



Z.A.C du
Parc des
Tulipes

99 AV Jean-Baptiste Clément,
93430 Villetaneuse

DAMIEN NOISEL

RAPPORT DE MISSIONS D'ALTERNANCE

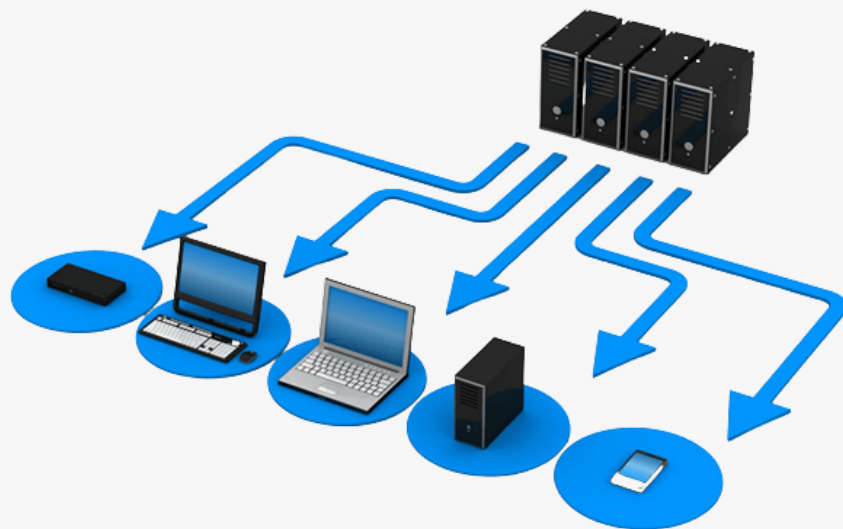


Figure 1: Représentation caricaturée d'une infrastructure

ANNÉE 2021-2022

DATES DE L'ALTERNANCE : 20 SEPTEMBRE 2021 AU 07 SEPTEMBRE 2022

MAÎTRE D'APPRENTISSAGE : MARIOT CÉDRIC
TUTEUR PÉDAGOGIQUE : BRUNET PHILIPPE

Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport d'alternance par des remerciements, à ceux qui m'ont beaucoup appris au cours de ces semaines passées à leur côté et même à ceux qui ont eu la gentillesse de faire de cette expérience un moment profitable.

Je tiens tout d'abord à remercier mon maître de stage, Mr Jérôme MARCHANDIAU, directeur des infrastructures IT¹ opérations sans qui rien n'aurait eu lieu.

Je souhaite ensuite adresser mes remerciements à Mr Cédric Mariot, responsable systèmes et bases de données ainsi qu'à toute son équipe. Ils ont su me faire confiance lors de cette aventure dans le monde professionnel et ont partagé leurs connaissances de manière très pédagogique. Je les remercie aussi pour leur disponibilité et la qualité de leur encadrement en entreprise.

Je voudrais enfin exprimer ma reconnaissance envers toute l'équipe infrastructure qui m'ont accueilli et apporté leur soutien moral et intellectuel tout au long de mon stage. Un grand merci à Mr Thierry LOPES, responsable réseaux et téléphonie ainsi qu'à Mr Laurent NAVARRO, ingénieur administration systèmes et applications pour leur patience et leur pédagogie.

¹ Voir lexique

Resumé

I am a student in a professional degree in computer networks and telecommunications, option Internet of things. This one took place from September 20th, 2021 to September 9th, 2022 at the company Manutan.

My apprenticeship consisted in various missions which had for goal to maintain the good functioning of the used systems, the safety of the data stored on the servers as well as the creation of new systems in a new network. All this in a context of security following a computer attack by ransomware that took place in February 2021.

In this concern of security, many new software and systems appeared within the infrastructure, my missions thus mainly consisted in the installation, the understanding and the maintenance of these new tools.

My first missions allowed me to discover the infrastructure while learning the IT practices to intervene on machines in use. The missions that followed had for main goal to automate the infrastructure through the use of scripting and Ansible, as well as the management of the flows between the different technologies through the use of a new tool named Waarp.

I learned a lot within the company because of the various missions. I also decided after this work-study program that I wanted to focus on infrastructure and Linux systems.

Introduction

Etudiant en licence professionnelle métiers des réseaux informatiques et des télécommunications option internet des objets, mon année s'est effectuée en alternance. Celle-ci s'est déroulée du 20 septembre 2021 au 9 septembre 2022. Au cours de cette alternance au sein de l'équipe système de l'entreprise Manutan, j'ai pu expérimenter le monde de l'entreprise sur un temps significatif ainsi qu'acquérir une expérience en tant que technicien systèmes.

L'entreprise Manutan est le leader européen de vente à distance d'équipements industriels et de bureau auprès des entreprises et des collectivités locales. Ses principaux concurrents sont les entreprises Amazon et Raja.

Mon alternance a consisté en diverses missions qui avaient pour but de maintenir le bon fonctionnement des systèmes utilisés, la sûreté des données stockées sur les serveurs ainsi que la création de nouveaux systèmes dans un nouveau réseau. J'ai eu l'occasion à maintes reprises d'intervenir directement sur les technologies physiques en me rendant, avec Cédric MARIOT, dans les datacenters.

Elle m'a permis de découvrir les différentes situations que pouvait rencontrer un administrateur système ainsi que les différentes façons de réagir aux situations. J'ai aussi pu voir de près les technologies utilisées afin de traiter et gérer les données et les utilisateurs.

En vue de rendre compte de manière fidèle et analytique de l'année passée au sein de la société Manutan, il apparaît logique de présenter à titre préalable l'entreprise ainsi que la culture qui lui est associée. Puis d'envisager le cadre de l'alternance : Quelques mois avant que j'intègre l'entreprise, celle-ci s'est faite cryptolocker des données durant le mois de juillet 2021 ce qui a rendu mon alternance d'autant plus intéressante puisqu'il y avait alors un besoin de sécurisation du système. Enfin, il sera précisé les différentes missions et tâches que j'ai pu effectuer au sein du service infrastructure systèmes et bases de données ainsi que les nombreux apports que j'ai pu en tirer.

Table des matières

| | |
|--|----|
| Remerciements | 2 |
| Resumé | 3 |
| Introduction | 4 |
| Table des matières | 5 |
| I. Manutan..... | 7 |
| A. Présentation | 7 |
| 1. L'entreprise | 7 |
| 2. Chiffres | 7 |
| 3. Site | 8 |
| B. Département Business-Technologie | 8 |
| 1. L'équipe | 8 |
| 2. L'infrastructure | 9 |
| 3. Les outils | 9 |
| II. L'attaque informatique | 10 |
| A. Les problèmes | 10 |
| 1. Vol de données | 10 |
| 2. Cryptolocker | 10 |
| B. Résolution | 10 |
| 1. Utilisation des backups | 10 |
| 2. Suppression des données corrompues | 10 |
| C. Prévention et renforcement des systèmes de protection | 10 |
| 1. Les décisions de l'entreprise | 10 |
| 2. Mon alternance | 11 |
| III. Missions | 12 |
| A. Mise à jour des backups Rubrik | 12 |
| 1. Change | 12 |
| 2. Téléchargement et installation | 13 |
| B. Déploiement de Centreon | 14 |
| 1. Récupération des VM | 14 |
| 2. Déploiement des agents | 14 |
| C. Le logiciel Waarp | 15 |

| | |
|----------------------------------|----|
| 1. Installation..... | 16 |
| 2. Le manager..... | 16 |
| 3. Scripting..... | 19 |
| 4. Documentation technique | 20 |
| D. Ansible..... | 20 |
| 1. Les Playbooks | 20 |
| 2. Ansible Tower..... | 21 |
| Bilan | 26 |
| Bibliographies | 27 |
| Lexique..... | 28 |
| Index..... | 30 |
| Annexes I..... | 31 |
| Annexes II | 32 |
| Table des figures | 34 |

I. Manutan

A. Présentation

1. L'entreprise

Manutan est une entreprise créée en 1966 par André Guichard et son fils Jean-Pierre Guichard.

Elle est le leader européen de vente à distance d'équipements industriels et de bureau auprès des entreprises et des collectivités locales.

Plus de Six-cent-mille produits sont en vente allant de la bureautique à la signalisation.

2. Chiffres

Les chiffres ci-contre sont la représentation pour la période 2020/2021.



Figure 2: Chiffres clés

3. Site

Le site de l'entreprise Manutan situé à Gonesse comprend six infrastructures :

- Un restaurant : des chefs cuisiniers, des repas variés, équilibrés avec des légumes et des fruits de saison
- Une université : permettant les formations dans un environnement conçu pour cela
- Un gymnase : situé dans le même bâtiment que l'université et comprenant une multitude de sports différents (tir à l'arc, badminton, ping-pong, escalade, foot en salle, squash, ...)
- Deux entrepôts : où se situent la logistique, MADREX (MANutan DReda EXtension) et le premier entrepôt
- Le tertiaire : lieu où se trouve les différents départements de l'entreprise sans compter la logistique

Manutan est la première entreprise à avoir visé la double certification HQE (Haute Qualité Environnementale) pour le tertiaire et le logistique.

Le tertiaire comprend trois étages, chaque étage a un ou plusieurs départements.



Figure 3: Site de Manutan Gonesse

B. Département Business-Technologie

1. L'équipe

Le département Business-Technologie se situe au premier étage du bâtiment tertiaire. Il comprend deux services :

- Le service IT comprenant l'infrastructure systèmes et bases de données, la digital workplace, le réseau et téléphonie, l'équipe de projet IT, les data analyst, le développement, le help Desk.
- Le Business comprenant le marketing

Je faisais partie de l'équipe systèmes et bases de données. Cette équipe, en me comptant, était composée à mon arrivée de neuf personnes puis plusieurs embauches ont été effectuées au cours de l'année et les externes ont été remplacés par des internes.

