Network Management & Monitoring

NAGIOS

Introduction

Network Monitoring Tools

- Availability
- Reliability
- Performance

Nagios actively monitors the availability of devices and services

Introduction

- Possibly the most used open source network monitoring software.
- Has a web interface.
  - Uses CGIs written in C for faster response and scalability.
- Can support up to thousands of devices and services.

Installation

In Debian/Ubuntu

```
# apt-get install nagios3
```

Key directories

```
/etc/nagios3
/etc/nagios3/conf.d
/etc/nagios-plugins/conf
/usr/lib/nagios/plugins
/usr/share/nagios3/htdocs/images/logos
```

Nagios web interface is here:

http://YourMachine/nagios3/

Plugins

Plugins are used to verify services and devices:

- Nagios architecture is simple enough that writing new plugins is fairly easy in the language of your choice.
- There are ***many, many*** plugins available (thousands).
  - ✓ http://exchange.nagios.org/
  - ✓ http://nagiosplugins.org/

Features

- Configuration done in text files, based on templates.
- Nagios reads its configuration from a directory. You determine how to divide your configuration files.
- Uses parallel checking and forking for scalability

Features cont.

- Utilizes topology to determine dependencies.
  - Differentiates between what is *down* vs. what is *unreachable*. Avoids running unnecessary checks and sending redundant alarms
- Allows you to define how to send notifications based on combinations of:
  - Contacts and lists of contacts
  - Devices and groups of devices
  - Services and groups of services
  - Defined hours by persons or groups.
  - The state of a service.

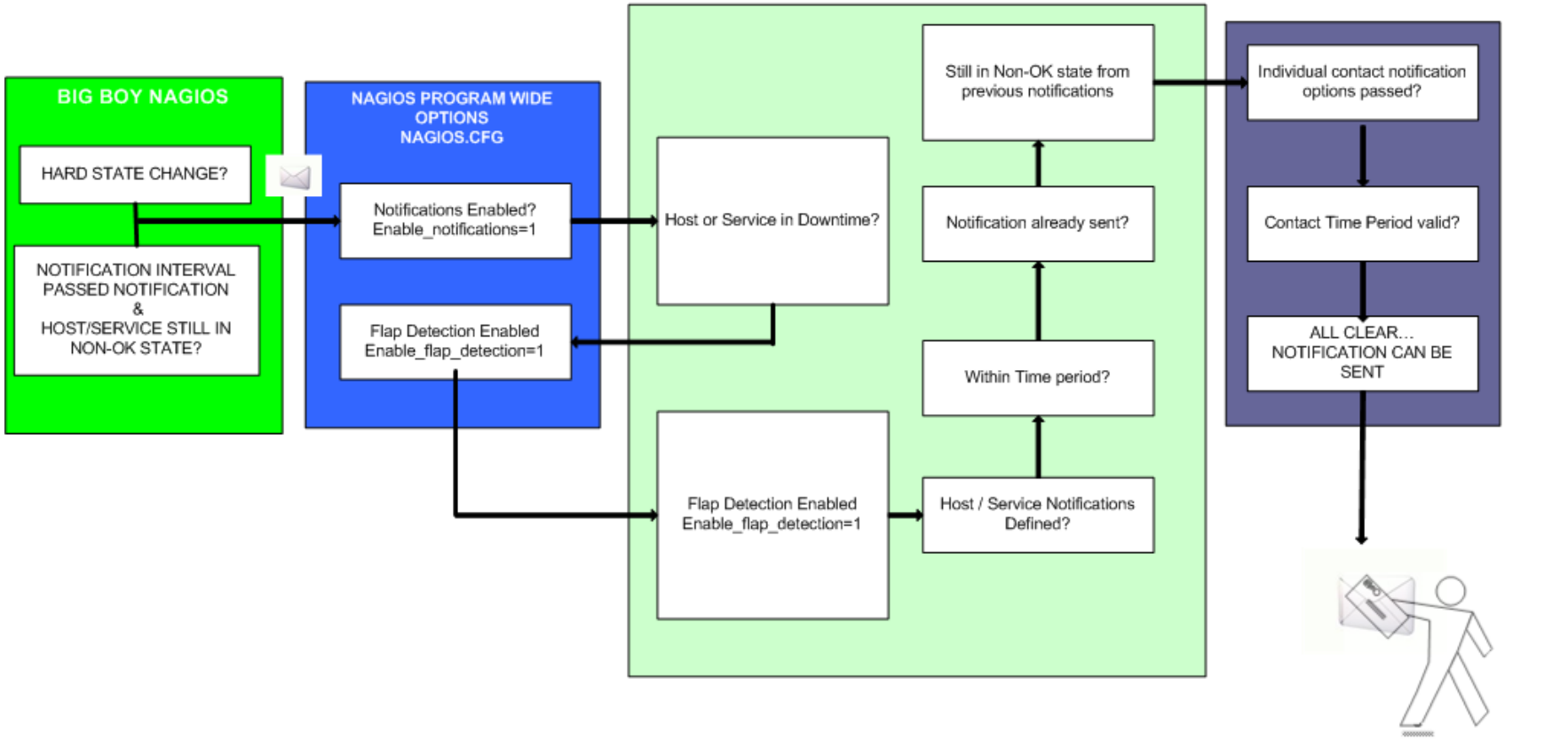
Notification Options (Host)

