



**CAMPUS
DES MÉTIERS
ET DES
QUALIFICATIONS**

**Informatique et électronique
de demain**

Auvergne-Rhône-Alpes

Rapport de stage

BTS SIO 1

CMQ IED – 19 mai -> 20 juin

RAPPORT DE STAGE

Ce document présentera l'objectif d'un stage en tant qu'étudiant en BTS SIO ainsi que les missions réalisées lors de ce dernier.

LOPES Raphael

Remerciements :

Je tiens à remercier Adlen Saadi, mon maître de stage pour m'avoir donné l'opportunité d'être stagiaire au CMQ IED ce qui a permis de valider ma première année de BTS. Je remercie aussi l'ensemble de l'équipe pour leur accueil et leur bienveillance.

Introduction : Objectif du stage

Pour valider la première année du BTS Services Informatiques aux Organisations (SIO), un stage en entreprise est obligatoire. Ce stage, qui dure entre 5 et 6 semaines, permet de se plonger dans le monde professionnel pour appliquer les compétences apprises en cours, comme la gestion de réseaux ou le développement d'applications. Il aide aussi à développer des savoir-faire pratiques et à mieux comprendre le métier d'informaticien tout en découvrant le fonctionnement d'une entreprise. C'est dans ce cadre que j'ai réalisé mon stage au CMQ IED, du 19 mai au 20 juin, où j'ai pu participer à des tâches concrètes en informatique et explorer les réalités de mon futur métier.

CMQ IED-AURA :

Le Campus des Métiers et des Qualifications Informatique et Électronique de Demain (CMQ IED-AURA), situé à Valence, est un réseau éducatif dédié à la formation en informatique et électronique. Basé principalement dans la Drôme il regroupe des lycées, centres de formation (CFA), universités et entreprises pour préparer les étudiants aux métiers du numérique, comme la programmation, la cybersécurité ou les réseaux. En tant que 2ème région de France pour l'informatique et l'électronique, Auvergne-Rhône-Alpes bénéficie d'un écosystème innovant que le CMQ IED-AURA soutient activement.

Le CMQ IED-AURA se concentre sur l'éducation informatique en créant des mallettes pédagogiques, conçues par des ingénieurs pédagogiques pour être utilisées dans les collèges et lycées. Ces ressources numériques et pratiques visent à enseigner l'informatique de manière accessible. Des médiateurs facilitent la collaboration entre les ingénieurs, les établissements scolaires et l'Éducation nationale pour déployer ces outils. Le campus porte aussi des projets innovants, comme Parallaxe 2050, un escape game mobile pour promouvoir les métiers du numérique auprès des jeunes, et Le Lab du Campus, un espace d'échange et d'idéation.

Durant mon stage, du 19 mai au 20 juin 2025, j'ai rejoint le service informatique du CMQ IED. Ce service, composé d'une petite équipe de 8, gère les plateformes numériques et les outils technologiques soutenant les projets pédagogiques, contribuant ainsi à la formation aux métiers de demain.

Voici un schéma montrant l'ensemble de l'équipe et leur rôle :



Organigramme EQUIPE

Campus des Métiers et des Qualifications Informatique et Electronique de Demain [CMQ IED_AURA]

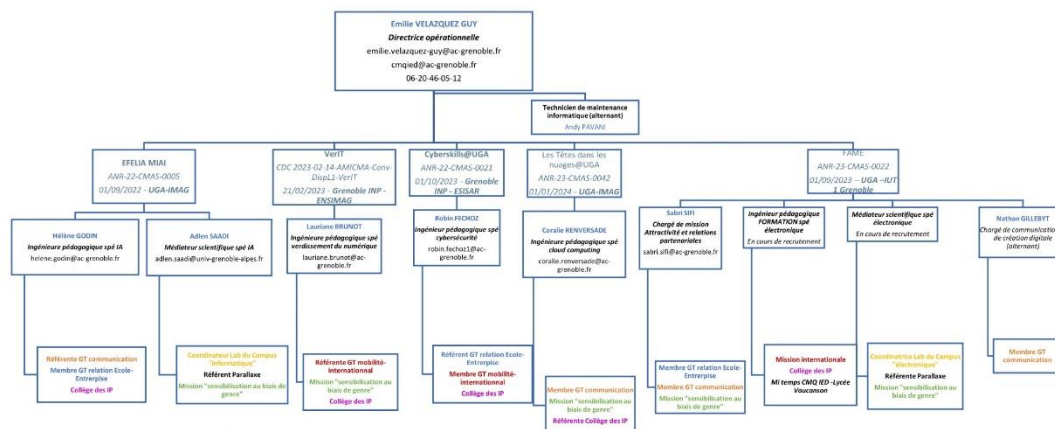
Thierry FEUTRY
Directeur du CMQ IED AURA
Proviseur Lycée polyvalent Algoud Laffemas-
Valence

Inspecteur d'Académie - Inspecteur Pédagogique
Régional
Sciences et Techniques Industrielles &
Technologie
Académie de Grenoble

Jean François MARION
Directeur du GIP FIPAG

Franck LENOIR
Secrétaire général du GIP FIPAG

Xavier OECHSLIN
Service des Affaires Générales du GIP FIPAG



Lab du Campus
09-74-85-52-74
09-78-80-01-01
07-68-49-27-23

[Page LinkedIn du Campus](https://www.linkedin.com/company/campus-des-metiers-et-des-qualifications)
<http://www.cmquied.fr/>

Lycée polyvalent Algoud-Laffemas
37-39 Rue Barthélémy de Laffemas BP 26 26901 VALENCE CEDEX 9

Note : Andy n'est plus présent, et Laurianne a été remplacé par Charlyne.

