Institut Universitaire de Technologie,
Aix-Marseille Université

RAPPORT DE STAGE de fin de deuxième année
Bachelor Universitaire de Technologie
Spécialité Réseaux et Télécommunications
Parcours Cybersécurité

Automatisation de l'infogérance et de la conformité d'un parc informatique

Kénan MEYLAN

ITIKA

Responsable entreprise : Maxime Longuet

Responsable académique : Arnaud Février

2023

Table des matières

1. Introduction

Plonger au cœur du département Administration système d'ITIKA a été une opportunité exceptionnelle qui a renforcé mes compétences et ma passion pour ce domaine en constante évolution. En tant que stagiaire administrateur système durant 10 semaines, j'ai pu participer à de nombreuses missions, et je vous parlerai dans ce rapport du projet qui a été la ligne directrice de mon stage, c'est-à-dire l'automatisation de l'infogérance d'un parc informatique.

L'objectif de ce projet était de mettre à jour le système actuellement utilisé, pour assurer la conformité et la stabilité du parc informatique, ainsi que l'amélioration du service d'infogérance de l'entreprise. Le processus de mise à jour est crucial pour le bon fonctionnement de l'entreprise mais aussi complexe car non sans conséquence car il pourrait faire dysfonctionner l'ensemble de notre parc informatique.
Nous verrons donc à travers ce rapport le travail réalisé pour essayer de répondre à cette problématique et proposer des solutions adaptées et efficaces.

Dans un premier temps, nous développerons sur l'entreprise et ses activités. Puis, nous verrons le cadre technique général du sujet, avant d'aborder la présentation du travail réalisé.
J'ai aussi pu participer à d'autres projets durant mon stage. Je ne pourrais malheureusement pas développer tout ceux-ci de la même manière que celui qui sera principalement présenté dans ce rapport, mais vous pourrez tout de même retrouver une brève description des principaux projets.

2. Présentation de l'entreprise

ITIKA est une société de services spécialisée dans la mise en place de solutions Open Source dans les domaines de l'infrastructure des technologies de l'information, des progiciels métiers et de l'internet.

Membre de l'APRIL, Association pour la Promotion et la Recherche en Informatique Libre, depuis sa création et utilisant des technologies Open Source dans toutes ses composantes, ITIKA est une Entreprise du Numérique Libre.

Les différents services proposés par l'entreprise sont très variés pour répondre à l'ensemble des problématiques réseaux et informatique, en s'appuyant donc uniquement sur des technologies Open Source. En effet, les prestations assurées par l'entreprise peuvent se diviser en 5 axes :

- **Administration système et infogérance :** Spécialisée dans l'Open Source, la société propose la gestion des serveurs d'applications web et mail. ITIKA propose principalement l'infogérance d'instances AWS, Amazon Web Services.

- **Intégration Logiciel Libre :** ITIKA propose aussi une gamme variée de briques Open Source à intégrer dans des infrastructures réseau, telles que des services web, de bases de données, de stockage de fichier NAS, Network Attached Storage ou Stockage en réseau, et des VPN, Virtual Private Network ou Réseau Privé Virtuel. Mais également de nombreux services pour entreprise tels que des ERP, Enterprise Resource Planning ou Progiciel de Gestion Intégré, GED, Gestion Electronique des Documents, et CRM, Customer Relationship Management ou Gestion de la Relation Client.

- **Développement progiciel :** La société propose aussi le développement de progiciels clés en main sous forme d'applications web dédiée au cœur du métier du client.

- **Développement Web :** ITIKA propose d'installer, d'héberger, de modifier et personnaliser les applications WEB.

- **Formations :** ITIKA propose aussi des formations à l'utilisation des produits fournis au clients et à l'utilisation de logiciels et systèmes Open Source.

L'entreprise a été fondée par Maxime Longuet, mon tuteur de stage, en 2004 et est basée dans le quatorzième arrondissement de Marseille, au 3 place de la Rotonde.

Du côté des clients du groupe, les différentes prestations proposées par ITIKA permettent de répondre à la demande de n'importe quel client. Les secteurs d'activités des différentes entreprises clientes sont donc très variés comme par exemple des entreprises de locations de canoë-kayak, des agences de marketing digital ou de croisières.

2.1. Membres de l'entreprise

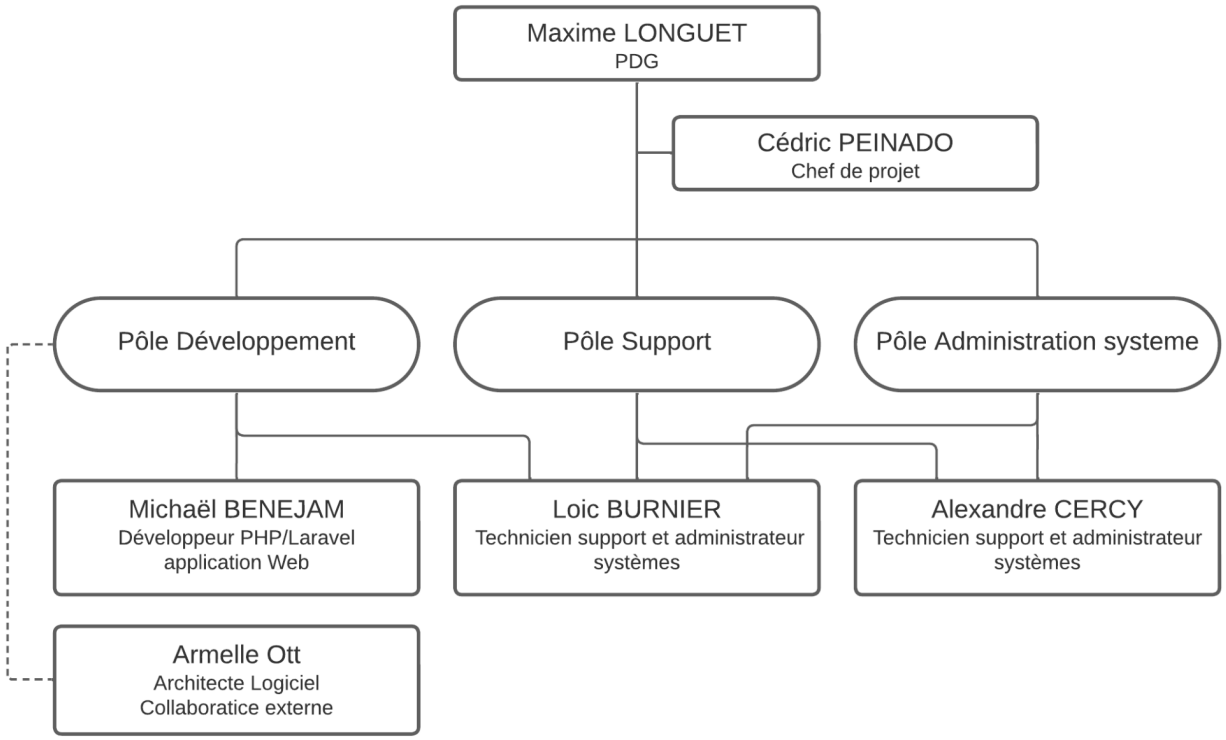


Figure 1 : Organigramme d'ITIKA

Comme vous l'aurez remarqué, l'entreprise est seulement constituée de 5 personnes. Cela influe donc sur le champ d'action de chaque membre de l'entreprise.
En effet, même si des postes sont identiques pour certaines personnes, les missions qui seront attribuées à celles-ci ne sont pas identiques et tout le monde se doit d'être assez polyvalent.

Pour plus de clarté, voici une brève explication :

Maxime Longuet est le Président Directeur Général, c'est-à-dire qu'il dirige chaque pôle que vous pouvez voir sur l'organigramme. Il est aussi développeur et administrateur système.

Cédric Peinado est le chef de projet. Il analyse les besoins des clients, les accompagne et les conseille. La partie Administrative est aussi gérée par M Peinado.

Pour le pôle Administration système ainsi que le pôle Support, nous pouvons retrouver Alexandre Cercy et Loïc Burnier. Ces anciens élèves de notre IUT s'occupent de l'infogérance sous Linux mais également des demandes et problèmes des clients.

