



# ÉCOLE CENTRALE LYON

UE PRO  
LABORATOIRE AMPÈRE

---

## Rapport du projet d'étude n°57

---

### *Élèves*

Armand DIDIERJEAN

Tristan CHEVREAU

Nathanaël LASCOUX

Anne AKA

Alexandre KOUADIO

Victor ANGOT

### *Commanditaires*

Antoine HAYNEZ

Benjamin DENISE

### *Tuteurs*

Daniel MULLER

René CHALON

### *Chargé de gestion de projet*

Thierry HOC

### *Chargé de communication*

Nicolas HOURCADE

### *Président de jury*

Frédéric DUBREUIL

## 1 Résumé

Ce projet vise à documenter et réhabiliter la salle des serveurs d'ÉCLAIR, la section informatique de l'association AEECL. Cette réhabilitation était nécessaire, car la salle des serveurs devenait obsolète et inutilisable par les membres de l'association. Le projet a été divisé en trois étapes : documenter la salle des serveurs, détecter ses problèmes, rechercher des solutions appropriées et enfin les mettre en place. Le présent rapport présente une partie de la documentation produite, des explications sur le processus de création de cette dernière et la motivation des choix effectués pour améliorer l'infrastructure (notamment le choix des technologies utilisées et leur configuration).

## 2 Abstract

This project aims to document, rehabilitate and retrofit ECLAIR's own server room. ECLAIR is the IT section of the AEECL association. This rehabilitate was desperately needed, as the server room was getting obsolete as well as becoming a blackbox to the members of the association. The project was divided into three steps : documenting the server room and its problems, searching for appropriate solutions and finally setting those up. The work done include but is not limited to : network infrastructure (firewall, DNS, reverse proxy...) and the production server. This paper includes most of the produced documentation (from the initial situation of the servers but also the work done on them) as well as explanations on the documenting process.

### 3 Remerciements

Nous adressons nos remerciements aux personnes qui nous ont aidés dans la réalisation de ce projet d'étude.

Nous remercions d'abord René Chalon et Daniel Muller, professeurs de l'école Centrale Lyon. En tant que tuteurs, ils nous ont guidés dans notre travail et ont mis en évidence les questions que nous devions nous poser. Nous tenons aussi à les remercier pour avoir accepté d'encadrer ce projet, qui au delà de la dimension scolaire, nous a stimulé tout le long d'année. Ce fut une vraie opportunité de travailler sur un projet aussi épanouissant qui n'aurait pas été possible sans eux.

Nous tenons aussi à remercier Thierry Hoc, professeur de l'ECL et conseiller en gestion de projet, pour ses précieux conseils sur la tenue d'un projet et tout ce qui s'y rapporte.

Nous remercions aussi Nicolas Hourcade, professeur de l'ECL et conseiller en communication, pour ses remarques et conseils sur la façon de présenter un projet à l'oral et sa bonne humeur permanente.

Nous souhaitons particulièrement remercier les membres actuels ainsi que les anciens membres d'ÉCLAIR pour leurs conseils et leur soutien tout au long du projet ; et plus spécifiquement Benjamin Denise (E2020), Yohan Beugin (E2017), Lucas Bourthole (E2017), François Homps (E2017) ainsi que Alexandre Marsone (E2016) pour leur expérience impressionnante, leur bienveillance à notre égard, leurs conseils, mais aussi pour leur réactivité.

## Table des matières

<b>1 Résumé</b>	<b>1</b>
<b>2 Abstract</b>	<b>1</b>
<b>3 Remerciements</b>	<b>2</b>
<b>4 Glossaire</b>	<b>6</b>
<b>5 Introduction</b>	<b>8</b>
5.1 Avant-propos . . . . .	8
5.2 Contexte . . . . .	8
5.3 Problématique . . . . .	8
5.4 Plan du rapport . . . . .	8
<b>6 Présentation du projet</b>	<b>10</b>
6.1 Identification des besoins de l'association . . . . .	10
6.2 Pilotage . . . . .	10
<b>7 Théorie des réseaux informatiques</b>	<b>12</b>
7.1 Prérequis théoriques . . . . .	12
7.2 LAN et VLAN . . . . .	13
7.3 RAID . . . . .	13
7.4 Reverse-proxy . . . . .	15
7.5 Pare-feu . . . . .	15
7.6 DNS . . . . .	16
7.7 SSH . . . . .	16
7.8 DHCP . . . . .	17
7.9 NTP . . . . .	17
7.10 Serveur web . . . . .	17
7.11 Virtualisation et conteneurisation . . . . .	18
7.11.1 Hyperviseur . . . . .	18
7.11.2 Docker . . . . .	18
<b>8 Rétro ingénierie</b>	<b>21</b>
8.1 Inventaire . . . . .	21
8.1.1 Physique . . . . .	21
8.1.2 Logiciel . . . . .	22
8.2 Le réseau physique du M16 et de la salle serveur . . . . .	23
8.2.1 Conventions de notation . . . . .	23
8.2.2 Cartographie du réseau . . . . .	26
8.3 Présentation de l'architecture réseau d'ÉCLAIR . . . . .	27
8.3.1 Communication avec les switchs . . . . .	28
8.3.2 Les VLAN utilisées . . . . .	30
8.4 Installation de la salle serveur . . . . .	32
<b>9 Réalisations</b>	<b>33</b>
9.1 Mise à niveau du hardware de l'infrastructure réseau . . . . .	33

9.1.1	Configuration des onduleurs . . . . .	33
9.1.2	Dimensionnement des serveurs . . . . .	33
9.1.3	Choix du serveur . . . . .	34
9.2	Mise à niveau logiciel de la salle serveur . . . . .	35
9.2.1	Mise à jour des systèmes d'exploitation . . . . .	35
9.2.2	Interface réseau . . . . .	35
9.2.3	Modification du reverse-proxy . . . . .	36
9.2.4	Mise à jour du pare-feu . . . . .	37
9.2.5	Migration vers un nouvel hyperviseur . . . . .	39
9.2.6	Mise en place de l'architecture SSH . . . . .	40
9.2.7	Migration du DNS, NTP et DHCP . . . . .	41
9.2.8	Système d'authentification unique (SSO) . . . . .	41
9.2.9	L'outil de passation de connaissance : le wiki . . . . .	44
9.3	Responsabilité sociétale et environnementale . . . . .	45
9.3.1	Impact environnemental . . . . .	45
9.3.2	Règlement général sur la protection des données . . . . .	46
<b>10</b>	<b>Prolongement du projet</b>	<b>47</b>
10.1	Stratégie de sauvegarde des données . . . . .	47
10.1.1	Données dites légères (règle de sauvegarde 3-2-1) . . . . .	47
10.1.2	Données dites lourdes . . . . .	48
10.2	Stratégie de restauration des données . . . . .	48
10.3	Monitoring . . . . .	48
<b>11</b>	<b>Conclusion</b>	<b>50</b>
<b>12</b>	<b>Bibliographie</b>	<b>51</b>
<b>13</b>	<b>Annexes</b>	<b>53</b>
13.1	Annexes théoriques . . . . .	53
13.2	Annexes spécifiques à notre installation . . . . .	81

## Table des figures

1	Grandes étapes du projet . . . . .	10
2	Tableau récapitulatif des protocoles les plus connus . . . . .	12
3	Comparaison des différents RAID (d'après IONOS) . . . . .	14
4	vue schématique d'un pare-feu . . . . .	16
5	Fonctionnement (très) simplifié du DNS . . . . .	16
6	Comparaison machine virtuelle et docker (d'après NetApp.com) . . . . .	19
7	Extrait de l'inventaire . . . . .	22
8	Câbles avant rangement et réorganisation . . . . .	23
9	Vue de la baie de brassage . . . . .	24
10	Vue des distributeurs . . . . .	25
11	Plan du M16 - étage associatif . . . . .	25
12	Testeur RJ45 (d'après Conrad) . . . . .	26
13	Routage des paquets . . . . .	27
14	Structure du réseau d'ÉCLAIR . . . . .	28
15	Câble ethernet-série (d'après Ebay) . . . . .	28
16	Câble USB-série (d'après Startech) . . . . .	29
17	Affichage et paramétrage de l'assignement des VLAN aux ports . . . . .	29
18	Affichage des informations système d'un des switches . . . . .	30
19	Baie de brassage . . . . .	32
20	Extrait du cahier des charges . . . . .	34
21	Exemple de configuration réseau (serveur de production) . . . . .	36
22	Fonctionnement du pare-feu d'ÉCLAIR . . . . .	38
23	Hyperviseur type 1 (d'après IT-Connect) . . . . .	39
24	Hyperviseur type 2 (d'après IT-Connect) . . . . .	39
25	Architecture SSH de l'association ÉCLAIR . . . . .	40
26	Diagramme séquentiel d'une authentification utilisant le protocole OAuth 2.0 . . . . .	43
27	Logo de docusaurus . . . . .	44
28	Bilan Carbone de la salle des serveurs d'ÉCLAIR . . . . .	45

## 4 Glossaire

Dans un souci de simplification, nous utiliserons les désignations suivantes :

- VM désignera les machines virtuelles.
- Association désignera une section de l'USEECL, de l'AEECL, un club ou une association indépendante.
- Machine, ou équipement réseau : tout appareil électronique, contribuant au fonctionnement d'un réseau. Nous utiliserons dans notre projet les équipements suivants : serveurs, ordinateurs, émetteurs Wi-Fi, switches, distributeurs

Ensuite, les termes techniques utilisés dans ce document sont définis ici :

- **Baie de brassage** : Armoire rassemblant les câbles entrant et sortant d'un point du réseau. Elle contient au même endroit les switches, distributeurs et tout autre machine ou dispositif gérant les données qui transitent par elle ; ceux-ci sont montés sur un "rack".
- **Bond** : L'agrégation de lien (ou bonding) permet de regrouper plusieurs interfaces physiques sous une même interface virtuelle (bond), afin de permettre la tolérance de panne et/ou l'augmentation du débit.
- **Cloud native** : Une application cloud-native se compose de services plus petits, indépendants et faiblement couplés.
- **Distributeur** : Support pour prises RJ45 ou fibre optique. Le distributeur permet de regrouper tous les câbles Ethernet ou fibre qui partent d'une baie de brassage (souvent par le plafond) vers des prises situées dans d'autres salles ou bâtiments.
- **DNS (Domain Name System)** : protocole permettant de faire correspondre IP et nom de domaine.
- **DNS wildcare** : enregistrement DNS générique, cet enregistrement permet d'appliquer des règles sur un ensemble de noms de domaine. Par exemple \*.eclair.ec-lyon.fr désigne tous les noms de domaine qui finissent par eclair.ec-lyon.fr.
- **(Adresse) IP (Internet Protocol)** : identifiant d'une machine dans un réseau.
- **Gateway** : nom générique d'un dispositif permettant de relier deux réseaux informatique de types différents (concrètement on indique l'adresse IP du serveur à qui on s'adresse pour communiquer avec l'autre réseau), par exemple un réseau local et le réseau de la DSI dans notre cas. On peut noter la présence du pare-feu entre le gateway des deux réseaux.
- **Daemon** : c'est un processus fonctionnant généralement en arrière-plan et qui est chargé d'une mission spécifique. Sous *Windows*, ils sont appelés "services".
- **Git** : outil de versionnement.
- **LAN (Local Area Network)** : regroupement de machines physiquement connectées entre elles et pouvant communiquer au sein d'un réseau local, sans passer par le réseau internet mondial.
- **Logiciel libre** : désigne une philosophie de conception et utilisation logicielle, ainsi que le logiciel en question. Le principe est que la modification, l'utilisation et la duplication du logiciel est permise, dans l'objectif de garantir des libertés aux utilisateurs [9].
- **Load Balancing** : processus de répartition d'un ensemble de tâches sur un ensemble de ressources, dans le but d'en rendre le traitement global plus efficace.

- **NAS** : serveur de stockage en réseau.
- **NTP** : Network Time Protocol (« protocole de temps réseau ») ou NTP est un protocole qui permet de synchroniser, via un réseau informatique, l'horloge locale d'ordinateurs sur une référence d'heure. Par abus de langage on désignera l'hôte d'un serveur NTP par "le NTP".
- **PAM (Pluggable Authentication Module)** : ensemble de bibliothèques Linux permettant de gérer les authentifications à travers tout le système.
- **Platform as a Service (PaaS)** : séparation des rôles où l'entité cliente (par exemple une association) maintient les applications proprement dites ; le fournisseur maintient la plate-forme d'exécution de ces applications : le matériel du ou des serveurs (la carte mère, la mémoire vive...), les logiciels de base (c'est-à-dire le ou les systèmes d'exploitation, le ou les moteurs de bases de données...) et l'infrastructure (de connexion au réseau, de stockage, de sauvegarde).
- **Ports** : identifiant d'une socket.
- **Protocole** : règle de communication adopté pour un échange entre deux machines (appelées client et serveur). Voici une liste des principaux protocoles informatiques :
  - SSL/TLS : protocole sécurisé d'échange de paquets.
  - HTTP : protocole de communication client-serveur (port 80)
  - HTTPS : protocole de communication client-serveur sécurisé par TLS (port 443)
  - SSH (Secure SHell) : protocole sécurisé de connexion à distance (port 22).
  - LDAP (Lightweight Directory Access Protocol) : protocole permettant de communiquer avec des annuaires.
- Réseau informatique : ensemble d'équipements reliés entre eux pour échanger des informations.
- **Socket** : interface de connexion entre deux machines.
- **Switch (ou commutateur)** : Abréviaison communément utilisée d'un "switch Ethernet". Machine équipée de ports RJ45, permettant de connecter entre elles plusieurs équipements au moyen de câbles Ethernet ou de fibres optiques et ainsi de diviser ou rassembler des flux de données. Ces machines sont équipées d'outils divers, tels que des compteurs de flux. Elles permettent aussi d'attribuer des VLAN à certains de leurs ports et ainsi d'imposer une certaine structure au réseau. *le pluriel "switches" sera employé, même si celui-ci ne fait pas l'unanimité.*
- **Trunk** : Un trunk est un lien entre deux équipements, le plus souvent entre deux switch, configuré de telle sorte que l'on peut y faire circuler des trames ethernet modifiées comportant des informations relatives au VLAN sur lequel elles transitent (d'après Cisco)
- **VLAN (Virtual Local Area Network)** : regroupement logiciel de machines qui ne sont pas forcément physiquement connectées entre-elles. Les machines se comportent alors comme si elles étaient sur un LAN.
- **VLAN Tagging** : permet d'identifier à quel VLAN est destiné un paquet.

## 5 Introduction

### 5.1 Avant-propos

Il est important de prendre en compte une singularité de ce projet d'étude. En effet, il a été commandité par l'association ÉCLAIR dont le mandat est renouvelé annuellement. Ainsi, en février, trois membres du projet ont intégré l'association et ont été nommés respectivement président, secrétaire général et trésorier (ce qui forme ce que l'on appelle communément le bureau d'une association : c'est l'organisme interne le plus à même de prendre des décisions importantes relatives à son organisation). Au même moment les commanditaires ont mis fin à leur mandat. Cette prise de fonction a permis d'accélérer ou de faciliter de nombreux processus (comme le dimensionnement du nouveau serveur ou les questions administratives) et a aussi été un facteur de motivation important. En contrepartie, il a fallu travailler sur la cohésion de l'équipe pour ne pas délaisser les autres membres du projet étant donné que les membres d'ÉCLAIR investissaient naturellement plus de temps dans le projet.

### 5.2 Contexte

ÉCLAIR (École Centrale de Lyon, Amis de l'Informatique et des Réseaux) est une section de l'AEECL (Association des Élèves de l'École Centrale de Lyon). ÉCLAIR prend en charge tout ce qui est lié de près ou de loin aux besoins informatiques associatifs de l'école. Pour ce faire, l'association dispose d'une salle des serveurs située au bâtiment M16.

La salle serveur d'ÉCLAIR a vieilli et est devenue incertaine et défaillante, il est donc nécessaire de la rénover pour subvenir aux besoins associatifs de l'école. En outre, le COVID-19 ayant profondément meurtri l'associatif Centralien, ÉCLAIR a peu à peu perdu le contrôle de son installation et de ses serveurs qui semblent maintenant être une boîte noire. Ce projet d'étude commandité par ÉCLAIR consiste donc en un audit de leur infrastructure, puis en la mise en place de mesures correctives, le tout en produisant une documentation destinée aux années suivantes. En d'autre termes, il s'agit d'appréhender l'infrastructure existante ainsi que ses problèmes puis de rechercher et proposer des solutions pour si possible les appliquer, ceci afin que ÉCLAIR puisse réutiliser et exploiter sereinement leurs serveurs pour développer de nouveaux projets et reprendre d'anciens services. Cette étude se fera au regard de considérations environnementales et des politiques de confidentialité et de sécurité en vigueur en Europe.

### 5.3 Problématique

Le problème à résoudre réside dans la question de la documentation de l'infrastructure ainsi que dans l'amélioration de cette installation en matière de *sécurité*, de *performance* et d'*impact environnemental*. En d'autre termes, on cherchera à proposer des améliorations de l'infrastructure réseau actuelle dans le but de la rendre plus facilement utilisable et sécurisée.

### 5.4 Plan du rapport

La structure adoptée dans ce rapport est la suivante :

- Il s'agit d'abord de présenter le projet dans sa globalité ainsi que les méthodes utilisées pour gérer celui-ci dans la section 6.
- Dans un second temps, nous présenterons des bases de la théorie des réseaux informatiques dans la section 7, ce qui sera utile pour la suite du rapport.
- La troisième partie (section 8) présente la première phase du PE à savoir la phase de documentation et de découverte de l'infrastructure existante.
- La quatrième partie (section 9) présente la deuxième phase du PE à savoir la phase d'amélioration et d'appropriation de l'infrastructure. En d'autres termes, nous expliquerons la conception puis la mise en place de solutions à des problèmes mis en lumière dans la section 8.
- La dernière partie du projet (section 10) ouvre sur les problèmes que nous n'avons pas pu traiter, de pistes d'amélioration et de solutions pas encore appliquées. Cette partie s'adresse particulièrement aux personnes qui seront impliqués l'année prochaine dans ce Projet d'Études, qui sera amené à être reconduit.

## 6 Présentation du projet

### 6.1 Identification des besoins de l'association

La conception d'un système d'informatique passe d'abord par l'étude et la compréhension du besoin de ses utilisateurs : il est évidemment inutile de construire un *data center* pour stocker les photos de vacances d'une famille.

La question de la raison d'être de l'infrastructure réseau d'ÉCLAIR s'est donc assez vite posée. Historiquement, ses fonctions étaient :

- L'administration du réseau internet de la résidence Paul Émile Comparat.
- L'hébergement d'un système d'un paiement (qui nécessitait une architecture réseau particulière).
- L'hébergement de logiciels libres et de sites web pour l'associatif Centralien.

Aujourd'hui, en discutant avec les anciens et les nouveaux membres de l'association il a été convenu que la direction souhaitée pour ÉCLAIR à l'avenir est la promotion de services et de logiciels libres ainsi que le développement de nouveaux outils (par exemple MyECL : une application web visant à faciliter la vie associative au quotidien). Ces projets nécessitent une infrastructure fiable, performante et facile à maintenir ainsi qu'une capacité de stockage importante. Nous avons donc commencé à construire une infrastructure classique d'hébergement (détaillée dans la suite) à partir de ce qui existait déjà.

### 6.2 Pilotage

Le recul nécessaire pour saisir les enjeux du projet a été long à acquérir. On peut diviser le projet en plusieurs phases (*figure 1*) :

- Phase 1 : Il a d'abord fallu nous former sur la gestion d'un réseau informatique et l'utilisation des systèmes d'exploitation basés sur Linux dont nous avons rédigé un guide pour aider à la prise en main (voir annexe 1A). Nous nous sommes formés pendant environ 2 mois de manière intensive puis cette formation s'est poursuivie en parallèle de la phase 2 jusqu'en décembre environ.
- Phase 2 : L'installation d'ÉCLAIR était peu documentée, il nous a fallu faire de la rétro ingénierie pour comprendre son fonctionnement en détail. Idem cette phase a empiété sur la phase 3 car nous avons fait certaines découvertes tardivement.
- Phase 3 : Une fois les besoins de l'association bien définis, nous avons progressivement mise à jour les machines. Puis nous avons configuré de nouveaux outils récents qui permettent d'améliorer la prise en main de l'installation et documenté ces derniers.

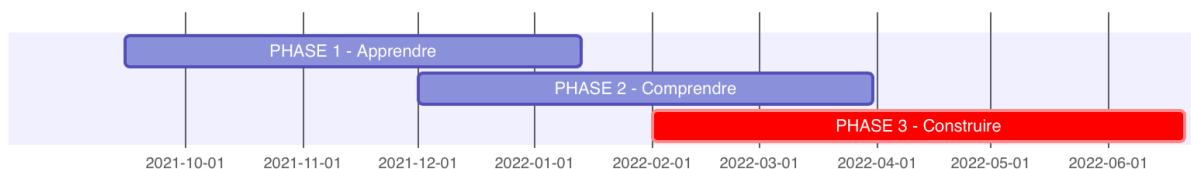


FIGURE 1 – Grandes étapes du projet

Les outils de pilotage utilisés par notre groupe sont détaillés dans le rapport de RVP1 et de RVP2.

## 7 Théorie des réseaux informatiques

Cette section vise à expliquer rapidement différentes briques élémentaires qui constituent une infrastructure réseau et se base en partie sur [2], pour réutiliser ces notions dans la suite du rapport. Nous détaillerons uniquement le fonctionnement de celles utilisées dans l'infrastructure d'ÉCLAIR afin de ne pas surcharger le rapport.

### 7.1 Prérequis théoriques

Pour comprendre le fonctionnement du réseau, de la transmission de simple bits à l'envoi de données complexes (comme des requêtes HTTP), l'ensemble des protocoles et systèmes réseau peut être catégorisé en 4 couches standard (RFC 1122, Internet STD 3). Le modèle OSI plus récent et complexe qui énonce 7 couches n'est pas présenté ici car il prête à confusion. Le modèle de la RFC 1122 (*figure 2*) est largement suffisant pour comprendre le fonctionnement des réseaux LAN et Internet.

Profondeur	Nom	Données	Protocoles	Rôle
1	Lien	Frames	ARP, MAC, PPP, VLAN	Communication entre machines d'un même réseau local (i.e. non séparées par un routeur) reliées par divers systèmes physiques (Câble, Wifi, etc.)
2	Internet	Paquets	ARPANET, IPv4, IPv6	Communication entre machines appartenant à des réseaux locaux différents par routage
3	Transport	Datagrammes	TCP, UDP, SCTP, ICMP	Assurer l'intégrité des données (contrôle, renvoi, etc.), la segmentation (découpage en datagrammes), la gestion du flux, la séparation du trafic via la notion de port
4	Application	Données	HTTP(S), (S)FTP, NTP, DHCP, DNS, IMAP, SMTP, POP, LDAP, SSH, XMPP, IRC, Telnet, SSL/TLS	Transmettre des données de haut niveau entre des applications tournant sur des machines potentiellement différentes et situées potentiellement sur des réseaux locaux différents

FIGURE 2 – Tableau récapitulatif des protocoles les plus connus

Considérons le scénario de l'envoi d'une requête HTTP d'un client vers un serveur. La forme de la requête elle-même suit les spécifications du protocole HTTP. Le protocole TCP découpe la requête en datagrammes. Le protocole IP transforme le datagramme en

paquet en ajoutant l'adresse IP du serveur et réalise le routage vers la gateway du réseau local. Le protocole MAC formate le paquet en frame en ajoutant l'adresse MAC de la gateway. La Gateway récupère la frame et la retransforme en paquet afin de la router. Après routage, le paquet redévient une frame envoyée à la gateway suivante. Lorsque le paquet parvient au serveur, il est retransformé en datagramme puis récupéré sur le bon port par TCP (80 pour HTTP). TCP régénère les données à partir des datagrammes et l'application (le serveur web) peut alors traiter les données.

Une analogie peut être faite avec l'utilisation des lettres au Moyen-âge. Celles-ci pouvaient être physiquement transportées à pied, à cheval ou par pigeon voyageur d'une ville à l'autre ce qui constitue la couche Lien. Dans chaque ville l'adresse inscrite sur l'enveloppe était vérifiée afin de déterminer la ville suivante (routage) avant d'être transmise au bon messager. Ceci correspond à la couche Internet. Une fois arrivée, le destinataire pouvait vérifier l'intégrité de la lettre en examinant son sceau ce qui constitue la couche Transport. Enfin, le contenu de la lettre était mis en page selon un certain protocole et dans une certaine langue ce que l'on peut rapprocher de la couche Application.

## 7.2 LAN et VLAN

Un LAN (Local Area Network) est un réseau informatique local. Il regroupe des machines dans un réseau physique délimité : une maison, une école, un laboratoire, ou un ensemble de bureaux.

VLAN pour Virtual LAN est un protocole de la couche *Lien* permettant de faire coexister plusieurs réseaux locaux virtuels sur un même ou plusieurs réseaux locaux physiques. Pour ce faire, un numéro de VLAN, nommé tag, est ajouté à chaque paquet dans le but d'identifier le VLAN auquel il appartient. Bien qu'un VLAN se comporte exactement comme un réseau local physique et qu'on puisse donc y faire coexister plusieurs réseaux IP, il est très courant d'associer un VLAN à un réseau IP. Dans ce cas, et si tous les réseaux locaux disposent de la même configuration, les VLAN permettent l'isolation de plusieurs réseaux de la couche Internet qui coexistent pourtant sur les mêmes réseaux locaux.

## 7.3 RAID

Les technologies RAID (ou *Redundant Array of Independent Disks*) [14] permettent d'assembler plusieurs disques sous forme d'une matrice. On identifie la configuration des disques par un numéro : RAID 0, 1, 5, 6 ou 10. L'assemblage de plusieurs disques physiques apparaît alors sur la machine comme un unique moyen de stockage, dont les propriétés dépendent du type de RAID choisi comme le montre la figure 3.

### Niveaux RAID : scénarios d'application

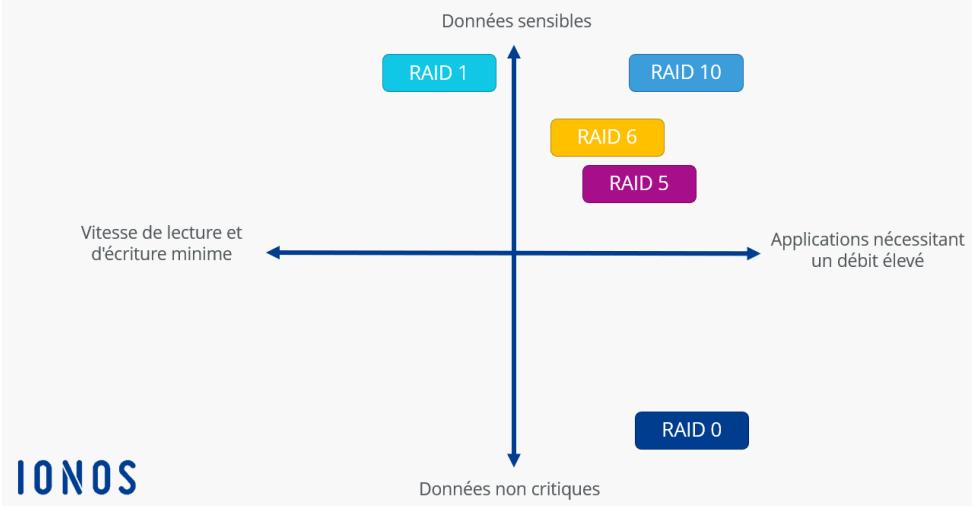


FIGURE 3 – Comparaison des différents RAID (d'après IONOS)

- Le RAID 0 permet d'augmenter la vitesse de lecture et écriture du stockage. Les machines écrivent alors une partie des données sur chaque disque.
- Le RAID 1 permet d'améliorer la redondance de l'information et donc la tolérance aux pannes de l'infrastructure : chaque donnée est écrite simultanément sur deux disques. Le stockage total de l'assemblage RAID correspond à la moitié de la capacité totale des disques.
- Le RAID 5 permet un compromis entre vitesse de lecture et redondance. Les données sont réparties sur plusieurs disques pour améliorer la vitesse de lecture et écriture tout en étant présentes en plusieurs exemplaires. Ce RAID nécessite un minimum de trois disques mais présente une capacité de stockage intéressante.
- Le RAID 10 (ou RAID 1+0) combine les concepts du RAID 1 et 0 : chaque données est écrite en deux exemplaires en parallèle sur plusieurs disques. Ce type RAID nécessite plus de disques que les autres.

La *figure 3* permet de comparer l'équilibre entre fiabilité et vitesse de lecture et écriture des différents RAID.

La technologie RAID à utiliser dépend alors de la vitesse et la fiabilité recherché. Nous avons décidé d'employer un RAID 5 pour la future machine d'ÉCLAIR afin d'améliorer les performances et la tolérance aux pannes de celle-ci, tout en limitant le coût et la quantité de disques à employer.

L'assemblage des disques peut-être logiciel (c'est-à-dire que le système d'exploitation détecte les différents disques et se charge de mettre le RAID en place), ou bien matériel (la mise en place du RAID est faite physiquement avant la connexion des disques au système d'exploitation). Cette deuxième possibilité est plus intéressante, un microprocesseur spécialisé est chargé de faire l'interface entre les disques et la machine. La majorité des serveurs de l'association sont configurés avec un RAID matériel.

Il est important de noter que si les technologies RAID peuvent garantir la redondance des données, cela ne peut être considéré comme une solution de sauvegarde, mais ne

permet que d'améliorer la tolérance aux pannes.

