Monitoring With Nagios

DevOps Training

Why Monitor?

Monitoring Principles

Types of Monitoring

- Environmental
- Network Performance
- Application Performance
- Network Device Status
- Server / System Status

Monitoring Models

Polling

- Actively query devices to determine status
- Schedule queries to minimize time between a failure and you knowing about the failure

Listening

- Devices tell you when something is wrong

Hybrid

Thresholds

Levels of Severity

- Normal Operation
- Warning
- Critical
- Off-line

Intervals

How many times do we try before declaring a host or service “dead”?

How often do we re-check the dead service?How often do we check a normally-operating host or service?

How often do we send out notifications after a problem has occurred?

Notifications

Who gets notified?

How do they get notified?

– Pager / SMS
– Email
– Phone call

Escalation

– Send a message to somebody else if the problem isn't resolved

– Automatic submission to trouble ticket system

Nagios

An open source monitoring tool that actively monitors availability of devices and services:

Popular: One of the most used open source network monitoring software packages.
Fast: Uses CGI functionality written in C for faster response and scalability.
Scalable: Can support up to thousands of devices and services. Modular Cool-Looking Web Interface®

Architecture

Nagios has server-agent architecture.

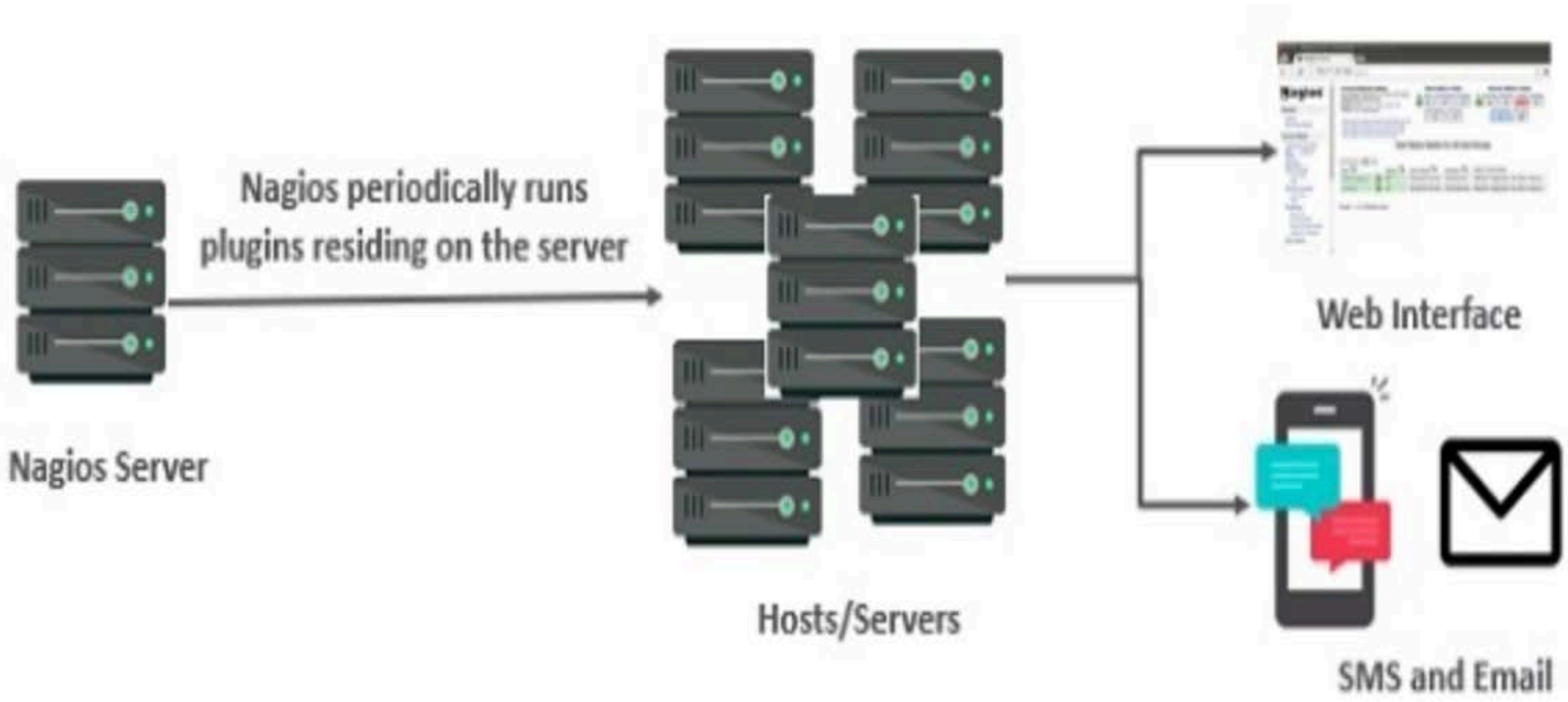
Nagios server is installed on the host and plugins are installed on the remote hosts/ servers which are to be monitored.

