

Avenue du ciseau 15,

1348 Ottignies-Louvain-la-Neuve

**Implémentation d’un système de surveillance et d’alerte externe d’une infrastructure réseau modulaire**

Travail de fin d’étude réalisé en vue de l’obtention du diplôme de bachelier en Informatique et Systèmes orientation Technologie de l’informatique

**Joel Cedric YEPGANG NAMENYI**



Rapporteur : **Marie-Noël VROMAN**

**Année Académique 2018-2019**

**Avant -propos**

La troisième année du cycle bachelier en Informatique et Systèmes est marquée au second semestre par l’élaboration d’un travail de fin d’études (TFE). Ce dernier a principalement pour objectif d’initier, de former et de familiariser l’étudiant que je suis à la réalisation d’un travail de recherche conciliant à la fois une étude théorique et un cas pratique à présenter devant un jury constitué d’enseignants et des professionnels. L’objet de ce travail de fin d’études, sa présentation et sa défense devraient permettre de prouver ma capacité à pouvoir mener à bien un projet autonome et personnel et donc à m’intégrer dans un milieu socio-professionnel.

~~Cette activité en abrégé TFE est une unité d’enseignement de 16 ECTS qui, valorise notre capacité d’insertion socio-professionnelle par une réalisation autonome et personnelle sur une période estimée à plus de 480 heures.~~

Cette unité d’enseignement (TFE) correspond à 16 ECTS et donc à une période de travail estimée à 480 heures.

**Remerciements**

Avant tout développement de ce rapport,il apparait judicieux de remercier ceux sans qui ce travail n’aurait pas été possible.

J’adresse ainsi mes remerciements à tous ceux qui ont eu la gentillesse de faire de ce travail de fin d’études un moment très intéressant. Il s’agit très spécifiquement de Mme Marie-Noël VROMAN mon rapporteur pour ses conseils, la patience et le professionnalisme pédagogique donc elle a fait preuve tout au long de nos multiples entretiens. Ainsi que, Mr Laurent Schalkwijk pour les échanges et les réponses à mes questions durant la réalisation de ce projet.

Je remercie également l’ensemble du corps professoral du Département Technologie de l’informatique de l’EPHEC pour le travail abattu ~~pour~~ durant ce cursus.

Enfin, je remercie également, ma famille, ma compagne et mes amis qui m’ont beaucoup soutenu moralement, physiquement et ont été toujours là depuis le début.

INTRODUCTION

Dans le cadre de notre parcours à l’Ephec, la dernière année est sanctionnée par le développement d’un sujet représentant notre projet de fin d’études. La finalité étant de mener à bien une mission venant d’un besoin exprimé ou non par une entreprise. Dans cette optique, l’élaboration de celui-ci nous permettra lors de nos multiples recherches d’acquérir une expérience socio-professionnelle puisqu’il intègre de la rigueur, de la méthode et de l’organisation. Le but visé par ce projet est de démontrer qu’avec nos acquis d’apprentissage accumulés durant notre cursus, ~~qu’~~on est capable non seulement de comprendre un problème concret, c’est-à-dire mener une analyse, rechercher des solutions et d’en réaliser des cas pratiques, mais aussi de prouver sa capacité d’insertion socioprofessionnelle par une réalisation autonome.

Ainsi, lors d’une discussion avec un ami, il m’a soulevé une problématique à laquelle il fait face chaque fois dans son réseau. Je le citerai mot pour mot : « chaque fois que je suis à l’extérieur, je ne suis prévenu d’aucune manière de l’arrêt de mes systèmes dans mon infrastructure et même je n’ai aucun suivi de ce qui s’est passé en mon absence. Et en plus si un système tombe, je devrais tout réinstaller à nouveau. ». Ainsi se posait le problème. À savoir : comment garder en permanence un œil sur son réseau tout en réduisant le temps d’intervention sur celui-ci lors des défaillances systèmes ? Dans ce cadre, il s’agit ici d’une :

**Implémentation d’un système de surveillance et d’alerte externe d’une infrastructure réseau modulaire**

Etant définie comme Théorie et traitement de l'information à l'aide de programmes mis en œuvre sur ordinateurs, l’informatique est en constante évolution et par conséquent très complexe quel que soit son domaine d’activité. Le monitoring (surveillance) vu ici comme composant de contrôle est une activité importante que doit incorporer dans ses services une entreprise hébergeant un parc informatique car, la grande partie des entreprises fonctionnent aujourd’hui à travers le réseau. Ainsi la disponibilité et la qualité de services conditionnent le bon fonctionnement de l’entreprise. Dès lors, comment mettre en place de manière sécurisée, un système de surveillance qui permettra de détecter rapidement certains processus défaillants ?   
Dans le souci de mieux gérer et contrôler l’activité de l’infrastructure, ce système non seulement garantira la disponibilité des équipements du parc informatique en cas de panne ou baisse de performance, mais aussi tentera de faire une prévention des problèmes pouvant survenir. Il sera donc question pour nous d’être informé ou d’être notifié en temps réel des possibles cas de problèmes, d’activité interrompue et de l’état de notre réseau qui peut être dû à une coupure d’électricité, coupure d’internet etc.

Tout en présentant le cahier de charges, ce rapport fera ressortir dans un premier temps une étude théorique du sujet, par la suite présentera l’architecture du système, puis décrira au travers de cette cette architecture le système de collecte et stockage des données qui seront par la suite analysées aux fins de créations des différentes actions(notifications).

1. CAHIER DE CHARGES

Bien que celui-ci soit réalisé dans un environnement de simulation, cette rubrique présente les directives qu’on essayera d’atteindre.

Partant du fait que, un client(virtuel) d’une entreprise(virtuelle) désire une solution de surveillance de son réseau pour prévoir des éventuels problèmes, Ci-dessous sont reprises les différentes étapes par lesquelles sont passées ce projet.

1. SYSTÈME DE MONITORING

Un serveur externe au réseau (plus script pour installation et config) qui servira à l’installation du moteur de contrôle. Dans la situation envisagée ce dernier sera en failover avec un autre serveur situé à l’intérieure du réseau.

Un serveur local sous Debian (plus script pour installation et config) qui servira à l’installation du moteur de contrôle et en failover avec le serveur externe.

Pour des raisons de sécurité, un serveur différent contiendra une base de données partagée aux deux servers en failover. Ainsi on conserve les informations en cas de basculement.

