

Formation SIO

Supervision des Réseaux et des Équipements

1. Introduction

Buts de la supervision

- ✓ Élément clé de la fourniture de services aux organisations
- ✓ Déceler les dysfonctionnements le plus rapidement possible
- ✓ Si possible avant qu'ils ne se produisent (maintenance pro-active)
- ✓ Donner les moyens d'agir rapidement aux techniciens
- ✓ Fournir à posteriori
 - ✓ Des statistiques de disponibilité
 - ✓ Des informations temporelles et géographiques des incidents

Objectifs de la supervision

- ✓ Objectifs techniques
- ✓ Surveillance du réseau
- ✓ Surveillance de l'infrastructure
- ✓ Surveillance des machines,
- ✓ Objectif applicatif
- ✓ Surveillance des applications
- ✓ Objectifs de respect du contrat de services
- ✓ Surveillance du respect des indicateurs
- ✓ Taux de disponibilité
- ✓ Temps de reprise sur incident
- ✓ Objectif métier
- ✓ Surveillance des processus métiers et des fonctions de l'entreprise.

Fonctions de la supervision

- ✓ Surveillance du fonctionnement et des performances des équipements et des services
- ✓ Visualisation de l'état instantané des éléments surveillés
- ✓ Mémorisation des événements
- ✓ Visualisation des historiques
- ✓ Recherche d'événements passés
- ✓ Réalisations de statistiques, de graphes
- ✓ Résolution automatique des problèmes, passage en dégradé
- ✓ Alerte graduée des personnels techniques (escalation)

Objets techniques surveillés

- ✓ Machines
- ✓ Matériels d'électronique active
- ✓ Liens
- ✓ Services
- ✓ Matériels spécifiques
- ✓ En bref, tout ce qui possède une adresse (IP ou autre) et qui est accessible par le réseau
- ✓ Dont on a la propriété, la maîtrise, ou pas

Méthode de surveillance

✓ Externe

- ✓ Toujours possible sous condition d'accès au socket
- ✓ L'élément surveillé voit le superviseur comme un client
- ✓ Permet la surveillance au moins jusqu'à la couche 4 OSI
- ✓ Limites
 - ✓ Pare-feu intermédiaire
 - ✓ Pare-feu local
 - ✓ Droits d'accès au delà de la couche 4
 - ✓ Droits d'accès aux informations en principe non publiques

✓ Interne

- ✓ Implique la maîtrise de l'équipement
- ✓ Peut induire des problèmes de sécurité

Méthode de surveillance

✓ Externe

- ✓ Toujours possible sous condition d'accès au socket
- ✓ L'élément surveillé voit le superviseur comme un client
- ✓ Permet la surveillance au moins jusqu'à la couche 4 OSI
- ✓ Limites
 - ✓ Pare-feu intermédiaire
 - ✓ Pare-feu local
 - ✓ Droits d'accès au delà de la couche 4
 - ✓ Droits d'accès aux informations en principe non publiques

✓ Interne

- ✓ Implique la maîtrise de l'équipement
- ✓ Peut induire des problèmes de sécurité

Représentation de l'information

- ✓ Textuelle
 - ✓ Affichage de l'état ([ex1](#), [ex2](#))
 - ✓ Affichage d'informations complémentaires ([ex3](#), [ex4](#))
- ✓ Graphique
 - ✓ Diagrammes en fonction du temps ([ex5](#), [ex6](#))
 - ✓ Schémas logiques du réseau ([ex7](#), [ex8](#))
- ✓ Géographique
 - ✓ Sur image/photographie réelle ([ex9](#))
 - ✓ Sur plan ou carte géographique ([ex10](#), [ex11](#), [ex12](#))

Logiciels de supervision

- ✓ Produits Constructeurs
 - ✓ Hp Openview
 - ✓ Cisco Prime Unified Communications Management Suite
- ✓ Produits Propriétaires
 - ✓ IT Supervisor
 - ✓ NetView
 - ✓ OP Manager
- ✓ Produits Libres
 - ✓ Cacti
 - ✓ Nagios
 - ✓ MRTG
 - ✓ Centreon
- ✓ Distributions libres complètes
 - ✓ Shinken
 - ✓ ZeroShell
- ✓ ...

