République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Sciences, Technologies et Sciences de la Matière

Département Informatique

Thème

Conception et réalisation d'un module complémentaire facilitant la configuration de la plateforme de monitoring « NAGIOS »

Pour l'obtention du diplôme de Master

Spécialité : Master Réseaux et Télécom

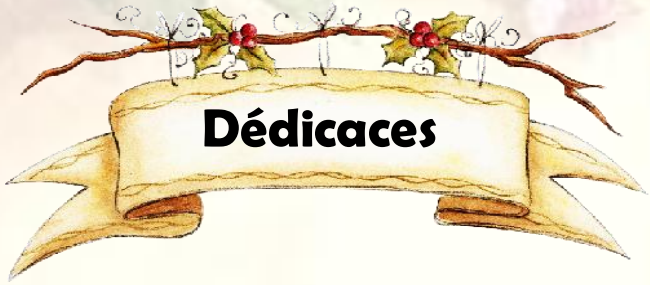
Réalisé par :

- MOHAMMED CHERIF Aissa Taki-Eddine
- SAIDI Wahiba

Dirigé par :

- Mr. MEGHAZI Hadj Madani

Année scolaire 2012-2013



On dédie ce mémoire de fin d'étude

A

Nos très chers parents

En témoignage de nos reconnaissance envers le soutien, les sacrifies et tous les efforts qu'ils ont fait pour notre éducation ainsi que notre formation

A

Nos chers frères**, et **Nos chères sœurs

Pour leur affection, compréhension et patience

A

Tous ceux qui ont une relation de proche ou de loin

Avec la réalisation du présent rapport.

Remerciement

*Nous tenons à remercier **DIEU** le tout puissant qui nous a donné la force et la volonté de réaliser chaque étape de notre vie avec conscience et satisfaction.*

*Au terme de ce travail, nous remercions les plus vifs et chaleureux, vont aux nos **parents**.*

*En premier lieu, nous voudrons exprimer vivement notre gratitude et nos sincères remerciements à **Mr MEGHAZI Hadj Madani** pour son encadrement et de nous avoir proposé le sujet de ce mémoire, et en nous faisant profiter de ces conseils judicieux, et de leur aides et guide avec une grande efficacité. Nous le félicitons de la patience dont il a su faire preuve à notre égard et qu'il soit assuré de notre plus profonde reconnaissance.*

*Nous adressons nos remerciements au membre du jury qui ont accepté d'examiner et d'évalués ce mémoire en lui apportant de l'intérêt : **Mr DAHMANI Youcef**, **NASSANE Samir***

Sans oublier de remercier tous les enseignants du département d'informatique pour le respect et leur politesse, leur gentillesse qu'ils nous ont montrée durant la période de notre formation, et tous nos collègues.

A cette occasion, nous tenons à témoigner notre reconnaissance à toute personne ayant aidé, de près ou de loin, à l'aboutissement de ce travail.

Résumé

Aujourd'hui, la supervision (Monitoring) des serveurs d'une entreprise et le contrôle permanent de leurs bon fonctionnement est une tâche plus que élémentaire afin d'assurer la disponibilité des services fournis.

Actuellement, il existe plusieurs plateformes qui offrent des outils facilitant les tâches de monitoring. Nagios est une de ces solutions, Nagios est libre, très puissant et fonctionnant sous Linux.

Le problème qu'on essaie de résoudre, dans le cadre de cette proposition de mémoire, réside dans la difficulté trouvée pour la configuration (Ajout, modification et mise à jour) des informations des serveurs concernés.

Si on veut résumer, l'objectif est de concevoir un module facilitant l'introduction de ces configurations dans Nagios en proposant une interface (GUI, Web) pour ceci.

Mots clés : *Monitoring, Nagios, Linux, Configuration avancée.*

Abstract

Today, supervision (monitoring) of a company's servers and Standing control their functioning is more than a simple task to ensure the availability of services.

Currently, there are several platforms that provide tools to facilitate the monitoring tasks. Nagios is one of those solutions, Nagios is free, very powerful and running Linux.

The problem we try to solve in the context of the proposed memory is the difficulty found for configuration (Adding, modifying and updating) information servers involved.

If we want to summarize, the goal is to develop a module to facilitate the introduction of these configurations in Nagios offering a GUI (Web) for this.

Keywords: Monitoring, Nagios, Linux, Advanced Setup.

Sommaire

Liste des figures

Introduction générale

Introduction générale

Au cours de la dernière décennie, La surveillance du réseau est un sujet qui a fait l'objet des recherches approfondies avec l'essor des technologies de l'information et des sciences informatiques. Avec tous les nouveaux sites Web hébergés sur des serveurs Web, des expansions d'entreprises mondiales, des réseaux ont évolué de façon significative.

L'administrateur réseau doit être en mesure de contrôler l'ensemble des dispositifs de son réseau et leur activités afin d'assurer la disponibilité des services fournis et éviter tout dérangement quant à l'accès de ces derniers. Chose qui est devenu incontournable dans un monde de compétition accrue où la réputation est un capital.

Cela a conduit à la montée de la nécessité de monitoring pour les administrateurs réseaux et systèmes pour être en mesure de surveiller ce qui se passe sur le réseau, ainsi que l'activité des serveurs de réseau, les services et les routeurs.

Pour garder un œil sur tout ce qui se passe sur son réseau d'entreprise ainsi que l'activité de ses serveurs, le monitoring est une solution d'actualité ! Considérée comme idéale, elle assure une tranquillité d'esprit et l'augmentation des profits tout en évitant des pertes causées par des défaillances système non détectées par les méthodes traditionnelles.

Dans ce mémoire, nous essayerons d'introduire et de toucher le concept de monitoring, en utilisant comme solution un logiciel de supervision libre portant le nom Nagios avec ces multiples configurations et proposant une interface (GUI) au but de suivre et surveiller l'état d'un ensemble de serveurs que nous allons les définir.

Chapitre I :

Administration Réseaux

Chapitre I : Administration réseaux

Introduction

Aujourd'hui l'administration des réseaux informatiques s'affirme comme une activité clé de toute entreprise, sa continuité dépend de la qualité de son système d'information qui gère son réseau. L'administrateur de réseau doit être en mesure à prendre en charge gérer les fonctions telles que la gestion des droits d'accès durant et après les heures de travail, le trafic et la sauvegarde des données. Il doit aussi contrôler la politique de sécurité régissant tous les types d'accès au réseau (accès interne, accès à distance et interconnexion avec des tierces parties), la surveillance et l'assurance de la fiabilité en générale du réseau. [1]

Pour profiter le maximum d'internet, l'administration des réseaux doit prévoir le développement de la mise en place d'outils indispensables pour la sécurité d'utilisation de cette dernière. Les entreprises utilisent différents produits et services tels que (les pare-feux, antivirus, gestion des courriers électroniques) qui jouent un rôle majeur et très important dans leur fonctionnement.

1. L'administration d'un réseau

C'est un ensemble des activités nécessaires qui englobe les moyens mis en œuvre pour offrir aux utilisateurs une qualité de service qui va permettre de guider l'évolution du système en fonction du trafic, les nouveaux services et technologies. Elle représente aussi la partie opérationnelle d'un système informatique (la surveillance de réseau informatique) et son réseau d'interconnexion, tel que le support technique, prévisions des coûts, et l'aide pour la gestion des ressources humaines.

1.1. Administrateur réseaux

Est une personne chargée de la gestion du réseau, c'est-à-dire de gérer les comptes et les machines d'un réseau informatique d'une organisation (**entreprise** par exemple). Cela peut concerner notamment des concentrateurs, commutateurs, routeurs, modems, pare-feux, proxies, connectivité Internet, les réseaux privés virtuels (***VPN***).

1.2. Les domaines d'activités de l'administration réseaux

L'ISO[1] distingue cinq domaines de l'administration réseaux :

- Gestion des pannes.
- Gestion des configurations.
- Audit des performances.
- Gestion de la comptabilité.
- Gestion de la sécurité. [2]

La gestion des pannes

- La détection de certains problèmes qui engendrent des pannes, isolation de côté afin de réparation.

Gestion des configurations

- Configuration des routeurs, commutateurs (switches), points d'accès Wi-Fi, etc.

Gestion des performances

- ***Évaluation***: collecter les données et établir des statistiques sur les données et sur les performances (temps de réponse, taux d'utilisation, débit, taux d'erreur, disponibilité).
- ***Gestion de trafic*** : satisfaire les besoins des users (à qui attribuer un grand dédit…).

Gestion de la comptabilité

- Gérer la charge des ressources pour empêcher toute surcharge (congestion).
- Gérer le coût d'utilisation des ressources et les facturer.
- Gérer les quotas d'exploitation des ressources (imprimantes, disques…). [3]

Gestion de la sécurité

- But : protéger les ressources du réseau et du système d'administration.
- Comment : Assurer les services de la sécurité (authentification, confidentialité, intégrité, disponibilité et non répudiation).
- Moyen : cryptographie + ***logiciel de supervision*** + audit + firewall + surveillance des journaux d'évènements :
    - Journal de sécurité.
    - Journal système.
    - Journal application.

[1] **ISO** : Organisation internationale de normalisation qui regroupe les instituts nationaux de normalisation de plus de cent pays.

2. Les fondements

- **Les tâches basiques**

Le but d'un réseau informatique est d'assurer le transport des données de manière automatique. Tout l'art de l'administrateur est de faire en sorte que le réseau puisse fonctionner de manière autonome, de façon à minimiser les interventions manuelles.

L'utilisation de DHCP permet de simplifier la configuration des ordinateurs et offre plus de souplesse pour modifier le plan d'adressage IP. Il faut absolument prendre en considération tout ce qui permet de simplifier le travail de l'administrateur réseau.

On aura beau avoir tous les systèmes redondants du monde, il est néanmoins nécessaire d'effectuer une surveillance rapprochée de son réseau afin d'être au courant des incidents et de pouvoir réagir en conséquence. Parmi les logiciels de surveillance, on peut citer *Nagios* qui outre sa gratuité, a l'avantage d'être particulièrement polyvalent. [4]

- **Un peu d'administration système**

L'administration réseau est rarement totalement découplée de l'administration système, pour la simple raison que le bon fonctionnement d'un réseau repose généralement sur un certain nombre de serveurs. Il n'est donc pas inutile de rappeler un certain nombre de règles élémentaires d'administration système.

Un serveur est rarement administré par une seule personne. Il est donc nécessaire de gérer l'accès concurrent aux fichiers modifiables par les administrateurs.

- **La sécurité est essentielle**

Lorsqu'on parle de sécurité, cela recouvre un domaine très vaste, comprenant entre autres le *contrôle d'accès*, *l'intégrité*, *la confidentialité* et *la disponibilité*. Parmi ces aspects, l'intégrité, elle est assurée en partie par les sommes de contrôle de TCP et d'UDP (rien n'interdisant un intrus placé sur le chemin d'un paquet de modifier les paquets et de recalculer les sommes de contrôle en conséquence), la confidentialité est assurée par des dispositifs de chiffrement spécifiques.

Le contrôle d'accès, quant à lui doit être imposé sur tous les équipements actifs et les serveurs, afin que seules les personnes autorisées y aient accès. Il est en effet particulièrement désagréable de voir son travail ruiné par un intrus qui aurait modifié, voire effacé la configuration de ses équipements. [4]

- **Les fichiers de configuration**

D'ailleurs, quasiment tous les équipements actifs acceptent de télécharger leur configuration, en totalité ou par morceaux, depuis un serveur (généralement par TFTP). Les mauvais administrateurs réseau s'enorgueillissent de pouvoir générer à la main de nombreux fichiers de configuration plus complexes les uns que les autres, au risque d'y introduire des erreurs de syntaxe ou plus grave, d'avoir à gérer la redondance des informations entre plusieurs fichiers.

En revanche, l'administrateur réseau futé et qui de plus applique le principe des tâches basiques adopte plutôt un autre principe lorsque cela est possible (ce n'est pas toujours le cas).

Il met au point un certain nombre de programmes qui permettent à partir d'informations élémentaires et non redondantes, de générer les différents fichiers de configuration qui en découlent.

- **Perl (Practical Extraction and Report Language)**

Est un langage de programmation particulièrement bien adapté aux manipulations de fichiers et de chaînes de caractères. Il dispose d'une immense bibliothèque de modules, permettant d'aller à l'essentiel et d'obtenir rapidement le résultat souhaité.

3. Les réseaux virtuels (VLAN)

Après l'arrivée des premiers commutateurs, des nouvelles possibilités sont apparues. Compte tenu de l'électronique interne des commutateurs, plus complexe que celles des répéteurs, il devenait possible de disposer de plusieurs segments Ethernet au sein d'un même commutateur (ce qui était d'ailleurs déjà possible à moindre échelle dans les répéteurs segmentables). Mais en ajoutant quelques en-têtes supplémentaires aux trames Ethernet, il devenait possible d'étendre la taille de ces segments Ethernet à l'ensemble d'un réseau de commutateurs interconnectés. Les réseaux virtuels (*virtual LAN, VLAN*) étaient nés. [4]

- **Principe**

Un VLAN est l'équivalent moderne des segments Ethernet de l'ancien temps. Tous les ordinateurs faisant partie d'un même VLAN sont capables de communiquer entre eux directement sans avoir à passer par un routeur. Un VLAN peut être local à un commutateur ou s'étendre à un ensemble de commutateurs reliés entre eux.

Dans le cas où une trame Ethernet doit être transportée d'un commutateur à un autre, il est nécessaire d'y rajouter quelques informations (en particulier le VLAN auquel elle appartient). C'est le but du standard IEEE 802.1Q.