2. L'infrastructure

Les systèmes comprennent toutes les machines physiques comme virtuelles. Cela comprenait 13 clusters :

- 87 machines hôtes (machines physiques)
- 1220 machines virtuelles
 - o 643 machines sous Linux
 - o 578 machines sous Windows

3. Les outils

Chaque équipe a ces propres outils pour travailler, dans l'équipe système nous avons :

- ServiceNow
- beyond Insight
- PRA
- vCenter
- Centreon/Zabbix

ServiceNow est l'outil de ticketing utilisé chez Manutan. Il permet aux employés d'effectuer des demandes aux différents services qui les reçoivent et les gèrent, permettant ainsi une décastration au fur et à mesure des tâches ainsi qu'un suivi des équipes. La plupart des tâches à effectuer ne proviennent donc pas d'un projet mais des utilisateurs et permettent le fonctionnement général de l'entreprise en aidant les employés et prestataires à régler les soucis de fonctionnement au niveau de l'infrastructure système et des données gérées.

Beyond Insight est une solution Beyond Trust comprenant deux parties : Password Safe ainsi que Team Password. Ces deux solutions permettent la gestion des mots de passe, que ce soit ceux des utilisateurs comme ceux des logiciels.

Le PRA ou Privilege Remote Access (accès privilégié à distance) est aussi une solution Beyond Trust. Celle-ci permet la connexion à distance de façon sécurisée des utilisateurs.

vCenter est le logiciel permettant la gestion des machines virtuelles. Il s'agit d'une solution VMware, leader et pionnier sur le marché de la virtualisation de serveurs.

Pour finir, il y a les outils de supervision. À mon arrivé, ils utilisaient le logiciel Zabbix pour effectuer celle-ci, mais l'entreprise voulait une supervision automatique et externe munie de process afin de permettre l'intervention en dehors des heures de bureau. Cette supervision externe était effectuée par l'entreprise Neurones et celle-ci utilisait le logiciel Centreon, nous sommes donc passés de Zabbix à Centreon.

II. L'attaque informatique

A. Les problèmes

1. Vol de données

L'attaque a eu lieu un week-end en février 2021. À ce moment aucune astreinte n'existait, il ne pût donc n'y avoir aucune résistance face à celle-ci. Les voleurs se sont introduits de l'intérieur du système et ont volé les données générales et bancaires de l'entreprise. Elles ont été retrouvées sur le DarkNet².

2. Cryptolocker

À la suite de leur vol, ils ont bloqué tous les systèmes, action aussi appelée cryptolocker³, ce qui empêchait l'utilisation de ceux-ci et induisait donc une perte de profit pour l'entreprise le temps du blocage de ces données. Afin de débloquent la situation, les voleurs demandaient une rançon. Leur attaque était donc une « ransomware⁴ » ou rançongiciel en français.

B. Résolution

1. Utilisation des backups

Par chance, les voleurs n'avaient pas pu bloquer les backups⁵. Plutôt que de payer, l'entreprise a donc repris une sauvegarde de leurs données et a essayé de régler la situation le plus vite possible afin d'avoir une perte la plus minime possible. Toutes les données du week-end ont été perdues mais l'entreprise a réussi à réduire les pertes.

2. Suppression des données corrompues

Après la remise en état des systèmes, une inspection de ceux-ci a eu lieu afin d'éradiquer toute corruption pouvant subsister. La faille a été trouvée et corrigée.

C. Prévention et renforcement des systèmes de protection

1. Les décisions de l'entreprise

Dans les six mois qui suivirent l'attaque, juste avant mon arrivée, les dirigeants prirent différentes décisions afin que cette situation ne se reproduise point. Ils décidèrent de refaire le

² Voir lexique

³ Voir lexique

⁴ Voir lexique

⁵ Voir lexique

réseau à partir de zéro, changer de superviseur pour avoir une supervision externe le week-end et la nuit permettant une intervention en cas de problème au moment où celui-ci se produit, changer l'emplacement des datacenters afin qu'ils soient à deux endroits différents, permettant ainsi une sécurité accrue des données stockées.

2. Mon alternance

C'est donc dans ce contexte que j'ai effectué mon alternance. Les missions m'ayant été confiées, lorsqu'elles faisaient partie d'un projet, étaient des missions ayant pour objectif de renforcer les systèmes existants⁶ dans le premier réseau et/ou d'aider à l'installation des nouveaux systèmes dans le second réseau. Ma mission principale était de m'occuper du logiciel Waarp que vous verrez dans la partie sur les missions effectuées.

⁶ Voir la troisième partie « Missions », p. 12

III. Missions

Au cours de mon alternance j'ai effectué deux missions :

- La première visait à mettre à jour des backups.
- La seconde visait à installer un nouveau superviseur⁷ pour ses machines.

A. Mise à jour des backups Rubrik

Rubrik Polaris est un logiciel en SAAS (Software As A Service ou logiciel en tant que service) c'est-à-dire qu'il permet l'utilisation d'un logiciel à distance sans avoir besoin d'utiliser des certificats car c'est une entreprise externe qui héberge le service.

1. Change

Pour ma première mission, je devais mettre à jour les serveurs de backup des machines hôtes. Pour ce faire, je devais au préalable créer des « change⁸ » puisque c'était une intervention sur une ressource relativement importante car permettant de ne pas perdre les données en cas de panne d'un appareil ou d'attaque. Un change est un ticket permettant la création d'une demande de réunion afin d'évaluer les risques, les impacts et le retour en arrière en cas d'erreur ou d'échec de l'intervention.

Je devais créer un change par machine. Ayant quinze serveurs, j'ai créé un change par serveur. Il y a une multitude de champs et on ne peut reprendre un ticket précédent afin de le dupliquer et d'en changer quelques lignes, le processus tenait assez occupé pour quelques heures.

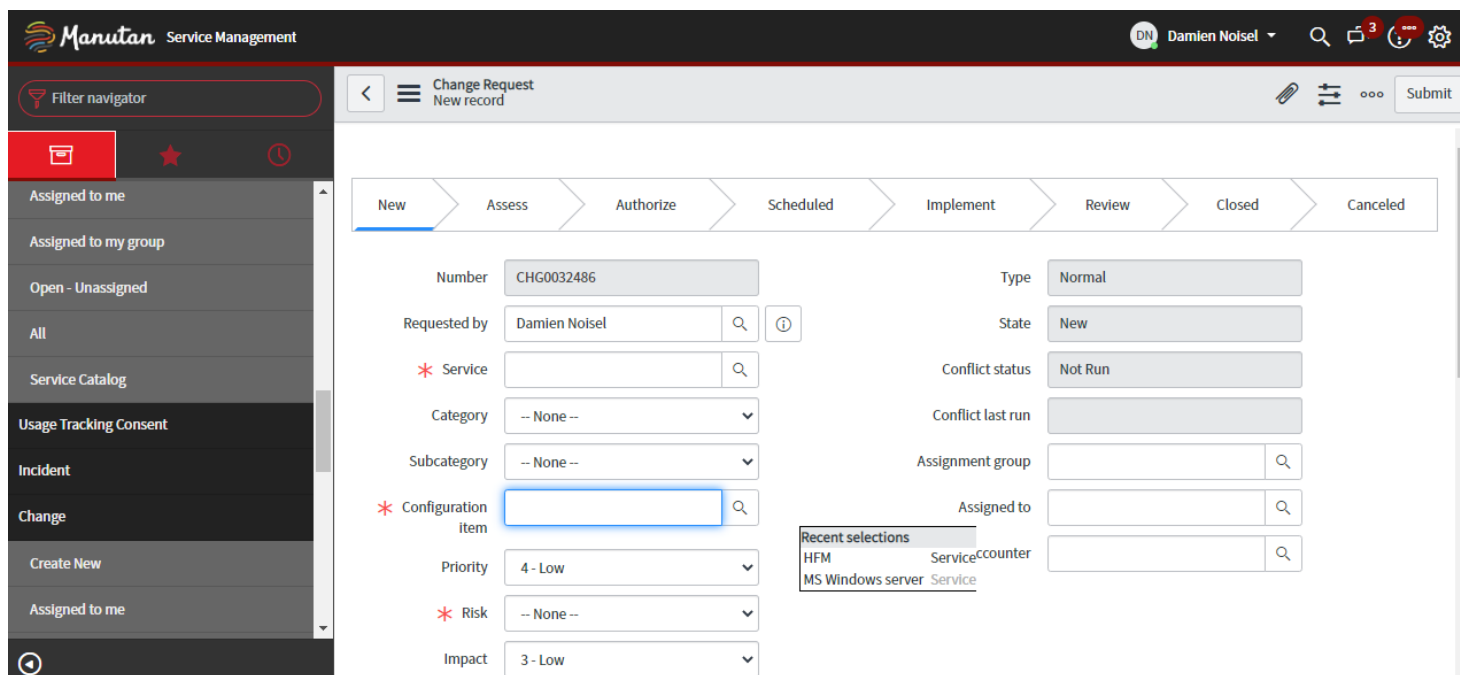
Vint alors le CAB⁹. Celui-ci a lieu chaque vendredi à 11h30 et accueille des sujets variés puisque toute personne de l'entreprise soumettant un change se doit d'y participer pour soutenir son action en argumentant et en répondant aux questions posées par les différents participants. Mon intervention fût très brève car l'impact et les risques de mon intervention étaient plutôt faibles et en cas d'échec il y avait une procédure à suivre. Ils ont donc accepté et ne m'ont posé qu'une question qui était sur la procédure à suivre en cas d'échec.

