



RAPPORT DE PROJET BTS CIEL 2025



Promotion : 2023-2025

Région GRANDEST

Crée par : KETTERER Anthony

Tuteur : MASSONSCIAUX Romaric

Table des matières

1.	Remerciements	4
2.	Présentation de l'entreprise.....	5
a.	Région GrandEst & Maison de Région.....	5
b.	Les effectifs et missions des agents de la Région Grand Est	7
c.	Siège de la Région GrandEst.....	8
d.	Chargés de Maintenance Informatique des Lycées (CMIL)	9
e.	Les Lycées de la Région Grand Est (Strasbourg)	12
3.	Principales tâches réalisées lors des deux années à la Région GrandEst.....	15
a.	L'assistance aux utilisateurs	15
b.	La gestion de tickets	16
c.	La masterisation	17
d.	Gestion de données et cybersécurité	18
e.	Gestion de projet et programmation	19
4.	Présentation du projet	20
a.	Contraintes techniques et économiques	20
b.	Matériels utilisés	21
c.	Diagrammes	23
5.	Introduction	27
6.	Configuration switch et partie capteur de température.....	28
a.	Configuration des interfaces du switch de cœur de réseau	28
b.	Programmation du microcontrôleur Arduino Uno.....	29
I.	Explication du capteur.....	29
II.	Explication du programme.....	31
7.	Proxmox VE (Virtual Environnement)	37
a.	Installation et configuration	37
b.	Interface web de Proxmox	38
8.	La configuration de la machine virtuel (VM).....	40
9.	Le protocole MQTT (Message Queuing Telemetry Transport)	42
a.	Explication du protocole	42
b.	Installation et configuration du protocole MQTT	43
c.	Tests du protocole MQTT	44
10.	Mise en place du VPN de la région GrandEst	45
a.	Configuration des interfaces du switch VPN.....	46
b.	Test de la mise en place du VPN	46
11.	Node-Red	47
a.	Certificats OpenSSL Node-Red	49

b.	Test des certificats Node-Red	50
12.	Simple Mail Transfer Protocol	51
13.	Base de données MariaDB & MySQL	53
a.	Script Python	55
14.	Installation et configuration de PHPMyAdmin.....	57
15.	Annexes.....	60
16.	Conclusion.....	61

1. Remerciements

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui ont contribué à la réussite de ce projet de mise en place réseau d'un capteur de température.

En premier lieu, je remercie chaleureusement la Région GrandEst, pour m'avoir accueilli au sein de ses équipes et pour m'avoir offert un cadre professionnel enrichissant. Je suis reconnaissant de la confiance qui m'a été accordée pour mener à bien ce projet et des moyens mis à disposition pour le réaliser dans les meilleures conditions. Je remercie aussi le lycée Louis COUFFIGNAL qui m'a permis de réaliser mon projet dans ses locaux.

Je tiens également à exprimer toute ma reconnaissance à Monsieur Romaric MASSON SCIAUX, mon tuteur de formation au sein de mon BTS, pour son accompagnement précieux, ses conseils avisés et sa disponibilité tout au long de cette expérience, pour son soutien pédagogique et son encadrement attentif. Ses conseils et son expertise m'ont permis de mieux appréhender les différents aspects théoriques du projet et de les appliquer dans un contexte concret. Ils m'ont également permis de progresser et de gagner en compétences, tant sur le plan technique que professionnel. Mes remerciements vont également à Monsieur Thierry BARBANSON qui a collecté les outils nécessaires au bon déroulement de mon projet.

J'adresse donc mes plus sincères remerciements pour la contribution donnée à la réussite de ce projet, qui a été pour moi une expérience enrichissante et formatrice.

2. Présentation de l'entreprise

a. Région GrandEst & Maison de Région

L'ancienne région Alsace, Champagne-Ardenne et Lorraine sont regroupées dans la Région GrandEst, une collectivité territoriale française. Située au nord-est de la France et frontalière de quatre pays : la Belgique, le Luxembourg, l'Allemagne et la Suisse, elle occupe une position stratégique en Europe.

Le rôle de la Région Grand Est est au sein du développement économique, social et culturel de son territoire est primordial. Elle soutient les entreprises grâce à des projets comme "Grand Est Transformation", qui offre des trajectoires de transformation axées sur quatre domaines principaux : le numérique, l'environnement, l'industrie et la santé.

De plus, la Région apporte son soutien à l'établissement et à la reprise d'entreprises en proposant des moyens de financement et de formation, ce qui favorise la compétitivité et l'innovation sur son territoire.

L'environnement, le développement des transports, la culture et le sport, l'amélioration de la qualité de vie de ses habitants sont également des domaines d'action de la Région Grand Est. En mettant en place 12 Maisons de la Région à travers tout le territoire, elle garantit une proximité avec les habitants afin de les soutenir dans leur vie quotidienne. Il existe plusieurs Maison de Région dans le GrandEst :

La maison de Région de Strasbourg se trouve
6 rue Oberlin, 67000 Strasbourg



Maison de Région Strasbourg



Carte régionale des maisons de région

**12 Maisons de Région
12 équipes CMIL (100 CMIL)
12 chefs de pôle CMIL**

Les Maisons de la Région exercent principalement une mission de relais entre les habitants et l'administration régionale en matière de diffusion de l'information et d'orientation des publics sur les dispositifs régionaux. Elles assurent par ailleurs des missions de gestion de certains services dédiés principalement aux citoyens tels que les transports scolaires et interurbains, les aides aux entreprises locales ou encore les subventions aux initiatives culturelles et associatives.

Ces Maisons de la Région permettent d'éviter aux citoyens et aux partenaires locaux de se déplacer jusqu'au siège principal situé à Strasbourg. Elles organisent des permanences et des événements locaux pour répondre efficacement aux besoins des territoires.

En complément, le siège de la Région, basé à Strasbourg, assume une fonction d'administration centrale et de prise de décisions stratégiques. Tandis que le siège se concentre sur la coordination des grandes politiques publiques, les Maisons de la Région garantissent une accessibilité accrue des services régionaux. Ensemble, ces structures assurent un équilibre entre centralisation stratégique et proximité, permettant de mieux répondre aux attentes des citoyens et des acteurs locaux.

b. Les effectifs et missions des agents de la Région Grand Est

La Région Grand Est emploie un total de 7 700 agents, répartis entre les services administratifs régionaux, les lycées et les Maisons de la Région.

Les services administratifs régionaux comptent 2 350 agents, qui sont répartis sur les 3 sites de Strasbourg, Châlons-en-Champagne et Metz.

Dans les lycées, la Région emploie 5 350 agents techniques des établissements d'enseignement (ATEE). Ces personnels sont chargés de l'entretien général, de l'accueil et de la maintenance des établissements scolaires. Leurs missions incluent l'entretien des locaux, la maintenance des infrastructures, la gestion des équipements techniques et la participation au bon fonctionnement quotidien des lycées.

Les Maisons de la Région, au nombre de 12, sont réparties sur l'ensemble du territoire du Grand Est. Chacune de ces Maisons compte en moyenne une équipe pluridisciplinaire d'une cinquantaine d'agents. Ces équipes sont principalement structurées autour de trois pôles de compétences :

- **Lycées** : gestion patrimoniale des installations et des agents de la Région.
- **Développement territorial** : comprenant le développement économique, l'emploi et la formation, la transition énergétique et l'ingénierie territoriale.
- **Transports et mobilités** : gestion des transports scolaires et interurbains ainsi que des mobilités durables.

