与 Nagios 整合成功。点击红色图标，将会出现以下图形数据。



Nagios 配置

配置文件说明

```
# tree /usr/local/nagios/etc/
/usr/local/nagios/etc/
|-- cgi.cfg                #cgi 配置文件
|-- linux                  #监控服务器自定义文件夹
|   |-- hosts              #自定义监控主机文件夹
|   |   |-- group.cfg      #自定义主机组配置文件
|   |   |-- localhost.cfg  #自定义主机配置文件
|   |   `-- services       #自定义监控主机服务文件夹
|       |-- localservice.cfg #自定义监控服务配置文件
|       `-- sevicegroup.cfg #自定义监控服务组配置文件
|-- nagios.cfg             #nagios 主配置文件
|-- objects
|   |-- commands.cfg       #命令配置文件
|   |-- contacts.cfg       #联系人配置文件
|   |-- templates.cfg      #组，监控时间段等配置文件
|   |-- timeperiods.cfg
`-- resource.cfg           #定义插件路径文件
```

配置 `cgi.cfg` 文件，将 `user_authentication` 修改为 0。

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

```
use_authentication=0
```

不然访问会出现以下错误提示：

`It appears as though you do not have permission to view information for any of the hosts 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.

配置 `nagios.cfg` 文件

增加以内容：

```
cfg_dir=/usr/local/nagios/etc/linux
```

配置 `commands.cfg` 文件

增加以下内容：

```
define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c
$ARG1$
}
```

添加联系人

配置 `contacts.cfg` 文件，添加以下内容。

```
define contact{
    contact_name          nagiosadmin
    use                   generic-contact
    alias                 Nagios Admin
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,u,r
    service_notification_commands notify-service-by-email,notify-service-by
-sms
    host_notification_commands notify-host-by-email
    email                 test@163.com
}
```

#email 定义联系人报警 mail 地址

#notify-service-by-email、notify-service-by-sms、notify-host-by-email、要与 `commands.cfg` 命令配置文件里的相对应

Nagios 配置验证

验证 nagios 配置文件是否正确，可使用以下命令验证：

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

Checking global event handlers...

Checking obsessive compulsive processor commands...

Checking misc settings...

Total Warnings: 0

Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

如果上面的 Total Errors 为 0 表示配置文件正确。

Nagios 启动

/etc/init.d/nagios start

stop	停止
restart	重启
reload	重新加载配置文件

验证客户端 nrpe 是否能正常连接

```
# /usr/local/nrpe/libexec/check_nrpe -H 192.168.1.126
```

NRPE v2.12

如出现 NRPE v2.12 说明 nrpe 服务正常。

Nrpe 配置

修改 nrpe.cfg 配置文件，vi /usr/local/nrpe/etc/nrpe.cfg，内容如下：

server_port=5666	#nrpe 端口
server_address=192.168.0.125	#nrpe 服务器 IP 地址
allowed_hosts=127.0.0.1,192.168.0.125	#允许访问的 IP 地址
nrpe_user=nagios	#nrpe 用户
nrpe_group=nagios	#nrpe 用户组

nrpe 启动

```
# /usr/local/nrpe/bin/nrpe -c /usr/local/nrpe/etc/nrpe.cfg -d
```

验证 nrpe 是否启动成功

```
# netstat -ntlp | grep :5666
tcp    0    0 192.168.0.125:5666  0.0.0.0:*    LISTEN    17898/nrpe
```

如果结果中有以上数据，表示启动成功。否则检查配置文件。

Nagios 配置实例

监控主机

在 nagios 的 etc/linux 目录创建 host 文件夹，并在该目录创建 v1.cfg 配置文件,内容如下：

mkdir /usr/local/nagios/etc/linux/host

vi /usr/local/nagios/etc/linux/host/v1.cfg

```
define host {
    host_name          192.168.0.126
    alias               web
    address             192.168.0.126
    check_command       check-host-alive
    use                 linux-server,generic-host,host-pnp
    max_check_attempts 10
    check_period        24x7
    event_handler       notify-host-by-email
    process_perf_data   1
    contact_groups      admins
    notification_interval 20
    notification_period 24x7
    notification_options d,r
    icon_image          web.gif
    statusmap_image     web.gd2
    2d_coords           10,200
    3d_coords           10,200,200
}
```

