République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique Université Abderrahmane Mira – Bejaia
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Administration et Sécurité des Réseaux

Thème

Mise en place d'un outil de supervision dans un réseau d'entreprise.

Cas d'étude : OPGI.

Réalisé par :

Melle HOUACINE Yasmine et *Melle* DJALI Lylia

Soutenu devant le jury composé de :

Président : Mr LARBI Ali	Université A. MIRA de Bejaïa
Examinateur : Mr MEHAOUED Kamal	Université A. MIRA de Bejaïa
Encadrant : Mr OUZEGGANE Redouane	Université A. MIRA de Bejaïa

Septembre 2020

Remerciements

En guise de reconnaissance, nous tenons à témoigner nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de notre projet de fin d'étude et à l'élaboration de ce modeste travail.

Nos sincères gratitudes à Mr OUZZEGANE Redouane pour la qualité de son enseignement, ses conseils et son intérêt incontestable qu'il porte à tous les étudiants.

Nous tenons à remercier l'ensemble du personnel de l'OPGI et Mme HADDADOU en particulier pour sa patience, ses conseils pleins de sens et pour le suivi et l'intérêt qu'elle a porté à nos travaux.

Nous remercions les membres du jury d'avoir accepté de juger et d'évaluer ce travail.

HOUACINE Yasmine & DJALI Lylia

Dédicace :

Je dédie ce travail :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mon frère Juba,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

A ma meilleure amie Lynda.

Merci d'être toujours là pour moi.

HOUACINE Yasmine

Dédicace :

Je dédie ce travail :

A toutes les personnes qui me sont chères et à toutes les personnes qui m'ont aidé de loin ou de près surtout ma mère, ma sœur, mes frères, mes belles sœurs et mes tantes, ces personnes qui m'ont soutenu tout au long de mes études.

Je vous aime énormément et je vous en serai toujours reconnaissante.

DJALI Lylia

Table des matières

Table des figures

Chapitre 1

Chapitre 2

Chapitre 3

Chapitre 4

Liste des tableaux

Liste des acronymes

ADSL	Asymmetric Digital Subscriber Line
API	Interface de programmation
ASN.1	Abstract Syntax Notation One
BDD	Base De Données
CMDB	Configuration Management DataBase
CPU	Central Processing Unit
DMTF	Distributed Management Task Force
HDD	Hard Disk Drive
IETF	Internet Engineering Task Force
IPMI	Intelligent Platform Management Interface
ISO	Organisation internationale de normalisation
IT	Infrastructure Informatique
J2SE	Java 2 Platform, Standard Edition
JMX	Java Management Extensions
MIB	Management Information Base
MRTG	Multi Router Traffic Grapher
NMAP	Network Mapper
NMS	Network Management Station
NRPE	Nagios Remote Plugin Executor
NSCA	Nagios Service Check Acceptor
PDU	Protocol Data Unit
SMI	Structure of Management Information
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
SSII	Société de Services et d'Ingénierie en Informatique
UDP	User Datagram Protocol

VPN	Virtual Private Network
WBEM	Web Based Enterprise Management
WMI	Windows Management Instrumentation

Introduction générale

Les entreprises quel que soit leur domaine veillent toujours à être dans le centre de la concurrence économique et à garder leur bonne réputation, et pour cela elles donnent beaucoup d'importance à leur système informatique avec toutes ses composantes puisque le système d'information et de communication est actuellement la clé de voute de 1'entreprise. II permet de collecter, de traiter, de communiquer toutes les informations stockées, archivées et générées par 1'entreprise. Toutes ces informations sont capitales pour sa survie et représentent son histoire et son savoir-faire.

Vu que le système informatique est au cœur des activités d'entreprise, sa maîtrise devient primordiale, puisque, il doit fonctionner pleinement et en permanence pour garantir la fiabilité et l'efficacité exigées. D'autre part, les problèmes liés au système informatique tels que les défaillances, les pannes, les coupures et les différents problèmes techniques doivent être réduits, du fait qu'une indisponibilité du système ou du réseau peut causer des pertes considérables.

Afin de minimiser le nombre de ces pertes, une sorte de surveillance et de contrôle s'avère obligatoire ; c'est ainsi que la notion de la ***« supervision informatique »*** a vu le jour et est devenue une tâche vitale pour tout système informatique.

La supervision informatique est une branche informatique qui doit assurer trois fonctionnalités : garantir la disponibilité du système et réseau de l'entreprise ; anticiper sur l'apparition d'éventuels problèmes ; garantir une durée d'intervention et de résolution minimale.

Notre travail consiste à la mise en œuvre et la configuration d'un outil de supervision réseaux. Pour bien comprendre le travail et bien cerner les problèmes techniques, nous avons réalisé un stage au niveau de l'OPGI, qui se charge de la promotion immobilière. Ceci nous a permis d'établir la problématique qui consiste en la détection de pannes et éviter la perte du temps et proposer une solution de supervision réseau à base de l'outil Nagios.

Pour bien présenter notre travail, notre mémoire sera structuré sur quatre parties, comme suit :

Dans le premier chapitre, nous procéderons à la présentation générale de l'organisme d'accueil ainsi que l'étude et la critique de l'existant, ceci nous a permis d'établir une problématique. Par la suite, les fondements sur lesquels la supervision se base, ainsi que les différents standards et protocoles qui permettent la supervision d'infrastructures réseaux et la présentation des différents outils de supervision feront l'objet du deuxième chapitre. En troisième lieu, nous ferons une présentation plus détaillée de la solution choisie avec l'ensemble de ces fonctionnalités ainsi que son architecture. En dernier chapitre, nous illustrerons les étapes de notre travail et nous réaliserons quelques tests.

Finalement, nous terminerons notre mémoire par une conclusion générale qui récapitulera les principales observations concernant l'évolution du travail et nous indiquons également comment les travaux réalisés tout au long de ce mémoire pourraient être améliorés.

Introduction

Dans ce premier chapitre, nous présenterons l'organisme d'accueil qui est l'OPGI (l'Office de Promotion et de Gestion de l'Immobilier de Bejaia), ceci se fera en passant par l'historique de cet organisme puis par une présentation générale de son organigramme ainsi qu'une présentation du service ou nous étions affectées puis nous parlerons du cadre du projet et nous finirons par l'étude préalable qui est une étape primordiale dans le déroulement du projet.

1. Présentation de l'organisme d'accueil

1.1. Présentation de l'OPGI

L'Office de Promotion et de Gestion Immobilière (OPGI) est doté, de la personnalité morale et de l'autonomie financière. Il est réputé commerçant dans ses rapports avec les tiers et est soumis aux règles de droit commercial.

Il peut exercer son activité sur l'ensemble du territoire national. Il est chargé dans le cadre de la mise en œuvre de la politique sociale de l'Etat, de promouvoir le service public en matière de logement, notamment pour les catégories sociales les plus démunies. En outre, il est chargé :

- De la promotion immobilière
- De la promotion foncière
- De la maîtrise d'ouvrage déléguée pour le compte de l'Etat ou pour tout autre opérateur.
- Des actions de prestation de service en vue d'assurer l'entretien, la maintenance, la réhabilitation et la restauration des biens immobiliers.
- De toutes actions visant l'accomplissement de sa mission.

Aussi il est habilité à gérer les biens immobiliers qui lui sont confiés notamment :

- La location et/ou la cession des logements et locaux à usage professionnel, commercial et artisanal.
- Le recouvrement des loyers et des charges locatives ainsi que les produits de la cession des biens immobiliers qu'il gère.
- La préservation des immeubles et de leurs dépendances en vue de leur maintien en état permanent d'habitabilité.

- L'établissement et la tenue à jour de l'inventaire des immeubles constituant le parc immobilier dont il assure la gestion.

L'Office de Promotion et de Gestion Immobilière est situé au niveau de la cité du commandant Mohamed FADHEL « Dit Si. H'MIMI »Ex. Pépinière BP 540 ter liberté-Bejaia.

1.2. Historique sur L'OPGI

OPGI (Office de Promotion et de Gestion Immobilière de Bejaia), crée en 1977, n'était alors présent qu'au seul chef-lieu de la wilaya de Bejaia, ce n'est qu'en 1984 suite à l'intégration de l'ex-service du logement de wilaya que ce dernier a élargi sa représentation au niveau des daïras.

L'organisation des OPGI à travers le temps a connu trois phases successives soit de l'ordonnance 76/93 du 23 octobre 1976 au décret 91/147 du 12 Mai 1991.

• L'ordonnance 76/93 du 23 octobre 1976

Érige l'OPGI en établissement public à caractère administratif dont la compétence territoriale s'étend à la wilaya ; il a pour objet la gestion d'un service public à caractère administratif ou industriel et commercial.

• Le décret N° 85-270 du 05/11/1985

Transforme l'organisation et le fonctionnement des OPGI en les désignant « établissements à caractère économique doté de la personnalité morale et de l'autonomie financière ; chargé de la gestion des programmes publics d'habitat à usage locatif de la prestation de services pour le compte de l'assemblée générale des copropriétaires et des locataires ainsi que du cadre de vie dans le cadre de convention établies avec les APC concernées. Les règles de fonctionnement sont celles édictées par le décret 83/200 du 19 mars 1983 précisant les conditions de création, d'organisation et de fonctionnement de l'établissement public local.

• **Le décret 91/147 du 12 Mai 1991**

Les transformant en établissement public à caractère industriel et commercial à vocation national, chargés de promouvoir le service public en matière de logement outre qu'ils sont chargés accessoirement de :

- La promotion immobilière ;
- La maîtrise d'ouvrage déléguée pour le compte de tout opérateur ;
- De la promotion foncière ;
- Des actions de prestation de services en vue d'assurer l'entretien, la maintenance, la réhabilitation et la restauration des biens immobiliers.

2. Organigramme de l'organisme

Les différentes structures de l'OPGI Bejaia sont présentées par l'organigramme ci-dessous :

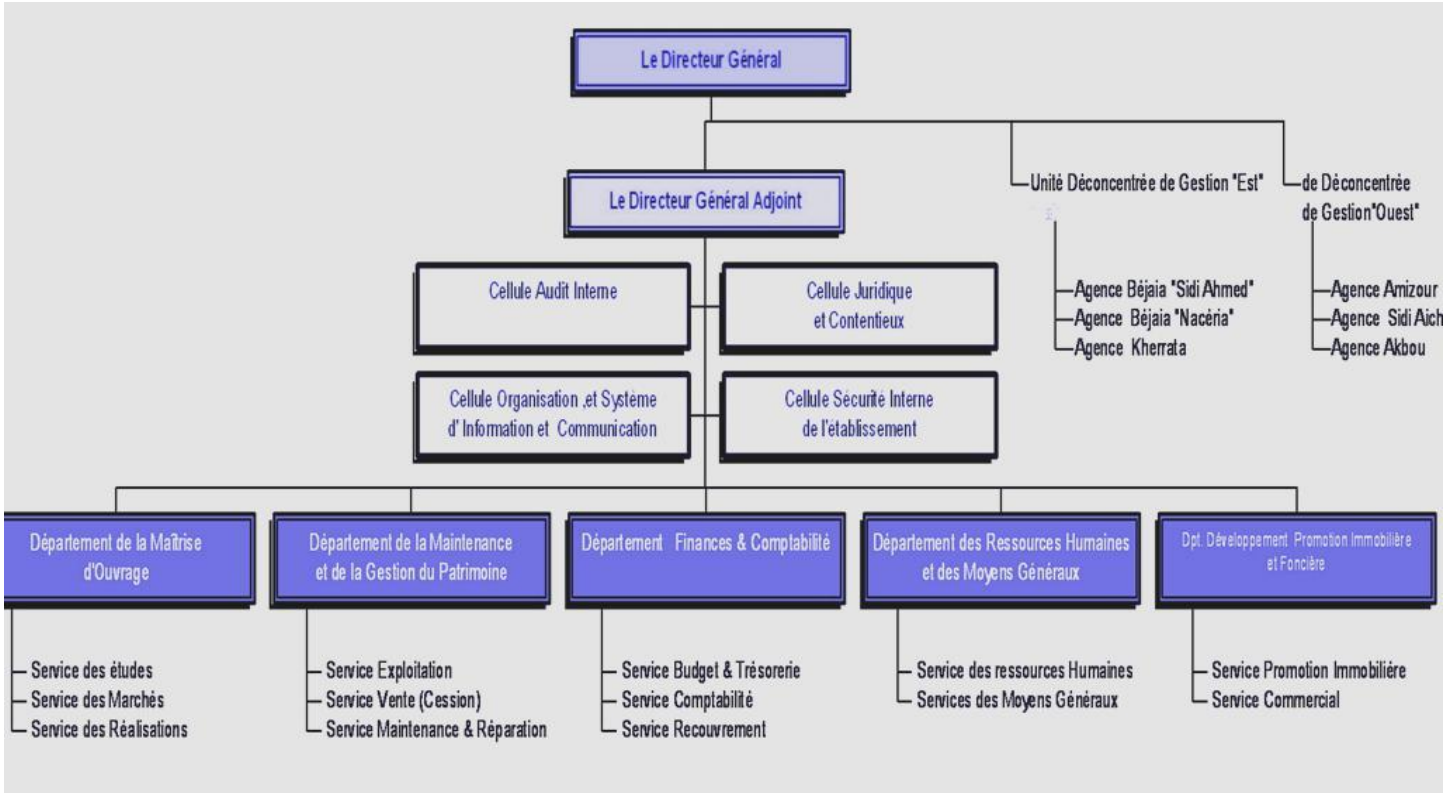


Figure 01: Organisme de l'OPGI de Béjaia

2.1. Présentation du service d'accueil

La cellule informatique composée de sept (07) éléments, un chef de cellule informatique, un ingénieur réseau, un programmeur applicatif, un technicien supérieur en informatique, et un apprenti encadré par l'ingénieur. La cellule informatique est chargée de :

- Développement et réalisation des projets informatiques.
- Administration du réseau.
- Gestion du parc informatique.
- Maintenance du système informatique.
- Introduction de nouvelles technologies.
- Formation du personnel aux techniques informatiques.
- Sauvegarde et archivage des données de l'entreprise.
- L'alimentation de la page Facebook de l'office.
- Maintenance du matériel informatique.

2.2. Organigramme du service d'accueil

Notre stage s'est déroulé au sein de la cellule informatique, qui est une cellule située au sous-sol de l'OPGI, ce schéma représente ses différents services

Figure 02: l'Organigramme du service d'accueil

3. Cadre du projet

Dans le cadre d'obtention d'un diplôme du mastère professionnel en Administration et sécurité des réseaux à l'université Abderrahmane Mira de Béjaia, il nous a été demandé d'élaborer ce petit mémoire suite à un stage. C'est dans ce cadre et pour l'année universitaire 2019/2020 que nous avons effectué le présent projet au sein de l'OPGI (Office de Promotion et de Gestion Immobilière) qui consiste à mettre en place un outil de supervision du réseau informatique.

Dans ce contexte s'introduit le travail à faire qui consiste à la Recherche, Implémentation et configuration d'une solution Open Source qui vise à superviser à distance les différents serveurs de la société avec gestion des alertes dans un environnement Multi plateformes.

4. Etude de l'existant (Situation Informatique)

Le parc informatique de l'Office de Promotion et de Gestion de l'Immobilier de Bejaia est composé d'un ensemble d'environ 94 postes de travail et de 21 imprimantes réseaux et locaux, et d'un serveur, un modem ADSL, un point d'accès, et de 5 switchs branchés à des panneaux de brassages.