Critères de choix

- ✓ Coûts
- ✓ Type d'équipement supervisables
- ✓ Nombre d'équipements
- ✓ Mode de licences
- ✓ Protocoles utilisés
- ✓ Overload induit sur le réseau et les équipements
- ✓ Modularisation
- ✓ Associativité
- ✓ Évolutivité
- ✓ Documentation
- ✓ Facilité de mise en œuvre et d'exploitation
- ✓ Qualité des graphes et des cartes
- ✓ Fonctionnalités

Points communs

- ✓ Architecture d'exploitation : Client-Serveur
- ✓ Architecture de surveillance
 - ✓ Client-Server (mode passif)
 - ✓ Manager-Agent (mode-actif)
- ✓ Interface d'exploitation graphique Web
- ✓ Pages dynamiques (php, ajax)
- ✓ Centralisation possible des données et des évènements dans une base de données
- ✓ Utilisation du protocole SNMP

2. Nagios Niveau Basique

- ✓ Avantages
 - ✓ Gratuit
 - ✓ Nombre d'équipements illimité
 - ✓ Gère tous les type d'équipements et de SE
 - ✓ Modularité totale sous forme de plugins
 - ✓ Associativité avec tous les produits libres
 - ✓ Cacti, MRTG, Nagvis, Munin, Glpi...
 - ✓ Protocoles standard
 - ✓ SNMP, ICMP...
 - ✓ Architectures monolithiques ou distribuées
 - ✓ Très bien documenté
 - ✓ Communauté très nombreuse et très active
 - ✓ Alerte par mail, sms, serveur vocal
 - ✓ Exploitable sur pc, [iphone](#), [Androide](#), [Blackberry](#)
- ✓ Inconvénients
 - ✓ Réputé (à tort) difficile à configurer
 - ✓ Ne gère pas en natif les architectures distribuées
 - ✓ Ne gère pas le planning des astreintes des personnels techniques lors de l'escalation

Principes de base

- ✓ Nagios est un ordonnanceur de tâches (check)
- ✓ Configurable en mode texte à la base
- ✓ Exploité par interface Web interactive
- ✓ Gère le stockage d'historisation mais peut alimenter une base de données externe (fonction de broking)
- ✓ Permet de regrouper les machines et les services selon une infinité de combinaisons
- ✓ Propose des graphes, des rapports et des schéma assez rudimentaires
- ✓ Se couple avec de nombreux autres logiciels
 - ✓ Pour enrichir ses fonctionnalités
 - ✓ Pour faciliter son exploitation
 - ✓ Pour faciliter sa configuration
- ✓ Repose sur des logiciels très courants
 - ✓ Apache2
 - ✓ Mysql

Installation

- ✓ Sur une machine Debian Based (Ubuntu & co)
 - ✓ apt-get install nagios3 nagios3-doc
 - ✓ Installe par dépendance tous les logiciels nécessaires
 - ✓ Apache2
 - ✓ Php5
 - ✓ Gd
 - ✓ ...
 - ✓ Installe les plugins de base
- ✓ Sur une machine Linux autre (y compris Mac OS X)
 - ✓ Installer préalablement les logiciels indispensables
 - ✓ Récupérer les sources et les installer selon la procédure habituelle
- ✓ Sur une machine Windows
 - ✓ Installer le programme NagWin fourni par Nagios

Contexte de démonstration

- ✓ Entreprise « i15 »
- ✓ Deux sites
 - ✓ Paris
 - ✓ Moscou
- ✓ Liaison par ligne spécialisée louée
- ✓ Deux routeurs
- ✓ Un serveur web à Moscou
- ✓ Un commutateur à Paris
- ✓ Quelques serveurs à Paris (un Mac, un Windows, quelques Linux)
- ✓ [Schéma](#) et plan d'adressage