1. APPAREILS À MONITORER

Une **VM (VMWare) Debian avec quelques** configurée et fonctionnel plus un script .sh pour tout réinstaller servira de machine à monitorer : plus Backup journalier vers le NAS

* Apache PHP 7.2 ,SSL (Let's Encrypt) ;
* MySQL Server ;
* OpenLDAP ;
* SSH ;
* Zabbix-agent.

Les pfsenses qui permettront non seulement la connexion des deux sites distants, mais aussi permettront les connexions de l’extérieur au réseau local avec un backup journalier vers le NAS. En outre les éléments à surveiller seront :

* Utilisation cpu/ram, hdd ;
* Statistique réseau : trafic en sur chaque interface (+/- temps réel, trafic écoulé sur la journée, la semaine, le mois, total) plus éventuellement par protocoles ;
* Logs pare-feu (20 derniers au moins) ;
* Etat des services, des interfaces ;
* Clients connectés sur les LANs, VPN et leurs IP/MAC .

Une machine Windows avec un agent Windows installé pour récupérer les informations sûres : Utilisation cpu/ram, hdd,

Switch (Cisco SG500-28P) avec agent SNMP pour surveiller son statut et l’état des ports. Un Nas (Synology et Qnap)

* Etats (On/Off) Utilisation cpu/ram, hdd
* Température du cpu, hdd
* Utilisateurs connectés Logs (20 derniers au moins)
* Etats de services
* Date et heure du dernier backup Infos backup (réussis, raté > log)

1. ETUDE THÉORIQUE DU PROJET

Afin d’avoir une idée claire lors de la réalisation de ce projet, il est important d’étudier la faisabilité de celui-ci à travers des thèmes existant en rapport avec celui-ci. Ainsi cette rubrique présentera l’importance de ce projet, étudiera les difficultés liées à celui-ci, pour en finir avec une étude comparative des différents moteurs de contrôle existant sur le marché.

1. IMPORTANCE DU PROJET

Au regard de ce sujet, il est surtout important dans la mesure où il intègre plusieurs notions acquises durant ma formation et certaines de mes multiples formations réalisées en ligne. Il s’agit entre autres de :

* **Gestion de projet*:*** c’est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Une bonne gestion des facteurs opérationnels et de méthodologie font qu'un projet puisse aboutir dans un triangle représentant l'équilibre qualité-coût-délai (QCD).
* **Sécurité informatique**: discipline qui vise la protection de l'intégrité et la confidentialité des informations stockées dans un système informatique, en outre le contrôle des utilisateurs et des droits d’accès aux données.
* **Service réseau** : fonctionnalité assurée par un ou plusieurs serveurs ayant pour fonction la fourniture d'informations à d'autres ordinateurs via une connexion réseau normalisée.
* **SGBD** : système qui stocke et gère des données de façon organisée et cohérente.
* **SQL**: langage permettant l’interrogation des données stockées dans les SGBD.

1. DIFFICULTÉS LIÉES AU PROJET

En ce qui concerne les difficultés de ce projet, elles vont de ses thèmes principaux qui sont monitoring et alerte qui jusqu’ici étaient encore nouveaux pour moi. Bien qu’abordés en parallèle durant le cursus à travers le mot SNMP, cela restait des mots auxquels il fallait trouver un sens avec le sujet. Face à cela, je me suis penché au préalable sur différentes solution de surveillance pour pouvoir à tout moment basculer de l’un vers l’autre cas de blocages sur une solution tierces.

C - ÉTUDES COMPARATIVES DES OUTILS DE SURVEILLANCE

Comme mentionné plus haut, il existe donc plusieurs systèmes de surveillance donc ici nous les classerons en deux catégories à savoir : les outils payants et les outils open source .

1. **Les outils payants**

On entend par outils payants un logiciel développé et commercialisé par une ou un groupe de personne. Ainsi, il en existe plusieurs dont ce rapport présente ceux ayant un peu parcouru la documentation. On y trouvera :

OP5 E-Monitor et PRTG qui sont des logiciels destinés aux serveurs. La surveillance et la gestion du réseau. L’utilisation de ce dernier est limitée à un essai de 30 jours après quoi, il nous faudra une licence.

1. **Les outils open source**

On entend par open source un logiciel gratuit qu’on peut non seulement télécharger et installer gratuitement, mais aussi configurer selon ses propres besoins. Il en existe plusieurs dont ce rapport présente ceux ayant lu la documentation. Il s’agit de :

* Zabbix
* Nagios

Table comparative des outils suscités

|  |  |  |  |
| --- | --- | --- | --- |
| **Catégories** | **Outils** | **Avantages** | **Inconvénients** |
| **Open source** | Zabbix  Nagios | Gratuit, facilité d’installation pour les débutants | Documentation et interface graphique complexe  Non réactivité du développeur sur le partage des modules |
| **Payante** | PRTG  OP5 | Offre un essaie de son environnement pour 30 en limité  Installation très facile | Logiciel à but commercial |

1. ENVIRONNEMENT DE TRAVAIL

Bien qu’ayant une entreprise virtuelle, tout produit se doit d’être développé aux fins d’utilisation. Et, pour sa facilité d’installation et de configuration, ce projet est élaboré sur des environnements et logiciels bien connus avec un accès facile aux différents serveurs par des clients appropriés. Ainsi on décrira cet environnement en deux catégories à savoir :

* Les OS (operating système)
* Os Windows utilisés pour l’installation des logiciels de simulation réseau.
* Os Debian ou Ubuntu pour l’installation des services et logiciels de contrôle.
* Les logiciels
* GNS3 ;
* VMWare ;
* Edraw max.

1. ARCHITECTURE DU SYSTÈME D’INFORMATION
2. **Architecture existante**

Comme tout travail, il convient d’analyser l’existant avant toute modification possible.