4.1. Architecture réseau de l'OPGI

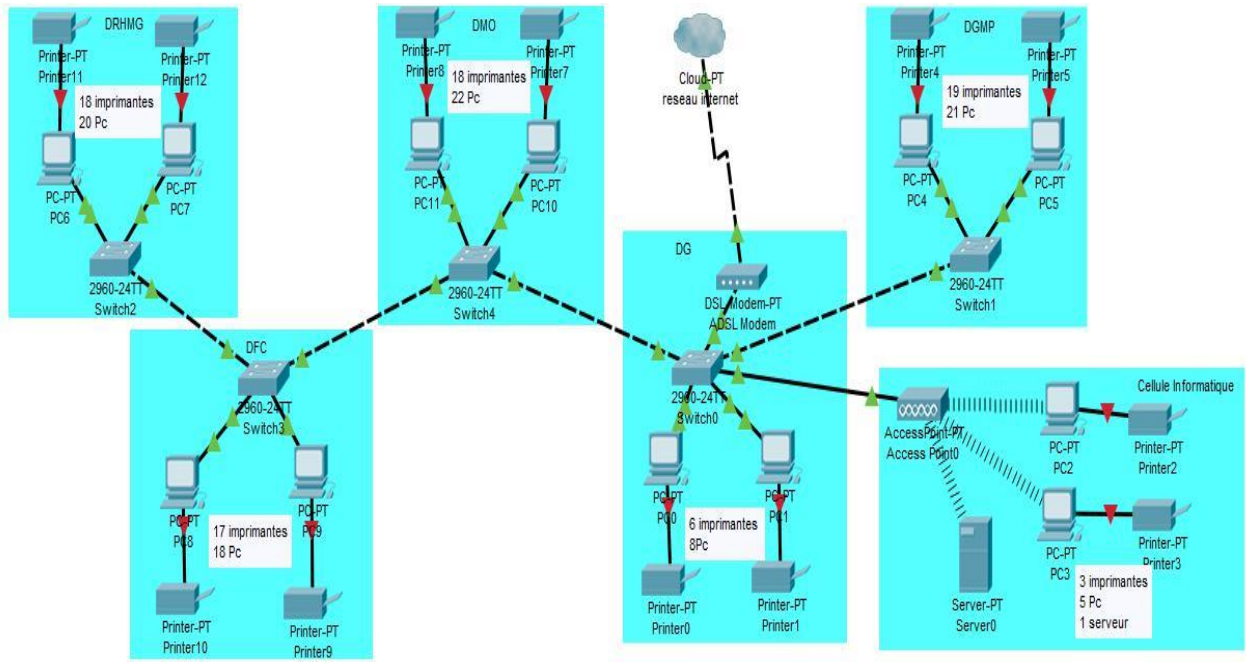


Figure 03 : Architecture réseau de l'OPGI

5. Critique de l'existant

Lors de l'étude que nous avons faite, nous avons relevé les problèmes suivants :

- Aucun outil de supervision système et réseau n'est mis en place au sein de l'entreprise.
- Pas d'alerte en cas de panne ou problème de fonctionnement sur le réseau.
- Plus le nombre des équipements et des services augmente plus il est difficile de détecter la défaillance des équipements (charge CPU, Etat mémoire, surcharge du disque….).
- Le seul moyen de détecter ces anomalies ne peut se faire que par la réception des différentes plaintes et réclamations du personnel de l'OPGI.
- L'ingénieur trouve des difficultés de vérifier la disponibilité des équipements surtout pour les locaux distants.
- Le diagnostic des pannes nécessite énormément de temps.

L'objectif de notre travail est donc de trouver une solution optimale pour la supervision de ses équipements, offrir la possibilité de devenir proactif face aux problèmes et le plus important, de pouvoir détecter et interpréter les causes et origines des problèmes rencontrés afin de les fixer le plus rapidement possible.

6. Solution proposée

Vu les inconvénients cités précédemment, nous proposons la mise en place d'un outil de supervision système et réseau open source qui assure les tâches suivantes :

- Déclencher des alertes lors de la détection des pannes.
- Offrir des renseignements supplémentaires voir charge CPU, espace disque, mémoire disponible, etc.
- Offrir à l'administrateur la possibilité de réagir le plus rapidement possible face aux pannes.
- Avoir une interface graphique compréhensible pour l'interaction entre l'utilisateur et le logiciel.
- Diagnostiquer l'état du réseau.

Conclusion

Ce chapitre a été conçu pour mieux connaître l'environnement du travail, en présentant l'entreprise d'accueil mais aussi pour comprendre les besoins et les attentes du client.

Le chapitre suivant sera consacré à effectuer le choix de l'outil à mettre en place.

Introduction

Pour pouvoir garantir une activité ainsi qu'une bonne notoriété de son entreprise, il est primordial de réduire au maximum les problèmes informatiques. C'est pour cela que les entreprises ont désormais recours à des sociétés de supervision informatique au travers de contrat de maintenance informatique.

Dans ce chapitre, nous allons présenter le concept de la supervision informatique et la manière dont il a été normalisé par l'ISO 7498/4, ensuite nous procédons à une étude de quelques outils de supervision.

1. La supervision en informatique

1.1. Définition

La supervision informatique est une technique de surveillance, d'analyses et d'alertes permettant de pallier les problèmes liés à tous les niveaux de fonctionnement informatique d'une entreprise.

L'utilisation de la supervision informatique a un but bien précis : rendre l'entreprise plus performante et surtout proactive.

La surveillance des équipements informatiques permet de détecter toute anomalie en temps réel, et de pouvoir ainsi la traiter dans les meilleurs délais. Selon le type d'anomalie relevée, le traitement se fera en maintenance locale ou à distance [1].

La supervision informatique peut concerner tout l'existant informatique et téléphonique de l'entreprise, le courant électrique, les disponibilités réseaux (fibres, ADSL (Asymmetric Digital Subscriber Line)), les serveurs, les imprimantes et les autres éléments actifs constituant le réseau (hubs, switchs, routeurs, etc.).

Elle doit répondre aux préoccupations suivantes :

- Technique : surveillance du réseau informatique, de l'infrastructure de l'entreprise ;
- Fonctionnelle : surveillance des machines informatiques et de production ;
- Applications : suivi des applications dans le cadre d'un processus métier [2].

1.2. Rôle de la supervision

Deux phases sont importantes pour que les administrateurs soient capables d'atteindre l'objectif visé par la supervision, à savoir, surveiller le système et garantir sa disponibilité même en cas d'anomalie. Nous pouvons citer les rôles suivants :

- ➔ Tenter de prévenir en cas de problème (défaillances matérielles ou interruption des services) et garantir une remontée d'information rapide ;
- ➔ Automatiser les tâches de récupération des applications et des services en assurant des mécanismes de redondance en une durée d'intervention minimale (par exemple : le redémarrage des services interrompus, l'arrêt de la machine en cas du surcharge du CPU (Cenral Process Unit), la sauvegarde des données en cas du risque de perte d'un disque dur en miroir, etc.)[3].

2. La supervision des réseaux

2.1. Définition

La supervision réseau fait référence à la surveillance du bon fonctionnement des réseaux informatiques et des services informatiques connectés sur ces réseaux.

La surveillance du réseau porte plus spécifiquement sur la qualité (bande passante) et la sécurité de la connexion Internet mais aussi, par extension, à l'état des services et matériels connectés : serveurs, imprimantes, postes de travail, etc.

La supervision réseau est un des 3 types de supervision informatique avec la supervision système (bas niveau) et la supervision applicative [4].

2.2. Principe

La supervision réseau peut être mise en œuvre sur la base d'analyse de logs, de résultats de commandes et de scripts locaux mais c'est surtout sur la base de protocoles standards comme le protocole SNMP pour que le monitoring des réseaux informatiques fonctionne. De nombreux logiciels existent pour ce fait La communauté du libre (Open Source) est particulièrement active dans le monitoring. Les logiciels permettent d'assister le technicien grâce aux alertes SMS, email mais aussi en proposant des solutions concrètes pour résoudre ou anticiper un problème.

Les logiciels offrent une couverture fonctionnelle plus ou moins large mais s'adaptent, la plupart du temps, à tout type d'entreprise [4].

Des solutions logicielles proposant la supervision du système d'information sont capables de vérifier l'état d'un équipement ou service donné à intervalle régulier. Les données de résultats sont exploitables sous 3 formes différentes : booléen (Le service est-il disponible ou non?), numérique (Quel est le temps de réponse de la machine ?) ou qualitatif (Quelle type d'erreur est renvoyée ?). Les solutions de supervision permettent également de remplir des rapports d'activité selon la nature du service surveillé, comme des graphes d'utilisation réseau, ou encore des historiques de changement d'état sur le temps [5].

2.3. Méthodes de la supervision

Deux grandes méthodes de supervision sont utilisées avec plusieurs variantes : les méthodes active et passive, détaillées dans les paragraphes suivants :

✓ **Supervision active**

La supervision active est la plus classique et la plus utilisée. Elle consiste en l'envoi de requêtes d'interrogation et de mesure par la plateforme de supervision. Elle a l'avantage d'être fiable : les vérifications se font de manière régulière et en mode question-réponse.

Cette méthode est composée de trois étapes :

- ➢ Le serveur envoie une requête vers la ressource supervisée.
- ➢ La ressource répond à la requête du serveur.
- ➢ Le serveur analyse l'information et détermine un état pour la ressource.

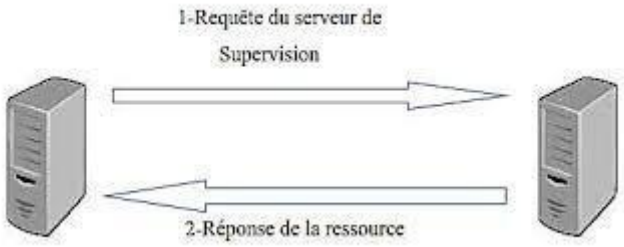


Figure 04 : Supervision active [27]

Les deux principaux protocoles de supervision active sont SNMP et WMI, ces deux protocoles sont à privilégier car non intrusifs : les agents sont natifs aux systèmes supervisés.

Certains protocoles d'administration peuvent également être utilisés pour la supervision: IPMI et JMX, les protocoles systèmes SSH et Telnet sont également très utilisés.

✓ **Supervision passive**

La supervision passive l'est du point de vue du serveur de supervision : ce sont les ressources supervisées qui transmettent des alertes au serveur de supervision :

- ➢ La ressource supervisée vérifie son état et transmet de manière autonome le résultat au serveur de supervision.
- ➢ Le serveur de supervision reçoit l'alerte et la traite.

Figure 05 : Supervision passive [27]

Le protocole standardisé et privilégié pour la supervision passive est SNMP avec le mécanisme de trappes. La communauté Nagios propose également un module passif dédié appelé NSCA (Nagios Service Check Acceptor)

3. La norme ISO 7498/4

L'ISO a également défini un modèle conceptuel de référence pour la gestion du réseau: la norme ISO 7498-4. Dans ce modèle sont identifiés cinq principaux domaines d'activité de la gestion du réseau: *faute, la configuration, la comptabilité, la performance, la sécurité*, généralement indiquée par l'acronyme FCAPS.

- Fault-management: Détection et gestion des fautes.
- Configuration : Gestion et configuration des équipements de réseau
- Accounting : (Comptabilité) collecte de données sur l'utilisation des ressources par les différents utilisateurs
- Performance : le suivi des performances du réseau et la mise en œuvre des procédures qui permettent de préserver les niveaux de performance requis

- Security : la gestion de la sécurité de l'infrastructure réseau afin d'empêcher l'utilisation à des fins non légitimes.

3.1. Gestion des défauts

Le terme Gestion des défauts fait référence à un ensemble d'activités pour détecter, isoler et corriger les défauts et défaillances dans un réseau. Les défauts peuvent être persistants ou transitoires. Les activités de gestion des défauts comprennent:

- la gestion et l'analyse des journaux d'erreurs des dispositifs.
- capacité de recevoir et réagir aux signaux de défaut émis par les dispositifs.
- identification des défaillances.
- l'exécution de tests de diagnostic.
- correction des défauts (au moyen d'interventions sur le matériel ou le logiciel des périphériques).

3.2. Gestion comptable

Les activités Gestion comptable permet de garder une trace de la quantité de ressources utilisées dans le système, par les utilisateurs individuels, des groupes d'utilisateurs ou d'applications. Les fonctions de gestion de la comptabilité comprennent :

- La production de rapports pour les utilisateurs, documentant les ressources consommées.
- L’imposition de limites d'utilisation (dimensions) et le seuil pour la charge.

3.3. Gestion de la configuration

Les activités gestion de la configuration comprennent l'identification des dispositifs, la collecte de données relatives à leur configuration, et l'imposition de nouvelles configurations, peut-être par la définition des valeurs paramétriques.

3.4. Gestion de la performance

Cette gestion comprend des fonctions pour:

- La collecte de données relatives au comportement des dispositifs.
- La collecte de données relatives aux services offerts par l'infrastructure.
- En cours d'exécution série de tests pour vérifier la qualité des services offerts par l'infrastructure réseau.
- Reconfiguration des dispositifs afin d'atteindre les objectifs de performance souhaités.

3.5. Gestion de la sécurité

Le but des activités gestion de la sécurité est de soutenir l'application des politiques de sécurité définies par les administrateurs réseau par le biais de fonctions spéciales qui incluent:

- L'utilisation des fonctions spéciales, des mécanismes de sécurité et des services de soutien ;
- La diffusion des informations pertinentes sur la sécurité ;
- Produire des rapports documentant des événements importants pour la sécurité [6].

4. Les standards de la supervision

Surveiller les systèmes d'information, permet de s'assurer d'une bonne disponibilité des services (Ex : l'arrêt d'un système de paiement par carte bancaire), ou la disponibilité d'un site de vente de billets en ligne (l'impact pour l'entreprise peut être très conséquent).

En détectant toutes les anomalies, on peut alerter par tout moyen à disposition (mail, sms, autres) et prévenir ainsi les défaillances. Une automatisation des tâches permettra de relancer les serveurs et d'intervenir à distance si nécessaire en toute sécurité, en relançant ainsi le service concerné [7].

Les systèmes de supervision utilisent des protocoles, très réglementés par la DMTF depuis 2005, parmi les principaux utilisés, nous relèverons 9 principaux protocoles dans la supervision :

4.1. Intelligent Platform Management Interface (IPMI)

C'est l'un des standards les plus utilisés, il concerne surtout les serveurs et cette interface intelligente de gestion de matériel permet, entre autres, de contrôler à distance certains composants très sensibles comme les sondes et autres ventilateurs.

4.2. Java Management Interface (JMX)

C'est l'API, qui permet de gérer une application en cours d'exécution. JMX est maintenant complètement intégré dans J2SE à partir de la version V. Certains experts estiment que le JMX est le SNMP de JAVA puisqu'il agit dynamiquement sur son comportement, génère des statistiques en temps réel sur son fonctionnement et notifie des dysfonctionnements.

4.3. Common Information Model (CIM)

Si l'on se base sur les écrits du DTMF, la norme CIM ou Protocole CIM, comprend en plus du méta modèle, une spécification et un schéma. Le méta modèle pour en définir la sémantique. La spécification qui définit les détails pour intégrer avec d'autres modèles de gestion. Le schéma, ensemble de classes avec ses propriétés qui fournit les descriptions des modèles en réel, incluant le cadre conceptuel structuré en couches distinctes ; modèle de base, schémas d'extension et le modèle commun.

4.4. Simple Network Management Protocol (SNMP)

C'est le protocole de communication et de gestion simplifiée du réseau. C'est le SNMP qui permet aux administrateurs de contrôler et de gérer (diagnostiquer) tous les éléments actifs du réseau. En langage SNMP on ne supervise pas, on manage, mais le résultat est similaire. Il est composé de 3 éléments essentiels : le superviseur, les nodes (ou nœuds en français) et les agents. Sans entrer dans les détails, c'est le SNMP qui permet de dialoguer entre le superviseur et les agents pour recueillir les objets dans la MIB.

4.5. Information Technology Infrastructure Library (ITIL)

C'est une norme, ensemble de bonnes pratiques diront d'autres, pour la bonne gestion d'un système d'information. Né en Grande Bretagne, et populaire en Europe depuis plus de 35 ans, il tend à s'implanter aux USA grâce à l'impulsion de certaines grandes SSII.

