Institut Universitaire de Technologie,

Aix-Marseille Université

RAPPORT DE STAGE

Diplôme Universitaire de Technologie

Spécialité Réseaux et Télécommunications

STAGE AU PÔLE INFORMATIQUE DE L'I2M

GUERRERO Maximilien

Institut de Mathématiques de Marseille

Responsable entreprise : Olivier Chabrol

Responsable académique : Tin Nguyen

2019

Remerciements

Je tenais à remercier l'équipe du service informatique du site de Château-Gombert et de Luminy pour son accueil et à son soutien technique. J'ai pu apprendre beaucoup grâce à celle-ci, non seulement en connaissances et compétences mais aussi en termes de rigueur.

Plus particulièrement, j'aimerais remercier Olivier CHABROL pour sa disponibilité et son engagement durant mon stage.

Je remercie également Pierre BARTHELEMY de m'avoir donné la possibilité de travailler avec le service informatique de Luminy.

Je remercie de même M. NGUYEN pour son encadrement pendant celui-ci ainsi que M. KORADJIAN de m'avoir mis en contact avec l'Institut de Mathématiques de Marseille.

Merci à Hugo BLACHERE et à son sérieux de m'avoir accompagné durant ce stage.

De façon générale, je remercie la direction d'avoir accepté ma candidature en tant que stagiaire dans leur institut et de m'avoir permis de ce fait de vivre une expérience très enrichissante.

Table des matières :

I. Introduction :

A. Présentation de l'entreprise :

L'I2M (Institut de Mathématiques de Marseille) est localisé sur trois sites :

- Château Gombert ;
- Luminy ;
- Saint Charles.

C'est une unité mixte de recherche du CNRS. L'I2M est le résultat de la fusion en 2014 du Laboratoire d'Analyse, Topologie et Probabilité (LATP) et de l'Institut de Mathématiques de Luminy (IML). On compte plus de 200 personnes actives au sein de l'I2M, dont en majorité sont enseignants-chercheurs et doctorants. Sa structure scientifique repose sur cinq groupes :

- Analyse Appliquée (AA) ;
- Arithmétique, Géométrie, Logique et Représentations (AGLR) ;
- Analyse, Géométrie, Topologie (AGT) ;
- Géométrie, Dynamique, Arithmétique et Combinatoire (GDAC) ;
- Mathématiques de l'Aléatoire (ALEA).

À la suite de la fusion, l'I2M conserve deux sites géographiques principaux, ce sera pourquoi nous serons environ un jour sur deux entre Luminy et Château-Gombert.

A Château-Gombert, les locaux se situent sur le site du technopôle. Dans le Centre de Mathématiques et d'Informatique (CMI), le rez-de-chaussée sert uniquement à l'enseignement et à l'administration. Les étages supérieurs, eux, sont des locaux de recherche. Le pôle informatique se trouve au premier étage avec les chercheurs.

Nos locaux à Luminy se situent dans le TPR2, qui est un bâtiment de l'université, toujours au niveau des bureaux des chercheurs.

B. Présentation du cadre technique :

Hugo et moi-même avons effectué notre stage au sein de la même entreprise, c'est pourquoi il sera cité plusieurs fois dans mon rapport. Durant mon stage, je suis resté avec le responsable du pôle informatique, Olivier CHABROL, et avec deux des assistants du pôle informatique, Guillaume CHAGNARD et Augustino DE SOUZA.

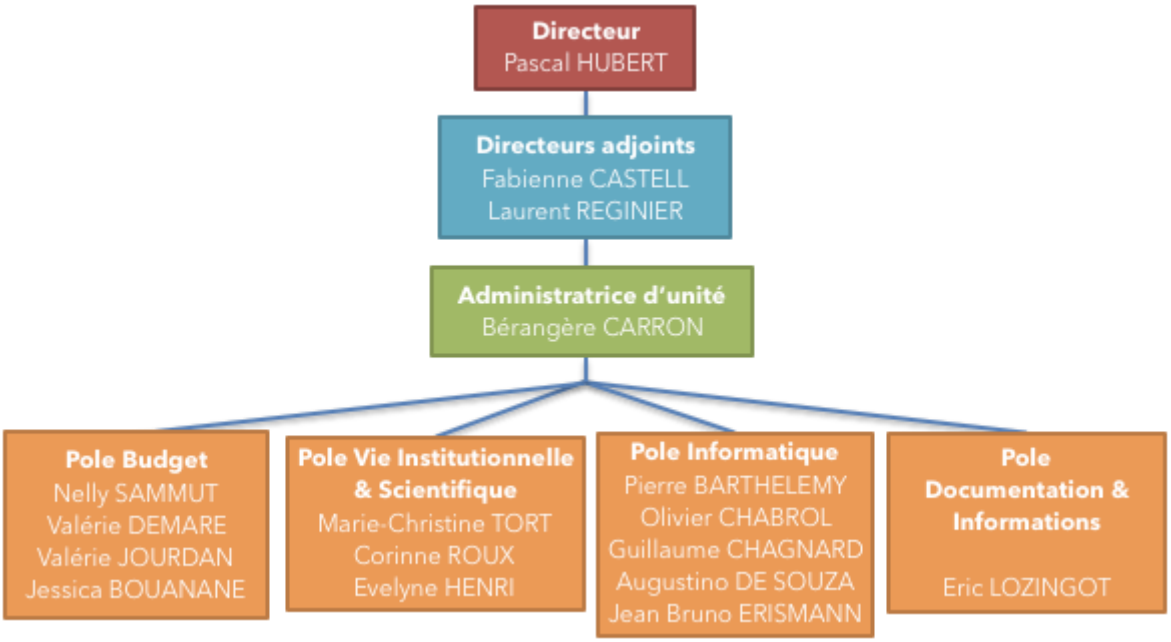


Figure 1 : Organigramme

Nous avons à disposition deux ordinateurs fixes sous Linux, qui est un système d'exploitation. Cependant nous préférons utiliser nos ordinateurs personnels pour une question de praticité, ce qui nous permet de travailler aussi bien à Luminy qu'à Château-Gombert.

Dans les locaux techniques de l'I2M nous avons accès à la salle des serveurs, où sont mis à disposition des casques anti-bruit.

Pour information, le parc informatique du site de Château Gombert est composé à l'heure actuelle de 66% d'ordinateurs sous OS X (Apple), de 32% d'ordinateurs sous Unix et de 2% d'ordinateurs sous Windows.

De plus, l'I2M est chargé de gérer toute la partie recherche du réseau, tandis que la DOSI, elle, gère la partie enseignement (faculté). Cette organisation complique la configuration du réseau.

II. Présentation des travaux réalisé durant le stage :

A. Réseau :

A.1. Topologie du réseau :

La topologie du réseau de Luminy est une topologie dite en étoile, c'est-à-dire que les équipements du réseau sont reliés à un système central qu'on appelle le nœud. Ce dernier peut être un routeur ou un switch. A Château-Gombert, la topologie est bien différente, dû à un problème de conception du bâtiment, qui a impacté directement l'état du réseau.

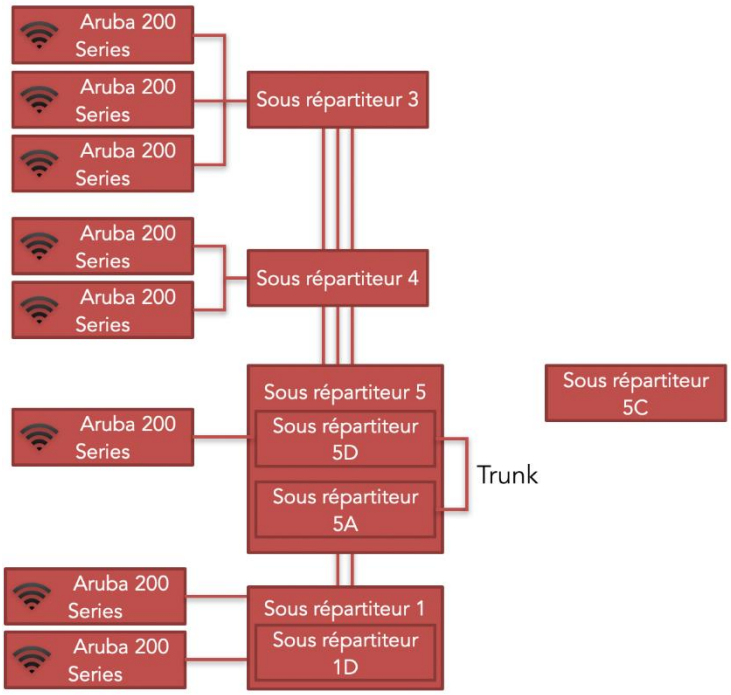


Figure 2 : Topologie réseau Château-Gombert

A.2. Les commutateurs :

Dès les premiers jours, Olivier CHABROL nous a montré comment s'organisait le réseau de l'I2M à Château Gombert, avec les différentes salles techniques et l'ensemble les **commutateurs***. Afin de conserver un historique des informations techniques, l'équipe informatique de l'I2M utilise un wiki (**dokuwiki***). Conscients du travail à fournir, nous avons été incités à le maintenir un maximum à jour, comme un journal de bord.