Pour le pôle Développement, Loic Burnier s'occupe aussi du développement des logiciels métier pour des clients, généralement en collaboration avec Michaël Benejam, et du développement d'outils en interne selon les besoins de l'entreprise. M Benejam s'occupe lui plus largement du développement de sites et applications web, mais également de la maintenance et de la rénovation de ceux-ci.
Vous aurez aussi pu remarquer Armelle Ott, qui est architecte logicielle mais qui est une collaboratrice externe. En effet, Mme Ott possède sa propre société, Alternative Micro, et n'intervient que sur certains projet de développement, tel que Eridu, un outil de gestion de ventes aux enchères

3. Présentation du cadre général technique

3.1. Contexte théorique

Chez ITIKA, de nombreux serveurs sont configurés chaque jour. Le but est donc d'automatiser toutes les configurations qui sont souvent effectuées sur les serveurs. L'entreprise est partenaire de Rudder, un logiciel libre de configuration automatique de serveurs et de gestion de parc informatique.
Il est compliqué de mettre à jour ce logiciel car il faut veiller à ce que chaque directive de configuration soit compatible avec la nouvelle version du logiciel.
En effet, les agents Rudder installés sur chaque serveur lancent les différentes commandes en tant qu'utilisateur root, c'est-à-dire avec tous les droits sur la machine. Une erreur, selon les directives, pourrait donc être fatale pour l'ensemble du parc informatique.

J'ai donc pu avoir accès à un compte AWS afin de lancer des instances qui m'ont servi de machines virtuelles pour différents tests.
J'ai aussi pu avoir accès à différents serveurs de test de l'entreprise afin de tester différentes directives Rudder sur des serveurs en conditions réelles.

3.2. Objectifs durant le stage

Les objectifs durant mon stage ont donc été clairement définis : Il fallait que j'installe un serveur Rudder en dernière version afin d'étudier et identifier les différentes tâches à faire tout au long du processus de migration.
D'autres objectifs tout au long de ma mission ont pu être définis comme par exemple l'ajout d'une application Web de monitoring, la configuration spécifique à la surveillance de processus et la remontée d'inventaire.

Cette mission est donc évidemment celle qui m'as pris le plus de temps, soit environ 70% de la durée de mon stage, mais j'ai aussi pu travailler sur d'autres projets, qui seront abordés dans une partie ultérieure au sein de ce rapport.

3.3. Cahier des charges pour les différentes missions

Une des compétences nécessaire pour ce stage était l'autonomie, car l'objectif n'était pas de perturber les personnes du service dans leur travail quotidien mais bien d'apporter une plus-value durant ma présence au sein de l'entreprise.
J'ai donc dû faire preuve d'autonomie et cela m'a permis d'apprendre beaucoup de choses. Cela ne m'as, bien sûr, pas empêché de communiquer et de demander de l'aide aux différentes personnes de l'entreprise.

De plus, les compétences requises lors de ces différentes missions sont évidemment une maîtrise des systèmes d'exploitation Linux. Un effort de rédaction et d'orthographe a aussi dû être fait afin de maintenir quotidiennement à jour le wiki de l'entreprise selon les différentes tâches qui doivent être documentées.

4. Présentation du travail réalisé

4.1. Qu’est-ce que Rudder ?

Rudder est une solution Open Source de gestion des configurations, de la conformité et de la sécurité des infrastructures informatiques, créée par la société Normation. Rudder offre une approche centralisée pour contrôler l'état des systèmes, détecter les différences par rapport aux politiques de sécurité et mettre en œuvre des mesures correctives.

Principalement conçu pour les infrastructures à grande échelle, Rudder simplifie la gestion des configurations en permettant aux administrateurs système de définir et d'appliquer des politiques de configuration cohérentes sur l'ensemble de leur infrastructure.

Les agents installés sur les nœuds gérés sont très légers et utilisent 10 à 20 Mo de RAM au maximum et sont extrêmement rapides. En effet, ils sont écrits en C et prennent moins de 10 secondes pour vérifier une centaine de règles. De plus, ils s'exécutent sur presque tous les types d'appareils. Ainsi, il est possible de gérer des serveurs physiques et virtuels dans le centre de données, des instances cloud et des périphériques IoT intégrés de la même manière. L'installation est autonome, via un seul package, et peut se mettre à jour automatiquement pour limiter la charge de gestion des agents.

Chaque nœud est connecté à un serveur central, écrit en Scala, qui définit les configurations à appliquer et collecte les rapports d'application.
Grâce à celui-ci, Rudder propose un ensemble de règles et de modèles préconfigurés, ainsi que la possibilité de définir des politiques de configuration personnalisées pour répondre aux besoins spécifiques des infrastructures.

En ce qui concerne la conformité et l'audit, Rudder permet de vérifier en continu la conformité des systèmes par rapport aux politiques de sécurité établies et de générer des rapports détaillés.

L'automatisation est un aspect clé de Rudder, offrant la possibilité d'automatiser des tâches de gestion des configurations, comme le déploiement de correctifs de sécurité et de mises à jour logicielles.

Le suivi des changements permet aussi de garder une trace de toutes les modifications apportées aux systèmes et de revenir à des états précédents si nécessaire. Rudder offre une interface utilisateur web qui facilite la configuration et l'administration des systèmes. Il nous est donc possible de visualiser facilement l'état de conformité du parc informatique dans sa globalité et de prendre des mesures correctives rapidement.

Enfin, il est important d’expliquer différentes notions propres à Rudder :
Premièrement, Rudder fonctionne grâce à des règles, c'est-à-dire des directives de configurations en fonction des nœuds sur lesquels ces directives vont être appliquées.
Une directive est une instance d’une technique, avec les paramètres à renseigner si cela est nécessaire, qui va être appliquée dans une règle.
Un paramètre est une paire clé/valeur, une technique est un ensemble de méthodes et enfin, une méthode est une action qui va être exécutée.
Une méthode peut renvoyer 3 états : un état de succès, un état d'échec, ou un état dit réparé si un fichier a été modifié au cours de l'exécution de la méthode. Une méthode peut ainsi être conditionnelle, c’est-à-dire exécutée selon une condition pouvant être la valeur de retour d’une méthode précédente. Nous pourrons voir ces différentes notions d’un point de vue pratique dans la suite de ce rapport.

4.2. Mise en place d'une infrastructure simplifié de test de Rudder

Afin de tester la dernière version de Rudder, je ne pouvais évidemment pas utiliser les serveurs en production, destinés aux entreprises clientes. J'ai donc dû mettre en place un parc informatique simplifié, seulement destiné à mes différentes missions. Comme mentionné précédemment, j'ai pu utiliser les services d'AWS afin de créer les machines virtuelles dont j'aurais besoin. Les machines virtuelles, dans AWS, sont appelées des instances. Une instance EC2 est une machine virtuelle qui est créée sur Amazon Elastic Compute Cloud. L'instance est créée à l'aide d'une image AMI, Amazon Machine Image, qui est préconfigurée avec un système d'exploitation.
Différents types d'instances EC2 sont proposés pour s'adapter à des cas d'utilisation spécifiques. Chaque type d'instance EC2 a des caractéristiques spécifiques en termes de ressources, de capacités de calcul, de mémoire, de stockage et de réseau. Il est possible de choisir le type d'instance qui correspond le mieux aux différents besoins spécifiques en fonction des exigences des applications.

Ici, pour notre serveur Rudder, après avoir lu la documentation disponible, nous pouvons voir que s'il y a moins de 50 nœuds, alors les valeurs minimales sont 2 Go de RAM et 2 cœurs pour le processeur. J'ai donc sélectionné une instance de type *t3.small* qui correspond exactement à ces caractéristiques, avec Debian en tant que système d'exploitation.

De plus, j'ai assigné une adresse IP élastique. Dans AWS, une adresse IP élastique est une adresse publique statique qui peut être associée à une instance EC2, contrairement aux adresses IP classiques qui sont attribuées de manière dynamique. Cela permet de garantir une connectivité constante même lorsque l'instance est arrêtée et redémarrée.

Une fois notre instance configurée, j'ai suivi la documentation de Rudder pour installer le serveur. Lorsque Rudder est installé, je me suis rendu, grâce à un navigateur, sur l'adresse de l'instance afin d'effectuer différentes configuration que nous allons voir.

