Sujet :

Mise en place d'un système de supervision Open source.

RAPPORT DE PROJET FIN D'ETUDE LASTIC3

Elaboré par

Othman Souli

Encadré par :

Mr Guemazi Mohamed

Société d'accueil :

ThinkTank

UNIVERSITE VIRTUELLE DE TUNIS

Année Universitaire : 2010/2011

Dédicace

A mes chers parents qui m'ont tellement donné et à qui je dois tout, en signe de gratitude et de reconnaissance.

A ma femme et ma petite fille qui ont apporté le bonheur et la paix a ma vie.

Qu'Allah les protège tous.

Othman Souli

Remerciements

Le travail présenté dans ce rapport de stage de fin d'étude a été effectué pour le compte de la société de service et d'ingénierie « Think Tank », sous la bienveillance du gérant Mohamed Guermazi que je tiens à remercier pour m'avoir accordé l'opportunité de réaliser ce stage dans la société en parallèle avec mes taches quotidiennes comme administrateur réseaux et de m'avoir encadré.

Table des matières

Table des figures

Liste des Tableaux

Cahier des charges

Titre du projet :

Mise en place d'un système de supervision Open source.

Travail demandé:

Recherche, Implémentation et configuration d'une solution Open Source qui vise à superviser à distance les différents serveurs de la société avec gestion des alertes dans un environnement multiplateformes.

Entreprise d'accueil :

Think Tank, société de services et d'ingénierie informatique.

Plan du travail :

Le but principal du projet est de pouvoir établir ou choisir et installer une station de surveillance des serveurs qui remplit les conditions suivantes :

- Coûts financiers les plus réduits possibles.
- Récupération des informations permettant la détection des pannes, l'indisponibilité des serveurs et de leurs services.
- Des renseignements supplémentaires de monitoring sur la charge CPU, Espace disque, mémoire disponible, input/output, etc…
- Gestion des alertes.
- Notification par mail ou SMS en cas de problème.
- Générer des rapports sur le fonctionnement des serveurs par mois.
- Générer des graphes (cartographie du réseau,…)
- Une interface graphique claire pour l'interaction utilisateur/Logiciel.

Introduction générale

Actuellement toutes les entreprises sont équipées d'un réseau local au minimum, et de réseaux de longues distances pour les plus importantes d'entre elles. Leurs parcs informatiques englobent une dizaine voir une centaine d'équipements, engendrés par des serveurs de bases de données et des serveurs de traitements.

Vu que ces systèmes informatiques sont au cœur des activités des entreprises, leur maitrise devient primordiale. Ils doivent fonctionner pleinement et en permanence pour garantir la fiabilité et l'efficacité exigées, et surtout travailler à réduire les problèmes de défaillances, les pannes, les coupures et les différents problèmes techniques qui peuvent causer des pertes considérables.

De ce fait, les administrateurs réseau font appel à des logiciels de surveillance et de supervision de réseaux afin de vérifier l'état du réseau en temps réel de l'ensemble du parc informatique sous leur responsabilité. Et être aussi informés automatiquement (par email, par SMS) en cas de problèmes. Grâce à un tel système, les délais d'interventions sont fortement réduits et les anomalies peuvent être aussitôt prises en main avant même qu'un utilisateur peut s'en apercevoir.

Ainsi, la supervision des réseaux s'avère nécessaire et indispensable. Elle permet entre autre d'avoir une vue globale du fonctionnement et des problèmes pouvant survenir sur un réseau mais aussi d'avoir des indicateurs sur la performance de son architecture.

Dans ce cadre, le présent rapport se base sur trois axes principaux :

- Présenter les notions de base de la supervision informatique et de ses logiciels les plus utilisés actuellement.
- Etudier la solution choisie parmi plusieurs en énumérant ses fonctionnalités et apports.
- la réalisation, et la mise en place de cette solution.

ChapitreI: Présentation du cadre du stage

Ce chapitre se focalise sur la présentation de l'entreprise accueillante et l'étude détaillée de l'existant où on cernera la problématique de mon sujet et on présentera la solution adoptée pour ce dernier.

1. Présentation de la société

Think Tank est une société de services et d'ingénierie informatique qui, partant de ses compétences dans les processus et les technologies adaptés aux systèmes d'informations, a su développer un savoir faire spécifique dans plusieurs domaines tel les méthodologies Orientée Objet, du langage JAVA, et de la plateforme J2EE.

Elle s'appuie en outre sur son centre de compétences « Architecture et Infrastructure » qui réunit un réseau d'experts dans les domaines suivants : systèmes et réseaux, sécurité et bases de données.

Ses prestations s'étendent du conseil en architecture technique jusqu'à l'exploitation globale du système d'information conçu, dans des centres de compétences sécurisés.

❖ **Historique :**

1999	-Création de Think Tank SARL, acquisition du projet Marketing Media Warehouse de BMW.
2001	-Signature d'un contrat de Partenariat avec la société Think Tank Business Solutions AG.
2003	-Signature d'un contrat de sous-traitance avec BearingPoint (Corporate KPMG Consulting)
2004	-Signature du contrat avec BearingPoint Munich pour la réalisation du projet PGI de Siemens AG (Power Generation Industrial)
2005	-Signature du contrat de sous-traitance avec SBB Suisse (Chemins de Fer Suisses) -Signature du contrat avec BearingPoint Frankfort pour la participation au méga projet de Reporting bancaire ABACUS Davinci qui utilise la norme financière Bale II
2007	Participation de 8 experts de Think Tank dans le projet du registre de commerce national de Tunisie dans le cadre du Programme de Modernisation Industrielle (PMI)

2008	Participation des experts de Think Tank dans l'action de mise à niveau du système d'information du département de propriété industrielle de l'INNORPI lancée par le PMI
2009	-Mise en place du système qui gère le registre de commerce Tunisien. Interlocuteurs Ministère de la Justice et des Droits de l'Homme et l'Institut National de Normalisation et de la Propriété Industrielle (INNORPI) - Mise en place du nouveau Portail dédié au registre de commerce de Tunisie - Mise en place de la première plateforme de dépôt des bilans en ligne dédiée aux experts comptables

Tableau 1. Historique de Think Tank

2. Etude de l'existant

2.1 Description de l'existant

Ce présent travail s'est déroulé dans un environnement comportant un parc informatique composé d'une dizaine de machines et de serveurs locaux et distants, dont le nombre est capable de se dupliquer en fonction du temps et des clients. La figure 1 présente l'architecture de Think Tank :

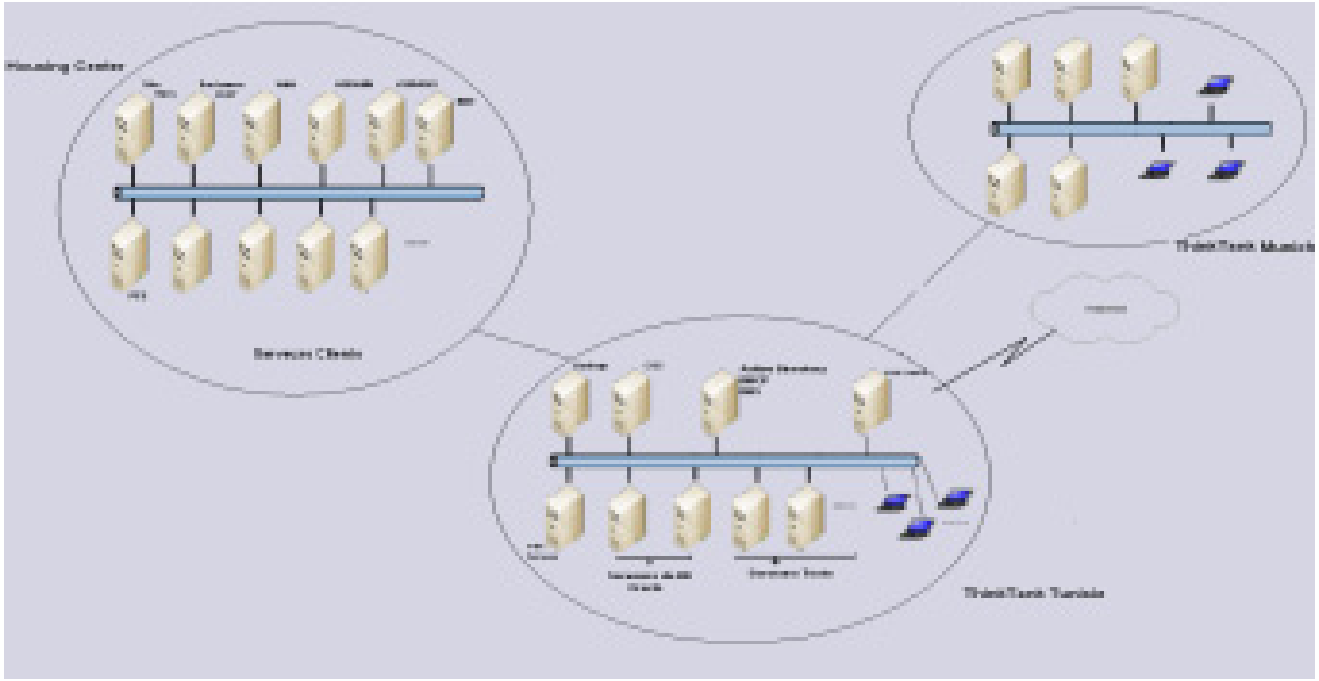


Figure 1. Architecture de Think Tank

Think Tank possède un très grand nombre de serveurs locaux en Tunisie, ainsi qu'elle héberge une dizaine de serveurs pour ses clients dans un centre d'hébergement en Allemagne (Housing Center).

2.2 Critique de l'existant

Ayant un très grand nombre de serveurs à gérer, l'administrateur est incapable de vérifier leurs disponibilité (en ligne ou pas), de déterminer la qualité des services qu'ils offrent, ni détecter la défaillance des équipements (charge CPU, Etat mémoire, surcharge du disque….), ni les surcharges et pénurie temporaire des ressources. Le seul moyen de détecter ces anomalies ne peut se faire que par la réception des différentes plaintes et réclamations des clients.

Se souciant de sa réputation et concerné par la satisfaction et le confort de ses clients, la société veut à tout prix éviter la confrontation à des clients mécontents d'où éviter le risque de les perdre, et ce en travaillant à offrir une meilleure qualité de services à ses clients en anticipant les pannes et en évitant les arrêts de longue durée gênant les services qui peuvent causer de lourdes conséquences aussi bien financières qu'organisationnelles.

Le but de ce projet est donc de trouver une solution optimale pour la gestion des serveurs et le monitoring de ses équipements en premier lieu, offrir la possibilité de devenir « pro actif » face aux problèmes rencontrés en un second lieu, et finalement et le plus important, de pouvoir détecter et interpréter en un simple coup d'œil les causes et origines des problèmes rencontrés afin de les fixer le plus rapidement possible.

2.3 Solution proposée

La gestion des serveurs distants et le monitoring de ses équipements étant le plus grand souci de l'administrateur, j'ai jugé nécessaire de mettre en évidence un outil pour contrôler le fonctionnement du réseau, d'étudier les données collectées et de définir des seuils d'alertes qui peuvent servir pour le déclenchement des alertes lors de détection des problèmes.

Il s'agit donc et sans doute d'une mise en place d'un système de supervision qui pourra grâce aux différentes fonctionnalités qu'il offre, anticiper les pannes en suivant méticuleusement le fonctionnement du système et en surveillant le statut des serveurs, des divers services réseaux et d'offrir des renseignements supplémentaires voir charge CPU, espace disque, mémoire disponible, etc.

Un système de supervision offrira à l'administrateur la possibilité de réagir le plus rapidement possible face aux pannes qui peuvent intervenir afin d'éviter un arrêt de production de trop longue durée.

3. Etude de choix

De nombreuses plateformes de supervision existent aujourd'hui. Certaines se contentent de gérer à temps réels l'état du réseau et préserve une vue globale sur le fonctionnement de son architecture, d'autres permettent également de connaître l'état des différents services, et d'autres qui offrent la possibilité de ressortir de nombreuses statistiques du réseau permettant une analyse assez fine.

3.1 Les offres éditeurs

S'assurant que la supervision est un marché porteur, les sociétés se pressent de plus en plus à investir dans des produits permettant la supervision et une meilleure gestion des réseaux.

Deux familles apparaissent, celle proposant des solutions généralistes pour la supervision des réseaux, des serveurs, des applications, des sites web,… comme les logiciels Patrol (BMC), d'Unicenter (Computer Associate), de la gamme openview (HP)...

D'autres offrent une supervision des domaines plus spécifiques citant comme logiciel panorama (Altaworks) qui gère uniquement l'aspect sécurité ou PathWAI (Candle) qui se penche principalement sur la supervision des applications.

- Ces solutions n'ont qu'un seul point commun : **un prix élevé**.

3.2 Les offres libres

Il existe des solutions de supervision libres et professionnelles. Parmi les plus répandues, reconnues du moment nous pouvons citer Nagios, Zabbix, BigBrother2 et OpenNMS.

L'avantage de ces logiciels libres est la gratuité, la disponibilité du code source et la liberté d'étudier et de modifier le code selon nos besoins et de le diffuser. De plus, il existe une communauté importante d'utilisateurs et de développeurs qui participent à l'amélioration des logiciels et apportent une assistance par la mise en ligne des documentations et les participations aux forums.

3.3 Choix du logiciel

Les différentes solutions commerciales déjà présentées (HPOpenview, Patrol, BigBrother, etc..) nécessitent un investissement important pour leur mise en place, et pour des raisons propres à l'entreprise, toutes ces solutions sont à écarter de mon liste de choix.

Parmi les solutions les plus connues, recommandées et surtout Libres, on citera Nagios et Zabbix. Voici un tableau comparatif des deux logiciels choisis.

	Zabbix	Nagios
Présentation	-Open source, libre -Multiplateformes -Homogène. -Moteur en C, interface web utilisateur en PHP, base de données SQL (MySQL, Oracle…) -Configuration centralisée sur une même interface graphique. ➔Peut monitorer de 3 manières : -LANCEMENT d'un processus sur les machines à monitorer pour collecter des données locales, grâce à l'agent Zabbix (obtenir des infos sans utiliser SNMP). -Requêtes SNMP. -Check externes qui sert à tester les services réseaux (rien à installer sur l'équipement surveillé, tests limités à des pings ou test de protocoles).	-Open source, Libre. -Conçu pour les plateformes Unix. -Modulaire. -Moteur en C, perl, sharp…, interface web en PHP, base de données SQL. -Configuration plus ou moins complexe ➔Peut monitorer de 3 manières : -L'utilisation des journaux d'exploitation par l'envoie des évènements issus des fichiers log en temps réel vers un serveur centrale offrant les informations nécessaires à la supervision. -Supervision active des services et infrastructure qui nous permet de garder l'historique des performances.
Fonctionnalités	-Offre une interface web de consultation et d'administration. -Peut générer des graphes. -Peut lever des alertes en envoyant des mails. -Supervise des équipements SNMP. -Gère les pannes et les performances	-Offre une interface web basée sur les CGL avec gestion des droits pour la consultation. -Génère des rapports de surveillance. -Il a la possibilité de monitorer à distance à travers un firewall. -Il peut définir des serveurs esclaves qui prennent le relais si le serveur maitre tombe en panne. -Surveillance des ressources des serveurs (CPU, mémoire…) -Surveillance des services réseaux. -Arrêt temporaire de la supervision locale ou globale. -Génère des graphes par l'interfaçage avec RRDTools.
Architecture	Architecture généralement basée sur : -Serveur Zabbix, le cœur et moteur de l'application programmé en C. -Agent Zabbix pour la collection des informations locales. -Une interface web d'administration et consultation des données. -Une base de données SQL.	Architecture généralement basée sur : -Le moteur de l'application qui sert à ordonnancer les tâches de supervision écrit en C. -Une interface web réalisée à l'aide des GCI, décrivant la vue d'ensemble su système et les anomalies possibles. -Plusieurs plugins qui peuvent être complétés en fonction des besoins.

Avantages	-Multiplateforme. -Utilise peu de ressources -Plus léger grâce à son homogénéité (Pas de plug-in à ajouter). -Mise à jour facile. -Configuration et utilisation aisée. -Interface vaste mais claire.	-Des plugins qui étendent les possibilités de Nagios. -Une très grande communauté qui participe activement au développement. -Un moteur performant -solution complète permettant le reporting, la gestion des pannes et d'alarmes, gestion des utilisateurs… -Des plugins permettent aux utilisateurs de développer facilement ses propres vérifications de services. -Possibilité de repartir la supervision entre plusieurs administrateurs. -Offre la possibilité de développer ses propres modules.
Inconvénients	-L'agent Zabbix communique les données en claire➔ nécessité de sécuriser les données. -Peu d'interfaçage avec d'autres solutions commerciales. -Communauté de développeurs limitée.	-Configuration complexe mais peut s'améliorer en ajoutant Centreon. -Interface peu ergonomique et intuitive.

Tableau 2. Tableau comparatif

Parmi ces solutions libres, les deux logiciels Zabbix et Nagios sont les plus répandus et les plus utilisés. Par rapport à mon projet, se sont les deux solutions les plus adaptées permettant de satisfaire pratiquement tous les besoins de la société, par les différentes fonctionnalités qu'elles offrent. Et compte tenu de ce critère Zabbix et Nagios restent à égalité et il me sera impossible de les départager.

Une des particularités captivantes de Nagios est sa modularité, on a ainsi estimé que Nagios a été plus adapté aux besoins de mon projet que Zabbix. En effet, grâce à ses plugins, Nagios possède une architecture facilement adaptable à l'environnement. Ces derniers pouvant être ajoutés, modifiés ou même personnalisés et permettent de spécifier les tâches pour aboutir au résultat voulu.

De plus Nagios est une solution stable, dispose d'une grande communauté de développeurs et est utilisé aussi bien dans les petites et moyennes infrastructures que dans les grands parcs informatiques et utilisé surtout par plusieurs entreprises de renommé, tels que Yahoo (100 000 serveurs), Yellow pipe Web Hosting (7000 serveurs) …

Bien que ce dernier soit réputé par sa configuration fastidieuse, il peut être couplé à Centreon un logiciel qui lui servira de couche applicative afin de faciliter la configuration et d'établir des interfaces IHM plus ergonomiques et compréhensibles.

4. Conclusion

Ce chapitre a été conçu pour familiariser l'environnement du travail en présentant l'entreprise d'accueil et l'architecture réseau dont elle dispose.

Les problèmes que rencontre la société se sont imposés suite à l'étude de l'existant et à sa critique, ce qui m'a permis de cerner la problématique de mon projet. J'ai par la suite proposé des solutions et leur étude a mon gérant et finalement nous avons posé notre choix sur la solution que nous jugeons la plus convenable à la société et à la formation que nous estimons acquérir qui est le logiciel de supervision libre « Nagios ».

Le chapitre suivant attaquera une étude approfondie de la solution choisie.

ChapitreII: Présentation de l'outil de supervision Nagios

Dans ce présent chapitre, Je commençe par définir la notion de la supervision et ses objectifs ensuite, analyser de prés les fonctionnalités de la solution proposée, son architecture, et les différents services qu'elle offre et finir par énumérer les différents fichiers de configurations sur quoi se base cette solution.