## 7.4 Reverse-proxy

Le reverse-proxy est installé du côté des serveurs web. L'utilisateur passe par son intermédiaire pour accéder aux applications de serveurs internes. Il sert donc à gérer les flux web entrants dans le réseau. Les applications d'un reverse proxy sont notamment :

- Mémoire cache : le reverse proxy peut décharger les serveurs Web de la charge de pages/objets statiques (pages HTML, images) par la gestion d'un cache web local. La charge des serveurs Web est ainsi généralement diminuée. On parle alors d'« accélérateur web » ou d'« accélérateur HTTP ».
- Intermédiaire de sécurité : le reverse proxy protège un serveur Web des attaques provenant de l'extérieur. En effet, la couche supplémentaire apportée par le reverse proxy peut apporter une sécurité additionnelle. La ré-écriture programmable des URL permet aussi de masquer et de contrôler, par exemple, l'architecture interne des sites web.
- Load balancing : le proxy inverse peut distribuer la charge d'un site unique sur plusieurs serveurs web applicatifs.
- Compression : le reverse proxy peut optimiser la compression du contenu des sites.
- Authentification : Le reverse proxy peut contrôler l'authentification avant accès à un serveur Web ne possédant pas cette fonction.

Lorsqu'un utilisateur veut accéder à un service hébergé par une machine, il contacte le reverse-proxy. Celui-ci retransmet alors les informations entre le client et la machine recherchée.

## 7.5 Pare-feu

Un pare-feu est un logiciel et matériel qui permet de séparer deux réseaux et d'augmenter la sécurité (*figure 4*). Il s'agit ici d'un serveur possédant deux cartes réseau : une première connectée au réseau local et une deuxième au réseau externe (accessible dans le cas d'ÉCLAIR par le réseau de la DSI). En faisant la jonction entre ces deux réseaux, l'objectif est de filtrer les connexions nous voulues et / ou non autorisées entre les deux. Cela permet d'empêcher le fonctionnement de certains protocoles trop permissifs, mais aussi pour certains pare-feux d'afficher ce qui a été bloqué pour voir si le réseau a été attaqué. La norme est d'avoir un pare-feu à l'entrée de chaque réseau géré par un organisme différent, c'est pourquoi ECLAIR possède son propre pare-feu, en plus de celui de la DSI.

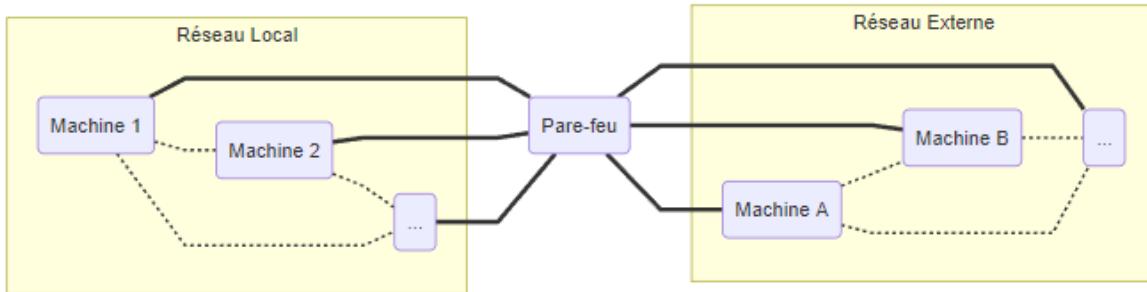


FIGURE 4 – vue schématique d'un pare-feu

## 7.6 DNS

Le DNS (Domain Name System, système de nom de domaine) [16] est en quelque sorte le répertoire téléphonique d'Internet. Les internautes accèdent aux informations en ligne via des noms de domaine (par exemple, `eclair.ec-lyon.fr` ou `myecl.fr`), tandis que les navigateurs interagissent par le biais d'adresses IP (Internet Protocol, protocole Internet). Le DNS traduit les noms de domaine en adresses IP afin que les navigateurs puissent charger les ressources web (*figure 5*)

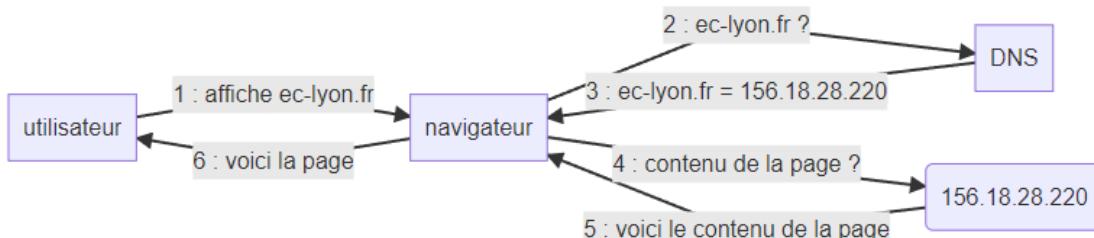


FIGURE 5 – Fonctionnement (très) simplifié du DNS

Le DNS d'ÉCLAIR est très simple. Une seule entrée est configurée, il s'agit d'une *wildcard* qui renvoie tous les sites terminant par `.eclair.ec-lyon.fr` vers le reverse proxy. Cette section ne nécessite pas plus de détail, cependant l'annexe 1B explique en détail le fonctionnement d'un DNS car cette information n'était pas connue de notre équipe lorsque nous avons commencé à nous renseigner dessus grâce à [15, 30, 4, 20].

## 7.7 SSH

### Bases

Le SSH [32] est un protocole sécurisé permettant de se connecter à distance à un serveur. Ceci permet d'administrer les serveurs depuis l'extérieur de la salle, qui n'est généralement pas très accueillante et confortable (bruit, températures élevées dans le cas d'ÉCLAIR, ...). On peut aussi avoir besoin d'utiliser les serveurs à des horaires ou depuis des lieux où on ne peut pas accéder à la salle, il est alors très utile de pouvoir s'y connecter depuis notre poste personnel.

Dans une architecture SSH, on distingue au moins deux machines : "le client", qui souhaite se connecter à "l'hôte". Le client est par exemple l'ordinateur personnel d'un administrateur système, qui souhaite se connecter à un serveur : l'hôte.

### Configuration simple

Sur une architecture simple le client est sur le même réseau que l'hôte. Les seules configurations à faire sont sur l'hôte, qui doit alors démarrer un service SSH (généralement sur le port 22). Par défaut dans la configuration la plus simple, n'importe qui peut tenter de se connecter sur l'hôte et essayer un mot de passe au hasard.

Il est alors possible de mettre en place des « Clés SSH » afin de sécuriser la connexion. Le client crée une paire de clés (une publique et une privée) et envoie sa clé publique sur l'hôte. Pour l'envoi de cette clé, il y a deux solutions : si le client peut encore se connecter par mot de passe il envoie alors la clé directement à l'hôte, sinon quelqu'un ayant accès au serveur (soit en local soit par une autre connexion SSH) doit y copier la clé. Une fois la clé copiée, on peut verrouiller tous les accès par mot de passe au SSH, seules les personnes ayant copié une clé sur le serveur pourront alors s'y connecter.

La problématique classique d'une entreprise qui aurait de nombreux serveurs, et surtout de nombreux clients est qu'il faudrait créer et copier une clé pour chaque client sur chaque serveur. Il n'y aurait aucun moyen de gérer simplement quels clients peuvent accéder à quelles machines, ou même de supprimer des clients. De plus, il n'est pas sécurisé de laisser tous les serveurs accessibles depuis un réseau public. Il existe des solutions permettant de palier ces problématiques, dont l'une mise en place par ÉCLAIR. Nous détaillerons le fonctionnement de cette solution dans la section 9.2.6.

## 7.8 DHCP

Dynamic Host Configuration Protocol (DHCP, protocole de configuration dynamique des hôtes) est un protocole réseau dont le rôle est d'assurer la configuration des paramètres IP d'une station ou d'une machine, en lui attribuant automatiquement une adresse IP et un masque de sous-réseau.

Concrètement, il attribue une adresse IP aux clients qui se connectent au réseau internet du M16 (administré par ÉCLAIR) ce qui permet de les identifier dans ce dernier.

## 7.9 NTP

Network Time Protocol (NTP) est un protocole utilisé pour synchroniser les heures d'horloge des ordinateurs dans un réseau. Il appartient et est l'une des parties les plus anciennes de la suite de protocoles TCP/IP. Le terme NTP s'applique à la fois au protocole et aux programmes client-serveur qui s'exécutent sur les ordinateurs.

## 7.10 Serveur web

Un serveur web est, soit un logiciel de service de ressources web (serveur HTTP), soit un serveur informatique (ordinateur) qui répond à des requêtes du World Wide Web sur

un réseau public (Internet) ou privé (intranet) en utilisant principalement le protocole HTTP.

Un serveur informatique peut être utilisé à la fois pour servir des ressources du Web et pour faire fonctionner en parallèle d'autres services liés comme l'envoi d'e-mails, l'émission de flux streaming, le stockage de données via des bases de données, le transfert de fichiers par FTP, etc.

## 7.11 Virtualisation et conteneurisation

### 7.11.1 Hyperviseur

La virtualisation consiste à créer plusieurs machines virtuelles (VM) à partir d'une seule machine physique à l'aide d'un logiciel appelé hyperviseur. Parce que les machines virtuelles fonctionnent de la même manière que des machines physiques, mais ne s'appuient sur les ressources que d'un seul système informatique, la virtualisation permet aux directions informatiques d'exécuter plusieurs systèmes d'exploitation sur un seul serveur (connu également sous le nom d'hôte). Pendant ce temps, l'hyperviseur attribue des ressources informatiques à chaque ordinateur virtuel, en fonction des besoins.

Un des gros avantages des machines virtuelles est la portabilité de celles-ci. On peut en effet faire des "snapshots" des machines, qui sont des images de tout le système à un instant t, et qui permet donc de sauvegarder tout le système. Une fois la snapshot créée, en cas de problème on peut en quelques clics remettre en place la machine entièrement. Sur une machine physique en l'état des choses à ÉCLAIR, il serait impossible de remettre en place la machine facilement, il faudrait alors réinstaller tout le système d'exploitation et refaire toutes les configurations à la main, un procédé long et laborieux.

Il existe deux types d'hyperviseurs, les types 1 ou "bare metal" sont des logiciels qui jouent aussi le rôle de système d'exploitation, ils sont installés directement sur la machine physique. Des exemples sont *xcp-ng*, *Proxmox*... Le deuxième type d'hyperviseur, bien plus connu est appelé "hosted". Ces hyperviseurs prennent la forme de simple logiciels qui s'installent sur des systèmes d'exploitations plus conventionnels, tels que *Debian*. Des exemples sont *VirtualBox* et *VMWare Fusion*.

Dans le cas des hyperviseurs de type 2, les méthodes de virtualisation peuvent varier en fonction du système d'exploitation de l'utilisateur. Par exemple, les machines Linux proposent un hyperviseur open source unique : le KVM (Kernel-based Virtual Machine). Le KVM faisant partie de Linux, il permet à la machine hôte d'exécuter plusieurs machines virtuelles sans hyperviseur distinct. L'hyperviseur utilisé par ECLAIR avant notre projet d'étude était QEMU basé sur KVM.

### 7.11.2 Docker

Docker est un outil utilisé par ÉCLAIR sur ses serveurs (et largement utilisé ailleurs) afin de gérer et d'organiser les différents services web ou réseaux proposés par l'association.

Le principe de Docker est de créer des containers qui sont des blocs spécialement créés pour l'utilisation d'un service (par exemple un serveur *Apache*, *NodeJs* ou *MySQL*). L'utilisation d'un container ressemble à celle d'une machine virtuelle excepté qu'un container repose sur le kernel linux de la machine hôte ce qui en fait un élément beaucoup plus léger

sur un serveur et exportable facilement sur d'autres serveurs. Un autre avantage à utiliser des containers est que chaque container est isolé du reste du système et n'a donc pas accès aux autres programmes, ce qui rajoute une couche de sécurité. Le container docker peut communiquer avec d'autres containers à travers ses ports ou au sein d'un même réseau de containers. Les principales différences avec la virtualisation sont résumés sur la *figure 6*.

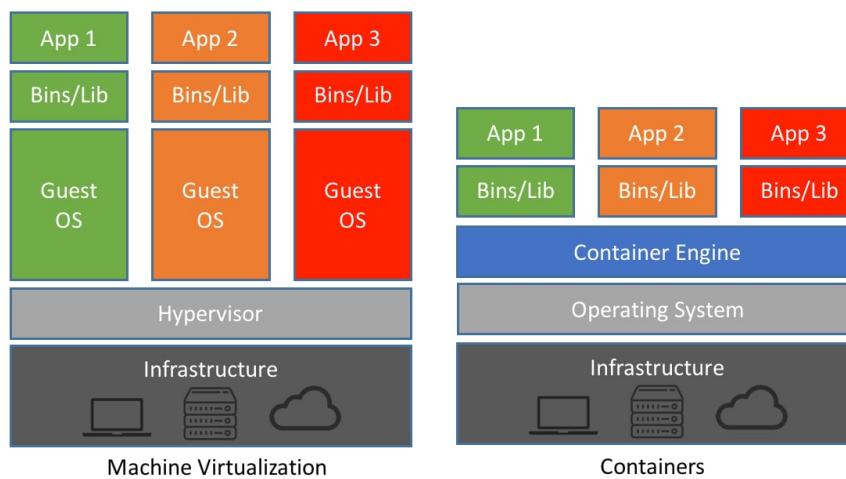


FIGURE 6 – Comparaison machine virtuelle et docker (d'après NetApp.com)

**Les images :** Une image est un ensemble de fichiers permettant de recréer un logiciel, un serveur, une distribution linux, etc ... dans un container. Les images se téléchargent à partir d'un "registry" ou bibliothèque en ligne, contenant une immense quantité d'images. Le principal registry Docker est le Docker hub, c'est celui qui est utilisé par défaut. La plupart des container partent donc d'une image pré-construite sur le registry puis ajoutent ses propres fichiers dans le container.

**Les volumes :** Un des inconvénients à l'utilisation de container seuls est qu'il est alors impossible de garder des fichiers après la suppression d'un container car ceux-ci sont tous supprimés. Cela est par exemple problématique lorsque l'on souhaite garder une base de données tout en supprimant une application devenue obsolète. Un autre problème est celui de pouvoir modifier facilement les fichiers contenus dans un container sans avoir à le reconstruire en entier. Ces deux problèmes peuvent être résolus grâce aux volumes. Ce sont des espaces mémoire partagés entre le serveur et un container docker (généralement stocké sur /var/lib/volumes/<volume>). Ce dossier sera donc accessible sur la machine hôte de Docker et dans le container Docker.

**Le Dockerfile :** Le Dockerfile permet la création d'images personnalisées. Il est écrit avec une syntaxe spéciale et décrit les différentes étapes (layers) nécessaires à la création d'une image, l'installation de certains packages spécifiques ou l'importation des fichiers de l'application.

**Docker-compose :** Docker-compose est un exécutable Docker permettant de créer facilement des applications multicontainers, créer les volumes associés à chaque container,

relier les ports entre la machine hôte du container et le container lui-même et permet l'ajout d'un grand nombre d'options à la création de containers. On l'utilise grâce à un fichier *docker-compose.yml*.

## 8 Rétro ingénierie

Le problème original ayant mené à la création de ce PE est le manque de documentation et de compréhension de l'association ÉCLAIR face à sa salle serveur. L'objectif du projet était donc, avant de mettre à niveau la salle, d'en récupérer le contrôle et la compréhension, tout ceci en l'absence d'une documentation solide et fiable sur laquelle s'appuyer. La partie de rétro-ingénierie a donc constitué une grande partie du travail fourni par le groupe du projet, en parallèle avec des formations permettant de comprendre les technologies que nous manipulions.

Cette phase du projet a été séparée en plusieurs parties : tout d'abord, nous avons réalisé un inventaire complet de l'infrastructure, que ce soit du point de vu physique ou informatique. Nous avons ensuite tenté de comprendre au mieux le fonctionnement du réseau du bâtiment associatif et où chaque port ethernet du bâtiment arrivait dans la baie de brassage de la salle serveur. Ceci nous a mené à tenter de comprendre l'entièreté de l'architecture réseau d'ÉCLAIR : que ce soit pour les serveurs de l'association sur des VLANS privées ou pour desservir l'étage associatif en réseau, sur des VLANS publiques.

### 8.1 Inventaire

Une des premières choses que nous avons fait dans le cadre de notre PE a été de lister le nombre de machines ainsi que leurs spécifications (hardware et software) sur un tableau. Ce document nous a été utile pendant tout le PE : pour des problématiques de dimensionnement, pour savoir où stocker les données les plus volumineuses, pour connaître les adresses IP de chaque machine, ...

#### 8.1.1 Physique

Cette partie consistait à lister l'entièreté des équipements physiques présents, que ce soit les serveurs, les onduleurs, les switchs et les distributeurs. Pour chacune de ces machines nous avons noté le nom de la machine, et pour les serveurs nous avons aussi noté leur configuration physique : modèle du processeur, nombre de coeurs, quantité de mémoire vive, quantité de mémoire sur disques durs, etc ... Une extract des informations récupérées est présentée dans la *figure 7*, concernant spécifiquement les machines.

Nom	Marque Modèle	Fonctions	Etat	OS	Disque dur / Espace restant	CPU / Total RAM	Accès BIOS
Artémis	HP Proliant DL160 G5	Serveur Compose Erebor	OK ?	Buster	2 HDD 72G + 2 HDD 500G		
Apollon	HP Proliant DL160 G5 (2008)	Dockers en production (dont wiki) + forge	OK ?	Stretch	2 HDD 72G + 2 HDD 500G disk 0-2 : UNCLAIMED disk 3 : sda, 1 volume disk 4 : sdb, 2 volumes	2x Intel Xeon E5405, 2GHz RAM : 9Go (24Go + 2512Mo)	F10
	HP P2000	Baie de stockage d'Atlas			10 HDD 2T	/	
Atlas	HP Proliant DL360p Gen 8	Hyperviseur pour les VM tts les VM st pbt dessus	OK ?	Wheezy	2 HDD 300 G + 1 HDD 900G ; 2x300Go RAID1 1x900Go RAID0	2x Intel Xeon E5-2620 (6 coeurs) RAM : 32 Go	F10 (puis F8 pour contrôleur RAID)

FIGURE 7 – Extrait de l'inventaire

Ce passage de l'inventaire n'a posé que peu de problèmes, puisqu'il suffisait de saisir quelques commandes pour obtenir toutes ces configurations.

### 8.1.2 Logiciel

L'inventaire logiciel en revanche a causé bien plus de difficultés. Il y avait deux objectifs : récupérer le nom et la version du système d'exploitation de chaque machine, puis lister les services que fournit cette machine pour comprendre son utilité dans l'infrastructure de l'association. La version du système d'exploitation se récupère sans difficulté sur la plupart des machines, c'est la seconde partie qui pose problème.

En effet, il est très compliqué de savoir quels services une machine fournit lorsque l'on n'est pas habitué à reconnaître les services par défaut d'une machine *Debian*. La partie problématique est que nous n'avons pas trouvé de solution simple pour lister seulement les services installés par l'utilisateur, il fallait donc trier les paquets présents par défaut sur le système pour qu'il fonctionne, et ceux installés par les membres d'ÉCLAIR pour donner son utilité à la machine. Nous avons donc commencé par lister tous les services, et chercher sur internet à quoi chacun d'entre eux servait. Le résultat de cette première méthode est trouvable en annexe 1G.

Cette méthode n'atteindra jamais complètement son but, et nous n'avons réussi à comprendre à quoi servait chaque machine seulement après avoir obtenu un document des anciens de l'association (promo 2016 et 2017) qui résumait globalement l'utilité des machines, sans préciser quelles solutions étaient utilisées pour remplir cette utilité.

C'est donc seulement plusieurs mois plus tard que nous avons pu officiellement déclarer cet inventaire terminé, après avoir gagné énormément de connaissances et pris beaucoup de distance par rapport à l'architecture.

## 8.2 Le réseau physique du M16 et de la salle serveur

L'étage associatif du M16 ainsi que les locaux associatifs au sous-sol, excepté le Faculty Club, sont historiquement alimentés en réseau internet par câble ethernet par ECLAIR. Cela est fait à partir d'un accès internet fourni par la DSI ainsi qu'un certain nombre de switchs situés dans la baie de brassage de la salle serveur. Cette baie de brassage distribue également le réseau aux machines de la salle serveur.

*Il faut noter que le Bazar, l'association de jeux vidéos et de jeux de société de Centrale, qui utilise le Faculty Club, cherche actuellement la baie de brassage reliée aux prises RJ45 de ses locaux pour pouvoir y connecter leurs consoles. Pour le moment, ces prises ne sont pas alimentées. ECLAIR et l'équipe du PE57 ne sont pas en mesure de les aider.*

Pour pouvoir comprendre au mieux la configuration réseau de la salle serveur (et par extension, de l'accès internet associatif), il a fallu documenter intégralement les branchements de la baie de brassage et la configuration des switchs qu'elle comprend. Dans la logique de transmettre les connaissances accumulées, des fiches d'utilisation des switchs ont été rédigées et sont disponibles en annexe 2A.

### 8.2.1 Conventions de notation

Parmi la documentation disponible sur la salle serveur, quasiment rien ne concernait la baie de brassage : il a donc fallu partir de zéro. Après avoir enlevé les matériels mutilés et amélioré la gestion des câbles (*figure 8*), nous avons choisi un certain nombre de conventions pour pouvoir nommer les éléments de la baie.

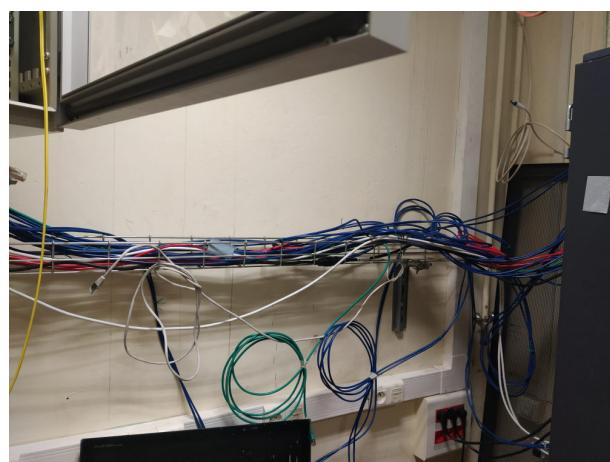


FIGURE 8 – Câbles avant rangement et réorganisation

Nous avons choisi de nommer chaque élément du rack selon la forme "Lettre identifiant la fonction" + "Position sur le rack". Ainsi, nous avons convenu des notations suivantes :

- F1, F2 : Arrivée fibre (F) de la DSI

- D3, D4 : Distributeurs (D) vers les ports RJ45 des locaux associatifs
- S5, S6, S7 : Switches (S) fournissant l'accès internet vers les distributeurs

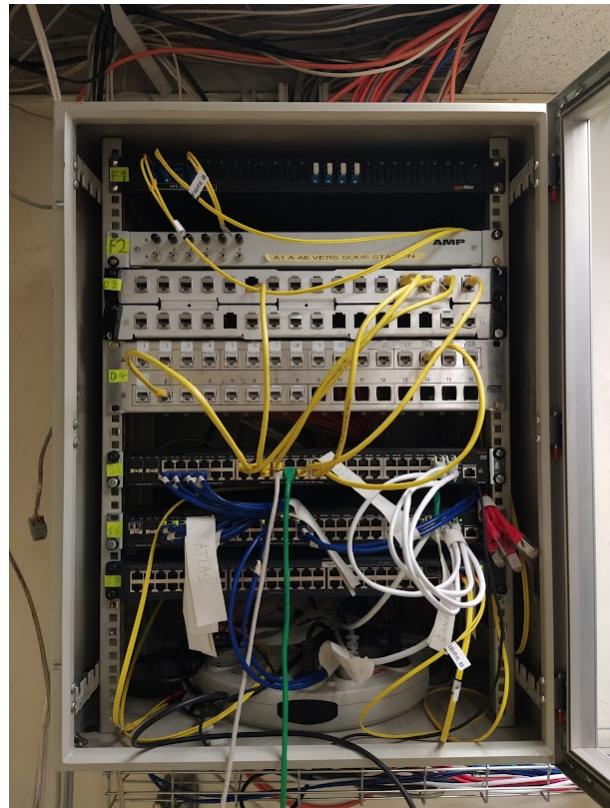


FIGURE 9 – Vue de la baie de brassage

Ensuite, pour identifier chaque emplacement présent sur les distributeurs D3 et D4 montrés sur la figure *figure 9*, nous avons choisi de compter leur position sur le tableau même si un emplacement était vide plutôt que d'utiliser les étiquettes déjà présentes dessus. En effet, ces étiquettes semblaient provenir d'années différentes et leur notation était incohérente (voir la *figure 10*) : probablement, il avait été entrepris de les décoller précédemment mais le travail n'avait pas été terminé. Il y a 16 emplacements par ligne et il y a 2 lignes par distributeur : ainsi, il y a 32 emplacements, de DX-1 à DX-32. Par exemple, le troisième emplacement de la seconde ligne du troisième distributeur est notée D3-19, qu'elle ait un port RJ45 d'installé dessus ou non. Cette notation se rapproche de celle d'un switch, où chaque port à déjà un numéro qu'importe que quelque chose soit branché dessus ou non. De la même manière, on note par exemple S6-3 le troisième port du switch 6 ou F1-2 la seconde arrivée fibre de la DSi.



FIGURE 10 – Vue des distributeurs

Pour ce qui est de l'arrivée des ports liés au distributeur, nous avons choisi d'utiliser le numéro de la salle tel que défini dans les plans du bâtiment (voir *figure 11*). En effet, si on utilisait le nom de l'association logeant dans le local, la notation deviendrait vite obsolète car il arrive qu'elles changent de place : sur la *figure 11*, on voit que les emplacements ne sont pas à jour. Avec ces notations, le deuxième prise RJ45 de la salle 103 serait noté 103b.



FIGURE 11 – Plan du M16 - étage associatif

A partir de ces notations, on peut alors définir clairement les connexions : par exemple,

S6-3 est connecté à D3-19 fournissant internet à la prise 103a. On peut ensuite établir des tableaux, ou des cartographies plus visuelles.

### 8.2.2 Cartographie du réseau

De par le manque de documentation de la baie de brassage, certains codes d'accès administrateur aux switches ont été perdus. Pour deux switches sur trois (S5 et S6), il a été possible avec une manipulation avec les boutons en façade (expliquée dans le manuel de ceux-ci [13]) de ne réinitialiser que le mot de passe en gardant la configuration inchangée. Le dernier, S7, ne peut pas être réinitialisé de cette manière et sa configuration n'a pas pu être documentée ni modifiée. Néanmoins, cela ne pose pas de très gros problèmes car ce switch paraît appartenir à la DSI ce qui laisse supposer que sa configuration est appropriée, et que dans tout les cas elle ne doit pas être modifiée par les membres du PE. Nous reparlerons de la configuration des switchs dans une partie suivante.

Pour déterminer à quel port du distributeur étaient reliées les prises des locaux associatifs, nous avons utilisé un testeur de ligne RJ45 comme celui *figure 12*. Ce dispositif est composé de deux parties à relier de part et d'autre de la ligne. Pour chaque pin du port RJ45 (qui en comporte 8), l'appareil teste la continuité de la ligne. Une lumière s'allume lorsque c'est le cas. Ainsi, si toutes les lumières s'allument successivement et sur les deux parties du testeur, il existe une connexion entre les deux côtés de la ligne. Il arrive parfois à cause de branchements ou de problèmes de masse que seulement un côté s'allume, c'est pourquoi il est important de bien vérifier visuellement les deux parties du testeur.



FIGURE 12 – Testeur RJ45 (d'après Conrad)

Toutes les connexions n'ont pas pu être trouvées malgré plusieurs tests fastidieux, mais nous avons pu trouver les suivantes (X signifie qu'il n'y a pas de port RJ45 à cet emplacement et ? que l'extrémité n'a pas été trouvée) :

D3-1	?	D3-17	?	D4-1	142a	D4-17	?
D3-2	?	D3-18	?	D4-2	142b	D4-18	114
D3-3	130a	D3-19	?	D4-3	142c	D4-19	?
D3-4	130b	D3-20	121a	D4-4	142d	D4-20	?
D3-5	124a	D3-21	X	D4-5	142e	D4-21	123b
D3-6	124b	D3-22	?	D4-6	142f	D4-22	123a
D3-7	123c	D3-23	113a	D4-7	142g	D4-23	?
D3-8	?	D3-24	113b	D4-8	142h	D4-24	120
D3-9	?	D3-25	?	D4-9	142i	D4-25	X
D3-10	119	D3-26	X	D4-10	142j	D4-26	X
D3-11	?	D3-27	X	D4-11	142k	D4-27	X
D3-12	?	D3-28	X	D4-12	142l	D4-28	X
D3-13	118a	D3-29	X	D4-13	142m	D4-29	X
D3-14	118b	D3-30	X	D4-14	142n	D4-30	X
D3-15	118c	D3-31	?	D4-15	142o	D4-31	X
D3-16	117	D3-32	?	D4-16	121b	D4-32	X

TABLE 1 – Tableau de correspondances entre les distributeurs D3 et D4 et les salles

### 8.3 Présentation de l'architecture réseau d'ÉCLAIR

Après avoir fait un plan de la structure du réseau physique du M16, nous avons dû documenter puis simplifier le fonctionnement "immatériel" de celui-ci.

Le réseau d'ÉCLAIR est inclus dans celui de la DS1. Ainsi tous les paquets qui arrive jusqu'au réseau local d'ÉCLAIR sont filtrés par le pare-feu de la DS1 puis par le pare-feu d'ÉCLAIR. Le reverse-proxy lit les entêtes des paquets et les dirige vers le destinataires. Idem dans l'autre sens (hôte vers clients).