Notre première mission est de mettre à jour la configuration des switchs en supprimant des **VLANs*** inutiles. Pour les supprimer, nous avons dû nous connecter en **SSH*** à tous les commutateurs. Pour cela, nous devons préciser l'algorithme de chiffrement qu'utilise le switch. Avec quelques recherches sur internet, nous les avons trouvés. Pour les HP, ce sera « 3des-cbc ».

Une fois connectés, nous devrons garder uniquement les numéro 1, 21 et 200. La commande à effectuer est la suivante : « no vlan id ».

```
s2610pwr-CMI_6a# show vlans

 Status and Counters - VLAN Information

  Maximum VLANs to support : 32
  Primary VLAN : DEFAULT_VLAN
  Management VLAN :

  VLAN ID Name                            | Status     Voice Jumbo
  ------- ------------------------------- + ---------- ----- -----
  1       DEFAULT_VLAN                    | Port-based No    No
  21      CMI-rech                        | Port-based No    No
  22      CMI-ens                         | Port-based No    No
  24      CMI-sunray                      | Port-based No    No
  26      sunray-kiosk                    | Port-based No    No
  27      jumpstart-re                    | Port-based No    No
  29      vlansgd                         | Port-based No    No
  114     PedagoPriv                      | Port-based No    No
  129     TOIP-old                        | Port-based No    No
  200     TOIP2                           | Port-based Yes   No
```

Figure 3 : VLANs

J'ai rencontré un problème quand l'un des switchs à mettre à jour n'a pas accepté la connexion SSH.

J'ai fait de nombreuses recherches afin d'en déterminer la cause. Le problème est que le switch Cisco était trop vieux pour accepter une connexion SSH comme le demandait **OpenSSH*.** Ce dernier nécessite une **clé de sécurité*** trop grande par rapport à ce que propose le switch. Nous avons résolu le problème en installant **Putty*** qui, lui, est plus permissif quant à la sécurité des clés.

Les autres commutateurs sont de la marque HP et sont plus récents ; nous n'avons donc eu aucun problème de mise à jour.

Pour savoir si la suppression de ces VLANs n'avait pas impacté l'état du réseau, nous avons effectué des contrôles à l'étage où ces commutateurs délivraient internet. Nous avons testé l'accès à ce dernier. Tout fonctionnait.

Grâce à ces contrôles, nous avons détecté que certains **téléphones IP*** n'arrivaient pas à s'attribuer d'adresses. Nous avons donc remis le VLAN 114 (TOIP) sur le switch en question et nous avons redémarré tous les terminaux bloqués.

B. Développement logiciel :

Une grande partie de mon stage sera consacré aux développements de logiciels.

B.1. Qu'est-ce que Symfony et pourquoi ?

Symfony est un framework PHP développé par la société française SensioLabs. Le PHP est un langage de programmation côté serveur permettant de rendre les pages d'un site web dynamiques. Un Framework est un ensemble cohérent de composants logiciels permettant de créer les fondations et les grandes lignes d'une application. Pour simplifier, il s'agit d'une bibliothèque de logiciels qui sera la boîte à outils d'un informaticien.

Symfony a intégré par défaut l'ORM (Object-Relational Mapping) doctrine. Une ORM est un logiciel qui permet aux développeurs de faire une correspondance entre un objet informatique et une base de données relationnelle.

De plus, SensioLabs a aussi développé un moteur de **template*** nommé twig. Ces templates seront destinées à l'affichage afin de bien distinguer les parties codes des parties graphiques. En général, le twig permet aux graphistes de travailler avec des développeurs en simplifiant la tâche du code à ces deux corps de métier.

Symfony est un framework basé sur le modèle vue contrôleur **(MVC)** ; pour faire simple c'est une façon d'organiser une application. Cette organisation est composée de trois types de modules ayant chacun ses responsabilités :

- Un modèle qui va contenir les données à afficher ;
- Une vue qui sera l'esthétique d'une interface graphique ;
- Un contrôleur qui contiendra la logique des actions effectuées par l'utilisateur, afin de concentrer la cohérence du programme.

Symfony étant l'un des plus gros framework PHP utilisé à travers le monde, il possède une forte communauté et une excellente documentation. C'est la raison de notre choix.

B.2. Travail réalisé sur le logiciel DHCP :

Ici, notre mission était de débugger certains points du projet qu'avaient effectué les stagiaires de l'année précédente. C'est une application web qui permet d'intégrer au **pool dhcp*** des utilisateurs afin qu'il puisse accéder à internet. Cette application permet aux techniciens qui ne sont pas habituer à manipuler le protocole DHCP à rajouter des hôtes via une interface web. Pour mener à bien cette mission, j'ai dû installer un **dual boot*** sur mon ordinateur personnel, afin d'avoir Windows et **Ubuntu*.** Dans le but de consolider mes compétences sous linux.

De plus, la documentation sur Ubuntu en administration système est très complète, ce qui me permet de ne jamais rester bloqué sur une étape d'une mission. L'installation du dual boot est assez simple ; il suffit de **flasher*** une clé USB afin de lui mettre une image du système d'exploitation de Ubuntu dans sa mémoire.

Nous aurons besoin de mettre la clé USB sur l'ordinateur et de le redémarrer. Lors du redémarrage nous devrons accéder au **BIOS*** de la machine et de démarrer sur la clé USB. Une fois l'ordinateur démarré sur Ubuntu, il ne nous reste plus qu'à faire l'installation de ce dernier et il nous proposera de s'installer dans une **partition du disque*** différente de Windows.

Quand l'installation se termine, à chaque redémarrage nous aurons le choix de démarrer soit sur Windows, soit sur Ubuntu, avec l'interface suivante :

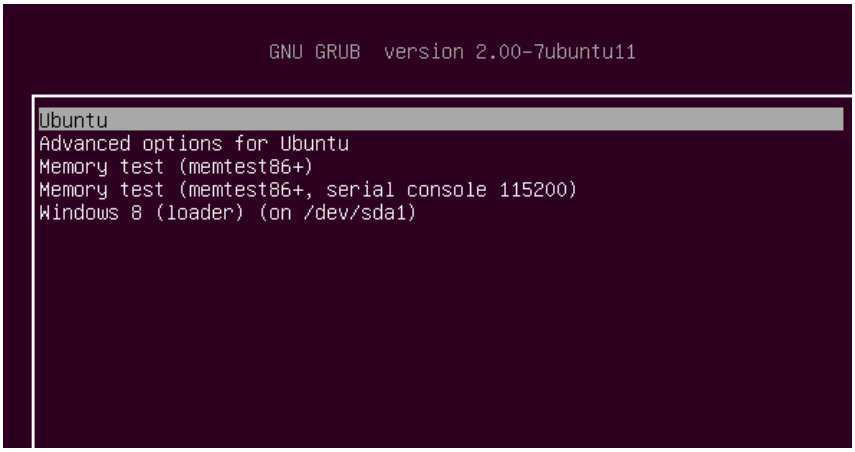


Figure 4 : Dual boot

De plus, pour héberger sur ma machine l'application DHCP et donc créer un environnement de test, j'ai dû installer et faire tourner un serveur apache2.

Pour cela, il faut installer le **service*** apache2 et php à l'aide des commandes suivantes (tout en mettant à jour la **bibliothèque aptitude***) :

```
apt update
apt upgrade
apt install apache2
apt install php php-xml
```

Figure 5 : Installation Apache

Puis, nous créons un hôte virtuel pour y importer la configuration correcte. Pour cela, nous devons se rendre dans le dossier /etc/apache2/sites-availables et créer un fichier avec l'extension .conf. Vous trouverez en annexe la configuration de l'hôte virtuel.

Après ces étapes de configuration, nous avons à exécuter une commande pour activer le site : `a2ensite`

Et autoriser le rewrite avec cette commande : `a2enmod rewrite`

Afin d'empêcher quiconque d'accéder à notre site nous mettons en place une demande d'authentification via un .htaccess basé sur un .htpsswd. Pour ce faire, nous insérons la ligne suivante : `admin:$apr1$RjR/23kz$PEe0Axuf4t3ioq.HfXkKv/`

L'objectif est d'abord de comprendre le code de nos prédécesseurs. Le projet étant réalisé avec Symfony, il m'a fallu aussi apprendre à maîtriser ce framework. J'ai pu débugger le bouton de date d'expiration. Le problème était lié à l'état du champ date d'expiration qui dépendait de la case à cocher qui était tout le temps décoché. On doit permettre à l'utilisateur de cocher la case. En fonction de la case à cocher, l'utilisateur doit pouvoir saisir ou non une date d'expiration.