1. La supervision

1.1 Définition

La supervision de réseaux peut être définie comme l'utilisation de ressources réseaux adaptées dans le but d'obtenir des informations (en temps réel ou non) sur l'utilisation ou la condition des réseaux et de leurs éléments afin d'assurer un niveau de service garanti, une bonne qualité et une répartition optimale et de ceux-ci.

La mise en place d'une supervision réseau, a donc pour principale vocation de collecter à intervalle régulier les informations nécessaires sur l'état de l'infrastructure et des entités qui y sont utilisés, de les analyser et de les rapporter.

1.2 Objectifs

L'objectif d'une supervision de réseaux peut ainsi se résumer en trois points :

- **Etre réactif** en alertant l'administrateur (e-mail ou sms) en cas de dysfonctionnement d'une partie du système d'information.
- **Etre pro actif** en anticipant les pannes possibles.
- Cibler le problème dès son apparition afin d'agir rapidement de la façon la plus pertinente possible.

2. Nagios

2.1 Présentation

Nagios est un logiciel libre distribué sous licence GPL qui permet de superviser un système d'information complet. Utilisé par de nombreuses sociétés, il fait l'objet de contribution et recherche très actives.

Etant le successeur de NetSaint dont la première version date de 1999, ce logiciel est considéré comme une évolution de ce dernier auquel a été ajoutée, entre autre, la gestion du protocole SNMP. Il apparaît sous le nom de Nagios le 10 mai 2002 aux conditions de la GNU General Public License.

Cet outil repose sur une plate-forme de supervision, fonctionnant sous Linux et sous la plupart des systèmes Unix. Il centralise les informations récoltées périodiquement par le fonctionnement modulaire dont il est caractérisé, ce qui le rend beaucoup plus attractif que ses produits concurrents. En revanche sa configuration peut se révéler complexe.

2.2 Fonctionnalités

Les fonctionnalités de Nagios sont très nombreuses, parmi les plus communes nous pouvons citer les suivantes :

- La supervision des services réseaux (SMTP, http…), des hôtes et des ressources systèmes (CPU, charge mémoire…)
- La détermination à distance et de manière automatique l'état des objets et les ressources nécessaires au bon fonctionnement du système grâce à ses plugins.
- Représentation coloriée des états des services et hôtes définies.
- Génération de rapports.
- Cartographie du réseau.
- Gestion des alertes.
- Surveillance des processus (sous Windows, Unix…).
- Superviser des services réseaux : (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, LDAP, etc.)
- La supervision à distance peut utiliser SSH ou un tunnel SSL.
- Les plugins sont écrits dans les langages de programmation les plus adaptés à leur tâche (Bash, C++, Python, Perl, PHP, C#, etc.)

Toutes ces fonctionnalités sont assurées grâce la gestion et supervision du réseau et ses différentes entités d'une manière centralisée. La figure 2 modélise cet aspect :

Figure 2. Centralisation d'informations par Nagios

2.3 Architecture

L'architecture de Nagios se base sur le paradigme serveur-agent. D'une manière spécifique, un serveur faisant office de point central de collecte des informations tandis que les autres machines du réseau exécutent un agent chargé de renvoyer les informations au serveur.

L'architecture globale de Nagios peut être décomposée en 3 parties coopératives entre elles :

- **Un noyau** qui est le cœur du serveur Nagios, lancé sous forme de démon et responsable de la collecte et l'analyse des informations, la réaction, la prévention, la réparation et l'ordonnancement des vérifications (quand et dans quel ordre).

C'est le principe de répartition des contrôles au mieux dans le temps qui nous évite la surcharge du serveur et des machines à surveiller.

- **Des exécutants :** ce sont les plugins dont un grand nombre est fourni de base, responsables de l'exécution des contrôles et tests sur des machines distantes ou locales et du renvoie des résultats au noyau du serveur Nagios

- **Une IHM :** C'est une interface graphique accessible par le web conçue pour rendre plus exploitable les résultats. Elle est basée sur les CGI (Common Gateway Interface) fournis par défaut lors de l'installation de Nagios qui interprètent les réponses des plugins pour les présenter dans l'interface.

Cette interface sert à afficher de manière claire et concise une vue d'ensemble du système d'information et l'état des services surveillés, de générer des rapports et de visualiser l'historique. D'une manière générale avoir la possibilité de détecter en un simple coup d'œil, les services ou hôtes ayant besoin d'une intervention de leur administrateur.

Il est possible de coupler Nagios à une base de données MySQL ou Postgres, lorsque le nombre d'objets à superviser devient conséquent. La figure 3 modélise l'architecture de Nagios.

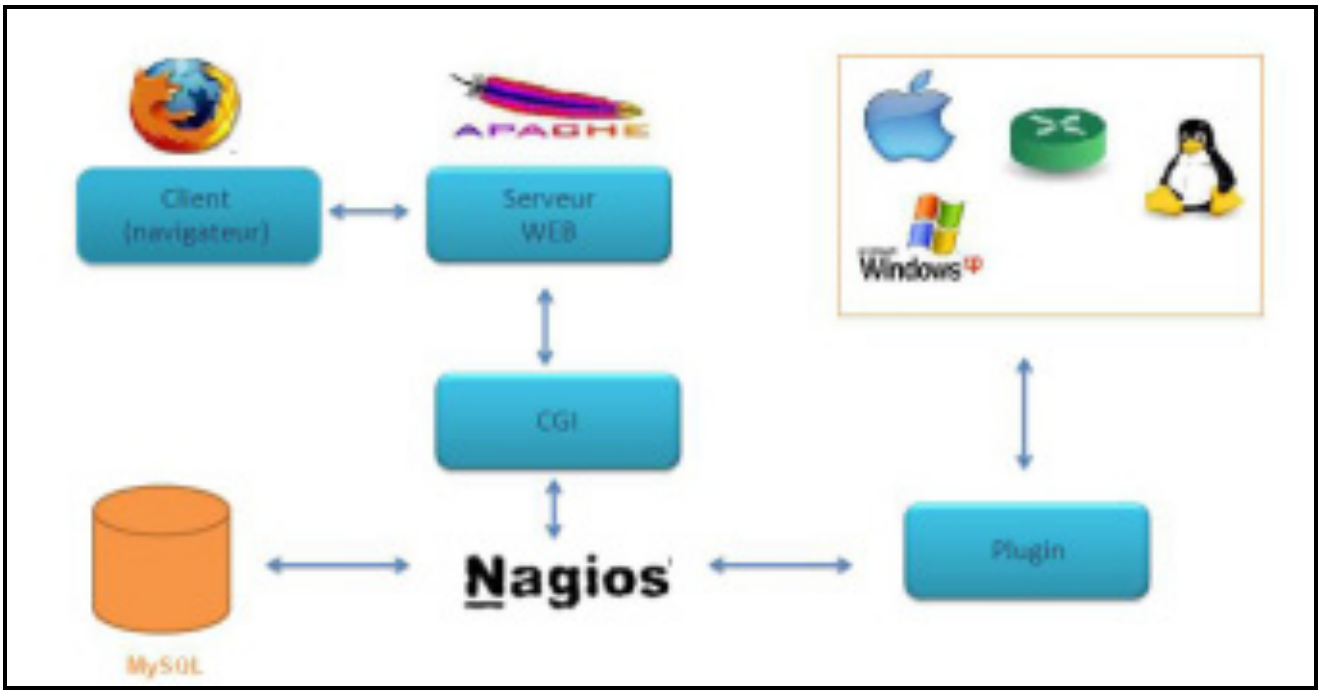


Figure 3. Architecture de Nagios

2.4 Plugins

Nagios fonctionne grâce à des plugins écris en Perl ou en C. Sans eux, il est totalement incapable de superviser et se résume en un simple noyau.

Ces plugins sont des programmes externes au serveur, des exécutables qui peuvent se lancer en ligne de commande afin de tester une station ou service. Ils fonctionnent sous le principe d'envoie de requêtes vers les hôtes ou services choisis lors d'un appel du processus de Nagios, et la transmission du code de retour au serveur principale qui par la suite se charge d'interpréter les résultats et déterminer l'état de l'entité réseau testée.

La relation entre le noyau et les plugins est assuré d'une part par les fichiers de configuration (définitions des commandes) et d'autre part par le code retour d'un plugin. Cette relation peut se résumer par le tableau 3:

Code retour	Etat	Signification
1	OK	Tout va bien
2	Warning	Le seuil d'alerte est dépassé
3	Critical	Le service a un problème
4	Unkown	Impossible de connaître l'état du service

Tableau 3. Signification des codes de retours

Nagios est livré avec un « package » de greffons standards regroupant les plus utilisés. Pour une utilisation basique et simple, ils devraient être suffisants. En voilà quelques exemples:

- **check_http :** Vérifie la présence d'un serveur web.
- **check_load :** Vérifie la charge CPU locale.
- **check_ping :** Envoie une requête Ping à un hôte.
- **check_pop :** Vérifie la présence d'un serveur POP3.
- **check_procs :** Compte les processus locaux.
- **check_smtp :** Vérifie la présence d'un serveur SMTP.
- **check_snmp :** Envoie une requête SNMP (passée en argument) à un hôte.
- **check_ssh :** Vérifie la présence d'un service SSH.
- **check_tcp :** Vérifie l'ouverture d'un port TCP (passé en argument).
- **check_users :** Compte le nombre d'utilisateurs sur la machine locale.

Il est possible de créer son propre plugin et l'interfacer avec Nagios tout en respectant les conventions des codes de retours précédemment expliqués.

La vivacité de la communauté Open Source et celle de Nagios 2 en particulier permet de disposer d'un grand nombre de plugins supplémentaires.

Comme on peut le constater sur la figure 4, les plugins peuvent fonctionner soit en effectuant des tests en local, à distance par le biais de divers moyen comme l'installation des agents NRPE ou NSClient ou autres.

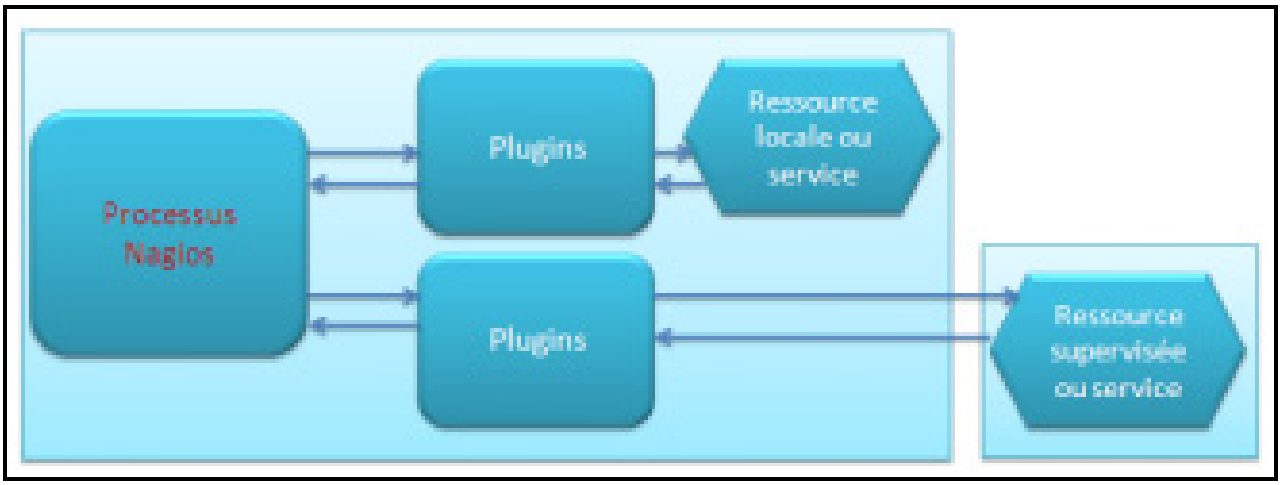


Figure 4. Principe de fonctionnement des plugins

2.5 Les fichiers de configuration

Nagios s'appuie sur différents fichiers textes de configuration pour construire son infrastructure de supervision. Nous allons à présent citer et définir ceux qui sont les plus importants :

- **Nagios.cfg** est le fichier de configuration principal de Nagios. Il contient la liste des autres fichiers de configuration et comprend l'ensemble des directives globales de fonctionnement.
- **Cgi.cfg** contient un certain nombre de directives qui affectent le mode de fonctionnement des CGI. Il peut être intéressant pour définir des préférences concernant l'interface web de Nagios.
- **Resource.cfg** permet de définir des variables globales réutilisables dans les autres fichiers. Etant inaccessible depuis les CGI qui génèrent l'interface, ce fichier peut être utilisé pour stocker des informations sensibles de configuration.
- **Commands.cfg** contient les définitions des commandes externes, telles que celles qui seront utiles pour la remontée d'alerte.
- **Checkcommands.cfg** contient les définitions des commandes de vérification prédéfinies et celles définies par l'utilisateur.
- **Hosts.cfg** définit les différents hôtes du réseau à superviser. A chaque hôte est associé son nom, son adresse IP, le test à effectuer par défaut pour caractériser l'état de l'hôte, etc.
- **Services.cfg** associe à chaque hôte ou à chaque groupe d'hôtes l'ensemble des services qui doivent être vérifiés.

- **Hostsgroups.cfg** définit des groupes d'hôtes pour regrouper des hôtes selon des caractéristiques communes. Un hôte peut appartenir à plusieurs groupes.
- **Contacts.cfg** déclare les contacts à prévenir en cas d'incident et définit les paramètres des alertes (fréquences des notifications, moyens pour contacter ces personnes, plages horaires d'envoi des alertes...).

3. Conclusion

Le présent chapitre a été introduit avec une brève présentation de la notion de supervision et ses enjeux. Ensuite j'ai décrit l'aspect de ma solution, énuméré ses fonctionnalités et modélisé son architecture. Finalement une partie a été consacrée pour la définition des différents fichiers de configuration générés par la solution de supervision Nagios, précédée par l'énumération des différents plugins de base responsable de l'exécution des tests nécessaires.

ChapitreIII: Les compléments de Nagios

Dans ce chapitre je vais présenter tout outils ou compléments que j'envisage ajouter à Nagios afin de mettre en valeur les fonctionnalités qu'elle offre optimiser , enrichir et garantir la mise en place d'une solution complète, facile à administrer et qui répond aux besoins déjà fixés.

1. NDOutils

1.1 Utilités

Il faut d'abord savoir que lorsque les greffons effectuent des tests, ils retournent au processus/ordonnanceur Nagios, deux types de données qui sont les états des hôtes et leurs services, ainsi que les données de performances qui par la suite seront enregistrées dans des fichiers plats.

Pour obtenir une information Nagios est obligé de lire et traiter ces fichiers en entier. Aussi chaque rafraichissement d'une page web depuis l'interface de Nagios implique une analyse complète de ces fichiers.

NDOutils vient alors optimiser l'exploitation de ces données en les exportant vers une base de données MySQL, ce qui a les avantages suivantes :

- Stockage des données à long terme.

- Permettre à un logiciel tiers comme « Centreon » d'accéder de manière optimisée aux données d'états et performances de Nagios et de partager ses données.

- Optimisation de l'exploitation des données et amélioration des performances ; il est plus rapide de rechercher des informations dans une base de données structurée, plutôt que dans un fichier de journalisation qu'il faut parcourir entièrement à chaque utilisation.

1.2 Présentation

NDOutils est un greffon chargé de transmettre les données remontées par Nagios (configuration des serveurs supervisés, les états des hôtes, les données de performance…) vers une base de données MySQL plutôt que de ne les garder que dans les fichiers plats.

De cette façon, les données seront plus souples à gérer. Grace à la possibilité de stockage à long terme, les données sont facilement exploitées et l'information devient aisément transformable de la manière que l'on souhaite.

NDOutils interagi avec Nagios indépendamment de Centreon.

1.3 Architecture

NDOutils se compose de deux modules :

- **Ndomod :** lancé automatiquement avec Nagios et responsable de l'exportation des données extraits des fichiers plats pour les déposer dans un socket (Unix, tcp).
- **Ndo2db :** démon nécessitant un script d'initialisation et responsable de l'ouverture de socket (Unix ou TCP) et place les données trouvées dans une base de donnée MySQL.

La figure 5 décrit l'architecture de NDOutils.

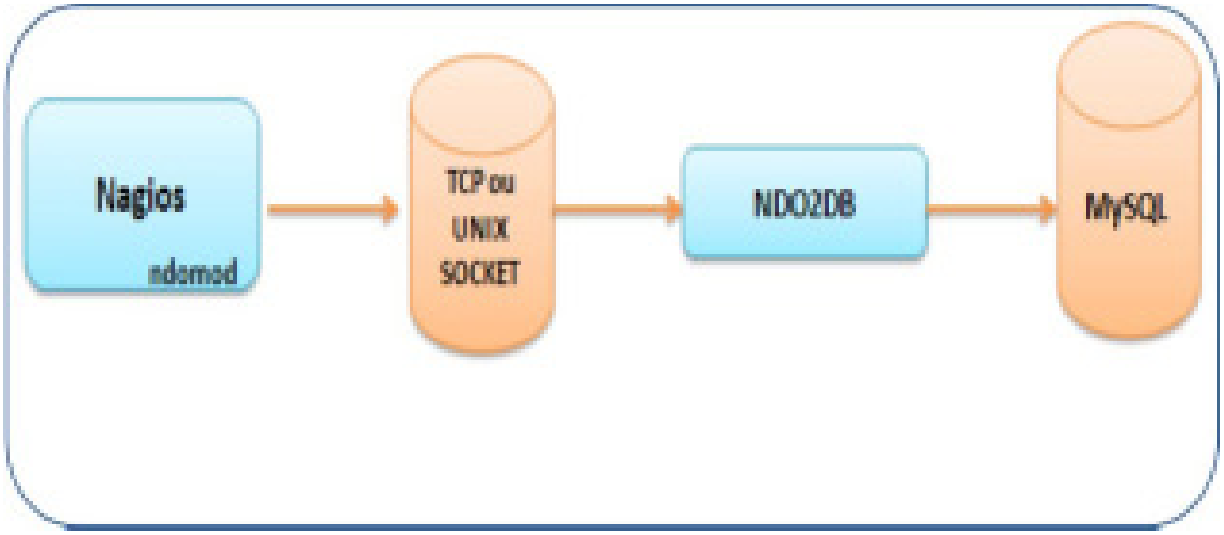


Figure 5. Architecture NDOutils

2. Centreon

2.1 Utilités

Sans aucun doute, Nagios est considéré comme étant une solution très puissante. Cependant, on peut lui reprocher d'être très compliquée à configurer vu le nombre important de fichiers dont elle dispose.

La modification manuelle de ces fichiers de configuration, à chaque ajout une hôte, un service, une commande.., augmentera le risque d'affronter beaucoup plus d'erreur.

On a donc choisi de coupler Nagios à Centreon pour remédier à ce problème en évitant la modification à la main de ces fichiers textes. Centreon n'ayant pas seulement le grand avantage de gérer automatiquement les nombreux fichiers de configuration de Nagios mais aussi une interface multiutilisateurs, intuitive et personnalisable avec intégration des droits d'accès en plus d'un compte rendu graphique plus pratique et élégant que celui offert par Nagios.

