

# GNU/Linux-Nagios



## Nagios 配置与应用

# GNU/Linux-Nagios

## 什么是 Nagios

Nagios 是一款开源的免费网络监视工具，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。在系统或服务状态异常时发出邮件或短信报警第一时间通知网站运维人员，在状态恢复后发出正常的邮件或短信通知。

# GNU/Linux-Nagios

## 什么是 Nagios

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，以及日志等等。



# GNU/Linux-Nagios

## Nagios 功能

- 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、简单地插件设计使得用户可以方便地扩展自己服务的检测方法；

# GNU/Linux-Nagios

## Nagios 功能

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间的关系，这种关系可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 EMail、短信、用户定义方式）；

# GNU/Linux-Nagios

## Nagios 功能

- 7、可以定义一些处理程序，使之能够在服务或者主机发生故障时起到预防作用；
- 8、自动的日志滚动功能；
- 9、可以支持并实现对主机的冗余监控；



# GNU/Linux-Nagios

## Nagios 功能

- 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；
- 11、可以通过手机查看系统监控信息；
- 12、可指定自定义的事件处理控制器；



# GNU/Linux-Nagios

## Nagios 建立

### 1. 安装 Apache

```
#yum install httpd -y
```

```
#rm -f /etc/httpd/conf.d/welcome.conf
```





# GNU/Linux-Nagios

## Nagios 建立

### 2. 配置 Apache

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

```
# 定义 86 行
```

```
ServerAdmin root@niliu.edu
```

```
# 定义 95 行
```

```
ServerName nos.niliu.edu:80
```



# GNU/Linux-Nagios

## Nagios 建立

### 2. 配置 Apache

# 定义 151: change  
AllowOverride All

# 定义 164

DirectoryIndex index.html index.cgi index.php

/\* 于文件末尾增加如下内容

ServerTokens Prod

KeepAlive On



# GNU/Linux-Nagios

## Nagios 建立

### 2. 配置 Apache

ServerTokens 指令, 可以隐藏 Apache 的一些敏感信息

ServerTokens Prod

服务器会发送 ( 比如 ) : Server: Apache

ServerTokens Major

服务器会发送 ( 比如 ) : Server: Apache/2



# GNU/Linux-Nagios



## Nagios 建立

### 2. 配置 Apache

#### ServerTokens 指令

#### ServerTokens Minor

服务器会发送 ( 比如 ) : Server: Apache/2.0

#### ServerTokens Min

服务器会发送 ( 比如 ) : Server: Apache/2.0.41

# GNU/Linux-Nagios

## Nagios 建立

### 2. 配置 Apache

#### ServerTokens 指令

#### ServerTokens OS

服务器会发送 ( 比如 ) : Server: Apache/2.0.41  
(Unix)

#### ServerTokens Full ( 或未指定 )

服务器会发送 ( 比如 ) : Server: Apache/2.0.41  
(Unix) PHP/4.2.2 MyMod/1.2



# GNU/Linux-Nagios

## Nagios 建立

### 2. 配置 Apache

#### KeepAlive 指令:

对于 HTTP/1.1 的客户端来说, 将会尽量的保持客户的 HTTP 连接, 通过一个连接传送多份 HTTP 请求响应。这样对于客户端来说, 可以提高 50% 左右的响应时间, 而于服务器端来说则降低了更多个连接的开销。不过这个依赖于客户端是否想保持连接。浏览器默认是保持连接的, 当你打开 100 个图片的网站时, 浏览器有可能只打开 2 个连接, 通过这两个连接传送数据, 而不是开 100 个连接。

# GNU/Linux-Nagios

## Nagios 建立

### 3. 配置 PHP

```
# yum install php php-mbstring php-pear -y
```

```
#vi /etc/php.ini
```

```
/* 定义 878 行, 时间区域
```

```
date.timezone=" Asia/Shanghai"
```



# GNU/Linux-Nagios

## Nagios 建立

### 4. 建立 index.php

```
#vi /var/www/html/index.php  
<?php phpinfo() ?>
```





# GNU/Linux-Nagios

## Nagios 建立

### 5. 启动 Apache

```
#systemctl start httpd
```

```
#systemctl enable httpd
```



# GNU/Linux-Nagios

## Nagios 建立

### 6. 测试 Apache 与 PHP

浏览器 <http://server>

### 7. 删除测试页

```
#rm -rf /var/www/html/index.php
```



# GNU/Linux-Nagios



## Nagios 建立

### 8. 安裝 Nagios

```
#yum --enablerepo=epel install nagios nagios-  
plugins-{ping,disk,users,procs,load,swap,ssh,http}  
-y
```

