Edureka Devops

Module-Nagios
Monitoring with Nagios

Nagios – What it is?

→A widely-used monitoring tool for trouble-shooting

- » Simple and open source
- » Network, system, service levels

→ With a notification system to inform administrators when something goes wrong

→ Nagios provides support to administrator(s) for detecting problems

- » Mail server failure
- » Hard drive overload
- » Network outage

Nagios -

→Colored area concept

» Green/Yellow/Red (Ok/Warning/Critical)

→ No performance analysis or display

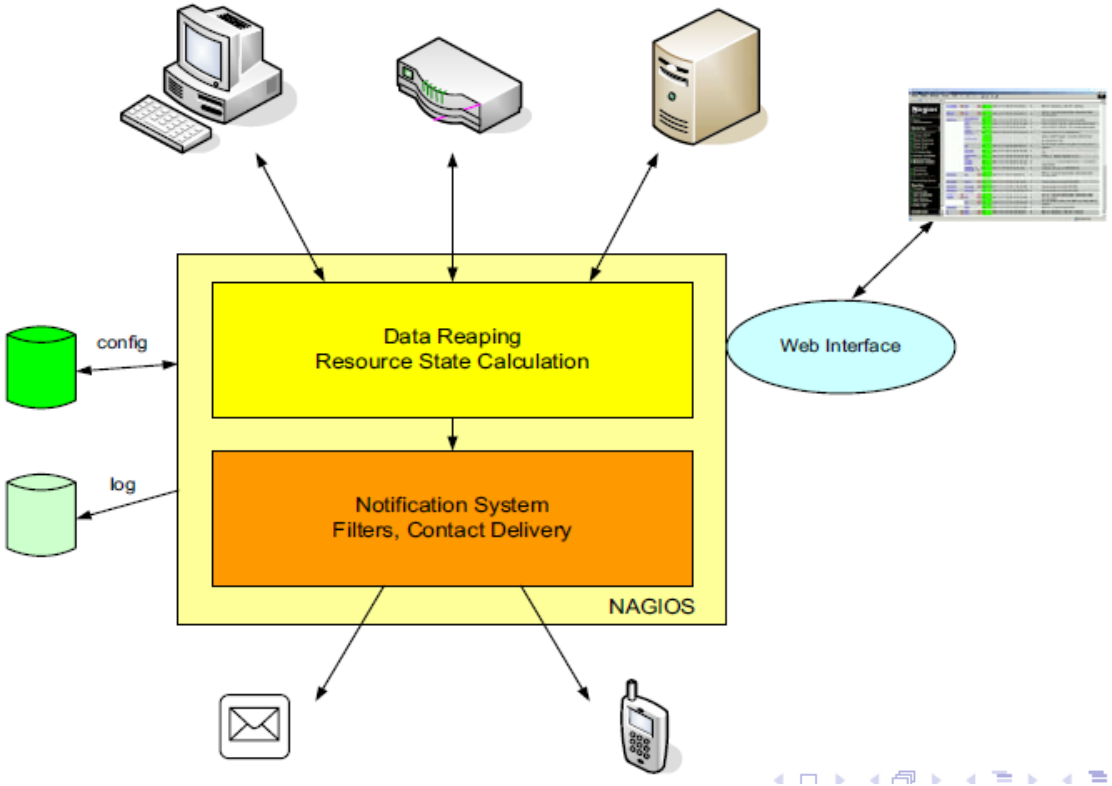
→ Checks using **external** commands (plugins)

→ Various possibilities for **remote** checks

→ Possibility for **passive checks** (from managed resources)

→ Web interface + notifications

Nagios Functional Architecture



Nagios

→Data reaping + notification system= *nagios processes*

» Can be run as a service (rc*X*.d, soft runlevel)

→ Web interface

» External web server (Apache)

» Bunch of cgi scripts (part of Nagios)

→ Configuration

» Simple text files

» Or a postgres database

→ Logs (local files)

→ Named pipe (unix domain socket) to enable nagios to **receive** commands (from cgi, passive asynchronous events)

Nagios

→Service

- » Service delivered by a software
- » Percentage of free space on a partition
- » Bandwith usage on a network interface ...