Nous devons également faire en sorte que, lorsque la date est décochée, elle renvoie une valeur par défaut (par exemple 01-01-0001).

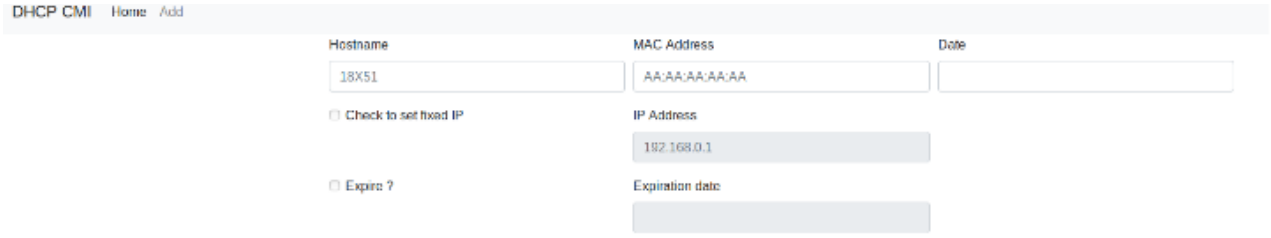


Figure 6 : Formulaire du DHCP

Il y avait une autre problématique liée au champ « description ». Lorsque l'utilisateur saisissait un caractère retour à la ligne, les précédentes étaient supprimées dans le dhcpd.conf. Je règle le problème en faisant en sorte que chaque retour à la ligne soit transformé par le caractère : « | » dans le fichier dhcpd.conf, qui sera invisible à l'œil de l'utilisateur.

Il n'y avait pas la possibilité de rechercher sur la page web. Nous avons donc mis en fonction le champ « Search » en faisant une requête AJAX (Asynchronous Javascript And XML). Cet outil permet de construire une application web dynamique et interactive. L'application finale ressemble à une longue liste d'adresse.

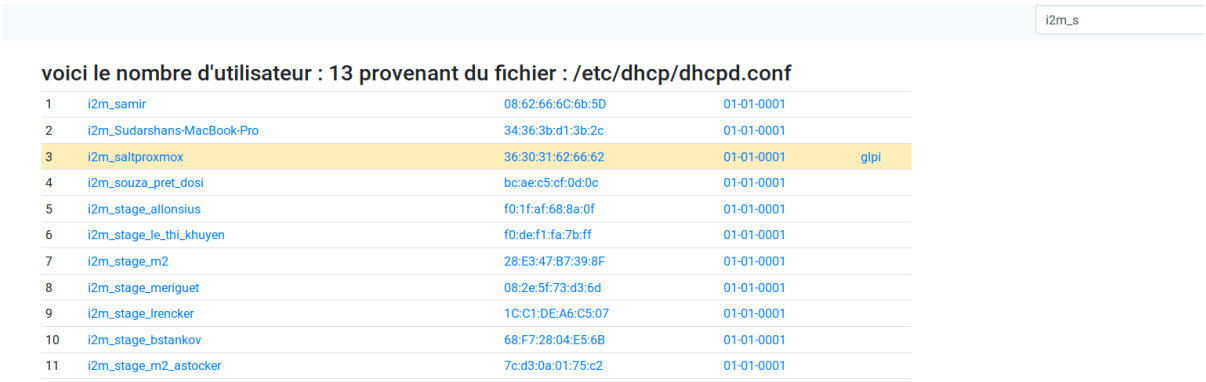


Figure 7 : Liste du DHCP

Le principal objectif de M. CHABROL en nous faisant pratiquer ces exercices était de nous initier à Symfony, afin de nous donner la prochaine mission.

C. Projet Keyring :

C'est notre seconde mission de développement, celle qui nous prendra le plus de temps.

Nous avons reçu par mail une demande d'application de gestion de clés et de badges par le pôle VIVS de Château-Gombert. Nous avons donc eu l'idée de créer une application web qui se calquerait sur le projet DHCP effectué.

Nous avons donc commencé à développer l'application avec les idées suivantes :

- Un onglet de création/gestion de clés et badges ;
- Un onglet de création de prêt en fonction des utilisateurs de l'application ;
- Un dernier onglet de création d'utilisateur.

C.1. Les outils utilisés :

Nous avons utilisé Symfony car au travers de l'application DHCP j'ai appris à l'utiliser. Nous avons aussi été incités à utiliser GitHub.

C'est un service web d'hébergement et de gestion pour développer des logiciels. Il permet à une équipe de développeurs de travailler ensemble et en même temps afin d'avancer le plus vite possible sur un projet.

C’est un outil que nous devons apprendre à utiliser durant notre stage car nous développons ensemble le projet Keyring.

De plus, GitHub permet de mettre en ligne notre code pour en faire du **versioning*** et dans le but éventuel de le partager aux autres utilisateurs de ce service, qui voudront peut-être faire une application avec la même utilité, ou au moins avec le même principe. Elle nous permet aussi de nous faire corriger par des collaborateurs s’il y a des erreurs.

Par exemple, Bootstrap est un framework **CSS*** qui permettra la simplification du développement de notre site. Il peut nous créer des boutons esthétiques le plus simplement possible.

JQUERY sert aussi simplifier le développement. Ce **software*** nous permettra, par exemple, comme nous le verrons plus tard, de nous rendre le travail de AJAX plus facile ainsi que les fonctions javascripts.

Date picker, lui, servira à nous afficher un calendrier sur l’interface graphique de notre site. Il permet également d’utiliser plusieurs formats de date et de les traduire entres elles.

C.2. Travail réalisé :

Nous allons commencer par créer la base de données suivante grâce à l’ORM doctrine. Cela nous donnera avec phpmyadmin, l’interface graphique ci-dessous :

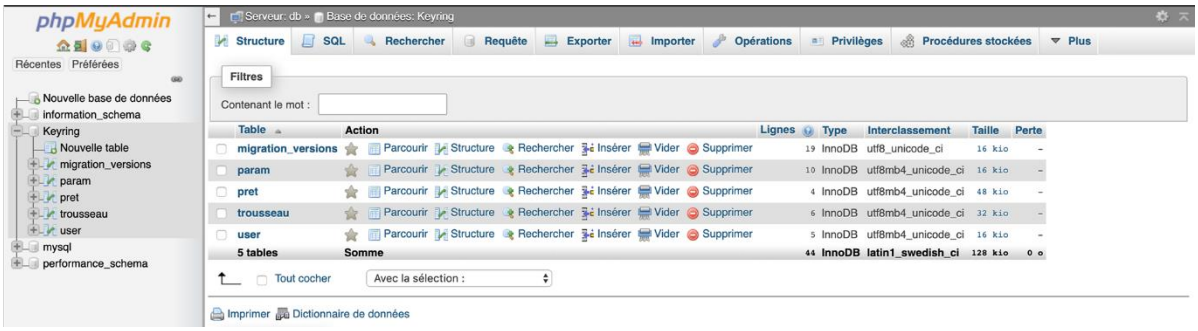


Figure 8 : Base de données sur phpmyadmin

Nous avons créé les quatre tables principales suivantes :

- Param ;
- Prêt ;
- Trousseau ;
- User.

La table param a pour but de simplifier le développement de l'application, car elle sera la liaison indirecte avec les autres tables.

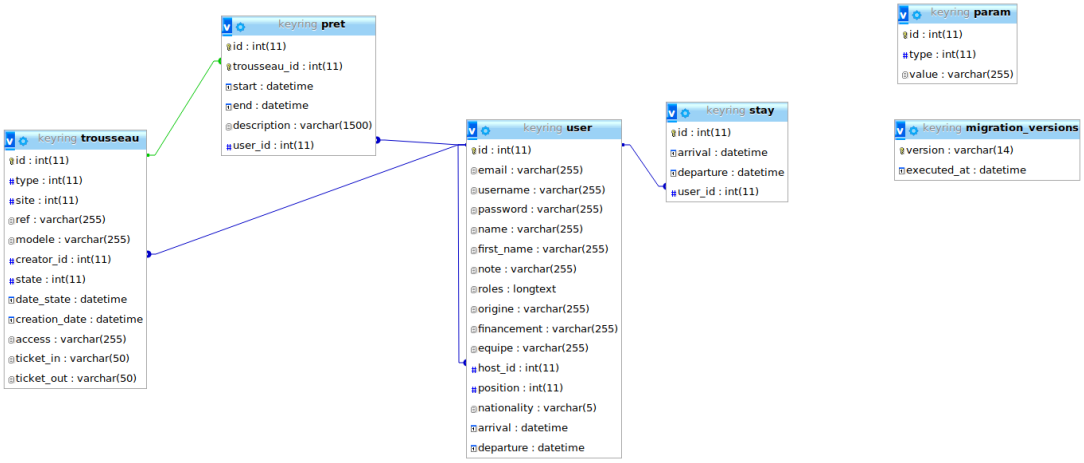


Figure 9 : Table relationnelle de notre base de données

Cela signifie que chaque élément de cette table a un identifiant unique qui est utilisé de la manière suivante : l'ID 1 représentera les sites (Château-Gombert, Luminy ou Saint-Charles) et l'ID 2 représentera les types (clés ou badges).