# GNU/Linux-Nagios

## Nagios 建立

### 9. 配置 Nagios

```
#vi /etc/httpd/conf.d/nagios.conf
```

```
/* 注释 22,23 行
```

```
#Require all granted
```

```
#Require local
```

在下面追加

```
Require ip 127.0.0.1 192.168.0.0/16
```



# GNU/Linux-Nagios

## Nagios 建立

### 9. 配置 Nagios

```
#vi /etc/httpd/conf.d/nagios.conf
```

```
/* 注释 53,54 行
```

```
#Require all granted
```

```
#Require local
```

在下面追加

```
Require ip 127.0.0.1 192.168.0.0/16
```



# GNU/Linux-Nagios

## Nagios 建立

### 9. 配置 Nagios

/\* 生成 Nagios 管理员账号及密码  
( 于 /etc/httpd/conf.d/nagios 中定义的口令文件 )  
( 于 /etc/nagios/ 中定义管理员为 nagiosadmin )

```
#htpasswd /etc/nagios/passwd nagiosadmin
```

# GNU/Linux-Nagios

## Nagios 建立

### 10. 启动 Nagios

```
#systemctl start nagios
```

```
#systemctl enable nagios
```

```
#systemctl restart httpd
```

```
#systemctl status httpd nagios
```



# GNU/Linux-Nagios

## Nagios 建立

### 11. 修正 Nagios-bug

//\* 如出现如下错误

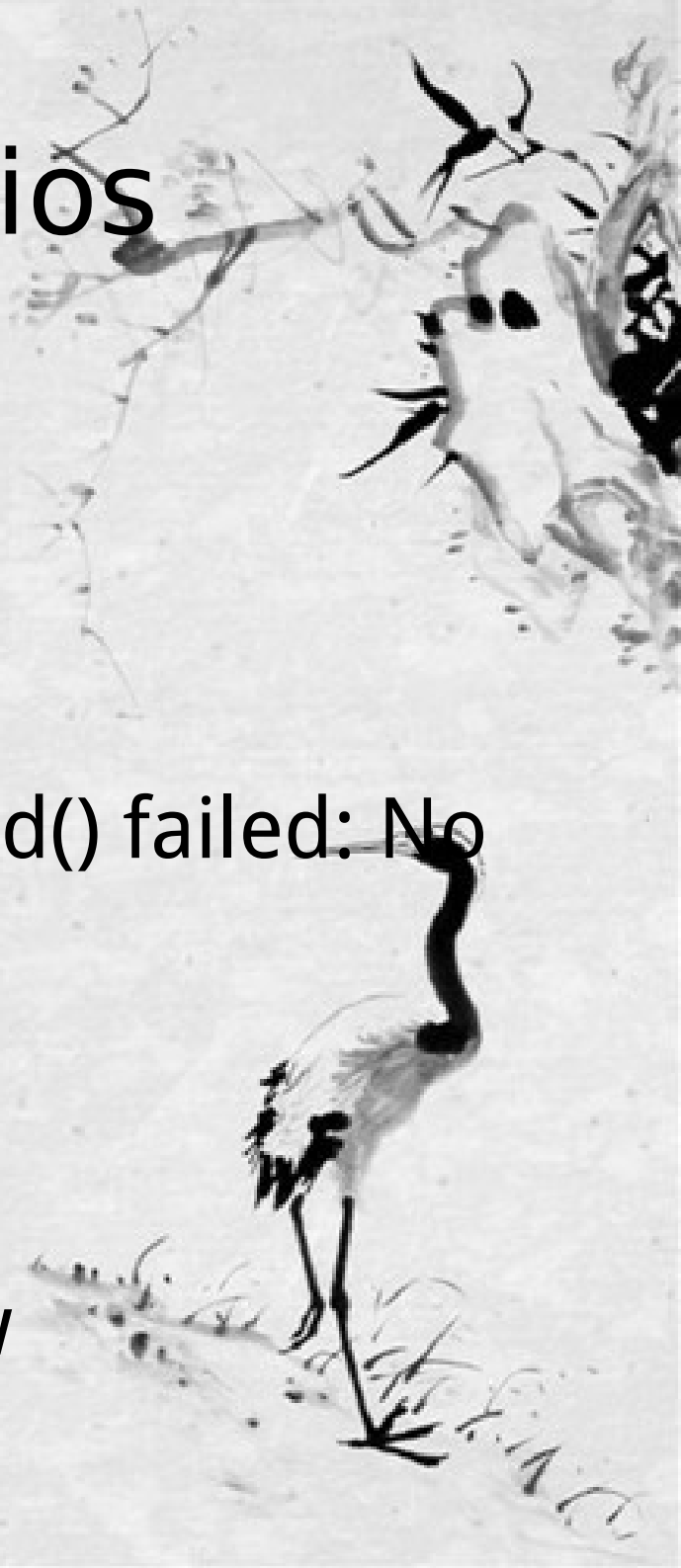
qh: Failed to init socket

'/var/log/nagios/rw/nagios.qh'. bind() failed: No such file or directory

请创建 socket 所需目录

```
#mkdir -v /var/log/nagios/rw
```

```
#chown nagios. /var/log/nagios/rw
```





