scmGalaxy

SCM – DevOps – Build – Release

Nagios Setup and Configuration

By – Rajesh Kumar

Twitter - RajeshKumarIN

Email – DevOps@RajeshKumar.xyz

Configuration Files

Located in /etc/nagios3/

Important files include:

- cgi.cfg Controls the web interface and security options.
- commands.cfg The commands that Nagios uses for notifications.
- nagios.cfg Main configuration file.
- conf.d/* All other configuration goes here!

Configuration files continued

Under conf.d/*   (*sample only*)

contacts_nagios3.cfg	users and groups
generic-host_nagios2.cfg	default host template
generic-service_nagios2.cfg	default service template
hostgroups_nagios2.cfg	groups of nodes
services_nagios2.cfg	what services to check
timeperiods_nagios2.cfg	when to check and who to notifiy

Configuration files continued

Under conf.d some other possible configfiles:

host-gateway.cfg	Default route definition
extinfo.cfg	Additional node information
servicegroups.cfig	Groups of nodes and services
localhost.cfg	Define the Nagios server itself
pcs.cfg	Sample definition of PCs (hosts)
switches.cfg	Definitions of switches (hosts)
routers.cfg	Definitions of routers (hosts)

Pre-installed plugins in Ubuntu

check_bgpstate check_hpjd check_mailq check_overcr
check_ssmtp check_breeze check_http check_mrtg
check_pgsql check_swap check_by_ssh check_icmp
check_mrtgtraf check_ping check_tcp check_clamd
check_ide_smart check_mysql check_pop check_time
check_cluster check_ifoperstatus check_mysql_query
check_procs check_udp check_dhcp check_ifstatus
check_nagios check_radius check_ups check_dig
check_imap check_nntp check_real check_users
check_disk check_ircd check_nntps check_rpc
check_wave check_disk_smb check_jabber check_nt
check_sensors check_dns check_ldap check_ntp check_spop
check_simap check_dummy check_ldaps check_ntp_peer
check_smtp check_file_age check_linux_raid check_ntp_time
check_snmp check_flexlm check_load check_nwstat

Nodes and services configuration

Based on templates

- This saves lots of time avoiding repetition
- Similar to Object Oriented programming

Create default templates with default parameters for a:

- generic node
- generic service
- generic contact

Generic node template

```
define host{
    name                     generic-host
    notifications_enabled        1
    event_handler_enabled        1
    flap_detection_enabled       1
    process_perf_data            1
    retain_status_information    1
    retain_nonstatus_information 1
    check_command                check-host-alive
    max_check_attempts           5
    notification_interval        60
    notification_period          24x7
    notification_options             d,r
    contact_groups               nobody
    register                     0
    }
```

Individual node configuration

```
define host{
        use                     generic-host
        host_name               switch1
        alias                   Core_switches
        address                 192.168.1.2
        parents                 router1
        contact_groups          switch_group
}
```

Generic service configuration

```
define service{
        name                                generic-service
        active_checks_enabled               1
        passive_checks_enabled              1
        parallelize_check                   1
        obsess_over_service                     1
        check_freshness                     0
        notifications_enabled                   1
        event_handler_enabled               1
        flap_detection_enabled                  1
        process_perf_data                   1
        retain_status_information               1
        retain_nonstatus_information            1
        is_volatile                         0
        check_period                        24x7
        max_check_attempts                  5
        normal_check_interval               5
        retry_check_interval                1
        notification_interval               60
        notification_period                 24x7
        notification_options                c,r
        register                            0
        }
```

