

**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**

**Modélisation et Gestion de l'infrastructure
physique d'un Campus**

Rayan SABQI

DIRNUM Luminy

Responsable entreprise : Pascal MOURET

Responsable académique : Tin NGUYEN

2024

Table des matières

1	Introduction	5
2	Présentation de l'entreprise	6
2.1	Aix-Marseille Université (AMU)	6
2.2	DirNum	6
2.2.1	Présentation générale	6
2.2.2	Les Missions	7
2.2.3	L'équipe réseaux de Luminy	8
3	Environnement de travail	9
3.1	Installation de mon poste de travail	9
3.2	Les Réunions	10
3.3	Maintenance/Surveillance en temps réel	11
3.3.1	ZABBIX	11
3.3.2	GLPI	12
3.4	Gestion du réseau	13
3.4.1	Différents Switchs	13
3.4.2	Le Wifi	14
4	Gestion de l'infrastructure physique	15
4.1	GRIP : Gestion Réseau au niveau de l'Infrastructure Physique	15
4.1.1	Description générale	15
4.1.2	Présentation de GRIP	15
4.2	Recherche et comparatif des solutions	18
4.3	Choix de la solution : NetBox	22
4.3.1	Installation	22
4.3.2	Plugins et développement	23
4.3.4	Migration des données	24
5	Conclusion	25
6	Remerciements	27
7	Glossaire	29
8	Sitographie	31
9	Table des illustrations	31

1 Introduction

Pour gérer les 40 bâtiments, 120 locaux techniques, environ 100 rocares en fibre optique et 6000 prises, présentes dans l'infrastructure de l'équipe réseaux de la Direction du Numérique de Luminy (DirNum Luminy), il est crucial de disposer d'outils de gestion des ressources permettant d'anticiper les améliorations sans inspection physique. Bien évidemment, j'ai pu compter sur l'aide de mon tuteur de stage, Pascal MOURET, responsable de l'équipe réseaux de la DirNum de Luminy.

Un des principaux objectifs aura été de rechercher, comparer et analyser les différentes solutions disponibles sur le marché. Puis, choisir le bon chemin à suivre en réglant les différents problèmes, qui pouvaient survenir, selon les besoins de l'entreprise. L'outil final devait permettre une vision centralisée du réseau, facilitant ainsi son exploitation et la planification de ses évolutions.

Dans un premier temps, je vais présenter l'entreprise et tout son écosystème. Ensuite, je passerai par l'environnement de travail et l'organisation qu'il y a autour. Dans un dernier temps, j'expliquerai de manière détaillée le processus effectué pour arriver à la solution finalement choisie, ainsi que les difficultés rencontrées. Enfin, je conclurai par un bilan des actions effectuées et une réflexion.

2 Présentation de l'entreprise

2.1 Aix-Marseille Université (AMU)

AMU, l'une des plus grandes universités francophones, a été créée en 2012 par la fusion de trois universités de la région Aix-Marseille. Elle est située dans la région Provence-Alpes-Côte d'Azur, avec des campus répartis entre Aix-en-Provence et Marseille. Celle-ci est actuellement sous la direction d'Éric BERTON élu Président le 6 janvier 2020.

Elle offre une large gamme de formations couvrant divers domaines académiques et professionnels. Elle est reconnue pour son excellence en recherche, et maintient des collaborations internationales. Avec plus de 80 000 étudiants, plus de 5 000 personnels et un réseau étendu d'instituts et de laboratoires comptant près de 2 000 BIATSS, AMU est un acteur majeur de l'enseignement supérieur en France.

2.2 DirNum

2.2.1 Présentation générale

La DirNum se trouve au centre d'AMU, car celle-ci permet d'assurer la sécurité et la disponibilité des services pour l'ensemble des campus. Dirigée par Jean-Philippe FLORET, celle-ci se compose de différents pôles transversaux par activités, ainsi que d'antennes de campus, comme on peut le voir sur la *Figure 1*. On y retrouve notamment, le pôle administratif, audiovisuel, environnement numérique, logiciel de gestion, ressources système, Web et le pôle réseaux. Dans nos antennes de campus, on retrouve les campus d'Etoile (avec pour site distant le Pharo), la Timone, Aix, Centre et Luminy (avec pour site distant la Ciotat).

La DirNum est pilotée par un Directeur, assisté par un Directeur adjoint, par deux Directeurs techniques infrastructures et systèmes d'information et par une responsable administrative.

Au niveau du personnel disponible, la DirNum compte environ 200 personnes à son actif réparties sur les différents pôles et antennes de campus cités plus haut. Parmi les antennes de campus, celle d'Aix qui a le plus de personnel à disposition, plus ou moins 35 personnes, et la Timone en qui en a le moins, environ 12 personnes.

Dans chaque antenne de campus, différentes missions de proximité sont présentes. Celles-ci vont servir plus précisément, comme l'indique leur nom, à aider les usagers du campus. On peut les considérer comme des équipes sur un campus, comme notamment l'équipe gestion de parc, Système, Assistance aux usagers mais aussi celle que j'ai intégré, l'équipe réseaux.

Il est important de noter que, par exemple, le pôle réseau et les équipes réseaux des campus sont deux entités différentes mais qu'ils travaillent tous en lien étroit les uns avec les autres.