# GNU/Linux-Nagios

## Nagios 建立

### 11. 访问 Nagios WEB

浏览器 -> [http://\\$srv\\_ip/nagios](http://$srv_ip/nagios)



# GNU/Linux-Nagios

## Nagios 通知设定

### 1. 安装 SMTP

```
#yum install postfix -y
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

```
#vi /etc/postfix/main.cf
```

```
/* 定义 75 行 ,myhostname  
myhostname = mail.niliu.edu
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

```
/* 定义 83 行  
mydomain = niliu.edu
```

```
/* 定义 99 行  
myorigin = $mydomain
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

/\* 定义 116 行

```
inte_interfaces = all
```

/\* 定义 164 行, 追加 \$mydomain

```
mydestination = $myhostname, localhost
```

```
$mydomain, localhost, $mydomain
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

/\* 定义 264, 定义自己的网络段

mynetworks = 168.168.0.0/16, 127.0.0.0/8

/\* 定义 419 行,

home\_mailbox = Maildir/



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

//\*574 行, 追加  
smtpd\_banner = \$myhostname ESMTP

//\* 在文件最底下追加如下内容



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

/\* 设置 MAIL 大小为 10M 一封。  
message\_size\_limit = 10485760

/\* 设置 MailBox 大小为 1G  
mailbox\_size\_limit = 1073741824





# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

/\* 设置 SMTP 认证

```
smtpd_sasl_type = dovecot
```

```
smtpd_sasl_path = private/auth
```

```
smtpd_sasl_auth_enable = yes
```

```
smtpd_sasl_security_options = noanonymous
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 2. 配置 SMTP

/\* 设置 SMTP 认证

```
smtpd_sasl_local_domain = $myhostname
```

```
smtpd_recipient_restrictions =  
permit_mynetworks,permit_auth_destination,per  
mit_sasl_authenticated,reject
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 3. 启动 Postfix

```
#systemctl start postfix  
#systemctl enable postfix
```



# GNU/Linux-Nagios

## Nagios 通知设定

4. 在 DNS 服务器的数据库中设定 MX 记录

```
#vi /var/named/niliu.db
```

```
.....
```

```
@      IN      MX      10      as.niliu.edu.
```

```
.....
```



# GNU/Linux-Nagios

## Nagios 通知设定

### 4. 启动 dns

```
#systemctl restart bind  
#systemctl enable bind
```



# GNU/Linux-Nagios

## Nagios 通知设定

5. 在 Nagios 服务器中设定 Nagios 收件人

```
#yum install mailx -y
```

```
#vi /etc/nagios/objects/contacts.cfg  
/* 设定 34 行 ,nagios 管理员邮件地址  
email root@localhost
```



# GNU/Linux-Nagios

## Nagios 通知设定

5. 在 Nagios 服务器中设定 Nagios 收件人

```
#systemctl restart nagios
```



# GNU/Linux-Nagios



## Nagios 通知设定

### 6. 在 Nagios WEB 中开启通知

Service-> 找到右侧带有红色”\” 标记 -> 点击 -  
> 找到右侧 ->Enable notifications for this service-  
>Commit





# GNU/Linux-Nagios

## Nagios 监控设定

### 1. 监控本地磁盘使用量

```
# vi /etc/nagios/objects/localhost.cfg
```

```
/* 在本地主机上监控磁盘的使用量。如果使用率  
低于 20% 空间将给予警告，如果空间使用率低于  
10% 将提出严重警告通知。可根据基线自己修改
```

# GNU/Linux-Nagios

## Nagios 监控设定

### 1. 监控本地磁盘使用量

```
# vi /etc/nagios/objects/localhost.cfg
```

```
//*76 行
```

```
define service{
```

```
    use
```

```
    host_name
```

```
    service_description
```

```
    check_command
```

```
    20%!10%!/
```

```
}
```

```
local-service
```

```
localhost
```

```
Root Partition
```

```
check_local_disk!
```