Basé sur la CMDB (Configuration Management DataBase), c'est une BDD (*Base De Données*) qui unifie les composants d'un système d'information et qui en plus, permet de comprendre l'organisation et de modifier la configuration si nécessaire. C'est la base même du système ITIL., en positionnant des blocs organisationnels et des flux d'informations, les recommandations ITIL, abordent des sujets aussi variés que :

- Les productions informatiques
- Les réductions des risques
- L'augmentation de la qualité du service
- L'efficacité globale du système d'information.

4.6. Standard Based Linux Instrumentation for Manageability (SBLIM)

SBLIM, Standards Based Linux Instrumentation for Manageability, est nommé par les experts en langage courant SUBLIME. Il s'applique aux machines LINUX et permet entre autres d'avoir accès aux technologies WBEM. Ce standard est exclusivement mis en avant par IBM qui en assure aussi le développement.

4.7. Web Based Enterprise Management (WBEM)

WBEM ensemble de standards de base intégrés dans les outils de supervision, pour faciliter l'échange entre plateformes et technologies. WBEM sont des standards Internet de gestion, surtout développés pour unifier les environnements dans l'informatique distribuée.

4.8. Web Services for Management (WS-MANAGEMENT)

WS-Management fournit la méthodologie pour échanger des informations d'administrations à travers les infrastructures IT, spécification fournie par le DTMF. Basé sur les Web Services (SOAP), il est très proche du protocole WBEM.

4.9. Windows Management Instrumentation

C'est le protocole pour les plateformes Windows, il étend le modèle du CIM, pour représenter les objets, dans cet environnement. Son interface cohérente et orientée objet, utilise les normes de l'industrie et permet aux informaticiens une utilisation simplifiée des tâches de gestion. L'accès aux données sous WMI, que ce soit en local ou à distance, est totalement transparent.

Tous ces protocoles sont normalisés et gérés par la DMTF : c'est un organisme mondial, où sont regroupés tous le grands constructeurs et donneurs d'ordre, qui met en place et gère les standards de la technologie. La DMTF simplifie la gérabilité de tous ces standards, grâce à la collaboration et à la participation des grandes sociétés mondiales de technologie, ainsi que des principaux constructeurs.

5. Le protocole SNMP

5.1. Présentation

Le protocole SNMP (Simple Network Management Protocol), conçu à l'initiative de CISCO, HP et Sun, puis normalisé par l'IETF et l'OSI, permet de contrôler à distance l'état des principaux constituants du réseau [8].

Les buts du protocole SNMP sont de :

- Connaître l'état global d'un équipement (actif, inactif, partiellement opérationnel...)
- Gérer les évènements exceptionnels (perte d'un lien réseau, arrêt brutal d'un équipement...) ;
- Analyser différents métriques afin d'anticiper les problèmes futurs (engorgement réseau...) ;
- Agir sur certains éléments de la configuration des équipements [9].

5.2. Les différentes versions du SNMP

Ce protocole d'administration, très répandu dans les réseaux locaux, est basé sur l'échange de messages entre les périphériques administrables et une station d'administration. Il existe 3 versions du protocole : SNMP v1, SNMP v2c et SNMP v3 [8].

- SNMP V1 : qui reste la version la plus utilisée car la plus « légère ».La sécurité de cette version est minimale car elle basée uniquement sur la chaîne de caractère appelée "communauté".
- SNMP V2C : est une version délaissée car trop complexe. Elle assure un niveau plus élevé de sécurité (authentification, cryptage…), des messages d'erreurs plus précis, autorise l'usage d'un Manager central. Cependant ce protocole utilise la structure d'administration de SNMP V1 (à savoir "communauté") d'où le terme SNMP V2C.
- SNMP V3 : permet de disposer des avantages de la version 2 sans en présenter les inconvénients. Elle définit un nouveau modèle de sécurité USM (User-basedSecurity Model) évitant le décryptage des messages de commande qui transitent sur le réseau et autorise des droits différents en fonction des utilisateurs.

Malgré tout la version SNMP V1 persiste encore sur les périphériques, plusieurs facteurs expliquent ce phénomène :

- Les infrastructures déployées en V1 ne sont plus modifiées, tout simplement car cela fonctionnait suffisamment à l'époque, du coup aucune modification n'y est appliquée.
- Les autres versions de SNMP ont été implémentées tardivement par les différents constructeurs.
- SNMP V1 demande très peu de ressources sur des petits équipements tels qu'une imprimante [10].

5.3. Architecture

Les différents éléments que l'on peut identifier avec le protocole SNMP sont synthétisés par le schéma ci-dessous.

- Les agents SNMP : ce sont les équipements (réseau ou serveur) qu'il faut superviser.
- Le superviseur SNMP : c'est une machine centrale à partir de laquelle un opérateur humain peut superviser en temps réel toute son infrastructure, diagnostiquer et résoudre les problèmes.
- La MIB : ce sont les informations dynamiques instanciées par les différents agents SNMP et remontées en temps réel au superviseur.

SNMP utilise le modèle client-serveur où le client est représenté par la station d'administration – NMS (Network Management Station) ou Manager – qui interroge des serveurs représentés par les agents SNMP implantés sur les nœuds administrables.

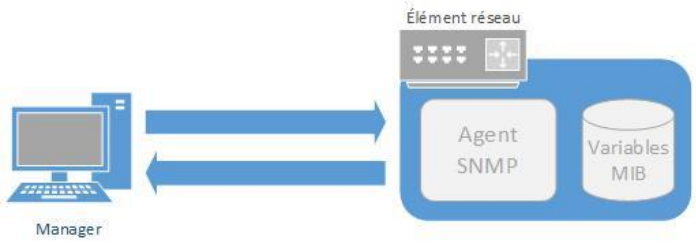


Figure 06: Architecture SNMP [8].

SNMP fonctionne au niveau 7 du modèle OSI mais s'appuie directement sur le protocole UDP il a donc besoin d'un numéro de port pour communiquer. Du fait qu'il utilise UDP, SNMP fonctionne en mode non connecté sans contrôle des données transmises.

5.4. Le manager

Rappelons que le Manager se trouvera sur une machine d'administration (un poste de travail en général). Il reste un client avant tout, étant donné que c'est lui qui envoie les différentes requêtes aux agents. Il devra disposer d'une fonction serveur, car il doit également rester à l'écoute des alertes que les différents équipements sont susceptibles d'émettre à tout moment. Le Manager dispose d'un serveur qui reste à l'écoute sur le port UDP 162 ainsi que d'éventuels signaux d'alarme appelés des "traps". Le Manager peut tout autant être installé sur une machine.

5.5. L'agent SNMP

L'agent est un programme qui fait partie de l'élément actif du réseau. L'activation de cet agent permet de recueillir la base de données d'informations et la rend disponible aux interrogations.

Les principales fonctions d'un agent SNMP :

- Collecter des informations de gestion sur son environnement local.
- Récupérer des informations de gestion telle que déni dans la MIB propriétaire.
- Signaler un évènement au gestionnaire.

Par ailleurs même si la principale fonction de l'agent est de rester à l'écoute des éventuelles requêtes du Manager et y répondre s'il y est autorisé, il doit également être capable d'agir de sa propre initiative, s'il a été configuré.

5.6. MIB

Chaque agent SNMP maintient une base de données décrivant les paramètres de l'appareil géré. Le Manager SNMP utilise cette base de données pour demander à l'agent des renseignements spécifiques. Cette base de données commune partagée entre l'agent et le Manager est appelée Management Information Base (MIB).

Généralement ces MIB contiennent l'ensemble des valeurs statistiques et de contrôle définis pour les éléments actif du réseau.

Un fichier MIB est écrit en utilisant une syntaxe particulière, cette syntaxe s'appelle SMI (Structure of Management Information), basée sur ASN.1.

En résumé, les fichiers MIB sont l'ensemble des requêtes que le Manager peut effectuer vers l'agent. L'agent collecte ces données localement et les stocke, tel que défini dans la MIB. Ainsi le Manager doit être conscient de la structure de la MIB afin d'interroger l'agent au bon endroit. La structure d'une MIB est une arborescence hiérarchique dont chaque nœud est défini par un nombre ou un Object IDentifier (OID). Chaque identifiant est unique et représente les caractéristiques spécifiques du périphérique géré. Lorsqu'un OID est interrogé, la valeur de retour est une séquence de chiffres séparés par des points [10].

Une MIB est un arbre très dense, il peut y avoir des milliers d'OID dans la MIB.

Voici un exemple de structure MIB :

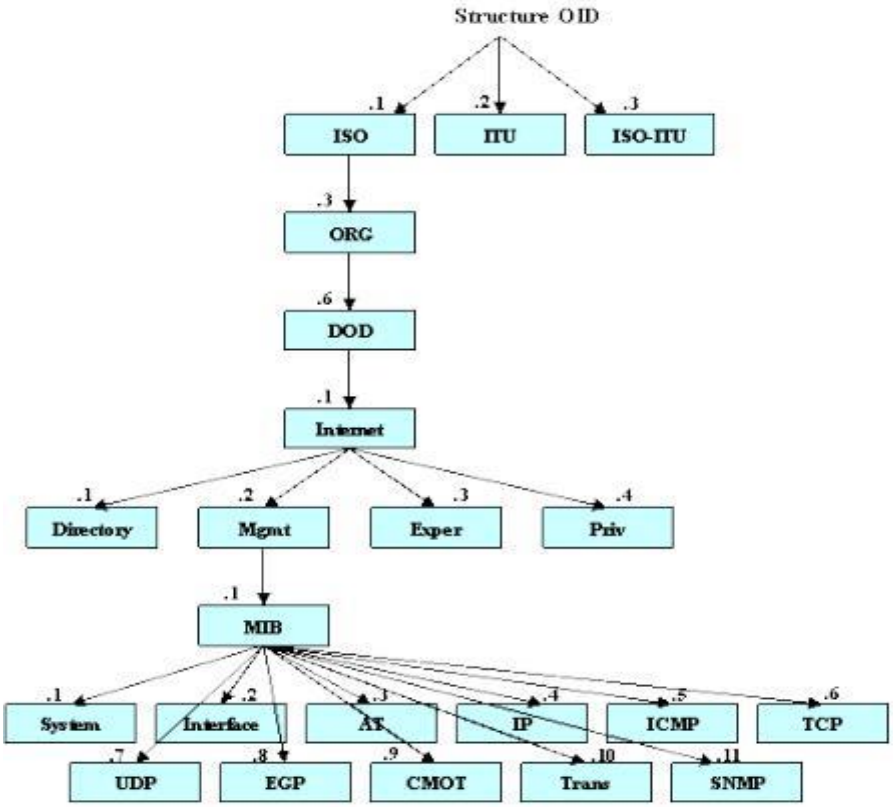


Figure 07 : Structure d'un OID [10].

Ainsi, pour interroger les différentes variables d'activité sur un appareil, il faudra explorer son arborescence MIB.

Ensuite, pour accéder aux variables souhaitées, on utilisera l'OID qui désigne l'emplacement de la variable à consulter dans la MIB. On aura par exemple sur un commutateur Nortel Passeport l'OID .1.3.6.1.4.1.2272.1.1.20 désignant le taux de charge du CPU.

5.7. Les requêtes SNMP

Le mécanisme de base du protocole SNMP est constitué d'échanges de type requête/réponse appelé PDU pour Protocol Data Unit. En fonction de la version du protocole SNMP utilisé, différentes commandes sont possibles. La structure des paquets utilisés par le protocole SNMP V1, est définie dans la RFC 1157. Les requêtes SNMP vont contenir une liste d'OID (Object identifier) à collecter sur l'agent SNMP.

Les types de requêtes du manager SNMP vers l'agent SNMP sont :

- Get Request : Le manager interroge un agent sur les valeurs d'un ou de plusieurs objets d'une MIB.
- Get Next Request : Le manager interroge un agent pour obtenir la valeur de l'objet suivant dans l'arbre des objets de l'agent. Cette interrogation permet de balayer des objets indexés de type tableau.
- Get Bulk Request : Introduite avec la version 2 du protocole SNMP, cette requête permet de mixer la commande GET et GETNEXT pour obtenir des blocs entiers de réponses de la part de l'agent.
- Set Request : Le manager positionne ou modifie la valeur d'un objet dans l'agent.Les réponses ou informations de l'agent vers le manager sont :
- Get Response : L'agent répond aux interrogations du manager.
- Trap : L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent n'attend pas d'acquittement de la part du manager.
- Notification : Introduite avec la version 2 du protocole SNMP. L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent n'attend pas d'acquittement de la part du manager.
- Inform : Introduite avec la version 2 du protocole SNMP. L'équipement génère un envoi vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent attend un d'acquittement de la part du manager et il y aura une retransmission en cas de non réponse. [10]

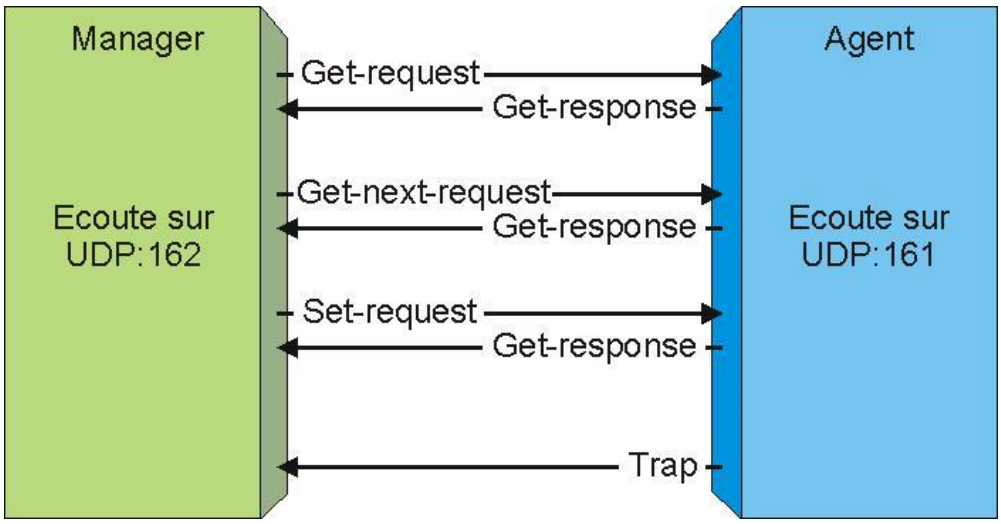


Figure 08 : Les échanges entre le manager et l'agent SNMP [26].

6. Logiciels OPEN SOURCE

Une supervision efficace passe donc par des logiciels de monitoring qui centralisent l'information de la santé du réseau pour le compte des directions des systèmes d'information.

Il existe de différents outils de supervision utilisés dans les entreprises. Nous allons voir les différents outils qui existent les comparer et voir celui qui est meilleur.

6.1. Nagios

Est un logiciel de supervision de réseaux créé en 1999 par Ethan Galstad. Il est considéré comme étant la référence des solutions de supervision Open Source. C'est un outil très complet pouvant s'adapter à n'importe quel type d'utilisation avec des possibilités de configuration très poussées. La modularité et la forte communauté (> 250 000) qui gravite autour de Nagios (en participant au développement de nombreux plugins et addons) offrent des possibilités en termes de supervision qui permettent aujourd'hui de pouvoir superviser pratiquement n'importe quelle ressource [11].

6.1.1. Avantages

Ce logiciel contient de nombreux avantages tels que :

- La supervision à distance peut utiliser SSH ou un tunnel SSL (notamment via un agent NRPE).
- Les plugins sont écrits dans les langages de programmation les plus adaptés à leurs tâches : scripts shell (Bash, ksh, etc.), C++, Perl, Python, Ruby, PHP, C#.
- La remontée des alertes est entièrement paramétrable grâce à l'utilisation de plugins (alerte par courrier électronique, SMS, etc…).