Ci-dessous un exemple d'une création de change :

⁷ Voir lexique

⁸ Voir lexique

⁹ Voir lexique



The screenshot shows the 'Change Request' form in the Manutan Service Management interface. The form is in the 'New' state and includes the following fields:

- Number: CHG0032486
- Requested by: Damien Noisel
- Type: Normal
- State: New
- Conflict status: Not Run
- Category: -- None --
- Subcategory: -- None --
- Configuration item: (Recent selections dropdown)
- Priority: 4 - Low
- Risk: -- None --
- Impact: 3 - Low

The 'Recent selections' dropdown for the Configuration item field shows the following options:

- HFM
- ServiceCounter
- MS Windows server
- Service

Figure 4: Page de création d'un ticket change sur Service Now

2. Téléchargement et installation

Le téléchargement et l'installation se faisaient tous deux sur le site de Polaris Rubrik, l'hébergeur des backups de l'entreprise Manutan. J'ai effectué les téléchargements en dehors des heures de travail en programmant à l'avance un lancement automatique.

Je devais télécharger après 21h pour éviter d'impacter la bande passante et de ralentir le travail dans l'entreprise. Je n'avais pas besoin d'être devant car un téléchargement ne représentait aucun risque envers la bonne marche de l'entreprise même en cas d'erreur. Les téléchargements se faisaient un à un pour ne pas ralentir les tâches automatiques ayant lieu pendant la nuit.

Vint alors la phase d'installation. Cette dernière demandait une surveillance car en cas d'échec ou de panne cela pouvait nuire au bon fonctionnement du backup sur le cluster concerné par l'installation. C'est pour cette raison que l'installation devait se faire après 14h, car c'est le matin que les backups peuvent être utiles en cas de problème ayant eu lieu pendant la nuit, et avant 16h pour qu'en cas de problème avec l'installation, on ait le temps de rétablir le backup.

Pour le rétablissement du backup, deux cas possibles :

- Le backup fait un retour sur sa version antérieure à l'installation et tout reprend son cours normal.
- Le backup échoue à retourner à sa version antérieure à l'installation, il faut alors entrer en contact avec l'équipe support de chez Polaris Rubrik afin qu'il règle la situation au plus vite car tant que ce n'est pas résolu, aucun backup n'a lieu.

Tout s'est déroulé sans encombre, puisque le créneau horaire était si court et que je ne pouvais donc lancer qu'une à deux installations par jour à raison d'une par heure (deux pour la plupart sauf pour deux clusters critiques pour l'entreprise). J'ai pris plusieurs semaines à compléter ma tâche. Les deux dernières installations ont été effectuées par mon collègue car j'étais en cours.

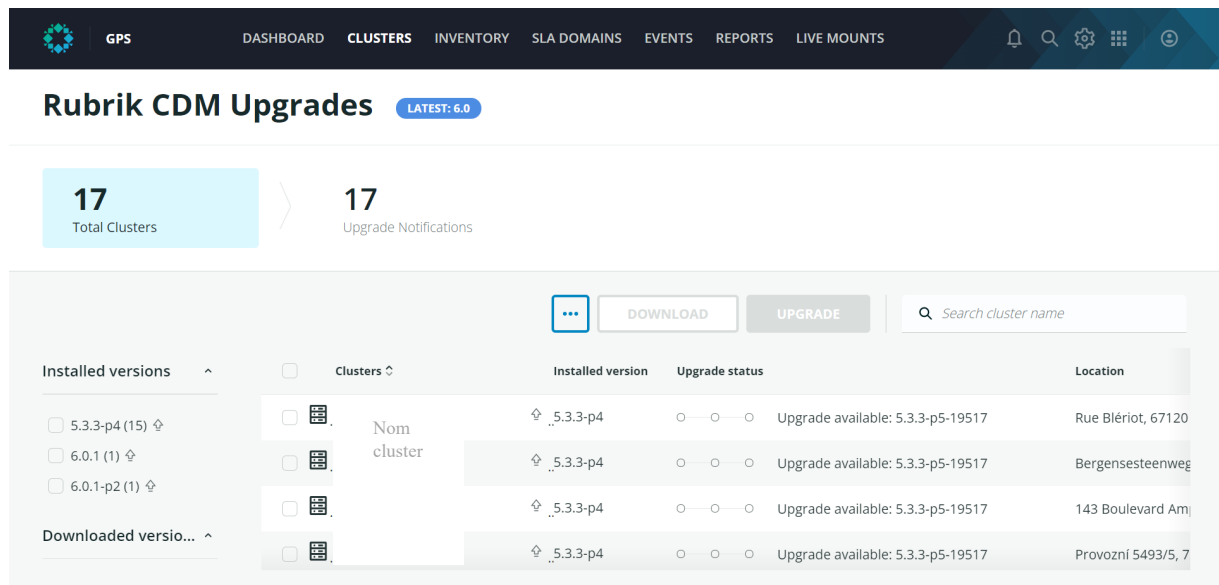


Figure 5: Liste des mises à jour par cluster sur le site Rubrik

B. Déploiement de Centreon

Centreon est un logiciel de supervision tel que Zabbix ou Nagios. Il permet donc la supervision des serveurs, c'est-à-dire la surveillance de l'infrastructure à partir de règles incrémentées au fur et à mesure des besoins tels que des alertes en cas de perte de signal provenant d'une machine ou des alertes en cas de manque de place sur un disque.

1. Récupération des VM

Ma seconde mission fût d'effectuer un déploiement sur bon nombre de machines Windows du parc.

2. Déploiement des agents

a) Automatisation

Pour l'installation du logiciel de supervision Centreon, le logiciel PDQ Deploy ainsi qu'une liste des serveurs sur lesquels le logiciel devait être installé ont été utilisés. Après, sur le logiciel, on crée des étapes en fonction des besoins, ces étapes se lancent alors à tour de rôle pour chaque serveur.

La première étape était de passer la liste des serveurs afin qu'elle puisse être lue.

La seconde étape était de se connecter à la machine avec ses identifiants administrateur.

La dernière étape était de lancer un exécutable contenant le logiciel Centreon puis, à la fin de l'installation, de lancer le script écrit au préalable permettant le redémarrage des services dont le logiciel de surveillance a besoin tel que SNMP et NSClient++.

Afin de ne pas être débordé par les problèmes, je prenais les serveurs vingt par vingt et les notais dans mon fichier texte que je passais au logiciel PDQ Deploy. Presqu'à chaque lancement il y avait au moins un serveur qui ne passait pas. Je les notais alors dans un fichier Excel puis passais aux serveurs suivants.

b) Manuel

Les serveurs notés comme n'étant pas passés, je devais les faire manuellement. C'est-à-dire me connecter à chacun d'entre eux et installer le logiciel ainsi que les services liés.

Certains n'étaient pas passés car je devais donner des comptes spécifiques tandis que d'autres n'étaient juste pas joignables pour diverses raisons. J'ai donc fait tout ceux auxquels je pouvais me connecter et là encore j'ai rencontré d'autres soucis.

Les serveurs sous Windows 2008 et plus ne posaient aucun souci, c'était sous Windows 2003 que c'était très compliqué.

c) Résolution

Nous avons pu résoudre certains problèmes tandis que d'autres ne pouvaient l'être par rapport au logiciel.

Pour les problèmes de connexion, j'ai pu en régler certains en arrêtant de passer par la connexion rdp¹⁰ et en passant directement via vCenter¹¹. Pour d'autres, elles n'étaient juste plus en utilisation et étaient déconnectées du réseau.

Ensuite sont venus les problèmes de services manquants. Pour NSClient++, c'était que l'exécutable s'était mal installé, du coup il fallait le réinstaller. Pour SNMP c'était plus compliqué car pour Windows 2008 et les versions ultérieures il y avait une commande PowerShell permettant l'installation, mais pour les Windows 2003, il fallait utiliser un fichier ISO contenant le service SNMP. Il fallait donc que la machine soit pourvue d'un lecteur de CD virtuel, or, ce n'était pas le cas de tous.

Mon collègue a alors créé un change pour pouvoir installer le lecteur sur les machines n'en étant pas pourvu car c'était des serveurs en cours d'utilisation et un redémarrage était nécessaire.

À la fin, tous nos serveurs n'avaient pas le logiciel de surveillance d'actif. Nous avons donc fait une réunion avec Neurons et il s'est avéré que leur logiciel n'était juste pas compatible avec les serveurs sous Windows 2003.

C. Le logiciel Waarp¹²

Waarp est un MFT (Managed File Transfer ou Gestionnaire de transfert de fichier), il permet un transfert automatisé et sécurisé des fichiers via un protocole propriétaire nommé R66 qui chiffre les données. Waarp en lui-même est un logiciel open source, c'est-à-dire que seul le manager fait partie d'une licence payante.

¹⁰ Voir lexique

¹¹ Voir lexique

¹² Voir le schéma fonctionnel en annexe

1. Installation

L'installation, comme pour tout autre logiciel, est différente d'un OS à un autre et puisque l'équipe système côté Linux est le « Product Owner ¹³ », l'installation est effectuée par nous que ce soit sur une machine Linux comme sur une machine Windows.

a) Windows

Pour commencer, chaque OS vient avec son fichier compressé, sous Windows c'est un fichier ZIP. Pour l'installation de celui-ci, il faut alors, créer un dossier dans « C:\Program Files\ », extraire le ZIP à l'intérieur, changer certaines lignes dans le fichier de configuration puis regarder si le service est bien lancé. Ensuite il faut arrêter le service puis aller sur Waarp Manager¹⁴ afin de créer la règle qui permettra à la machine de communiquer à travers les passerelles à la machine cible. Une fois créée, cette règle est sous forme de fichier devant être ajouté avec les fichiers de configuration précédemment modifiés. Une commande doit être utilisée dans le but de rendre actifs les changements. Pour finir il faut réactiver le service.