Mission 1 : Scan réseau et Inventaire informatique

Mon stage a commencé le 19 mai, après le départ d'Andy, qui était l'alternant qui s'occuper du réseau. J'ai donc demandé à Robin si Andy avait laissé un plan du réseau, un inventaire ou autre trace de son travail. Malheureusement il n'avait rien laissé.

J'ai donc repris le poste admin et installé les outils nécessaires au monitoring du réseau, comme Nmap et aussi une VM Kali Linux pour avoir la totalité des outils réseaux préinstallé.

J'ai fait un scan Zenmap (version graphique de Nmap) avec la commande suivante :

nmap -sS -sV -T4 -O -v -n 192.168.10.0/24

-O = OS détection.

-sSS = Scan TCP SYN.

-sV = Détection des versions de service.









-n = Désactiver la résolution DNS.

-T4 = Pour signaler que le réseau est stable.

Je vais lister tous les appareils que j'ai détecter sur le réseau grâce aux scans :

Liste équipements :

192.168.10.1 (PC-Admin) :

Nmap Output		Ports / Hosts		Topology	Host Details	Scans
	Port	Protocol	State	Service	Version	
	135	tcp	open	msrpc	Microsoft Windows RPC	
	139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn	
	445	tcp	open	microsoft-ds		
	2179	tcp	open	vmrpd		
	3000	tcp	open	http	Uvicorn	
	5060	tcp	open	sip	MicroSIP sipd 3.21.6	
	5061	tcp	open	sip-tls		
	5357	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)	

192.168.10.50 (Imprimante) :

Nmap Output					
Ports / Hosts					
Topology					
Host Details					
Scans					
	Port	Protocol	State	Service	Version
●	80	tcp	open	http	
●	443	tcp	open	https	
●	515	tcp	open	printer	
●	631	tcp	open	ipp	
●	9100	tcp	open	jetdirect	


192.168.10.100 (Switch GS1915-8) :


<http://192.168.10.100/login.html>

ZYXEL

GS1915-8


Enter User Name/Password and click to login.






Login

The Switch is being managed by Nebula.
Please use [the local credential password on NCC](#) to login.



Visit Nebula for Your Network Management.

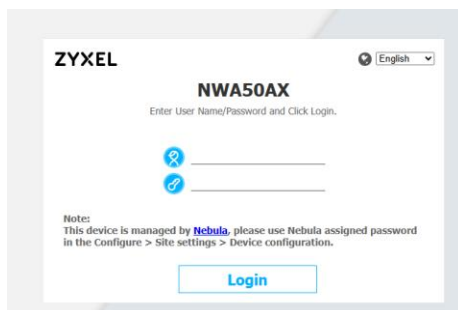
Go Now











Nmap Output					
Ports / Hosts					
Topology					
Host Details					
Scans					
	Port	Protocol	State	Service	Version
●	22	tcp	open	ssh	OpenSSH 3.9p1 (protocol 2.0)
●	23	tcp	open	telnet	HP Integrated Lights Out telnetd
●	80	tcp	open	http	Allegro-Software-RomPager
●	443	tcp	open	tcpwrapped	

192.168.10.101 (Box WiFi NWA50AX) :

<http://192.168.10.101/>



The image shows the login page of a ZYXEL NWA50AX device. At the top left is the ZYXEL logo. To the right is a language dropdown menu set to 'English'. Below the logo, the model 'NWA50AX' is displayed, followed by the instruction 'Enter User Name/Password and Click Login.' There are two input fields: one for the username (indicated by a key icon) and one for the password (indicated by a key icon with a slash). Below these fields is a 'Login' button. A note at the bottom states: 'Note: This device is managed by Nebula, please use Nebula assigned password in the Configure > Site settings > Device configuration.'


Nmap Output					
Ports / Hosts					
Topology					
Host Details					
Scans					
	Port	Protocol	State	Service	Version
	21	tcp	open	ftp	
	22	tcp	open	ssh	ZyXEL ZyWALL sshd (protocol 2.0)
	80	tcp	open	http	lighttpd
	82	tcp	filtered	xfer	
	83	tcp	filtered	mit-ml-dev	
	84	tcp	filtered	ctf	
	443	tcp	open	http	lighttpd
	10082	tcp	filtered	amandaidx	


192.168.10.102 (Switch GS1915-24EP) :

ZYXEL

GS1915-24EP


Enter User Name/Password and click to login.






Login





The Switch is being managed by Nebula.
Please use [the local credential password on NCC](#) to login.