# GNU/Linux-Nagios

## Nagios 监控设定

### 2. 使用 nagios 插件检测指定分区

#### 1) 显示插件所能提供的选项

```
#/usr/lib64/nagios/plugins/check_disk -h
```

```
#check_disk -w 80% -c 90% -p /dev/sda1
```



# GNU/Linux-Nagios

## Nagios 监控设定

### 2. 使用 nagios 插件检测指定分区

2) 指定插件的命令配置文件定义插件的调用。

```
# vi /etc/nagios/objects/commands.cfg
define command{
    command_name    check_sda1
    command_line    $USER1$/check_disk -w
$ARG1$ -c $ARG2$ -p $ARG3$
}
```

# GNU/Linux-Nagios

## Nagios 监控设定

2. 使用 nagios 插件检测指定分区

3) 调用插件编辑检测内容

```
#vi /etc/nagios/objects/localhost.cfg
```



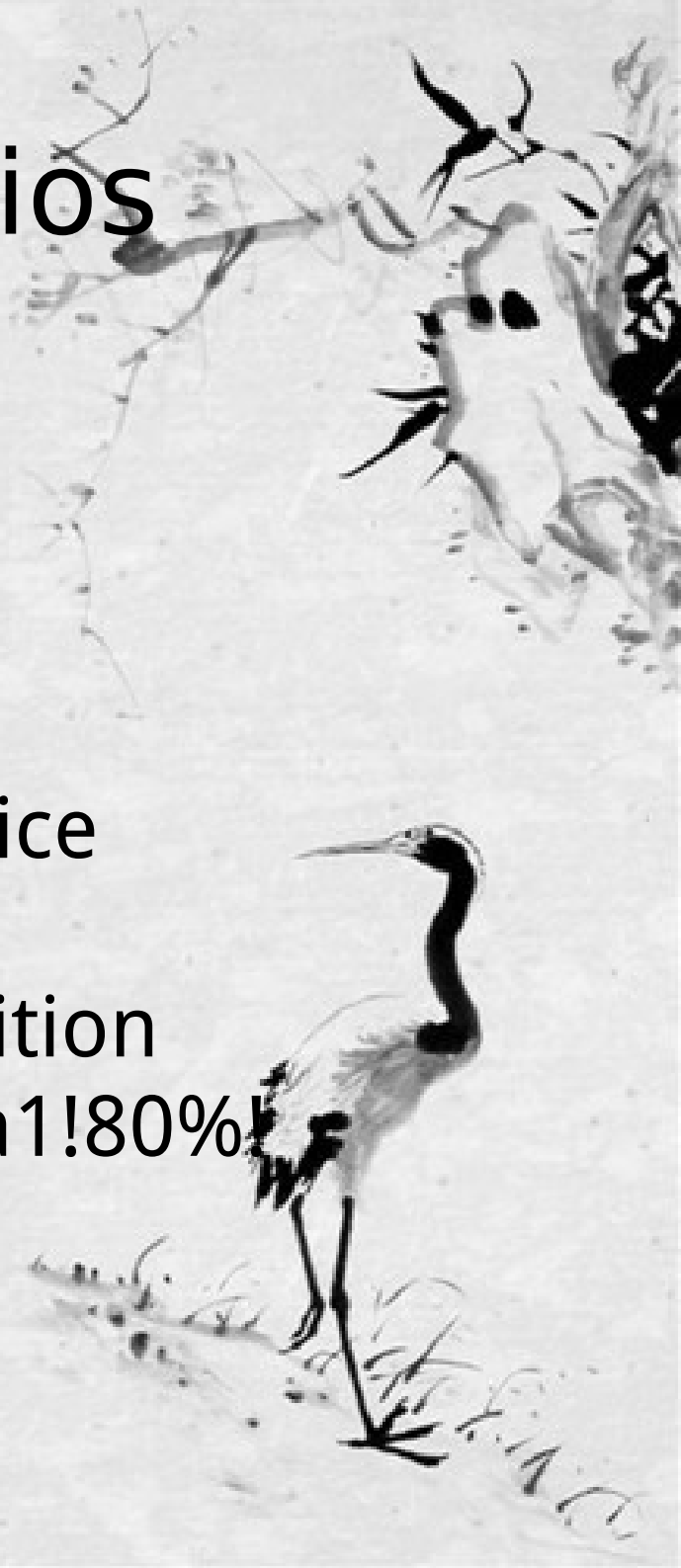
# GNU/Linux-Nagios

## Nagios 监控设定

2. 使用 nagios 插件检测指定分区

3) 调用插件编辑检测内容

```
define service{  
    use                local-service  
    host_name          localhost  
    service_description Boot Partition  
    check_command       check_sda1!80%!  
90%!/dev/sda1  
}
```