Ces Maisons de Région ont pour mission de gérer 243 lycées publics fréquentés par 211 500 lycéens, d'assurer la formation de 48 600 demandeurs d'emploi, d'accompagner plus de 260 000 entreprises dans leur développement et de soutenir les collectivités locales dans leurs projets.

En résumé, la Région Grand Est emploie 2 350 agents dans ses services administratifs, 5 350 agents techniques dans les lycées et environ 600 agents répartis dans les 12 Maisons de la Région, chacun contribuant au bon fonctionnement et au développement du territoire à travers des missions spécifiques.

c. Siège de la Région GrandEst

Le siège de la Région Grand Est, situé à Strasbourg, remplit des fonctions essentielles au fonctionnement et à la gestion stratégique de la région. En tant que siège décisionnel, il accueille les séances plénières et les réunions des commissions du Conseil régional, où sont prises les grandes décisions politiques et budgétaires.

En tant qu'administration centrale, il supervise la mise en œuvre des politiques publiques régionales dans des domaines variés tels que l'éducation, les transports, l'économie ou encore l'environnement. Il assure également la coordination globale des actions régionales et la gestion des budgets.

Le siège gère les grands dossiers stratégiques, notamment en matière d'infrastructures, de développement économique et de partenariats internationaux. Il joue également un rôle d'interface avec les grandes entreprises ainsi que les institutions nationales et européennes, renforçant ainsi l'influence et la coopération de la région à différents niveaux.

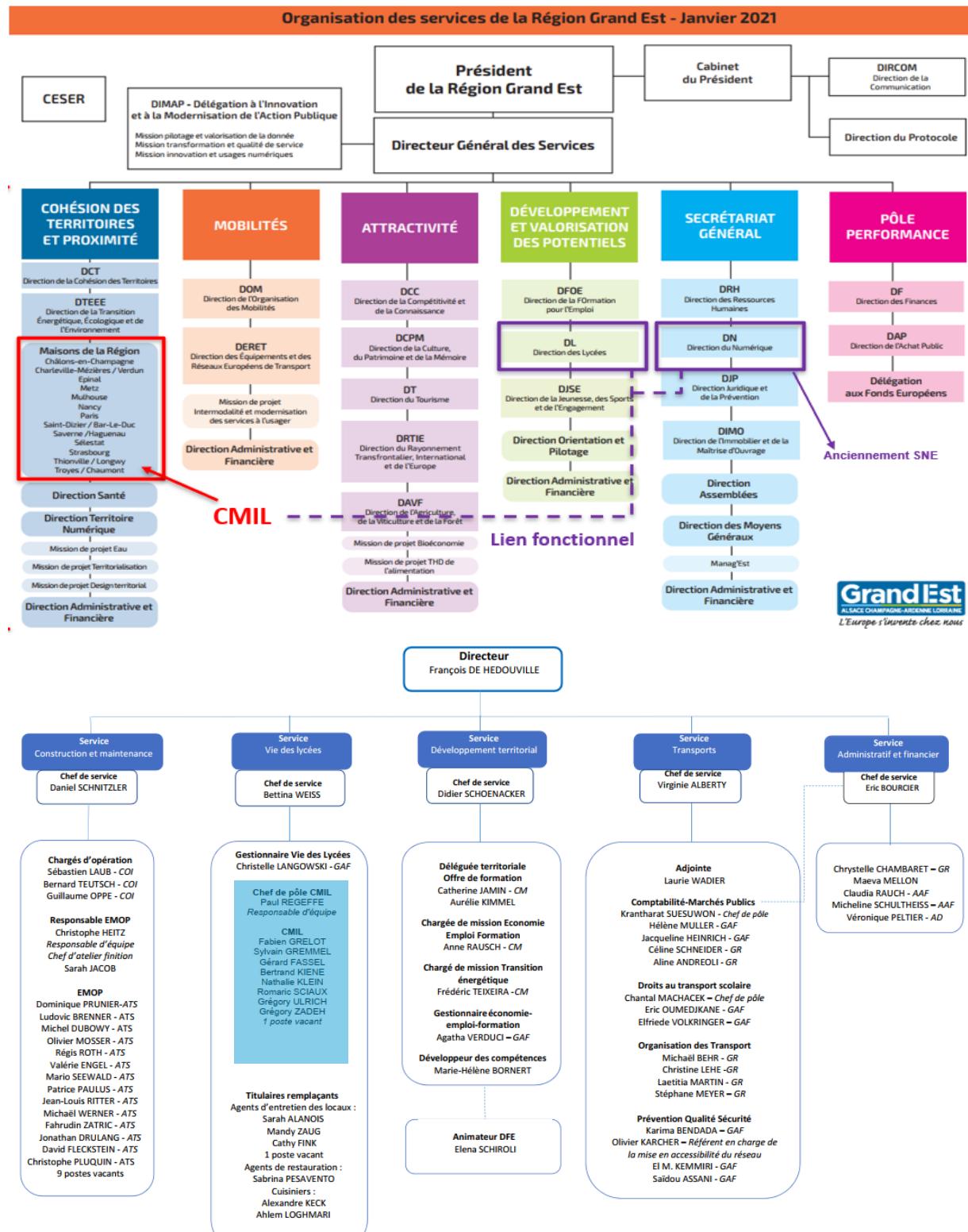


Siège de la Région GrandEst situé 1 Place Adrien Zeller 67070 Strasbourg.

d. Chargés de Maintenance Informatique des Lycées (CMIL)

Quelle est la place du CMIL dans la collectivité de la Région GrandEst ?

Les CMIL étaient durant une certaine période attaché au SNE (Système Numérique Educatif). A cause de la disparition de ce dernier, ils sont rattachés dans la catégorie DN (Direction du Numérique) reliant également la DL (Direction des Lycées).



Rôles des CMIL

Les CMIL interviennent uniquement dans les établissements publics

Les CMIL appliquent la politique régionale du SNE et des Elus en matière de numérique

1) Maintenance de niveau 1 et 2 :

- de l'infrastructure (serveurs et réseaux)
- du parc informatique pédagogique
- de la téléphonie

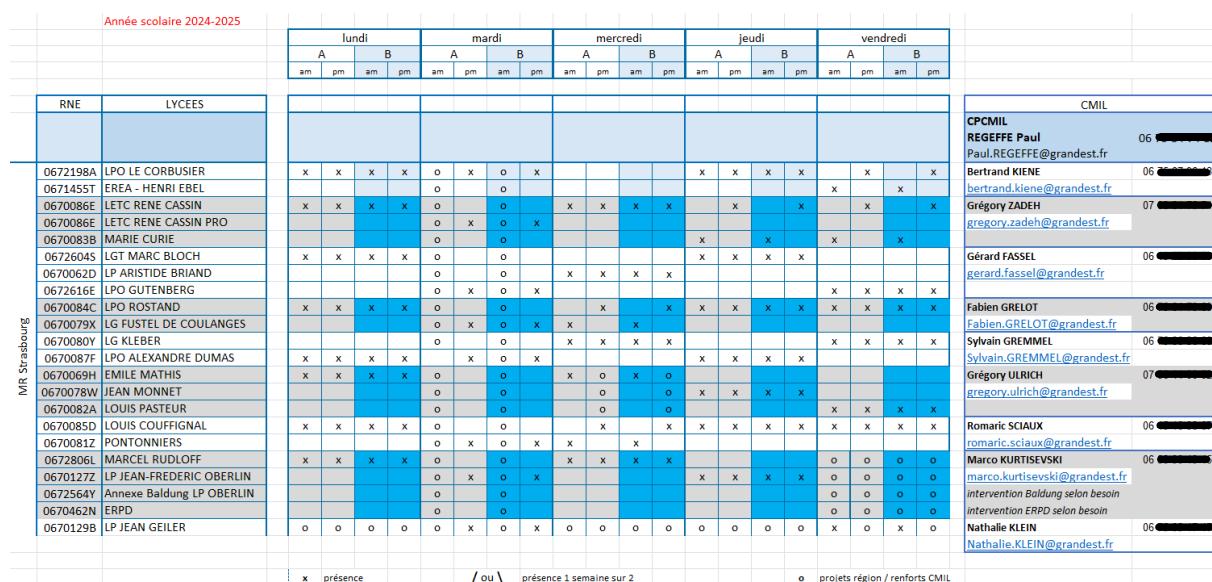
2) Administration, contrôle et gestion :