注： host_name	主机名
alias	别名
address	IP 地址
check_command	监控命令
use	使用模版
max_check_attempts	最大检测次数
event_handler	事件通知方式
process_perf_da	是否生成数据图表(1 表是是,0 表示否)
contact_groups	联系组
notification_interval	通知时间间隔
notification_period	监控时间段
notification_options	报警通知设置(d 表示主机宕机， r 表示主机恢复)
icon_image	主机图标
statusmap_image	主机拓扑图标

2d_coords	2D 坐标
3d_coords	3D 坐标
use	linux-server,generic-host,host-pnp 这里要加入 host-pnp,主机才会有显示 PNP4Nagios 生成的图形。

监控服务-磁盘分区监控

(监控客户端 192.168.0.126 的/opt 分区大小)

在 nagios 的 etc/linux 目录创建 service 文件夹, 并在该目录创建 s1.cfg 配置文件, 内容如下:

```
mkdir /usr/local/nagios/etc/linux/service
```

```
vi /usr/local/nagios/etc/linux/service/s1.cfg
```

```
define service {
    host_name          192.168.0.126
    service_description /opt
    use                 generic-service
    check_command        check_nrpe!check_disk0
    is_volatile          0
    max_check_attempts  3
    check_interval       1
    retry_interval       1
    check_period         24x7
    notification_interval 5
    notification_period  24x7
    notification_options w,u,c,r
    contact_groups       admins
}
```

注:check_nrpe!check_disk0 这里的 check_disk0 与客户端 nrpe.cfg 配置文件里名称相同。客户端 nrpc.cfg 相应配置文件内容如下:

```
command[check_disk0]=/usr/local/nrpe/libexec/check_disk -w 20 -c 10 -p /dev/hda1
```

注: host_name 主机名, 与 host 文件夹里的主机配置文件里的 host_name 名称相同

service_description	服务描述
use	使用模版
check_command	监控命令
is_volatile	声音警告功能关闭
max_check_attempts	最大检测次数
check_interval	检测时间
retry_interval	重试间隔

check_period 检测时间段
notification_interval 通知时间间隔
notification_period 通知时间段
notification_options 报警通知设置
 w = send notifications on a WARNING state,
 u = send notifications on an UNKNOWN state,
 c = send notifications on a CRITICAL state,
 r = send notifications on recoveries (OK state),
 f = send notifications when the service starts and stops flapping, and
 s = send notifications when scheduled downtime starts and ends
contact_groups 联系组

监控服务-端口监控

在 nagios 的 etc/linux 目录创建 service 文件夹，在 s1.cfg 配置文件中,新增以下内容：

```
define service {
    host_name          192.168.0.126
    service_description tcp_8080
    use                generic-service
    check_command       check_nrpe!check_tcp_8080
    is_volatile         0
    max_check_attempts 3
    check_interval      1
    retry_interval      1
    check_period        4x7
    notification_interval 5
    notification_period 24x7
    notification_options w,u,c,r
    contact_groups      admins
}
```

注:check_nrpe! check_tcp_8080 这里的 check_tcp_8080 与客户端 nrpe.cfg 配置文件里名称相同。客户端 nrpc.cfg 相应配置文件内容如下：

```
command[check_tcp_8080]=/usr/local/nrpe/libexec/check_tcp -p 8080
```

监控服务-http 状态监控

在客户端 **nrpe.cfg** 配置文件里自定义一条命令。

```
command[check_http_status]=/usr/local/nrpe/libexec/check_http -H www.bbs.com -I 192.168.0.126 -u /index.html -t 60
```

-H 表示主机名,也就是域名

-I 表示主机名对应的 IP 地址

-u 表示 uri 地址

-t 表示超时 (单位: 秒)

整条命令意思是,监控 192.168.0.126 服务器上的 www.bbs.com 域名下的 index.html 首页页面。

单独执行

```
# /usr/local/nrpe/libexec/check_http -H www.bbs.com -I 192.168.0.126 -u /index.html -t 60
```

