NaemonBox Manual Documentation

Release 0.0.7

NaemonBox Team

September 16, 2016

Contents

Contents:

CHAPTER 1

Introduction

1.1 Target audience

Before you start to dive deep into the documentation, we think it is fair to let you know if you get the right information out of the document or not. This documentation is intended for system administrators who want to get into monitoring and cluster management. It is also intended for user who only have to operate with the software but don't do any configurations.

The handbook provides also some background information about LINUX® commands in general, i.e. in context with package installation.

It does not deal with general monitoring themes or basic principles of monitoring or cluster management. If you wish to learn something about that, please save your time and look for a more suitable document in the world wide web or in your local specialised bookstore.

1.2 Prerequisite

Below list shows you which prerequisite of users and administrators should be fulfiled to get into monitoring or cluster management with Naemonbox software.

- Experience with LINUX® in general
- Experience with the LINUX® command line, e.g bash , zsh or other
- Experience with standard html browser
- Experience with network settings

CHAPTER 2

About Naemonbox

Naemonbox is an extansive Software Package for monitoring devices. Monitoring requires some clarification of the concepts used and how they are defined in NaemonBox. This means to observe, record, collect and display different aspects of hardware and software from activities point of view. This monitored aspects could be close to hardware like CPU Temperature, CPU Voltage, FAN Spin from NET devices but also close to services running on monitored Operating System like SSH daemons, POSTFIX daemons, HTTP services or checks the availability of devices via ping. It notifies users of outages, generates performance data for reporting, creates automated ticket with GLPI in the release of an alarm from Naemon. It allows you to centralize, and analyze log messages with cacti and rsyslog, and many other possibilities ...

A whole new way to share IT content with the various actors of an information system (Governance, Administrators, Technicians, Operators, ...). And exciting new connections between apps and devices. All that and more make NaemonBox better than ever.

Official site

2.1 Project

NAEMONBOX is a tool that allows you to install and easy to use your own Monitoring server. Having the Nagios/Naemon tools already installed and configured for you, will bring you more than you expect ...

NAEMONBOX comes with a PHP based web-tool to ease configuration and administration on One central storage. It manage groups, users and corresponding permissions and notifications. It provide Flexible preselection by device and monitoring categories with beautiful graphing of your performance data. It run distributed monitoring instances as master/worker instances to increase performance and availability in complex networks. It gives customizable map with Nagvis, well integrated with Thruk. You won't need to install another web interface.

- Easy to install : install is mainly done with the install script of the release tarball.
- Easy for new users : once installed, Naemonbox provide a single WebUI to interface with all modules and packs.
- Easy to migrate from Nagios : we want Nagios configuration and plugins to work in with Naemon. Plugins provide great flexibility and are a big legacy codebase to use. It would be a shame not to use all this community work
- Debian-platform : Naemonbox is only available for Debian OS.

This is basically what Naemonbox is made of. Maybe add the “keep it simple” Linux principle and it’s perfect. There is nothing we don’t want, we consider every features / ideas.

2.2 Features

Naemonbox has a lot of features, we started to list some of them in the last paragraph. Let’s go into details :

- NAEMON: core monitoring application. Role separated daemons : we want a daemon to do one thing but doing it good. Naemon have at least 4 daemon called worker.
- CACTI and PNP4NAGIOS : Perfomance Management, Rsyslog.
- WEATHERMAP : Mapping bandwith.
- NAGVIS : Customizable mapping.
- GLPI/FUSION : Management and inventory.
- SNMPTT : SNMP Trap translation.
- BACKUP MANAGER : Command line backup tool to make daily archives.
- MEDIAWIKI : the wiki software that powers Wikipédia.
- PSDASH: A linux system information web dashboard using psutils and flask.
- NRPE : allows you to remotely execute Nagios plugins on other Linux/Unix machines. This allows you to monitor remote machine metrics (disk usage, CPU load, etc). NRPE can also communicate with Windows agent addons like NSClient++, so you can check metrics on remote Windows machines as well.
- NSCA : allows you to integrate passive alerts and checks from remote machines and applications with Naemon. Useful for processing security alerts, as well as redundant and distributed Naemon setups.
- WEBMIN : a web-based interface for system administration.

The table below provides naemon system (only) recommandation based on one poller

Monitored Nodes /Hosts	Monitored Services	Hard Drive Space	CPU Cores	RAM
50	250	40 GB	1-2	1-4 GB
100	500	80 GB	2-4	4-8 GB
>500	>2500	120 GB	>4	>8 GB

3.1 System requirements

- An x64-compatible hardware
- 4 GB free disk space
- 1 GB of RAM.
- 1 processor core - 1 GHz CPU

3.2 Recommended system requirements

- An x64-compatible hardware
- 20 GB plus the required disk space recommended essentialy for /var. Disk space needed by mysql and rrd files
- 2 processors core or hyper-thread for each virtualized CP - 2 GHz+ CPU.
- 2 GB of RAM.

3.3 Client Operating Systems

- Windows: 2000,XP or later, 2003,2008 or later
- Linux/Unix: 2.4+ kernel Linux distributions, Solaris 9+ , FreeBSD 6.4+, AIX 5.2/5.3
- VMware ESX (i)
- NetBotz Rack Monitor (APC)
- NetApp Storage system

3.4 Openvz VPS installation

To use NaemonBox on openvz VPS, first you need to do as root (according to your timezone, change the third command line below):

```
cd /etc/
rm localtime
ln -s /usr/share/zoneinfo/Europe/Paris ./localtime
```

3.5 GNU/Linux Debian 7 (or later) Installation

Naemonbox require for running a machine with Debian GNU/Linux 7 or later ready (or based on Debian) that has network access. A video installation instructions in **expert mode** of Debian GNU / Linux 8 (codename " jessie ") on the 64-bit PC architecture (" amd64 ") is available here for French users.

Once you have access to your server, either directly or by SSH, you can install Naemonbox using the install script.

Get the latest tarball here .

3.6 Installing Naemonbox

A video installation instructions of Naemonbox is avalaible here. When installing from a released tarball, you need to run as root.

```
tar zxvf naemonbox-VerNum.tar.gz
cd naemon
./install
```

Go to url http://your_ip_adress/

- Login/password : admin/admin
- Wiki Login/password : wikiadmin/admin

Naemonbox is compatible with Nagios configuration.

CHAPTER 4

Getting Started

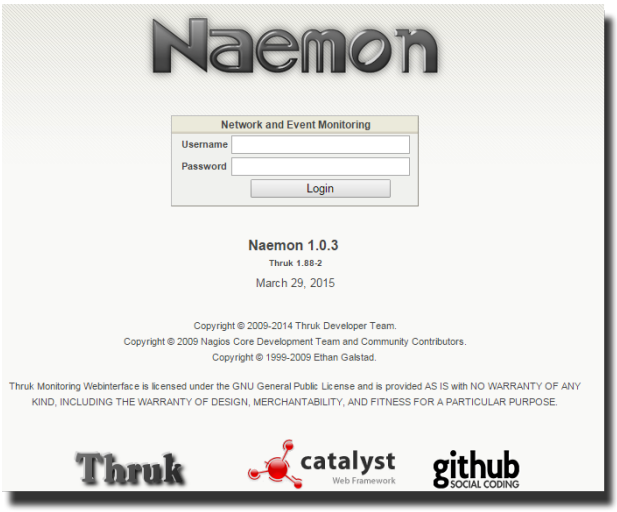
This is a quick guide to the basics of Naemonbox, written from a new user's perspective. We don't talk about advanced concepts for all apps. Visit the project website that provide complete documentation.

4.1 Step one

Before using Naemonbox, you'll need to know the fundamentals and Linux commands. I recommend you read the product documents:

- Naemon documentation available.
- Cacti documentation available.
- Pnp4Nagios.
- Weathermaps.
- Nagvis.
- Glpi.
- Fusion Inventory.
- SNMPTT.
- Mediawiki.
- Psdash.
- NRPE.
- NSCA.
- Webmin.

4.2 Step two



Connect to applications

- Use root and your password to connect by ssh or TTY
- Use admin / admin for web application (naemon, nagvis, cacti, glpi, phpmyadmin, webmin)
- Use wikiadmin / admin for the wiki.

You are strongly suggested to change credentials of the admin default user. Ready? Let's go!

Configure the monitoring

there are 2 ways :

1. Manualy, you can edit nagios/naemon config files. Not recommanded because you need to use an editor in text mode (vi, nano...).
2. Use Naemon web config tool to configure and manage naemon. That is what we will detail in the next step

4.3 Step three

The Basics workings are all the elements that are involved in the monitoring and notification logic. There are described in Configuring Naemon section.

4.4 Step four

How to monitor remote devices or services ? Several ways are possibles according to the host type's.

- Linux host (debian like, Centos...) : install SNMP agent and/or NRPE
- Windows Host : install snmp and/or nsclient++ or via WMI (Naemonbox is ready for WMI)
- Network Host (switch, router, firewall) : enable SNMP
- Network services (http, ftp, smtp, pop...) : Many plugins are availables in /usr/lib/nagios/plugins

CHAPTER 5

Configuring Naemon

5.1 Introduction

One of the features of Naemon' object configuration format is that you can create object definitions that inherit properties from other object definitions.

Tip: Also, read up on the object tricks that offer shortcuts for otherwise tedious configuration tasks.

Note:

When creating and/or editing configuration files, keep the following in mind:

1. Lines that start with a '#' character are taken to be comments and are not processed
2. Directive names are case-sensitive
3. Characters that appear after a semicolon (;) in configuration lines are treated as comments and are not processed

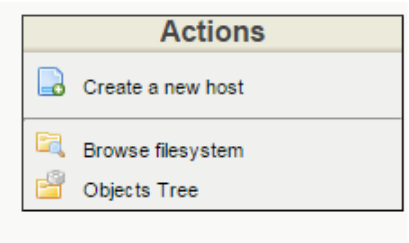
An explanation of how object inheritance works can be found here.

I strongly suggest that you familiarize yourself with object inheritance once you read over the documentation presented there, as it will make the job of creating and maintaining object definitions much easier than it otherwise would be.

Now it is time to create some configuration object definitions in order to monitor a new Windows machine. We will start by creating a basic host group for all Windows machines for one site.

5.2 Actions

In the Config Tool / Object settings menu it is possible to perform certain "generic" actions on the various objects.



5.2.1 Create a new Hosts / Services /... / Contactgroups

The creation of a new object is done via the **Create a new ...** instruction next to the **Actions menu**

5.2.2 Browse filesystem

A browser that allows you to navigate to the directory containing Naemon configuration files located in **/etc/naemon/conf.d/** folder.

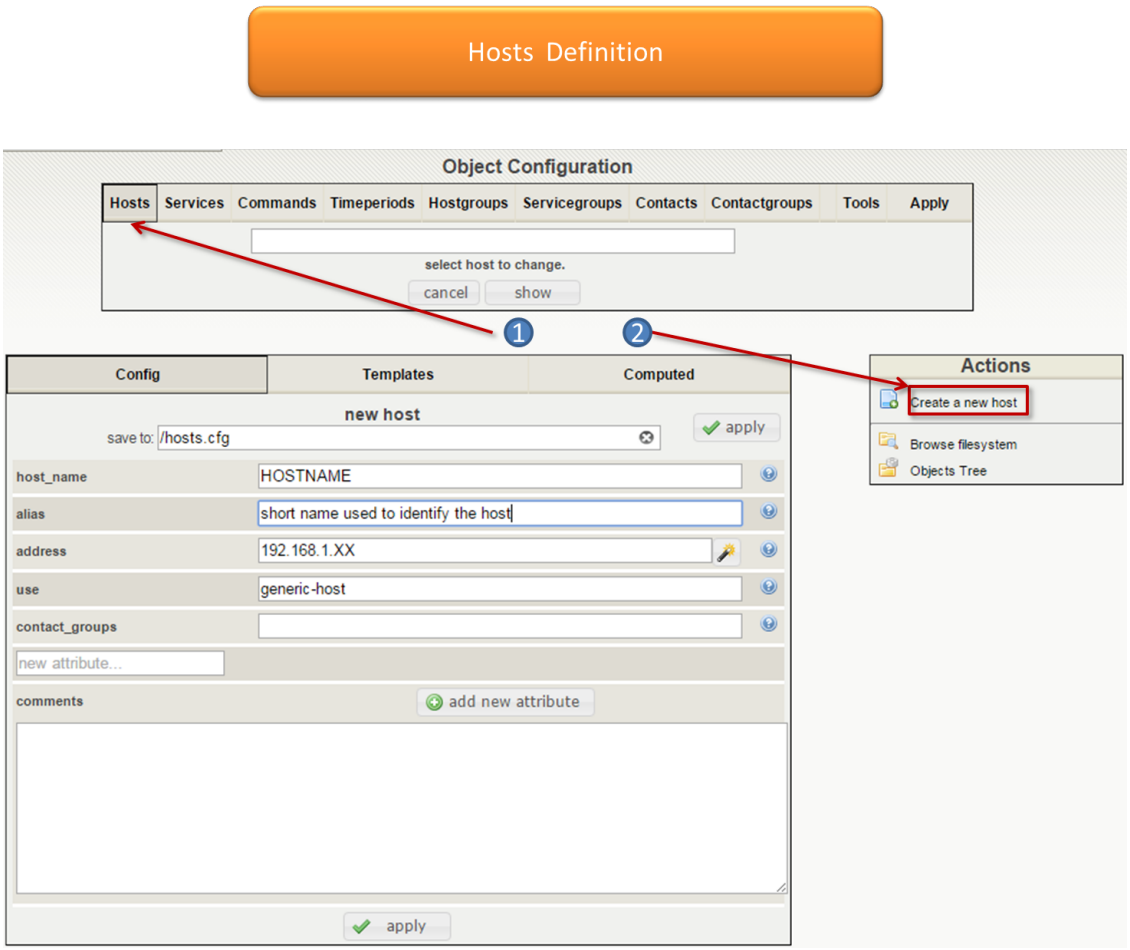
5.2.3 Objects Tree

An Objects Browser inside Naemon configuration files located folder in **/etc/naemon/conf.d/** folder.

5.3 Hosts Definition

A host definition is used to define a physical server, workstation, device, etc. that resides on your network.

All additions of hosts are done in the menu: **Config Tool ==> Object Configuration ==> Hosts ==> Create a new host**.



5.3.1 Create a host

Directive Descriptions

host_name: This directive is used to define a short name used to identify the host. It is used in host group and service definitions to reference this particular host. Hosts can have multiple services (which are monitored) associated with them. When used properly, the $HOSTNAME$ macro will contain this short name.

alias: This directive is used to define a longer name or description used to identify the host. It is provided in order to allow you to more easily identify a particular host. When used properly, the $HOSTALIAS$ macro will contain this alias/description.

address: This directive is used to define the address of the host. Normally, this is an IP address, although it could really be anything you want (so long as it can be used to check the status of the host). You can use a FQDN to identify the host instead of an IP address, but if DNS services are not available this could cause problems. When used properly, the $HOSTADDRESS$ macro will contain this address.

use: Link to the the template you use

contact_groups: This is a list of the short names of the contact groups that should be notified whenever there are problems (or recoveries) with this host. Multiple contact groups should be separated by commas. You must specify at least one contact or contact group in each host definition.

new attribute: A field used to add a new directive wich is filled with the **add new attribute** button.