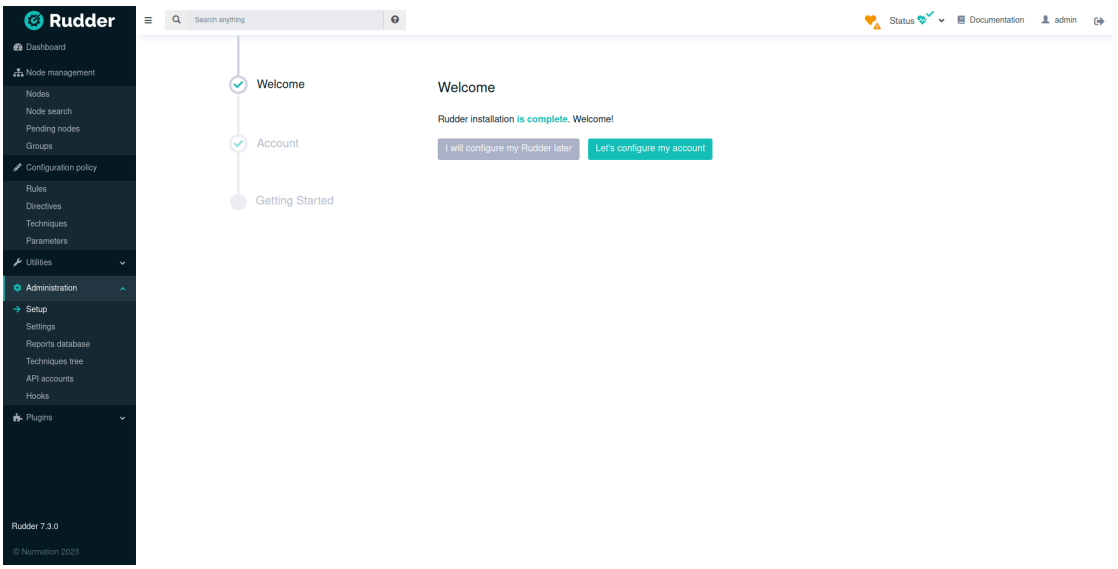


Figure 2 : Interface de Rudder

Il est important de noter qu'AWS fonctionne avec un VPC, Virtual Private Cloud. C'est un service d'AWS qui permet de créer un réseau virtuel isolé dans le cloud. Il est donc nécessaire d'ajouter ce réseau dans Rudder. Pour cela, j'ai dû me rendre dans l'onglet `Administration` → `Settings` → `General` → `Allowed Networks` afin ajouter l'adresse `172.31.0.0/16`.

Il nous est donc désormais possible d'ajouter des serveurs, appelés nœuds dans Rudder. Après avoir installer le paquet de l'agent Rudder, j'ai pu le connecter à notre serveur principal grâce aux commandes suivantes :

```
sudo rudder agent policy-server 172.31.2.225
sudo rudder agent run
```

Sur le serveur dans l'onglet `Node Management` → `Pending Node`, nous pouvons désormais voir le nouveau node que l'on vient de connecter :

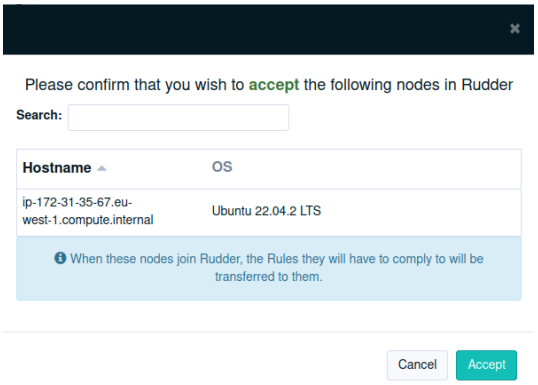


Figure 3 : Ajout d'un noeud sur le serveur Rudder

Nous avons désormais accès aux différents nœuds ajoutés dans l'onglet `Node Management` → `Node`.

4.3. Centralisation des inventaires des serveurs du parc informatique

Un inventaire fait référence à une liste détaillée et organisée de tous les éléments matériels et logiciels qui composent un nœud. L'inventaire est essentiel pour la gestion et la maintenance du parc informatique car cela nous permet d'avoir un suivi constant des différents serveurs à la disposition des clients.

Actuellement, GLPI, Gestionnaire Libre de Parc Informatique, est utilisé en production chez ITIKA, mais la mission qui m'a été assignée a été d'automatiser les envois d'inventaire vers le serveur GLPI grace a Rudder. J'ai donc utilisé un clone du serveur GLPI actuellement en utilisation dans l'entreprise afin d'évoluer dans un environnement de test complètement identique à la situation réelle.

Après avoir étudié la documentation des plugins de Rudder, j'ai pu découvrir le plugin GLPI dans Rudder, officiellement distribué par Normation. Un des prérequis était d'installer le plugin FusionInventory sur notre serveur GLPI. L'installation est simple, il suffit seulement de télécharger l'archive depuis Github puis la décompresser dans le dossier `plugin` de GLPI.
Une fois cela effectué, sur le panel Web, dans l'onglet `Accueil` → `Configuration` → `Plugins`, nous pouvons voir que le plugin FusionInventory a été ajouté à la liste.
Nous avons d'ailleurs directement un pop-up qui apparaît pour nous demander si on veut activer le plugin.

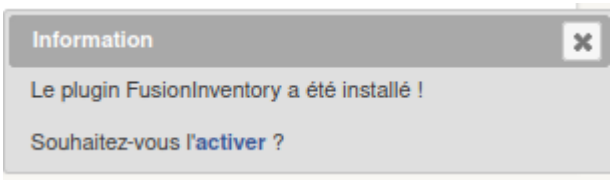


Figure 4 : Activation du plugin Fusion Inventory dans GLPI

Sur le serveur Rudder, j'ai ensuite installé le plugin GLPI qui sert à envoyer les inventaires récupérés par Rudder vers notre serveur GLPI. Pour cela, voici les différentes commandes :

```
sudo nano /opt/rudder/etc/rudder-pkg/rudder-pkg.conf
      [Rudder]
      url = https://download.rudder.io/plugins
      username = itika
      password = <MOT DE PASSE>
sudo rudder package update
sudo rudder package list --all
sudo rudder package install rudder-plugin-glpi
sudo nano /opt/rudder/etc/glpi.conf
      GLPI_FUSION_URL=https://glpi-server-test.itika.net/plugins/fusioninven
       tory/
```

J'ai ensuite pu mettre en place la liaison automatique des inventaires et des ordinateurs dans GLPI. Une des contraintes imposées était qu'il fallait que chaque ordinateur ait précédemment été créé manuellement et qu'il soit lié en fonction de son FQDN, Full Qualified Domain Name ou nom de domaine pleinement qualifié, c'est-à-dire son nom d'hôte complet.

J'ai donc pu utiliser à mes fins de test 3 serveurs que l'on nommera :

- `kenan01.itika.net`
- `kenan02.itika.net`
- `kenan03.itika.net`

Nous pouvons donc créer, dans GLPI dans l'onglet `Accueil` → `Parc` → `Ordinateurs` → `Ajouter`, les ordinateurs portant les mêmes noms que précédemment cités.
J'ai ensuite dû paramétrer les règles de liaison des inventaires récupérés grâce au plugin FusionInventory avec les ordinateurs précédemment créés dans GLPI. Pour cela, dans l'onglet `Administration` → `FusionInventory` → `Rules` → `Equipment import and link rules`, il faut désactiver toutes les règles qui ne serviront pas et activer seulement les règles `Computer name` et `Unknown device import` afin d'éviter les conflits d'importations. Nous pouvons ensuite modifier la règle `Computer name` de la manière suivante :

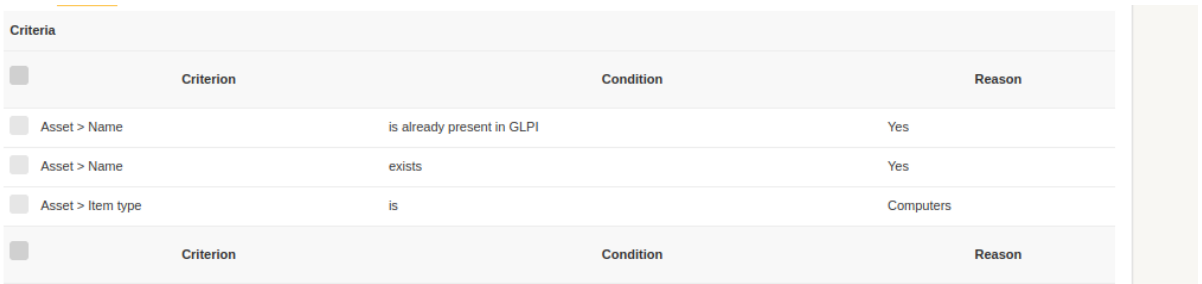


Figure 5 : Règle de liaison "Computer name" dans FusionInventory

Cependant, la variable `Name` utilisée par FusionInventory et GLPI correspond à la version courte du nom d'hôte, soit le nom d'hôte coupé au premier point. Il est important de comprendre que dans l'entreprise Itika, les nombreux serveurs qui sont infogérés appartiennent à des entreprises différentes, et les noms de domaines doivent donc être propres à chaque machine. Les noms courts ne sont pas assez signifiants pour une bonne infogérance.
Dans FusionInventory nous n'avons pas de possibilité de liaison en fonction du FQDN, nous devons donc modifier dans le fichier d'inventaire en XML, Extensible Markup Language ou langage de balisage extensible, la balise `HARDWARE/NAME` en fonction du contenu de la balise `RUDDER/HOSTNAME`.