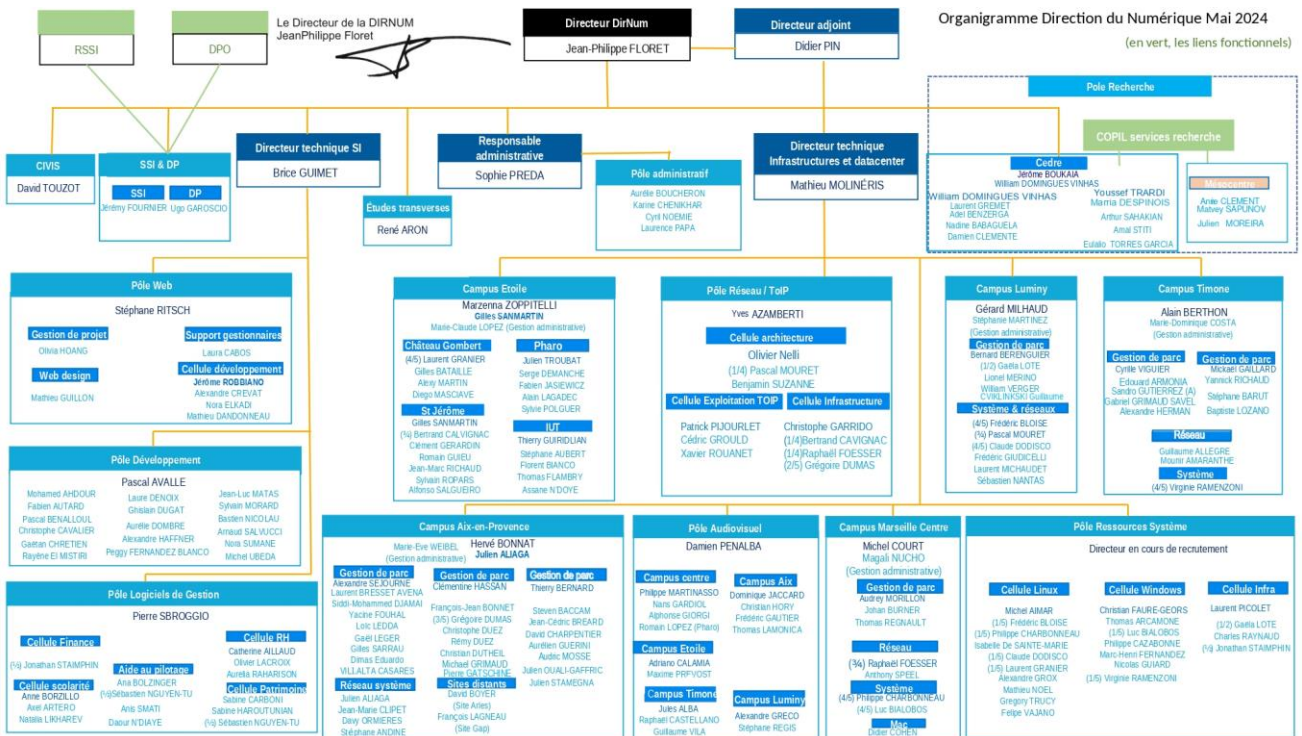


Figure 1 : Organigramme hiérarchique de la DirNum

2.2.2 Les Missions

La DirNum est responsable de la mise en œuvre, de l'administration et de la maintenance du réseau métropolitain, des réseaux de campus, ainsi que de la connexion à Internet via le Réseau national de télécommunications pour la technologie, l'enseignement et la recherche (RENATER). De plus, la DirNum assure la mise en place et la gestion du Datacenter de l'université, en s'occupant des serveurs, de la protection et de la sauvegarde des données.

La DirNum est également chargée du développement, de l'acquisition et de la maintenance d'applications de gestion. On peut alors citer, les proiciels de gestion de la scolarité, de la gestion financière et des ressources humaines. Elle va aussi s'occuper du développement et de la maintenance des sites web de l'université et va fournir une gamme de services numériques via l'environnement numérique de travail (ENT), incluant les messageries, les listes de diffusion, le partage de documents et l'organisation du travail.

Enfin, la DirNum gère un large parc de matériels informatiques administratifs et pédagogiques, ainsi que les logiciels et les équipements audiovisuels, tels que la vidéo-projection et la sonorisation. Elle va offrir un soutien aux usagers pour garantir l'accès et l'utilisation efficace des services numériques, ce qui va ainsi contribuer de manière significative au bon fonctionnement de l'université.

2.2.3 L'équipe réseaux de Luminy

Au sein de la Dirnum campus Luminy se trouve l'équipe réseaux. Celle-ci se compose à l'heure actuelle son responsable, Pascal MOURET, ainsi que ses équipiers, Laurent MICHAUDET, Sébastien NANTAS et Frédéric GIUDICELLI. Ils vont se charger des problèmes réseaux sur le campus, par extension faire de la veille sur celui-ci à l'aide d'outils programmés dans ce cadre. Ils sont aussi là pour aider les usagers, qui pourraient être par exemple des laboratoires de recherche, pour divers problèmes, que ça soit wifi, filaire ou même au niveau des équipements.

Une bonne partie du travail consiste aussi à faire évoluer le réseau. La subtilité est qu'il ne suffit pas seulement de faire marcher ce qui existe déjà, mais d'en découvrir toujours plus et évoluer de façon permanente tout en imaginant comment mettre en place ces évolutions. On peut aussi citer la gestion du datacenter qui se trouve sur le campus de Luminy.

3 Environnement de travail

3.1 Installation de mon poste de travail

L'installation de Kali Linux sur un poste a été ma première tâche pour bien débiter ce stage. Pour pouvoir travailler convenablement, le système d'exploitation que je connaissais le mieux, via mes cours à l'Institut Universitaire de Technologie (IUT), était Linux avec sa distribution Kali.

La démarche à suivre n'était pas si compliqué grâce à l'utilisation d'une clé Universal Serial Bus (USB) simple et du logiciel Balena Etcher pour rendre une clé USB "bootable". J'ai dû commencer par télécharger Balena Etcher, une fois l'application installée, il a fallu l'ouvrir et sélectionner le fichier ISO de Kali Linux (récupéré au préalable sur leur site officiel). Ensuite, choisir la clé USB comme cible pour l'installation. Cliquer sur "Flash" pour démarrer le processus. Balena Etcher écrira alors l'image ISO sur la clé USB, la rendant bootable. Une fois tout ceci terminé, la clé USB était prête à être utilisée pour installer Kali Linux sur ma machine.

Dans mon cas avec l'utilisation de linux (sur un poste de travail temporaire pour pouvoir rendre la clé USB "bootable"), j'ai pu récupérer l'application Balena Etcher au format AppImage, qui est une archive qui contient le logiciel et toutes les bibliothèques dont il dépend. Après avoir téléchargé le fichier, il a fallu lui attribuer les droits d'exécution avec la commande "**chmod +x nomFichier.AppImage**", puis le lancer avec "**./nomfich.AppImage**". Une fois le "flashing" terminé, il ne manquait qu'à insérer la clé USB dans l'ordinateur cible, le redémarrer, et configurer le Basic Input/Output System (BIOS) ou l'Unified Extensible Firmware Interface (UEFI) pour démarrer à partir de la clé USB.

La dernière étape pour finaliser l'installation de mon poste aura été de suivre la documentation officielle de Kali Linux.

Dans le cas présent, j'ai dû passer par une installation graphique, puis choisir la langue, le nom d'utilisateur (Hostname), le nom complet de l'utilisateur, ainsi que l'Username et le mot de passe qui servent à l'authentification. Et pour finir, le choix du disque qui aura été partitionné pour permettre l'installation définitive des metapackages et du GRUB boot loader.

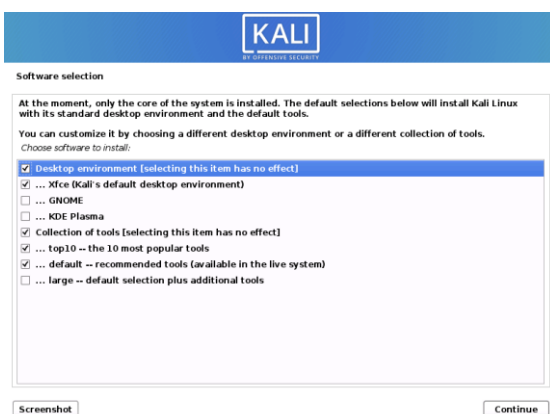


Figure 2 : metapackages

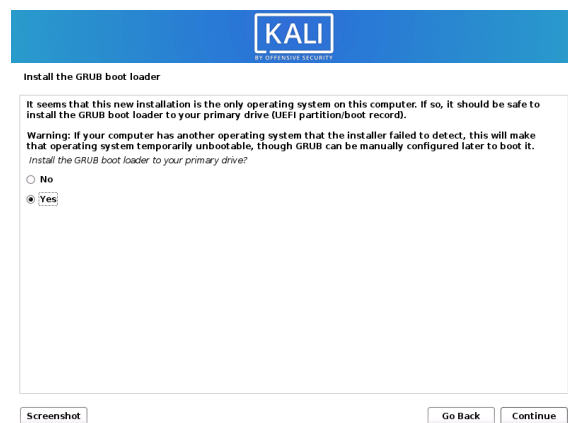


Figure 3 : GRUB boot loader

3.2 Les Réunions

Un grand point important de mon stage, les réunions qui ont lieu environ deux fois dans le mois. Celles-ci sont faites pour réunir l'ensemble du service et partager les informations de chaque équipe, donc, souvent, j'ai été amené à découvrir les nouveautés ou les actualités des équipes comme celle de la gestion de Parc ou Système. Lorsqu'une personne est en télé-travail, celle-ci a quand même l'obligation d'y participer à l'aide du logiciel Zoom ou sa version Web.