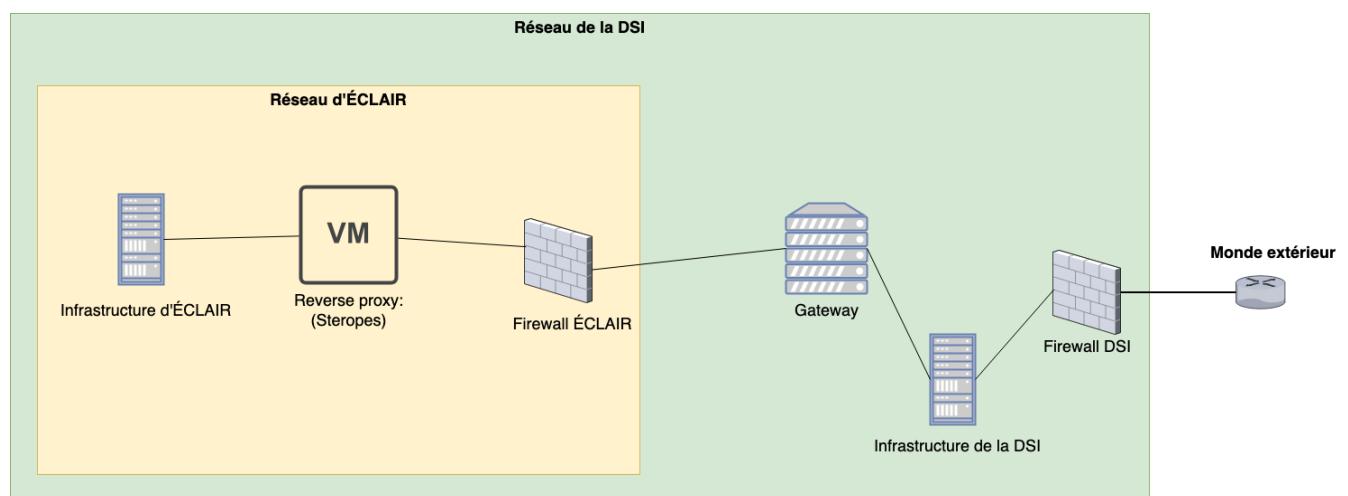


FIGURE 13 – Routage des paquets

L'organisation des serveurs est la suivante. Deux VLAN sont utilisés (la 156 et la 10). La 156 est relié à la DS1 (la porte d'entrée du réseau est le gateway). La 10 permet de communiquer aux machines entre elles (en local).

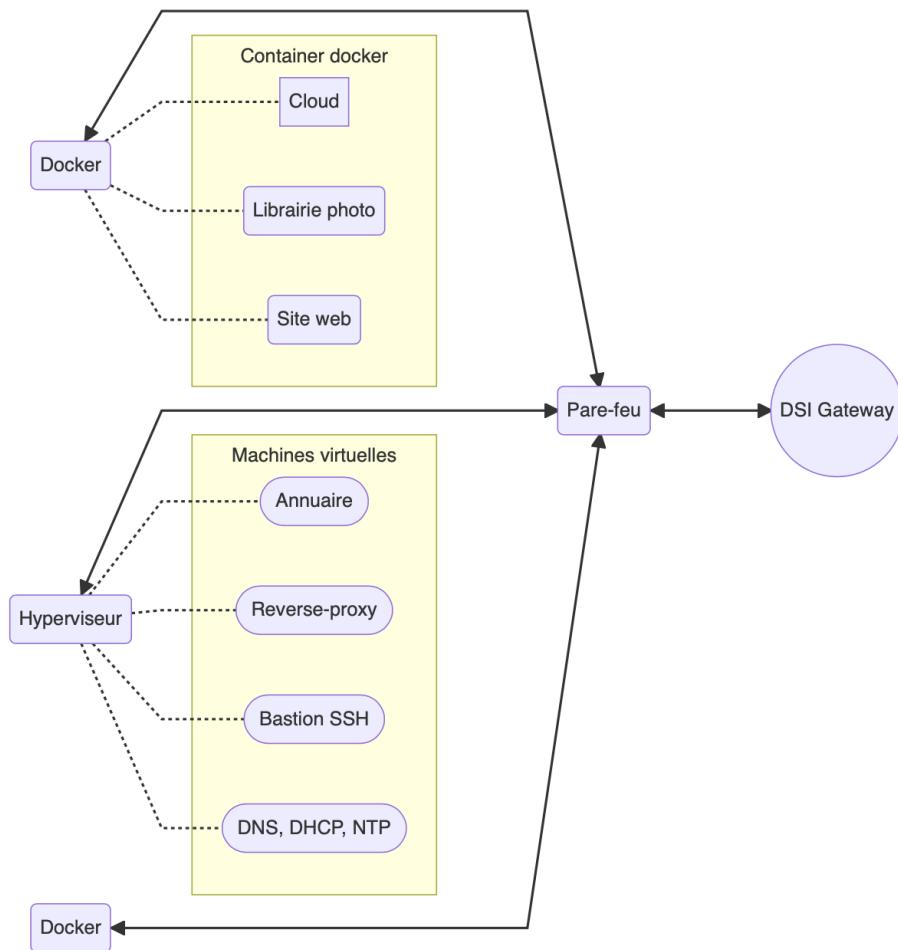


FIGURE 14 – Structure du réseau d'ÉCLAIR

### 8.3.1 Communication avec les switchs

Pour savoir comment fonctionne le réseau du M16, il a fallu communiquer avec les switchs pour connaître la manière dont ils sont configurés. Pour cela, nous avons commencé par utiliser un port série ethernet, suivi d'un convertisseur USB série tels que présentés *figure 15* et *figure 16* pour brancher au port "console" de S5 et S6 un ordinateur.



FIGURE 15 – Câble ethernet-série (d'après Ebay)



FIGURE 16 – Câble USB-série (d'après Startech)

Une autre manière de faire, découverte plus tard durant le PE, consiste simplement à utiliser le port micro-USB en façade du switch (qui n'est pas marquée explicitement en tant que port "console"), mais cela demande des drivers supplémentaires. Cette alternative est amenée à devenir le moyen principal de communication avec les switchs, car depuis *Windows 11* le driver de notre convertisseur USB-série n'est plus compatible, cette version ayant introduit des changements dans la gestion des périphériques.

Du côté de l'ordinateur, nous avons utilisé le logiciel open-source et gratuit *Tera Term*. Ce logiciel bien documenté ([29]) permet d'afficher le résultat d'une communication série sous la forme d'une console. En figure *figure 17* et *figure 18*, sont présentés des exemples d'affichage sur *Tera Term* de la communication série.

```

orome                                         27-Jan-1990 22:46:08
===== CONSOLE - OPERATOR MODE =====
Status and Counters - General System Information

System Contact      :
System Location     :

Software revision   : V0.16.02.0012      Base MAC Addr    : ecebb8-3a4000
ROM Version         : V0.16.01.0001      Serial Number   : CN79JYJ078

Up Time             : 26 days          Memory - Total  : 369,500,672
CPU Util (%)        : 0               Memory - Free   : 270,758,272

IP Mgmt - Pkts Rx : 101,153        Packet - Total  : 6600
                                Pkts Tx : 100,987        Buffers - Free   : 4859
                                                               Lowest   : 4838
                                                               Missed  : 0

Actions-> Back | Help
Return to previous screen.

Use arrow keys to change action selection and <Enter> to execute action.

```

FIGURE 17 – Affichage et paramétrage de l'assignement des VLAN aux ports

Switch Configuration - VLAN - VLAN Port Assignment				
Port	DEFAULT_VLAN	erebor	LAN	switch
1	No	No	Untagged	No
2	No	Tagged	No	No
3	No	No	Untagged	No
4	No	Tagged	No	No
5	No	No	Untagged	No
6	No	Tagged	No	No
7	No	No	Untagged	No
8	No	Tagged	No	No
9	No	No	Untagged	No
10	No	No	Untagged	No
11	No	No	Untagged	No
12	No	No	Untagged	No

Actions-> Cancel Edit Save Help

Cancel changes and return to previous screen.

Use arrow keys to change action selection and <Enter> to execute action.

FIGURE 18 – Affichage des informations système d'un des switches

Un tutoriel détaillé de l'utilisation de la communication série pour les switches S5 et S6 est disponible en annexe 2A.

### 8.3.2 Les VLAN utilisées

En utilisant le port série des switches, nous avons pu extraire la configuration des VLAN du réseau d'ECLAIR. Les VLAN utilisées sont les suivantes :

- *Default\_VLAN* (1) : Cette VLAN doit toujours exister et n'est pas supprimable. Elle est obligatoire pour certaines communications entre switches ou certains protocoles.
- *DMZ* (10) pour Dimilitarized Zone : c'est la VLAN privée des serveurs, non accessible depuis l'extérieur de la salle
- *erebor* (11) : Cette VLAN est une relique du système expérimental de paiement d'ECLAIR appelé *erebor*. Dans le cas où celui-ci serait à nouveau redéployé dans une forme ou une autre, celui-ci a été conservé.
- *LAN* (156) : Cette VLAN relie les locaux associatifs et une partie des serveurs
- *switch* (172) : C'est une VLAN particulière qui sert apparemment au trunk entre S5 et S6.

On présente dans les tableaux suivants le résultat de cette recherche :

Port(s)	Relié à	Default_VLAN	LAN	erebor	switch
1,3,5,7			Untagged		
2,4,6,8	Artémis			Tagged	
9-28	Locaux M16		Untagged		
29-36		Untagged			
37-44				Untagged	
45-58	trk26	Untagged	Tagged	Tagged	Tagged
49-52		Untagged			

TABLE 2 – Configuration VLAN de S5

Port(s)	Relié à	Default_VLAN	LAN	erebor	DMZ	switch
1-4	trk1 (Atlas)	Untagged		Tagged	Tagged	Tagged
5-8	trk2 (Apollon)	Untagged	Tagged	Tagged	Untagged	
9-12	trk3 (vide)	Untagged	Tagged	Tagged	Untagged	
13-14	trk4 (vide)	Untagged	Tagged	Tagged	Untagged	
15-16	trk5 (vide)	Untagged	Tagged	Tagged	Untagged	
17-18		Untagged	Tagged		Untagged	Tagged
19-22		Untagged				
23-24	trk6 (Gaïa)	Untagged	Tagged	Tagged	Untagged	
25-42		Untagged				
43	Heimdall IN		Untagged			
44	Gandalf		Untagged			
45-48	trk26	Untagged	Tagged	Tagged		Tagged
49-50			Untagged			
51-52		Untagged				

TABLE 3 – Configuration VLAN de S6

L'équipe du PE à très peu changé la configuration de ces switchs, et n'a utilisé quasiment que la VLAN 156 "LAN", car nous considérons que notre compréhension du fonctionnement des VLAN, des trunk et communications réseau ne sont pas suffisantes, et que les modifications mettaient trop en danger le bon fonctionnement du réseau de la salle et par conséquent l'avancement du PE. Néanmoins, nous pouvons faire les hypothèses suivantes :

- Trk26 est le trunk qui relie les deux switches, et qui permet de les "fusionner". Ce qui passe dans S5 est repliqué dans S6 et inversement.
- Les ports S5-9 à S6-28 sont réservés à la distribution du réseau aux locaux associatifs, d'où leur configuration "LAN : Untagged"
- La VLAN DMZ n'est présente que sur S6 et n'est pas configurée sur Trk26 (donc n'est pas sur S5), car celle-ci ne contient que les machines de la salle serveur.

Nous espérons que les prochaines générations du PE pourront comprendre en profondeur cette ancienne configuration et mieux la documenter. Nous attirons l'attention sur le fait que les tests sur ce genre de configurations nous paraissent complexes et multifactoriels (configuration réseau de la machine testant, pare-feu, paramètres implicites...), ce

qui rend cette partie chronophage et fastidieuse. Changer les paramètres lorsque d'autres personnes travaillent sur les serveurs sans les impacter est impossible, c'est pourquoi nous n'avons pas pu consacrer le temps nécessaire à cela (travailler là dessus à deux ou trois impliquait plus ou moins de mettre "au chômage" les membres du PE restants s'ils voulaient travailler sur les machines). En effet, si cette configuration est mal faite, le réseau associatif dans son entièreté peut ne plus fonctionner, ce qui impacte aussi les associations. Par exemple, le *SdeC*, auquel nous nous excusons pour avoir parfois sorti leurs imprimantes du réseau lors de nos séances de PE.

#### 8.4 Installation de la salle serveur

La baie de brassage visible sur la *figure 19* est composé d'un onduleur (voir section 9.1.1) ainsi que d'un multiplexeur qui permet de changer l'affichage de l'écran salon le serveur sur lequel on travaille. Enfin, la baie contient une baie de stockage qui est concrètement un disque dur externe avec beaucoup de stockage.

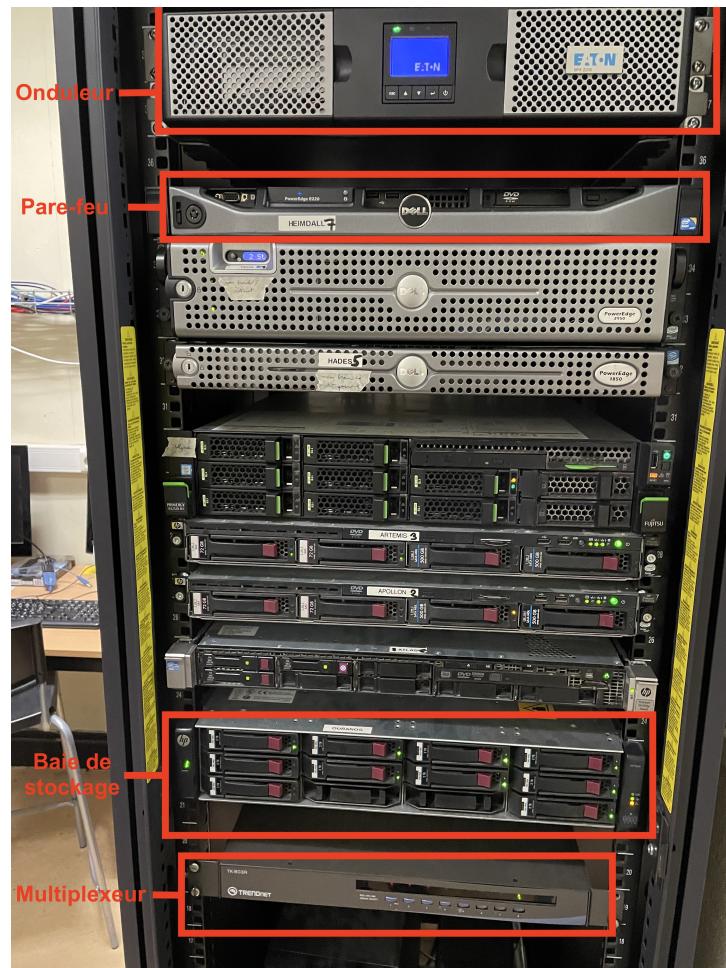


FIGURE 19 – Baie de brassage

## 9 Réalisations

La requête du commanditaire du projet d'étude était de remettre à niveau l'infrastructure réseau d'ÉCLAIR. Cette mission s'articulait en trois axes d'après le sujet qui nous a été communiquée en septembre. Il s'agissait d'effectuer un dimensionnement des serveurs en fonction des besoins de l'association (hardware), de faire une refonte de l'infrastructure réseau du point de vue logiciel et enfin de faire un bilan carbone de cette dernière.

### 9.1 Mise à niveau du hardware de l'infrastructure réseau

#### 9.1.1 Configuration des onduleurs

On compte deux (2) onduleurs dans la salle serveurs d'ÉCLAIR. Ces onduleurs sont de type EATON 9PX2200. Cette série offre une protection avancée pour petits et moyens datacenters, salles informatiques et infrastructures.

Le EATON 9PX2200 est doté de connecteurs série et USB, ainsi que d'un emplacement pour carte communication (en option sauf sur le modèle Netpack où elle est automatiquement incluse). Le By-Pass interne assure la continuité de service en cas de panne de l'appareil.

Les fonctions majeures des onduleurs sont l'alimentation sans interruption des machines auxquelles ils sont reliés, la protection des appareils informatiques contre les risques électriques comme les coupures de courant, les surtensions, les sous-tensions, etc. Un autre aspect intéressant est qu'ils lissent le courant de manière à ne pas endommager les composants électroniques.

L'une des fonctions importantes des onduleurs est la possibilité de gestion à distance de la salle des serveurs en décidant, grâce à un logiciel, de l'ordre d'extinction des machines lors d'une coupure d'électricité. En effet lors d'une coupure d'électricité, l'on peut recevoir des messages (SMS) ou des mails pour être alerté et décider de quoi faire. Ainsi, l'on aura le temps de sauvegarder les données ou de terminer certaines tâches avant d'éteindre minutieusement les machines.

Le logiciel responsable de cette fonctionnalité est : Network Shutdown module ; logiciel développé par Eaton et disponible gratuitement en ligne. Il faut de plus connecter l'onduleur, grâce au port Ethernet qu'il possède à internet (sur la LAN 156). Ces fonctionnalités nécessitent une carte de communication qui est en cours d'achat et qui sera installée par la prochaine itération du projet ou par notre groupe dans le cadre de notre mandat associatif.

#### 9.1.2 Dimensionnement des serveurs

Le dimensionnement d'un serveur faisant partie d'une infrastructure réseau diffère en fonction des besoins de l'association et du nombre d'utilisateurs censés être supportés par l'infrastructure. Le dimensionnement d'un serveur varie également en fonction des rôles, services et applications que ce dernier est amené à héberger ou exécuter. Concrètement, cela consiste à choisir les composants d'un serveur. Classiquement on se borne à choisir :

- La marque
- Le processeur

- La mémoire vive (RAM)
- Les disques durs
- L'alimentation
- Le contrôleur RAID
- Les interfaces réseau (essentiellement le nombre de ports ethernet et leurs débits dans notre cas)

Pour connaître nos besoins en terme de puissance de calcul, nous avons consulté les membres d'ÉCLAIR afin d'établir un cahier des charges fonctionnel (dont un extrait est visible *figure 20*) basé sur les services qu'il souhaitaient déployer sur un horizon de 1 à 2 ans. Nous avons aussi pris en compte la priorité de déploiement de ces services (certains étaient plus utiles que d'autres). Par exemple, déployer une suite bureautique collaborative coûte cher et ne semble pas être le service avec le plus de demande en raison de la forte compétitivité de *Microsoft* et *Google*. Il en est ressorti que la salle serveur devrait être capable d'héberger :

- Stockage cloud (instance Nextcloud)
- Galerie photo (instance Piwigo)
- Une plateforme d'hébergement vidéo
- Un serveur de communication (instance compatible avec le protocole Matrix)
- MyECL (plateforme de gestion développée par ÉCLAIR)
- Un outil de gestion de projet
- Des bases de connaissances reposant sur le projet Docausorus
- Les sites internet de différentes associations, PE et projets
- D'autres services avec un coût négligeable en puissance de calcul et en stockage (raccourcisseur d'URL, sondages ...).

Etat	Fonctionnalité / Services	Nécessité	Machine	Besoin Puissance	Besoin Stockage	Besoin Fiabilité	Flux anticipé
Cloud	Nextcloud	5	D	1	5	4	3
Suite Bureautique	Collabora Online	3	C	4	0	3	4
	Onlyoffice						
	Etherpad						
Prise de notes	Hedgedoc	2	A	?	1	2	1
Git		4	C	4	2	5	2

FIGURE 20 – Extrait du cahier des charges

En se référant aux configurations minimales et recommandées pour un nombre estimé de 1000 utilisateurs nous avons pu estimer la configuration requise pour notre nouvelle machine en prenant en compte ce que nous possédions déjà grâce à l'inventaire produit précédemment.

### 9.1.3 Choix du serveur

Cette partie du projet se mêle aux responsabilités associatives et a donc surtout été réalisée en dehors des heures dédiées au projet. En effet, il a d'abord fallu obtenir un finan-

cement pour l'achat de la machine. Ce financement a été obtenu grâce à la participation de trois acteurs :

- Le *Forum Perspectives* (une association centralienne) a réalisé un appel à projet qui consistait à redistribuer l'argent généré par leur mandat. Grâce aux votes des Centraliens, l'association a pu obtenir 2 000 € après avoir monté un dossier.
- Le BDE, auprès de qui nous avons déposé un autre dossier nous a accordé une subvention de 5 000 €
- *CGI*, une entreprise de conseil en informatique a sponsorisé l'association à hauteur de 1 500 €

Un devis a ensuite été réalisé en travaillant avec les conseillers techniques d'une entreprise de vente de serveurs reconditionnés basés sur le cahier des charges établi auparavant. Nous avons aussi utilisé nos propres connaissances accumulées au fil des recherches de machines ([5, 6]) pour avoir un avis critique sur ces conseils. Le choix de matériel reconditionné par rapport à du matériel neuf s'est basé sur des critères de budget. En effet, l'achat de matériel neuf a du sens lorsque on a des besoins de fiabilité très importants et un budget conséquent ce qui n'était pas notre cas. Le serveur considéré à l'achat est présenté en annexe 2F.

## 9.2 Mise à niveau logiciel de la salle serveur

### 9.2.1 Mise à jour des systèmes d'exploitation

Lors de l'installation, *Debian* se configure avec une adresse IP attribuée par le DHCP. Il faut donc connecter physiquement la machine à un port du switch qui n'est pas tagué (il laisse passer tous les paquets) pour finir de télécharger le système. On choisit un RAID 5 pour un bon compromis entre stabilité et efficacité puis on demande à *Linux* de créer automatiquement les partitions du RAID (qui apparaît alors comme un unique disque). Après avoir saisi quelques informations comme le gateway et le nom de domaine *Debian* est (ré)installé. Tout cela a été fait après s'être formé à Linux ([26]).

### 9.2.2 Interface réseau

Il faut ensuite configurer les interfaces réseau. Une interface réseau est une interface logicielle vers le matériel réseau. Le noyau *Linux* distingue deux types d'interfaces réseau : physiques et virtuelles. L'interface réseau physique représente un périphérique matériel réseau réel tel qu'un port ethernet alors qu'une interface réseau virtuelle est construite comme une fusion ou autre opération sur des interfaces réseau physique.

Il faut d'abord charger le module du kernel permettant d'utiliser des VLAN. Pour ce faire on ajoute la ligne '8021q' au fichier `/etc/modules`.

La configuration des interfaces réseau *figure 21* a été intégralement repensée par notre groupe. Elle repose sur le concept de bond [12]. Un bond fusionne plusieurs interfaces réseau (eno1 et eno2 par exemple) et crée une interface virtuelle. Cette interface virtuelle à un débit plus important que les interfaces initiales individuellement car on réalise du load-balancing entre toutes les interfaces réseau avec le protocole LACP. On déclare ensuite le tag des VLAN ainsi que des paramètres de configuration des bond.

Une subtilité de la configuration est que l'on crée des alias de bond (bond.156 :1) afin de créer des interfaces virtuelles pour les container Docker. On peut faire un parallèle avec

la configuration réseau qui doit être faite sur les machines virtuelles.

```

1 auto lo
2
3 iface lo inet loopback
4
5 allow-hotplug eno1 eno2 # Autoriser le hotplug des cable ethernet
6
7 iface eno1 inet manual # On supprime l'interface réseau
8 iface eno2 inet manual
9
10 auto bond0 # déclaration du bond
11 iface bond0 inet manual
12     bond_slaves eno1 eno2 # liste des interfaces composant le bond
13     bond_mode 802.3ad    # mode 4 : augmente la bande passante et gère la tolérance de panne.
14     bond_downdelay 200   # temps en millisecondes pour qu'une interface soit détectée down
15     bond_updelay 200    # idem pour up
16     bond_lacp rate 1     # Définit le type d'intervalle entre chaque packet LACPDU pour le mode 802.3ad
17
18 auto bond0.156 # bond taguant les paquets de la LAN 156
19 iface bond0.156 inet static
20     address 156.xxx.xxx.xxx
21     netmask 255.255.xxx.xxx
22     gateway 156.xxx.xxx.xxx # le gateway s'obtient avec ip route avec _defaut via xxx.xx.xx.xxx dev eno_ ...
23
24 auto bond0.10 # bond taguant les paquets de la LAN 10
25 iface bond0.10 inet static
26     address 10.xxx.xxx.xxx
27     netmask 255.xxx.xxx.xxx
28
29 ### Container Docker n°1
30 auto bond0.156:1
31 iface bond0.156:1 inet static
32     address 156.xxx.xxx.xxx # IP du container Docker
33     netmask 255.255.xxx.xxx

```

FIGURE 21 – Exemple de configuration réseau (serveur de production)

### 9.2.3 Modification du reverse-proxy

Le choix du reverse-proxy pour la nouvelle infrastructure construite dans le cadre du Projet d'Étude a été longuement réfléchi. Les trois technologies ayant été considérées sont :

- NGINX, standard actuel (lourd et complexe). Reverse-proxy en vigueur au début du projet.
- Traefik, très adapté à des container Docker.
- Caddy, minimalistique et facile à prendre en main

Initialement, *NGINX* était utilisé à ÉCLAIR. Ce logiciel open-source a été conçu avec des objectifs de performance et d'optimisation. En contrepartie la configuration est très lourde. Il a été utilisé dans le cadre de notre projet jusqu'en début mai. Toutefois, *NGINX* a une configuration complexe (voir annexe 2H). Un autre problèmes que nous avons rencontrés avec *NGINX* est qu'il fallait renouveler manuellement les certificats SSL des sites web tous les trois mois avec *Certbot*. Par ailleurs, il fallait rajouter des lignes à la configuration pour augmenter le débit montant et descendant accordé à un container *docker*. *NGINX* fixe une limite par défaut ce qui alourdissait les fichiers de configurations.

*Traefik* [17] possède de nombreux avantages : il renouvelle automatiquement les certificats SSL (si on configurer *Certbot* et *Let's Encrypt*) et possède un dashboard web intégré. Par ailleurs il découvre automatiquement les container *Docker* ce qui permet de faire toute la configuration d'un service que l'on veut héberger dans le docker-compose. Le fonctionnement de *Traefik* n'est pas conventionnel (elle repose sur un système de provider, qui sont en réalité des API, que l'on indique dans le fichier de configuration et qui échangent avec *Traefik*) ce qui peut rendre la passation délicate. Un des problèmes que nous avions et que si on veut l'utiliser comme reverse proxy sur plusieurs serveurs *Docker* (un cluster) il faut configurer un orchestrateur docker ce qui complexifierait inutilement l'installation.

Le choix final s'est donc porté sur *Caddy* ([33]) pour sa simplicité, sa modularité (plugins open-source développés par la communauté de *Caddy*) et sa fonctionnalité permettant de renouveler automatiquement les certificats SSL sans avoir besoin de le configurer. En un sens, *Caddy* est simpliste mais se prête parfaitement aux besoins actuel d'ÉCLAIR. Il est de plus spécialisé dans la gestion des conteneurs Dockers [3]. Or il est déconseillé de conserver un outil inutilement complexe dont on n'exploite très peu les possibilités. L'annexe 2I montre les fichiers de configuration de *Caddy*, à savoir le docker-compose et le CaddyFile (celle de *NGINX* est disponible en annexe 2H). On remarque que la configuration de *Caddy* est largement simplifiée par rapport à *NGINX*.

#### 9.2.4 Mise à jour du pare-feu

En plus du pare-feu de la DSI situé en amont, ÉCLAIR possède son propre pare-feu. Cela permet d'éviter de propager à l'ensemble du réseau de l'école des problèmes qui proviendraient du réseau associatif, mais aussi de mieux isoler les installations de la DSI et d'ÉCLAIR d'une manière générale. Celui-ci est situé en sortie de S7, car ce switch sert pour redistribuer l'accès fibré au réseau fourni par la DSI à ÉCLAIR (*figure 22*).