# GNU/Linux-Nagios

## Nagios 监控设定

### 2. 使用 nagios 插件检测指定分区

```
#systemctl restart nagios
```

刷新 Nagios WEB 页面



# GNU/Linux-Nagios



## 为 Nagios 添加一个监控项目

### 1. 查询 Nagios 监控插件

```
# yum --enablerepo=epel search nagios-plugins-
```



# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 2. 添加 Nagios 监控插件

```
# yum --enablerepo=epel install nagios-plugins-ntp -y
```

# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 3. 查看 Nagios 监控插件使用方法

```
#/usr/lib64/nagios/plugins/check_ntp_time -h
```



# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 4. 配置 commands.cfg 定义插件调用方法

```
#vi /etc/nagios/objects/commands.cfg
```

```
define command{  
    command_name check_ntp_srv  
    command_line $USER1$/check_ntp_time -H  
$ARG1$ -w $ARG2$ -c $ARG3$  
}
```

# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 5. 配置 localhost.cfg 调用插件检测

```
#vi /etc/nagios/objects/localhost.cfg
define service{
    use                local-service
    host_name          localhost
    service_description NTP_TIME
```



# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 5. 配置 localhost.cfg 调用插件检测

/\* 检测 192.168.188.49, 如果延迟 1 秒提出警告, 延迟 2 秒提出严重警告

```
check_command      check_ntp_srv!  
192.168.188.49!1!2  
/* 开启通知  
notifications_enabled 1  
}
```



# GNU/Linux-Nagios

## 为 Nagios 添加一个监控项目

### 6. 启动 Nagios

```
#systemctl restart nagios
```

### 7. Nagios WEB 页检测



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 1. 设定 Nagios 服务器

#### 1) 设定服务器监控配置文件所在目录

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

```
/* 取消 52 行注释
```

```
cfg_dir=/etc/nagios/servers
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

1. 设定 Nagios 服务器

2) 创建目录

```
#mkdir -v /etc/nagios/servers
```

3) 修改相关权限

```
#chgrp nagios /etc/nagios/servers
```

```
#chmod 750 /etc/nagios/servers
```





# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 1. 设定 Nagios 服务器

### 4) 配置目标监控主机 ( 所要监控的服务器 )

```
#vi /etc/nagios/servers/noc.cfg
```

```
define host{  
    use                linux-server  
    host_name          noc  
    alias              noc  
    address            192.168.188.112  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 1. 设定 Nagios 服务器

### 4) 配置目标监控主机 ( 所要监控的服务器 )

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description PING  
    check_command       check_ping!100.0,10%!  
300.0,40%  
}
```

# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 1. 设定 Nagios 服务器

#### 4) 配置目标监控主机 (所要监控的服务器)

/\* 检测 ping, 如果 ping 包平均往返超过 100ms 且丢包率超过 10% 给予警告。如果 ping 包平均往返超过 300ms 且丢包率超过 40% 给予严重警告。

```
    check_command    check_ping!100.0,10%!
300.0,40%
}
```

# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

1. 设定 Nagios 服务器

5) 启动 Nagios

```
#systemctl restart nagios
```



# GNU/Linux-Nagios



## 增加一个监控 Linux 主机

### 2. 配置目标服务器

#### 1) 为目标服务器添加 Nagios 插件

```
#yum --enablerepo=epel install nrpe nagios-  
plugins-{ping,disk,users,procs,load,swap,ssh} -y
```

# GNU/Linux-Nagios



## 增加一个监控 Linux 主机

### 2. 配置目标服务器

//\*NRPE(Nagios Remote Plugin Execution) 插件一起安装。 NRPE 插件可以使用类似代理的方式在远程系统上做检查。

# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 2. 配置目标服务器

#### 2) 配置 nrpe.cfg

```
#vi /etc/nagios/nrpe.cfg
```

```
/* 定义 81 行, 允许 Nagios 服务器访问
```

```
allowed_hosts=127.0.0.1,192.168.188.211
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 2. 配置目标服务器

```
/* 定义 97 行, 允许由 Nagios 服务传递监控参数  
dont_blame_nrpe=1
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 2. 配置目标服务器

/\* 注释 210-214 行本地检测

```
#command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5 -c 10  
#command[check_load]=/usr/lib64/nagios/plugins/check_load -w 15,10,5 -c 30,25,20  
#command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1  
#command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10 -sZ  
#command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 150 -c 200
```

# GNU/Linux-Nagios



## 增加一个监控 Linux 主机

### 2. 配置目标服务器

//\* 开启 223-226 行, 由服务器传递检测参数

