# Nagios详细安装配置

## 一,安装:

### 1.相关环境

| Host Name | IP | OS | Arch |
|---|---|---|---|
| duangr-1 | 192.168.56.10 | CentOS 6.4 | x86_64 |
| duangr-2 | 192.168.56.11 | CentOS 6.4 | x86_64 |
| duangr-3 | 192.168.56.12 | CentOS 6.4 | x86_64 |

### 2.Nagios主节点192.168.56.10需要安装:

- nagios
- nagios-plugin
- nrpe
- apache

**Nagios从节点192.168.56.11/12需要安装:**

- nagios-plugin
- nrpe

### 3.主机环境检查(全部节点)

yum install -y wget httpd php gcc glibc glibc-common gd gd-devel make net-snmp openssl openssl-devel perl

安装后检查

rpm –q wget httpd  php gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel openssl perl

若有缺失,请先安装. 可通过如下几个镜像网站下载相关安装包:

- http://rpm.pbone.net/
- http://mirrors.163.com/centos/6.4/os/x86_64/Packages/
- http://mirrors.sohu.com/centos/6.4/os/x86_64/Packages/

### 4.创建用户nagios(全部节点)

useradd nagios

groupadd nagcmd

usermod -a -G nagcmd nagios

### 5.安装nagios主程序(主节点安装)

tar zxvf nagios-4.0.4.tar.gz

cd nagios-4.0.4

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

make all

make install

make install-init

make install-config

make install-commandmode

make install-webconf

cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/

chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (自检)

/etc/init.d/nagios start

/etc/init.d/httpd start

/usr/local/apache2/bin/apachetl -t   检测apache配置文件

创建一个默认的WEB登录账户

```
htpasswd –c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

将nagios添加为服务

```
chkconfig --add nagios
chkconfig --level 35 nagios on
chkconfig --add httpd
chkconfig --level 35 httpd on
```

### Nagios Plugin 安装(全部节点安装)

先检查xinetd服务是否安装　 rpm -qa | grep xinetd

没有则安装xinetd 包　 yum -y install xinetd

```
tar zxvf nagios-plugins-2.0.tar.gz
cd nagios-plugins-2.0
./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
make install
```

如果出现mysql相关的编译错误,是mysql的默认安装路径被修改导致的,调整with-mysql后重新make

```
1. ./configure --prefix=/usr/local/nagios  --with-mysql=/usr/local/mysql
2. make && make install
```

### 6.安装NRPE(全部节点安装)

```
1. tar -zxf nrpe-2.15.tar.gz
2. cd nrpe-2.15
3. ./configure --enable-command-args
4. make all
5. make install-plugin
```

下面步骤只需要在被监控节点执行

```
1. make install-daemon && make install-daemon-config && make install-xinetd
```

被监控节点配置

如果是被监控节点,需要配置NRPE已守护进程运行(通过xinetd来运行)

1、更改/etc/xinetd.d/nrpe文件，设置允许nagios主节点服务器连接

```
 vi /etc/xinetd.d/nrpe
 only_from       = 127.0.0.1 192.168.56.10
```

2、在/etc/services结尾增加：

```
 nrpe      5666/tcp       # NRPE
```

3、增加对参数的支持

```
 vi /usr/local/nagios/etc/nrpe.cfg
  dont_blame_nrpe=1
  allowed_hosts=127.0.0.1,192.168.56.10
```

4.更改目录权限（必须，不然nrpe服务起不来）

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

5、启动nrpe

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

或者重启service xinetd restart

如报错： xinetd: unrecognized service

则安装xinetd 包

```
rpm -qa | grep xinetd
yum -y install xinetd
```

注：有时候在改完nrpe.cfg后执行以上重启步骤不行，则需要先杀死进程，在启动nrpe服务

```
pkill nrpe;/usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg  -d
```

6、验证nrpe是否监听

```
netstat -anpt | grep nrpe
```

7、测试nrpe是否正常运行

```
/usr/local/nagios/libexec/check_nrpe -H 127.0.0.1

NRPE v2.15
```

**开启nagios并访问**

/etc/init.d/nagios start 或 service nagios restart

http://192.168.56.10/nagios/

如果打开网页nagios状态为disable，因查看SElinux状态（需为关闭状态）。

1、/usr/sbin/sestatus -v      ##如果SELinux status参数为enabled即为开启状态

