Document d'installation FAN 2.1

Filename : FAN_Documentation_EN_v2.1-1

Version : 2.1

Date : 12/04/2011

Authors : Olivier LI-KIANG-CHEONG, Manuel OZAN, Charles JUDITH

Contents

1 PRESENTATION OF FAN

The purpose of FAN is to supply an installation CD which includes the most-used tools in the Nagios community. The FAN CD-ROM is ISO-certified. It is thus very easy to install.

A large number of tools are also being distributed, which makes the implementation of an efficient monitoring platform much easier.

1.1 Linux-based operating system

FAN is based on CentOS. All CentOS packages remain available, so that you can keep all the advantages of CentOS while having the Nagios tools already installed and configured for you.

Integrated tools:

- Nagios: core monitoring application;
- Nagios plug-ins: plug-ins to monitor different equipments;
- Centreon: Web interface for Nagios (Centreon is one of the best for this purpose!);
- NagVis: advanced mapping (geographical, functional, by services…);
- NDOUtils: stores the Nagios data into a MySQL database;
- NRPE: makes it possible to monitor the Windows servers (the NRPE daemon is not provided);

2 SUBJECTS NOT COVERED

The following subjects will not be covered by this documentation:

- The detailed use of Nagios plug-ins;
- The string theory.

3 EXISTING SOFTWARE

3.1 Nagios

Nagios™ (formerly Netsaint) is an application for system and network monitoring. It monitors the hosts and services you have specified, and informs you about the state of your systems. It is an open-source software under GPL licence.

It is a modular program which can be broken down into 3 parts:

1. The application engine which schedules the monitoring tasks.

2. The Web interface, which gives an overview of the information system and the possible anomalies.
3. The plug-ins, a hundred mini-programs or so, which can be configured according to the user's needs in monitoring each service or resource available on all computers or network devices of the Information System.

Description of the program:

- Monitoring of network services: (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, LDAP, etc…)
- Monitoring of server resources (processor load, hard disk usage, paged memory usage), and all this on the most-widespread operating systems.
- Interface with the SNMP protocol.
- The Remote Monitoring can use SSH or a SSL tunnel.
- The plug-ins are written in programming languages which are best-adapted to their tasks: script shell (Bash, ksh...), C++, Perl, Python, Ruby, PHP, C#, etc...

- The checking of services is performed in parallel.
- It is possible to create a network hierarchy in order to be able to differentiate between an unreachable and a crashed server.
- The alert notification is fully configurable through plug-ins (alerts by e-mail, text message, etc…).
- Alerts are acknowledged by the administrators.
- Alert escalation management.
- Control of visibility: the users can have their access restricted to some devices.
- Oscillation management (changes from a "normal" state to an "error" state within a short period of time).
- Each test returns a particular state:
    1. OK (everything is fine)
    2. WARNING (the alert threshold has been exceeded)
    3. CRITICAL (the service has a problem)
    4. UNKNOWN (it is impossible to know the state of the service)

3.2 Centreon

Centreon is a network monitoring software based on the **Nagios** open-source tool.

Centreon has a user-friendly interface which makes it possible for a large number of users (including non-technical people) to view the state of the system, especially with graphics. However, technicians still have access to the Nagios technical information.

In July 2007, the **Oreon** software changed names to become **Centreon**.

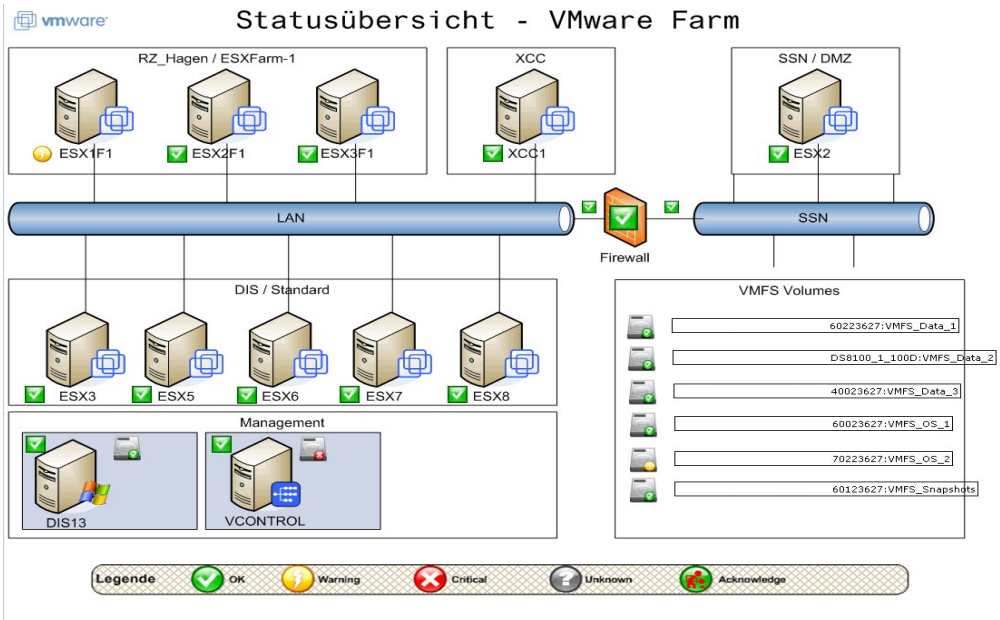
The program includes:

- An intuitive and customisable multi-user interface;
- An advanced configuration interface allowing the user to configure the area to be monitored;
- Configuration help;
- Management of all Nagios configuration files (cgi, nagios.cfg...);
- A Nagios configuration load module;
- Compatibility with Nagios 1.x, Nagios 2.x, Nagios 3.x;
- A configuration validity check with the Nagios debugger;
- Network server/hardware ID files which include all the basic information on these types of resources;
- Advanced and customisable graphic representations;
- Intelligent management of access rights, including resources as well as interface pages;
- A system of modules which makes it possible to include other applications into Centreon;
- A full incident report;
- A real-time calculation system for quality of service which notifies the user whenever quality of service decreases;
- A Java map which offers a simplified version of the information system's state (property of the Merethis Company).

3.3 Nagvis

Nagvis is visualisation module. It makes it possible to create functional views of monitoring. Nagvis can be paired with a network diagram in order to send the Nagios data to the diagram in real-time.

Example of a Nagvis diagram :



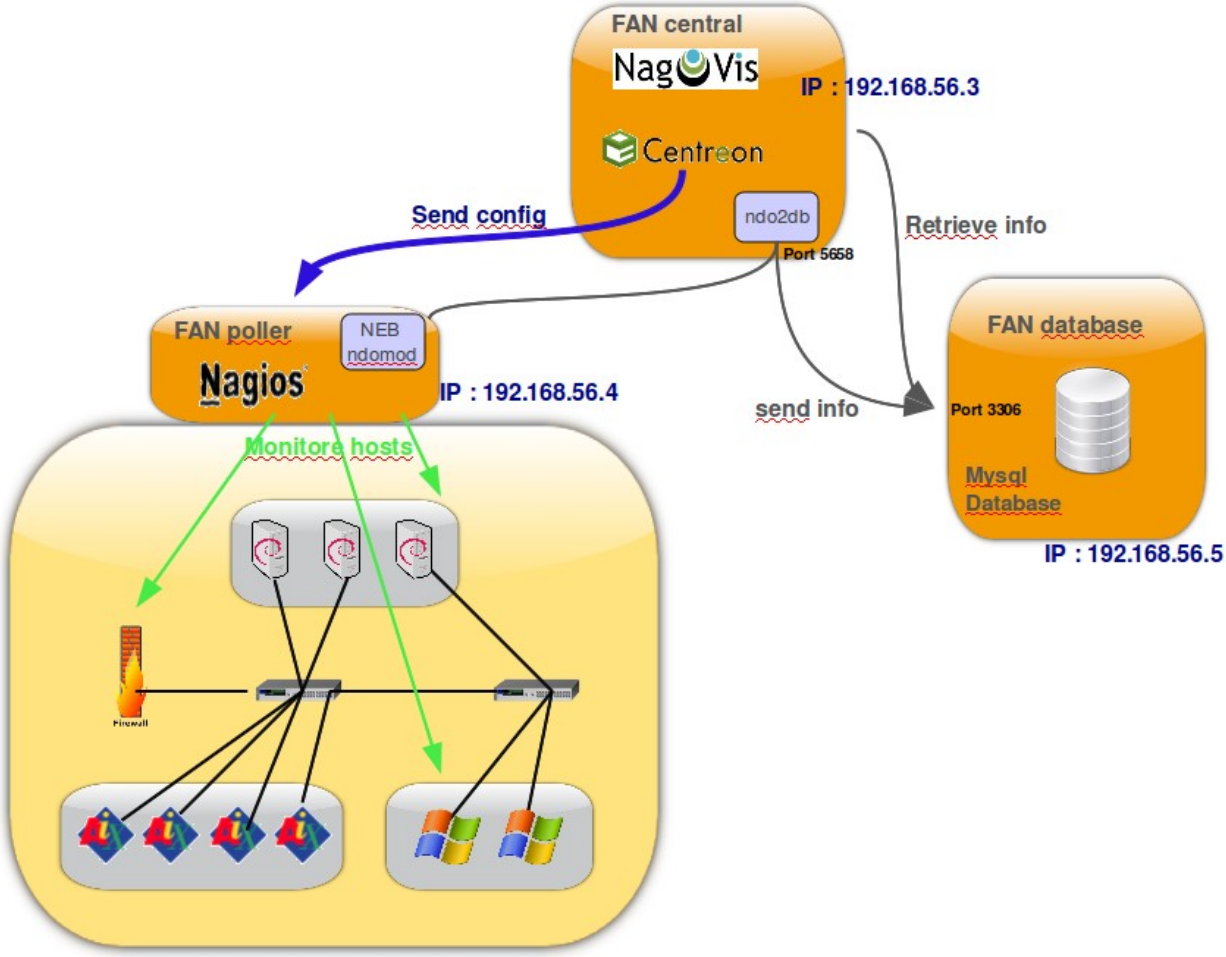
4 INSTALLATION OF FAN

Installing FAN is similar to installing a standard CentOS. It is quick and intuitive. No installation help is necessary. It requires 1 Gb. Since FAN 2.1, you can configure the distributed monitoring.