→ Service check

- » Provides state information on a service
- » Returns a value: OK, WARNING, CRITICAL (exit status 0, 1, 2), UNKNOWN (exit status 3, due to time out or plugin runtime trouble) to reflect the Nagios view about this service
- » Can be local (OS calls) or remote (ICMP, NRPE, SNMP ...)
- » Is implemented by a plugin (external command/script)

Nagios

→Service states are the mirror of what nagios observes

→ States: OK, WARNING, CRITICAL, UNKNOWN

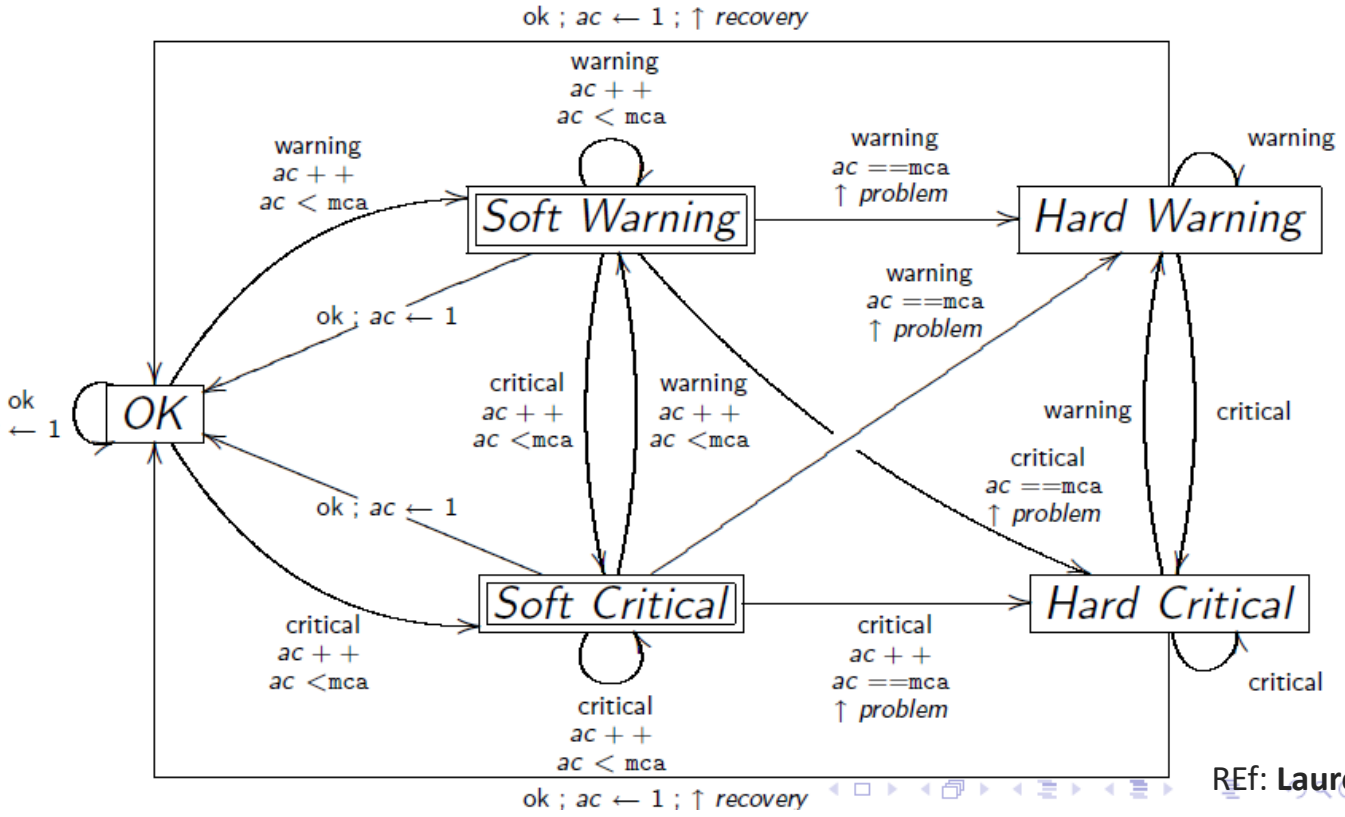
→ Transitions from one state to another one based on results provided by checks

→ Critical and warning states are shadowed by related *soft* states

- » A service goes first to a *soft* state
- » Attempt count mecanism to reach a definitive *hard* state

→ User notification can only occur when *hard* states are reached

Nagios Service State Diagram



REf: **Laurent Andrey, R´emi Badonnel**

Course Url

Nagios Config...