- **Le standard IEEE 802.1Q**

Approuvé le 8 décembre 1998, le standard 802.1Q du comité 802 de l'IEEE (*Institute of Electrical and Electronics Engineers*) est aujourd'hui le standard de fait pour l'identification des trames faisant partie d'un réseau virtuel.

Il a succédé à ISL (*Inter-Switch Link*), protocole propriétaire développé par *Cisco* (et également repris par quelques autres constructeurs). Le principe général est de rajouter dans chaque trame Ethernet destinée à être transmise d'un commutateur à un autre quelques en-têtes supplémentaires contenant en particulier l'identifiant du réseau virtuel auquel elle appartient (VID, VLAN Identifier), qui est un numéro sur 12 bits, de 0 à 4094 (4095 est réservé et, en pratique, 0 et 1 sont inutilisés).[4]

- **En pratique**

Les constructeurs de commutateurs imposent en général des limitations quant au nombre de réseaux virtuels que leurs matériels peuvent gérer. Les plus petits modèles ne savent souvent en gérer que quelques dizaines alors que les châssis offrent un éventail plus large. Il convient donc d'y prendre garde lors de la définition des VLAN de son réseau et du choix de son matériel. Il faut également prendre en compte les possibilités offertes par les commutateurs des différents constructeurs.

Les plus simples ne permettent pas de placer un port (et donc les ordinateurs qui y sont connectés) que dans un seul VLAN de manière statique, les plus évolués permettent de choisir le VLAN de manière dynamique en fonction de divers paramètres (adresse Ethernet, protocole, etc.).

4. Simple Network Management Protocol (SNMP)

SNMP permet à l'administrateur réseau depuis un poste de contrôle central, d'obtenir et de modifier divers éléments de configuration d'équipements actifs et de logiciels. SNMP est un paradoxe dans le monde de l'administration réseau conçu comme un protocole censé unifier et

simplifier la gestion des matériels et des logiciels, il n'a pas encore réussi à s'imposer auprès des administrateurs pour des raisons parfaitement valables :

- SNMP n'est pas si simple que ça.
- Sa mise en œuvre (autrement que pour de petits besoins ponctuels) nécessite des investissements gigantesques en logiciels d'administration.
- Sa sécurité laisse à désirer (c'est un comble pour un protocole dédié à l'administration) bien que la situation se soit bien améliorée avec SNMPv3.

En fait, l'utilisation systématique de SNMP est difficile à justifier lorsqu'on n'administre pas un gigantesque réseau composé de matériels et de logiciels hétérogènes et qu'on n'a pas quelques centaines de milliers d'euros à y consacrer.

Cependant, dans certains cas, SNMP peut tout de même s'avérer fort utile si l'on sait l'utiliser à bon escient.

- **Historique de SNMP**

Trois versions successives de SNMP se sont succédé :

- SNMPv1, en 1990, décrit dans les RFC 1155 à 1157.
- SNMPv2, en 1996, décrit dans les RFC 1901 à 1908.
- SNMPv3, en 1999, décrit dans les RFC 2571 à 2575.

Bien que SNMPv3 soit maintenant assez ancien, rares sont les matériels qui sont capables de le gérer. Nous nous concentrerons donc sur SNMPv2.

- **Sécurité**

Le protocole permettant la gestion des équipements actifs du réseau devrait logiquement être le plus sécurisé de tous les protocoles avec en particulier, un contrôle d'accès et une confidentialité irréprochables. En effet, le contrôle d'accès était approximatif et la confidentialité inexistante.SNMPv3, quant à lui accorde enfin une large part de sa spécification au contrôle d'accès, la confidentialité étant naturellement laissée à IPsec.

- **Principes**

- **Les agents**

Chaque équipement ou logiciel pouvant être interrogé par SNMP comporte un serveur appelé agent. Celui-ci écoute sur le port UDP 161.

➢ Les MIBS

Les paramètres pouvant être examinés ou modifiés par l'agent sont définis dans une MIB (*Management Information Base*), qui est un document décrivant ces différents paramètres, leur type (nombre entier, chaîne de caractères...), s'ils sont modifiables ou non (dans l'absolu, indépendamment des droits d'accès), etc.
Chaque élément d'une MIB est défini par un nom et un numéro (comme d'habitude, on retrouve cette dualité, les noms étant plus faciles à manipuler pour nous, pauvres humains, et les nombres plus faciles à manipuler par les ordinateurs). [5]

- **Les opérations**

Sans rentrer dans le détail du protocole et des formats de paquets, il est néanmoins intéressant de savoir un minimum comment fonctionne le dialogue entre un agent SNMP et un logiciel d'administration.

Il existe quatre types de messages : *Get, Get next, Set, Trap*. Les types de messages précédents étaient envoyés par le logiciel d'administration à l'agent, celui-ci fonctionne en sens inverse. Il est très utile pour effectuer une surveillance passive du réseau, les équipements se plaignant si nécessaire.

- **Net-snmp**

La plupart des équipements réseau intègrent un agent SNMP. Certains logiciels également, de même que les systèmes d'exploitation commerciaux. Les UNIX libres, quant à eux, utilisent la suite logicielle ***net-snmp***. Celle-ci regroupe un agent, divers outils d'interrogation ainsi qu'une bibliothèque de fonctions C permettant de construire des agents SNMP ou d'intégrer des capacités d'interrogation dans d'autres logiciels.

- **MRTG**

MRTG est un logiciel permettant d'interroger divers types d'appareils (routeurs, commutateurs, ordinateurs) et de représenter sous forme graphique le trafic réseau sur chacune de leurs interfaces.

5. Les annuaires

On désigne sous le terme générique annuaire tout service permettant d'obtenir des informations à partir d'une base centrale ou répartie. Les annuaires sont rarement indispensables mais ils apportent un confort non négligeable soit aux utilisateurs (c'est le cas du DNS) soit à l'administrateur réseau (c'est le cas de DHCP).

Les annuaires peuvent généralement être gérés soit par plusieurs serveurs simultanément, soit par un serveur principal et par un ou plusieurs serveurs secondaires qui en prennent le relais en cas de défaillance. Étant donné l'importance que prennent les annuaires dès qu'on commence à les utiliser, il est indispensable de profiter de ces capacités de redondance.

5.1. Domain Name System (DNS)

Le DNS est l'annuaire le plus ancien et certainement le plus utilisé. En effet, dans un réseau IP, chaque machine est repérée par une adresse, qui est un nombre sur 32 bits pour IPv4 et sur 128 bits pour IPv6. Le DNS est un moyen d'associer un nom à chaque adresse IP et vice versa (ceci est une vision réductrice mais elle correspond à l'utilisation principale du DNS).il est décrit dans les RFC 1033 à 1035 (datant de novembre 1987). De nombreux autres RFC décrivent des extensions au DNS.il utilise les ports UDP et TCP 53.

- **Principe de fonctionnement**

Le DNS est une base de données répartie. Par rapport au système précédent, on ne dispose plus d'un fichier unique de correspondances entre adresses IP et noms mais de plusieurs et chaque site maintient les informations correspondant à ses machines et les met à la disposition du reste de l'Internet par un protocole approprié.

Pour savoir quel serveur interroger, le système de nommage a été hiérarchisé. À cet effet, l'Internet a été découpé en domaines. Un domaine correspond à un ensemble de machines dépendant administrativement de la même entité.

- **Le système de nommage**

Toute hiérarchie commençant quelque part, la racine du DNS s'appelle « **.** », sous cette racine ont été créés des top-level domains (TLD) permettant d'effectuer un premier rangement

des niveaux inférieurs. Parmi les TLD, on retrouve : des domaines fonctionnels créés à l'origine par et pour les américains, ce qui explique leur vision un peu réduite des choses :

- ✓ ***com*** pour les entreprises ;
- ✓ ***edu*** pour les universités ;
- ✓ ***gov*** pour les institutions gouvernementales ;
- ✓ ***int*** pour les organismes internationaux ;
- ✓ ***mil*** pour les organismes militaires (l'Internet ayant été inventé par les militaires américains) ;
- ✓ ***net*** pour les fournisseurs d'accès à l'Internet ;
- ✓ ***org*** pour les autres organisations.

Des domaines nationaux créés par la suite selon la norme ISO 3166-1 :

- ✓ ***au*** pour l'Australie ;
- ✓ ***de*** pour l'Allemagne ;
- ✓ ***fr*** pour la France, etc…

➢ Les serveurs

Chaque zone dispose d'un ou de plusieurs serveurs DNS. S'il y en a plusieurs, l'un d'eux est dit ***maître*** et les autres sont ses ***esclaves*** (on parlait de primaire et de secondaires dans l'ancienne terminologie du DNS). Le ***serveur maître*** est le vrai détenteur des informations de la zone, les esclaves se contentent de les recopier.

À chaque modification des informations d'une zone, le serveur maître avertit ses esclaves pour qu'ils puissent se mettre à jour. Toute modification sur le maître se répercute donc très rapidement sur ses esclaves. L'opération de mise à jour s'appelle ***un transfert de zone***. Le serveur maître et ses esclaves font autorité sur les zones qu'ils gèrent (c'est-à-dire qu'eux seuls détiennent les tables de correspondance officielles pour ces zones).

➢ La résolution de nom

Si nous avons besoin de savoir l'adresse IP associée à une machine de la zone ***univ-tiaret.dz,*** nous allons s'adresser à l'un des serveurs qui font autorité pour cette zone. Mais quels sont ces serveurs ?

Comme la structure du DNS est hiérarchique, il suffit de le demander à l'un des serveurs de la zone ***dz***. Si nous ne les connaissons pas non plus, nous allons demander leurs

adresses à l'un des serveurs de la racine. Puisqu'il faut bien qu'une arborescence commence quelque part, ces derniers sont connus (il y en a actuellement treize, répartis sur la planète) et permettent donc de toujours pouvoir descendre l'arbre du DNS. L'ensemble de ce processus s'appelle la résolution de nom.

- **Interrogation de DNS avec dig**

Le DNS est généralement interrogé de manière transparente par divers programme mais il est également possible de l'interroger explicitement, pour tester un serveur ou par simple curiosité. Les principaux programmes permettant ceci sont ***dig***, ***nslookup*** et ***host***. Plus spartiate, ***dig*** est néanmoins plus précis et les puristes le préfèrent aux autres, principalement parce qu'il renvoie des résultats dans un format directement exploitable par un serveur de noms.

5.2. BIND (Berkeley Internet Name Domaine)

Est un logiciel utilisé sur quasiment tous les serveurs de noms du monde, développé par l'**Internet Software Consortium**(**ISC**). BIND est livré en standard avec tous les systèmes UNIX mais il s'agit rarement de la dernière version.

Or il est important de toujours utiliser la dernière version de BIND pour éviter l'exploitation des failles de sécurité connues et pour profiter des dernières fonctionnalités (BIND ayant énormément évolué ces dernières années).

5.3. LDAP (Lightweight Directory Access Protocol)

LDAP est l'annuaire qui monte en ce moment. Il est principalement utilisé pour centraliser les bases d'authentification ou les carnets d'adresses électroniques mais il s'agit d'un annuaire généraliste pouvant être utilisé pour presque n'importe quoi. LDAP est décrit dans le RFC 2251 et utilise le port TCP 389.

- **Principes**

LDAP est l'adaptation dans le monde IP de DAP *(Directory Access Protocol*), qui permet d'accéder à des annuaires X.500 dans le monde ISO. C'est ce qui explique la structure particulière des bases LDAP et leur système de nommage. À la différence des systèmes de

gestion de bases de données (SGBD), les bases LDAP sont optimisées pour la lecture. En conséquence, elles peuvent aisément être répliquées pour assurer une meilleure disponibilité.

Une base LDAP est composée d'entrées, chaque entrée regroupe un certain nombre d'attributs et a un DN (*Distinguished Name*) qui doit être unique au sein de l'ensemble de la base.

Chacun des attributs a un type et une ou plusieurs valeurs. Chaque entrée a un attribut spécial indiquant la classe de l'entrée. Celle-ci contrôle quels attributs sont autorisés dans l'entrée. Ainsi il existe une classe « *organisation* », une classe « *personne* », etc. Les entrées sont définis dans un format de fichier spécial appelé LDIF (LDAP Data Interchange Format) et décrit dans le RFC 2849.

- **OpenLDAP**

Le serveur LDAP le plus répandu sous UNIX est OpenLDAP, dont la dernière version est la 2.4.21.

Autres annuaires

Network Information System (*NIS*), auparavant appelé Yellow Pages (*YP*) est un système créé par Sun Microsystems et permettant le partage de divers bases d'informations (notamment `/etc/passwd` et `/etc/group`) au sein d'un ensemble de machines sous UNIX. Il est aujourd'hui en voie de disparition au profit de LDAP.

6. La messagerie

La messagerie est certainement l'un des services réseau les plus utilisés. Il existe de nombreux systèmes de messagerie propriétaires mais aujourd'hui la majorité des messageries utilise le protocole SMTP (Simple Mail Transfer Protocol). Quelques messageries d'entreprise utilisent également le protocole X.400 mais uniquement en interne, la communication vers l'extérieur se faisant par l'intermédiaire d'une passerelle SMTP.

SMTP est décrit dans le RFC 5321. Le format des courriers électroniques est décrit dans le RFC 5322. SMTP utilise le port TCP 25.