6.1.2. Inconvénients

Parmi ses inconvénients nous citons :

- Difficile à installer et à configurer
- Dispose d'une interface compliquée
- Ne permet pas d'ajouter des hosts via Web
- Besoin d'un autre outil comme CACTI pour faciliter sa configuration

- Pas de représentations graphiques
- Les mises à jour de la configuration se font en mode « lignes de commandes » et doivent être réalisées côté supervision comme côté serveur à superviser.

6.2. Zabbix

Zabbix est un outil de supervision, ambitionnant de concurrencer Nagios et MRTG. Il permet de superviser réseau, systèmes (processeur, disque, mémoire, processus,...). Zabbix offre des vues graphiques (générés par RRDtool) et des alertes sur seuil. Le «serveur ZABBIX» peut être décomposé en 3 parties séparées: Le serveur de données, l'interface de gestion et le serveur de traitement. Chacune d'elles peut être disposée sur une machine différente pour répartir la charge et optimiser les performances. Un agent ZABBIX peut aussi être installé sur les hôtes Linux, UNIX et Windows afin d'obtenir des statistiques de la charge CPU, l'utilisation du réseau, l'espace disque... Le logiciel peut réaliser le monitoring via SNMP. Fonctionnalité intéressante, il est possible de configurer des "proxy Zabbix" afin de répartir la charge ou d'assurer une meilleure disponibilité de service [12].

6.2.1. Avantages

Ses avantages sont les suivants :

- Richesse des sondes et tests possibles (supervision d'applications Web, par exemple).
- Réalisation de graphiques, cartes ou screens.
- Mise à jour de la configuration via l'interface Web de Zabbix.
- Serveur Proxy Zabbix. Surveillances des sites web: temps de réponse, vitesse de transfert.
- Ses agents sont assez légers (écrits en C).

6.2.2. Inconvénients

Et ses avantages sont :

- Interface est un peu vaste, la mise en place des templates n'est pas évidente au début : petit temps de formation nécessaire.
- L'agent Zabbix communique par défaut en clair les informations, nécessité de sécuriser ces données (via VPN par exemple).
- Peu d'interfaçage avec d'autres solutions commerciales.

6.3. Centreon (Anciennement Oreon)

Créé en 2003 par des français souhaitant améliorer Nagios, il a été repris par une nouvelle entreprise nommée Merethis il se présente comme une évolution de celui-ci pour tout d'abord son interface mais aussi ses fonctionnalités. Il s'appuie également sur les technologies Apache et PHP pour l'interface web, MySQL pour le stockage des données de configuration et de supervision [13].

6.3.1. Avantages

On en cite quelques avantages :

- Une installation complète et automatique des packages nécessaires à l'utilisation de NAGIOS.
- Facilite la configuration de Nagios.
- Une découverte automatique du réseau via NMAP.
- Graphe le résultat des alertes, système de reporting.

6.3.2. Inconvénients

Et quelques inconvénients :

- Requiert plus de ressources matérielles que Nagios.

6.4. Check_MK

L'agent Check_MK est un agent basé sur le système Nagios, il permet de le rendre plus intuitif et plus complet avec la supervision des services et l'ajout de sondes personnalisées. Cette collecte est possible par l'installation d'un plugin sur les clients (check-mk-agent), et d'un paquet sur le serveur (check-mk-server) pour superviser le parc. Ainsi, il est alors possible d'obtenir des informations relatives au CPU, à la RAM, sur le(s) HDD, les partitions, la quantité de Swap utilisée, le nombre de processus actifs et bien plus encore suivant les matériels supervisés [14].

6.4.1. Avantages

Ce logiciel a des avantages :

- Installation et configuration facile
- L'interface Web est beaucoup plus intuitive et qui intègre des outils, comme PNP4Nagios et RRDtool.
- L'interface permet une configuration entièrement graphique.
- Check_MK est capable de réaliser un inventaire automatique des services disponibles sur un hôte à superviser.

6.4.2. Inconvénients

Et un inconvénient :

- Offre plus de services sur l'environnement Unix

6.5. Interprétation des résultats

La Comparaison générale des outils de supervision internes a été étudiée avec un diagramme radar et un tableau comparatif :

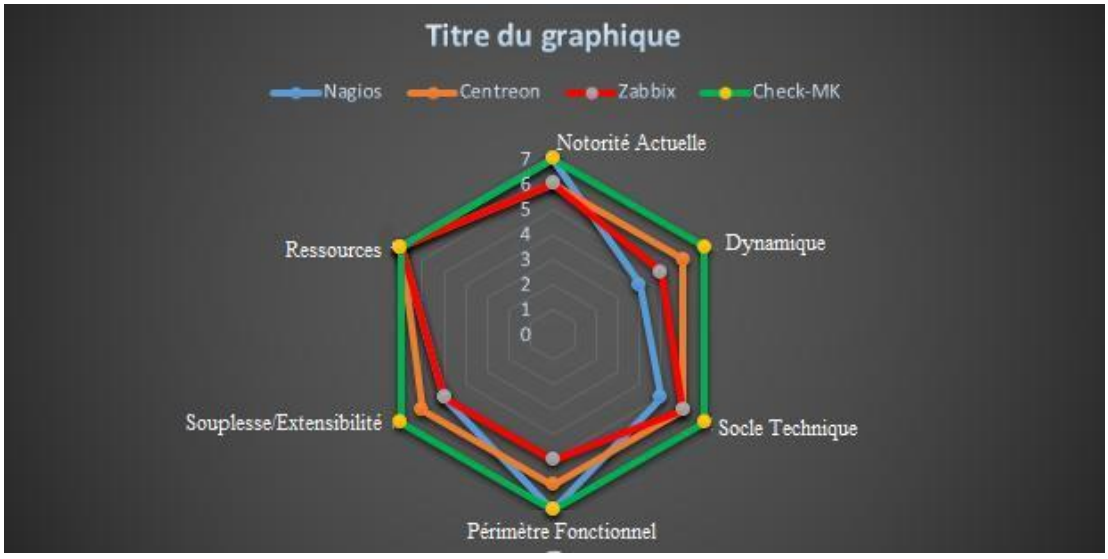


Figure 09: Diagramme radar [15].

Critères fonctionnels	Centreon	Nagios	Zabbix	Check_mk
Environnement d'installation	Unix	Unix	Unix	Unix
Basé sur	PHP	C++	PHP, C, C++	Python
Protocole	SNMP, SMTP,pop3,NNTP,ICMP,HLDP	SNMP, SMTP, pop3, NNTP, ICMP,HLDP	HTTP,FTP, SMTP,SSH, IMAP	SNMP
Gestion d'une authentification et des rôles	Oui	Oui	Oui	Oui
Création de graphes simple à partir des mesures	Oui	Non	Oui	Oui
Création de graphes complexes avec mise en relation des métriques des services supervisés	Non	Non	Oui	Oui
Utilisation d'agents sur les machines cibles	NRPE NSclient	NRPE NSclient	Agent Windows/ Unix	Check_MK Win
Installation et configuration simple	Oui	Non	Oui	Oui
Intégration simple d'un nouvel host dans un système de configuration centralisée	Oui	Non	Oui	Oui
Possibilité de mettre en place simplement une supervision centralisée entre plusieurs sous-réseaux ou sites	Oui	Non	Oui	Oui
Utilisation de RRD Tools	Oui	Non	Non	Oui

Tableau 01: Tableau comparatif [15].

Conclusion

Les logiciels de supervision réseau sont à la base du bon fonctionnement d'une architecture réseau et permet de réagir rapidement en cas de problèmes ou pannes. Qu'elles soient Open Source ou propriétaires, chacune a ses particularités qui lui sont propres.

Dans le chapitre suivant, nous allons présenter l'outil choisi et décrire son mode de fonctionnement.

Introduction

Dans ce chapitre, nous commençons par expliquer la raison pour laquelle nous avons choisi Nagios comme outil de supervision pour notre projet, ensuite présenter ce dernier, son architecture et son principe de fonctionnement, puis nous présentons les compléments de notre solution qui sont les agents spécialisés en supervision à distance NSCA et NRPE ainsi que ses fichiers de configuration.

1. Pourquoi utiliser Nagios

Comme vu dans le tableau comparatif du chapitre précèdent, l'outil qui répond aux critères fonctionnels dans le choix d'une solution open source stable, performante et ayant une forte communauté, Nagios sort largement vainqueur. Cette solution est en effet la référence en matière de supervision dans le monde de l'open source.

2. Présentation de Nagios

Nagios est anciennement connu sous le nom de Netsaint. Plus exactement, Nagios est une évolution de Netsaint auquel a été ajoutée, entre autres, la gestion du protocole SNMP.

Nagios est un outil développé principalement par Ethan Galstad, très connu dans le monde de l'entreprise et des professionnels réseaux, il est conçu pour fonctionner sous un système d'exploitation Linux (compatible UNIX). Cet outil propose de superviser les machines et les services d'un réseau via des plugins indépendants, chacun responsable d'un test particulier et vous alertent lorsque les systèmes vont mal et quand ils vont mieux [16].

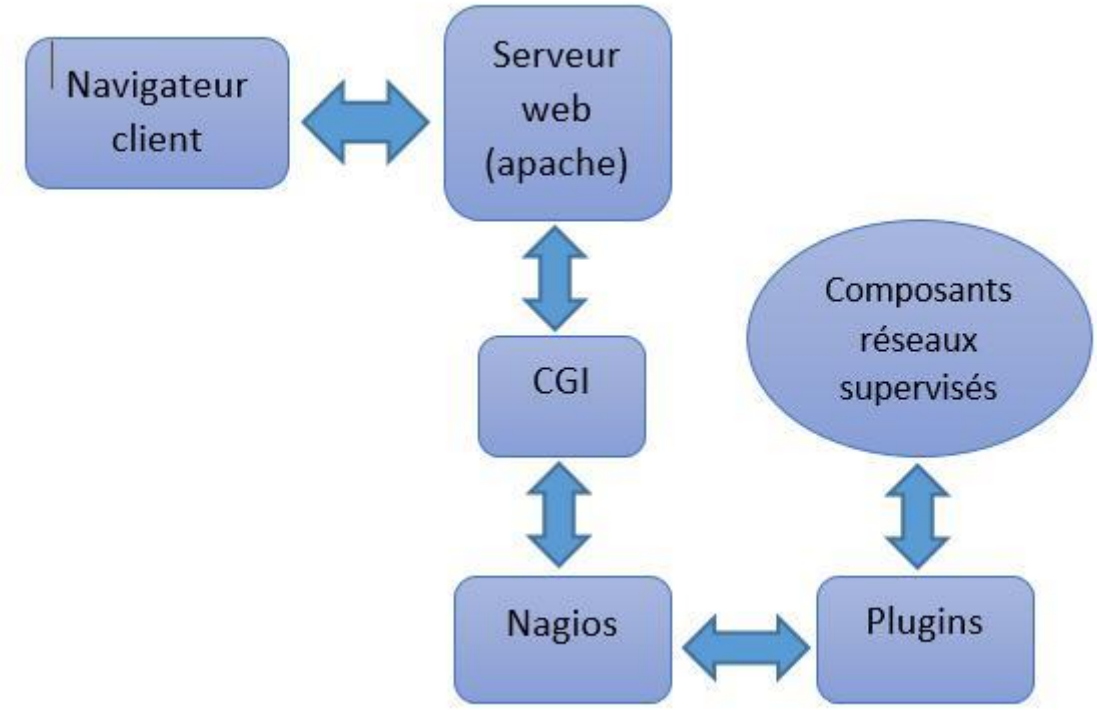
Quelques-unes des fonctionnalités incluses dans Nagios :

- Surveillance des services réseaux (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Surveillance des ressources des hôtes (charge processeur, utilisation des disques, etc.)
- Système simple de plugins permettant aux utilisateurs de développer facilement leurs propres vérifications de services.
- Notifications des contacts quand un hôte ou un service a un problème et quand celui-ci est résolu (via email, pager, sms, ou par tout autre méthode définie par l'utilisateur)
- Possibilité de définir des gestionnaires d'événements qui s'exécutent pour des évènements sur des hôtes ou des services, pour une résolution proactive des problèmes
- Support pour l'implémentation d'un système de surveillance redondant [17].

3. 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** (Pages Web accessible par Navigateur) qui affiche de manière claire et concise l'état des services surveillés.
- **Des greffons (plugins) :** Un greffon est un programme exécutable ou script (perl, shell, etc.) [18].



- **Figure 10 : Architecture de Nagios [18].**

4. Mode de fonctionnement

Nagios ne possède aucun mécanisme interne pour surveiller le statut des équipements et des applications et collecter les informations. Afin que Nagios réponde parfaitement aux attentes prévues, il doit reposer sur des programmes externes appelés des plugins pour superviser les machines connectées sur le réseau. Nagios est considéré comme un ordonnanceur de tâches et d'événements.

Son principe de fonctionnement est assez simple : à chaque instant donné, Nagios exécute les plugins pour collecter des informations sur les états des machines. Un plugin est généralement un script (Shell, PHP, Perl, etc.) ou un programme et il peut être exécuté localement dans le serveur Nagios ou à distance dans les machines clientes à l'aide des agents. Le retour de l'exécution d'un plugin est utilisé par Nagios pour connaître l'état de la machine contrôlée. La configuration de Nagios est basée sur des objets. Les principaux objets sont : l'hôte, le service et le contact. Un hôte est la machine à superviser connectée au réseau. Le service est l'élément à superviser sur la machine. En cas d'alerte, Nagios informe les responsables qui sont représentés par les objets contacts. La configuration de Nagios est sauvegardée dans des fichiers plats.

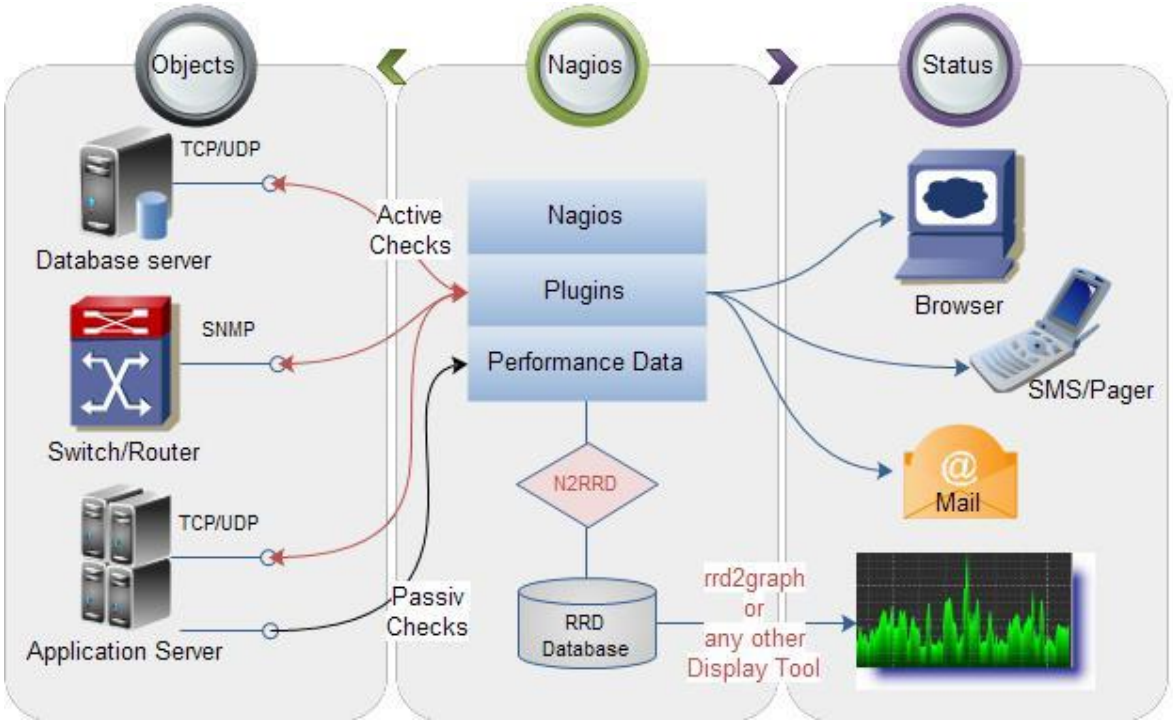


Figure 11 : Principe de fonctionnement de Nagios [19]

5. Les plugins

La principale force de Nagios est sa grande modularité qui lui permet de s'adapter aux besoins des utilisateurs. Il est ainsi possible de surveiller un grand nombre de paramètres sur les machines de votre réseau. Nous allons dans ce billet évoquer les différentes méthodes que l'on peut utiliser pour récupérer ces informations.