Configuration par défaut

- ✓ Nagios gère la machine sur laquelle il est implanté
- ✓ Il surveille
 - ✓ Le remplissage des disques
 - ✓ Le nombre de processus
 - ✓ La charge processeur
 - ✓ Le nombre d'utilisateurs
 - ✓ Les services http et ssh
- ✓ On visualise l'état via l'interface Web [Voir](#)
- ✓ On ne peut pas interagir
- ✓ Toutes les fonctionnalités sont présentes
- ✓ Il fonctionne en démon
- ✓ Il ne renseigne pas de base de données externe

Emplacement et rôle des fichiers de configuration

- ✓ Configuration de Nagios et son interface
 - ✓ [/etc/default/nagios3](#)
 - ✓ [/etc/nagios3/cgi.cfg](#)
 - ✓ [/etc/nagios3/nagios.cfg](#)
- ✓ Plugins de supervision
 - ✓ [/usr/lib/nagios/plugins/check_XXXX](#)
- ✓ Modèle d'exploitation des plugins (commandes)
 - ✓ [/etc/nagios-plugins/config/XXXX.cfg](#)
- ✓ Définition des machines, des groupes, des services...
 - ✓ [/etc/nagios3/conf.d/XXXX.cfg](#)
- ✓ Répertoire d'interactivité avec l'interface web
 - ✓ [/var/lib/nagios3/rw/](#)

Modification initiale

- ✓ Réglage de l'interface graphique
 - ✓ Éditer `/etc/nagios3/cgi.cfg`
 - ✓ Régler la fréquence de rafraîchissement de l'écran
 - ✓ `refresh_rate=5`
- ✓ Autoriser l'interactivité avec le serveur web
 - ✓ Éditer `/etc/nagios3/nagios.cfg`
 - ✓ `check_external_command=1`
 - ✓ Ajouter `www-data` dans le groupe `nagios`
 - ✓ `adduser www-data nagios`
 - ✓ Donner les droits d'écriture au groupe `nagios` donc à `www-data` sur `/var/lib/nagios3/rw/`
 - ✓ `chmod g+rwX /var/lib/nagios3/rw`
 - ✓ `chmod u+s /var/lib/nagios3/rw`
 - ✓ Redémarrer le démon `nagios3`
 - ✓ `/etc/init.d/nagios3 restart`
- ✓ Autoriser l'exploitation de tous les plugins, même ceux qui utilisent des commandes nécessitant les droits de root
 - ✓ `chmod u+s /usr/lib/nagios/plugins/*`

Architecture de la configuration des objets

- ✓ Un hôte (host) utilise un modèle (template)
- ✓ Un hôte est supervisé par une commande
- ✓ Un groupe d'hôtes (hostgroup) contient un ou plusieurs hôtes
- ✓ Un service utilise un modèle
- ✓ Un service est présent sur un hôte ou sur un groupe d'hôtes
- ✓ Un service est supervisé par une commande
- ✓ Une commande utilise un plugin
- ✓ Un hôte ou un service est géré à certaines heures (timeperiod) par un administrateur (contact) ou un groupe d'administrateurs (contactsgroup)
- ✓ Un groupe de services est un ensemble de couples hôte/service qui assurent généralement un processus métier ou un ensemble fonctionnel de l'entreprise
- ✓ [Schéma](#)

Groupes d'hôtes

- ✓ On peut assembler les hôtes selon les besoins
- ✓ Il faut pour des raisons pratiques créer des groupes
 - ✓ Par système d'exploitation (linux, mac, win2k3...)
 - ✓ Par service rendu (http, ssh, ftp...)
 - ✓ Par type de matériels (routeur, switch, serveurs...)
 - ✓ Par marque de matériels (switch cisco, switch 3com...)
 - ✓ Par surveillance à effectuer (mémoire, température, occupation disque...)
- ✓ Car une commande surveille un service rendu par un groupe d'hôtes

Groupes d'hôtes facultatifs

- ✓ On créera autant d'autres groupes que de besoin pour une vision synthétique des données
- ✓ Groupes géographiques (Paris, Moscou,
 - ✓ bâtiment B...)
- ✓ Groupes fonctionnels (liaison Paris-Moscou, dmz...)
- ✓ Groupe métiers (service clients, paye...)
- ✓ ...