结果如下:

```
HTTP OK HTTP/1.1 200 OK - 236 bytes in 0.001 seconds |time=0.001160s;;;0.000000  
size=236B;;;0
```

成功返回 http 状态 200, 表示 http 状态监控成功。

在服务端添加 http 状态监控服务

在 nagios 的 etc/linux 目录创建 service 文件夹, 在 s1.cfg 配置文件中,新增以下内容:

```
define service {  
    host_name          192.168.0.126  
    service_description http-status  
    use                generic-service  
    check_command       check_nrpe!check_http_status  
    is_volatile         0  
    max_check_attempts  3  
    check_interval      1  
    retry_interval      1  
    check_period        24x7  
    notification_interval 5  
    notification_period 24x7  
    notification_options w,u,r,c  
    contact_groups      admins  
}
```

注:check_nrpe!check_http_status 与客户端的 nrpe.cfg 配置文件定义的名称相同。

添加手机短信报警

通过 linux 下飞信客户端实现手机短信报警。

Linux 飞信客户端下载

<http://www.it-adv.net/fetion/downng/fetion20091117-linux.tar.gz>

linux 飞信安装

```
# tar zxvf fetion20091117-linux.tar.gz
# mv /fetion20091117-linux /usr/local/fetion
# cd /usr/local/fetion
# cp lib* /usr/lib
```

设置权限 chown -R nagios.nagios /usr/local/fetion

飞信短信测试

```
/usr/local/tools/fetion/fetion      --mobile=159*****      --pwd=2233***
--to=158*** --msg-gb="messages"
```

手机能接收到"messages", 说明 fetion 发送短信息成功。

使 linux 下的 fetion 发送信息支持中文件,只需要修改--msg-gb=""为--msg-utf8=""

修改 commands.cfg 配置文件,在末尾添加以内容:

```
define command{
    command_name notify-service-by-sms
    command_line      /usr/local/fetion      --mobile=139369*****
--pwd=fetionpass      --to=$CONTACTPAGER$ --msg-utf8="$HOSTNAME$\n
$SERVICEDESC$ is $SERVICESTATE$\n on $TIME$ result is
$SERVICEOUTPUT$" $CONTACTPAGER$
}
```

修改 contacts.cfg 配置文件，添加飞信好友，报警联系人。内容如下：

```
define contact{
    contact_name          nagiosadmin
    use                   generic-contact
    alias                 Nagios Admin
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,u,r
    service_notification_commands notify-service-by-email,notify-service-by-sms
    host_notification_commands notify-host-by-email
    email                 test@163.com
    pager                 15900000000,13900000000
}
```

注:pager 为添加飞信好友手机号，可添加多个，以(, 分号)隔开。

MySQL 监控

创建 mysql 监控用户

在要监控 mysql 主机上面建立 mysql 用户，用作 mysql 监控服务使用。

```
MySQL>grant all on *.* to nagios_user@'%' identified by 'nagios_user';
MySQL>flush privileges;
```

安装 check_mysql_health 插件

在 nagios 监控服务器上安装 check_mysql_health 插件。

下载地址:http://www.consol.de/fileadmin/opensource/Nagios/check_mysql_health-2.0.3.tar.gz

```
# tar zxvf check_mysql_health-2.0.3.tar.gz
# cd check_mysql_health
# ./configure --prefix=/usr/local/nagios
# make
# make install
# yum install -y perl-DBD-MySQL.i386 perl-DBD-Pg.i386
```

注:check_mysql_health 使用需要这 2 个发安装包

在运行./check_mysql_health 出现以下错误提示,

```
# ./check_mysql_health
```

```
-bash: ./check_mysql_health: yes: bad interpreter: No such file or directory
```

解决方法

因为 check_mysql_health 是用 perl 写的, 将 check_mysql_health 的第一行由原来的#! yes -w 修改为#!/usr/bin/perl, 这样就 OK 了