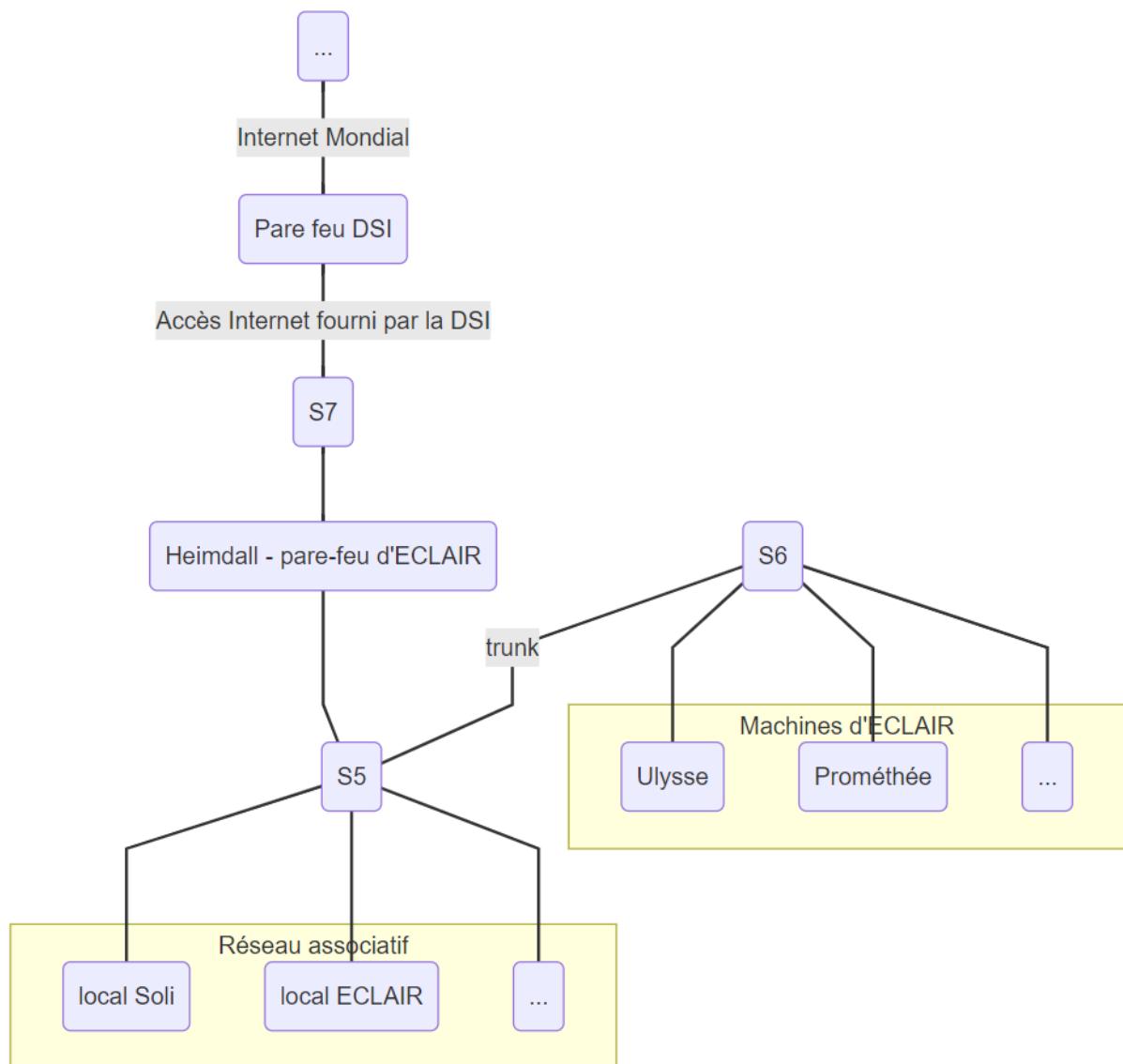


FIGURE 22 – Fonctionnement du pare-feu d’ÉCLAIR

ÉCLAIR utilise un pare-feu basé sur une distribution *pfsense*, dans son édition libre. Celui-ci a une documentation riche que nous avons beaucoup utilisée ([23]). Initialement le pare-feu d’ÉCLAIR présentait une configuration inutilement complexe, issue de la gestion du réseau internet de la résidence Comparat.

Provenance	Destination	Protocole (port)	Règle
Externe	Reverse-proxy	HTTPS (443)	autoriser
Externe	Reverse-proxy	HTTP (80)	autoriser
Externe	DNS	DNS (53)	autoriser
Interne	Externe	*	autoriser
Externe	Interne	*	interdire

TABLE 4 – Nouvelle configuration du pare feu

Nous avons alors simplifié cette configuration (table 4), ainsi que mis à jour *pfsense*

qui présentait trop de versions de retard : le système de mise à jour automatique était devenu lui-même obsolète. Nous en avons profité pour rajouter des descriptions plus précises et claires de ces règles. Les prochaines versions de *pfsense* s'installeront normalement automatiquement. En annexe 2J, le lecteur pourra trouver un tutoriel succinct sur l'utilisation de cet outil, en particulier sur le fonctionnement des règles. Nous pouvons remarquer que par défaut, toutes les connexions provenant de l'extérieur du réseau sont bloquées et toutes celle provenant de l'intérieur sont autorisées. On autorise ensuite uniquement les connexions de l'extérieur vers le reverse-proxy (ports 80 et 443) ainsi que vers le DNS (port 53).

### 9.2.5 Migration vers un nouvel hyperviseur

L'hyperviseur anciennement utilisé par ÉCLAIR, *QEMU* est un hyperviseur de type 2, sur une version qui était très vieillissante. Nous avons donc opté pour *Proxmox* [25], une solution open-source de type 1. Un hyperviseur de type 1 est un système qui s'installe directement sur la couche matérielle du serveur. On parle d'hyperviseur natif (*figure 23*). L'hyperviseur de type 2 qui est un logiciel qui s'installe et s'exécute sur un système d'exploitation déjà en place. On parle d'hyperviseur hébergé (*figure 24*). Nous avons choisi d'installer un hyperviseur type 1 car nous n'utilisions pas le système d'exploitation de l'hôte et avions besoin d'un seul hyperviseur sur cette machine. Les systèmes de type 1 sont allégés de manière à se « concentrer » sur la gestion des systèmes d'exploitation invités c'est-à-dire ceux utilisés par les machines virtuelles qu'ils contiennent. Ceci permet de libérer le plus de ressources possible pour les machines virtuelles. *Proxmox* est basé sur l'utilisation d'une interface web.

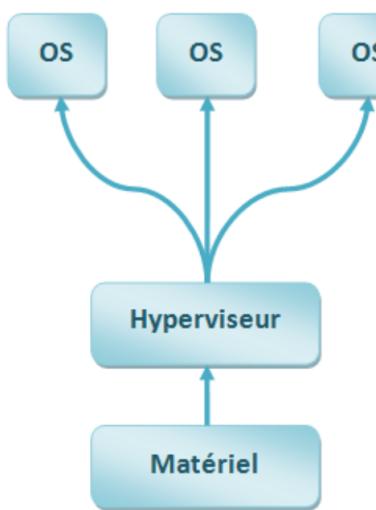


FIGURE 23 – Hyperviseur type 1  
(d'après IT-Connect)

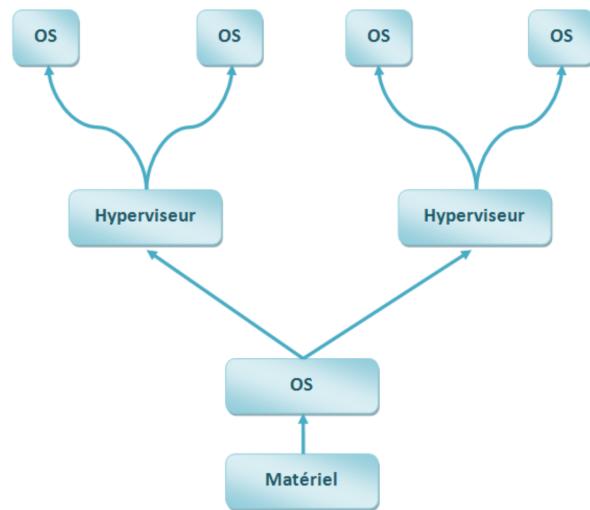


FIGURE 24 – Hyperviseur type 2  
(d'après IT-Connect)

Le réseau d'un hyperviseur est généralement plus complexe que celui de simples machines, comme l'hôte se charge souvent de faire le routage de ses machines virtuelles. Dans notre cas, chaque machine remplissant un rôle précis, elles nécessitent une adresse IP extérieure unique, par exemple pour les mettre dans le reverse-proxy. De plus, *Proxmox* nécessite des bridges sur lesquels créer des machines virtuelles. Tout ceci a grandement

compliqué la configuration réseau de l'hyperviseur et a donc beaucoup ralenti sa mise en place.

Une fois sa configuration réseau terminée, *Proxmox* rend très simple la gestion de ses machines virtuelles. Il est alors très rapide de créer, configurer et sauvegarder les machines dont on a besoin. Ces procédures sont décrites dans l'annexe 2G.

### 9.2.6 Mise en place de l'architecture SSH

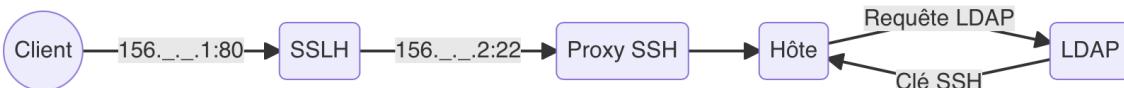


FIGURE 25 – Architecture SSH de l'association ÉCLAIR

Dans le cas d'ÉCLAIR la configuration SSH est assez avancée, et est composée de plusieurs briques. Lorsque le client souhaite accéder à l'un des serveurs, il doit d'abord passer par trois points : du SSLH, un proxy SSH, ainsi qu'un serveur LDAP.

Le serveur SSLH est la réponse à un problème propre à l'infrastructure d'ÉCLAIR : les seuls ports auxquels la DSI nous autorise à accéder depuis l'extérieur du réseau de l'École Centrale de Lyon sont les ports 80 et 443, qui permettent de faire du web (trames HTTP et HTTPS). Or, si les ports 80 et 443 sont déjà utilisés par du web, il est normalement impossible d'y faire passer aussi du SSH. C'est ici qu'entre en jeu le SSLH : c'est un multiplexeur de ports, qui permet de faire passer plusieurs protocoles par un seul port. La trame envoyée par le client commence donc par entrer dans le serveur SSLH par le port 80, et est redirigée sur le port 22 du proxy SSH.

Celui-ci remplit le rôle de « bastion » [28], qui est normalement renforcé et doit résister aux attaques de l'extérieur. Dans un cas général, il n'est pas trouvable simplement sur le réseau car il ne répond pas aux requêtes ICMP (les ping, qui permettent généralement de savoir si une machine est présente sur l'adresse IP qu'on teste). Ce bastion a pour mission de relier toutes les machines du réseau à l'extérieur. En effet, les serveurs sont tous sur un réseau privé, et ne sont normalement pas accessibles depuis l'extérieur du réseau de Centrale sans passer par (au moins) une machine qui filtre les informations et les personnes qui passent sur le réseau. Une fois que le bastion nous a authentifié (par clé SSH dans le cas d'ÉCLAIR), il nous renvoie finalement sur la machine hôte à laquelle on veut se connecter.

Lorsque la requête arrive sur cette machine, l'authentification fonctionne par clé, mais la clé n'est pas sur la machine hôte... En effet, cela impliquerait de copier la clé publique du client sur tous les hôtes, ce qui serait trop long et laborieux. C'est pourquoi un serveur LDAP a été mis en place. Ce serveur est en fait un annuaire qui stocke tous les utilisateurs avec leur clé SSH publique et les machines auxquelles ils ont accès. Une fois la configuration faite, c'est un script appelé par le fichier de configuration du daemon SSH qui se charge de contacter le serveur LDAP [1] et de récupérer la clé [31]. L'authentification est alors faite à l'aide de PAM [18] (Pluggable Authentication Modules) qui se chargera seul de

créer une session au client s'il ne s'est jamais connecté sur la machine. Le client est alors authentifié, et les opérations de connexion sont terminées.

Ce projet ayant pour objectif de recréer la documentation de l'association ÉCLAIR pour les prochains mandats, on pourra trouver en annexe toutes les instructions nécessaires à la configuration de l'architecture SSH :

- En annexe 1C on trouvera toutes les informations nécessaires à la compréhension et la mise en place de nombreuses configurations SSH.
- En annexe 2B on trouvera les instructions précises à suivre pour intégrer une nouvelle machine à l'architecture SSH d'ÉCLAIR.
- En annexe 1D on trouvera de même les informations permettant de comprendre le fonctionnement et la configuration d'un serveur LDAP.
- En annexe 2C on trouvera les instructions permettant de réinstaller de zéro la machine virtuelle "ldap", qui fournit le service LDAP.
- En annexe 1E on trouvera des bases sur la configuration du module PAM de *Linux*, ce fichier ne permet en revanche pas de remonter ou comprendre entièrement l'architecture, chose nécessitant une certaine expérience et une compréhension assez avancée du module PAM.
- En annexe 1F on trouvera les instructions nécessaires à l'installation et la configuration du service SSLH.
- Finalement, en annexe 2D on trouvera le fichier permettant de remettre en place entièrement la machine virtuelle "eole", qui fournit le service de proxy SSH.

### 9.2.7 Migration du DNS, NTP et DHCP

Une fois toutes les machines virtuelles recrées et repensées sur le nouvel hyperviseur *Proxmox*, il a fallu migrer le DNS, NTP et DHCP. Les logiciels utilisés sont ceux directement intégrés à Debian (respectivement bind9, ntpdate et isc-dhcp-server [16, 27, 10]). Ainsi, étant donnée que les configurations étaient basique mais ne nécessitait pas de refontes, nous avons simplement recopié les fichiers configurations et les avons déployé sur les nouvelles machines virtuelles.

### 9.2.8 Système d'authentification unique (SSO)

Les systèmes d'authentification unique (Single Sign-on ou SSO) ont pour objectif de simplifier l'accès et l'authentification à différents services.

Ce processus a pour avantage de simplifier grandement la gestion des utilisateurs. Ceux-ci ont un unique jeu d'identifiant et mot de passe, réduisant les risques d'hameçonnage (phishing en anglais) et de pertes d'accès, qui leur permet de se connecter à l'ensemble des services utilisant le SSO.

On distingue plusieurs catégories de SSO :

- Les SSO internes permettent de se connecter à un ensemble de services proposés par une institution. Nous pouvons citer dans cette catégorie le CAS (Système d'Authentification Centralisé) de l'école.
- Les SSO sociaux, permettent de se connecter à un service au moyen d'un service tiers. *Google* propose ainsi un service de connexion pour des plateformes tierces.

Dans le cadre d'ÉCLAIR, un système de connexion centralisé a tous son sens. En effet, l'association propose et souhaite proposer des services multiples aux élèves et aux personnels de l'école. Ce système d'authentification permettrait alors de simplifier la création et la gestion des comptes de ses utilisateurs tout en limitant les risques d'attaques informatiques.

Il existe différents protocoles [24] permettant de se connecter, ceux-ci peuvent être plus ou moins anciens et sont adaptés à des situations différentes. Actuellement l'utilisation des protocoles *OAuth 2.0* [11] et *Openid connect* [22] sont de plus en plus importants. Il s'agit des deux protocoles que nous avons décidé d'implémenter pour l'infrastructure d'ÉCLAIR.

On distingue, dans le processus d'authentification, les concepts suivants :

- Ressource : une information dont l'accès est restreint, par exemple le nom de l'utilisateur ou son adresse email.
- Serveur d'autorisation ou d'authentification : le serveur chargé d'autoriser ou authentifier les utilisateurs. Il s'agit du cœur du SSO.
- Client : serveur, logiciel, application à laquelle le client cherche à être connecté par le biais du serveur d'authentification.
- Authentifier : prouver que l'utilisateur est bien celui qu'il prétend être. Pour cela, le serveur demande généralement à l'utilisateur une preuve de son identité, sous la forme d'un mot de passe, d'un secret, ou d'une donnée biométrique tel une empreinte digitale.
- Autoriser : permettre à un utilisateur d'avoir accès à une ressource protégée.
- Token : donnée témoignant d'une autorisation ou d'une identité.
- Json Web Token (JWT) : format particulier de token.

Les spécifications du protocole *OAuth 2.0* commencent à être écrites en 2009, dans un effort de modernisation et d'unification des nombreux protocoles et outils d'authentification alors existants. Le protocole *Openid Connect* vise de son côté à étendre le protocole *OAuth 2.0*, en ajoutant notamment la possibilité d'obtenir des informations sur l'utilisateur et d'authentifier celui-ci de manière générale, et non d'autoriser son accès à une ressource.

Les spécifications de ces deux protocoles définissent plusieurs processus visant à autoriser ou authentifier un utilisateur. Le processus recommandé aujourd'hui est le protocole *Authorization code Grant*, il est adapté à l'authentification sur des services tiers (par exemple sur une instance Nextcloud) ainsi que l'autorisation à un service interne, depuis une page web et même de puis une application mobile, sous réserve d'utiliser l'extension *PKCE*.

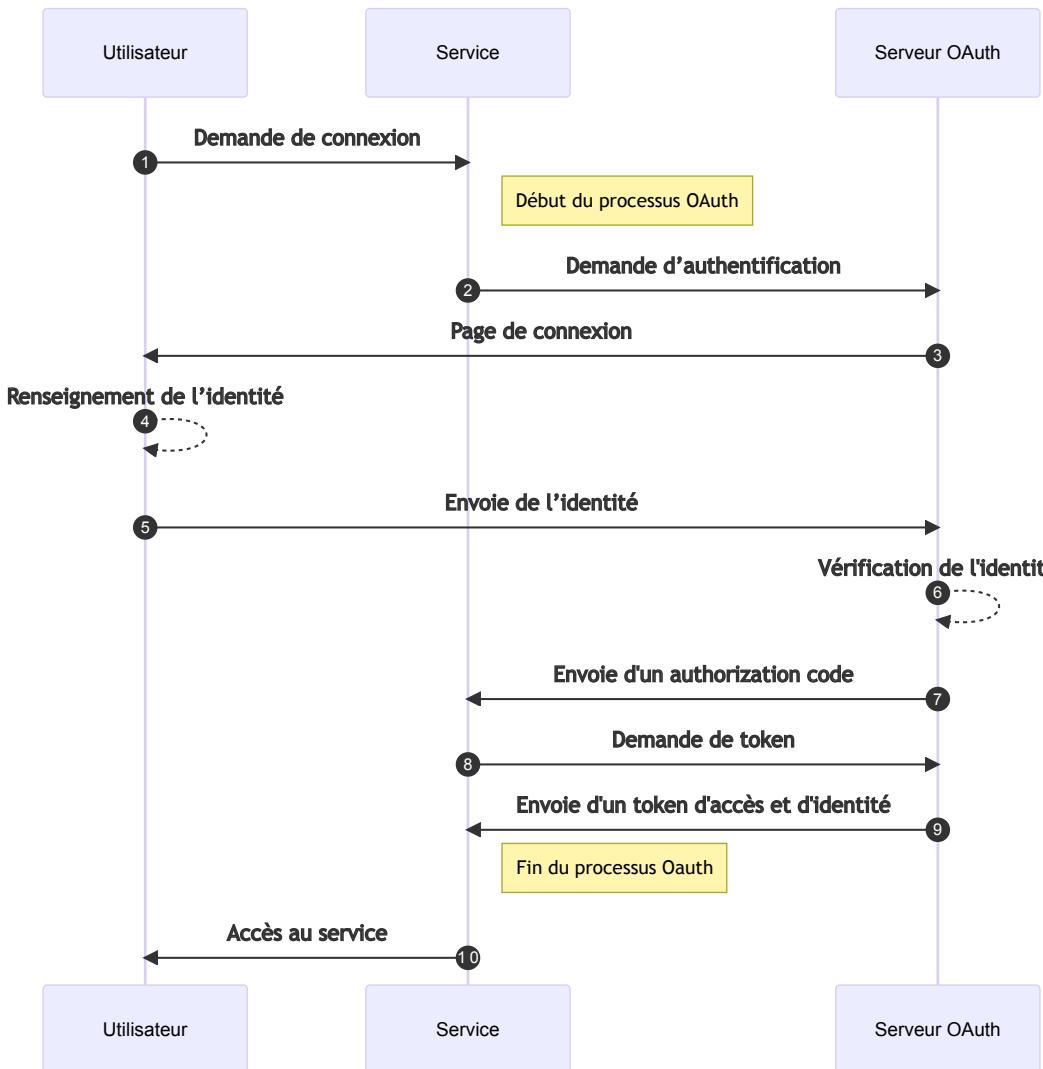


FIGURE 26 – Diagramme séquentiel d'une authentification utilisant le protocole OAuth 2.0

Le processus de connexion à un service, par exemple à l'instance Nextcloud proposée par ÉCLAIR, au moyen du SSO via le protocole *OAuth 2.0*, représenté dans le diagramme 26, ce déroulera comme suit :

1. L'utilisateur accède à la page de Nextcloud (le *service*) et demande à se connecter
2. Le service demande au serveur d'authentifier l'utilisateur
3. Celui-ci propose à l'utilisateur une page de connexion où il peut entrer une preuve de son identité
4. L'utilisateur renseigne les éléments demandés : son email et son mot de passe
5. Il envoie alors ces éléments au serveur d'authentification. Le service n'aura alors jamais accès à ces informations confidentielles et précieuses
6. Le serveur d'authentification vérifie l'email de l'utilisateur et le hash de son mot de passe
7. Le serveur d'authentification envoie un code d'autorisation au service
8. Le service demande un token d'accès et d'identité à l'utilisateur
9. Le service reçoit un token d'accès et d'identité du serveur d'authentification
10. Le processus OAuth est terminé
11. L'utilisateur accède au service

7. Il envoie ensuite un code d'autorisation, de durée de validité très courte au service
8. Le service peut alors échanger son code d'autorisation contre des tokens
9. Le serveur d'authentification peut envoyer un token d'autorisation ou d'identité
10. Le service peut alors valider l'identité de l'utilisateur et donc lui donner accès au service

Nous avons donc implémenté les spécifications *OAuth 2.0* et *Openid connect* du protocole l'*Authorization code grant* dans un serveur *Python*, afin que le SSO soit compatible avec les autres services d'ÉCLAIR.

### 9.2.9 L'outil de passation de connaissance : le wiki

Un des buts de notre Projet d'Étude est de fournir une documentation de l'infrastructure afin de limiter les problèmes de passation de connaissance d'un mandat à un autre. Le support de ce wiki est donc un choix important. Après un état de l'art des solutions de documentation les plus répandues, nous avons identifié plusieurs possibilités.

Nous avons défini le cahier des charges suivant :

- Nous devons avoir une copie du site ou de ses sources accessible en cas de panne des serveurs
- Un système d'authentification est nécessaire, le wiki contenant des données sensibles
- L'utilisation de

À ce titre, une instance *MediaWiki* [21], l'outil utilisé par ÉCLAIR par le passé, ne correspond pas au cahier des charges. En dehors de considérations esthétiques, les données stockées sous forme d'une base de données rendant les sauvegardes compliquées.

Le *Markdown* est un langage de balisage, léger, pratique et facilement extensible. Il est donc tout indiqué pour la rédaction de documents techniques, tout en étant très facile à convertir en page web.

Il a alors été décidé d'opter pour le *Docusaurus* [7] (*figure 27*) et de le connecter au SSO que nous avons développé (partie 9.2.8).

Docusaurus est un projet open source permettant de réaliser des sites de documentation et qui repose sur le langage Markdown. Des exemples de wiki créés avec Docusaurus sont disponibles sur leur page de présentation : <https://docusaurus.io/showcase>.



FIGURE 27 – Logo de docusaurus

## 9.3 Responsabilité sociétale et environnementale

### 9.3.1 Impact environnemental

**Bilan carbone** Le Bilan Carbone est un outil de diagnostic conçu pour comprendre et analyser l'activité d'une entité en matière d'émissions directes et indirectes de gaz à effet de serre. L'objectif général est de trouver un moyen d'économiser de l'énergie et de réduire au maximum l'émission de gaz carbonique.

Pour faire un bilan carbone, il faut calculer les émissions induites ou indirectes ce qui constitue l'exercice le plus délicat. Par souci de facilité nous avons uniquement comptabilisé la consommation moyenne électrique sur une semaine mais n'avons pas pris en compte la pollution générée par la création des machines.

Dans notre cas, après avoir relevé les puissances que consomment les différentes machines (grâce aux onduleurs) et effectué le bilan carbone de la salle des serveurs d'ECLAIR, il ressort que celle-ci émet environ 191 g de carbone par heure.

Les différentes valeurs des puissances relevées sont contenues dans le tableau ci-dessous :

Machines	Puissances en kWh
Serveurs + switchs	1,1
Climatisation	0,68
Onduleurs	0,10847
Disques durs	0,0194
Total	1,90787

Avec en France :  $0,1 \text{ kg de } CO_2 \approx 1 \text{ kWh}$ .

Le diagramme circulaire *figure 28* permet de visualiser les proportions d'émission de chaque machine de la salle :

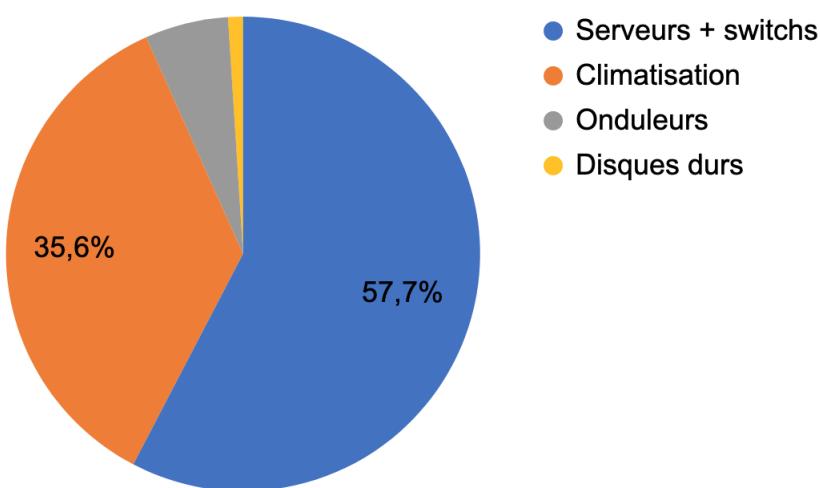


FIGURE 28 – Bilan Carbone de la salle des serveurs d'ECLAIR

Ce bilan a permis d'évaluer l'impact de nos machines en matière d'émission des gaz à effet de serre dans l'environnement afin de pouvoir mettre en place un plan de réduction de ces émissions.

**Pistes d'amélioration :** Il est relativement complexe de réduire la consommation énergétique d'une salle serveur. On travaille généralement à optimiser les logiciels en production dessus. Dans notre cas, le rack possède un nombre important de machines. Certaines sont sous-exploitées et possèdent une puissance de calcul très faible (certaines machines ont 4Go de RAM). Pourtant, le système d'exploitation de ces machines consomme beaucoup d'énergie inutilement. Il a été convenu de centraliser tous les petits services sur une seule machine à laquelle on a ajouté de la RAM, ce qui a permis de retirer deux machines. On diminue ainsi l'empreinte carbone. Il a ensuite fallu recycler ces machines, pour cela nous avons contacté des professionnels basés à Lyon.

### 9.3.2 Règlement général sur la protection des données

Le RGPD, Règlement Général sur la Protection des Données sert à encadrer le traitement et l'utilisation des données personnelles, données permettant d'identifier directement ou indirectement une personne, collectées par les entreprises ou les administrations de l'Union Européenne. En général, le RGPD a pour but de renforcer la protection des données personnelles des européens et aussi de responsabiliser les organismes publiques et privées.

En tant que membres du projet d'étude et de mise à niveau de la salle serveur d'ECLAIR, nous travaillons sur des machines qui abritent des données appartenant à des élèves ou des associations de l'École Centrale, nous devons donc d'être en conformité avec le RGPD afin d'assurer la sécurité des informations personnelles détenues. La mise en conformité RGPD se fait en différentes étapes [19]

- En premier lieu, il faut faire une liste exhaustive de toutes les données collectées, la population concernée et le moyen de récupération de ces données. Aussi, dans le cas de collecte de données sensibles (opinion politique, données concernant la santé...) nous ne sommes pas autorisés à les traiter sans le consentement des personnes concernées ; il faut donc se poser la question de savoir les données sont sensibles ou pas.
- Deuxièmement, les données doivent être collectées pour atteindre un objectif fixé posé sur un fondement légal comme l'énonce le principe de minimisation : ce principe prévoit que les données à caractère personnel doivent être adéquates, pertinentes et limitées à ce qui est nécessaire au regard des finalités pour lesquelles elles sont traitées. Les informations doivent être conservées sur une durée prédéfinie liée à l'objectif ; une fois cette durée atteinte, les informations doivent être supprimées ou indemnisées.
- Il est aussi important d'être capable de supprimer les données des clients s'ils le demandent et de signaler aux autorités en cas de piratage ou de violation des données.

Avec le déclin de la salle serveur, les services hébergés dessus ont progressivement diminués. Aujourd'hui, la partie en production consiste seulement en des sites statiques et un cloud de stockage pour l'association audiovisuelle *Pixels*. Ainsi, pour l'instant le respect de la RGPD est évident.

De plus, le wiki possède maintenant une check-list qui vérifie que chaque service mis en ligne respecte le RGPD.

## 10 Prolongement du projet

Ce projet d'étude a permis d'identifier les problèmes et failles majeures de l'infrastructure d'ÉCLAIR. Si certaines solutions ont été mises en place, d'autres n'ont pas pu l'être. Si certaines tâches n'ont pas pu être mises en place, d'autres n'ont pas été mises en place car il s'agit de projets plus conséquents qui dépassent le cadre du PE. Ces tâches, développées ci-dessous, ne rentraient pas dans les objectifs du PE, qui consistait surtout en la détection de problèmes et dans la proposition de solutions, mais pas nécessairement dans leur implémentation. Elles feront l'objet d'une itération du projet d'étude dans l'année à venir.

### 10.1 Stratégie de sauvegarde des données

La stratégie de sauvegarde (plus couramment appelée *backup plan*) est le point le plus important que nous n'avons pas pu traiter. En effet, il était prévu de mettre en place des sauvegardes régulières et redondantes. Cependant jusqu'en mars environ nous manquions de recul sur les fichiers clés à sauvegarder afin de pouvoir restaurer les machines en cas de panne. Un plan de sauvegarde sans plan de restauration est absolument inutile. Nous nous sommes donc contenté de faire bêtement des images disques de nos serveurs sur des disques durs externes. Cette solution n'est pas satisfaisante car elle est inutilement coûteuse en stockage, risquée et les sauvegardes sont difficiles à exploiter.

Aujourd'hui, nous avons acquis ce recul, ce qui nous a permis d'identifier les données sensibles qui nécessitent une copie. Parmi ces données on distingue les données légères et les données lourdes.

#### 10.1.1 Données dites légères (règle de sauvegarde 3-2-1)

On qualifiera les données de "données légères" lorsqu'on a affaire à des fichiers de configuration ou de base de données ayant de tailles inférieures à quelques mégaoctets.

