# Gestión de Redes: Nagios

## Network Startup Resource Center
## www.nsrc.org

# Introducción

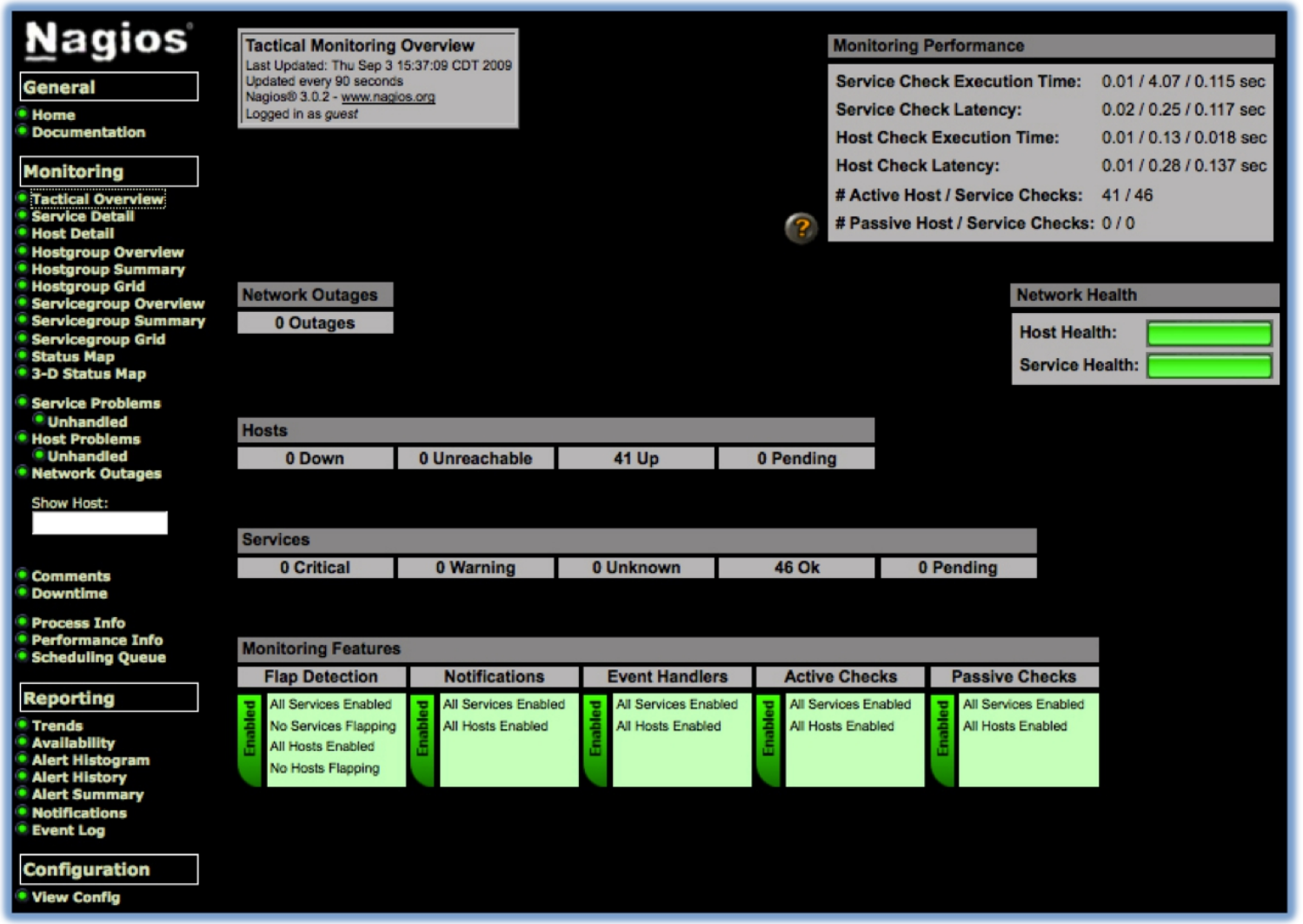
- Probablemente el software libre de monitorización más utilizado
- Interfaz web para visualizar el estado, revisar la historia de eventos, planificar bajas por mantenimiento
- Envía alarmas por e-mail. Puede configurarse para usar otros mecanismos (ej. SMS)

# Introducción

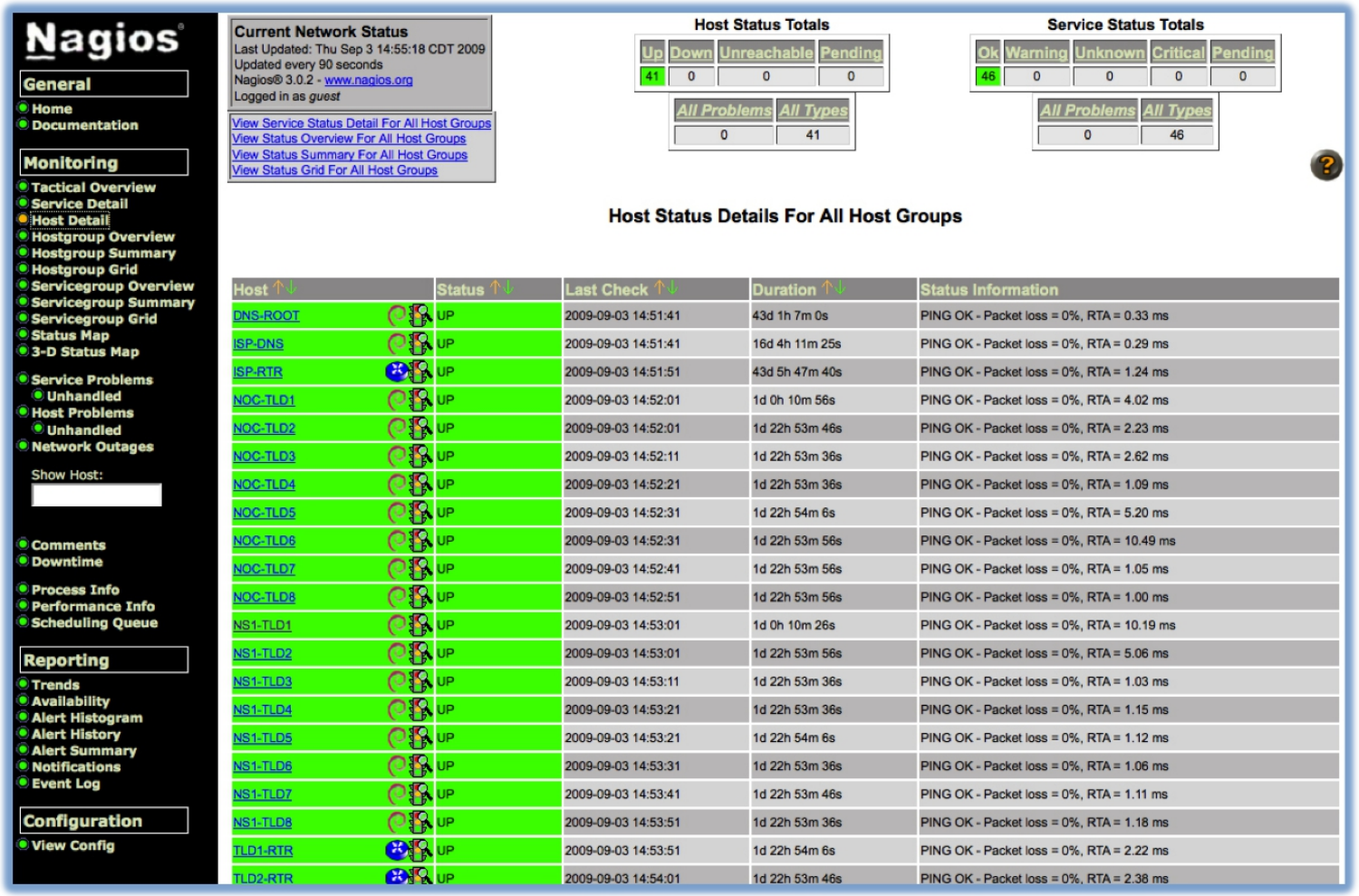
Nagios monitoriza activamente la disponibilidad

