Institut Universitaire de Technologie,
Aix-Marseille Université

RAPPORT DE STAGE
Diplôme Universitaire de Technologie
Spécialité Réseaux et Télécommunications

Les Différentes Formes de Supervision

Morgan KATHAPERMALL

ITIKA GROUPE

Responsable entreprise : Maxime LONGUET

Responsable académique : Arnaud FÉVRIER

2019

TABLE DES MATIERES

1. INTRODUCTION

L'infrastructure informatique est aujourd'hui un élément clé pour les entreprises peu importe leur taille : TPE, PME, Grands Comptes, … Le système d'information est devenu un élément central de l'activité des différents services et doit fonctionner pleinement et en permanence pour garantir l'efficacité de l'entreprise. Son rôle intervient à tous les niveaux : les réseaux, les terminaux utilisateurs, les serveurs d'applications ainsi que les données, avec comme objectif, la garantie du bon fonctionnement de l'entreprise.

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.

La supervision informatique désigne l'ensemble des outils et ressources déployés pour veiller au bon fonctionnement de votre système d'information. Le but est de mettre en place une maintenance préventive afin d'éviter les interruptions de service et de détecter en amont les failles des infrastructures informatiques pour contrer les cyber-attaques. Le monitoring 7J/7 permet de vérifier en permanence que les pare-feu et antivirus sont actifs et que les serveurs sur lesquels sont répliquées et sauvegardées vos données fonctionnent correctement.
Ainsi, un service de supervision informatique vous permettra d'avoir un système d'information opérationnel et disponible.

Durant mon stage j'ai pu aborder différents aspects de la supervision, et nous allons étudier tout cela ensemble au travers de ce rapport.

2. PRESENTATION DE L'ENTREPRISE

ITIKA est une petite société de services spécialisée dans le logiciel libres. Elle est fondée en 2004 par Maxime Longuet. Etant devenue une SAS (Société par Action Simplifiées) et nommée ITIKA-GROUPE le 1er mai En 2011 il est rejoint par Mr Peinado, qui deviendra son associé. Elle actuellement basés au 3 place de la Rotonde 13014 Marseille.

L'entreprise ne se limite pas à un domaine précis, elle propose :

• L'administration système et infogérance : Spécialiste dans les plateformes de production Open Source. La société propose la gestion des serveurs d'applications web et mail. Tout dernièrement ITIKA propose l'infogérance d'instances AWS, Amazon Web Services.

• L'Intégration Logiciel Libre : ITIKA propose une gamme variée de briques Open Source à intégrer dans des infrastructure réseau. Expert dans les briques libres d'infrastructure (WEB, DATABASE, FICHIER, PROXY, VPN, EMAIL...) Mais également dans les nombreux produits Libre et Open source disponible pour des entreprise (ERP, GED, CRM, WIKI, SOCIAL NETWORKING...).

• Le Développement progiciel : Forte de l'expérience de développements d'outils métiers de M. Longuet, la société propose le développement de progiciels clés en main sous forme d'applications web, dédiée au cœur du métier du client.

• Le Développement web : Spécialiste des solutions WEB Libre, ITIKA est à même d'installer, d'héberger, de modifier et personnaliser les applications de type CMS (Gestion de contenu) pour l'élaboration de site plaquette ou portail d'information.

• Formation : Formations à l'utilisation des produits fournis aux clients, transfert de compétences, formations à l'utilisation de logiciels et systèmes Open Source.

L'équipe est composée de 5 personnes à plein temps et 3 externes. Cette dernière est principalement composée de développeurs et administrateurs systèmes.

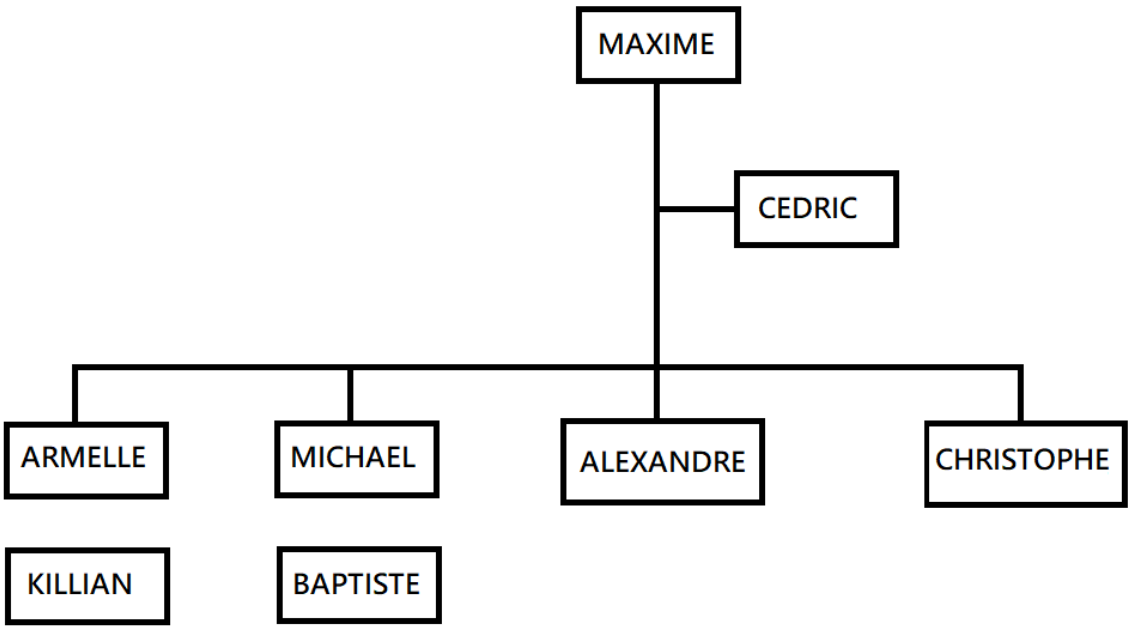


Figure 1 - Organigramme

Comme vous l'aurez remarqué, le fait que l'entreprise est petite le système de hiérarchie est chamboulé, Pour Résumer :

Maxime Longuet est le Président et le Directeur Technique que ce soit au niveau de l'infrastructure système ou du développement web. Il est aussi un administrateur système

Cédric Peinado est le chef de projet, c'est-à-dire qu'il analyse les besoins des clients et qu'il les accompagne. Il s'occupe également de la partie facturation et comptabilité. Il exerce cette double casquette quelque soir les prestations de l'entreprise.

Pour les développeurs nous avons :

Armelle qui est une Architecte logicielle externe

Michael qui est un développeur backend il produira du code pur qui servira au progiciel. Il code principalement en PHP.

Killian qui est également développeur backend externe

Baptiste qui est développeur front, il s'occupera d'intégrer le programme et crée l'interface graphique pour le client notamment avec des codes CSS, Cascading Style Sheets ce lannguage permet de mettre en forme des pages web par exemple.

Concernant l'administration système nous avons :

Alexandre, il s'occupe de l'infogérance sous Linux mais également de la partie support technique. C'est-à-dire qu'il gère aussi les demandes et problèmes des clients qui sont envoyés sous forme de ticket électronique

Christophe est un administrateur système externe tient le même rôle qu'Alexandre mais sous le système d'exploitation Windows.