5.1. Les plugins locaux

En standard, SNMP ne remonte que des informations systèmes basiques. Pour aller plus loin et surveiller des processus plus complexe, Nagios a mis en place un système de type plugins locaux. Un plugin local est un script localisé sur le serveur Nagios (/usr/lib/nagios/plugins sous Linux, c'est pour cela que l'on dit qu'il est local). Ce script, lancé à la demande de Nagios, doit retourner un code dont la signification est la suivante:

- Code 0: OK - Tout va bien
- Code 1: WARNING - Alerte
- Code 2: CRITICAL - Alerte critique
- Code 3: UNKNOWN - Problème lors de l'exécution du plugin

En plus de ces codes, un plugin peut fournir d'autres informations (sous la forme d'une chaîne de caractères) qui seront affichées à côté du statut de la machine [20].

5.2. Les principaux plug-ins

- ➔ check_disk : Vérifie l'espace occupé d'un disque dur
- ➔ check_http : Vérifie le service "http" d'un hôte
- ➔ check_ftp : Vérifie le service "ftp" d'un hôte
- ➔ check_mysql : Vérifie l'état d'une base de données MYSQL
- ➔ check_nt : Vérifie différentes informations (disque dur, processeur …) sur un système d'exploitation Windows
- ➔ check_nrpe: Permet de récupérer différentes informations sur les hôtes
- ➔ check_ping: Vérifie la présence d'un équipement, ainsi que sa durée de réponse
- ➔ check_pop: Vérifie l'état d'un service POP (serveur mail)
- ➔ check_snmp : Récupère divers informations sur un équipement grâce au protocole SNMP [21].

6. La supervision active et passive

Nagios peut utiliser différentes méthodes dans le but de récolter les informations sur les machines du réseau. Une méthode dite active, et une autre passive. Les deux se basent sur l'exécution d'un processus démon (daemon) sur la machine à surveiller.

Ces deux méthodes se combinent généralement pour une efficacité optimale de la supervision. Nous rappelons que dans tout système d'exploitation multitâche, un démon est un programme informatique qui s'exécute en arrière-plan, et non sous le contrôle direct d'un utilisateur.

6.1. Les plugins actifs avec NRPE

A la différence des plugins locaux (ceux qui s'exécute sur la machine serveur, concernant ses propres ressources), le module/démon NRPE (Nagios Remote Plugin Executor) permet l'exécution de plugins dit actifs directement sur les machines à surveiller. Dans ce cas, la demande d'exécution du greffon actif est faite à l'initiative de la machine serveur Nagios.

La procédure interne, comme illustrée par la figure 12, est la suivante: Le serveur Nagios demande, via le client NRPE, l'exécution du plugin P sur la machine M. Le daemon NRPE hébergé sur la machine M, reçoit la requête d'exécution du plugin P. Ensuite l'exécution de ce plugin sur la machine M. Le daemon NRPE de la machine M récolte les informations suit à l'exécution du greffon P et envoi le résultat au serveur Nagios. Enfin le serveur Nagios interprète les résultats et lance le traitement adéquat.

Ce type de procédure permet d'assurer une surveillance distante. Il faudra toutefois ouvrir un port de communication pour permettre au NRPE de communiquer avec son client et récupérer les informations d'état concernant les machines déportées.

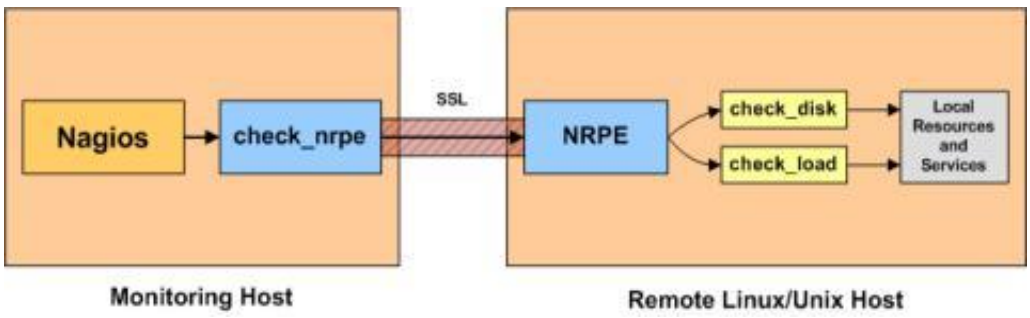


Figure 12: La supervision active [22]

NRPE est déclenché à l'initiative du serveur Nagios. Ce mode de fonctionnement présente des limites. Par exemple dans le cas où les machines à surveiller son derrière un réseau sécurisé, NRPE ne permet que les connexions sortantes de celui-ci, ou encore si le processus à surveiller demande une fréquence d'exécution très courte. L'échange des informations n'est plus assuré. Dans ce cas nous avons recours aux greffons dits passifs [23].

6.2. Les plugins passifs avec NSCA

Le module NSCA propose l'exécution de plugins passifs sur les machines à surveiller. Leur exécution est déclenchée suite à des critères préalablement définis sur les machines distantes. Par exemple, le dépassement de 75% de la capacité de stockage, la détection d'une activité réseau anormale ou simplement des checks périodiques sous forme de mises à jour auto déclenchées.

La procédure interne est la suivante : Le daemon NSCA sur une machine M lance l'exécution du plugin P suite à un critère de déclenchement vérifié. En effet le plugin P est exécuté sur la machine M. Le daemon NSCA de la machine M récolte les informations suite à l'exécution du greffon P et envoi le résultat au serveur Nagios. Enfin le serveur Nagios interprète les résultats et lance le traitement adéquat.

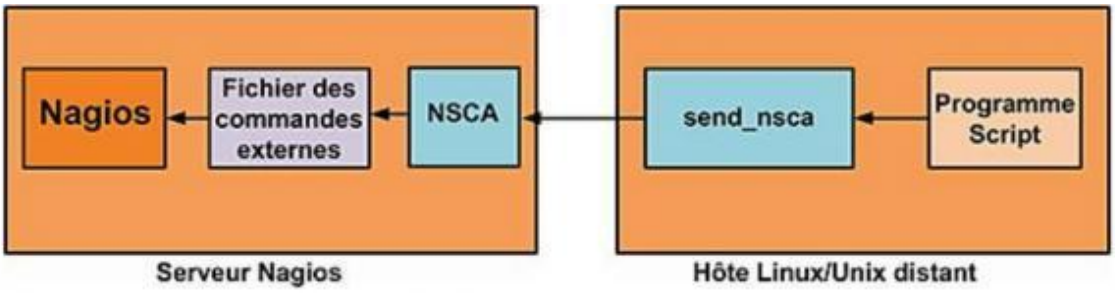


Figure 13: La supervision passive [24]

Nous remarquons que la demande d'exécution du greffon est faite non pas par l'initiative de la machine serveur Nagios, mais par l'initiative de la machine distance elle-même (la machine à superviser) suite à un critère de déclenchement vérifié [23].

7. Les fichiers de configuration

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

✓ **Nagios.cfg** est le fichier de configuration principal de Nagios. Il contient la liste des autres fichiers de configuration et comprend l'ensemble des directives globales de fonctionnement.

✓ **Cgi.cfg** contient un certain nombre de directives qui affectent le mode de fonctionnement des CGI. Il peut être intéressant pour définir des préférences concernant l'interface web de Nagios.

✓ **Resource.cfg** permet de définir des variables globales réutilisables dans les autres fichiers. Étant inaccessible depuis les CGI qui génèrent l'interface, ce fichier peut être utilisé pour stocker des informations sensibles de configuration.

✓ **Commands.cfg** contient les définitions des commandes externes, telles que celles qui seront utiles pour la remontée d'alerte.

✓ **Checkcommands.cfg** contient les définitions des commandes de vérification prédéfinies et celles définies par l'utilisateur.

✓ **Hosts.cfg** définit les différents hôtes du réseau à superviser. A chaque hôte est associé son nom, son adresse IP, le test à effectuer par défaut pour caractériser l'état de l'hôte, etc.

✓ **Services.cfg** associe à chaque hôte ou à chaque groupe d'hôtes l'ensemble des services qui doivent être vérifiés.

✓ **Hostgroups.cfg** définit des groupes d'hôtes pour regrouper des hôtes selon des caractéristiques communes. Un hôte peut appartenir à plusieurs groupes.

✓ **Contacts.cfg** déclare les contacts à prévenir en cas d'incident et définit les paramètres des alertes (fréquences des notifications, moyens pour contacter ces personnes, plages horaires d'envoi des alertes...) [25].

Conclusion

Après avoir présenté l'outil de supervision à utiliser, expliqué et étudié son fonctionnement, nous passerons dans le chapitre suivant à sa mise en place au sein de l'entreprise.

Introduction

A travers ce chapitre, nous allons décrire la phase de réalisation de notre application. Nous allons commencer par la spécification des différents environnements de développement, matériels et logiciels. Ensuite nous décrirons les points les plus intéressants de l'application, tout en donnant un aperçu sur les différentes parties développées au cours de ce projet.

1. Environnement de travail

1.1. Environnement matériel

Dans notre projet, nous avons utilisé un ordinateur portable modèle Toshiba Satellite C55-B1325 avec la configuration suivante :

- Intel(R) Pentium(R) CPU N3520.
- 4Go de RAM
- Disque dur de capacité 146 Go.
- Système d'exploitation Ubuntu 16.04 LTS pour la mise en œuvre du logiciel Nagios.

1.2. Environnement logiciel

Au niveau de l'installation nous avons choisi les versions de 2018 compatibles avec le système Ubuntu 16.04 LTS pour fournir une meilleure interprétation sur des équipements à superviser. Les versions utilisées sont listées comme suit :

- L'outil de supervision Nagios 4.4.2.
- Les plugins de Nagios, nagios-plugins-release-2.2.1.
- L'agent NSClient++ pour la supervision des machines Windows, nscp-0.5.2.35.

2. Installation et configuration de Nagios

Afin d'installer Nagios sur notre machine, nous avons tout d'abord mis à jour notre serveur Ubuntu et installé les packages nécessaires, puis installé Apache et PHP 7, ensuite nous avons téléchargé et installé Nagios et ses plugins, enfin nous avons démontré la façon dont nous avons accède à l'interface Nagios.

2.1. Mise à jour du serveur Ubuntu 16.04 et installation des packages nécessaires

Nous avons commencé par exécuter les commandes suivantes pour mettre à jour la liste des packages et mettre à niveau tous les packages installés:

---

sudo apt update

sudo apt upgrade

Ensuite, nous avons installé des packages pré requis nécessaires à la construction de Nagios et ses plugins avec la commande suivante:

---

sudo apt install autoconf gcc libc6 make wget unzip libgd2-xpm-dev

---

2.2. Installation de Apache et PHP 7

Pour installer Apache, PHP 7 et tous les modules nécessaires sur notre serveur Ubuntu 16.04, nous avons exécuté la commande suivante :

---

sudo apt install apache2 php libapache2-mod-php7.0

Une fois l'installation d'Apache est terminée, exécuter la commande suivante pour permettre à Apache de démarrer au démarrage :

---

sudo systemctl enable apache2.service

2.3. Téléchargement et installation de Nagios

Télécharger la dernière version stable de l'archive tar de Nagios dans le répertoire / **tmp** du serveur en exécutant la commande wget suivante :

```
Wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.4.2/nagios-4.4.2.tar.gz -O /tmp/nagioscore.tar.gz
```

Une fois le téléchargement terminé, accéder au répertoire / **tmp**, extraire l'archive en exécutant la commande suivante:

```
cd /tmp
sudo tar xf nagioscore.tar.gz
```

Une fois l'archive Nagios Plugins extraite, accédez au répertoire **nagios-4.4.2**:

```
cd /tmp/nagios-4.4.2
```

L'étape suivante consiste à exécuter le script **./configure** qui vérifiera le système pour les bibliothèques et binaires manquants et préparera le code source de Nagios pour le processus de construction :

```
sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

La sortie ressemblera à ce qui suit:

```
*** Configuration summary for nagios 4.4.2 2018-08-16 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagios
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  /run/nagios.lock
   Check result directory:  /usr/local/nagios/var/spool/checkresults
           Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/apache2/sites-enabled
             Mail program:  /usr/bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                HTML URL:  http://localhost/nagios/
                 CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):  /usr/sbin/traceroute


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

Maintenant que la configuration est terminée, nous exécutons le processus de compilation à l'aide de la commande **make all**

---

make all

Une fois terminée, la commande imprimera la sortie suivante:

```
*** Support Notes *****************************************

If you have questions about configuring or running Nagios,
please make sure that you:

     - Look at the sample config files
     - Read the documentation on the Nagios Library at:
          library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       support.nagios.com

************************************************************

Enjoy.
```

Exécuter la commande suivante pour créer l'utilisateur et le groupe Nagios et ajouter Apache au groupe Nagios:

---

sudo make install-groups-users

sudo usermod -a -G nagios www-data

L'étape suivante consiste à installer Nagios à l'aide de la commande make install:

---

sudo make install

Exécuter la commande suivante pour installer les exemples de fichiers de configuration Nagios:

```
sudo make install-config
```

Installer le script d'initialisation afin de pouvoir gérer le service Nagios à l'aide de la commande **systemctl:**

```
sudo make install-daemoninit
```

Ensuite, installer les fichiers de configuration du serveur Web Apache avec:

```
sudo make install-webconf
```

Redémarrer le service Apache:

```
sudo systemctl restart apache2
```

Exécuter la commande suivante pour créer un compte utilisateur appelé **nagiosadmin:**

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

Exécuter ensuite la commande suivante pour installer et configurer le répertoire de commande externe:

```
sudo make install-commandmode
```

2.4. Installation des plugins Nagios

Installer les dépendances suivantes nécessaires pour télécharger et compiler les plugins Nagios:

---

sudo apt install autoconf gcc libc6 libmcrypt-dev make libssl-dev wget bc gawk dc build-essential snmp libnet-snmp-perl gettext

---

Télécharger la dernière version stable de l'archive tar de Nagios dans le répertoire **/ tmp** de votre serveur avec:

---

wget --no-check-certificate -O /tmp/nagios-plugins.tar.gz https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz

---

Une fois le téléchargement terminé, accédez au répertoire **tmp**, on extrait l'archive avec les commandes suivantes:

---

cd /tmp

sudo tar xf nagios-plugins.tar.gz

Une fois l'archive Nagios Plugins extraite, accéder au répertoire **nagios-plugins-release-2.2.1**:

---

cd /tmp/nagios-plugins-release-2.2.1

Les commandes suivantes vérifieront système pour les bibliothèques et binaires manquants et prépareront le code source des plugins Nagios pour le processus de construction:

```
./tools/setup

./configure
```

Une fois la configuration terminée, nous lançons le processus de compilation à l'aide de la commande **make**:

```
make
```

Installer les plugins Nagios à l'aide de la commande **make install**:

```
sudo make install
```

