

Nagios lab parts 1 to 3

Nagios Installation and Configuration

Introduction

Goals

- Install and configure Nagios

Notes

- Commands preceded with “\$” imply that you should execute the command as a general user - not as root.
- Commands preceded with “#” imply that you should be working as root.
- Commands with more specific command lines (e.g. “rtrX>” or “mysql>”) imply that you are executing commands on remote equipment, or within another program.

Exercises

PART I

Log in to your virtual machine as the sysadm user.

Nagios Installation has already been done. We do these steps for practice

Install Nagios Version 3

```
$ sudo apt install nagios3 nagios3-doc
```

Nagios web administration password: Userid: nagiosadmin

```
$ sudo apt install --reinstall iputils-ping
```

At this point you will have a web server installed on your host, but you may need to start it. To enable and start your web server do:

```
$ sudo systemctl enable apache2
$ sudo systemctl start apache2
```

Let's see that apache2 is running as we expect:

```
$ sudo systemctl status apache2
```

Press 'q' to quit if the output fills the page.

You may need to configure Nagios to start whenever your host starts and then you will need to start the service. To do this do:

```
$ sudo systemctl enable nagios3
$ sudo systemctl start nagios3
```

Let's verify that nagios is now running as we expect:

```
$ sudo systemctl status nagios3
```

Press 'q' to quit if the output fills the page.

See Initial Nagios Configuration

Open a browser, and go to your machine like this:

```
http://oob.hostX.campusY.ws.nsrc.org/nagios3/
```

At the login prompt, login as:

```
User Name: nagiosadmin
Password: <CLASS PASSWORD>
```

Click on the "Hosts" link on the left of the initial Nagios page to see what has already been configured.

Click on the "Services" link to see what local services are being monitored¹.

Add Routers, PCs and Switches

We will create three files, routers.cfg, switches.cfg and pcs.cfg and make entries for the devices in your campus. If you want, you can simply create a single file for all items - Nagios will read any file named *.cfg and sort out the details on its own.

Creating the “routers.cfg” file

If you want some help to understand what your campus network looks like take a look at the detailed network diagram for campus1 linked on the main page for your workshop.

For each group you will end up monitoring each item in your campus, this includes:

Routers

- bdr1.campusY.ws.nsrc.org
- core1.campusY.ws.nsrc.org
- transit1.nren.ws.nsrc.org

Switches

- dist1-b1.campusY.ws.nsrc.org
- dist1-b2.campusY.ws.nsrc.org

Hosts

- host1.campusY.ws.nsrc.org
- host2.campusY.ws.nsrc.org
- host3.campusY.ws.nsrc.org
- host4.campusY.ws.nsrc.org
- host5.campusY.ws.nsrc.org
- host6.campusY.ws.nsrc.org
- srv1.campusY.ws.nsrc.org

```
$ cd /etc/nagios3/conf.d
$ sudo editor routers.cfg
```

“NOTE:” Y is the number of your campus (1, 2, 3, 4, 5, or 6)

```
define host {
    use          generic-host
    host_name    transit1.nren
    alias        Campus Y Transit Provider Router
    address      transit1.nren.ws.nsrc.org
}

define host {
    use          generic-host
    host_name    bdr1.campusY
    alias        Campus Y Border Router
    address      bdr1.campusY.ws.nsrc.org
}

define host {
```

```
use          generic-host
host_name    core1.campusY
alias        Core Router 1, Campus Y
address      core1.campusY.ws.nsrc.org
}
```

Now save the file and exit the editor.

Let's verify that our changes are working. On the command line do:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg
```

If you don't have any errors (warnings are OK), then you can reload the Nagios configuration:

```
$ sudo systemctl reload nagios3
```

And, in a web browser view:

```
http://oob.hostX.campusY.ws.nsrc.org/nagios3/
```

and click on hosts. You should now see your routers listed. They may still be waiting to be checked. Eventually they should turn green once Nagios runs a check.

Now we will do the same steps for our campus switches.

Creating the switches.cfg file

```
$ cd /etc/nagios3/conf.d                (just to be sure)
$ sudo editor switches.cfg
```

In this file add the following entries. You can COPY and PASTE, but be sure to update "Y" with your campus number:

```
define host {
    use          generic-host
    host_name    dist1-b1.campusY
    alias        Distribution Switch 1, Building 1, Campus Y
    address      dist1-b1.campusY.ws.nsrc.org
}

define host {
    use          generic-host
    host_name    dist1-b2.campusY
    alias        Distribution Switch 1, Building 2, Campus Y
```

```
    address    dist1-b2.campusY.ws.nsrc.org
}
```

Save the file and exit.

Creating the pcs.cfg File

Now we create entries for the 6 hosts (host1 through host6) and the campus shared server (srv1).