3. CENTREON

3.1 Qu'est-ce donc ?

Centreon est un logiciel de supervision informatique. Ce logiciel libre de supervision système et réseau est basé sur Nagios. Il assure une surveillance permanente des machines dans un parc et génère automatiquement des alertes en cas de dysfonctionnement.

En termes de supervision réseau, Centreon relève la disponibilité et les temps de réponses des services basés sur les protocoles comme HTTP, FTP, SMTP et bien d'autres protocoles.

Au niveau de la supervision système, le protocole SNMP pour Simple Network Management Protocol qui est essentiel au monitoring, il va permettre de surveiller les ressources systèmes tel que la charge l'occupation des partitions de disques, l'utilisation des capacités mémoires ou encore la bande passante des interfaces. L'implémentation du protocole sur le système Linux permet également d'exécuter

Pour information SNMP est un protocole relativement simple. Pourtant l'ensemble de ses fonctionnalités est suffisamment puissant pour permettre la gestion des réseaux hétérogènes complexes. Il est aussi utilisé pour la gestion à distance des applications : les bases de données, les serveurs, les logiciels, etc.

L'environnement de gestion SNMP est constitué de plusieurs composantes : la station de supervision, les éléments actifs du réseau, un protocole et les variables MIB.

Pour Management Information Base, la MIB est la base de données contenant les variables, aussi appelés OID. Les différentes composantes du protocole SNMP sont les suivantes :

Les éléments actifs du réseau sont les équipements ou les logiciels que l'on cherche à gérer. Cela va d'une station de travail à un concentrateur, un routeur, un pont, etc. Chaque élément du réseau dispose d'une entité dite agent qui répond aux requêtes de la station de supervision. Les agents sont des modules qui résident dans les éléments réseau. Ils vont chercher l'information de gestion comme par exemple le nombre de paquets en reçus ou transmis.

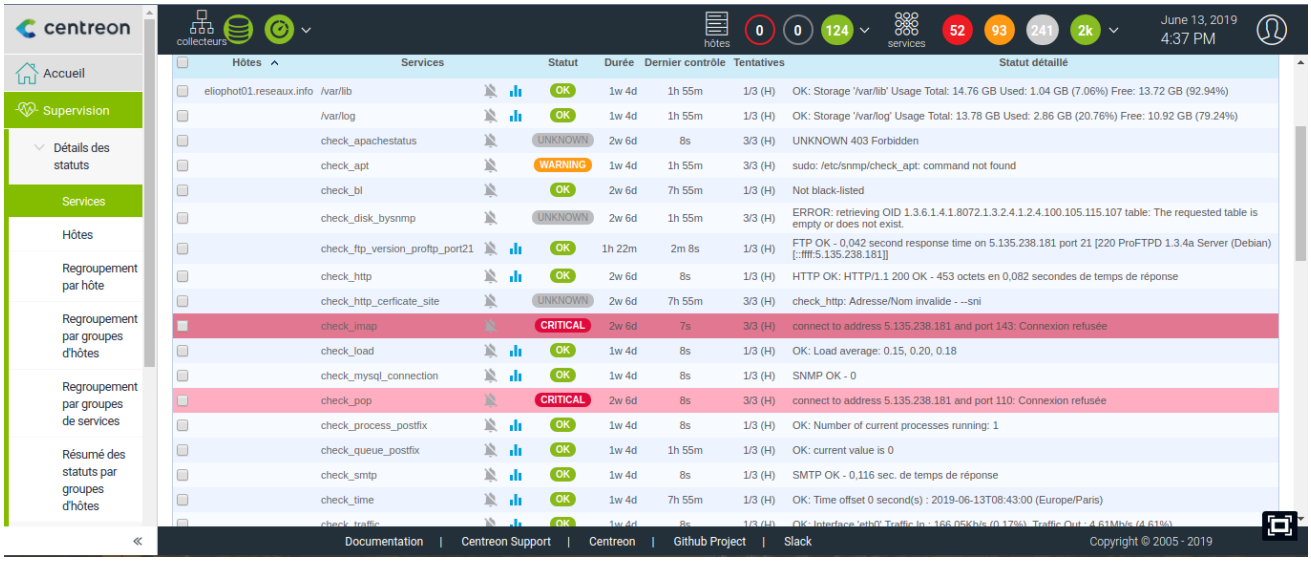


Figure 2 - Interface de Centreon

Sur cette image vous pouvez voir l'interface web de Centreon.
Commençons par le haut, vous pouvez apercevoir le bandeau noir ce dernier contient à sa droite deux case contenant les noms hôtes et services.

Pour la partie hôte vous notifie simplement du nombre de serveurs joignables en vert et indisponibles en rouge, le gris correspondant à un état inconnu de l'hôte.

Pour la partie service, Nagios est le moteur de la plate-forme, il peut être vu comme un ordonnanceur de tâche qui gère l'exécution de chaque programme de surveillance. Il traite les résultats retournés et crée des notifications si les services ne répondent pas aux exigences de performances, définis par l'administrateur avec les seuils d'alerte ou si ceux-ci ne sont tout simplement plus joignable sur le réseau.

Le logiciel vous informe du nombre de services actif en fonctionnement en vert. En gris vous verrez tous les services dont l'état est inconnu c'est à dire que le service ne retourne pas de valeur suite à un mauvaise configuration du fichier ou une erreur d'OID. En orange pour un service en phase d'avertissement ou en rouge pour un service en phase critique. Par exemple si vous voulez surveiller l'espace disque d'un serveur pourrez définir qu'à 70% d'utilisation le service passera en avertissement et à 80% il passera en critique.

Vous avez ensuite la possibilité d'être notifié par mail puis par SMS c'est ce que l'on appelle l'escalade.
Au préalable il faudra renseigner les informations d'un contact et le mettre dans le groupe d'escalade et également choisir les services et hôtes à surveiller Il faut choisir judicieusement ce qu'un veut car chaque SMS est payant.