```
command[check_users]=/usr/lib64/nagios/plugins/check_users -w $ARG1$ -c $ARG2$  
command[check_load]=/usr/lib64/nagios/plugins/check_load -w $ARG1$ -c $ARG2$  
command[check_disk]=/usr/lib64/nagios/plugins/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$  
command[check_procs]=/usr/lib64/nagios/plugins/check_procs -w $ARG1$ -c $ARG2$ -s $ARG3$
```

# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 2. 配置目标服务器

```
//* 开启 nrpe
```

```
#systemctl start nrpe
```

```
#systemctl enable nrpe
```



# GNU/Linux-Nagios



## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

#### 1) 安装 nrpe 插件

```
#yum --enablerepo=epel install nagios-plugins-nrpe -y
```

# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

#### 2) 定义监控主机配置文件及目录

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

```
/* 取消 52 行注释
```

```
cfg_dir=/etc/nagios/servers
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

#### 3) 对目录进行权限操作

```
#mkdir -v /etc/nagios/servers
```

```
#chgrp nagios /etc/nagios/servers
```

```
#chmod 750 /etc/nagios/servers
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

#### 4) 配置所需监控信息

```
#vi /etc/nagios/objects/commands.cfg
```

```
/* 于文件最后追加
```

```
define command{
```

```
    command_name      check_nrpe
```

```
    command_line      $USER1$/check_nrpe -H
```

```
$HOSTADDRESS$ -c $ARG1$
```

```
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

### 4) 配置所需监控信息

```
#vi /etc/nagios/servers/noc.cfg
```





# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

#### 4) 配置所需监控信息

```
define host{  
    use                LinuxSrv  
    host_name          noc  
    alias              noc  
    address            192.168.188.212  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description PING  
    check_command       check_ping!100.0,10%!  
300.0,40%  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

/\* 监控远程磁盘使用量

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description Root Partition  
    check_command       check_nrpe!check_disk\  
20%\!10%\!  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

/\* 检测远程当前登陆账户数量

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description Current Users  
    check_command      check_nrpe!  
    check_users\!20\!50  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

/\* 监控远程服务器进程及类型

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description Total Processes  
    check_command      check_nrpe!  
    check_procs\!250\!400\!RSZDT  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 3. 配置 Nagios 服务器

/\* 监控远程服务器当前负载

```
define service{  
    use                generic-service  
    host_name          noc  
    service_description Current Load  
    check_command       check_nrpe!check_load\  
5.0,4.0,3.0\!10.0,6.0,4.0  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Linux 主机