2.2 Présentation

Centreon est une couche applicative Web venant se greffer à Nagios pour offrir une administration moins rudimentaire basée sur deux fonctionnalités principales :

- **Une seconde interface de monitoring** : Centreon propose une interface plus sobre, ainsi qu'une façon différente de traiter les données remontées par Nagios.
- **Puissante interface de configuration** : Centreon autorise en effet à l'utilisateur de modifier intégralement la configuration de Nagios depuis son navigateur internet, plutôt qu'en modifiant manuellement les fichiers éparpillés sur le disque.

Cet outil utilise ses propres bases de données MySQL crées automatiquement lors de son installation pour récupérer toutes les données d'états et de performances de Nagios pour les traiter et les afficher dans sa propre interface graphique.

Cet outil construit ses propres graphiques grâce aux RRDTools, des bases de données particulièrement adaptées à la construction graphique.

2.3 Architecture

- **Centreon et Base de données**

Centreon interagit principalement avec la base de données MySQL pour remonter les données fournies par Nagios et stockées dans la base grâce à NDO.

Lors de son installation Centreon crée automatiquement trois schémas dans la base de données MySQL :

- **Centstatus:** C'est la base dans laquelle NDOUtils stocke les données extraites des fichiers plats de Nagios et sur laquelle Centreon pointe pour pouvoir remonter les mêmes données.

Ces données sont visualisées dans l'interface monitoring de Centreon.

- **Centstorage:** Traite et stocke les données de performances remontés de Nagios via NDOutils vers la base de données MySQL, avant leurs intégration en base RRD. Responsable de la création des parties métrologiques de Centreon qui sont le reporting et la génération des graphs.

Ces données sont visualisées dans la partie « Reporting » et « Views » de Centreon.

- **Centreon:** Collecte les informations de configuration, et stocke les fichiers objets de Nagios (Host, Services, Périodes, etc...). Grâce aux fonctions d'Import/Export, Centreon peut générer de nouveaux fichiers de configuration pour Nagios.

❖ **Centreon et démons**

Pour un fonctionnement sain, Centreon a besoin que ses deux démons soient lancés :

- **Centstorage** : Centstorage est l'outil qui exploite les données remontées par Nagios pour Centreon. C'est un programme écrit en Perl, associé à Centreon. A chaque modification du fichier de données perfdata, centstorage met à jour deux bases de données « Centstorage » et « RRD ».

- **Centcore** : Dans le cas où l'architecture adoptée est distribuée (serveur centrale pour la supervision et d'autres serveurs fils), Centcore permettra à cette architecture de bien communiquer ensemble, en se chargeant de la transmission des données entre ses différents serveurs. Aussi Responsable du déploiement de la configuration de Centreon vers Nagios.

La figure 6 est un schéma récapitulatif décrivant l'interaction entre les différentes couches logicielles de l'association entre Nagios3/Centreon2.

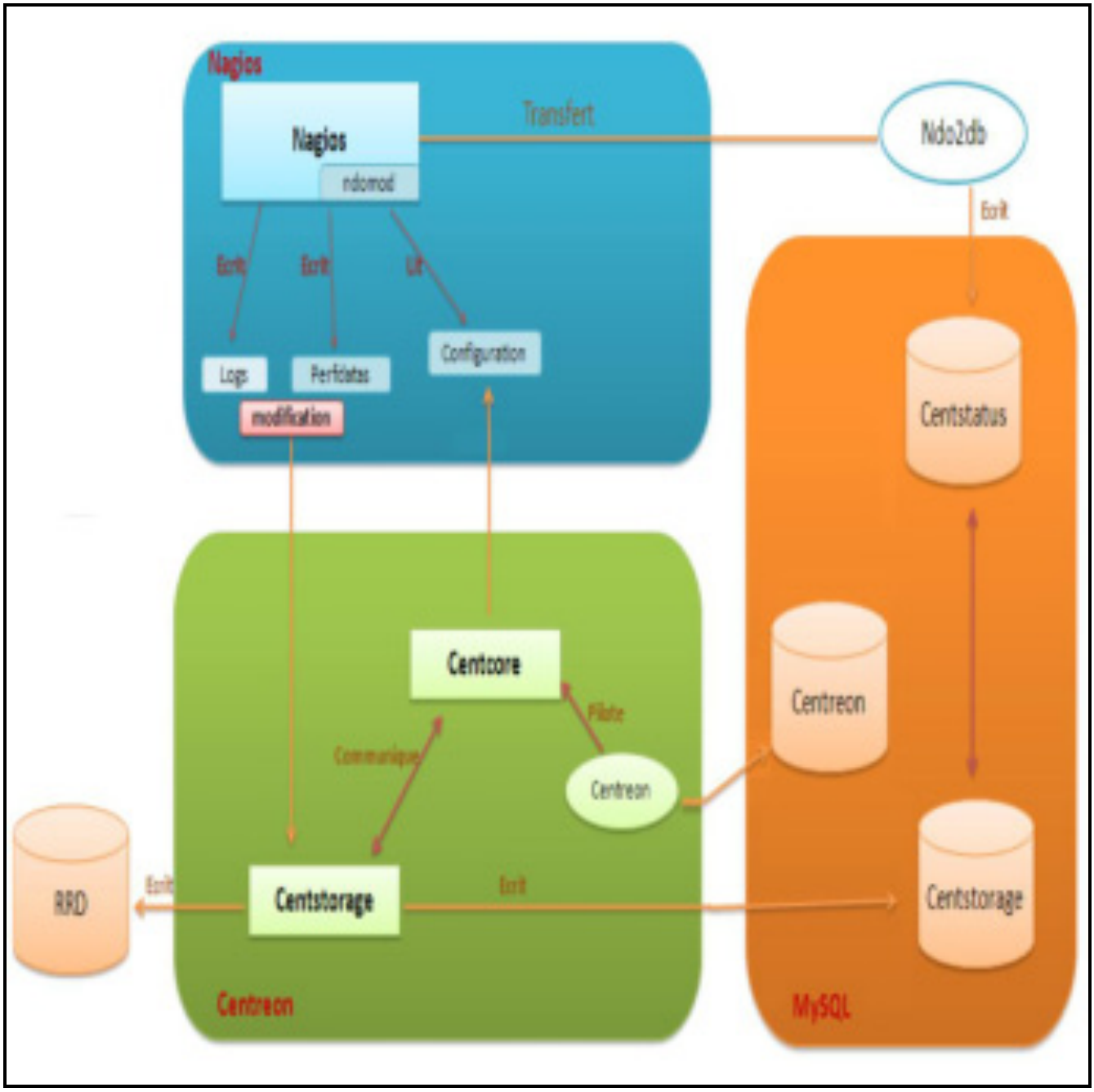


Figure 6. Interaction entre Nagios et Centreon

3. NSClient pour la supervision des serveurs Windows

3.1 Présentation :

C'est un plugin permettant de récupérer un nombre important d'information à surveiller sur une machine Windows.

Le plugin se livre avec un ensemble de commandes check qui nous permet de dégager d'importantes informations comme :

- **CLIENTVERSION** : retourne la version de l'agent NSClient.
- **CPULOAD :** Retourne la charge moyenne du système.
- **UPTIME** : Retourne la durée écoulée depuis le dernier redémarrage de la machine.
- **USEDDISKSPACE :** Retourne la taille et le pourcentage du disque utilisé.
- **MEMUSE :** Retourne la taille de la mémoire utilisée, et la taille restante.
- **SERVICESTATE** : Retourne le statut (démarré, arrêté) d'un ou plusieurs services Windows.
- **PROCSTATES :** Vérifie si un ou plusieurs processus sont démarrés.
- **COUNTER :** Interroge n'importe quel compteur de performance.

3.2 Architecture

NSClient se base sur une architecture client/serveur (Figure 7). La partie cliente nommée check_nt, doit être disponible sur le serveur Nagios et on doit vérifier son existence parmi les plugins délivrés avec Nagios-plugins sinon l'installer. La partie serveur NSClient++ est à installer sur chacune des machines Windows à surveiller.

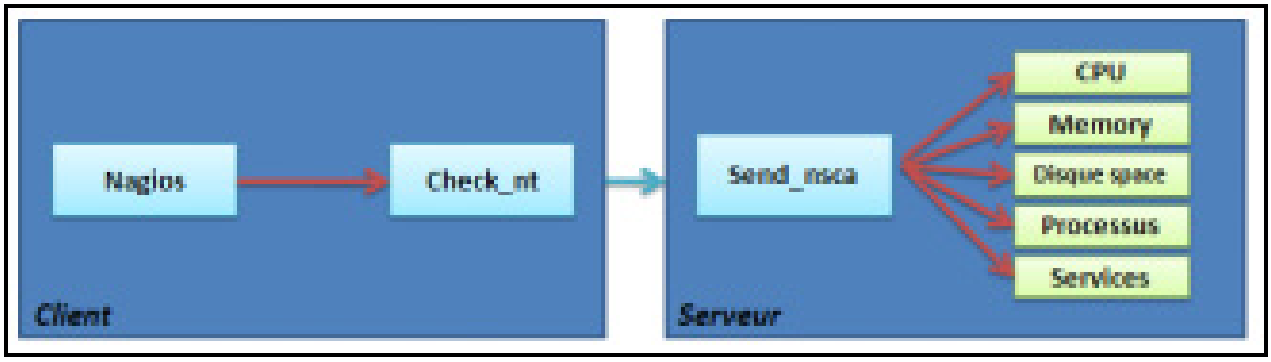


Figure 7. Architecture NSClient

4. NRPE pour la supervision des serveurs Linux

4.1 Présentation

NRPE (Nagios Remote Plugin Executor) est un agent de supervision qui vous permet de récupérer les informations à distance lors de la supervision d'un serveur Linux. Il a le grand avantage d'exécuter les commandes dans la machine à superviser ce qui permet ainsi de répartir les charges.

Il est livré avec un ensemble de commandes check définis par défaut dans son fichier de configuration et nécessite l'installation des plugins Nagios aussi.

4.2 Architecture

NRPE se base sur une architecture client/serveur (Figure 8). La partie cliente nommée check_nrpe, doit être disponible sur le serveur Nagios et on doit vérifier son existence parmi les plugins délivrés avec Nagios-plugins sinon l'installer. La partie serveur NRPE est à installer sur chacune des machines Windows à surveiller.

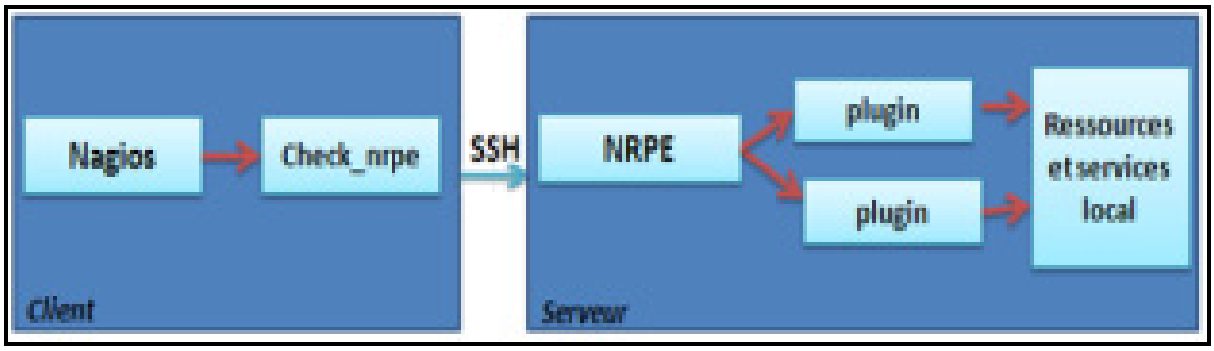


Figure 8. Mécanisme du NRPE

Procédure de fonctionnement :

- Le serveur Nagios demande l'exécution d'un plugin sur la machine distante.
- Le daemon NRPE hébergé sur la machine distante, reçoit la requête d'exécution du plugin.
- Le plugin est exécuté sur la machine distante.
- Le daemon NRPE de la machine distante envoie le résultat du plugin au serveur Nagios.
- Le serveur Nagios interprète les résultats reçus.

5. Conclusion

Le but de ce chapitre était de présenter les compléments que j'ai choisis à Nagios. Certains ont été choisis pour leur nécessité comme les greffons NRPE et NSClient, et d'autres participaient surtout à l'amélioration de la manipulation et l'utilisation de Nagios, et surtout facilité de sa configuration.

Le chapitre suivant entamera l'aspect technique de mon projet, de la mise en place jusqu'aux exemples d'utilisations.

ChapitreIV: Mise en place du système de supervision

Au sein de ce dernier chapitre, je vais présenter le chronogramme de réalisation de mon projet, de l'environnement de travail et enfin quelques captures écrans des interfaces de Nagios/Centreon.

1. Chronogramme

Le chronogramme suivant a été suivi tout au long de la réalisation de mon projet

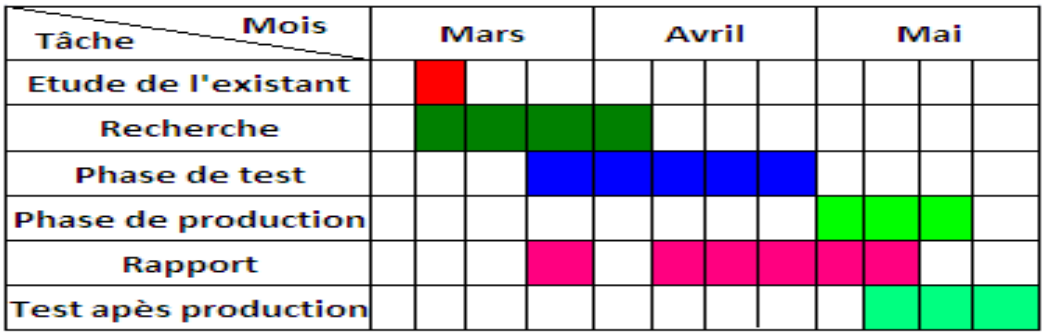


Figure 9. Chronogramme du projet

2. Environnements de mise en place

2.1 Environnement matériel

- **Phase de test :** j'ai au cours de cette phase installé une machine virtuelle sur ma machine personnelle pour tester la solution choisie et s'adapter à sa mise en place, mais aussi de s'assurer si elle répond vraiment aux besoins fixés par la société avant de passer à la phase de production, et ce en essayant de tester deux serveur serveurs distants Windows et Linux.
- **Phase de production :** Une douzaine de serveurs Windows et Linux à superviser.

  J'ai installé un serveur Linux (hardware et system exploitation) pour y déployer Nagios/Centreon qui a les caractéristiques suivantes ;
  - Système Suse Linux entreprise 11 (i589).
  - Microprocesseur Intel® Pentium® de vitesse 2.80GHz.
  - Connexion internet.

2.2 Environnement logiciel

- La solution de supervision Nagios-3.2.3.
- Les greffons de Nagios, Nagios-plugins-1.4.15
- La couche applicative associée à Nagios pour faciliter sa configuration et son administration Centreon-2.1.10
- Le plugin NDOutils-1.4b9 pour le stockage des données de Nagios dans la base de données MySQL et le partage de ces données avec Centreon.
- Le plugin NSClient pour la supervision des serveurs Windows.
- Le plugin NRPE-2.1.12 pour la supervision des serveurs Linux.

3. Mise en place de Nagios/Centreon et les plugins

3.1 Pré-requis Nagios/Centreon

En plus des plugins Nagios a besoin de satisfaire certaines dépendances.

Les pré-requis à l'installation sont donc :

- Dépendances LAMP : Apche2, PHP5, MySQL
- Librairies Perl
- Les librairies graphiques : GD, libgd libpng, libjpeg…
- Compilateur : gcc, gcc-gc++

3.2 Installation de Nagios/Centreon

Les étapes d'installation et de configuration de « Nagios3-2.3 » et ses plugins « Nagios-plugins-1.4.15 », « Centreon-2.1.10 » et « NDOUtils-1.4b9 » seront détaillées dans l'annexe A.

3.3 Installation de NSClient

Pour la supervision des serveurs Windows, je vais installer le greffon NSClient sur la machine distante et vérifier la présence de la commande « check_nt » parmi les plugins installé de Nagios. Les étapes d'installation seront détaillées dans l'annexe B.

3.4 Installation de NRPE

Pour la supervision des serveurs Linux, je vais installer le greffon « NRPE-2.1.12 » sur la machine distante et vérifier la présence de la commande « check_nt » parmi les plugins installé de Nagios. Les d'installation étapes seront détaillées dans l'annexe C.

4.Interfaces de Nagios/Centreon

4.1 Centreon

- **Authentification**

La première étape à faire avant d'accéder à l'interface de Centreon c'est d'ouvrir un navigateur web et écrire dans la barre de navigation « http://localhost/centreon ». Une page d'authentification s'affiche demandant le nom de l'utilisateur ainsi que le mot de passe comme l'indique la figure 10.

Figure 10. Page d'authentification

➢ **Tactical Overview**

La figure 11 est la première vue après l'authentification, elle nous propose l'essentiel des informations importantes qui sont : l'état de fonctionnement du système d'information supervisé, le nombre d'alertes actuelles, etc.

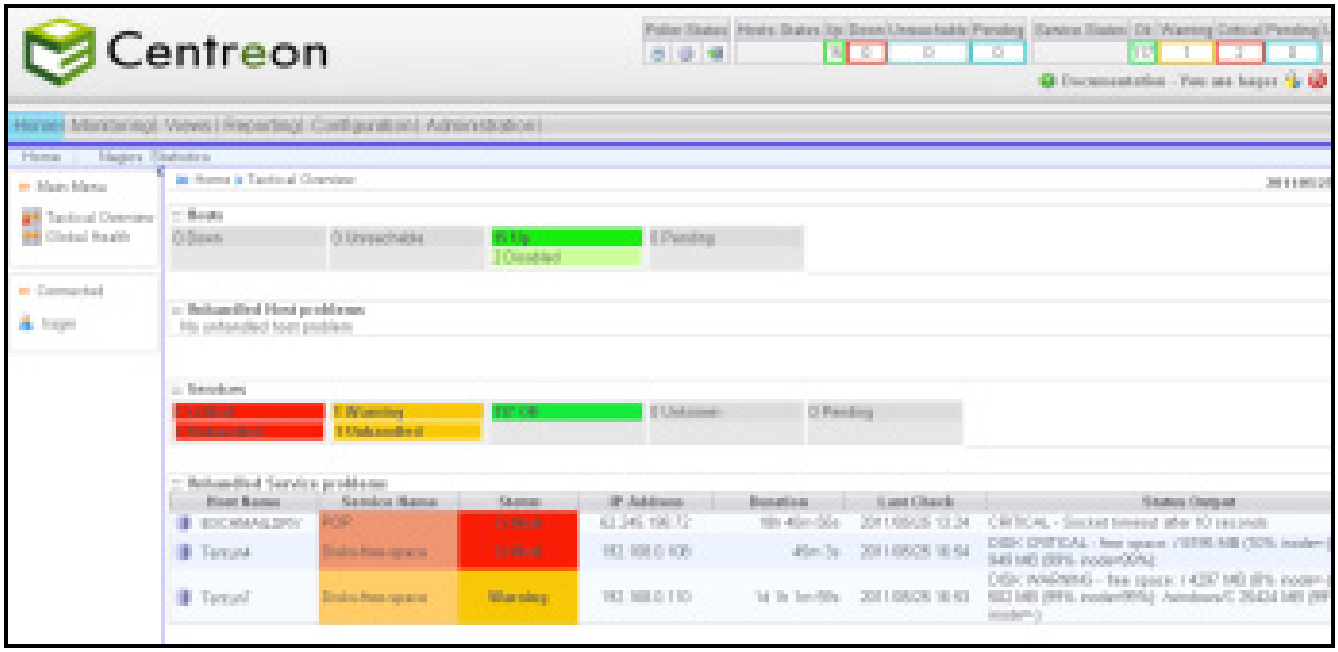


Figure 11. Interface de Vue Globale

➢ **Santé globale**

Cette vue nous permet d'avoir en représentation dite en "camembert", un état de santé globale de notre supervision.