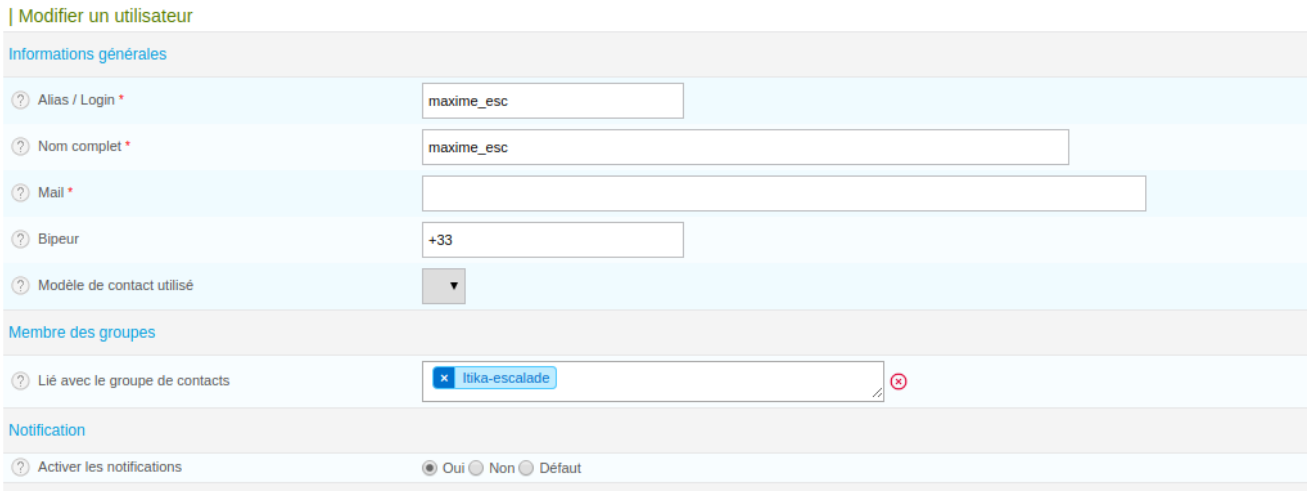


Figure 3 - Informations d'un utilisateur en vue de l'intégrer au groupe d'escalade

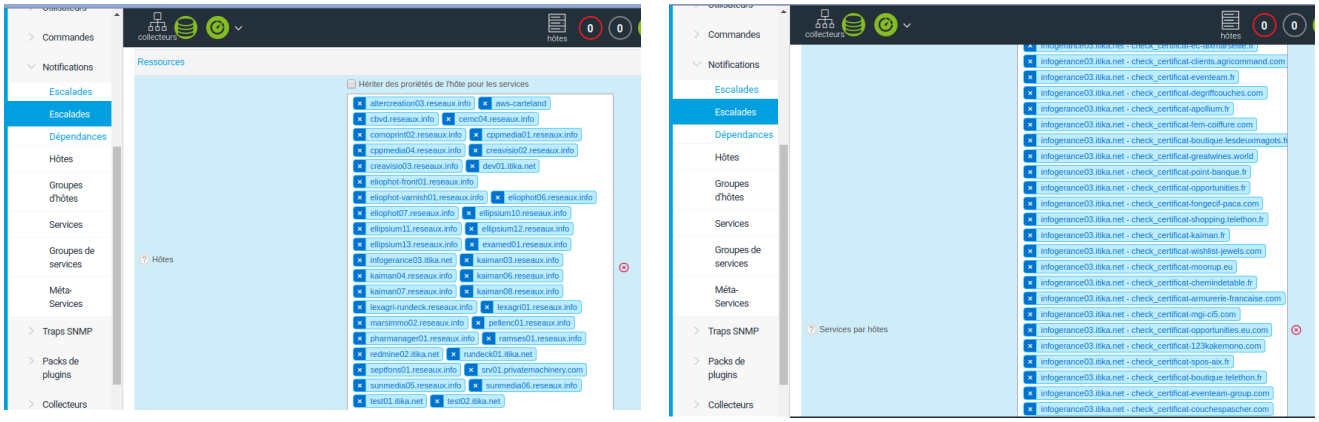


Figure 4 - Un exemple de ressources à surveiller

Ensuite il suffit de définir le nombre de mail dans un intervalle de temps puis demander à Centreon de basculer sur l'envoi de messages sur votre téléphone. Cela permet à ITIKA d'avoir une grande réactivité en cas de problème majeur sur un hôte.

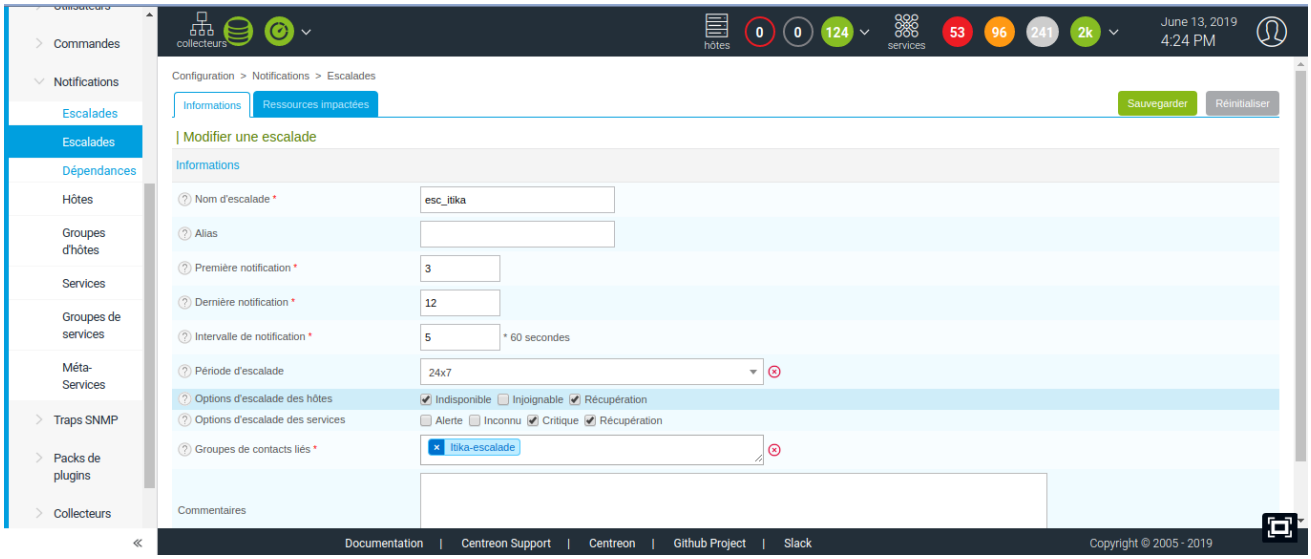


Figure 5 - Configuration des temps et nombre de notifications

Son interface évoluée offre une approche de Nagios plus simple, et un système de configuration plus pratique ainsi que des fonctions avancées de reporting et de tracé de graphiques permettant de suivre très précisément l'état de son réseau et de ses machines et de conserver un historique de tous les événements.

Chaque aspect de la supervision est assuré par plusieurs programmes spécialisés, appelés « plugins ». C'est par leur intermédiaire que Nagios vérifie régulièrement l'état des services réseaux et des systèmes supervisés.

3.2 Ma Mission

3.2.1 Création d'un hôte

Ma mission sur ce logiciel fut de migrer les hôtes du Centreon vieillissant nommé infogérance01 vers un nouveau, nommé infogérance03, avec une interface web plus moderne mais surtout des sondes plus précises. Le but étant de pouvoir surveiller les machines et trouver le plus rapidement possible l'origines des erreurs en cas de pannes et prévoir ces pannes pour les éviter.

Pour effectuer cette tâche j'ai tenu un tableur en répertoriant chaque machine sur infogérance01 pour ne pas me perdre et m'assurer de ne pas oublier une sonde.

Deuxièmement pour migrer les machines il faut d'abord migrer ce qu'on appelle les hôtes, ça peut être un serveur, un ordinateur ou n'importe quel équipement d'un réseau informatique.