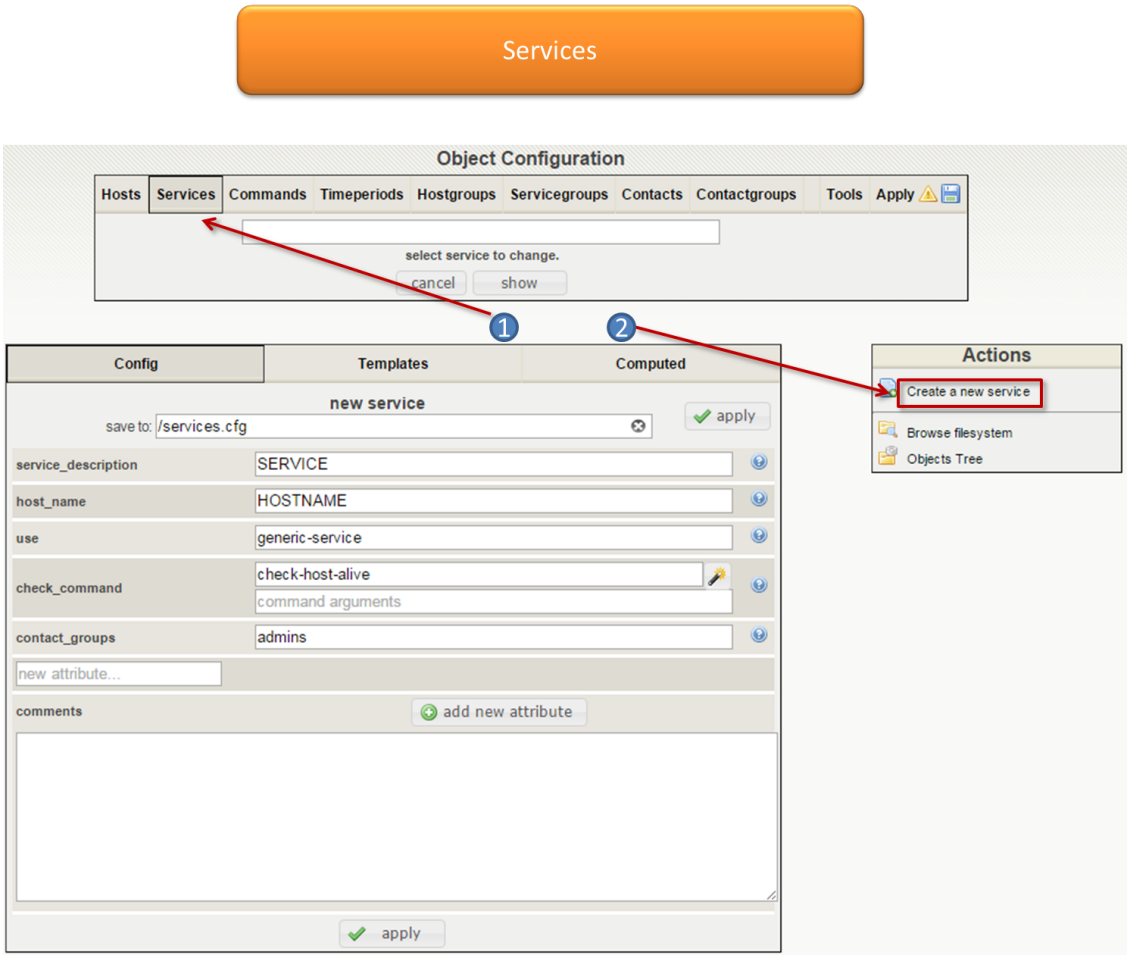
A descrition of all directives is located here.

apply Click on apply to save.

5.4 Services

A service is a check point linked / attached to a host. E.g.: Percentage of partition use on a server, ink level in a printer.

All additions of services are done in the menu: **Config Tool ==> Object Configuration ==> Services ==> Create a new service**.



5.5 Commands

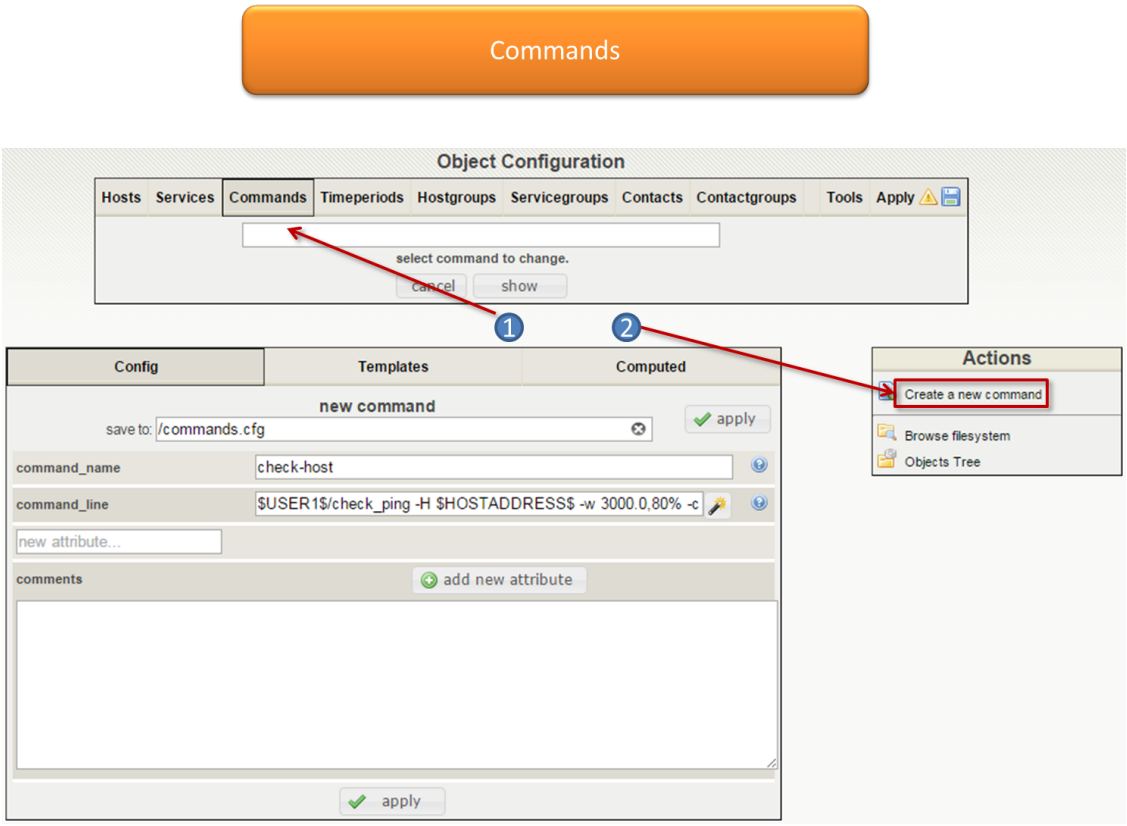
5.5.1 Definition

A command is the definition of a line of command which uses a script or an application to perform an action. It is possible execute this command by specifying arguments.

There are three types of command:

- **Verification** commands are used by the schedulers to verify the status of a host or of a service.
- **Notification** commands are used by the schedulers to alert the contacts (via mail, SMS, etc.).
- **Miscellaneous** commands are used by the additional modules (to perform certain actions), by the scheduler for data processing, etc.

All the commands can be configured in the menu: **Config Tool ==> Object Configuration ==> Commands ==> Create a new command**.



5.6 Time periods

5.6.1 Description

A time period is a list of times during various days that are considered to be "valid" times for notifications and service checks. It consists of time ranges for each day of the week that "rotate" once the week has come to an end.

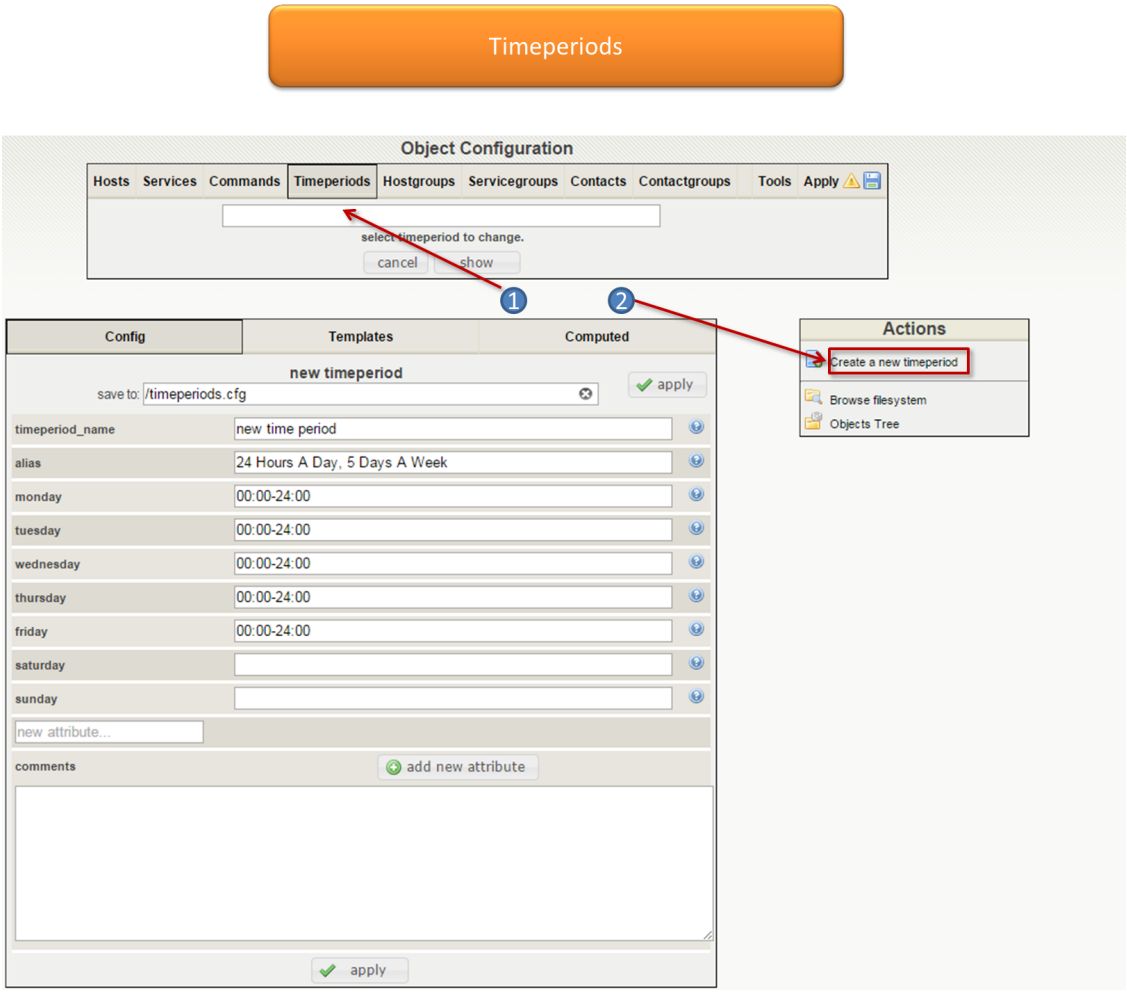
Different types of exceptions to the normal weekly time are supported, including: specific weekdays, days of generic months, days of specific months, and calendar dates.

Time periods apply to two types of actions:

- Execution of check commands
- Sending of notifications

5.6.2 Configuration

The configuration of time periods is done in the menu: **Config Tool ==> Object Configuration ==> Timeperiods ==> Create a new timeperiod**.



Basic options

- The **Time period name** and **Alias** fields define the name and description of the time period respectively.
- The fields belonging to the **Time range** sub-category define the days of the week for which it is necessary to define time periods.
- The **Exceptions** table enables us to include days excluded from the time period.

Syntax of a time period

When creating a time period, the following characters serve to define the time periods :

- The character ":" separates the hours from the minutes. E.g.: HH:MM
- The character "-" indicates continuity between two time periods
- The character "," serve s to separate two time periods

Here are a few examples:

- 24 hours a day and 7 days a week: 00:00-24:00 (to be applied on every day of the week).
- From 08h00 to 12h00 and from 14h00 to 18h45 (to be applied on weekdays only).

5.7 Contacts

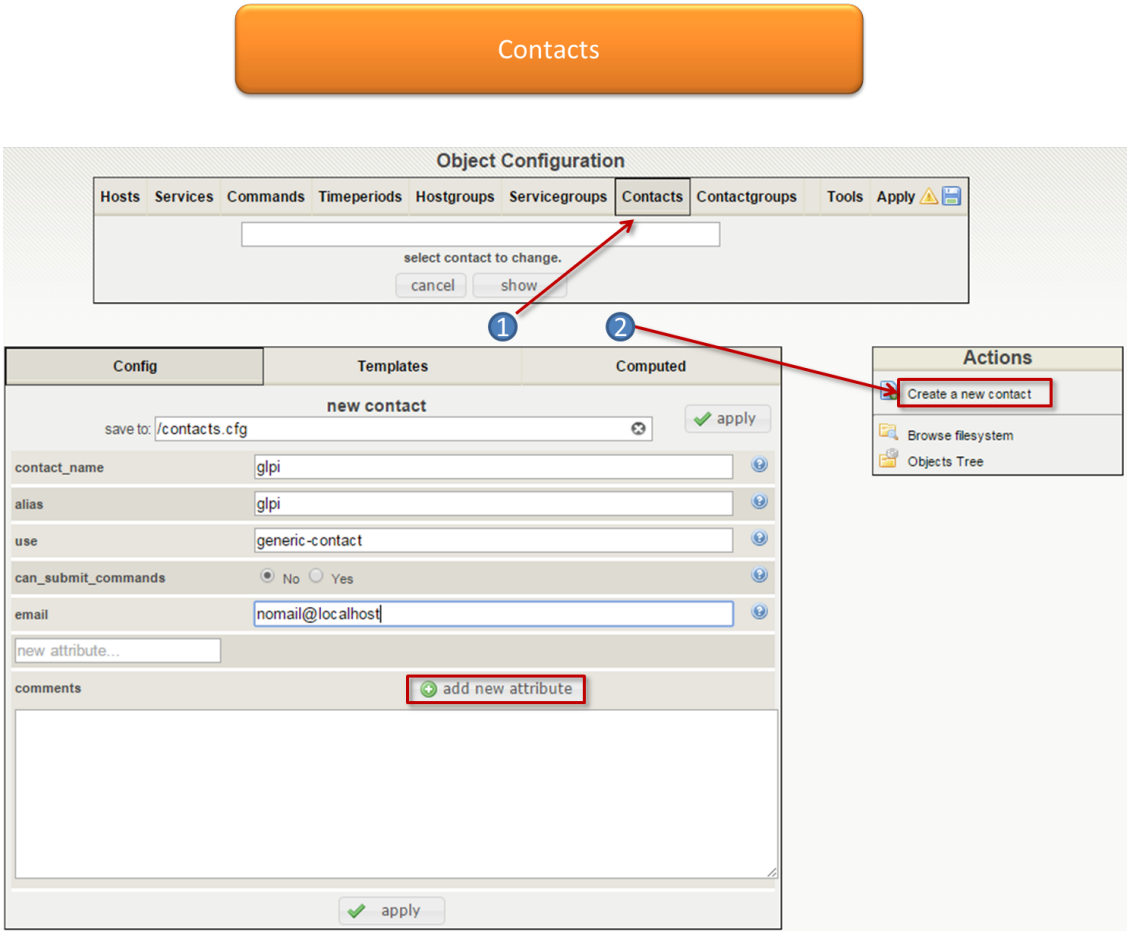
5.7.1 Definition

The contacts in Naemon are used to:

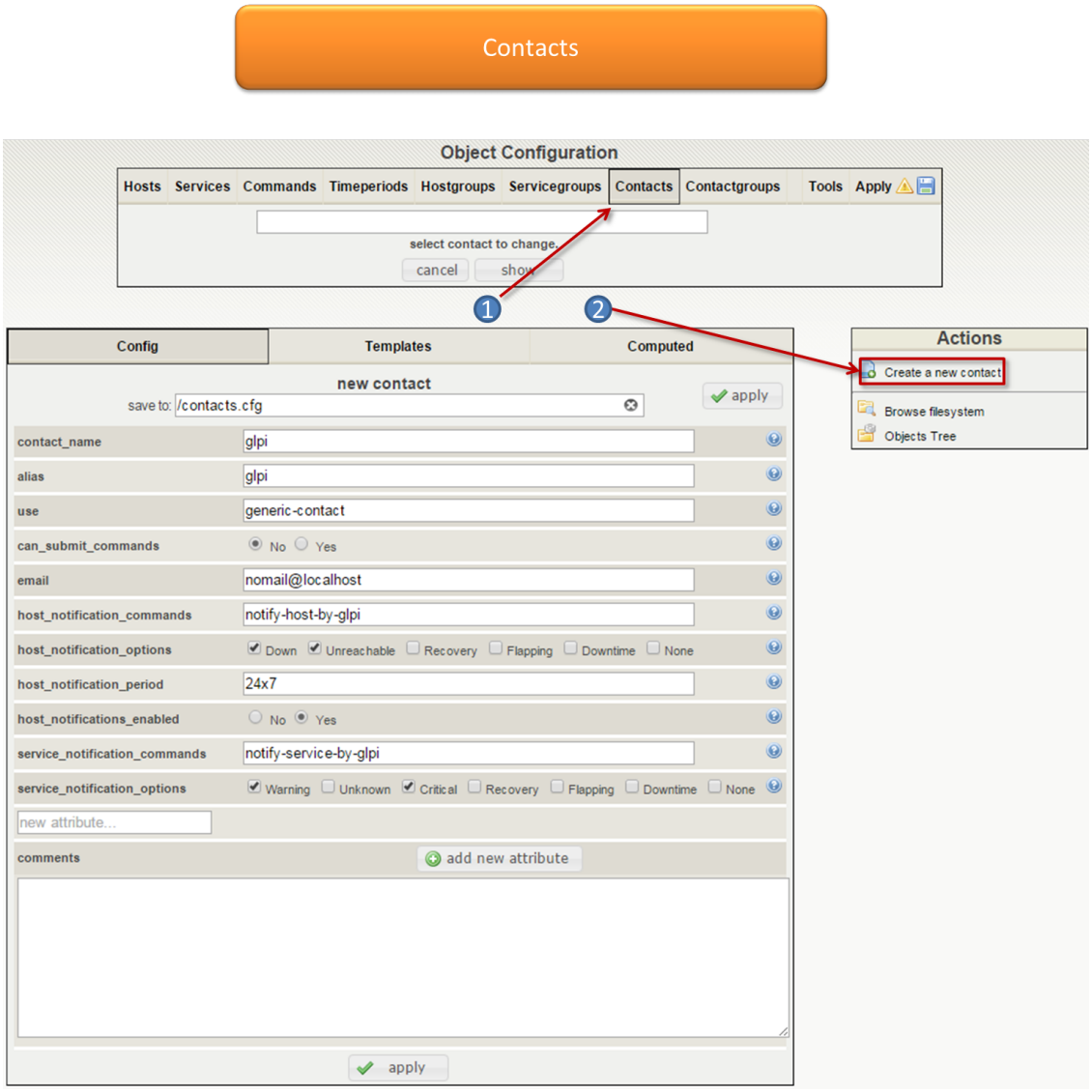
- Log in to the Naemon web interface: each contact has its own rights of authentication to the web interface.
- Be warned in the event of a problem on your network (notification).

To add a contact, simply go to the menu: **Config Tool ==> Object Configuration ==> Contacts ==> Create a new contact**.

To display a contact, click on the **Object Configuration** field under **Contacts**, then click on the **Show** button.



According to your needs, add more attributes by clicking on **add new attributes**, fill the field you just selected and **click on finish**.



5.7.2 General information

- The **contact_name** field defined the login to access the web interface.
- The **alias** field is used to define a longer name or description for the contact.
- The **use** field allows us to link the contact to a Model of contact.
- The **can_submit_commands** directive is used to determine whether or not the contact can submit external commands to Naemon from the CGIs. Values: 0 = don't allow contact to submit commands, 1 = allow contact to submit commands.
- The **email** field contain the e-mail address of the contact (to send out an alert email to the contact).
- The **contactgroups** is used to associated the contact to one or more groups of contacts.
- The **host_notification_commands** field serves to choose the notification command to a host.
- The **host_notification_options** field serves to define states for which notifications can be sent out to this contact..
- The **host_notification_period** field serves to choose the time period for which notifications can be sent.

- The **host_notification_enable** directive allows us to enable the sending of notifications to the user.
- The **service_notification_commands** field serves to choose the notification command to a service.
- The **service_notification_options** field serves to define states for which notifications can be sent out to this contact..
- **minimum_value**: This directive is used as the value that the host or service hourly value must equal before notification is sent to this contact.
- **pager**: This directive is used to define a pager number for the contact.
- **addressx**: Address directives are used to define additional "addresses" for the contact.
- **retain_status_information**: This directive is used to determine whether or not status-related information about the contact is retained across program restarts.
- **retain_nonstatus_information**: This directive is used to determine whether or not non-status information about the contact is retained across program restarts.

5.7.3 User Configuration

You can either edit or create, simply go to the menu: **Config Tool ==> User settings**.

- The **Usernamed** field serves to select an existing user to change or to create a new user (just type his name) to access the Naemon web interface.
- The **Contact Exists** field let us edit the user settings
- The **Password** and **Confirm Password** fields contain the user password.

You can now set to yes or no the global authorization functionality when determining what the users have access to. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Tip: A sample CGI configuration file (*/etc/naemon/cgi.cfg*) is installed for you.

- This CGI allows you to view objects (i.e. hosts, host groups, contacts, contact groups, time periods, services, etc.) that you have defined in your object configuration file(s).
- You must be authorized for configuration information in order to any kind of configuration information.

Note: The defaut admin user has all authorisations set to **yes**.

5.8 Groups

A group allows us to group together one or more objects. There are three kinds of groups: hosts, services and contacts.

The hosts groups and services groups serve mainly for viewing graphics or to group the objects. Contact groups are used mainly for the configuration of ACLs.

5.8.1 Host Groups

To add a host group:

1. Go to the menu: **Config Tool ==> Object Configuration ==> Hostgroups**

2. In the Action menu, click on **Create a new hostgroups**

- The **Host Group Name** and **Alias** defines the name and the alias of the host group.
- The **Linked Hosts** list allows us to add hosts in the hostgroup.
- The **Notes** field allows us to add optional notes concerning the host group.
- The **Notes URL** field defined a URL which can be used to give more information on the hostgroup.
- The **Action URL** field defined a URL normally use to give information on actions on the hostgroup (maintenance, etc.).
- The **Icon** field indicates the icon to be use for the host group.
- The **Map Icon** is the icon use for mapping.
- The **RRD retention** field is expressed in days, it serves to define the duration of retention of the services belonging to this hostgroup in the RRD database. It will be the default duration defined in the menu: " **Administration ==> Options ==> CentStorage** " if this value is not defined.
- The **Status** and **Comments** fields allow to enable or disable the host group and to make comments on it.

5.8.2 Service Groups

To add a service group:

1. Go into the menu: **Config Tool ==> Object Configuration ==> Servicegroups**
2. In the Action menu, click on **Create a new servicegroup**

- The **Service Group Name** and **Description** fields describes the name and the description of the service group.
- The **Linked Host Services** list allows us to choose the various services that will be included in this group.
- The **Linked Host Group Services** list allows us to choose the services linked to a host group that will be part of this group.
- The **Linked Service Templates** list allows to deploy a service based on this template on all hosts linked to this group.
- The **Status** and **Comments** fields allow to enable or disable the service group and to make comment on it.

5.8.3 Contact Groups

To add a group of contacts:

1. Go into the menu: **Config Tool ==> Object Configuration ==> Contactgroups**
2. In the Action menu, click on **Create a new contactgroup**

- The **Contact Group Name** and **Alias** fields define the name and the description of the contact group.
- The **Linked Contacts** list allows us to add contacts to the contact group.
- The **Status** and **Comment** fields allow to enable or disable the group of contacts and to make comment on it.

Note: For more information refer to the associated chapter covering groups.

5.9 Tools

Tools can find some issues from a cross reference check.

Go in the menu: **Config Tool ==> Object Configuration ==> Tools ==> Check Object References**

5.10 Apply

Show the output before apply config changes

Config Parse error: verify the syntax of an object reference

1. Verify your changes: view diff chenged files

2. Save changes to disk: showing the path file to save your changes to disk

3. Check your configuration changes: check your configuration changes

4. Reload your monitoring core: reload your monitoring core

Discard changes: discard all unsave changes

CHAPTER 6

Configuration by example

6.1 Configuring Naemon

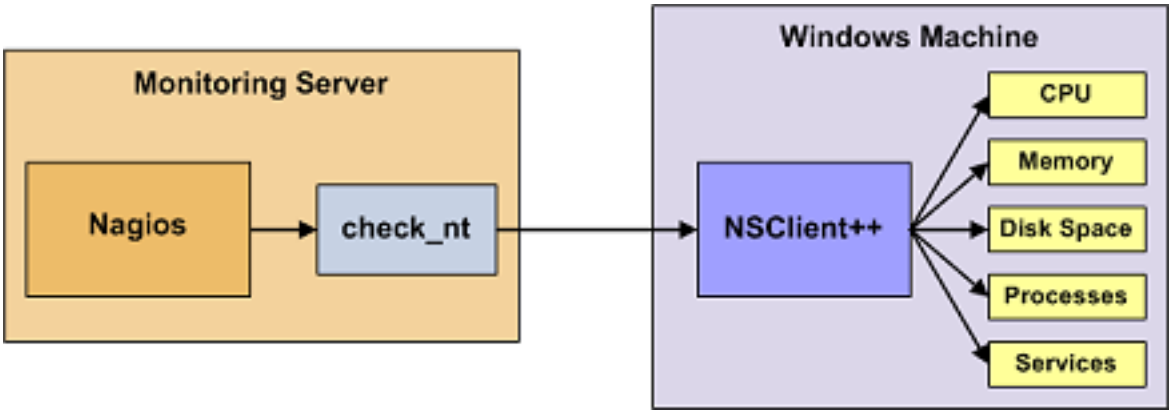
6.1.1 Introduction

This document describes how to monitor local and Windows machines attributes, such as:

- Memory Usage

- CPU load
- Disk Usage

6.1.2 Overview



Monitoring services or attributes of a Windows machine requires the installation of an agent. This agent acts as a proxy between the Nagios plugin and the actual service or attribute of the Windows machine to be monitored . Without installing an agent on the Windows machine, Naemon would be unable to monitor local services or attributes of the Windows machine.

For this example, we will install NSClient ++ on the Windows machine and using the plugin check_nrpe that will communicate with the addon NSClient .

6.1.3 Steps

There are several steps you need to follow in order to monitor a new Windows machine:

- Install a monitoring agent on the Windows machine
- Create new host and service definitions for monitoring the Windows machine
- Restart the Naemon Service

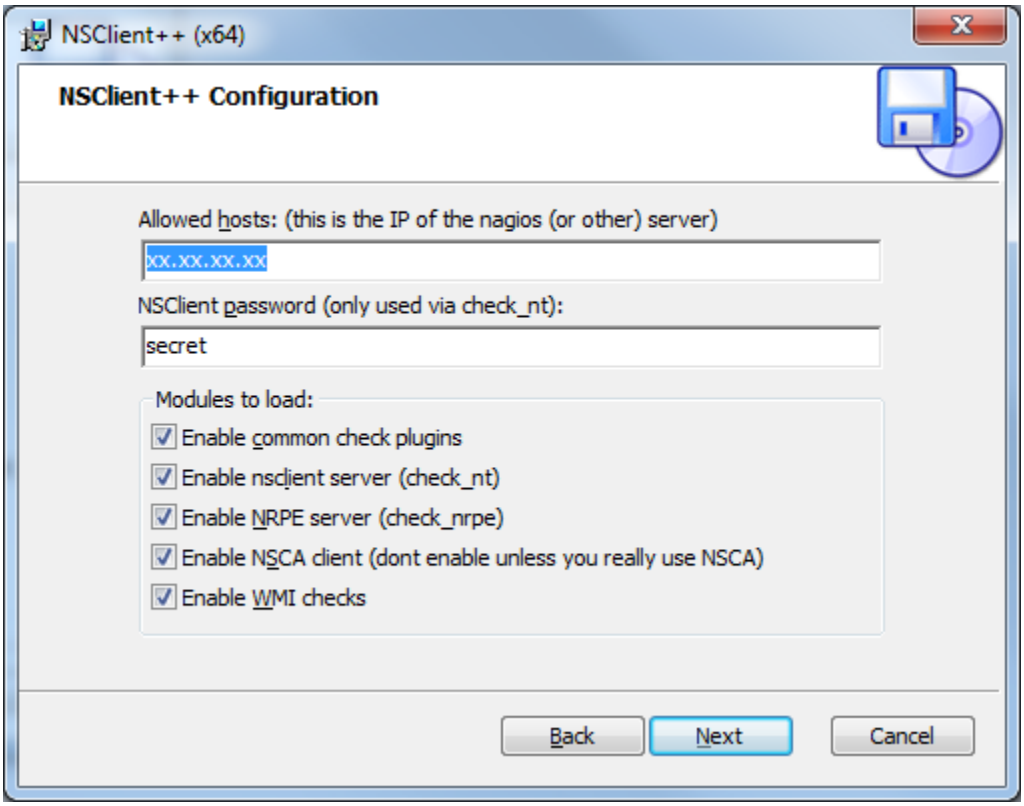
6.2 Windows Agent Installation

Naemon recommends using NSClient ++. These instructions allow a NSClient basic installation and Naemon configuration to monitor the Windows machine.