Pour cela, j'ai modifié la fonction `send_inventory` dans le fichier du plugin GLPI de Rudder `/opt/rudder/bin/glpi-plugin` :

```
send_inventory(){
    FQDN=$(xmlstarlet sel -T -t -v "/REQUEST/CONTENT/RUDDER/HOSTNAME"
"$1")
    xmlstarlet edit -L --update "/REQUEST/CONTENT/HARDWARE/NAME"
--value "${FQDN}" "$1"
    curl -w "%{http_code}" -H "Content-Type: Application/x-compress"
-k -s --data @"$1" "${GLPI_FUSION_URL}" || echo "Error sending $1"
}
```

Notons qu'il est nécessaire d'installer le paquet `xmlstarlet` pour un bon fonctionnement du script.

Ainsi, nous pouvons forcer, depuis les différents noeuds, à envoyer leur inventaire au serveur Rudder:

```
sudo rudder agent inventory
```

Nous pouvons ensuite envoyer tous les inventaires Rudder à notre serveur GLPI :

```
sudo /opt/rudder/bin/glpi-plugin send-all
```

Nous pouvons voir dans `Accueil` → `Administration` → `FusionInventory` → `Agent` que les inventaires ont automatiquement été liés à des ordinateurs que nous avons préalablement créés en fonction du FQDN dans l'inventaire.

Name	Entity	Last contact	locked	Device_id	Computer link	Version	Token
kenan01.itika.net-2023-05-04-14-55-17	Entité racine	09-05-2023 14:13	No	kenan01.itika.net-2023-05-04-14-55-17	kenan01.itika.net	**INVENTORY** : v2.4.3	
kenan02.itika.net-2023-05-04-14-55-17	Entité racine	09-05-2023 14:13	No	kenan02.itika.net-2023-05-04-14-55-17	kenan02.itika.net	**INVENTORY** : v2.4.3	
kenan03.itika.net-2023-05-04-14-55-17	Entité racine	09-05-2023 14:13	No	kenan03.itika.net-2023-05-04-14-55-17	kenan03.itika.net	**INVENTORY** : v2.4.3	
Name	Entity	Last contact	locked	Device_id	Computer link	Version	Token

Figure 6 : Liaison des différents inventaires et des ordinateurs

En revanche, une fois que nous avons déployé cette solution dans un environnement de production, nous nous sommes rendus compte que les inventaires étaient remis à zéro à chaque fois. Cela a engendré une énorme quantité de lignes dans notre historique car chaque logiciel était noté comme désinstaller puis réinstaller.
Après avoir discuté avec David Durieux, principal développeur du plugin FusionInventory, j'ai pu effectuer la mise à jour de GLPI en version 9.5.13, qui est la dernière version compatible avec notre serveur. Ensuite, j'ai pu mettre à jour les différents plugins.
J'ai finalement dû effectuer un nettoyage de la base de données et plus particulièrement de la table `glpi logs grâce` à la commande suivante :

```
DELETE FROM `glpi_logs` WHERE `user_name` = "Plugin_FusionInventory";
```

Ensuite, il a fallu envoyer une nouvelle fois la liste de tous les inventaires de notre serveur Rudder.

Grâce à cela, l'entreprise dispose désormais d'un système de centralisation des inventaires des serveurs du parc informatique. Cela est une grande plus-value dans le domaine de l'infogérance car GLPI et FusionInventory nous permettent désormais de savoir, par exemple, quand et quels logiciels ont été mis à jour et d'avoir un suivi des différentes améliorations matérielles qui ont été faites.
Si un problème survient, nous avons désormais un accès rapide aux différentes informations dont nous avons besoin. Cela permet désormais d'être opérationnel plus rapidement en cas de panne afin de minimiser la durée dans laquelle le serveur ne sera plus joignable.

4.4. Installation d'outils de monitoring sur les différents nœuds

Un autre besoin de l'entreprise que nous avons pu évoquer au cours de mon stage était la surveillance des performances système en temps réel. J'ai donc été chargé de déployer une solution de monitoring automatisée avec Rudder.
Le monitoring est un processus de surveillance et d'évaluation continu. Pour cela, nous utiliserons Netdata offre une interface conviviale qui permet de visualiser les métriques système en temps réel. Il collecte des données sur divers aspects tels que l'utilisation du processeur, la mémoire, le réseau et le stockage. Les données sont ensuite stockées afin de pouvoir avoir une visualisation globale.

J'ai donc pu mettre en place une technique pour l'installation et la configuration de Netdata. Comme mentionné précédemment, une technique est une suite de méthode exécutable selon des conditions, et nous allons voir cela sur l'image ci dessous :

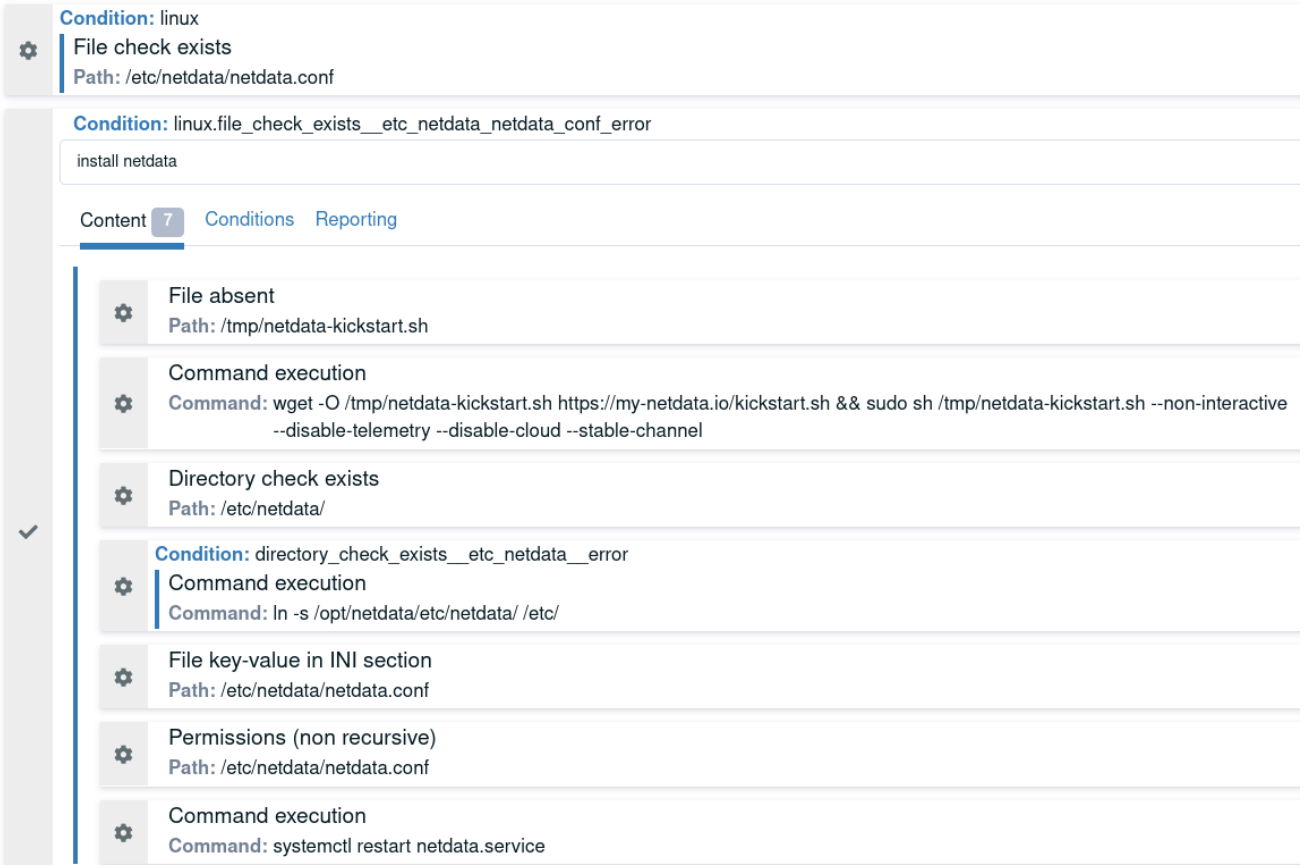


Figure 7 : Liaison des différents inventaires et des ordinateurs

Tout d'abord, si la directive est exécutée sur un agent Linux, alors Rudder vérifie la présence du fichier `netdata.conf`. En effet, je n'ai pas pu utiliser la directive `Packages check installed`, car Netdata peut être installé avec un paquet seulement sur quelques distributions. Or, Netdata peut être installé sur un panel de distribution plus grand grâce à des versions disponibles sur Github.
Cette installation est donc possible grâce au script que nous supprimons pour éviter tout conflit de permissions, que nous téléchargeons puis que nous exécutons avec différents paramètres propre à l'exécution du script.
Ensuite, si le dossier `/etc/netdata` existe, c'est que Netdata a été installé depuis les paquets, sinon c'est qu'il a été installé depuis Github et nous pouvons faire un lien symbolique depuis `/opt/netdata/etc/netdata`.
Rudder modifie ensuite le fichier de configuration de Netdata de sorte à changer le port par défaut grâce à la directive propre au fichier INI. Sa structure consiste en des sections délimitées par des crochets, avec des paires clé-valeur spécifiant les options à l'intérieur de chaque section.