Pour commencer, saisissez toutes les informations concernant la machine à superviser dans le menu de configuration.

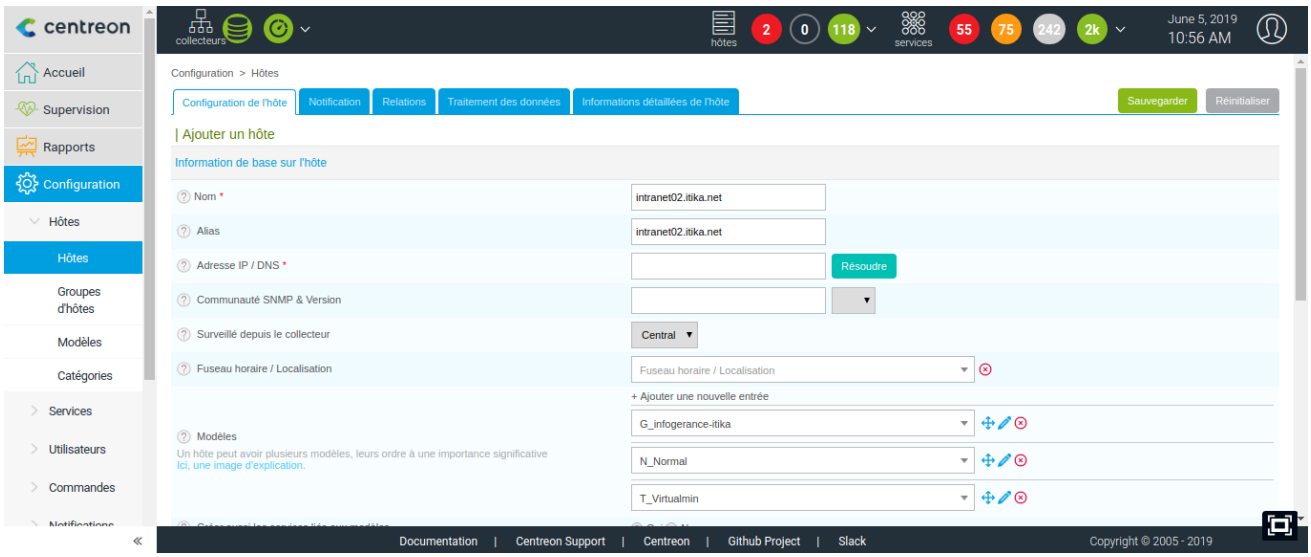


Figure 6 - Menu de configuration d'un hôte

Une fois que vous aurez entré le nom de votre équipement l'adresse IP de ce dernier. Centreon vous proposera trouver l'IP de votre machine à partir de son nom, cela fonctionnera uniquement si vous aurez préalablement configurer votre hôte sur la partie DNS.

Et après avoir renseigner le nom et la version de la communauté SNMP vous pouvez ajouter des modèles. Les modèles vous permettront de créer une liste prédéfinie de sondes, si vous appliquer le modèle à un hôte les sondes seront automatiquement implémenter.

Il vous restera juste à exporter la nouvelle configuration vers le collecteur qui vérifiera qu'il y a aucune erreur dans la nouvelle configuration et l'appliquer

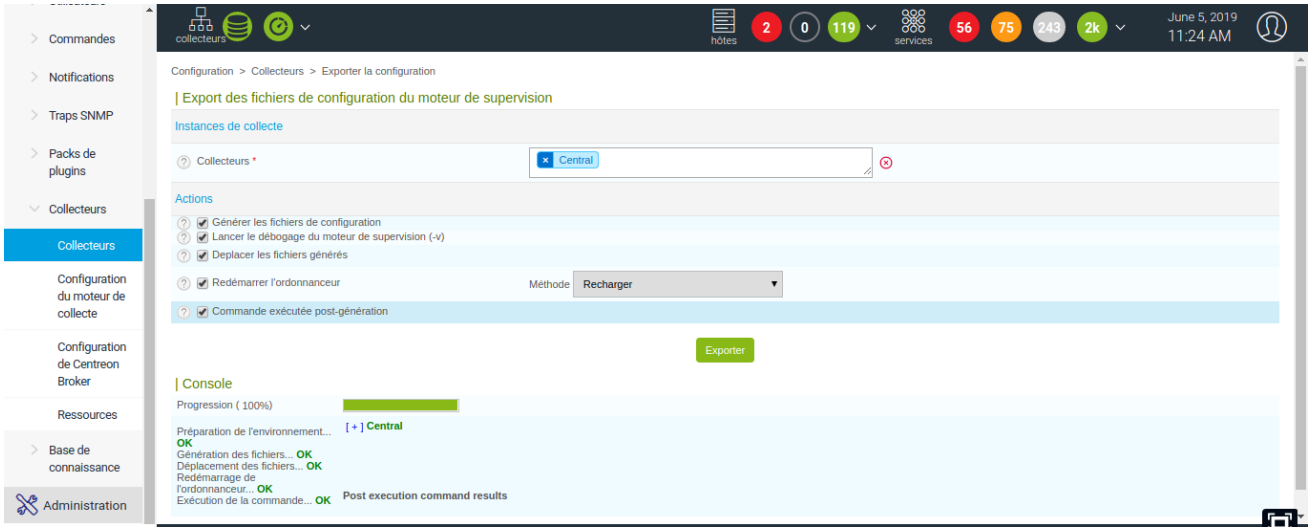


Figure 7 – Export de la configuration vers le collecteur

3.2.2 Comment créer un service

Maintenant que l'hôte est installé nous allons voir de quoi est composer un service, pour rappel, ce dernier nous permet d'avoir tt les informations que l'on souhaite. Tout d'abord un service c'est une seule valeur c'est à dire que si on veut surveiller la bande passante et l'espace disque on aura 2 services.