nebula

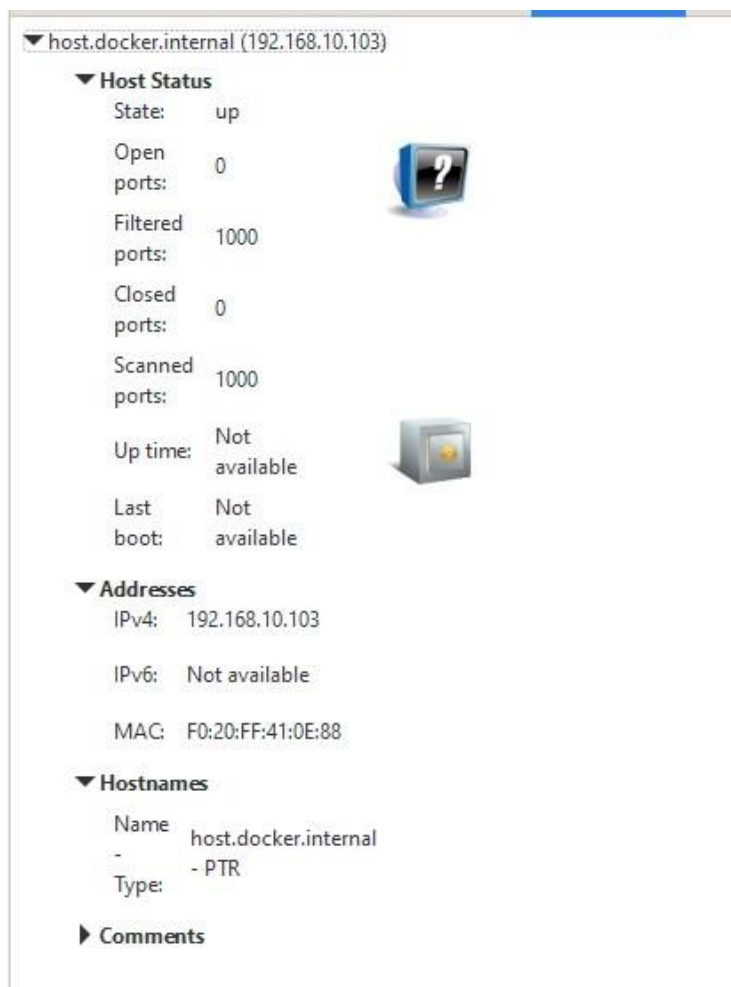
Visit Nebula for Your Network Management.

[Go Now](#)




Nmap Output		Ports / Hosts		Topology	Host Details	Scans
	Port	Protocol	State	Service	Version	
	22	tcp	open	ssh	OpenSSH 3.9p1 (protocol 2.0)	
	23	tcp	open	telnet	HP Integrated Lights Out telnetd	
	80	tcp	open	http	Allegro-Software-RomPager	
	443	tcp	open	tcpwrapped		

192.168.10.103 (host.docker.internal)













Très peu d'information sur cette machine, pas de port ouvert.

192.168.10.104 (Windows) :

Nmap Output					
Ports / Hosts					
Topology					
Host Details					
Scans					
	Port	Protocol	State	Service	Version
	3000	tcp	open	http	Node.js Express framework

Le site est inaccessible.

192.168.10.107 (Android TV) :

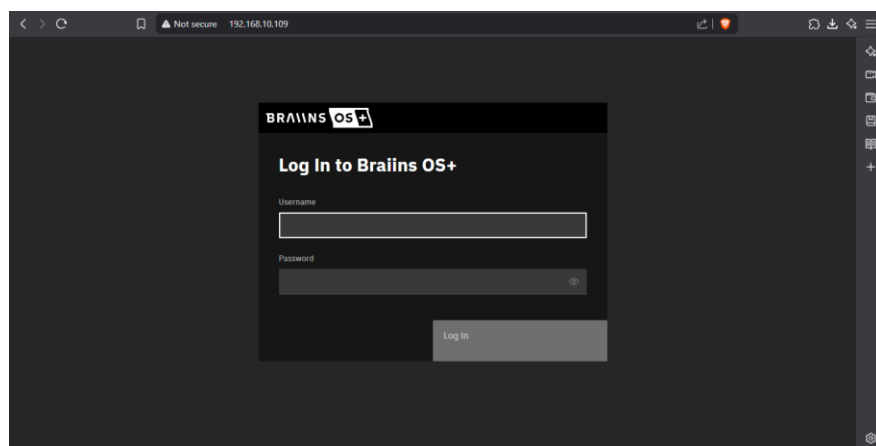
Nmap Output		Ports / Hosts		Topology	Host Details	Scans
	Port	Protocol	State	Service		Version
	30	tcp	filtered			
	2190	tcp	filtered	tivoconnect		
	3003	tcp	filtered	cgms		
	3880	tcp	filtered	igrs		
	8008	tcp	open	http		
	8009	tcp	open	ajp13		
	8443	tcp	open	https-alt		
	9000	tcp	open	cslistener		
	10000	tcp	filtered	snet-sensor-mgmt		
	12345	tcp	filtered	netbus		

192.168.10.108 (Inconnu, all TCP port down) :

▼ 192.168.10.108	
▼ Host Status	
State:	up
Open ports:	2
Filtered ports:	0
Closed ports:	998
Scanned ports:	1000
Up time:	Not available
Last boot:	Not available
▼ Addresses	
IPv4:	192.168.10.108
IPv6:	Not available
MAC:	86:3B:85:FA:FD:C9
► Comments	

192.168.10.109 (OpenWrt Chaos Calmer)

OS pour système embarquer, dans notre cas c'est un ASIC (mineur bitcoin)



Nmap Output					
Ports / Hosts			Topology	Host Details	Scans
Port	Protocol	State	Service	Version	
22	tcp	open	ssh	Dropbear sshd 2020.81 (protocol 2.0)	
53	tcp	open	domain	dnsmasq 2.90	
80	tcp	open	http		
8000	tcp	open	http	LuCI Lua http config	
8081	tcp	open	blackice-icecap		

192.168.10.110 (Inconnu) :















```
Nmap scan report for 192.168.10.110
Host is up (0.019s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE    SERVICE VERSION
7/tcp     filtered echo
37825/tcp  filtered unknown
50579/tcp  filtered unknown
51470/tcp  filtered unknown
```

Il faudrait fermer le port 7 car c'est un protocole auto reply qui peut flooder le réseau et participer à une attaque DDOS