4.1 Distributed Monitoring

This feature is available since FAN 2.1. A distributed architecture is based on :

- 1 central monitoring servers
- 1 database server
- and several pollers monitors.



The central server consolidates all monitoring data and offers a user interface which also offers the possibility to monitor and manage the central server and the poller monitors. The poller monitors send their check results to the database server. This type of setup permits distribution of checks – for any type of reason f.e. remote locations, DMZ, etc.

You need install minimun 2 FAN servers :

- fan-database
- fan-central, it may also be considered as fan-poller

But, we recommend to install 3 FAN servers :

- fan-database
- fan-central
- fan-poller

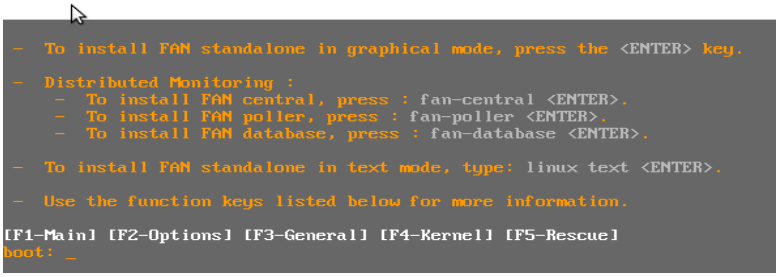
4.2 Various modes to install FAN

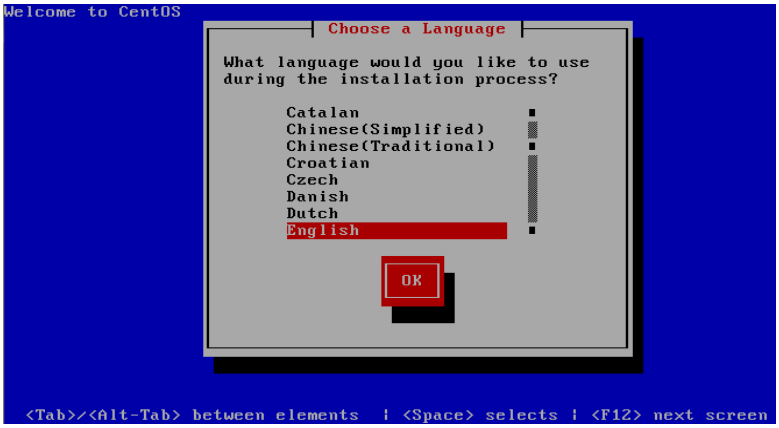
When starting the installation of FAN, several solutions available to you:

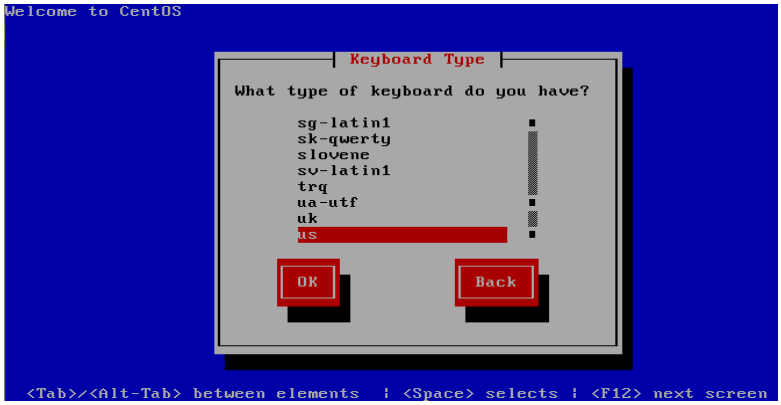
- Standalone installation (including Nagios, Centreon and database on the same server)
- FAN central (includinf Nagios, Centreon, Nagvis)
- FAN poller (including Nagios)
- FAN database (including MySQL)