6.1. SMTP (Simple Message Transfert Protocole)

Un protocole utilisé pour transférer les messages électroniques sur les réseaux. Un serveur SMTP est un service qui écoute sur le port 25, son principal objectif est de router les mails à partir de l'adresse du destinataire. [6]

- **Fonctionnement**

Le service SMTP est divisé en plusieurs parties, chacune assurant une fonction spécifique:

- **MUA** (*Mail User Agent)* : c'est le client de messagerie (*Exemples* : Outlook, ThunderBird).
- **MTA** (*Mail Transfert Agent)* : c'est l'élément principal d'un serveur SMTP car c'est lui qui s'occupe d'envoyer les mails entre les serveurs. En effet, avant d'arriver dans la boite mail du destinataire, le mail va transiter de MTA en MTA. Il est possible de connaitre l'ensemble des MTA par lesquels le mail est passé, pour cela il suffit d'afficher la source du message.
- **MDA** (*Mail Delivery Agent)* : c'est le service de remise des mails dans les boîtes aux lettres (les espaces mémoires réservés) des destinataires, il intervient donc en fin de la chaine d'envoi d'un mail. [6]

- **Relation avec DNS**

Comment déterminer quel est le serveur de messagerie du site destinataire ? Le RFC 2219 indique bien qu'il est censé s'appeler ***mail.domaine*** mais c'est loin d'être le cas sur tous les sites. De plus, comment faire s'il y a plusieurs serveurs ?

Le DNS dispose donc d'un RR de type MX, qui indique quel est le serveur de messagerie associé à une zone :

Gmail.com IN MX 10 www.gmail.com

Le serveur de messagerie pour la zone ***gmail.com*** est donc la machine www.gmail.com . Le nom de cette machine est precede d'un entier naturel indiquant la priorité relative du serveur. Cet entier est appelé *le poids du serveur*. En effet, une zone peut disposer de plusieurs serveurs de messagerie et il faut pouvoir connaître la priorité de chaque serveur. *Plus le poids est petit, plus*

le serveur est prioritaire. Ainsi, si le serveur de poids le plus faible ne répond pas, le courrier sera envoyé au deuxième, sinon au troisième et ainsi de suite. La valeur des poids n'a pas d'importance, seule en a leur position relative dans l'ordre croissant.

Hotmail.fr IN MX 0 mail.hotmail.fr
IN MX 10 mail0.hotmail.fr

Dans l'exemple précédent, le courrier sera tout d'abord envoyé au serveur **mail.hotmail.fr**. Si celui-ci ne répond pas, il sera alors envoyé au serveur **mail0. hotmail.fr**.

6.2. Serveurs de messagerie

Il existe de nombreux serveurs de messagerie. Sous UNIX, les plus répandus sont : **Sendmail, Postfix**, et dans une moindre mesure : **Gmail, Exim**.

Sendmail, le logiciel de messagerie historique (sa première version remonte au début des années 1980), est encore utilisé sur de très nombreux serveurs de messagerie. Cependant, un système de configuration complexe (même s'il est maintenant masqué par des outils plus simples), une structure monolithique et une longue histoire de problèmes de sécurité (dont le dernier remonte pourtant à 1997) lui font aujourd'hui préférer d'autres logiciels, notamment ***Postfix***.

Conclusion

Nous avons entamé notre mémoire avec ce chapitre qui a pour objet « l'introduction des tâches d'administration réseaux et systèmes dans les réseaux d'entreprise ». Dans lequel nous avons mis l'accent sur l'importance de cette dernière qui est considérée maintenant comme une mesure de l'avancé, la fiabilité et le sérieux des SI des grandes firmes.

Parmi les tâches d'administration réseaux on trouve « la supervision », qui peut être réalisée en utilisant de diverses techniques. Alors, c'est quoi la supervision ? Quels sont ses outils ? Ce sont unes des questions qui vont être vu en détail dans le prochain chapitre.

Chapitre II :

La supervision

Chapitre II : La supervision

Introduction

Un bon administrateur réseau doit savoir à tout moment l'état des différentes machines et des différents services. Cependant, l'administrateur ne peut pas se permettre de passer son temps devant un tableau avec des voyants verts en attendant qu'un voyant passe au rouge pour agir, son temps est occupé à d'autres tâches, donc il ne peut pas surveiller les statuts des machines en permanence.

L'examen quotidien des logs systèmes est un bon début, mais, si un problème survient, on s'en rend compte seulement le lendemain, ce qui peut être trop tard.

Pour simplifier leur travail, les administrateurs utilisent généralement ce qu'on appelle un « *moniteur de supervision informatique* », un tel moniteur permet d'avoir une vue globale du fonctionnement de réseau ainsi que du niveau de performances des systèmes, et d'alerter par différents moyens, l'apparition d'une anomalie.

Dans ce chapitre, nous allons présenter les notions de base concernant la supervision informatique.

1. Définition

La supervision désigne un ensemble de concepts recouvrant la surveillance du bon fonctionnement d'un système informatique (matériels, services, applicatifs) en production. On peut la définir aussi comme un domaine vaste de l'informatique inclut donc plusieurs activités : Surveiller, Visualiser, Analyser, Piloter, Agir …

On distingue trois types :

1.1. Supervision système

La supervision système porte principalement sur les trois types principaux de ressources système: processeur, mémoire et stockage.

1.2. Supervision réseau

Cette supervision porte sur la surveillance de manière continue de la disponibilité des services en ligne du fonctionnement, des débits, de la sécurité mais également du contrôle des flux.

1.3. Supervision des applications

La supervision des applications (*ou supervision applicative*) permet de connaître la disponibilité des machines en terme de services rendus en testant les applications hébergée par les serveurs. À titre d'exemple, un serveur Web peut avoir une supervision système et réseau avec des signaux au vert, et la machine ne sera pourtant pas disponible au sens du service Web si apache n'est pas présent ou n'est pas en mesure de servir des pages Web.

2. La supervision dans les entreprises

La supervision informatique permet de superviser l'ensemble du système d'Information de l'entreprise : Le réseau et ses équipements, les serveurs, les périphériques et les applications.

L'informatique est au cœur de l'entreprise, quel que soit son secteur d'activité. On peut facilement comparer la place que joue l'informatique au sein d'une entreprise à celle que joue le système nerveux chez l'être humain. En effet, il est au centre de l'activité, et doit fonctionner pleinement et en permanence pour garantir l'activité. Certaines ramifications même comme le réseau et les terminaux utilisateurs doivent aussi fonctionner, à l'instar des nerfs du système dans le corps humain.

Les problèmes liés à l'informatique doivent donc être réduits au minimum, car une indisponibilité du système d'information peut être la cause d'une grande perte.

Deux phases sont donc importantes pour les directeurs informatiques *: garantir la disponibilité du système en cas de panne* (par des mécanismes de redondance...) mais aussi tenter de prévenir en cas de problème, et le cas échéant, *garantir une remontée d'information rapide et une durée d'intervention minimale :* c'est le rôle de la supervision.

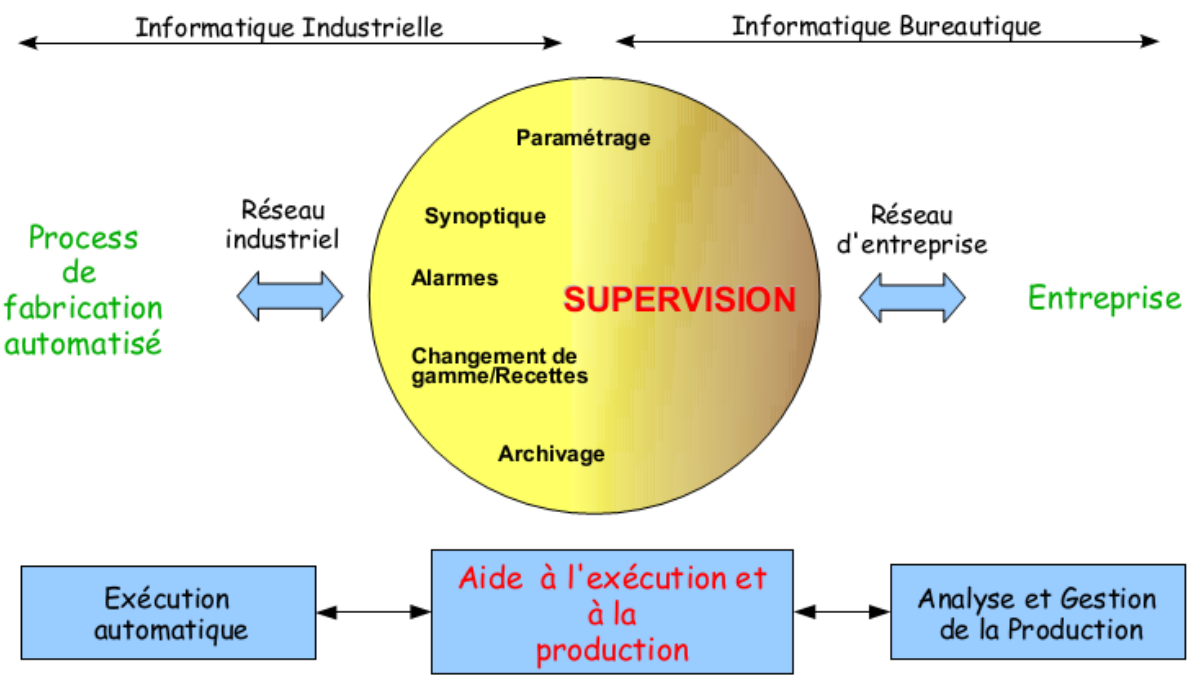


Figure 1 : La supervision dans les entreprises [7].

La figure ci-dessus représente la supervision dans les entreprises, dans la première partie, qui représente l'informatique industrielle, il y a un processus de fabrication automatisé travaille en équivalence avec le poste de supervision par l'intermédiaire d'un réseau industriel et dans la deuxième partie qui représente l'informatique bureautique, il y a l'entreprise avec son tour de travailler en équivalence de l'autre côté avec le poste de supervision par un réseau d'entreprise.

Et tout cela conduit à l'exécution automatique et l'aide à l'exécution et à la production ce qui montre finalement l'analyse et gestion de la production.

3. Les modules de supervision

Autour de la supervision, plusieurs modules coexistent :

- **Supervision des applications**

Elle porte sur la surveillance de manière continue de la disponibilité des services en ligne, du fonctionnement du réseau, des débits et bande passante, et de la sécurité, etc.

- **Supervision système**

C'est la vérification de la santé d'un serveur coté ressources matérielles (la mémoire, le CPU, le disque dur, etc.).

- **Exécution de commandes**

Qui sont des actions ou programmes lancés automatiquement.

- **Envoi d'alertes**

C'est une émission de message d'alerte sous forme sonore, visuelle ou encore par e-mail.

- **Cartographie**

Présente la vue d'ensemble de l'architecture informatique surveillé.

- **Rapports d'activité (reporting)**

Comme les tableaux de bord et les histogrammes.

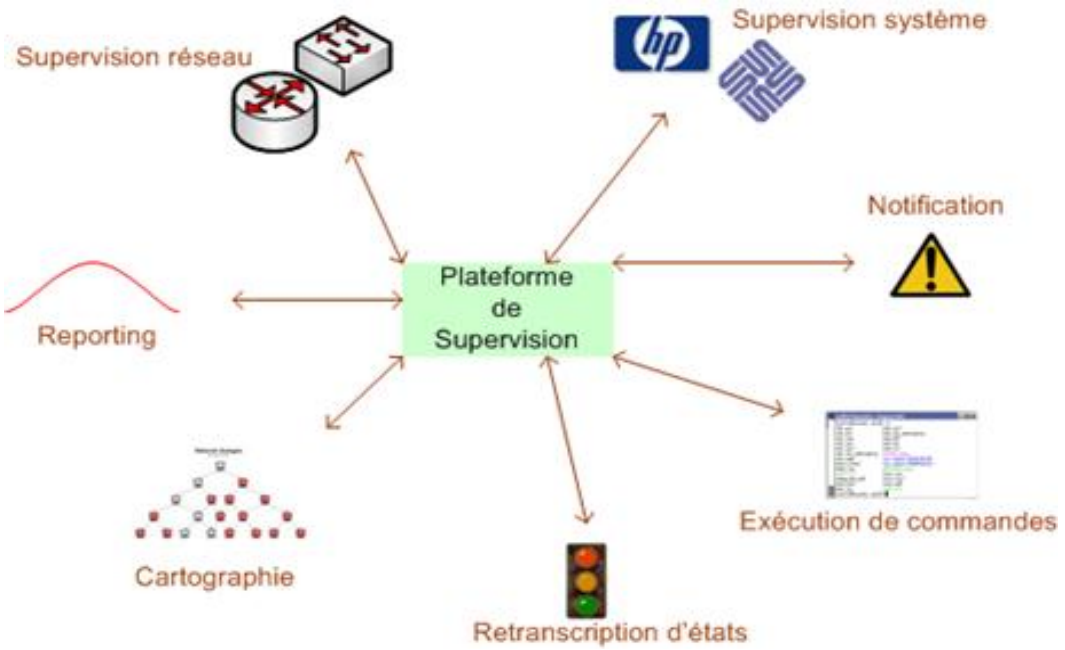


Figure 2 : Les modules de supervision.

Comment superviser ?

Il existe plusieurs méthodes pour superviser le système d'information :

- Analyser les fichiers de log.
- Récupérer des résultats de commandes et de scripts locaux ou distants.
- SNMP : Simple Network Management Protocol. (Nous avons l'expliqué dans le premier chapitre)

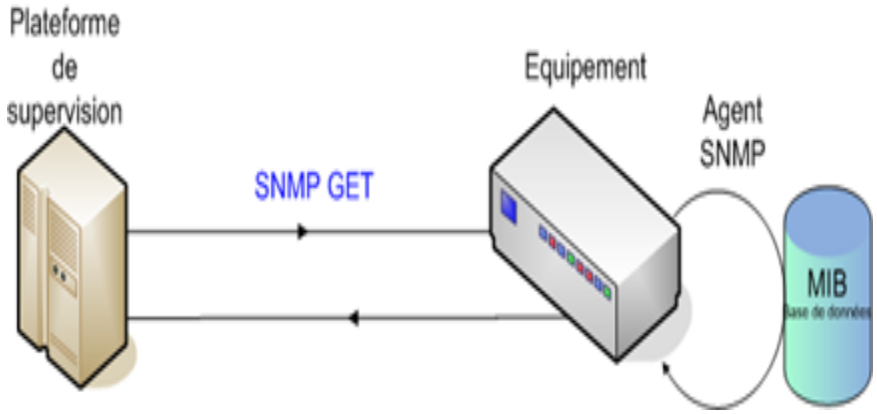


Figure 3 : Gouverne de supervision.