Pour vérifier le bon fonctionnement à la suite de l'installation, il faut soit aller sur internet à partir de la machine sur laquelle vient d'être installé le logiciel puis taper « localhost :8088 », un site doit apparaître, autrement une réinstallation doit être faite. Cela signifie qu'il faut tester l'échange en envoyant un fichier à la machine cible.

b) Linux

L'installation sur Linux est relativement semblable si ce n'est les chemins de création des différents dossiers qui diffèrent ainsi que la méthode pour obtenir les fichiers de configuration devant aussi être changés.

2. Le manager

Le manager comme dit précédemment est payant, il demande une licence pour rendre son utilisation possible, autrement, Waarp étant un logiciel open source il est tout à fait possible de l'installer et de l'utiliser. Cet outil développé par l'entreprise Waarp permet une centralisation et une simplification de la gestion des différents besoins auxquels répond Waarp tel que la possibilité de créer un échange de fichier sécurisé avec des partenaires extérieurs à l'entreprise ou aux différentes machines à l'intérieur.

Il comprend plusieurs onglets :

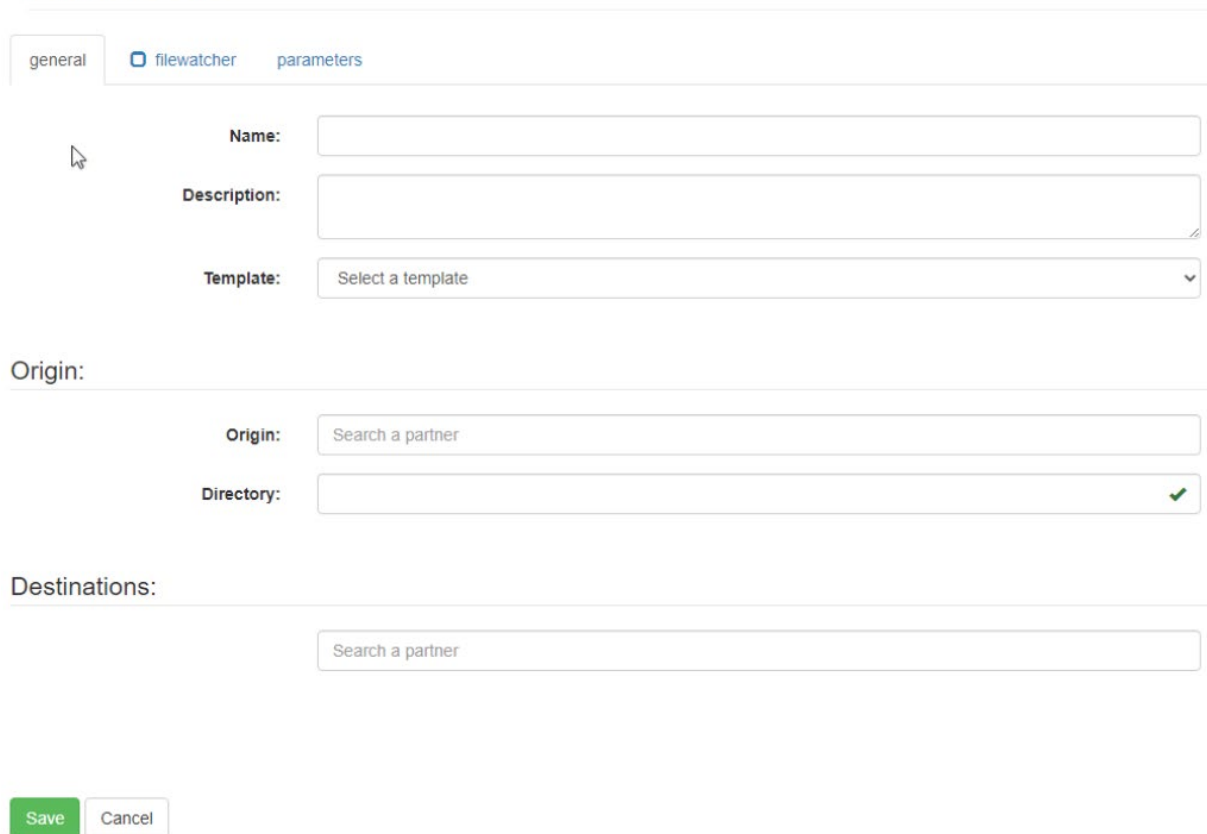
- La gestion des flux¹⁵
- La gestion des différents sites (en l'occurrence zone grise et zone verte)
- La gestion des partenaires
- Un journal de suivi des différentes actions permettant un maintien continu du logiciel
- Un onglet d'administration permettant la gestion des droits des utilisateurs

¹³ Voir lexique

¹⁴ Voir « partie B, Le logiciel Waarp, 2. Le manager »

¹⁵ Voir lexique

New Flow



general filewatcher parameters

Name:

Description:

Template:

Origin:

Origin:

Directory:

Destinations:

Save Cancel

Figure 6: Création d'un flux

Ci-dessus l'onglet de création de flux. On peut voir que tous les logiciels représentés dans mon rapport sont en anglais c'est aussi le cas pour la plupart des autres en raison du caractère international de l'entreprise Manutan.

Le champ « nom » permet juste d'identifier le flux parmi les autres, le champ « description » permet quant à lui de savoir de quel flux il s'agit.

« Template » permet de choisir un moyen de transfert ayant été configuré en amont tel que le choix de copier (copy) le fichier depuis le dossier d'origine ce qui permet de garder le fichier source ou bien de le déplacer (move) ce qui fait disparaître le fichier source.

Les champs « Origin » et « Destinations » ont pour but de définir d'où provient le fichier et à quel endroit il faut le transférer.

Pour finir avec cette capture d'écran, on peut voir deux onglets de plus en dessous du titre de la page soit « filewatcher » et « parameters ». Parameters comme son nom l'indique permet de configurer le flux créé tandis que filewatcher est un outil pouvant être activé au besoin permettant de surveiller le dossier ciblé et dans le cas où un fichier y apparaîtrait, Waarp le détecterait et l'envverrait de manière automatique jusqu'à la destination configurée.

Partners

General information

| | | |
|------------------------|--------------------------|---|
| Partner type: * | Type | ✓ |
| External: ⓘ | <input type="checkbox"/> | |
| Site: * | No site | ✓ |
| Application: | | |
| Environment: | No environment | ✓ |
| Description: | | |
| <div>Save Cancel</div> | | |

Figure 7: Création d'un partenaire

La suite est la création du partenaire.

Le premier champ permet de choisir le type de flux allant être utilisé :

- Tls
- R66
- FTP
- SFTP

C'est la façon dont les données vont être envoyées et le chiffage leur étant associés.

Si le partenaire est extérieur au réseau il faut cocher la checkbox.

Jusqu'à maintenant je n'ai ni eu besoin du champ *Environment* ni du champ *Application*.

Ensuite le site représente la zone dans laquelle il se situe (gris, vert ou extérieur).

Comme cité plus haut, un monitoring est assuré par le manager. À droite nous pouvons voir à quoi ressemble la surveillance des serveurs sur lesquels le logiciel Waarp est installé avec un code couleur en fonction de la situation :

- Gris : le serveur n'est pas joignable
- Rouge : l'hôte existe mais n'est pas accessible
- Noir : le serveur est accessible et connecté

Ci-dessous la représentation des deux statuts possibles pour la surveillance des transferts entre machines.

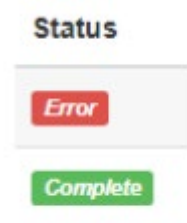


Figure 9: Surveillance des transferts Waarp

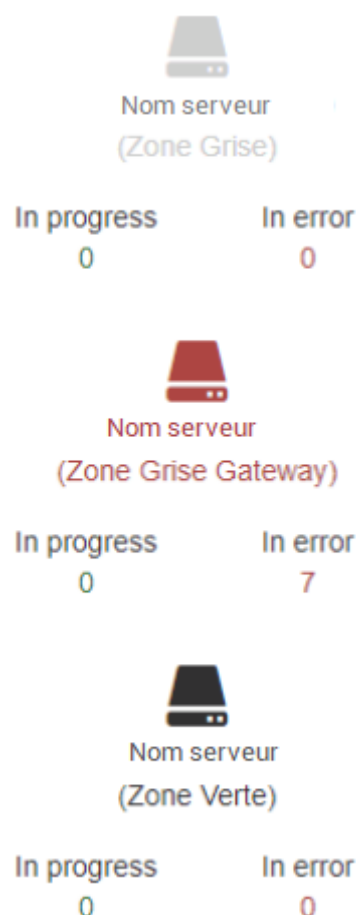


Figure 8: Surveillance des serveurs Waarp

3. Scripting¹⁶

L'une de mes tâches en plus de l'installation et du maintien de Waarp était d'écrire un script d'automatisation de l'envoi des fichiers reçus via Waarp vers deux stockages internes :

- Gitlab (pour faciliter le développement)
- Nexus (pour le stockage des fichiers binaires ¹⁷et des fichiers compressés)

Mon script regardait le dossier d'arrivée et en fonction de ça envoyait le fichier vers Nexus ou Gitlab et cela, dans le bon dossier. J'ai aussi effectué la gestion d'erreurs car :

- Un fichier vide ne peut être envoyé
- Un dossier doit être créé en amont sur Gitlab (Nexus crée si le dossier n'existe pas déjà) sinon une erreur se produit
- Un nom de fichier contenant des caractères spéciaux ou des espaces ne s'envoie pas

¹⁶ Script en annexe

¹⁷ Voir lexique

4. Documentation technique

Une autre tâche m'a été confiée, celle d'écrire un document technique permettant à tout utilisateur même non initié à la technologie Waarp de pouvoir demander la création d'un transfert et de l'utiliser.