Les commandes

- ✓ Une commande est une interface entre Nagios et le plugin
- ✓ Elle définit donc l'usage fait du plugin
- ✓ Un plugin peut être utilisé de différentes manières pour effectuer diverses surveillances
- ✓ Il est de coutume de définir l'ensemble des commandes pour un même plugin dans un fichier portant le nom du plugin
- ✓ Les commandes permettent de simplifier l'usage des plugins en masquant leur implémentation technique et en définissant des valeurs par défaut
- ✓ Les commandes utilisent des macros pour transmettre les paramètres au plugin
- ✓ Les macros sont préfixées et suffixées par \$ (\$ARG1\$)
- ✓ Pour définir la surveillance d'un service il faut regarder la définition de la commande afin de connaître la syntaxe et les paramètres à fournir
- ✓ Les commandes sont dans /etc/nagios-plugins/config/
- ✓ [Exemple : fichier des commandes du plugin check_ssh](#)

Fichiers de configuration

- ✓ Les fichiers de configuration sont libres
 - ✓ Ils peuvent contenir n'importe quels objets
 - ✓ Être in-homogènes
 - ✓ En quantité quelconque
 - ✓ Sous n'importe quel nom
 - ✓ Mais il est préférable de garder la structure par défaut et de l'enrichir
- ✓ Deux contraintes
 - ✓ Les fichiers doivent être suffixés .cfg
 - ✓ Stockés dans /etc/nagios3/conf.d/
- ✓ Exemples
 - ✓ [TimePeriod](#), [Contact](#), groupe de contacts
 - ✓ [HostTemplate](#), [Hôtes](#), [Groupe d'Hôtes](#)
 - ✓ [ServiceTemplate](#), [Services](#)

3. Réalisation basique pas à pas

- ✓ Création d'un hôte
 - ✓ [Objet créé dans le fichier hosts.cfg](#)
 - ✓ [Test1 Test2](#)
- ✓ Création de deux groupes d'hôtes
 - ✓ [Objets créés dans le fichier mesgroupes.cfg](#)
 - ✓ [Test](#)
- ✓ Création d'un service
 - ✓ [Objet créé dans le fichier messervices.cfg](#)
 - ✓ [Test1 Test2](#)
- ✓ Supervision du contexte exemple
 - ✓ Les fichiers de configurations
 - ✓ [Hôtes](#)
 - ✓ [Service ajouté](#)
 - ✓ [Services d'origine](#)
 - ✓ [Groupes](#)

4. Exploitation

- ✓ Ce qu'on peut voir
 - ✓ États [vue1](#) [vue2](#) [vue3](#)
 - ✓ Ok, Warning, Critical
 - ✓ Flapping
 - ✓ Historique
 - ✓ [Schémas](#)
 - ✓ [Rapports](#)
 - ✓ [Graphes](#)
- ✓ Ce qu'on peut faire
 - ✓ Lancer un contrôle
 - ✓ Activer/désactiver la surveillance
 - ✓ ...

5. Nagios niveau 2

Deux principaux problèmes dans ce qu'on vient de mettre en place

- ✓ On ne voit que l'aspect public des éléments surveillés
 - ✓ Réponse de l'hôte
 - ✓ Réponse des services
 - ✓ Ça ne permet pas de faire de la maintenance pro-active
 - ✓ Température
 - ✓ Remplissage disque
 - ✓ ...
- ✓ Si un élément tombe en panne
 - ✓ Tous ceux situés en aval ont l'air d'être en panne
 - ✓ Dans un réseau étendu, on va donc déranger des dizaines d'administrateurs à travers le monde
 - ✓ Graphiquement on ne voit pas d'où vient la panne

Voir la partie privée d'un équipement

✓ Quatre possibilités

1. Écrire des plugins qui lancent les commandes via ssh sur une machine distante
 - ✓ Authentification ssh par certificat
 - ✓ Très fort overload réseau et équipement pour l'ouverture du tunnel ssh
 - ✓ Rarement utilisée
2. Implanter des agents (NRPE ou NS-client) qui vont exécuter les plugins localement, à la demande de Nagios, et lui renvoyer les résultats [schéma](#)
 - ✓ Surveillance dite « active »
 - ✓ Nrpe sous linux et Mac OS X, NS-client sur Windows [schéma](#)
 - ✓ Ne permet pas de surveiller les architectures distribuées
 - ✓ Non utilisable sur des matériels d'électronique active
3. Implanter des démons (NSCA) sur les machines à surveiller qui déclencheront eux-même les contrôles et feront remonter l'information vers Nagios [schéma](#)
 - ✓ Pas de contrôle par Nagios
 - ✓ Diminue l'overload réseau
 - ✓ Permet de surveiller les architectures distribuées
 - ✓ Non utilisable sur des équipements d'électronique active
4. Utiliser des agents SNMP