4. Sécurité

Le serveur de supervision ayant accès à énormément d'informations, il doit être considéré comme un point potentiellement dangereux et doit être sécurisé au maximum. [8]

5. Besoins

Les besoins en matière de supervision de systèmes et de réseaux sont assez variés. Certaines entreprises souhaitent simplement une visibilité en temps réel de leur fonctionnement, d'autres veulent également des alarmes ou des actions automatiques en cas de problème, des mesures de performances ou un historique pour l'aide à la décision, voire l'ensemble. [8]

6. Le marché de la supervision

Le marché de la supervision peut être découpé en deux grandes sous-parties :

- ❖ Les offres éditeurs.
- ❖ Les offres du monde libre.

6.1. Les offres éditeurs

Les gros éditeurs logiciels ont rapidement compris que *la supervision était une ressource clé pour les entreprises,* qui de plus en plus, utilisent leur système d'information et ont donc besoin d'une disponibilité toujours plus grande de leur infrastructure informatique.

Par conséquent, la supervision est un domaine dans lequel les sociétés n'hésitent pas à investir depuis quelques années. Ayant rapidement compris cela, les gros éditeurs logiciels sont donc très vites entrés dans la course aux logiciels de supervision.

Aujourd'hui, la majorité des gros éditeurs logiciels propose des outils complets de supervision. On retrouve, parmi les plus connus :

- ❖ HP : la gamme Openview (NNM, OVO, ...).
- ❖ BMC : Patrol.
- ❖ IBM : Tivoli.
- ❖ Computer Associates : Unicenter TNG.

6.2. Les offres du monde libre

Depuis une dizaine d'années déjà, plusieurs projets de supervision ont vu le jour au sein de la communauté du logiciel libre. Il suffit pour cela d'aller faire un petit tour sur ***sourceforge*** pour se rendre compte de la multitude de projets émergeants autour de la supervision système et réseau.

Que Superviser ?

Technique : surveillance du réseau, de l'infrastructure et des machines (Processeur, Mémoire, Stockage).

Applicative : surveillance des applications et des processus métiers.

Contrat de service : surveillance respect des indicateurs.

Métier : surveillance des processus métiers de l'entreprise.

Conclusion

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. La supervision est un des moyens indispensables pour favoriser la croissance de rendement d'une entreprise. Parmi tous les logiciels libres de supervision, ***Nagios*** est très certainement le plus répandu et également le plus suivi par la communauté de développeur. Dans le chapitre suivant on étudiera en détails tous qui concernent ce logiciel libre.

Chapitre III :

NAGIOS

Chapitre III : NAGIOS

Introduction

Un des besoins les plus exprimés en matière de gestion de réseau est la surveillance des services. C'est donc dans une démarche de qualité de service, et de manière à pouvoir réagir dans les plus brefs délais, que de nombreuses solutions de supervision de services ont vu le jour, dont Nagios.

1. Histoire de Nagios

Nagios est développé par *Ethan Galstad* et débute son histoire en ***1999*** sous le nom de ***NetSaint***. Quelque temps plus tard, à cause d'un problème de propriété intellectuelle sur le nom, il devient ***Nagios***. Actuellement en version 3, il a plus de neuf ans d'existence.

Nagios à l'origine était destiné uniquement pour les systèmes Linux, mais actuellement, elle peut se déployer sur n'importe quel système Unix.

2. Définition

Nagios est un logiciel libre de surveillance (***Monitoring***) des réseaux et systèmes, très connu dans le monde de l'entreprise et des professionnels réseaux. Il permet de surveiller les hôtes et les services spécifiés dans son fichier de configuration, et d'alerter les administrateurs systèmes et réseaux en cas d'évènements (Mauvais ou Bon).Nagios permet la supervision active et passive.

3. Quelques fonctionnalités de Nagios

- Surveillance des services réseaux tels que: SMTP, HTTP, FTP, SSH, etc.
- Surveillance des ressources machines telles que: Charge de processeur, Utilisation de l'espace disque, utilisation de la mémoire, etc.
- Rotation automatique des fichiers journaux.

● Interface Web optionnelle permettant de visualiser l'état actuel du réseau, les notifications et les fichiers journaux.
● Conception des simples greffons (***plugins***) permettant aux utilisateurs de développer leurs propres vérificateurs de services.
● Notification par mail ou sms lorsqu'un problème survient sur un service ou une machine.
● Support pour l'implémentation d'un système de surveillance redondant.
● Etc...

4. Mode de licence

Nagios est distribué sous les termes de la GNU « *General Public Licence* » Version 2 comme publiée par la *Free Software Foundation* (FSF). Cette licence donne la permission légale de copier, distribuer et/ou modifier Nagios sous certaines conditions. [9]

5. Concepts et principe

Nagios ne possède aucun mécanisme interne pour surveiller le statut des équipements et des applications. Il repose sur des programmes externes appelés greffons (plugins). Nagios peut être assimilé à un planificateur de tâches. Il exécute un greffon à intervalle régulier lorsqu'un service ou un hôte doit être surveillé. [10]

5.1. Nagios ne fait rien sans ses plugins

Nagios ne sait rien faire tout seul. Il ne peut même pas vérifier le bon état du serveur sur lequel il est hébergé. Son auteur a en effet considéré qu'il ne pouvait prévoir toutes les vérifications qu'un tel outil doit intégrer. Il a donc décidé de n'en mettre aucune au sein de *Nagios* et de laisser la responsabilité des vérifications à des ***plugins*** que l'utilisateur devra fournir à l'outil. [10]

6. Atouts de Nagios par rapport aux autres outils open source

Nagios n'est pas le seul outil de supervision open source. Par rapport à ses concurrents, sa plus grande force réside dans sa modularité complète. Il n'a pas de domaine de prédilection et peut observer tout ce que ses plugins sont capables de rapporter.

D'autres outils open source de supervision existent, mais ils ne sont pas aussi modulaires ou performants que Nagios. On trouve aussi sur le marché des outils de même envergure, mais non complètement libres.

6.1. Performances de Nagios

En matière de performances, Nagios n'a rien à envier aux outils de supervision propriétaires. Il permet avec un serveur modeste, de surveiller près de 10 000 éléments.

Si cette performance est déjà tout à fait honorable, Nagios propose des options pouvant sensiblement augmenter cette valeur. L'architecture même de Nagios permet de surveiller autant de nœuds que souhaite l'utilisateur. La supervision peut être distribuée entre plusieurs serveurs, tout en centralisant l'administration sur une unique console.

7. Architecture de Nagios

Nagios peut être décomposé en trois parties:

- ***Un ordonnanceur :*** chargé de contrôler quand et dans quel ordre les contrôles des services sont effectués.
- ***Une interface graphique (IHM)*** : qui affiche de manière claire et concise l'état des services surveillés.
- ***Des sondes(Greffons) :*** Les sondes de Nagios sont des petits scripts ou programmes qui sont à la base des vérifications.

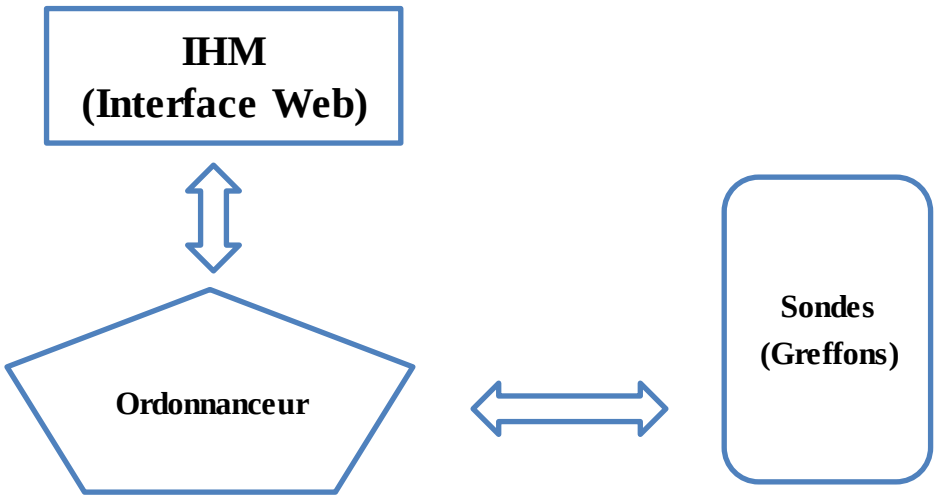


Figure 4 : Architecture de Nagios.

7.1. Les plugins

Un plugin est un programme exécutable ou script (perl, shell, etc.) capable de fournir au moteur :

- **Un code de retour**

=> 0 = tout va bien (OK).
=> 1 = avertissement (WARNING).
=> 2 = alerte (CRITICAL).
=> 3 = inconnu (UNKNOWN).

- **Un court message descriptif**

En option, un plugin peut retourner des informations de performance permettant à Nagios de les interpréter pour tracer des graphiques.

➢ Principe de base

Nagios est un moteur d'ordonnancement de vérifications diverses assurées par des plugins. La relation entre le moteur principal et les plugins se fait d'une part dans la configuration de Nagios, pour que Nagios sache quelles vérifications lancer sur, ou à destination de quelles machines ou services, d'autre part par le code retour ainsi que la sortie standard d'un plugin. Ces plugins fonctionnent soit en local sur la machine Nagios, soit effectuent des tests à distance.

➢ Exécution des plugins

Les plugins peuvent fonctionner localement (directement sur la machine supervisée) ou à distance (au travers du réseau). Pour l'exécution à distance des plugins, il existe plusieurs possibilités:

- Par le biais d'autres serveurs de supervision Nagios distant. Cette méthode est utilisée dans le cadre de la supervision distribuée.
- Par les agents d'exécution de tests, dont voici deux des principaux agents proposés par Nagios:
  - **NRPE (Nagios Remote Plugin Executor)** : il constitue une méthode de surveillance dite active. En effet, l'initiateur et l'ordonnanceur des tests est la machine Nagios : le plugin ***check _nrpe*** permet à la machine Nagios d'envoyer des instructions au démon NRPE situé sur la machine distante.

- **NSCA (Nagios Service Check Acceptor)** : il s'agit là d'une méthode passive : le client NSCA est installé, configuré et lancé sur chaque hôte distant de sorte à envoyer des résultats de tests à la machine Nagios.

➢ Ecriture des plugins

Le projet Nagios fournit en standard un bon nombre de plugins de base, mais la simplicité de leur mode de fonctionnement permet à l'administrateur d'en écrire pour ses propres besoins.

8. Fonctionnement de Nagios

Le principe de supervision de Nagios repose sur l'utilisation de plugins, l'un installé sur la machine qui supporte Nagios, et l'autre sur la machine que l'on souhaite superviser. Un plugin est un programme modifiable qui peut être écrit dans plusieurs langages possibles selon les besoins, et qui servent à récupérer les informations souhaitées.

Nagios par l'intermédiaire de son plugin, contact l'hôte souhaité et l'informe des informations qu'il souhaite recevoir.

Le plugin correspondant installé sur la machine concernée reçoit la requête envoyée par Nagios et ensuite va chercher dans le système de sa machine les informations demandées.

Il renvoi sa réponse au plugin Nagios, qui ensuite le transmet au moteur de Nagios afin d'analyser le résultat obtenu et ainsi mettre à jour l'interface web.

Il existe deux types de récupération d'informations: La récupération ***active*** et la récupération ***passive***.

La différence entre les deux types est l'initiative de la récupération. Dans le premier type, à savoir le type actif, c'est Nagios qui a toujours cette initiative. C'est lui qui décide quand il envoie une requête lorsqu'il veut récupérer une information.

Alors que lors d'une récupération passive, l'envoi d'information est planifié en local, soit à partir d'une date, soit en réaction à un événement qui se déroule sur la machine administrée.

La demande d'informations se fait grâce à l'exécution d'une commande de la part de Nagios. Une commande doit obligatoirement comporter des arguments afin de pouvoir chercher les bonnes informations sur les bonnes machines.

Ces arguments sont l'adresse IP de l'hôte sur lequel aller chercher l'information, la limite de la valeur de l'information recherchée pour laquelle l'état « Attention » sera décidé, idem pour la valeur « Critique », et enfin d'autres options qui varient selon le plugin utilisé.

Pour ne pas devoir à créer une commande par machine supervisée et par information recherchée, nous pouvons remplacer les arguments par des variables, et ainsi réutiliser la commande plusieurs fois, en remplaçant la bonne variable. Nous avons alors la possibilité de travailler avec des services. Lors de la création d'un service, il faut l'associer à un ou plusieurs hôtes puis à une commande.

Ensuite Nagios remplace automatiquement la variable de l'adresse IP dans la commande, grâce à la liste d'hôtes associée au service. Puis on doit définir manuellement dans le service les autres variables nécessaires à la commande.

Une fois que Nagios a reçu les informations dont il avait besoin sur l'état des hôtes, celui-ci peut construire des notifications sur l'état du réseau, afin d'en informer l'administrateur.