192.168.10.111 (Windows 11) :




```
Nmap scan report for 192.168.10.111
Host is up (0.012s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

192.168.10.119 (Linux server Issabel PBX VoIP + mail + Web + MySQL)

Nmap Output	Ports / Hosts			Topology	Host Details	Scans
	Port	Protocol	State	Service	Version	
	22	tcp	open	ssh	OpenSSH 8.0 (protocol 2.0)	
	25	tcp	open	smtp	Postfix smtpd	
	80	tcp	open	http	Apache httpd 2.4.37 ((rocky) OpenSSL/1.1.1k)	
	110	tcp	open	pop3		
	143	tcp	open	imap		
	443	tcp	open	http	Apache httpd 2.4.37 ((rocky) OpenSSL/1.1.1k)	
	993	tcp	open	tcpwrapped		
	995	tcp	open	tcpwrapped		
	1720	tcp	open	h323q931		
	3306	tcp	open	mysql	MariaDB 5.5.5-10.3.39	
	4445	tcp	open	upnotifyp		
	5060	tcp	open	sip	Issabel-4.1 (Status: 401 Unauthorized)	
	8088	tcp	open	http	Asterisk 18.19.0	
	8089	tcp	open	http	Asterisk 18.19.0	

Utilisation du protocole SIP (Session Initiation Protocol) qui renvoie le code 401, code http, sauf que ce protocole gère les connexions et donc les login/password, mais sans HTTPS/SSL les identifiants sont en clair sur le réseau.

192.168.10.246 (OpenWrt) :

Nmap Output	Ports / Hosts			Topology	Host Details	Scans
	Port	Protocol	State	Service	Version	
	139	tcp	open	netbios-ssn	Samba smbd 4	
	445	tcp	open	netbios-ssn	Samba smbd 4	
	5357	tcp	open	http	BaseHTTPServer 0.6 (Python 3.7.3)	

Serveur Samba pour le stockage, surement un NAS car c'est un OS OpenWrt (système embarqué), je n'ai pas accès au serveur car je n'ai pas les id.

Le serveur web renvoie cette erreur :

← → ↻ ⚠ Non sécurisé 192.168.10.246:5357

Error response

Error code: 501

Message: Unsupported method ('GET').

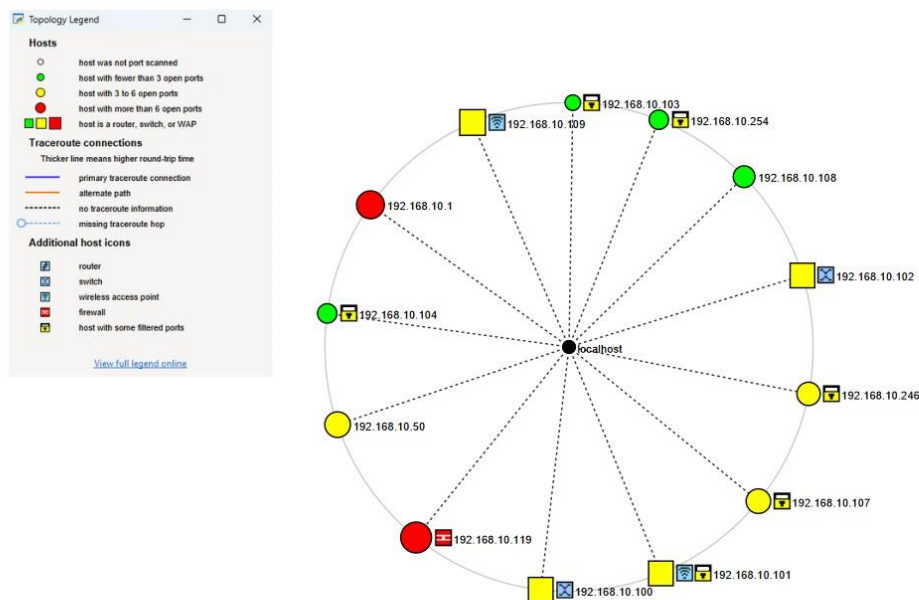
Error code explanation: HTTPStatus.NOT_IMPLEMENTED - Server does not support this operation.

192.168.10.254 (Linux – Gateway du reseau) :

Nmap Output					
Ports / Hosts					
Topology					
Host Details					
Scans					
	Port	Protocol	State	Service	Version
●	53	tcp	open	domain	dnsmasq 2.90
●	80	tcp	closed	http	
●	443	tcp	closed	https	

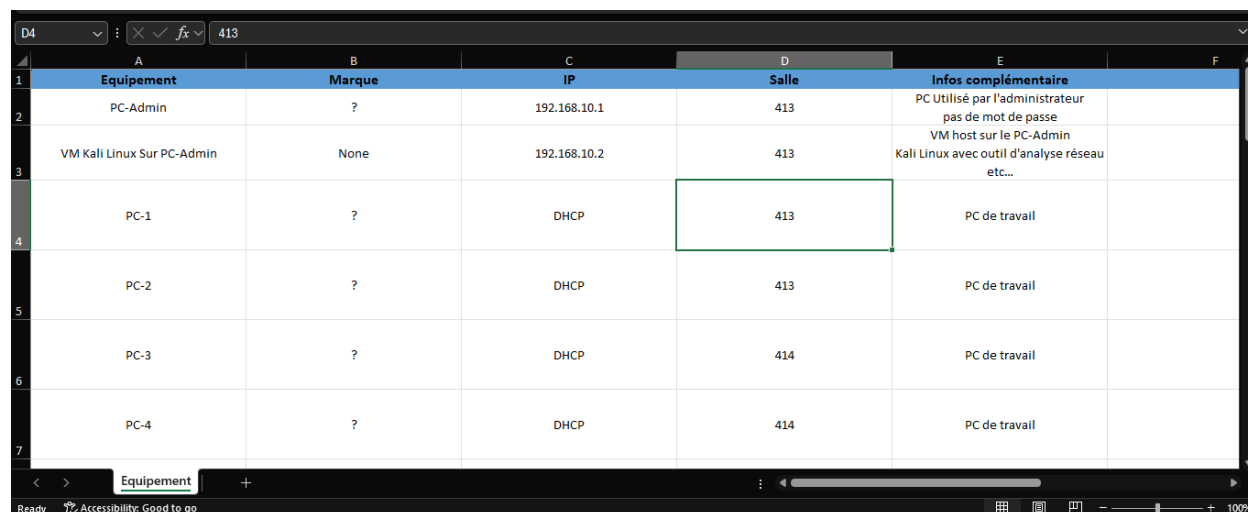
Surement le routeur, la page web est fermé, donc pas d'administration web possible.

Voici la topologie réseau :



Inventaire Excel :

L'intégralité du matériel que j'ai trouvé sur le réseau et disponible par le CMQ a été écrit sur un fichier excel.



	A	B	C	D	E	F
	Equipement	Marque	IP	Salle	Infos complémentaire	
1	PC-Admin	?	192.168.10.1	413	PC Utilisé par l'administrateur pas de mot de passe	
2	VM Kali Linux Sur PC-Admin	None	192.168.10.2	413	VM host sur le PC-Admin Kali Linux avec outil d'analyse réseau etc...	
3	PC-1	?	DHCP	413	PC de travail	
4	PC-2	?	DHCP	413	PC de travail	
5	PC-3	?	DHCP	414	PC de travail	
6	PC-4	?	DHCP	414	PC de travail	
7						

Récapitulatif mission :

Cette mission m'a permis de mettre en place mes capacités d'analyses réseaux, de détection d'éventuelle faille sur des machines et la mise en place d'un inventaire pour lister les équipements réseaux. Malheureusement je n'ai pas pu aller plus loin, comme voir la configuration des switches/routeurs car les mots de passes étaient détenus par l'ancien admin qui n'a jamais pu nous les donner (appeler plusieurs fois).

Mission 2 : Réalisation d'une plateforme Attaque/Défense

Robin a missionné Jordan (stagiaire MMI) et moi-même de réaliser une app web pédagogique adresser a des lycéens bac pro et/ou BTS CIEL/SIO. Cette plateforme inspirer de rootme, doit proposer une mise en situation d'attaque/défense dans le milieu de la cybersécurité.

J'ai commencé par réaliser un google slide avec l'aspect théorique et le fonctionnement du jeu, j'ai par la suite ajouté les technologies que j'ai utilisé pour créer l'app. Il y'a aussi des devis server car mon app fonctionneras avec VMs.