1. Download the latest stable version from http://www.nsclient.org
2. Install NSClient++ and use the “Complete” setup type to make sure you got all features. On the next page use the default path for nsclient.ini and make sure “Install sample configuration” are checked
3. Enter all hosts that are allowed to connect to NSClient++, separate multiple IP’s with ”,”. Make sure to check the following:

- Check **Enable common check plugins**
- Check **Enable nsclient server (check_nt)**
- Check **Enable NRPE server (check_nrpe)**

- Check **Enable NSCA client**
- Check **Enable WMI checks**



NSClient++ should be installed and set up to start automatically. This should be enough to start with some basic Windows monitoring.

6.3 How-to Monitor

6.3.1 Introduction

One of the features of Naemon' object configuration format is that you can create object definitions that inherit properties from other object definitions.

Tip: Also, read up on the object tricks that offer shortcuts for otherwise tedious configuration tasks.

Note:

When creating and/or editing configuration files, keep the following in mind:

1. Lines that start with a '#' character are taken to be comments and are not processed
2. Directive names are case-sensitive
3. Characters that appear after a semicolon (;) in configuration lines are treated as comments and are not processed

An explanation of how object inheritance works can be found here.

I strongly suggest that you familiarize yourself with object inheritance once you read over the documentation presented there, as it will make the job of creating and maintaining object definitions much easier than it otherwise would be.

Now it is time to create some configuration object definitions in order to monitor a new Windows machine. We will start by creating a basic host group for all Windows machines for one site.

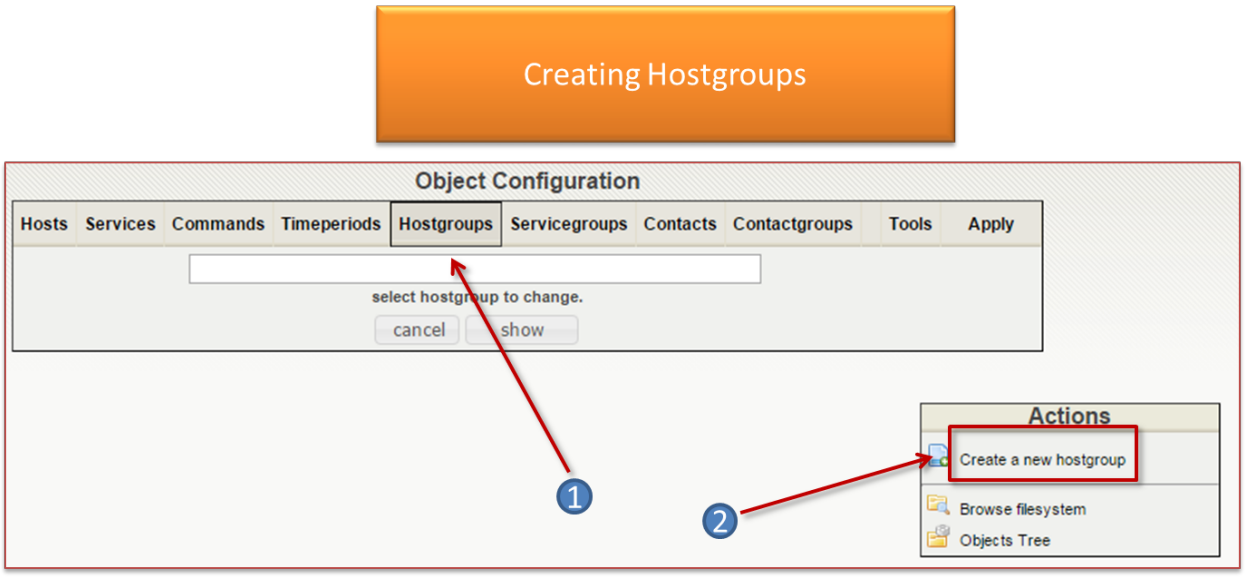
6.4 Host Group Definition

A host group definition is used to group one or more hosts together to simplify configuration.

6.4.1 Add an hostgroup

For editing , go through :

1. Naemon Setup menu **Config Tool ==> Object settings ==> Hostgroups.**
2. **Create** or **Clone.**
3. Make changes and click on **Apply.**



6.5 Command Definition

A command define the command line that uses a script or an application to perform an action . You can run this command by specifying arguments.

There are three types of commands:

- Audit controls are used by the schedulers to check the status of a host or service .
- Notification commands are used by the schedulers to alert contacts (via mail, SMS ...) .
- Various commands are used by add-ons (to perform certain actions) by the scheduler for data processing ...

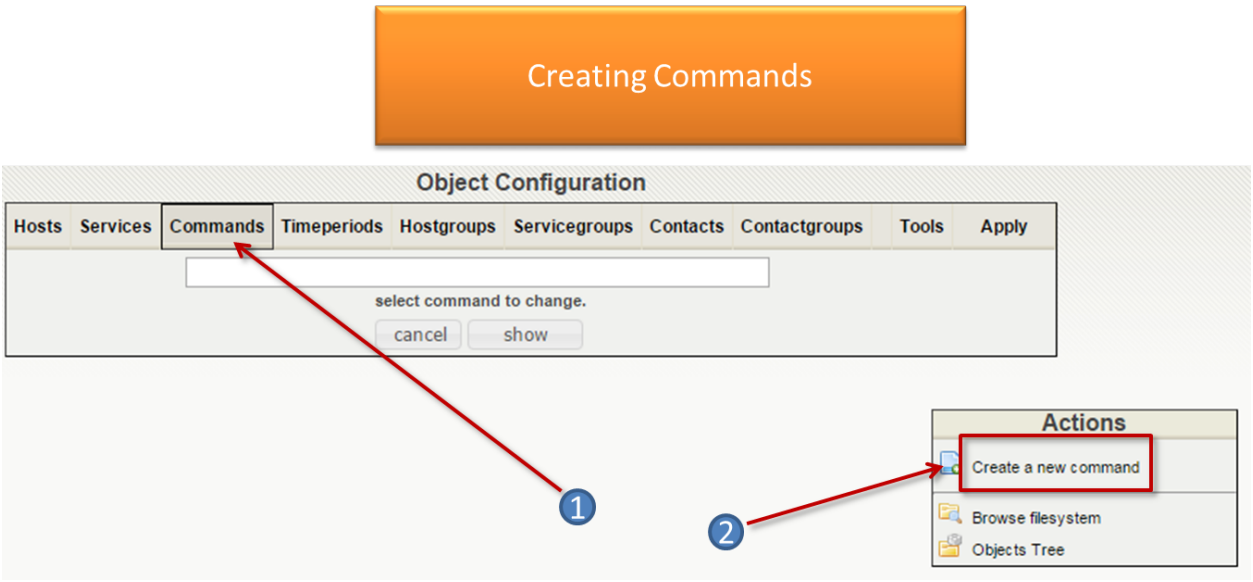
6.5.1 Add a command

Configuration

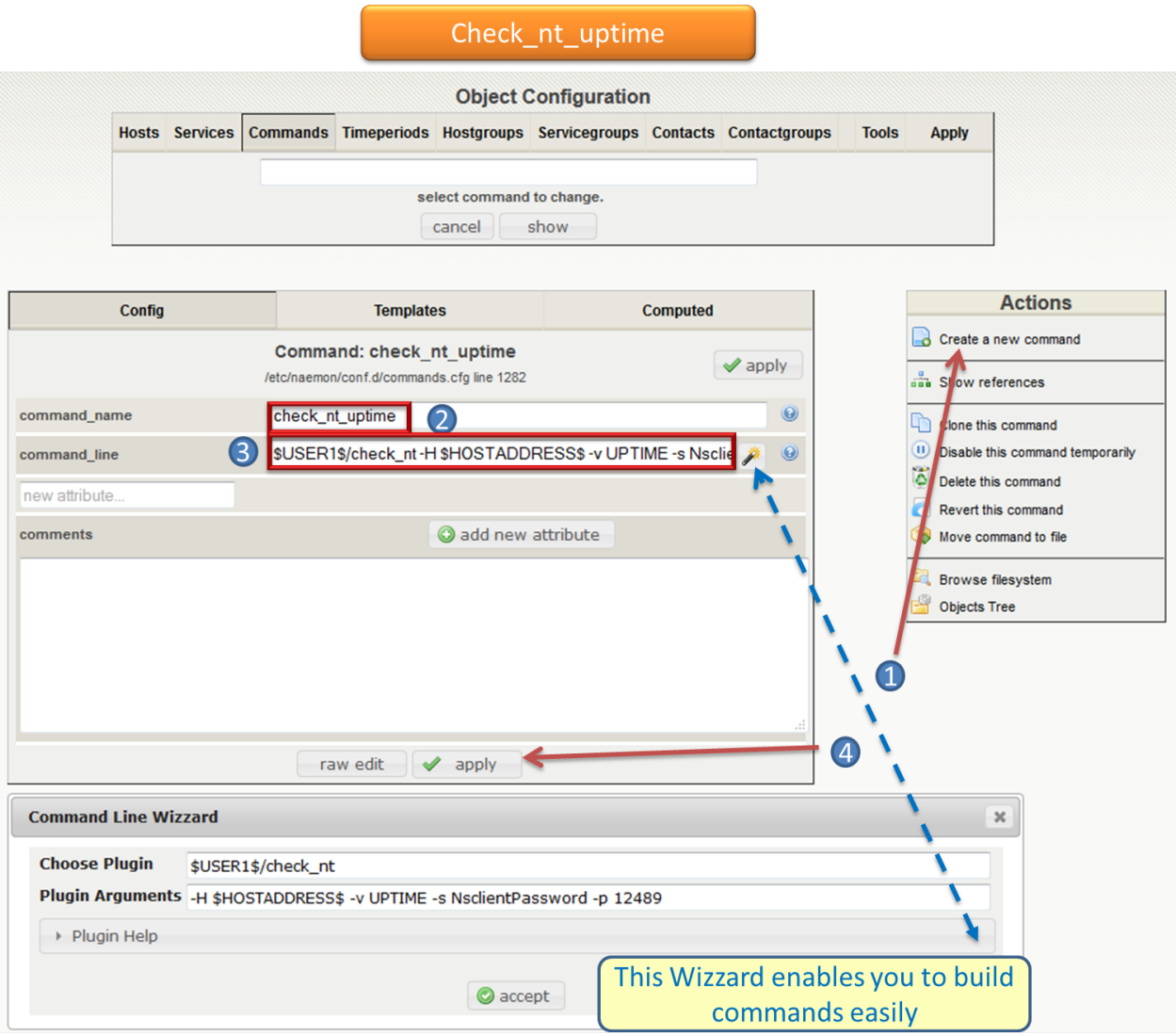
check_nt_uptime

We will add a "command / plugin " check_nt_uptime that will be used by the system start service we will create for our "host" . For editing , go through

1. **Config Tool Menu ==> Object Configuration ==> Commands**
2. **Create a new command**



3. Enter the command name **check_nt_uptime**.
4. Enter the following command line $USER1$/check_nt -H $HOSTADDRESS$ -v UPTIME -s NsclientPassword -p 12489
5. Save, click on **Apply**.

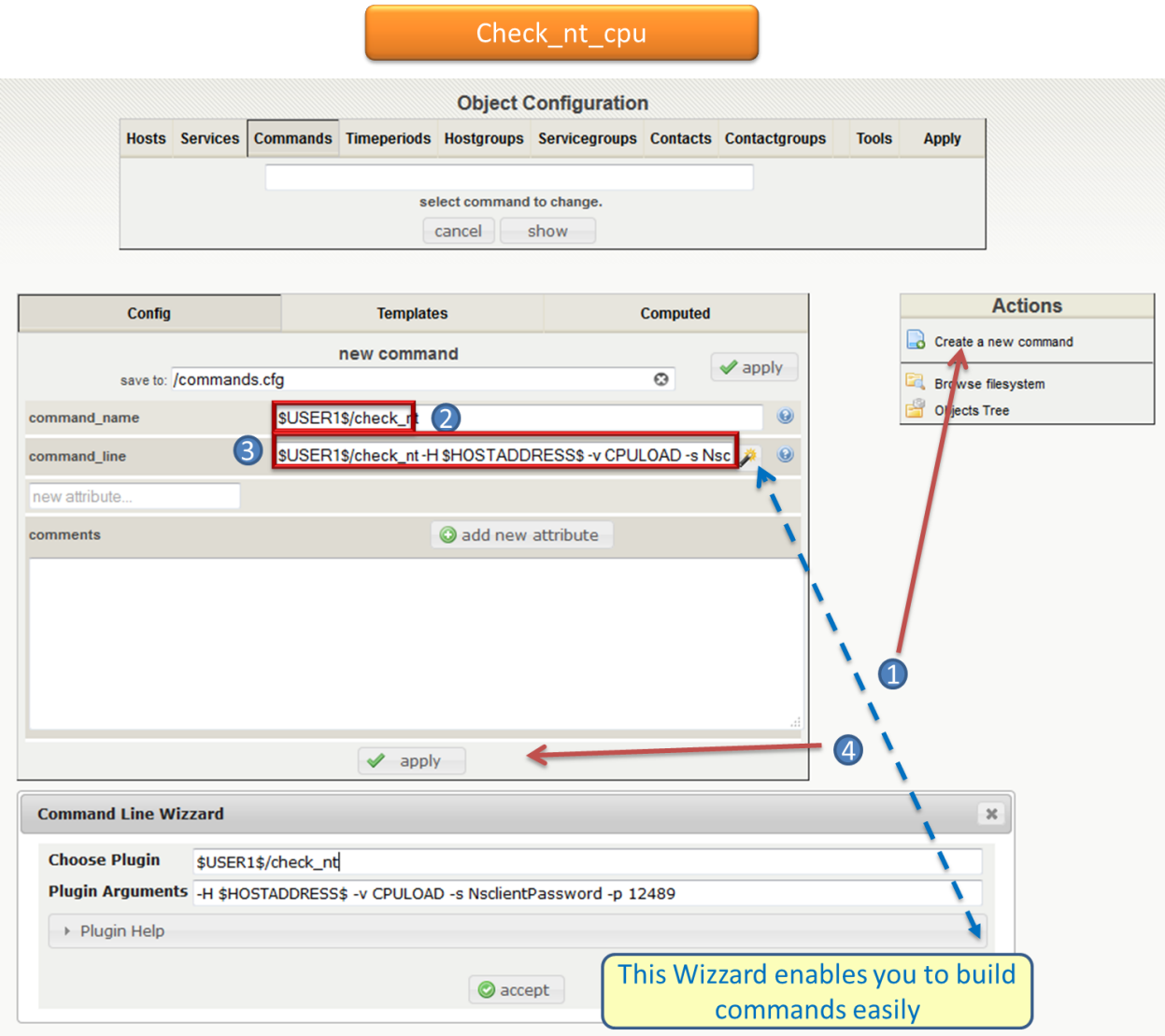


The command is now present in Naemon configuration. We can now associate it to a service.

check_nt_cpu

We will add a “command / plugin ” check_nt_cpu that will be used by the cpu use service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_nt_cpu
- Enter the following command line $USER1$/check_nt -H $HOSTADDRESS$ -v CPULOAD -s NsclientPassword -p 12489
- Save and click on apply