- de Nodos (dispositivos)
- y Servicios

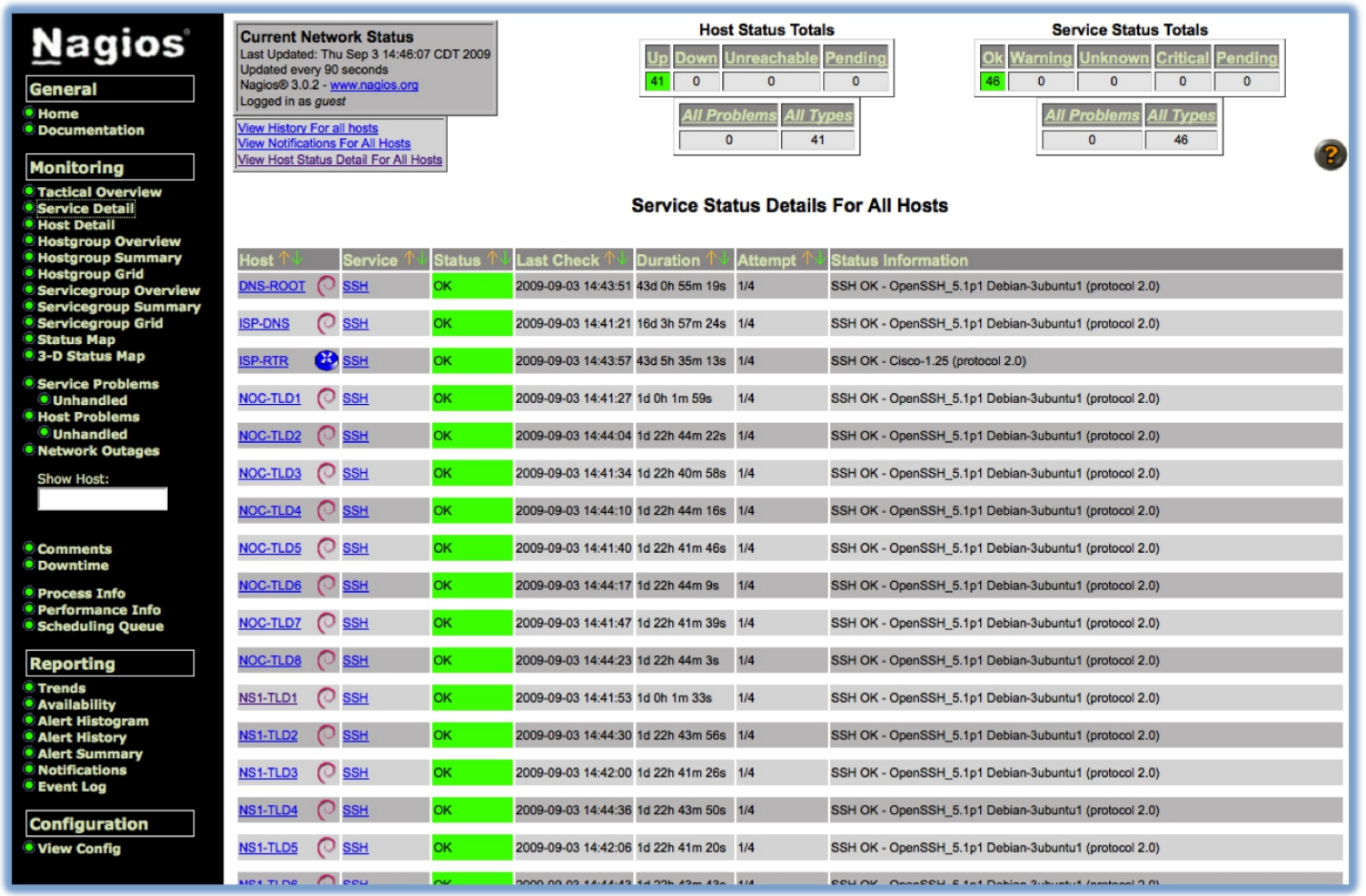
# Nagios: Vista General



# Vista Detalles Hosts



# Vista Detalles Servicios



# Funcionalidades

Utiliza información topológica para determinar las dependencias.

- Diferencia entre lo que está *caído* vs. lo que está *inalcanzable*. Evita hacer comprobaciones y enviar alarmas innecesarias

Permite definir cómo enviar alarmas basado en una combinación de:

- Contactos y listas de contactos
- Dispositivos y sus grupos
- Servicios y sus grupos
- Horarios definidos por personas y grupos.
- El estado de un servicio.

# Plugins

Los plugins se usan par verificar servicios y nodos:

- La arquitectura de Nagios hace suficientemente simple el escribir nuevos plugins en el lenguaje de su preferencia.
- Existen **muchos, muchos** plugins disponibles (miles)..

✓http://exchange.nagios.org/

✓http://nagiosplugins.org/

# Plugins Pre-instalados en Ubuntu

## /usr/lib/nagios/plugins

```
2. nsrc@s1: ~ (ssh)
nsrc@s1:~$ ls /usr/lib/nagios/plugins
check_apt       check_disk      check_hpjd          check_jabber    check_mysql       check_ntp_time  check_real      check_ssh     check_wave
check_breeze    check_disk_smb  check_http          check_ldap      check_mysql_query check_nwstat    check_rpc       check_ssmtp   negate
check_by_ssh    check_dns       check_icmp          check_ldaps     check_nagios      check_oracle    check_rta_multi check_swap    urlize
check_clamd     check_dummy     check_ide_smart     check_load      check_nntp        check_overcr    check_sensors   check_tcp     utils.pm
check_cluster   check_file_age  check_ifoperstatus  check_log       check_nntps       check_pgsql     check_simap     check_time    utils.sh
check_dbi       check_flexlm    check_ifstatus      check_mailq     check_nt          check_ping      check_smtp      check_udp
check_dhcp      check_ftp       check_imap          check_mrtg      check_ntp         check_pop       check_snmp      check_ups
check_dig       check_host      check_ircd          check_mrtgtraf  check_ntp_peer    check_procs     check_spop      check_users
nsrc@s1:~$
```

## /etc/nagios-plugins/config

```
2. nsrc@s1: ~ (ssh)
nsrc@s1:~$ ls /etc/nagios-plugins/config/
apt.cfg     disk-smb.cfg  fping.cfg  http.cfg      mail.cfg   netware.cfg  pgsql.cfg   real.cfg     tcp_udp.cfg
breeze.cfg  dns.cfg       ftp.cfg    ifstatus.cfg  mailq.cfg  news.cfg     ping.cfg    rpc-nfs.cfg  telnet.cfg
dhcp.cfg    dummy.cfg     games.cfg  ldap.cfg      mrtg.cfg   nt.cfg       procs.cfg   snmp.cfg     users.cfg
disk.cfg    flexlm.cfg    hppjd.cfg  load.cfg      mysql.cfg  ntp.cfg      radius.cfg  ssh.cfg
nsrc@s1:~$
```

# Cómo funcionan los plugins

- Periódicamente Nagios ejecuta un plugin para verificar el estado de cada servicio. Las posibles respuestas son:
  - OK
  - WARNING
  - CRITICAL
  - UNKNOWN
- Si un servicio no está OK, entra en un estado de error “soft”. Después de un número de reintentos (3 por defecto), entra en un estado de error “hard”. En este momento se envía una alarma.
- Es posible también activar manejadores de eventos (event handlers) externos basándose en transiciones de estados

# Cómo funcionan los plugins (Cont.)

## Parámetros

- Intervalo de chequeo normal
- Intervalo de reintento (i.e. cuando no es OK)
- Número máximo de reintentos
- Ventana de ejecución de los chequeos
- Ventana para el envío de las alarmas

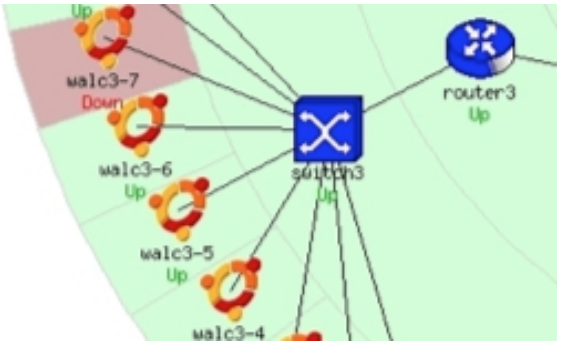
## Programación de tareas

- Nagios distribuye sus chequeos a lo largo del intervalo para balancear la carga
- La interfaz web muestra la hora del próximo chequeo