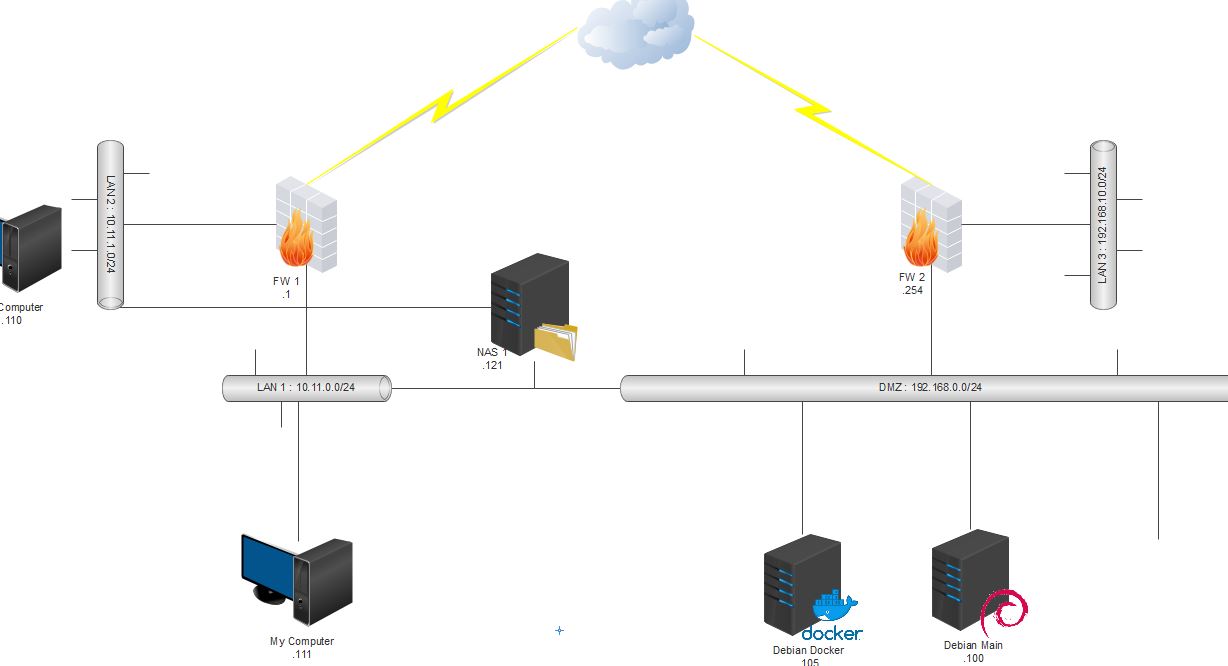


Fig.1 réseau existant

Le réseau ainsi défini, correspond à l’architecture initiale ou de base qui a permis d’élaborer une réflexion sur ce travail. Il est principalement constitué de  :

* Trois LAN : LAN1, LAN2 , LAN3
* Une DMZ qui héberge les machines du réseau interne qui ont besoin d’être accessibles depuis l’extérieur
* Deux firewalls donc un sur le site principale FW1 et un autre sur le site distant FW2
* Un NAS pour la sauvegarde journalière des données.

Afin de le rendre évolutif et mieux adapté aux standings d’un réseau moderne, il a fallu procéder à une analyse et une mise à niveau correspondant aux attentes du cahier de charge et au perspective ~~avenir~~ d’évolution qui permettrait de mieux contrôler les incidents et les interventions potentiels. Ainsi après révision de l’ensemble de réseau, il a fallu envisager des possibles accessibilités externes et internes qui permettraient de maintenir le réseau en activité permanente.

1. **Architecture définitive**

Avant de détailler toutes les étapes de réalisation de ce travail, il est important de donner un aperçu du schéma global. À partir de celui-ci chaque étape de réalisation pourra être mieux visualisé et pourra dès lors montrer son utilité et le but de son intégration. A ceci sera joint une explication du choix de chaque technologie.

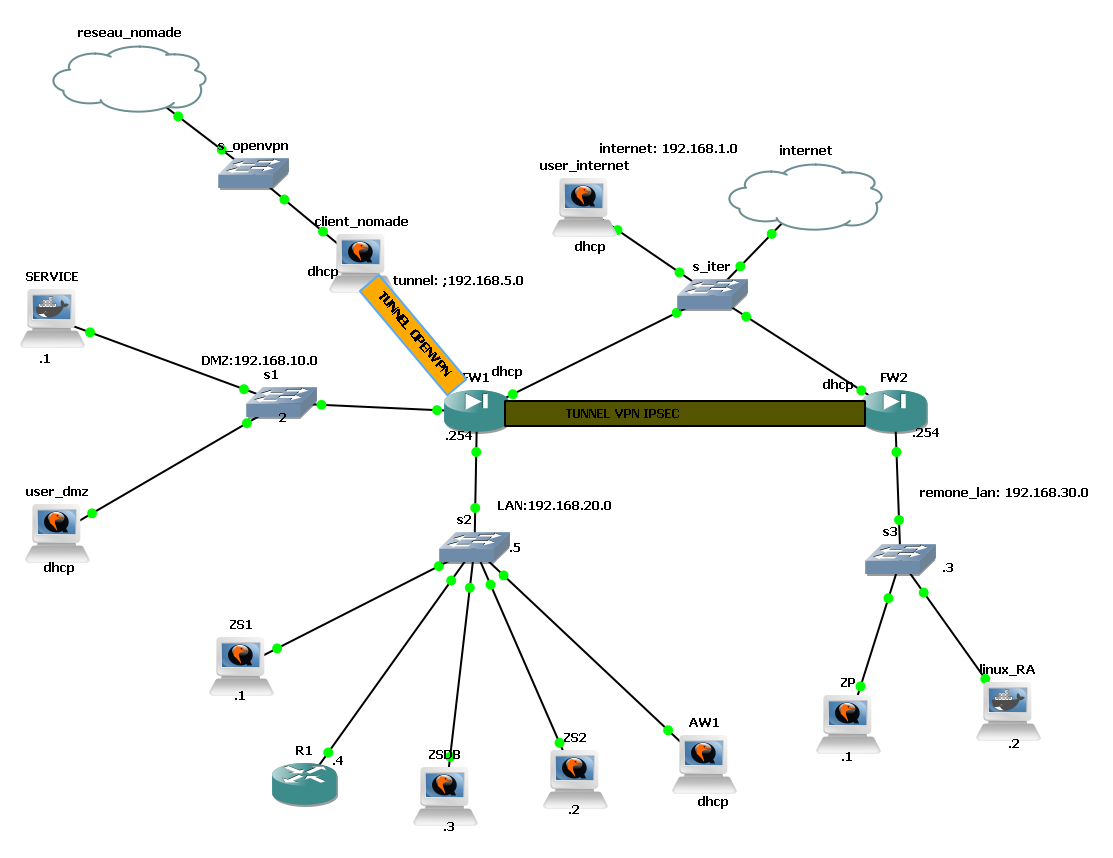


Fig.2 réseau définitif

**Description**

En accord avec le cahier de charge, on observe bien au travers de ce schéma des éventuels ajouts qui peuvent se remarquer avec les termes comme Openvpn qui permettra la connexion par exemple d’un client nomade, tunnel ipsec qui permettra la connexion des deux sites distants, Zabbix pour l’installation du système de contrôle. À noter que tous ces termes seront développés plus bas dans la mise en place des systèmes, services et solution du système de notifications.