Lorsque Nagios effectue une notification, il attribue des états aux hôtes, ainsi qu'aux services.

Un hôte peut avoir les états suivants :

- **UP** : en fonctionnement.
- **DOWN** : éteint.
- **UNREACHABLE**: inaccessible.
- **PENDING** : en attente**.**

Les différents états d'un service sont :

- **OK** : en fonctionnement**.**
- **WARNING** : Attention.
- **CRITICAL :** critique.
- **PENDING** En attente.
- **UKNOWN :** Inconnu.

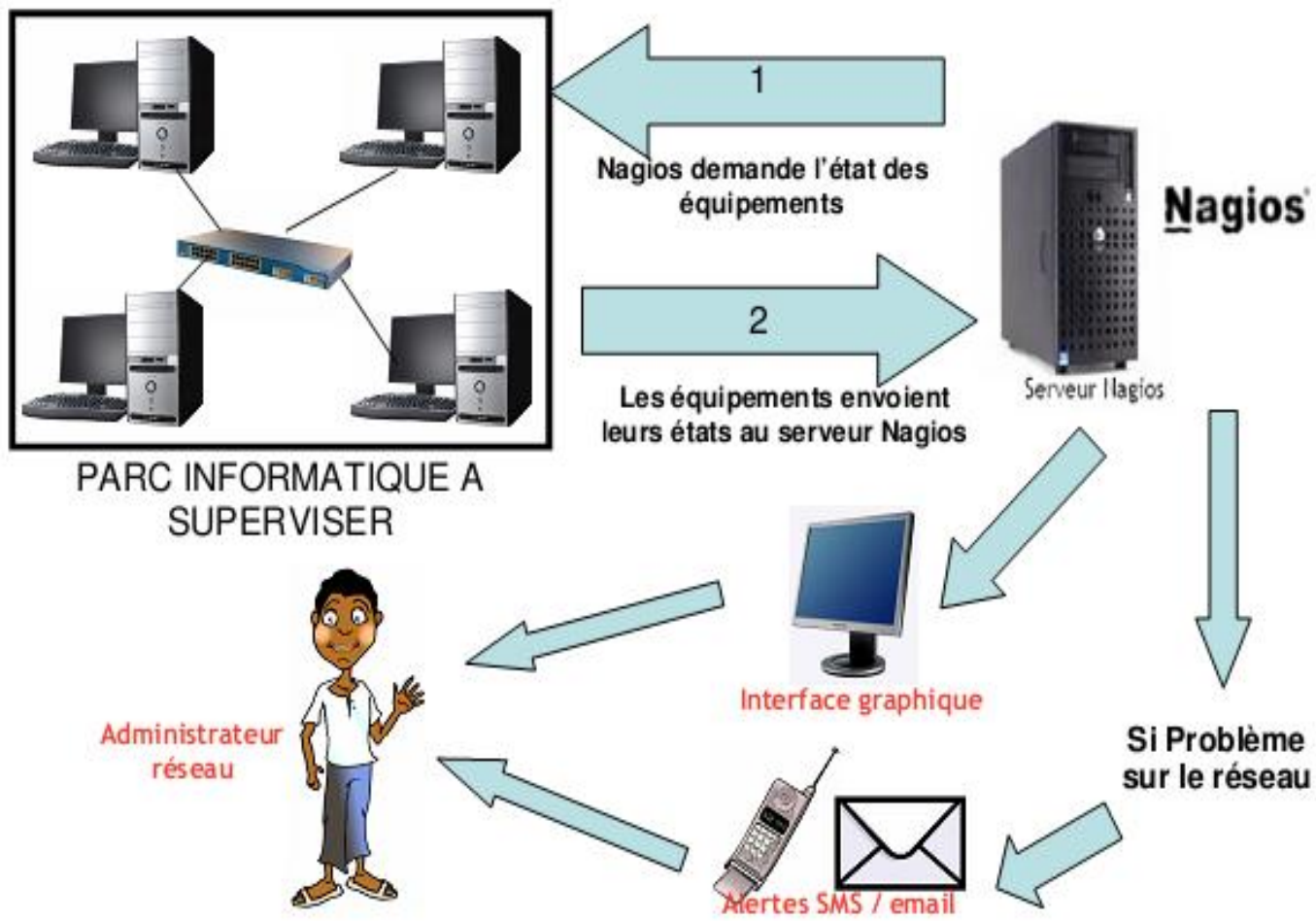


Figure 5 : Fonctionnement de Nagios

9. Mise en place de Nagios

La mise en place de Nagios passe par la récupération des sources sur le site « SourceForge », l'installation et la configuration.

9.1. Récupération des sources

Pour installer Nagios, nous aurons besoin de deux archives : Nagios **(nagios-3.5.0.tar.gz)** et ses plugins de base **(nagios-plugins-1.4.16.tar).** Ces archives sont disponibles en téléchargement sur le site officiel de Nagios http://www.nagios.org . Nous avons fait des tests avec le système Linux Ubuntu (*Machine virtuelle*) dans sa version 11.10

9.2. Installation de Nagios

Nagios, l'outil libre de monitoring réseau. Nous allons donc décrire l'installation de cette nouvelle monture sous un OS Linux Ubuntu en version 11.10.

☞ **Pre-requis**

Nous avons d'abord besoin d'installer un serveur web et les librairies de bases nécessaires pour la compilation de Nagios. Pour cela, il faut utiliser les commandes suivantes:

```
# sudo apt-get install apache2
# sudo apt-get install build-essential
```

Il faut également créer un utilisateur et un groupe dédié au processus Nagios (pour d'évidente raison de sécurité) :

```
# sudo -s
# /usr/sbin/useradd nagios
# passwd nagios
# /usr/sbin/groupadd nagios
# /usr/sbin/usermod -G nagios nagios
# /usr/sbin/groupadd nagcmd
# /usr/sbin/usermod -G nagcmd nagios
# /usr/sbin/usermod -G nagcmd www-data
```

☞ **Compiler les sources de Nagios**

Une fois les sources décompressées :

```
# tar xzf nagios-3.5.0.tar.gz
# cd nagios
```

Nous allons lancer la compilation grâce aux commandes suivantes :

```
# ./configure --with-command-group=nagcmd
# make all
# make install
# make install-init
# make install-config
# make install-commandmode
# ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

Il faut ensuite installer l'interface Web :

```
# make install-config
# make install-commandmode
# make install-webconf
# sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
# service apache2 restart
```

Puis finir par la compilation des plugins de base :

```
# cd ..
# tar xzf nagios-plugins-1.4.16.tar.gz
# cd nagios-plugins-1.4.16
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
# make
# make install
```

9.3. Configuration de Nagios

Avant de pouvoir lancer Nagios, il faut éditer les fichiers de configuration qui se trouve dans le répertoire ***/usr/local/nagios/etc***. Pour faire un premier test, le seul fichier à configurer avant d'exécuter Nagios est le fichier ***/usr/local/nagios/etc/objects/contacts.cfg*** et de changer l'adresse email de contact (nagiosadmin) ou seront envoyés les mails en cas d'alerte.

Pour vérifier que la configuration de Nagios est bonne (qu'il n'y a pas d'erreur dans les fichiers de configuration), nous pouvons utiliser l'option **"-v"** de la commande Nagios:

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

S'il n'y a pas d'erreur, on peut continuer. Afin que nagios et apache2 puissent démarrer automatiquement au démarrage du serveur, faites ceci :

```
# chkconfig --add apache2
# chkconfig --add nagios
# chkconfig nagios on
# chkconfig apache2 on
```

Si nous n'avons pas d'erreurs, nous pouvons passer à la dernière étape.

9.4. Lancement de Nagios

Pour lancer Nagios, nous pouvons utiliser les commandes:

```
# /etc/init.d/nagios start
```
ou bien
```
# service nagios start
```

Pour le redémarrage de Nagios (par exemple si nous modifions les fichiers de configurations):

```
# /etc/init.d/nagios restart
```
ou bien
```
# service nagios restart
```

9.5. Utilisation de Nagios

L'accès à l'interface Web de Nagios se fait par l'URL suivante: http://localhost/nagios/.

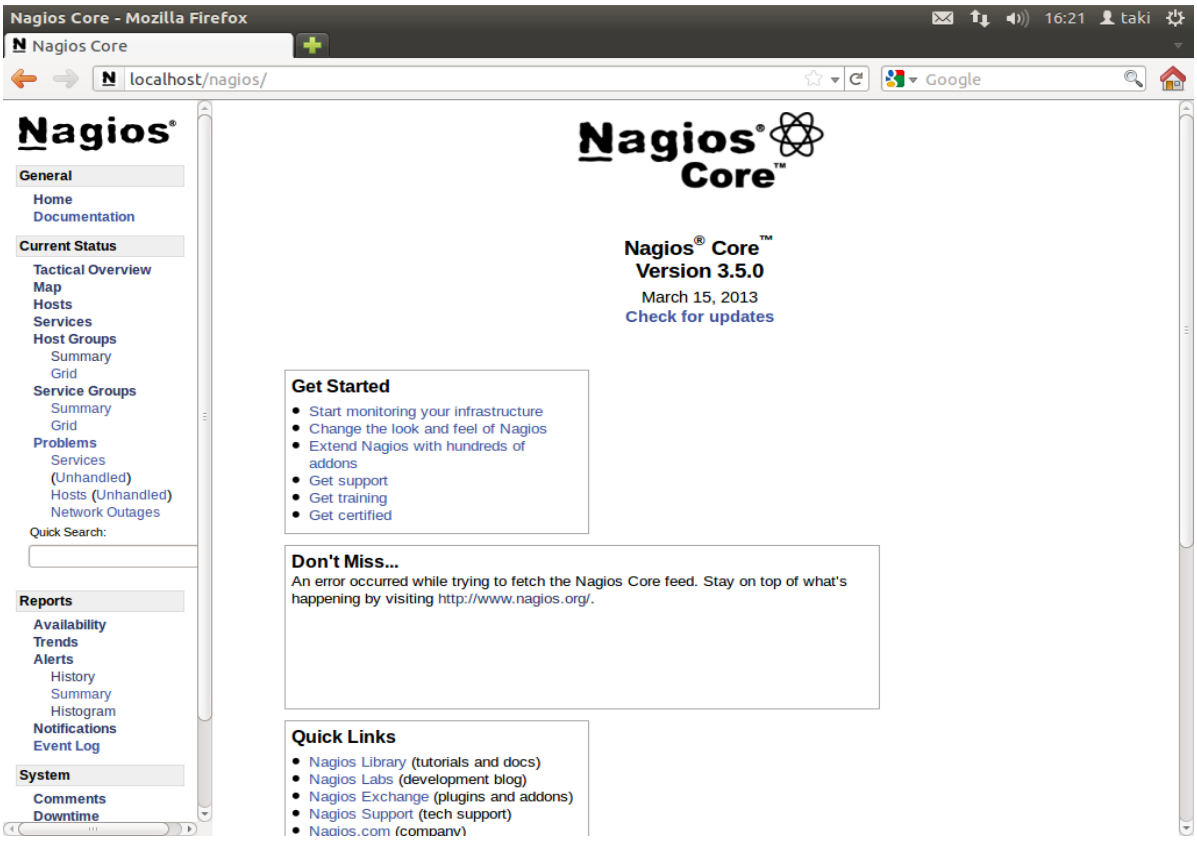


Figure 6 : L'interface Web de Nagios.

10.Données à définir dans Nagios

Les informations dont a besoin Nagios afin d'accomplir sa tâche. Ces éléments devront être définis dans les fichiers de configuration de Nagios. Ces fichiers sont généralement situés dans le répertoire **/etc** de l'arborescence de Nagios, par défaut **/usr/local/nagios**. Ils sont construits sur le modèle suivant :

```
define type {
parametre1=valeur1 ; commentaire
parametre2=valeur2
}
```

Ici nous voyons la déclaration d'un objet « **type** » qui a comme valeur « **valeur1** » pour « **parametre1** ». Certains paramètres sont indispensables, d'autres sont optionnels. Les commentaires commencent par le signe « **;** » .Nous ne verrons ici que les paramètres obligatoires et les plus importants des paramètres optionnels.

- **Commandes de vérification**

Nagios a besoin qu'on lui fournisse les commandes responsables des vérifications des éléments distants. Ce sont des commandes qui déterminent l'état des éléments distants et donnent l'information à Nagios. Elles récupèrent également les données de performances.

Pour les définir, on doit instancier l'objet `command`. Ces instances figurent dans le fichier `commands.cfg`.

Ces objets sont simples et ne comportent que deux propriétés :

- `command_name`: c'est le nom de la commande tel qu'on va pouvoir l'utiliser dans le reste de la configuration Nagios.
- `command_line`: c'est la commande que doit lancer Nagios.

On remarque dans l'exemple ci-dessous une valeur un peu particulière « `$HOSTADDRESS$` ». C'est en fait une macro qui est positionnée lors du lancement de la commande par Nagios.

Elle peut changer de valeur suivant le contexte, ici elle est égale à l'adresse réseau de l'élément que l'on veut surveiller.

Exemple de commande