Le premier objectif et donc la première partie consiste à expliquer le principe de Waarp et la gestion des fichiers par le logiciel afin d'éviter au maximum un retour sur les fichiers n'étant pas envoyés (les logs créés par la gestion d'erreur permettent de définir quels fichiers ne se sont pas envoyés et pourquoi mais ne permettent pas de « gérer l'erreur » au sens propre du terme mais juste de stocker les fichiers et de créer un log d'erreur).

Le second objectif était d'expliquer à ses mêmes utilisateurs comment récupérer leur fichier à partir de Gitlab ou de Nexus (en l'occurrence principalement Nexus, Gitlab n'étant utilisé que pour le développement et non le stockage).

D. Ansible

De façon à pouvoir d'administrer les différents systèmes plus simplement et dans un souci d'automatisation permanent, la solution Ansible a été choisie puisqu'elle est puissante et qu'elle est sous la maintenance de l'entreprise RedHat, qui constitue plus de 95% de nos machines Linux. Cette solution permet en peu de ligne d'effectuer des actions sur une ou plusieurs machines (hôtes) distantes à l'aide de « Playbooks » et de façon centralisée grâce à Ansible Tower.

Ma mission était d'écrire un Playbooks permettant de modifier le contenu d'un fichier de configuration sur une machine Linux. Cette mission étant simple elle m'a permis de me familiariser rapidement avec l'environnement Ansible et ces outils tel qu'Ansible Tower.

1. Les Playbooks

Ce sont des fichiers de code programmés en langage YAML ou JSON qui permettent de demander aux machines cibles d'effectuer certaines actions. Les Playbooks se doivent d'être « idempotents » c'est-à-dire que lors du lancement d'un, celui-ci se doit d'avoir le même résultat à chaque lancement. Ci-dessous mon premier programme pour tester :



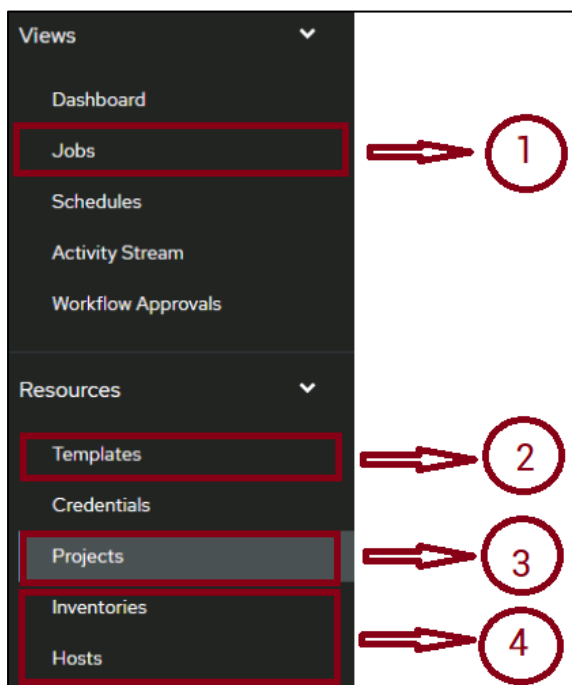
```
1 ---
2 - name: This is a hello-world example
3   hosts: localhost
4
5   tasks:
6
7     - name: Create a file called '/tmp/testfile.txt' with the content 'hello world'.
8       ansible.builtin.copy:
9         content: hello worldn
10        dest: /tmp/testfile.txt
```

Figure 10: Playbook / affichage de "hello worldn"

Ce programme permet juste la création d'un fichier nommé *testfile.txt* et situé sur la machine où est installé Ansible Tower (localhost) dans le dossier */tmp/* avec pour contenu le text *hello worldn*.

2. Ansible Tower

Comme dit précédemment, Ansible Tower permet de réunir tous les outils nécessaires pour structurer son code. Il y a de nombreux onglets dont certains permettant d'effectuer des statistiques. Nous ne nous intéresserons ici qu'aux onglets ayant des contenus me permettant de remplir ma tâche :



J'ai utilisé quatre onglets :

- Jobs (1)¹⁸ : permet d'avoir l'historique de l'utilisation d'Ansible Tower ainsi que de pouvoir relancer une tâche anciennement utilisée
- Templates (2)¹⁹ : qui permet la sélection du lancement souhaité, le Playbook à utiliser ainsi que l'inventaire à utiliser
- Projects (3)²⁰ : qui permet la jonction entre Ansible Tower et le Playbook
- Inventories et Hosts (4)²¹ : qui sont liés l'un à l'autre

Figure 11: Onglets importants d'Ansible Tower

Je ne parlerai pas de l'onglet *Credentials* car il permet de créer des entrées login/mot de passe et je n'en ai, dans mon cas, pas eu le besoin. Il reste néanmoins un onglet important pour permettre une utilisation complète et optimale d'Ansible.

Ansible Tower permet aussi de créer des utilisateurs pour la plateforme et de manipuler leur droit, chose assez précise puisque les droits peuvent être changés pour chaque onglet et chaque objet présent sur Ansible Tower.

¹⁸ Voir page 25

¹⁹ Voir page 24

²⁰ Voir page 23

²¹ Voir page 22

a) Inventories et Hosts

Les onglets *inventories* et *hosts* sont liés l'un à l'autre dans le sens où les hôtes sont des adresses IP ou des FQDN²² représentant une machine (physique ou virtuelle) et qu'un inventaire est composé d'un ou de plusieurs de ces hôtes. A la création d'un hôte, il est demandé dans quel inventaire il doit être placé, il faut donc créer l'inventaire avant l'hôte.

Pour exemple, les captures d'écran suivantes représentent un exemple d'hôte et d'inventaire. Nous pouvons alors voir que les hôtes sont inclus dans des inventaires et que leur nom ne correspond pas à une adresse car leur adresse IP ou FQDN est codée à l'intérieur. Chaque hôte est en soit un objet contenant des informations. Bien entendu rien n'interdit de choisir une adresse IP ou un FQDN comme nom.

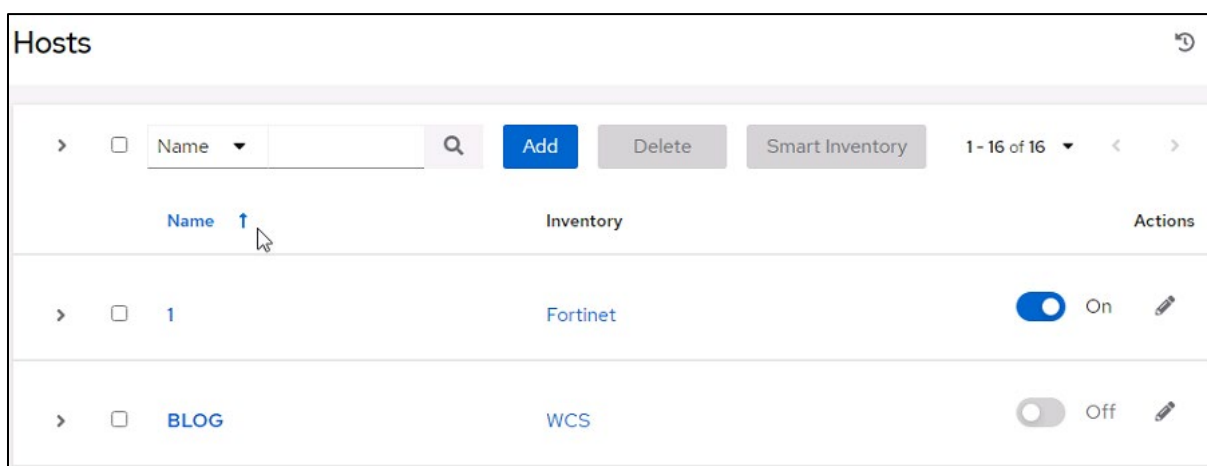


Figure 12: Onglet "Hosts" sur Ansible Tower

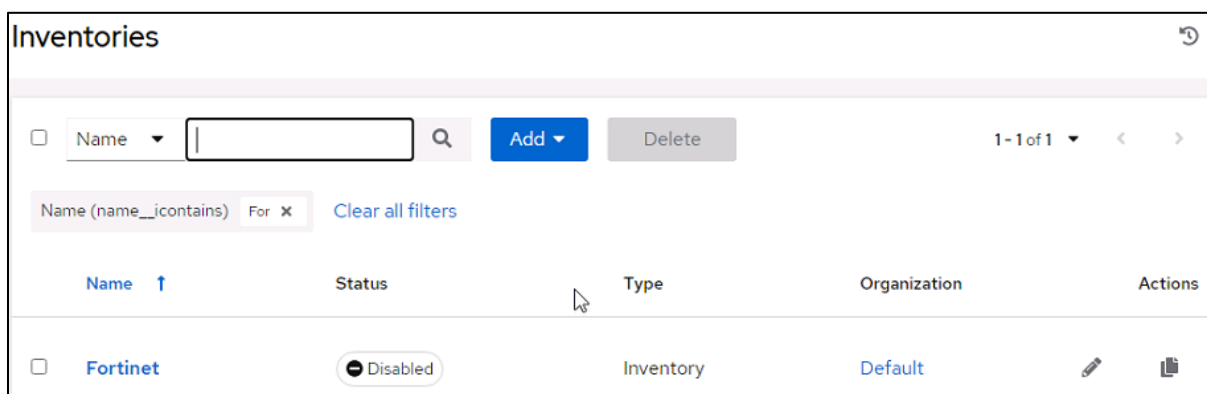


Figure 13: Onglet "Inventories" sur Ansible Tower

²² Voir lexique

b) Projects

La création du projet permet de lier le dossier où se situe le *Playbook* avec Ansible Tower pour permettre son utilisation²³. Comme nous pouvons le voir sur la capture d'écran suivante, l'utilisation d'un dépôt Git afin de conserver son code et de pouvoir l'utiliser est possible et c'est d'ailleurs l'option ayant été retenue par l'entreprise. Tous les *Playbooks* seront donc stockés sur Gitlab. Il faut ensuite donner l'URL vers le dépôt où se situe le code devant être utilisé. Plusieurs codes peuvent s'y trouver (en .yaml ou .json pour une utilisation avec Ansible).