### 4. 启动 Nagios 服务

```
#systemctl start nagios
```

```
#systemctl enable nagios
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

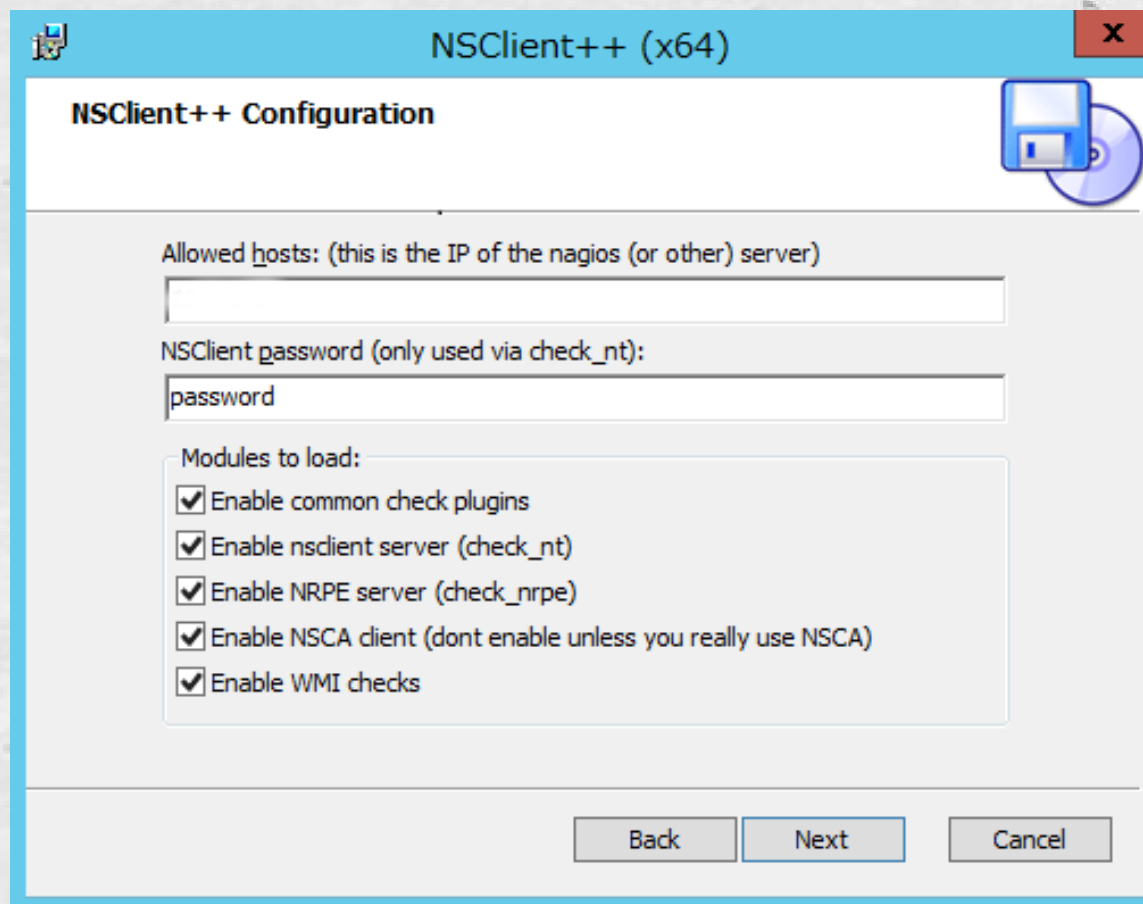
1. 准备一台 Windows2012/2000 R2 Server
2. 到 <http://sourceforge.net/projects/nscplus/> 下载 NSClnet++ 软件并安装到 Win Srv 上



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

3. 填入 Nagios 服务区 IP 及 Nagios 链接所需的密码等

The image shows a screenshot of the NSClient++ Configuration window. The window has a blue title bar with the text "NSClient++ (x64)" and a red close button. Below the title bar is a white header area with the text "NSClient++ Configuration" and a blue floppy disk icon. The main content area is light gray and contains three sections: "Allowed hosts: (this is the IP of the nagios (or other) server)" with an empty text box, "NSClient password (only used via check\_nt):" with a text box containing the word "password", and "Modules to load:" with a list of five checked items: "Enable common check plugins", "Enable nsclient server (check\_nt)", "Enable NRPE server (check\_nrpe)", "Enable NSCA client (dont enable unless you really use NSCA)", and "Enable WMI checks". At the bottom of the window are three buttons: "Back", "Next", and "Cancel".

NSClient++ (x64)

NSClient++ Configuration

Allowed hosts: (this is the IP of the nagios (or other) server)

NSClient password (only used via check\_nt):

password

Modules to load:

- ☒ Enable common check plugins
- ☒ Enable nsclient server (check\_nt)
- ☒ Enable NRPE server (check\_nrpe)
- ☒ Enable NSCA client (dont enable unless you really use NSCA)
- ☒ Enable WMI checks

Back Next Cancel

# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

4. 开启 -> 服务 -> 设置 NSClient++(x64) 启动



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

#### 1) 配置 commands.cfg

# 修改 210 行：添加 Windows 所设定的密码，链接端口为 12489

```
command_line $USER1$/check_nt -H  
$HOSTADDRESS$ -p 12489 -v $ARG1$ $ARG2$ -s  
password
```

# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

#### 2) 配置 winsrv.cfg

```
#cd /etc/nagios/server/winsrv.cfg
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
define host{  
    use                windows-server  
    host_name          winsrv  
    alias              winsrv  
    address            192.168.188.213  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
define hostgroup{  
    hostgroup_name      5iblueservers  
    alias                5iblueservers  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
/* 检测 ping
define service{
    use                generic-service
    host_name           winsrv
    service_description PING
    check_command        check_ping!
    100.0,10%!300.0,40%
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
/* 检测 NSClient++ 版本
define service{
    use                generic-service
    host_name           winsrv
    service_description NSClient++ Version
    check_command        check_nt!
                        CLIENTVERSION
}
```





# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
/* 检测 uptime
define service{
    use                generic-service
    host_name           winsrv
    service_description Uptime
    check_command        check_nt!UPTIME
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

/\* 检测 cpu 负载

```
define service{
```

```
    use                generic-service
```

```
    host_name          winsrv
```

```
    service_description CPU Load
```

```
    check_command       check_nt!CPULOAD!-l
```

```
    5,80,90
```

```
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

/\* 检测内存使用量

```
define service{
```

```
    use                generic-service
```

```
    host_name          winsrv
```

```
    service_description Memory Usage
```

```
    check_command       check_nt!MEMUSE!-w
```

```
    80 -c 90
```

```
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

/\* 检测磁盘使用量