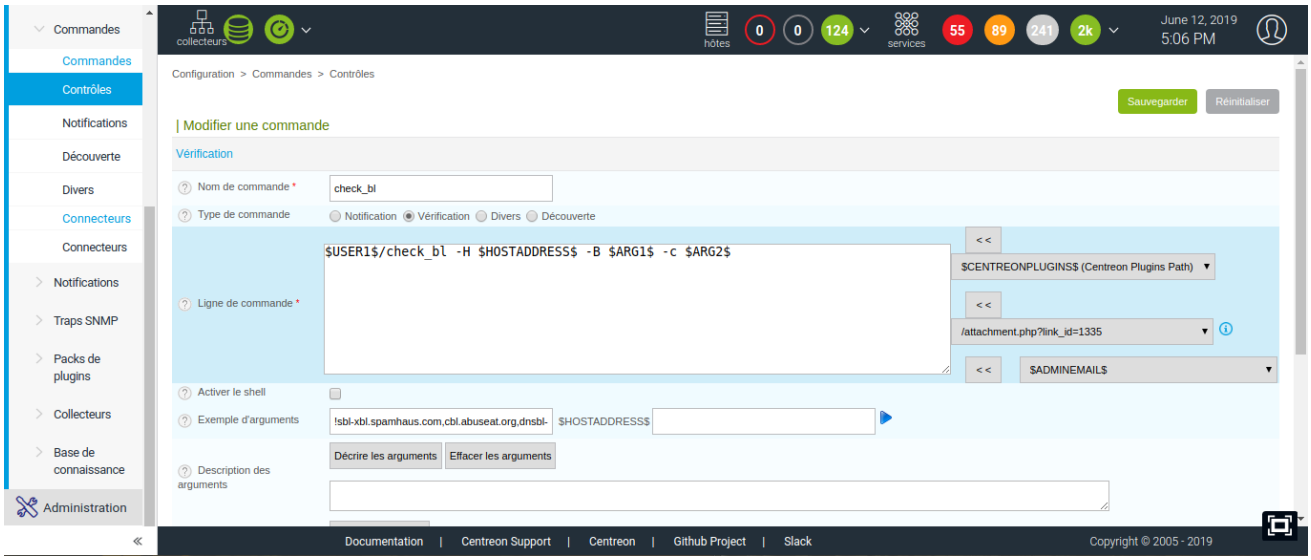


Figure 8 – Création de la ligne de commande

Nous allons analyser le service de blacklistage. Il s'agit de vérifier si l'équipement que l'on surveille est rejeté.

A la base, un service c'est une ligne de commande, donc c'est dans le menu commande de contrôle que tout débute. Comme dit précédent on appelle un plugin via la commande **$USER1$/check_bl** et on spécifie les options avec un système d'argument comme si vous exécutez un script. Ici le **-H** servira au plugin pour l'adresse IP par la translation DNS le **-B** servira pour indiquer ou vérifier le blacklistage et le **-c** correspondant à critique est notre seuil d'alerte.

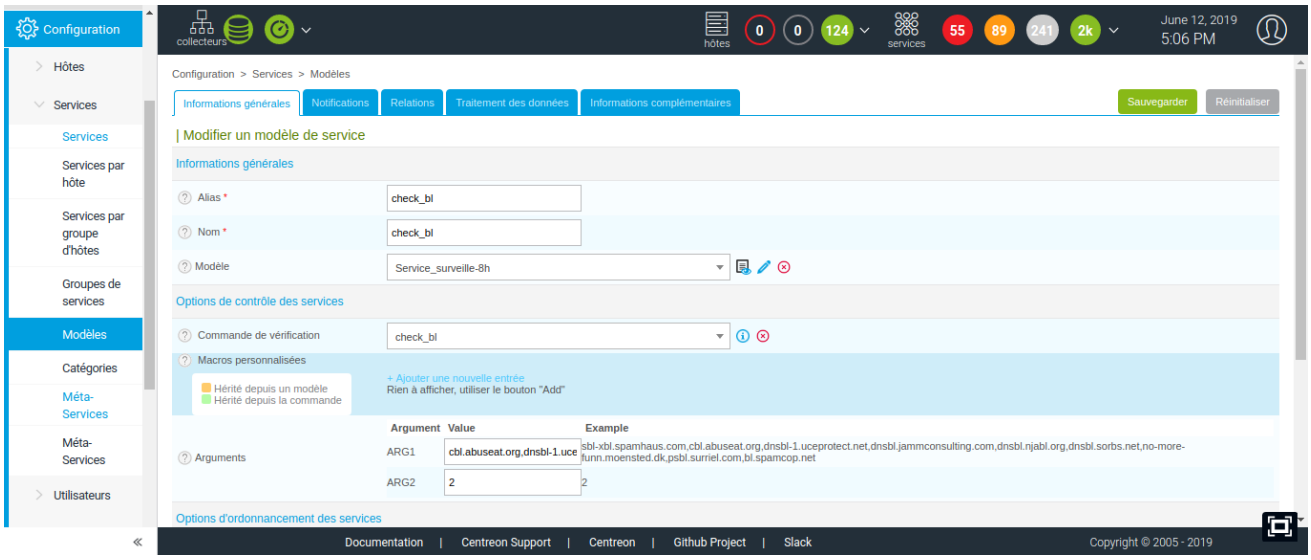


Figure 9 – Configuration d'un modèle à partir de la ligne de commande précédente

Notre ligne de commande créé nous nous rendons maintenant dans le menu Modèles. A ce moment nous sommes à la moitié du parcours il suffit de donner un nom, puis de donner l'intervalle de surveillance du service pour sinon le collecteur vous retournera une erreur. Et dans l'onglet commande de vérification on va venir appeler la commande précédemment créée.

Dans la partie Arguments vous pouvez spécifier les variables à utiliser par le plugin. Ici on spécifie le serveur à interroger pour le blacklistage et une valeur à laquelle le service nous notifiera de son état critique autrement dit si le l'entité surveiller est blacklister.

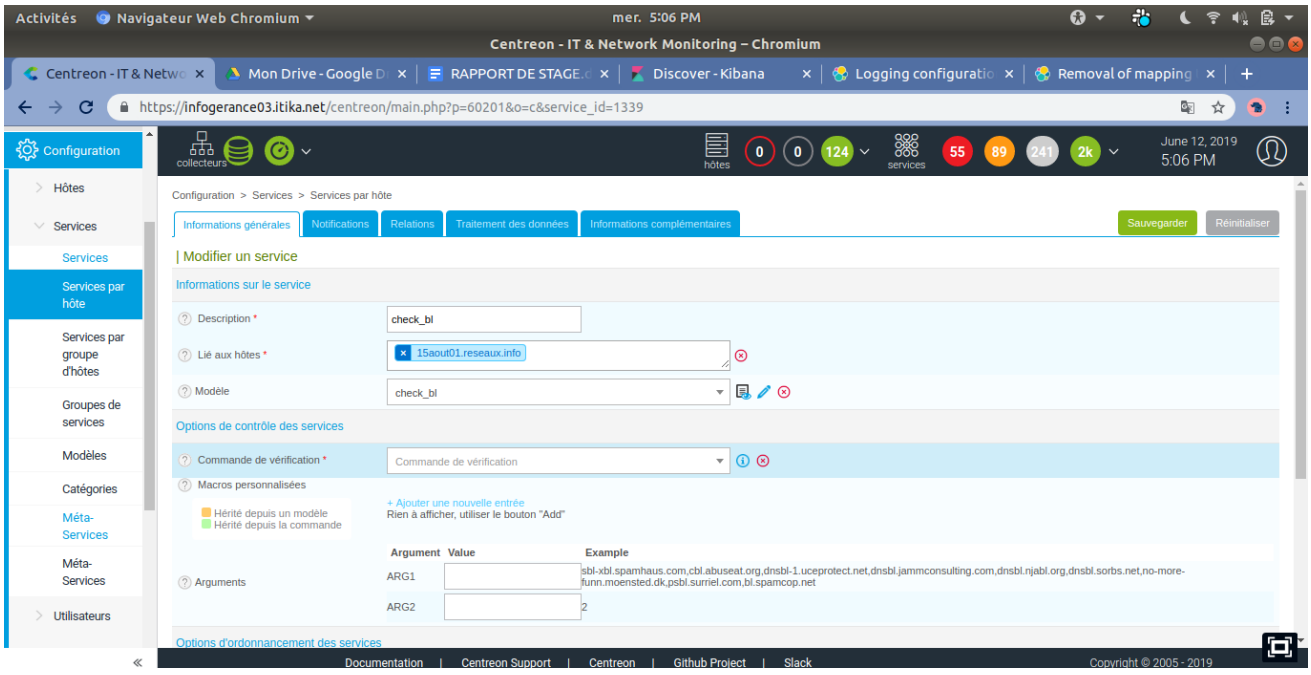


Figure 10 – Création et Implémentation du service sur un hôte

Enfin, on crée enfin notre service qu'on lie à un hôte et dont si besoin on peut affiner les arguments Le service appelle notre modèle qui remonte à son tour à notre ligne de commande l'ordre. Mais il faut également savoir que si vous préciser une première fois les arguments dans le modèle puis que vous les changer dans le menu service, c'est la modification étant la plus proche du résultat final qui prendra le dessus.