1. MISE EN PLACE DE L’OSSATURE DU SYSTÈME DE CONTRÔLE

En rapport avec le cahier de charge ce système doit pouvoir surveiller tous les équipement réseau se trouvant dans parc informatique d’une entreprise. Pour ce faire ce système sera divisé en plusieurs sous-systèmes à savoir : la mise place du système de simulation réseau, l’intégration des différents équipements réseaux nécessaires dans ce système, installation et configuration au niveau des équipements des différents technologies et services en accord avec le cahier de charges, mise en place du système de notification.

1. **SYSTÈME DE SIMULATION**

Comme dit plus haut, il est choisi ici comme logiciels de virtualisation GNS3 et VMWARE.

1. **VM(Virtual machine) Workstation pro**

Ce logiciel est un outil de virtualisation de poste de travail créé par la société [VMware](https://fr.wikipedia.org/wiki/VMware), il peut être utilisé pour mettre en place un environnement de test pour développer de nouveaux logiciels, ou pour tester l'architecture complexe d’un système d’exploitation avant de l’installer réellement sur une machine physique.

Nous avons opté pour une version premium enfin de bénéficier de tout l’environnement qu’offre cette plateforme.

**Avantages de cet hyperviseur**

* **Partitionnement**

Il exécute plusieurs systèmes d’exploitation et repartit les ressources entre les différentes machines virtuelles.

* **Isolation**

Il assure la protection de la sécurité au niveau matériel et les performances accrues tout en déployant le contrôle des ressources.

* **Encapsulation**

Il enregistre des fichiers donnant états des différentes machines virtualisés.

L’ordinateur physique sur lequel nous avons installé VMware Workstation sera pour la suite de ce projet système hôte et son [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) operating system hôte. Pour exécuter VMware Workstation, le système hôte et le système d'exploitation hôte doivent satisfaire à des exigences matérielles et logicielles spécifiques. Pour permettre à gns3 d’être hypervisé car celui-ci est gourmand en ressources.

1. **GNS3**

**Gns3** , pour  Graphical Network Simulator est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre) permettant l'émulation ou la [simulation](https://fr.wikipedia.org/wiki/Simulation_r%C3%A9seau) de réseaux informatiques. Publié pour la première fois en 2008, Il permet la combinaison de périphériques virtuels et réels, utilisés pour simuler des réseaux complexes. Il utilise le logiciel d'émulation [Dynamips](https://en.wikipedia.org/wiki/Dynamips) pour simuler [Cisco IOS](https://en.wikipedia.org/wiki/Cisco_IOS) .  Il est utilisé par de nombreuses grandes entreprises, notamment [Exxon](https://en.wikipedia.org/wiki/Exxon) , [Walmart](https://en.wikipedia.org/wiki/Walmart) , [AT & T](https://en.wikipedia.org/wiki/AT%26T) et [NASA](https://en.wikipedia.org/wiki/NASA) , et est également populaire pour la préparation des examens de certification professionnelle en réseau. En 2015, le logiciel avait été téléchargé 11 millions de fois.

**Raison de son utilisation**

* Il utilise des fonctions issues du monde de Linux. Sous Windows ou Mac, ses fonctions peuvent être mal interprétées. Le serveur étant sous Ubuntu, cela permet à GNS3 d'exécuter ces fonctions sans problème.
* Il utilise la puissance des logiciels de VMware Workstation et son rendement est supérieur à la simple émulation.
* Il permet d’effectuer une simulation concrète de son réseau avant un déploiement définitif en environnement réel ou physique.
* Il permet l’installation de périphériques autres que les routeurs Cisco. Notons qu’en fonction de la taille du réseau à mettre en place , cette dernière pourra demander plus de ressource. Son utilisation dans ce projet nécessite que l’on crée une liaison réseau à la VMware Workstation, ce qui permet la sauvegarde de toutes manipulations effectuées sur gns3 vers Workstation.

Pour pouvoir installer ces logiciels, nous nous sommes procuré une machine faisant office de serveur hôte. Cette **machine serveur**disposera des ressources suivantes

* Système d’exploitation Windows 7 ;
* 64 Go de mémoire RAM (random Access memory) ;
* Processeur Intel Xeon 3.5Ghz ;
* 250 Go de disque dur ssd .