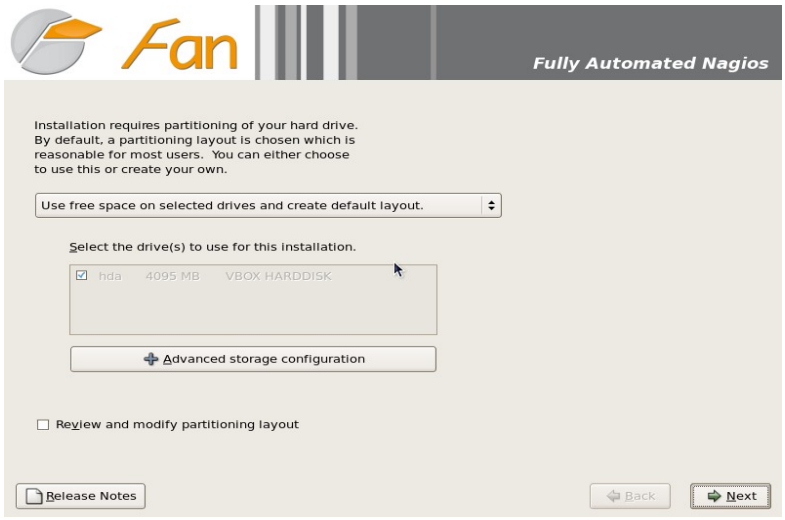
4.3 Installation

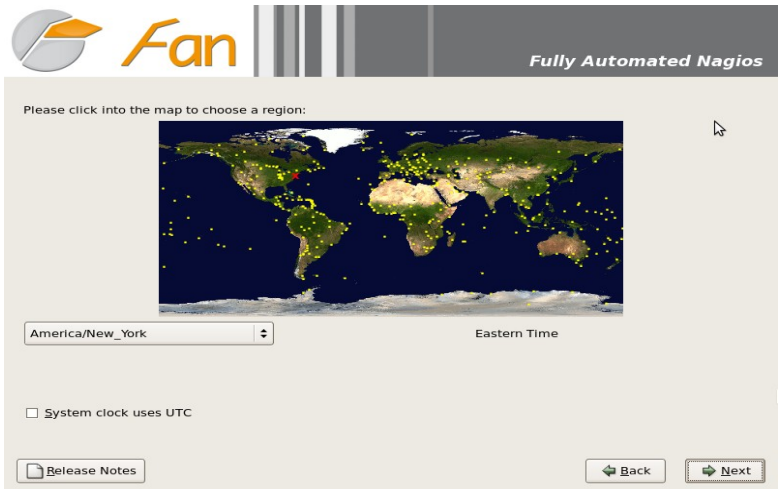
Here are the installation steps:

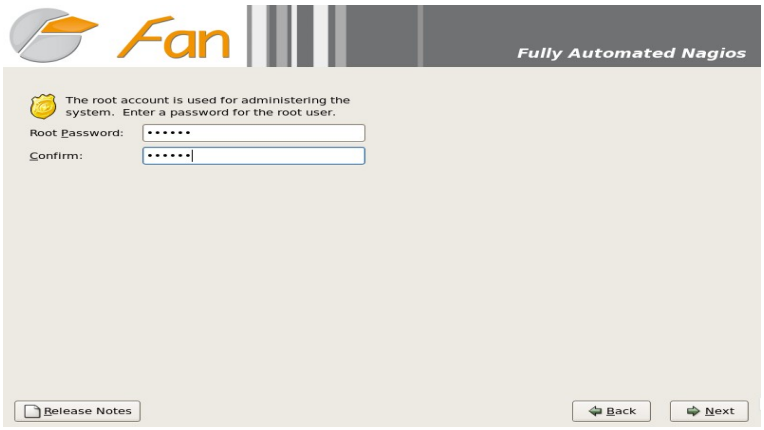





Welcome to CentOS
Keyboard Type
What type of keyboard do you have?
sg-latin1
sk-qwerty
slovene
sv-latin1
trq
ua-utf
uk
us
OK
Back
<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen


Fan
Fully Automated Nagios
Installation requires partitioning of your hard drive.
By default, a partitioning layout is chosen which is
reasonable for most users. You can either choose
to use this or create your own.
Use free space on selected drives and create default layout.
Select the drive(s) to use for this installation.
hda 4095 MB VBOX HARDDISK
Advanced storage configuration
Review and modify partitioning layout
Release Notes
Back
Next


Fan
Fully Automated Nagios
Please click into the map to choose a region:
America/New_York
Eastern Time
System clock uses UTC
Release Notes
Back
Next


Fan
Fully Automated Nagios
The root account is used for administering the system. Enter a password for the root user.
Root Password:
Confirm:
Release Notes
Back
Next


Fan
Fully Automated Nagios
Congratulations, the installation is complete.
Remove any media used during the installation process and press the "Reboot" button to reboot your system.
Release Notes
Back
Reboot

4.4 Configure the distributed monitoring

If you chose to install FAN distributed mode, you must read this chapter to configure your various servers (fan-central, fan-poller, fan-database).

4.4.1 Configure FAN database and poller

We need to temporarily allow root to login via ssh on fan poller and database.

Connect to fan-poller and modify this file /etc/ssh/sshd_config

```
PermitRootLogin yes
```

Redémarrez sshd

```
# service sshd restart
```

4.4.2 Configure FAN central

Connect to fan-central and run **system-config-distributed-monitoring** script

First configure acces fan-central to fan-database and answer questions **:**

```
# system-config-distributed-monitoring
Choose an action to do (addpoller or configdatabase):configdatabase
Give me the IP address of database server :192.168.56.5 <== IP de votre fan-database
Give me the root password of database server :
What's the IP address of fan-central (default 192.168.56.3) :[enter] <== Modifiez-le si
nécessaire
[INFO] You must enable root user to login by ssh to database server "192.168.56.5" by
ssh.
If this is not the case, edit /etc/ssh/sshd_config on database server and add
"PermitRootLogin yes"
Do you want continu ? [y/n], default to [n]:y
Stopping ndo2db: done.
Starting ndo2db: done.
Stopping Centcore
Waiting for centcore to exit . done.
Starting Centcore
Stopping centreon data collector Collector : centstorage
Waiting for centstorage to exit . done.
Starting centstorage Collector : centstorage
```

Check Centreon on fan-central :