- des serveurs pédagogiques
- des ressources partagées pédagogiques

3) Assistance et accompagnement :

- Assister les ARL (Agents Région des Lycées) pour la connexion aux outils informatiques Région (Messagerie, Mobilattee, C'est, Traj'Est...)
- Accompagnement des personnels établissement pour la connexion aux outils Région (GESMAT, Extranet, Déclyc, Webgerest)
- Assistance aux personnels référents des établissements à l'usage des outils lycée 4.0 en début d'année scolaire

Répartition des CMIL par établissement



Répartition des lycées par CMIL

Qu'est-ce qu'un CMIL et quel est son travail au sein du lycée ?

Un CMIL est un professionnel dont la mission est d'assurer le bon fonctionnement des infrastructures informatiques au sein des établissements scolaires. Il joue un rôle indispensable pour garantir que les outils numériques nécessaires aux activités pédagogiques soit toujours opérationnels. Les systèmes et activés administratives sont désormais géré par le **RAIP (Relais d'Assistance Informatique de Proximité)**.

Au quotidien, le CMIL est responsable de la maintenance des équipements informatiques, tels que les ordinateurs, tablettes, imprimantes et autres dispositifs numériques. Il intervient pour réparer les pannes, effectuer les mises à jour et s'assurer de leur performance optimale. Il est également en charge de la gestion des réseaux locaux et de la connectivité internet, garantissant une connexion fiable et sécurisée pour les élèves, enseignants et personnels administratifs.

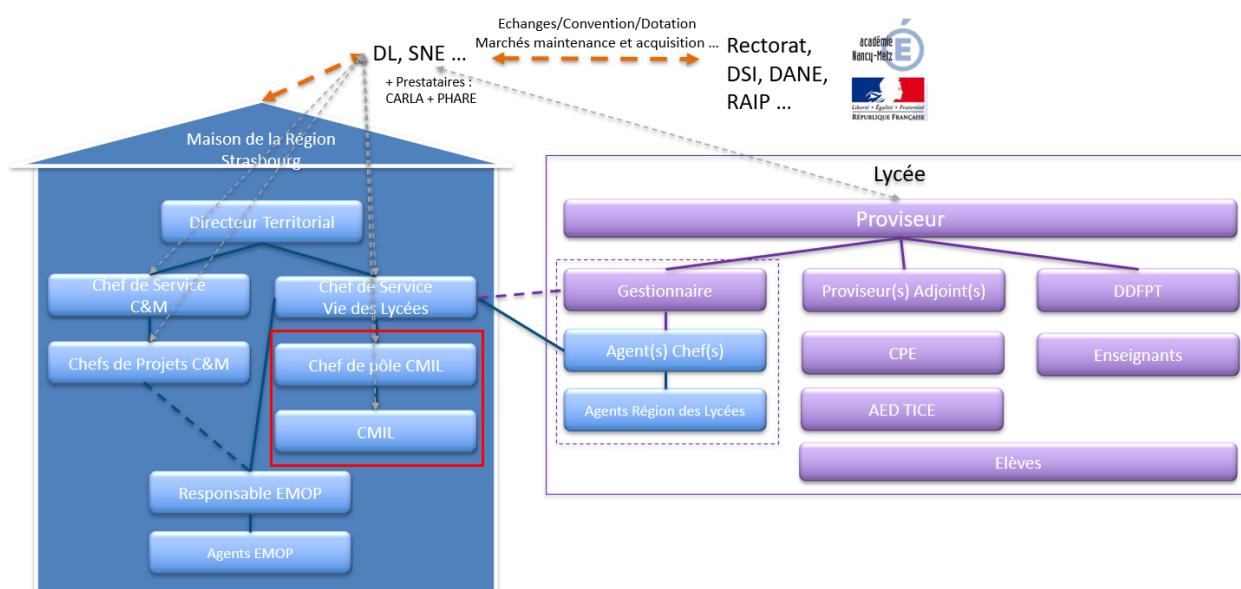
En cas de problème technique, le CMIL offre un support direct aux utilisateurs, qu'il s'agisse de résoudre des bugs, de dépanner des logiciels ou d'accompagner les utilisateurs dans la prise en main des outils numériques. Par ailleurs, il veille à la sécurité informatique en protégeant les systèmes contre les cybermenaces et en garantissant la confidentialité des données.

Le travail du CMIL est essentiel pour accompagner la transformation numérique des lycées. Grâce à son expertise, il contribue activement à créer un environnement technologique efficace, sécurisé et adapté aux besoins des élèves et des équipes pédagogiques.

e. Les Lycées de la Région Grand Est (Strasbourg)

SUR LA REGION GRAND EST	SUR LA MAISON DE LA REGION - STRASBOURG
<ul style="list-style-type: none"> ✓ 4 rectorats : <ul style="list-style-type: none"> ▪ Strasbourg ▪ Reims ▪ Nancy-Metz ▪ DRAAF Grand Est pour les lycées agricoles 	<ul style="list-style-type: none"> ✓ 1 rectorat : <ul style="list-style-type: none"> ▪ Strasbourg
<ul style="list-style-type: none"> ✓ 342 Etablissements Publics Locaux d'Enseignements : <ul style="list-style-type: none"> ▪ 237 lycées publics et agricoles ▪ 105 lycées privés 	<ul style="list-style-type: none"> ✓ 25 Etablissements Publics Locaux d'Enseignements : <ul style="list-style-type: none"> ▪ 18 lycées publics, 1 ERPD, 1 EREA ▪ 7 lycées privés
<ul style="list-style-type: none"> ✓ Plus de 220 000 lycéens : <ul style="list-style-type: none"> ▪ ~183 000 lycéens dans le public ▪ ~37 000 lycéens dans le privé 	<ul style="list-style-type: none"> ✓ Plus de 12 000 lycéens : <ul style="list-style-type: none"> ▪ Chiffres hors lycées privés

Explication du lien entre la MRS (Maison de Région Strasbourg) et les lycées.



Lien entre la MRS et les lycées.

Le fonctionnement d'un lycée.