```
define command {
command_name check_tcp
command_line /usr/local/nagios/libexec/check_tcp -H $HOSTADDRESS$ -p
$ARG1$
}
```

➢ **Arguments des commandes**

Les commandes peuvent prendre des arguments, comme c'est le cas dans notre exemple `check_tcp` ci-dessus. Les arguments sont de la forme **`$ARGn$`**, «`n`» pouvant aller de **1** à **32**. Ils peuvent être donnés lors de l'appel de la commande, par un hôte ou par un service. Ceci permet entre autres, d'avoir une seule définition de la commande pour vérifier un port TCP et de spécifier, par exemple dans le service, le numéro de port que l'on souhaite surveiller.

- **Périodes de temps**

Nagios a besoin de savoir quand superviser les éléments et quand avertir les utilisateurs. Ces périodes de temps peuvent varier suivant les environnements. Il faut que l'utilisateur puisse les définir librement.

➢ **Définition des périodes de temps**

L'objet qui se charge de cela est `timeperiod`. Ses instances figurent dans le fichier `timeperiods.cfg`. Cet objet a comme propriétés :

- `timeperiod_name`: c'est le nom qui sera utilisé dans le reste de la configuration.
- `alias`: c'est un nom d'affichage dans les interfaces.
- `Sunday,monday`, etc. : pour chaque jour, on peut préciser un intervalle de temps qui sera pris en considération.

Les périodes de temps peuvent être exprimées simplement, ici par exemple, les horaires d'ouverture du service :

Exemple : Péroide de travail

```
define timeperiod{
timeperiod_name workhours
alias Work Hours
Sunday       08:00,17:00
monday       08:00,17:00
tuesday      08:00,17:00
wednesday    08:00,17:00
 thursday    08:00,17:00
}
```

- **Hôtes**

Également nomme ***hosts***, nœuds ou ressources, ce sont les éléments que Nagios supervise. Dans le cadre de la supervision système, c'est le serveur à surveiller; pour la supervision réseau, il peut s'agir d'un switch. En cas de problème, il alerte un ou plusieurs contacts. Cet élément est la base de la supervision dans Nagios. [10]

États d'un nœud

Un hôte peut avoir trois états :

- `UP:` il est en état de répondre.
- `DOWN`: il est indisponible.
- `UNREACHABLE`: l'état n'est pas connu car il est situé, en termes de réseau, derrière un élément qui est HS (inaccessible).

➢ **Définition d'un hôte**

Un nœud possède des propriétés particulières comme son *adresse réseau* ou bien les *personnes à contacter* en cas de problème. C'est un objet *host* au sens de Nagios. Ces objets figurent dans le fichier `localhost.cfg`. L'ensemble des propriétés indispensables sont les suivantes :

- `host_name` : nom de l'hôte tel qu'il sera utilisé dans le reste de la configuration.
- `alias` : nom qui est affiché aux utilisateurs.
- `address` : adresse réseau de l'hôte.

- **max_check_attempts** : nombre de vérifications que Nagios doit tenter avant de le déclarer réellement DOWN.
- **check_period** : période de temps pendant laquelle le nœud est supervisé.
- **contacts** : contacts à prévenir en cas de souci.
- **contact_groups** : groupes de contacts à prévenir en cas de souci.
- **notification_interval** : intervalle de temps en minutes, entre les notifications d'erreur ; si cette valeur est à zéro, il ne sera envoyé qu'une seule notification.
- **notification_period** : période de temps appliquée aux notifications des contacts.

Deux autres propriétés sont facultatives mais fortement conseillées :

- **check_command** : commande qui vérifie l'état de l'hôte. Elle est optionnelle car, dans certains cas particuliers comme la supervision passive, elle n'est pas nécessaire.
- **notification_options** : ce sont les états des hôtes qui doivent faire l'objet d'une notification. Si l'option n'est pas spécifiée, Nagios considère que tous les états doivent être remontés. [10]

Les états possibles sont :

- **d** : lorsqu'un hôte passe en état **DOWN**.
- **u** : lorsqu'un hôte passe en état **UNREACHABLE**.
- **r** : lorsqu'un hôte revient en état **UP**.

- **f** : lorsqu'un hôte commence à faire le « yoyo » (**flapping**).
- **s** : lorsqu'un hôte arrive dans une période de maintenance définie par un administrateur.
- **n** : est utilisé si un contact ne veut recevoir aucune notification.

Exemple de définition

Définissons un hôte représentant un serveur nommé **ikat**. Il est vérifié avec la commande **check-host-alive** qui envoie simplement un ping vers l'adresse du serveur, ce test est effectué toutes les 5 minutes. En cas de problème, ce test est réitéré deux autres fois :

La propriété **max_check_attempts** =3=2+ (1 test déjà effectué). Si le problème persiste, une notification est envoyée au groupe **admins**.

Si le problème n'est pas résolu 30 minutes après, une autre notification est envoyée, et ainsi de suite. Lorsque le problème est résolu, le compteur repasse à zéro.

Définition d'un hôte

```
define host{
host_name ikat
alias Serveur Web 1
address 192.168.1.2
check_command check-host-alive
check_interval 5
retry_interval 1
max_check_attempts 3
check_period 24x7
contact_groups admins
notification_interval 30
notification_period 24x7
notification_options d,u,r
}
```

Les hôtes sont les éléments de base de la configuration de Nagios. Ils sont simples à définir et à répertorier, car ils correspondent aux serveurs et aux éléments réseau qui composent un parc.

- **Services**

Les services sont les points supervisés sur les hôtes. Dans le cas d'un serveur, il s'agira par exemple, de s'assurer du bon fonctionnement d'une application particulière, ou bien de vérifier si la charge du serveur est acceptable. Les hôtes et les contacts à alerter sont indispensables dans la définition des services.

➢ **Etats des services**

Un service peut avoir plusieurs états :

- **OK**: tout va bien pour l'élément surveillé.
- **WARNING**: quelque chose commence à aller mal comme par exemple, un disque qui est presque rempli.

- **CRITICAL**: la situation est très grave et demande une intervention immédiate. C'est le cas, par exemple, d'un disque plein.
- **UNKNOWN**: la commande de vérification n'a pas pu obtenir les informations souhaitées. Par exemple, les arguments fournis à la commande ne sont pas bons.

➢ Définition d'un service

Tout comme les autres objets, les services possèdent des propriétés indispensables :

- **service_description**: nom du service.
- **host_name**: nom de l'hôte sur lequel se trouve le point à surveiller.
- **check_command**: commande de vérification pour obtenir l'information souhaitée. On peut lui fournir des arguments en les séparant par le caractère « **!** ».
- **max_check_attempts**: nombre de tentatives au bout desquelles la situation est considérée comme sûre.
- **check_interval**: période entre deux tests en temps normal.
- **retry_interval**: période entre deux tests lorsqu'il y a un souci.
- **check_period**: période de temps durant laquelle le service est supervisé.
- **notification_interval**: intervalle de temps entre deux notifications. Tout comme pour les nœuds, si cette valeur est à zéro, une seule notification est envoyée.
- **notification_period**: période de temps durant laquelle les notifications sont envoyées.
- **contacts**: contacts à prévenir.
- **contact_groups**: groupes de contacts à prévenir.

Une autre propriété est importante mais facultative :

- **notification_options**: ce sont les états qui pour un service, doivent faire l'objet d'une notification. Comme pour les hôtes, si ce paramètre n'est pas positionné, tous les états sont pris en compte. Ces états sont :
  - **w**: lorsqu'un service passe en état **WARNING**.
  - **u**: lorsqu'un service passe en état **UNKNOWN**.
  - **c**: lorsqu'un service passe en état **CRITICAL**.
  - **r, f, s, n**: mêmes options que pour les hôtes, mais appliquées aux services.

Exemple de définition

Définissons un service **http** vérifiant que le port 80 (HTTP) est bien ouvert sur le serveur **ikat**. Le numéro de port est ici passé en argument numéro 1 à la commande **check_tcp** définie précédemment. Si la commande avait en besoin d'un second argument, on l'aura ajouté à la suite, en mettant un autre «**!**» entre les arguments. Par exemple check_**tcp!80!5**, 80 étant **$ARG1$** dans la commande et 5, **$ARG2$**.

Il est à noter que ceci vaut pour tous les appels de **check_command** et donc pour les hôtes également. Les paramètres **contacts** et **contact_groups** n'ont pas besoin d'être présents tous les deux. Si un seul est positionné, la définition est valide.

Cette vérification est faite toutes les 5 minutes. En cas de problème, un test supplémentaire est effectué (**max_check_attempts** = 3=2+1 test déjà effectué) au bout de 3 minutes. Si le problème est toujours présent, une notification est envoyée aux **admins**. Au bout de 30 minutes, si le problème n'est pas toujours résolu, une autre notification est envoyée et ainsi de suite.

Définition d'un service

```
define service{
host_name ikat
service_description Http
check_command check_tcp! 80
max_check_attempts 2
check_interval 5
retry_interval 3
check_period 24x7
notification_interval 30
notification_period 24x7
notification_options w,c,r
contact_groups admins
}
```

- **Contacts**

Les contacts sont les personnes qui reçoivent les notifications d'alertes de Nagios. Les hôtes et les services se voient accrocher des contacts. Nagios doit savoir qui prévenir lorsqu'un problème surgit. Nous allons avoir un contact par utilisateur de Nagios.

Les contacts doivent avoir des périodes de notification. Certains souhaitent recevoir les alertes uniquement sur certaines plages horaires.

Ceci est tout particulièrement vrai lorsqu'on évoque les envois de SMS. Il est inutile que ces messages partent en pleine journée. Certains autres ne veulent recevoir que les alertes critiques et pas les simples avertissements. Tout ceci est possible avec Nagios.

➢ Définition d'un contact

Les objets contenant les contacts sont tout simplement `contact`. Ils sont définis dans le fichier `contacts.cfg` et possèdent les propriétés suivantes :

- `contact_name`: c'est le nom du contact tel qu'il sera utilisé dans le reste de la configuration.
- `host_notifications_enabled`: accepte ou non les notifications concernant les hôtes.
- `service_notifications_enabled`: accepte ou non les notifications concernant les services.
- `host_notification_period`: période de temps où les notifications d'erreurs sur les hôtes sont acceptées.
- `service_notification_period`: période de temps où les notifications d'erreurs sur les services sont acceptées.
- `host_notification_options`: états qui sur les hôtes, doivent faire l'objet d'une notification. Les valeurs possibles sont les mêmes que pour le paramètre `notification_options` des hôtes.

Ces états sont :

- `d`: lorsqu'un hôte passe en état `DOWN`.
- `u`: lorsqu'un hôte passe en état `UNREACHABLE`.
- `r`: lorsqu'un hôte revient en état `UP`.

- `f`: lorsqu'un hôte commence à faire le « yoyo » (flapping).
- `s`: lorsqu'un hôte arrive dans une période de maintenance définies par un administrateur.
- `n`: est utilisé si un contact ne veut recevoir aucune notification.

- `service_notification_options`: états qui sur les services, doivent faire l'objet d'une notification. Les valeurs possibles sont les mêmes que pour le paramètre `notification_options` des services.

Ces états sont :

- `w`: lorsqu'un service passe en état `WARNING`.
- `u`: lorsqu'un service passe en état `UNKNOWN`.
- `c`: lorsqu'un service passe en état `CRITICAL`.
- `r,f,s,n`: mêmes options que pour les hôtes, mais appliquées aux services.

- `host_notification_commands`: commande de notification qui est utilisée pour avertir d'un évènement sur un hôte. [10]
- `service_notification_commands`: commande de notification qui est utilisée pour avertir d'un évènement sur un service.

Un autre paramètre important, quoique facultatif, est :

- `email`: l'adresse e-mail du contact.

Exemple de définition de contact

Voici une définition de contact. Il se nomme Aissa Takieddine. Il souhaite être averti tout le temps et ce par e-mail.

Définition du contact Taki-Eddine :

```
define contact{
contact_name                    Taki-Eddine
alias                           Aissa Takieddine
host_notifications_enabled      1
service_notifications_enabled   1
service_notification_period     24x7
host_notification_period        24x7
service_notification_options    w,u,c,r
host_notification_options       d,u,r
service_notification_commands   notify-by-email
host_notification_commands      host-notify-by-email
email                           centreon1226@gmail.com
```

Nous remarquons que *service_notification_commands* et *host_notification_commands* sont au pluriel, ceci n'est pas quelconque. Un contact peut être averti de plusieurs manières à la fois. On peut l'alerter par e-mail et par SMS en même temps.

La définition dans ce cas, est très simple: il suffit de définir les commandes séparées par des virgules :

Définition de plusieurs commandes de notification :

```
service_notification_commands notify-by-email    , notify-by-email
host_notification_commands host-notify-by-email  , host-notify-by-email
```

Avec une telle définition, à chaque envoi de courriel va correspondre un envoi de SMS. Dans ce cadre, il n'est pas possible d'envoyer les SMS à certaines heures de la nuit seulement, sauf si la commande qui envoie les SMS gère les horaires elle-même.

Le plus simple dans ce genre de cas, consiste à dupliquer la définition du contact : l'une des copies définit l'envoi d'e-mail sur une période **24x7**et l'autre gère l'envoi de SMS sur une période en heures non travaillées.

➢ Groupes de contacts

Il est rare qu'un administrateur soit seul à devoir recevoir une alerte. On peut créer un groupe de contacts dans lequel sont placés plusieurs contacts devant recevoir les

mêmes alertes. Il est possible de définir ce groupe à notifier. L'information sera alors redirigée automatiquement vers les membres du groupe. La définition est très simple. L'objet associé est `contactgroup` et se trouve dans le fichier « **contacts.cfg** » aux côtés des contacts.

Il ne possède que quatre propriétés :

- `contactgroup_name`: nom du groupe tel qu'il sera utilisé dans le reste de la configuration.
- `alias`: nom d'affichage pour ce groupe.
- `members`: liste des contacts du groupe séparés par des virgules.
- `contactgroup_members`: liste des groupes de contacts faisant parti du groupe.

Voici un exemple de définition d'un groupe de contacts regroupant les administrateurs **Taki-Eddine** et **Wahiba** incluant également les membres du groupe admins-linux.

Définition d'un groupe de contacts

