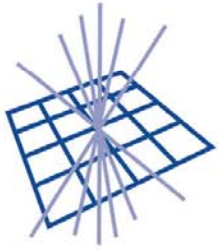
Nagios:
An introduction and Brief Tutorial

Chris Brew
SciTech/PPD

Science & Technology Facilities Council
Particle Physics Department



GridPP
UK Computing for Particle Physics

Outline

- What is Nagios?
- Getting Started
  - Hosts, Commands, Services, Timeperiods and Contacts
  - Remote Checks with NRPE
  - Hostgroups and Servicegroups
  - Templates
  - Config File(s)
  - Active v. Passive checks
- Going Further
  - Writing you own Checks
  - NSCA
  - Service Hierarchies
  - Eventhandlers
  - Modifying the Web Pages

What is Nagios

- "Nagios is an enterprise-class monitoring solutions for hosts, services, and networks released under an Open Source license."

  www.nagios.org

- "Nagios is a popular open source computer system and network monitoring application software. It watches hosts and services that you specify, alerting you when things go bad and again when they get better."

  www.wikipedia.org

In My Opinion

A Badly configured Nagios is one of the most effective senders of SPAM to sysadmins there is but with a small amount of work it can be an invaluable tool for understanding your cluster

Installation

- Nagios RPMs for RHEL (and so SL/SLC) available from the DAG repository
- 4 Main component RPMS
  - nagios – the main server software and web scripts
  - nagios-plugins – the common set of check scripts used to query services
  - nagios-nrpe – Nagios Remote Plugin Executor
  - nagios-nsca – Nagios Service Check Acceptor
- Setup is simply a matter on installing the RPMs, configuring your web server and editing the config files to suit your setup

Architecture

- Simplest setup has central server running Nagios daemon that runs local check scripts which the status of services on that and remote hosts
- A host is a computer running on the network which runs one or more services to be checked
- A service is anything on the host that you want checked. Its state can be one of: OK, Warning, Critical or Unknown
- A check is a script run on the server whose exit status determines the state of the service: 0, 1, 2 or -1

hosts

```
define host{
        host_name                       my-host
        alias                           my-host.domain.ac.uk
        address                         168.192.0.1
        check_command                   check-host-alive
        max_check_attempts              10
        check_period                    24x7
        notification_interval           120
        notification_period             24x7
        notification_options            d,r
        contact_groups                  unix-admins
        register                        1
        }
```

Services

```
define service{
        name                            ping-service
        service_description             PING
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              4
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  unix-admins
        notification_options            w,u,c,r
        notification_interval           960
        notification_period             24x7
        check_command    check_ping!100.0,20%!500.0,60%
        hosts                           my-host
        register                        1
        }
```

Command

- Commands wrap the check scripts

```
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H     $HOSTADDRESS$ -w
  99,99% -c 100,100% -p 1
        }
```

- and the alerts

```
define command{
        command_name    notify-by-email
        command_line    /usr/bin/printf "%b" "***** Nagios
  *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService:
  $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
  $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
  Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "** $NOTIFICATIONTYPE$
  alert - $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **"
  $CONTACTEMAIL$
        }
```

Check Scripts

- The standard nagios-plugins rpm provides over 130 different check scripts, ranging from check_load to check_oracle_instance.p via check_procs, check_mysql, check_mssql, check_real and check_disk
- Writing you own check scripts is easy, can be in any language.
  - Active scripts just need to set the exit status and output a single line of text
  - Passive checks just write a single line to the servers command file

Contacts

- Contacts are the people who receive the alerts:

```
define contact{
        contact_name                    chris
        alias                           Chris Brew
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           someone@somewhere
        }
```

- Contactgroups group contacts:

```
define contactgroup{
        contactgroup_name       unix-admins
        alias                   Unix Administrators
        members                 chris
        }
```

Time Periods

- Time periods define when things, checks or alerts, happen:

```
define timeperiod{
        timeperiod_name 24x7
        alias           24 Hours A Day, 7 Days A Week
        sunday          00:00-24:00
        monday          00:00-24:00
        tuesday         00:00-24:00
        wednesday       00:00-24:00
        thursday        00:00-24:00
        friday          00:00-24:00
        saturday        00:00-24:00
        }
```