Fonction	Employeur	Description / Missions
Directeur Territorial	Région Grand Est	N+3 des CMIL. Directeur de la Maison de la Région. Manage et coordonne les services de la Maison de la Région, gère le budget, applique les politiques régionales votées par les élus.
Chef de Service Vie des Lycées	Région Grand Est	N+2 des CMIL. Manage et coordonne le service Vie des Lycées. A en charge la gestion RH des Agents Région des Lycées (ARL) + EMOP (Equipe Mobile d'Ouvriers Professionnels)
ARL (Anciennement ATTEE)	Région Grand Est	ARL : Agents Région des Lycées. On en charge l'accueil, l'entretien, la maintenance et la restauration.
Chef de pôle CMIL	Région Grand Est	N+1 des CMIL. Manage et coordonne l'équipe CMIL
CMIL	Région Grand Est	Chargés de la Maintenance Informatique pédagogique dans les lycées
EMOP	Région Grand Est	EMOP : Equipe Mobile d'Ouvriers Professionnels. A en charge les travaux dit « lourds » et spécialisés (équipement de cuisine, Les EMOP sont mobiles et interviennent sur plusieurs lycées.
Chef de Service C&M (Construction et Maintenance)	Région Grand Est	Chef du Service en charge de la maintenance des bâtiments dans les lycées
Chefs de projets COI	Région Grand Est	Chargé d'Opérations Immobilières. En charge de la maintenance des bâtiments dans les lycées
DL	Région Grand Est	Direction des Lycées
SNE	Région Grand Est	Service Numérique Educatif
Phare	Prestataire Grand Est	Hotline Phare, support technique, MOE travaux courant faibles 4,0 ...

Qu'est-ce que le lycée 4.0 ?**✓ Les enjeux politiques :**

- Permettre l'accès et favoriser l'usage du numérique
- Préparer les lycéens à la transformation digitale et lutter contre la fracture numérique
- Accroître les compétences individuelles et préparer la vie professionnelle des lycéens
- Alléger le poids des cartables

✓ Le principe : Remplacer les manuels papiers par des ressources numériques

- Manuels scolaires numériques
- Contenus numériques enrichis : vidéos, animations, contenus dynamiques.
- ENT (Espace Numérique de Travail) : monbureaunumerique.fr
- Usage collectif des ressources numériques (en classe)
- Usage individuel des ressources numériques (à domicile / en classe)

✓ Le matériel : Bring Your Own Device (BYOD) ou EIM (Equipement Individuel Mobile)

- Lycées 4.0 vague 1 et vague 2 (2017 et 2018) : Achat à tarif subventionné région d'un pc ou d'une tablette
- Lycées 4.0 vague 3 et 4 (2019 à nos jours) : PC portable gratuit pour tous les lycéens hors adaptations spécifiques (modèle unique 14")

3. Principales tâches réalisées lors des deux années à la Région GrandEst

Lors de ces deux années travaillant à la Région GrandEst, j'ai pu réaliser plusieurs activités intéressantes comme la configuration d'appareils réseaux (switch), la configuration d'un logiciel informatique sur Linux qui permettra de gérer les futurs tickets du lycée Couffignal (GLPI) ou encore booster (rajout de RAM, de SSD) des postes qui seront utiles pour le réseau pédagogique. Parmi toutes les activités réalisées, en voici trois que j'ai principalement approfondies :

a. L'assistance aux utilisateurs

Cela signifie répondre le plus clairement possible à leurs questions, résoudre leurs problèmes informatiques et, principalement, assister le personnel pédagogique en réinitialisant leurs mots de passe en cas d'impossibilité de connexion au domaine.

NOM	PRÉNOM	PROFIL	IDENTIFIANT	ÉTABLISSEMENT	CLASSE	SOURCE
KETTERER	Anthony	Tous	aketterer2	1570002Z		Manuel

Interface web acl.numerique-educatif.fr pour la réinitialisation de mots de passe

En me connectant à ce site avec mon compte ACL (Accès aux Contenus en Ligne) qui est un logiciel mis en place par la Région Grand Est, avec des droits d'administrateurs, je peux :

- Rechercher le nom d'une personne ayant un incident avec son compte ACL.
- Visualiser des identifiants afin que l'utilisateur puisse se reconnecter au domaine.
- S'ils ne fonctionnent pas, avoir l'autorisation d'en créer d'autres fonctionnels.
- Activer et désactiver le compte.
- Supprimer le compte si la personne n'est plus active dans aucun lycée de Strasbourg.

b. La gestion de tickets

Le ticketing est un système de gestion des demandes et incidents informatiques utilisé par les services d'assistance et de support technique. Il permet d'organiser, suivre et résoudre efficacement les problèmes rencontrés par les utilisateurs.

Lycée polyvalent Louis Couffignal | Liste des tickets 64

Filtre

Mots clés

Créé entre

Rémonté depuis

Indifférent

Priorité

Indifférent

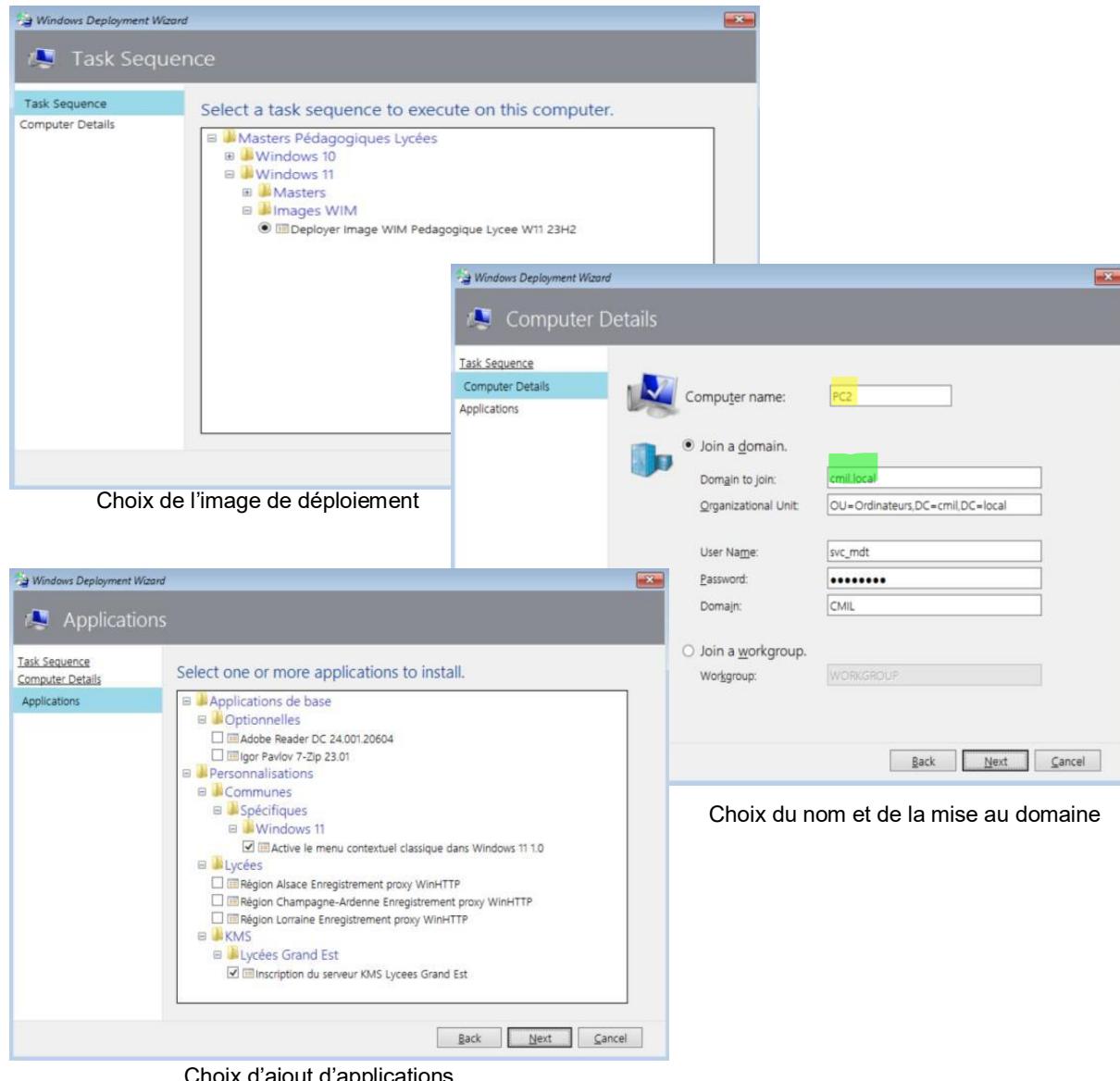
Interface web e-maintenance (tickets)