# Jerarquía: El concepto de *padres*

**Los nodos pueden tener *padres*:**

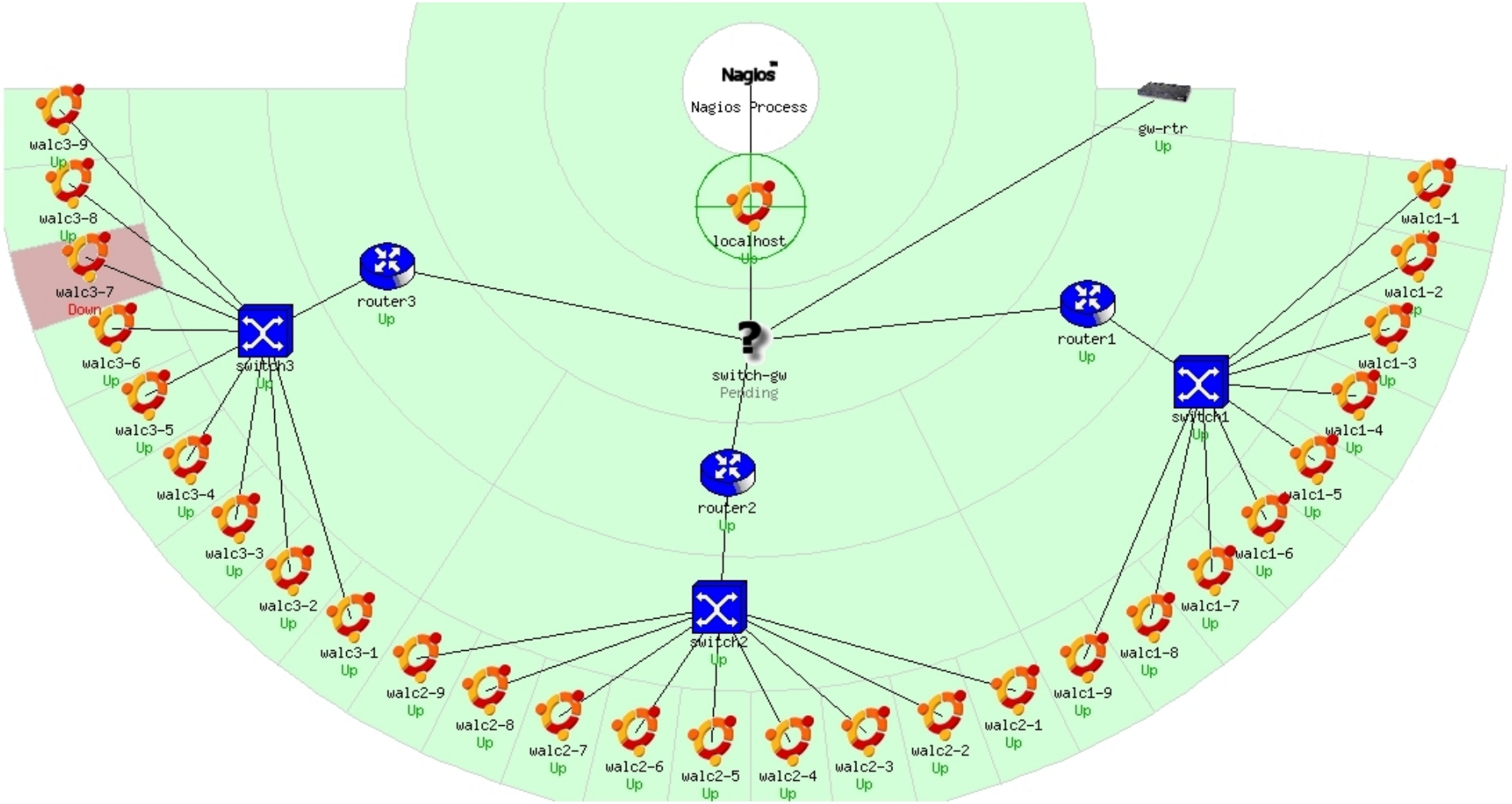
- El *padre de un* PC conectado a un switch sería el switch.
- Nos permite especificar las dependecias entre dispositivos.
- Evita enviar alarmas cuando los padres *no responden.*
- Un nodo puede tener varios *padres* (dual homed).



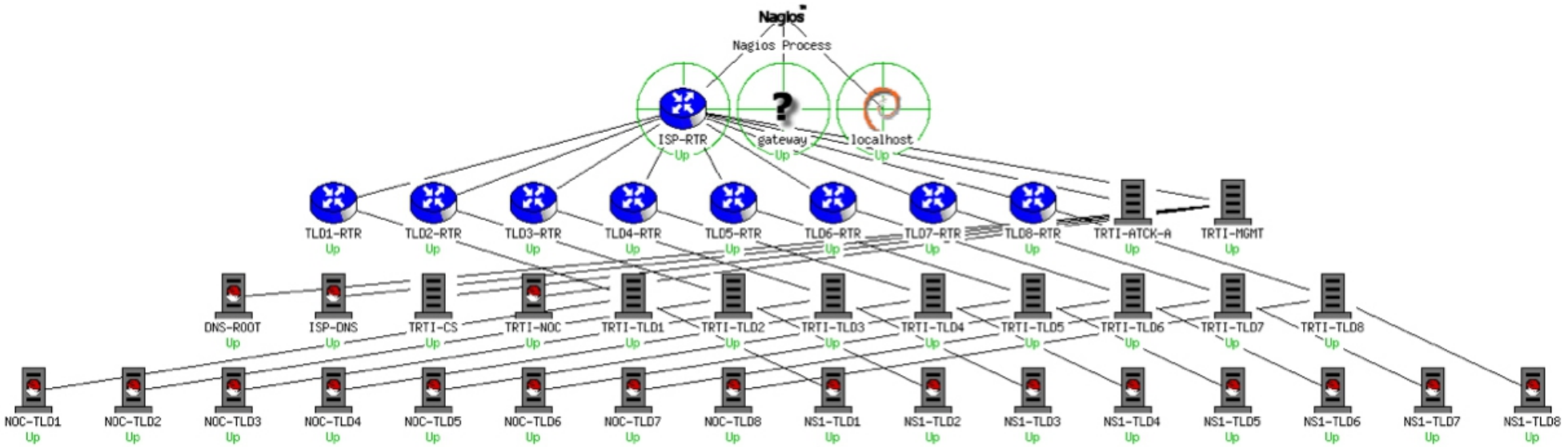
# Punto de Vista de Red

- El lugar donde se coloque el servidor Nagios determinará su punto de vista de la red.
- El servidor Nagios se convierte en la *raíz* del árbol de dependencias jerárquicas

# Punto de Vista de Red



# Vista Red en Árbol Colapsado



# Demostración de Nagios

http://noc.ws.nsrc.org/nagios3/

nagiosadmin: lab_password

# Instalación

**En Debian/Ubuntu**

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

Directorios clave

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

La interfaz web de Nagios está en:

http://pcN.ws.nsrc.org/nagios3/

# Configuración de nodos y servicios

**Basado en plantillas**

- Esto ahorra mucho tiempo ya que evita la repetición

**Hay plantillas por defecto con parámetros por defecto para:**

- *Nodo genérico* (generic-host_nagios2.cfg)
- *Servicio genérico* (generic-service_nagios2.cfg)

- Los parámetros individuales se pueden sobreponer
- Los valores por defecto son razonables

# Configuración

- La configuración se define en archivos de texto

  /etc/nagios3/conf.d/*.cfg

- Detalles en http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html

- La configuración por defecto se distribuye entre varios archivos por tipo de objeto, pero en realidad se pueden organizar a su gusto

- Siempre verifique antes de reiniciar Nagios, de lo contrario su sistema se caerá!

```
nagios3 –v /etc/nagios3/nagios.cfg
```

# Monitorizar un nodo

**pcs.cfg**

```
define host {
   host_name pc1
   alias     pc1 in group 1
   address   pc1.ws.nsrc.org
   use       generic-host
}
```

Heredar parámetros de esta plantilla

- Ésta es una configuración mínima
- Simplemente está haciendo ping al nodo; Nagios le advertirá que no está monitorizando ningún servicio
- El nombre del archive puede ser cualquiera acabado en **`.cfg`**
- Organice sus nodos como le convenga, por ejemplo, nodos relacionados en el mismo archivo

# Plantilla genérica de nodo

## generic-host_nagios2.cfg

```
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 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   ; DON'T REGISTER THIS DEFINITION -
                                       ; IT'S NOT A REAL HOST, JUST A TEMPLATE!
}
```