Mise en œuvre de NRPE

- ✓ Installation du plugin NRPE sur le serveur Nagios
 - ✓ apt-get install nagios-nrpe-plugin
- ✓ Installation de l'agent NRPE et des plugins sur le client linux
 - ✓ apt-get install nagios-nrpe-server nagios-plugins
- ✓ Exploitation du plugin check_nrpe
 - ✓ Comme une commande normale mais :
 - ✓ Il y a au moins un paramètre
 - ✓ Qui indique le plugin à exécuter sur l'hôte distant

Construire les dépendances

- ✓ Notion de parent
 - ✓ Toujours du point de vue de Nagios
 - ✓ Un équipement peut avoir plusieurs parents
 - ✓ Le parent est indiqué dans la définition du « host »
 - ✓ [Exemple Test](#)
- ✓ Amélioration du schéma
 - ✓ On ajoute des icônes
 - Fournis par nagios
 - Des photos
 - ...
 - ✓ Dans un objet hostextinfo
 - ✓ La taille et le format des images sont imposés
 - ✓ [Exemple Test](#)
- ✓ Établissement des dépendances entre services (servicedependency)
 - ✓ Fixe les dépendances entre les services
 - ✓ Assez complexe à concevoir

6. Nagios niveau 3

- ✓ Écrire ses propres plugins
 - ✓ Le langage est libre
 - ✓ Un plugin peut accepter 0..n paramètres
 - ✓ Un plugin doit fournir un état de sortie (exit), parmi 4, après l'exécution d'un algorithme
 - ✓ 0 → OK
 - ✓ 1 → Warning
 - ✓ 2 → Critical
 - ✓ 3 → Unknown
 - ✓ Un plugin peut générer un message sur une ou plusieurs lignes.
Ce message sera inclus dans les vues détaillées de Nagios
 - ✓ Un plugin est stocké (en principe) dans /usr/lib/nagios/plugins
 - ✓ On peut le tester indépendamment de Nagios
 - ✓ Et vérifier la valeur de retour, sous linux avec : `echo $?`

Exemple trivial de plugin

- ✓ Objectif : compter le nombre d'utilisateurs potentiels, déclarés sur une machine, en comptant le nombre de répertoires dans /home/
- ✓ Le [programme](#) du plugin
- ✓ Test du plugin
- ✓ Définition de la [commande](#) associée au plugin
 - ✓ Stockée dans /etc/nagios-plugins/config/
- ✓ Intégration dans la surveillance de la machine hôte
 - ✓ Définition d'un nouveau [service](#) utilisant la nouvelle commande
- ✓ [Test1](#) [Test2](#)
- ✓ Intégration dans les autres machines
 - ✓ Copie du plugin sur chaque machine
 - ✓ Définition du service pour le groupe de machines

7. Nagios niveau 4

- ✓ Surveiller le fonctionnement des équipements réseau avec SNMP
- ✓ Fonctionnement similaire à NRPE
- ✓ Mais s'appuie sur un protocole normalisé
- ✓ Principe
 - ✓ Un agent dans l'équipement à surveiller
 - ✓ Qui récupère des données d'après leur OID (Object IDentifier)
 - ✓ Le tout est défini par la Mibs de l'équipement
 - ✓ Définie par des RFC
 - ✓ Comporte une partie propre à chaque constructeur
 - ✓ Cohérente même avec des constructeurs différents
- ✓ Activation sur l'équipement
- ✓ Nagios utilisera le plugin `check_snmp`