Host state:

When configuring a host you have the following notification options:

- **d:** DOWN
- **u:** UNREACHABLE
- **r:** RECOVERY
- **f:** FLAPPING
- **n:** NONE

NAGIOS - NOTIFICATION FLOW DIAGRAM



NOTE: The flow will only continue when each of the listed filters are satisfied.

CONTACT GETS THE NOTIFICATION MESSAGE

How checks work

- A node/host/device consists of one or more service checks (PING, HTTP, MYSQL, SSH, etc.)
- Periodically Nagios checks each service for each node and determines if state has changed. State changes are:
  - CRITICAL
  - WARNING
  - UNKNOWN
- For each state change you can assign:
  - Notification options (as mentioned before)
  - Event handlers

How checks work continued

Parameters

- Normal checking interval
- Re-check interval
- Maximum number of checks.
- Period for each check

- Node checks only happen when services respond.
  - A node can be:
    - DOWN
    - UNREACHABLE

How checks work continued

- By default Nagios does a node check 3 times before it will change the node's state to down.
- No response states goes to *warning* then *critical*

The concept of “parents”

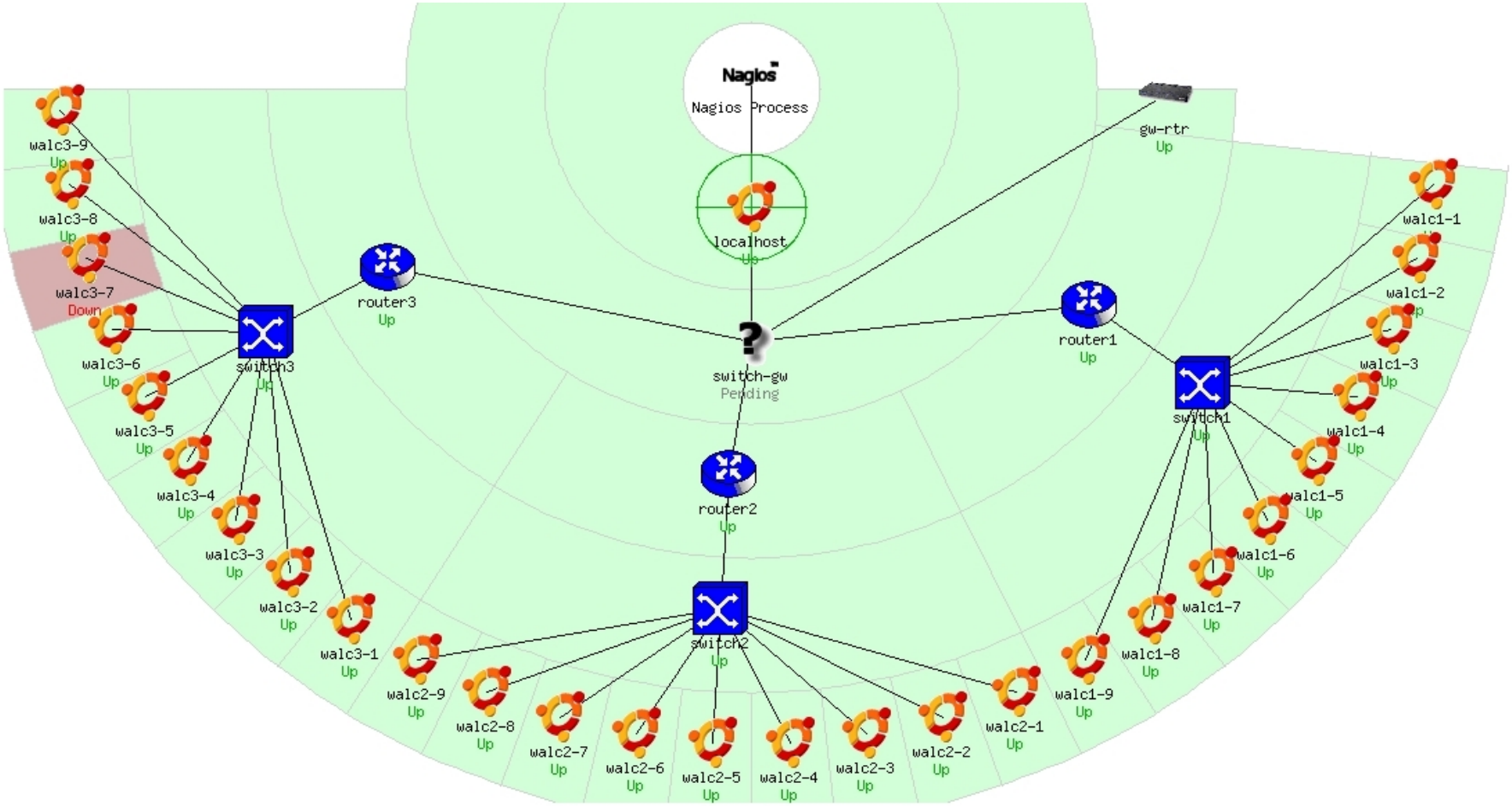
Nodes can have parents:

- The parent of a PC connected to a switch would be the switch.
- Allows us to specify the dependencies between devices.
- Avoids sending alarms when parent does not respond.
- A node can have multiple parents (dual homed).

Network viewpoint

- Where you locate your Nagios server will determine your point of view of the network.
- The Nagios server becomes the “root” of your dependency tree

Network viewpoint

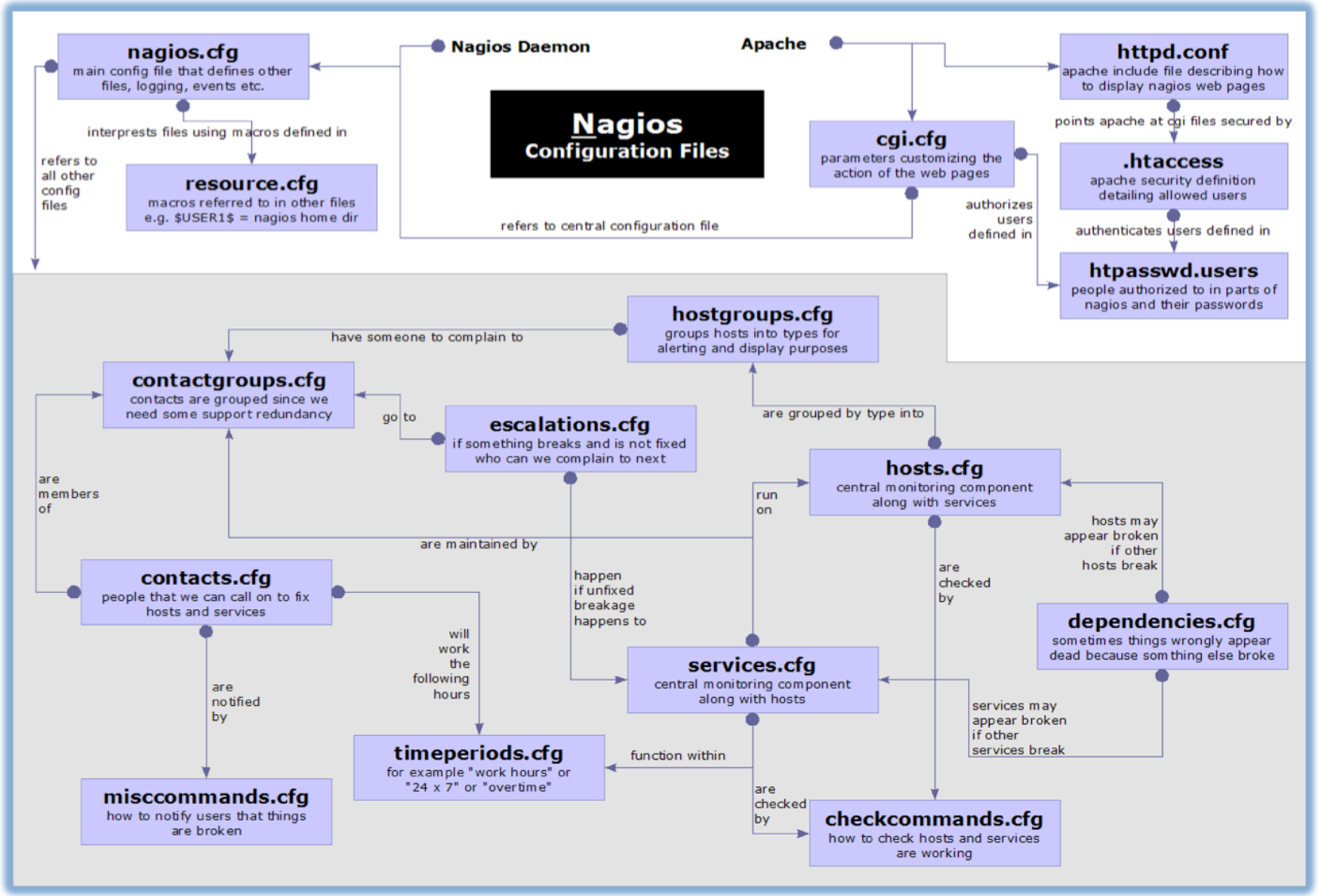


Demo Nagios

Configuration Files

- Lots!
- Can seem complex at first
- **Object oriented**
  - Objects (devices or services) inherit attributes.
  - Apply functionality to *groups of devices.*
  - Do not apply functionality to individual objects. Does not scale!
  - Once you understand Nagios configs the rest is easy…

Configuration files (Official)



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/*

contacts_nagios2.cfg	users and groups
extinfo_nagios2.cfg	make your UI pretty
generic-host_nagios2.cfg	default host template
generic-service_nagios2.cfg	default service template
host-gateway_nagios3.cfg	host at default gw definition
hostgroups_nagios2.cfg	groups of nodes
localhost_nagios2.cfg	definition of nagios host
services_nagios2.cfg	what services to check
timeperiods_nagios2.cfg	when to check who to notify

Configuration files continued

Under conf.d some other possible config files:

- servicegroups.cfg — Groups of nodes and services
- pcs.cfg — Sample definition of PCs (hosts)
- switches.cfg — Definitions of switches (hosts)
- routers.cfg — Definitions of routers (hosts)

Pre-installed plugins in Ubuntu

/usr/lib/nagios/plugins