Il arrivait aussi parfois où seulement notre équipe réseaux faisait des réunions, ce qui permettait de prendre le moment de voir l'avancement de chacun précisément. De mon côté, j'ai eu l'occasion d'y participer pour présenter l'avancement sur mon sujet de stage, les aspects techniques, les points positifs ou même négatifs, ainsi que les difficultés rencontrées. Cela m'a réellement permis de pousser ma réflexion technique tout en étant à l'oral devant des personnes tout aussi techniques, et d'un autre côté j'ai aussi pu avoir des avis constructifs et pertinents de la part de mes équipiers.

Chaque présentation était faite sur un écran dans la salle de réunion, où était projeté le partage d'écran d'un ordinateur. La personne qui présidait la réunions (ou les personnes, car s'il y a plusieurs équipes, chaque responsable la représentera), utilisait un logiciel de mind-mapping du nom de Freeplane. C'est un logiciel libre et gratuit conçu pour la cartographie mentale et la cartographie de l'information. Il permet d'organiser les idées et grands points pour découler ensuite sur des plus petits, au plus on déplie la carte, au plus les grands points se précisent et se détaillent pour en arriver à des points plus précis au bout.

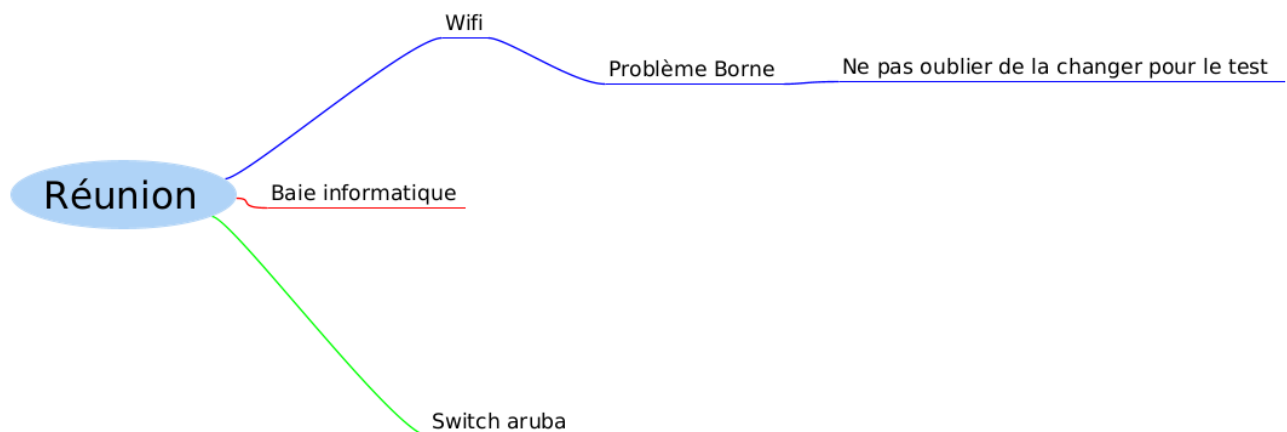


Figure 4 : Exemple sur le logiciel Freeplane

3.3 Maintenance/Surveillance en temps réel

3.3.1 ZABBIX

Zabbix est un logiciel open source développé par Alexei Vladishev, distribué sous General Public License (GPL). Il permet de superviser et de surveiller en temps réel l'infrastructure informatique, y compris l'état du réseau comme ses serveurs et ses équipements (switchs, routeurs et imprimantes par exemple).

Zabbix fonctionne en utilisant un système de serveur et d'agents pour surveiller les infrastructures informatiques. Le serveur stocke les données dans des bases MySQL ou PostgreSQL et peut collecter des informations via des protocoles comme le Simple Network Management Protocol (SNMP) par exemple.

Pour faciliter la collecte de données, le logiciel propose des agents que l'on peut installer sur les équipements. Mais le plus commun est l'utilisation de SNMP pour récupérer les informations (il faut bien sûr que le matériel supporte le protocole SNMP).

L'interface web de Zabbix est intuitive et permet de visualiser tous les équipements surveillés sous forme de graphiques, cartes et tableaux. Cela rend la gestion des infrastructures IT simple et visuellement accessible.

Dans le cas de la DirNum Luminy, l'utilisation de Zabbix se fait sur un serveur dédié et leur permet de centraliser l'information dans un affichage sous forme de tableau avec des entrées comme, le nom (Host), le groupe (Host Groups), la sévérité (severity), le problème, et le temps passé depuis l'incident (Age). En ce qui concerne la colonne "Problem", elle est remontée intrinsèquement par Zabbix mais pour des cas très précis, elle peut être gérée par des scripts (dont le langage n'a pas d'importance) qui vont fixer des conditions sur le retour d'informations de l'équipement et via ces conditions créer les remarques à afficher. Il est possible de voir directement les incidents en levant simplement la tête vers les écrans dédiés à l'affiche de Zabbix dans la salle.

Il ne faut pas oublier de préciser que l'affichage présenté ci-dessous a été réalisé à l'aide de Grafana, un outil qui permet de faire des Dashboards plus esthétiques, comme j'ai pu le voir à l'ITUT. Grafana aura surtout servi à agréger plusieurs Zabbix (celui de l'équipe Système et celui de l'équipe réseaux par exemple),

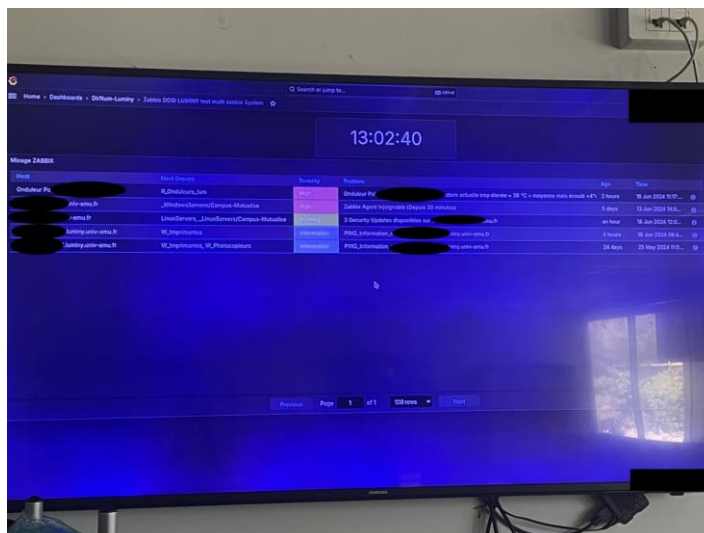


Figure 5 : Ecran Zabbix

3.3.2 GLPI

GLPI, qui signifie Gestionnaire Libre de Parc Informatique, est un logiciel gratuit de gestion des actifs et des services informatiques. Il offre des fonctionnalités conformes à l'ITIL pour le service desk, le suivi des licences et l'audit des logiciels.

Le logiciel facilite le traitement des demandes, en simplifiant les processus de gestion des services, des incidents et des problèmes. Cela assure que les requêtes des utilisateurs sont traitées rapidement et professionnellement, améliorant ainsi la qualité globale du service.

Il permet de gérer les actifs et les configurations informatiques, en suivant les ordinateurs, les périphériques, les imprimantes réseau et leurs composants associés. Depuis la version 10, il intègre une gestion dynamique des inventaires, garantissant une base de données de configuration toujours à jour et fournissant des informations précises sur les actifs.