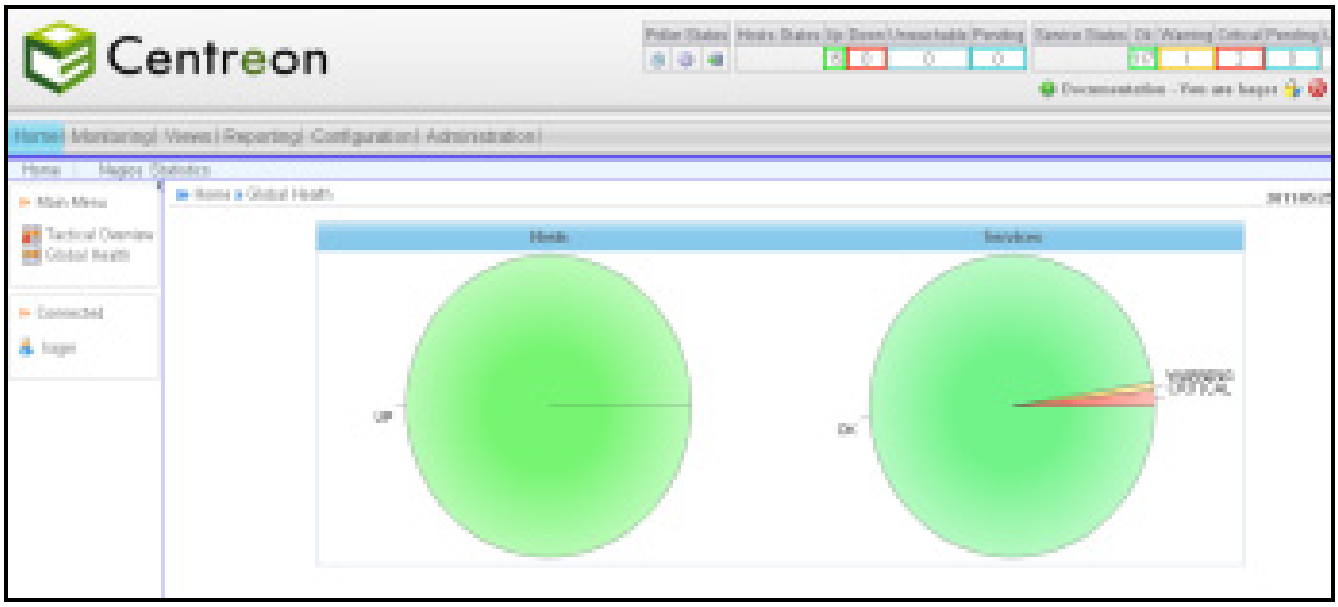


Figure 12. Interface de la santé globale

➢ **Statistique de Nagios**

Dans cette vue, on retrouve les performances de notre supervision (temps de check, latence etc..) et des graphiques traçant l'historique de performance de chacun de nos instances.

- ***Temps de check, latence***

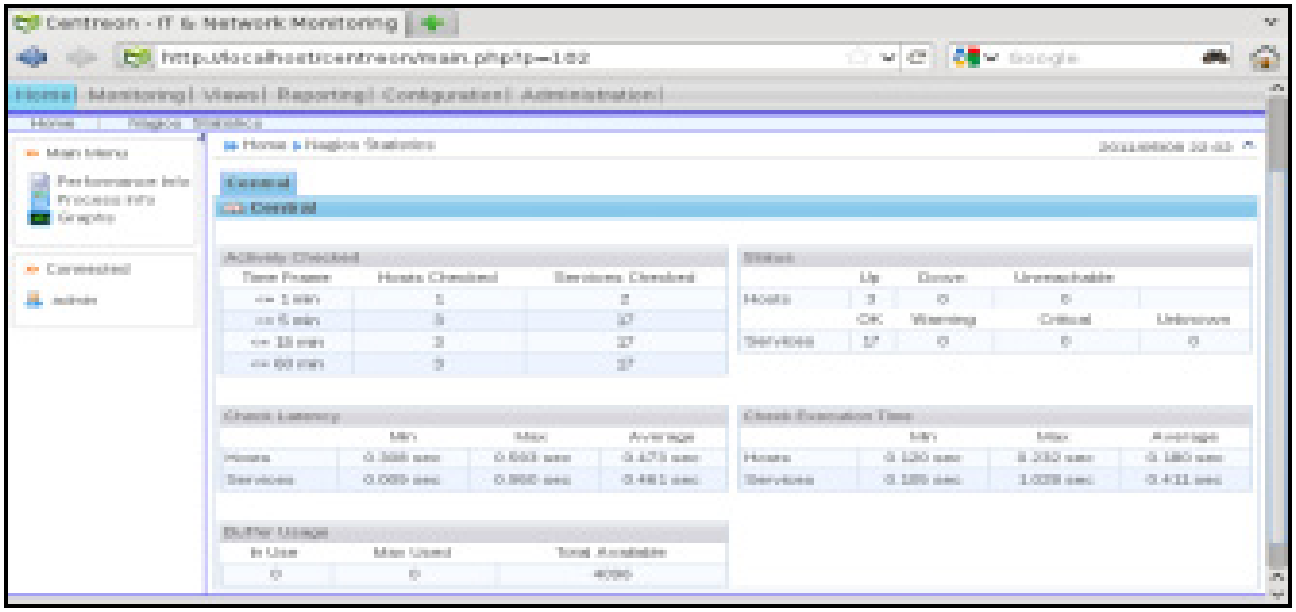


Figure 13. Interface des statistiques Nagios

- ***Graphique de performance***

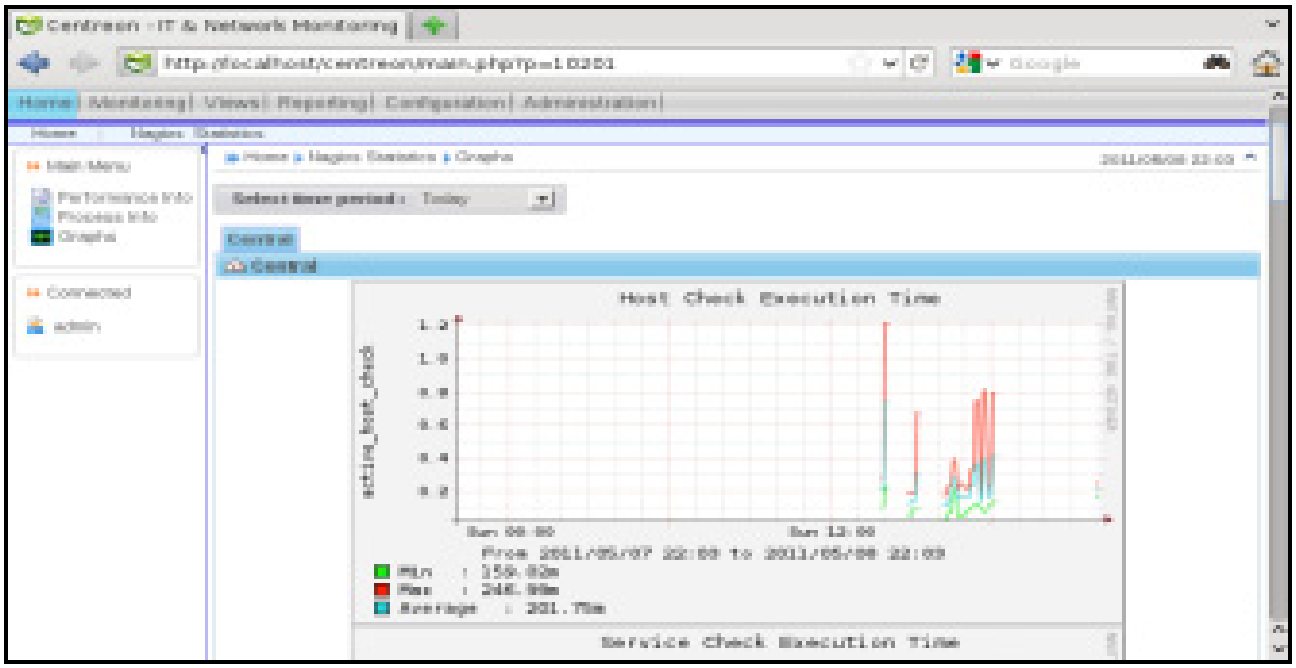


Figure 14. Interface de graphiques de performance

➢ **Monitoring**

Cette vue va nous permettre d'accéder à nos hôtes et nos services supervisés.

- ***Les hôtes***

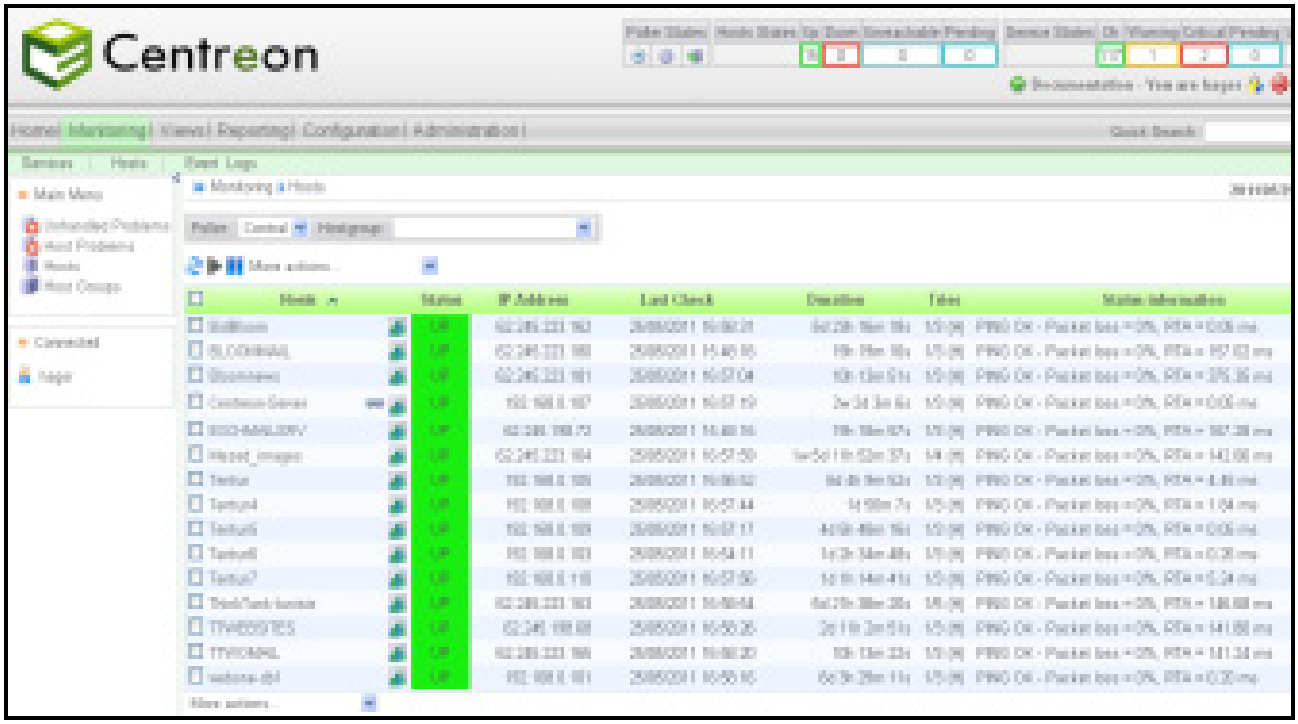


Figure 15. Interface des hôtes supervisées

- ***Les services***

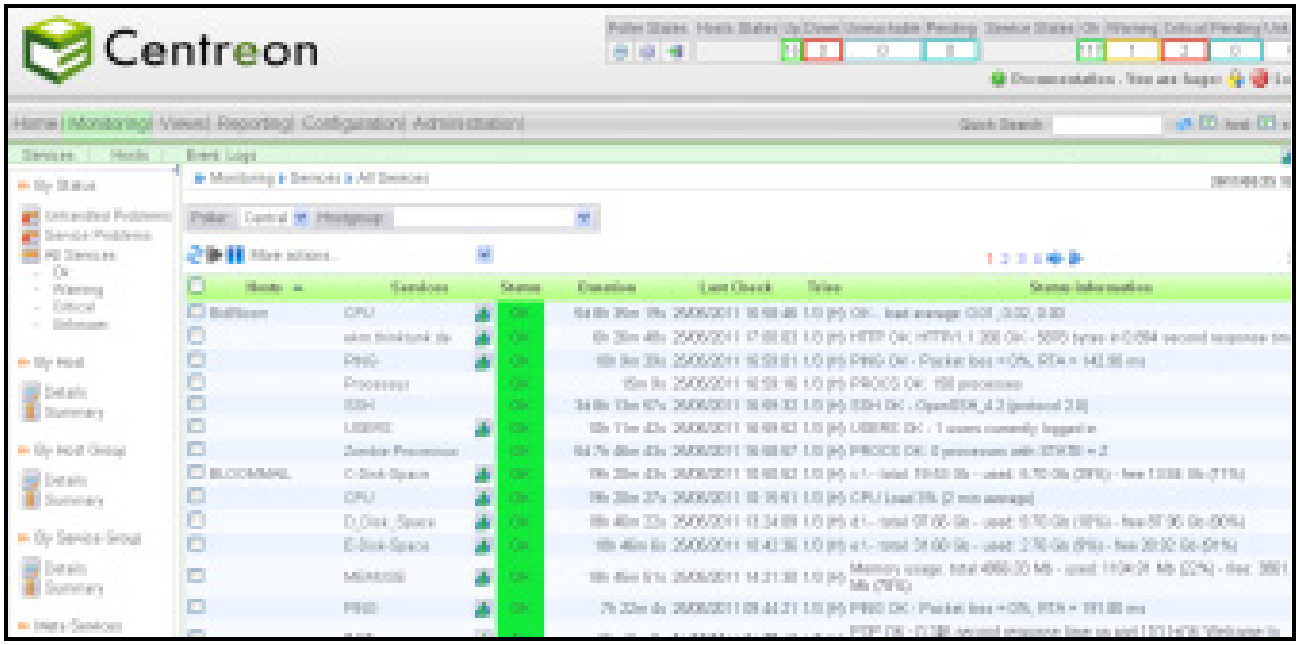


Figure 16. Interface des services supervisés

➢ **Event logs**

Dans cette vue, nous aurons accès à tout l'historique des journaux d'évènements concernant Centreon (Nagios).

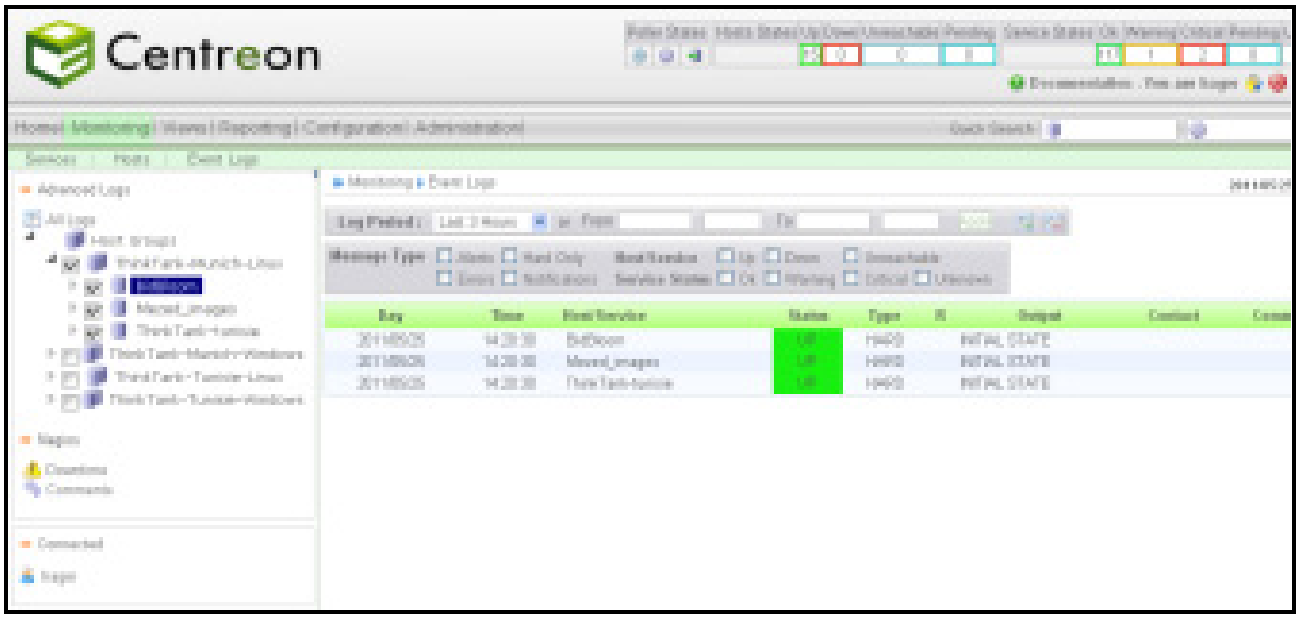


Figure 17. Interface des journaux d'évènements

➢ **Views**

Cette vue permet de voir, de créer, de paramétrer des Templates de graphiques pour les exploiter ensuite pour vos hôtes et services.

Figure 18. Interface de Views

➢ **Reporting**

Cette vue vous permet d'avoir des statistiques de fiabilité de chaque hôte sur une période de temps données.