- Connect to http://fan-central/centreon/

- and try to login to nagiosadmin/nagiosadmin

Second add poller fan-poller into fan-central :

```
# system-config-distributed-monitoring
Choose an action to do (addpoller or configdatabase):addpoller
Give me the new Poller Name ? (no space) :fan-poller  <== Ajouter une description
Give me the IP address of "fan-poller" :192.168.56.4 <== IP de votre fan-poller
Give me the root password of "fan-poller" :
What's the IP address of fan-central (default 192.168.56.3) :[enter] <== Modifiez-le si
nécessaire
[INFO] You must enable root user to login by ssh to "fan-poller" by ssh.
If this is not the case, edit /etc/ssh/sshd_config on "fan-poller" and add
"PermitRootLogin yes"
Do you want continu ? [y/n], default to [n]:y
Check if nagios user has a ssh key
Nagios user has a ssh key
Create a random password for nagios user on "fan-poller"
Copy ssh key to poller "fan-poller"
Add configuration poller in to centreon database
Stopping Centcore
Waiting for centcore to exit . done.
Starting Centcore
```

4.5 Disable root login by ssh

Connect to fan-poller and modify this file /etc/ssh/sshd_config

```
PermitRootLogin no
```

Restart sshd service

```
# service sshd restart
```

Make the same operation on fan-database.

4.6 Add a new poller

If you want to add a new poller :

- Install a FAN poller
- Permit root login by ssh
- Run configure_distributed_monitoring script

```
# system-config-distributed-monitoring addpoller
```

- Disable root login

4.6.1 Delete a poller

If you want to delete a poller,

Go to Centreon webui.

- Delete Administration>Configuration>Nagios>NagiosCFG
- Delete Administration>Configuration>Centreon>Ndomod
- Delete Administration>Configuration>Centreon>Pollers

Go to Nagvis and delete the backend for the poller.

4.6.2 Gestion des trap snmp par poller

This feature is NOT available and stable into Centreon 2.1.13.

5 FIRST CONFIGURATION

In order to be able to use our new platform, a little configuration is required. You must at least configure:

- The network (IP address, routes, DNS…)
- The hostname

5.1 Network interface

The following command allows you to configure the server's network interfaces :

```
# system-config-network
```

or

```
# vi /etc/sysconfig/networking/devices/ifcfg-eth0
```

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
ONBOOT=yes
HWADDR=00:0c:29:72:44:a3
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=192.168.1.21
```

```
GATEWAY=192.168.1.1
```

5.2 Configuring the routes

```
# route add -net 0.0.0.0 gw 10.166.200.252 netmask 255.255.255.0  (10.166.200.252
being the gateway)
```

Other routes will not be taken into account during startup.

To do so, you need to put them into a text file:

```
# vi /etc/sysconfig/network-scripts/route-eth0
```

```
GATEWAY0=10.166.200.254
NETMASK0=255.255.0.0
ADDRESS0=10.174.0.0
```

5.3 Restart the network interface

```
# service network restart
```

5.4 Backing up/Restoring the network configuration

5.4.1 Backup

```
# system-config-network-cmd -e > /tmp/network-config
```

5.4.2 Restoration

```
# system-config-network-cmd -i -c -f /tmp/network-config
```

The -i option indicates the data import, the -c option triggers the deletion (before import) of the existing configuration and the -f option specifies which file to import.

5.5 The DNS

```
# vi /etc/resolv.conf
```

```
nameserver monDNS
nameserver DNSpublic
search mondomaine
```

5.6 The machine's name

```
# vi/etc/sysconfig/network
```

```
HOSTNAME=FAN (où FAN est le nouveau nom :-)
```

Then :

```
# hostname FAN (se re-loguer)
```

5.7 Installing the graphical environment

For those who can not dispense GUI:

```
# yum --exclude=nautilus-sendto groupinstall "GNOME Desktop Environment" "X Window
System"
# startx
# system-config-display (for display configuration)
```

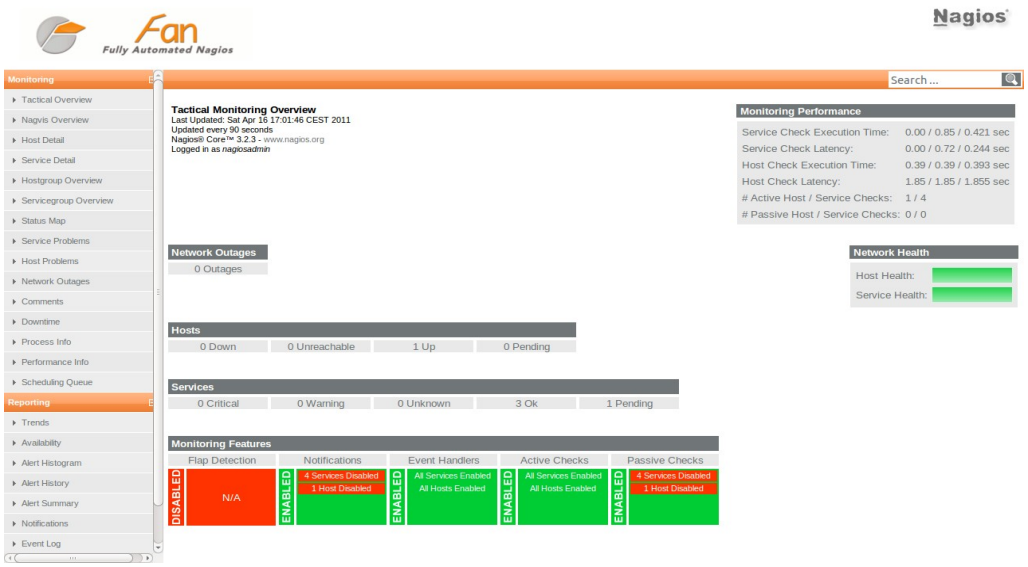
6 First steps

All monitoring tools have now been installed and configured (just what we needed!).

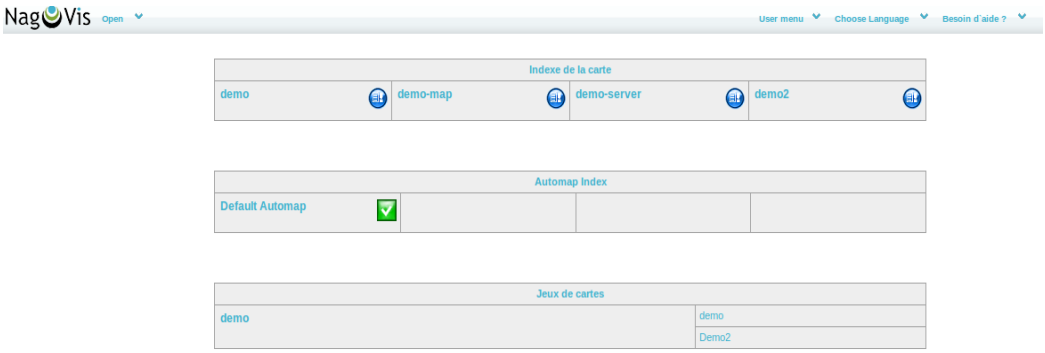
For those who can't wait, it is possible to access the project home page (from a network computer) via: **http://ip-serveur/**

This home page contains all the different services offered by FAN. You just have to click on Nagios, for example, to access the Nagios interface.
As indicated above, the default login and password are: **nagiosadmin**/**nagiosadmin**

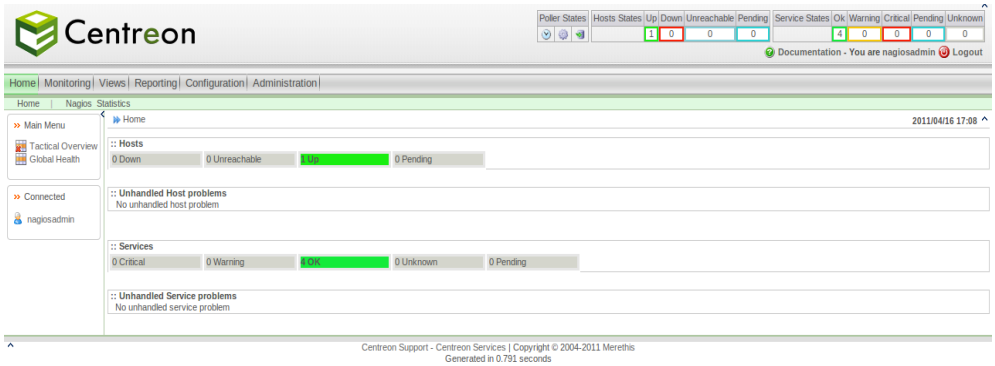
6.1 Nagios



6.2 Nagvis



6.3 Centreon



7 prerequisites

7.1 Defining the requirements

Before the first line of command, it is important to precisely define the requirements.
The following questions should be asked :

- ✔ Which device to monitor ?
- ✔ Which service to monitor ?
- ✔ Who will receive the e-mails ?
- ✔ Who will use this platform and modify it ?

This step is very important. If enough details are given, the configuration of Nagios/Centreon will be made much easier.

There is no "miracle method", but the following advice can be useful :

- ✔ Make a list of all the devices to be monitored (name and IP address);
- ✔ Identify the critical services and attach them to the devices;
- ✔ Set up a logical alert notification policy (define contacts and contact groups);
- ✔ Create a network diagram which details the dependency of the devices;

8 CONFIGURING NAGIOS

The FAN project offers different configuration choices: The platform administrator can choose to only use Nagios. I will now give more details about platform configuration and I will use the Nagios text file configuration method.

First of all, it is important to know where the main files and directories are located.

8.1 Important directories

- /etc/nagios : directory containing configuration files
- /usr/lib/nagios : directory containing CGI files and Nagios plugins
- /usr/share/nagios : directory containing monitoring web files.

8.2 Description of files