La DirNum de luminy utilise GLPI dans le cadre de l'inventaire de ses matériels, cela leur permet notamment de pouvoir répondre rapidement à des besoins divers, ou des demandes de matériels pour les usagers du campus (que ça soit des ordinateurs ou même des modules d'alimentation). Lié à cela, on retrouve aussi l'ajout d'un plugin créé dans le cadre du ticketing, celui-ci est disponible depuis seulement quelques mois et a été développé par une entreprise directement aux besoins de la DirNum. Il permet de régler le problème qui concernait les usagers du ticketing pour leurs différentes demandes, je m'explique.

Imaginons qu'un étudiant n'ait plus accès à sa messagerie ENT, celui-ci va envoyer une demande au service informatique (logiquement), mais maintenant imaginons que le problème ne vienne pas du service informatique, mais des ressources humaines qui ont oublié de renseigner l'étudiant dans un document, alors dans ce cas-là, le service informatique dira simplement qu'il va falloir se rediriger vers le bon service (ce qui va forcer l'étudiant à envoyer une nouvelle demande). Comme on le remarque rapidement, cela devient une perte de temps pour les services et l'utilisateur ainsi qu'un flou pour celui-ci (car ici nous étions dans le cas où l'utilisateur a pu recevoir une réponse avec le bon service à contacter).

Donc finalement, le plugin créé sur mesure va prendre la demande de l'utilisateur et via différents choix, à faire en quelques clics de sa part, diriger un peu plus vers où doit s'adresser la demande. Puis si la demande n'est pas assez spécifique, elle est envoyée à l'antenne de Campus qui va s'occuper de réorienter la demande vers le bon service. Maintenant l'utilisateur a largement plus de chances d'avoir une demande résolue et n'a plus à savoir précisément quel est le service vers qui il faut se tourner. Voici, *Figure 6*, où doit se diriger l'utilisateur pour faire sa demande depuis l'ENT. Puis, *Figure 7*, où il faut cliquer une fois sur la page.



Figure 6 : Logo sur l'ENT



Figure 7 : Demande de Ticket

3.4 Gestion du réseau

3.4.1 Différents Switchs

Je voulais revenir sur un point qui durant mon stage m'a interpellé, j'ai pris en main la configuration d'un switch HP A3100. J'ai découvert et noté dans mon cahier les commandes essentielles de Comware 5/7, ce qui m'a permis de réaliser plusieurs tâches importantes. J'ai commencé par créer un utilisateur et configurer un mot de passe chiffré tout en activant les privilèges nécessaires pour cet utilisateur.

Ensuite, j'ai activé le service SSH pour pouvoir me connecter au switch directement depuis un terminal, sans avoir à passer par Putty. En parallèle, j'ai pris soin de configurer les interfaces et d'établir des ports access pour les VLANs, assurant ainsi une bonne organisation. Enfin, j'ai enregistré toutes les configurations effectuées pour m'assurer qu'elles soient sauvegardées et opérationnelles après chaque redémarrage.

Bien que les commandes et les interfaces puissent varier légèrement entre les switchs Cisco et le Comware utilisé par HP, les principes sous-jacents restent les mêmes. Cette familiarité m'a permis de m'adapter rapidement et efficacement à ce nouvel environnement, en appliquant mes connaissances précédentes tout en apprenant de nouvelles commandes spécifiques au HP A3100

Voici les différentes commandes type comware que j'ai pu voir (liste non-exhaustive) :

Description sur une interface :

```
[Switch-example]interface GigabitEthernet 1/0/32
[Switch-example-GigabitEthernet1/0/32]description PC-Management
[Switch-A5500-GigabitEthernet1/0/32]quit
```

Création d'un utilisateur :

```
[Switch-example]local-user admin
New local user added.
```

Configuration de son mot de passe :

```
[Switch-example-luser-admin]password simple bonjour
```

Configuration du niveau d'autorisation de l'utilisateur :

```
[Switch-example-luser-admin]authorization-attribute level 3
```

Autoriser l'utilisateur à utiliser des services:

```
[Switch-example-luser-admin]service-type ftp
[Switch-example-luser-admin]service-type ssh
[Switch-example-luser-admin]service-type terminal
[Switch-example-luser-admin]service-type telnet
[Switch-example-luser-admin]service-type web
```

3.4.2 Le Wifi

J'ai eu l'opportunité d'échanger avec Frédéric GIUDICELLI qui m'a généreusement expliqué les rouages de son travail actuel. Il m'a détaillé le processus de remplacement des bornes WiFi sur le campus.

D'abord, le contrôleur joue un rôle crucial dans la gestion des bornes WiFi. Il vérifie régulièrement les versions des bornes et, s'il détecte une mise à jour de l'OS des bornes sur son serveur, il la déploie automatiquement vers les bornes qui en ont besoin.

Quant au serveur DHCP, son rôle est d'attribuer une adresse IP à chaque borne et de faciliter son ajout au contrôleur.

Les étapes de remplacement sont bien définies :

- Identification des équipements obsolètes à remplacer.
- Collecte des informations de l'ancien appareil (nom, ancienne IP, MAC, numéro de série, numéro inventaire) et mise à jour avec les nouvelles informations (nouveau nom, nouvelle IP, notion d'inventaire, commentaires).
- Configuration de la nouvelle borne via le contrôleur pour vérifier sa détection et son activation avec la nouvelle IP.
- Quand ces opérations ont été effectuées, remplacement de l'ancienne borne par la nouvelle. Il est impératif de choisir une nouvelle IP disponible pour éviter un conflit d'IP sur le contrôleur.
- Enfin, l'installation physique de la borne au bon emplacement sur le campus.

Le placement des bornes n'est pas aléatoire et se fait en fonction des besoins de wifi exprimés. La disposition est faite de façon à optimiser la couverture Wifi.

4 Gestion de l'infrastructure physique

4.1 GRIP : Gestion Réseau au niveau de l'Infrastructure Physique

4.1.1 Description générale

GRIP est une interface web dédiée à la gestion de l'infrastructure physique d'un réseau. Elle permet de modéliser les différents éléments des locaux techniques ainsi que les connexions entre eux, facilitant ainsi l'analyse de leurs interactions. Celle-ci a été majoritairement faite en langage de programmation Perl.

Associée à une base de données, GRIP permet de modéliser les locaux techniques et les armoires réseau qu'ils contiennent, en détaillant la disposition interne des éléments. Elle offre aussi la possibilité de représenter les liaisons entre les armoires et les précâblages des locaux, en stockant des informations sur les caractéristiques des liaisons en fibre optique ou en cuivre, comme les types de connectiques utilisées.

Grâce à cette interface, il est facile de visualiser le nombre de liaisons disponibles et leurs caractéristiques, ainsi que le nombre et la localisation des armoires réseau. Elle permet également de voir les équipements réseau administrables ou non, ainsi que les ports libres disponibles. En somme, GRIP fournit une vue d'ensemble de l'utilisation du réseau, facilitant ainsi la planification de ses évolutions. Le Logiciel est disponible à l'installation gratuitement depuis le lien SourceSup présent dans la partie Sitographie. En ce qui concerne son installation il suffit de suivre le document d'installation se nommant "INSTALLATION.txt".

4.1.2 Présentation de GRIP

Au cours de l'installation initiale, un utilisateur avec tous les droits est créé. Par la suite, il est possible de créer de nouveaux utilisateurs et de leur attribuer des droits spécifiques selon les besoins. Une fois connecté à l'application, l'utilisateur arrive sur la page de synthèse, la page d'accueil de GRIP, qui offre une vue d'ensemble de l'état de l'infrastructure et des éléments qui la composent (Figure 8).

