

## 3. Adding A Remote Server To Icinga

[Page 1](#) [Page 2](#) [Page 3](#) [Page 4](#)

Monitoring localhost is nice, but of course, it would be even better if we could monitor all of our servers in one location. This is possible with Icinga, and this chapter describes how we can add our second Ubuntu 11.10 server to the setup.

To do this, we need to install the Nagios NRPE (Nagios Remote Plugin Executor) server on the server to be monitored, and the Nagios NRPE plugin on Icinga server (monitoring server). The NRPE server will listen on server to be monitored; and Icinga server will connect to it using the NRPE plugin and pass commands to it that the NRPE server will execute on the monitored server; finally, it will pass back the results to Icinga server.

First we install the nagios-nrpe-plugin package on Icinga server:

Icinga server

```
apt-get install nagios-nrpe-plugin
```

```
Nagios web administration password:<- nagiosadmin_password  
Password confirmation:<- nagiosadmin_password
```

Monitored server

Now we go to monitored server:

Install the nagios-nrpe-server package :

```
apt-get install nagios-nrpe-server
```

Now open /etc/nagios/nrpe.cfg :

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

We must configure the NRPE server to allow Icinga server to connect, therefore we add IP-ADDRESS Icinga server to the allowed\_hosts line:

```
[...]  
# ALLOWED HOST ADDRESSES  
# This is an optional comma-delimited list of IP address or hostnames  
# that are allowed to talk to the NRPE daemon.  
#  
# Note: The daemon only does rudimentary checking of the client's IP  
# address. I would highly recommend adding entries in your /etc/hosts.allow  
# file to allow only the specified host to connect to the port  
# you are running this daemon on.  
#
```

```
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

allowed_hosts=127.0.0.1,192.168.0.100
[...]
</file>
```

(If you don't do this, you will get the following error when you run

```
/usr/lib/nagios/plugins/check_nrpe -H 192.168.0.101 on Icinga server:
```

```
root@Icinga_server:/etc/nagios-plugins/config#
/usr/lib/nagios/plugins/check_nrpe -H IP_ADDRESS_SERVER.....here it is
IP address of monitored server\
```

```
CHECK_NRPE: Error - Could not complete SSL handshake.\
```

```
root@Icinga_server:/etc/nagios-plugins/config# )
```

Also, Icinga server needs to be allowed to pass command line arguments to the NRPE server, so still in the same file we set

```
dont_blame_nrpe
to
1
:
```

```
<code>[...]
```

```
# COMMAND ARGUMENT PROCESSING
```

```
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
```

```
#
#<nowiki> *</nowiki><nowiki>*</nowiki><nowiki>*</nowiki> ENABLING THIS OPTION
IS A SECURITY RISK!<nowiki> *</nowiki><nowiki>*</nowiki><nowiki>*</nowiki>
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
```

```
#
# Values: 0=do not allow arguments, 1=allow command arguments
```

```
dont_blame_nrpe=1
[...]
</file>
```

(If you don't do this, you will see the error

```
CHECK_NRPE: Received 0 bytes from daemon. Check the remote server logs for
error messages.
```

for lots of remote service checks in the Icinga web interface, and in /var/log/syslog on monitored server you will see these errors:

```
Aug 23 14:20:20 monitored_server nrpe[11496]: Error: Request contained command arguments, but argument option is not enabled!\
```

```
Aug 23 14:20:20 monitored_server nrpe[11496]: Client request was invalid, bailing out...
```

```
)
```

Finally we must add command definitions for each service check we want to run on monitored server and that is not already defined. I want to run the the check\_procs

```
,
check_all_disks
, and
check_mysql_cmdlinecred
checks on monitored server; these are not defined in
/etc/nagios/nrpe.cfg
, so I add them now (I also want to run the
check_users
and
check_load
checks, but these are already defined):
```

```
<code>[...]
command[check_procs]=/usr/lib/nagios/plugins/check_procs -w 250 -c 400
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w '20%' -c '10%'
-e
command[check_mysql_cmdlinecred]=/usr/lib/nagios/plugins/check_mysql -H
localhost -u 'nagios' -p 'howtoforge'
[...]
```

(If you don't do this, you will get errors like

```
NRPE: Command 'check_all_disks' not defined
NRPE: Command 'check_mysql_cmdlinecred' not defined
NRPE: Command 'check_procs' not defined
```

in the Icinga web interface.)

As you see I have hardcoded the command line arguments because using variables like command[check\_procs]=/usr/lib/nagios/plugins/check\_procs -w

```
$ARG1$
```

-C

\$ARG2\$ did not work for me. But still, when we configure the service checks for monitored server on Icinga server, we will have to pass command line arguments to these checks; monitored server will ignore these because I have hardcoded the comand line arguments into /etc/nagios/nrpe.cfg , but if you leave them out, you will get errors like /usr/lib/nagios/plugins/check\_nrpe: option requires an argument - 'a' in the Icinga web interface.