```
$ sudo editor pcs.cfg
```

For each campus place this entry at the top of the pcs.cfg file (replace “Y” with your campus number):

```
define host {
    use          generic-host
    host_name    srv1.campusY
    alias        Server, Campus Y
    address      srv1.campusY.ws.nsrc.org
}
```

(Note: the DNS name resolves to both IPv4 and IPv6 addresses, and Nagios will use IPv6 by default. You could instead use a literal IP address like 100.68.Y.X or 2001:db8:Y:1::X)

Here is a sample entry for host1. You can continue with the remaining hosts using this example.

```
define host {
    use          generic-host
    host_name    host1.campusY
    alias        Host 1, Campus Y
    address      host1.campusY.ws.nsrc.org
}
```

Now repeat this for all your remaining hosts. You should have a pcs.cfg file with one entry for srv1.campusY defined and your other 6 hosts, or a total of 7 entries.

REPEAT THE NEXT THREE STEPS EACH TIME YOU MAKE CHANGES TO YOUR CONFIGURATION

Verify that your configuration files are OK

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg
```

You will get some warnings like the ones below. You can ignore them for now.

```
Checking services...
  Checked 6 services.
Checking hosts...
Warning: Host 'bdr1.campus6' has no services associated with it!
Warning: Host 'core1.campus6' has no services associated with
it!
Warning: Host 'srv1.campus6' has no services associated with it!

etc....
...
Total Warnings: 11
Total Errors: 0
```

Things look okay - No serious problems were detected during the check. Nagios is saying that it's unusual to monitor a device just for its existence on the network, without also monitoring some service.

Reload/Restart Nagios

```
$ sudo systemctl reload nagios3
```

HINT: You will be doing this a lot. If you do it all on one line, like this, then you can use arrow-up and call back the command:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg && sudo systemctl
reload nagios3
```

The '&&' ensures that the reload only happens if the config is valid.

Verify via the Web Interface

Go to the web interface (<http://oob.hostX.campusY.ws.nsrc.org/nagios3>) and check that the hosts you just added are now visible in the interface. Click on the "Hosts" item on the left of the Nagios screen to see this. You may see it in "PENDING" status until the check is carried out.

View Status Map

Go to <http://oob.hostX.campusY.ws.nsrc.org/nagios3/>

Click on the "Map" item on the left. You should see all your hosts with the Nagios process in the middle. The "?" are because we have not told Nagios what type of host each item is (router, switch, AP, PC running Linux, etc...). We will do this later.

PART II - Configure Service Check for the Campus srv1 Server

Configuring

Now that we have our hardware configured we can start telling Nagios what services to monitor on the configured hardware, how to group the hardware in interesting ways, how to group services, etc.

Associate a service check for your campus server

```
$ sudo editor hostgroups_nagios2.cfg
```

- Find the hostgroup named “ssh-servers”. In the members section of the definition change the line:

```
members          localhost
```

to

```
members          localhost,srv1.campusY
```

where “Y” is your campus number

Exit and save the file.

Verify that your changes are OK:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg
```

Reload the Nagios configuration to see the new service association with your host:

```
$ sudo systemctl reload nagios3
```

In the Nagios web interface, find the “Services” link (left menu), and click on it.

You should be able to find your recent change:

```
srv1.campusY  SSH      PENDING  ...
```

PART III - Defining Services for all Servers

Determine what services to define for what devices

To start we are simply using ping to verify that our servers and network devices are responding or “Up”. So far we are only monitoring ssh on your campus server.