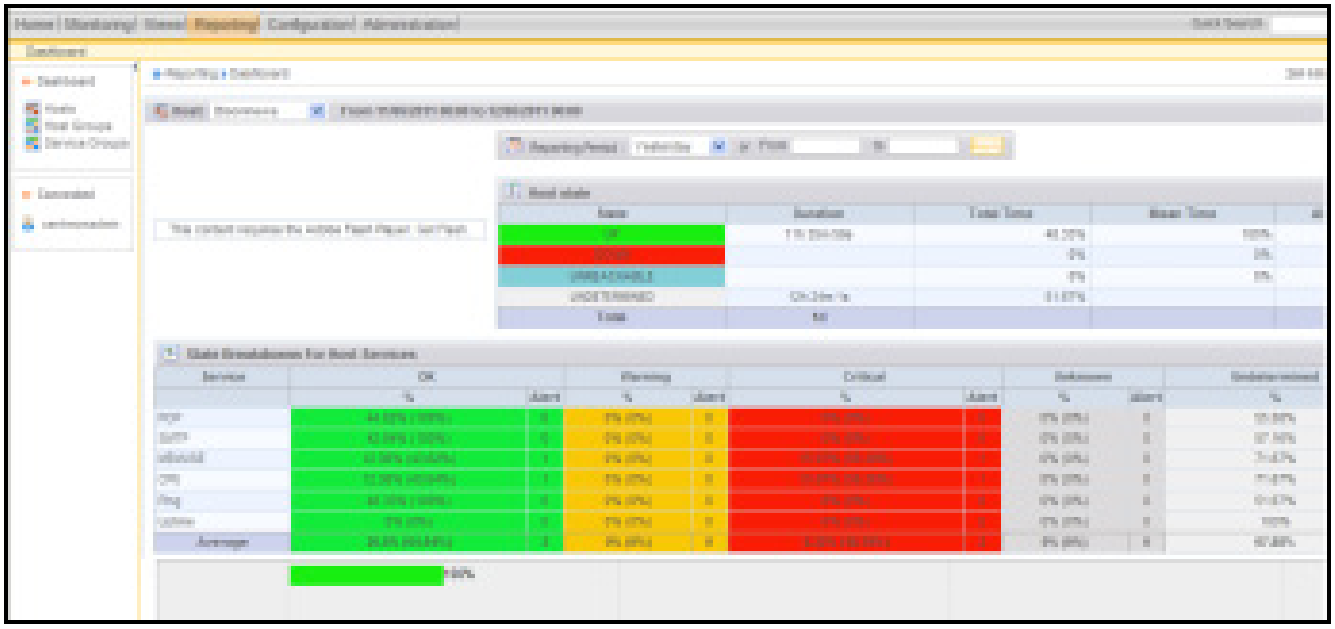


Figure 19. Interface des rapports

5. Exemple d'Utilisations

5.1 Utilisation des Templates pour l'ajout et la supervision des serveurs Windows

Depuis le serveur Nagios, il faut utiliser le plugin standard check_nt pour interroger le démon distant NSClient++ en mode NSClient. Ce plugin possède de nombreuses options:

-H : Hostname

-p : 12489 (port par défaut)

-s : password (définit dans le fichier .NSCI

-w : seuil d'avertissement (warning),

-c : seuil critique

-V : Variable à interroger (MEMUSE, USEDDISKSPACE, UPTIME,..).Ils seront expliqués ultérieurement.

Voici les étapes nécessaires pour l'ajout et la supervision d'un serveur Windows depuis l'interface de Centreon :

❖ **<u>Etape1</u> : Ajout des commandes dans l'interface suivante de Centreon :**

Dans l'interface **Configuration/Commands**, On doit ajouter les commandes checks qui nous permettront de relever les informations de supervision voulues depuis le serveur distant.

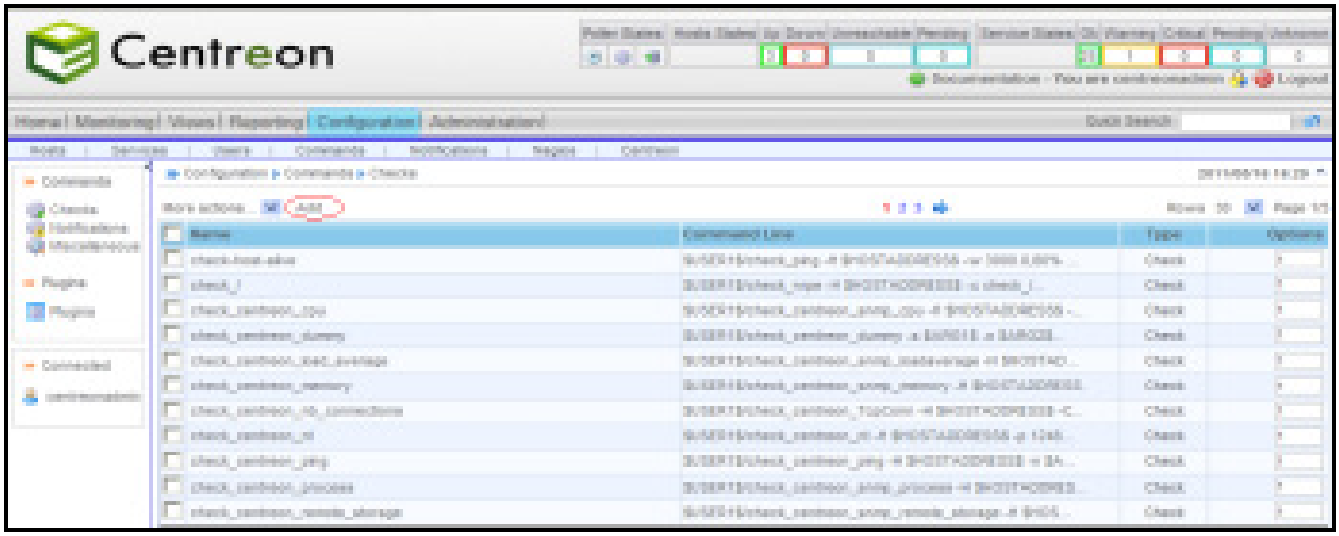


Figure 20. Interface des listes des commandes

L'appuie sur « **add** » nous ramène à l'interface suivante pour la définition des commandes:

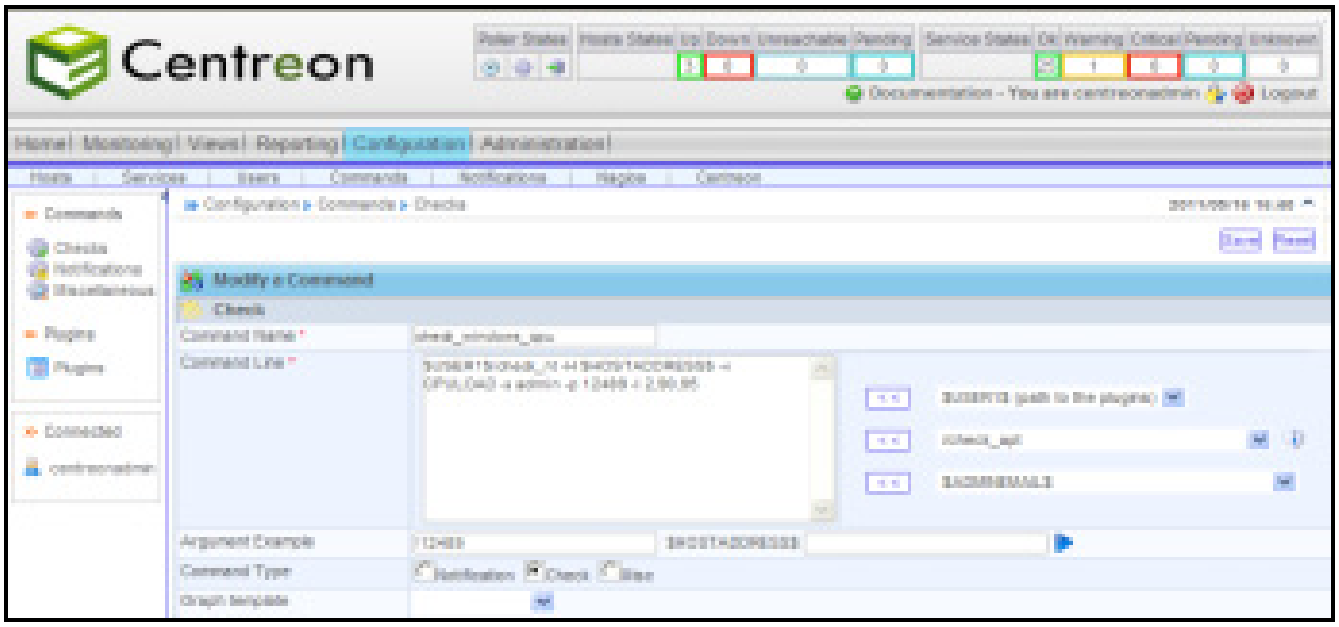


Figure 21. Interface de définition des commandes

De la même manière toutes ces commandes seront définies :

Commandes	Syntaxes	Significations
Check-Wind-CPU	**check_nt** -H $HOSTADDRESS$ -v CPULOAD -s admin -p 12489 -l 2,90,95	Permet de déterminer la charge moyenne système durant les x dernières minutes, avec les seuils « warning » 90% et « critical » 95%. (-l 2,90,95).
Check-Wind-C	**check_nt -H** $HOSTADDRESS$ **-v** USEDDISKSPACE **-s** admin **-p** 12489 **-l** c	Permet de déterminer la taille du disk C (-l)
Check-Wind-D	**check_nt -H** $HOSTADDRESS$ **-v** USEDDISKSPACE **-s** admin **-p** 12489 **-l** d	Permet de déterminer la taille du disk D (-l)
Check-Wind-E	**check_nt -H** $HOSTADDRESS$ **-v** USEDDISKSPACE **-s** admin **-p** 12489 **-l** e	Permet de déterminer la taille du disk E (-l)
Check-Wind-UPTIME	**check_nt -H** $HOSTADDRESS$ **-v** MEMUSE **-s** admin **-p** 12489 **-w** 80 -c 90	Permet de déterminer la taille et pourcentage de la mémoire utilisée, libre et totale. A 80% de la mémoire utilisée l'état du service devient « warning » et à 90% l'état du service devient « critical »
Check-Wind-MEMUSE	**check_nt -H** $HOSTADDRESS$ **-v** UPTIME **-s** pdw **-p** 12489	Permet de déterminer la durée écoulée depuis le dernier redémarrage

Tableau 4. Les commandes NSClient

- **<u>Etape2</u> : Associer chaque commande à un Template de service :**

- L'option « Add » nous renvoie vers une interface ou nous devons définir notre « Service Template » et l'associer à sa commande relative.
- Ainsi on définit les Templates propre à chaque commande crée dans la partie 1.

La figure 22 suivante, présente l'interface d'ajout des nouveaux Services Templates.

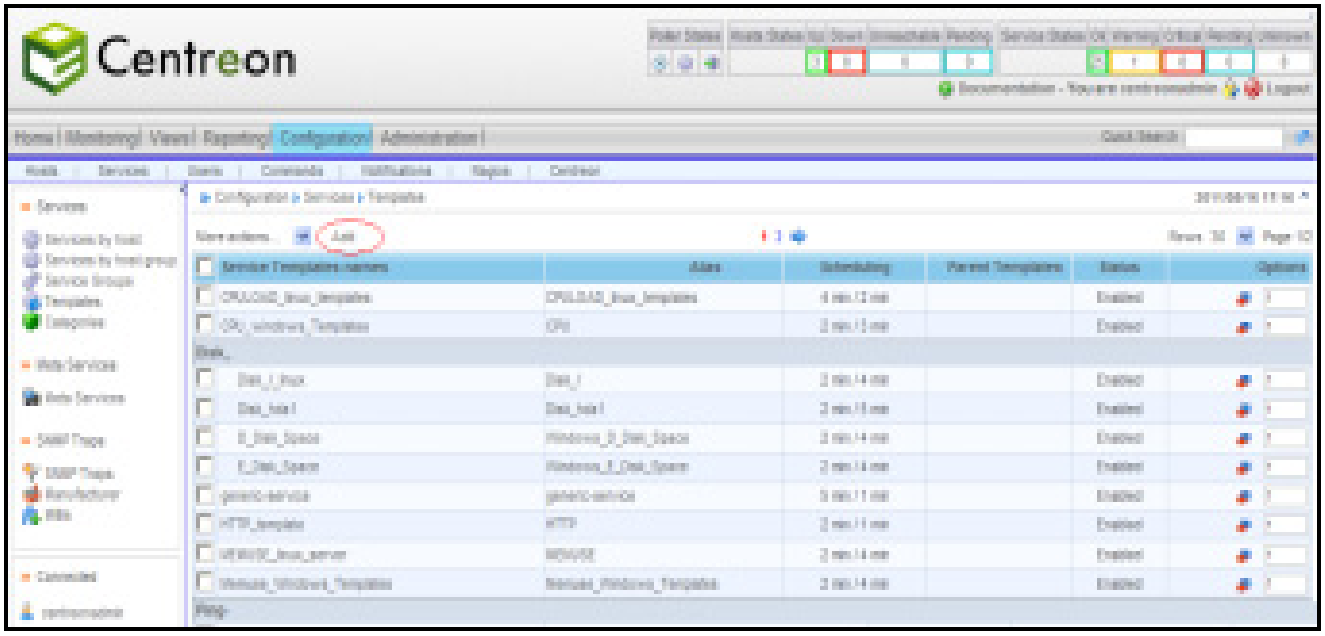


Figure 22. Interface de liste des Templates des services

❖ **<u>Etape 3 : Associer les « Services Templates » à un « Host Template » :</u>**

Dans l'interface **commands/hosts/Templates** on crée le Template « Windows-servers »

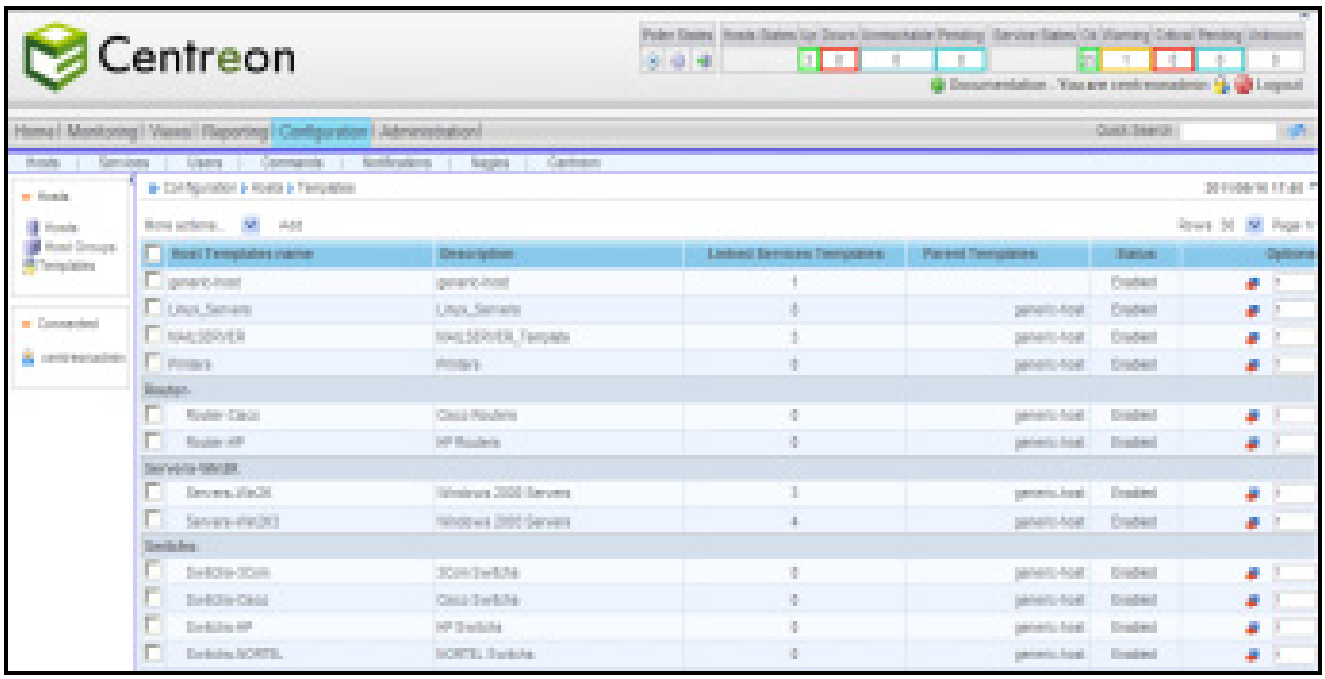


Figure 23. Interface de liste des Templates des hôtes

- L'option « **add** » nous ramène à une interface où nous devrons définir notre nouvelle Template.
- Toutes les « Services Templates » crées dans la partie précédentes doivent s'associer à notre « Host Template » nommée « Windows-Servers-Template ». Comme on le voit dans l'interface commands>Hosts>Templates>Relation, une liste des « services Templates » apparaît, on doit donc sélectionner et ajouter les services qu'on voudra lier à ce « Windows-Servers-Templates ».

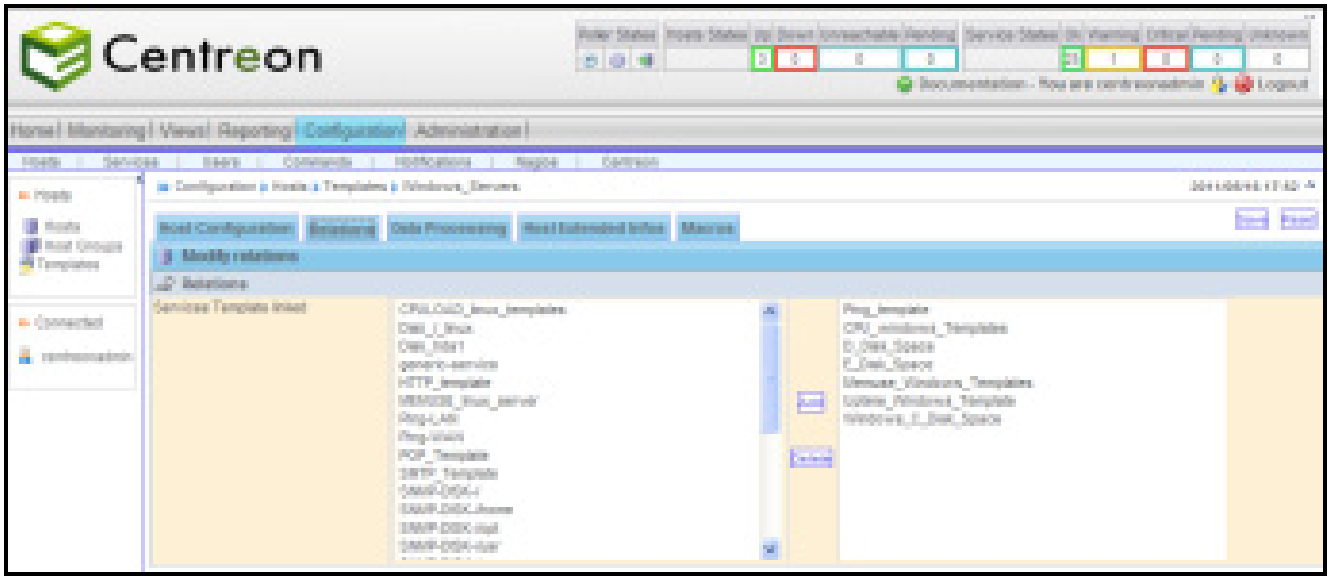


Figure 24. Interface d'association des Templates de services à un Template d'hôte

Ainsi Il ne nous reste plus qu'à ajouter les coordonnées du serveur Windows à configurer (nom, adresse IP, Host Template,…) à travers l'interface **Configuration/Hosts/Add**.