Mise en œuvre de SNMP

- ✓ Objectif : surveiller le débit d'un port en
- ✓ auto-négociation (10/100Mbps) d'un commutateur
- ✓ Création de la
[commande et des services associés à check_snmp](#)
- ✓ [Exploitation sur le commutateur du contexte](#)
- ✓ On pourrait améliorer les choses par la création d'un plugin synthétique qui :
 - ✓ Boucle sur N ports
 - ✓ Exécute le plugin check_snmp
 - ✓ Vérifie si tous les ports sont à 100Mbps
 - ✓ Liste les ports qui ne respectent pas cette contrainte

Exploitation de SNMP sur une machine

- ✓ Installation de l'agent SNMP : Installation du démon SNMP, du protocole et des Mibs
 - ✓ apt-get install snmpd snmp snmp-mibs-downloader
- ✓ Installation des mibs et configuration du démon snmpd
 - ✓ Éditer le fichier /etc/default/snmpd de la machine à surveiller
 - ✓ Ajouter la MIBS désirée
 - ✓ MIBS=UCD-SNMP-MIBS
 - ✓ Configurer le démon snmpd pour :
 - ✓ Qu'il écoute le réseau
 - ✓ AgentAddress udp:161
 - ✓ Lui indiquer la hiérarchie des Oid à rendre visible à tous
 - ✓ View systemonly included 1.3.6.1.4
 - ✓ Redémarrer le démon snmpd
- ✓ Exploitation avec le plugin check_snmp sur Nagios
 - ✓ Test avec snmpwalk
 - ✓ snmpwalk -v « version snmp » -c « communauté » « adresseIP » « oid »
 - ✓ Exemple : snmpwalk -v 1 -c public 192.168.1.50 1.3.6.1.4.1.2021.9.1.6.1
 - ✓ Donne la taille en Ko du premier point de montage (son nom est en 1.3.6.1.4.1.2021.9.1.3.1)
- ✓ Attention aux problèmes de sécurité car on publie des informations sensibles
- ✓ [Intégration](#) de la surveillance de la machine d'après la [commande](#) pré-établie snmp_disk

8. Plus loin avec Nagios

- ✓ Configuration graphique des objets
 - ✓ Plus simple (humm... à voir !), plus intuitif (humm... pas sûr !)
 - ✓ Structure stockée dans une base mysql
 - ✓ Génère les fichiers de configuration Nagios d'après la base de données
 - ✓ Ne permet plus d'avoir une gestion manuelle des fichiers
- ✓ Enrichissement
 - ✓ des rapports
 - ✓ des graphes
 - ✓ des schémas
- ✓ Distribution des serveurs Nagios à travers le réseau

Configuration graphique de Nagios

- ✓ Deux logiciels principalement
 - ✓ Nconf
 - ✓ Simple
 - ✓ Intuitif
 - ✓ Facile à installer
 - ✓ Facile à utiliser
 - ✓ Ne permet que la configuration de Nagios en générant les fichiers .cfg
 - ✓ L'exploitation reste inchangée
 - ✓ [Exemple](#)
 - ✓ Centreon
 - ✓ Très riche de possibilités
 - ✓ Interface complexe
 - ✓ Installation complexe
 - ✓ Permet la configuration ET l'exploitation de Nagios
 - ✓ Enrichi Nagios de nombreuses fonctionnalités

9. Centreon

- ✓ Considéré comme le complément indispensable de Nagios
- ✓ Permet la configuration graphique de Nagios
- ✓ Génère des rapports, des historiques, des graphes
 - ✓ 100% paramétrables
 - ✓ De bien meilleure qualité
- ✓ Permet la distribution des processus de supervision sur plusieurs serveurs Nagios
- ✓ Utilise MySql par défaut mais peut utiliser Postgresql
 - ✓ Pour stocker la configuration de Nagios
 - ✓ Pour stocker les informations transmises par Nagios
- ✓ Utilise NDoUtils
 - ✓ Pour récupérer les informations de Nagios
 - ✓ Nagios voit NdoUtils comme un « broker »