4. HISTORISER UN BASH

4.1 Qu'est-ce qu'un Bash ?

Dans la suite de Centreon, je me suis penché sur un autre projet qui vise à avoir une supervision le plus poussée possible. L'idée principale de ce projet fut de savoir quelles commandes sont exécutés dans un BASH et en avoir une trace instantanée.

Tout d'abord un BASH, acronyme de Bourne Again SHell, est une variante d'un SHELL, ce dernier est un programme ayant pour fonction d'assurer l'interface entre l'utilisateur et le système Linux. C'est un interpréteur de commandes. Un SHELL, que l'on peut également appelée communément "terminal", est la méthode la plus courante de gestion des serveurs Linux. Les interpréteurs de commandes disponibles en environnement Unix et Linux ont en commun les fonctionnalités suivantes.

- Ils proposent un jeu de caractères spéciaux permettant de déclencher des actions particulières.
- Ils possèdent des commandes internes et des mots clés parmi lesquels certains sont utilisés pour faire de la programmation
- Ils utilisent des fichiers d'initialisation permettant à un utilisateur de paramétrer son environnement de travail.

Donc chaque Shell propose ses propres caractères spéciaux, commandes internes, mots clés et fichiers de paramétrage. Heureusement, les interpréteurs les plus utilisés actuellement dérivent tous du Shell et ont, par conséquent, un certain nombre de fonctionnalités en commun. A partir de maintenant nous utiliserons le mot Shell comme l'équivalent de Bash.

4.2 La solution proposée

4.2.1 Le daemon auditd

Pour réaliser tout ceci l'entreprise m'a mis à disposition une machine réservée à la recherche et développement. J'ai ensuite puis me connecter au serveur en SSH* et avec un port spécifique pour plus de sécurité.

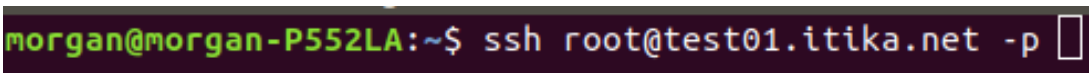


Figure 11 – Commande connexion SSH

J’ai tout d’abord guidé mes recherches sur le daemon* auditd. Ce dernier est l’élément principal du cadre dans l’espace utilisateur. Il est responsable de la réception des événements d’audit envoyés par le noyau et du stockage de ces derniers sur le système de fichiers. Il permet donc de reporter les moindres faits et gestes. Mais je me suis très rapidement heurté à un problème, le daemon ne voulait pas se lancer. Et ce dysfonctionnement est causée par le système de conteneurisation de LXC.

Pour résumer LXC (de l’anglais Linux Containers) est un système de virtualisation, utilisant l'isolation comme méthode de cloisonnement au niveau du système d'exploitation. Il est utilisé pour faire fonctionner des environnements Linux isolés les uns des autres dans des conteneurs partageant le même noyau et une plus ou moins grande partie du système hôte. Le conteneur apporte une virtualisation de l'environnement d'exécution (processeur, mémoire vive, réseau, système de fichier…) et non pas de la machine. Pour cette raison, on parle de « conteneur » et non de « machine virtuelle ». Ce système de cloisonnement empêche auditd de parcours le système librement.

4.2.2 La commande History

Je me suis donc tourné vers la commande history. En quelques mots, cette commande répertorie les commandes entrées dans un bash par un utilisateur dans un fichier de log qui sauvegarder sur la machine. La répertoriage se fait dès la déconnexion de l’utilisateur

La problématique fut que lorsque la session de l’utilisateur est close par la machine à cause d’un délai d’inactivité dès lors la commande history n’enregistre pas les commandes de la session courante et tout est perdu. De plus pour des raisons de facilité il a fallu un programme le plus court et simple possible pour le mettre en place sur toutes les machines

On m’a alors demandé de trouver une solution à ce problème très dérangeant pour les administrateurs réseaux.

J’ai tout d’abord consulté le la documentation de la commande et après de nombreux test il s’est avéré possible et j’en suis arrivé à cette ligne.

```
PROMPT_COMMAND="history -a;history -n;history -r;$PROMPT_COMMAND"
```

Figure 12 - Options de la commande history

Le -a de la commande history permet d’ajouter les nouvelles lignes d'historique (lignes d'historique entrées depuis début de la session Bash en cours) dans le fichier d’historique.

Le -n va ajouter les lignes d'historique non lues dans le fichier d'historique à la liste de l'historique en cours. Ce sont des lignes annexées à l’historique depuis le début de la session Bash en cours.

Et pour finir le - r lit le fichier d'historique actuel et ajoute son contenu à la liste d'historique.

La commande **PROMPT_** a exactement les mêmes propriétés que celle de la calculatrice, c’est-à-dire que cette instruction permet d’écrire la ligne sans notre intervention.

On pouvait maintenant avoir son historique de commande sans craindre une déconnection inopinée mais pour une personne. Or les les administrateurs se retrouvent souvent à deux sur le même serveur et en même temps. Il a alors fallu que je synchronise les terminaux entre eux.

Pour cela je me suis tourné vers les options du shell notamment avec la commande shopt qui permet de personnaliser son shell :

```
#Options shell
shopt -s histappend
shopt -s lithist
#Mise à jour d'history en direct
PROMPT_COMMAND="history -a;history -n;history -r;$PROMPT_COMMAND"
```

Figure 13 - Options de la commande shopt

Dans les deux cas, le -s permet tout simplement d'activer l'option qui suit.

Comme ici l'option lithist qui si activé, et que l'option cmdhist est activée, sauvegarde les commandes multi-lignes dans l'historique avec des nouvelles lignes intégrées.

Et l'option histappend elle est définie, la liste d'historique est annexée au fichier nommé par la valeur de la variable HISTFILE lorsque l'interpréteur de commandes sort, plutôt que d'écraser le fichier.