```
# ls /etc/nagios/
```

- cgi.cfg: CGI configuration file;
- localhost.cfg: definition of host "localhost" (Nagios, in other words);
- ndomod.cfg: NDOUtils configuration file;
- resource.cfg: possibility to define sensitive information (identifier, password…);
- command-plugins.cfg: definition of the Check commands;
- nagios.cfg: main Nagios configuration file;
- ndomod-load.cfg: NDOUtils configuration file ("broker_module" location);
- send_nsca.cfg: NSCA configuration file;
- commands.cfg: definition of commands (Check and Notification commands);
- nrpe.cfg: NRPE server configuration file;
- htpasswd.users: stores the usernames and passwords having access to Nagios (encrypted);
- ndo2db.cfg: NDOUtils configuration file;
- nsca.cfg: NSCA server configuration file.

8.3 Methodology

In order to add a host with services to monitor, several files need to be configured.
The following examples will help you understand how to configure Nagios. It is intended for people who have no or little knowledge of Nagios, and the following information is given as advice.
In order to simplify configuration, you can :

- Create a "conf.d" directory located in "/etc/nagios/" where you will place all your configuration files.
  Depending on your monitoring architecture (multi-site or not), you can create a directory with the company's or the site's name.

Create files named :

- servers_nameofsite.cfg;
- printers_ nameofsite.cfg;
- switches_ nameofsite.cfg;
- routers_ nameofsite.cfg.

In this way, the different devices will be defined according to their types.

We also advise you to create the following files:

- contacts.cfg : to define contacts;
- dependances.cfg : to manage dependencies;
- extinfo.cfg: to add graphical functionalities (icon…);
- services.cfg: to define services;
- hostgroups.cfg: to define host groups;
- generic-host.cfg: to define host templates;
- generic-service.cfg: to define service templates;
- time-period.cfg: to define notification periods.

9 Example of configuration

servers_nameofsite.cfg file :

```
#declaring a server                          ; comment
define host {
      host_name     SRVLEMANS           ; device name
      alias         Server Le Mans      ; alias
      address       10.166.200.100      ; IP address
      use           generic-host        ; device type
}

#declaring a server
define host {
      host_name     Fax-Server
      alias         Fax Server
      address       10.166.200.183
      use           generic-host
}
```

routers_ nameofsite.cfg file :

```
#declaring a router
define host {
      host_name     ASA-5505
      alias         Cisco Router ASA-5505
      address       10.166.200.252
      use           generic-host
}

#declaring a router
define host {
      host_name     Google
      alias         Search engine
      address       www.google.com
      use           generic-host
      parents       ASA-5505     ; device it depends on (geographically) (status_map)
}
```

The files switches_nameofsite.cfg and printers_nameofsite.cfg have the same configuration type.

hostgroups.cfg file :

```
#All devices
define hostgroup {
      hostgroup_name      All
      alias               All devices
      members             *
}

# declaring a group
define hostgroup {
      hostgroup_name      LINUX Servers       ; name of group
      alias               Axians LINUX Servers; alias
      members             nagios              ; group member, corresponds to the
host_name
}

# declaring a group
define hostgroup {
      hostgroup_name      WINDOWS Servers
      alias               Axians WINDOWS Servers
      members             SRVLEMANS, Fax Server
}
```

Example of configuration

services.cfg file :

```
define service{
      use                 generic-service     ; used template
      host_name           nagios              ; name of the affected host
      service_description User Number         ; name of service
      check_commandcheck_users!20!50   ; used command (commands.cfg)
}

define service{
      use                 generic-service
      hostgroup_name      srv-linux           ; name of the affected group
      service_description Total Processus
      check_commandcheck_procs!400!800
      }

define service{
      use                 generic-service
      host_name           nagios
      service_description Current Load
      check_commandcheck_load!5.0!4.0!3.0!10.0!6.0!4.0
      }
```

generic-host.cfg file :

```
define host{
      name                generic-host ; 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_enabled1      ; 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 ; default test command (ping)
            max_check_attempts        10
            notification_interval           0
            notification_period       24x7
            notification_options            d,u,r
            contact_groups            admins
      register            0
      }
```

Example of configuration

generic-service.cfg file :

```
define service{
      name                generic-service     ; The 'name' of this service template
      active_checks_enabled     1             ; Active service checks are enabled
      passive_checks_enabled    1             ; Passive service checks are
enabled/accepted
      parallelize_check         1             ; Active service checks should be
parallelized (disabling this can lead to major performance problems)
      obsess_over_service       1             ; We should obsess over this service (if
necessary)
      check_freshness           0             ; Default is to NOT check service
"freshness"
      notifications_enabled           1             ; Service notifications are enabled
      event_handler_enabled     1             ; Service event handler is enabled
      flap_detection_enabled    1             ; Flap detection is enabled
      failure_prediction_enabled1             ; 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
            notification_interval           0       ; Only send notifications on status
change by default.
            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
      }
```

extinfo.cfg file :

```
define hostextinfo{
      hostgroup_name      LINUX servers
      notes               LINUX servers
      icon_image          base/debian.png
      icon_image_alt      Debian GNU/Linux
```

```
      vrml_image          debian.png
      statusmap_image     base/debian.gd2
      }
```

time-period.cfg file

