

NAGIOS MONITORING

On Master server

1. Go to Nagios working directory

```
$ cd /usr/local/etc/nagios
```

2. Copy Nagios sample files

To do this, make sure you are in Nagios working directory from the first step

There are Nagios's sample files in the directory, which we need to copy them

as working copies. The easiest is to become temporarily root, and rename the files:

```
$ sudo -s
```

You can then copy the commands below:

```
cp cgi.cfg-sample cgi.cfg
cp nagios.cfg-sample nagios.cfg
cp resource.cfg-sample resource.cfg
cp objects/commands.cfg-sample objects/commands.cfg
cp objects/contacts.cfg-sample objects/contacts.cfg
cp objects/localhost.cfg-sample objects/localhost.cfg
cp objects/templates.cfg-sample objects/templates.cfg
cp objects/timeperiods.cfg-sample objects/timeperiods.cfg
```

```
# exit
$
```

3. Create monitoring configuration files for DNS Servers

While are you still in Nagios working directory, create a new file for DNS servers monitoring

Change the following below:

```
xx:      your group number
yy:      group number of your slave server
MYTLD:   your zone name
```

```
$ sudo vi objects/dns-servers.cfg
```

- - - - - cut below - - - - -

Define the host server

```
define host{
    use                frebsd-server
    host_name          master
    alias              master
    address            10.10.xx.1
}
```

```
define host{
    use                frebsd-server
```

```

        host_name      cache
        alias          cache
        address        10.10.xx.2
    }

    define host{
        use             freebsd-server
        host_name       slave
        alias           slave
        address         10.10.yy.1
    }

    ### Define the group

    define hostgroup{
        hostgroup_name  dns-servers
        alias           DNS Servers
        members         cache,master,slave
    }

    ### Define Services

    define service {
        use             generic-service      ; Name of service
    template to use
        hostgroup_name  dns-servers
        service_description  PING
        check_command   check_ping!100.0,20%!500.0,60%
    }

    define service {
        use             generic-service      ; Name of service
    template to use
        hostgroup_name  dns-servers
        service_description  Check DNS
        check_command   check_dns!www.MYTL
    }

    - - - - - cut end - - - - -

```

4. Add check_dns service

By default, Nagios doesn't configure check_dns service, we need to add line below into commands.cfg

```
$ sudo vi objects/commands.cfg
```

- Place this below the check_nt service check definition:

```

- - - - - cut below - - - - -

define command{
    command_name  check_dns
    command_line  $USER1$/check_dns -s $HOSTADDRESS$ -H $ARG1$
}

- - - - - cut end - - - - -

```

5. Enable DNS monitoring group in Nagios configuration files

Now you have DNS Server monitoring files (dns-server.cfg) created. We need to enable on nagios.cfg file for monitoring

```
$ sudo vi /usr/local/etc/nagios/nagios.cfg
```

Look for the line "cfg_file=/usr/local/etc/nagios/objects/localhost.cfg" and add below content

```
- - - - - cut below - - - - -  
  
# Definition for DNS servers  
cfg_file=/usr/local/etc/nagios/objects/dns-servers.cfg  
  
- - - - - cut end - - - - -
```

6. Enable Nagios service

To enable nagios service add the line below in /etc/rc.conf

```
nagios_enable="YES"
```

7. Start Nagios service

To start nagios service

```
$ sudo service nagios start
```

Nagios should be running right now without any error report.

8. Create Apache password authentication file for Nagios

Nagios required authentication via http to gain admin access to its web interface.

Let's create the password file

```
$ sudo -s  
# htpasswd -c /usr/local/etc/apache22/nagios.auth nagiosadmin  
# exit  
$
```

Set your desired password (perhaps, use the class password)

Password file is now created

9. Create Nagios' Apache configuration for web interface access

Go to Apache Includes directory

```
$ cd /usr/local/etc/apache22/Includes
```

If there already is a nagios.conf file in the directory, we will remove and rebuild it. Our new configuration is significantly different. To do this do:

```
$ sudo rm nagios.conf
```

Then edit the file:

```
$ sudo vi nagios.conf
```

Add the lines below into the file:

- - - - - cut below - - - - -

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

```
ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/
Alias /nagios/ /usr/local/www/nagios/
```

```
<Directory "/usr/local/www/nagios/cgi-bin">
    Options ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthName "Nagios Access"
    AuthType Basic
    Require valid-user
    AuthUserFile /usr/local/etc/apache22/nagios.auth
</Directory>
```

```
<Directory "/usr/local/www/nagios">
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthName "Nagios Access"
    AuthType Basic
    Require valid-user
    AuthUserFile /usr/local/etc/apache22/nagios.auth
</Directory>
```

- - - - - cut end - - - - -

10. Now, edit the main apache config.

```
$ sudo vi /usr/local/etc/apache22/httpd.conf
```

Find the line:

```
DirectoryIndex index.html
```

And change it to:

```
DirectoryIndex index.html index.php
```

11. Enable and start Apache web service if required, by adding this to /etc/rc.conf

```
$ sudo vi /etc/rc.conf
```

```
apache22_enable="YES"
```

(Save file and exit)

```
$ sudo service apache2 restart
```

12. Access to Nagios web page via `http://10.10.xxx.1/nagios/`

You will need to use "nagiosadmin" as the user and the password you specified previously using the "htpasswd" command.

Take a look around the Nagios interface to see what is being reported.