En d'autres termes, cette directive écrit dans le fichier la section `web` suivante :

```
[web]
default port = 22445
```

Rudder assigne ensuite au fichier les permissions conformes puis redémarre le service Rudder.
Une fois notre technique sauvegardée, nous devons en faire une instance, c'est-à-dire créer une directive que nous allons assigner à `Global configuration for all nodes` afin que tous les nœuds connectés à notre serveur Rudder utilisent la technique.

Une fois que nous avons sauvegardé notre directive, et que nous avons patienté pour que les directives soient mises à jour sur les agents, ou après avoir lancé manuellement la commande `sudo rudder agent run -u` sur nos agents, nous pouvons voir que nous avons bien accès à notre panel netdata sur le port 22445.

4.5. Monitoring des processus PHP FPM

Un avantage de Netdata est sa modularité grâce à son ajout de plugin. Un des besoins de l'entreprise qui a pu être développé grâce au déploiement de Netdata était la centralisation des différentes pages de monitoring des processus PHP FPM, FastCGI Process Manager ou gestionnaire de processus FastCGI. En effet, de nombreux serveurs possèdent plusieurs sites étant configurés avec PHP FPM.

J'ai pu utiliser un serveur appelé `dev02.itika.net`, qui est un serveur de développement au sein de l'entreprise. En effet, sur le serveur dev02, il y a déjà de nombreux sites, ainsi que plusieurs versions de PHP FPM et plusieurs processus par version. Cela a pu m'être utile pour tester notre solution de monitoring dans un environnement plus proche des serveurs de production et s'assurer qu'elle fonctionne quelque soit l'environnement, les versions, et les configurations.

De plus, sur l'intégralité de nos serveurs, les différents sites et configurations des processus PHP FPM sont gérés depuis Virtualmin, ce qui nous permet de paramétrer directement les prochaines configurations. En revanche, nous ne nous pencherons pas sur la configuration de Virtualmin dans ce rapport. Les configurations déjà existantes doivent être modifiées et j'utiliserai donc celles-ci.
Pour cela, nous devons dans chaque fichier de configuration d'un processus du dossier `fpm/pool.d` des différentes versions de PHP ajouter la ligne `pm.status_path = /fpm-status` nous permettant d'activer la page de status du processus.

Ensuite, dans chaque configuration Apache des sites utilisant un processus PHP FPM, j'ai du rajouter la ligne `ProxyPass /fpm-status fcgi://localhost:<PORT>/fpm-status` afin de rediriger toutes les requêtes vers `/fpm-status`, soit rediriger vers la page de statut que nous avons précédemment activée. Notons aussi que `<PORT>` est la valeur du port utilisé dans le même fichier de configuration.

Une fois nos différentes pages de statut activées, j'ai pu automatiser le monitoring de celles-ci dans Netdata grâce à Rudder. Dans une première partie, j'ai pu avoir besoin d'un script qui renvoie un JSON, JavaScript Object Notation, que nous allons traiter par la suite dans Rudder.

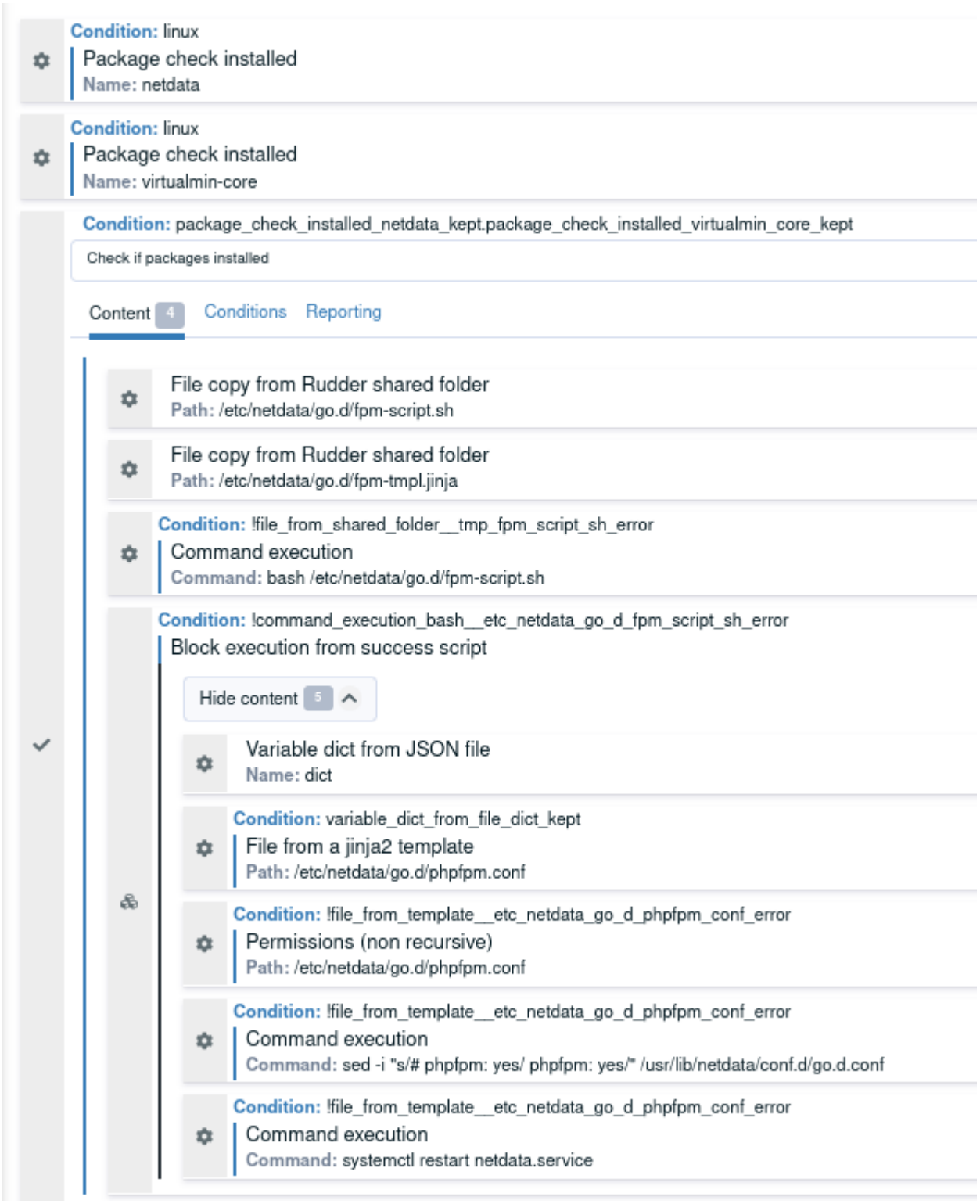
Pour cela, j'ai pu écrire le suivant qui renvoie un code de retour 1, si le JSON est identique, car cela permet de renvoyer une erreur de la méthode dans Rudder et utiliser cette condition pour ne pas redémarrer Netdata à chaque fois que la directive est exécutée :

```
#!/bin/bash
list_vmin=($(virtualmin list-domains --php-mode fpm --multiline |
grep -e "Username:" -e "URL:"  | cut -d ":" -f 2-))

declare -A dict
touch /etc/netdata/go.d/fpm.json

for (( el=0; el<"${#list_vmin[@]}"/2; el++ ));
do
 dict[${list_vmin[$(($el*2))]}]=${list_vmin[$(($el*2+1))]}
done

data='{ "fpm":[] }'

for i in "${!dict[@]}"
do
 data=$(echo $data | jq ".fpm += [{\"domain\": \"${i}\",
\"username\": \"${dict[$i]}\" }]")

done

echo "$data" > /etc/netdata/go.d/fpm-test.json
if ! cmp -s /etc/netdata/go.d/fpm.json
/etc/netdata/go.d/fpm-test.json
then
 mv /etc/netdata/go.d/fpm-test.json /etc/netdata/go.d/fpm.json
else
 exit 1
fi
```