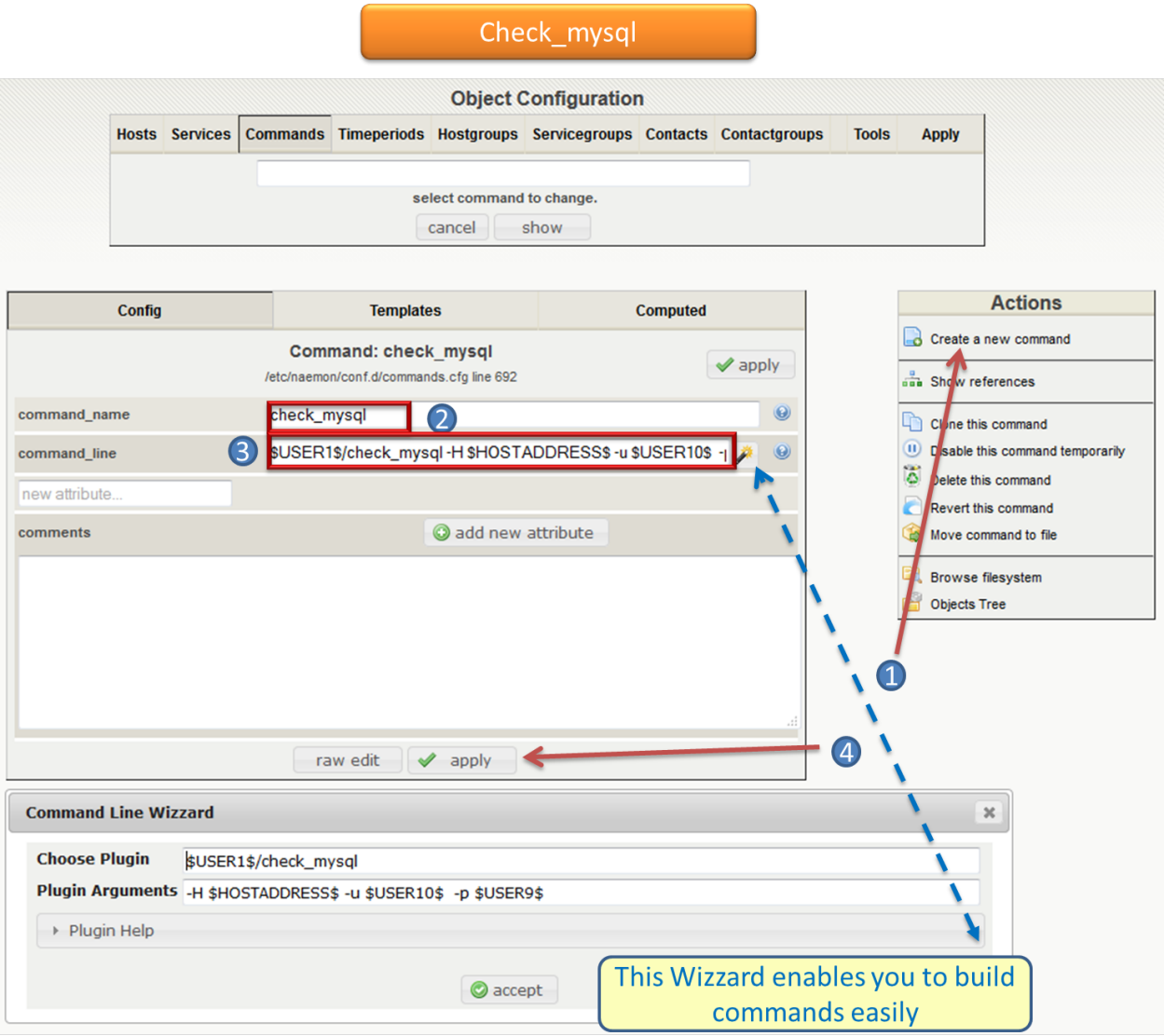


The command is now present in Naemon configuration. We can now associate it to a service.

check_mysql

We will add a “command / plugin ” check_mysql that will be used by mysql service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_mysql
- Enter the following command line : $USER1$/check_mysql -H $HOSTADDRESS$ -u user -p Password
- Save and click on apply

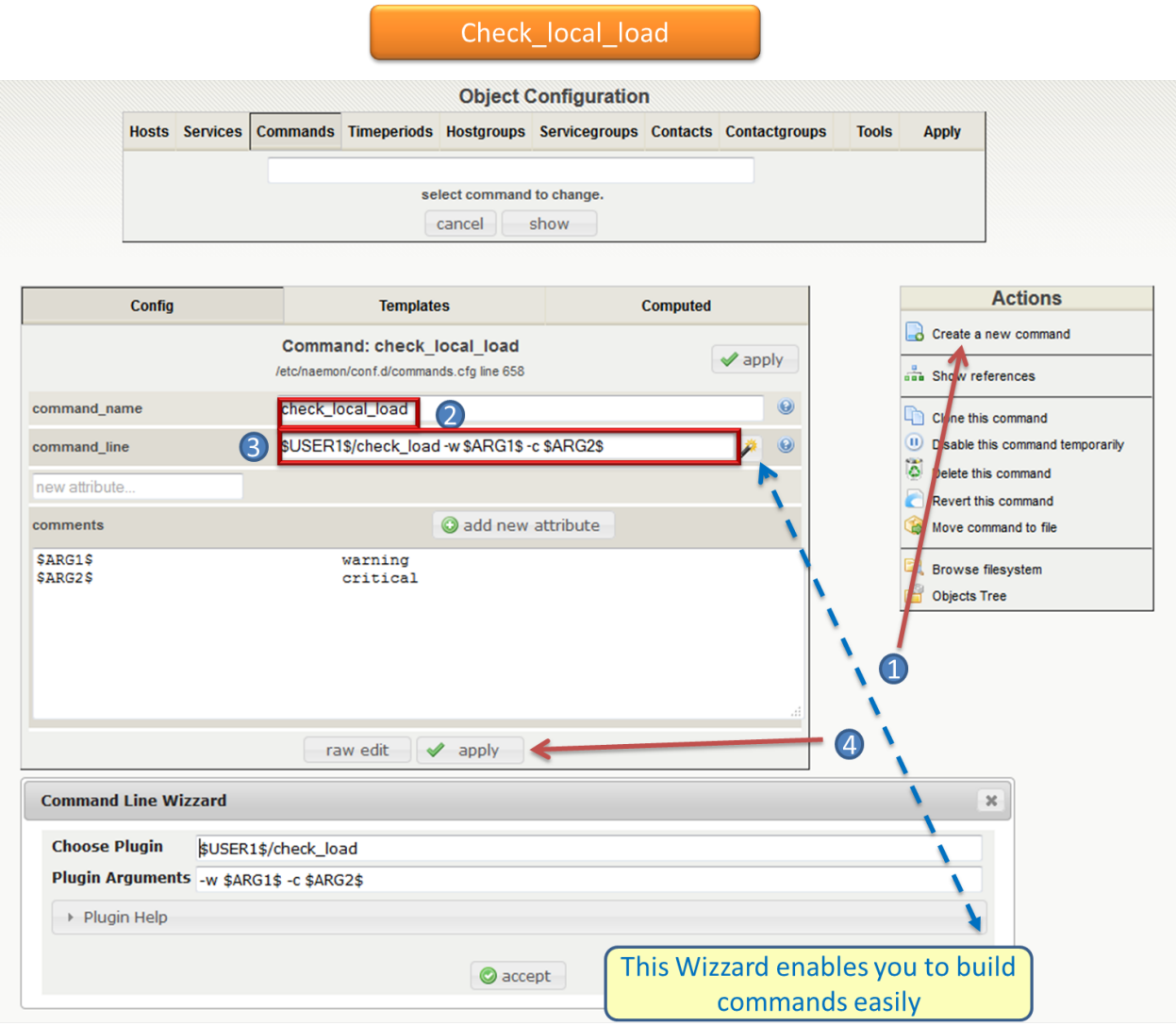


The command is now present in Naemon configuration. We can now associate it to a service.

check_local_load

We will add a “command / plugin ” check_local_load that will be used by Current Load service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_local_load
- Enter the following command line $USER1$/check_load -H $HOSTADDRESS$ -w 5.0,4.0,3.0 -c 10.0,6.0,4.0
- Save and click on apply

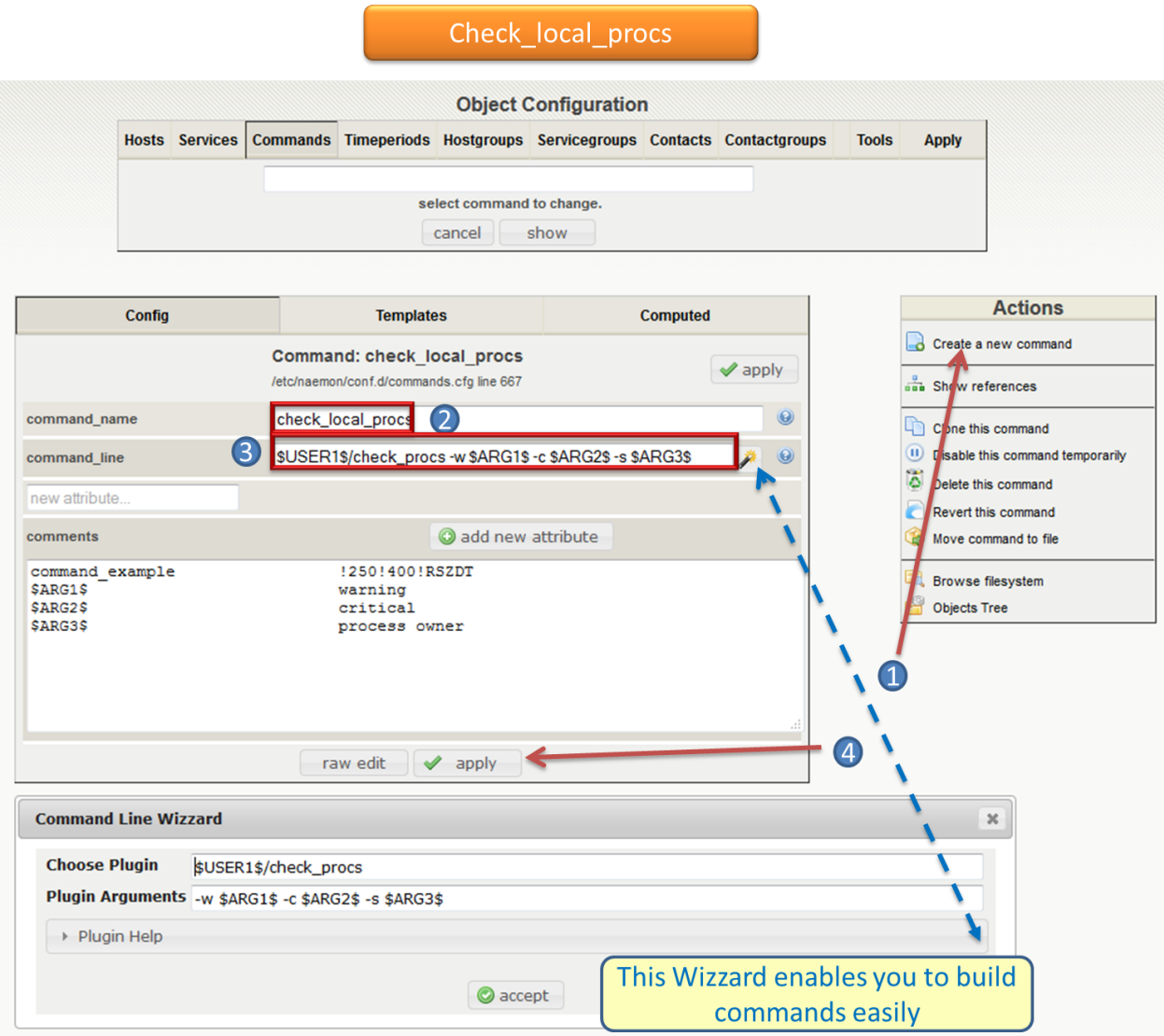


The command is now present in Naemon configuration. We can now associate it to a service.

check_local_procs

We will add a “command / plugin ” check_local_procs that will be used by Total Processes service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_local_load
- Enter the following command line $USER1$/check_procs -w 250 -c 400 -s RSZDT
- Save and click on apply

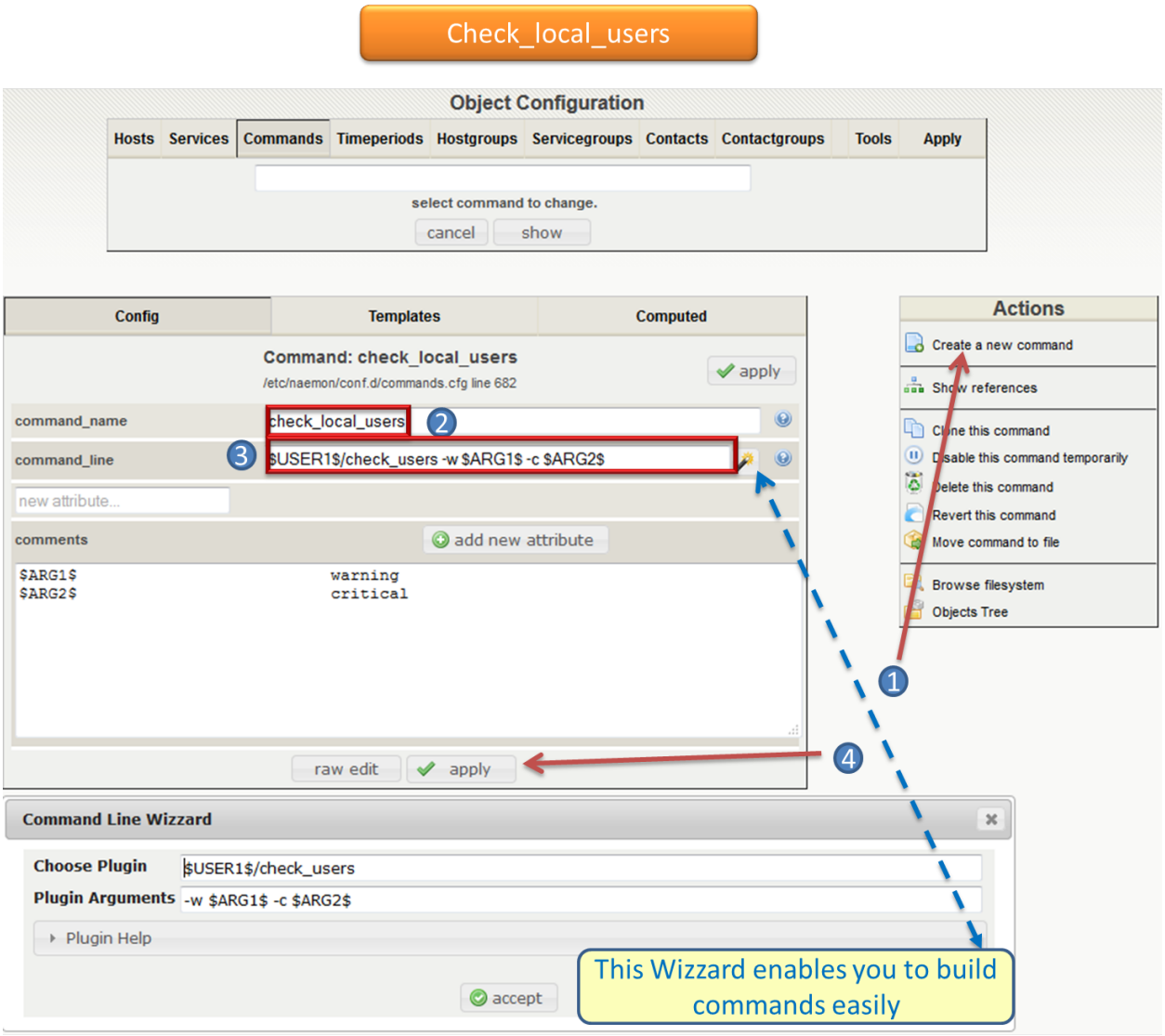


The command is now present in Naemon configuration. We can now associate it to a service.