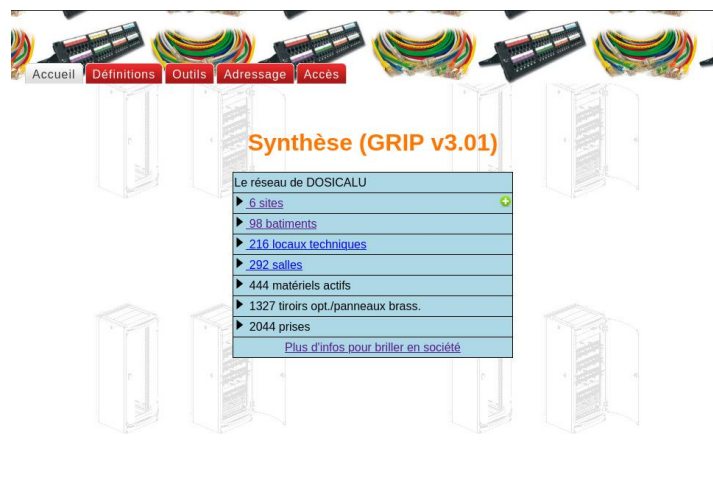


Figure 8 : Accueil GRIP

Un réseau dans GRIP se compose de plusieurs éléments clés (visibles sur la *Figure 8*) :

- Les sites (exemple : Luminy),
- Les bâtiments, rattachés à un site,
- Les locaux (de brassage) ou Salles, rattachés à un bâtiment,
- Les baies ou emplacements (si ce n'est pas dans une baie), rattachés à un local,
- Les prises, rattachées à un local,
- Les équipements actifs ou passifs, rattachés à une baie ou à un emplacement.

La page de synthèse affiche le nombre d'éléments de chaque catégorie. Lorsque la création d'un élément est possible, une icône « + » apparaît à côté du nom de la catégorie. Si un ou plusieurs éléments existent dans une catégorie, un clic sur la catégorie correspondante affiche tous les éléments, permettant ainsi de les sélectionner et de les gérer. Voici ci-dessous (*Figure 9*) l'exemple :

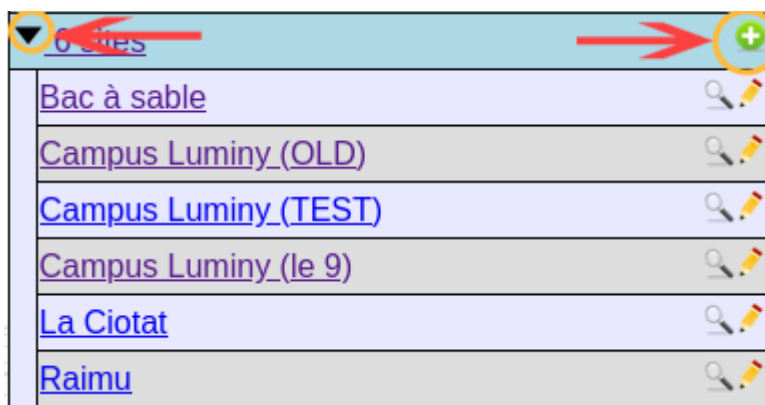


Figure 9 : liste déroulante et icône de création

Pour créer un élément, il faut d'abord sélectionner l'élément de la catégorie supérieure auquel il sera rattaché. Par exemple, pour ajouter une baie, il faut d'abord sélectionner le local correspondant. Une fois la baie créée, des matériels passifs ou actifs peuvent y être ajoutés, définis par une description et une composition. Ici, *Figure 10*, j'ai choisi le site Bac à sable pour pouvoir ensuite dérouler les locaux et en créer un (*Figure 11*) :

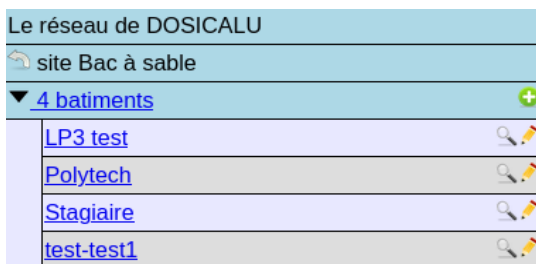


Figure 10 : Choix du site parent

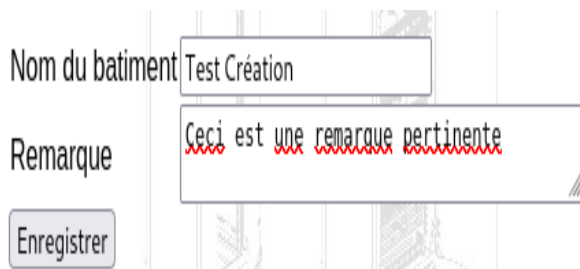


Figure 11 : Création d'un Bâtiment

Au démarrage, la base de données de GRIP ne contient aucune définition de matériels par défaut pour éviter d'inclure des définitions obsolètes. Cependant, des exemples sont disponibles dans le dossier « descriptions » de l'installation. Pour importer une description, il suffit d'aller dans l'onglet « Définitions », puis « Matériels », et de suivre le lien « Liste des descriptions ». Un fichier, *Figure 12*, de définition (au format *.grip_desi_export) peut alors être importé et, une fois validé, le matériel devient utilisable dans les baies de brassage.

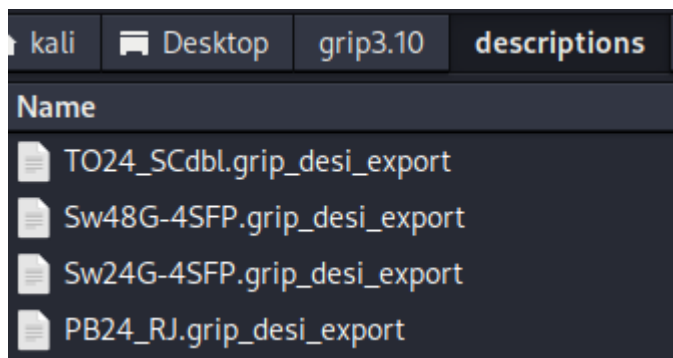


Figure 12 : Définition importable

GRIP permet également de créer ses propres matériels ou de modifier les descriptions existantes. Les définitions de matériels contiennent des informations telles que le numéro de port et les emplacements pour le brassage et le précâblage.

Pour ajouter des matériels dans une baie (*Figure 13*), il faut naviguer depuis la page de synthèse jusqu'au local correspondant, puis cliquer sur le nom de la baie. Les nouveaux matériels peuvent être placés en bas de la page. Une fois un matériel ajouté, il est possible de brasser ses ports vers d'autres équipements du même local ou, pour les matériels de type lien (comme les tiroirs optiques ou les panneaux de brassage), d'ajouter des précâblages vers des prises ou d'autres matériels de type lien (ce qui en somme veut dire que nous pouvons joindre d'autres locaux, d'autres bâtiments de cette manière).

Le choix sur le domaine VLAN Trunking Protocol (VTP), *Figure 13*, va servir à définir si ce matériel peut créer, modifier et supprimer des VLANs tout en propageant les informations de VLAN à tous les commutateurs du domaine (Serveur VTP), ou à l'inverse recevoir seulement les informations des serveurs VTP en les appliquant localement (Client VTP).