La table « prêt » sert à lister les différents emprunts des utilisateurs en fonction d'un site, d'un type, d'un nom, …etc.

La table des « trousseaux » est utile afin de lister les différentes clés et/ou badges avec leurs états : si elles sont volées, hors services, actives, …etc. ; elle sert aussi à créer un prêt.

Dans le but de travailler et d'avancer au mieux, j'abandonne l'idée du serveur apache2 qui m'a causé de légers problèmes auparavant, dû à sa configuration qui est assez laborieuse pour une application de ce genre.

J'opte donc pour la création d'un serveur web Symfony.

Il y a un avantage qui m'a fait me pencher sur cette méthode : c'est sa simplicité de mise en place ; il suffit de deux commandes dans notre terminal :

```
#Serveur web de test de notre application :

#Installation dépendances
composer require --dev symfony/web-server-bundle

#Lancement du serveur web de test (A la racine du projet)
php bin/console server:start
```

Figure 10 : Installation et lancement du serveur web Symfony

J'ai préféré présenter l'application sous forme de listes. Il y en aura 3 :

- Les clés/badges ;
- Les prêts ;
- Les utilisateurs.

Cette méthode permet d'agir facilement sur les éléments ci-dessus. Pour modifier ou supprimer un élément, il suffit de se rendre sur la liste de l'élément en question.

Nous avons donc rajouté des boutons permettant de supprimer et de modifier les prêts/clés/utilisateurs. Nous avons également pensé à rajouter une alerte lors du clic de la suppression, afin d'éviter de se tromper.

L'alerte demande si oui ou non nous sommes surs de vouloir supprimer l'objet en question :

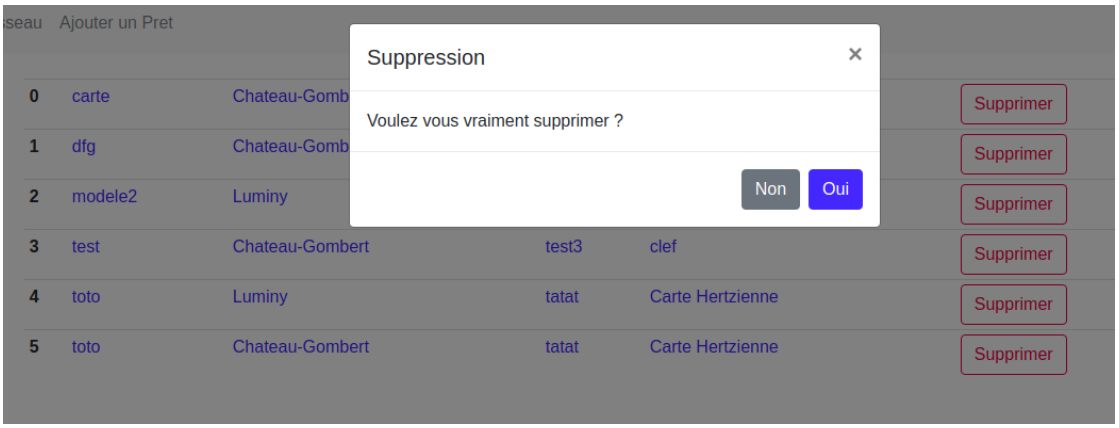


Figure 11 : Alerte lors de la suppression

Le bouton de modification, lui, sera plus compliqué à mettre en place car lorsque l'on modifie un prêt par exemple, il faut que les champs soient déjà préremplis.

Le prêt déjà existant aura donc des valeurs. Celles-ci devront déjà être saisies au moment de la modification.

Nous devons définir des rôles pour les utilisateurs afin d'avoir une sécurité sur notre application, c'est-à-dire que pas n'importe quel utilisateur puisse modifier son prêt. Il y aura donc deux rôles qui seront respectivement :

- Utilisateur ;
- Administrateur.

Les utilisateurs n'ont le droit d'accès à aucune page de l'application. Pour simplifier, les « utilisateurs » comme nous l'appelons sur notre site n'est qu'un outil permettant à l'administrateur de créer un prêt. Seuls les « administrateurs » seront les réels utilisateurs de cette application.

Afin de rendre la sécurité sur notre page plus importante, nous avons ajouté le firewall de Symfony. Il sert à gérer les droits d'accès de l'application pour bloquer les pages aux personnes qui auront un compte avec des droits restreints.

Dans l'interface graphique, nous avons ensuite rajouté les dates grâce à date picker permettant d'afficher un agenda. On aura simplement à cliquer sur un jour et cela traduira cette date au format (J/M/A). Cela permettra d'avoir une date de fin pour la visite d'un utilisateur ou pour la fin d'un prêt.

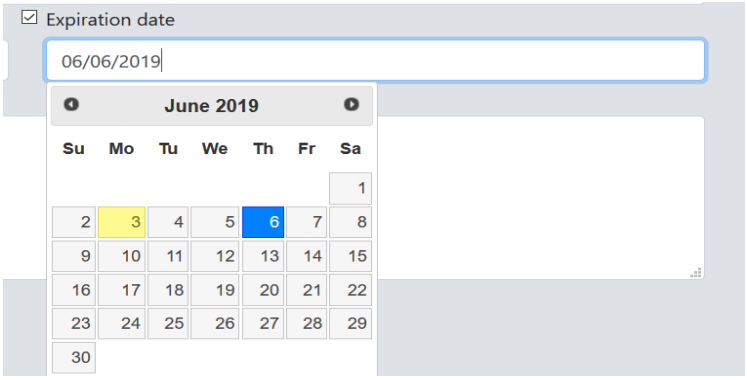


Figure 12 : Interface graphique du calendrier

Ensuite, il a fallu se pencher sur la partie de gestion et de création des utilisateurs. Pour cela, nous avons dû demander au pôle administratif ce qu'il était important d'avoir comme information pour un invité. Nous avons donc des champs qui décrivent par qui l'invité est financé, son université/lieu d'origine, s'il fait partie d'une équipe ….

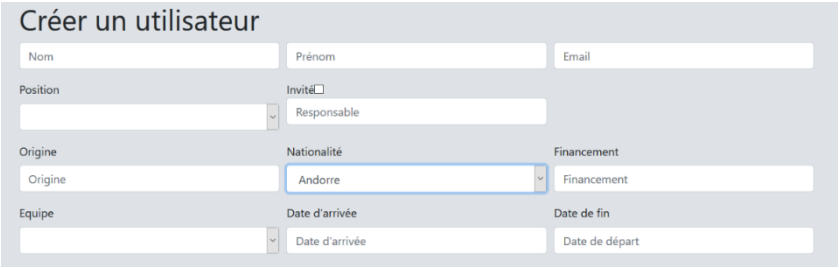


Figure 13 : Formulaire de création d'utilisateur

Une fois que nous avons créé un utilisateur, il nous faut créer une clé afin de pouvoir lui attribuer cette dernière et de créer un prêt. A la création d'une clé, nous aurons des champs correspondant à un type (clé ou badge), à un lieu, une référence, son état, …etc.

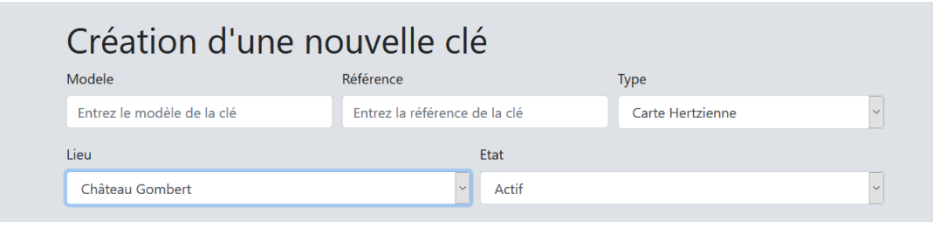


Figure 14 : Formulaire de création d'une clé

Une fois que nous avons créé une clé et un utilisateur, nous pouvons procéder à la création d'un prêt. Ce fût le système de création le plus complexe à mettre en place. Afin d'améliorer l'ergonomie de notre application, nous avons pensé à rajouter de l'auto-remplissage pour le champ « utilisateur ». C'est au moment de rentrer certains caractères que l'application nous proposera les utilisateurs contenant ces derniers. Ce fût la première difficulté car j'ai dû faire en sorte de lire en base de données les utilisateurs contenant les caractères tapés.

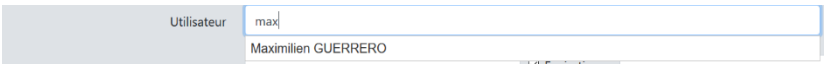


Figure 15 : Autocompletion par nom d'utilisateur

Puis, j'ai pensé à créer un filtre entre les lieux et les différents types. Lorsque l'utilisateur rentrera un site, par exemple Luminy, et un type, par exemple badge hertzien, nous avons fait en sorte que les choix possibles pour les clés soient uniquement les badges hertziens de Luminy.

