Nagios Installation and Configuration

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. "RTR-GW>" or "mysql>")
  imply that you are executing commands on remote equipment, or within
  another program.

Exercises

Exercises Part I

0. Log in to your virtual machine as the sysadm user.

1. Install Nagios Version 3

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

You will be prompted for nagiosadmin password. Give it the normal workshop
password.

2. See Initial Nagios Configuration

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

```
        http://pcN.ws.nsrc.org/nagios3/
```

At the login prompt, login as:

```
        user: nagiosadmin
        pass: <CLASS PASSWORD>
```

Browse to the "Host Detail" page to see what's already configured.

3. Remove the File host-gateway_nagios3.cfg

```
        $ sudo bash
        # cd /etc/nagios3/conf.d
        # rm host-gateway_nagios3.cfg
```

4. Update the File hostgroups_nagios2.cfg

```
        # editor hostgroups_nagios2.cfg
```

Go to the bottom of the file and find the entry:

```
define hostgroup {
        hostgroup_name  ping-servers
                alias           Pingable servers
                members         gateway
        }
```

Change the members line so that this looks like:

```
define hostgroup {
        hostgroup_name  ping-servers
                alias           Pingable servers
                members         rtrX
        }
```

Where "rtrX" is the router for your group. Now save and exit the from the file.

5. Add Routers, PCs and Switches

```
--------------------------------
```

We will create three files, routers.cfg, switches.cfg and pcs.cfg and make entries for the hardware in our classroom.

5a. Creating the switches.cfg file

```
----------------------------------
```

```
        # editor switches.cfg
```

In this file add the following entry:

```
define host {
    use         generic-host
    host_name   sw
    alias       Backbone Switch
    address     10.10.0.253
}
```

Save the file and exit.

5b. Creating the routers.cfg file

```
---------------------------------
```

We have 10 total routers. These are rtr1-rtr9 and gw-rtr. We will define entries for each of these.

```
        # editor routers.cfg
```

```
define host {
    use         generic-host
    host_name   gw-rtr
    alias       Classrooom Gateway Router
    address     10.10.0.254
}

define host {
    use         generic-host
    host_name   rtr1
    alias       Group 2 Gateway Router
    address     10.10.1.254
}

define host {
    use         generic-host
    host_name   rtr2
    alias       Group 2 Gateway Router
    address     10.10.2.254
}

define host {
    use         generic-host
    host_name   rtr3
    alias       Group 3 Gateway Router
    address     10.10.3.254
}

define host {
    use         generic-host
    host_name   rtr4
    alias       Group 4 Gateway Router
    address     10.10.4.254
}

define host {
    use         generic-host
    host_name   rtr5
    alias       Group 5 Gateway Router
    address     10.10.5.254
}
```

```
define host {
    use         generic-host
    host_name   rtr6
    alias       Group 6 Gateway Router
    address     10.10.6.254
}

define host {
    use         generic-host
    host_name   rtr7
    alias       Group 7 Gateway Router
    address     10.10.7.254
}

define host {
    use         generic-host
    host_name   rtr8
    alias       Group 8 Gateway Router
    address     10.10.8.254
}

define host {
    use         generic-host
    host_name   rtr9
    alias       Group 9 Gateway Router
    address     10.10.9.254
}

define host {
    use         generic-host
    host_name   ap1
    alias       Wireless Access Point 1
    address     10.10.0.251
}

define host {
    use         generic-host
    host_name   ap2
    alias       Wireless Access Point 2
    address     10.10.0.252
}
```

Now save and exit from the file.

5c. Creating the pcs.cfg File

Now we will create entries for all the Virtual Machines in our classroom. Below
we give you the first few entries. You should complete the file with as many PCs
as you wish to add. We recommend thet, at least, you add the 4 PCs that are members
of your group as well as an entry for the classroom NOC.

```
        # editors pcs.cfg
```

```
define host {
    use         generic-host
    host_name   noc
    alias       Workshop NOC machine
    address     10.10.0.250
}

#
# Group 1
#

define host {
    use         generic-host
    host_name   pc1
    alias       pc1
    address     10.10.1.1
}

define host {
    use         generic-host
    host_name   pc2
    alias       pc2
    address     10.10.1.2
}

define host {
    use         generic-host
    host_name   pc3
    alias       pc3
    address     10.10.1.3
}

define host {
    use         generic-host
    host_name   pc4
    alias       pc4
    address     10.10.1.4
}
```

You can save and exit from the file now, or you can continue to add more PC entries.
If you have not added PCs for your group be sure to do that before you exit from the
file.