修改 commands.cfg 配置文件, 在末尾添加以下内容:

```
define command{
    command_name check_mysql_health
    command_line $USER1$/check_mysql_health --hostname $ARG1$ --port
$ARG2$ --username $ARG3$ --password $ARG4$ -w $ARG5: -c $ARG6: --mode $ARG7$
}
```

创建 mysql 监控服务配置文件

Vi /usr/local/nagios/etc/linux/service/check_mysql.cfg, 内容如下:

```
define service {
    host_name                192.168.125
    service_description      check_mysql_slave
    use                      generic-service
    check_command             check_mysql_health!192.168.1.126!3306! \
nagios_user!nagios_pass!150!200!threads-connected
    is_volatile              0
    max_check_attempts       3
    check_interval           1
    retry_interval           1
    check_period             24x7
    notification_interval    5
    notification_period      24x7
    notification_options     w,u,c,r
    contact_groups           admins
    register                 1
}
```

注:

check_mysql_health!192.168.1.126!3306!nagios_user!nagios_pass!150!200!threads-connected
is_volatile

192.168.1.126 为 mysql 数据库 IP 地址

3306 为 mysql 数据库端口

Nagios_user 为 mysql 数据库用户名

Nagios_pass 为 mysql 数据库密码
150 为 mysql 报警连接数
200 为 mysql 报警严重连接数
threads-connected 为 mysql 连接数

Mysql 主从监控

在主 **mysql** 数据库上创建监控账号

```
Mysql>grant Replication client on *.* to 'nagios'@'%' identified by 'nagios';  
Mysql>flush privileges;
```

在从 **mysql** 数据库上创建 **check_mysql_slave** 监控脚本，并存放在 **/usr/local/nrpe/libexec** 目录下。

vi /usr/local/nrpe/libexec/check_mysql_slave 内容如下：

```
#!/bin/sh  
slave_info=$(mysql -unagios -pnagios -e "show slave status\G" | grep  
Running | awk '{print $2}'))  
error=$(mysql -unagios -pnagios -e "show slave status\G" | grep Last))  
  
if [ "${slave_info[0]}" = "Yes" -a "${slave_info[1]}" = "Yes" ];then  
    echo "Mysql_Slave_is_Running...YES"  
    exit 0  
else  
    echo "Mysql_Slave_is_Running...NO ${error[@]}"  
    exit 2
```

使用 **mysql** 命令，**show slave status\G** 查看 **mysql** 主从运行状态，

Slave_IO_Running: Yes

Slave_SQL_Running: Yes

这 2 项是否为 Yes, 如果为 Yes, 表示 **mysql** 主从运行正常，否则将输出错误信息。

注：使用数组将 **mysql** 主从状态记录，如果失败，将会输入 **mysql** 出错误提示。

exit 0 表示成功

exit 2 表示异常，报警

设置执行权限

chmod +x /usr/local/nrpe/libexec/check_mysql_slave

修改 nrpe.cfg 配置文件

vi /usr/local/nrpe/etc/nrpe.cfg, 在末尾新增以下内容:
command[check_mysql_slave]=/usr/local/nrpe/libexec/check_mysql_slave

单独执行/usr/local/nrpe/libexec/check_mysql_slave 如果 mysql 正常将会出现以下成功提示:

```
# /usr/local/nrpe/libexec/check_mysql_slave
Slave_is_Running...YES
```