Ces deux lignes en plus tt fonctionnent parfaitement je mets tout cela dans une boucle if avec pour condition de d'exécuter cette commande uniquement si le shell de la machine est un bash. Et voici donc le résultat final :

```
if [ "$SHELL" = "/bin/bash" ]; then
        #Options shell
        shopt -s histappend
        shopt -s lithist
        #Mise à jour d'history en direct
        PROMPT_COMMAND="history -a;history -n;history -r;$PROMPT_COMMAND"
```

Figure 14 - Programme complet

Lorsqu'un shell est lancé il va appeler différent fichier et pour automatiser ces lignes de commandes à chaque démarrage d'un bash, il faut s'intéresser à l'ordre dans lequel sont lancée les fichiers au démarrage de sessions je me suis intéressé au bashrc. Mais intégrer ce fichier au bashrc. Cette solution ne convenait pas car lorsque l'utilisateur passe en "root" c'est à dire en superutilisateur* les commandes entrées n'étaient plus prises en compte par la commande history.

Mon tuteur m'a ensuite expliqué la hiérarchie existante entre les fichiers lors démarrage d'un shell. La hiérarchie étant complexe, on se concentrera sur le /etc/profile de base. Ce fichier démarre en paramétrant des fonctions d'aide et quelques paramètres de base pour bashrc qui lui-même paramètre un terminal. Il spécifie des paramètres d'historique de bash et, pour des raisons de sécurité, il désactive la conservation d'un fichier d'historique permanent pour l'utilisateur *root*. Il paramètre aussi une invite utilisateur par défaut. Il appelle ensuite de petits scripts à finalité unique dans le répertoire /etc/profile.d pour fournir la plupart de l'initialisation.

Puis au /etc/profil.d on a créé un fichier en .sh pour qu'il soit exécuté et qu'il contre la règle de sécurité et nous permette d'avoir les commandes entrées.

ELASTICSEARCH

5.1 Présentation

Pour finir mon stage je me suis positionné sur un projet en Recherche et Développement l'idée principale du projet se concentre sur un serveur contenant logstash pour trier les mails d'alertes des machines du parc informatique et stocker ces données dans elasticsearch puis on visualise le tout via kibana. C'est ce qu'on appelle ELK cela signifie **E**lasticSearch, **L**ogstash et **K**ibana. Il s'agit de coupler les 3 logiciels pour obtenir une **solution** d'analyse de log performante et **complète**. Les outils sont :

- ElasticSearch : est une base de données pour l'indexation et la recherche des données. Il fournit un moteur de recherche distribué et multi-entité à travers une interface REST. C'est un logiciel libre écrit en Java et publié en open source sous licence Apache.

- Logstash : Logstash est un moteur de collecte et de traitement des données via plug-in.
  Les processus du service lisent les données dans la file d'attente, et les traitent à l'aide de l'un des plug-ins filtre configurés, en séquence. Logstash est prêt à l'emploi et est doté de nombreux plug-ins qui ciblent des types de traitements spécifiques. C'est ainsi que les données sont analysées, traitées et enrichies. Dès lors que les données ont été traitées, ils sont envoyés aux plug-ins de sortie appropriés, chargés de formater et de transférer les données (par exemple, à Elasticsearch).

- Kibana : est un dashboard interactif et paramétrable permettant de visualiser les données stockées dans ElasticSearch elle permet la simplification de la lecture des données en les montrant sous forme de graphiques le tout personnalisable à souhait

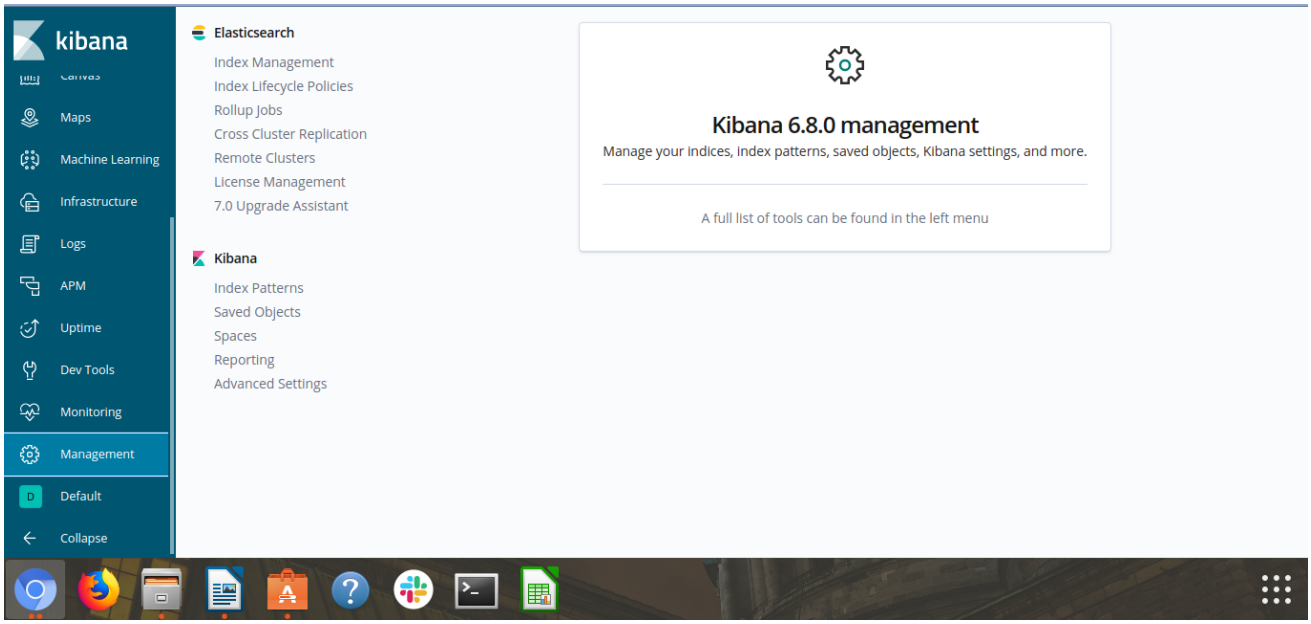


Figure 15 – Interface Kibana

J’ai tout d'abord effectué la mise à jour des ces trois paquets. Etant donné que pour des raisons de sécurité ITIKA n’utilise pas la version la plus récentes, ils m’ont expliquer que pour contrôler la version de mise à jour que sélectionnera apt il suffit de créer un document dans /etc/apt/sources.list.d/ et d’y insérer une ligne de code donné par l’entreprise qui a créé ElasticSearch et de préciser la version voulue. J’ai alors précisé la version 6.x et apt a trouvé la version 6.8.0.

Si j’ai choisi la version 6 ce n’est pas par hasard, c’est surtout var à partir de la version 6 ElasticSearch et Kibana propose un système de login sécurisé via un certificat ssl.

Malheureusement je n’ai pas eu le temps eu le temps d'implémenter ce système car le stage c’est terminé.

6. CONCLUSION

Durant ce stage, j'ai découvert le monde de l'entreprise et son fonctionnement. J'ai compris qu'au sein d'une structure de petite taille il fallait être polyvalent et autonome. ITIKA m'a permis de voir les bases du métier d'administrateur système, ce métier requiert de solides connaissances de Linux pour par exemple maîtriser complètement un programme et les comprendre. Le fossé entre la théorie de l'école et les besoins du monde du travail me conforte dans le fait d'aller en licence ASUR en cursus professionnel pour essayer le concilier les connaissances

7. REMERCIEMENTS

Je tiens à remercier M. Cercy pour tous ses conseils et sa gentillesse durant cette année.