Lien vers le slide :

<https://docs.google.com/presentation/d/1zo8Zyt0ap03LvxmVKonP1ViTlgmaE3KidZg-xVL4FsM/edit?usp=sharing>

Ensuite j'ai réalisé un schéma explicatif de l'infrastructure technique du projet (fichier SVG) https://github.com/raphdzn/CMQ/blob/main/sch%C3%A9ma_infra.svg

Pour réaliser l'app, j'ai utilisé Python Flask pour le serveur web, sqlite3 pour la DB (simple d'utilisation), html/css/js la base pour les sites. Pour l'instant le site est une maquette mais pour un déploiement définitif, il vaut mieux utiliser une DB de type PostgreSQL car plus stable et plus puissant.

1^{ère} étape : Base de données

Structurer la base de données et réfléchir à quel élément et table j'aurai besoin pour réaliser mon application web, sachant qu'il y'a des utilisateurs et des données à gérer, voici ma DB :

Name	Type	Schema
▼ Tables (10)		
> challenges		CREATE TABLE challenges (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, description TEXT, flag TEXT, points INTEGER, level TEXT)
> defenses		CREATE TABLE defenses (id INTEGER PRIMARY KEY AUTOINCREMENT, team_id INTEGER, challenge_id INTEGER, proof TEXT)
> documentation_articles		CREATE TABLE documentation_articles (id INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT NOT NULL, content TEXT NOT NULL, category TEXT, tags TEXT, diff
> matches		CREATE TABLE matches (id INTEGER PRIMARY KEY AUTOINCREMENT, challenge_id INTEGER, vm_id TEXT, ip_address TEXT, status TEXT)
> solved_challenges		CREATE TABLE solved_challenges (id INTEGER PRIMARY KEY AUTOINCREMENT, team_id INTEGER NOT NULL, challenge_id INTEGER NOT NULL, user_id INTEGER, --
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
> team_matches		CREATE TABLE team_matches (id INTEGER PRIMARY KEY AUTOINCREMENT, team_id INTEGER, match_id INTEGER, access_token TEXT)
> teams		CREATE TABLE teams (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT UNIQUE, password TEXT, role TEXT, score INTEGER DEFAULT 0, creator_id INTEGE
> user_teams		CREATE TABLE user_teams (id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER, team_id INTEGER, UNIQUE(user_id, team_id))
> users		CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT UNIQUE, password TEXT, preferred_role TEXT, score INTEGER DEFAULT 0)
▼ Indices (0)		
▼ Views (0)		
▼ Triggers (0)		

La DB est une sqlite3, parfait pour une maquette, sachant qu'elle est légère à faire tourner sur un environnement de production et je m'enlève de potentiel erreurs de logiciel PostgreSQL ou autre DB.

La manipulation de la DB se fera via le backend codé en python.

2^{ème} étape : Backend

Pour réaliser le server j'ai choisi Python et son framework Flask pour réaliser des serveurs web. Python est un langage que je maîtrise, Flask est simple à mettre en place, je ne voulais pas perdre de temps avec du JavaScript et du React car je ne maîtrise pas assez ces technologies.

J'ai donc créé un environnement virtuel et installé tous les packages requis grâce à PIP :

Name	Date modified	Type	Size
Include	5/22/2025 9:02 AM	File folder	
Lib	5/22/2025 9:02 AM	File folder	
Scripts	6/6/2025 8:58 AM	File folder	
.gitignore	5/22/2025 9:02 AM	Text Document	1 KB
pyvenv.cfg	5/22/2025 9:02 AM	Configuration Sou...	1 KB

Venv permet d'isoler les packages python pour ne pas créer de problèmes de versioning entre les différents projets que je réalise sur ma machine, on peut aussi facilement partager les packages requis grâce à un fichier « requirements.txt » qui contient les packages et leur version, on peut alors installer tout d'un coup grâce à PIP. Parfait pour faire la succession de mon projet à un autre dev.

Le fichier backend « app.py » contient les imports des différents packages :