check_local_users

We will add a “command / plugin ” check_local_users that will be used by Current users service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_local_users
- Enter the following command line $USER1$/check_users -w 20 -c 50
- Save and click on apply

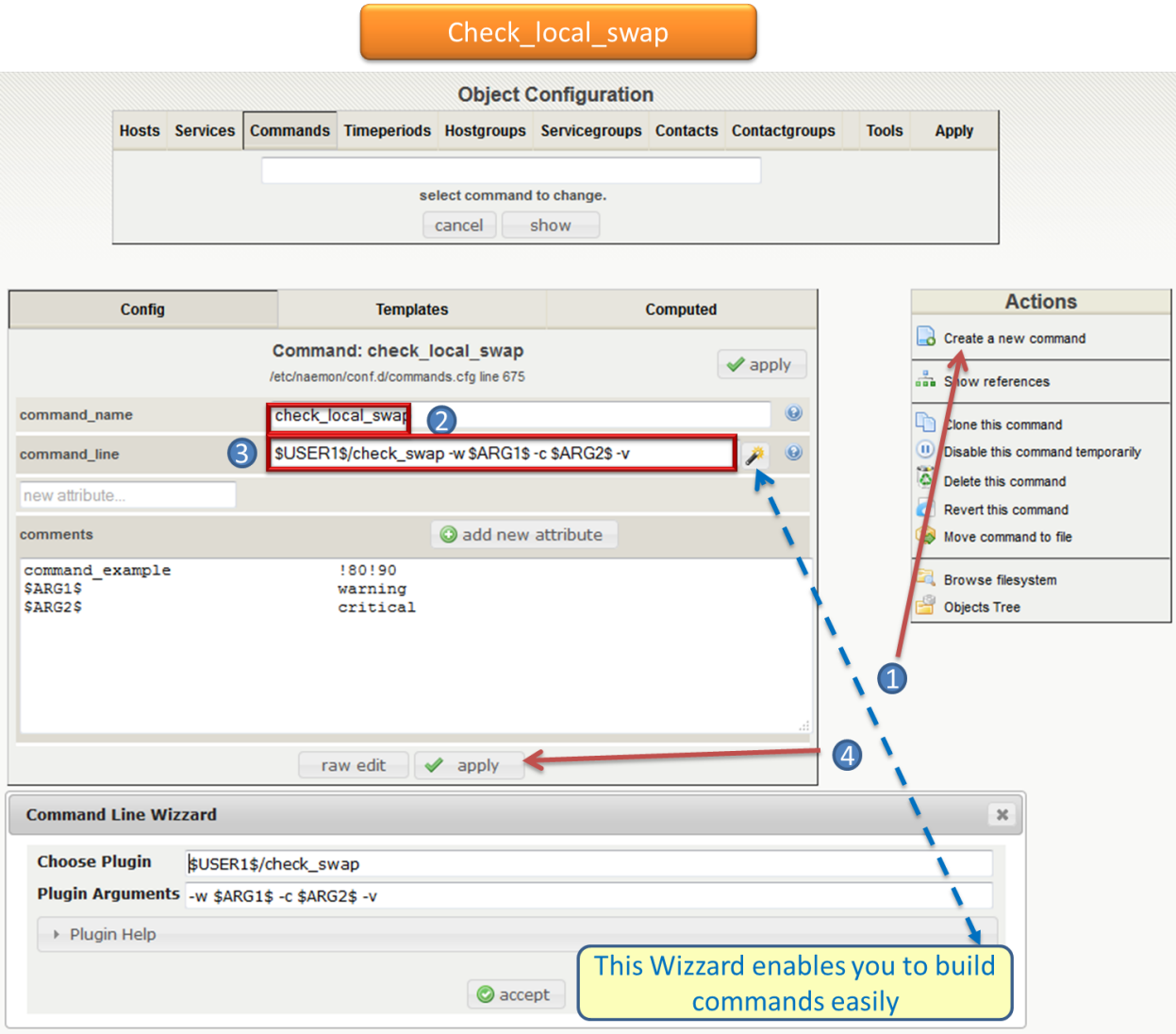


The command is now present in Naemon configuration. We can now associate it to a service.

check_local_swap

We will add a “command / plugin ” check_local_swap that will be used by swap usage service we will create for our “host” .

- Config Tool Menu / Object Configuration / Commands
- Enter the name of the command check_local_swap
- Enter the following command line $USER1$/check_procs -w 20 -c 10
- Save and click on apply



The command is now present in Naemon configuration. We can now associate it to a service.

6.5.2 Add a service

We will add a service “system start” to find out how long the system is started, to oversee our “host ”.

System Start

We will go through the Naemon Setup menu Config Tool > Object settings > Services.

- Completing the “system start” Service Description
- Enter the host name S34XXXXXXX
- Choose Systeme_Start service model
- Add a contact group Supervisors
- Save and click apply

The service is now present in Naemon configuration.

CPU Use To know the CPU load

We will go through the Naemon Setup menu Config Tool > Object settings > Services.

- Completing the "cpu_use" Service Description
- Enter the host name S34XXXXXXX
- Choose Win-Cpu_Use service model
- Add a contact group Supervisors
- Save and click apply

The service is now present in Naemon configuration.

CURRENT Load To know the local load

We will go through the Naemon Setup menu Config Tool > Object settings > Services.

- Completing the "local_load" Service Description
- Enter the host name S34XXXXXXX
- Choose generic-service service model
- Add a contact group Supervisors
- Save and click apply

The service is now present in Naemon configuration.

CURRENT Users To know the numbers of users connected

We will go through the Naemon Setup menu Config Tool > Object settings > Services.

- Completing the "Current_Users" Service Description
- Enter the host name S34XXXXXXX
- Choose generic-service service model
- Add a contact group Supervisors
- Save and click apply

The service is now present in Naemon configuration.

6.5.3 Network status

Each monitored server consists of several services (DHCP - WINS - SQL - TINA etc ...). Each monitored service uses a command. To check a service on the server, take control of the server and start a NET START command line or open the Services management method

To monitor the McAfee status services , we create a template *TMP-McAfee_Services* that each host will be associated to McAfee_Service Setting the Service Template : *TMP-McAfee_Services*

- Name: *TMP-McAfee_Services*
- Service Description : McAfee_Services
- Service Model used : generic Service
- Command verification : check_nt_services
- Arguments: 'McAfee Framework Service!McShield McAfee!McAfee Task Manager!McAfee Validation Trust Protection Service'

McAfee_Service Definition

This service uses the command check_nt_services

- Command name : check_nt_services
- Command line: $USER1$/check_nt -H $HOSTADDRESS$ -v SERVICESTATE -s NsclientPassword -p 12489 -d SHOWALL -l $ARG1$,$ARG2$,$ARG3$,$ARG4$

Macro $ARG1$, $ARG2$, $ARG3$ ... match the arguments placed in the command. eg: "McAfee Framework Service!McShield McAfee!McAfee Task Manager!McAfee Validation Trust Protection Service'

Service : traffic (naemon) To know the traffic up and down from the NIC

- In the Config Tool / Services menu.
- Completing the description (eg traffic)
- Choose a service model (eg generice-Service)
- Select the check command : check_traffic
- Arguments : eth0!80!90!1
- Save and click on apply

The service is now present in Naemon configuration, we need to export it to apply config changes

Export Naemon Configuration Files Menu Config Tool/Object settings and then click Apply to save your change to disk, check your configuration changes, reload your monitoring core

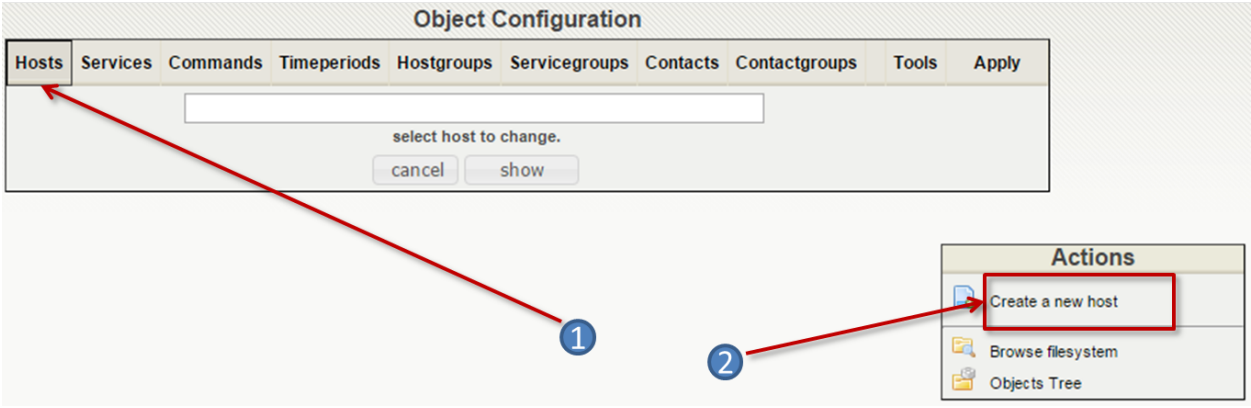
6.6 Host Definition

A host definition is used to define a physical server, workstation, device, etc. that resides on your network.

6.6.1 Add a host

We will add a Windows server-based host in our Naemon configuration. For editing , go through :

1. Naemon Setup menu **Config Tool ==> Object settings ==> Hosts.**
2. **Create a new host.**



3. Make changes and click on **Apply.**

- Host name (“host name”) : S34XXXXXXX
- Host Description (“Alias”) : Web Server
- IP address / DNS : 10.xx.xxx.xxx
- Add a template (template) associated with this host . A Template is the centralization of characteristics common to a machine.
- Then select the template : Servers-Win2k3
- Fill the Control Period : 24x7
- Add a contact group : Supervisors
- Save and click on apply

At this point, the host www is in the Naemon configuration

We will now export the new configuration changes to Naemon by clicking on Apply. View diff of changed files compares files

- Save changes to disk dumps the configuration .
- Check your configuration checks changes if there is no error
- Reload your monitoring core recover Naemon .

6.6.2 access , authentication and authorization management

6.6.3 Create a host

- Click on the Config Tool menu/Object Configuration/Contact
- Click Create a new Contact

Fill the fields according to your criteria (full name , Alias / Login , generic contact, Email, Allow can_submit _commands)

6.6.4 User Configuration

- Click on the Setup menu Tool/User Configuration
- Select the account in the username field
- Create a password and confirm, then click “SAVE”

  Editing the cgi.cfg file

By default, a contact will be entitled to access objects which it is associated , make change according to your needs :

- show_context_help=0
- use_authentication=1
- use_ssl_authentication=0
- default_user_name=nagiosadmin
- authorized_for_system_information=nagiosadmin,hotline,
- authorized_contactgroup_for_system_information=
- authorized_for_configuration_information=nagiosadmin
- authorized_contactgroup_for_configuration_information=
- authorized_for_system_commands=nagiosadmin
- authorized_contactgroup_for_system_commands=
- authorized_for_all_services=nagiosadmin,hotline
- authorized_contactgroup_for_all_services=
- authorized_for_all_hosts=nagiosadmin,hotline
- authorized_contactgroup_for_all_hosts=
- authorized_for_all_service_commands=nagiosadmin
- authorized_contactgroup_for_all_service_commands=
- authorized_for_all_host_commands=nagiosadmin
- authorized_contactgroup_for_all_host_commands=
- authorized_for_read_only=
- authorized_contactgroup_for_read_only=
- refresh_rate=90
- escape_html_tags=1
- action_url_target=_blank
- notes_url_target=_blank
- lock_author_names=1
- host_unreachable_sound=../media/critical.wav
- host_down_sound=../media/critical.wav
- service_critical_sound=../media/critical.wav
- service_warning_sound=../media/warning.wav
- service_unknown_sound=../media/unknown.wav

CHAPTER 7

Nagvis Configuring Overview

Nagvis is the most advanced mapping module. It is both flexible, scalable and consider the under cards. Nagvis will help make connections between cards, insert background images, icons or pictures ..

7.1 Prerequisites

To create or edit maps you need Visio diagram as design software . To upload file you need to connect securely to the shared resources on the server (Ex. Bitvise TBM) . This tool allows accessing remote files over an encrypted connection as if they were files on a local drive, without requiring SFTP or SCP file transfers.

7.2 Create a map

The map must be like on the model below :

7.3 Integration of the map in Nagvis

Opne Bitvise. The diagrams are stored in /usr/local/NagVis/share/userfiles/images/maps/

7.4 Create the map in Nagvis

Go to Otions Menu / Manage maps In Create a map complete the field as follows :

- Map Name (site_name)
- Map icon : std_small
- Background Map . Select from the pull down menu, the map you just downloaded on the Linux server MAP_Name.png .

7.5 Adding elements to map

Now that we have our map , we have to add elements to our map .

We already have a default panel item with Nagvis :

- Icon
- Line
- Special

The icons and lines offer us to link the item you will choose :

- A host
- A service
- A group of hosts (hostgroup in Naemon)
- A service group (servicegroup in Naemon)
- A map (icon summarizing the overall status of the card and serving as a link to there)

We will add two elements to our map : service and host. To add proceed as follows: In the Edit menu a map / icon Add / Host A window opens the Create Object. In the host_name field, select the server name and confirm your choice by clicking on the "Save" button. The mouse pointer changes to a + sign , select the server to modify.

You can unlock the object added , while clicking , support and move the object. To enable this feature , go to the menu Edit Map , Select Lock / Unlock All. Go to the Edit Map manu / Add icon / Service

A window opens **Create Object**. In the host_name field, select the server name , then in the service_description field, select the service name and confirm your choice by clicking on the "Save" button. The mouse pointer changes to a + sign , click on the server to modify.

7.6 Modify Object

select the MAP you want to modify. Move your mouse to the object you want to change , once in edit mode :

- Modify : In the case you have to change to the server name.
- Delete : If you delete the server or stop his monitoring.

7.7 Authentication / Authorization

Account management and access go through two modules:

7.8 Managing Users :

In the Personal menu / Manage Users menu, we can:

- Create a user with password initialization .
- Assign a role to a user
- Delete user
- Change password

7.9 Role Management :

Asigned different roles what the user is able to perform . Permissions are hard-coded into Nagvis .

In the Personal menu / Manage Roles menu, we can:

- Create Role
- Set permissions for a role

An authorization is composed of three elements:

- Module: The different modules (Map , Automap , Rotation , ...)
- Action: The action can (view, edit , delete, ...)
- Object : The different object (A Particular user , map, ...)