Ces deux étapes seront faites, comme nous l'avons vu au-dessus, avec la technologie AJAX. Nous avons donc les champs utilisateurs, clés, dates, … afin de créer le prêt.

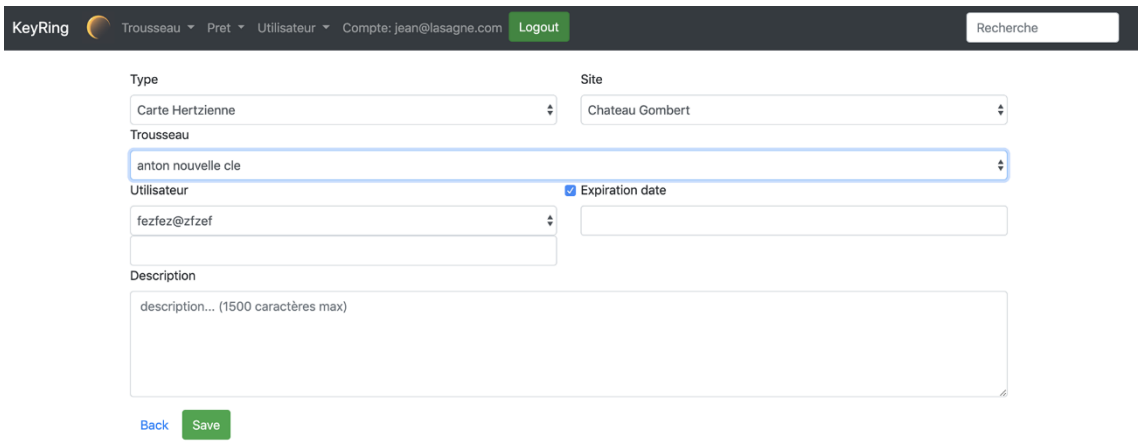


Figure 16 : Formulaire de création de prêt

Une fois ces étapes terminées, selon nous, l'application était prête à l'emploi. Nous sommes donc allés voir le pôle administratif afin d'optimiser l'application et de voir ce qui n'allait pas. Nous aurons pour mission, à la suite de cette entrevue, de rajouter un champs ticket ENT pour la création de clés, car certains badges nécessitaient la demande d'activation via un numéro de ticket ENT.

De plus, le pôle administratif nous a demandé d'ajouter la possibilité de créer un tableau Excel à partir d'une liste de notre site. Je l'ai fait sous forme de boutons où il suffit de cliquer pour le générer. Et aussi de pouvoir concevoir un document PDF à partir d'un prêt, afin de le faire signer par un emprunteur.

Nous avons rendu le site plus lisible en changeant légèrement son esthétique. Le lien du site se trouvera en annexe.

Après quelques utilisations, nous avons constaté que certains points n'étaient pas clairs aux yeux des utilisateurs du pôle administratif. Nous avons par exemple rendu possible la création d'un utilisateur directement depuis le menu « création de prêt », car le fait de devoir changer de menu pour la création d'un nouvel utilisateur n'était pas pratique.

Nous avons aussi ajouté des graphiques afin de voir le pourcentage de clés inactives/perdues/volées, … et le pourcentage de clés par rapport au site.

Cette application sera utilisée par les pôles administratifs des 3 sites de l'I2M. Certains champs n'étaient pas nécessaires pour certaines personnes tandis que d'autres les pensaient obligatoires.

Par exemple, le champ « ticket ENT » a posé un problème car à Luminy, les tickets ENT pour activer les badges ne sont pas utilisés. Il a donc fallu rendre l'application la plus ergonomique possible pour les utilisateurs des 3 pôles.

D. Aide aux utilisateurs :

Durant notre stage, nous avons eu l'occasion d'aider le personnel de l'I2M. Nous avons reçu par exemple une personne qui n'arrivait pas accéder au réseau WI-FI « Eduroam ». Lorsque nous avons pris possession de son ordinateur pour essayer de nous connecter, plusieurs problèmes se sont posés à nous ; son **OS*** était arch linux. Je n'avais jamais vu ce système d'exploitation. De plus, tout était en italien.

Nous avons donc dû communiquer avec l'utilisateur afin qu'il nous guide à travers le système pour que nous puissions répondre à sa demande.

Le problème venait du fait que la méthode d'authentification qu'il utilisait était la mauvaise ; nous avons donc dû la changer. Pour savoir laquelle utiliser, nous avons demandé à M. CHABROL, qui a su nous guider.

Nous avons aussi aidé des architectes à installer et à mettre en route un vidéo projecteur, trouver des adaptateurs pour des étudiantes qui devaient passer un oral, dépanner des problèmes d'imprimante, etc….

Certains utilisateurs nous ont posé des problématiques plus complexes. Par exemple, le programme d'un mathématicien n'arrivait pas à compiler. Il s'agissait d'un programme écrit en **LaTeX***. Après réflexion avec l'utilisateur, nous avons compris qu'il s'agissait d'un caractère qui ne devait pas être présent et qui empêchait la compilation du programme. Pour nous, ce fût une découverte puisque nous ne connaissions pas ce langage.

E. Création et mise en place d'un VPN :

La problématique est que le personnel de l'I2M n'accède pas à l'**intranet*** avec une adresse IP extérieure au réseau local.

E.1. Qu'est-ce qu'un VPN et OpenVPN ?

Le VPN, Virtual Private Network, ou réseau privé virtuel, est, de manière générale, utilisé pour se protéger des dangers sur internet. C'est une solution qui est de plus en plus répandue dans le monde de l'informatique, autant de manière professionnelle que personnelle.

Il s'agit d'un réseau privé que nous pouvons créer afin de nous protéger sur notre réseau interne. Le VPN peut se mettre en place avec différents logiciels et il va permettre de chiffrer nos données en créant un tunnel sécurisé grâce à un chiffrement **AES 256***, contrairement au proxy.

De plus, **un proxy*** ralentira la connexion car, généralement gratuit, les serveurs proxy ne peuvent pas investir beaucoup d'argent en **bande passante***. Le VPN est donc une meilleure solution en général. Ce dernier peut éventuellement servir à éviter l'interception des données par le **FAI***.

Pour nous, l'utilité principale du VPN sera de créer un tunnel permettant de répondre à la demande du personnel de l'I2M.

Nous avons choisi l'option de prendre un software gratuit et libre à mettre en place et à configurer ; il s'agit d'OpenVPN. Ce dernier crée une connexion dite **point-to-point*** ou **site-to-site*** et permet l'accès à distance facilement. Il utilise la technique de **chiffrement SSL/TLS***. Nous allons utiliser la méthode de couples noms d'utilisateur/mot de passe.

Un autre avantage qu'a ce logiciel est qu'il tourne sur une multitude d'environnements dont Mac OS, Linux et Windows, soit l'intégralité des machines de l'I2M.

E.2. Travail réalisé sur le VPN :

Pour la réalisation de ce projet, M. CHABROL nous a donné accès au serveur annu1, qui tourne sous le système d'exploitation Proxmox dans sa version 2.13.

Ce dernier est une solution de virtualisation libre basée sur **Linux KVM***. Il permet de créer des « **containers*** » et des **machines virtuels (VM)*** et de les superviser. La différence entre un conteneur et une machine virtuelle est qu'à chaque machine virtuelle, nous allons affecter un système d'exploitation, tandis qu'un conteneur, lui, partage un OS avec d'autres conteneurs. On aura donc une grosse économie de performances car un conteneur ne prendra uniquement ce dont il a besoin en ressources.