→Object-oriented representation

» A nagios object describes a specific unit: a service, an host, a contact, a contactgroup ... with attributes and values

» kind of inheritance mechanisms, dependencies amongst objets

→ Set of configuration files

» Main file: nagios.cfg (ref. to other cfg. files)

» 1 file per object type: services.cfg, hosts.cfg, contacts.cfg,

» checkcommands.cfg, misccommands.cfg, timeperiods.cfg ...

Nagios

→A service **have** to be linked to an *host*

→ Only *UP* and *DOWN* states

→ User notification (problem, recovery)

→ Same external checks than services

- » UP = (WARNING or OK), DOWN = CRITICAL
- » Typically: ICMP-based checks

→ No active checks if related services are **OK**

→ contact, contactgroup: to specify who to notify, how to notify and when to notify

→ Used in host and service definitions

→ command: to execute check plugins or to send user notifications

→ Links Nagios attributes found in definitions (ex: host name) to plugin parameters

→ Already provided for most common plugins (see checkcommands.cfg file)

→ Basic wrapping to send emails (misccommands.cfg file)

Nagios – Examples (host.cfg)

```
d e f i n e host {
host name                       w e b1
a l i a s                       web1  l i n u x machine
address                         192.168.33.80
check                           command check–host–a l i v e
max                             check attempts 3
check p e r i o d               24 x7
n o t i f i c a t i o n i n t e r v a l   180 # 3 hours
n o t i f i c a t i o n p e r i o d       24 x7
n o t i f i c a t i o n o p t i o n s     d , r , u# down , recovery , unreachable
contact groups                  a d m i n i s t r a t o r s
}
```

Nagios Example (service.cfg)

```
define service {
host name                       web1
service_description             http service
check_command                   check_http
max_check_attempts              3
normal check interval           5
retry check interval            1
check period                    24x7
notification interval           180
notification period             24x7
notification options            w, c, r, f, u
# warning, critical, recovery, flapping, unreachable
contact groups                  administrators
}
```

NAgios Examples (checkcommands.cfg)

```
d e f i n e command{
command name check –host –a l i v e
command line $USER1$/ check icmp  –H $HOSTADDRESS$
}
d e f i n e command{
command name check http
command line $USER1$/ check http  –H $HOSTADDRESS$
}
```

NAgios Examples (contacts.cfg)

```
define contact {
contact name                    sriram
alias                           The Boss
service notification period     24 x7
host notification period        24 x7
service notification options    w, u , c , r
host notification options       d , r
service notification commands   notify –by –email
host notification commands      host –n otify –by –email
email                           sriram@nowhere.com
}
define contactgroup {
contactgroup name               administrators
alias                           group of administrators
members Andrey, Sriram, Raaj
}
```

NAgios Commands

→Local Command (PS, dfm uptime)

» check_disk -w 30% -c 15% -p /var

» check_load -w 2.0,1.0,0.5 -c 4.0,2.0,1.0

» check_procs -w 150 -c 250 --metric=PROCS

→Network Service Checking – Done from Nagios Machine

» check_icmp -H 1.14.1.2 -w 100.0,20% -c 200.0,40%

» check_tcp -H boston.loria.fr -p 7000

» check_ftp -H ftp.inria.fr -p 21 -e 220

» check_http -H http://www.esial.uhp-nancy.fr

» check_smtp, check_imap, check_pop

NAgios Other checks

→via check_by_ssh

→using NRPE (check_nrpe/nrpe daemon)

→SNMP (using check_snmp plugin)

→NSCA (Nagios Service Check Acceptor) (Acceptor/sendor Pair)

Thank you!