```
SELinux status:                 enabled
```

2、getenforce                 ##也可以用这个命令检查

关闭SELinux：因使用1+2结合的方式关闭SElinux

1、临时关闭（不用重启机器）：

```
setenforce 0                  ##设置SELinux 成为permissive模式
                                 ##setenforce 1 设置SELinux 成为enforcing模式
```

2、修改配置文件需要重启机器（永久生效）：

修改/etc/selinux/config 文件

将SELINUX=enforcing改为SELINUX=disabled（关闭） permissive（记录警告而不是禁止未经批准的操作）

## 二:配置:

### 7.配置远程被监控节点

### 7.1.1 修改配置文件

1. # su - nagios

2. $ vi /usr/local/nagios/etc/nrpe.cfg

修改为如下配置内容：

1. command[check_users]=/usr/local/nagios/libexec/check_users -w $ARG1$ -c $ARG2$

2. command[check_load]=/usr/local/nagios/libexec/check_load -w $ARG1$ -c $ARG2$

3. command[check_disk]=/usr/local/nagios/libexec/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$

4. command[check_procs]=/usr/local/nagios/libexec/check_procs -w $ARG1$ -c $ARG2$ -s $ARG3$

5. command[check_procs_args]=/usr/local/nagios/libexec/check_procs  $ARG1$

6. command[check_swap]=/usr/local/nagios/libexec/check_swap -w $ARG1$ -c $ARG2$

以上监控命令功能:

| | |
|---|---|
| check_users | 监控登陆用户数 |
| check_load | 监控CPU负载 |
| check_disk | 监控磁盘的使用 |
| check_procs | 监控进程数量,状态包括 RSZDT |
| check_swap | 监控SWAP分区使用 |

### 7.1.2 重启 xinetd服务

配置完上述命令后,重启 xinetd服务

1. service xinetd restart

### 7.1.3 校验配置

检查监控命令配置是否ok

```
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_users  -a 5 10
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_load   -a 15,10,5 30,25,20
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_disk    -a 20% 10% /
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_procs -a 200 400 RSZDT
/usr/local/nagios/libexec/check_nrpe -H localhost -c check_swap  -a 20% 10%
```

### 7.1.4 防火墙设置（开启5666端口）

vi /etc/sysconfig/iptables

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 5666 -j ACCEPT
```

service iptables restart

## 7.2 配置监控服务主节点

### 7.2.1 主节点配置测试

```
/usr/local/nagios/libexec/check_nrpe -H 192.168.56.11
  NRPE v2.15
/usr/local/nagios/libexec/check_nrpe -H 192.168.56.12
  NRPE v2.15
/usr/local/nagios/libexec/check_nrpe -H 192.168.56.12  -c check_users
  USERS OK - 0 users currently logged in |users=0;5;10;0
```

### 7.2.2 nagios.cfg(nagios主配置文件)

(使用 nagios 用户)

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

```
#cfg_file=/export/home/nagios/etc/objects/localhost.cfg      (注释掉)
cfg_dir=/export/home/nagios/etc/servers
```

主配置文件声明了监控脚本的存储路径为 ./servers，默认没有此目录，需要手工创建

nagios 会读取 servers 目录下面后缀为.cfg的全部文件作为配置文件

```
cd /usr/local/nagios/etc
mkdir servers
cd servers
```

### 7.2.3 定义监控的主机组

声明一个监控的主机组,将主机环境中提到的三台主机全部加入监控

vi /export/home/nagios/etc/servers/group.cfg

新文件，内容如下：

```
define hostgroup{
    hostgroup_name      duangr-server
    alias               duangr Server
    members             duangr-1,duangr-2,duangr-3
}
```

解释下上面的配置：

- hostgroup_name:   主机组的名称，可随意指定
- alias:                  主机组别名，可随意指定
- members:              主机组成员，多个主机名称之前使用逗号分隔。另外主机名称必须与 define host 中host_name 一致。