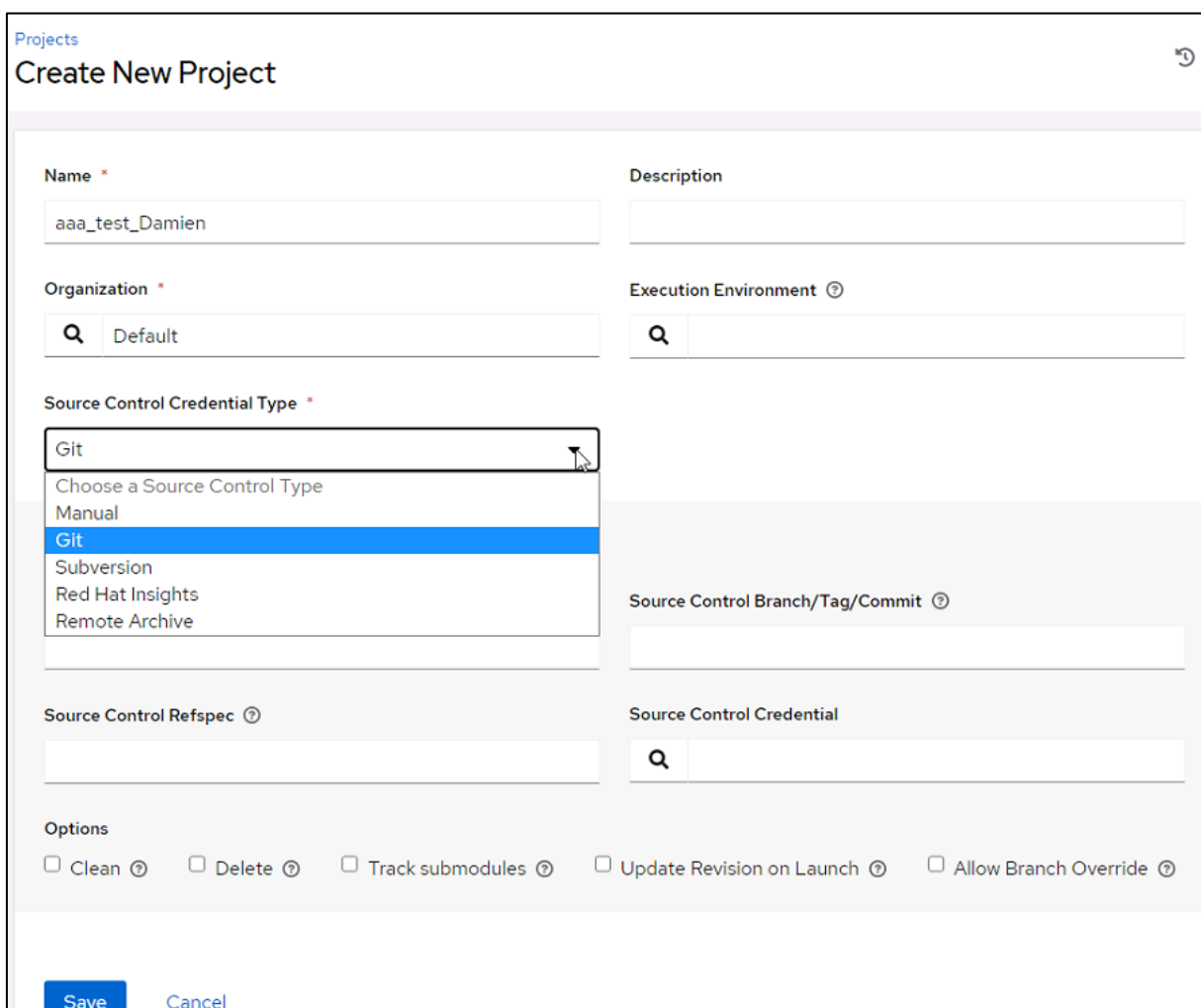
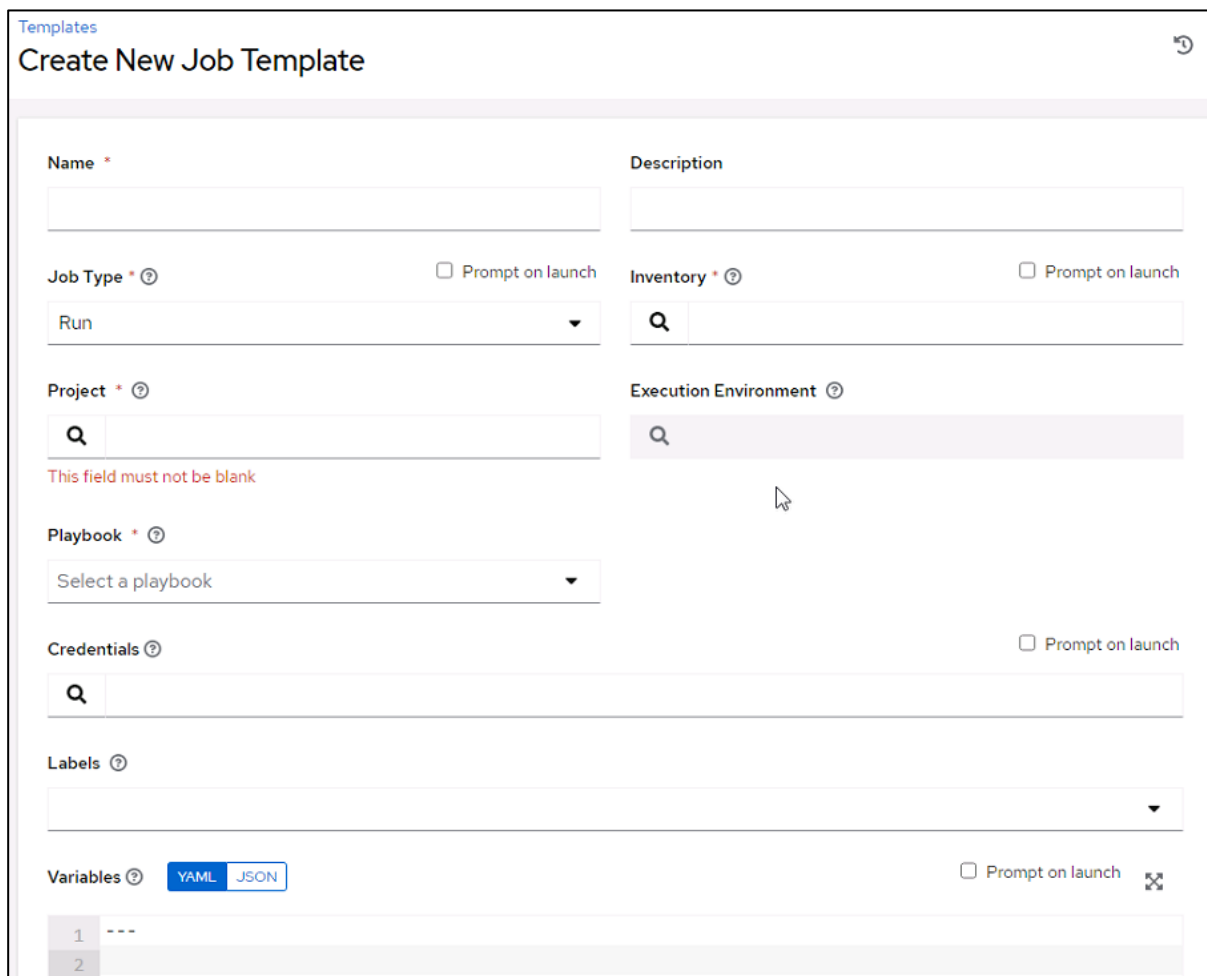


Figure 14: Création d'un projet Ansible

²³ Voir *Templates* page 23-24

c) Templates

Les templates sont destinés à créer des Playbooks ou des Workflows²⁴. Pour se faire, il faut choisir si on souhaite jouer le Playbook ou juste le tester dans le champ *Job type*. On choisit ensuite son *Inventory* et son *Project* comme expliqué précédemment. Pour finir il faut choisir un *Playbook*, les choix qui apparaissent dépendent à quel dossier est lié le projet car il peut y avoir plusieurs programmes présents dans un seul dossier. Au besoin, la possibilité d'ajouter des «*credentials*²⁵ » est autorisée.



The screenshot shows the 'Create New Job Template' form. It has a header 'Templates' and a title 'Create New Job Template'. The form is divided into several sections:

- Name ***: A text input field.
- Description**: A text input field.
- Job Type ***: A dropdown menu with 'Run' selected. There is a checkbox 'Prompt on launch' next to it.
- Inventory ***: A search input field with a magnifying glass icon. There is a checkbox 'Prompt on launch' next to it.
- Project ***: A search input field with a magnifying glass icon. Below it is a red error message: 'This field must not be blank'.
- Execution Environment**: A search input field with a magnifying glass icon.
- Playbook ***: A dropdown menu with 'Select a playbook' selected.
- Credentials**: A search input field with a magnifying glass icon. There is a checkbox 'Prompt on launch' next to it.
- Labels**: A search input field with a magnifying glass icon.
- Variables**: A section with tabs for 'YAML' and 'JSON'. There is a checkbox 'Prompt on launch' and a 'X' icon next to it.