CHAPTER 8

Centralize Windows logs with CACTI

When it comes to management, monitoring , one of the major commandments is based on the importance and attention that must be given to the logs. This is a unique source of information to have on hand in order to exploit and thus back to the source of the problem. To support its analysis and its exploitation, we will centralize event logs. This requires that all messages are transmitted to a central server Cacti with syslog plugin, an rsyslog server. " Rsyslog " centralizes the various event logs on the monitoring server . We can quickly and efficiently locate these failures on a network.

8.1 Windows client installation

We will use a "syslog" agent which will allow us to send traps compatible with RFC 3164 standard syslog format.

8.2 Prerequisites

Must be installed

- Microsoft.Net framework 2.
- El2esl

Go to All Programs and launch el2sl Configurator. This window opens :

- **gate** , enter the monitoring server address. Leave the default port
- You can change for each log , record types that will be sent to the server according to their order of importance (critical [2] Warning [4] ...)
- Click "close" .

8.3 Create a rule deletion

To create rules deletion, click on the red cross to the right of the log and then enter a name for your rule .

For example, to delete all events from the **naemon** server, just click on the red cross and complete the following steps. Here we remove all UDP connection messages from the monitoring server . There it's finished. Well logs !

8.4 Management of authentication and access permissions

Now you want to configure Cacti to provide authentication and additional permissions . This configuration is done from the web console cacti . We want to restrict viewing an account on the [Console] and [syslog] tab . In the Utilities / User Management menu, click add to create our first account.

- Then fill in the [User Name], [Full Name] .
- Add a password (X2) . Enabled Check the box to activate the account .
- In the [Realm permissions] check the boxes [Console Access] and [Plugin - > Syslog User] .

CHAPTER 9

Fusioninventory Client Installation

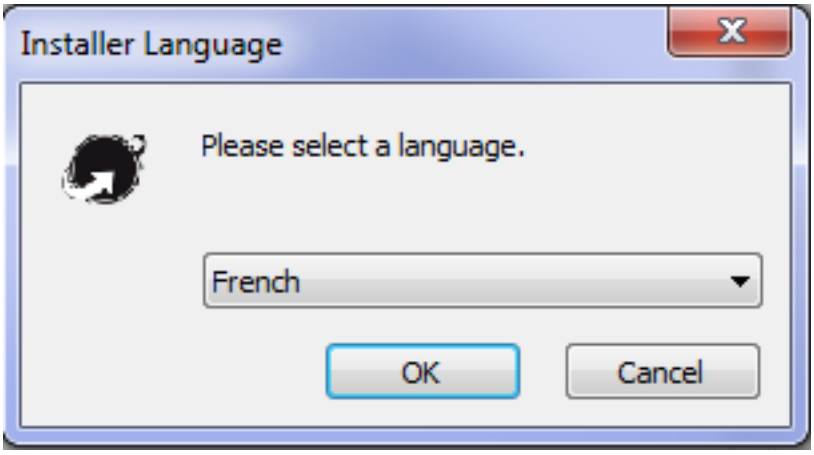
9.1 Windows Client

- Select the version to install for your architecture :

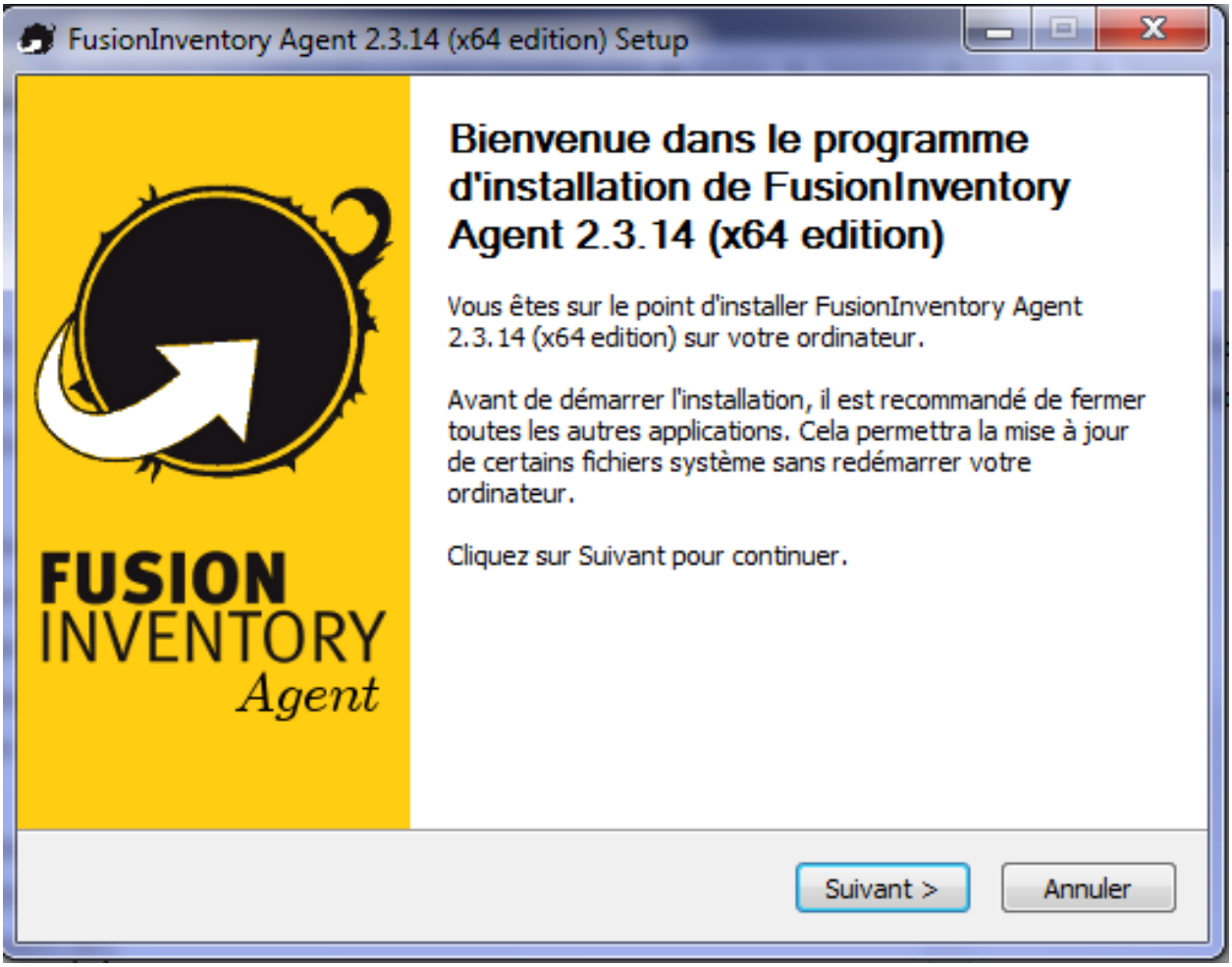
2.3.14

fusioninventory-agent_windows-x64_2.3.14-portable.exe	22/12/2014 16:52	5,885 Mo	1156	f442b8dddabc5ad3a17683cba53e2b3a
fusioninventory-agent_windows-x64_2.3.14.exe	22/12/2014 16:54	7,862 Mo	5730	9f83ea6d5a9c97370c3f8f83a255ccb3
fusioninventory-agent_windows-x86_2.3.14-portable.exe	22/12/2014 16:56	5,662 Mo	583	21f32fdbdbdcf5ca075daee510fc49dd
fusioninventory-agent_windows-x86_2.3.14.exe	22/12/2014 16:57	7,575 Mo	9943	64c0b8ba39b1514d82f1590a027a54af
fusioninventory-agent_windows_installer_en.html	22/12/2014 16:49	41,843 ko	1032	089e6715367ff6608abfe28017455304

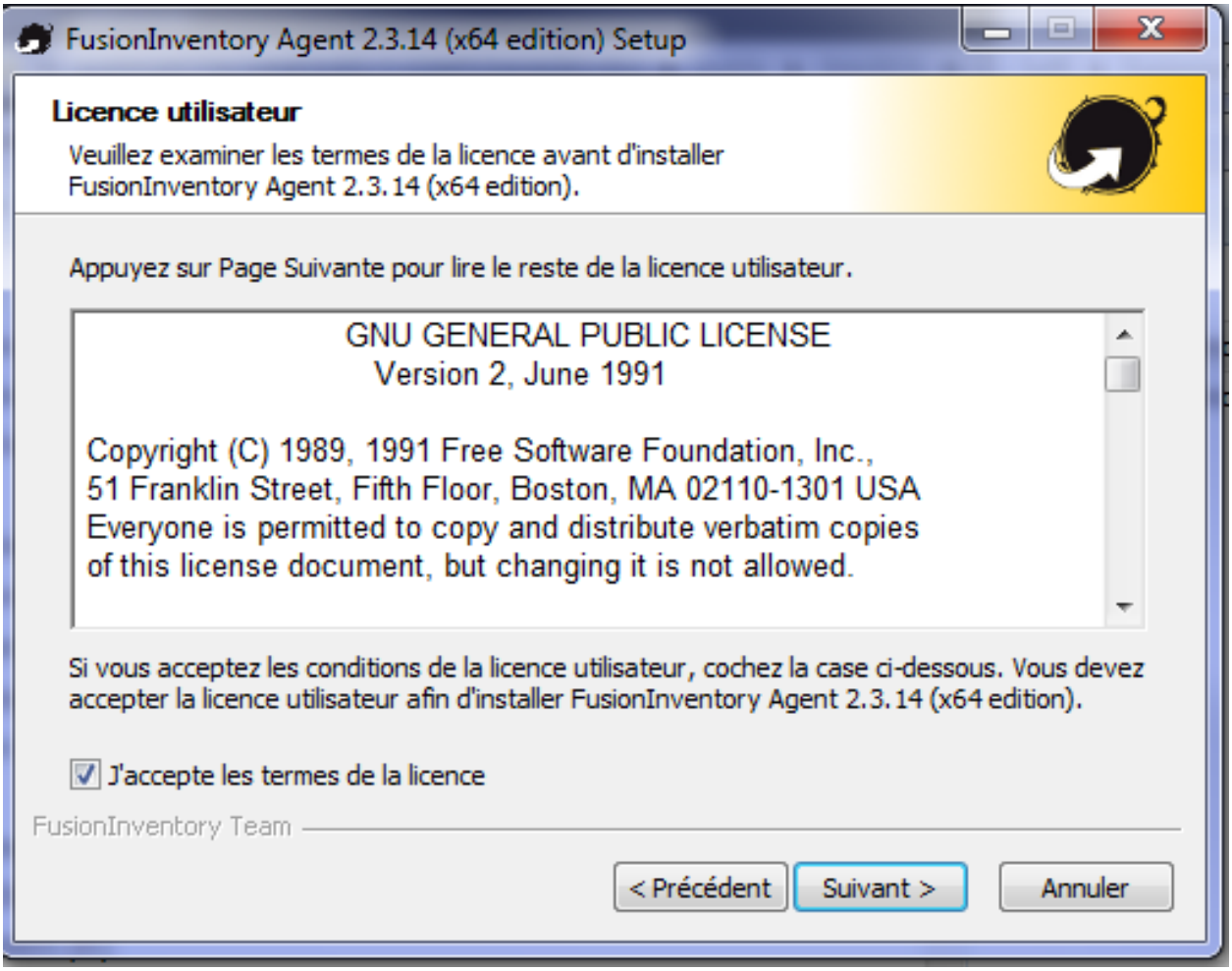
- The installer detects the system language installed . Leave blank and click OK.