```
define timeperiod{
      timeperiod_name workhours
      alias           Standard Work Hours
      monday          09:00-17:00
      tuesday         09:00-17:00
      wednesday       09:00-17:00
      thursday        09:00-17:00
      friday          09:00-17:00
      }
```

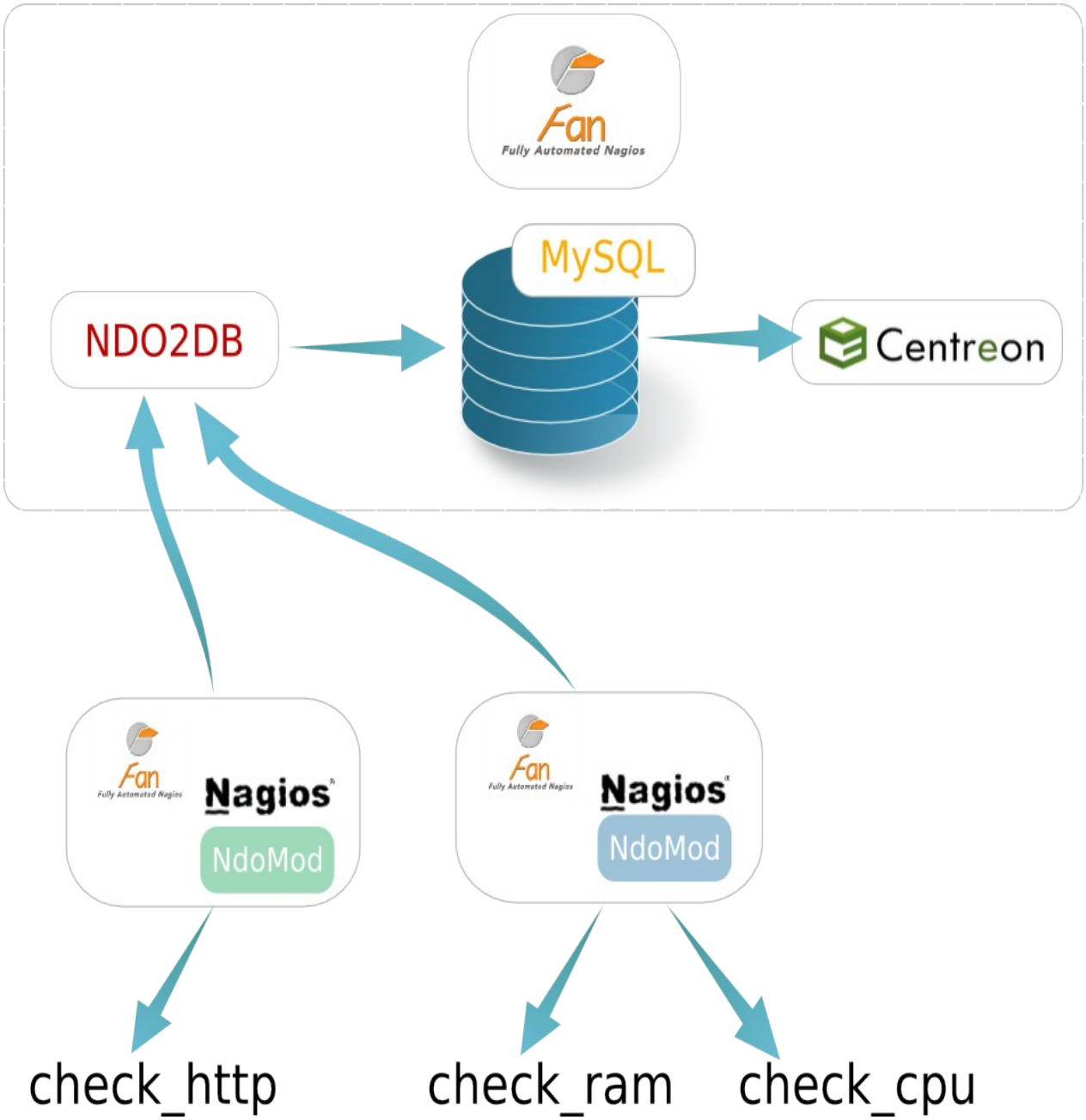
dependances.cfg file :

```
define hostdependency {
      host_name                ASA-5505
      dependent_host_name           google
      notification_failure_criteria     d     ; d -> down
      }
# If the ASA-5505 device is down, then no notification will be sent about the Google
device.
```

If you want more information about these files, please refer to the official Nagios documentation at:
http://nagios.sourceforge.net/docs/3_0/toc.html

Please note:
I won't go into too much detail about the different possible monitoring tests, since it is not the purpose of this documentation. However, you will find many links on this subject in the appendix.

With the above examples of configuration, you can quickly obtain a monitoring platform of this type:


Fan
Fully Automated Nagios
MySQL
NDO2DB
Centreon
Fan
Fully Automated Nagios
Nagios
NdoMod
Fan
Fully Automated Nagios
Nagios
NdoMod
check_http
check_ram
check_cpu

10 Useful links

Site officiel de Nagios : http://nagios.sourceforge.net/docs/3_0/quickstart.html
Communauté francophone de la supervision libre : http://wiki.monitoring-fr.org/nagios/start
Documentation de Nagios traduite en français : http://doc.monitoring-fr.org/
Blog de Nicolargo : http://blog.nicolargo.com/nagios-tutoriels-et-documentations
Site de plugins Nagios : http://www.exchange.nagios.org/
Site de Centreon : http://www.centreon.com
Site de Nagvis : http://www.nagvis.org
Site de plugins Nagios : https://www.monitoringexchange.org/