2.5. L'accès à Nagios

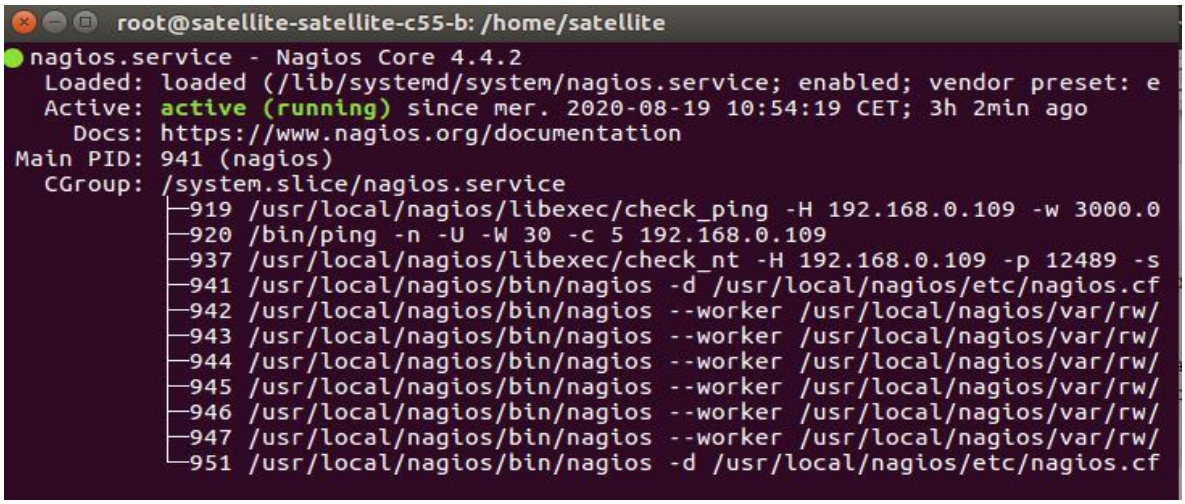
Maintenant que les plugins Nagios et Nagios sont installés sur notre système Ubuntu, nous démarrons le service Nagios avec la commande suivante :

```
systemctl start nagios
```

Vérifier l'état du service avec:

```
systemctl status nagios
```

La sortie ressemblera à quelque chose comme ci-dessous:



Il ne reste plus que d'accéder à “http: // votre-domaine-ou-ip/nagios” via le navigateur pour avoir la fenêtre de connexion, et entrer les informations que nous avons défini précédemment avec la commande htpasswd comme indiqué sur l’image ci-dessous :

Figure 14 : Fenêtre de Connexion

Après l'authentifier nous accédons à la page d'accueil de Nagios comme indiqué sur l'image ci-dessous :

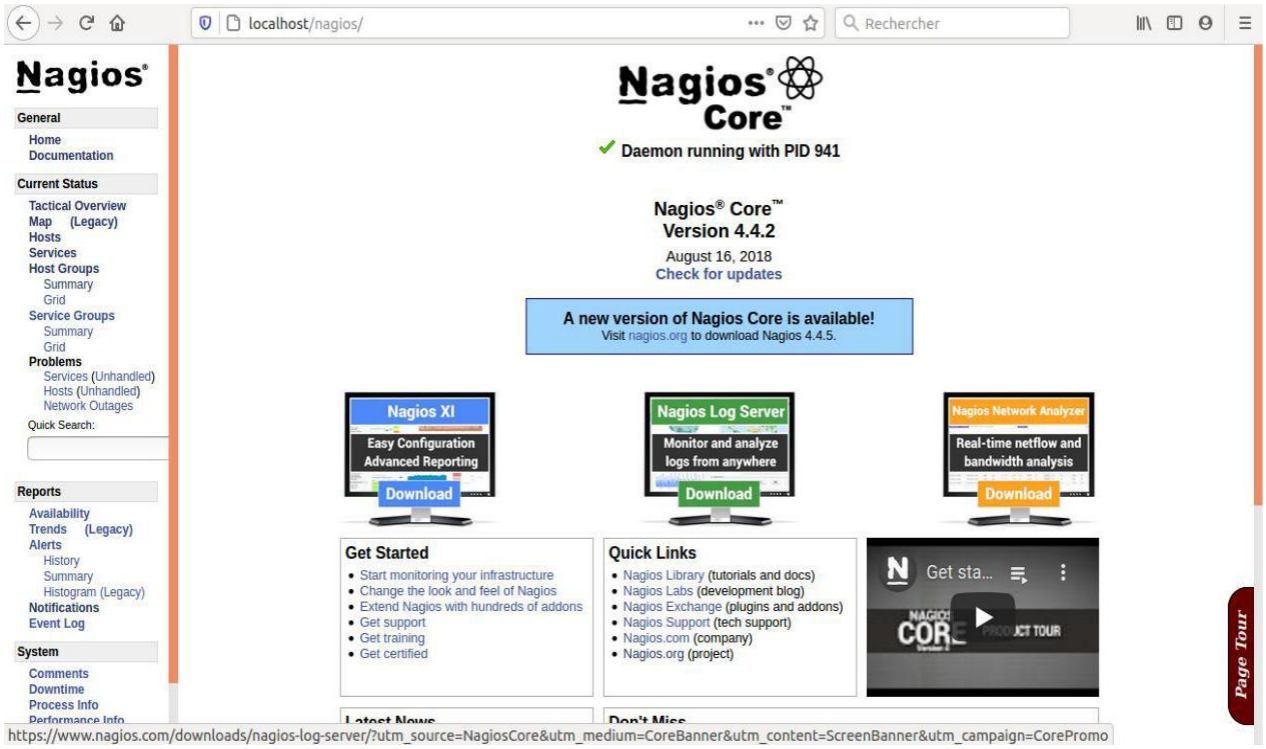


Figure 15 : Page d'accueil Nagios

3. Machines Windows

Pour pouvoir superviser une machine Windows, nous avons commencé par installer un agent Windows dans chaque machine à superviser, ensuite nous avons configuré Nagios.

3.1. Installation de l'agent Windows

Avant de pouvoir superviser les attributs et services privés des machines Windows, nous allons devoir installer un agent sur ces machines. Nous avons choisi pour cela l'addon NSClient++.

Les étapes d'installation de NSClient++ et de configuration de Nagios sont comme suit :

1) Télécharger la dernière version de NSClient++ selon le système d'exploitation depuis https://nsclient.org/download/
2) Installer NSClient++

Nous configurons NSClient++ lors de son installation comme indiqué dans la figure suivante :

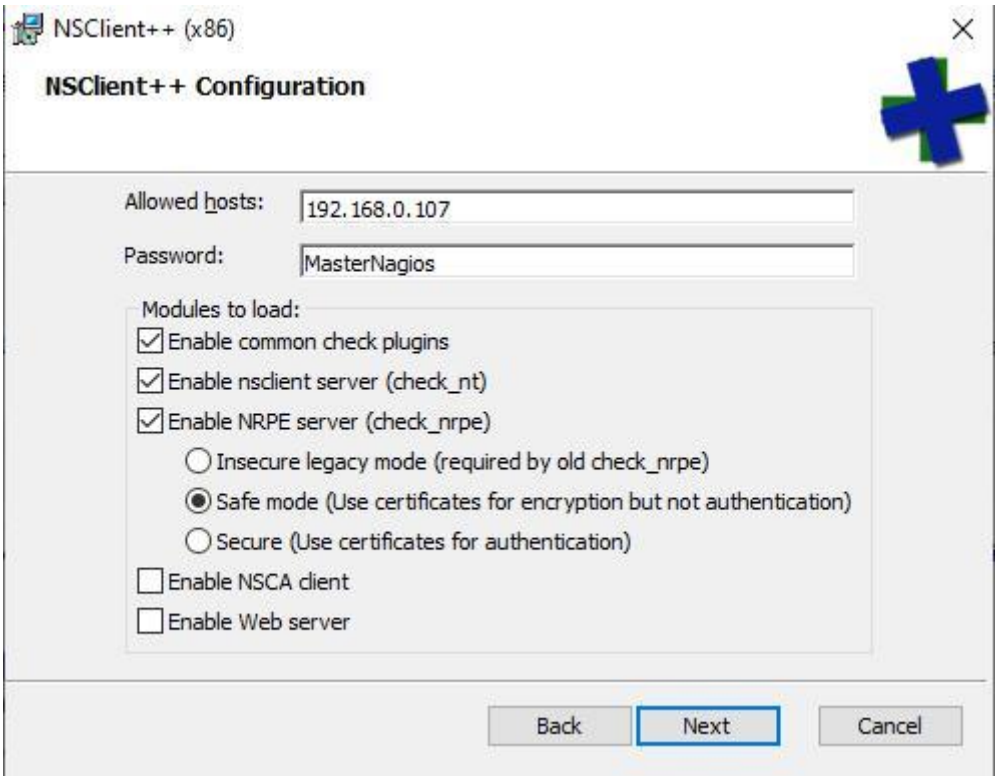


Figure 16 : Fenêtre de configuration NSClient++

3.2. Configuration de Nagios

1) Dans le fichier /usr/local/nagios/etc/nagios.cfg décommenter la ligne

---

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

---

2) Modifier le fichier **windows.cfg**

---

nano /usr/local/nagios/etc/objects/windows.cfg

---

3) Ajouter une nouvelle définition d'hôte pour la machine Windows que vous souhaitez superviser :

```
#LES PC DE L'OPGI

#PC 1
define host {

    use                     windows-server          ; Inherit default values from a template
    host_name               Pc_IT               ; The name we're giving to this host
    alias                   PC1       ; A longer name associated with the host
    address                 192.168.0.105             ; IP address of the host
}
```

Figure 17 : Déclaration d'un serveur Windows

4) Ajoutez la définition des services suivants:

- pour contrôler la version de l'addon NSClient++ tournant sur le serveur Windows.

```
###############################################################################
#
# SERVICE DEFINITIONS
#
###############################################################################

# Create a service for monitoring the version of NSClient++ that is installed
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     NSClient++ Version
    check_command           check_nt!CLIENTVERSION
}
```

Figure 18 : Définition du service NSClient++ Version

- Pour superviser le temps écoulé depuis le dernier re/démarrage du serveur Windows.

```
# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     Uptime
    check_command           check_nt!UPTIME
}
```

Figure 19 : Définition du service Uptime

- Pour superviser la charge CPU du serveur Windows et générer une alerte CRITICAL si la charge CPU des 5 dernières minutes est égale à 90% ou plus ou une alerte WARNING si la charge CPU des 5 dernières minutes est égale à 80% ou plus.

```
# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     CPU Load
    check_command           check_nt!CPULOAD!-l 5,80,90
}
```

Figure 20 : Définition du service CPU load

- Pour superviser l'utilisation de la mémoire du serveur Windows et générer une alerte CRITICAL si l'utilisation de la mémoire est égale à 90% ou plus ou une alerte WARNING si l'utilisation de la mémoire est égale à 80% ou plus.

```
# Create a service for monitoring memory usage
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     Memory Usage
    check_command           check_nt!MEMUSE!-w 80 -c 90
}
```

Figure 21 : Définition du service Memory Usage

- Pour superviser l'espace utilisé du disque C:\ du serveur Windows et générer une alerte CRITICAL si l'espace utilisé du disque est égale à 90% ou plus ou une alerte WARNING si l'espace utilisé du disque est égale à 80% ou plus.

```
# Create a service for monitoring C:\ disk usage
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     C:\ Drive Space
    check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90
}
```

Figure 22 : Définition du service C:\ Drive Space

- Ajoutez la définition de service suivante pour superviser l'état du service W3SVC et générer une alerte CRITICAL si ce service est arrêté.

```
# Create a service for monitoring the W3SVC service
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     W3SVC
    check_command           check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
}
```

Figure 23 : Définition du service W3SVC

- Ajoutez la définition de service suivante pour superviser l'état du processus Explorer.exe et générer une alerte CRITICAL si ce processus ne tourne pas.

```
# Create a service for monitoring the Explorer.exe process
# Change the host_name to match the name of the host you defined above

define service {

    use                     generic-service
    host_name               PcYasmine,Pc_IT,Pc_info
    service_description     Explorer
    check_command           check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
}
```

Figure 24 : Définition du service Explorer

5) Protection par mot de passe

Dans le fichier /usr/local/nagios/etc/commands.cfg, modifier la définition de la commande **check_nt** pour inclure l'argument *-s <PASSWORD>*

```
define command {

    command_name    check_nt
    command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s MasterNagios -v $ARG1$ $ARG2$
}
```

Figure 25 : Définition de la commande chech_nt

6) Redémarrage de Nagios

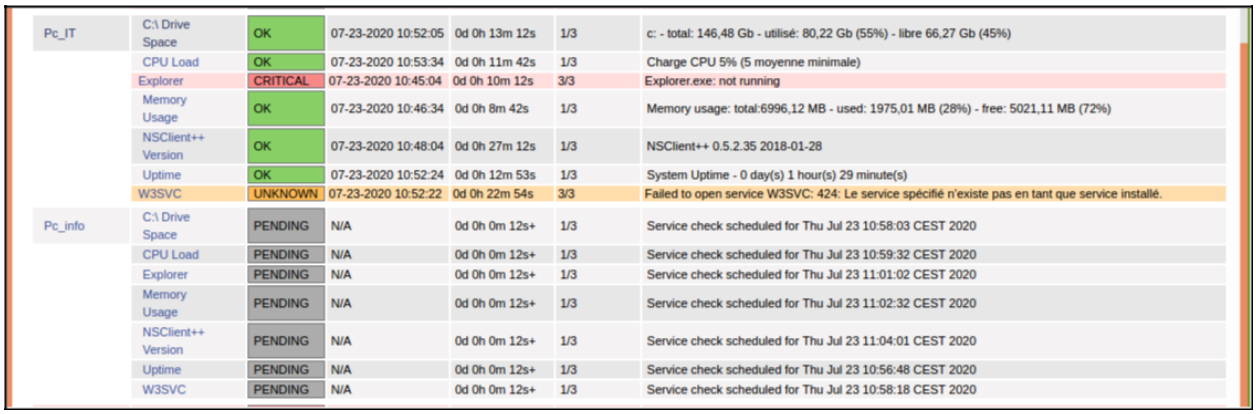
Redémarrer Nagios avec la commande suivante :

---

service nagios restart

3.3. Les tests

Nous laissons le programme faire la collecte et le traitement des données pendant quelques minutes avant d'actualiser l'interface de Nagios.



Pc_IT	C:\ Drive Space	OK	07-23-2020 10:52:05	0d 0h 13m 12s	1/3	c: - total: 146,48 Gb - utilisé: 80,22 Gb (55%) - libre 66,27 Gb (45%)
	CPU Load	OK	07-23-2020 10:53:34	0d 0h 11m 42s	1/3	Charge CPU 5% (5 moyenne minimale)
	Explorer	CRITICAL	07-23-2020 10:45:04	0d 0h 10m 12s	3/3	Explorer.exe: not running
	Memory Usage	OK	07-23-2020 10:46:34	0d 0h 8m 42s	1/3	Memory usage: total:6996,12 MB - used: 1975,01 MB (28%) - free: 5021,11 MB (72%)
	NSClient++ Version	OK	07-23-2020 10:48:04	0d 0h 27m 12s	1/3	NSClient++ 0.5.2.35 2018-01-28
	Uptime	OK	07-23-2020 10:52:24	0d 0h 12m 53s	1/3	System Uptime - 0 day(s) 1 hour(s) 29 minute(s)
	W3SVC	UNKNOWN	07-23-2020 10:52:22	0d 0h 22m 54s	3/3	Failed to open service W3SVC: 424: Le service spécifié n'existe pas en tant que service installé.
Pc_info	C:\ Drive Space	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 10:58:03 CEST 2020
	CPU Load	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 10:59:32 CEST 2020
	Explorer	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 11:01:02 CEST 2020
	Memory Usage	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 11:02:32 CEST 2020
	NSClient++ Version	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 11:04:01 CEST 2020
	Uptime	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 10:56:48 CEST 2020
	W3SVC	PENDING	N/A	0d 0h 0m 12s+	1/3	Service check scheduled for Thu Jul 23 10:58:18 CEST 2020

Figure 26 : Résultat du check des Pc_IT et PC_info

La figure 26 illustre les états des services pour chaque hôte:

- Pc_IT : après avoir collecté les données :
  - ➔ L'état du l'espace disque est : OK. En fait pour un disque de 146,48 Gb, 80,22 Gb sont utilisés représentant 55% de la taille totale du disque et 66,27 Gb sont libres ce qui est équivalant à 45%.
  - ➔ - L'état de la charge du processeur est : OK.
  - ➔ - L'état du service explorer est critique en fait il n'est plus démarré.
  - ➔ - L'état de l'utilisation de la mémoire est : ok. En fait la taille totale de la mémoire est 6996,12 Gb. Seulement 28% ce cette mémoire est utilisé.
  - ➔ L'état du service UPTIME est OK. Le serveur est actif depuis 1 heure et 29 minutes.
  - ➔ L'état de W3SVC est alarmant. Car ce service n'est pas installé sur cette machine.
- Pc_info : en attente d'avoir les données demandées.