# Sobreponer valores por defecto

Los valores heredados se pueden sobreponer en el nodo

**pcs.cfg**

```
define host {
   host_name                    pc1
   alias                        pc1 in group 1
   address                      pc1.ws.nsrc.org
   use                          generic-host
   notification_interval        120
   contact_groups               admins,managers
}
```

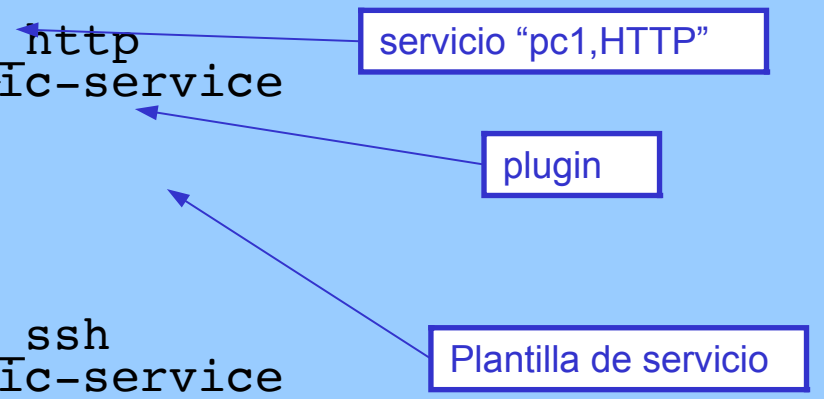
# Definición de servicios (forma directo)

**pcs.cfg**

```
define host {
    host_name       pc1
    alias           pc1 in group 1
    address         pc1.ws.nsrc.org
    use             generic-host
}

define service {
    host_name               pc1
    service_description     HTTP
    check_command           check_http
    use                     generic-service
}

define service {
    host_name               pc1
    service_description     SSH
    check_command           check_ssh
    use                     generic-service
}
```



# Comprobaciones de servicio

- La combinación de nodo+servicio es un identificador único para el chequeo, ej:
- "pc1,HTTP"
- "pc1,SSH"
- "pc2,HTTP"
- "pc2,SSH"
- *check_command* hace referencia al plugin
- *service template* causa que se hereden los parámetros sobre frecuencia de comprobación, y a quién y cuándo enviar las alarmas

# Plantilla genérica de servicio

```
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
        failure_prediction_enabled      1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
          notification_interval           0
          is_volatile                     0
          check_period                    24x7
          normal_check_interval           5
          retry_check_interval            1
          max_check_attempts              4
          notification_period             24x7
          notification_options            w,u,c,r
          contact_groups                  admins
        register                        0    ; DONT REGISTER THIS DEFINITION
        }
```

**generic-service_nagios2.cfg**

(comments have been removed)

# Sobreponiendo valores por defecto

De nuevo, los valores por defecto se pueden sobreponer para cada servicio

**services_nagios2.cfg**

```
define service {
   host_name               pc1
   service_description     HTTP
   check_command           check_http
   use                     generic-service
   contact_groups          admins,managers
   max_check_attempts      3
}
```

# Chequeos de servicio repetidos

- Frecuentemente monitorizamos el mismo servicio en múltiples nodos
- Para evitar la duplicación, es más conveniente definir un *service check* para todos los nodos en un *hostgroup*

# Crear grupos de nodos (hostgroups)

**hostgroups_nagios2.cfg**

```
define hostgroup {
   hostgroup_name   http-servers
   alias            HTTP servers
   members          pc1,pc2
}

define hostgroup {
   hostgroup_name   ssh-servers
   alias            SSH servers
   members          pc1,pc2
}
```

# Monitorizando servicios en hostgroups

```
define service {
    hostgroup_name          http-servers
    service_description     HTTP
    check_command           check_http
    use                     generic-service
}

define service {
    hostgroup_name          ssh-servers
    service_description     SSH
    check_command           check_ssh
    use                     generic-service
}
```

**services_nagios2.cfg**

*si el hostgroup "http-servers" contiene a pc1 y pc2 entonces Nagios crea chequeos de HTTP para cada nodo. Los chequeos de servicio se llaman "pc1,HTTP" y "pc2,HTTP"*

# Vista alternativa

- “Este hostgroup contiene a estas PCs”

**or:**

- “Esta PC pertenece a estos hostgroups”

- No es necesaria la sección “members” en el archivo de hostgroups

# Membresía de grupo alternativa

**pcs.cfg**

```
define host {
   host_name     pc1
   alias         pc1 in group 1
   address       pc1.ws.nsrc.org
   use           generic-host
   hostgroups    ssh-servers,http-servers
}

define host {
   host_name     pc2
   alias         pc2 in group 1
   address       pc2.ws.nsrc.org
   use           generic-host
   hostgroups    ssh-servers,http-servers
}
```

Nodos y servicios convenientemente definidos en el mismo sitio

# Otros usos de los hostgroups

Elegir iconos para el mapa de estado

**pcs.cfg**

```
define host {
   host_name    pc1
   alias        pc1 in group 1
   address      pc1.ws.nsrc.org
   use          generic-host
   hostgroups   ssh-servers,http-servers,debian-servers
}
```

**extinfo_nagios2.cfg**

```
define hostextinfo {
   hostgroup_name    debian-servers
   notes             Debian GNU/Linux servers
   icon_image        base/debian.png
   statusmap_image   base/debian.gd2
}
```

# Opcional: Servicegroups

- Los servicios se pueden agrupar usando un "servicegroup"
- De manera que los servicios relacionados o dependientes se puedan ver juntos en la interfaz web
- Los servicios deben estar previamente definidos

**servicegroups.cfg**

```
define servicegroup {
   servicegroup_name   mail-services
   alias          Services comprising the mail platform
   members        web1,HTTP,web2,HTTP,mail1,IMAP,db1,MYSQL
}
```

# Configurar la topología

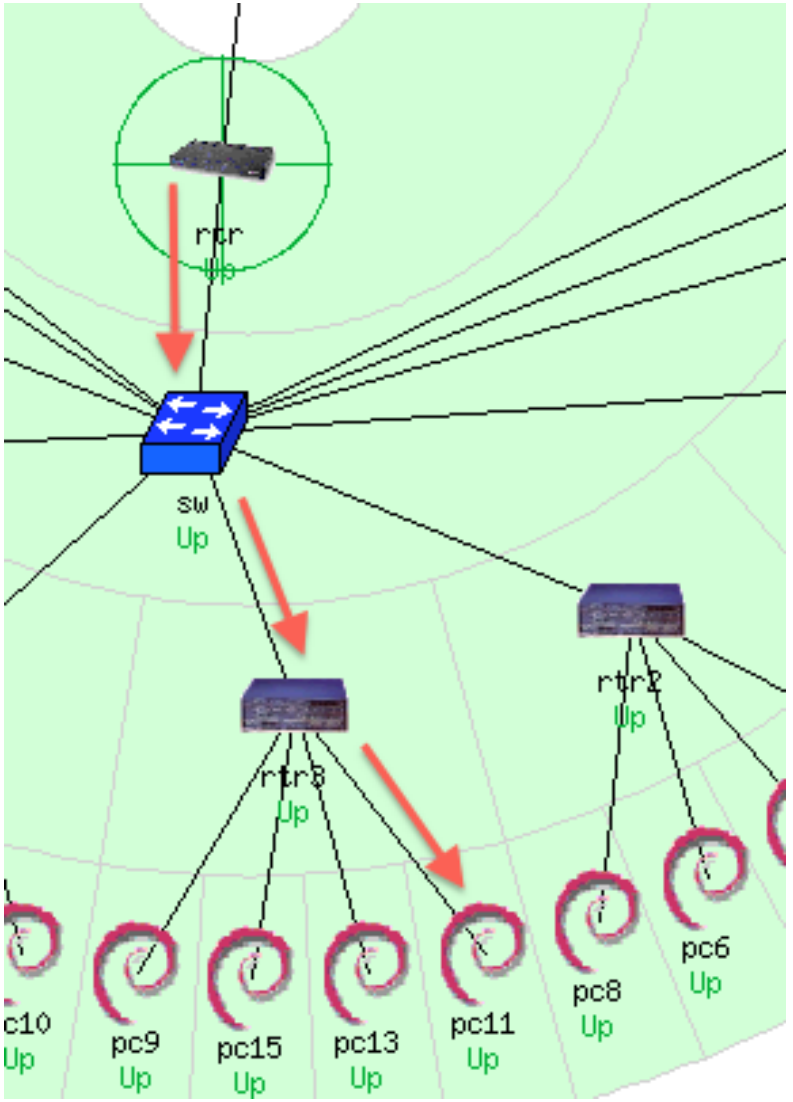
**pcs.cfg**

```
define host {
   host_name    pc1
   alias        pc1 in group 1
   address      pc1.ws.nsrc.org
   use          generic-host
   parents      rtr1
}
```