STEPS 6a - 6c SHOULD BE REPEATED WHENEVER YOU UPDATE THE CONFIGURATION!
===

6a. Verify that your configuration files are OK

```
        # nagios3 -v /etc/nagios3/nagios.cfg
```

    ... You should get some warnings like :

```
Checking services...
        Checked 7 services.
Checking hosts...
Warning: Host 'gw-rtr' has no services associated with it!
Warning: Host 'rtr1' has no services associated with it!
Warning: Host 'rtr2' has no services associated with it!

etc....
...
Total Warnings: N
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.

6b. Reload/Restart Nagios

```
        # service nagios3 restart
```

Not always 100% reliable to use the "restart" option due to a bug in the Nagios init
script. To be sure you may want to get used to doing:

```
        # service nagios3 stop
        # service nagios3 start
```

6c. Verify via the Web Interface

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

HINT: You will be doing this a lot. If you do it all on one line, like this, then you can hit cursor-up and rerun all in one go:

```
        # nagios3 -v /etc/nagios3/nagios.cfg && /etc/init.d/nagios3 restart
```

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

7. View Host Detail and Status Map

Go to http://pcN.ws.nsrc.org/nagios3

Click on the "Host Detail" item on the left. Are all the hosts you have defined listed? Are they up?

Click on the "Status Map" item on the left. You should see all your hosts with the Nagios process in the middle.

PART II
Configure Service check for the classroom NOC

----

0. 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.

1. Associate a service check for our classroom NOC

```
    # editor hostgroups_nagios2.cfg
```

    - Find the hostgroup named "ssh-servers". In the members section of the defintion
      change the line:

```
members                 localhost
```

    to

```
members                 localhost,noc
```

Exit and save the file.

Verify that your changes are OK:

```
        # nagios3 -v /etc/nagios3/nagios.cfg
```

Restart Nagios to see the new service assocation with your host:

```
        # /etc/init.d/nagios3 restart
```

Click on the "Service Detail" link in the Nagios web interface to see your new entry.

PART III
Defining Services for all PCs

----

0. For services, the default normal_check_interval is 5 (minutes) in
   generic-service_nagios2.cfg. You may wish to change this to 1 to speed up
   how quickly service issues are detected, at least in the workshop.

1. Determine what services to define for what devices

   - This is core to how you use Nagios and network monitoring tools in
     general. So far we are simply using ping to verify that physical hosts
     are up on our network and we have started monitoring a single service on
     a single host (your PC). The next step is to decide what services you wish
     to monitor for each host in the classroom.

   - In this particular class we have:

```
     routers:  running ssh and snmp
     switches: running telnet and possibly ssh as well as snmp
     pcs:      All PCs are running ssh and http and should be running snmp
               The NOC is currently running an snmp daemon

     So, let's configure Nagios to check for these services for these
     devices.
```

2.) 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. The initial entry in the file looked like:

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

What do you think you should change? Correct, the "members" line. You should add in entries for all the classroom pcs, routers and the switches that run ssh. With this information and the network diagram you should be able complete this entry.

The entry will look something like this:

```
define hostgroup {
        hostgroup_name  ssh-servers
                alias           SSH servers
                members         localhost,pc1,pc2,pc3,pc4...,pc32,….ap1,ap2,s1,s2,noc,rtr1,rtr2…rtr9,gw-rtr
        }
```

Note: leave in "localhost" - This is your PC and represents Nagios' network point of view. So, for instance, if you are on "pc3" you would not include "pc3" in the list of all the classroom pcs as it is represented by the "localhost" entry.

The "members" entry will be a long line and will likely wrap on the screen.

Remember to include all your PCs and all your routers that you have defined. Do not include any entries if they are not already defined in pcs.cfg, switches.cfg or routers.cfg.

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

```
# nagios3 -v /etc/nagios3/nagios.cfg
```

If everything looks good, then restart Nagios

```
# /etc/init.d/nagios3 stop
# /etc/init.d/nagios3 start
```

and view your changes in the Nagios web interface.

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