```
define contactgroup{
contactgroup_name admins
alias Administrateurs web
members Taki-Eddine, Wahiba
contactgroup_members admins-linux
}
```

Il ne faut pas hésiter à définir des groupes de contacts. Ils facilitent grandement la configuration des services et de nœuds, car il n'est pas nécessaire de recopier à chaque fois tous les membres des groupes, au risque d'en oublier de temps en temps.

11. Les avantages et les inconvénients de Nagios

11.1. Les avantages

- C'est un logiciel libre.
- Information constante et en temps réel sur l'état du parc informatique.
- Richesse des plugins (nagios.org et nagiosexchange.org).
- Possibilité d'analyse grâce à l'édition de rapports.

11.2. Les inconvénients

- Interface complexe et pas très intuitive.
- Configuration fastidieuse, nombre important de fichiers de configuration.
- Configuration de bout en bout.

Conclusion

Lorsque l'on recherche un outil de supervision, le choix d'une solution open source est très avantageux. Dans ce monde, Nagios est la référence. Sa conception est simple et sa plus grande force est sa modularité, c'est pour ça on a choisi ce logiciel dans notre projet afin d'essayer de résoudre le problème qui réside dans la difficulté trouvée pour la configuration (Ajout, modification, et mise à jour) des informations des serveurs concernés.

L'objectif majeur c'est de concevoir un module facilitant l'introduction de ces configurations dans Nagios en proposant une interface GUI pour ceci. Les possibilités de Nagios sont très étendues. La seule limite réside dans l'obtention des informations de supervision. Si l'administrateur est capable de les obtenir en ligne de commande, alors l'intégration dans Nagios est possible.

Chapitre IV :

Fonctionnement De Nagios

Chapitre IV : Fonctionnement de Nagios

Introduction

La gestion d'un parc de serveur est un travail de chaque instant. Un bon administrateur système doit savoir à tout moment l'état des différentes machines et des différents services. Pour se simplifier le travail, nous allons décrire dans le chapitre précédent le moniteur de supervision « Nagios » qui a comme but de surveiller les services et les machines. Concernant les machines supervisées, on va expliquer dans ce chapitre la configuration de Nagios d'un côté, et la configuration des machines quelle que soient Windows ou bien Linux de l'autre côté et la méthode utilisée pour les supervisés avec Nagios.

1. Machines à surveiller sous Windows

1.1. Configuration de Nagios

La première fois que nous configurons Nagios pour surveiller une machine Windows, nous aurons besoin de faire un peu de travail supplémentaire. Nous avons seulement besoin de faire cela pour la première machine Windows.

- Modifions le fichier de configuration principal de Nagios.

```
root@taki-VirtualBox:~# gedit gksudo /usr/local/nagios/etc/nagios.cfg
```

- Décommettons la ligne faisant référence aux templates de configuration des machines Windows (retirons le premier signe dièse (#) de la ligne suivante dans le fichier de configuration principal).

```
# cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Enregistrons le fichier et quittons et comme ça nous avons dit à Nagios de regarder le fichier "**windows.cfg**" pour y trouver les définitions des hôtes Windows. Cette méthode est intéressante car nous pouvons définir toutes nos machines Windows dans le même fichier, mais

nous préférons avoir un fichier par serveur Windows pour une meilleure lisibilité et maintenance. Pour ce faire, on ne l'utilise que pour exemple de fichiers. [11]

- Créons un répertoire où stocker tous les fichiers de configuration de nos machines Windows et précisons-le à Nagios.

```
root@taki-VirtualBox:~# mkdir /usr/local/nagios/etc/serveurs_windows
root@taki-VirtualBox:~# gksudo gedit /usr/local/nagios/etc/nagios.cfg
```

Trouvons les lignes commentées commençant par #cfg_dir=… Créons une ligne identique en précisant le chemin vers le répertoire que nous avons créé. Décommentez-la puis commentons la ligne **cfg_file=/usr/local/nagios/etc/objects/windows.cfg** car nous ne l'utiliserons plus. Voici le résultat :

```
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
...
...
cfg_dir=/usr/local/nagios/etc/serveurs_windows
#cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

- Copions le fichier windows.cfg en ikat[2].cfg dans ce répertoire.

```
root@taki-VirtualBox:~# cp /usr/local/nagios/etc/objects/windows.cfg
/usr/local/nagios/etc/serveurs_windows/ikat.cfg
```

- Modifions ce fichier en remplaçant "**winserver**" par "**ikat**" (nom de notre serveur Windows). Puis changeons l'adresse IP par la nôtre ou par i**kat** étant donné que nous disposons d'un DNS. Faisons une petite vérification

```
root@taki-VirtualBox: /usr/local/nagios/etc/serveurs_windows#
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
root@taki-VirtualBox: /usr/local/nagios/etc/serveurs_windows#
/etc/init.d/nagios restart
```

Il n'y a pas d'erreur, tout fonctionne bien, on peut le vérifier sur notre site Nagios http://localhost/nagios. Dans le fichier ikat.cfg, nous testons plusieurs services présents par défaut.

[2] ikat : Le nom de serveur Windows a supervisé

Notre serveur windows appartient au hostgroup « ***hostgroup_windows***», ce groupe doit être défini une seule fois pour toutes les machines Windows que nous surveillons. Comme nous souhaitons faire un fichier par serveur, nous définirons ce hostgroup dans un fichier à part et l'enlèverons du fichier **ikat.cfg**.

```
root@taki-VirtualBox:~#
gedit /usr/local/nagios/etc/serveurs_windows/hostgroup_windows.cfg
```

```
# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be
a member of this group

define hostgroup{
    hostgroup_name  windows-servers     ; The name of the hostgroup
    alias           Serveurs Windows     ; Long name of the group
    members         ikat                  ; Serveurs membre du groupe
separes        par des virgules
```

Voilà, à ce stade, Nagios surveille notre machine **ikat** et nous pouvons rajouter d'autres serveurs Windows (***machine.cfg*** dans le répertoire ***serveurs_windows***).

1.2. Installation de l'agent

Pour notre projet, nous allons installer l'addon **NSClient++** sur la machine Windows et utiliser le plugin **check_nt** pour communiquer avec NSCLient++. Ce plugin est déjà installé vu que Nagios l'est. Nous pouvons le trouver dans le fichier "***commands.cfg***".

```
# 'check_nt' command definition
define command{
  command_name     check_nt
  command_line     $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -v
$ARG1$  $ARG2$
    }
```

2. Machines à surveiller sous Linux

Nous avons besoin d'un agent sur les serveurs à surveiller et des plugins Nagios. Nous utiliserons pour plugins « ***NRPE*** ».

NRPE (*Nagios Remote Plugin Executor*) est un "Addon" pour Nagios qui permet d'exécuter des plugins sur un serveur Linux/Unix distant. Cela permet de surveiller des ressources locales (charge du processeur, utilisation de la mémoire, espace disque...) qui ne sont normalement pas disponibles depuis d'autres machines. Afin d'interroger le client NRPE il faudra utiliser le greffon "**check_nrpe**" sur notre serveur Nagios .Pour ce faire il va falloir installer **nrpe** sur notre serveur Nagios, et sur toutes les machines à surveiller. De plus il faudra installer les plugins Nagios sur chaque serveur distant à surveiller.

2.1. Installation des plugins NRPE

Avant de commencer l'installation, nous devons d'abord nous assurer de disposer de la librairie "**libssl-dev**", sinon, il faut l'installer. Elle correspond à "**openssl-devel**".

après le téléchargement des plugins ***nrpe-2.14***, plaçons-nous dans le répertoire ***/usr/local/src*** et il faut les copier dans le bureau et les glisser vers le terminal et enlever les « ' ».

```
root@taki-VirtualBox: /usr/local/src# tar xzf 'home/taki/bureau/nrpe-2.14.tar.xzf'
```

➔ Compiler et installer NRPE :

```
root@taki-VirtualBox: /usr/local/src# cd /usr/local/src/nrpe-2.14
root@taki-VirtualBox: /usr/local/src/ nrpe-2.14# ./configure --disable-ssl --
enable-command-args
root@taki-VirtualBox: /usr/local/src# make all && make install
```

➔ Modifions le fichier ***/etc/services*** et rajoutons la ligne

```
root@taki-VirtualBox: /usr/local/src#  gedit /etc/services
```

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

➔ Copions le fichier de configuration **nrpe.cfg** dans le bon répertoire

```
root@taki-VirtualBox:~#  cp /usr/local/src/nrpe-2.14/sample-config/nrpe.cfg
/usr/local/nagios/nrpe.cfg
```

➔ Modifions le fichier de configuration pour activer la prise d'arguments :

```
root@taki-VirtualBox:~# gedit /usr/local/nagios/nrpe.cfg
```

```
dont blame nrpe=1
```

```
root@taki-VirtualBox:~# cat /usr/local/nagios/nrpe.cfg | grep dont_blame_nrpe
```

```
dont_blame_nrpe=1
# command arguments *AND* the dont_blame_nrpe directive in this
```

➔ Création du fichier **/etc/xinetd.d/nrpe** pour définir le service, si le répertoire **/etc/xinetd.d** n'existe pas, c'est que xinetd n'est pas installé. Sous OpenSuse, il devrait être installé par défaut, mais sous Ubuntu il faudra l'installer via la commande : **apt-get install xinetd**. Ensuite, créez le fichier **/etc/xinetd.d/nrpe**.

```
root@taki-VirtualBox:~# gedit /etc/xinetd.d/nrpe
```

```
service nrpe
{
    flags           = REUSE
    socket_type     = stream
    port            = 5666
    wait            = no
    user            = nagios
    group           = nagios
    server          = /usr/local/nagios/bin/nrpe
    server_args     = -n -c /usr/local/nagios/nrpe.cfg -i
    log_on_failure += USERID
    disable         = no
    only_from       = 127.0.0.1 192.168.1.253
}
```

192.168.1.253 est l'IP de notre serveur "**taki**[3]". Quelques paramètres importants :

Quelques explications d'options

server_args

- **-n** : déshabilite ssl.
- **-i** : utilise xinetd.

[3] **taki** : Le nom de notre serveur qui contient l'outil de supervision Nagios

- **only_from** : permet de lister les hôtes autorisés à contacter nrpe.

➔ Redémarrage de xinetd :

```
root@taki-VirtualBox:~# /etc/init.d/xinetd restart
```

2.2. Configuration de Nagios

Il faut définir un objet **command** et les services dans **server**[4]**.cfg**. Commençons par créer un répertoire **serveurs_linux** où déposer notre fichier **server.cfg**.

```
root@taki-VirtualBox:~# mkdir /usr/local/nagios/etc/serveurs_linux/
```

➔ Rajoutons le chemin de ce répertoire dans **nagios.cfg**.

```
root@taki-VirtualBox:~# gedit /usr/local/nagios/etc/nagios.cfg
```

```
cfg_dir=/usr/local/nagios/etc/serveurs_windows
cfg dir=/usr/local/nagios/etc/serveurs linux
```

Maintenant, créons notre fichier "**server.cfg**".

```
root@taki-VirtualBox:~# gedit /usr/local/nagios/etc/serveurs_linux/server.cfg
```

➔ Redémarrage de Nagios.

Le script **/etc/init.d/nagios start/stop/restart** n'est pas toujours d'une fiabilité sans faille. Afin d'éviter d'avoir plusieurs démons nagios qui tourneraient et qui mettraient à jour la même interface CGI. Voici comment éviter ce désagrément :