Je tiens également à remercier tout particulièrement M. Longuet, mon tueur (Gérant de la société), pour son accueil, son écoute, sa confiance et tous ses conseils.

Son expérience et sa passion pour son métier m'ont permis d'accroitre, de façon exponentiellement, mes connaissances durant ce stage. Ce fut un réel plaisir d'apprendre avec une personne autant passionnée par son métier.

Enfin, un grand merci à l'ensemble des membres d'ITIKA pour leur accueil, leur gentillesse et leur disponibilité durant ce stage. Ce fut une expérience enrichissante et pleine d'intérêt.

8. GLOSSAIRE

Web : Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites.

PHP : Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.

User-agent : il s'agit d'une chaine de caractère stockée dans le chaque navigateur. Elle contient entre autres les informations de version, du naviguateur (Système d'exploitationn numéro de build...)

MIB : Management Information Base. Il s'agit d'un registre contenant tous les Identifiants d'actions (OID) utilisable par un administrateur via SNMP.

HTTP : L'HyperText Transfer Protocol (Protocole de transfert hypertexte), plus connu sous l'abréviation HTTP, est un protocole de communication client-serveur développé pour le World Wide Web.
Debian : Debian est un système d'exploitation et une distribution de logiciels libres.

CMS : CMS est l'acronyme de content management system, soit, en français, « système de gestion de contenu ». Il s'agit d'un programme informatique utilisant une base de données et permettant de gérer de A à Z l'apparence et le contenu d'un site web.

Secure SHell (**SSH**) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible de voir ce que fait l'utilisateur.

Daemon : Un daemon , parfois traduit par démon, désigne un type de programme informatique, un processus ou un ensemble de processus qui s'exécute en arrière-plan plutôt que sous le contrôle direct d'un utilisateur.

Superutilisateur : Le super-utilisateur (root) dans GNU/Linux est l'utilisateur qui a les droits d'accès administratifs à votre système. Les utilisateurs normaux n'ont pas ces droits pour des raisons de sécurité.

Root : compte super utilisateur des systèmes sous Unix.

Apt : il s'agit avec Aptitude d'un des gestionnaires de paquets de Debian.

9. BIBLIOGRAPHIE - SITOGRAPHIE

https://www.axido.fr/tout-savoir-sur-la-supervision-informatique/

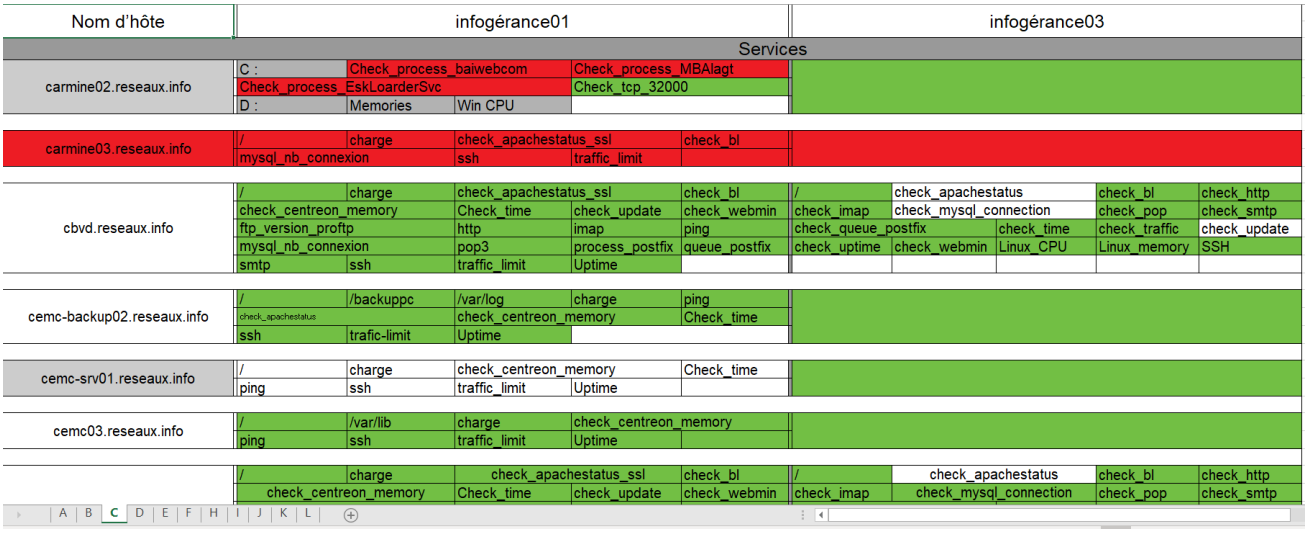
https://linuxcontainers.org/fr/

https://fr.wikipedia.org/wiki/Daemon_(informatique)

https://documentation-fr.centreon.com/docs/centreon/en/2.8.x/administration_guide/02h.html

https://linuxfr.org/news/proxmox-la-vi

10. ANNEXE



Nom d'hôte	infogérance01				infogérance03				
	Services								
carmine02.reseaux.info	C :	Check_process_baiwebcom		Check_process_MBAlagt					
	Check_process_	EskLoarderSvc		Check_tcp_32000					
	D :	Memories	Win CPU						
carmine03.reseaux.info	/	charge	check_apachestatus_ssl	check_bl					
	mysql_nb_connexion		ssh	traffic_limit					
cbvd.reseaux.info	/	charge	check_apachestatus_ssl	check_bl	/	check_apachestatus		check_bl	check_http
	check_centreon_memory		Check_time	check_update	check_webmin	check_imap	check_mysql_connection	check_pop	check_smtp
	ftp_version_proftp		http	imap	ping	check_queue_postfix		check_time	check_traffic
	mysql_nb_connexion		pop3	process_postfix	queue_postfix	check_uptime	check_webmin	Linux_CPU	Linux_memory
	smtp	ssh	traffic_limit	Uptime					
cemc-backup02.reseaux.info	/	/backuppc	/var/log	charge	ping				
	check_apachestatus		check_centreon_memory		Check_time				
	ssh	trafic-limit	Uptime						
cemc-srv01.reseaux.info	/	charge	check_centreon_memory		Check_time				
	ping	ssh	traffic_limit	Uptime					
cemc03.reseaux.info	/	/var/lib	charge	check_centreon_memory					
	ping	ssh	traffic_limit	Uptime					
	/	charge	check_apachestatus_ssl	check_bl	/	check_apachestatus		check_bl	check_http
	check_centreon_memory		Check_time	check_update	check_webmin	check_imap	check_mysql_connection	check_pop	check_smtp

Voici un fragment du tableur que j'ai tenu tenir pour la migration.

Lorsque la case est verte, cela signifie que la migration a réussi.

Si elle est grise c'est que le service a été migré mais ne retourne pas de valeur

En rouge le service n'a pas été migrer car je n'ai pas les compétences pour créer la ligne de commande qu'il faut