Lorsque des professeurs ou autre responsable pédagogique ont un problème en lien avec un système numérique, ils se connectent avec leurs comptes ACL sur l'outil de ticketing du lycée Louis COUFFIGNAL (e-maintenance) et ouvre un ticket. Pour résoudre un ticket, il doit passer par différentes étapes :

- Le ticket est créé par l'utilisateur.
- Il est pris en charge par l'administrateur.
- L'administrateur va sur place, résout le problème et teste la fonctionnalité.
- Il envoie un message à l'utilisateur afin de lui dire qu'il a résolu le problème et lui demande de tester de lui-même.
- Si le problème n'est pas résolu pour l'utilisateur, l'administrateur revérifie l'incident sur place.
- Si le problème est résolu pour l'utilisateur, le ticket est clôturé.

c. La masterisation

L'installation d'images sur des postes via MDT (Microsoft Deployment Toolkit) avec des applications et des logiciels précis mis en place selon les différentes sections (général, technologique et professionnel). Ce processus permet de gagner du temps en évitant une installation manuelle répétitive et assure une configuration standardisée adaptée aux besoins des l'utilisateurs.



Lorsqu'un professeur demande l'installation de plusieurs logiciels dans une salle de classe, les étapes suivantes sont mises en œuvre :

- Lancement du boot réseau (PXE) via la touche F12 sur le pc en question puis Entrer.
- Choix de l'image à mettre en place.
- Choix du nom du poste (ici **PC2**) et de l'inscription au domaine (**cmil.local**).
- Installation de l'image sur le poste.

d. Gestion de données et cybersécurité

La configuration des VLANs (Virtual Local Area Networks) a joué un rôle essentiel dans l'optimisation et la sécurisation du réseau. En segmentant le trafic, ils permettent d'organiser les flux de données, d'isoler les différents services et de réduire les risques liés aux conflits réseaux et aux accès non autorisés. Cette isolation améliore la gestion des communications internes en limitant la congestion et en optimisant les performances. De plus, l'implémentation des VLANs a permis d'adapter l'infrastructure aux besoins spécifiques des utilisateurs, garantissant ainsi une communication fluide et sécurisée entre les différentes entités du réseau.

```
interface Vlan-interfacel106
description SCES-ACAD
ip binding vpn-instance VRF-ACAD
ip address 10.1.1.24 255.255.255.24
#
interface GigabitEthernet1/0/1
port link-mode bridge
description UpLink_FO_SR-1-0
port link-type trunk
port trunk permit vlan all
#
interface GigabitEthernet1/0/2
port link-mode bridge
description UpLink_FO_SR-2-1
port link-type trunk
port trunk permit vlan all
#
interface GigabitEthernet1/0/3
port link-mode bridge
description UpLink_FO_SR-5-0
port link-type trunk
port trunk permit vlan all
```

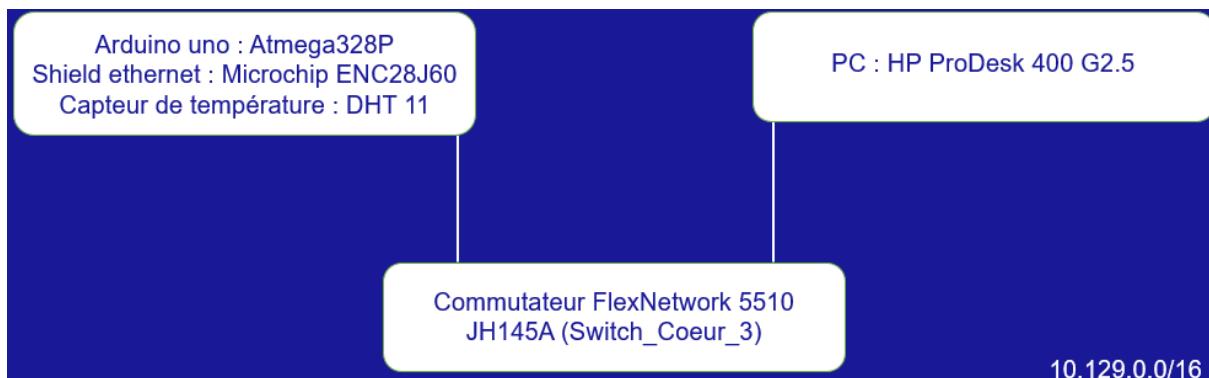
Interfaces du commutateur de cœur de réseau

Les étapes suivantes permettront de créer des VLANs et d'en attribuer à des interfaces spécifiques :

- System-view
- vlan <VLAN_ID>, name <VLAN_NAME> (creation du vlan)
- interface vlan <VLAN_ID>, ip adresse <IP_ADDRESS> <SUBNET_MASK> (liaison d'une adresse ip avec le vlan)
- interface range fa0/1 – 24 (sélection des interfaces de 1 à 24)
- switchport mode access (mise en place en mode access des interfaces)
- switchport access vlan <VLAN_ID> (inscription du vlan sur les interfaces)
- Write memory (sauvegarde)

e. Gestion de projet et programmation

Pour finaliser la liste d'activités effectuées au sein du lycée Louis Couffignal de Strasbourg, la gestion de projet a occupé une place prépondérante durant ma dernière année, avec environ 50 heures consacrées à la gestion du projet lui-même, et 150 heures allouées à la partie technique et pratique.



La programmation du microcontrôleur a permis de détecter les variations et d'envoyer les données au serveur Mosquitto en utilisant MQTT. L'installation et la configuration du serveur Proxmox sur le HP ProDesk 400 G2.5 ont été des étapes fondamentales de ce projet, suivies par la mise en place d'une machine virtuelle avec le protocole MQTT et le service Node-RED. Ce dernier a été configuré avec des certificats OpenSSL afin de sécuriser les communications. Le protocole SMTP et la base de données (Maria DB) ont également été intégrés via Node-RED pour gérer l'envoi d'emails lors d'une température anormale et pour stocker les informations issues du capteur. Cette partie a duré environ cent cinquante heures.

Enfin, des documentations partagées ont été créées pour détailler chaque étape et des supports informatiques ont été préparés pour expliquer clairement le fonctionnement de l'ensemble du projet. Cette partie a duré une cinquantaine d'heure.

Quant à la programmation du microcontrôleur, elle a été rédigée, ajustée et testée afin de garantir le **bon** fonctionnement du capteur ainsi que de l'ensemble du projet. Ce capteur, étant la pièce maîtresse, devait être parfaitement opérationnel pour assurer la réussite du projet.

4. Présentation du projet

a. Contraintes techniques et économiques

Réglementation entreprise : Utilisation obligatoire du plan des plages d'adresses IP organisé par la Région GrandEst mais également de la sécurité.