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

/etc/nagios-plugins/config

apt.cfg	disk-smb.cfg	ftp.cfg	ldap.cfg	mysql.cfg	ntp.cfg	radius.cfg	ssh.cfg
breeze.cfg	dns.cfg	hppjd.cfg	load.cfg	netware.cfg	pgsql.cfg	real.cfg	tcp_udp.cfg
dhcp.cfg	dummy.cfg	http.cfg	mail.cfg	news.cfg	ping.cfg	rpc-nfs.cfg	telnet.cfg
disk.cfg	flexlm.cfg	ifstatus.cfg	mrtg.cfg	nt.cfg	procs.cfg	snmp.cfg	users.cfg

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    ; The name of this host template
        notifications_enabled           1               ; Host notifications are enabled
        event_handler_enabled           1               ; Host event handler is enabled
        flap_detection_enabled          1               ; Flap detection is enabled
        failure_prediction_enabled      1               ; Failure prediction is enabled
        process_perf_data               1               ; Process performance data
        retain_status_information       1               ; Retain status information across program restarts
        retain_nonstatus_information    1               ; Retain non-status information across program restarts
                check_command                   check-host-alive
                max_check_attempts              10
                notification_interval           0
                notification_period             24x7
                notification_options            d,u,r
                contact_groups                  admins
        register                        0       ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
        }
```

Individual node configuration