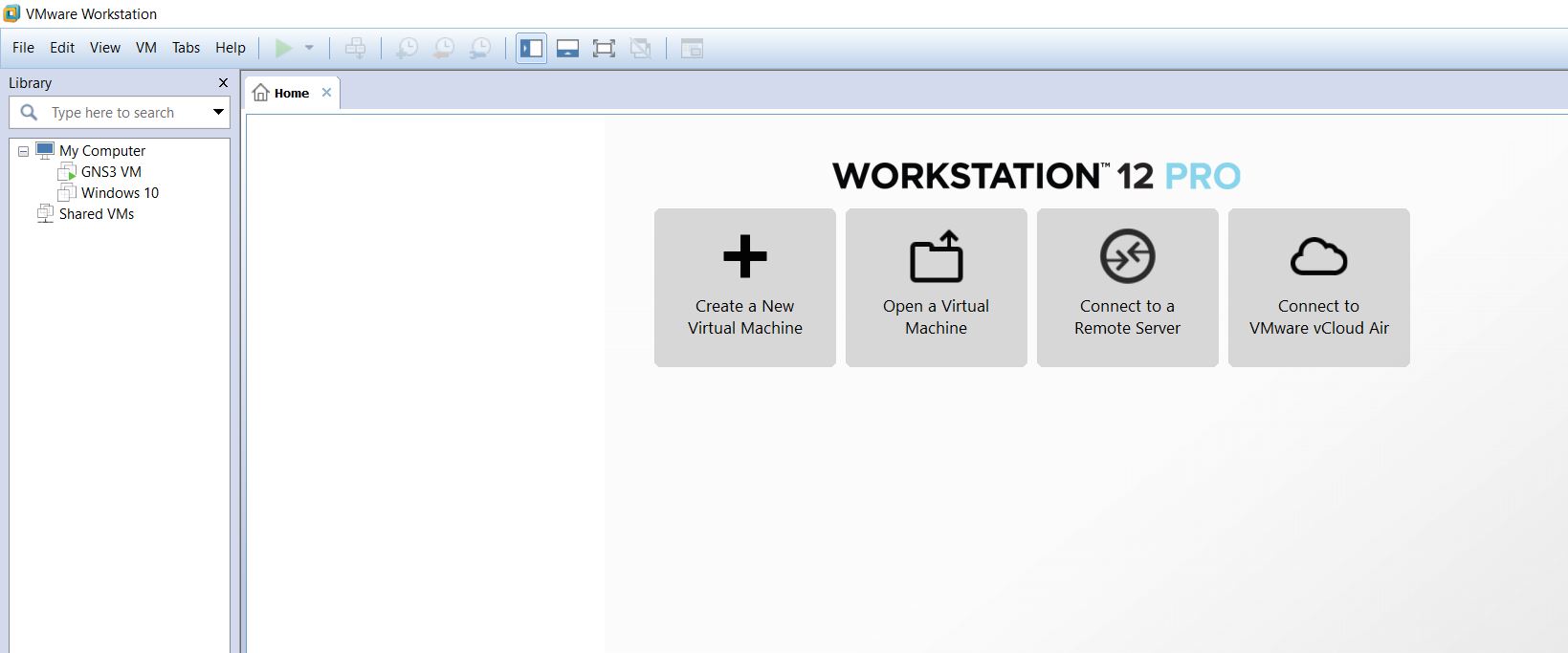


Fig.3 Vmware Workstation

L’installation de gns3 terminée, il nous a été possible par la suite d’importer des images (Appliance) correspondant à un ensemble de périphériques déjà présents dans le système de contrôle de gns3.

Une Appliance sera ici, une machine ou un périphérique déjà présent sur gns3 qui ne peut être utilisé que si l’image correspond à cette dernière est bien installée.

Son importation ne sera possible que depuis le site officiel de gns3. Il nous sera demandé, de créer un compte utilisateur pour y accéder.

1. **Procédures d’installation d’une Appliance**

voir annexes fig.3 (Ajout d’une Appliance) et fig.4 (Appliance installé)

Les Appliance installées , il a fallu procéder à la mise en place de l’ossature du réseau général(voir fig.2 architecture générale du réseau) qui nous a permis d’effectuer des configurations et des tests.

1. **SYSTÈME D’INTEGRATION DE L’ARCHITECTURE RESEAU**

**(**Installation et configuration)

Comme , nous pouvons le constater dans la figure reprenant les Appliance installées mentionnée ci-haut, il est important de commencer par configurer les éléments en présence.

La section suivante explique le pourquoi et le choix des périphériques utilisés pour ce projet. Elle reprend dans un ordre bien précis, la démarche entreprise qui a permis d’atteindre la mise en place d’une telle architecture.

1. **Firewall et sécurité**
2. **PfSense**

PfSense est un [routeur](https://fr.wikipedia.org/wiki/Routeur)/[pare-feu](https://fr.wikipedia.org/wiki/Pare-feu_(informatique)) [open source](https://fr.wikipedia.org/wiki/Opensource) basé sur le système d'exploitation [**FreeBSD**](https://fr.wikipedia.org/wiki/FreeBSD). À l'origine il est un [fork](https://fr.wikipedia.org/wiki/Fork_(d%C3%A9veloppement_logiciel)) de [m0n0wall](https://fr.wikipedia.org/wiki/M0n0wall), il utilise le pare-feu à états [Packet Filter](https://fr.wikipedia.org/wiki/Packet_Filter), des fonctions de routage et de [NAT](https://fr.wikipedia.org/wiki/Network_address_translation) lui permettant de connecter plusieurs réseaux informatiques. Il comporte l'équivalent [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) des outils et services utilisés habituellement sur des routeurs professionnels [propriétaires](https://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire). PfSense convient pour la sécurisation d'un [réseau domestique](https://fr.wikipedia.org/wiki/R%C3%A9seau_local) ou de petite entreprise.

Après une brève installation manuelle pour assigner les [interfaces réseaux](https://fr.wikipedia.org/wiki/Carte_r%C3%A9seau), il est possible ensuite de l’administrer à distance depuis une [interface web](https://fr.wikipedia.org/wiki/Interface_web). Comme sur les distributions Linux, pfSense intègre aussi un [gestionnaire de paquets](https://fr.wikipedia.org/wiki/Gestionnaire_de_paquets) pour installer des fonctionnalités supplémentaires, comme un proxy, serveur VoIP etc.

1. **Fonctionnalités**

Il existe une multitude de fonctionnalités liées à pfsense , parmi lesquelles nous pouvons citer les quelques fonctions suivantes :

* Le filtrage

Il permet d’effectuer un filtrage IP source et destination, port du protocole, IP source et destination pour le trafic TCP (Transmission Control Protocol , est un protocole de transport fiable, en [mode connecté](https://fr.wikipedia.org/wiki/Mode_connect%C3%A9) ) et UDP. Ce qui est important lorsque l’on veut limiter les connexions sur une base des règles. cette fonctionnalité de filtrage est activée par défaut sur pfsense et sa désactivation entrainera certains dysfonctionnements NFS (Network File System qui à l'origine est un [protocole](https://fr.wikipedia.org/wiki/Protocole_de_communication)  qui permet à un [ordinateur](https://fr.wikipedia.org/wiki/Ordinateur) d'accéder via un [réseau](https://fr.wikipedia.org/wiki/R%C3%A9seau_informatique) à des [fichiers](https://fr.wikipedia.org/wiki/Fichier_(informatique)) distants).

L’on peut aussi désactiver cette fonctionnalité (filtrage) si l’on souhaite que notre pfsense fonctionne comme un routeur pur ce qui n’est pas le cas dans ce projet.

* Le portail captif

Il permet de forcer l'authentification, ou la redirection vers une page pour l'accès au réseau. Ceci est communément utilisé sur les réseaux de points chauds (Hot Spots) ou encore pour une connexion distante mais est également largement utilisé dans les réseaux d'entreprise pour une couche supplémentaire de sécurité sur l'accès sans fil ou Internet.