Les données légères à sauvegarder sur les serveurs sont :

- /etc (répertoire des fichiers de configurations). Plus précisément :
  - /etc/network/interfaces : configuration réseau
  - /etc/resolv.conf
  - /etc/bind/named.conf
  - /etc/bind/named.conf.options
  - /etc/bind/named.conf.local
  - /var/cache/bind
- /home/Docker (pour le serveur de production) : dossier où l'on place les docker-compose et où l'on monte les volumes avec des données importantes (les volumes non montés sont dans le système Docker soit dans /var)

Un standard lorsque l'on parle de sauvegarde est la règle 3-2-1. Elle représente un cadre idéal de sécurisation des données (même si aucune stratégie de sauvegarde n'est parfaite). La règle de sauvegarde 3-2-1 préconise de :

- Disposer de 3 copies de ses données. On peut garder le fichier principal sur serveur et le copier en 2 exemplaires sur des périphériques différents. Ainsi, plus on fait de copies des données, moins on prend le risque de tout perdre.

- Sauvegarder sur 2 supports différents. Si l'un des supports de sauvegarde est défaillant, les données sont sauvées grâce au deuxième support.
- Conserver 1 copie de sauvegarde hors-site (localisation géographique différente)

L'idéal est de stocker sa sauvegarde principale en « local » (dans nos locaux) et d'en posséder une copie dans le Cloud (en mode externalisée).

Ces préconisations minimisent le risque de perte de données et s'inscrit dans une politique de sauvegarde fiable et sûre.

### 10.1.2 Données dites lourdes

On qualifiera les données de "lourdes" lorsque l'on parle de répertoire qui sont de l'ordre du téra octet (classiquement des photos où des vidéos). Il est clair qu'appliquer la règle 3-2-1 est très onéreux dans ce cas. En effet, stocker une grande quantité de données dans le Cloud coûte très cher. L'idéal serait d'avoir une copie sur un serveur NAS en dehors du M16 (à la DSI par exemple) en plus de bénéficier de la garantie offerte par le RAID.

## 10.2 Stratégie de restauration des données

Le DRP, “Disaster Recovery Plan” (ou Plan de reprise d'activité ou Plan de continuité d'activité) permet, en complément de la solution de sauvegarde, d'assurer la reconstitution de l'infrastructure informatique et donc la remise en route de l'activité de l'association. Il est aussi crucial que le plan de sauvegarde.

## 10.3 Monitoring

Le monitoring ou la supervision informatique permet d'analyser en temps réel l'état du système informatique et l'état du réseau informatique à des fins préventives. Il permet d'alerter en cas de dysfonctionnement des systèmes d'information et de pouvoir ainsi agir le plus rapidement possible afin d'éviter une indisponibilité des systèmes. Le monitoring réseau ou le monitoring de serveurs est important dans de nombreux aspects des ressources informatiques d'une entreprise. On parle dans la suite de ce paragraphe de White-box monitoring c'est à dire qu'on a accès à toutes les informations de l'infrastructure.

ÉCLAIR ne dispose plus actuellement de système de monitoring. Nous avons identifié deux étapes nécessaires pour déployer une surveillance active du système, ceci suite à une formation de 3h qui nous a été dispensé par une entreprise : *Padok*. D'abord il faut choisir un outil, les deux que nous avons étudié sont :

- Approche cloud native : Prometheus (base de données et analyse des logs) + Node-Exporter + Grafana (Dashboard)
- Zabbix (logiciel libre permettant de surveiller l'état de divers services réseau, serveurs et autres matériels réseau et produisant des graphiques dynamiques de consommation des ressources anciennement utilisé par ÉCLAIR)

La deuxième partie, plus complexe, consiste à choisir les métriques qui sont adaptés à la surveillance de l'infrastructure d'ÉCLAIR. En d'autre termes quelles informations ont

un sens et comment poser des critères et des limites sur ces grandeurs. *Google* à récemment publié quatre "golden metrics" [8] qu'il faut idéalement parvenir à mesurer :

- Latence : c'est le temps qu'il faut pour répondre à une requête. Il est important de faire la distinction entre la latence des requêtes réussies et de celles qui échouent. Par exemple, une erreur HTTP 500 déclenchée en raison d'une perte de connexion à une base de données ou à un autre backend peut être traitée très rapidement ; cependant, comme une erreur HTTP 500 indique l'échec d'une demande, la prise en compte des 500 dans la latence globale peut donner lieu à de mauvaises interprétations. D'autre part, une erreur lente est pire qu'une erreur rapide ! Il est donc important de mesurer la latence des erreurs, plutôt que de se contenter de filtrer les erreurs.
- Trafic : une mesure de la demande qui pèse sur le système (dépend du nombre d'utilisateur). Pour un service Web, cette mesure correspond généralement aux requêtes HTTP par seconde, éventuellement pondérés selon la nature des requêtes (par exemple, contenu statique ou dynamique). Pour un système de streaming audio, cette mesure peut porter les sessions simultanées. Pour un système de stockage cette mesure peut être les envois et les réceptions par seconde.
- Erreurs : Taux de requêtes qui échouent, soit explicitement (par exemple, HTTP 500), soit implicitement (par exemple, une réponse HTTP 200 positive, mais associée à un contenu erroné), soit à cause de contraintes que l'on s'impose. Lorsque les codes de réponse des protocoles sont insuffisants pour exprimer toutes les conditions d'échec, des protocoles secondaires (internes) peuvent être nécessaires pour suivre les modes d'échec partiels.
- Saturation : Le niveau de "remplissage" d'un système. Il s'agit d'une mesure de la fraction du système, qui met l'accent sur les ressources les plus limitées (par exemple, dans un système où la mémoire est limitée, il faut afficher la mémoire). De nombreux systèmes voient leurs performances se dégrader avant d'atteindre une utilisation de 100%, il est donc essentiel d'avoir un objectif d'utilisation. Dans les systèmes complexes, la saturation peut être complétée par une mesure de charge de plus haut niveau.

Pour obtenir des métriques de qualité il faut progressivement affiner ce que l'on mesure et que l'on affiche. Le choix de métriques adéquat ne peut se faire qu'avec de l'expérience au fur et à mesure des problèmes rencontrés. Il fera l'objet d'une des missions de la prochaine itération du projet.

## 11 Conclusion

Pour conclure, ce projet d'étude est né d'une volonté de l'association ÉCLAIR de reprendre le contrôle de son infrastructure serveur. Ainsi, en se formant pour comprendre son fonctionnement il a été possible de la reconstruire presque entièrement en documentant tout le processus.

Actuellement, les serveurs sont fonctionnels et en production, on peut donc considérer que le projet a été réussi. La documentation que nous avons produite permet en théorie à un débutant de reproduire le travail effectué dans la dernière phase de notre projet en quelques semaines. Elle permet aussi d'utiliser les serveurs d'ÉCLAIR en tant que PaaS (Platform As A Service), par exemple pour l'association audiovisuelle de centrale *Pixels*.

Toutefois, il reste encore un travail important à faire en matière de sécurité, de sauvegarde et de facilitation d'utilisation de l'infrastructure. Ceci laisse envisager des perspectives d'évolution qui feront l'objet d'une prochaine itération de notre projet.

## 12 Bibliographie

### Références

- [1] Karim BUZDAR. *How to configure LDAP client in Debian 10.* <https://linuxhint.com/configure-ldap-client-debian/>. 2021.
- [2] René CHALON. *Une introduction aux réseaux informatiques.* diaporama. 2021.
- [3] Tyler CHARBONEAU. <https://www.docker.com/blog/deploying-web-applications-quicker-and-easier-with-caddy-2/>.
- [4] CLOUDFLARE. *DNS server types.* <https://www.cloudflare.com/learning/dns/dns-server-types/>. 2022.
- [5] DELL. <https://www.dell.com/fr-fr>.
- [6] Western DIGITAL. <https://www.westerndigital.com/>.
- [7] DOCUSAURUS. *Docusaurus.* <https://docusaurus.io/>.
- [8] Rob EWASCHUK. *Monitoring Distributed Systems.* <https://sre.google/sre-book/monitoring-distributed-systems/>. 2017.
- [9] Free Software FOUNDATION. "Qu'est-ce que le logiciel libre ?" <https://www.gnu.org/philosophy/free-sw.fr.html>.
- [10] Jean Pierre GIRAUD. *DHCP Server.* [https://wiki.debian.org/fr/DHCP\\_Server](https://wiki.debian.org/fr/DHCP_Server). 2013.
- [11] Dick HARDT. *The OAuth 2.0 Authorization Framework.* RFC 6749. 2012. DOI : 10.17487/RFC6749. URL : <https://www.rfc-editor.org/info/rfc6749>.
- [12] Ian HOEFFNER. *Bonding.* <https://wiki.debian.org/Bonding>. 2008.
- [13] HP. *Aruba 2540 Switches : Installation and Getting Started Guide.*
- [14] IONOS. *Qu'est-ce que la solution RAID.* <https://www.ionos.fr/digitalguide/serveur/securite/raid/>.
- [15] ITDVDS. *Understanding How DNS Works in Depth.* <https://www.youtube.com/watch?v=T-eghY-9WdE>. 2017.
- [16] Franck JONCOURT. *Bind9.* <https://wiki.debian.org/fr/Bind9>. 2008.
- [17] traefik LABS. *Traefik Proxy Documentation.* <https://doc.traefik.io/traefik/>.
- [18] Denis LAXALDE. *LDAP PAM.* <https://wiki.debian.org/fr/LDAP/PAM>. 2009.
- [19] *Le règlement européen sur la protection des données.* <https://www.cnil.fr/fr/reglement-europeen-protection-donnees>. 2018.
- [20] *List of DNS record types.* [https://en.wikipedia.org/wiki/List\\_of\\_DNS\\_record\\_types](https://en.wikipedia.org/wiki/List_of_DNS_record_types).
- [21] MEDIAWIKI. *MediaWiki.* <https://www.mediawiki.org/wiki/MediaWiki>.
- [22] et al. N. SAKIMURA. *OpenID Connect Core 1.0.* [https://openid.net/specs/openid-connect-core-1\\_0.html](https://openid.net/specs/openid-connect-core-1_0.html).
- [23] NETGATE. *Basic Firewall Configuration Example.* <https://docs.netgate.com/pfsense/en/latest/recipes/example-basic-configuration.html>.

- [24] Aaron PARECKI. *OAuth 2.0 Simplified*. <https://www.oauth.com/>.
- [25] *Proxmox wiki main page*. [https://pve.proxmox.com/wiki/Main\\_Page](https://pve.proxmox.com/wiki/Main_Page). 2022.
- [26] ROOT-ME. "Une plateforme rapide, accessible et réaliste pour tester vos compétences en hacking". <https://www.root-me.org/>.
- [27] David SINQUIN. *NTP*. <https://wiki.debian.org/fr/NTP?action=info>. 2015.
- [28] Carl TASHIAN. *DIY SSH Bastion Host*. <https://smallstep.com/blog/diy-ssh-bastion-host/>. 2020.
- [29] T. TERASHINI. *Tera Term Help*.
- [30] DNS Made Easy VIDEOS. *DNS Explained*. [www.youtube.com/watch?v=72snZctFFtA](https://www.youtube.com/watch?v=72snZctFFtA). 2012.
- [31] WARLORD. *SSH Authorizedkeys and LDAP*. [https://warlord0blog.wordpress.com/2020/05/16/ssh-authorized\\_keys-and-ldap/](https://warlord0blog.wordpress.com/2020/05/16/ssh-authorized_keys-and-ldap/). 2020.
- [32] Laurenz WISKOTT. *SSH*. <https://wiki.debian.org/fr/SSH>. 2007.
- [33] ZEROSSL. *Caddy Documentation*. <https://caddyserver.com/docs/>.

## 13 Annexes

### 13.1 Annexes théoriques

L'ensemble des annexes ont été constitué par notre PE en se basant sur des sources diverses (indiquées dans la bibliographie). Ce travail de compréhension de la théorie des réseaux informatiques nous a occupé pendant pratiquement les quatre premiers mois du projet.

# Annexe 1A

## [Mémento] Linux

---

### [Mémento] Linux

Guide de survie :

Raccourcis clavier :

RTFM :

Navigation :

Gestion des fichiers :

Recherche de fichiers :

Manipulation de fichiers :

Extraire, trier et filtrer :

Flux de redirection :

Archiver et compresser :

Utilisateurs et droits :

Coté serveurs :

SSH :

Surveiller l'activité du système :

Installer des programmes :

Exécuter des programmes en arrière-plan :

Exécuter un programme à une heure différée :

Transférer des fichiers :

Analyser le réseau et filtrer le trafic :

Gérer les services :

Annexe :

Raccourci `top` :

Raccourci `less` :

Regex :

Ajouter et supprimer des règles :

Pratiquer !

---

## Guide de survie :

### Raccourcis clavier :

- `Ctrl + A` : ramène le curseur au début de la ligne.
- `Ctrl + E` : ramène le curseur à la fin de la ligne.
- `Ctrl + U` : supprime ce qui se trouve à gauche du curseur.
- `Ctrl + K` : supprime ce qui se trouve à droite du curseur.
- `Ctrl + Y` : colle ce que viens d'être supprimé.
- `Ctrl + C` : arrête un processus lancé en console.
- `Ctrl + Z` : met en pause l'exécution du programme.
- `Ctrl + D` : équivalent de exit/logout.
- `Ctrl + L` : nettoie l'écran.

## RTFM :

- `man` : ouvre le manuel (relatif à une commande).
- `-h` ou `--help` : aide succincte relativ à la commande.
- `apropos` : cherche les mots clés dans toutes les pages du manuel et affiche la liste des commandes concernées.

## Navigation :

---

- `pwd` : affiche le path du répertoire courant.
- `cd` : change le répertoire courant.

## Gestion des fichiers :

---

### Recherche de fichiers :

- `which` : renvoie l'emplacement de la commande.
- `ls` : liste les éléments du repertoire courant.
  - `-a` : affiche les fichiers et dossiers cachés.
  - `-l` : liste détaillée.
  - `-h` : affiche la taille.
  - `-F` : indique le type d'élément.
  - `-t` : trie par date de dernière modification.
- `du` : taille occupée par les dossiers.
  - `-a` : affiche la taille des dossiers et des fichiers.
  - `-h` : affiche la taille (human readable).
- `locate` : recherche rapide (necessite une maj de la base de donnée).
- `find` : recherche approfondie.
- [Méthode avancées](#)

### Manipulation de fichiers :

- `cat` : affiche tout le fichier.
- `less` : affiche le fichier page par page (il existe plusieurs [raccourcis](#) qui facilite la navigation).
- `head` & `tail` : affiche le début et la fin d'un fichier (`tail -f` pour afficher les modifications en temps réel).
- `touch` : crée un fichier.
- `mkdir` : crée un dossier.
- `cp` : copie un fichier (`-R` pour les dossiers).
- `mv` : déplace/renomme un fichier.
- `rm` : supprime un fichier.
- `ln` : crée un lien physique (`ln -s` crée un lien symbolique).

## Extraire, trier et filtrer :

- `grep` : filtre des données.
  - `-i` : ignore la casse.
  - `-n` : indique le numero de la ligne.
  - `-v` : inverser la recherche (ignorer un mot).
  - Utilisation possible d'[expressions régulières](#).
- `sort` : trie les lignes.
  - `-o` : écris dans un fichier.
  - `-r` : trie en ordre inverse.
  - `-n` : trie des nombres.
- `wc` : compte le nombre de lignes-mots-octets.
- `uniq` : supprime les doublons (`-c` compte le nombre d'occurrences).
- `cut` : coupe une partie du fichier (`-d` indique le délimiteur et `-f` le champ à conserver).

## Flux de redirection :

- `>` : redirige une le résultat d'une commande dans un nouveau fichier.
- `>>` : idem à la fin d'un fichier déjà existant.
- `2>` : redirige les messages d'erreurs (`2>&1` mis à la fin de la commande redirige tout)
- `<` : lis depuis un fichier.
- `<<` : lis un fichier progressivement (appuyer sur entrée à chaque ligne).
- `|` : chaîne les commandes.

## Archiver et compresser :

- `tar` : assemble des fichiers dans une archive (utiliser `-cvf`).
  - `-c` : crée une archive (.tar).
  - `-v` : affiche le détail des opérations.
  - `-f` : assemble l'archive dans un fichier.
  - `-tf` : affiche le contenu de l'archive sans l'extraire.
  - `-rvf` : ajoute un fichier.
  - `-xvf` : extrait les fichiers de l'archive.
- `gzip` : compresse avec LZ77 (s'utilise sur une archive).
- `bzip2` : compresse avec Burrows-Wheeler (s'utilise sur une archive).
- `zcat` : affiche directement un fichier compressé.
- `unzip` & `unrar` : à installer (decompresse les .zip et .rar).

## Utilisateurs et droits :

---

- `sudo` : exécute une commande en root.
- `chmod` : modifie les droits d'accès.

Droit	Chiffre
r (read)	4
w (write)	2
x (execute)	1

## Côté serveurs :

### SSH :

- `sudo /etc/init.d/ssh start` : lance le serveur (`stop` pour l'arrêter).
- `/etc/ssh/sshd_config` : fichier de configuration (`sudo /etc/init.d/ssh reload` pour que le changement soit pris en compte).
- `ssh` : établit la connection avec le serveur.
  - `-p` pour préciser le port.
  - `-i` pour spécifier une clé privée.
- Authentification par clé :
  1. `ssh-keygen -t rsa` : génère une paire de clés publique/privée.
  2. `ssh-copy-id -i id_rsa.pub login@ip` : La clé est automatiquement ajoutée à `~/.ssh/authorized_keys`.
  3. se connecter !
 

*Remarque* : pour générer une passphrase sécurisée utiliser `openssl rand -hex 12`
  4. `ssh-add` : autorise à ne taper la passphrase qu'une seule fois.

## Surveiller l'activité du système :

- `w` : informations générale sur l'état du système (uptime, charge, liste des connectés, ...).
- `lshw` : affiche des informations sur l'ensemble du hardware de la machine (à installer)
- `tload` : affiche le graphe de l'évolution de la charge.
- `ps` : liste statique des processus lancé depuis la console (non actualisée en temps réel).
  - `-ef` : liste tous les processus.
  - `-ejH` : affiche les processus en arbre.
  - `-u UTILISATEUR` : liste les processus lancés par un utilisateur.
- `top` : liste dynamique des processus (comme pour `less` la [navigation](#) se fait avec le clavier).
- `kill` : tue un processus (PID en argument).
- `killall` : tue plusieurs processus (commande en argument).
- `halt` : arrête l'ordinateur (dérive de `shutdown`).
- `reboot` : redémarre l'ordinateur (dérive de `shutdown`).

## Installer des programmes :

- `apt-get update` : met à jour le cache des paquets.
- `apt-cache search` : recherche un paquet.
- `apt-get install` : installe un paquet.
- `apt-get autoremove` : supprime un paquet (rajouter `autoremove` pour supprimer les dépendances).
- `apt-get upgrade` : met à jour tous les paquets.

## Exécuter des programmes en arrière-plan :

- `&` : lance un processus en arrière-plan (à mettre en fin de commande).
- `nohup` : détache le processus de la console.
- `bg` : passe le processus en tache de fond (à utiliser après un `Ctrl + Z`).
- `fg` : reprend un processus au premier plan.
- `jobs` : affiche les processus en tache de fond.
- `screen` : A venir (à installer).

## Exécuter un programme à une heure différée :

- `at` : exécute une commande à une heure précise (lire `man date` pour le format de la date).
- `atq` : liste les *jobs* en attente.
- `atrm` : supprime un *job* en attente.
- `sleep` : met en pause le système.
- `crontab` : exécute une commande régulièrement (penser à rediriger la sortie).
  - `-e` : modifie la crontab.
  - `-l` : affiche la crontab actuelle.
  - `-r` : supprime la crontab.

## Transférer des fichiers :

- `wget` : télécharge des fichiers depuis l'adresse HTTP ou FTP (`-c` pour reprendre un téléchargement en cours).
- `scp` : copie des fichiers sur le réseau (d'un ordinateur à un autre).
- `ftp` : connecte à un serveur FTP.
  - `put` : envoie un fichier vers le serveur.
  - `get` : télécharge un fichier depuis le serveur.
- `sftp` : un FTP sécurisé.
- `rsync` : synchronise des fichiers pour une sauvegarde (mettre `-av` en paramètre). Pour une sauvegarde plus complète cf `man` (ex : `rsync -av --delete --backup -- backup-dir=...)`

## Analyser le réseau et filtrer le trafic :

- `host` : convertit une IP en nom d'hôte et inversement.
- `whois` : tout savoir sur un nom de domaine.
- `ifconfig` : liste des interfaces réseau.
- `netstat` : affiche des statistiques sur le réseau.
  - `-i` : statistiques des interfaces réseau.
  - `-uta` : liste toutes les connexions ouvertes.
  - `-lt` : liste les connexions en état d'écoute.
  - `-s` : statistiques résumées.
- `iptables` : configuration du pare-feu ([quelques commandes utiles](#)).

## Gérer les services :

- `systemctl` : gère le démarrage et l'arrêt des processus (unité de type service).
  - `start` : lance un service (`stop` pour l'arrêter).
  - `restart` : `stop` puis `start`.
  - `reload` : met à jour la configuration.
  - `status` : renvoie l'état du service.
  - `enable` : lance le service au démarrage (`enable` pour ne pas le lancer).
  - `list-units` : liste toutes les unités actives (`--type=service` affiche uniquement les unités de type "service").
- `journalctl` : affiche les logs générés par `systemd`.
  - `-b` : depuis le dernier démarrage du service.
  - `-u` : pour un service uniquement.

---

## Annexe :

### Raccourci `top` :

- q : ferme top
- h : affiche l'aide, et donc la liste des touches utilisables.
- B : met en gras certains éléments.
- f : ajoute ou supprime des colonnes dans la liste.
- F : change la colonne selon laquelle les processus sont triés.
- u : filtre en fonction de l'utilisateur désiré.
- k : tue un processus.
- s : change l'intervalle de temps entre chaque rafraîchissement de la liste (par défaut: toutes les trois secondes).

## Raccourci less :

- `d` : affiche les onze lignes suivantes (soit une moitié d'écran).
- `b` : retourne en arrière d'un écran.
- `y` : retourne d'une ligne en arrière.
- `u` : retourne en arrière d'une moitié d'écran (onze lignes).
- `q` : arrête la lecture du fichier. Cela met fin à la commande less.
- `=` : indique où vous en êtes dans le fichier (numéro des lignes affichées et pourcentage).
- `h` : affiche l'aide
- `/` : tapez / suivi du texte que vous recherchez pour lancer le mode recherche.
- `n` : touche « Résultat suivant ».
- `N` : pareil que n, mais pour revenir en arrière.

## Regex :

[Mémento des expressions régulières](#)

## Ajouter et supprimer des règles :

- `-A chain` : ajoute une règle en fin de liste pour la `chain` indiquée (`INPUT` ou ``OUTPUT`, par exemple).
- `-D chain rulenumber` : supprime la règle n° `rulenumber` pour la `chain` indiquée.
- `-I chain rulenumber` : insère une règle au milieu de la liste à la position indiquée par `rulenumber`. Si la position `rulenumber` n'est pas indiquée, la règle sera insérée en premier, tout en haut dans la liste.
- `-R chain rulenumber` : remplace la règle n° `rulenumber` dans la `chain` indiquée.
- `-L` : liste les règles.
- `-F chain` : vide toutes les règles de la `chain` indiquée. Cela revient à supprimer toutes les règles une par une pour cette `chain`.
- `-P chain regle` : modifie la règle par défaut pour la `chain`.

## Pratiquer !

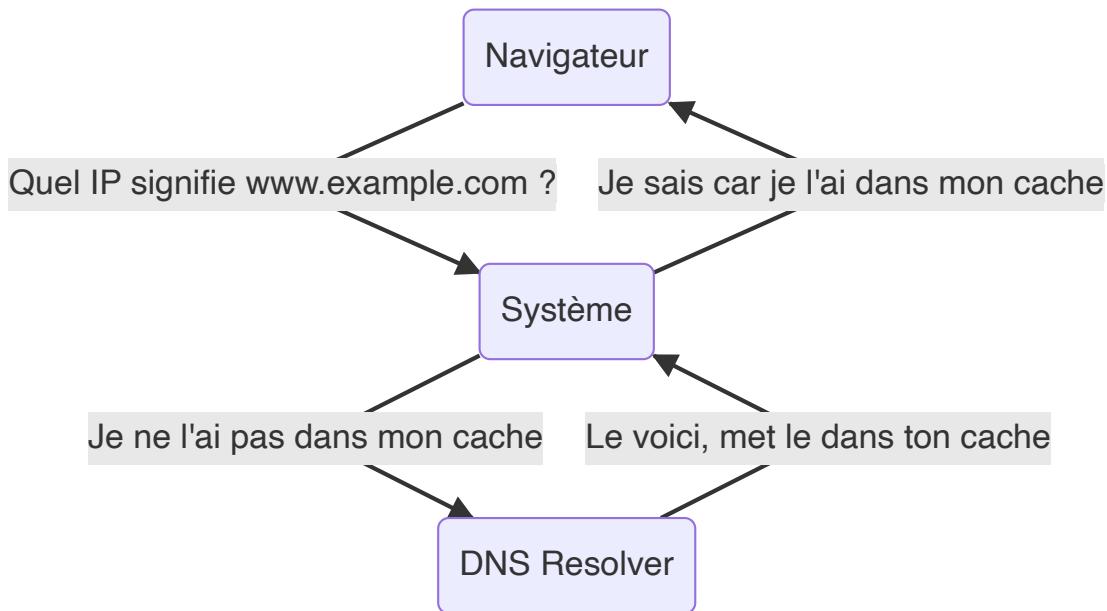
- [otherthewire](#)
- [pwn.college](#)
- [Root Me](#)

# Annexe 1B

## DNS (Domain Name System)

### Fonctionnement général

Un utilisateur tape `www.exemple.com` dans son navigateur. Le navigateur a besoin d'une adresse IP pour se connecter au site car `www.exemple.com` n'est qu'une chaîne de caractères. Celui-ci regarde d'abord dans le **cache** de l'ordinateur s'il n'y a pas déjà une adresse IP correspondante en mémoire. S'il ne trouve pas, il envoie une demande appelée **query** à son **DNS Resolver** (aussi appelé **Resolving Name Server**). Ce DNS resolver (en abrégé DNSR) va alors effectuer la recherche de l'IP. On appelle ce processus le **DNS lookup**.



### Fonctionnement du DNS Resolver

Le DNS Resolver est un serveur que chacun peut choisir et configurer, mais le plus souvent on utilise celui de son fournisseur d'accès internet (FAI). Le DNS Resolver va contacter successivement (récursevement) des **serveurs DNS**, c'est-à-dire des gros annuaires pointant vers d'autres serveurs. À chaque itération, le serveur DNS vérifiera qu'il n'a pas déjà la correspondance en cache ; s'il ne l'a pas, il pointera si possible vers un autre serveur DNS.

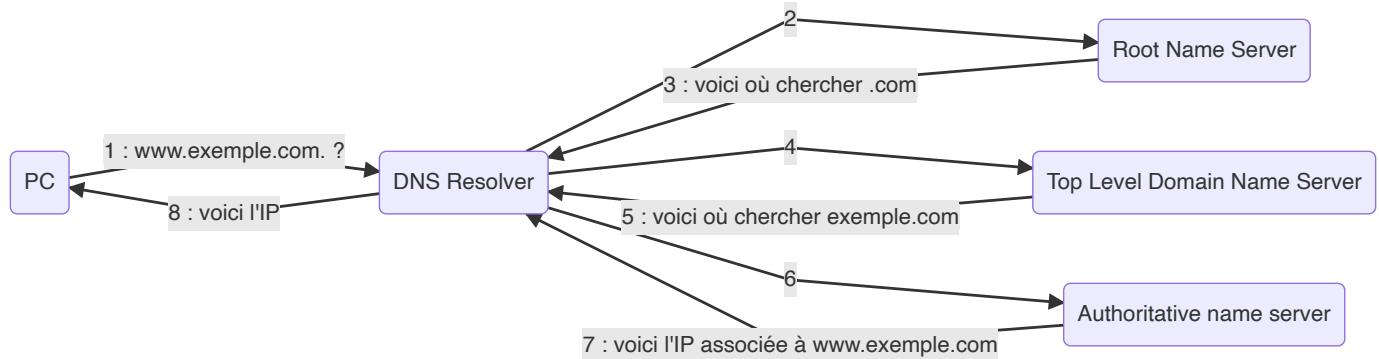
reprenons l'exemple de `www.exemple.com`. En réalité, il y a un dernier point à la fin, qui est implicite : lorsque l'on contacte `www.exemple.com`, on contacte en fait `www.exemple.com.`.

Ce point représente la racine des noms de domaines, et il permet de contacter le **Root Name Server** (RNS).