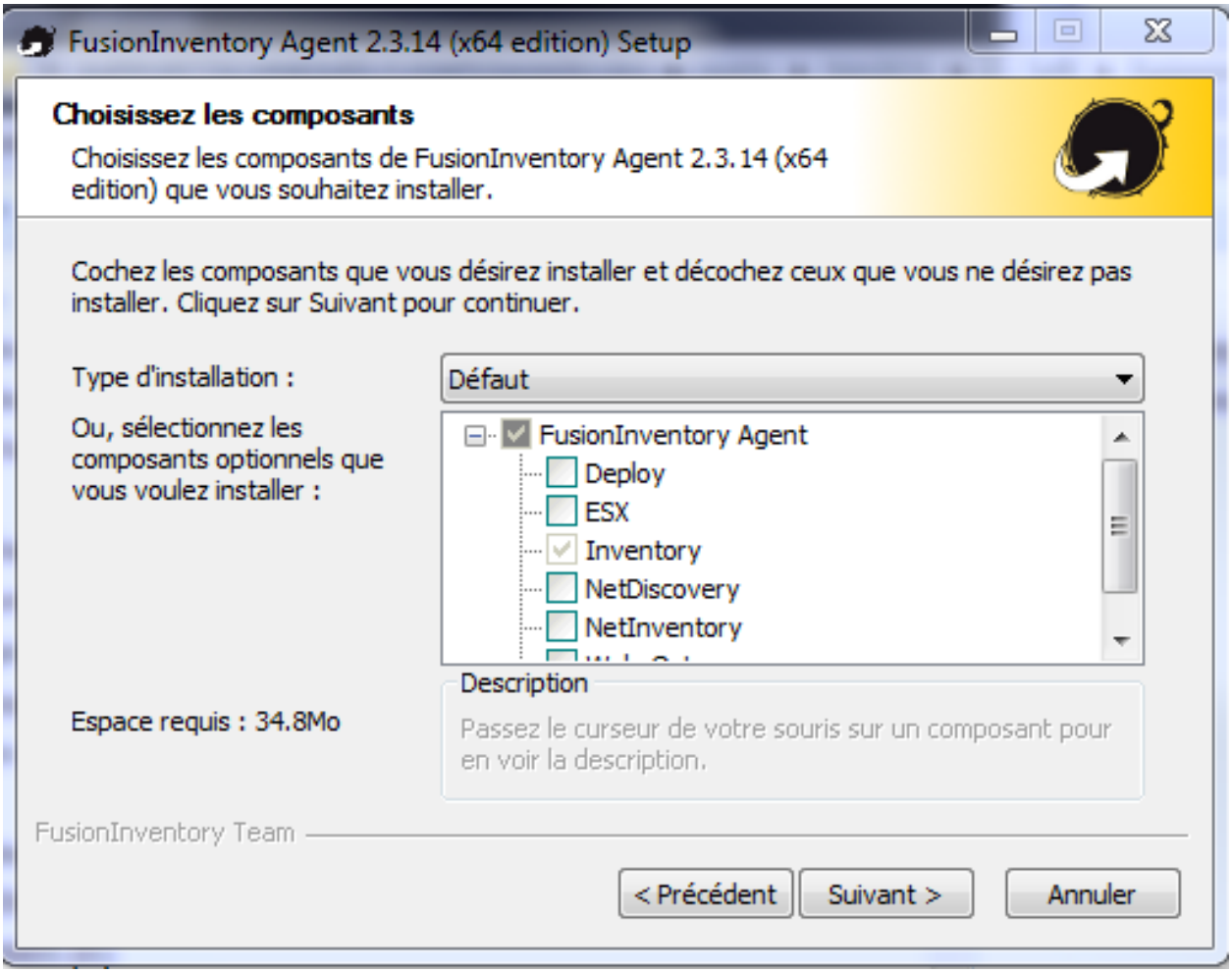
- Click Next



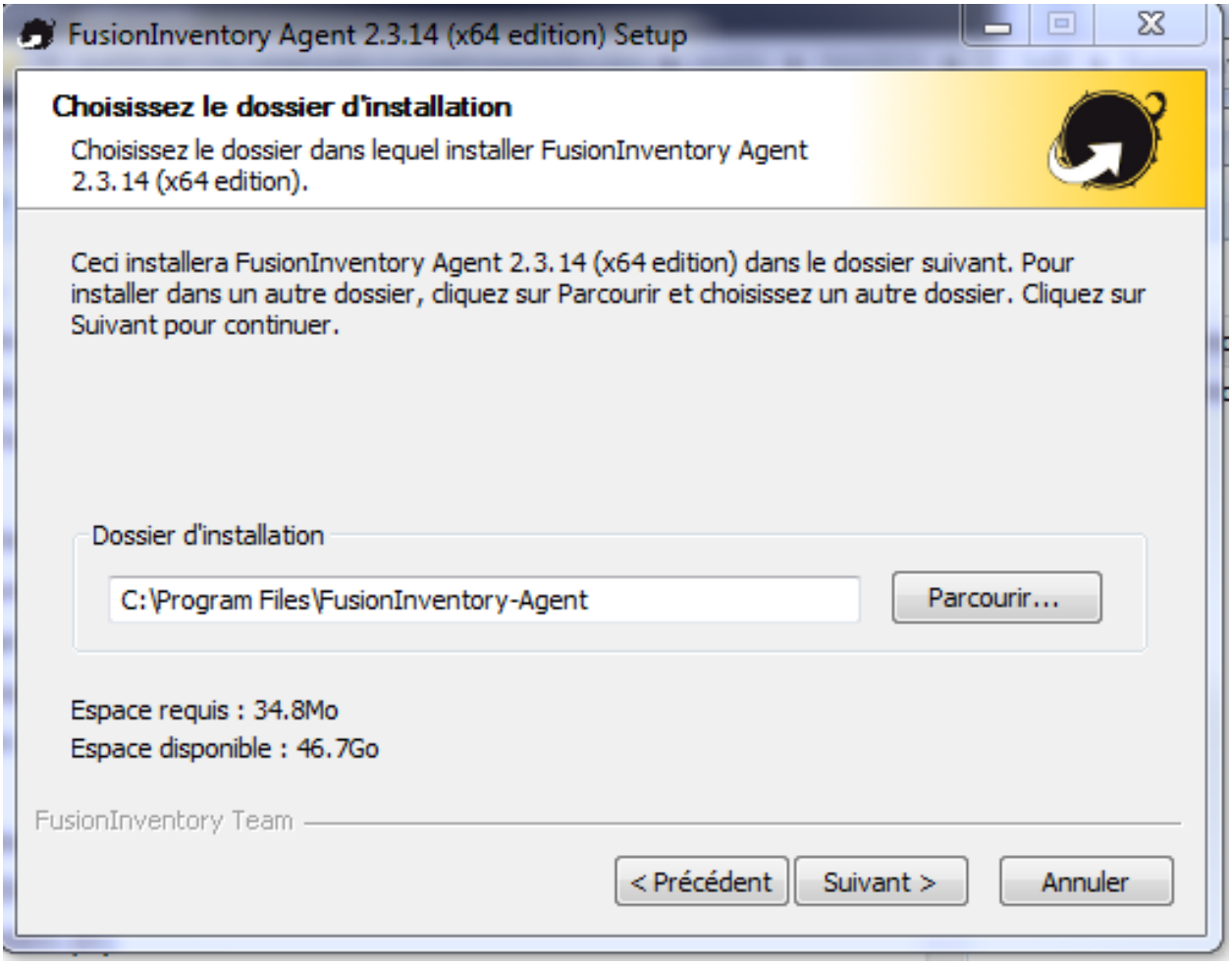
- Agree with the license terms and click Next


FusionInventory Agent 2.3.14 (x64 edition) Setup
Licence utilisateur
Veuillez examiner les termes de la licence avant d'installer FusionInventory Agent 2.3.14 (x64 edition).
Appuyez sur Page Suivante pour lire le reste de la licence utilisateur.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Si vous acceptez les conditions de la licence utilisateur, cochez la case ci-dessous. Vous devez accepter la licence utilisateur afin d'installer FusionInventory Agent 2.3.14 (x64 edition).
J'accepte les termes de la licence
FusionInventory Team
< Précédent
Suivant >
Annuler

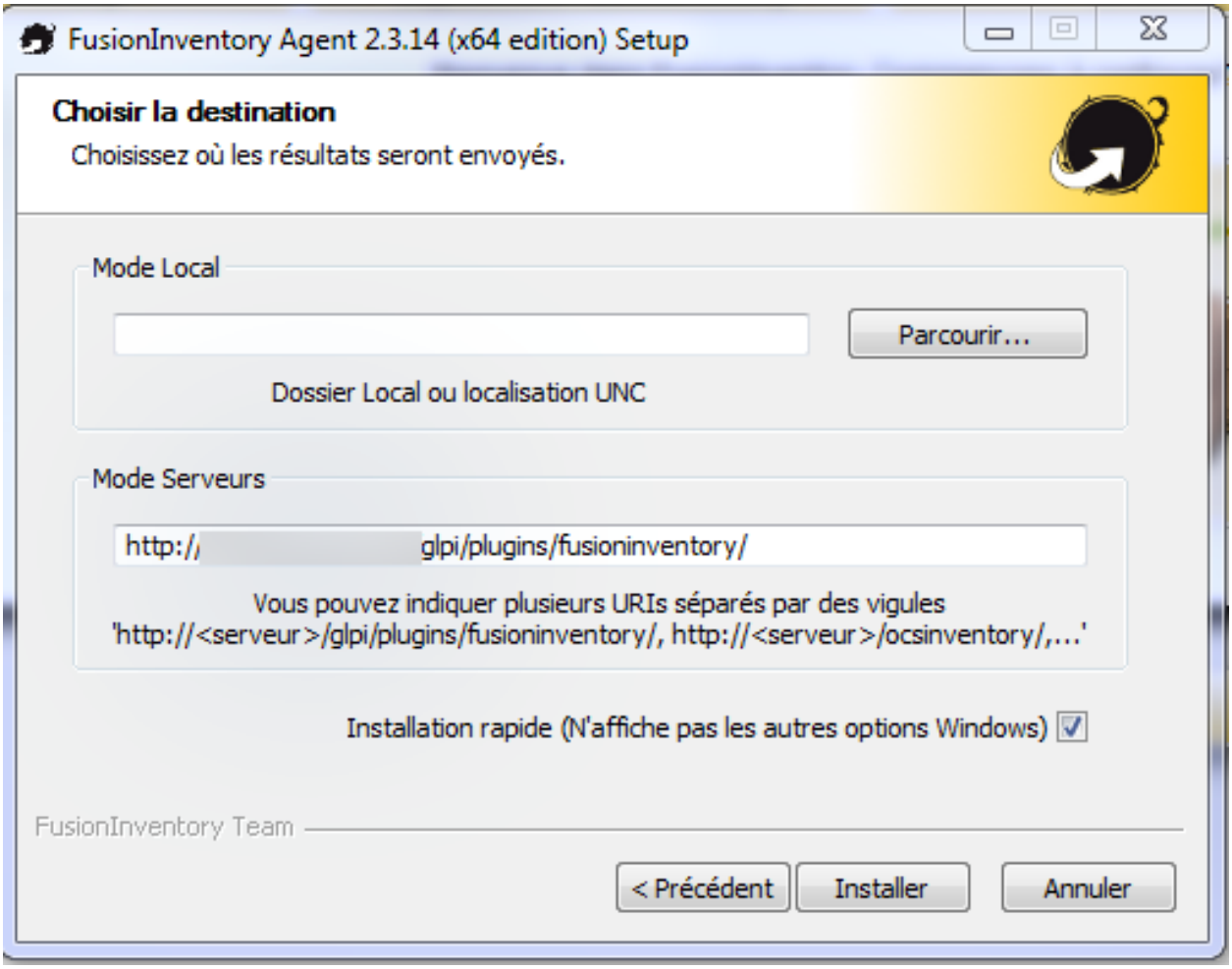
- Click Next


FusionInventory Agent 2.3.14 (x64 edition) Setup
Choisissez les composants
Choisissez les composants de FusionInventory Agent 2.3.14 (x64 edition) que vous souhaitez installer.
Cochez les composants que vous désirez installer et décochez ceux que vous ne désirez pas installer. Cliquez sur Suivant pour continuer.
Type d'installation :
Défaut
Ou, sélectionnez les composants optionnels que vous voulez installer :
FusionInventory Agent
Deploy
ESX
Inventory
NetDiscovery
NetInventory
Description
Passez le curseur de votre souris sur un composant pour en voir la description.
Espace requis : 34.8Mo
FusionInventory Team
< Précédent
Suivant >
Annuler

- Click Next



- In the field Servers Mode, enter the IP @ of GLPI server : http://your_ip_address/glpi/plugins/fusioninventory/



- Click Install , Next, and Close.

fusioninventory-agent_windows-7.png

9.2 ESX Client

FusionInventory can contact a ESX/ESXi/vCenter serveur using the VMware SOAP API. It will identify the ESX server and the associated virtual machine. At the end, it will push XML inventory of the machines to the server. Do the following commands, as root :

```
# fusioninventory-esx --user root --password 'password' --host @IP --directory /tmp
# fusioninventory-injector -v --file /tmp/HOSTNAME-2013-11-04-07-13-32.ocs -u http://your_ip_address/
```

9.3 GLPI Console

It does take a little bit more time, we could take a cup of coffee and then we can see the machine appear in GLPI inventory . Some alerts depending on the criticality threshold in Naemon to automatically trigger the creation of incident tickets in GLPI .

CHAPTER 10

Naemonbox Architecture

10.1 Distributed Monitoring

High availability and load balancing out-of-the box

10.1.1 Classification availability

The classification of systems in terms of availability commonly leads to 7 classes, not taking into account class (system available 90 % of the time , and therefore unavailable more than one month a year) to the ultra class (available 99.99999 % of the time and therefore unavailable only three seconds per year) : these classes are the number 9 in the percentage of time that the class systems are available

Type	Downtime (per year)	percentage availability	class
unmanaged	50,000 (34 days , 17 hours and 20 min)	90 %	1
managed	5000 (3 days , 11 hours and 20 min)	99 %	2
well managed	500 (8:20 minutes)	99,9 %	3
fault tolerance	50 (a little less than an hour)	99,99 %	4
high availability	5 minutes	99,999 %	5
very high availability	0,5 (30 secondes)	99,9999 %	6
high critical availability	0,05 (3 secondes)	99,99999 %	7

10.1.2 Introduction

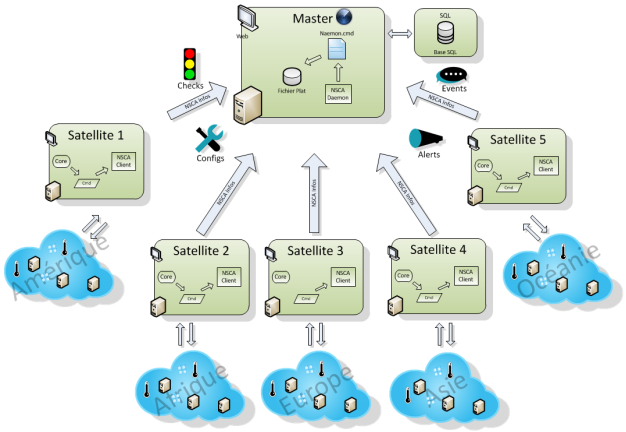
A server can fail, and so does the network. That's why you can (should) define multiple processes as well as spares in the Naemonbox architecture. Monitoring complex environments has never been easier. Activate the built-in cluster feature and you can begin configuring your high availability and distributed monitoring setup.

10.1.3 High availability clusters

In Naemonbox, clustered instances assign an 'active zone master'. This master writes to the Mysql database and manages configurations, notifications and check distribution for all nodes. Should the active zone master fail, another instance is automatically assigned to this role. Furthermore, each instance carries a unique ID to prevent conflicting database entries and split-brain behaviour. With continuous synchronisation of program states as well as check results, this design gives Naemonbox the edge over active-passive clusters using Pacemaker in Nagios. It also makes fail-safe monitoring much easier to scale in Naemonbox

10.1.4 Distributed monitoring

Where operations are dispersed across multiple sites, Naemonbox enables distributed monitoring too. Thanks to Naemonbox cluster zoning, satellite instances can be demarcated into their own secure zones to report to a central NOC. Satellites can be simple checkers or fully equipped with local MySQL database, user interface and other extensions too. Replication can be isolated to occur only between the master zone and each individual satellite, keeping satellite sites blind to one another. If a satellite goes rogue, check results are saved for retroactive replication access once the connection is restored.

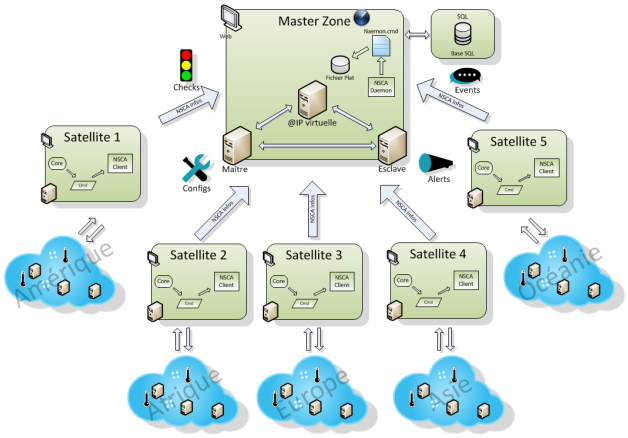


10.1.5 Distributed, high availability monitoring

Combine high availability clusters with a distributed setup, and you have a best practice scenario for large and complex environments. Satellite nodes can be scaled to form high availability clusters and cordoned off into secure zones. Load balancing and replication within them can be managed by an active zone instance to reflect different levels of hierarchy. An instance in the satellite zone can be a simple checker, sending results to the active satellite node for replication and display on a local interface. In turn the active satellite node can relay results to the NOC master zone for global reports.

Important: TODO Put all of the Naemonbox supervisory parameters with formulas and examples.

10.1.6 Diagrams



CHAPTER 11

Troubleshooting

11.1 FAQ - Naemonbox troubleshooting

11.1.1 FAQ Summary

Naemonbox users, developers, administrators possess a body of knowledge that usually provides a quick path to problem resolutions. The Frequently Asked Questions questions are compiled from user questions and issues developers may run into.

Have you consulted at all the resources available for users and developers.

__Before posting a question to the forum:__

- Read the through the Getting Started tutorials
- Search the documentation wiki
- Use this FAQ section.
- Bonus: Update this FAQ if you found the answer and think it would benefit someone else

Doing this will improve the quality of the answers and your own expertise.

CHAPTER 12

Indices and tables

- genindex
- modindex
- search