在 nagios 服务端创建监控 mysql 主从配置服务文件

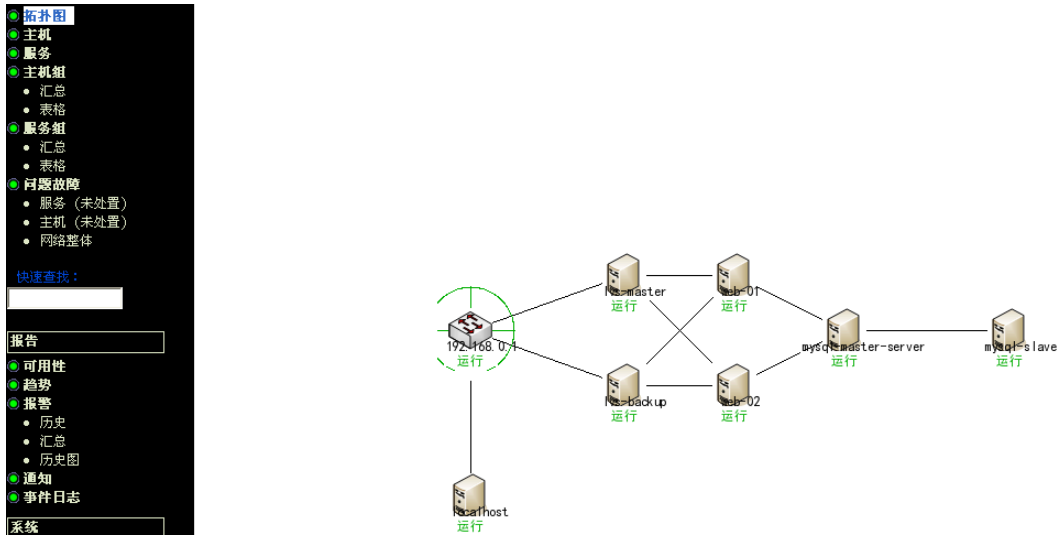
vi /usr/local/nagios/etc/linux/service/check_mysql_slave.cfg
内容如下:

```
define service {
    host_name                192.168.125
    service_description      check_mysql_slave
    use                      generic-service
    check_command             check_nrpe!check_mysql_slave
    is_volatile               0
    max_check_attempts       3
    check_interval            1
    retry_interval            1
    check_period              24x7
    notification_interval    5
    notification_period       24x7
    notification_options      w,u,c,r
    contact_groups            admins
    register                  1
}
```

注: check_command check_nrpe!check_mysql_slave 指定监控 mysql 主从脚本

Nagios 部份功能载图

拓扑图



主机



主机组

当前状态

总览

拓扑图

主机

服务

主机组

- 汇总
- 表格

服务组

- 汇总
- 表格

问题故障

- 服务 (未处置)
- 主机 (未处置)
- 网络整体

快速查找：

报告

可用性

趋势

报警

- 历史
- 汇总
- 历史图

通知

事件日志

[所有主机组的服务详细状态](#)
[所有主机组的主机详细状态](#)
[所有主机组的汇总状态](#)
[所有主机组的服务状态表](#)

所有主机组的服务概要

Linux Servers (linux-servers)				
主机名	别名	状态	服务	动作
192.168.0.1	lan-gateway	运行	无符合条件的服务	
localhost	localhost-server	运行	正常状态: 7 (OK)	
lvs-backup	mysql-slave	运行	无符合条件的服务	
lvs-master	varnish	运行	无符合条件的服务	
mysql-master-server	nginx	运行	正常状态: 7 (OK)	
mysql-slave	oracle-server	运行	无符合条件的服务	
web-01	192.168.0.22	运行	无符合条件的服务	
web-02	varnish	运行	无符合条件的服务	

服务

总览

拓扑图

主机

服务

主机组

- 汇总
- 表格

服务组

- 汇总
- 表格

问题故障

- 服务 (未处置)
- 主机 (未处置)
- 网络整体

快速查找：

报告

可用性

趋势

报警

- 历史
- 汇总
- 历史图

通知

事件日志

[所有主机的详细详细状态](#)