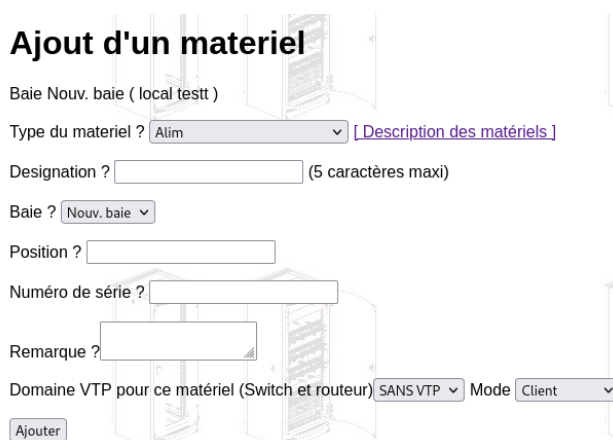


Figure 13 : Ajout d'un matériel sur GRIP

4.2 Recherche et comparatif des solutions

Au cours de mon stage et en vue de répondre à la problématique de mon sujet, je me suis mis activement à la recherche d'une solution déjà disponible sur le marché pour pouvoir la comparer à GRIP, mais aussi avec les autres disponibles. J'ai alors dû établir certains critères de sélection, en commençant d'abord par le fait d'être une solution OpenSource et entièrement gratuite (ou partiellement si quelques fonctionnalités jugées inutiles). Ensuite, en prenant ces deux directions, on retrouvait les critères au niveau de l'interface (sa disposition/son harmonie), la facilité de prise en main de la solution, mais plus essentiellement si elle coche les plus de cases dans les besoins de l'entreprise. Ces besoins étant l'accès simple à l'information voulue ainsi qu'une visualisation graphique du matériel, tout en pouvant garder à jour facilement l'infrastructure et l'améliorer en utilisant directement la solution.

Présentation rapide des solutions non retenues pour les tests :

Pandora FMS

Description :

Pandora FMS est un logiciel de supervision et de monitoring qui utilise le protocole SNMP pour découvrir les services voisins.

Problèmes :

- Manque de fonctionnalités graphiques
- Interface complexe à prendre en main
- Solution payante pour obtenir plus de fonctionnalités ou du support
- Implémentation fastidieuse

Remarque :

Pandora FMS est jugé moins utile car ses fonctionnalités peuvent être remplacées par NetDisco, qui est gratuit.

Device42

Description :

Device42 est une solution de gestion de la configuration et des actifs IT, Information Technology, reconnue pour ses fonctionnalités graphiques avancées.

Points forts :

- Interface graphique très développée

Problème :

- Solution payante

NetDisco

Description :

NetDisco est une application web open-source pour la gestion et la cartographie des réseaux. Il est particulièrement efficace pour découvrir et visualiser les équipements réseau en utilisant le protocole SNMP.

Points Forts :

- Découverte Automatisée avec SNMP
- Visualisation de Réseau
- Open Source, donc gratuit et personnalisable selon les besoins spécifiques.

Problèmes :

- Interface Graphique Limitée : L'interface peut paraître rudimentaire par rapport à des solutions commerciales plus avancées.
- Manque de fonctionnalité par rapport à GRIP sur la visualisation de l'infrastructure, des baies par exemple

Nous pouvons ensuite passer aux logiciels qui ont été retenus pour les tests :

OpenDCIM

Description :

OpenDCIM est un logiciel OpenSource de gestion des infrastructures des centres de données. Avec pour créateurs clés : Jose Miguel Gomez, Wilbur Longwisch et Scott Miliken

Points forts :

- Interface simple à prendre en main
- Installation rapide
- Possibilité de réaliser schématiquement un local/une salle avec la disposition de ses baies
- Importation de divers matériels ou différentes définitions de lieux en masse possible
- Tracabilité d'une connexion entre deux matériels

Problèmes :

- Interface inesthétique, Netbox proposant largement mieux et avec plus de fonctionnalités
- Comporte des bugs mineurs d'affichage

NetBox

Description :

NetBox est un logiciel de gestion de l'infrastructure et des adresses Internet Protocol (IP) (IPAM, Internet Protocol Address Management) avec une interface épurée et des fonctionnalités avancées.

Points forts :

- Interface épurée et très intéressante
- API disponible et facilement utilisable (documentation disponible)
- Base de données PostgreSQL
- Importation de données en CSV, YAML et JSON
- Grande variété de plugins très intéressants
- Définition de matériels disponible via import avec un fichier sur le GitHub de la communauté
- Toujours en développement avec des mises à jour régulières
- Visualisation des baies disponible
- Communauté très investie et répond rapidement en cas de problème

GLPI

Description :

GLPI est un logiciel libre de gestion et d'inventaire de parc : gestion des services informatiques et gestion des services d'assistance.

Points forts :

- Interface équivalente à celle de NetBox
- Plugins disponibles (mais moins nombreux que sur NetBox)

Problèmes :

- Manque de visualisation graphique (exemple : la visualisation nécessaire des baies)
- Plugins souvent obsolètes et abandonnés, rendant leur installation impossible, qui est le point plus contraignant
- Fonctionnalités limitées sans les plugins, la solution est plutôt utilisée pour faire de l'inventaire et donc perd toute son utilité sans les plugins (si on se base sur les besoins de l'entreprise)
- Certains plugins nécessitent un paiement pour les utiliser
- Une communauté qui a presque abandonné les plugins (voir complètement), impossible de les tester en restant sur la nouvelle version de GLPI (GLPI 10), donc plus aucune utilité de continuer les tests sur cette solution

Après avoir essayé plusieurs solutions, j'ai trouvé que NetBox répondait le mieux aux besoins de l'entreprise. Son interface moderne et intuitive la rend vraiment agréable à utiliser, et la visualisation graphique des racks et des équipements facilite grandement la gestion et le suivi de l'infrastructure réseau, comme le permettait GRIP. En plus, NetBox permet de la gestion des adresses IP, des VLANs et des connexions entre équipements, ce qui améliore la cohérence et la fiabilité des données (même si dans notre cas, la solution ne sera pas utilisée pour l'adressage IP ou des VLANs). L'API REST de NetBox offre aussi une grande flexibilité pour automatiser les tâches courantes et intégrer la solution avec d'autres outils, dans notre cas elle servira à la migration des données. Enfin, la communauté active et les mises à jour régulières garantissent que NetBox reste toujours adapté aux besoins évolutifs des entreprises.

Avec ce choix de solution, j'avais différentes voies possibles. La première option serait de tout migrer sur NetBox d'un coup. Cette approche représenterait un travail énorme et risquerait de ne pas être faisable dans les délais. La seconde option, plus pragmatique, consisterait à procéder à des migrations par étapes. Cela permettrait de garder un lien entre GRIP et NetBox, en basculant petit à petit sur l'utilisation et l'ajout d'éléments sur NetBox. Cette méthode graduelle offrirait la flexibilité nécessaire pour intégrer NetBox sans perturber les opérations courantes et garantirait une transition en douceur tout en évaluant continuellement l'efficacité de la nouvelle solution.

Je tiens à préciser que, bien que j'aie finalement opté pour la recherche d'une nouvelle solution, une option pour mon projet de stage aurait été de continuer à développer l'application déjà existante, GRIP. Après une longue réflexion et plusieurs journées d'exploration de GRIP et de son code, j'ai décidé, avec l'accord de mon tuteur, de mettre cette idée de côté.