A chaque ajout d'un serveur Windows on n'a pas à refaire les mêmes étapes, on doit seulement lui affecter le « Host Template » adéquat pour que tous les services associés à ce Template apparaissent automatiquement.

❖ **<u>Etape4 : Exportation de la configuration vers Nagios</u>**

En fait lorsque nous modifions la configuration dans Centreon, nous ne faisons que modifier l'état de la base Centreon. Les modifications ne sont pas encore prises en compte par les différents collecteurs Nagios.

Pour effectuer cette mise à jour, il faut se rendre au menu **Configuration / Nagios** puis cliquer sur les boutons:

- **"Move export files":** pour déplacer physiquement les fichiers de configuration dans l'arborescence Nagios.
- **"Restart Nagios":** Pour le redémarrage de Nagios afin que la configuration soit prise en compte.
- Puis cliquer sur "Export"

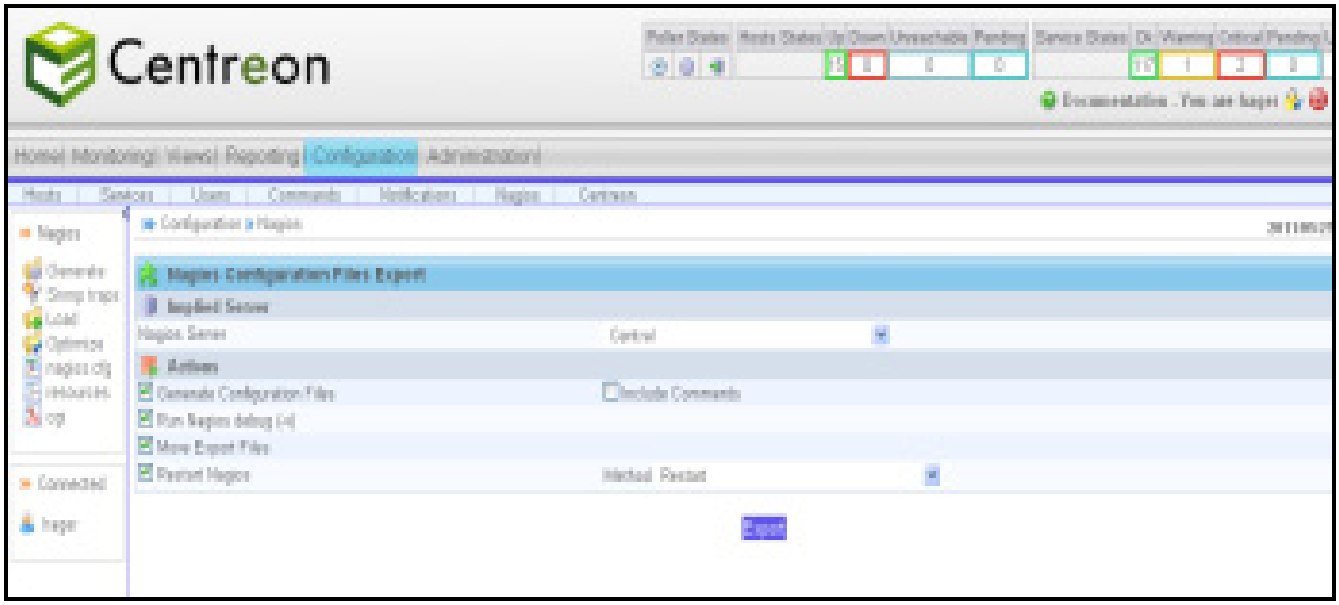


Figure 25. Interface d'exportation

Si tout se passe bien, nous ne devrions pas avoir de messages d'erreurs comme suit:

Quelques minutes après l'exportation, l'hôte ajouté apparaitra dans l'interface « Monitoring » de Centreon, accompagné de ses services. On peut remarquer dans cette figure, la figure 26, l'apparition du nom de la nouvelle hôte, encerclé par la couleur rouge, ainsi que ses services relatifs comme le prouve la figure 27.

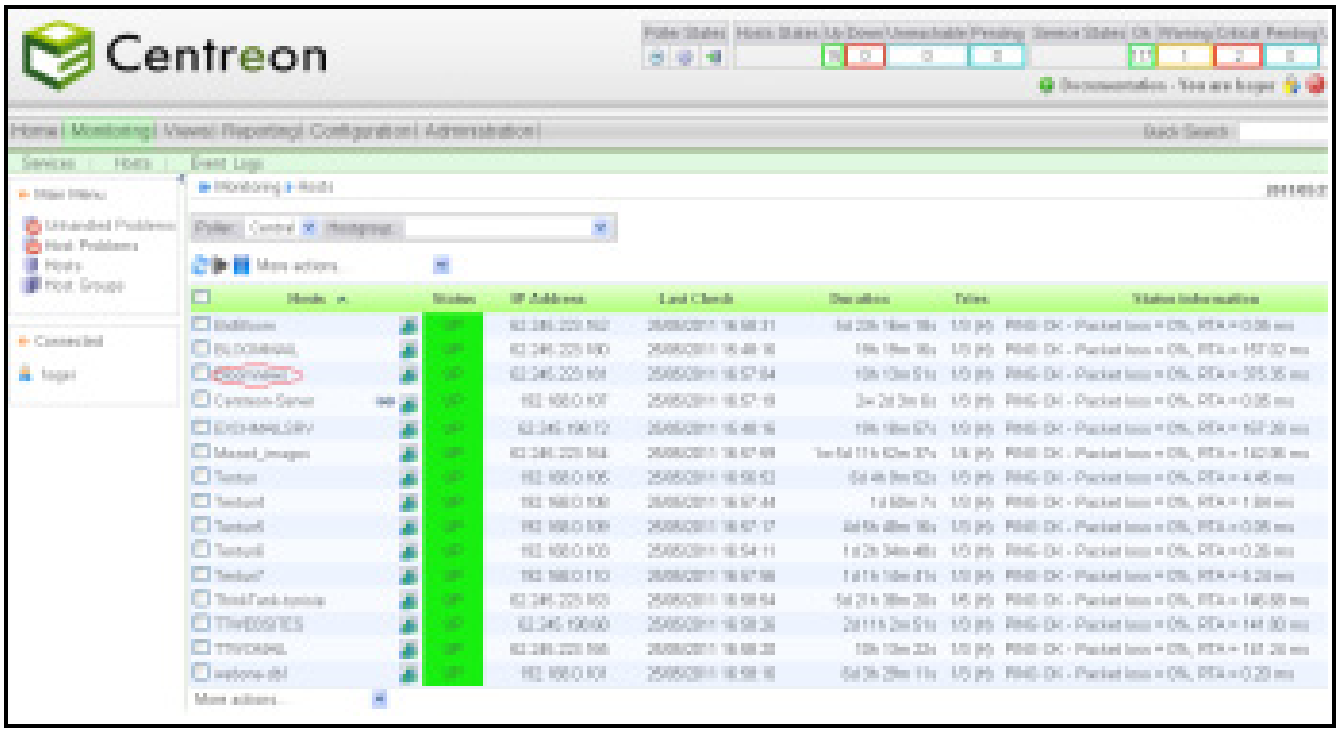


Figure 26. Etat des hôtes supervisés dans Centreon

Les services apparaissent automatiquement avec l'ajout d'hôte, ce qui st claire dans cette interface :

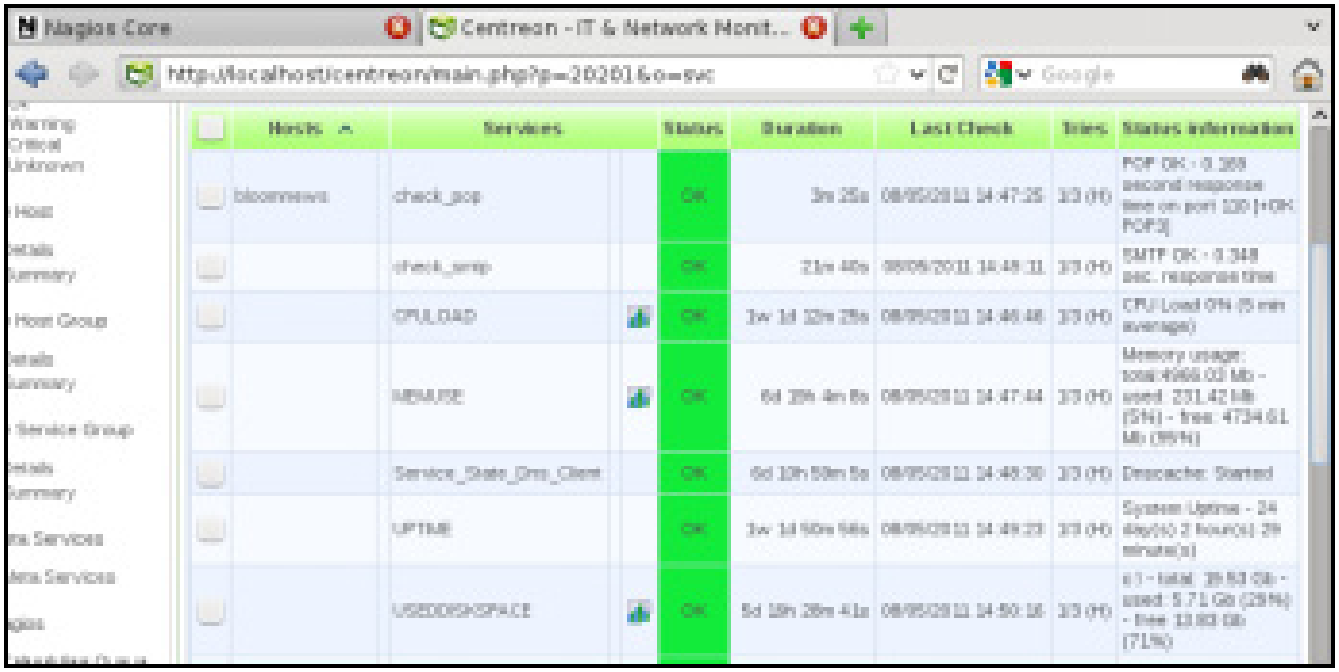


Figure 27. Etat des services supervisés dans Centreon

La même hôte et ses services apparaissent dans l'interface de Nagios après l'exportation de Centreon vers Nagios :

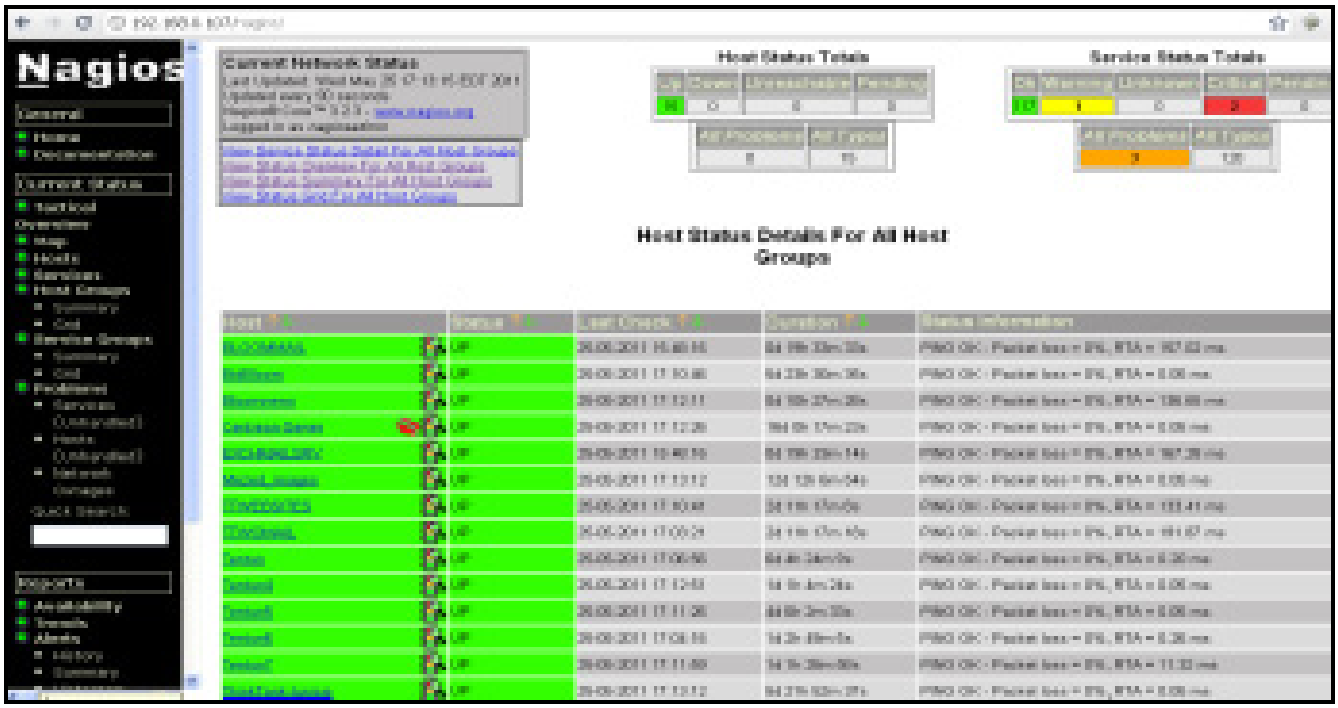


Figure 28. Interface des hôtes supervisées dans Nagios

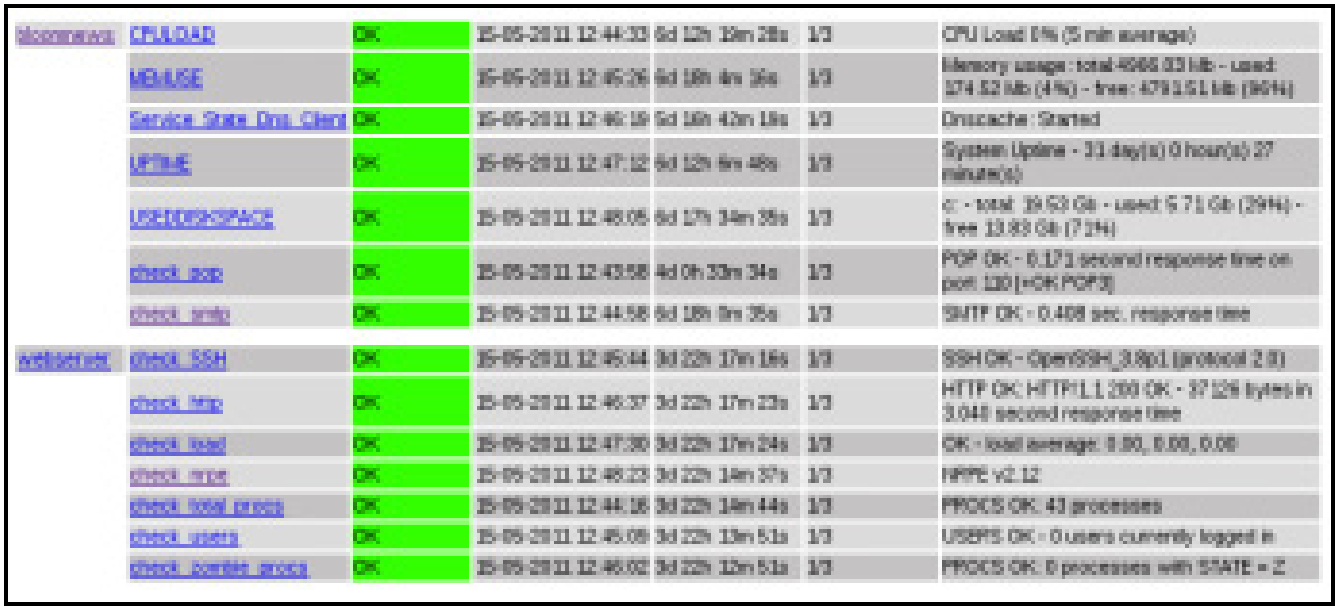


Figure 29. Interface des services supervisés dans Nagios

5.2 Utilisation des Templates pour l'ajout et la supervision des serveurs Linux

Puisque NRPE a la particularité d'exécuter les commandes réclamé par le serveur Nagios dans la machine Linux distante à superviser, on doit avoir cet ensemble de commandes définies dans le fichier de configuration nrpe.cfg de la machine à superviser.

Ligne ajoutée	Significations
Command[check_users]=/usr/local/nagios/libexec/check_users	Permet de déterminer le nombre d'utilisateurs connectés.
Command[check_load]=/usr/local/nagios/libexec/check_load –w 15,10,5 –c 30,25,20	Permet de déterminer la charge CPU
Command[check_hda1]=/usr/local/nagios/libexec/check_disk –w 20% -c 10% -p /dev/hda1	Permet de déterminer l'espace disk restant sur la partition /dev/hda1
Command[check_/]=/usr/local/nagios/libexec/check_disk –w 20% -c 10% -p /	Permet de déterminer l'espace disk restant sur la partition /
Command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -s Z	Permet de déterminer le nombre de processus zombie.
Command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs	Permet de déterminer le nombre de processus.
Command[check_memory]=/usr/local/nagios/libexec/check_memory.pl–w 10% -c 5%	Permet de déterminer la taille restante de la mémoire.

Tableau 5. Les commandes NRPE

➢ **Remarques :**

- Les variables check_users, check_disk, etc.…Sont déjà présents dans le fichier /usr/local/nagios/libexec à l'installation des plugins dans la machines distante, dont les fonctionnalités sont déjà expliqués dans le chapitre 2 de mon rapport.

- la variable check_memory est un script perl à ajouter parmi ceux déjà existants dans /usr/local/nagios/libexec. Voir [annexe D]
- Ces commandes seront appelés depuis le serveur nagios seulement par leur nom indiqué entre [ ] et de la manière suivante :

Check_nrpe –H <@machine distante> **-c** <nom de la commande>

Voici les étapes nécessaires pour l'ajout et la supervision d'un serveur Linux depuis l'interface de Centreon :

❖ **<u>Etape1</u> : Ajout des commandes dans l'interface suivante de Centreon :**

Dans l'interface Configuration>Commands, On doit ajouter les commandes checks qui nous permettront de relever les informations de supervision voulues depuis le serveur distant.