Il est important de mettre ce script dans le dossier `/var/rudder/configuration-repository/shared-files/` car grâce à une méthode Rudder, il nous est possible de transférer des fichiers du serveur au nœud depuis ce dossier.
De plus, Rudder nous permet d'éditer des fichiers en fonction d'une template Jinja2. Dans le même dossier que le script, nous allons donc créer le fichier `fmp-tmpl.jinja` dans lequel j'ai écrit le template dont Rudder va se servir pour créer le fichier de configuration pour Netdata :

```
jobs :
{% for item in vars.vmin.dict['fpm'] %}
  - name : {{item['username'] | replace (".","-") }}
    url  : {{item['domain']}}fpm-status?full&json
{% endfor %}
```

J’ai ensuite pu mettre en place la technique Rudder ci-dessous:



Grâce à cette technique, si nous sommes sur un agent Linux, nous vérifions premièrement que Netdata et Virtualmin soient installés. Dans ce cas là, nous allons exécuter le bloc de méthodes suivant :
Nous copions le script et le template Jinja2 depuis le dossier `/var/rudder/configuration-repository/shared-files` sur le serveur Rudder vers l’agent dans le dossier `/etc/netdata/go.d/`
Nous exécutons ensuite le script si la méthode de copie ne renvoie pas d’erreur.

Ensuite, si le script ne renvoie pas d'erreur, un nouveau bloc est exécuté :

Rudder récupère le JSON dans la variable `vmin.dict`, crée le fichier de configuration `/etc/netdata/go.d/phpfpm.conf` grâce à notre template Jinja2, met les bonnes permissions à ce fichier, décommente la ligne dans le fichier de configuration `/usr/lib/netdata/conf.d/go.d.conf`.
On redémarre ensuite le service pour que les changements soient appliqués. Notons aussi qu'une fois cette technique sauvegardée, nous devons en faire une directive sur le groupe de nœuds souhaité.

Une fois l'exécution des directives relancée manuellement grâce à la commande `rudder agent run -u` nous pouvons vérifier le contenu du fichier de configuration.

```
cat /etc/netdata/go.d/phpfpm.conf
    jobs :
      - name : michael-abogest-dev02-itika-net
        url  : http://michael.abogest.dev02.itika.net/fpm-status?full&json

      - name : maxime-abogest-dev02-itika-net
        url  : http://maxime.abogest.dev02.itika.net/fpm-status?full&json

      - name : loic-abogest-dev02-itika-net
        url  : http://loic.abogest.dev02.itika.net/fpm-status?full&json

      - name : test-abogest-dev02-itika-net
        url  : http://test.abogest.dev02.itika.net/fpm-status?full&json

      - name : maxime-wktools-dev02-itika-net
        url  : http://maxime.wktools.dev02.itika.net/fpm-status?full&json
```

Dans Netdata, nous pouvons retrouver dans le panneau de droite les liens vers les sections de monitoring. Nous avons désormais accès à nos différentes métriques de manière automatique lorsqu'un nouveau site est déployé grâce a Rudder.

4.6. Mise à jour officielle du parc informatique

Vers la fin de mon stage, j'ai eu la chance de participer à une visioconférence avec l'ensemble de l'équipe Infrastructure d'ITIKA ainsi que de Benoit Peccatte, Architecte Logiciel chez Normation. Lors de cette visioconférence, Mr Peccatte était chargé de la mise à jour de Rudder sur l'ensemble du parc informatique. Plusieurs étapes ont clairement pu être définies lors du processus de migration vers une version plus récente. Afin de ne pas compromettre les différents serveurs en production, nous avons dû mettre en place une démarche claire et construite. Cela nous aurait aussi permis de pouvoir corriger un problème dans un temps très réduit, qui pourrait survenir sur les différents serveurs malgré nos précautions.

En collaboration avec le prestataire, nous avons donc tout d'abord corrigé les problèmes existant avant tout processus de mise à jour. En effet, certains nœuds n'étaient plus connectés au serveur principal à cause d'un problème de configuration d'un relais. Il était plus pertinent de corriger ce problème dans un premier temps pour être sûr qu'il ne soit pas apparu à cause de la mise à jour.

Ensuite, nous avons mis à jour toutes les directives obsolètes sur notre serveur afin qu'elles puissent continuer de fonctionner après la mise à jour du serveur. Si nous n'avions pas fait cela, il est possible que les différentes directives obsolètes ne soient plus supportées dans les versions plus récentes.

Une fois ces étapes terminées, nous avons pu mettre à jour notre serveur Rudder. Pour cela, nous avons dû suivre une démarche structurée.
En effet, pour éviter les problèmes de compatibilité, nous avons fait une mise à jour progressive. Notre serveur était en version 6.2 et nous avons premièrement mis à jour vers la version 7.2.

Ensuite, nous avons corrigé la configuration d'Apache sur les certificats SSL car dans les nouvelles versions, la communication avec les nœuds fonctionne avec un certificat différent de celui utilisé pour accéder en HTTPS au nom de domaine depuis un navigateur.

Nous avons aussi mis à jour les différents plugins utilisés tel que le plugin de détection des serveurs vulnérables en fonction des CVE, Common Vulnerabilities and Exposures ou vulnérabilités et expositions communes. Nous avons aussi mis à jour les différents relais dans la version stable la plus récente.

Enfin, une fois confirmé le bon fonctionnement de toutes les parties que nous avons mises à jour, nous avons mis à jour le serveur principal dans la dernière version stable, c'est-à-dire la 7.3, mais qui n'était pas directement compatible avec la 6.2. En effet, de nombreux composants avaient été modifiés voire supprimés au cours des différentes versions.

Nous avons ensuite mis en place une technique afin de mettre à jour les agents sur tous les nœuds.
Une fois que notre mise à jour était effectuée et fonctionnelle, nous avons pu avoir une présentation des nouvelles fonctionnalités disponibles, telles que les paramètres globaux, qui sont les paramètres de chaque agent paramétrable depuis le serveur. Nous avons par exemple utilisé cette nouvelle fonctionnalité pour diminuer la verbosité de la journalisation dans les journaux systèmes Syslog.

Mr Peccatte a aussi pu nous renseigner sur les axes d'améliorations de notre serveur Rudder, tel que la migration des méthodes de synchronisation avec les relais. En effet, entre le serveur et un relais, CFEngine est utilisé par défaut. Malgré les nombreuses améliorations depuis la version 6.2, la méthode de synchronisation la plus optimisée est d'utiliser RSync à la place.

Ma participation à la visioconférence avec l'équipe d'Infrastructure d'ITIKA et Benoît Peccatte m'a permis de comprendre l'importance d'une approche structurée lors de la mise à jour de Rudder sur l'ensemble d'un parc informatique et de nombreux serveurs en production. Les bénéfices personnels que j'ai tirés de cette expérience sont multiples.
Tout d'abord, j'ai pu observer la démarche effectuée pour résoudre des problèmes existants, ce qui a renforcé ma capacité à identifier et corriger les erreurs de configuration.
De plus, en mettant à jour les directives obsolètes sur notre serveur, j'ai acquis une compréhension plus approfondie de l'importance de maintenir la compatibilité avec les versions plus récentes. La démarche structurée que nous avons adoptée, en effectuant une mise à jour progressive, m'a permis de découvrir les étapes nécessaires pour éviter les problèmes de compatibilité, et garantir un bon fonctionnement continu et sur le long terme.
Dans l'ensemble, cette expérience a renforcé mes compétences techniques, ma compréhension des processus de mise à jour et m'a ouvert de nouvelles perspectives pour améliorer les performances et la sécurité des systèmes d'infrastructure.

4.7. Autres missions réalisées

4.7.1. Déploiement d'un gestionnaire de mot de passe

Aujourd'hui, ITIKA utilise les services de LastPass. La dépendance externe de ce service est trop forte et la société préfère internaliser ce service sur une solution libre et communautaire. L'utilisation des services SaaS, Software as a Service ou logiciel en tant que service, aussi connue que Lastpass pose aussi le problème d'être une cible de choix pour les hackers comme le démontrent les derniers incidents de sécurité subis par LastPass. De plus, internaliser le service permet l'accès à celui-ci grâce au VPN de la société.