Nagios sends a signal through a process scheduler to run the plugins on the local/remote hosts/servers.

Plugins collect the data (CPU usage, memory usage etc.) and sends it back to the scheduler.

Then the process schedules send the notifications to the admin/s and updates Nagios GUI.

Architecture



Features

Type of availability is largely delegated to plug-ins:

The product's architecture is simple enough that writing new plugins is fairly easy in the language of your choice.

There are many, many, many plug-ins available.

Features

The Nagios package in Ubuntu comes with a number of pre-installed plugins:
apt.cfg breeze.cfg dhcp.cfg disk-smb.cfg disk.cfg dns.cfg dummy.cfg
flexlm.cfg fping.cfg ftp.cfg games.cfg hppjd.cfg http.cfg ifstatus.cfg
ldap.cfg load.cfg mail.cfg mrtg.cfg mysql.cfg netware.cfg news.cfg nt.cfg
ntp.cfg pgsql.cfg ping.cfg procs.cfg radius.cfg real.cfg rpc-nfs.cfg
snmp.cfg ssh.cfg
tcp_udp.cfg telnet.cfg users.cfg vsz.cfg
There are many more available (e.g.)...

http://sourceforge.net/projects/nagiosplugins

Features

Fast and Scalable : Compiled, binary CGIs and common plug-ins for faster performance.
Parallel checking and forking of checks to support large numbers of devices.
Uses “intelligent” checking capabilities.
Attempts to distribute the server load of running Nagios (for larger sites) and the load placed on devices being checked.
Configuration is done in simple, plain text files, that can contain much detail and are based on templates.
Nagios reads it's configuration from an entire directory. You decide how to define individual files.

Features

Topology Aware: To determine dependencies.
Differentiates between what is down vs. what is not available. This way it avoids running unnecessary checks. This is done using parent-child relationships between devices.

Notifications: How they are sent is 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.

Features

Service state:
When configuring a service you have the following notification options: d: DOWN: The service is down (not available)

u: UNREACHABLE: When the host is not visible

r: RECOVERY: (OK) Host is coming back up

f: FLAPPING: When a host first starts or stops or it's state is undetermined.

n: NONE: Don't send any notifications

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 (scripts, actions to take)

How checks works

Parameters: Set in /etc/nagios3/nagios.cfg:
Normal checking interval
Re-check interval
Maximum number of checks.
Period for each check
Services check(s) only happen when a node responds (ping check or “is alive = yes”):
Remember a node can be:
DOWN
UNREACHABLE

How checks works

In this manner it can take some time before a host changes its state to “down” as Nagios first does a service check and then a node check.
By default Nagios does a node check 3 times before it will change the nodes state to down.

You can, of course, change all this.
/etc/nagios3/nagios.cfg

Lots of configuration settings and combinations
Default settings have been tested for large install