4. Configuration d'une imprimante

Nous commençons par éditer le fichier de configuration principal de Nagios "**/usr/local/nagios/etc/nagios.cfg"** en décommentant la commande suivante :

#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

Puis nous éditons le fichier "**/usr/local/nagios/etc/objects/printer.cfg"** afin d'ajouter les définitions des hôtes pour les imprimantes réseau que nous souhaitons superviser et leurs services associés :

```
##############################################################################
#
# HOST DEFINITIONS
#
##############################################################################

# Define a host for the printer we'll be monitoring
# Change the host_name, alias, and address to fit your situation

define host {

    use                     generic-printer         ; Inherit default values from a template
    host_name               Bro_J4120               ; The name we're giving to this printer
    alias                   Brother_DCP_J4120DW      ; A longer name associated with the printer
    address                 192.168.0.105           ; IP address of the printer
    hostgroups              network-printers        ; Host groups this printer is associated with
}
```

Figure 27 : Définition d'une l'imprimante IP

Nous avons ajouté deux définitions de service; une pour pouvoir superviser l'état de l'imprimante; Le service utilise le plugin **check_hpjd** pour vérifier l'état de l'imprimante toutes les 10 minutes par défaut, plus la définition de service permettant de pinger l'imprimante toutes les 10 minutes par défaut; C'est utile pour superviser le RTA, les paquets perdus et la connectivité réseau :

```
###############################################################################
#
# SERVICE DEFINITIONS
#
###############################################################################

# Create a service for monitoring the status of the printer
# Change the host_name to match the name of the host you defined above
# If the printer has an SNMP community string other than "public",
# change the check_command directive to reflect that

define service {

    use                     generic-service         ; Inherit values from a template
    host_name               Bro_J4120              ; The name of the host the service is associated with
    service_description     Printer Status          ; The service description
    check_command           check_hpjd!-C public    ; The command used to monitor the service
    check_interval          10                      ; Check the service every 10 minutes under normal conditions
    retry_interval          1                       ; Re-check the service every minute until its final/hard state is determined
}



# Create a service for "pinging" the printer occasionally.
# Useful for monitoring RTA, packet loss, etc.

define service {

    use                     generic-service
    host_name               Bro_J4120
    service_description     PING
    check_command           check_ping!3000.0,80%!5000.0,100%
    check_interval          10
    retry_interval          1
}
```

Figure 28 : Définition des services de l'imprimante IP

Une fois que nous avons ajouté les définitions d'hôte et de service au fichier printer.cfg, nous somme prêt à commencer la supervision de l'imprimante et pour cela nous aurons besoin de redémarrer Nagios:

---

service nagios restart

4.1. Le test

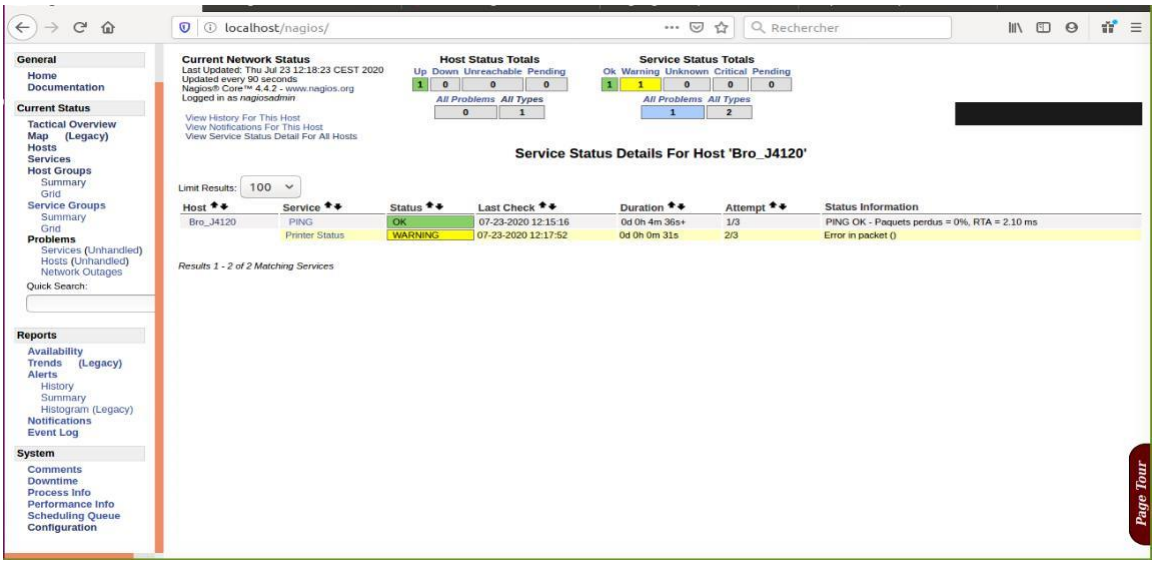


Figure 29 : Résultat du check de l'imprimante IP

5. Configuration d'un switch

Les switchs et les routeurs peuvent facilement être supervisés en les pingant pour déterminer le nombre de paquets perdus, RTA, etc. Si votre switch supporte SNMP, vous pouvez superviser l'état des ports, etc. avec le plugin **check_snmp** et la bande passante avec **check_mrtgtraf** plugin (en utilisant MRTG).

Le plugin **check_snmp** sera compilé et installé seulement si vous avez les paquets net-snmp et net-snmp-utils installés sur votre système.

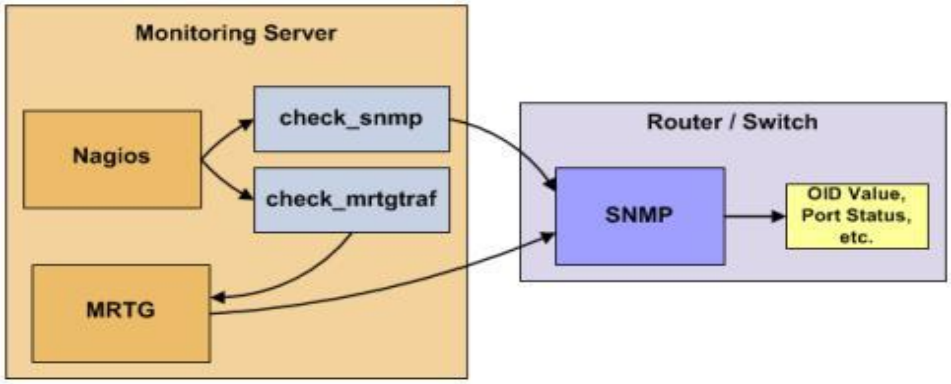


Figure 30: Vue globale sur la supervision d'un routeur/switch [28]

5.1. Configuration de Nagios

Supprimer le caractère (#) du début de la ligne suivante du fichier de configuration principal **nagios.cfg** :

#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

Dans le fichier switch.cfg Ajouter une nouvelle définition d'hôte pour le switch à superviser :

```
##############################################################################
#
# HOST DEFINITIONS
#
##############################################################################

# Define the switch that we'll be monitoring

define host {

    use                     generic-switch                  ; Inherit default values from a template
    host_name               switch1                 ; The name we're giving to this switch
    alias                   switch_informatique             ; A longer name associated with the switch
    address                 192.168.0.1                      ; IP address of the switch
    hostgroups              switches                        ; Host groups this switch is associated with
}
```

Figure 31 : Définition du switch

Nous ajoutons maintenant les services pour superviser les différents aspects du switch :

- Supervision des paquets perdus et de la RTA: pour pouvoir superviser les paquets perdus et le temps moyen de réponse entre le serveur Nagios et le switch toutes les 5 minutes en conditions normales.

```
###############################################################################
#
# SERVICE DEFINITIONS
#
###############################################################################

# Create a service to PING to switch

define service {

    use                     generic-service                 ; Inherit values from a template
    host_name               switch1,S2                      ; The name of the host the service is associated with
    service_description     PING                            ; The service description
    check_command           check_ping!200.0,20%!600.0,60%  ; The command used to monitor the service
    check_interval          5                               ; Check the service every 5 minutes under normal conditions
    retry_interval          1                               ; Re-check the service every minute until its final/hard state is determined
}
```

Figure 32 : Définition du service Ping

Supervision de l'information d'état SNMP: Pour les deux services suivant le switch ou routeur doit supporter SNMP

- Ajoutez la définition de service suivante pour superviser le temps écoulé depuis la mise sous tension de votre switch.

```
# Monitor uptime via SNMP

define service {

    use                     generic-service                 ; Inherit values from a template
    host_name               switch1,S2
    service_description     Uptime
    check_command           check_snmp!-C public -o sysUpTime.0
}
```

Figure 33 : Définition du service Uptime

- Ajouter la définition de service pour Vérifier si un port/interface particulier du switch est dans un état up.

```
# Monitor Port 1 status via SNMP

define service {

    use                     generic-service                 ; Inherit values from a template
    host_name               switch1,S2
    service_description     Port 1 Link Status
    check_command           check_snmp!-C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB
}
```

Figure 34 : Définition du service Port 1 Link Status

- Supervision de la bande passante/trafic

Pour superviser l'usage de la bande passante des switchs et routeurs nous utilisons MRTG, où sont stockées les données ainsi que les seuils.

```
# Monitor bandwidth via MRTG logs

define service {

    use                     generic-service                 ; Inherit values from a template
    host_name               switch1,S2
    service_description     Port 1 Bandwidth Usage
    check_command           check_local_mrtgtraf!/var/lib/mrtg/192.168.1.253_1.log!AVG!1000000,1000000!5000000,5000000!10
}
```

Figure 35 : Définition du service Port 1 Bandwidth Usage

Pour finir on redémarre le service Nagios.

5.2. Le test

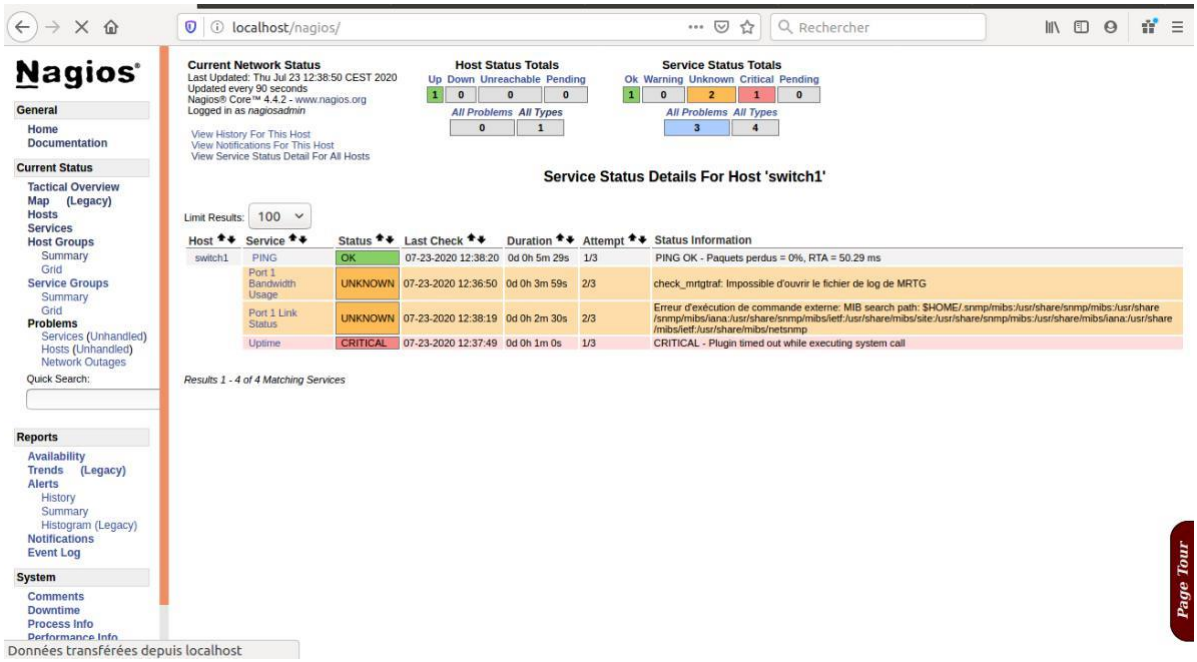


Figure 36: Résultat du check du switch1

Après avoir installé et configurer le manager SNMP sur Ubuntu et activer le protocole SNMP sur les switchs on obtient les résultats suivants :

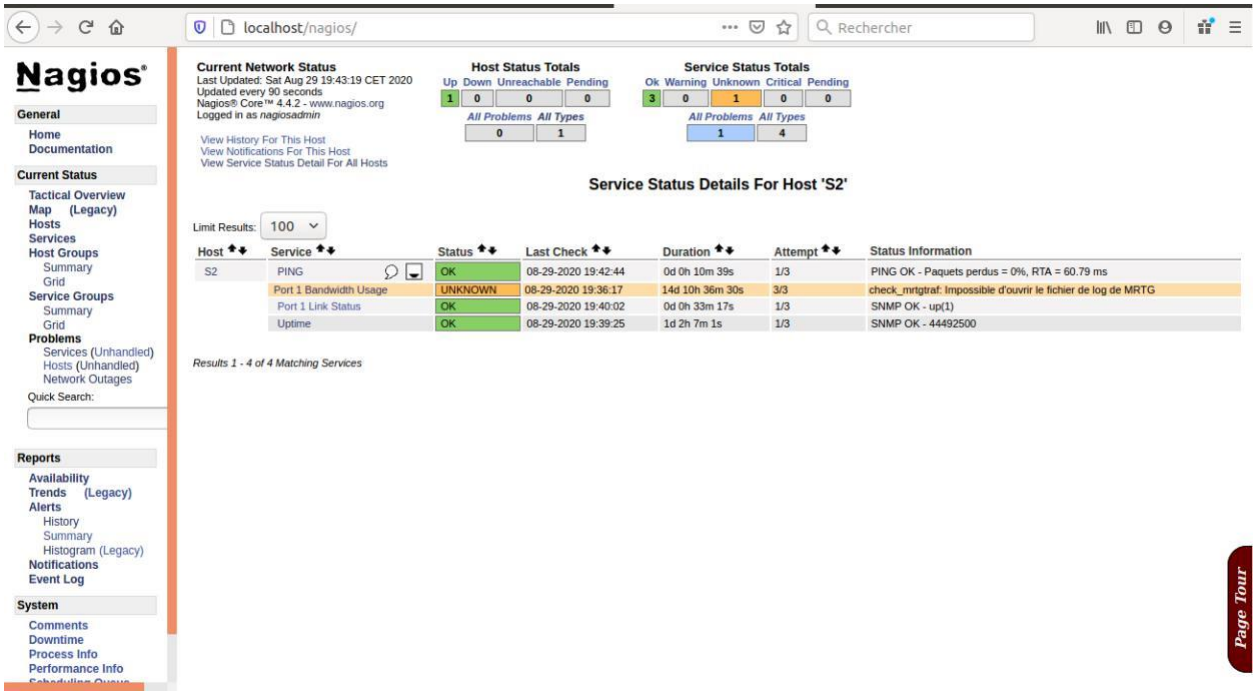


Figure 37 : Résultat du check du s2

6. Notification par mail

Malgré l'existence d'une interface web permettant de voir l'état d'un hôte ou service en temps réel, la notification des contacts reste toujours obligatoire. Pour envoyer les notifications par mail depuis Nagios il faut d'abord installer l'outil correspondant, cela peut se faire de plusieurs manières : utiliser SSMTP, Postfix ou bien encore Sendmail.