* VPN

PfSense offre quatre options de connectivité VPN: IPSec, OpenVPN, PPTP et L2TP.

* Serveur DHCP et relais

PfSense comprend à la fois les fonctionnalités de serveur DHCP et de relais DHCP.

Une particularité de son utilisation est l’apport technologique lié à sa disponibilité ( load balacing), confidentialité assurée par ces différents protocoles (https authentification, ipsec, OpenVPN etc.)

- sa simplicité d’installation et d’administration qui requiert néanmoins des connaissances informatiques.

Dans notre cas , l’installation de pfsense est utilisée pour exploiter l’ensemble de ces services . Mais encore grâce à sa fonction vpn, j’ai pu établir une connexion distante entre les deux sites mais aussi établir une connexion sécurisée d’un utilisateur externe.

1. **LES VPNS**

Un vpn est un réseau privé virtuel crypté dans le réseau Internet, qui permet à une société dont les locaux seraient géographiquement dispersés de communiquer et partager des documents.

Il dispose de plusieurs protocoles dont deux ont particulièrement retenu notre attention, il s’agit de ipsec et OpenVPN.

1. **Protocol IPSEC**

C’est un protocole de tunneling qui permet d’établir une connexion entre deux sites distants tout comme OpenVPN mais ce dernier nécessite préalablement l’installation d’un logiciel agent sur le client pour que ce dernier puisque contacter le serveur. A la différence des autres , il permet de véhiculer d’autres protocoles tels que SSH, RDP, SMB, SMTP, IMAP, etc.

Il fallut dans un premier temps fixer les adresses IP au niveau des interfaces Wan des deux firewalls. Ce qui a permis dans notre cas d’activer et d’installer sur chacun des firewalls ce protocole, l’un faisant office à la fois de client sur le FW2 et de serveur sur le FW. A ceci s’ajoute, une clé de chiffrement AES 256 bits de part et d’autre. Cette installation s’est soldée par un test de connectivité entre les deux sites. Ce qui a permis d’intégrer une configuration OpenVPN.

1. **Protocol OPENVPN**

Tout comme ipsec il établit une connexion distante entre sites à la différence qu’ici c’est le client qui contacte en premier le firewall à travers une clé partagée que seul le serveur reconnaitra.

En effet, lors de la configuration de OpenVPN j’ai été amené à créer une clé de chiffrement dont une copie a été exportée sur l’ordinateur client. Sur ce même serveur, il a fallu créer un utilisateur et un mot spécifique pour ce client.

Il a donc fallu par la suite installer sur le client un logiciel OpenVPN client et lui attacher le fichier de configuration exporté enfin qu’il puisse s’authentifier et se faire reconnaitre par le serveur.

L’avantage d’une telle connexion est qu’elle permet à un utilisateur distant d’accéder à son réseau local depuis n’importe quel site ou ordinateur.

1. **SOLUTION DE SURVEILLANCE**

Le monitoring ou monitorage est une activité de [surveillance](https://fr.wikipedia.org/wiki/Surveillance) et de [mesure](https://fr.wikipedia.org/wiki/Mesure_physique) d'une activité [informatique](https://fr.wikipedia.org/wiki/Informatique). On parle aussi de [supervision](https://fr.wikipedia.org/wiki/Supervision).

Les raisons de cette pratique peuvent être variées notamment pour des raisons de :

* [Performance](https://fr.wikipedia.org/wiki/Performance), en termes de temps de réponse par exemple ;
* [Disponibilité](https://fr.wikipedia.org/wiki/Disponibilit%C3%A9), indépendamment des performances ;
* D'[intégrité](https://fr.wikipedia.org/wiki/Int%C3%A9grit%C3%A9), l'état des processus sur une machine linux par exemple.

Dans le contexte actuel , travaillant dans un environnement de test, nous nous sommes servis d’un système d’exploitation Ubuntu 16.4 enfin de déployer ma solution.

Lors de notre première élaboration de solution, nous envisagions de mettre dans l’un des réseaux locaux un serveur qui devait faire du monitoring et un autre en externe enfin de faire du failover.

Mais après réévaluation des possibles risques et de perte d’information ou d’indisponibilité de ce serveur ,nous avons recouru à une autre solution qui fut de mettre en place deux serveurs qui nous permettraient de faire du basculement entre les deux si jamais l’un tombait. Cependant, enfin de pousser un peu plus l’idée , il fut judicieux de mettre entre ces deux serveurs une solution appelée pacemaker que nous développerons dans la section suivante. Cela étant, et à des fins plus amélioratives, nous nous sommes posé la question de savoir ce qui se passerait si jamais l’un des serveurs présents en local était stoppé et qu’adviendra-t-il de ce dernier une fois qu’il sera rétabli. D’où, l’idée d’une base de données partagée entre les deux serveurs.

Dans la suite de ce rapport nous développerons les aspects importants et pertinents en ce qui concerne la mise en place d’un tel système de surveillance.

Il convient d’avoir sur chacune des machines que nous nommerons Zabbix serveur1 (ZS1) et Zabbix serveur2 (ZS2) des os ( operating system) Ubuntu 16.04 identiques.

La version Xénial de Ubuntu et la base de données MySQL ont été utilisées sur celle-ci.

Mais avant d’entrer dans le vif de l’installation, il convient de faire une présentation de l’utilité de chaque logiciel installé dans les manipulations qui couvriront la supervision d’un telle infrastructure réseau.

1. **ZABBIX**

Zabbix est une plateforme gratuite de supervision et d’alerte en temps réel. Son  
principal objectif est la supervision de l’ensemble d’équipements des réseaux. Il permet  
à l’administrateur de surveiller ses équipements réseaux . Ce logiciel intègre la gestion des pannes et la gestion des performances.

L’architecture Zabbix repose sur quatre composants donc un Zabbix serveur, Zabbix agent, Zabbix frontend, Zabbix proxy et MySQL.

-Zabbix serveur

Il tourne exclusivement sur linux ou distribution( Debian ou Ubuntu), il est chargé de collecter les données, les analyser, réaliser des graphes, les stocker dans une base de données.

Sur Ubuntu pour son installation il suffit de faire :

#apt-get Install Zabbix-server-MySQL puis de se rendre dans le fichier de configuration /etc./zabbix/zabbix\_server.conf

- zabbix-agent

C’est un logiciel installé sur les équipements qui devront être surveillés. Il est possible de l’installer sur des systèmes d’exploitation tels que Windows, linux, mac etc.