```
from flask import Flask, request, jsonify, render_template, redirect, url_for, session
import sqlite3
import bcrypt
import secrets
import logging
```

Création de l'app Flask :

```
app = Flask(__name__)
app.secret_key = secrets.token_hex(16) # Clé secrète pour les sessions

# Configurer les logs pour le débogage
logging.basicConfig(level=logging.DEBUG)
```

Pour lancer le serveur :

```
# Initialiser la base de données au démarrage
with app.app_context():
    init_db()

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Le reste du code contient des « routes » qui sont appelés via le frontend du code et le protocole HTTP/S :

```
@app.route('/admin/documentation/delete/<int:article_id>', methods=['DELETE'])
def admin_delete_documentation(article_id):
    if 'admin_id' not in session:
        return jsonify({"message": "Accès admin requis."}), 401




    conn = None
    try:
        conn = sqlite3.connect('ctf.db')
        c = conn.cursor()
        delete_result = c.execute("DELETE FROM documentation_articles WHERE id = ?", (article_id,))
        conn.commit()

        if delete_result.rowcount > 0:
```

99% des routes sont en interactions avec la DB car l'app repose exclusivement sur un système de donnée et de vérification d'accès.

3^{ème} étape : Frontend

Le frontend est la partie visible par les utilisateurs, le serveur Flask gère automatiquement la redirection et l'affichage des pages via un dossier « templates » :

Name	Date modified	Type	Size
 admin.html	6/18/2025 2:49 PM	Brave HTML Docu...	35 KB
 home.html	6/20/2025 10:49 AM	Brave HTML Docu...	73 KB
 login.html	6/20/2025 11:19 AM	Brave HTML Docu...	19 KB

Pour la réalisation des pages, j'ai commencé simplement par le HTML pour structurer mes pages, ensuite le CSS pour le design et pour finir, le JS qui permet l'interaction avec le serveur via les routes grâce à HTTP/S.

4^{ème} étape : Serveur

Flask propose via une méthode de lancer un serveur de test via « app.run » :

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

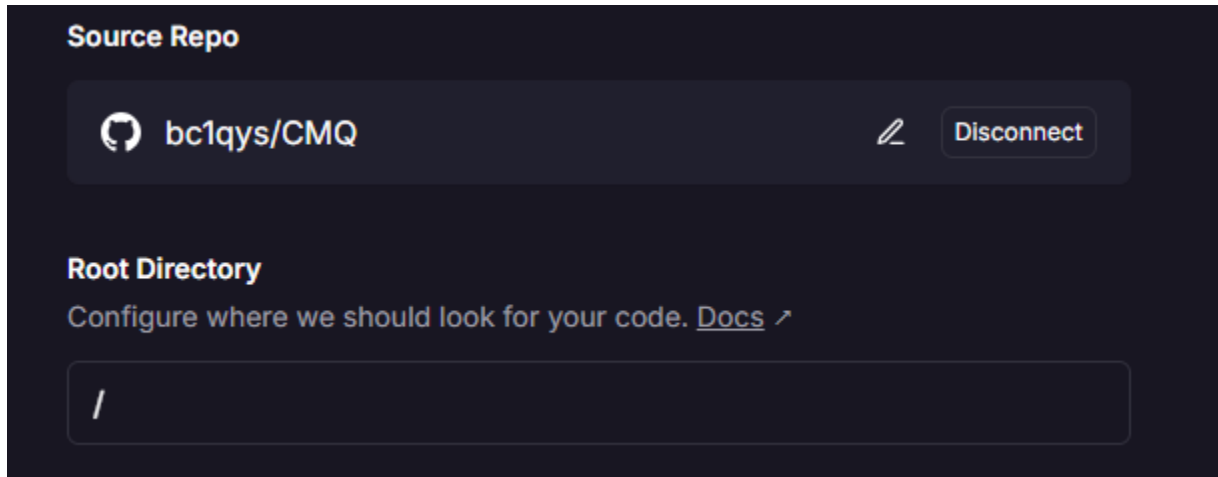
Pour tester le serveur en local c'est largement suffisant mais Flask signale qu'il faut utiliser d'autre technologies comme Gunicorn, CherryPy et Waitress qu'on verra juste après.

Gunicorn, CherryPy et Waitress sont des serveurs WSGI qui permettent de gérer la charge du serveur et le multi-threading. J'ai donc choisi Gunicorn, celui recommandé par la communauté Python.