At the bottom, there are two numbered rows (1 and 2) for adding variables.

Figure 15: Création d'un "Template" Ansible

²⁴ Voir lexique

²⁵ Voir lexique

d) Jobs

L'onglet Jobs consiste en un historique de lancement sous Ansible Tower.

La figure suivante illustre différents états qu'il est possible d'obtenir lorsque l'on regarde. On peut y voir quatre états sur cinq :

- Running : signifie qu'un lancement a été effectué. Dans notre cas, c'était un lancement de projet (Source Control Update) car à chaque modification du programme, il faut recharger le projet pour qu'elles soient prises en compte
- Pending : indique que le Playbook a été mis en attente pendant son exécution et peut être remis en route
- Failed : avertie qu'une erreur est présente dans le programme et que son exécution est donc impossible
- Successful : l'exécution du programme c'est bien passée, mais cela ne signifie pas que le résultat soit celui qui était attendu
- Canceled : signifie que l'exécution a été stoppée et ne sera pas reprise

| Name | Status | Type | Start Time | Finish Time | Actions |
|---------------------|------------|-----------------------|------------------------|------------------------|---------|
| 2683 – WCS8 | Running | Source Control Update | 8/18/2022, 10:33:24 AM | | |
| 2682 – WCS Inst all | Pending | Playbook Run | | | |
| 2680 – WCS Inst all | Failed | Playbook Run | 8/18/2022, 10:29:49 AM | 8/18/2022, 10:30:12 AM | |
| 2681 – WCS8 | Successful | Source Control Update | 8/18/2022, 10:29:40 AM | 8/18/2022, 10:29:49 AM | |

Figure 16: Onglet "Jobs" sur Ansible Tower

Bilan

Afin de préparer mon rapport, j'ai fait un rapport journalier de mes tâches et de mes actions telles que les réunions auxquelles j'ai pris part comme celles concernant le codage de scripts. Je me suis servi du site de Manutan nommé M-Life pour obtenir mes chiffres et les informations sur la société en plus des questions posées à mes différents collègues.

Les différentes tâches ont été bénéfiques aussi bien pour mes compétences personnelles que pour mon intégration et ma compréhension de l'infrastructure. La mission sur Rubrik étant la première m'ayant été confiée, elle m'a amenée à me servir des différents outils ainsi que de connaître le moyen d'intervenir sur un logiciel dit en « production²⁶ ».

J'ai eu une certaine difficulté à m'adapter à Waarp. Au départ je ne comprenais pas le concept de « flux », de « partenaires » et les différences entre les protocoles utilisés, après plusieurs explications et une utilisation presque quotidienne de ceux-ci, la compréhension est arrivée. L'entreprise Waarp a décidé, dans un contexte de sécurisation et de simplification, de modifier leur code en passant du JAVA au GOLANG.

Une autre de mes difficultés a alors été de m'adapter pour l'apprentissage de nouveaux langages. Le GOLANG pour Waarp et l'Ansible dans un souci d'automatisation permanent. Ces deux langages devenant très utilisés pour les logiciels systèmes, ces connaissances et compétences me seront utiles plus tard.

En conclusion de mon alternance, j'ai décidé de poursuivre mes études en tant que DevOps pour rester dans les systèmes ainsi que m'orienter vers les systèmes Linux.

²⁶ Voir lexique

Bibliographies

IBM. (2019). *What is managed file transfer?* Récupéré sur [www.ibm.com: https://www.ibm.com/topics/managed-file-transfer](https://www.ibm.com/topics/managed-file-transfer)

Manutan. (2020). *Chiffres clés*. Récupéré sur M-Life.

TOUBA, K. (2022). Consignes de présentation du rapport. Villetaneuse, 92, Ile-de-France.

WAARP SAS. (2021). *OPEN SOURCE MANAGED FILE TRANSFER WAARP*. Récupéré sur [waarp.fr: https://waarp.fr/](https://waarp.fr/)

Wikipedia, C. (s.d.). *Software as a service*. Récupéré sur [fr.wikipedia.org: https://fr.wikipedia.org/wiki/Software_as_a_service](https://fr.wikipedia.org/wiki/Software_as_a_service)

Lexique

¹ **IT (Information Technologie)** : technologie de l'information en français, le service systèmes et infrastructures auquel j'appartenais fait partie du pôle IT comprenant tous les services technologiques compris dans l'entreprise.

² **Darknet** : réseau privé dont les connexions ne sont faites qu'entre pairs de confiance sous couvert d'anonymat (votre adresse IP n'est pas visible).

³ **Cryptolocker** : procédé visant à chiffrer un système entier afin de le rendre inutilisable à moins de trouver la clé de déchiffrement.

⁴ **Ransomware** ou rançongiciel en français : procédé visant à demander une rançon en échange de la restitution de données et/ou le déblocage d'un système.

⁵ **Backup** : sauvegarde d'un système pouvant être utilisée en cas de besoin pour revenir à un point antérieur.

⁷ **Superviseur** : logiciel permettant de surveiller le maintien des services présents sur une machine. Une alerte est envoyée par mail en cas de problème.

⁸ **Change** : ticket créé en cas d'intervention sur un système en production. Il permet de parler de l'intervention à faire auprès des services en charge et de peser les besoins et les risques de l'intervention.

⁹ **CAB** : nom de la réunion dans laquelle les change sont décidés.

¹⁰ **RDP (Remote Desktop Protocol)** : protocole qui permet à un utilisateur de se connecter sur un serveur exécutant Microsoft Terminal Services donc un serveur ayant un OS Windows.

¹¹ **vCenter** : utilitaire de gestion pour vmWare, il permet de rassembler toutes les machines virtuelles créées en un site et de les gérer.

¹³ **Product Owner** : représente la ou les personnes ayant les droits sur l'application et la maintenant en fonction.

¹⁵ **Flux** : présente une analogie avec une bande transporteuse où les éléments sont traités séquentiellement

¹⁷ **Fichier binaire** : est un fichier qui ne contient pas de texte. Il est utilisé pour stocker des données sous forme d'octets, qui sont généralement interprétés comme autre chose que des caractères textuels. Ces fichiers contiennent généralement des instructions dans leurs en-têtes pour déterminer comment lire les données qui y sont stockées. Ils peuvent être utilisés pour stocker tout type de données dans un ordinateur.

²² **FQDN (Fully Qualified Domain Name)** : nom de domaine qui donne la position exacte de son nœud dans l'arborescence DNS en indiquant tous les domaines de niveau supérieur. On parle également d'un domaine absolu

²³ **Workflows** : succession de Playbook enclenchée de manière automatique

²⁴ **Credentials** : représente en anglais les login et mots de passe de connexion

²⁴ **Production** : désigne toute application qui, dès lors de son arrêt, implique une perte d'exploitation pour l'entreprise.

Index

| | |
|---------------------|--------------------------------|
| administrateur..... | 4, 14 |
| adresse IP..... | 22, 28 |
| Ansible Tower..... | 20, 21, 23, 25 |
| Automatisation..... | 14, 15 |
| backup..... | 12, 13 |
| CAB..... | 12, 28 |
| change..... | 12, 15, 28 |
| cluster..... | 13 |
| cryptolocker..... | 4, 10 |
| déploiement..... | 14 |
| flux..... | 16, 17, 18, 26, 34 |
| FQDN..... | 22, 28 |
| Gitlab..... | 19, 20, 33 |
| installation..... | 13 |
| IT2, 8 | |
| Manuel..... | 15 |
| Manutan..... | 4, 7, 8, 9, 13 |
| Nexus..... | 19, 20, 32 |
| Playbook | |
| Playbook..... | 21, 23, 24, 25, 28, 34 |
| Playbooks..... | 6, 20, 23, 24 |
| Polaris | |
| Rubrik..... | 13 |
| sauvegarde..... | 10, 28 |
| script..... | 14 |
| serveurs..... | 4, 9, 12 |
| supervision..... | 9, 11, 14 |
| téléchargement..... | 13 |
| Waarp..... | 11, 15, 16, 17, 19, 20, 26, 31 |
| Windows..... | 9, 14 |