GRIP a été entièrement développée par mon tuteur, Pascal Mouret, et deux de ses collègues. Se plonger dans le code d'une autre personne demande déjà un temps considérable pour comprendre sa logique. En plus de cela, je n'avais jamais programmé en langage PERL, ce qui ajoutait une difficulté supplémentaire. De nombreuses bibliothèques et fonctions souvent utilisées dans le programme ont été intégralement créées par mon tuteur et ses collègues. J'ai d'abord tenté d'apprendre progressivement les subtilités de ce langage, mais j'ai rapidement décidé de chercher de meilleures solutions afin de pouvoir proposer un résultat concret à la fin de mon stage.

4.3 Choix de la solution : NetBox

4.3.1 Installation

Dans cette partie je ne vais pas présenter l'installation complète de NetBox, car celle que j'ai pu faire au cours de mon stage est disponible sur le Wiki de l'équipe réseaux et comporte des IPs à ne pas montrer. Je vais plutôt présenter les différentes étapes et ce qui est utilisé pour pouvoir l'installer.

Prérequis :

- Un serveur avec un système d'exploitation compatible, tel que Ubuntu ou CentOS.
- Accès root ou des privilèges sudo.
- Une base de données PostgreSQL.
- Python 3.6 ou plus récent.

Dans mon cas, j'ai d'abord fait une installation sur ma machine physique pour faire tous les tests nécessaires, puis je suis passé sur un serveur de production où j'ai pu faire l'installation (ce qui permet d'avoir la solution et de faire un snapshot pour revenir en arrière en cas de problème).

1. PostgreSQL database :

PostgreSQL occupe une position centrale et essentielle au sein de l'infrastructure de NetBox en tant que système de gestion de base de données. Il est responsable du stockage sécurisé, de la gestion efficace et de la récupération rapide des données relatives à l'infrastructure réseau. Une configuration précise est cruciale pour assurer le bon fonctionnement de NetBox, sans oublier le fait que pour pouvoir bien utiliser l'API, la nécessité d'un accès à une base structurée et un accès aux données bien rangées se fait bien ressentir.

2. Redis

L'utilisation de Redis permet à NetBox de gérer efficacement les opérations de mise en file d'attente, notamment pour les tâches asynchrones telles que la gestion des tâches de traitement en arrière-plan, les notifications en temps réel, et d'autres processus nécessitant une exécution parallèle sans bloquer les opérations principales de l'application.

En tant que base de données clé-valeur en mémoire, Redis permet à NetBox de stocker temporairement des données fréquemment utilisées en mémoire vive, offrant ainsi des temps de réponse plus rapides et une réduction de la charge sur la base de données principale.

3. NetBox

Cette partie est simplement l'installation de tous les paquets nécessaires à l'application, la récupération de son archive pour l'installer, ainsi qu'une bonne partie sur ses fichiers de configuration. On va notamment créer le premier super-utilisateur, renseigner les IP autorisées à pouvoir joindre NetBox, ou encore lancer un script qui va nous créer l'environnement virtuel Python, l'installation des paquets Python nécessaires et bien d'autres paramètres qui vont finaliser l'installation.

4. Unicorn

Unicorn, ou Green Unicorn, est un serveur Hypertext Transfer Protocol (HTTP) Web Server Gateway Interface (WSGI) rapide et léger conçu pour Python. NetBox l'utilise comme interface entre son application Python et le serveur HTTP, assurant ainsi des performances optimales et une gestion robuste des requêtes HTTP.

Il est choisi pour sa capacité à gérer efficacement de multiples connexions simultanées en utilisant un modèle de processus maître-travailleur. Cela permet à NetBox de traiter les requêtes de manière asynchrone et de s'adapter facilement aux pics de charge, assurant des performances élevées même lors d'une utilisation intensive.

5. Serveur HTTP

Comme vu un bon nombre de fois lors de mon parcours à l'IUT, l'installation d'un serveur HTTP est le béaba en réseaux et informatique. Ici, on a le choix entre Nginx ou Apache en guise de serveur HTTP, personnellement j'ai choisi de passer par Nginx mais j'ai également pu passer par Apache, ce qui n'a absolument rien dégradé à mon installation. Je tiens à préciser que cette étape sert pour héberger l'applications web.

4.3.2 Plugins et développement

Le point qui fait la différence sur NetBox est réellement ses plugins et leur utilité. Ici, je vais présenter les plugins que j'ai décidé de garder et d'utiliser. Les sites permettant de les récupérer (sur GitHub) seront mis dans la partie **Sitographie**.

Device View :

Ce plugin permet de visualiser un matériel, ses ports et la possibilité de cliquer sur un port pour voir sa trace (d'où il est connecté et vers où). C'est le plugin majeur car il ajoute un aspect graphique à NetBox. Pour le faire fonctionner il faut utiliser des templates (modèles) qui utilisent le `grid-template-area` en CSS. On peut trouver en Annexe un exemple de modèle que j'ai pu réaliser pour représenter un matériel en particulier

Reorder Rack :

Ce plugin permet de réordonner, comme son nom l'indique, une baie en utilisant un outil de « drag and drop » (glisser et déposer). Son utilisation est simple, on se rend sur la baie voulue, puis clique le bouton « Reorder », enfin on clique sur le matériel sans relâcher et on le dépose à l'endroit voulu.

Topology view :

Ce plugin va permettre de créer une carte de la topologie réseau à partir des appareils dans NetBox. Les connexions sont basées sur les câbles qui ont été créés. Le plugin prend en charge le filtrage par nom, site et rôle de périphérique.

NetBox-ui-plugin :

Celui-ci fait la même chose que le « Topology view » mais à quand même été choisi pour les possibilités de filtrage et d'utilisation. Le point assez intéressant est que l'on peut regrouper plusieurs périphériques pour minimiser la carte et la rendre pliante et dépliant. La différence avec l'autre est que celui-ci n'est disponible qu'en choisissant un site, on ne peut pas le faire fonctionner depuis un bâtiment ou un local.

4.3.4 Migration des données

En annexe se trouve l'un des codes utilisés pour la migration des données (celui des locaux car il reprend la même méthode que pour les bâtiments ou les baies).

Préparation du fichier CSV :

Tout d'abord, j'ai préparé un fichier CSV nommé `locauxFinal.csv`. Ce fichier contient les informations nécessaires pour chaque location à créer dans NetBox. Chaque ligne du fichier CSV représente une entrée distincte pour une nouvelle location, comprenant des détails tels que le nom de la location (`name`), le slug (`slug`) pour l'identification unique dans l'URL, le type de la location (`type`), et éventuellement le slug du parent (`parent`) pour les locaux.

Structure du script Python :

En utilisant Python, j'ai structuré un script qui interagit avec l'API REST de NetBox à l'aide de la bibliothèque `requests`. J'ai configuré les paramètres de base, y compris l'URL de l'API de NetBox (`NETBOX_URL`) et le token d'API (`API_TOKEN`) pour l'authentification. Le script contient des fonctions dédiées pour la création de locations (`create_location`) et la récupération de l'ID du parent à partir de son slug (`get_parent_id`).

Création de locations dans NetBox :

La fonction principale du script (`migrate_locations`) parcourt chaque ligne du fichier CSV. Pour chaque ligne :

- Elle nettoie et normalise le nom et le slug de la location pour s'assurer qu'ils respectent les exigences de format de NetBox.
- Elle détermine le type de la location (bâtiment ou local) en fonction du champ `type` du CSV.
- Si la location est un local, elle vérifie si un parent est spécifié dans le CSV et obtient l'ID du parent à partir de son slug.