Now let’s add monitoring of services for our various servers and network devices:

In this class we, so far, have:

- routers: running ssh and ntp
- switches: running ssh and ntp
- pcs: All PCs are running ssh and http and will be running snmp (including srv1)

The classroom NOC is currently running an snmp daemon we can monitor if you wish.

Verify that SSH is running on the routers and workshop PCs images

In the file “services_nagios2.cfg” there is already an entry for the SSH service check, so you do not need to create this step. Instead, you simply need to re-define the “ssh-servers” entry in the file /etc/nagios3/conf.d/hostgroups_nagios2.cfg. We have just changed this file, and added the srv1.campusY host, so it looked like:

```
# A list of your ssh-accessible servers
define hostgroup {
    hostgroup_name  ssh-servers
        alias      SSH servers
        members     localhost,srv1.campusY
    }
```

Now update with your remaining campus items as well as the transit router. We will give one complete example below:

```
define hostgroup {
    hostgroup_name  ssh-servers
        alias      SSH servers
        members     host1.campusY, host2.campusY,
host3.campusY, host4.campusY, \
                    host5.campusY, host6.campusY,
srv1.campusY, transit1.nren, \
                    bdr1.campusY, core1.campusY, dist1-
b1.campusY, dist1-b2.campusY
    }
```


NOTES:

We have removed localhost from the entry above. We will do the same for our remaining hostgroups.

The “members” entry will be a long line and will likely wrap on the screen. If you want to start additional entries on newline then use "" to indicate a newline like this:

Only include entries you have defined previously. So, include all servers, routers and switches you have configured previously.

Be sure you change “Y” to your campus number.

Once you are done, run the pre-flight check and reload Nagios:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg && sudo systemctl
reload nagios3
```

... and view your changes in the Nagios web interface.

To continue with hostgroups you can add additional groups for later use, such as all your campus routers. Go ahead and edit the file hostgroups_nagios2.cfg again:

```
$ sudo editor hostgroups_nagios2.cfg
```

and add the following to the end of the file (COPY and PASTE this):

```
# A list of our virtual routers

define hostgroup {
    hostgroup_name routers
    alias          Cisco Routers and Switches for
CampusY
    members
bdr1.campusY,core1.campusY,transit1.nren
}
```

Save and exit from the file. Verify that everything is OK:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg
```

If everything looks good, then reload the Nagios configuration

```
$ sudo systemctl reload nagios3
```

Check that http is running on all your campus servers

Much like the ssh-servers hostgroup check we will create a check for http running across all of your hosts that run http in your campus. This will include:

Hosts

- host1.campusY.ws.nsrc.org
- host2.campusY.ws.nsrc.org
- host3.campusY.ws.nsrc.org
- host4.campusY.ws.nsrc.org
- host5.campusY.ws.nsrc.org
- host6.campusY.ws.nsrc.org
- srv1.campusY.ws.nsrc.org

Edit the file hostgroups_nagios2.cfg:

```
$ sudo editor hostgroups_nagios2.cfg
```

And look for this entry:

```
# A list of your web servers
define hostgroup {
    hostgroup_name http-servers
        alias          HTTP servers
        members        localhost
    }
```

Update this entry to add your hosts listed above. You can remove the localhost entry.

If you have questions or are confused please ask an instructor for help.

When you are done making the change, save your file and check that everything is OK:

```
$ sudo nagios3 -v /etc/nagios3/nagios.cfg
```

If everything looks good, then reload Nagios

```
$ sudo systemctl reload nagios3
```

Now go to the Nagios web interface and click on the Hosts menu choice on the left of the page.

You are ready to go on to the next set of exercises.

-
1. If you see error "DISK CRITICAL - /sys/kernel/debug/tracing is not accessible: Permission denied" then you have come across a minor bug (<https://bugs.launchpad.net/ubuntu/+source/nagios-plugins/+bug/15164510>) .

It is safe to ignore this, but if you want to fix it you would edit `/etc/nagios-plugins/config/disk.cfg` and change the `command_line` of the `check_all_disks` command definition like this:

```
# 'check_all_disks' command definition
define command{
    command_name    check_all_disks
    command_line    /usr/lib/nagios/plugins/check_disk -w
'$ARG1$' -c '$ARG2$' -e -X tracefs -X cgroup -X tmpfs
}
```