```
define service{  
    use                generic-service  
    host_name           winsrv  
    service_description C:\ Drive Space  
    check_command        check_nt!  
                        USEDDISKSPACE!-l c -w 80 -c 90  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

```
/* 检测 Explorer
define service{
    use                generic-service
    host_name           winsrv
    service_description Explorer
    check_command        check_nt!
PROCSTATE!-d SHOWALL -l Explorer.exe
}
```



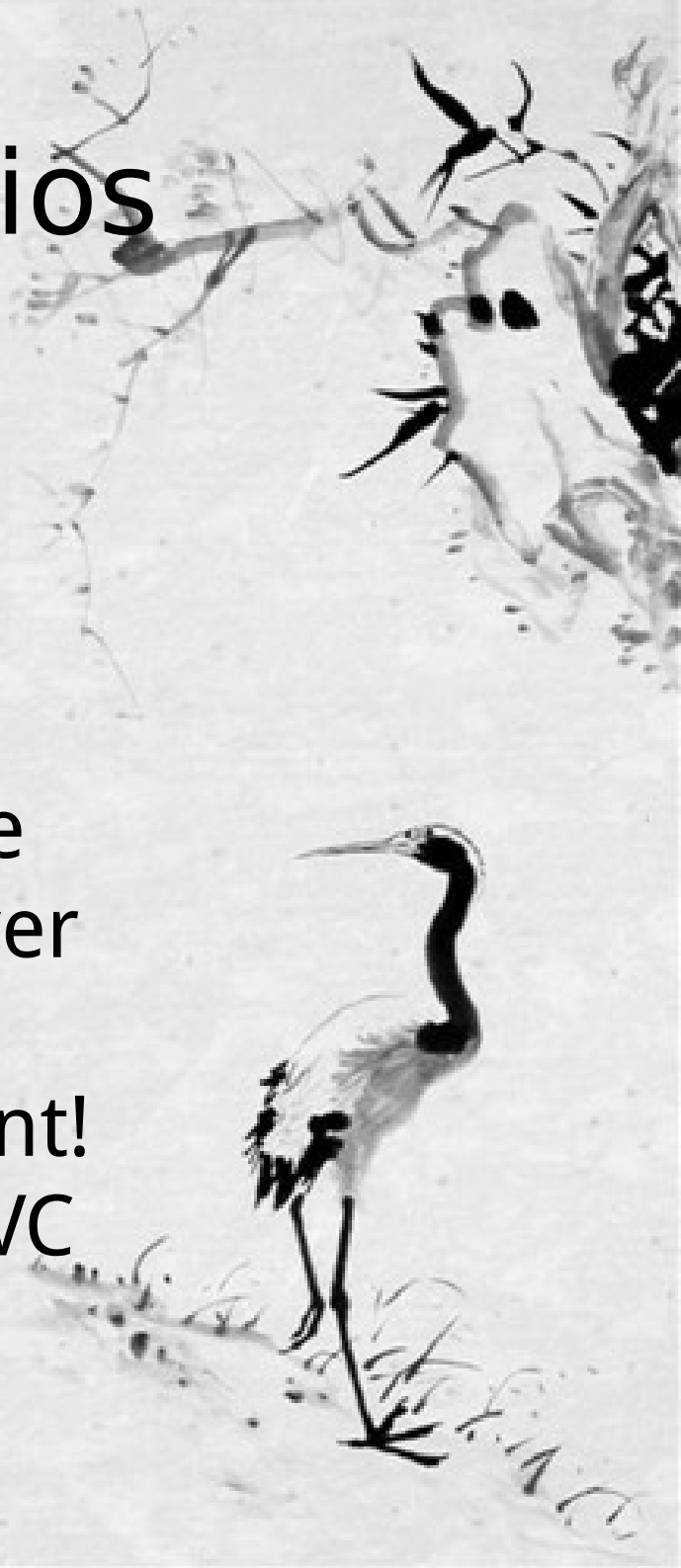
# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 5. 于 Nagios 服务器配置

/\* 检测 IIS 服务

```
define service{  
    use                generic-service  
    host_name          5iblueserver  
    service_description W3SVC  
    check_command       check_nt!  
SERVICESTATE!-d SHOWALL -I W3SVC  
}
```



# GNU/Linux-Nagios

## 增加一个监控 Windows 主机

### 6. 重启 Nagios 服务

```
#systemctl restart nagios
```

### 7. Nagios WEB 页监控