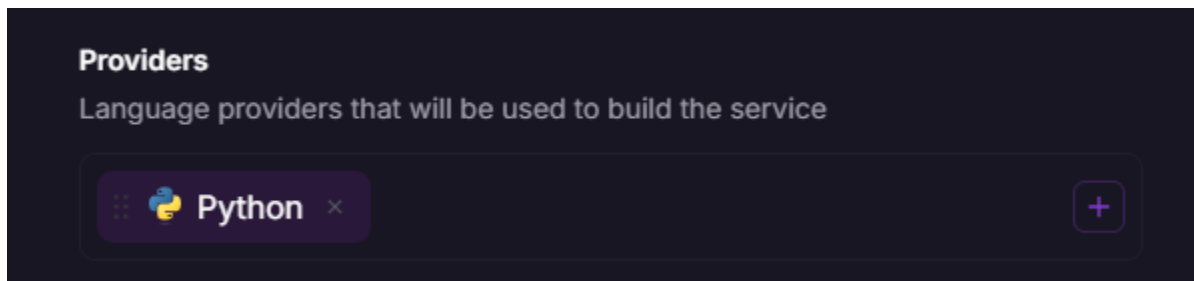
Hébergement du projet :

Au tout début du projet, je lancer le server via Flask en hébergement local, mais cela n'était pas optimisé, je ne pouvais pas partager l'avancée du projet à Robin et le serveur consommé pas mal de batterie sur mon PC.

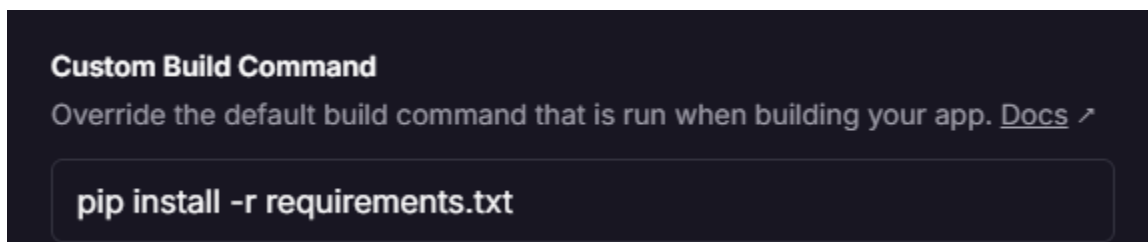
J'ai utilisé railway pour héberger mon projet directement via github. Railway me fournit un server qui récupère le projet directement sur github. Il faut paramétrer Railway pour qu'il exécute du Python :



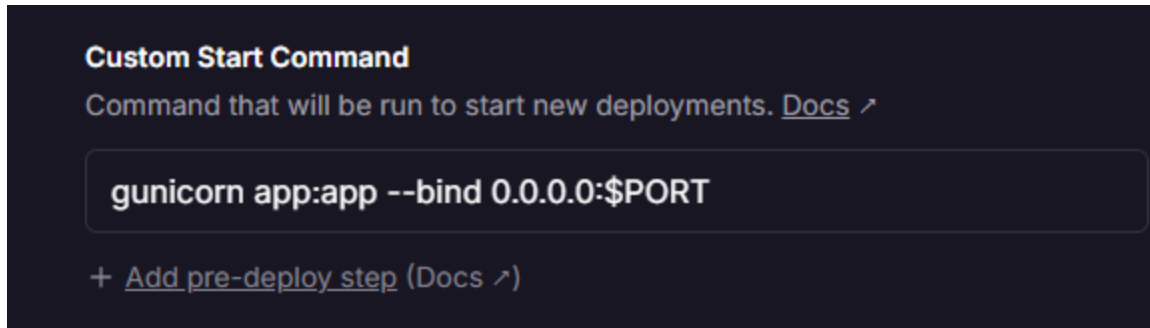
Il faut lier son projet github et signaler où est la racine du projet, dans mon cas /.



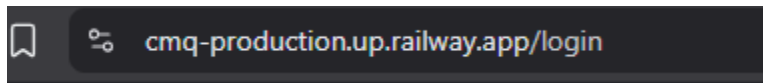
Il faut ajouter le langage Python.



La build command qui permet d'ajouté les packages depuis mon fichier « requirements.txt », pas besoin de venv car il y'a qu'un projet sur la machine.



La commande pour lancer le serveur.



Railway me génère donc une URL où je peux accéder depuis n'importe où à mon projet.

Développement :

J'ai donc passé la majorité de mon stage à développer cette plateforme, le plus long a été de définir les routes sur le serveur Flask, je devais penser à chaque fonctionnalité notamment pour le côté admin qui est complexe. La gestion de la base de données a été compliqué car je ne suis pas très à l'aise avec le SQL donc j'avais des problèmes de logique au sein de mon application, comme le fait qu'un joueur puisse rejoindre plusieurs équipes, lancé plusieurs match etc...J'ai donc du rajouté des vérifications au seins des routes.

Je ne me suis pas spécialement organisé dans un ordre précis mais j'avancé selon mes envies, du design un jour, du HTML un autre jour etc...Avec du recul j'aurai du mieux m'organisé et m'occuper en 1^{er} de la logique du site avant le design.

Le site est disponible ici : <https://cmq-production.up.railway.app/>

Présentation du site à Robin :

A mon dernier jour de stage j'ai présenté l'avancer de mon projet à Robin, l'ingénieur pédagogique, je lui ai fait une démonstration du site, son fonctionnement coté élève et administrateur. Présenté les faiblesses du site comme le fait que le système de point de la défense n'est pas super bien désigné. Je lui ai aussi partagé les aces d'amélioration possible, comme migrer le site vers une solution JavaScript et React.js, bien plus puissant et logique pour réaliser une app web.

J'ai aussi rédigé un guide explicatif sur le fonctionnement du site, son utilité, comment il est conçu et son installation complète.