```
root@taki-VirtualBox:~# /usr/local/nagios/bin/nagios -v
/usr/local/nagios/etc/nagios.cfg
root@taki-VirtualBox:~# /etc/init.d/nagios start
```

Remarque

La question qui se pose : pourquoi n'avons-nous pas créé de fichier hostgroup comme pour les serveurs Windows ? En fait la définition du hostgroup « **hostgroup_linux**» pour les linux se trouve dans le fichier **/usr/local/nagios/etc/objects/localhost.cfg** (fichier de configuration du

[4] **Server** : C'est notre serveur Linux à superviser.

serveur de supervision). Nous pourrions décider d'ôter de ce fichier la définition du hostgroup pour la mettre dans un fichier à part, ce qui serait bien sur beaucoup propre.

Conclusion

Nous avons définir dans ce chapitre la configuration de Nagios qui est installé au niveau de serveur de supervision et l'installation de l'agent NSClient++ sur la machine Windows en utilisant le plugin ***check_nt*** pour communiquer avec lui, et ce plugin devrait déjà être installé sur le serveur Nagios. Concernant les machines Linux, on a vu l'installation de plugin NRPE qui est un agent de supervision qui nous permet de récupérer les informations à distance, donc il suffit d'installer le démon sur la machine distante et de l'interroger à partir du serveur Nagios.

Comme nous avons dit auparavant, notre but majeur de ce mémoire est de réaliser et proposer une interface graphique (GUI) pour faciliter aux débutants dans le domaine Monitoring d'utiliser Nagios pour surveiller leurs machines (Windows & Linux), c'est pour cela on va détailler le fonctionnement de notre application dans le chapitre prochain.

Chapitre V :

Conception et implémentation

Chapitre V : Conception et implémentation

Introduction

Le but de notre travail est de construire un module complémentaire pour faciliter la configuration de Nagios en fournissant une interface graphique permettant d'ajouter et modifier un serveur Windows ou bien Linux en manipulant ainsi les fichiers de configuration à partir de notre application.

- **Pre-requis**

Afin d'ajouter un serveur quel que soit son type sous Nagios pour la supervision, on doit suivre les étapes suivants :

Coté serveur

On doit installer dans un premier temps ce qu'on appelle les plugins et le démon NRPE. Ensuite, en ajoutant le fichier de configuration dans le répertoire approprié tout dépend de type de serveur que l'on souhaite d'ajouter, puis, l'ajout de nom de serveur dans le groupe d'hôtes dans le même répertoire.

Coté client

Concernant les serveurs Windows, on doit installer l'application cliente « NSClient++ » avec l'ajout de l'adresse IP de serveur Nagios et un mot de passe spécifique pendant l'installation de l'addon (NSClient++). Pour les serveurs Linux, on doit installer le démon NRPE. Au cas où l'administrateur a utilisé pour Nagios autre que le port par défaut, il faut donc le spécifier afin que les communications entre l'application cliente et le serveur Nagios se feront sans problèmes

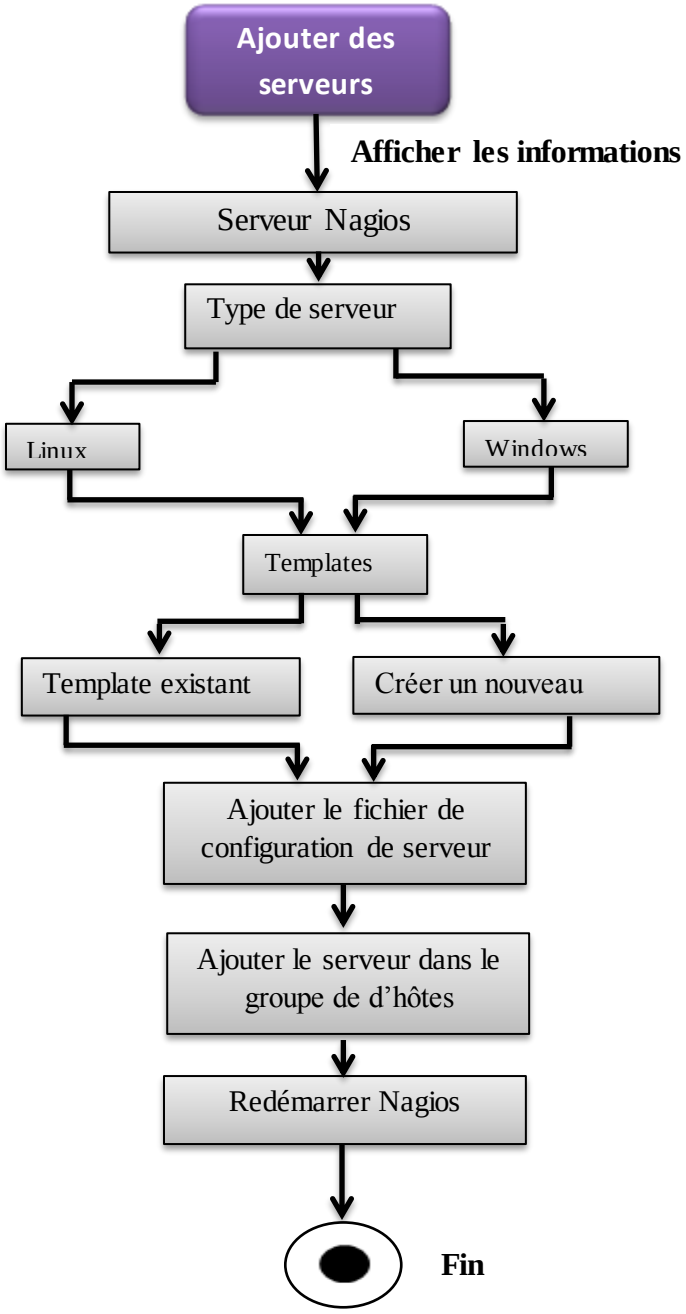


Figure 7 : Diagramme d'ajout des serveurs.

1. Les outils de développement

Notre application est implémentée en utilisant le langage **JAVA**, sous **NetBeans**, dans l'environnement Linux (on a utilisé des machines virtuelles).

1.1. JAVA

Le choix de langage JAVA a été motivé par les raisons suivantes :

- Java assure une totale indépendance des applications vis-à-vis de l'environnement d'exécution : c'est à dire que toute machine supportant Java est en mesure d'exécuter un programme sans aucune adaptation (ni recompilation, ni paramétrage de variables d'environnement).
- JAVA possède une bibliothèque immense d'objets prêts à l'emploi, ce qui facilite pleinement la procédure d'implémentation.

1.2. Netbeans

NetBeans est un IDE (Environnement de Développement Intégré), un environnement libre et facile à utiliser. Il supporte plusieurs langages de programmation à savoir JAVA, C++, PHP, et bien d'autre. Il fournit plusieurs outils tels qu'un éditeur de texte doté d'un pré-compilateur avancé, un gestionnaire de projets. Ainsi que des outils de débogage et de test des programmes. C'est un outil qui facilite énormément la phase de développement et des tests.

2. Les interfaces

La configuration dans Nagios selon notre travail est faite comme suit :

2.1. La page principale

L'exécution de notre application se conduit à la page principale :

Figure 8 : Fenêtre principale.

- Une barre de menu comportant les menus (Fichier, Option)

Figure 9 : La barre de menu.

- Le menu Fichier contient à son tour :

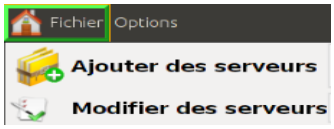


Figure 10 : La barre de menu « Fichier ».

- Le menu option contient Accéder à la base de donnée.

Figure 11 : La barre de menu « Options ».

2.2. Ajouter des serveurs

Dans le menu « Fichier » au-dessus on trouve « **Ajouter des serveurs** », lorsqu'on clique sur cet item, on obtient une fenêtre qui contient deux boutons, l'un pour l'ajout des serveurs Windows et l'autre pour les serveurs Linux :

Figure 12 : Fenêtre « Ajouter des serveurs ».

Pour ajouter un serveur Windows au but de le supervisé avec Nagios, on clique sur le bouton « **Ajouter des serveurs Windows** », ce qui conduit à ouvrir la fenêtre suivante :

Figure 13 : Fenêtre « Ajouter des serveurs Windows ».

Le principe de fonctionnement de cette fenêtre est de remplir les champs vides avec les informations concernant l'hôte « serveur » et cliquer sur le bouton « **Valider** » pour afficher le résultat sur le « JTextArea » afin de vérifier les données fournis s'elles sont justes sinon les modifiées.

Pour mieux comprendre on va prendre un exemple : un administrateur veut ajouter un serveur Windows sous le nom « **CD1** », alors le résultat de validation après le remplissage des champs vides bien sûr est démontré dans la figure suivante :

Figure 14 : Fenêtre « Exemple d'ajout d'un serveur Windows»

P.S : Le chapitre 3 explique tous les détails concernant les informations d'hôte, les délais de supervision et les contacts.

Jusqu'ici, on a seulement fait que l'étape de renseignement sur la configuration du serveur que l'on souhaite d'ajouter, rien n'est encore inscrit dans les fichiers de configurations de Nagios. Pour ce faire on doit cliquer sur le bouton « **Enregistrer** » qui en background fait plusieurs tâches à la fois :

- Créer un fichier qui contient les informations de serveur dans le répertoire approprié, dans ce cas : ***/usr/local/nagios/etc/serveurs_windows/***.
- Pour compléter l'ajout de serveur, il faut ajouter son nom dans le fichier de groupe des hôtes **« hostgroup_windows.cfg ».**
- Redémarrer Nagios automatiquement sans ouvrir un terminal :

```
try {
Process test = Runtime.getRuntime().exec("gksudo service nagios restart");
       } catch (Exception e) {}
```

Après avoir terminé l'ajout complet de notre serveur, on peut consulter l'interface web de Nagios pour vérifier la réussite de ce qu'on a déjà fait et débuter la supervision de serveur, alors il suffit de cliquer sur le bouton « **Aller vers l'interface Web** », et enfin, sortir de la fenêtre en cours avec le bouton « **Quitter** ».

Pour ajouter un serveur Linux, il suffit de cliquer sur le bouton « **Ajouter des serveurs Linux** », ce qui conduit à ouvrir la fenêtre suivante :

Figure 15 : Fenêtre « Ajouter des serveurs Linux».

Ce qui concerne le principe de fonctionnement, c'est le même cas des serveurs Windows, le seul changement est situé dans le chemin d'emplacement d'hôte : (`"/usr/local/nagios/etc/serveurs_linux/"`), et pour le groupe d'hôtes :

(`"/usr/local/nagios/etc/serveurs_linux/hostgroup_linux.cfg"`).

2.3. Modifier des serveurs

Dans le menu « **Fichier** » on trouve aussi « **Modifier des serveurs** », Lorsqu'on clique sur cet item, on obtient une fenêtre qui contient deux boutons, l'un pour modifier des serveurs Windows et l'autre pour les serveurs Linux :

Figure 16 : Fenêtre « Modifier des serveurs».

Pour modifier un serveur Windows on clique sur le bouton « **Modifier des serveurs Windows** », ce qui conduit à ouvrir la fenêtre suivante :

Figure 17 : Fenêtre « Modifier des serveurs Windows» .

Lorsqu'on clique sur le bouton « **Ouvrir le fichier** », une boite d'ouverture est survenue vers les fichiers des serveurs Windows, alors on choisit le serveur que nous voulons de le modifier, faire la modification et l'enregistrer avec le bouton « **Enregistrer** », les mêmes étapes concernant les serveurs linux.

Figure 18 : Fenêtre « Modifier des serveurs Linux».

Conclusion

La réalisation de cette application permet de déduire qu'elle est facilitée aux débutants dans le domaine de supervision avec Nagios, une meilleure utilisation de ce dernier avec beaucoup d'avantages tels que le grand gain de temps et l'obtention des résultats pour améliorer les performances concernant les machines ou bien les hôtes a supervisés.

Conclusion générale

Conclusion générale

Pour conclure, l'administration réseaux et systèmes est un facteur de réussite primordiale pour les entreprises puisqu'elle leur permet le renforcement de la sécurité et guider les gérants de l'entreprise vers une bonne évolution du système d'information en fournissant une meilleure qualité de services.

La supervision du réseau donne les informations nécessaires à la gestion du réseau et trouve leurs tendances avec une rapidité de localisation des problèmes réseau.

Dans la deuxième partie de notre mémoire, nous avons démontré que le monitoring est un bon complément de l'administration réseaux et système ayant pour but la surveillance des réseaux en cours leurs maintenant en bonne santé toute en assurant la disponibilité et l'amélioration des performances.

Concernant le troisième chapitre, on a étudié l'une des solutions open source qui est un outil de monitoring réseau connu sous le nom de « Nagios », ayant pour fonction de surveiller les hôtes et quelques services spécifiques, alertant l'administrateur des états des machines et équipements présents sur le réseau de l'entreprise.

Pour le quatrième chapitre, nous avons fait une démonstration sur le fonctionnement de Nagios afin de connaitre au mieux son fonctionnement et son rôle de supervision à partir de la mise en place d'une configuration au niveau des machines ou bien les hôtes que l'on souhaite de surveiller.

Ce travail effectué dans le cadre d'un projet professionnel qui nous a permis de comprendre les concepts de base de la supervision des réseaux avec l'utilisation du logiciel Nagios sous linux et en le mettant en coordination avec une interface (GUI) qu'on a développé, cette dernière va permettre à un utilisateur novice une configuration avancée de Nagios.

Nous souhaitons que notre travail constitue un support d'aide aux débutants dans le domaine leurs permettant de bien assimiler la supervision informatique.

Bibliographie

Bibliographie

[2] : Sovanna. T., 'Administration et supervision réseau', Octobre 2009, Rapport TP.
http://www.lacl.fr/tan/Reseau/resSNMP.pdf

[3] : Thierry B., Matthieu V., 'Les outils d'administration et de supervision réseau « L'exemple de Nagios »' , Décembre 2004, Compte rendu.
http://web.univ-pau.fr/~cpham/M2SIR/BIBLIO/DOC04-05/Nagios.pdf

[4] : Baudoin M., 'Administration de réseaux'.
http://www.babafou.eu.org/ensta/b1-4/b1-4.pdf

[5] : Coponat P., Reynier S., 'Supervision réseau', Rapport.
http://master-info.univ-lyon1.fr/M2SIR/2006/supervision.pdf

[7] : Bonnet P., 'Introduction à la supervision', Novembre 2010, Mémoire Master SMaRT (Présentation PowerPoint). Université Lille 1 : Sciences et technologie.
http://www-lagis.univ-lille1.fr/~bonnet/supervision/Cours_intro_super.pdf

[8] : Manseri. H., 'Mise en œuvre d'un système de supervision au sein du service SI2 du LIRMM', Du 16 avril au 30 septembre 2009, Rapport de stage.
http://www2.lirmm.fr/IC/Supports/FMIN420-Stages/Archives_Rapports_FMIN420/Rapport_de_stage_Hakima_MANSERI.pdf

[9] : Mabo E., Niang A., 'La supervision avec NAGIOS, sécurité des systèmes informatiques', janvier 2009, Mémoire Master.
http://www.matael.info/documents/Supervision-Nagios-Centreon.pdf

[10] : Jean G., '***Nagios*** 3 pour la supervision et la métrologie : Déploiement, configuration et optimisation', Eyrolles, 2009, pp 32-51. ISBN : 978-2-212-1247.

Webographie

[1] :http://www.nationalrt.com/administration-des-reseaux-informatique.html,

[Administration des réseaux informatique], consulté le 02/03/2013.

[6] : http://www-igm.univ-mlv.fr/~dr/XPOSE2004/abouvet/smtpPres.htm

[Les protocoles de messagerie : SMTP, POP et IMAP « présentation SMTP »], consulté le 07/03/2013.

[11] : http://www.nationalrt.com/administration-des-reseaux-informatique.html

[Nagios pour débutants], consulté le 27/05/2013.