Nodo padre

- Esto significa “pc1 está del otro lado de rtr1”
- Si rtr1 cae, pc1 se marca como “unreachable” en lugar de “down”
- Evita una cascada de alarmas si cae rtr1
- También permite a Nagios dibujar el mapa

# Otra vista de la configuración




**RTR**

```
define host {
    use                 generic-host
    host_name           rtr
    alias               Gateway Router
    address             10.10.0.254     }
```

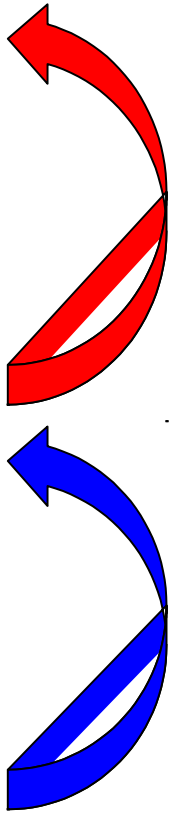
**SW**

```
define host {
    use                 generic-host
    host_name           sw
    alias               Backbone Switch
    address             10.10.0.253
    parents             rtr     }
```

**RTR3**

```
define host {
    use                 generic-host
    host_name           rtr3
    alias               router 3
    address             10.10.3.254
    parents             sw      }
```

**PC11…**



# Notificaciones fuera de línea (out of band)

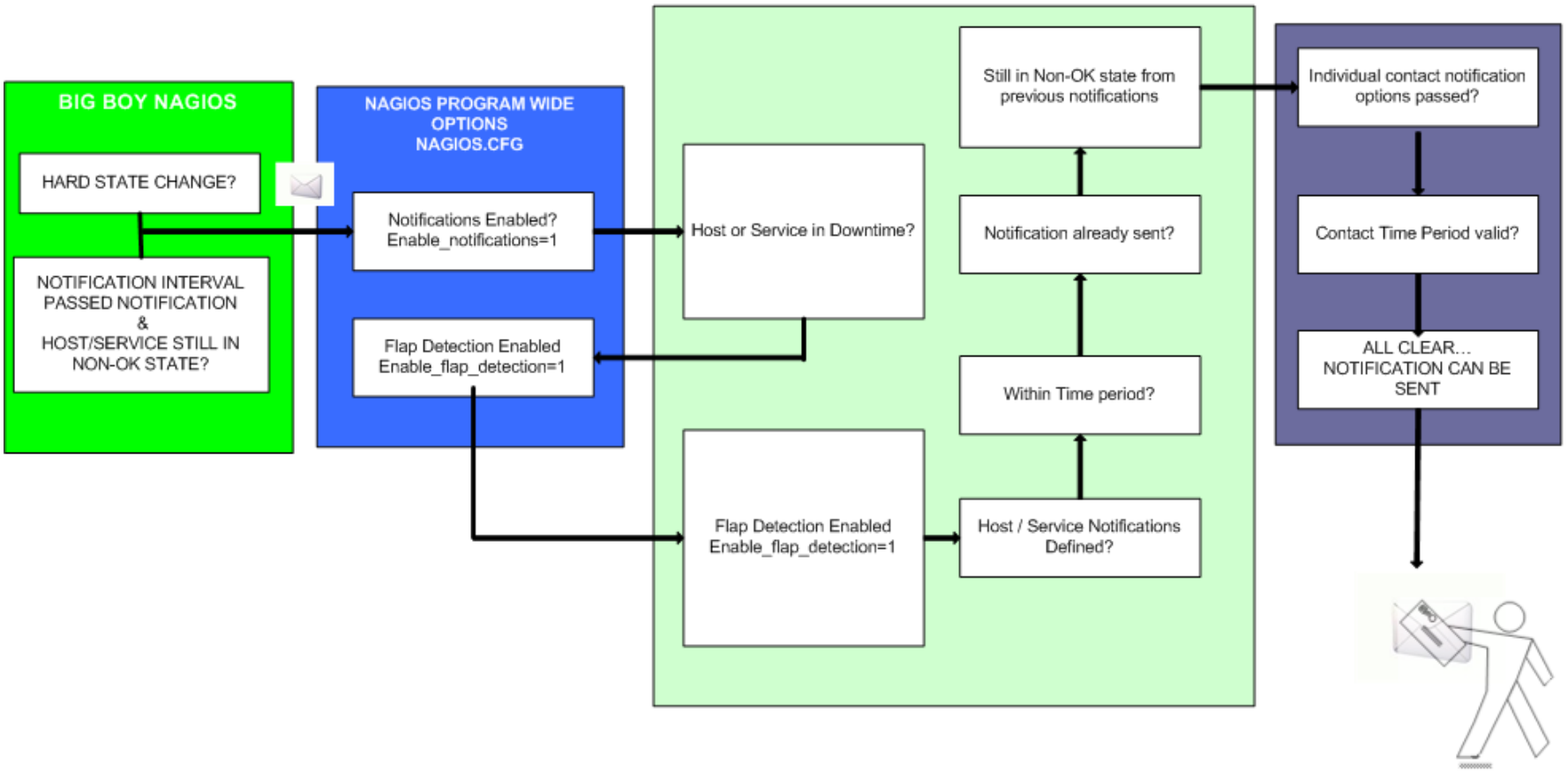
Una cosa crítica que recordar: un sistema de mensaje o SMS que no dependa de su red.

- Puede usar un teléfono celular conectado al servidor Nagios, o un dispositivo USB con tarjeta SIM
- Puede usar herramientas como:

**gammu**: http://wammu.eu/
**gnokii**: http://www.gnokii.org/
**sms-tools**: http://smstools3.kekekasvi.com/

Yo uso una Raspberry Pi con Kannel: http://www.kannel.org/

# NAGIOS - NOTIFICATION FLOW DIAGRAM




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

CONTACT GETS THE NOTIFICATION MESSAGE

# Referencias

- **Sitio web de Nagios**
  http://www.nagios.org/
- **Sitio web de plugins para Nagios**
  http://www.nagiosplugins.org/
- *Nagios System and Network Monitoring*, by Wolfgang Barth. Buen libro sobre Nagios.
- **Sitio web de plugins (no-oficial)**
  http://nagios.exchange.org/
- **Un tutorial de Nagios para Debian**
  http://www.debianhelp.co.uk/nagios.htm
- **Consultoría comercial de Nagios**
  http://www.nagios.com/

# Detalles Adicionales

Algunas slides adicionales que puede encontrar útiles o informativas…

# Más Funcionalidades

- Permite confirmer un evento.
  - Un usuario puede añadir comentarios a través de la interfaz gráfica.
- Se pueden definer períodos de mantenimiento.
  - Por dispositivo o grupo de dispositivos
- Mantiene estadísticas de disponibilidad y genera reportes
- Puede detector “flapeo” (cambios rápidos) y suprimir notificaciones adicionales.
- Permite varios métodos de notificación:
  - e-mail, buscapersonas, SMS, winpopup, audio, etc...
- Permite definer niveles de notificación para el escalado

# Opciones de Notificación de Nodo

**Estado Nodo:**

Cuando se configure un nodo puedes ser notificado en las siguientes condiciones:

- **–d:** DOWN
- **–u:** UNREACHABLE
- **–r:** RECOVERY
- **–f:** FLAPPING (start/end)
- **–s:** SCHEDULED DOWNTIME (start/end)
- **–n:** NONE