所有主机的正常状态

主机名	别名	服务	状态	最近检查时间	持续时间	尝试次数	状态信息
localhost	localhost-server	/	正常(OK)	2010-09-08 18:02:30	15日 0时 4分 50秒	1/3	DISK OK - free space: /11413 MB (64% inode=96%);
	localhost-server	boot	正常(OK)	2010-09-08 18:02:58	7日 6时 37分 34秒	1/3	DISK OK - free space: /boot 77 MB (82% inode=99%);
	localhost-server	check-swap	正常(OK)	2010-09-08 18:02:32	0日 7时 17分 0秒	1/3	SWAP OK - 100% free (2004 MB out of 2004 MB)
	localhost-server	check-mysql-slave	正常(OK)	2010-09-08 18:02:46	0日 7时 17分 25秒	1/3	Slave_is_Running...YES
	localhost-server	check-total-procs	正常(OK)	2010-09-08 18:02:36	0日 7时 16分 56秒	1/3	PROCS OK: 114 processes
	localhost-server	http-status	正常(OK)	2010-09-08 18:02:09	0日 7时 16分 24秒	1/3	HTTP OK: HTTP/1.1 200 OK - 264 bytes in 0.001 seconds
	localhost-server	load	正常(OK)	2010-09-08 18:02:30	15日 0时 4分 50秒	1/3	OK - load average: 0.22, 0.07, 0.02
mysql-master-server	nginx	boot	正常(OK)	2010-09-08 18:02:13	0日 7时 17分 17秒	1/3	DISK OK - free space: /boot 168 MB (93% inode=99%);
	nginx	load	正常(OK)	2010-09-08 18:02:50	0日 7时 17分 42秒	1/3	DISK OK - free space: /opt 12419 MB (90% inode=99%);
	nginx	load	正常(OK)	2010-09-08 18:02:24	0日 7时 17分 13秒	1/3	OK - load average: 0.00, 0.00, 0.00
	nginx	ping	正常(OK)	2010-09-08 18:02:58	6日 0时 27分 39秒	1/5	PING OK - Packet loss = 0%, RTA = 0.24 ms
	nginx	procs	正常(OK)	2010-09-08 18:02:26	0日 7时 17分 10秒	1/3	PROCS OK: 189 processes
nginx	users	users	正常(OK)	2010-09-08 18:02:48	0日 7时 17分 39秒	1/3	USERS OK - 1 users currently logged in

共13个匹配的服务已全部显示

服务组

当前状态

总览

拓扑图

主机

服务

主机组

- 汇总
- 表格

服务组

- 汇总
- 表格

问题故障

- 服务 (未处置)
- 主机 (未处置)
- 网络整体

快速查找:

报告

可用性

趋势

报警

- 历史
- 报警

所有服务组的服务详细状态

所有服务组的汇总状态

所有服务组的服务状态表

所有服务组的服务概要

负载均衡 (系统负载均衡)

主机名	别名	状态	服务	动作
localhost	localhost-server	运行	正常状态 1 (OK)	
mysql-master-server	nginx	运行	正常状态 1 (OK)	

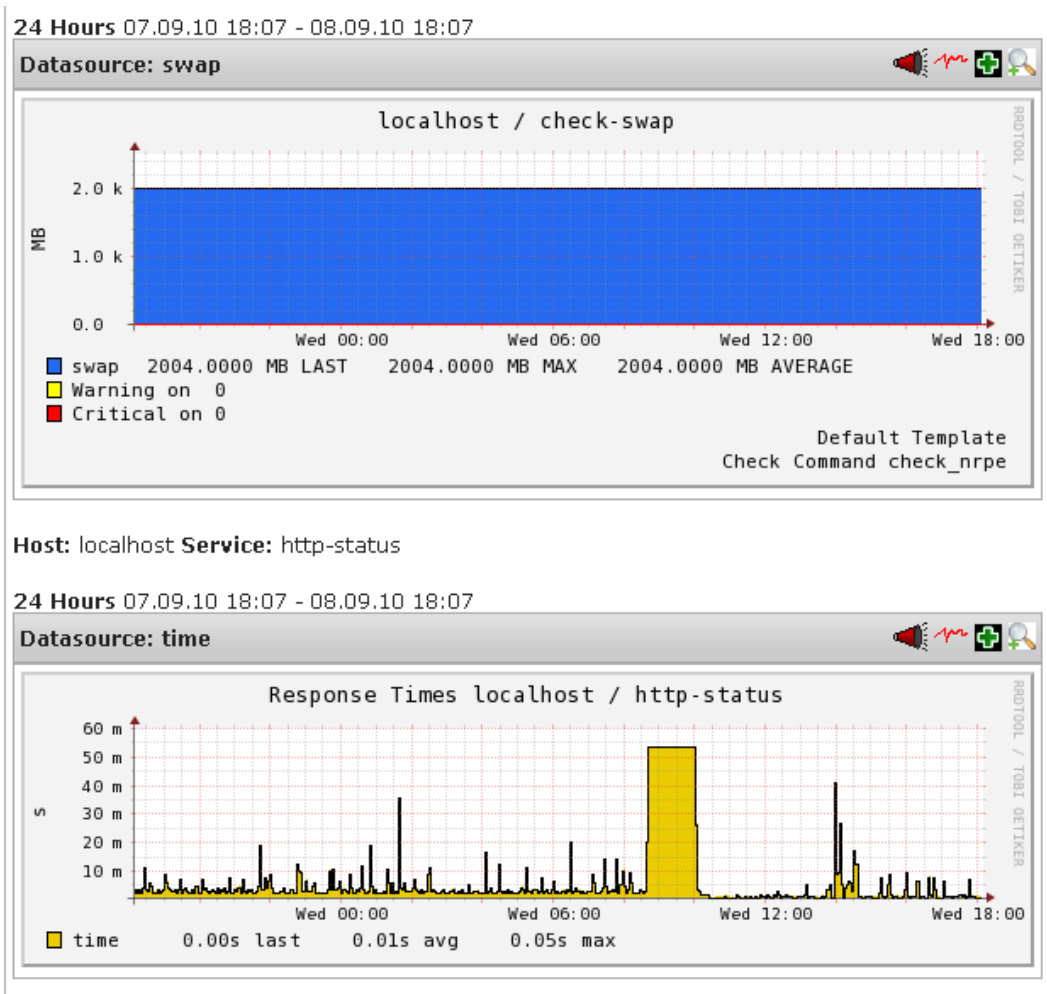
load (localhost)