```
# editor hostgroups_nagios2.cfg
```

and add the following to the end of the file:

```
# A list of our virtual routers
define hostgroup {
        hostgroup_name  cisco7200
                alias           Cisco 7200 Routers
                members         rtr1,rtr2,rtr3,rtr4,rtr5,rtr6,rtr7,rtr8,rtr9
        }
```

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

```
# nagios3 -v /etc/nagios3/nagios.cfg
```

If everything looks good, then restart Nagios

```
# service nagios3 stop
# service nagios3 start
```

3.) Check that http is running on all the classroom PCs.

- This is almost identical to the previous exercise. Just make the change to the HTTP service adding in each PC (no routers or switches). Remember, you don't need to add your machine as it is already defined as "localhost".

PART IV
Adding Parent Relationships

Each item is a child of either a switch or a router in our classroom, EXCEPT for your gateway router (rtrX) and the other members of your group. We are now going to add a "parents" statement for each device we have configured.

If you are unsure of the parent relationships you can look at our classroom Network Diagram. Remember, the parent relationships are from the point of view of your Nagios instance running on your pc.

1. Adding Parents to switches.cfg

```
        # cd /etc/nagios3/conf.d
        # editor switches.cfg
```

Update the entry:

```
define host {
    use         generic-host
    host_name   sw
    alias       Backbone Switch
    address     10.10.0.253
}
```

to be

```
define host {
    use         generic-host
    host_name   sw
    alias       Backbone Switch
    address     10.10.0.253
    parents     rtrX
}
```

Where "rtrX" is the gateway router for your group. I.E., for group 1 you would use "rtr1", for group 2, "rtr2" and so forth.

Save and exit from the file.

2. Adding Parents to routers.cfg

```
        # editor routers.cfg
```

For each entry we will add a "parents" line. So, for the gw-rtr definition at the top of the file this should now look like:

```
define host {
    use         generic-host
    host_name   gw-rtr
    alias       Classrooom Gateway Router
    address     10.10.0.254
    parents     sw
}
```

For all the remaining rtrX entries you should, also, add a line that says:

```
    parents     sw
```

EXCEPT For the rtrX entry for your group. There should be NO PARENTS entry.

So, if you are in group 2, then the entries for groups 1, 2 and 3 would look like:

```
define host {
    use         generic-host
    host_name   rtr1
    alias       Group 2 Router
    address     10.10.1.254
    parents     sw
}

define host {
    use         generic-host
    host_name   rtr2
    alias       Group 2 Router
    address     10.10.2.254
}

define host {
    use         generic-host
    host_name   rtr3
    alias       Group 3 Router
    address     10.10.3.254
    parents     sw
}
```

Update the rest of the file correctly and then save and exit from the file.

3. Adding Parents to pcs.cfg

For all the PC entries you should add a "parents" line that has the router
for that PC's group. For the noc the parent is the core switch or "sw"

```
#
# Classroom NOC
#

define host {
    use         generic-host
    host_name   noc
    alias       Workshop NOC machine
    address     10.10.0.250
    parents     sw
}
```

For PCs in Group 1 entries look like:

```
#
# Group 1
#

define host {
    use         generic-host
    host_name   pc1
    alias       pc1
    address     10.10.1.1
    parents     rtr1
}

define host {
    use         generic-host
    host_name   pc2
    alias       pc2
    address     10.10.1.2
    parents     rtr1
}
```

etc…

Do this for all the PCs in the remaining groups.

BUT, FOR THE 4 ENTRIES FOR THE PCS IN YOUR GROUP DO NOT ADD ANY PARENTS STATEMENT!

Save and exit from the file.

4. Restart Nagios and See the Updated Status Map

```
        # service nagios3 restart
```

If you have errors, fix these and try restarting again.

Open a web browser to http://pcN.ws.nsrc.org/nagios3 and click on the "Status Map" link on the left. Your map should now look quite different. You should a map that represents the Nagios world point of view from your machine.

PART V
Create More Host Groups

0. In the web view, look at the pages "Hostgroup Overview", "Hostgroup Summary", "Hostgroup Grid". This gives a convenient way to group together hosts which are related (e.g. in the same site, serving the same purpose).

1. Update /etc/nagios3/conf.d/hostgroups_nagios2.cfg

    - For the following exercises it will be very useful if we have created or update the following hostgroups:

      debian-servers
      routers
      switches

      If you edit the file /etc/nagios3/conf.d/hostgroups_nagios2.cfg you will see an entry for debian-servers that just contains localhost. Update this entry to include all the classroom PCs, including the noc (this assumes that you created a "noc" entry in your pcs.cfg file). Remember to skip your PC entry as it is represented by the localhost entry.