主机的定义，后面会说到。

### 7.2.4 定义监控的主机

下面开始定义具体的主机

### 7.2.4.1 本地主机监控配置

先定义本地主机 duangr-1

vi /export/home/nagios/etc/servers/duangr-1.cfg

新文件，内容如下：

```
define host{
      use              linux-server
      host_name        duangr-1
      alias            duangr-1
      address          192.168.56.10
}


define service{
      use                  local-service
      host_name            duangr-1
      service_description       Host Alive
      check_command          check-host-alive
      }
define service{
      use            local-service
      host_name           duangr-1
      service_description     Users
      check_command        check_local_users!20!50
      }
define service{
      use            local-service
      host_name           duangr-1
      service_description    CPU
      check_command  check_local_load!5.0,4.0,3.0!10.0,6.0,4.0              }
define service{
      use           local-service
      host_name          duangr-1
      service_description    Disk Root
      check_command      check_local_disk!20%!10%!/
      }
define service{
      use           local-service
      host_name          duangr-1
      service_description   Disk Home
      check_command    check_local_disk!20%!10%!/export/home              }
define service{
      use          local-service
      host_name         duangr-1
```

```
    service_description   Zombie Procs
    check_command      check_local_procs!5!10!Z
    }
define service{
    use          local-service
    host_name       duangr-1
    service_description     Total Procs
    check_command    check_local_procs!250!400!RSZDT
    }
define service{
    use          local-service
    host_name      duangr-1
    service_description    Swap Usage
    check_command       check_local_swap!20!10
    }
```

说明下，由于是此主机也是监控服务主节点所在主机，因此可以使用check_local_* 的相关命令来进行监控。

这个文件中已经将常用的监控项配置进去。

**7.2.4.2远程主机监控配置**

再定义远程主机duangr-2和duangr-3

定义远程主机的监控之前，需要先定义check_nrpe命令

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

在文件的最后面添加如下内容：

```
# 'check_nrpe' command definition
define command{
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$
    }
define command{
    command_name    check_nrpe_args
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$ -a $ARG2$
    }
```

定义duangr-2主机的监控配置

$ vi /usr/local/nagios/etc/servers/duangr-2.cfg

新文件，内容如下：

```
define host{
    use            linux-server
    host_name              duangr-2
    alias                 duangr-2
    address               192.168.56.11
    }
define service{
    use                 local-service
    host_name                    duangr-2
    service_description          Host Alive
    check_command              check-host-alive
    }
define service{
    use                    local-service
    host_name                  duangr-2
    service_description           Users
    check_command      check_nrpe_args!check_users!5 10
    }
define service{
   use                local-service
   host_name              duangr-2
   service_description         CPU
  check_command   check_nrpe_args!check_load!15,10,5 30,25,20
    }
define service{
    use            local-service
    host_name      duangr-2
    service_description  Disk Root
    check_command   check_nrpe_args!check_disk!20% 10% /
    }
define service{
 use               local-service
 host_name              duangr-2
  service_description      Disk /export/home
check_command  check_nrpe_args!check_disk!20% 10% /export/home
    }
define service{
   use           local-service
   host_name        duangr-2
   service_description      Procs Zombie
   check_command    check_nrpe_args!check_procs!5 10 Z
   }

define service{
   use                   local-service
   host_name                  duangr-2
   service_description          Procs Total
  check_command check_nrpe_args!check_procs_args!"-w400 -c600"             }
define service{
    use               local-service
    host_name            duangr-2
    service_description      Swap Usage
    check_command     check_nrpe_args!check_swap!20% 10%
    }
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; 下面是一些常用进程的监控,主要是云平台相关进程
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; 监控crond进程
define service{
        use                local-service
        host_name               duangr-2
        service_description     PS: crond
 check_command    check_nrpe_args!check_procs_args!"-c1:1 -Ccrond"
        }
;; 监控zookeeper进程
define service{
        use            local-service
        host_name           duangr-2
        service_description    PS: QuorumPeerMain
check_command       check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.quorum.QuorumPeerMain"
        }
;;监控storm的从节点进程
define service{
        use           local-service
        host_name       duangr-2
        service_description    PS: supervisor
        check_command check_nrpe_args!check_procs_args!"-c1:1 -Cjava -adaemon.supervisor"
        }
;; 监控storm的主节点进程
define service{
    use                local-service
    host_name          duangr-2
    service_description       PS: nimbus
    check_command        check_nrpe_args!check_procs_args!"-c1:1 -Cjava -adaemon.nimbus"
        }
;; 监控MetaQ进程
define service{
        use             local-service
        host_name           duangr-2
        service_description     PS: MetaQ
        check_command        check_nrpe_args!check_procs_args!"-c1:1 -Cjava -ametamorphosis-server-w"
        }
;; 监控Redis进程
define service{
        use             local-service
        host_name       duangr-2
        service_description       PS: redis-server
        check_command         check_nrpe_args!check_procs_args!"-c1:1 -Credis-server"
    }
;; 监控hadoop主节点NameNode进程
define service{
    use           local-service
    host_name       duangr-2
    service_description    PS: NameNode
    check_command      check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.namenode.NameNode"
    }
监控hadoop主节点SecondaryNameNode进程
define service{
    use          local-service
    host_name      duangr-2
    service_description   PS: SecondaryNameNode
    check_command    check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.namenode.SecondaryNameNode"
    }
;; 监控hadoop主节点ResourceManager进程
define service{
    use               local-service
    host_name       duangr-2
    service_description  PS: ResourceManager
    check_command                  check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.resourcemanager.ResourceManager"
    }
;; 监控hadoop从节点DataNode进程
define service{
    use                        local-service
    host_name          duangr-2
    service_description    PS: DataNode
    check_command       check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.datanode.DataNode"
    }
;;监控hadoop从节点NodeManager进程
define service{
    use            local-service
    host_name       duangr-2
    service_description    PS: NodeManager
    check_command     check_nrpe_args!check_procs_args!"-c1:1 -Cjava -aserver.nodemanager.NodeManager"
    }
```