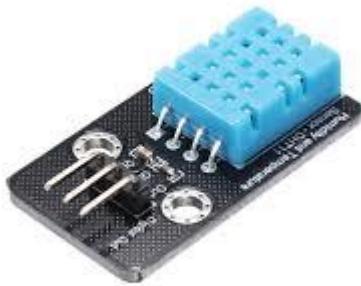
Contraintes économiques à respecter :

- **Budget** : Pour la totalité des parties du capteur, la section des BTS CIEL présente au 3^{ème} étage du bâtiment C du lycée Couffignal les possède déjà. Le serveur Proxmox, le poste VPN ainsi que les commutateurs, la Région Grand Est les possède également grâce à une ancienne dotation informatique.
- **Coûts de développement** : 0€. Tout sera réalisé par mes soins avec l'aide de certains collègues.
- **Coûts de maintenance** : 0€. La maintenance sera effectuée en interne par l'administrateur réseau du lycée COUFFIGNAL.
- **Contraintes de temps** : Environ 150 heures ont été nécessaires pour la réalisation technique du projet et une cinquantaine d'autre pour la partie rédaction de documentations et de support pour la présentation de ce projet.

Contrainte technique : L'administration des réseaux centralisée par la Région GrandEst ainsi que la recherche de ressources pour la configuration des appareils.

Contraintes liées aux risques et imprévus : Il est possible lors de l'accomplissement du projet que des problèmes et des imprévus arrive. Cela peut impacter le temps passé sur le projet ou encore une réparation ou changement de matériel.

b. Matériels utilisés

Type de matériels	Caractéristique technique
	<ul style="list-style-type: none"> • Microcontrôleur : Atmega328P • Tension de fonctionnement : 5V • Tension d'entrée (recommandée) : 7-12V. • Broches numériques d'entrée/sortie : 14 (dont 6 peuvent être utilisées comme sorties PWM) • Broches d'entrée analogiques : 6 • Sensibilité : $0.6 * V_{cc}$ (3V pour $V_{cc} = 5V$)
	<ul style="list-style-type: none"> • Capteur : DHT 11 • Gamme de mesure de température : 0 à 50 °C • Précision de mesure de température : ± 2 °C • Tension d'alimentation : 3.3 à 5.5V • Courant de fonctionnement : 0.3 mA (en mode de mesure) 60 µA (en mode veille) • Type de sortie : Signal numérique unique sur une broche • Sensibilité : 1°C
	Commutateur FlexNetwork 5510 JH145A (Switch_Coeur_3).
	<p>Commutateur FlexNetwork 5130 JG932A (SR-C009-DEPLOY).</p> <p>Sur ce commutateur est branché le poste créé pour assurer une connexion à distance.</p>
	Commutateur intermédiaire en C-011 FlexNetwork 5130 JG932A (SR-5C-2-0).

Type de matériels	Caractéristique technique
	<ul style="list-style-type: none"> Contrôleur Ethernet : Microchip ENC28J60 Interface : SPI (Serial Peripheral Interface) Tension de fonctionnement : 5V Connecteur RJ45 : Pour la connexion au réseau Ethernet Vitesse de communication : 10 Mbps Protocole de communication : TCP/IP, UDP, HTTP, etc.
	HP Pro Desk 400 G2.5 (Serveur Proxmox)
	<p>HP Pro Desk 400 G2.5 (VPN-MQTT)</p> <p>Ce poste a été configuré pour une connexion à distance.</p>

c. Diagrammes

L'image correspond au digramme de déploiement mais également du synoptique réseau de mon projet informatique, il comporte tous les outils, logiciels, protocoles et matériels nécessaires au bon déroulement et à la réussite de mon projet.

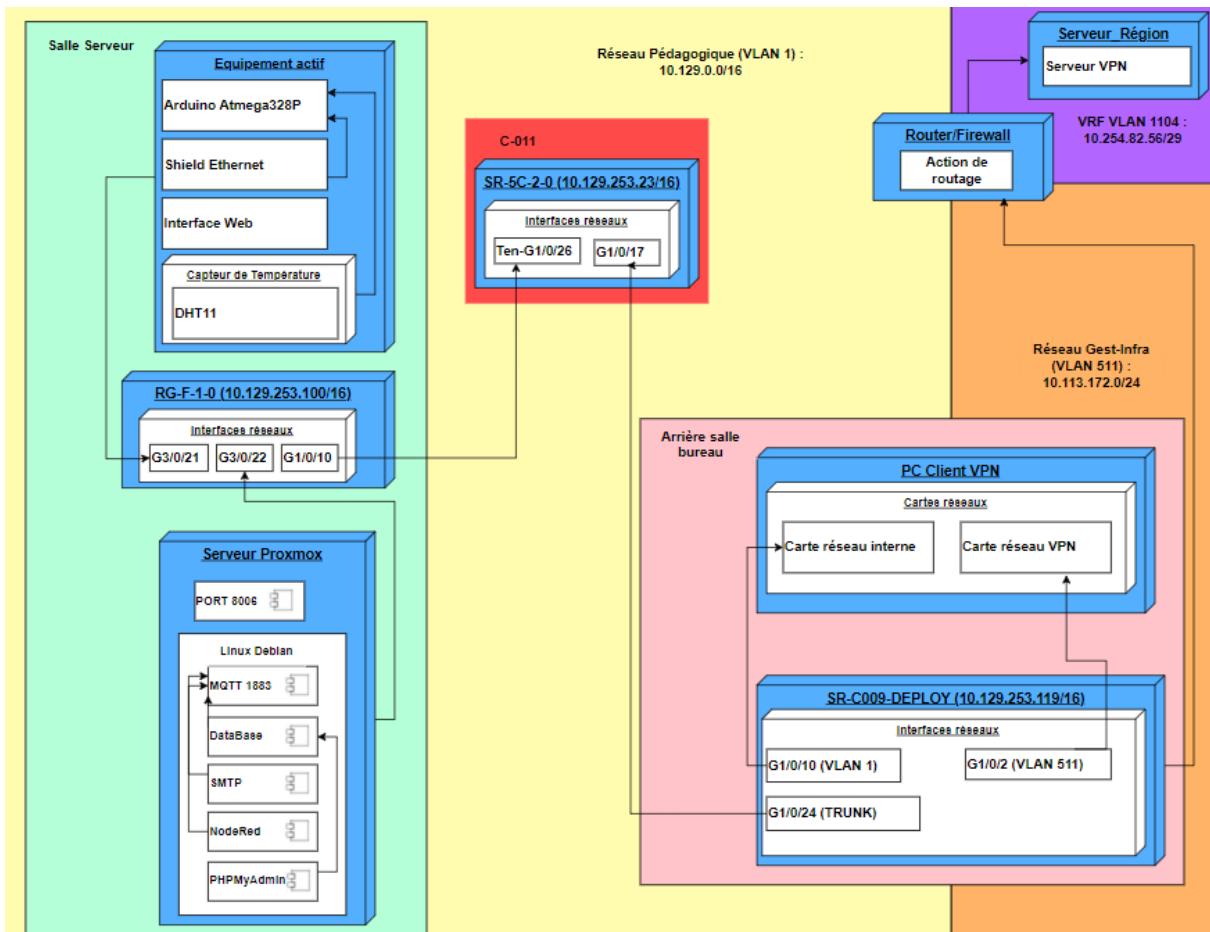
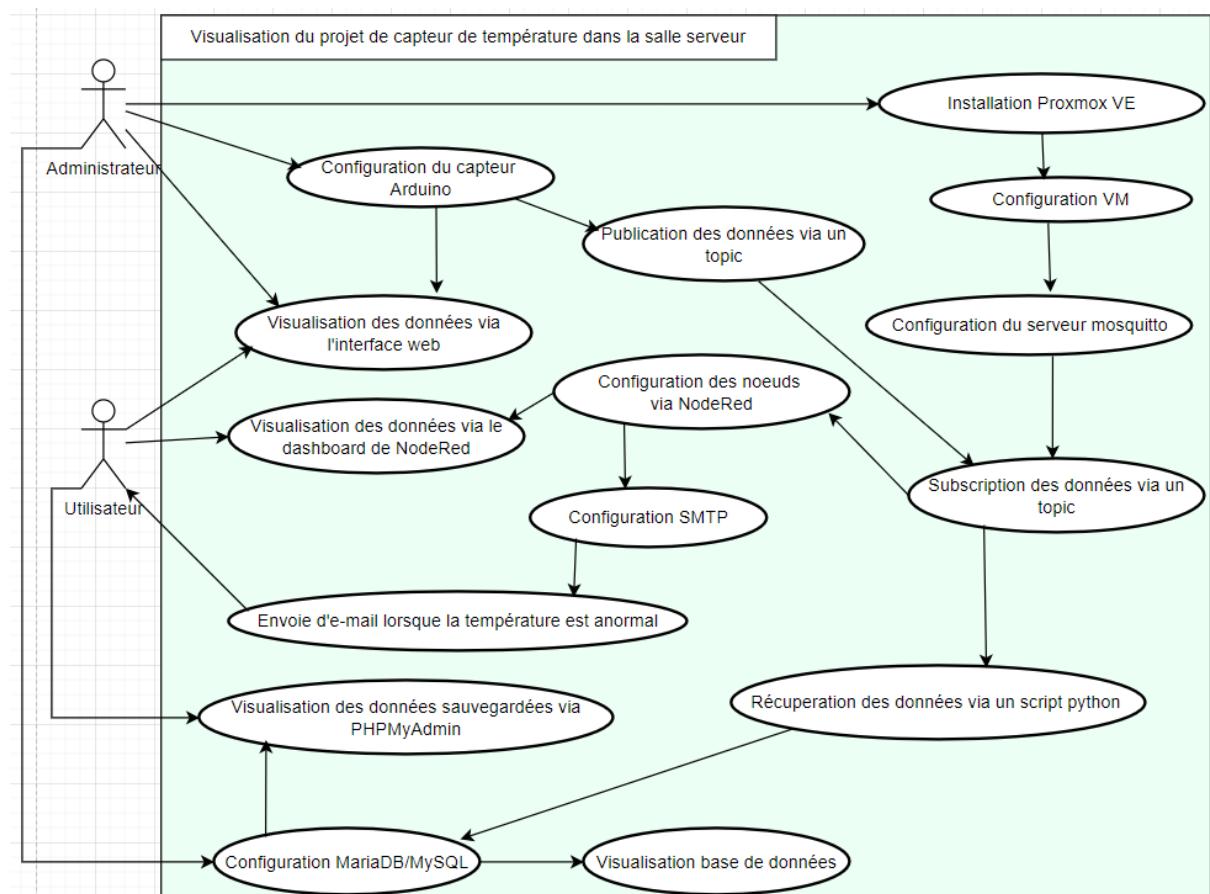