Il interagit, collecte, envoie les données récupérées et écoute sur le port 10050 par défaut.

Commande d’installation ;

#apt-get Install zabbix-agent  
Fichiers configurations : /etc/zabbix/zabbix\_agent.conf  
/etc/zabbix/zabbix\_agent.conf

Notons que la collecte d’informations se fera auprès des différents périphériques, par exemple un switch ou un routeur où nous utiliserons le snmp pour le monitorer

- zabbix-front-end

Ce logiciel va permettre à travers une interface web de mettre en forme les données  
(graphiques), consulter les dernières mesures, aussi d'administrer.

#apt-get Install zabbix-frontend-PHP  
Fichiers configurations : /etc/zabbix/apache.conf  
/etc/zabbix/dbconfig.php

1. **FAILOVER CLUSTER**

Les deux serveurs une fois installés et configurés, il fallut mettre en place le failover cluster pour s’assurer du basculement.

En effet, le failover dans notre cas est un dispositif mis en place entre nos deux serveurs formant des nœuds où chacun fonctionne indépendamment de l’autre. Il aura pour objectif, d’assurer la haute disponibilité et fonctionne selon ce principe ; quand un nœud tombe en panne ou est indisponible pour une quelconque raison, un autre nœud du cluster prend la relève et assure le relais pour maintenir la disponibilité de la surveillance en entendant le rétablissement de l’autre nœud.

Pour le mettre en pratique dans notre cas, il fallut procéder à l’installation de pacemaker et de corosysnc.

1. **Pacemaker**

C’est un gestionnaire de cluster haute disponibilité. Il est chargé de démarrer, arrêter et de superviser les ressources d’un cluster. Il a pour rôle de basculer entre deux ou plusieurs machines.

Pour qu’une telle installation fonctionne, nous avons référé ces champs dans zabbix enfin de pouvoir identifier et de renseigner chaque basculement au niveau du serveur.

* Zabbix1 - premier système de basculement avec zabbix1.example.com comme domaine
* Zabbix2 - deuxième système de basculement avec zabbix2.example.com
* Zabbix – pour le nom du système avec la base de données
* Zabbix - IP de basculement qui est une adresse IP flottante du réseau sur lequel, sont situés mes deux serveurs.

# /etc/hosts

Nous avons ensuite modifié et renseigné les différents champs d’adresse IP respectifs enfin qu’ils puissent se contacter mutuellement.

Notons que le nom des machines doit être diffèrent pour des raisons de configuration plus lisible futur. Le fichier /etc/hostname permet de le faire.

1. **Corosync**

L’installation de pacemaker s’appuyant sur d’autres logiciels tels heartbeat ou corosync pour agir et surveiller des postes, il nous fallut permettre aux deux machines de s’envoyer des informations d’où l’usage de corosync.

Mais avant, il fallut créer une clé sécurisée entre les deux serveurs qui permettra de communiquer de manière sécuriser.

En effectuant la commande corosync-keygen, cette dernière créera le fichier /etc/corosync/authkey

Cette clé par la suite sera transmise au serveur zabbix2 , mais avant un client ssh doit être installé pour faciliter la connexion au serveur distant.

Apt Install openssh-client

etc/ssh/sshd\_config

Décommenter la ligne PermitRootLogin=yes et commenter la ligne prohibit-password

Ce qui permettra à au root distant de s’y connecter .

Sudo scp /etc/corosync/authenkey root@zabbix2:

Cette opération une fois terminée ,nous a permis de renseigner l’adresse IP ou le nom d’hôte auprès des différents nœuds dans le fichier /etc/corosync/corosync.conf qui n’était autre que zabbix1 et zabbix2.

Il Fallait par la suite activer le daemon corosync dans le fichier /etc/default/corosync et le mettre

START=yes

Puis redémarrer les deux démons sur les deux machines

/etc/init.d/corosync start

En effectuant sudo crm il nous affichera l’état du cluster et enfin de vérifier que le basculement fonctionne bien , il nous a suffi d’introduire « sudo crm node standby» sur l’un des serveurs et d’aller vérifier sur l’autre.

1. **INTEGRATION DE LA BASE DE DONNÉES PARTAGÉE**

Sur le serveur nommé zabbix data base (ZSDB) dédié au partage de la base de données , la collecte et la sauvegarde de l’état de la BD des différents serveurs zabbix1 et zabbix2. Jusqu’ici se faisait en local. Nous avons ainsi procédé à l’installation d’une base de données secondaire ou de backup nommée mariadb.

Mariadb est un système de gestion de base de données relationnelles. A l’exemple de MySQL mais cette dernière est un fork plus communautaire et ouvert et à la fois 100% compatible MySQL. Car l’ensemble de son comportement reste similaire à celui de MySQL à l’exception de quelques commandes qui changent.

Pour son installation, il est possible de retrouver sur le dépôt officiel de Ubuntu des packages correspondant et il peut aussi être installé manuellement via :

Apt Install mariadb-server.

Nous l’avons sécurisée en créant un nouvel utilisateur et un mot de passe pour se connecter à la base de données et aussi la création d’une base de données.

Il fallut ensuite se rendre sur l’un des serveurs zabbix de surveillance enfin de transférer le script du fichier de base de données local vers la base de données partagée.

Bien évidemment, une connexion cliente ssh est nécessaire pour ce type de transfert car certaines informations d’authentification seront demandées pour se connecter au serveur distant.

1. **ZABBIX PROXY**

Enfin de décharger les serveurs zabbix(ZS1 OU ZS2), de surveiller les emplacements distants mais surtout de simplifier la maintenance du réseau distribué.t , nous avons mis en place sur le réseau distant un serveur zabbix proxy enfin de collecter des données de performances et de disponibilité pour le compte des serveurs zabbix .

Nous avons pu remarquer ainsi que l’utilisation d’un proxy est un moyen plus simple d’implémenter une surveillance centralisée et distribuée. Car toutes les données récoltées sont stockées localement avant d’être transmises aux serveurs principaux.

Ainsi, aucune donnée n'est perdue en raison de problèmes de communication temporaires avec le serveur. Les paramètres ProxyLocalBuffer et ProxyOfflineBuffer dans le [fichier de configuration](https://www.zabbix.com/documentation/3.0/manual/appendix/config/zabbix_proxy) du [proxy](https://www.zabbix.com/documentation/3.0/manual/appendix/config/zabbix_proxy) contrôlent la durée de conservation locale des données.