```
    # editor /etc/nagios3/conf.d/hostgroups_nagios2.cfg
```

Update the entry that says:

```
# A list of your Debian GNU/Linux servers
define hostgroup {
        hostgroup_name  debian-servers
                alias           Debian GNU/Linux Servers
                members         localhost
        }
```

So that the "members" parameter contains something like this. Use your classroom network diagram to confirm the exact number of machines and names in your workshop.

```
                members         localhost,pc1,pc2,pc3,pc4,pc5,pc6,pc7,pc8,pc9
                                pc10,pc11,pc12,pc13,pc14,pc15,pc16,pc17,pc18,
                                pc19,pc20,pc21,pc22,pc23,pc24,pc25,pc26,pc27,
                                pc28,pc29,pc30,pc31,pc32,pc33,pc34,pc35,pc36
```

Be sure that the line wraps and is not on separate lines. Otherwise you will get an error when you go to restart Nagios. Remember that your own PC is "localhost".

- Once you have done this, add in two more host groups, one for routers and one for switches. Call these entries "routers" and "switches".

- When you are done be sure to verify your work and restart Nagios.

      - Remember to skip your pc entry as it is represented by the localhost entry.

2. Go back to the web interface and look at your new hostgroups

PART VI
Extended Host Information ("making your graphs pretty")
----

1. Update extinfo_nagios2.cfg

    - If you would like to use appropriate icons for your defined hosts in
      Nagios this is where you do this. We have the three types of devices:

      Cisco routers
      Cisco switches
      Ubuntu servers

      There is a fairly large repository of icon images available for you to
      use located here:

      /usr/share/nagios/htdocs/images/logos/

      these were installed by default as dependent packages of the nagios3
      package in Ubuntu. In some cases you can find model-specific icons for
      your hardware, but to make things simpler we will use the following
      icons for our hardware:

```
/usr/share/nagios/htodcs/images/logos/base/debian.*
/usr/share/nagios/htdocs/images/logos/cook/router.*
/usr/share/nagios/htdocs/images/logos/cook/switch.*
```

    - The next step is to edit the file /etc/nagios3/conf.d/extinfo_nagios2.cfg
      and tell nagios what image you would like to use to represent your devices.

```
# editor /etc/nagios3/conf.d/extinfo_nagios2.cfg
```

      Here is what an entry for your routers looks like (there is already an entry
      for debian-servers that will work as is). Note that the router model (3600)
      is not all that important. The image used represents a router in general.

```
define hostextinfo {
        hostgroup_name   routers
        icon_image       cook/router.png
        icon_image_alt   Cisco Routers (3600)
        vrml_image       router.png
        statusmap_image  cook/router.gd2
}
```

      Now add an entry for your switches. Once you are done check your
      work and restart Nagios. Take a look at the Status Map in the web interface.
      It should be much nicer, with real icons instead of question marks.

PART VII
Create Service Groups
----

1. Create service groups for ssh and http for each set of pcs.

   - The idea here is to create three service groups. Each service group will
     be for a quarter of the classroom. We want to see these PCs grouped together
     and include status of their ssh and http services. To do this edit
     and create the file:

```
# editor /etc/nagios3/conf.d/servicegroups.cfg
```

     Here is a sample of the service group for group 1:

```
define servicegroup {
        servicegroup_name       group1-servers
        alias                   group 1 servers
        members                 pc1,SSH,pc1,HTTP,pc2,SSH,pc2,HTTP,pc3,SSH,pc3,HTTP,pc4,SSH,pc4,HTTP
        }
        - Note that the members line should wrap and not be on two lines.

        - Note that "SSH" and "HTTP" need to be uppercase as this is how the service_description is
          written in the file /etc/nagios3/conf.d/services_nagios2.cfg

        - You should create an entry for other groups of servers too

    - Save your changes, verify your work and restart Nagios. Now if you click on
      the Servicegroup menu items in the Nagios web interface you should see
      this information grouped together.