Diagramme de déploiement & synology du projet

Ce diagramme de cas d'utilisation permet de visualiser le fonctionnement complet du projet.



Ce diagramme de séquence est essentiel pour modéliser les interactions dynamiques de mon système.

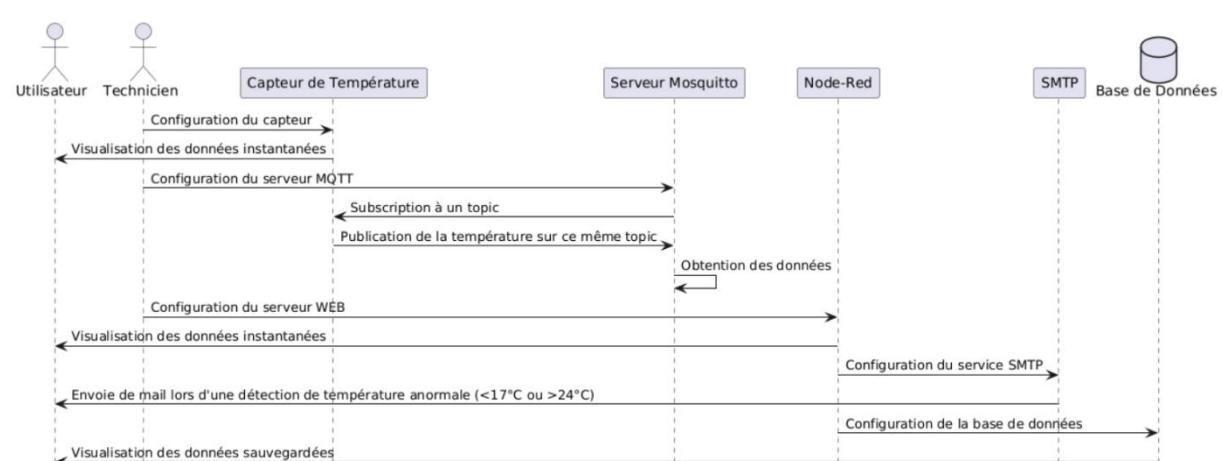


Diagramme de Séquence

Explication du Gantt :

Segments	Description
1 à 2	Analyse des besoins de l'entreprise et repérage de l'infrastructure réseau du projet
3 à 5	Rédaction, amélioration et vérification du cahier des charges
6	Configuration switchs
7 à 10	Concentration sur le capteur (programmation, test en local, requête ICMP, cahier de recette).
11 à 15	Concentration sur le serveur Proxmox (installation, configuration, vérification et test, création de groupes, d'utilisateurs et de permissions, cahier de recette).
17 à 20	Concentration sur la machine virtuel (initialisation, configuration réseau, vérification et test, cahier de recette).
21 à 25	Concentration sur la mise en place du protocole MQTT (ressources, configuration, vérification et test, cahier de recette).
26 à 29	Concentration sur le VPN (Virtual Private Network) (configuration, vérification et test, cahier de recette).
30 à 31	Préparation et oral de la revue 2 au lycée COUFFIGNAL.
32 à 37	Concentration sur la mise en place de Node-Red (ressources, installation, configuration, programmation, rectification des incidents, vérification et test, cahier de recette).
38 à 41	Concentration sur les certifications OpenSSL de Node-Red (ressources, installation, configuration, vérification et test, cahier de recette).
42 à 47	Concentration sur le protocole SMTP via Node-Red (ressources, installation, configuration, vérification et test, cahier de recette).
48 à 49	Préparation et oral de la revue 1 à l'UIMM Alsace.
50 à 55	Concentration sur MariaDB (ressources, installation, script python, configuration, vérification et test, cahier de recette).
56 à 62	Concentration sur PHPMyAdmin (ressources, installation, configuration, vérification et test, cahier de recette).
63 à 65	Rédaction des dernières documentation et finalisation du mémoire
66 à 67	Préparation et oral de la revue 3 au lycée COUFFIGNAL.
68 à 69	Préparation et oral de la soutenance à l'UIMM Alsace.