Architecture de communication

- ✓ Nagios « refourgue » ses résultats à Ndo
[schéma après les avoir exploitées](#)
- ✓ Le démon NDo
 - ✓ Écoute les données transmises par Nagios
 - ✓ Les stocke dans une base de données MySql
- ✓ Cette architecture permet :
 - ✓ À une multitude d'applications (dont Centreon) de récupérer ces données dans la base de données
 - ✓ De centraliser les données de plusieurs serveurs Nagios répartis dans le réseau
 - ✓ À plusieurs administrateurs de travailler simultanément en plusieurs points du réseau sur le même jeu de données

Installation de Centreon

- ✓ Sous forme de sources à installer
- ✓ Avec un programme de configuration complexe, en perl
- ✓ Nécessite d'avoir bien assimilé l'architecture de communication et le fonctionnement de Nagios
- ✓ Trop complexe pour être abordée ici mais disponible sur internet
- ✓ Heureusement, plusieurs distributions Linux gratuites dédiées à l'administration réseau existent dont : FullyAutomatedNagios (FAN)
 - ✓ Basée sur CentOS, fork de Red Hat Enterprise
 - ✓ Sont installés et correctement configurés de base
 - ✓ Nagios
 - ✓ Centreon
 - ✓ Nagvis
 - ✓ Mysql, NdoUtils, php5, gd, ... tout ce qu'il faut pour que ça fonctionne
 - ✓ Disponible en téléchargement à <http://www.fullyautomatednagios.org>

Exploitation de Centreon

- ✓ Interface Graphique qu'il suffit d'explorer
- ✓ Ne pas oublier d'exporter la configuration vers Nagios
- ✓ Écrase les fichiers de Nagios
- ✓ Rend l'interface de Nagios inutilisable (ou presque) mais elle n'est plus nécessaire
- ✓ Permet de définir tous les objets Nagios
 - ✓ Les hosts
 - ✓ Les services
 - ✓ Les commandes
 - ✓ Les groupes
 - ✓ Les administrateurs
 - ✓ ...
- ✓ Permet d'exploiter les données de supervision produites par Nagios
- ✓ Offre beaucoup plus de fonctionnalités que Nagios
- ✓ Implique quand même d'avoir bien compris et assimilé l'architecture de Nagios
- ✓ [Vue1](#) [Vue2](#) [Vue3](#) [Vue4](#) [Vue5](#) [Vue6](#)

10. Cacti

- ✓ Évolution de MRTG
- ✓ Fonctionne indépendamment de Nagios
- ✓ En reprend en partie l'architecture
- ✓ Entièrement basée sur SNMP
- ✓ Très orienté sur la supervision :
 - ✓ Des éléments réseau
 - ✓ Des trafics
 - ✓ Du QOS
- ✓ Intègre les Mibs Cisco et des autres grands constructeurs
- ✓ Peut aussi surveiller les machines si elles offrent le service SNMP

Installation de Cacti

- ✓ Sur une machine Linux debian, Fedora, Gentoo ou Suse
 - ✓ Installation par les dépôts (apt-get install cacti) pour Debian
 - ✓ Installe, par dépendance, tous les logiciels nécessaires de base (apache2, php5, gd, ...)
- ✓ Sur une machine Linux autre (y compris Mac OS X)
 - ✓ Installer les logiciels indispensables
 - ✓ Récupérer les sources et les installer selon la procédure habituelle
- ✓ Sur une machine Windows
 - ✓ Installation par fichier .zip
- ✓ Sous Mac Os X
 - ✓ Ce n'est pas vraiment prévu
 - ✓ Mais il existe sur le net des procédures pour y arriver
 - ✓ Ça semble marcher...

Exploitation de Cacti

- ✓ 100% full web
- ✓ Interface graphique qu'il suffit d'explorer
- ✓ Il faut bien connaître le protocole SNMP
- ✓ Assez complexe tout de même à configurer
- ✓ [Vue1](#) [Vue2](#) [Vue3](#) [Vue4](#)

11. Nagvis

- ✓ Permet de placer l'état des matériels sur des images
 - ✓ Sur des plans logiques de réseau
 - ✓ Sur des plans d'architecture
 - ✓ Sur des photos de baies
 - ✓ Sur des cartes géographiques
 - ✓ ...
- ✓ En fait sur n'importe quelle image de fond
- ✓ [Ex1 Ex2 Ex3](#)