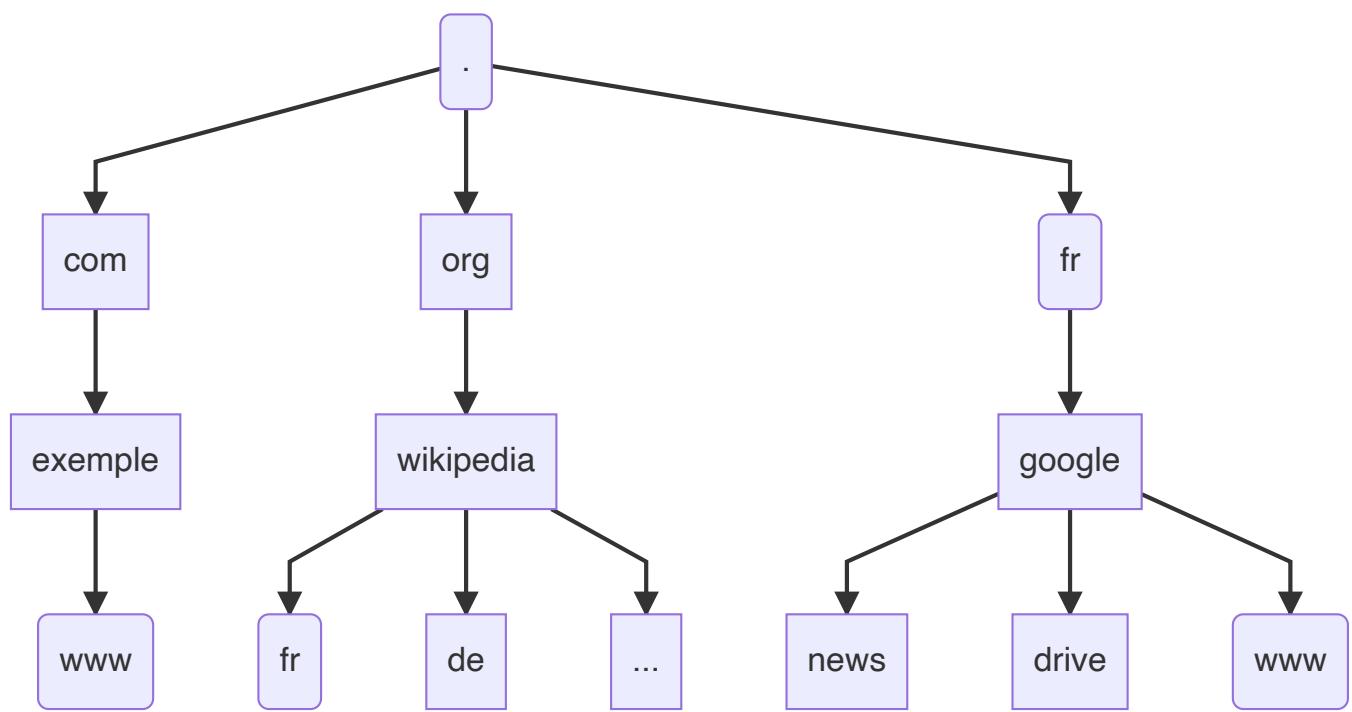
Le RNS va rediriger vers le bon **Top Level Domain Name Server** (TLDNS), c'est-à-dire les serveurs DNS gérant ".com", ".net", etc.

Ensuite, le TLDNS redirigera vers le bon **Authoritative Name Server** (ANS). Ceux-là sont un peu différents : Il ne proposera pas un nouveau serveur DNS à interroger mais l'adresse IP du site demandé si celle-ci est disponible.

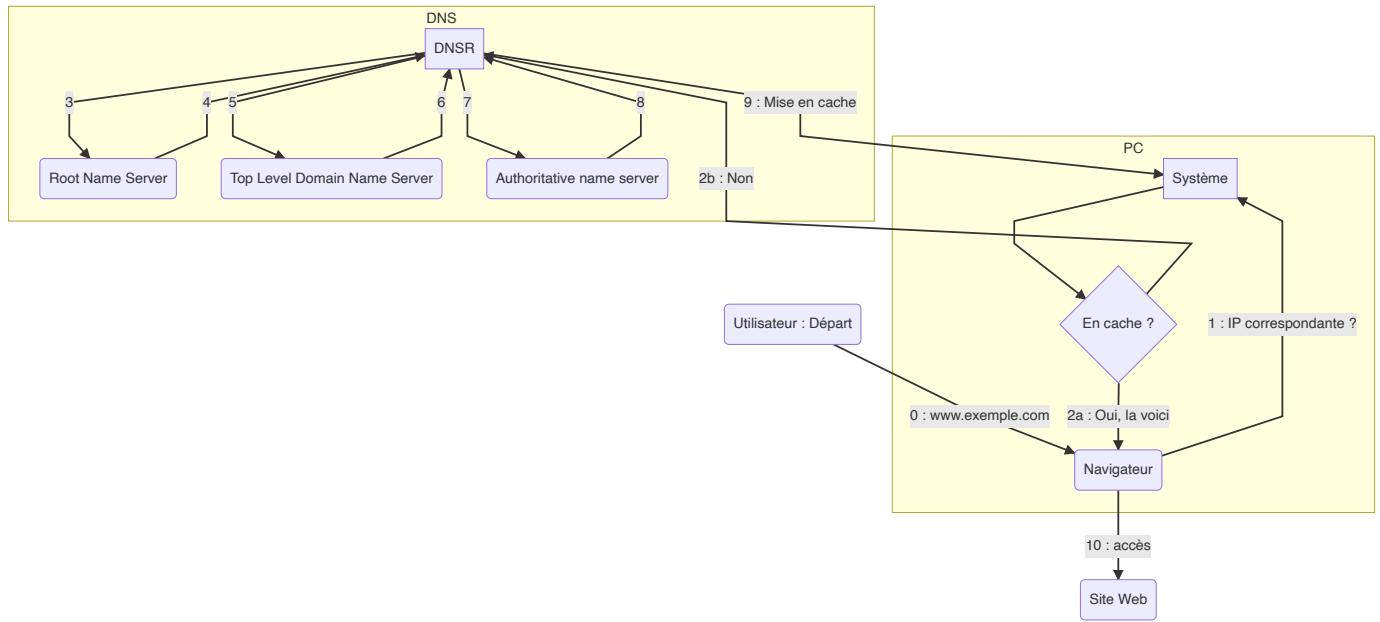
Il est important de noter que les serveurs DNS ne se parlent *jamais* directement : c'est toujours le DNS Resolver qui contacte successivement les nouvelles adresses que l'on lui donne. C'est pourquoi le DNS Resolver fait partie d'une autre classe de serveurs appelée **Recursive Name Servers**.



Exemple de hiérarchie de noms de domaine :



## Diagramme final



## Précisions

- On appelle **Fully Qualified Domain Name** (FQDN) le nom de domaine absolu (ici `www.exemple.com.` ). On peut parler de nom relatif lorsque les parties supérieures sont omises (par exemple ici `www.exemple.`). Si un DNS resolver n'a pas à faire avec un FQDN, il complète avec des valeurs par défaut (search et domain dans /etc/resolv.conf ou DHCP). En règle générale, il vaut mieux travailler avec des FQDN quand on interroge son DNS resolver.
- En réalité, les serveurs DNS ont souvent des copies qui répondent à leur place en cas de problème, ou

bien des serveurs DNS "alternatifs" si le premier ne répond pas. Parfois, les DNSR contactent d'autres DNSR, ou alors un même utilisateur contacte deux DNSR différents.

- Comme expliqué précédemment, on peut aussi utiliser un autre DNSR que celui de son FAI. En effet, il existe des DNS Resolver publics, nommés simplement **Public DNS Resolvers**, comme ceux de google. Cela peut permettre de contourner des privations des autres DNSR qui feraient par exemple exprès de ne pas rendre d'adresse IP pour certains sites web.

## Enregistrements et mise en cache

On appelle un enregistrement un **Resource Record** (RR) qui décrit une **zone DNS** (c'est à dire un domaine, le plus souvent unique). Il existe des dizaines de types de RR différents (voir source 9), mais nous présenterons ici que quelques unes. Les RR ont une durée de vie appelée **Time To Live (TTL)**, ce qui signifie qu'elles ne restent pas en cache éternellement.

Type	ID Type	Nom	Fonction	Exemple
A	1	Address record	Relie un nom d'hôte à l'adresse IP d'un serveur.	<code>nomdomaine.com 3600 IN A 46.31.152.2</code>
AAAA	28	IPv6 address record	Idem, pour IPv6	<code>nomdedomaine.com 3600 IN A 2001:0db8:0000:85a3:0000:0000:ac1f:8001</code>
CNAME	5	Canonical Name Record	Donne un autre nom de domaine pour une adresse particulière	<code>bar.exemple.com CNAME foo.exemple.com</code>
DNAME	39	Delegation Name Record	Donne un autre nom pour les sous-domaines d'un domaine	<code>bar.exemple.com DNAME foo.exemple.com</code>
NS	2	Name Server	Précise quel serveur DNS s'occupe d'une telle zone DNS. Celle-ci ne doit jamais pointer vers un CNAME. Elle est toujours associée avec une RR de type "A" correspondant à la valeur pointée.	<code>exemple.com. 1800 IN NS names.exemple.com (et names.exemple.com A 1.2.3.4)</code>
MX	15	Mail Exchanger Record	Spécifie un serveur mail responsable d'accepter les messages à la place d'un domaine. ex <i>d'utilisation</i> : en déclarer plusieurs pour faire du load balancing et de la redondance	<code>example.com. 1936 IN MX 10 onemail.example.com. example.com. 1936 IN MX 10 twomail.example.com.</code>
SRV	33	Service	Spécifie quel serveur est responsable d'un service	<code>_service._protocole « IN SRV » priorité poids port cible" exemple: _sip_tls.icoadmin.com. 3600 IN SRV 5 20 443 sipdir.online.microsoft.com.</code>
...	...	...	...	...

## En fonction des OS

- Sous Windows : C://Windows/System32/drivers/etc/hosts permet de changer l'annuaire des IP juste sur ce PC (prioritaire sur le cache).
- Par OS :
  - Sous Linux, il n'y a pas de cache DNS au niveau de l'OS. Le cache DNS dépend donc de ce qui a été installé, et les commandes et paramètres aussi. On peut citer par exemple :
    - `systemd-resolved.service` pour *Ubuntu 18.04* et autres distributions courantes
    - `nscd.service` pour les distributions basées sur *RedHat*

- `dnsmasq.service` pour d'autres
  - Sous Windows, il faut s'intéresser à `ipconfig` en tapant par exemple `ipconfig /displaydns`
- *Exemple d'un enregistrement en cache sous windows :*

```
1 Nom d'enregistrement. : steropes.eclair.ec-lyon.fr
2 Type d'enregistrement : 1
3 Durée de vie . . . . : 75746
4 Longueur de données . : 4
5 Section . . . . . : Réponse
6 Enregistrement (hôte) : 156.18.24.10
```

# Annexe 1C

## Configurer un SSH

### Configurer un SSH

Vocabulaire

Mise en place de base

Mise en place sur le serveur

Mise en place sur le client

Création d'une clé SSH

Fichier de configuration rapide

Mise en place avec un bastion

Connexion

Concepts de base

Renforcement du bastion

Fermer des ports

Changer le port par lequel on utilise le SSH

Bloquer la réponse aux pings

Mise en place

Mettre en place les clés SSH

Limiter les sessions

## Vocabulaire

Le **client** est l'ordinateur qui cherche à se connecter, et **l'hôte** est le serveur qui va héberger la connexion. Dans le cas d'une architecture plus avancée on aura plutôt un ensemble de serveurs (**les hosts**) qui possède une LAN privée, et accède à internet en passant à travers une machine pare-feu. Derrière ce pare-feu (ou sur la même machine), on va mettre une machine qui sera **le bastion**. Cette machine sera renforcée pour résister à des attaques, ceci sera fait en fermant tous les ports et en empêchant de répondre aux pings. On a aussi **un client**, qui sera l'utilisateur qui veut se SSH sur une des machines derrière le bastion.

## Mise en place de base

Source : [Site](#)

### Mise en place sur le serveur

On commence par installer les dépendances nécessaires sur le serveur :

```
1 | >>> sudo apt-get update
2 | >>> sudo apt-get install openssh-server
```

On redémarre le SSH avec systemctl :

```
1 | >>> sudo systemctl status ssh    # Affiche le statut du SSH sur le serveur.  
2 | >>> sudo service ssh stop  
3 | >>> sudo service ssh start  
4 | # Ou bien :  
5 | >>> sudo systemctl restart ssh
```

Finalement, on récupère l'adresse ip du hôte (serveur sur lequel on est) :

```
1 | >>> ip a
```

En cas de présence d'un firewall sur le serveur, il faut lui préciser qu'on accepte le ssh (ou ouvrir le port correspondant, par défaut 22) :

```
1 | >>> sudo ufw allow ssh
```

## Mise en place sur le client

On installe les dépendances nécessaires :

```
1 | >>> sudo apt update  
2 | >>> sudo apt install openssh-client
```

Pour se connecter (attention, il faut être sur le même réseau) :

```
1 | >>> ssh username@HostIPaddress
```

Le système mettra souvent une mise en garde lors de la première connection, que l'on accepte.

## Création d'une clé SSH

On peut commencer par vérifier les clés installées sur le client : si cette ligne ne renvoie rien c'est qu'il n'y en a pas.

```
1 | >>> ls -l ~/.ssh/id*
```

On va ensuite créer la paire de clés (par défaut en 2048-bit RSA). La machine va alors demander où les placer, par défaut dans `~/.ssh/id_rsa` qui convient parfaitement. Il affiche ensuite la clé publique, l'empreinte de la clé et son image randomart.

```
1 | >>> ssh-keygen
```

On copie ensuite la clé sur le serveur, plus simplement en utilisant :

```
1 | >>> ssh-copy-id user@hostname  
2 | # On remplace user par le nom d'utilisateur et hostname par le nom du serveur
```

Attention, lors de la première connexion, on risque de recevoir un message disant que l'authenticité ne peut pas être établie, on tape alors yes et entrée. Le système va alors chercher un fichier `id_rsa.pub`, qui contiendrait la clé publique. Il va alors demander le mot de passe. Finalement, on devrait voir un message précisant le nombre de clés ssh ajoutées : 1.

Pour ajouter une clé sur un serveur qui répond sur un autre port que le 22 (par exemple 80), il suffira d'exécuter cette commande :

```
1 | >>> ssh-copy-id "-p 80 user@hostname"
```

Par défaut, les clés SSH sont stockées dans le fichier `~/.ssh/authorized_keys`, mais il est possible de les stocker ailleurs (par exemple sur un serveur LDAP distant). Dans ce cas, on peut aller chercher les clés SSH à l'aide d'un script. Pour indiquer ça dans le fichier de configuration SSH, on écrira :

```
1 | # Si le script est /usr/bin/ldap_ssh_key :
2 |
3 | AuthorizedKeysCommand /usr/bin/ldap_ssh_key
4 | AuthorizedKeysCommandUser nobody
```

## Fichier de configuration rapide

On va commencer par créer un fichier `config` dans le dossier `.ssh` du client. Dans ce fichier on va écrire des lignes de la forme suivante :

```
1 | Host debian
2 |   User nath
3 |   Port 22
4 |   Hostname 172.16.224.201
5 |
6 | # Avec une clé SSH :
7 | Host debian
8 |   User nath
9 |   Port 22
10 |  Hostname 172.16.224.135
11 |  IdentityFile ~/.ssh/id_rsa
```

Pour se connecter au serveur d'adresse 172.16.224.201, il suffira alors de taper `ssh debian`.

## Mise en place avec un bastion

### Connexion

Pour se connecter à un hôte caché derrière le bastion (ici un proxy SSH, `-J` correspond à l'option `ProxyJump` de SSH), le client tapera :

```
1 | ssh -J $BASTION $HOST
```

Il pourra sinon écrire cette condition dans son fichier `~.ssh/config` avec les lignes :

```
1 Host $HOST
2 ProxyJump $BASTION
```

## Concepts de base

Pour renforcer le bastion, les premières étapes sont :

- Fermer tous les ports non utilisés par le SSH
- Changer le port SSH (le mettre sur un autre port que le 22), et potentiellement n'autoriser que certains utilisateurs (selon l'IP)
- Désactiver l'utilisateur root

## Renforcement du bastion

### Fermer des ports

On commence par voir quels ports sont ouverts et écoutent l'extérieur, il y a de nombreuses commandes mais on va utiliser `ss` (`netstat` commence à être vieille et est remplacée par `ss`) :

`ss` : Informations à propos des ports réseau de la machine

- `-u` : Ports utilisant le protocole UDP
- `-t` : Ports utilisant le protocole TCP
- `-l` : Port qui écoutent (listening)
- `-p` : Affiche les procédés qui ont ouvert les ports
- `-n` : Numérique, utilise les adresses IP au lieu des noms de domaines

Pour voir quels services utilisent quels ports (on peut aussi le grep avec un service précis) :

```
1 >>> less /etc/services
```

Pour gérer l'ouverture des ports, on va utiliser [ufw](#) (uncomplicated firewall) :

```
1 # Mise en place du firewall :
2 >>> sudo apt install ufw
3 >>> sudo ufw enable
4
5 # Afficher les règles et l'état du firewall (plusieurs possibilités):
6 >>> sudo ufw status verbose
7 >>> sudo ufw status numbered
8
9 # Ajouter une règle d'application :
10 >>> sudo ufw app list      # Pour lister les app qui ont des règles disponibles
11 >>> sudo ufw allow "OpenSSH" # Pour mettre en place une règle d'application
12 # Pour voir les fichiers de config de ces profils d'applications, on va dans
13 /etc/ufw/applications.d
```

```

14 # Ouvrir des ports :
15 >>> sudo ufw allow 80          # Pour accepter TCP et UDP sur un port
16 >>> sudo ufw allow 80/tcp     # Pour accepter seulement le TCP
17 >>> sudo ufw allow from 192.168.2.128 to any port # Pour accepter une seule machine
18   sur tous les ports
19 >>> sudo ufw allow from 192.168.2.128 to any port 80 # Pour accepter une seule
20   machine sur un seul port
21
22 # Fermer des port
23 >>> sudo ufw deny 80          # Ferme le port 80
24 >>> sudo ufw deny from 192.168.2.0/24    # Pour bloquer tout un sous-réseau
25
26 # Supprimer toutes les règles :
27 >>> sudo ufw reset
28 # Supprimer une règle
29 >>> sudo ufw status numbered      # Repérer le numéro de la règle
30 >>> sudo ufw delete $REGLE_N

```

## Changer le port par lequel on utilise le SSH

On va commencer par ouvrir le bon port (dans notre cas on utilisera le 80) :

```

1 | >>> sudo ufw default deny incoming
2 | >>> sudo ufw allow 80/tcp

```

On modifie ensuite le port de ssh pour le serveur avec `sudo vi /etc/ssh/sshd_config`. On décommente la ligne qui précise le port, et on le change (80 dans notre cas). Bien penser à redémarrer le SSH après les modifs :

```

1 | >>> sudo service sshd restart

```

## Bloquer la réponse aux pings

Les pings ne scannent pas des ports, mais envoient des paquets ICMP auxquels le système répond par défaut. On cherche à supprimer ce comportement pour rendre plus compliquée l'intrusion et l'exploration de notre architecture.

On va modifier des paramètres du kernel, qui se trouvent dans le fichier `/etc/sysctl.conf` : il suffit d'ajouter cette ligne à la fin du fichier `net.ipv4.icmp_echo_ignore_all = 1`. On peut ensuite mettre en place ce changement sans redémarrer avec la ligne :

```

1 | >>> sudo sysctl -p

```

# Mise en place

Une fois toutes les étapes ci-dessus suivies et le port SSH changé, on va suivre ces étapes pour mettre en place le système. Normalement à ce niveau on doit pouvoir se connecter avec un proxyjump sur l'hôte, mais il faudra aussi se connecter sur le bastion avant. On va donc créer une session qui ne nécessite pas de se connecter. On mettra ensuite en place les clés ssh.

## Mettre en place les clés SSH

Attention, il est important de faire cette étape avant de limiter les droits de connexion des sessions. On va échanger une clé entre le **client** et le **bastion** (cf plus haut pour échanger des clés). Dans ce cas on peut ne pas mettre de passphrase (moins sécurisé) si on veut ne pas avoir 2 mots de passe à saisir pour se connecter sur un serveur. Pour la deuxième clé, on la crée aussi sur le client, et on la copie (théoriquement à la main, ou bien en la copiant d'abord sur le bastion) sur le host (dans `~/.ssh/authorized_keys`). Finalement, on a nos 2 clés qui sont mise en place, et seulement la seconde nécessite un mot de passe.

## Limiter les sessions

On commence par désactiver la connexion en ssh au serveur pour tous les utilisateurs normaux. Ceci se fait dans le fichier `/etc/ssh/sshd_config` :

```
1 # Prohibit regular SSH clients from allocating virtual terminals, forward X11, etc:
2 PermitTTY no
3 X11Forwarding no
4 PermitTunnel no
5 GatewayPorts no
6
7 # Prohibit launching any remote commands:
8 ForceCommand /usr/sbin/nologin
```

Puis on redémarre le démon ssh : `sudo service sshd restart`.

# Annexe 1D

## LDAP

**Attention :** ce tutoriel décrit comment installer LDAP dans le but de l'utiliser avec PAM pour s'authentifier. L'objectif est ici d'utiliser la base de donnée du LDAP comme base de donnée d'utilisateurs.

## Définition

LDAP signifie Lightweight Directory Access Protocol. C'est un protocole qui permet de stocker des données dans des bases non relationnelles, sous forme d'annuaire (pyramide). Pour comprendre les bases de ce protocole, le tutoriel d'OpenClassroom est un bon point de départ.

## Installation

Notre backend LDAP sera slapd, que l'on commence par installer avec :

```
1 | >> sudo apt install slapd
```

On rentre alors les paramètres nécessaires (le nom d'entité demandé sera le "o" de. organization du LDAP). On peut ensuite entrer les paramètres suivants en tapant :

```
1 | >> sudo dpkg-reconfigure slapd
```

Une fois slapd correctement installé, penser à l'activer et à le lancer :

```
1 | >> sudo systemctl enable slapd
2 | >> sudo systemctl start slapd
```

Les fichiers importants sont :

- Les fichiers de configurations : `/etc/ldap/slapd.d/`
- Les templates de la base de donnée : `/etc/ldap/schema/`
- La base de donnée elle-même : `/var/lib/ldap/`

## Monter la base de donnée

Les groupes et les entrées de la base de donnée sont définies par des fichiers LDIF (LDAP Data Interchange Format), qui permettent de représenter en texte la base de donnée. Dans ces fichiers chaque entrée est représentée par une série de ligne, et les entrées distinctes se différencient par un saut de ligne. On peut faire des lignes plus longues en ajoutant un espace au début d'une ligne, qui est alors considérée comme la continuité de la précédente.

Pour ajouter des données dans la base on va donc créer un fichier LDIF et le remplir de la façon suivante :

```

1 >> touch newdata.ldif
2 >> vi newdata.ldif
3 # Ajouter les données dans le fichier (ici pour créer un groupe d'utilisateurs):
4
5 # users, eclair.ec-lyon.fr
6 dn: ou=users,dc=example,dc=com
7 objectClass: organizationalUnit
8 objectClass: top
9 ou: users

```

Pour préciser quel est l'objet que l'on crée, on utilise des objectClass dont une liste non exhaustive est présentée ici :

- `top` : Classe présente sur tous les objets, qui permet de définir leurs propriétés de base
- `organizationalUnit` : Signifie que l'on crée une "ou", soit un groupe

D'autres exemples sont : `account`, `posixAccount`, `shadowAccount`.

Mais lorsque l'on veut créer des tables précises, par exemple un ensembles d'utilisateurs avec leurs clés SSH publiques, il nous faut d'autres types de classes. Ces classe sont alors définies avec des schema dans `/etc/ldap/schema`.

On ajoute ensuite notre entrée / dossier avec `ldapadd` :

- `-D` : précise la base dans laquelle on écrit (correspond à `-b` pour `ldapsearch`)
- `-f` : précise le fichier à ajouter
- `-h` : précise le serveur LDAP (en adresse IP)
- `-H` : précise le serveur LDAP (en nom de domaine)
- `-w` : authentification simple avec mot de passe en ligne de commande
- `-x` : Authentification simple
- `-Y` : pour pouvoir préciser le mode d'authentification (souvent utilisé avec `-Y EXTERNAL`)

La commande serait donc la suivante :

```
1 >> sudo ldapadd -xWD cn=admin,dc=eclair,dc=ec-lyon,dc=fr -f groups.ldif
```

## Ajouter un schéma personnalisé

Lorsque l'on veut créer des attributs plus avancés dans une base de donnée, il est nécessaire de les définir au préalable pour que LDAP les reconnaisse. Ainsi, on va devoir ajouter un nouveau schéma à LDAP, qui définira ces attributs. Prenons l'exemple d'ÉCLAIR, qui stocke ses utilisateurs ainsi que leurs clés SSH publiques sur son annuaire LDAP.

On commence par créer le fichier de définition des clés (trouvable sur internet) :

```

1 dn: cn=openssh-lpk,cn=schema,cn=config
2 objectClass: olcSchemaConfig
3 cn: openssh-lpk
4 olcAttributeTypes: ( 1.3.6.1.4.1.24552.500.1.1.13 NAME 'sshPublicKey'
5 DESC 'MANDATORY: OpenSSH Public key'
6 EQUALITY octetStringMatch
7 SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
8 olcObjectClasses: ( 1.3.6.1.4.1.24552.500.1.1.2.0 NAME 'ldapPublicKey' SUP top
AUXILIARY
9 DESC 'MANDATORY: OpenSSH LPK objectclass'
10 MAY ( sshPublicKey $ uid )
11 )

```

**ATTENTION :** les deux espaces à gauche des lignes de l'attribut et de la classe ne sont pas des tabulations !!!

On va ensuite ajouter ce fichier dans le LDAP avec la commande suivante :

```
1 | >> sudo ldapadd -Y EXTERNAL -H ldap:// -f openssh-lpk.ldif
```

On peut alors créer une entrée avec les lignes suivantes dedans :

```

1 | objectClass: ldapPublicKey
2 | sshPublicKey: ssh-rsa.....root@local

```

## Faire des recherches dans la base

On utilise la commande `ldapsearch`, avec un certain nombre de paramètres et d'options :

- `-x` : Utilise une authentification simple au lieu de SASL
- `-b` : On recherche directement dans un endroit de la base de donnée au lieu de la racine
- `-h` : Précise le serveur LDAP (donc celui sur lequel est située la base de donnée) avec une IP
- `-H` : Précise le serveur LDAP avec un nom de domaine

On écrit donc la base de la recherche, puis ensuite on ajoute des filtres. Si l'on veut tout voir, on tapera seulement :

```
1 | >> ldapsearch -x -b dc=eclair,dc=ec-lyon,dc=fr -h 156.----.----.---
```

Pour préciser des filtres si l'on veut par exemple voir seulement la clé publique d'un seul utilisateur on peut exécuter :

```
1 | >> ldapsearch -x -b dc=eclair,dc=ec-lyon,dc=fr -h 156.----.----.--- '(&
(objectClass=posixAccount)(uid="fdebouck"))' 'sshPublicKey'
```

Ici on cherche dans les `eclair.ec-lyon.fr` sur le serveur `156.----.----`. On cherche à récupérer le paramètre `sshPublicKey`, en filtrant les entrées telles qu'elles soient de la classe `posixAccount`, et que l'uid soit `"fdebouck"`. Si l'on avait voulu une seule condition on aurait pu écrire `'(uid=' "fdebouck" ')'`.

## Installer une interface web pour gérer la base de donnée

Cette interface web sera gérée par `phpldapadmin`.

Si cela fonctionne, on installe directement le paquet :

```
1 | >> sudo apt install phpldapadmin
```

Si apt ne trouve pas le paquet, il faut le cloner directement depuis le repo de debian :

```
1 | >> wget  
http://ftp.fr.debian.org/debian/pool/main/p/phpldapadmin/phpldapadmin_1.2.6.3-  
0.2_all.deb  
2 | >> sudo apt install ./phpldapadmin_1.2.6.3-0.2_all.deb
```

On peut ensuite ajouter des templates personnalisés en ajoutant des fichiers xml dans  
`/etc/phpldapadmin/templates/modification/custom_blablabla.xml`

## Annexe 1E

# Linux PAM

PAM signifie Pluggable Authentication Modules, c'est un ensemble de bibliothèques qui permettent de précisément gérer l'authentification de différents services du système. PAM possède un fichier de configuration `/etc/pam.conf`, qui sera ignoré s'il existe un dossier de configuration `/etc/pam.d`. Dans ce dossier on peut créer un fichier de configuration pour chaque service pour lequel on souhaite utiliser PAM. Par exemple si on le configure pour le SSH, on va éditer le fichier `/etc/pam.d/sshd`.

## Configuration

### Simple

La configuration se fait ligne par ligne, et chaque ligne est sous cette forme :

```
1 | type control-flag module module-arguments
```

On peut indiquer 4 types différents :

- `account` : Vérifications liées au compte, est-ce que l'utilisateur peut utiliser le service concerné et est-ce que le mot de passe de l'utilisateur a expiré.
- `auth` : Authentifie un utilisateur et setup ses credentials.
- `password` : Met à jour les mots de passe des utilisateurs et gère l'authentification.
- `session` : Gère les actions faites au début et à la fin de la session.

Les control-flags peuvent prendre ces valeurs :

- `requisite` : Il est nécessaire que le module soit successful pour que l'authentification ait lieu. En cas d'erreur l'utilisateur sera notifié.
- `required` : Il est nécessaire que le module soit successful pour continuer. En cas d'erreur l'utilisateur ne sera notifié que quand tous les modules du même type auront été testés.
- `sufficient` : Si les modules précédents sont successful, on ignore les suivants et le service peut continuer
- `optional` : Le succès ou l'échec de ce module n'est pas enregistré
- Rarements utilisés : `include` et `substack` existent aussi

Une liste des modules disponibles est présente sur [ce site](#).

### Plus avancée

Pour des configurations plus avancées on peut créer des lignes de cette forme :

```
1 | type [value1=action1 value2=action2 ...] module module-arguments
```

Avec `valueN` correspondant au code de retour de la fonction invoquée dans le module pour laquelle la ligne est définie. Les valeurs possibles sont trouvables [le site de documentation de PAM](#). Les actions possibles sont les suivantes :

- `ignore` : Si cette action est utilisée avec un stack de modules, le statut renvoyé par les modules n'est pas pris en compte par l'application.
- `bad` : Indique que le code de retour doit être pris comme indication de l'échec du module. Si c'est le premier module en échec, son statut sera utilisé pour tout le stack.
- `die` : Équivalent à bad mais termine le stack de modules immédiatement
- `ok` : Ce return code devra contribuer directement au return code du stack de modules
- `done` : Équivalent à bad mais termine le stack de modules immédiatement
- `N` (un chiffre) : équivalent à ok mais saute directement au Nième module suivant
- `Reset` : Nettoie toute la mémoire de l'état du stack de module et recommence avec le stack suivant

## Annexe 1F

# SSLH

---

Voir [ici](#).