Proxmox permet, une fois installé sur un serveur ou une machine, d'avoir une interface web simple d'utilisation, utile à la gestion des containers et des VMs. Le serveur annu1 aura donc 7 containers et une machine virtuelle qui sera notre VPN et des espaces de stockages (backups,fury,local,…)

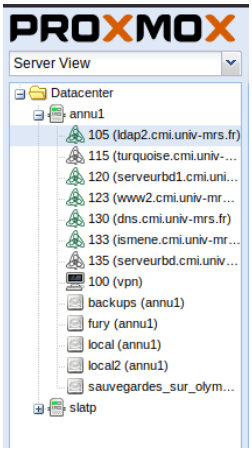


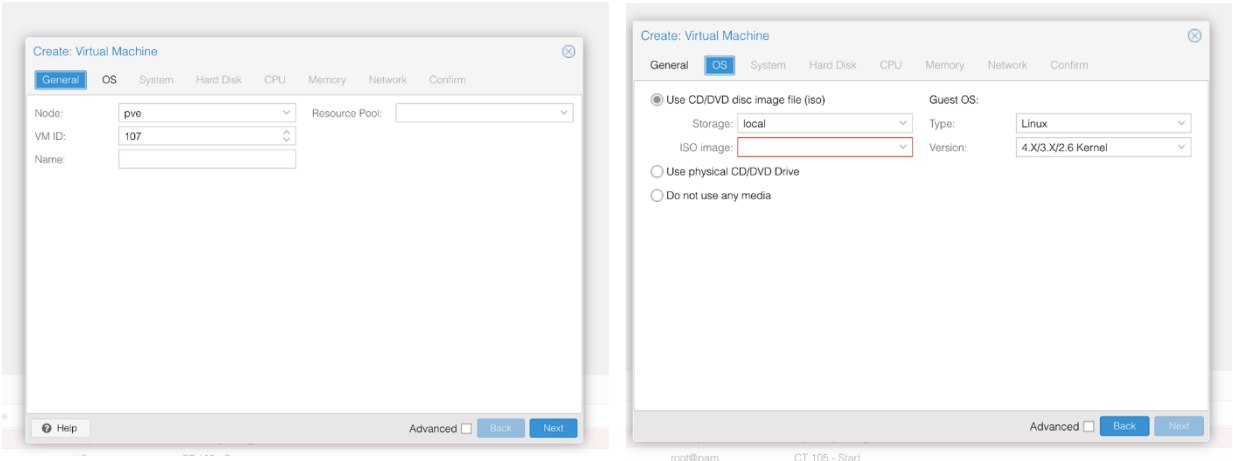
Figure 17 : Conteneur et VM de annu1

J'ai donc dû comprendre et apprendre à me servir de ce nouvel outil.

Nous avons commencé par la création de la machine virtuelle qui est finalement une étape simple, puisque tout est faisable depuis l'interface web où nous aurons un bouton « create VM ».

Figure 18 : Boutons sur l'interface web de Proxmox

Ensuite, Proxmox nous mène à une page de création de machine virutelle où on pourra rajouter une adresse IP, une **interface réseau***, un espace de stockage, etc…, soit toutes les étapes d'une configuration d'une VM classique.



Figures 19 et 20 : Principales étapes de la création d'une VM

Nous avons choisi le système d'exploitation **Ubuntu server***, car, comme sa version classique, sa documentation est très complète. Une fois la machine correctement lancée et bien configurée, nous avons pu lancer une connexion SSH sur cette dernière afin d'en faire notre VPN.

Lors de la première connexion, nous avons mis à jour la bibliothèque aptitude avec les commandes que l'on a vu dans la partie **B.2.** Nous avons aussi lancé l'installation des paquets d'OpenVPN avec la commande suivante. De plus, l'installation de ce script nous permettra la simplification du déploiement de OpenVPN :

```
curl -O https://raw.githubusercontent.com/Angristan/openvpn-
install/master/openvpn-install.sh
```

On lui donne ensuite les droits d'exécution avec la commande suivante :

```
chmod +x openvpn-install.sh
```

Et on lance le script avec celle-ci :

```
sudo ./openvpn-install.sh
```

Une fois les commandes lancées, le script nous propose de rentrer des paramètres comme l'adresse IP, le nom du VPN , etc …

Une fois le VPN mis en place et qu'il fonctionnait bien, nous avons eu l'idée de rajouter de la sécurité dans notre serveur. L'idée était de bloquer une adresse IP qui fait trop de fois d'affilée un mot de passe faux lors de la connexion sur le serveur.

Nous avons donc utilisé le software fail2ban, qui permet de faire, après configuration, exactement ce que nous demandions. Ce service fonctionne en regardant les **logs*** de OpenVPN. Or, il nous est impossible d'utiliser ce service. L'authentification avec le mot de passe se fait côté client, donc il n'y a aucune trace des tentatives de mot de passe côté serveur.

Etant donné que nous ne voulions pas laisser un service sans sécurité, nous avons réfléchi à une autre solution. Nous avons par la suite trouvé un projet GitHub d'une personne qui permettait l'authentification du client côté serveur à l'aide d'une petite commande à ajouter qui, elle, s'effectuera sur le client :

```
sudo openvpn --config nomdufichier.ovpn --auth-user-pass
```

Comme nous pouvons le voir, l'extension « --auth-user-pass » permet l'inscription des tentatives de connexion dans les logs côté serveur.

Un autre problème fut que le script n'assurait pas d'authentification. Nous avons par la suite découvert que le script n'arrivait pas à lire les logs. Il nous a fallu supprimer la ligne contenant l'utilisateur et le groupe dans le fichier server.conf. Nous avons donc réussi à faire en sorte que le fail2ban fonctionne.

Au bout de 2 faux essais, l'utilisateur sera banni pendant 12 heues. Si cet utilisateur est banni mais qu'il s'est juste trompé deux fois, il sera possible de le retirer de cette liste de bannis en nous sollicitant.

Nous aurons seulement besoin de taper la commande suivante sur le serveur afin de réaccueillir la personne :

`fail2ban-client status openvpn` : (liste les banni(e)s)

`fail2ban-client set openvpn unbanip [@ip]` : (réaccueillir l'ip voulu)

Une fois tout cela fini, nous avons donc pu mettre en service notre VPN afin que le personnel de l'I2M puisse accéder à son intranet depuis une adresse externe.

F. Mise à jour de Proxmox sur le serveur annu1 :

Comme nous l'avons vu précédemment, le système d'exploitation de annu1 est un Proxmox en version 2.13.

Nous avons proposé de mettre à jour le serveur afin d'avoir la dernière version et tous les nouveaux outils que propose Proxmox. Actuellement, sa version la plus récente est la 5.4 qui est donc celle que nous allions installer.

Avant de mettre à jour le serveur, nous avons dû faire des **backups*** pour pallier d'éventuels problèmes, afin que si rien ne fonctionne comme prévu, que nous puissions remettre en état le serveur rapidement.

M. CHABROL nous a donc prêté une tour avec un processeur assez puissant, du fait que beaucoup d'utilisateurs sollicitent le serveur.

Nous avons réalisé cette dernière étape pour que pendant la maintenance, les utilisateurs ne perdent pas leurs accès. Nous avons transféré les containers et la machine virtuelle sur cet ordinateur avec la méthode suivante :

- Installer Proxmox 5.4 sur le nouveau serveur (sur l'ordinateur qui nous a été fourni par M. CHABROL) ;
- Faire les backups sur l'ancien serveur, se connecter sur la nouvelle machine en SSH et une fois la connexion établie, faire un **SCP*** afin d'obtenir le fichier zippé du backup à l'aide de la commande suivante :
  scp tonfichier.tar root@adresseip:/var/lib/vz/dump/

Une fois le transfert correctement effectué dans le bon dossier, nous devions voir sur l'interface web du nouveau serveur l'apparition de ce que nous avions transféré.

Nous devions restaurer, sans oublier de décocher « conteneurs non privilégié ». Pour ces derniers, cette option n'était pas disponible pour la restauration de machine virtuelle, car, au préalable, nous avions défini ces droits.

Avant d'éteindre l'ancien conteneur, nous devions d'abord remettre les paramètres réseau, c'est-à-dire son adresse IP, son **gateway***, etc…

Normalement, lorsque l'on met à jour entre deux versions de Proxmox qui n'ont pas beaucoup de versions d'écart, cette étape n'est pas nécessaire.

Une fois les conteneurs et la VM mise sur le nouveau serveur, la mission était donc de mettre à jour le serveur annu1. Pour cela, nous nous sommes rendus directement dans la salle des serveurs avec une clé USB avec Proxmox stocké sous sa version 5.4.

Au début, le serveur ne détectait pas cette dernière en tant que périphérique. Après un temps de réflexion, nous avons passé dans BIOS le prochain démarrage en **UEFI***. Cette fois-ci, la clé est bien apparue et nous avons pu installer la nouvelle version de la même manière que l'on installe un Ubuntu, ou comme nous avions pu le faire auparavant sur la machine qui nous avait été fournie.

Nous avons dû refaire les étapes décrites ci-dessus pour repasser les conteneurs et la VM de la tour au vrai serveur.

Cependant, nous avons eu un problème lorsque certains conteneurs, qui démarraient à partir du point de montage de la machine, ne démarraient plus.

La nouvelle version de Proxmox n'autorise en fait pas un conteneur à démarrer ainsi. Nous avons donc fait un montage **NFS*** sur le serveur et un **bind*** entre ce dernier et notre conteneur.