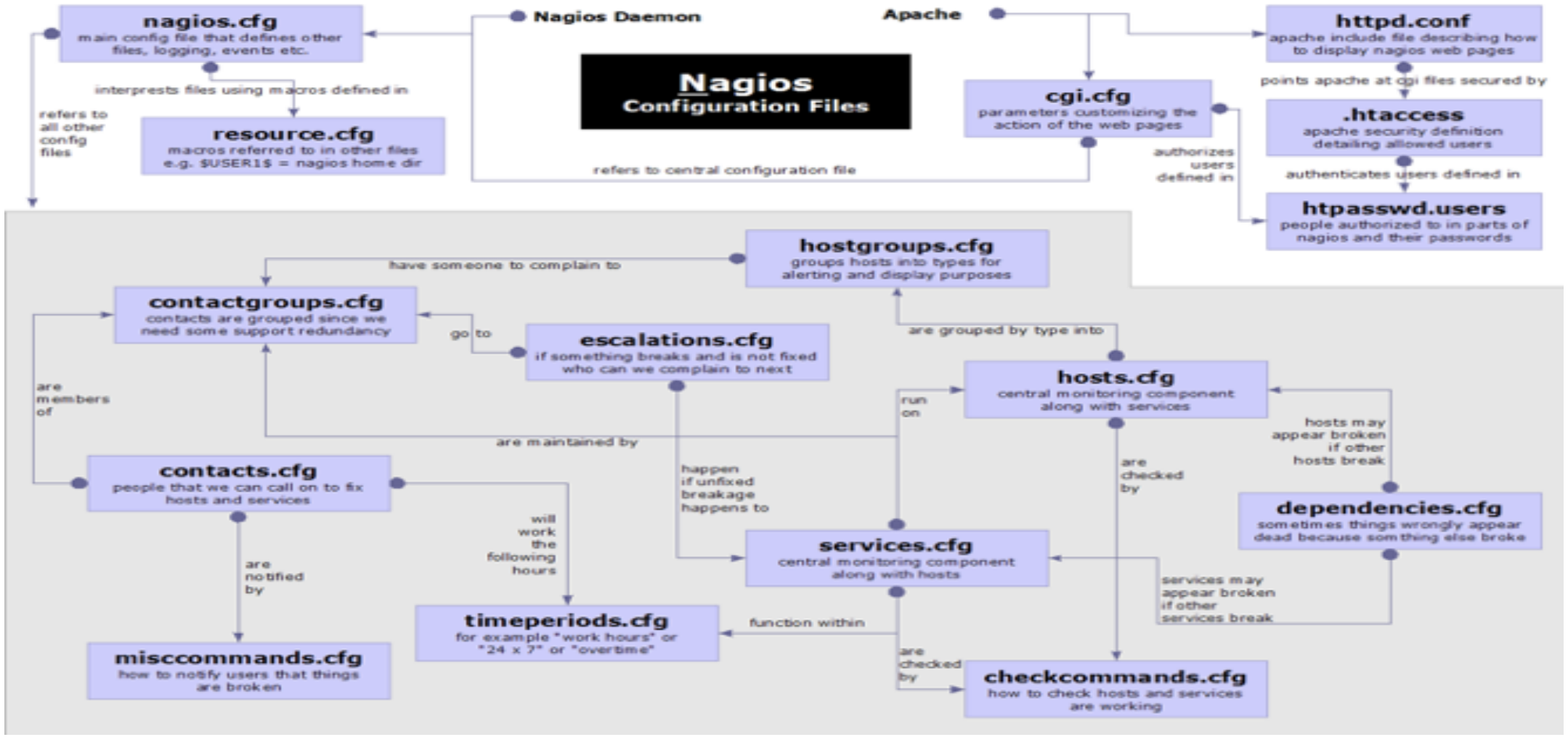
Concept of parents

Nodes can have parents.

For example, the parent of a PC connected to the switch mgmt-sw1 would be mgmt-sw1.

This allows us to specify the network dependencies that exist between machines, switches, routers, etc.
This avoids having Nagios send alarms when a parent does not respond.
Note: A node can have multiple parents.

Nagios Configuration files



Configuration files

Located in /etc/nagios3/ (in Ubuntu)

Important files include:

cgi.cfg : Controls the web interface and security options.

commands.cfg: The commands that Nagios uses for notifications (i.e. sending email)

nagios.cfg: Main configuration file.

conf.d/* : All other configuration goes here!

Configuration files

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/servers.cfg	Sample definition of PCs (hosts)
switches.cfg	Definitions of switches (hosts)
routers.cfg	Definitions of routers (hosts)

Main configuration file

Global settings

File: /etc/nagios2/nagios.cfg

Says where other configuration files are.
General Nagios behavior:
For large installations you should tune the installation via this file.
See: Tunning Nagios for Maximum Performance
http://nagios.sourceforce.net/docs/2_0/tuning.html

Time Periods

conf.d/timeperiods_nagios2.cfg: defines the base periods that control checks, notifications, etc.
Defaults: 24 x 7
Could adjust as needed, such as work week only.
Could adjust a new time period for “outside of regular hours”, etc.

Configuring Service/Host Checks

Define how you are going to test a service.

Located in /etc/nagios-plugins/config, then adjust in /etc/nagios3/conf.d/services_nagios2.cfg

```
# 'check-host-alive' command definition
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 2000.0,60% -c
5000.0,100% -p 1 -t 5
 }
```

Notification Commands

Allows you to utilize any command you wish. You can do this for generating tickets in RT:

```
# 'notify-by-email' command definition
define command{
        command_name    notify-by-email
        command_line    /usr/bin/printf "%b" "Service: $SERVICEDESC$\nHost: $HOSTNAME$\nIn: $HOSTALIAS$\nAddress: $HOSTADDRESS$
\nState: $SERVICESTATE$\nInfo: $SERVICEOUTPUT$\nDate: $SHORTDATETIME$" | /bin/mail -s '$NOTIFICATIONTYPE$: $HOSTNAME$/
$SERVICEDESC$ is $SERVICESTATE$' $CONTACTEMAIL$

        }
```

From: nagios@nms.localdomain
To: grupo-redes@localdomain
Subject: Host DOWN alert for switch1!
Date: Thu, 29 Jun 2006 15:13:30 -0700

Host: switch1
In: Core_Switches
State: DOWN
Address: 111.222.333.444
Date/Time: 06-29-2006 15:13:30
Info: CRITICAL - Plugin timed out after 6 seconds

THANK YOU!

training@laksans.com