PART VIII
Configure Guest Access to the Nagios Web Interface
-----------------------------------------------------------------------------

1. Edit /etc/nagios3/cgi.cfg to give read-only guest user access to the Nagios
   web interface.

    - By default Nagios is configured to give full r/w access via the Nagios
      web interface to the user nagiosadmin. You can change the name of this
      user, add other users, change how you authenticate users, what users
      have access to what resources and more via the cgi.cfg file.

    - First, lets create a "guest" user and password in the htpasswd.users
      file.

    # htpasswd /etc/nagios3/htpasswd.users guest

      You can use any password you want (or none). A password of "guest" is
      not a bad choice.

    - Next, edit the file /etc/nagios3/cgi.cfg and look for what type of access
      has been given to the nagiosadmin user. By default you will see the following
      directives (note, there are comments between each directive):

      authorized_for_system_information=nagiosadmin
      authorized_for_configuration_information=nagiosadmin
      authorized_for_system_commands=nagiosadmin
      authorized_for_all_services=nagiosadmin
      authorized_for_all_hosts=nagiosadmin
      authorized_for_all_service_commands=nagiosadmin
      authorized_for_all_host_commands=nagiosadmin

      Now let's tell Nagios to allow the "guest" user some access to
      information via the web interface. You can choose whatever you would
      like, but what is pretty typical is this:

      authorized_for_system_information=nagiosadmin,guest
      authorized_for_configuration_information=nagiosadmin,guest
      authorized_for_system_commands=nagiosadmin
      authorized_for_all_services=nagiosadmin,guest
      authorized_for_all_hosts=nagiosadmin,guest
      authorized_for_all_service_commands=nagiosadmin
      authorized_for_all_host_commands=nagiosadmin

    - Once you make the changes, save the file cgi.cfg, verify your
      work and restart Nagios.

    - To see if you can log in as the "guest" user you may need to clear
      the cookies in your web browser. You will not notice any difference
      in the web interface. The difference is that a number of items that
      are available via the web interface (forcing a service/host check,
      scheduling checks, comments, etc.) will not work for the guest
      user.



PART IX
Optional Exercises
-------------------------------------------------------------------------------

1. Check that nagios is Running
-------------------------------
```

As opposed to just checking that a web server is
running on the classroom PCs, you could also check that the nagios3
service is available, by requesting the /nagios3/ path. This means
passing extra options to the check_http plugin.

For a description of the available options, type this:

```
      # /usr/lib/nagios/plugins/check_http
      # /usr/lib/nagios/plugins/check_http --help
```

and of course you can browse the online nagios documentation or google
for information on check_http. You can even run the plugin by hand to
perform a one-shot service check:

```
     # /usr/lib/nagios/plugins/check_http -H localhost -u /nagios3/
```

So the goal is to configure nagios to call check_http in this way.

```
define command{
        command_name    check_http_arg
        command_line    /usr/lib/nagios/plugins/check_http -H '$HOSTADDRESS$' $ARG1$
        }

define service {
        hostgroup_name                  nagios-servers
        service_description             NAGIOS
        check_command                   check_http_arg!-u /nagios3/
        use                             generic-service
}
```

and of course you'll need to create a hostgroup called nagios-servers to
link to this service check.

Once you have done this, check that Nagios warns you about failing
authentication (because it's trying to fetch the page without providing
the username/password). There's an extra parameter you can pass to
check_http_arg to provide that info, see if you can find it.

WARNING: in the tradition of "Debian Knows Best", their definition of the
check_http command in /etc/nagios-plugins/config/http.cfg
is *not* the same as that recommended in the nagios3 documentation.
It is missing $ARG1$, so any parameters to pass to check_http are
ignored. So you might think you are monitoring /nagios3/ but actually
you are monitoring root!

This is why we had to make a new command definition "check_http_arg".
You could make a more specific one like "check_nagios", or you could
modify the Ubuntu check_http definition to fit the standard usage.

2. Check that SNMP is running on the classroom NOC

- First you will need to add in the appropriate service check for SNMP in the file
  /etc/nagios3/conf.d/services_nagios2.cfg. This is where Nagios is impressive. There
  are hundreds, if not thousands, of service checks available via the various Nagios
  sites on the web. You can see what plugins are installed by Ubuntu in the nagios3
  package that we've installed by looking in the following directory:

```
    # ls /usr/lib/nagios/plugins
```

  As you'll see there is already a check_snmp plugin available to us. If you are
  interested in the options the plugin takes you can execute the plugin from the
  command line by typing:

```
    # /usr/lib/nagios/plugins/check_snmp
    # /usr/lib/nagios/plugins/check_snmp --help
```

  to see what options are available, etc. You can use the check_snmp plugin and
  Nagios to create very complex or specific system checks.

- Now to see all the various service/host checks that have been created using the
  check_snmp plugin you can look in /etc/nagios-plugins/config/snmp.cfg. You will
  see that there are a lot of preconfigured checks using snmp, including:

```
      snmp_load
      snmp_cpustats
      snmp_procname
      snmp_disk
      snmp_mem
      snmp_swap
      snmp_procs
      snmp_users
      snmp_mem2
      snmp_swap2
      snmp_mem3
      snmp_swap3
      snmp_disk2
      snmp_tcpopen
      snmp_tcpstats
      snmp_bgpstate
      check_netapp_uptime
      check_netapp_cupuload
      check_netapp_numdisks
      check_compaq_thermalCondition
```

And, even better, you can create additional service checks quite easily. For the case of verifying that snmpd (the SNMP service on Linux) is running we need to ask SNMP a question. If we don't get an answer, then Nagios can assume that the SNMP service is down on that host. When you use service checks such as check_http, check_ssh and check_telnet this is what they are doing as well.

- In our case, let's create a new service check and call it "check_system". This service check will connect with the specified host, use the private community string we have defined in class and ask a question of snmp on that ask - in this case we'll ask about the System Description, or the OID "sysDescr.0" -

- To do this start by editing the file /etc/nagios-plugins/config/snmp.cfg:

```
# joe /etc/nagios-plugins/config/snmp.cfg
```

At the top (or the bottom, your choice) add the following entry to the file:

```
# 'check_system' command definition
define command{
       command_name    check_system
       command_line    /usr/lib/nagios/plugins/check_snmp -H '$HOSTADDRESS$' -C
'$ARG1$' -o sysDescr.0
        }
```

You may wish to copy and paste this vs. trying to type this out.

Note that "command_line" is a single line. If you copy and paste in joe the line may not wrap properly and you may have to manually add the part:

```
'$ARG1$' -o sysDescr.0
```

to the end of the line.

- Now you need to edit the file /etc/nagios3/conf.d/services_nagios2.cfg and add in this service check. We'll run this check against all our servers in the classroom, or the hostgroup "debian-servers"

- Edit the file /etc/nagios3/conf.d/services_nagios2.cfg

```
# joe /etc/nagios3/conf.d/services_nagios2.cfg
```

At the bottom of the file add the following definition:

```
# check that snmp is up on all servers
define service {
        hostgroup_name                  snmp-servers
        service_description             SNMP
        check_command                   check_system!xxxxxx
        use                             generic-service
        notification_interval           0 ; set > 0 if you want to be renotified
}
```

The "xxxxxx" is the community string previously (or to be) defined in class.

Note that we have included our private community string here vs. hard-coding

```
      it in the snmp.cfg file earlier. You must change the "xxxxx" to be the snmp
      community string given in class or this check will not work.

    - Now we must create the "snmp-servers" group in our hostgroups_nagios2.cfg file.
      Edit the file /etc/nagios3/conf.d/hostgroups_nagios2.cfg and go to the end of the
      file. Add in the following hostgroup definition:

# A list of snmp-enabled devices on which we wish to run the snmp service check
define hostgroup {
           hostgroup_name       snmp-servers
                   alias        snmp servers
                   members      noc
          }

        - Note that for "members" you could, also, add in the switches and routers for
          group 1 and 2. But, the particular item (MIB) we are checking for "sysDescr.0"
          may not be available on the switches and/or routers, so the check would then fail.

    - Now verify that your changes are correct and restart Nagios.

    - If you click on the Service Detail menu choice in web interface you should see
      the SNMP check appear for the noc host.

    - After we do the SNMP presentation and exercises in class, then you could come
      back to this exercise and add in all the classroom PCs to the members list in the
      hostgroups_nagios2.cfg file, snmp-servers hostgroup definition. Remember to list
      your PC as "localhost".
```