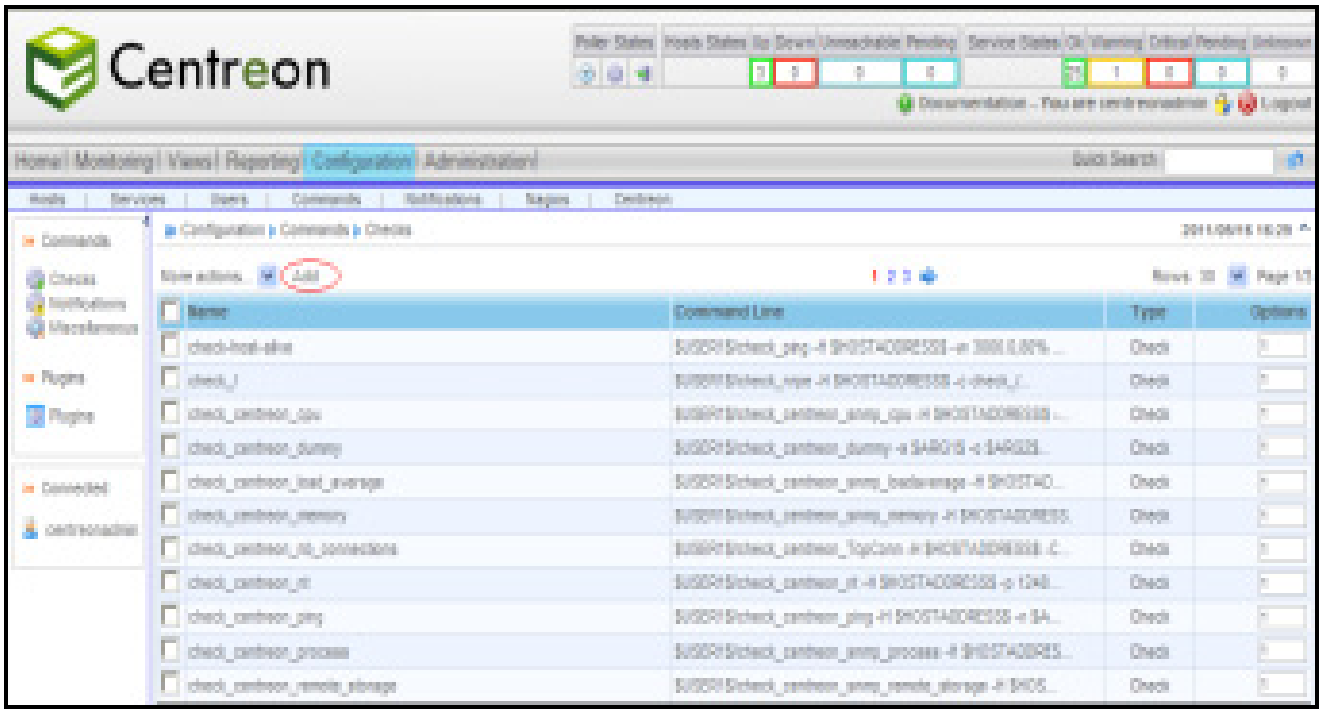


Figure 30. Interface des services supervisés dans Nagios

➢ L'appuie sur « **add** » nous ramène à l'interface suivante pour la définition des commandes:

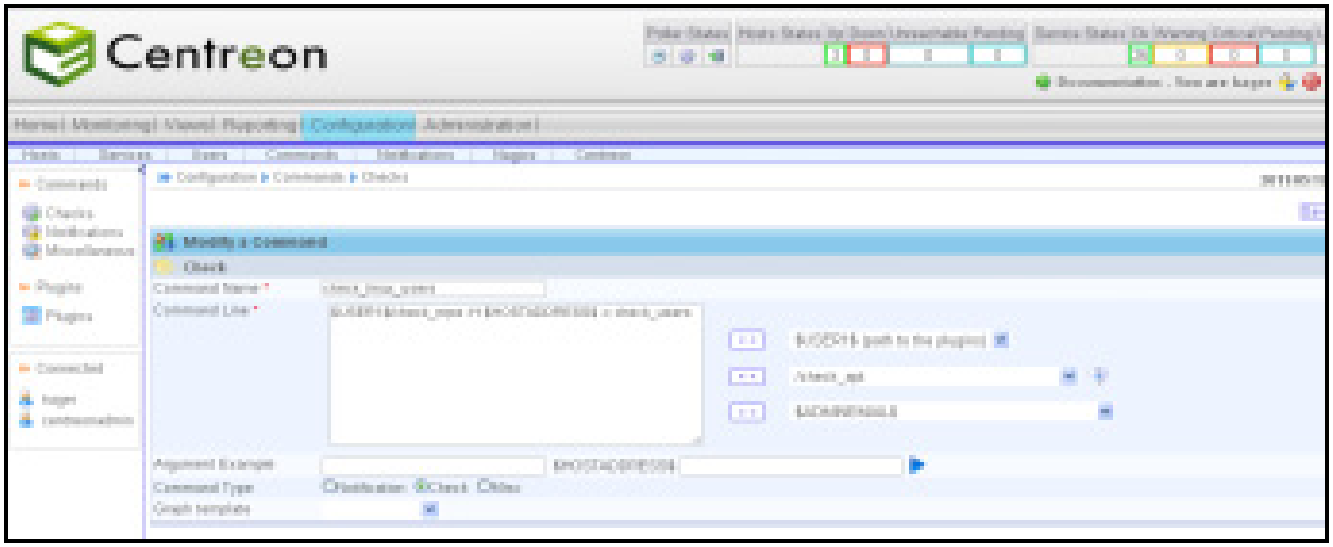


Figure 31. Interface des services supervisés dans Nagios

De la même manière toutes ces commandes seront définies :

❖ **<u>Etape2</u> : Associer chaque commande à un Template de service :**

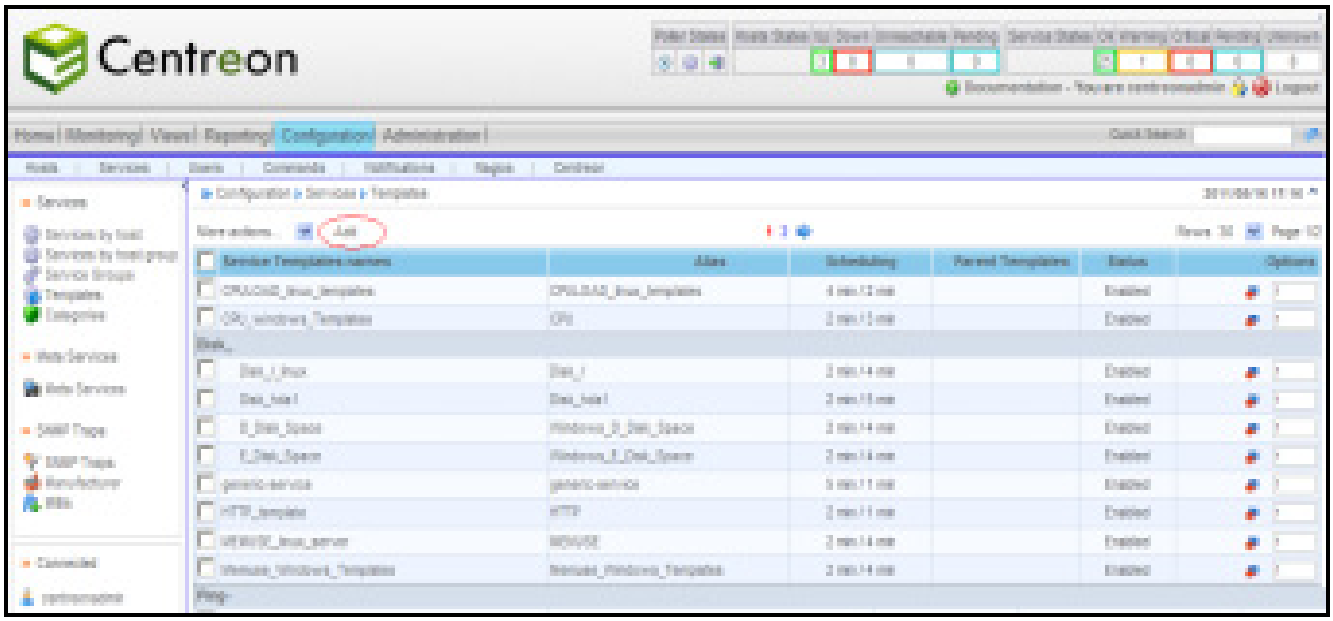


Figure 32. Interface des services supervisés dans Nagios

- L'option « Add » encerclé par la couleur rouge, comme le montre la figure 32, nous renvoie vers une interface ou nous devons définir notre « Service Template » et l'associer à sa commande relative.
- Ainsi on définit les Templates propre à chaque commande crée dans la partie 1.

❖ **<u>Etape 3 : Associer les « Services Templates » à un « Host Template » :</u>**

Dans l'interface présenté par la figure 33, on crée le Template « Linux_servers »

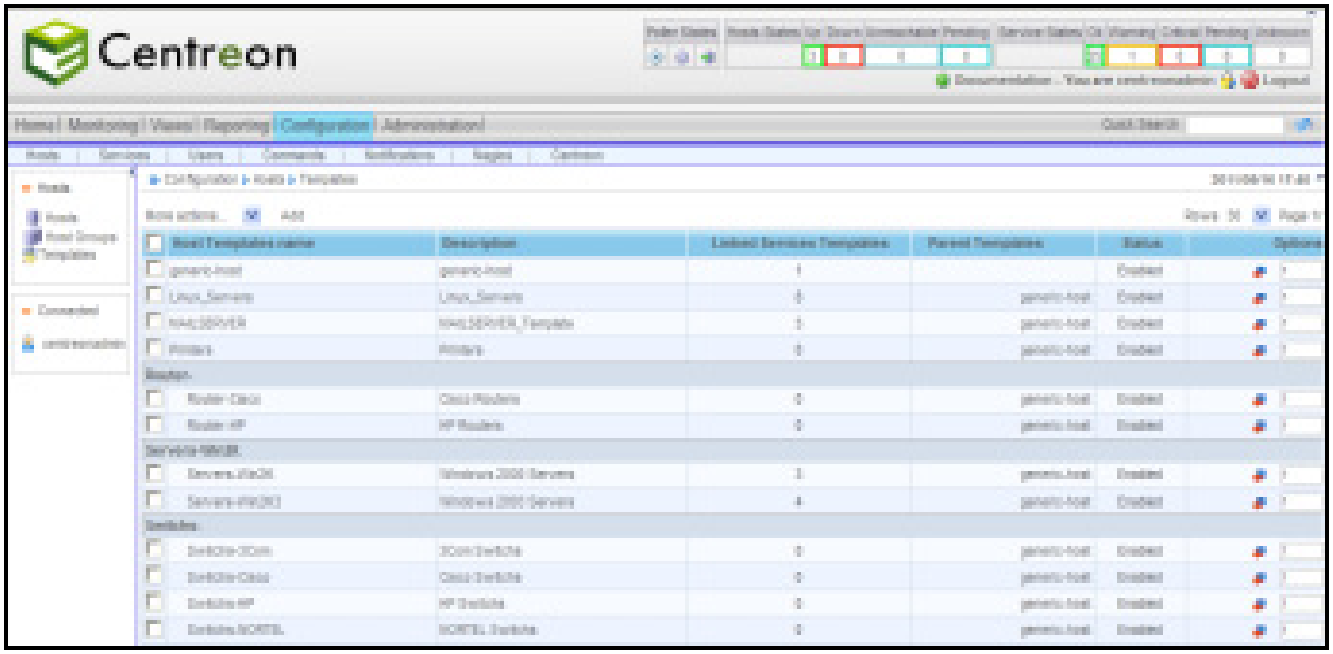


Figure 33. Liste des Templates des hôtes

- L'option « add » nous ramène à une interface où nous devrons définir notre nouvelle Template.
- Tout les « Services Templates » crées dans la partie précédente doivent s'associer à notre « Host Template » nommé « Linux-Servers-Template ». Comme on le voit dans l'interface **commands/Hosts/Templates/Relation**, une liste des « services Templates » apparaît, on doit donc sélectionner et ajouter les services qu'on voudra lier à ce « Linux-Servers-Templates ».

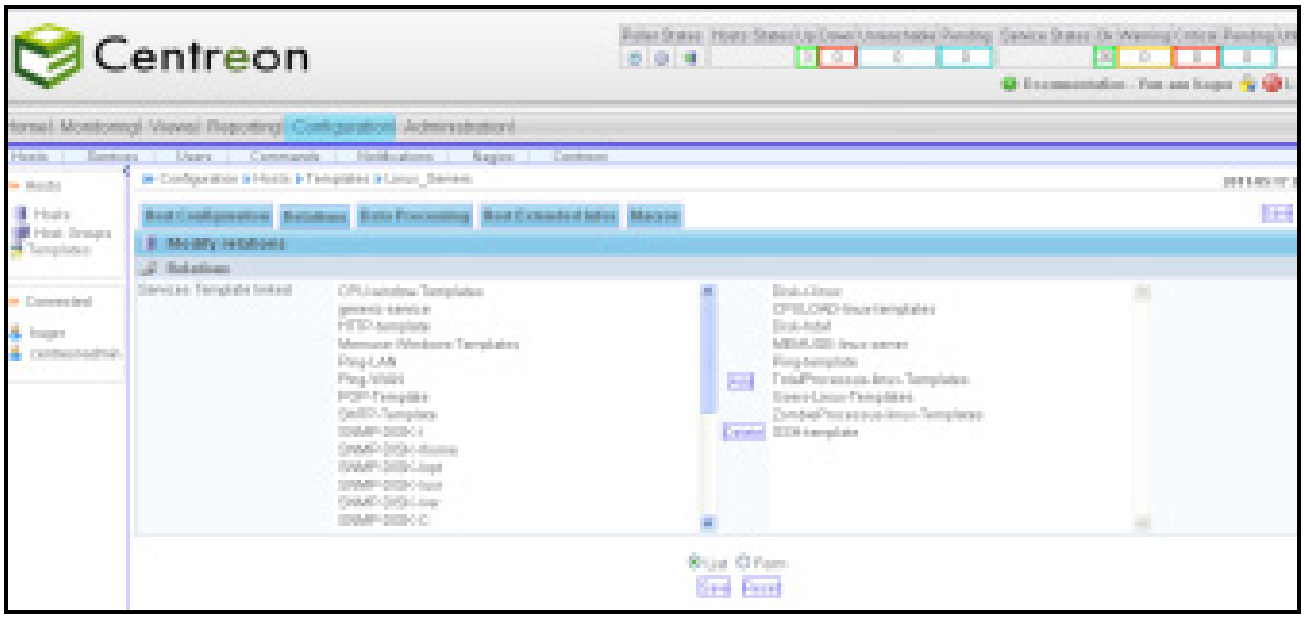


Figure 34. Liste des Templates de service à associer à une hôte

- Ainsi Il ne nous reste plus qu'à ajouter les coordonnées du serveur Linux à configurer (nom, adresse IP, Host Template,…) à travers l'interface **Configuration/Hosts/Add**.

A chaque ajout d'un serveur Linux on n'a pas à refaire les mêmes étapes, on doit seulement lui affecter le « Host Template » adéquat pour que tous les services associés à ce Template apparaissent automatiquement.

5.3 Notification par mail

En plus d'être informé visuellement par l'interface de Centreon ou Nagios, on peut paramétrer l'envoi des mails pour indiquer la perte d'un hôte ou d'un service. Cela permet d'avoir des informations supplémentaires, et d'avoir un historique de l'activité durant la nuit lorsque l'on ouvre sa boite mail le matin.

En premier lieu on aura besoin d'installer les éléments postfix et mailx et avoir accès à un serveur SMTP (propre à la société). Voir [Annexe C]

Nagios possède déjà les commandes de notification « host-notify-by-email » et « service-notify-by-email » dans la partie configuration> commands>notifications qui seront paramétrées à des hôtes ou services lors de leur création, ainsi on gardera la même configuration à chaque nouvel ajout.

Il nous reste qu'à informer le système des utilisateurs et groupes d'utilisateurs à notifier lors de l'apparition d'un problème et de sélectionner la durée de notification.

Cette configuration est claire dans la figure 35.

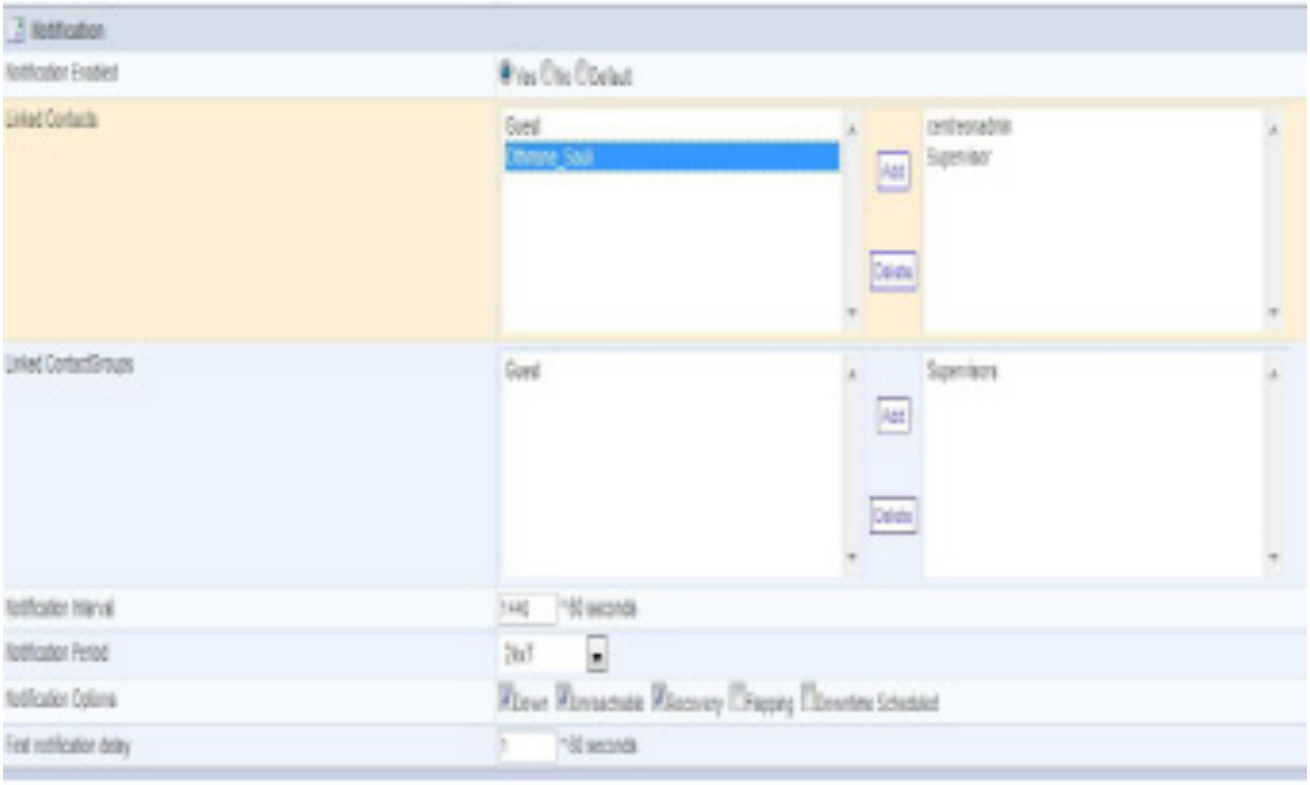
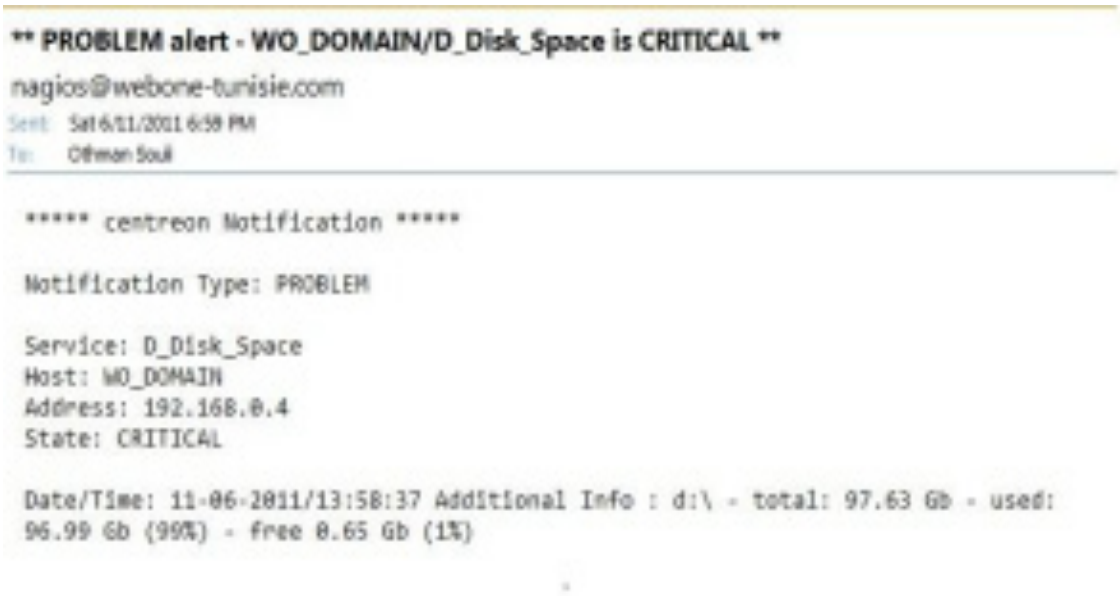


Figure 35. Configuration des notifications

Cette interface modélise la configuration d'envoie de notification, où nous devrons sélectionner les utilisateurs concernées par la réception de ces notifications, ainsi on définira l'intervalle de notification, la période de notification (24*7, workhours,..) et le type de notifications.