主机名	别名	状态	服务	动作
mysql-master-server	nginx	运行	正常状态 1 (OK)	

load (mysql-master-server)

主机名	别名	状态	服务	动作
mysql-master-server	nginx	运行	正常状态 1 (OK)	

PNP4Nagios 监控图形

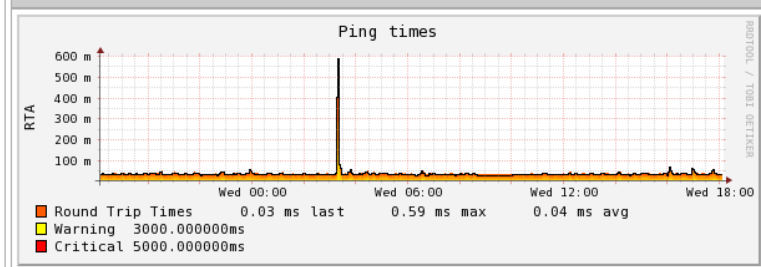


Service overview for "localhost"

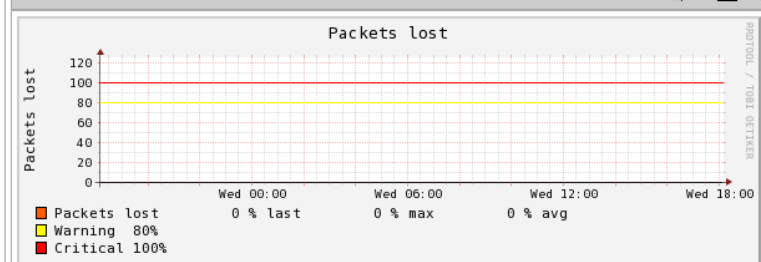
Host: localhost Service: Host Perfddata

24 Hours 07.09.10 18:07 - 08.09.10 18:07

Datasource: Round Trip Times



Datasource: Packets Lost



Search

Actions



My basket

192.168.0.1::_HOST_
[show basket](#)

Status

Host: [localhost](#)
Last Check: 08.09.10 18:07

Time ranges

- [Custom time range](#)
- [4 Hours](#)
- [24 Hours](#)
- [One Week](#)
- [One Month](#)
- [One Year](#)

Services

- [Host Perfddata](#)
- [check-swap](#)
- [http-status](#)
- [load](#)
- [/](#)
- [/boot](#)