Now save the file and restart the NRPE server:

```
/etc/init.d/nagios-nrpe-server restart
```

Now check if the NRPE server is listening:

Icinga server

```
netstat -tap | grep nrpe

root@monitored_server:~# netstat -tap | grep nrpe
tcp        0      0 *:nrpe                *:*                    LISTEN
23668/nrpe
root@monitored_serv:~#
```

Now go back to Icinga server and configure to check if it can connect to the NRPE server on monitored server:

```
/usr/lib/nagios/plugins/check_nrpe -H IP_ADDRESS ;this time IP address of
monitored server
```

output should be as follows in case of success:

```
root@Icinga_server:~# /usr/lib/nagios/plugins/check_nrpe -H IP_ADDRESS ;IP
address of monitored server
NRPE v2.12
root@Icinga_server:~#
```

Monitored server

We want to check MySQL on monitored server; because we use the NRPE daemon, we can run the check locally on monitored server, i.e., we don't have to open MySQL to the outside to allow Icinga server to run the check. Therefore I create the MySQL user nagios for localhost and localhost.localdomain instead of for lp address of Icinga sever and server1.example.com :

```
mysql -u root -p

GRANT USAGE ON *.* TO nagios@localhost IDENTIFIED BY 'howtoforge';
GRANT USAGE ON *.* TO nagios@localhost.localdomain IDENTIFIED BY 'howtoforge';
FLUSH PRIVILEGES;
```

```
quit;
```

Now we go back to Icinga server...

```
Icinga server
```

.. and create the Icinga configuration for monitored\_server:

```
vi /etc/icinga/objects/server2_icinga.cfg
```

```
use                                generic-service                ; Name of service
template to use
    host_name                       server2.example.com
    service_description             Disk Space
    check_command                   check_nrpe!check_all_disks!20%!10%
}
define service{
    use                               generic-service
    host_name                         monitored_server.example.com ; OR
IP_ADDRESS
    service_description              Current Users
    check_command                    check_nrpe!check_users!20!50
}
define service{
    use                               generic-service
    host_name                         monitored_server.example.com ; OR
IP_ADDRESS
    service_description              Total Processes
    check_command                    check_nrpe!check_procs!250!400
}
define service{
    use                               generic-service                ; Name of
service template to use
    host_name                       monitored_server.example.com ; OR
IP_ADDRESS
    service_description              Current Load
    check_command                    check_nrpe!check_load!5.0!4.0!3.0!10.0!6.0!4.0
}
define service{
    use                               generic-service
    host_name                         monitored_server.example.com ; OR
IP_ADDRESS
    service_description              MySQL
    check_command                    check_nrpe!check_mysql_cmdlinecred!nagios!howtoforge
}
```

```
define service{
    use                generic-service
    host_name          monitored_server.example.com ; OR
IP_ADDRESS
    service_description SMTP
    check_command      check_smtp
}
define service{
    use                generic-service
    host_name          monitored_server.example.com ; OR
IP_ADDRESS
    service_description POP3
    check_command      check_pop
}
define service{
    use                generic-service
    host_name          monitored_server.example.com ; OR
IP_ADDRESS
    service_description IMAP
    check_command      check_imap
}
```

(As I've mentioned before, although I have hardcoded the command line arguments for some commands into `/etc/nagios/nrpe.cfg` on `monitored_server`, we still need to add command line arguments to certain these checks here.)

As you see, I use `check_nrpe` for some checks and pass the actual check (like `check_all_disks`) as a command line argument to `check_nrpe`. These are the checks that will be executed locally by the NRPE server on `monitored_server`

. `check_nrpe` is not needed for all checks. Checks that test a connection from the outside like `check_ping` or `check_smtp` can be run from `server1`.

To check the SSH and HTTP services on `monitored_server`

, we can EITHER add the following stanzas to `/etc/icinga/objects/server2_icinga.cfg` ...

```
[...]
define service {
    use                generic-service
    host_name          monitored_server.example.com ; OR
IP_ADDRESS
    service_description SSH
    check_command      check_ssh
}
define service {
    use                generic-service
    host_name          monitored_server.example.com ; OR
```

```
IP_ADDRESS
    service_description    HTTP
    check_command          check_http
}
```

... OR we add

monitored\_server.example.com to the http-servers and ssh-servers hostgroups in /etc/icinga/objects/hostgroups\_icinga.cfg :

```
vi /etc/icinga/objects/hostgroups_icinga.cfg
```

```
# Some generic hostgroup definitions

# A simple wildcard hostgroup
define hostgroup {
    hostgroup_name    all
                    alias        All Servers
                    members<nowiki>    *</nowiki>
}

# A list of your Debian GNU/Linux servers
define hostgroup {
    hostgroup_name    debian-servers
                    alias        Debian GNU/Linux Servers
                    members        localhost,monitored_server.example.com
}

# A list of your web servers
define hostgroup {
    hostgroup_name    http-servers
                    alias        HTTP servers
                    members        localhost,monitored_server.example.com
}

# A list of your ssh-accessible servers
define hostgroup {
    hostgroup_name    ssh-servers
                    alias        SSH servers
                    members        localhost,monitored_server.example.com
}
```

Restart Icinga:

```
/etc/init.d/icinga restart
```

Afterwards you should find server2 in the Icinga web interface:



[Page 1](#) [Page 2](#) [Page 3](#) [Page 4](#)

From:  
<http://vlired.cu/dokuwiki/> - **ICT Network Project**

Permanent link:  
[http://vlired.cu/dokuwiki/icinga:icinga\\_configuration\\_page3](http://vlired.cu/dokuwiki/icinga:icinga_configuration_page3)

Last update: **2015/06/29 12:10**