Les images avant/après :

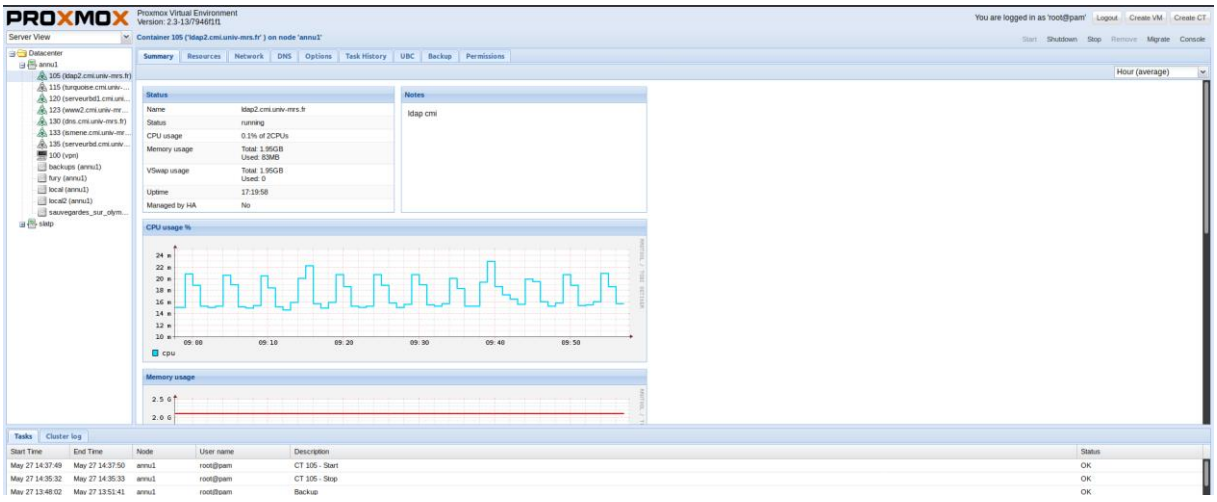


Figure 21 : Ancienne interface web de proxmox (v 2.13)

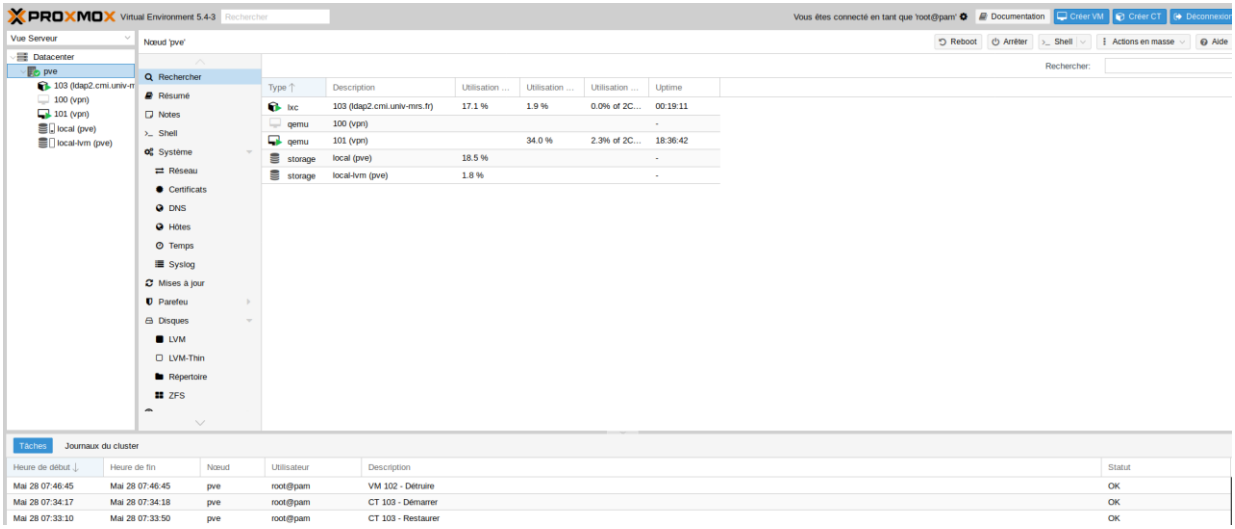


Figure 22 : Nouvelle interface web de proxmox (v 5.4)

G. Proposition de projet à l'I2M :

Hugo et moi-même avons proposé à M. CHABROL une solution autre que Proxmox pour créer et superviser les conteneurs et les machines virtuelles.

Cet outil est Docker ; il nous permet, de la même façon que Proxmox, d'avoir des conteneurs. Nous avons présenté à M. CHABROL un diaporama comparant ce dernier à Proxmox.

Docker permet donc la création de conteneurs mais aussi d'en lier plusieurs entre eux. Par exemple nous pouvons lier un conteneur **Nextcloud*** à un conteneur de base de données, dans le but de créer un serveur de stockage Cloud très simplement.

De plus, la documentation de Docker est très complète et les conteneurs se trouvent déjà préconfigurés dans une bibliothèque disponible sur le site officiel : https://www.docker.com/.

H. Réseaux Wi-Fi et imprimante :

Nous avons été sollicités pour le placement d'une borne wifi au premier étage de l'I2M. Le but était de donner l'accès **Eduroam*** à un couloir où les bureaux étaient mal desservis en réseaux Wi-Fi.

Pour cela, nous avons fixé la borne au plafond et nous avons dû raccorder le câble à un connecteur mural, de manière à ce que ce dernier ne soit pas visible et ne gêne pas le passage. Nous l'avons donc fait passer dans le faux plafond et par une gaine de câble électrique, déjà présente.

Figure 23 et 24 : Installation de la borne Wi-Fi

Un peu plus tard durant le stage, nous avons dû déplacer une imprimante, beaucoup plus récente que toutes les autres, du premier étage afin que toutes les personnes de cet étage puissent en bénéficier. Nous avons donc dû la raccorder directement au local technique sur un switch pour que les utilisateurs puissent imprimer en local.

Figure 25 : Passage du câble dans le faux plafond

III. Conclusion :

Durant mon stage, j'ai principalement travaillé sur le développement web mais j'ai aussi eu des missions relevant de l'administration système.

Les tâches orientées vers l'administration des systèmes informatiques m'ont permis non seulement de consolider mes connaissances sur le sujet mais aussi d'acquérir de nouveaux savoirs dans ce domaine. De plus, le développement web n'étant pas ma spécialité, cela m'a donné l'opportunité d'élargir mon spectre de connaissances et compétences dans ce domaine.

Le contact avec les utilisateurs m'a aussi permis d'apprendre à être à l'écoute et à la disposition des personnes, que ce soit pour des questions ou des doutes.

Mes différentes missions m'ont permis d'approfondir mes compétences et mes connaissances dans le domaine de l'Administration informatique en général. Grâce à ce stage, j'ai non seulement découvert le monde de l'entreprise dans un service informatique mais aussi toutes les responsabilités qu'impliquent le fait de travailler dans ce dernier.

Cette expérience de dix semaines a confirmé ma volonté de travailler dans l'Administration des Systèmes et des Réseaux. Qui plus est, mon cursus en IUT me permet d'avoir une base solide dans ce domaine, ce qui, par conséquent, pourrait me permettre de devenir un alternant en Licence Professionnelle à la fois motivé et qualifié.

IV. Glossaire :

Commutateur (switch) : c'est un équipement qui relie plusieurs segments dans un réseau.

Dokuwiki : Moteur de wiki libre. Permet d'avoir un historique des travaux effectués.

VLAN : Réseau local virtuel permettant d'améliorer la gestion du réseau.

SSH : Secure Shell est un programme informatique et un protocole de communication sécurisé

OpenSSH : ensemble d'outils informatiques permettant des communications sécurisées.

Clé de sécurité : Majoritairement utilisée pour la connexion sans fil à des terminaux.

Putty : Emulateur de terminal, client pour les protocoles SSH, Telnet, TCP ….

Téléphone IP : Téléphone utilisant la technologie VoIP (voix sur IP).

SQL : Langage informatique servant à exploiter les bases de données relationnelles.

Template : Modèle de conception de logiciel.

Pool DHCP : Plage d'adresse permettant l'accès au DHCP* (Dynamic Host Configuration Protocol).

DHCP : Protocole réseau qui assure la configuration automatique réseau d'un hôte.

Dual Boot : Possibilité de démarrer un ordinateur sous deux OS différents.

Ubuntu : Système d'exploitation open source basé sur Debian.

Flasher : Possibilité de démarrer un ordinateur sur une clé USB.

BIOS : Basic Input Output Sytem, ensemble de fonctions contenues dans la mémoire morte de la carte mère permettant d'effectuer des opérations de base.

Partition de disque : Consiste à créer des « zones » sur le disque dont les données ne seront pas mélangées.

Service : Programme lancé par l'utilisateur ou l'ordinateur.

Bibliothèque aptitude : Là où l'on stock tous les services vérifiés par Debian.

Date picker : Interface graphique permettant à l'utilisateur de sélectionner une date dans un calendrier.

CSS : Cascading Style Sheets est un langage informatique permettant la présentation des documents HTML.

Software : Est la traduction en anglais pour le mot « logiciel ».

OS : Operating System, ensemble de programmes dirigeant l'utilisation des ressources d'un ordinateur.

LaTeX : Langage et système de composition de documents.

AES 256 : Advances Encryption Standard, est un algorithme de chiffrement, 256 est la taille en bits de la clé de chiffrement.