说明下,由于duangr-2是远程主机,因此使用check_nrpe_args命令来监控.

这个文件中已经将常用的监控项配置进去, 同时还包含了hadoop、storm、zookeeper、metaq、redis的相关进程监控，主要的监控思路是判断进程是否存在。

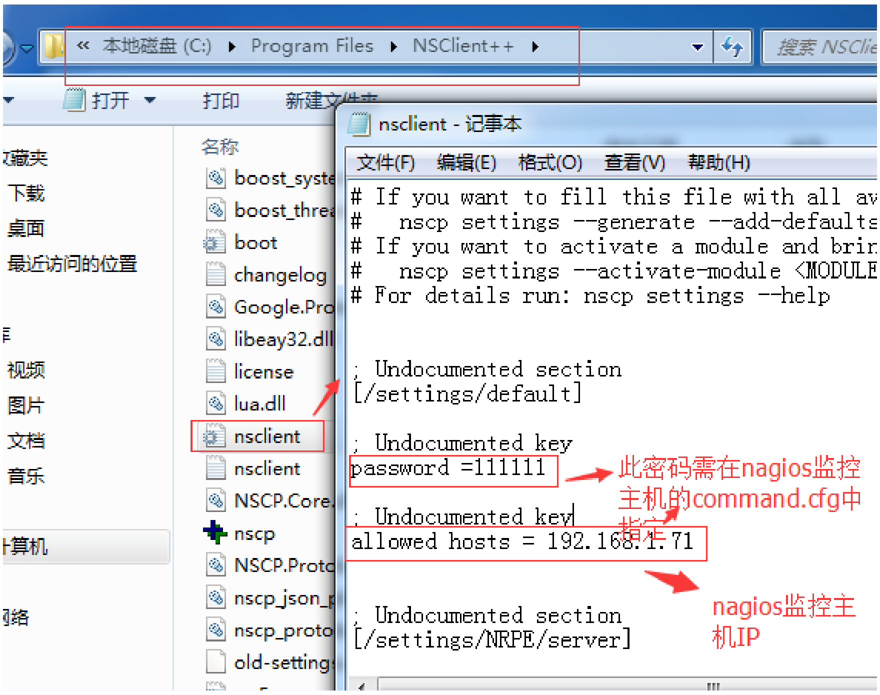
定义duangr-3主机的监控配置

vi duangr-3.cfg

内容与duangr-2.cfg类似,只需要修改 host_name 、alias、 address即可.

#### 7.2.4.3 nagios监控Windows主机（需安装nsclient）

客户端配置：<br>
安装Nsclient略,安装后配置：

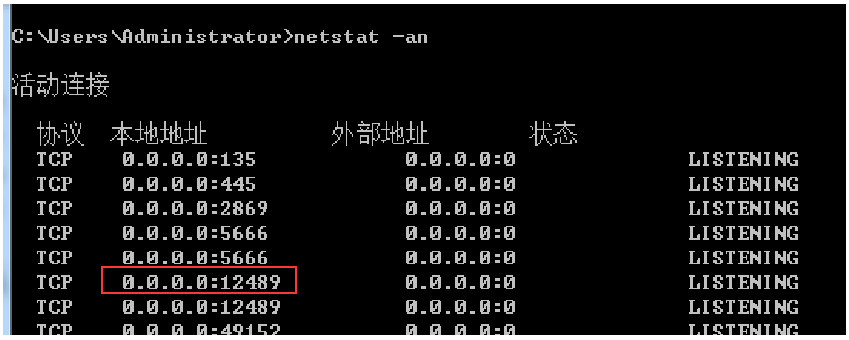


```
# If you want to fill this file with all av
#   nscp settings --generate --add-defaults
# If you want to activate a module and brin
#   nscp settings --activate-module <MODULE
# For details run: nscp settings --help


; Undocumented section
[/settings/default]

; Undocumented key
password =111111

; Undocumented key
allowed hosts = 192.168.1.71

; Undocumented section
[/settings/NRPE/server]
```