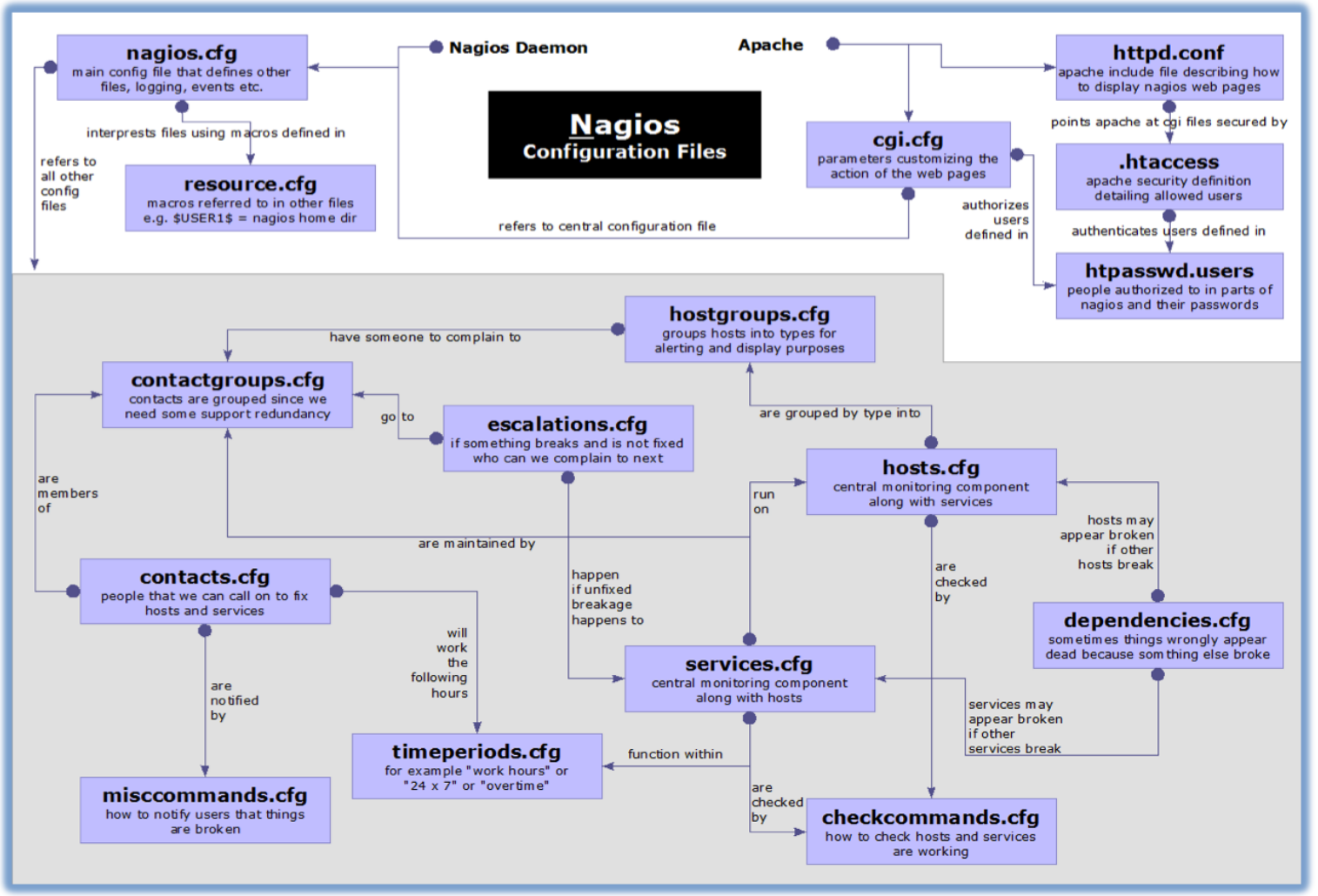
# Opciones de Notificación de Servicio

**Estado Servicio:**

Cuando se configure un servicio puedes ser notificado en las siguientes condiciones:

- **w:** WARNING
- **c:** CRITICAL
- **u:** UNKNOWN
- **r:** RECOVERY
- **f:** FLAPPING (start/end)
- **s:** SCHEDULED DOWNTIME (start/end)
- **n:** NONE

# Ficheros de Configuración



# Ficheros de Configuración Debian/Ubuntu

**Situados en /etc/nagios3/**

Los siguientes con ficheros importantes:

- nagios.cfg — Fichero de configuración principal.
- cgi.cfg — Controla la interfaz web y las opciones de seguridad.
- commands.cfg — Los comandos que Nagios usa para las notificaciones.
- conf.d/* — El resto de la configuración está aquí!

# Más ficheros de configuración

**Dentro de conf.d/***

- contacts_nagios2.cfg — usuarios y grupos
- extinfo_nagios2.cfg — hace la IU bonita
- generic-host_nagios2.cfg — plantilla de host por defecto
- generic-service_nagios2.cfg — plantilla de servicio por defecto
- host-gateway_nagios3.cfg — definición de router de salida
- hostgroups_nagios2.cfg — grupos de nodos
- localhost_nagios2.cfg — definición de host en nagios
- services_nagios2.cfg — qué servicios conmprobar
- timeperiods_nagios2.cfg — cuándo comprobar, a quién notificar

# Más ficheros de configuración

**Dentro de conf.d/ algunos otros posibles ficheros de configuración:**

- servicegroups.cfg — Grupos de nodos y servicios
- pcs.cfg — Definición de ejemplo de PCs (hosts)
- switches.cfg — Definiciones de switches (hosts)
- routers.cfg — Definiciones de routers (hosts)

# Detalles Configuración Principal

**Parámetros globales**

**Fichero:** `/etc/nagios3/nagios.cfg`

- Dice dónde están otros ficheros de configuración.
- Comportamiento general de Nagios:

– Para grandes instalaciones debes afinar su configuración usando este fichero.

– Ver: *Tunning Nagios for Maximum Performance*
**http://nagios.sourceforge.net/docs/3_0/tuning.html**

# Configuración CGI

## /etc/nagios3/cgi.cfg

- Se puede cambiar el directorio CGI si se quiere
- Autenticación y autorización en Nagios usa:
- Activar autenticación usando el mecanismo .htpasswd de Apache, o usando RADIUS o LDAP.
- Se pueden asignar derechos a los usuarios usando las siguientes variables:

- authorized_for_system_information
- authorized_for_configuration_information
- authorized_for_system_commands
- authorized_for_all_services
- authorized_for_all_hosts
- authorized_for_all_service_commands
- authorized_for_all_host_commands

# Perídos de Tiempo (ventanas)

Define los períodos de tiempo en los que se permiten los chequeos, notificaciones, etc.

- Por defecto: 24 x 7
- Ajustar según se necesite, por ejemplo solamente horas de trabajo entre semana.
- Configurar un Nuevo period de tiempo para “fuera de horario regular”, etc.

```
# '24x7'
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
        }
```

# Configurar Chequeos de Service/Host

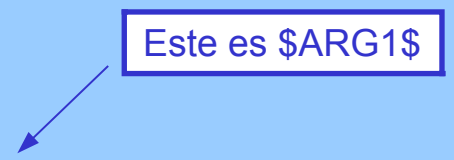
```
define command {
  command_name  check_ssh
  command_line  /usr/lib/nagios/plugins/check_ssh '$HOSTADDRESS$'
}

define command {
  command_name  check_ssh_port
  command_line  /usr/lib/nagios/plugins/check_ssh -p '$ARG1$' '$HOSTADDRESS$'
}
```

**/etc/nagios-plugins/config/ssh.cfg**

- Como se ve, el mismo plugin se puede usar de distintas maneras (“commands”)
- Comando y argumentos se separan con un símbolo de exclamación (!)
- ejemplo, chequear SSH en un puerto no estándar, se puede hacer como:

```
define service {
   hostgroup_name          ssh-servers-2222
   service_description     SSH-2222
   check_command           check_ssh_port!2222
   use                     generic-service
}
```



# Comandos de Notificación

Usar cualquier comando que se quiera!