Remote checks with NRPE

- NRPE is a daemon that runs on a remote host to be checked and a corresponding check script on the Master Nagios server
- Nagios Daemon runs the check_nrpe script which contacts the daemon which runs the check script locally and returns the output:

Nrpe.cfg (on remote host):

```
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c
   30,25,20
```

Nagios.cfg (on Master server):

```
define command{
        command_name    check_nrpe_load
        command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c check_load
        }
```

Basic WN NRPE Config

```
pid_file=/var/run/nrpe.pid
server_port=5666
nrpe_user=nagios
nrpe_group=nagios
dont_blame_nrpe=0
debug=0
command_timeout=60
command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5
   -c 10
command[check_zombie_procs]=/usr/.../check_procs -w 5 -c 10 -s Z
command[check_wn_ssh]=/usr/.../check_wn_ssh.sh
command[check_load]=/usr/.../check_load -w 30,20,10 -c 45,30,20
command[check_total_procs]=/usr/.../check_procs -w 300 -c 400
command[check_system_disk]=/usr/.../check_disk -w 20% -c 10%
   -p /dev/sda2
command[check_scratch_disk]=/usr/.../check_disk -w 20% -c 10%
   -p /dev/sda5
```

Host and Service Groups

- Host and service groups let you group together similar hosts and services:

```
define hostgroup{
        hostgroup_name  4-ServiceNodes
        alias           RALPP Service Nodes
        }
define servicegroup{
        servicegroup_name       topgrid
        alias                   Top Grid Services
        }
```

- Plus a hostgroups or a servicegroups line in the host or service definition

Templates

- You can define templates to make specifying hosts and services easier:

```
define host{
        name                    generic-unix-host
        use                     generic-host
        check_command           check-host-alive
        max_check_attempts      10
        check_period            24x7
        notification_interval   120
        notification_period     24x7
        notification_options    d,r
        contact_groups          unix-admins
        register                0
        }
```

- Reduces a host definition to:

```
define host{
        use                     generic-grid-frontend-host
        host_name               heplnx201
        alias                   heplnx201.pp.rl.ac.uk
        address                 130.246.47.201
}
```

Config Files

- Main nagios.cfg file can have include statements to pulling other setting files or directories of files
  - The standard example config files are confusingly spred over several possible files, many of which need editing to get anything working.
- My current set up has the config spread over multiple files and directories.
  - One set of top level files defining global settings, commands, contact, hostgroups, servicegroups, host-templates, service-templates, time-periods, resources (user variables)
  - One directory for each host group containing one file defining the services and one defining the hosts

Active vs Passive Checks

- For some services running a script to check their state every few minutes (so called active checking) is not the best way.
  - Service has its own internal monitoring
  - One script can efficiently check the status of multiple related services
- The nagios service can be set to read “commands” from a named pipe
  - Any process can then write in a line updating the status of a service (passive check)
  - Web frontend’s cgi script can also write commands to the file to disable checks or notifications for a host or service for example.

Going Further 1

- NSCA, is a script/daemon pair that allow remote hosts to run passive checks and write the results into that nagios servers command file.
  - Checking operation on remote host calls send_nsca script which forwards the result to the nsca daemon on the server which writes the result into the command file
  - Can be used with eventhandlers to produce a hierarchy of Nagios servers
- Service Hierachies, services and hosts can depend on other services or hosts so for instance:
  - If the web server is down don't tell me the web is unreachable
  - If the switch is down don't send alerts for the hosts behind it

Going Further 2

- Event Handlers, as well as just telling you a service is down Nagios can attempt to rectify the fault by running an eventhandler
- The cgi scripts, templates and style sheets that build the web pages can be edited to add extra information
- Nagios has a myriad of other features not touched upon here from state stalking to flap detection, notification escalations to scheduling network, host or service downtimes

Conclusions

- Nagios is a vary useful tool for the time poor sysadmin but can be very daunting when you first look at it.
- My advice is:
  - Install it on you test node (though this may well end up as your master server)
  - Run a few check scripts by hand to get the feel for them
  - Set up a simple config file that runs a few check on the local host
  - Install nrpe on the host and nrpe and nagios-plugins on a remote host
  - Run check nrpe by hand to get it working then add a couple of simple checks on the remote host
  - NOW THINK ABOUT HOW YOU WANT TO ORGANISE YOU CONFIG FILES
  - Now add hosts and service until you run out, then write some more