La mission qui m'a été attribuée a donc été d'étudier les différentes solutions Open Source disponibles, de faire une présentation de la solution puis de déployer en interne la solution.

De plus, cette solution a pu être proposée à différents clients et pour cela, il fallait qu'elle soit maîtrisée en interne dans un premier temps.

Durant cette mission, j'ai donc dû, :

- ➢ Faire une étude préalable des différentes solutions en fonction des besoins de l'entreprise.
- ➢ Connaître les prérequis qui nous permettront de choisir la solution d'hébergement appropriée.
- ➢ Vérifier si nous pouvons facilement migrer ou importer nos mots de passe depuis LastPass.
- ➢ Déterminer si les fonctionnalités de partage entre les comptes utilisateurs répondent à nos besoins.
- ➢ Tester le bon fonctionnement de la solution choisie sous Linux, MacOS et iPhone.
- ➢ Étant donné le mode de fonctionnement, j'ai aussi dû réfléchir à une solution de secours en cas de défaillance du service d'hébergement.

Après avoir étudié les nombreuses solutions d'hébergement de mot de passe, mon maître de stage et moi en sommes venus à une conclusion : VaultWarden est la solution la plus cohérente pour nos besoins.

En effet, nous avions premièrement étudié la solution Bitwarden, mais en raison des différents prérequis (développé en .NET, Microsoft Server SQL, besoin de beaucoup de RAM, besoin de plus de place initiale, ...), nous ne pouvons nous tourner vers cette solution.

VaultWarden a donc été la plus adaptée, car reprend exactement les mêmes fonctionnalités que BitWarden, mais en utilisant un programme plus léger que ce soit en RAM ou sur le disque, écrit en Rust, utilisant une base de données de notre choix. Vault Warden est d'ailleurs très bien documenté, mis à jour fréquemment, et déployable avec ou sans Docker.

De plus, côté client, VaultWarden utilise les applications de BitWarden, ce qui simplifie l'installation depuis les postes utilisateurs.

Une fois les différentes phases de test concluantes, j'ai pu déployer VaultWarden dans un environnement de production dans l'entreprise, effectuer différentes opérations de sécurisation du serveurs telles que la mise en place d'un firewall, un audit de sécurité système, et la mise en place de différents scanners.

Grâce à ce projet, j'ai donc pu développer mes compétences sur Docker et l'outil Compose. J'ai aussi pu découvrir le firewall CSF, ConfigServer Firewall, et LFD, Login Failure Daemon, pour la protection contre les attaques Brute Force ainsi que de différents outils tels que RKHunter pour la détection des rootkits, Nikto pour la détection des vulnérabilités du serveur web et Lynis pour les audits de sécurité système.

4.7.2. Etude de mise à jour du logiciel de supervision du parc informatique

En raison de la récente annonce de Centreon du 25 mai prévoyant la fin de Centreon sur CentOS 7, j'ai aussi dû au cours de mon stage me renseigner sur ce logiciel et tester la migration vers Debian 11. Centreon est un outil de supervision et de monitoring fonctionnant sur un principe de « check » à l'aide du protocole SNMP.

Afin d'effectuer une prochaine migration, nous avons eu besoin de savoir si les fonctionnalités que nous utilisons sur notre Centreon actuel fonctionne dans la nouvelle version. En effet, nous utilisons la version communautaire des plugins disponible avec les fonctionnalités suivantes :

- **storage** : Pour avoir l'espace disque des serveurs
- **load** : Pour la charge serveur
- **time** : Pour savoir si le serveur est à la bonne heure
- **interfaces** : Pour le trafic total d'une interface réseau
- **uptime** : Pour la durée de fonctionnement depuis son dernier allumage
- **cpu** : Pour l'utilisation du processeur
- **memory** : Pour les différentes valeur d'utilisation de la mémoire RAM
- **numeric-value** : Pour récupérer la valeur de la queue Postfix des mails en attente

J'ai pu utiliser deux serveurs de l'entreprise afin d'avoir des serveurs en conditions d'utilisations réelles, supervisés depuis le Centreon actuel. J'ai ainsi pu récupérer les commandes des plugins de la version actuelle, installer Centreon sur une machine en Debian 11 et adapter les commandes sur cette nouvelle version.

4.7.3. Migration du serveur de mail

L'utilisation d'un serveur de messagerie en entreprise est essentielle pour faciliter la communication interne et externe. Il permet d'envoyer, recevoir et gérer les e-mails de manière centralisée, offrant un contrôle sur les données sensibles, une gestion efficace des flux de communication et la possibilité de personnaliser les adresses e-mail pour renforcer l'identité de l'entreprise.
Actuellement en place dans l'entreprise, Zimbra est le logiciel utilisé par tous les employés en tant que boîte mail professionnelle. La dernière mise à jour remonte à la sortie de Zimbra 8, soit Septembre 2012. La mise à jour vers une version plus récente était possible, mais Zimbra est en déclin et la pérennité du produit n'est plus assurée.
C'est pourquoi, j'ai été chargé de l'installation de Carbonio, une amélioration de Zimbra par une entreprise différente, afin de pouvoir faire un Proof of Concept de cette solution.
Un proof of concept, ou PoC, est une démonstration pratique pour valider la faisabilité d'une idée, d'un concept ou d'une technologie. Il s'agit d'un prototype fonctionnel à petite échelle qui vise à prouver la viabilité d'un projet avant d'investir davantage de ressources. Le PoC permet de tester les fonctionnalités clés, d'évaluer les performances et de démontrer la valeur potentielle d'une solution.
Ainsi, j'ai pu étudier le fonctionnement complet d'un serveur de mail. J'ai donc pu comprendre le fonctionnement des enregistrements DNS MX, des relais SMTP, et de synchronisation IMAP des mails

4.7.4. Migration du Wiki

La solution de systèmes de gestion de contenu de type wiki en fonctionnement dans l'entreprise était vieillissante. Le wiki est utilisé au quotidien afin d'accéder et de partager facilement des informations au sein de l'entreprise, mais n'était pas mis à jour.
Ainsi, durant mon stage j'ai été chargé d'étudier les différentes solutions possibles selon différentes contraintes. En effet, j'ai dû rechercher un wiki qui puisse gérer l'accès aux ressources en fonction de groupes d'utilisateurs, qui puisse prendre en charge la syntaxe Markdown, et facultativement qui puisse stocker directement sur le disque les articles plutôt que dans une base de données.

5. Conclusion

Dans ce rapport, vous aurez donc pu trouver un résumé des dix semaines passées au sein de l'entreprise ITIKA et le bilan des différentes tâches qui m'ont été assignées.
Grâce à celles-ci, j'ai eu l'opportunité de contribuer activement à l'amélioration de l'infrastructure et de la gestion du parc informatique de l'entreprise. Au cours de ces dix semaines, j'ai réalisé plusieurs tâches essentielles qui ont permis de renforcer l'efficacité et la stabilité du système.

La mise en place d'instances de test de Rudder et le déploiement des différents outils de monitoring a pu être un aspect crucial de mon travail.
Grâce à cette surveillance en temps réel, il est désormais possible de détecter et de résoudre rapidement les problèmes de performance et de stabilité. Les diverses autres tâches qui m'ont été assignées ont aussi permis une gestion plus efficace des ressources et l'amélioration de la collaboration au sein de l'équipe.
Les solutions aux différentes missions qui m'ont été assignées sont désormais en activité au sein de l'entreprise soit dans un environnement de test, ou soit seulement au sein de l'entreprise, mais aussi désormais proposées aux clients.

Les différents cahiers des charges qui avaient pu être mis en place ont été respectés et j'ai réussi à mener à bien chaque objectif et missions qui m'ont été assignés.
En plus de la satisfaction personnelle que j'ai pu tirer lors de l'accomplissement de chaque mission, j'ai compris les différents enjeux de celles-ci.

Ce stage a été une expérience très enrichissante qui m'a permis d'apporter une réelle contribution à l'amélioration de l'infrastructure informatique de l'entreprise.
Grâce à la formation que j'ai reçue à l'IUT, j'ai pu mettre en pratique mes connaissances techniques et développer mes compétences en administration système mais aussi en gestion de projet. En effet, les enseignements théoriques et pratiques m'ont préparé de manière adéquate pour aborder les défis rencontrés lors du stage.