Podemos usar esto para generar tickets en 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:      router_group@localdomain
Subject: Host DOWN alert for TLD1-RTR!
Date:    Thu, 29 Jun 2006 15:13:30 -0700


Host: gw
In: Core_Routers
State: DOWN
Address: 192.0.2.100
Date/Time: 06-29-2006 15:13:30
Info: CRITICAL - Plugin timed out after 6 seconds
```

# Configuración de Grupo de Servicio

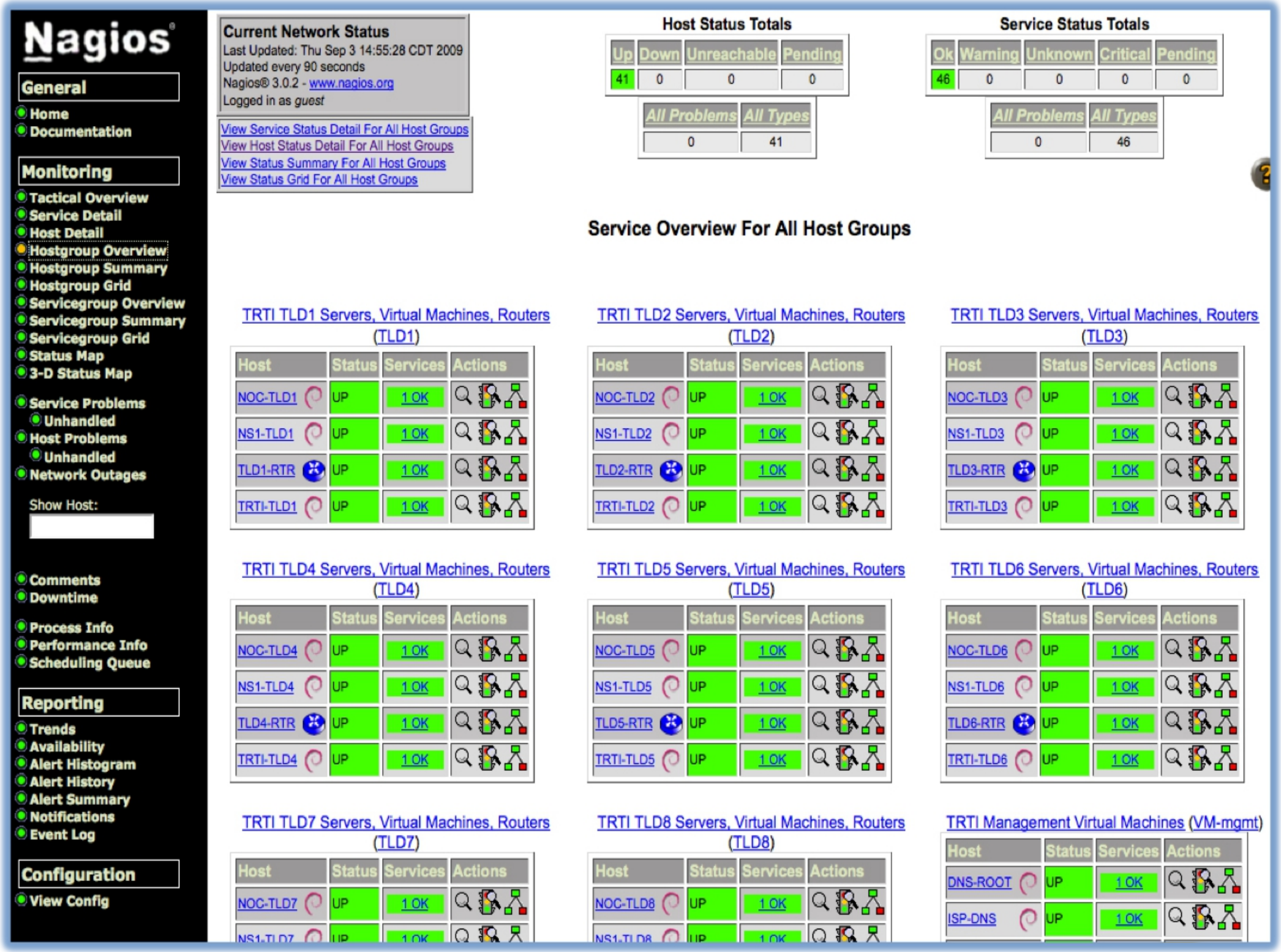
```
# check that ssh services are running
define service {
        hostgroup_name              ssh-servers
        service_description         SSH
        check_command               check_ssh
        use                         generic-service
        notification_interval       0
}
```

La “service_description” es importante si se planea crear un Grupo de Servicio (servicegroup). Debajo un ejemplo de definición de Grupo de Servicio:

```
define servicegroup{
   servicegroup_name    Webmail
   alias                web-mta-storage-auth
   members              srvr1,HTTP,srvr1,SMTP,srvr1,POP,   \
                        srvr1,IMAP,srvr1,RAID,srvr1,LDAP,  \
                        srvr2,HTTP,srvr2,SMTP,srvr2,POP,   \
                        srvr2,IMAP,srvr2,RAID,srvr2,LDAP
}
```

# Capturas de Pantalla

Algunas capturas de pantalla de una instalación de Nagios.


Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Hostgroup Overview
Hostgroup Summary
Hostgroup Grid
Servicegroup Overview
Servicegroup Summary
Servicegroup Grid
Status Map
3-D Status Map
Service Problems
Unhandled
Host Problems
Unhandled
Network Outages
Show Host:
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log
Configuration
View Config
Current Network Status
Last Updated: Thu Sep 3 14:55:28 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as guest
View Service Status Detail For All Host Groups
View Host Status Detail For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
41 0 0 0
All Problems All Types
0 41
Service Status Totals
Ok Warning Unknown Critical Pending
46 0 0 0 0
All Problems All Types
0 46
Service Overview For All Host Groups
TRTI TLD1 Servers, Virtual Machines, Routers (TLD1)
Host Status Services Actions
NOC-TLD1 UP 1 OK
NS1-TLD1 UP 1 OK
TLD1-RTR UP 1 OK
TRTI-TLD1 UP 1 OK
TRTI TLD2 Servers, Virtual Machines, Routers (TLD2)
Host Status Services Actions
NOC-TLD2 UP 1 OK
NS1-TLD2 UP 1 OK
TLD2-RTR UP 1 OK
TRTI-TLD2 UP 1 OK
TRTI TLD3 Servers, Virtual Machines, Routers (TLD3)
Host Status Services Actions
NOC-TLD3 UP 1 OK
NS1-TLD3 UP 1 OK
TLD3-RTR UP 1 OK
TRTI-TLD3 UP 1 OK
TRTI TLD4 Servers, Virtual Machines, Routers (TLD4)
Host Status Services Actions
NOC-TLD4 UP 1 OK
NS1-TLD4 UP 1 OK
TLD4-RTR UP 1 OK
TRTI-TLD4 UP 1 OK
TRTI TLD5 Servers, Virtual Machines, Routers (TLD5)
Host Status Services Actions
NOC-TLD5 UP 1 OK
NS1-TLD5 UP 1 OK
TLD5-RTR UP 1 OK
TRTI-TLD5 UP 1 OK
TRTI TLD6 Servers, Virtual Machines, Routers (TLD6)
Host Status Services Actions
NOC-TLD6 UP 1 OK
NS1-TLD6 UP 1 OK
TLD6-RTR UP 1 OK
TRTI-TLD6 UP 1 OK
TRTI TLD7 Servers, Virtual Machines, Routers (TLD7)
Host Status Services Actions
NOC-TLD7 UP 1 OK
TRTI TLD8 Servers, Virtual Machines, Routers (TLD8)
Host Status Services Actions
NOC-TLD8 UP 1 OK
TRTI Management Virtual Machines (VM-mgmt)
Host Status Services Actions
DNS-ROOT UP 1 OK
ISP-DNS UP 1 OK

**Current Network Status**
Last Updated: Fri Sep 4 13:29:20 CDT 2009
Updated every 90 seconds
Nagios® 3.0.2 - www.nagios.org
Logged in as *guest*

View Service Status Detail For All Service Groups
View Status Summary For All Service Groups
View Service Status Grid For All Service Groups

**Host Status Totals**

| Up | Down | Unreachable | Pending |
|---|---|---|---|
| 41 | 0 | 0 | 0 |

| All Problems | All Types |
|---|---|
| 0 | 41 |

**Service Status Totals**

| Ok | Warning | Unknown | Critical | Pending |
|---|---|---|---|---|
| 53 | 0 | 0 | 1 | 0 |

| All Problems | All Types |
|---|---|
| 1 | 54 |

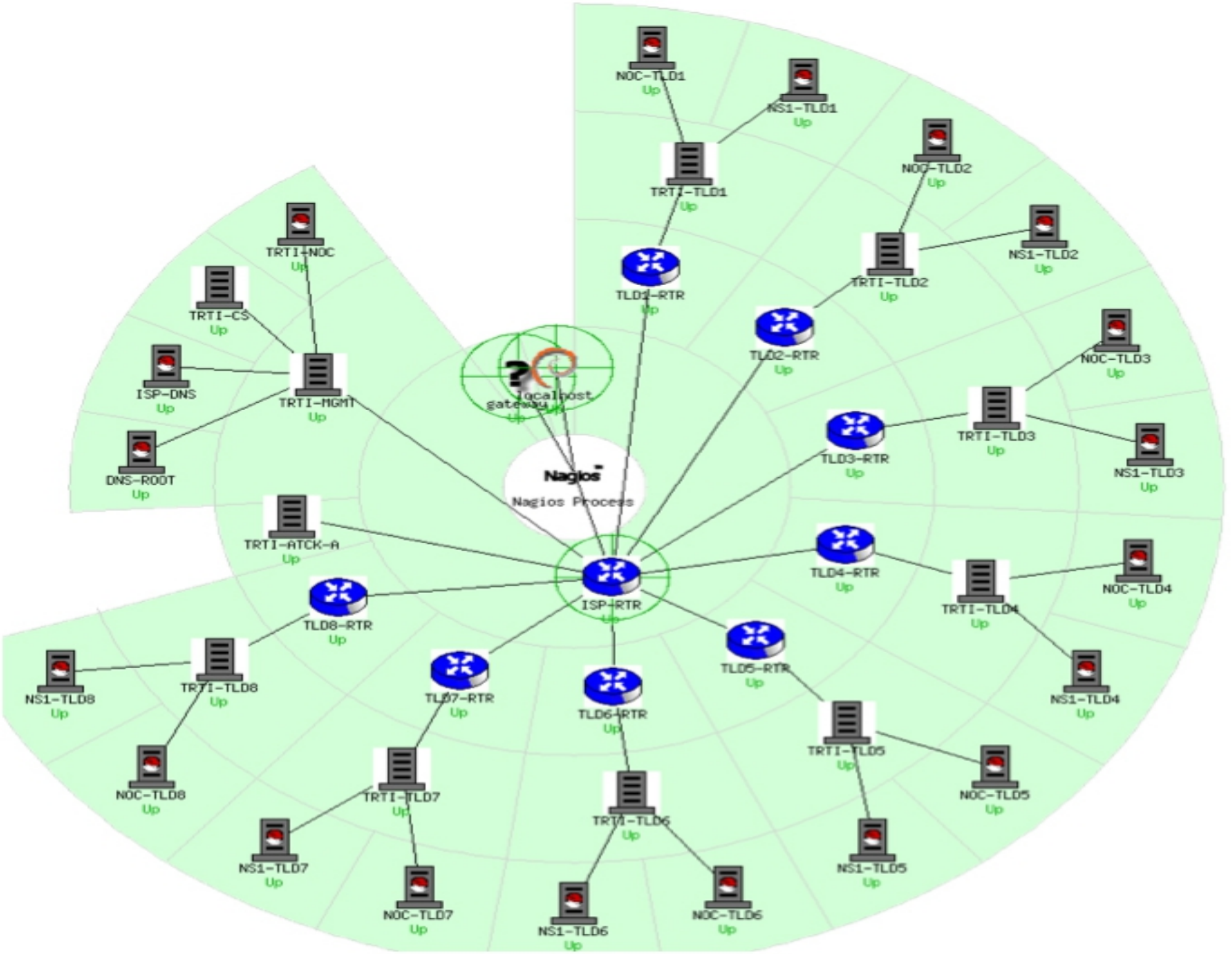
# Service Overview For All Service Groups

TLD Servers running Nagios (NAGIOS)

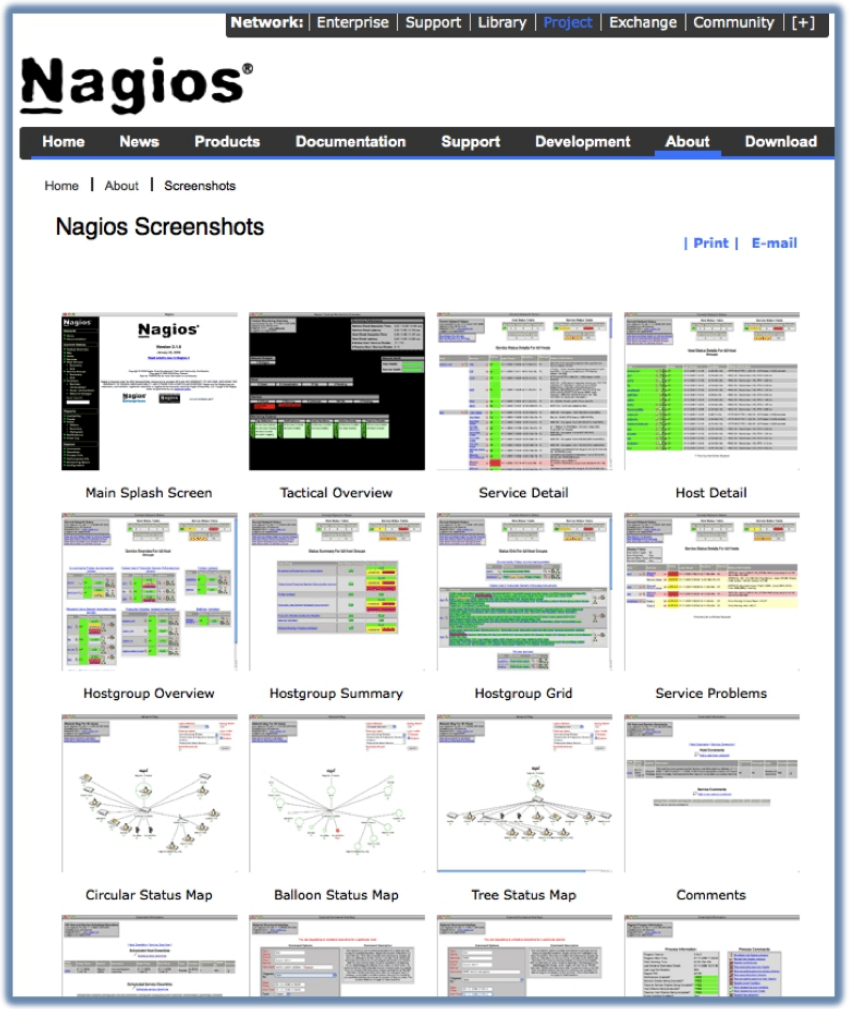
| Host | Status | Services | Actions |
|---|---|---|---|
| NS1-TLD1 | UP | 1 OK | |
| NS1-TLD2 | UP | 1 OK | |
| NS1-TLD3 | UP | 1 OK | |
| NS1-TLD4 | UP | 1 OK | |
| NS1-TLD5 | UP | 1 OK | |
| NS1-TLD6 | UP | 1 OK | |
| NS1-TLD7 | UP | 1 OK | |
| NS1-TLD8 | UP | 1 OK | |

TLD Servers running SSH (SSH)

| Host | Status | Services | Actions |
|---|---|---|---|
| NS1-TLD1 | UP | 1 OK | |
| NS1-TLD2 | UP | 1 CRITICAL | |
| NS1-TLD3 | UP | 1 OK | |
| NS1-TLD4 | UP | 1 OK | |
| NS1-TLD5 | UP | 1 OK | |
| NS1-TLD6 | UP | 1 OK | |
| NS1-TLD7 | UP | 1 OK | |
| NS1-TLD8 | UP | 1 OK | |


NOC-TLD1
Up
NS1-TLD1
Up
TRTI-TLD1
Up
NOC-TLD2
Up
TRTI-TLD2
Up
NS1-TLD2
Up
TRTI-NOC
Up
TRTI-CS
Up
TLD1-RTR
Up
TLD2-RTR
Up
NOC-TLD3
Up
ISP-DNS
Up
TRTI-MGMT
Up
localhost
gateway
Up
TRTI-TLD3
Up
TLD3-RTR
Up
NS1-TLD3
Up
DNS-ROOT
Up
Nagios
Nagios Process
TRTI-ATCK-A
Up
TLD4-RTR
Up
NOC-TLD4
Up
ISP-RTR
Up
TRTI-TLD4
Up
TLD8-RTR
Up
TLD5-RTR
Up
NS1-TLD8
Up
TRTI-TLD8
Up
TLD7-RTR
Up
TLD6-RTR
Up
NS1-TLD4
Up
TRTI-TLD5
Up
NOC-TLD8
Up
TRTI-TLD7
Up
NOC-TLD5
Up
TRTI-TLD6
Up
NS1-TLD7
Up
NS1-TLD5
Up
NOC-TLD7
Up
NS1-TLD6
Up
NOC-TLD6
Up

# Más Capturas de Pantalla




Muchas más muestras de capturas de pantalla disponibles en:

http://www.nagios.org/about/screenshots

# Nagios Lab Classroom Topology