Proxy : Logiciel informatique qui joue le rôle d'intermédiaire entre deux hôtes afin de faciliter/surveiller leurs échanges.

Bande passante : Débit maximal d'une voie de transmission.

FAI : C'est un Fournisseur d'Accès Internet.

Intranet : Réseau informatique utilisé à l'intérieur d'une entreprise.

Point-to-point : Type de réseau qui permet la connexion d'une paire de terminaux.

Site-to-site : Communication entre 2 LAN distants.

Chiffrement SSL/TLS : Se sont deux techniques de chiffrements différentes, la plus sécurisée étant TLS car approuvées par l'IETF.

Linux KVM : Permet la virtualisation matérielle ce qui accélère la virtualisation des OS.

Containers : En Français, conteneurs, permet de stocker des « objets », par exemple un serveur apache2.

Machine virtuelle : Appareil informatique créé par un logiciel d'émulation.

Interface réseau : Généralement, c'est la carte réseau de l'ordinateur.

Ubuntu server : Version serveur de l'OS de Ubuntu, elle est donc sans interface graphique.

Logs : Historique d'évènement stocké dans un fichier texte.

Backups : Sauvegarde de données, qui peuvent être ponctuelles.

SCP : Secure Copy. Transfert via SSH des fichiers et des répertoires entre des machines.

Gateway : En français, passerelle par défaut, système matériel permettant de faire la liaison entre deux réseaux.

UEFI : C'est une interface qui succède, sur certaines cartes mères, au BIOS.

Nextcloud : Logiciel libre d'hébergement de fichier.

Eduroam : Vise à offrir un accès sans fil à internet aux personnels et étudiants des établissements d'enseignement supérieur.

NFS : Permet à un ordinateur d'accéder, via un réseau, à des fichiers distants.

Bind : Action permettant de lier des éléments entre eux.

Versioning : Garde une trace de l'évolution d'un programme

V. Bibliographie :

https://doc.ubuntu-fr.org/

https://doc.ubuntu-fr.org/openvpn

https://www.fail2ban.org/wiki/index.php/Main_Page

https://doc.ubuntu-fr.org/fail2ban

https://getbootstrap.com/docs/4.3/getting-started/introduction/

http://support.i2m.univ-amu.fr/doku.php

https://www.globalsign.fr/fr/blog/difference-entre-ssl-et-tls/

https://github.com/Angristan/OpenVPN-install

https://github.com/olivierChabrol

Annexe :

Journal de bord :

Semaine 1 :

- Découverte des locaux ;
- Suppression des vlans inutilisé ;
- Apprentissage de Symfony ;
- Apprentissage du code de nos prédécesseurs.

Semaine 2 :

- Debug du champ de date ;
- Debug du champ description ;
- Mise en fonction du champ « rechercher » ;
- Lecture et compréhension de la demande du pôle administratif.

Semaines 3 à 6 :

- Développement de l'application Keyring ;
- Apprentissage de GitHub ;
- Création des bases de données avec Doctrine/phpmyadmin ;
- Discussion à l'optimisation de notre application ;
- Optimisation de notre application.

Semaines 6 à 8 :

- Mise en place d'un serveur VPN avec openVPN ;
- Mise en place de fail2ban ;
- Backups du serveur annu1 ;
- Mise à jour de la version de Proxmox sous annu1.

Semaines 8 à 10 :

- Rédaction de notre rapport ;
- Préparation à l'oral ;
- Finition de la mise à jour de annu1.

Réparation de la date d'expiration par défaut pour le service DHCP :

```
if(!$("#expirationDate").is(":checked")) {
      $('#expirationDate').prop('disabled', true);
    }

 $('#expirationDateCheck').on('change', function(event) {
      if ($('#expirationDateCheck').is(':checked')) {
        $('#expirationDate').prop('disabled', false);
    }

 if($('#expirationDate').val() != "01-01-0001"){
        $("#expirationDateCheck").prop('checked', true);
    }
```

On formule donc un code html/Javascript pour faire fonctionner notre requête Ajax en fonction de notre recherche **d'hôte/Adresse Mac ou de Date de création**

```
function searchHost()
{
    $.ajax({
      url:"/listHost",
  method: "post",
  data: {host: $("#searchHost").val()},
  dataType: 'json',
  success: function (data) {
     $("#tablehost").find("tr").remove();
     $js = JSON.parse(data);
     $("#nbHost").text($js["dataLenght"]);
    var numLine = 0;
    $.each($js["data"], function(index, element) {
      var classCss = "";
      numLine = numLine + 1;
      if (element.fixedIpAddress != "")
      {
        classCss = 'class="table-warning"';
      }
      $('#tablehost').append('<tr '+classCss+'> <td id="ligne">' + numLine + '</td>

    });
  }
    });
}
```

Notre tableau sera donc interactif en fonction de notre recherche, les informations relatives aux hôtes sont parsés dans un document au format JSON puis affichés en HTML.

```
public function jsonSerialize() {
  return [
      'hostname' => $this->_hostname,
      'dateModif' => ($this->_dateModif == null ?"01-01-0001":$this->_dateModif->format('d-m-Y')),
      'macAddress' => $this->_macAddress,
      'fixedIpAddress' => ($this->_fixedIpAddress == null ?"":$this->_fixedIpAddress)

  ];
```

Installation d'un fail2ban pour éviter les attaques par force brute. Nous allons installer le paquet suivant :

```
apt install fail2ban
```

Nous allons ensuite créer dans /etc/fail2ban/filter.d le fichier openvpn.conf

```
touch openvpn.conf
```

Nous allons ensuite éditer ce fichier :

```
nano openvpn.local
```

Et y ajouter le code suivant :

(Ligne pouvant varier selon l'applicatif, difficulté de l'écriture de l'expression régulière : failregex = .*<HOST>:\d+ SENT CONTROL \[\S+\]: 'AUTH_FAILED'.*

```
 # Fail2Ban filter for selected OpenVPN rejections
#
[Definition]
#
# Example message (other matched messages not seen in the testing server's logs):
# Fri Sep 23 11:55:36 2016 TLS Error: incoming packet authentication failed from [AF_INET]59.90.146.160:51223

failregex =.*<HOST>:\d+ SENT CONTROL \[\S+\]: 'AUTH_FAILED'.*

ignoreregex =
```

Ensuite nous irons dans le dossier /etc/fail2ban/jail.d

```
cd /etc/fail2ban/jail.d
```

et crée le fichier openvpn :

```
touch openvpn
```

puis y ajouter le code suivant :

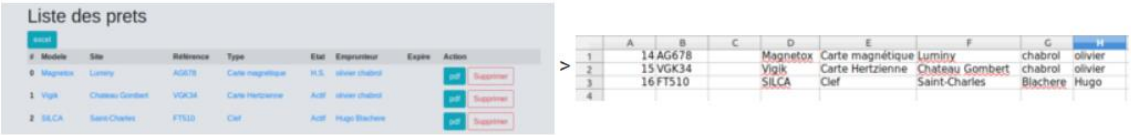
```
# Fail2Ban configuration fragment for OpenVPN

[openvpn]
enabled  = true
port     = 1194
protocol = udp
filter   = openvpn
logpath  = /var/log/openvpnas.log
maxretry = 2
```

Puis on redémarre Open-VPN :

```
systemctl restart openvpn
```

Rajout d'un bouton qui va produire un document excel par rapport a la liste des prêts.



Notre application web est disponible en ligne avec le lien suivant (vous n'aurez pas de compte vous ne pourrez donc pas vous authentifier) : http://keyring.i2m.univ-amu.fr/login

Méthode d'authentification d'eduroam : PEAP

Configuration de l’hôte virtuel :

```
<virtualHost *:80>
  ServerName dhcpcmi.i2m.univ-amu.fr
  DocumentRoot /var/www/dhcp/public
  <Directory "/var/www/dhcp/public/">
      AuthType Basic
      AuthName "Restricted Content"
      AuthUserFile /etc/apache2/.htpasswd
      Require valid-user
      AllowOverride None
      Order Allow,Deny
      Allow from All
      <IfModule mod_rewrite.c>
          Options -MultiViews
          RewriteEngine On
          RewriteCond %{REQUEST_FILENAME} !-f
          RewriteRule ^(.*)$ index.php [QSA,L]
      </IfModule>
  </Directory>
  ErrorLog /var/log/apache2/dhcp_error.log
  CustomLog /var/log/apache2/dhcp_access.log combined
  <Directory /var/www/dhcp/public/bundles>
      <IfModule mod_rewrite.c>
          RewriteEngine Off
      </IfModule>
  </Directory>
  # optionally set the value of the environment variables used in the application
  #SetEnv APP_ENV prod
  #SetEnv APP_SECRET <app-secret-id>
  #SetEnv DATABASE_URL "mysql://db_user:db_pass@host:3306/db_name"
</VirtualHost>
```