Annexes I

Waarp Schéma fonctionnel

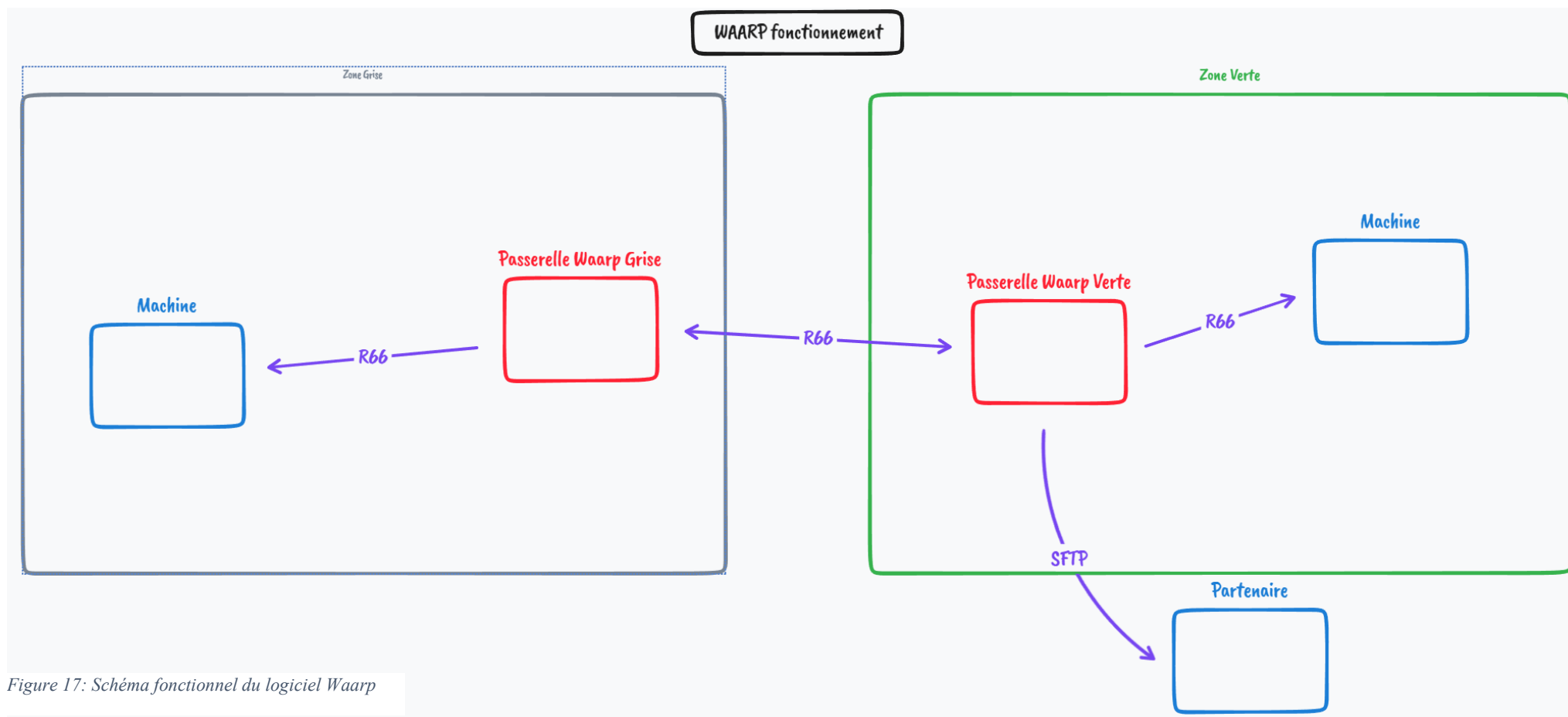


Figure 17: Schéma fonctionnel du logiciel Waarp

Annexes II

Script Nexus

```

USER=
PWD=
FAILURE=/var/lib/waarp/migmnia-warpas2/in/FAILURE/
declare -a toSend #=a table / -A map
toSend=( "SECU" "TIBCO" "DOLLARU" "AX2K12" "SNOW" "DWK" "HFM" "TTMON" "EXTREAM" "SLIM4" "LINKKI" "INOW" )
count=0

#Directory verification and file upload function.
chk_repo() {
    #Verify the existence of a file.
    if [[ -f "$path$file" ]]
    then
        #Take repository registered in the table.
        for repoSend in ${toSend[@]}
        do
            #Verify the existence of the repo in the table. If 1 in return, try to send file, ether send a log and don't touch the file
            if [ "$repo" = "$repoSend" ]
            then
                #Upload the file on Nutanix server.
                cat <<EOF | curl -v --user "$USER:$PWD" --upload-file $path$file http://          : /repository/waarp-inbound/$repo/$file 2>/root/tmp.txt
                #Verifies that the file has been sent by the return code. If 1 delete the file send, ether keep file in a precise repertory to dont forget to send it by hand.
                if [[ "$(cat /root/tmp.txt | grep -i HTTP/1.1)" =~ .*201.* ]]
                then
                    echo "[${TIMESTAMP}] SUCCESS to send '$file' to Nexus: 'waarp/$repo', deleting." | tee -a /var/log/inotify/transfert.log
                    rm $path$file
                else
                    echo "[${TIMESTAMP}] Fail to send '$file' to Nexus: 'waarp/$repo'. Error: '$(cat /root/tmp.txt | grep -i '^HTTP/1.1')'. Moved to $FAILURE/$repo" | tee -a /var/log/inotify/transfert.log
                    mv $path$file $FAILURE$repo
                    fi
                    break
                    count=0
                else
                    count=$((count + 1))
                    if [ $count -eq ${#toSend[@]} ]
                    then
                        echo "[${TIMESTAMP}] Fail to send '$file'. Rule on $repo doesn't exist." | tee -a /var/log/inotify/transfert.log
                        count=0
                    fi
                fi
            done
        else
            echo "Invalid path: $path$file doesn't exist."
        fi
    }

#Monitoring the activities of a directory.
#According to the events, we associate an action.
/usr/bin/inotifywait -mqr --event create /var/lib/waarp/ /in/ | while read line; do
    #takes the variables that are useful to us.
    event=$(echo $line | /usr/bin/awk '{print $2}')
    path=$(echo $line | /usr/bin/awk '{print $1}')
    #to rename files without the event.
    fileold=$(echo $line | sed 's:\/.*:CREATE :g')
    file=$(echo $fileold | sed 's:\/.*:;:g')
    mv "$path$fileold" "$path$file"
    #takes only the directory in which the file was downloaded to use after.
    repo=$(echo $path | sed -r 's:\/.*:\/in::g' | sed -r 's:\/.*:;:g')
    #Take TIMESTAMP just for the logs.
    TIMESTAMP=$(date +%s)
    case $event in
        CREATE)
            chk_repo
            rm /root/tmp.txt
        esac
    done | tee -a /var/log/inotify/transfert.log

```

Figure 18: Script d'automatisation Nexus

Script Gitlab

```
declare -A toKeep
toSend=([SECU]=2 [TIBCO]=12 [DOLLARU]=13 [AX2K12]=14 [SNOW]=19 [DWK]=29 [HFM]=35 [TTMON]=37)
toKeep=( 'INFRA' )
TOKEN=""
FAILURE=/var/lib/waarp/migmnia-warpas2/in/FAILURE/
count=0

chk_file() {
    if [[ -f "$path$file" ]]; then
        for repoSend in ${!toSend[@]}
        do
            md5sum $path$file > checksum.md5
            sleep 1
            while [ $(md5sum -c /root/checksum.md5 | awk '{print $2}') -ne "OK" ]
            do
                sleep 1
                md5sum $path$file > checksum.md5
            done
            if [ "$repo" = "" ]
            then
                echo "[${TIMESTAMP}] This file must be sent by hand." | tee -a /var/log/inotify/transfert.log
            elif [ "$repo" = "$repoSend" ]
            then
                #echo "curl -k --header \"PRIVATE-TOKEN: $TOKEN\" --upload-file $path https://          /api/v4/projects/${toSend[$repo]}/packages/generic/test/0.0.1/$file"
                curl -k --header "PRIVATE-TOKEN: $TOKEN" --upload-file $path$file https://          /api/v4/projects/${toSend[$repo]}/packages/generic/my_package/0.0.1/$file $1>/root/tmp.txt
                sleep 15
                if [[ "$(cat /root/tmp.txt)" =~ .*201.* ]]
                then
                    echo "[${TIMESTAMP}] SUCCESS to send '$file' to Gitlab '$repo: ${toSend[$repo]}', deleting." | tee -a /var/log/inotify/transfert.log
                    rm $path$file
                else
                    echo "[${TIMESTAMP}] Fail to send '$file'. Error: '$(cat /root/tmp.txt)'. Moved to $FAILURE/$repo" | tee -a /var/log/inotify/transfert.log
                    mv $path$file $FAILURE/$repo/
                fi
                break
                count=0
            else
                count=$((count + 1))
                if [ $count -eq ${#toSend[@]} ]
                then
                    echo "[${TIMESTAMP}] Fail to send '$file'. Rule on $repo doesn't exist." | tee -a /var/log/inotify/transfert.log
                    count=0
                fi
            fi
        done
    else
        echo "Invalid path"
    fi
}

/usr/bin/inotifywait -mqr /var/lib/waarp/migmnia-warpas2/in/ | while read line; do
    # variable
    evenement="echo $line | /usr/bin/awk '{print $2}'"
    path="echo $line | /usr/bin/awk '{print $1}'"
    file="echo $line | /usr/bin/awk '{print $3}'"
    repo="echo $path | sed -r 's:^\./in::g' | sed -r 's:/.%$::g'"
    TIMESTAMP="date"
    case $evenement in
        CREATE)
            chk_file
        esac
    done | tee -a /var/log/inotify/transfert.log
done -f /root/tmp.txt
```

Figure 19: Script d'automatisation Gitlab

Table des figures

| | |
|---|----|
| Figure 1: Représentation caricaturée d'une infrastructure | 0 |
| Figure 2: Chiffres clés | 7 |
| Figure 3: Site de Manutan Gonesse..... | 8 |
| Figure 4: Page de création d'un ticket change sur Service Now | 13 |
| Figure 5: Liste des mises à jour par cluster sur le site Rubrik | 14 |
| Figure 6: Création d'un flux | 17 |
| Figure 7: Création d'un partenaire | 18 |
| Figure 8: Surveillance des serveurs Waarp | 19 |
| Figure 9: Surveillance des transferts Waarp..... | 19 |
| Figure 10: Playbook / affichage de "hello worldn" | 20 |
| Figure 11: Onglets importants d'Ansible Tower | 21 |
| Figure 12: Onglet "Hosts" sur Ansible Tower | 22 |
| Figure 13: Onglet "Inventories" sur Ansible Tower | 22 |
| Figure 14: Création d'un projet Ansible | 23 |
| Figure 15: Création d'un "Template" Ansible | 24 |
| Figure 16: Onglet "Jobs" sur Ansible Tower | 25 |
| Figure 17: Schéma fonctionnel du logiciel Waarp..... | 31 |
| Figure 18: Script d'automatisation Nexus | 32 |
| Figure 19: Script d'automatisation Gitlab..... | 33 |