此密码需在nagios监控主机的command.cfg中指定

nagios监控主机IP

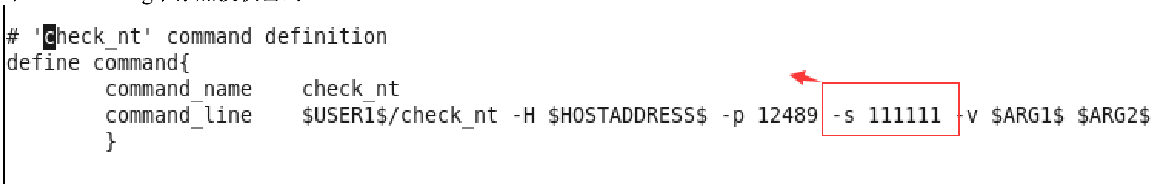
重启NSClient服务，检查12489端口如否启用



```
C:\Users\Administrator>netstat -an

活动连接

  协议  本地地址          外部地址        状态
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2869           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5666           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5666           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:12489          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:12489          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING
```

nagios配置：

在command.cfg中添加授权密码



```
# 'check_nt' command definition
define command{
        command_name    check_nt
        command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s 111111 -v $ARG1$ $ARG2$
        }
```

重启nagios服务

测试：

/usr/local/nagios/libexec/check_nt -H 主机IP -p 12489 -s 密码 -v UPTIME

如果反馈的信息是System Uptime - 0 day(s) 8 hour(s) 44 minute(s) 则表示连接正常。

如果显示为could not fetch information from server，则有以下可能

密码不正确（最傻的可能，也是经常发生的可能）

服务器上有防火墙，需要开放12489端口。

**7.2.4.4 邮件监控**

定义监控人邮件地址

vi /usr/local/nagios/etc/objects/contacts.cfg

```
1. define contact{
2.        contact_name                    nagiosadmin             ; Short name of user
3.        use                             generic-contact         ; Inherit default values from generic-contact template (defined above)
4.        alias                           Nagios Admin            ; Full name of user
5.        email                           yourname@domain.com
6.                                                                ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
7.        }
```

除了配置监控邮件的接收人外，还要确保：

- rpm -qa | grep sendmail
- yum -u install sendmail
- chkconfig --level 35 sendmail on
- service sendmil restart
- echo "This is test mail" | mail -s 'Test mail' 18717991553@139.com   //测试
- 本主机与邮件服务器互通
- 本主机SendMail可以使用外部SMTP服务发送邮件

### 7.2.4.5 校验配置

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

### 7.2.4.6 启动

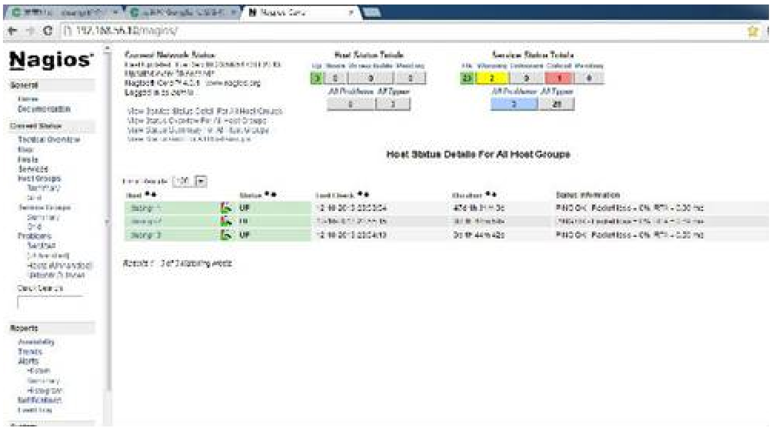
1. /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

nagios已经是一个服务，也可以执行如下操作：

1. service nagios start/stop/restart/status

2. 8. 监控页面

3. http://192.168.56.10/nagios

1. 

2.