Ce stage a renforcé mon intérêt pour le domaine de l'infrastructure informatique et a confirmé mon choix de poursuivre mes études dans ce domaine. J'ai pu constater l'importance d'une infrastructure solide et bien gérée pour le bon fonctionnement d'une entreprise. Cela a influencé positivement mon projet professionnel en me donnant une vision plus concrète de ma future carrière.

6. Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont su donner de leur temps tout au long de mon stage et de la rédaction de ce rapport.

Je tiens tout d'abord à remercier Maxime Longuet, mon encadrant de stage, pour son précieux accompagnement, ses conseils et sa disponibilité tout au long de cette expérience professionnelle. Sa présence m'a permis de découvrir de nouveaux horizons, d'approfondir mes connaissances et de relever de nouveaux défis.

Mes remerciements vont également à toute l'équipe d'ITIKA pour leur chaleureux accueil, leur soutien et leur collaboration durant mon stage. Leur expertise et leurs partages d'expérience m'ont été extrêmement bénéfiques dans l'acquisition de nouvelles compétences.

Je souhaite exprimer ma reconnaissance envers ces collègues de travail, qui ont contribué à créer une atmosphère agréable et motivante au sein de l'équipe. Leur support et leurs échanges ont été une source d'inspiration et d'apprentissage constant.

Enfin, j'aimerais adresser mes sincères remerciements à mon établissement d'enseignement, l'IUT Réseaux et Télécommunications Marseille Luminy, ainsi qu'à mon tuteur académique, Arnaud Février, pour m'avoir donné l'opportunité de réaliser ce stage et d'enrichir mes connaissances. Leurs enseignements ont été un socle solide sur lequel j'ai pu construire mes compétences professionnelles.

Je suis reconnaissant envers toutes les personnes qui ont contribué à la réalisation de ce stage et de ce rapport. Leurs précieux apports ont grandement enrichi mon expérience et mes connaissances.

Merci à tous.

7. Glossaire

Adresse IP élastique : Adresse IP publique qui peut être associée de manière flexible à une instance ou à un service dans le cloud, permettant une connectivité persistante

Apache : Logiciel de serveur web open source permettant de fournir des services d'hébergement de sites web et de servir des pages web aux clients qui y accèdent

Attaque Brute Force : Méthode utilisée pour trouver un mot de passe ou une clé en testant une à une toutes les combinaisons possibles

Audit (sécurité système) : Évaluation méthodique des mesures de sécurité mises en place dans un système informatique afin de détecter les vulnérabilités, d'identifier les éventuelles failles de sécurité et de proposer des recommandations pour renforcer la protection du système contre les menaces potentielles

AWS : Plateforme de services cloud proposée par Amazon

C : Langage de programmation impératif de bas niveau inventé dans les années 70

CentOS : Distribution de système d'exploitation open source basée sur le code source du système d'exploitation Red Hat Enterprise Linux

CFEngine : Outil de gestion de configuration open source permettant d'automatiser et de gérer la configuration et la conformité des systèmes informatiques à grande échelle

CVE : Common Vulnerabilities and Exposures, système de référencement standardisé utilisé pour identifier et suivre les vulnérabilités de sécurité informatique

Directive (Rudder) : Règle de configuration définie pour un ou plusieurs nœuds du système

EC2 : Service d'Amazon Web Services qui fournit des instances virtuelles dans le cloud

Enregistrement DNS MX : Entrée Mail Exchanger dans le système de noms de domaine (DNS) qui spécifie les serveurs de messagerie responsables de recevoir les e-mails

ERP : Enterprise Resource Planning, système intégré de gestion des ressources d'entreprise permettant de planifier et de gérer efficacement les processus commerciaux

Fichier INI : Fichier de configuration utilisé pour stocker des paramètres et des options de configuration structurées sous forme de sections et de clés-valeur

Firewall : Dispositif de sécurité réseau qui surveille et contrôle le trafic entrant et sortant d'un réseau en appliquant des règles de filtrage pour bloquer ou autoriser certaines connexions

FQDN : Fully Qualified Domain Name, nom complet d'un domaine dans le système de noms de domaine

GED : Gestion Électronique de Documents, logiciel permettant de gérer les documents de manière électronique

GLPI : Gestionnaire Libre de Parc Informatique, solution logicielle open source pour la gestion des services informatiques et l'inventaire du parc informatique

Image AMI : Amazon Machine Image, copie statique d'une instance de machine virtuelle dans le service Amazon Elastic Compute Cloud

Inventaire : Liste détaillée et organisée de tous les éléments matériels et logiciels qui composent un serveur ou un ordinateur

Jinja2 : Moteur de modèle Python utilisé pour générer des fichiers de texte en combinant des modèles prédéfinis avec des données dynamiques

JSON : JavaScript Object Notation, format de données textuel dérivé de la notation des objets du langage JavaScript

Markdown : Langage de balisage léger non formaté

Méthode (Rudder) : Instruction à exécuter qui retourne 3 états (succès, erreur, ou réparé) après son exécution

Monitoring : Surveillance informatique permettant de s'assurer de la disponibilité et de la performance de l'ensemble du parc informatique

NAS : Network Attached Storage, serveur de stockage de fichiers en réseau

Paramètre (Rudder) : Paire clé/valeur globale ou locale pouvant être utilisée dans une directive

PHP FPM : Interface permettant la communication entre un serveur Web et PHP basée sur le protocole FastCGI

Proof of Concept : Étude de faisabilité d'un projet à réaliser afin de démontrer la viabilité du projet

Progiciel : Ensemble de logiciels munis d'une documentation, conçus pour répondre à des besoins spécifiques et permettre une utilisation autonome

Relais (Rudder) : Nœuds spéciaux permettant de répartir la charge ou d'interconnecter les réseaux des agents et du serveur Rudder

Relais SMTP : Passerelle de courrier utilisé pour distribuer les mails vers l'extérieur du réseau

Règles (Rudder) : Liaison entre une ou plusieurs directives en fonction d'un ou plusieurs groupes de noeuds

Rootkit : Ensemble de logiciels conçu pour permettre à des cybercriminels de prendre le contrôle d'un serveur

SaaS : Software as a Service, modèle d'exploitation commerciale des logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur

Scala : Langage de programmation à multiples paradigmes désigné pour exprimer des motifs de programmation communs de façon concise, élégante et robuste

Syslog : Protocole définissant un service de journaux d'événements d'un système informatique

Synchronisation IMAP : Protocole qui synchronise en permanence les messages contenus sur le serveur et sur le poste de travail

Technique (Rudder) : Ensemble logique de méthodes exécuté les unes à la suite des autres selon ou non des conditions et des opérateurs logiques

Utilisateur Root : Compte d'administration le plus élevé sur un système Unix/Linux, ayant des privilèges et un accès complets à toutes les fonctionnalités du système

Virtualmin : Panneau de contrôle d'hébergement de domaine et de site Web qui permet de créer et de gérer des domaines, ainsi que de simplifier à la fois l'automatisation et les tâches

VPC : Virtual Private Cloud, service permettant de créer un réseau virtuel isolé dans le cloud

VPN : Virtual Private Network, réseau privé virtuel utilisé pour créer une connexion réseau privée entre des appareils via Internet

XML : Extensible Markup Language, langage de structuration de données, utilisé notamment pour la gestion et l'échange d'informations

8. Références

8.1. Sitographie

"Documentation Rudder", **Normation**, https://docs.rudder.io

"Carbonio CE Documentation", **Zextras**, https://docs.zextras.com/carbonio-ce/html/index.html

"Unofficial Bitwarden compatible server written in Rust, formerly known as bitwarden_rs", **Dani Garcia**, https://github.com/dani-garcia/vaultwarden/wiki

"Documentation Centreon", **Centreon**, https://docs.centreon.com/fr/

"FusionInventory documentation", **Fusion Inventory**, https://documentation.fusioninventory.org/

"GLPI installation", **GLPI Project/Teclib'**, https://glpi-install.readthedocs.io/en/latest/

8.2. Bibliographie

"Postfix La référence", **K. Dent**, Edition O'REILLY, 2004

"UNIX Utilisateur - Seconde Edition", **A. Berlat/J.-F. Bouchaudy/G. Goubet**, Editions Tsoft/Eyrolles, 2003

"LINUX Administration - Quatrième édition - Noyau 2.6", **J.-F. Bouchaudy/G. Goubet**, Editions Tsoft/Eyrolles, 2004

"LINUX Administration - Tome 3 - Sécuriser un serveur Linux", **J.-F. Bouchaudy**, Editions Tsoft/Eyrolles, 2008