SSLH est un multiplexeur de ports, qui permet donc de faire passer plusieurs services sur un seul port. Un cas d'usage typique est de faire passer du SSH par le port 80 ou 443, car les entreprises ferment souvent le port 22 pour des raisons de sécurité. Il est alors nécessaire de faire passer le SSH et le HTTP / HTTPS par le même port, c'est l'utilité de SSLH.

## Première mise en place

---

On commence par installer le paquet :

```
1 | >>> sudo apt install sslh
```

Le fichier de configuration est `/etc/default/sslh`. La configuration se fera sur la ligne `DAEMON_OPTS` (voir partie Configuration). Pour activer le protocole, il faut changer la ligne du fichier de config `RUN=no` à `RUN=yes`, et initialiser le daemon :

```
1 | >>> sudo systemctl enable sslh
2 | >>> sudo systemctl start sslh
```

## Configurations :

---

Elles sont font donc sur la ligne `DAEMON_OPTS` du fichier `/etc/default/sslh`. Voici les options :

- `--user sslh` : nécessite cet utilisateur pour fonctionner
- `--listen 0.0.0.0:80` : écoutera toutes les interfaces sur le port 80
- `--ssh 127.0.0.1:22` : pour rediriger les flux ssh sur le port 22 du localhost (on peut aussi écrire directement localhost)
- `--anyprot eole.ec-lair.ec-lyon.fr:8080` : renvoie tous les autres protocoles sur la machine portant ce nom de domaine

On peut aussi saisir des options (trouvables avec `man sslh`) :

- `-n` : Numérique, évite de faire une requête DNS à chaque requête
- `-t NUM` : Timeout, temps avant d'envoyer la connection au protocole de timeout (SSH en général), par défaut 2 secondes

## Installation d'une solution sur Docker

---

On utilisera la solution de [Yves Rutschle](#). Il faut au préalable avoir installé docker et git sur la machine.

On commence par cloner le repo git :

```
1 | >> git clone https://github.com/yrutschle/sslh.git
```

On se place ensuite dans le dossier `sslh` créé et on construit l'image docker

```
1 >> cd sslh  
2 >> sudo make docker
```

On le configure ensuite dans le `docker-compose.yml` :

```
1 sslh:  
2   image: registry.eclair.ec-lyon.fr/sslh:latest  
3   restart: unless-stopped  
4   hostname: sslh  
5   ports:  
6     - "156.----:443:443"  
7   command: --listen=0.0.0.0:443 --tls=$REVERSE_PROXY.eclair.ec-lyon.fr:443 --  
8     ssh=$PROXY_SSH.eclair.ec-lyon.fr:22  
9   volumes:  
10    - ./config.cfg:/etc/sslh.cfg  
11  depends_on:  
12    - caddy
```

Ici `tls` désigne les protocoles HTTP et HTTPS.

# Annexe 1G

## Principaux services Linux

---

Pour obtenir les services qui tournent sur une machine, il suffit de taper `systemctl` et de trouver les lignes en `.service`. On peut aussi utiliser `systemctl | grep service`.

- `Fail2ban` : Analyse les logs de certains services pour bloquer des tentatives de connexions comme le bruteforce
- `Bind9` : DNS (très utilisé)
- `cron` : Outil pour exécuter automatiquement des scripts à des horaires précis
- `dbus` : Permet de faire communiquer des applications entre elles à travers une API standardisée
- `getty` : ? Permet de se connecter à un terminal tty, demande l'identifiant et le renvoie à l'application
- `kmod` : Gère les modules du kernel Linux
  - `kmod-static-nodes` : crée une liste de device nodes
- `nscd` : ? Gère le cache
- `nsclid` : Gère les connexions depuis un LDAP
- `ntp` : Network Time Protocol, synchronise l'horloge de la machine avec celles du réseau
- `rsyslog` : Transfère les messages log sur un réseau IP
- `zabbix` : Collecte des informations de performances de l'OS pour les envoyer sur un serveur (celui sur lequel est installé zabbix server)
- `exim4` : Serveur de messagerie par défaut de Debian
- `nginx` : serveur web, aussi utilisé comme reverse proxy, cache HTTP et load balancer
- `sslh` : SSH / HTTP multiplexer, gère les connexions entre les 2 protocoles
- `ufw` : Uncomplicated Firewall, pare-feu créé pour être facile à utiliser

### 13.2 Annexes spécifiques à notre installation

Les annexes suivantes devraient permettre aux futurs membres d'éclair de configurer entièrement les machines, ce sont des tutoriels propres à l'installation d'ECLAIR.

## Annexe 2A

# Comment communiquer avec les switchs ?

---

Brancher votre ordinateur avec un port usb (mini) (pour S5 "tulkas" et S6 "orone"). Sinon (pour S7 en particulier), chaque switch a un port *console* dans lequel connecter le câble ethernet-série-usb mais cette méthode impose de télécharger divers drivers, plus disponibles pour *windows 11*.

Il faut installer **Tera Term VT** ou autre console permettant d'afficher une communication série, et se connecter au port COM concerné avec une résolution de 9600 bauds.

## S5 - aruba 2540 JL355A - "tulkas"

---

- Ce switch est exactement le même que le S6. Se référer donc à S6 !

## S6 - aruba 2540 JL355A - "orone"

---

Instructions pour le firmware VC.16.02.0012

- Presser entrée une fois dans la console.
- Il commence par afficher le modèle du switch et un petit paragraphe sur la licence. Presser une touche (`press any key to continue`).
- En fonction de ce qui a été fait avant, il affichera `orone>` ou `orone#`.
  - Si l'affiche `orone>`, il faut se connecter en temps que manager. Entrer la commande `enable`. Si l'indique un mot de passe et un login et que ni vous ni personne ne le connaît, prendre un trombone et appuyer sur le bouton "clear" en façade pendant 10s (entre 5s et 15s). Attention, pas plus ni moins car cela ne fait pas la même action. Attention aussi à ne pas confondre avec le bouton "reset". Ensuite, redémarrez la connexion.
  - Si l'affiche `orone#`, vous êtes connecté en tant que manager.
- Vous pouvez maintenant rentrer des commandes. Quelques commandes utiles :
  - `help` vous donne une liste des commandes utilisables
  - `menu` transforme le terminal en menu "graphique". Celui-ci est bien fait et utilisable pour avoir une vue d'ensemble (ex : pour voir l'ensemble des VLAN par ports, aller dans les menus 2 puis 8 puis 3).
  - `link-test` teste si une adresse mac particulière est présente sur le réseau.
  - `show cpu` donne une moyenne d'utilisation du CPU sur les 300 dernières secondes.
  - Il est possible d'appuyer sur TAB pendant l'écriture d'une commande pour afficher la liste des paramètres possibles de celle-ci.
- Une liste des erreurs que vous pouvez rencontrer :
  - `invalid input` : la commande est inexistante ou pas applicable (ex : `enable` en étant déjà manager)
  - `incomplete input` : la commande manque de paramètres.
    - ex : la commande `show` seule ne fait rien : `show vlan` affiche les VLAN paramétrées et leur ID. Penser au TAB !
    - Une fonction peut avoir plusieurs paramètres : ex : `show vlan 11`.

- Si certaines fonctions ne marchent pas, certains menus sont vides, etc ... Vérifiez que vous êtes bien connectés en tant que manager (cf plus haut)

# Annexe 2B

## Ajouter un serveur sur le SSH

L'objectif est de pouvoir accéder à ce serveur à travers l'architecture SSH utilisant le LDAP, et de bloquer toute autre connexion SSH.

### Config globale

On commence par changer le fichier de configuration SSH serveur : `/etc/ssh/sshd_config`

```

1 # This is the sshd server system-wide configuration file. See
2 # sshd_config(5) for more information.
3
4 # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
5
6 # The strategy used for options in the default sshd_config shipped with
7 # OpenSSH is to specify options with their default value where
8 # possible, but leave them commented. Uncommented options override the
9 # default value.
10
11 Protocol 2
12 #Port 22
13 #AddressFamily any
14 #ListenAddress 0.0.0.0
15 #ListenAddress ::

16
17 #HostKey /etc/ssh/ssh_host_rsa_key
18 #HostKey /etc/ssh/ssh_host_ecdsa_key
19 #HostKey /etc/ssh/ssh_host_ed25519_key
20
21 # Ciphers and keying
22 #RekeyLimit default none
23
24 # Logging
25 #SyslogFacility AUTH
26 #LogLevel INFO
27
28 # Authentication:
29
30 #LoginGraceTime 2m
31 PermitRootLogin no
32 StrictModes yes
33 #MaxAuthTries 6
34 #MaxSessions 10
35
36 PubkeyAuthentication yes
37
38 # Expect .ssh/authorized_keys2 to be disregarded by default in future.
39 #AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

```

```

40
41 #AuthorizedPrincipalsFile none
42
43 AuthorizedKeysCommand /usr/bin/ldap_ssh_key
44 AuthorizedKeysCommandUser nobody
45
46 # For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
47 #HostbasedAuthentication no
48 # Change to yes if you don't trust ~/.ssh/known_hosts for
49 # HostbasedAuthentication
50 #IgnoreUserKnownHosts no
51 # Don't read the user's ~/.rhosts and ~/.shosts files
52 #IgnoreRhosts yes
53
54 # To disable tunneled clear text passwords, change to no here!
55 PasswordAuthentication no
56 #PermitEmptyPasswords no
57
58 # Change to yes to enable challenge-response passwords (beware issues with
59 # some PAM modules and threads)
60 ChallengeResponseAuthentication no
61
62 # Kerberos options
63 #KerberosAuthentication no
64 #KerberosOrLocalPasswd yes
65 #KerberosTicketCleanup yes
66 #KerberosGetAFSToken no
67
68 # GSSAPI options
69 #GSSAPIAuthentication no
70 #GSSAPICleanupCredentials yes
71 #GSSAPIStrictAcceptorCheck yes
72 #GSSAPIKeyExchange no
73
74 # Set this to 'yes' to enable PAM authentication, account processing,
75 # and session processing. If this is enabled, PAM authentication will
76 # be allowed through the ChallengeResponseAuthentication and
77 # PasswordAuthentication. Depending on your PAM configuration,
78 # PAM authentication via ChallengeResponseAuthentication may bypass
79 # the setting of "PermitRootLogin without-password".
80 # If you just want the PAM account and session checks to run without
81 # PAM authentication, then enable this but set PasswordAuthentication
82 # and ChallengeResponseAuthentication to 'no'.
83 UsePAM yes
84
85 #AllowAgentForwarding yes
86 #AllowTcpForwarding yes
87 #GatewayPorts no
88 X11Forwarding yes

```

```

89 #X11DisplayOffset 10
90 #X11UseLocalhost yes
91 #PermitTTY yes
92 PrintMotd no
93 #PrintLastLog yes
94 #TCPKeepAlive yes
95 #UseLogin no
96 #UsePrivilegeSeparation sandbox
97 #PermitUserEnvironment no
98 #Compression delayed
99 ClientAliveInterval 60
100 ClientAliveCountMax 5
101 #UseDNS no
102 #PidFile /var/run/sshd.pid
103 #MaxStartups 10:30:100
104 #PermitTunnel no
105 #ChrootDirectory none
106 #VersionAddendum none
107
108 # no default banner path
109 #Banner none
110
111 # Allow client to pass locale environment variables
112 AcceptEnv LANG LC_*
113
114 # override default of no subsystems
115 Subsystem sftp /usr/lib/openssh/sftp-server
116
117 # Example of overriding settings on a per-user basis
118 #Match User anoncvs
119 # X11Forwarding no
120 # AllowTcpForwarding no
121 # PermitTTY no
122 # ForceCommand cvs server

```

On crée ensuite le fichier qui s'occupe d'aller chercher la clé de l'utilisateur dans `/usr/bin/ldap_ssh_key` :

```

1#!/bin/bash
2set -eou pipefail
3IFS=$'\n\t'
4
5result=$(ldapsearch -x -b dc=eclair,dc=ec-lyon,dc=fr -h 156.18.24.6 '(&
6(objectClass=posixAccount)(uid=\"$1\"))' 'sshPublicKey')
7attrLine=$(echo "$result" | sed -n '/^ /{H;d};/sshPublicKey:/x;$g;s/\n
8*//g;/sshPublicKey:/p')
9
10if [[ "$attrLine" == sshPublicKey::* ]]; then
11    echo "$attrLine" | sed 's/sshPublicKey:: //'
12    base64 -d
13elif [[ "$attrLine" == sshPublicKey:* ]]; then
14    echo "$attrLine"
15fi

```

```

11 echo "$attrLine" | sed 's/sshPublicKey: //'
12 else
13   exit 1
14 fi

```

On change les droits de ce fichier pour qu'il soit seulement lisible et exécutable par les utilisateurs non root :

```
1 | >> sudo chmod 755 /usr/bin/ldap_ssh_key
```

Installer les paquets nécessaires au fonctionnement de LDAP avec PAM :

```
1 | >> sudo apt install libpam-ldapd libnss-ldapd ldap-utils
```

On configure ensuite PAM avec ce script (à copier sur la machine et à exécuter après avoir changé ses droits :

```

1 #!/bin/sh
2 # Recopie la conf' dans les bons fichiers
3 # À lancer en root
4 # by Florent Gattoni, 2015
5
6 cat > /etc/pam.d/common-account << "EOF"
7 account required pam_unix.so
8 account sufficient pam_succeed_if.so uid < 1000 quiet
9 account [default=bad success=ok user_unknown=ignore] pam_ldap.so
10 account required pam_permit.so
11 EOF
12
13 cat > /etc/pam.d/common-auth << "EOF"
14 auth sufficient pam_unix.so nullok_secure
15 auth requisite pam_succeed_if.so uid >= 1000 quiet
16 auth sufficient pam_ldap.so use_first_pass
17 auth required pam_deny.so
18 EOF
19
20 cat > /etc/pam.d/common-password << "EOF"
21 password sufficient pam_unix.so md5 obscure min=4 max=8 nullok try_first_pass
22 password sufficient pam_ldap.so
23 password required pam_deny.so
24 EOF
25
26 cat > /etc/pam.d/common-session << "EOF"
27 session required pam_unix.so
28 session required pam_mkhomedir.so skel=/etc/skel/
29 session optional pam_ldap.so
30 EOF
31

```

```

32 cat > /etc/pam.d/common-session-noninteractive << "EOF"
33 session [default=1]      pam_permit.so
34 session requisite      pam_deny.so
35 session required       pam_permit.so
36 session required     pam_unix.so
37 session optional      pam_ldap.so
38 EOF
39
40 cat > /etc/pam.d/su << "EOF"
41 @include common-auth
42 @include common-account
43 @include common-session
44 EOF
45
46 cat > /etc/pam.d/sshd << "EOF"
47 auth sufficient      pam_ldap.so
48 auth sufficient      pam_unix.so
49 account sufficient    pam_permit.so
50 EOF
51
52 cat > /etc/pam.d/system-auth << "EOF"
53 auth      required      pam_env.so
54 auth      sufficient    pam_unix.so nullok try_first_pass
55 auth      requisite     pam_succeed_if.so uid >= 500 quiet
56 auth      required      pam_deny.so
57
58 account      required      pam_unix.so
59 account      sufficient    pam_succeed_if.so uid < 500 quiet
60 account      required      pam_permit.so
61
62 password      requisite    pam_cracklib.so try_first_pass retry=3
63 password      sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authok
64 password      required      pam_deny.so
65
66 session      optional      pam_keyinit.so revoke
67 session      required      pam_limits.so
68 session      [success=1 default=ignore] pam_succeed_if.so service in crond quiet
use_uid
69 session      required      pam_unix.so
70 EOF
71
72 cat > /etc/nsswitch.conf << "EOF"
73 passwd:        files ldap
74 group:         files ldap
75 shadow:        files ldap
76
77 hosts:          files dns ldap
78 networks:      files ldap
79

```

```
80 protocols:      db files ldap
81 services:       db files ldap
82 ethers:         db files ldap
83 rpc:            db files ldap
84
85 netgroup:       nis ldap
86 aliases:        ldap
87 EOF
```

**Attention :** il est très important d'avoir une session root démarrée pendant l'exécution de ce script, et de copier l'intégralité du dossier `/etc/pam.d` avant de l'exécuter. Ainsi, si il y a une erreur dans la nouvelle configuration, on peut toujours récupérer l'ancienne et ne pas perdre la machine...

De plus, à l'heure de la rédaction de ce rapport ce script n'a pas encore été réécrit et n'est pas prêt à être utilisé... Il a tout de même été placé dans le rapport pour illustrer et donner une idée de la manière dont les configurations doivent être faites.

# Annexe 2C

## Remise en place de LDAP

On commence par installer le backend LDAP et le lancer :

```

1 >> sudo apt install slapd
2 >> sudo dpkg-reconfigure slapd
3 >> sudo systemctl enable slapd
4 >> sudo systemctl start slapd

```

On ajoute le schéma permettant de stocker des clés SSH, on remplit donc un fichier

`/etc/ldap/schema/openssh-lpk.ldif` :

```

1 dn: cn=openssh-lpk,cn=schema,cn=config
2 objectClass: olcSchemaConfig
3 cn: openssh-lpk
4 olcAttributeTypes: ( 1.3.6.1.4.1.24552.500.1.1.1.13 NAME 'sshPublicKey'
5   DESC 'MANDATORY: OpenSSH Public key'
6   EQUALITY octetStringMatch
7   SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
8 olcObjectClasses: ( 1.3.6.1.4.1.24552.500.1.1.2.0 NAME 'ldapPublicKey' SUP top
AUXILIARY
9   DESC 'MANDATORY: OpenSSH LPK objectclass'
10  MAY ( sshPublicKey $ uid )
11  )

```

On ajoute ce schéma à la base de donnée avec la commande (en étant placé dans le répertoire

`/etc/ldap/schema/`) :

```

1 >> sudo ldapadd -Y EXTERNAL -H ldapi:/// -f openssh-lpk.ldif

```

On définit ensuite les organizational unit *groups* et *users*, et on définit les groupes *sysadmin* et *dev* à l'aide de ce fichier :

```

1 # users, eclair.ec-lyon.fr
2 dn: ou=users,dc=eclair,dc=ec-lyon,dc=fr
3 objectClass: organizationalUnit
4 objectClass: top
5 ou: users
6
7 # groups, eclair.ec-lyon.fr
8 dn: ou=groups,dc=eclair,dc=ec-lyon,dc=fr
9 objectClass: organizationalUnit
10 objectClass: top
11 ou: groups
12
13 # dev, groups, eclair.ec-lyon.fr

```

```

14 dn: cn=dev,ou=groups,dc=eclair,dc=ec-lyon,dc=fr
15 cn: dev
16 gidNumber: 9501
17 objectClass: posixGroup
18
19 # sysadmin, groups, eclair.ec-lyon.fr
20 dn: cn=sysadmin,ou=groups,dc=eclair,dc=ec-lyon,dc=fr
21 cn: sysadmin
22 gidNumber: 9500
23 objectClass: posixGroup

```

On ajoute le fichier avec la commande suivante :

```
1 >> sudo ldapadd -xWD cn=admin,dc=eclair,dc=ec-lyon,dc=fr -f groups.ldif
```

On installe ensuite l'interface web `phpldapadmin` :

```

1 >> wget
http://ftp.fr.debian.org/debian/pool/main/p/phpldapadmin/phpldapadmin_1.2.6.3-
0.2_all.deb
2 >> sudo apt install ./phpldapadmin_1.2.6.3-0.2_all.deb

```

Il faut alors changer le fichier de configuration suivant :

```

1 # /etc/phpldapadmin/config.php
2
3 # Dans "Support for attrs display order" décommenter / changer les lignes suivantes
4 :
5 $config->custom->appearance[ 'attr_display_order' ] = array(
6   'cn',
7   'uid',
8   'gecos',
9   'uidNumber',
10  'gidNumber',
11  'loginShell',
12  'homeDirectory',
13  'userPassword',
14  'shadowLastChange',
15  'shadowMax',
16  'shadowWarning'
17 );
18
19 # Dans "Define your LDAP servers in this section"
20 # Changer le nom du site :
21 $servers->setValue( 'server', 'name', 'Eclair LDAP' );
22
23 # Changer tous les 'dc=example,dc=com' en 'dc=eclair,dc=ec-lyon,dc=fr' :
24 $servers->setValue( 'server', 'base', array( 'dc=eclair,dc=ec-lyon,dc=fr' ) );

```

```
24 | $servers->setValue('login','bind_id','cn=admin,dc=eclair,dc=ec-lyon,dc=fr');
```

Et créer les fichiers suivants :

```

1 /etc/phpldapadmin/templates/modification/custom_group.xml
2
3 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
4 <!DOCTYPE template SYSTEM "template.dtd">
5
6 <template>
7 <askcontainer>1</askcontainer>
8 <description>New Custom Eclair Group</description>
9 <icon>ldap-ou.png</icon>
10 <invalid>0</invalid>
11 <rdn>cn</rdn>
12 <regexp>^cn=[a-zA-Z0-9]*,ou=groups</regexp>
13 <title>Custom: Eclair Group</title>
14 <visible>1</visible>
15
16 <objectClasses>
17 <objectClass id="posixGroup"></objectClass>
18 </objectClasses>
19
20 <attributes>
21 <attribute id="cn">
22   <display>Group Name</display>
23   <order>1</order>
24   <page>1</page>
25 </attribute>
26 <attribute id="gidNumber">
27   <display>GID Number</display>
28   <order>2</order>
29   <page>1</page>
30   <readonly>1</readonly>
31   <spacer>1</spacer>
32   <value>=php.GetNextNumber(/;gidNumber;;;;9500)</value>
33 </attribute>
34 <attribute id="memberUid">
35   <display>Users</display>
36   <order>3</order>
37   <page>1</page>
38   <value><![CDATA[=php.MultiList(/;(objectClass=posixAccount);uid;%cn%)]]>
39 </value>
40 </attribute>
41 </attributes>
42 </template>
```

```

1 /etc/phpldapadmin/templates/modification/custom_ou.xml
2
3 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
4 <!DOCTYPE template SYSTEM "template.dtd">
5
6 <template>
7 <askcontainer>1</askcontainer>
8 <description>New Organisational Unit</description>
9 <icon>ldap-ou.png</icon>
10 <invalid>0</invalid>
11 <rdn>ou</rdn>
12 <!-- <regexp>^o=.*,</regexp> -->
13 <title>Generic: Organisational Unit</title>
14 <visible>1</visible>
15
16 <objectClasses>
17 <objectClass id="organizationalUnit"></objectClass>
18 </objectClasses>
19
20 <attributes>
21 <attribute id="ou">
22   <display>Organisational Unit</display>
23   <hint>don't include "ou="</hint>
24   <order>1</order>
25   <page>1</page>
26 </attribute>
27 </attributes>
28
29 </template>
```

```

1 /etc/phpldapadmin/templates/modification/custom_user.xml
2
3 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
4 <!DOCTYPE template SYSTEM "template.dtd">
5
6 <template>
7 <askcontainer>1</askcontainer>
8 <description>New Custom Eclair User Account</description>
9 <icon>ldap-user.png</icon>
10 <invalid>0</invalid>
11 <rdn>cn</rdn>
12 <regexp>^cn=[a-zA-Z0-9]*,ou=users</regexp>
13 <title>Custom: Eclair Account</title>
14 <visible>1</visible>
15
16 <objectClasses>
17 <objectClass id="account"></objectClass>
18 <objectClass id="posixAccount"></objectClass>
```

```

19 <objectClass id="top"></objectClass>
20 <objectClass id="shadowAccount"></objectClass>
21 <objectClass id="ldappublicKey"></objectClass>
22 </objectClasses>
23
24 <attributes>
25 <attribute id="cn">
26   <display>Common Name</display>
27   <icon>ldap-user.png</icon>
28   <onchange>=autoFill(uid;%cn%)</onchange>
29   <order>1</order>
30 </attribute>
31 <attribute id="uid">
32   <display>login</display>
33   <icon>ldap-user.png</icon>
34   <onchange>=autoFill(homeDirectory;/home/users/%uid%)</onchange>
35   <order>2</order>
36 </attribute>
37 <attribute id="gecos">
38   <display>Full name, position</display>
39   <icon>ldap-user.png</icon>
40   <spacer>1</spacer>
41   <order>3</order>
42 </attribute>
43
44 <attribute id="uidNumber">
45   <display>UID Number</display>
46   <icon>terminal.png</icon>
47   <readonly>1</readonly>
48   <value>=php.GetNextNumber(/;uidNumber;;;;1100)</value>
49   <order>4</order>
50 </attribute>
51 <attribute id="gidNumber">
52   <display>GID Number</display>
53   <icon>terminal.png</icon>
54   <readonly>1</readonly>
55   <value>=php.GetNextNumber(/;gidNumber;;;;1100)</value>
56   <order>5</order>
57 </attribute>
58 <attribute id="loginShell">
59   <display>Login shell</display>
60   <icon>terminal.png</icon>
61   <type>select</type>
62   <value id="/bin/bash">/bin/bash</value>
63   <value id="/bin/sh">/bin/sh</value>
64   <default>/bin/bash</default>
65   <order>6</order>
66 </attribute>
67 <attribute id="homeDirectory">
```

```
68    <display>Home directory</display>
69    <icon>terminal.png</icon>
70    <spacer>1</spacer>
71    <order>7</order>
72  </attribute>
73
74
75  <attribute id="userPassword">
76    <display>Password</display>
77    <helper>
78      <display>Encryption</display>
79      <id>enc</id>
80      <!--<value>=php.PasswordEncryptionTypes( )</value>-->
81      <value>md5</value>
82    </helper>
83    <icon>lock.png</icon>
84    <post>=php.PasswordEncrypt(%enc%;%userPassword%)</post>
85    <verify>1</verify>
86    <order>8</order>
87  </attribute>
88  <attribute id="shadowLastChange">
89    <display>Shadow Last Change (days since epoch)</display>
90    <icon>lock.png</icon>
91    <readonly>1</readonly>
92    <value>=php.Function(daysSinceEpoch; )</value>
93    <order>9</order>
94  </attribute>
95  <attribute id="shadowMax">
96    <display>Shadow Max (days)</display>
97    <icon>lock.png</icon>
98    <readonly>1</readonly>
99    <value>99999</value>
100   <order>10</order>
101  </attribute>
102  <attribute id="shadowWarning">
103    <display>Shadow Warning (days)</display>
104    <icon>lock.png</icon>
105    <readonly>1</readonly>
106    <value>7</value>
107    <order>11</order>
108  </attribute>
109  <attribute id="sshpublicKey">
110    <display>SSH Public Key</display>
111    <icon>lock.png</icon>
112    <order>12</order>
113  </attribute>
114 </attributes>
115
116 </template>
```

```

1 /etc/phpldapadmin/templates/creation/custom_user.xml
2
3 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
4 <!DOCTYPE template SYSTEM "template.dtd">
5
6 <template>
7 <askcontainer>1</askcontainer>
8 <description>New Custom Eclair User Account</description>
9 <icon>ldap-user.png</icon>
10 <invalid>0</invalid>
11 <rdn>cn</rdn>
12 <regexp>^ou=users</regexp>
13 <title>Custom: Eclair Account</title>
14 <visible>1</visible>
15
16 <objectClasses>
17 <objectClass id="account"></objectClass>
18 <objectClass id="posixAccount"></objectClass>
19 <objectClass id="top"></objectClass>
20 <objectClass id="shadowAccount"></objectClass>
21 <objectClass id="ldappublicKey"></objectClass>
22 </objectClasses>
23
24 <attributes>
25 <attribute id="cn">
26   <display>Common Name</display>
27   <icon>ldap-user.png</icon>
28   <onchange>=autoFill(uid;%cn%)</onchange>
29   <order>1</order>
30 </attribute>
31 <attribute id="uid">
32   <display>login</display>
33   <icon>ldap-user.png</icon>
34   <onchange>=autoFill(homeDirectory;/home/users/%uid%)</onchange>
35   <order>2</order>
36 </attribute>
37 <attribute id="gecos">
38   <display>Full name, position</display>
39   <icon>ldap-user.png</icon>
40   <spacer>1</spacer>
41   <order>3</order>
42 </attribute>
43
44 <attribute id="uidNumber">
45   <display>UID Number</display>
46   <icon>terminal.png</icon>
47   <readonly>1</readonly>
48   <value>=php.GetNextNumber(/;uidNumber;;;;1100)</value>
49   <order>4</order>

```

```

50 </attribute>
51 <attribute id="gidNumber">
52   <display>GID Number</display>
53   <icon>terminal.png</icon>
54   <readonly>1</readonly>
55   <value>=php.GetNextNumber(/;gidNumber;;;;1100)</value>
56   <order>5</order>
57 </attribute>
58 <attribute id="loginShell">
59   <display>Login shell</display>
60   <icon>terminal.png</icon>
61   <type>select</type>
62   <value id="/bin/bash">/bin/bash</value>
63   <value id="/bin/sh">/bin/sh</value>
64   <default>/bin/bash</default>
65   <order>6</order>
66 </attribute>
67 <attribute id="homeDirectory">
68   <display>Home directory</display>
69   <icon>terminal.png</icon>
70   <spacer>1</spacer>
71   <order>7</order>
72 </attribute>
73
74
75 <attribute id="userPassword">
76   <display>Password</display>
77   <helper>
78     <display>Encryption</display>
79     <id>enc</id>
80     <!--<value>=php.PasswordEncryptionTypes()</value>-->
81     <value>md5</value>
82   </helper>
83   <icon>lock.png</icon>
84   <post>=php.PasswordEncrypt(%enc%;%userPassword%)</post>
85   <verify>1</verify>
86   <order>8</order>
87 </attribute>
88 <attribute id="shadowLastChange">
89   <display>Shadow Last Change (days since epoch)</display>
90   <icon>lock.png</icon>
91   <readonly>1</readonly>
92   <value>=php.Function(daysSinceEpoch; )</value>
93   <order>9</order>
94 </attribute>
95 <attribute id="shadowMax">
96   <display>Shadow Max (days)</display>
97   <icon>lock.png</icon>
98   <readonly>1</readonly>