Elle appelle ensuite la fonction `create_location` avec les paramètres nécessaires pour créer la location dans NetBox, en incluant le site ID, le type de la location, et éventuellement l'ID du parent.

Exécution du script :

Pour exécuter le script, j'ai fourni le chemin absolu du fichier CSV (`CSV_FILE_PATH`) ainsi que l'ID du site dans NetBox où je voulais intégrer les nouvelles locations (`SITE_ID`). J'ai veillé à ce que le fichier CSV soit correctement formaté et contienne toutes les données requises selon les spécifications préétablies. Le script a été conçu pour gérer les erreurs potentielles et a affiché des messages de validation ou d'erreur pour chaque opération effectuée.

5 Conclusion

Au cours de mon stage, j'ai entrepris une exploration approfondie des solutions de gestion d'infrastructures réseau disponibles sur le marché, dans le but de répondre efficacement aux besoins spécifiques de l'entreprise. Après avoir évalué diverses options, mon choix s'est orienté vers NetBox.

Ce stage m'a offert une expérience précieuse dans le domaine de la gestion d'infrastructures réseau et de l'intégration de solutions logicielles. J'ai pu développer mes compétences techniques en Python et en gestion de bases de données, tout en acquérant une compréhension approfondie des défis liés à la gestion d'infrastructures à grande échelle (campus).

J'ai dû faire face à plusieurs défis significatifs qui ont influencé le déploiement de NetBox au sein de l'entreprise. L'un des principaux obstacles a été la contrainte de temps pour mener à bien la migration complète vers NetBox. En raison de la diversité et de la complexité des équipements réseau, certaines configurations n'ont pu être migrées car demandant plus de temps de réflexion et de développement. J'ai tout de même pu effectuer des configurations manuelles pour certains locaux (disponible en annexe) afin de démontrer l'efficacité de l'application.

À l'avenir, je vois NetBox comme une composante stratégique, offrant non seulement une gestion efficace, mais également une base solide pour l'automatisation et l'évolutivité des opérations de l'entreprise sur l'infrastructure. Je suis reconnaissant pour cette opportunité d'apprentissage et je suis prêt à sur cette voie pour en apprendre toujours plus en réseaux et en informatique.

Finalement, j'ai pu arriver à un résultat dont je suis fier, malgré les problèmes restants, et à l'aide des cours que j'ai pu suivre via mes deux premières années en réseaux et télécommunications, j'ai pu facilement comprendre certains concepts ou utiliser mes connaissances tout en en acquérant plus pour aller plus loin.

6 Remerciements

Le stage en entreprise effectué au sein de l'équipe Réseaux de la Dirnum Luminy est ma première expérience professionnelle dans le milieu des Réseaux et Télécommunications.

Je voudrais remercier mon maître de stage, Pascal MOURET, pour m'avoir fait confiance, donné des responsabilités pendant la durée du stage ainsi que pour toutes ses explications.

Je tiens aussi à remercier Laurent MICHAUDET, Sébastien NANTAS et Frédéric GIUDICELLI, pour leur accueil, ainsi que pour leurs nombreux conseils, leur écoute, mais également pour le temps qu'ils m'ont accordés.

Je remercie également tout le personnel de la Dirnum Luminy, notamment Gérard MILHAUD et Stéphanie MARTINEZ pour m'avoir accueilli et intégré dans un environnement convivial.

7 Glossaire

BIATSS : ensemble de fonctions et de personnels de la Fonction publique composés de Bibliothèques, Ingénieurs, Administratifs, personnels Techniques, Sociaux et Santé.

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

Bootable : Contient un programme qui permet de démarrer un ordinateur sur le périphérique dit « bootable ».

Metapackage : Regroupe plusieurs packages, facilitant l'installation d'un ensemble de logiciels.

GRUB boot loader : Programme de démarrage utilisé pour lancer un système d'exploitation.

Mind-mapping : Transformer une longue liste d'idées en un schéma attrayant.

Ticketing : Système de gestion de requêtes créant un ticket pour chacune.

Baies : Armoires technique qui centralisent des éléments de réseaux informatiques et de téléphonie.

Equipements actifs/passifs : Passifs : Composants qui connectent et dirigent les signaux sans alimentation électrique. Actifs : Dispositifs qui traitent et amplifient les signaux en nécessitant une alimentation électrique.

Précâblage : Partie du câblage qui resterait fixe, par exemple l'arrière d'une prise.

Tiroir optique : Equipement passif qui regroupe tous les câbles fibres optiques dans un même endroit, pour ensuite les connecter à des équipements réseaux actifs.

VLAN : Virtual local area network, réseau local virtuel, est un réseau informatique logique indépendant.

Commutateur : Permet de relier (virtuellement) deux périphériques à l'aide de liens physiques.

Bug : Défaut de conception d'un programme informatique.

API : Application programming interface, interface logicielle qui permet de « connecter » un logiciel à un afin d'échanger des données.

CSV/YAML/JSON : Formats de données structurées.

Snapshot : Sauvegarde à un instant T (point de restauration) d'un système

Tâches asynchrones : opération qui s'exécute indépendamment du flux principal du programme, permettant à d'autres opérations de continuer sans attendre sa fin

Super-Utilisateur : Utilisateur ayant tous les droits sur un logiciel ou système

Processus Maître-Travailleur : modèle d'architecture où un processus maître distribue des tâches à plusieurs processus travailleurs qui exécutent les tâches en parallèle

Token : Jeton sécurisé d'authentification pour des systèmes informatiques ou des API

8 Sitographie

<https://www.kali.org/docs/installation/hard-disk-install/> (Doc. Installation Kali Linux), Juin 2024

<https://www.syloe.com/glossaire/logiciel-zabbix/> (Qu'est-ce que Zabbix ?), Mai 2024

<https://sourcesup.renater.fr/projects/grip/> (Doc. GRIP), Juin 2024

<https://netboxlabs.com/docs/netbox/en/stable/installation/> (Installation NetBox), Juin 2024

<https://github.com/peterbaumert/netbox-device-view> (Doc. Plugin), Juin 2024

<https://github.com/netbox-community/netbox-reorder-rack> (Doc. Plugin), Mai 2024

<https://github.com/netbox-community/netbox-topology-views> (Doc. Plugin), Mai 2024

<https://wiki.opendcim.org/wiki/index.php/Ubuntu> (Doc. Installation OpenDCIM), Mai 2024

https://faq.teclib.com/03_knowledgebase/procedures/install_glpi (Doc. Installation GLPI), Mai 2024

<https://demo.netbox.dev/api/> (demo API NetBox pour mieux comprendre), Juin 2024

9 Table des illustrations

<i>Figure 1 : Organigramme hiérarchique de la DirNum</i>	7
<i>Figure 2 : GRUB boot loader</i>	9
<i>Figure 3 : metapackages</i>	9
<i>Figure 4 : Exemple sur le logiciel Freeplane</i>	10
<i>Figure 5 : Ecran Zabbix</i>	11
<i>Figure 6 : Logo sur l'ENT</i>	12
<i>Figure 7 : Demande de Ticket</i>	12
<i>Figure 8 : Accueil GRIP</i>	15
<i>Figure 9 : liste déroulante et icône de création</i>	16
<i>Figure 10 : Choix du site parent</i>	16
<i>Figure 11 : Création d'un Bâtiment</i>	16
<i>Figure 12 : Définition importable</i>	17
<i>Figure 13 : Ajout d'un matériel sur GRIP</i>	17