Postfix est un élément étroitement lié à Nagios. Effectivement, il sert à l'envoi des notifications vers votre serveur de messagerie et a été conçu comme une alternative plus rapide, plus facile à administrer et plus sécurisée que l'historique Sendmail ; c'est pour cela qu'on la choisie comme solution.

Les notifications dans Nagios font appel à plusieurs éléments :

- d'une part le MTA (Mail Transport Agent) (Postfix) installé sur le serveur Nagios
- d'autre part les 2 commandes utilisées pour envoyer les mails
  - ➔ notify-service-by-email
  - ➔ notify-host-by-email
- les contacts à notifier

6.1. Configuration de Postfix

Pour utiliser Postfix nous devons installer :

---

sudo apt-get install postfix mailutils libsasl2-2 ca-certificates libsasl2-modules

---

Dans le fichier main.cfg ajouter les lignes suivantes :

---

sudo nano /etc/postfix/main.cf

```
# Enables SASL authentication for postfix
smtp_sasl_auth_enable = yes
# Disallow methods that allow anonymous authentication
smtp_sasl_security_options = noanonymous
# Location of sasl_passwd we saved
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
# Enable STARTTLS encryption for SMTP
smtp_tls_security_level = encrypt
# Location of CA certificates for TLS
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

Modifier le fichier sasl_passwd

---

sudo nano /etc/postfix/sasl/sasl_passwd

Convertissez le fichier sasl_passwd en fichier de base de données et supprimez le fichier d'origine du serveur. Nous pouvons utiliser la commande postmap pour la conversion.

---

postmap /etc/postfix/sasl/sasl_passwd

Modifiez la sécurité et la propriété du fichier de mots de passe pour restreindre l'accès de l'utilisateur root et en lecture-écriture uniquement

---

chown -R root:postfix /etc/postfix/sasl

chmod 750 /etc/postfix/sasl

*chmod 640 /etc/postfix/sasl/sasl_passwd**

Redémarrer Postfix

sudo service postfix restart

6.2. Configuration de Nagios

Définir les contacts dans le fichier contacts.cfg

```
define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact template (defined above)
    alias                   nagiosadmin             ; Full name of user
    email                   nagiosyas@gmail.com  ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
    service_notification_commands           notify-service-by-email
    host_notification_commands              notify-host-by-email
}
```

Figure 38 : Définition du contact

Définition des commandes de notification dans le fichier commands.cfg

```
define command {

    command_name    notify-host-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$
\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mail -s "** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$
is $HOSTSTATE$ **" $CONTACTEMAIL$
}


define command {

    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost:
$HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n" | /usr/bin/
mail -s "** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$
 }
```

Figure 39 : Commandes d'envoi par mail

Enfin, on modifie le paramètre du compte Google pour autoriser les applications non Google moins sécurisées à utiliser l'authentification pour envoyer des e-mails via SMTP en notre nom.

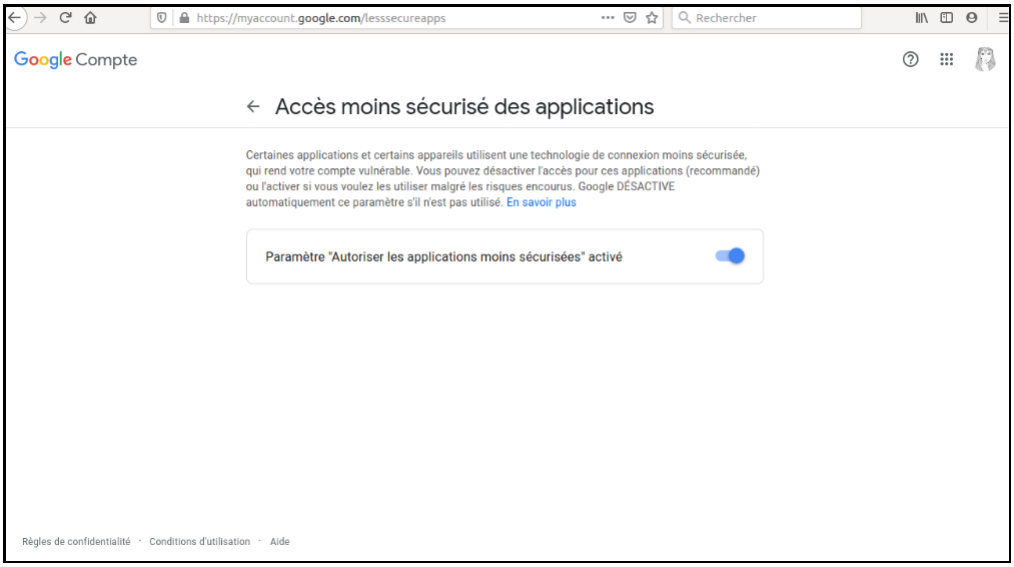


Figure 40 : Paramètre Google

Le résultat de cette configuration est présenté dans la figure suivante :

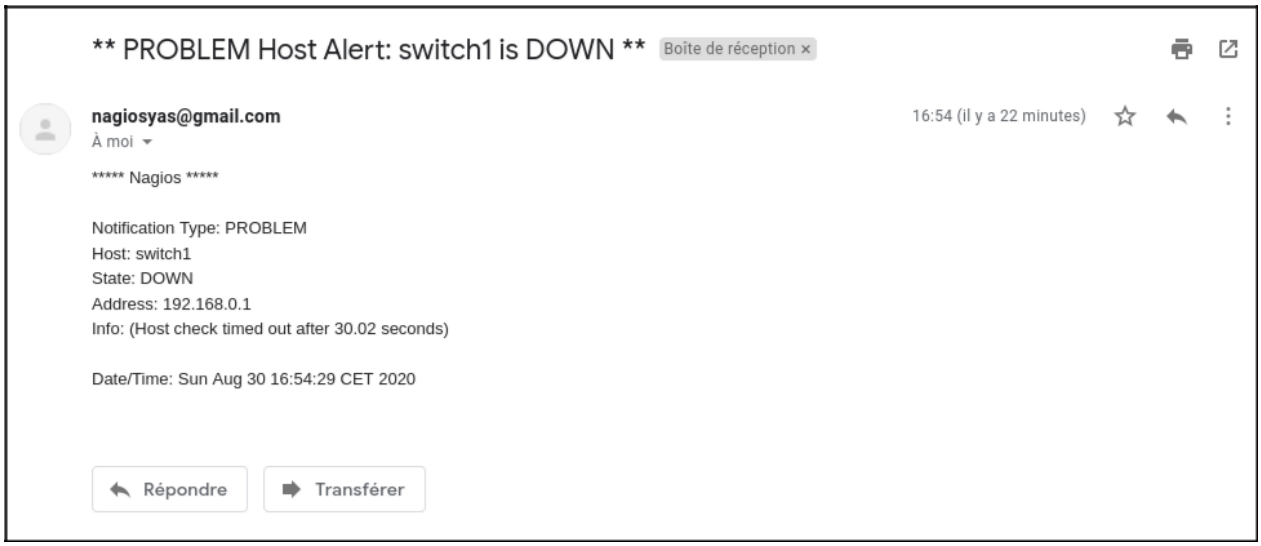


Figure 41 : Mail d'alerte reçue

Conclusion

Dans ce chapitre, nous avons décrit l'aspect pratique de notre projet, où nous avons expliqué les étapes de préparation, configuration et installation de Nagios, et tester cette solution sur quelques équipement de l'organisme d'accueil OPGI.

Conclusion générale

L'objectif de notre projet était de permettre à l'administrateur réseaux de l'OPGI de mieux superviser les équipements et les services de son réseau. En effet une solution de supervision permet de gagner du temps lors du diagnostic des pannes et de faciliter les tâches de l'administrateur réseaux.

Plus le nombre des équipements et des services informatiques augmente, plus les tâches de l'administrateur deviennent trop compliquées et trouve des difficultés de les assurer ce qui engendre une perte du temps et un travail non-accomplie.

Notre travail consistait à mettre en place un outil de supervision système et réseau.

Dans un premier lieu, nous avons fait une étude comparative entre les différentes solutions open source existantes sur le marché, nous avons aussi fait une étude détaillée de l'existant et dégagé ses limites afin de fixer la solution retenue.

Dans la partie réalisation, nous avons mis en place l'outil Nagios et nous l'avons configuré sur les différentes machines, imprimantes et switch de l'entreprise pour les mieux superviser et alerter l'administrateur par mail en cas de pannes.

Comme perspectives, nous proposons l'amélioration de ce travail par :

- La supervision des services de bases de données

La supervision des serveurs et routeurs dès que l'administrateur finit leur installation.

Bibliographie

[1] Axido *'Axido, Votre transition digitale'.* Page Web. URL : https://www.axido.fr/tout-savoir-sur-la-supervision-informatique/. Date de consultation : Mars 2020.

[2] Appvizer *'Appvizer'*. Page Web. URL: https://www.appvizer.fr/magazine/services-informatiques/supervision-info/supervision-informatique. Date de consultation: Mars 2020.

[03] Youssouf N'TCHIRIFOU *'MEMOIRE Online'.* Page Web. URL: https://www.memoireonline.com/04/12/5604/m_Monitoring-dune-infrastructure-informatique-sur-base-doutils-libres16.html#toc55. Date de consultation: Mars 2020.

[4] APPVIZER *'Appvizer'*. Page Web. URL: https://www.appvizer.fr/magazine/services-informatiques/supervision-applicative/supervision-reseau-enjeux-bonnes-pratiques-logiciels. Date de consultation: Mars 2020.

[05] Supinfo école superieure *'SUPINFO'*. Page Web. URL: https://www.supinfo.com/articles/single/2954-fondamentaux-supervision. Date de consultation: Mars 2020.

[06] BookWiki *'Book wiki encyclopédie libre'.* Page Web. URL: https://boowiki.info/art/services-reseau/gestion-de-reseau.html#goto-3. Date de consultation: Mars 2020.

[07] Clever Technologie *'clever Technologie'*. Page Web. URL: https://www.supinfo.com/articles/single/2954-fondamentaux-supervision. Date de consultation: Mars 2020.

[8] Pierre-Alain Goupille *'TECHNOLOGIE DES ORDINATEURS ET DES RÉSEAUX'.*Livre. 8e édition

[9] FrameIP.com '*FRAMEIP.COM Partage des connaissances du monde TCP/IP'.*Site web.URL : http://www.frameip.com/snmp/. Date de consultation: Mars 2020.

[10] www.Developpez.com '*Developpez.com.* Site web.URL: http://ram-0000.developpez.com/tutoriels/reseau/SNMP/. Date de consultation: Mars 2020.

[11] Supinfo école superieure *'SUPINFO'.* Siteweb.URL: https://www.supinfo.com/articles/single/3124-comparaison-outils-supervison. Date de consultation: Mars 2020.

[12] UPEM - IGM *'Comparatif des outils de supervision'* Site web.URL: http://www-igm.univ-mlv.fr/~dr/XPOSE2010/supervision/zabbix.html. Date de consultation: Mars 2020.

[13] Centreon Community '*GUIDE PRATIQUE de la Communauté Centreon'*. PDF. URL: https://www.centreon.com/wp-content/uploads/2018/08/2018_07_Guide_communautaire_Centreon.pdf. Date de consultation: Mai 2020.

[14] Adrien Pichard *'Rapport de stage 2eme année de BTS SIO'* URL: https://www.apichard.fr/ressources/documentations/Stage%20Econocom/Rapport%20de%20stage.pdf. Date de consultation: Mai 2020.

[15] Supinfo école superieure, '*comparaison entre les outils de supervision'*, Site web URL : https://www.supinfo.com/articles/single/3124-comparaison-outils-supervison. Date de consultation : Mars 2020.

[16] David Imanache - Nicolas Joubert *'Olivier Mayaud Informatique & Réseaux – 3'*. Livre

[17] Wikimonitoring *'DOCUMENTATIONSURNAGIOS'*.PDF.URL: https://wiki.monitoring-fr.org/_media/nagios-doc-2x-fr.htmldoc.pdf. Date de consultation Avril 2020.

[18] Elie MABO et Amadou NIANG Etudiants en Master Informatique, Option Sécurité des Systèmes Informatiques , '*La supervision avec NAGIOS'.* Livre.

[19] GrizzlyDev, *'Supervision : l'atout prévention de votre informatique'* Site web URL :https://www.grizzlydev.com/supervision-latout-prevention-de-votre-informatique/ Publié le 17 octobre 2017, Date de consultation Avril 2020.

[20] Nicolas Hennion aka Nicolargo, *'Ebook version 1.0'*, PDF : ebook-nicolargo-nagios-v1_0.pdf téléchargeable sur "https://tsoungui.fr/ebook-nicolargo-nagios-v1_0.pdf"

[21] Nagios entreprises, *'NAGIOS EXCHANGE'*, site web URL :https://exchange.nagios.org/ ,Date de consultation juin 2020.

[22] MONITORING-FR, *'addons officiels'*, site web. URL :https://www.monitoring-fr.org/solutions/nagios/addons/ ,Date de consultation Juin 2020.

[23] WIKI MONIITORING-FR.ORG, *'Superviser un hôte Windows avec NSClient++'* , site web. URL :https://wiki.monitoring-fr.org/nagios/nagios-nsclient-host ,Date de consultation Aout 2020.

[24] WIKI MONIITORING-FR.ORG, *'protocole ncsa'*, site web URL :https://wiki.monitoring-fr.org/nagios/addons/nsca, Date de consultation Aout 2020.

[25] Nagios entreprises, *'Nagios Version 2.x Documentation '*, PDF : nagios-doc-2x-fr.htmldoc.pdf ,Date de consultation Aout 2020.

[26] Ecole ingenieurs 2000 *'Manipulation de Traps SNMP en C et Java'*, PDF :http://www-igm.univ-mlv.fr/~dr/XPOSE2004/rjourdan/concept.html, Date de consultation Aout 2020.

[27] Mémoire réalisé par IKEDICHE Sara ,*' Etude et planification d'un système de supervision (SCADA) sous le logiciel Labview'* ,lien pour le mémoire : http://193.194.80.11/xmlui/bitstream/handle/123456789/2645/memoire%20pfe.pdf?sequence=1&isAllowed=y ,Date de consultation Aout 2020 .

[28] Nagios entreprises, ’Nagios version 3x documentation : Chapitre 14. Supervision des routeurs et des switchs’, site web. URL :https://doc.monitoring-fr.org/3_0/html/gettingstarted-monitoring-routers.html, Date de consultation Septembre 2020.

RESUME

Pour assurer la disponibilité permanente de leur infrastructure informatique, les entreprises ont rapidement compris que la supervision était devenue une ressource-clé. Reçues en mars 2020, OPGI nous a confié la mise en place d'un outil de supervision de son système d'information à travers un stage de 03 mois (mais à cause des circonstances on n'a pas y était pour toute la durée), c'est une solution qui va permettre la supervision complète de son parc informatique. Ainsi la solution NAGIOS a été retenue. Nous l'avons choisi après une étude comparative des outils de supervision qu'on trouve sur internet. Certains critères qui ont été considérés sont : logiciel libre, grandes performances, adaptabilité et fonctionnalités. L'outil déployé permet de contrôler tout type de système d'information. En somme, l'objectif de ce stage est de coupler la puissance de NAGIOS au système d'information de l'OPGI que ces derniers peuvent exploiter !

ABSTRACT

To ensure the continued availability of their IT infrastructure, companies quickly realized that supervision had become a key resource. Received on March 2020, OPGI entrusted us with the implementation of a monitoring tool for its information system through an internship of 03 months (which couldn't be properly done because of the current circumstances), it is a solution that will allow the complete supervision of its computer park. So, the NAGIOS solution was chosen. We chose it after a comparative study of the supervision tools found on the internet. Some criteria that have been considered are: free software, great performance, adaptability and features. In short, the aim of this internship is to combine the power of SHINKEN with the SOFTNET information system that they can deploy!