```

```

99    <value>99999</value>
100   <order>10</order>
101  </attribute>
102  <attribute id="shadowWarning">
103    <display>Shadow Warning (days)</display>
104    <icon>lock.png</icon>
105    <readonly>1</readonly>
106    <value>7</value>
107    <order>11</order>
108  </attribute>
109  <attribute id="sshpublicKey">
110    <display>SSH Public Key</display>
111    <icon>lock.png</icon>
112    <order>12</order>
113  </attribute>
114 </attributes>
115
116 </template>

```

```

1 /etc/phpldapadmin/templates/creation/custom_user.xml
2
3 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
4 <!DOCTYPE template SYSTEM "template.dtd">
5
6 <template>
7 <askcontainer>1</askcontainer>
8 <description>New Custom Eclair Group</description>
9 <icon>ldap-ou.png</icon>
10 <invalid>0</invalid>
11 <rdn>cn</rdn>
12 <regexp>^ou=groups</regexp>
13 <title>Custom: Eclair Group</title>
14 <visible>1</visible>
15
16 <objectClasses>
17 <objectClass id="posixGroup"></objectClass>
18 </objectClasses>
19
20 <attributes>
21 <attribute id="cn">
22   <display>Group Name</display>
23   <order>1</order>
24   <page>1</page>
25 </attribute>
26 <attribute id="gidNumber">
27   <display>GID Number</display>
28   <order>2</order>
29   <page>1</page>
30   <readonly>1</readonly>

```

```

31   <spacer>1</spacer>
32   <value>=php.GetNextNumber(/;gidNumber;;;;9500)</value>
33 </attribute>
34 </attributes>
35
36 </template>
```

Il faudra ensuite ajouter un premier utilisateur à la main sous la forme :

```

1 # nlascoux, users, eclair.ec-lyon.fr
2 dn: cn=nlascoux,ou=users,dc=eclair,dc=ec-lyon,dc=fr
3 cn: nlascoux
4 uid: nlascoux
5 uidNumber: 1117
6 gidNumber: 1117
7 loginShell: /bin/bash
8 homeDirectory: /home/users/nlascoux
9 shadowLastChange: 18916
10 shadowMax: 99999
11 shadowWarning: 7
12 objectClass: account
13 objectClass: posixAccount
14 objectClass: top
15 objectClass: shadowAccount
16 objectClass: ldapPublicKey
17 sshPublicKey: ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQDmVb1LbpTuKkxvtaq3elHWYI4v0ALW/OhTDB5/uJE/4wbVOolwKSA
ZVDgtOnMXHmSspWc3PbM28NGDTth81678/m//b4IZ8hyX/RowMDaeHjVuX8MDyN/HGbT/+PiLgLtOkL/sgM
Eeem9Kyym9WkUgZJstYcXXi/074cljkoE5e+nyjVkzHmEw5x33cFPYTd7a7OV4AAehVptDau/7KyMQNc7Rf
y4JODNmT9H4bzL5SMnYvx5DrGdy5vtOE5WL1V2oKgPzU1ENndI96+nb1qKdz/TEUv2GdJ6+7YwARLN2+T+N
6WopiHqnI1MOUvr9P0MsPVNtzMtRTobkok0jlu0C4JsyXnzZyDirlvGQLLrfPJS4WDy7QtfKX04jdR1E1Jg
hGKTfTLP1D1xQW0BLSDYrQ1TRFK99AWemo9fEEzM7jyFoPa3LfOFOtv03G7b8pSJ9msYZHkuJkfFDVGvt4I
AvdCuCXTKOd+05jv3a+7uMGrvK/RUzprl06IBy0TVCWOfs/bn9TeJuvgqgf6uDB/e60rSTzL0poj4FoDUN
Vpcl20eMD8ydIQeKzCUP7kXm3qEJNvNVBp0ydeuBt8SFKXZ2H3xVPlgvGNb3N1NkFjjRUYh1m6crpI9BXfi
82vpqmYY8wHdt9kDWVh1oGdmiebdh5FyEN4u4bzFZ31Ex4FCw== MBNath@MBPNath.local
```

Puis avec la commande :

```
1 | >> sudo ldapadd -xWD cn=admin,dc=eclair,dc=ec-lyon,dc=fr -f user.ldif
```

# Annexe 2D

## Remise en place du proxy SSH

---

On copie simplement le fichier de configuration SSH de l'ancien eole (`/etc/ssh/sshd_config`):

```

1  # $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $
2
3  # This is the sshd server system-wide configuration file. See
4  # sshd_config(5) for more information.
5
6  # This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
7
8  # The strategy used for options in the default sshd_config shipped with
9  # OpenSSH is to specify options with their default value where
10 # possible, but leave them commented. Uncommented options override the
11 # default value.
12
13 Port 22
14 #AddressFamily any
15 #ListenAddress 156.18.24.2
16 #ListenAddress ::

17 HostKey /etc/ssh/ssh_host_rsa_key
18 HostKey /etc/ssh/ssh_host_dsa_key
19 HostKey /etc/ssh/ssh_host_ecdsa_key
20 HostKey /etc/ssh/ssh_host_ed25519_key

22
23 # Ciphers and keying
24 #RekeyLimit default none

25
26 # Logging
27 SyslogFacility AUTH
28 LogLevel INFO

29
30 # Authentication:

31 LoginGraceTime 2m
32 #PermitRootLogin without-password
33 #PermitRootLogin no
34 StrictModes yes
35 #MaxAuthTries 6
36 #MaxSessions 10

38
39 #RSAAuthentication yes
40 PubkeyAuthentication yes

41
42 # Expect .ssh/authorized_keys2 to be disregarded by default in future.
43 #AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
44

```

```
45 #AuthorizedPrincipalsFile none
46
47 #AuthorizedKeysCommand none
48 #AuthorizedKeysCommandUser nobody
49
50 # For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
51 #HostbasedAuthentication no
52 # Change to yes if you don't trust ~/.ssh/known_hosts for
53 HostbasedAuthentication no
54 #IgnoreUserKnownHosts no
55 # Don't read the user's ~/.rhosts and ~/.shosts files
56 #IgnoreRhosts yes
57
58 # To disable tunneled clear text passwords, change to no here!
59 PasswordAuthentication no
60 PermitEmptyPasswords no
61
62 # Change to yes to enable challenge-response passwords (beware issues with
63 # some PAM modules and threads)
64 ChallengeResponseAuthentication no
65
66 # Kerberos options
67 #KerberosAuthentication no
68 #KerberosOrLocalPasswd yes
69 #KerberosTicketCleanup yes
70 #KerberosGetAFSToken no
71
72 # GSSAPI options
73 #GSSAPIAuthentication no
74 #GSSAPICleanupCredentials yes
75 #GSSAPIStrictAcceptorCheck yes
76 #GSSAPIKeyExchange no
77
78 AllowAgentForwarding yes
79 AllowTcpForwarding yes
80 #GatewayPorts no
81 X11Forwarding no
82 X11DisplayOffset 10
83 #X11UseLocalhost yes
84 #PermitTTY yes
85 PrintMotd no
86 PrintLastLog yes
87 TCPKeepAlive yes
88 #UseLogin no
89 #UsePrivilegeSeparation sandbox
90 #PermitUserEnvironment no
91 #Compression delayed
92 #ClientAliveInterval 0
93 #ClientAliveCountMax 3
```

```

94 #UseDNS no
95 #PidFile /var/run/sshd.pid
96 #MaxStartups 10:30:100
97 PermitTunnel yes
98 #ChrootDirectory none
99 #VersionAddendum none
100
101 # no default banner path
102 #Banner none
103
104 # Allow client to pass locale environment variables
105 AcceptEnv LANG LC_*
106
107 # override default of no subsystems
108 Subsystem sftp /usr/lib/openssh/sftp-server
109
110
111 # Set this to 'yes' to enable PAM authentication, account processing,
112 # and session processing. If this is enabled, PAM authentication will
113 # be allowed through the ChallengeResponseAuthentication and
114 # PasswordAuthentication. Depending on your PAM configuration,
115 # PAM authentication via ChallengeResponseAuthentication may bypass
116 # the setting of "PermitRootLogin without-password".
117 # If you just want the PAM account and session checks to run without
118 # PAM authentication, then enable this but set PasswordAuthentication
119 # and ChallengeResponseAuthentication to 'no'.
120 UsePAM yes

```

On crée aussi un dossier pour les utilisateurs : `/home/users`

Pour créer les sessions utilisateurs on crée un script dans `/home/create_sessions.sh` :

```

1 # Script qui cree un une session correctement et definit un mot de passe
2 # Pour l'utiliser, taper /home/create_sessions.sh $USERNAME
3 # Attention : il faut un compte sudo ou être en root
4
5 sudo useradd -m -d /home/users/$1 -s /bin/bash $1
6
7 cd /home/users/$1
8 sudo mkdir .ssh
9 sudo touch .ssh/authorized_keys
10
11 read -p "SSH key : " KEY
12 sudo echo $KEY >> .ssh/authorized_keys
13
14 echo "Penser à creer un mdp pour la session avec \\"sudo passwd $1\\\""

```

## Annexe 2E

# Création d'une VM

## Création de la VM

- Commencer par se connecter sur le réseau éclair, puis accéder à l'interface web de l'hyperviseur : `https://156.18.24.201:8006`. Se connecter, les credentials sont trouvables sur le gestionnaire de mot de passe éclair.
- Cliquer sur `create VM` en haut à droite de la fenêtre
- Remplir les informations :
  - L'ID de la VM correspondra à 100 + dernier chiffre de l'IP de la machine (exemple pour eole .2, id 102)
  - Remplir ensuite les champs selon nécessaire jusqu'à `Network`. La plupart des champs sont à laisser par défaut, il suffit de sélectionner l'OS, et d'indiquer la taille du disque, le nombre de coeurs et la RAM disponible.
  - Dans `Network`, choisir `vmbro0` pour le bridge et laisser `no VLAN` pour le tag.
  - Cliquer sur `Finish` dans la dernière page

## Configuration de base (réseau et sudo)

- Lancer la VM et installer l'OS en entier (après la spécification du nom de la machine on indiquera `eclair.ec-lyon.fr` pour le domaine de la VM)
- Une fois l'OS installé on va pouvoir se connecter en root pour installer `sudo` et donner des droits sudo à l'utilisateur principal
- Modifier le fichier `/etc/network/interfaces` et y mettre cette configuration (pour la VM eole, de numéro 2) :

```

1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 source /etc/network/interfaces.d/*
5
6 # The loopback network interface
7 auto lo
8 iface lo inet loopback
9
10 # The primary network interface
11 auto ens18
12 iface ens18 inet static
13   address 156.18.24.2/21
14   gateway 156.18.31.254
15   dns-nameservers 156.18.24.1 156.18.19.5 156.18.22.3
16   dns-search eclair.ec-lyon.fr

```

**Attention :** vérifier que `resolvconf` soit bien installé pour que le DNS se configure correctement, cela avant de relancer la VM avec la nouvelle config.

- On éteint ensuite la VM (pas besoin de redémarrer le réseau) avec `sudo shutdown now`
- On se rend alors dans le hardware de la VM pour :
  - Remove l'adaptateur CD / DVD
  - Ajouter un `Network device` avec `vmbr1`, toujours sans tag de VLAN
- On relance la VM
- On ajoute alors à la fin du fichier `/etc/network/interfaces` (encore pour eole) :

```
1 auto ens19
2 iface ens19 inet static
3   address 10.18.24.2/24
4   dns-nameservers 10.18.24.1
5   dns-search dmz.eclair.ec-lyon.fr
```

On peut finalement redémarrer la VM et vérifier que tout fonctionne.

Si ce n'est pas le cas, vérifier que les noms des interfaces réseau correspondent à ceux indiqués dans le fichier de config.

# Server Dell R440

## 8SFF

REF



(11)853667731



5G1D8K3



Full description

- 2 x Intel Xeon Gold 6126 (12C 19.25M Cache 2.60 GHz)
- 4 x 16GB DDR4 RDIMM 2666MHz
- 8 x Tray Caddy 2.5"
- Dell PowerEdge RAID Controller H730 (1Gb+FBWC)
- iDRAC 9 Enterprise
- Broadcom 5720 DP 1Gb Network Interface Card
- 2 x Power supply 550W Hot Plug
- Rack mount kit 19"
- 5 years warranty from Servermall



Dimensions: 43x649x434

Weight: 14 kg

Official warranty: 5 years warranty from Servermall

Price **5 330,-**Price excl.VAT **4 405,-**

### Features

The Dell EMC PowerEdge R440 is 2-socket, 1U rack server designed to run complex workloads using highly scalable memory, I/O, and network options. The systems feature the Intel Xeon Scalable Processor family (Skylake-EP), up to 16 DIMMs, PCI Express (PCIe) 3.0 enabled expansion slots, and a choice of network interface technologies to cover NIC and OCP.

The PowerEdge R440 is a general-purpose platform capable of handling demanding workloads and applications, such as high-performance computing, web tech, infrastructure scale out, surveillance and site security. The PowerEdge R440 adds new storage capacity options, making it well-suited for software defined storage and data-intensive applications that require greater storage, while not sacrificing I/O performance.

**5**  
year

Warranty



Free shipping



Onsite warranty

Up to 70%  
cheaper

# Annexe 2G

## ProxMox

### Installation et configuration

#### Installation :

- Télécharger l'image ISO (celle nommée Proxmox VE, pour Virtual Environment) depuis [le site de proxmox](#)
- Faire une clé bootable
  - Raspberry pi imager
  - Utilitaire fdisk ?
  - Balena etcher sur mac ?
- Brancher la clé sur la machine
- Démarrer la machine et accéder au bios (voir touche en fonction de la machine)
- Sélectionner la clé comme device de boot (attention, si on ne boot pas en UEFI, il est possible que la machine ne voit pas la clé)

#### Configuration (sur la machine) :

- Config de base : créer un mdp, rentrer un mail...
- Config réseau : adresse, hostname, gateway, DNS...
- Config disques durs, création de partitions...

Une fois configurée, la machine va reboot. Il faut alors réaccéder au bios pour sélectionner le bon disque de démarrage. Finalement, on peut se connecter à l'interface web de la machine via `http://$IPaddress:8006`.

#### Première VM

Il faut commencer par upload une image ISO : on va dans le disque local, dans `ISO Images`, puis on clique sur upload. Pour faire les choses proprement, on va ensuite créer une seconde partition logique `vms`, qui acceptera les images disque (important). Finalement, la dernière étape avant de créer la VM est de créer un bridge réseau. Ceci se fait en allant dans la node, puis réseau et créer.

#### Gestion des VM en ligne de commande

Si l'on arrive pas à accéder à l'interface web (qui semble d'une stabilité modérée), on pourra se SSH sur la machine hôte, et utiliser la commande `qm`. Il suffit de la taper pour voir les commandes possibles, mais la plus utile sera de lancer une VM avec `qm start $VM_ID`. Si le réseau des machines est correctement configuré on pourra alors directement se SSH dessus. C'est une bonne solution car le SSH semble plus stable que l'interface web à travers VNC.

#### Création de snapshots

On commence par créer une partition logique qui acceptera les backups. Il suffit ensuite de se placer dans la section "backup" d'une VM, puis de cliquer sur `backup now`. La plupart des options par défaut fonctionnent très bien.

Attention cependant le fonctionnement des snapshots sur Proxmox est étrange, quand on veut rétablir une snapshot, il crée en fait une nouvelle VM à partir de l'image sauvegardée de la précédente.

## Configuration réseau avec des VLAN

---

La configuration réseau de Proxmox est assez ardue quand on veut le connecter à des VLANs, car les VM nécessitent d'être ajoutées sur des bridges. Il faut alors créer des bridges que l'on met sur un bond, en précisant qu'ils doivent être `vlan-aware`, et donner les numéros des VLAN. De plus, l'adresse IP doit être indiquée seulement sur les bridges, pas sur les interfaces ou les bonds. Le fichier de configuration final est le suivant :

```
1 auto lo
2 iface lo inet manual
3
4 auto enp5s0
5 iface enp5s0 inet manual
6 auto enp9s0
7 iface enp9s0 inet manual
8
9 auto bond0
10 iface bond0 inet manual
11   bond-slaves enp5s0 enp9s0
12   bond-mimon 100
13   bond-mode 802.3ad
14
15 auto bond0.10
16 iface bond0.10 inet manual
17 auto bond0.156
18 iface bond0.156 inet manual
19
20 auto vmbr0
21 iface vmbr0 inet manual
22   address 156.18.24.201/21
23   gateway 156.18.31.254
24   bridge-ports bond0.156
25   bridge-stp off
26   bridge-fd 0
27   bridge-vlan-aware yes
28   bridge-vids 156
29
30 auto vmbr1
31 iface vmbr1 inet manual
32   address 10.18.24.201/24
33   bridge-ports bond0.10
34   bridge-stp off
```

```
35 bridge-fd 0
36 bridge-vlan-aware yes
37 bridge-vids 10
```

# NGINX

## Annexe 2H

### Configuration

Il faut d'abord installer nginx : `sudo apt install nginx`

Les fichiers et les dossiers de configuration utilisés sont :

```
1 /etc/nginx/nginx.conf # fichier de configuration gloable du serveur
2 /etc/nginx/proxy_param
3 /etc/nginx/sites-enabled/ # liens symboliques vers les fichiers de site-available
4 /etc/nginx/sites-available/ # contient les fichiers de configurations de vos sites
ou services
5 /etc/nginx/mime.types
```

nginx.conf :

```
1 user www-data; # définit l'utilisateur et le groupe avec lequel le daemon Nginx
sera lancé
2 worker_processes 1; # (essayer "nombre auto" en param)
3 pid /run/nginx.pid;
4 include /etc/nginx/modules-enabled/*.conf;
5
6 events {
7     worker_connections 1024; # equal to "ulimit -n"
8     multi_accept on; # le processeur acceptera toutes les nouvelles connexions en
même temps
9 }
10
11 http {
12     ##
13     # Basic Settings
14     ##
15     server_tokens off; # active ou désactive l'émission de la version Nginx
16     charset utf-8;
17     sendfile on;
18     tcp_nopush on; # optimise la quantité d'informations envoyée
19     tcp_nodelay on; # optimiser les délais d'envoi des information
20
21     keepalive_timeout 10; # le premier paramètre est obligatoire définit un délai
pendant lequel une connexion cliente KeepAlive restera ouverte côté serveur
22     client_header_timeout 20; # définit la taille au delà de laquelle la requête sera
enregistrée dans un fichier
23     client_body_timeout 20; # taille max des données envoyées par un client
24     reset_timedout_connection on;
25     send_timeout 20;
26     types_hash_max_size 2048;
```

```
27 server_names_hash_bucket_size 64; # fix long domain name issue
28
29 include /etc/nginx/mime.types; # cette liste indique la nature des fichiers
30 renvoyées par le serveur
31 default_type application/octet-stream;
32 ##
33 # SSL Settings
34 ##
35 ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE (on
36 utilise les protocoles TLS car SSLv3 est plus vulnérable)
37 ssl_prefer_server_ciphers on;
38 ##
39 # Log Settings
40 ##
41 access_log off;
42 error_log /var/log/nginx/error.log; # dossier ou on récupère les logs
43
44 ##
45 # Gzip Settings # active la compression des pages sauf pour les navigateurs
46 outdated
47 ##
48 gzip on;
49 gzip_buffers 16 8k;
50 gzip_comp_level 6;
51 gzip_min_length 1400; # minimum MTU (POE) -> no compression if it feets in a frame
52 !
53 gzip_proxied expired no-cache no-store private auth;
54 gzip_vary on;
55 gzip_disable "msie6";
56 gzip_types # types des fichiers à compresser
57   application/atom+xml
58   application/javascript
59   application/json
60   application/ld+json
61   application/manifest+json
62   application/rss+xml
63   application/vnd.api+json
64   application/vnd.geo+json
65   application/vnd.ms-fontobject
66   application/x-font-ttf
67   application/x-javascript
68   application/x-web-app-manifest+json
69   application/xhtml+xml
70   application/xml
71   font/opentype
72   font/truetype
73   image/bmp
```

```

72 image/svg+xml
73 image/x-icon
74 text/cache-manifest
75 text/css
76 text/javascript
77 text/plain
78 text/vcard
79 text/vnd.rim.location.xloc
80 text/vtt
81 text/x-component
82 text/x-cross-domain-policy
83 text/xml;
84
85 ##
86 # Virtual Host Configs
87 ##
88 include /etc/nginx/conf.d/*.conf;
89 include /etc/nginx/sites-enabled/*;
90 }

```

proxy\_params : on paramètre les entêtes des paquets qui doivent être ajoutées par le reverse-proxy.

```

1 proxy_set_header Host $http_host;
2 proxy_set_header X-Real-IP $remote_addr;
3 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
4 proxy_set_header X-Forwarded-Proto $scheme;
5 proxy_set_header Upgrade $http_upgrade;
6 proxy_set_header Connection "upgrade";

```

etc/conf.d/ssl.conf

```

1 ssl_session_cache shared:SSL:20m;
2 ssl_session_timeout 10m;
3 ssl_buffer_size 1400;
4
5 ssl_certificate ssl/bundle.pem;
6 ssl_certificate_key ssl/star_eclair_ec-lyon_fr.key;
7
8 ssl_ciphers
"EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA256:EECDH+ECD
SA+SHA384:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA384:EDH+aRSA+AESGCM:EDH+aRSA+SHA256:EDH+a
RSA:EECDH:!aNULL:!eNULL:!MEDIUM:!LOW:!3DES:!MD5:!EXP:!PSK:!SRP:!DSS:!RC4:!SEED";

```

ex : sites-available/wiki (fichiers de configuration d'un service)

```

1 server {
2   listen 443 http2 ssl; ## listen for ipv4
3   server_name wiki.eclair.ec-lyon.fr wiki;

```

```
4 include error_location;
5
6 location / {
7     include proxy_params;
8     proxy_pass http://10.18.24.201:8080;
9 }
10 }
11
12 server {
13     listen 8080;
14     server_name wiki.eclair.ec-lyon.fr wiki;
15     return 301 https://wiki.eclair.ec-lyon.fr $request_uri;
16 }
```

Il faut ensuite activer le server block en créant un lien symbolique : `sudo ln -s /etc/nginx/sites-available/wiki /etc/nginx/sites-enabled/wiki`

Si le proxy Nginx et le serveur Apache sont sur des machines différentes, il faut indiquer l'adresse IP du proxy dans le fichier de configuration du module `/etc/apache2/mods-available/rpaf.conf`

# Annexe 2I

## Fichiers de configuration de Caddy et du SSLH

```
1 # docker-compose.yml
2
3 services:
4   sslh:
5     image: registry.eclair.ec-lyon.fr/sslh:latest
6     restart: unless-stopped
7     hostname: sslh
8     ports:
9       - "156.18.24.10:443:443"
10    command: --listen=0.0.0.0:443 --tls=caddy:443 --ssh=eole.eclair.ec-lyon.fr:22
11    volumes:
12      - ./config.cfg:/etc/sslh.cfg
13    depends_on:
14      - caddy
15
16  caddy:
17    image: caddy:2-alpine
18    restart: unless-stopped
19    ports:
20      - "156.18.24.10:80:80"
21      # - "156.18.24.10:443:443"
22      # - "156.18.24.10:443:443/udp"
23    volumes:
24      - ./Caddyfile:/etc/caddy/Caddyfile:ro
25      - ./data/caddy_data:/data
26    container_name: caddy
```

```
1 # Caddyfile
2
3 {
4   servers :443 {
5     protocol {
6       experimental_http3
7     }
8   }
9 }
10
11 (common) {
12   header /* {
13     -Server
14   }
15 }
```

```
17 # adresse du site {  
18 #   reverse_proxy 10.xxx.xxx.xxx:PORT  
19 # }
```

## Annexe 2J

# pfsense

L'infrastructure d'ÉCLAIR utilise un Parefeu : [pfsense](#).

Celui-ci est installé sur la machine Heimdall. Elle possède deux ports réseaux, un branché sur le switch d'entrée du réseau de la DS1. Le deuxième port est branché sur l'entrée du switch Orome (`s6`) et correspond à l'entrée de notre réseau.

L'accès à la console de la machine (un dérivé de FreeBSD sur lequel repose pfsense) n'est pas très utile au quotidien et permet de la restauration d'un fichier de config ainsi que les configs réseau.

L'administration du pare-feu se fait au travers de l'interface graphique de pfsence : <https://heimdall.eclair.ec-lyon.fr/> (les identifiants sont sur Bitwarden)

Trois interfaces sont configurées :

- LAN : réseau local, tout est autorisé par défaut
- WAN : réseau extérieur, tout est interdit par défaut
- BRIDGE0 : l'union de LAN et WAN (un *bridge* réalise le lien entre LAN et WAN)

C'est interfaces sont configurées sur les *Network port* suivants

- LAN : `bge1 (f8:bc:12:3d:f7:a8)`
- WAN : `bge0 (f8:bc:12:3d:f7:a6)`

On ajoute sur ces trois interfaces des règles. La majorités de nos règles sont à ajouter à BRIDGE0 afin de les appliquer au réseau local et externe.

## Sauvegarde et restauration

Il est facile de réaliser une sauvegarde ou de restorer celle-ci depuis l'interface web. Une nouvelle sauvegarde devrait être réalisée après chaque modification.

## Mise à jour

Il est possible de mettre à jour pfsense depuis l'interface web. Il suffit d'aller dans : [System/Update/System update](#)

Attention, il semblerait que le logiciel prétende parfois à tort qu'il est à jour. Il suffit alors de changer de branche de mise à jour (stable, déprécié ou expérimental) puis revenir. Nous avons mis à jour vers la dernière version .4.x

Lorsque la mise à jour échoue, il est facile de faire une sauvegarde de la config de pfsense et de le réinstaller.

## Installation

1. Réaliser une clef d'installation de pfsense ([télécharger une image](#))
2. Réaliser l'installation :

1. Choisir `Install`
2. Choisir `Clavier`
3. Choisir `BIOS` pour l'installation, `disque entier` puis `MBR`
4. Configure comme dns primaire l'ip de Prométhée

Il faut activer le DHCP lors de l'installation, celui-ci sera désactiver lors de la restauration d'une sauvegarde

3. Configuration des interfaces de base (nécessaire pour que la machine soit accessible depuis le réseau local, afin de restorer le fichier de config)
  1. Choisir `WAN (wan) -> bge0 -> 156.18.24.210/21` et `LAN (lan) -> bge1 -> 10.18.24.210/24`
  2. Se connecter à la nouvelle IP de Heimdall depuis un ordi branché en ethernet sur un port des switch
  3. Restaurer la sauvegarde

## Réalisation d'une règle

Attention : il faut toujours ajouter une **description** explicite de chaque règle

Note, pour faire une règle vers Steropes, il faut utiliser l'IP `156.18.24.10` et non `10.18.24.10`

Après une mauvaise manipulation, il est possible de restaurer depuis la console un état précédent. Il faut utiliser la commande `15`.

pfsense est munie d'une règle *anti-lockout* qui empêche de configurer une règle qui nous bloquerait hors de Heimdall.

Il sera nécessaire d'ajouter une règle lorsque nous voudrons mettre de nouveaux services sur des ports particuliers (autres que 80 et 443)

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<code>302 /2.48 GiB</code>	IPv4 TCP	BRIDGE0 net	*	*	*	*	none		Accéder à tout les protocoles, depuis la LAN, vers l'extérieur	
<input type="checkbox"/>	<code>57 /15.25 MiB</code>	IPv4 *	156.18.16.0/20	*	156.18.16.0/20	*	*	none		Traffic Centrale avec les sites courants de la DSi	
<input type="checkbox"/>	<code>0 /32 KIB</code>	IPv4 TCP	*	*	156.18.24.10	80 (HTTP)	*	none		Ouverture du port 80, pour le http, vers Steropes	
<input type="checkbox"/>	<code>1 /1.58 MiB</code>	IPv4 TCP	*	*	156.18.24.10	443 (HTTPS)	*	none		Ouverture du port 443, pour le https, vers Steropes	
<input type="checkbox"/>	<code>0 /0 B</code>	IPv4 ICMP any	*	*	*	*	*	none		Autoriser le protocole ICMP (ping, message d'erreur) partout (a réactiver au besoin)	
<input type="checkbox"/>	<code>0 /0 B</code>	IPv4 UDP	*	*	156.18.24.10	53 (DNS)	*	none		Ouverture du port 53, pour le DNS, vers Steropes	

Add   Add   Delete   Save   Separator

Par défaut, toutes les requêtes devraient être bloquées.

- On ouvre les requêtes vers Steropes sur les ports 53, 80 et 443
- On ouvre les connexions sur tous les ports depuis le réseau local : cela permet d'accéder à internet depuis le M16 et depuis les machines
- On accepte spécifiquement les connexions vers les machines de la DSI

**Check-list de rapport de Projet d'Etudes**  
**A remplir par les rédacteurs (élèves)**  
**et à insérer en dernière page du rapport**

**A développer**

Renseigner la case par le nom du responsable, ou la date ou une simple croix lorsque la vérification a été faite.

Vérification présence	Vérification qualité
-----------------------	----------------------

**Contenu**

Résumé en français	X	X
Résumé en anglais	X	X
Table des matières	X	X
Table des figures	X	X
Introduction	X	X
Conclusion générale	X	X
Bibliographie	X	X
Citation des références dans le texte	X	X

**Forme**

Vérification orthographe	X	X
Pagination	X	X
Homogénéité de la mise en page	X	X
Lisibilité des figures	X	X