Il est bien de noter que , le proxy Zabbix est un collecteur de données. Il ne calcule pas les déclencheurs, ne traite pas les événements et n'envoie pas d'alertes.

1. **SYSTEMES DE MONITORING**

Pour effectuer une surveillance sur le réseau , l’agent zabbix devait être présent et activé sur chacun des périphériques du réseau.

De même, il fallait aussi installer et activer en parallèle les services à monitorer. Pour par la suite l’intégrer au système de surveillance zabbix server.

1. **INTEGRATION D’UN HÔTE À MONITORER**

Les hôtes zabbix sont des périphériques que nous souhaitons surveiller( serveurs, commutateurs, poste de travail etc.) . La création d’un hôte est l’une des premières tâches de surveillance sur notre serveur zabbix. Sur zabbix, ils sont organisés en groupe.

L’ajout d’un hôte nécessite une bonne documentation de l’environnement zabbix. Il est important de bien assigner l’hôte au bon groupe enfin de mieux se repérer au moment d’élaborer sa séquence de surveillance.

1. **INTEGRATION DU SYSTÈME D’ALERTE**

Le système d’alerte est un ensemble de mesure mis en place pour assurer la disponible d’un service ou d’un périphérique surveillé.

Il est recommandé de bien cerner les composantes qui le définissent et leurs principes de fonctionnement.

1. **Terminologie**

**Item :** un item est un élément qui teste des services et collecte des données.

Comme nous l’avons mentionné un peu plus haut la collecte des données n’est faisable que via un agent car ce dernier se changera de récupérer pour chaque item à intervalle de temps régulier l’ensemble d’information et de les envoyer au serveur zabbix.

**Trigger :** génère un évènement en réaction à une certaine valeur ou donnée remontée par un item.

**Trigger = évènement = statut**

**Action :** envoi des alertes (notifications), en fonction d’évènements précis générés par des triggers.

**Action** **= alerte = notification**

1. **Principe de fonctionnement**

Dans zabbix, la génération d’une alerte est la résultante de plusieurs conditions réalisées entre les différentes terminologies citées. Le schéma suivant montre la relation entre ces trois éléments.

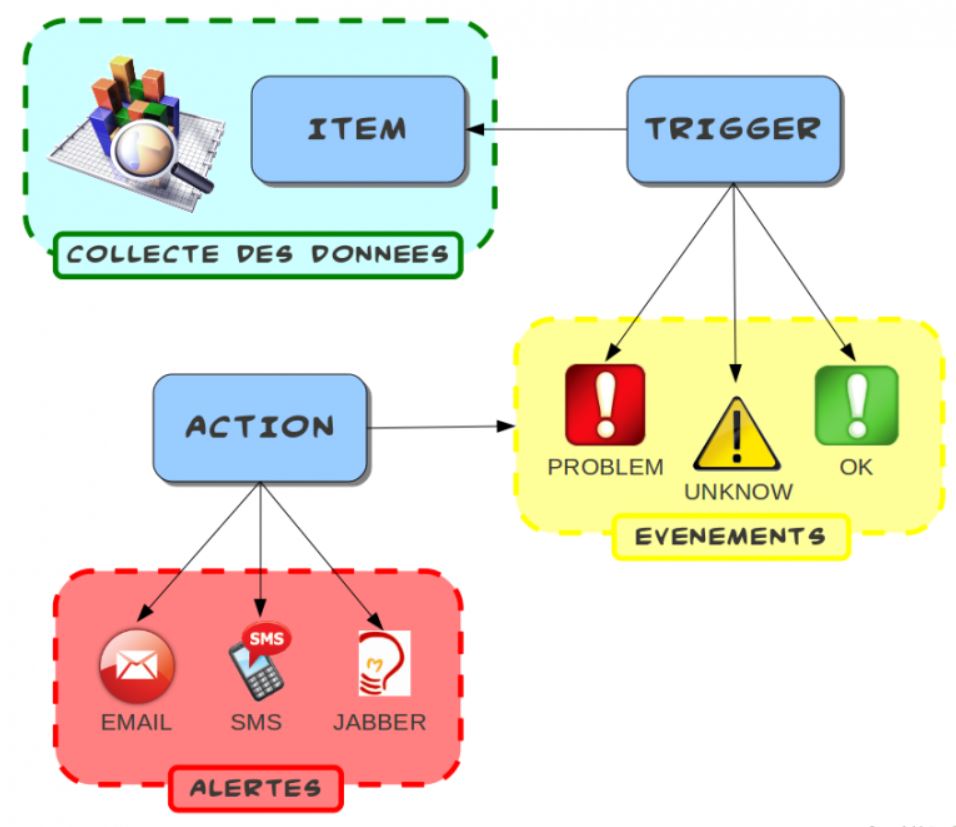


Fig.5

Dans le processus de génération d’une alerte, le premier maillon de la chaîne est l’item. Cet élément collecte les données à surveiller, comme par exemple « Est-ce que le port imp. est ouvert ? », ensuite, le trigger chargé de surveiller cet item, en fonction de ses conditions et des valeurs remontées par l’item, il génère un évènement de type PROBLEM, UNKNOW ou OK (avec différents niveaux de criticités possibles).

Pour créer une alerte, le 3ème et dernier élément de Zabbix à entrer en jeu est l’action. Celui-ci fonctionne comme le trigger, à la différence qu’il surveille les évènements créés par le trigger (au lieu des valeurs remontées par les items pour ce dernier) , selon ses conditions, il génère alors une alerte (ou notification) de type EMAIL.

Switches

Pfsense



Web pages

SNMP

Agent Windows

Agent linux

**Architecture Zabbix**



**Réseau distant**

Agents linux

Agents Windows

Nas snmp

Web pages

IPMI interface

**Difficultés rencontres**

**Améliorations possibles**

1. AMELIORATIONS
2. CONCLUSION

Dire ce qui est concrètement terminé, ta satisfaction par rapport au travail.

Dire ce que tu changerais si tu recommençais maintenant, ce que ce sujet t’a appris.

Donner les améliorations futures.

Dire en quoi ce travail virtualisé est utile pour l’adapter à un vrai réseau.

Parler de ce que tu as envie de faire : ton avenir professionnel, ce que tu peux mettre en avant sur ton CV suite à tes 3 années d’études à l’EPHEC et suite à la réalisation de ce TFE