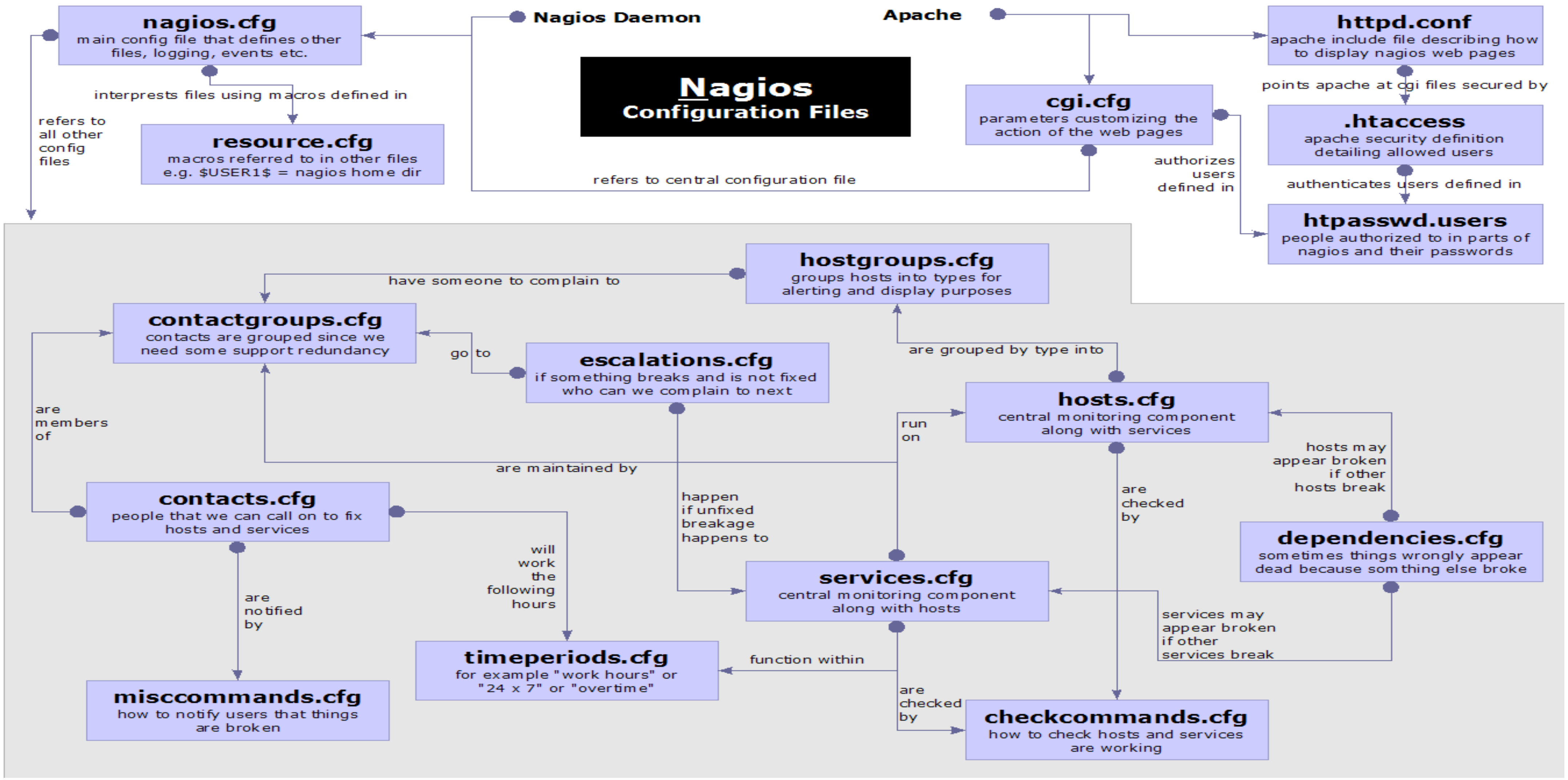
Individual service configuration

```
define service{
        host_name                switch1
        use                      generic-service
        service_description           PING
        check_command            check-host-alive
        max_check_attempts       5
        normal_check_interval    5
        notification_options          c,r,f
        contact_groups           switch-group
}
```

Beeper and sms messages

- It's important to integrate Nagios with something available outside of work
  - Problems occur after hours... (unfair, but true)
- A critical item to remember: an SMS or message system should be independent from your network.
  - You can utilize a modem and a telephone line
  - Packages like sendpage, qpage or gnokii can help.

Nagios Configuration



Nagios Configuration Files

Nagios Daemon

Apache

nagios.cfg
main config file that defines other files, logging, events etc.

interpresting files using macros defined in

resource.cfg
macros referred to in other files e.g. $USER1$ = nagios home dir

refers to all other config files

refers to central configuration file

cgi.cfg
parameters customizing the action of the web pages

authorizes users defined in

httpd.conf
apache include file describing how to display nagios web pages

points apache at cgi files secured by

.htaccess
apache security definition detailing allowed users

authenticates users defined in

htpasswd.users
people authorized to in parts of nagios and their passwords

hostgroups.cfg
groups hosts into types for alerting and display purposes

have someone to complain to

contactgroups.cfg
contacts are grouped since we need some support redundancy

go to

escalations.cfg
if something breaks and is not fixed who can we complain to next

are grouped by type into

hosts.cfg
central monitoring component along with services

are members of

run on

are maintained by

happen if unfixed breakage happens to

are checked by

hosts may appear broken if other hosts break

contacts.cfg
people that we can call on to fix hosts and services

will work the following hours

are notified by

services.cfg
central monitoring component along with hosts

dependencies.cfg
sometimes things wrongly appear dead because somthing else broke

services may appear broken if other services break

function within

timeperiods.cfg
for example "work hours" or "24 x 7" or "overtime"

are checked by

misccommands.cfg
how to notify users that things are broken

checkcommands.cfg
how to check hosts and services are working

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
        }
```

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

thank you!