```
define host{
        use                     generic-host
        host_name               gw-rtr
        alias                   Main workshop router
        address                 192.0.2.1
        contact_groups          router_group
}
```

Generic service configuration

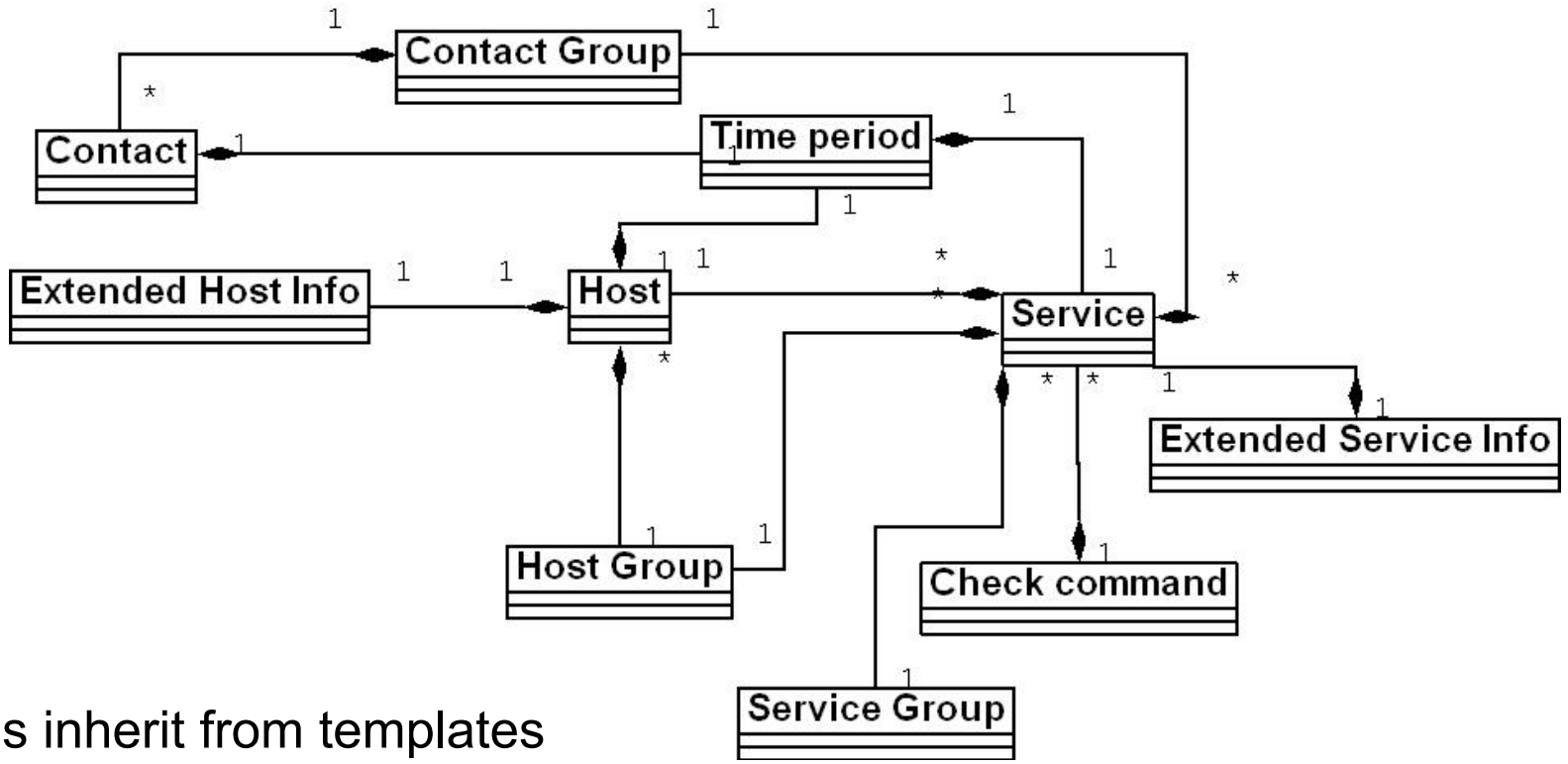
```
define service{
        name                            generic-service
        active_checks_enabled           1
        passive_checks_enabled          1
        parallelize_check               1
        obsess_over_service             0
        check_freshness                 1
        notifications_enabled           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{
        hostgroup_name                  servers
        service_description             PING
        check_command                   check-host-alive
        use                             generic-service
        max_check_attempts              5
        normal_check_interval           5
        notification_options            c,r,f
        notification_interval           0 ; set > 0 if you want to be renotified
}
```

c: Critical
r: Recovering
f: Flapping

Configuration Flow



Items inherit from templates

We start with a host

- Place multiple hosts in a group
- Define parents
- Add a service check to the group
- Add extended info, if any

OoB Notifications

A critical item to remember: an SMS or message system that is independent from your network.

- You can utilize a cell phone connected to the Nagios server
- You can use packages like:

  gnokii: http://www.gnokii.org/
  qpage: http://www.qpage.org/
  sendpage: http://www.sendpage.org/

References

- **Nagios web site**
  http://www.nagios.org/
- **Nagios plugins site**
  http://www.nagiosplugins.org/
- *Nagios System and Network Monitoring*, by Wolfgang Barth. Good book about Nagios.
- **Unofficial Nagios plugin site**
  http://nagios.exchange.org/
- **A Debian tutorial on Nagios**
  http://www.debianhelp.co.uk/nagios.htm
- **Commercial Nagios support**
  http://www.nagios.com/

Questions?

?