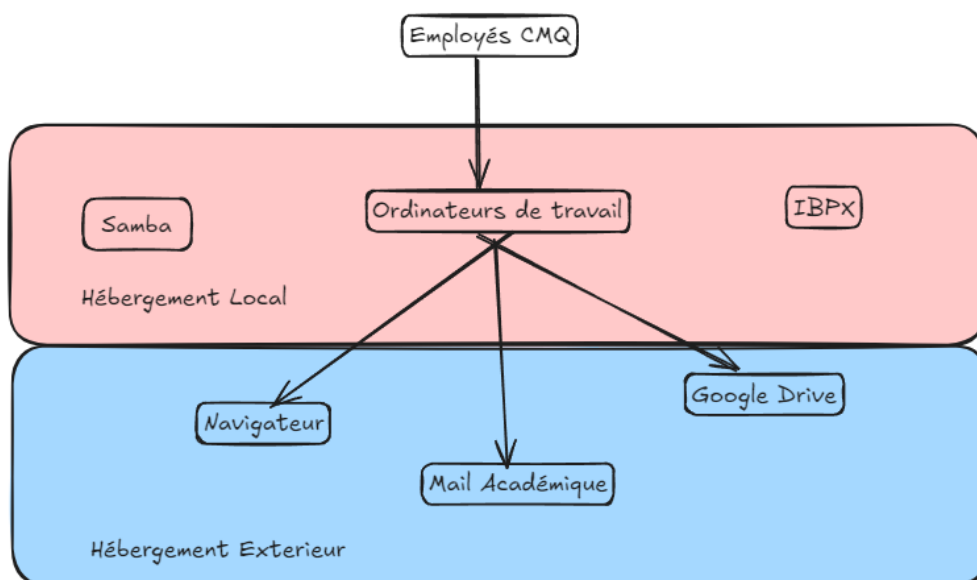
Récapitulatif de mission :

Cette mission était assez éloignée de mon parcours d'étude, réaliser un site web est plus pour les SLAM, mais j'ai bien aimé réaliser ce dernier. Le site est conçu pour des lycéens qui veulent s'entraîner à la cybersécurité, c'est donc un domaine que je maîtrise et je ne me suis pas senti en difficulté.

Mission 3 : Rdv avec MegaO pour la reprise du réseau informatique :

Jeudi 13 juin, MegaO venez dans les locaux pour reprendre en main le réseau informatique. Emilie (la directrice) m'a demandé d'inspecter les différents locaux où est entreposé le matériel informatique pour préparer le rdv. Le jour-j j'ai donc accompagné Emilie et Réagis (MegaO) pour la visite, j'ai donc expliqué comment structurer l'infrastructure réseau, les serveurs et leurs rôles etc...J'ai ensuite partagé mon excel concernant la liste de matériel pour que MegaO propose une offre adéquate aux besoins du CMQ.

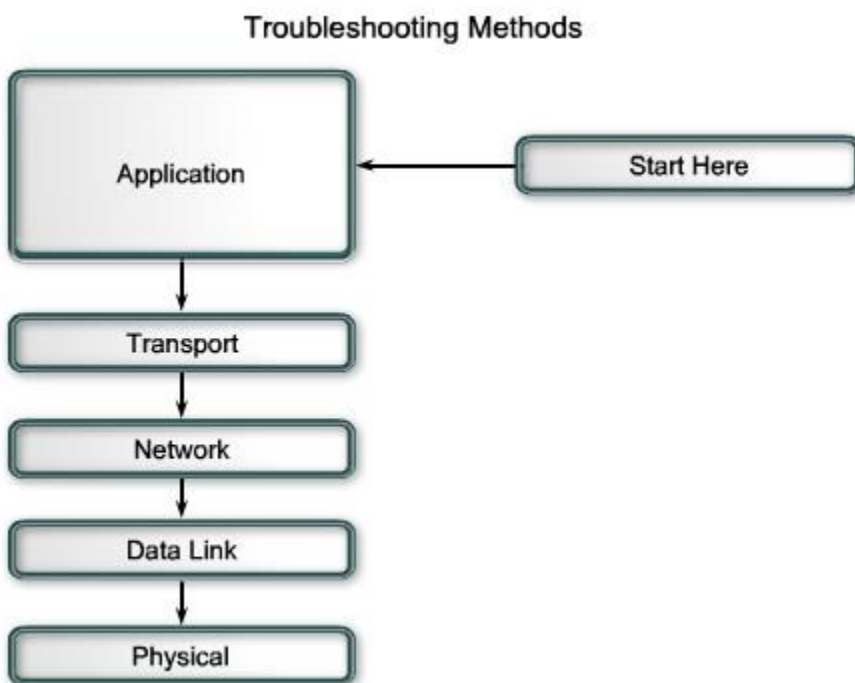
J'ai aussi expliqué comment les employés utilisent l'informatique du CMQ :



A la suite du rdv, j'ai débranché 2 machines qui appartenait à l'ancien admin réseau (Andy). Un serveur Samba et IBPX que personne n'utiliser et un ASIC qui est une machine qui mine du Bitcoin.

Mission 4 : Maintenance du réseau internet :

Durant la dernière semaine de stage, le réseau internet est tombé en panne, avec Alessio (stagiaire CIEL), nous avons donc décider de réparer le réseau en utilisant la méthode « bottom-up approach » :



- 1) Nous avons vérifié si le routeur fonctionnait bien et c'était le cas, donc on remonte d'un niveau.
- 2) Vérification du switch, pas de led qui clignoté alors que les câbles étaient branchés et internet fonctionné, donc nous avons changer le switch par le même model et les leds sont revenus mais le WiFi ne fonctionnait toujours pas.
- 3) En dernier, le repeteur, après l'avoir branché en direct depuis la box, il fonctionnait pendant environ 3h et après le WiFi s'éteignait, nous l'avons remplacé par d'autres modèles mais toujours la même erreur.

Nous avons conclu que le problème venait des répéteurs WiFi mais nous n'avons pas d'autre modèles pour essayer, le switch, le routeur fonctionnait parfaitement puisqu'en câblant directement nos ordinateurs, nous avions internet.