Ainsi on recevra un mail dans ma boite ayant la forme suivante :



** PROBLEM alert - WO_DOMAIN/D_Disk_Space is CRITICAL **

nagios@webone-tunisie.com

Sent: Sat 6/11/2011 6:58 PM

To: Othman Soufi

```
***** centreon Notification *****

Notification Type: PROBLEM

Service: D_Disk_Space
Host: WO_DOMAIN
Address: 192.168.0.4
State: CRITICAL

Date/Time: 11-06-2011/13:58:37 Additional Info : d:\ - total: 97.63 Gb - used:
96.99 Gb (99%) - free 0.65 Gb (1%)
```

Figure 36. Exemple de notification

6. Conclusion

Dans ce chapitre j'ai penchés sur l'aspect pratique de mon projet, en détaillant les étapes de la mise en place et l'utilisation de ma solution, et j'ai ainsi pu prouver l'apport important de Centreon à Nagios, qui est principalement, la facilité de la configuration, mais aussi la livraison de comptes rendus et d'analyses plus rapidement et d'une manière beaucoup plus précise pour le seul but de gagner et optimiser la gestion de son temps.

Conclusion générale

Le domaine de la supervision est un domaine important de l'administration systèmes et réseaux. En constante évolution, les solutions libres de supervision ont prouvé qu'elles avaient leur place dans la sphère professionnelle.

Et comme je l'ai déjà explicité dans mon étude, la supervision est un des moyens indispensables pour favoriser la croissance de rendement d'une entreprise. Le propos de ce projet était de choisir une solution qui répandait aux besoins organisationnels et financiers de l'entreprise et il n'y'avait pas mieux pour combler ce besoin que Nagios.

L'association de Nagios et de Centreon a permis la constitution d'une solution de monitoring à la fois puissante et efficace.

Centreon agit comme un intermédiaire entre l'administrateur et les fichiers de configuration de Nagios. Il enregistre dans une base de données les configurations effectuées par l'administrateur, puis il modifie les fichiers de configuration de Nagios en fonction du contenu de la base de données. Ce qui a permis de simplifier grandement le travail de l'administrateur, contrairement à l'utilisation de Nagios seul.

Ce stage ma permis d'acquérir maintes connaissances dans le monde de la supervision des réseaux informatiques, et surtout la maitrise de l'environnement Unix.

Références netographiques

1. **http://www.nagios.org/** : le site officiel de Nagios
2. **http://www.nagios.sourceforge.net/** : documentation complète sur les fichiers de Nagios
3. **http://www.nagios.org/support/**
4. **http://www.centreon.com/** : Le site officiel de Centreon
5. **http://dokuwiki.ruusan.org/administration/nagios :** Un site d'installation de Nagios et Centreon
6. **http://wiki.monitoring-fr.org/infra/postfix :** Un tutoriel pour l'installation et la configuration de POSTFIX.

Annexe A

❖ Installation de Nagios-3.2.3

➢ Installation des librairies et pré-requis nécessaires

Installation de puis le yast :

- **Apache2** :apache2, apache2-mod-php5, apache2-mod-perl
- **PHP5 :** php5, php5-gd,php5-Ldap,php5-mysql
- **Libraries GD:** gd, libpng, libjpeg
- **Compilateur:** gcc, gcc-c++
- **Interpréteur perl:** perl,perl-config-Inifiles,perl-crypt-dos,perl-digest-hmac

➢ Installation de Nagios

Pour éviter des problèmes de sécurité, il est préférable de lancer «Nagios» avec un compte utilisateur normal.

- Création d'un compte utilisateur et un groupe dédié au processus nagios.
- Création d'un groupe "nagcmd" permettant l'exécution des commandes externes à travers l'interface web.
- Rajout des utilisateurs Nagios et Apache à l'intérieur du groupe "nagcmd".

```
# cd /usr/sbin
# useradd nagios
# passwd nagios
# groupadd nagios
# groupadd nagcmd
# usermod -G nagios,nagcmd nagios
# usermod -G nagcmd wwwrun
```

- Téléchargement et installation de Nagios

```
#   cd /usr/local/src
#   wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
#  tar xzf nagios-3.2.3.tar.gz
#   cd nagios-3.2.3
#   ./configure –prefix=/usr/local/nagios –with-command-group=nagcmd –enable-event-broker
#   make all
#   make install
#   make  install-init
#   make install-config
#   make install-commandmode
#   make install-webconf
```

➔**Make all :** Compiler les codes sources

Make install : Installer les binaires

Make install-init : Installer les scripts de démarrage

Make install-config : Installer les fichiers de configuration

Make install-commandmode: Installer et configurer les permissions

Make install-webconf : Installer les fichiers de configuration de Nagios dans le répertoire conf d'apache2**.**

- Création d'un compte nagiosadmin pour se connecter à la page web nagios.

```
# htpasswd2  -c  /usr/local/nagios/etc/htpasswd.users    nagiosadmin
```

➢ Installation des plugins de base : nagios-plugins-1.4.15

- Téléchargement et installation des plugins.

```
#  cd /user/local/src
#wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
#   tar xzf nagios-plugins-1.4.15.tar.gz
#   cd nagios-plugins-1.4.15
#   ./configure  --with-nagios-user=nagios  --with-nagios-group=nagios
#   make
#   make install
```

- On peut tester s'il y'a aucune erreur dans la configuration de nagios suite à l'exécution de la commande suivante :

```
# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

➔Cette commande de vérification se fera automatiquement grâce à Centreon et à chaque modification de la configuration de Nagios.

- Il vaut mieux changer la permission du répertoire /usr/local/nagios/var/rw afin de pouvoir effectuer des actions depuis l'interface de nagios.

```
# chown nagios:nagcmd /usr/local/nagios/var/rw
```

❖ Installation NDOUtils-1.4b9

➢ Installation des librairies et pré-requis nécessaires

Installation de puis le yast :

- **MySQL**: mysql, mysql-client.

➢ Installation de NDOutils

```
# cd /usr/local/src
# wget http://prdownloads.sourceforge.net/sourceforge/nagios/ndoutils-1.4b9.tar.gz
# tar xzf ndoutils-1.4b9.tar.gz
# cd ndoutils-1.4b9
# ./configure --prefix=/usr/local/nagios --enable-mysql --disable-pgsql --with-ndo2db-user=nagios --with-ndo2db-group=nagios --with-mysql=/usr/lib/mysql
# make
```

- Installation des binaires et installation des fichiers de configuration des binaires Ndo2db et ndomod.

```
# cp src/ndomod-3x.o /usr/local/nagios/bin/ndomod.o
# cp src/ndo2db-3x /usr/local/nagios/bin/ndo2db
# cp src/log2ndo src/file2sock /usr/local/nagios/bin
# cp config/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db/cfg
# cp config/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
# chmod 774 /usr/local/nagios/bin/ndo*
# chown nagios:nagios /usr/local/nagios/bin/ndo*
# chown nagios:nagios /usr/local/nagios/etc/ndo*
```

- Récupération du script d'utilisation :

```
# cp daemon-init /etc/init.d/ndo2db
# chmod +x /etc/init.d/ndo2db
```

❖ Installation Centreon-2.1.10

```
# cd /usr/local/src
# wget http://downloads.centreon.com/index.php?id=144
# tar xzf centreon-2.1.10.tar.gz
# cd centreon-2.1.10
# ./install.sh -i
```

```
you accept GPL license ?
[y/n], default to [n]:
> y

Do you want to install : Centreon Web Front
[y/n], default to [n]:
> y

Do you want to install : Centreon Nagios Plugins
[y/n], default to [n]:
> y

Do you want to install : Centreon Snmp Traps process
[y/n], default to [n]:
> y

Do you want me to create this directory ? [/usr/local/centreon]
[y/n], default to [n]:> y
Path /usr/local/centreon OK

Do you want me to create this directory ? [/etc/centreon]
[y/n], default to [n]:> y
/usr/local/nagios/bin/nagios OK

Where is your NDO ndomod binary ?
default to [/usr/sbin/ndomod.o]>
 /usr/local/nagios/bin/ndomod.o   OK

Do you want me to configure your sudo ? (WARNING)
[y/n], default to [n]:> y
Configuring Sudo OK

Do you want to add Centreon Apache sub configuration file ?
[y/n], default to [n]:> y
Create '/etc/apache2/conf.d/centreon.conf' OK
Configuring Apache OK

Do you want to reload your Apache ?
[y/n], default to [n]:> y
Reloading Apache service OK

Do you want me to create this directory ? [/var/run/centreon]
[y/n], default to [n]:> y
Path /var/run/centreon OK

Do you want me to create this directory ? [/var/lib/centreon]
[y/n], default to [n]:> y
Path /var/lib/centreon OK
```

```
Do you want me to install CentStorage init script ?
[y/n], default to [n]:> y
CentStorage init script installed OK

Do you want me to install CentStorage run level ?
[y/n], default to [n]:> y

Do you want me to install CentCore init script ?
[y/n], default to [n]:> y
CentCore init script installed OK

Do you want me to install CentCore run level ?
[y/n], default to [n]:> y

Do you want me to create this directory ? [/var/lib/centreon/centplugins]
[y/n], default to [n]:> y
Path /var/lib/centreon/centplugins OK
```

- Une deuxième partie concerne la suite d'installation de Centreon à travers l'interface graphique où on définisse l'administrateur de Nagios, l'accès à la base de données, les noms des bases de Centreon à créer…

Annexe B

❖ Installation de NSClient

➢ Partie Serveur (Machine Windows Distante)

Il faudra installer et configurer NSClient++ sur le serveur Windows

- Télécharger la version NSClient-0.3.8.75.
- Dézipper le client sous le répertoire C:\NSClient++-Win32-0.3.8.
- Ouvrir une commande DOS (cmd.exe)
- Entrer les commandes suivantes :

  C:\cd NSClient++-Win32-0.3.8

  C:\cd NSClient++-Win32-0.3.8\NSClient++.exe/install

L'installation est donc achevée, vérifions donc que le service est autorisé à "Interagir avec le bureau" (marquer Local system account et Allow service to interact with desktop dans l'onglet « Log On » du gestionnaire de service) en ouvrant le gestionnaire des services.

- On passe maintenant à la modification du fichier de configuration sous c://nsclient/NSC.INI.

- Décommenter dans la première section [modules] tout les modules sauf **CheckWMI.dll** et **RemoteConfiguration.dll**

- Décommenter la ligne **allowed_hosts** dans la section [Settings] et ajoutant l'adresse du serveur Nagios aussi pour des mesure de sécurité on a la possibilité d'attribuer un password pour accéder à NSClient.

[Setting]

;# OBFUSCATED PASSWORD

; This is the same as the password option but here you can store the password in an obfuscated manner.

; *NOTICE* obfuscation is *NOT* the same as encryption, someone with access to this file can still figure out the

; password. Its just a bit harder to do it at first glance.

;obfuscated_password=Jw0KAUUdXlAAUwASDAAB

PASSWORD

This is the password (-s) that is required to access NSClient remotely. If you leave this blank everyone will be able to access the daemon remotly.

password=admin

ALLOWED HOST ADDRESSES

This is a comma-delimited list of IP address of hosts that are allowed to talk to the all daemons. If leave this blank anyone can access the deamon remotly (NSClient still requires a valid password).The syntax is host or ip/mask so 192.168.0.0/24 will allow anyone on that subnet access

allowed_hosts= 192.168.0.107

- **Démarrage NSClient:**

  C:\cd NSClient++-Win32-0.3.8\NSClient++.exe/**start**

- **Arrêt NSClient**

  C:\cd NSClient++-Win32-0.3.8\NSClient++.exe/**stop**

➢ Partie Cliente (serveur Nagios)

Juste on doit vérifier la présence de la commande check_nt sous /usr/local/nagios/libexec sinon le télécharger et l'ajouter parmi les autres commandes.

➔Depuis le terminal du serveur nagios testons si la machine Windows distante répond en tapant la commande suivante qui doit renvoyer la version de NSClient++ installée :

```
#cd /usr/local/nagios/libexec
#./check_nt  -H 62.245.223.181 –s admin –p 12489 –v CLIENTVERSION

                NSClient++ 0.3.8.75
```

➔Maintenant que tout est prêt dans la machine Windows distante à superviser, on a plus qu'à ajouter la machine au serveur Nagios et essayer de récupérer les informations nécessaires grâce à la commande **check_nt** qui permet d'interroger à distance l'agent NSClient.

Annexe C

❖ Installation de NRPE

➢ Partie Cliente (Serveur Linux)

Accéder au serveur Linux à superviser en tant que **root** et suivre les étapes suivantes :

- Création d'un utilisateur et groupe.

```
# cd /usr/sbin
# useradd nagios
# passwd nagios
# groupadd nagios
# usermod -G nagios nagios
```

- Téléchargement, décompression et Installation des plugins Nagios Nagios-plugins-1.4.15

```
#mkdir downloads
#cd downloads
#wget http://osdn.dl.sourceforge.net/sourceforge/nagiosaplug/nagios-plugins-1.4.15.tar.gz
# tar xzf nagios-plugins-1.4.6.tar.gz
#cd nagios-plugins-1.4.6
#./configure
#make
#make install
#chown nagios.nagios /usr/local/nagios
#chown –R nagios.nagios /usr/local/nagios/libexec
```

- Téléchargement, décompression et Installation du plugin nrpe-2.12.

```
#wget    http://osdn.dl.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
#tar xzf nrpe-2.12.tar.gz
#cd nrpe-2.12
#./configure
#make all
#make install-plugin
#make install-daemon
#make install-daemon-config
#make install-xinetd
```

➔L'installation est donc achevée, Passons à la configuration de /usr/local/nagios/etc/nagios/nrpe.cfg.

```
Allowed_host =  @ du serveur nagios
```

Et ajouter la ligne suivante dans /etc/services :

```
nrpe            5666/tcp            # NRPE
```

➔Finalement lancer le deamon XINETD relatif à NRPE :

```
# /etc/init.d/xinetd   start
```

➔On peut aussi utiliser les commandes suivante pour stopper, redémarrer ou déterminier l'état du processus (démarré, stoppé) :

```
# /etc/init.d/xinetd   stop
# /etc/init.d/xinetd   status
# /etc/init.d/xinetd   restart
```

➢ Au niveau du serveur Nagios

Au niveau du serveur serveur Nagios on refait les mêmes étapes pour l'installation de NRPE.

- Les plugins sont déjà installés.
- Téléchargement, décompression et Installation du plugin nrpe-2.12.

```
#wget    http://osdn.dl.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
#tar xzf nrpe-2.12.tar.gz
#cd nrpe-2.12
#./configure
#make all
#make install-plugin
#make install-daemon
#make install-daemon-config
#make install-xinetd
```

➔Finalement lancer le deamon XINETD relatif à NRPE :

```
# /etc/init.d/xinetd   start
```

➔Depuis le terminal du serveur nagios testons si la machine Windows distante répond en tapant la commande suivante qui doit renvoyer la version de NSClient++ installée :

```
#cd /usr/local/nagios/libexec
#./check_nrpe –H  @serveur-distant
```

➔Vérifier que les requêtes (TCP sur port 12489) ne sont pas bloquées par un firewall sinon ajouter une règle pour autoriser le Firewall IPtable.

Annexe D

Notification par mail

❖ Définitions

➢ Serveur SMTP

Serveur smtp signifie « Serveur Simple Mail Transfert Protocol » et se traduit par « protocole simple de transfert de courrier » en français. Un serveur smtp est un serveur de courrier. Il gère le transfert du courrier électronique vers les différents serveurs de messagerie électronique. De plus, il permet l'envoi de mail à partir des ordinateurs clients. C'est pourquoi, il est utile de spécifier un serveur pop et un serveur smtp lors de la configuration du logiciel du mail

➢ Postfix

Il sert à l'envoi des notifications vers votre serveur de messagerie.

➢ Mailx :

Offrant un binaire « mail » commande qui permet de tester l'envoie des mails.

❖ Installations

➢ Installer mailx et posix depuis les sources, à travers l'interface yast ou la commande

```
# Zypper install mailx, postfix
```

❖ Configuration :

➢ Modifier le fichier /etc/potfix/mail.cfg comme suit :

myhostname = MoniterServ

mydestination = rfronteau-laptop, localhost.localdomain, localhos

relayhost = `smtp.orange.fr`

- **Myhostname :** Le nom du myhostname doit être équivalent au nom de votre machine (nom qu'on retrouve dans /etc/hosts ou /etc/hostname).
- **Le mydestinitation** doit être identique au myhostname. En aucun cas c'est 2 valeurs doivent être différentes car ça ne fonctionnera pas ou plus.
- **relayhost** sert à renseigner l'IP ou le nom DNS du serveur de messagerie utiliser pour router votre courrier.

➢ On modifie depuis l'interface de nagios dans **Configuration>command>notifications** on modifie le script de **« host-notify-by-email » et « service-notify-by-email »** en y ajoutant la commande de test **« mail »** au lieu de @MAIL@ :

/usr/bin/printf "%b" "*** centreon Notification *****\n\nType:$NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\nDate/Time: $DATE$/$TIME$" | mail -s "Host $HOSTSTATE$ alert for $HOSTNAME$!" $CONTACTEMAIL$**