5. Introduction

Actuellement, le lycée Louis COUFFIGNAL de STRASBOURG ne dispose d'aucun capteur de température dans sa salle serveur alors que cela serait indispensable. Les différentes personnes ayant accès aux serveurs physiques n'ont absolument aucune solution connectée permettant de contrôler à distance les paramètres physiques de cette salle. L'ensemble de ces paramètres ont un impact direct sur la durée de vie des appareils. Ce projet a pour but de développer/programmer et installer un capteur connecté de température qui sera positionné dans la salle serveur du lycée COUFFIGNAL et en cas de dysfonctionnement de la climatisation ou simplement d'une hausse drastique de la température dans la pièce, des alertes seront émises. Il sera fixé à hauteur moyenne pour mesurer de façon optimale la température dans de bonnes conditions de fonctionnement car la climatisation fonctionne sur les hauteurs de la pièce ce qui pourrait fausser les données.

Le projet consiste à créer une maquette avec un capteur de température connecté relié à un système informatique permettant de générer des alarmes en cas de dépassement de seuil, de gérer ces alarmes afin d'éviter des arrêts non-contrôlés des systèmes informatiques. La solution technique vise à mettre en oeuvre un broker de données de type serveur Mosquitto¹ sur une machine Linux Debian hébergée par Proxmox². Le capteur connecté sera réalisé avec un Arduino Uno (Atmega328P) équipé d'un Shield Ethernet (Microchip ENC28J60) et d'un capteur de température DHT11 qui se connectera à ce serveur MQTT. L'objectif du projet consiste à développer un système de collecte de données de température de la salle serveur. Le système sauvegardera dans une base de données les informations transmises de ce capteur. Une interface Node-Red sera mis en œuvre qui permettra d'afficher les températures en temps réel et de consulter l'historique des données de températures. Un système d'alerte déclenchera des événements en cas de dépassement de seuil.

¹Un serveur Mosquitto est un broker MQTT (Message Queuing Telemetry Transport) qui facilite la communication entre différents dispositifs dans un réseau IoT (Internet of Things).

²Un Proxmox est principalement utilisé pour gérer des environnements virtualisés et offre des fonctionnalités avancées pour la gestion de machines virtuelles.

Tous fichiers/documents pris durant le développement de ce projet seront uploadées à l'URL suivant : <https://github.com/AnthonyKTTR67/projet>.

⚠ Cependant, pour des raisons de cybersécurité, vous n'aurez pas accès à ce dossier.

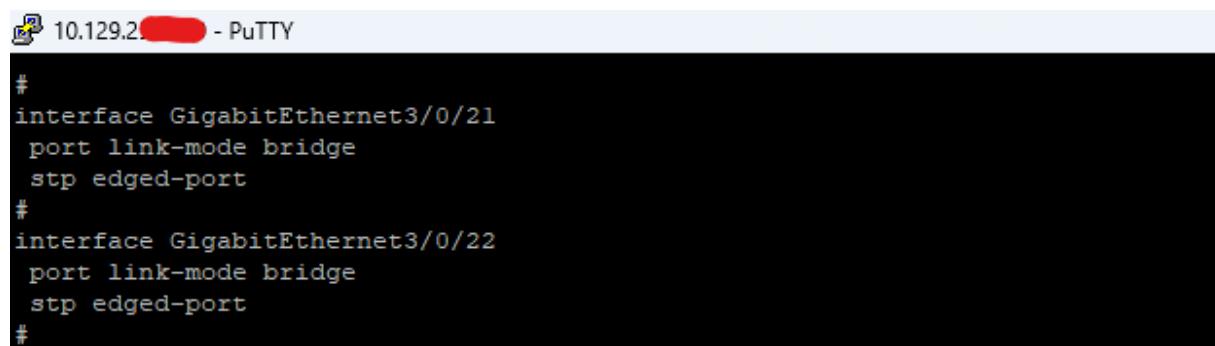
Un autre dossier GitHub a été créé pour une meilleure visualisation des photos et/ou captures d'écrans présentes dans ce document :

https://github.com/AnthonyKTTR67/Projet.Photo_Report/.

6. Configuration switch et partie capteur de température

a. Configuration des interfaces du switch de cœur de réseau

Pour permettre la communication entre des appareils connectés à des switchs, il est nécessaire de configurer initialement les interfaces de ces switchs. Cette configuration est cruciale pour établir une communication réseau efficace et sécurisée entre les différentes machines de mon projet.



```
#  
interface GigabitEthernet3/0/21  
port link-mode bridge  
stp edged-port  
#  
interface GigabitEthernet3/0/22  
port link-mode bridge  
stp edged-port  
#
```

Interface capteur et serveur Proxmox

Le mode "link-mode bridge" sur un port signifie que le port fonctionne comme un pont (bridge) en réseau, permettant la communication entre deux segments de réseau sans faire de routage. Par défaut, tous les ports sont dans le VLAN 1 et fonctionnent en mode access, sauf indication contraire. Le comportement du port en mode bridge est influencé par l'adresse MAC, car le switch utilise l'adresse MAC pour apprendre et aiguiller les paquets Ethernet correctement.

Le commandement stp edged-port permet de configurer le port actuel en tant que port de bordure (edge port) dans un réseau utilisant le protocole Spanning Tree Protocol (STP). Un port de bordure est utilisé pour connecter des terminaux tels que des ordinateurs, des imprimantes ou encore des capteurs, ce qui accélère la convergence du réseau et améliore sa stabilité.

L'interface GigabitEthernet3/0/21 est connectée au capteur de température, tandis que le serveur est connecté à l'interface GigabitEthernet3/0/22.

b. Programmation du microcontrôleur Arduino Uno

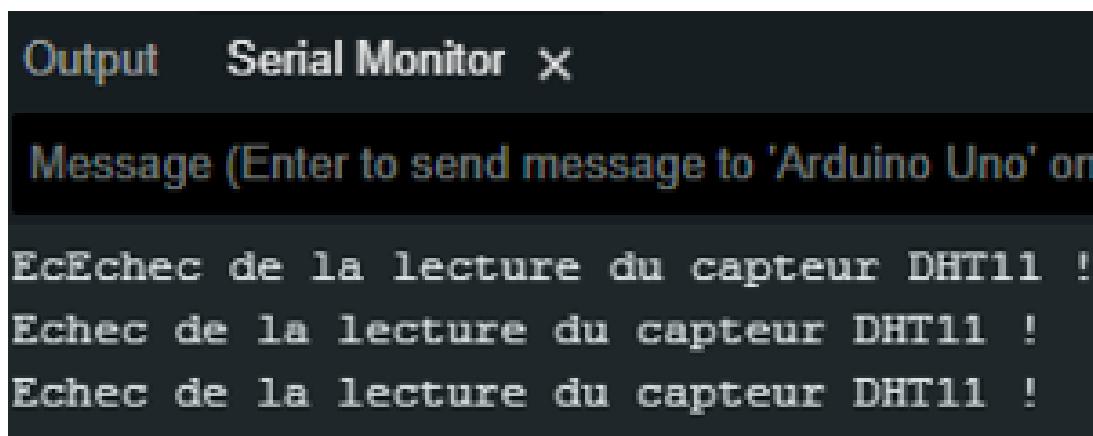
I. Explication du capteur

La programmation du microcontrôleur vise à garantir la collecte, le traitement et la transmission des données provenant du capteur DHT11 par le biais d'une connexion réseau, offrant ainsi une supervision constante et en temps réel des températures dans la salle serveur. La première étape de cette programmation consiste à effectuer des recherches en ligne, à trouver des ressources et des documentations qui faciliteraient la réalisation de connexions appropriées, la programmation et la compréhension du langage employé, ici C++.

L'assemblage des trois appareils a été appliqué convenablement en faisant très attention de ne pas inverser les broches du capteur avec celles du microcontrôleur car dans le pire des cas, le capteur peut extrêmement chauffer et devenir inutilisable. Le branchement est donc le suivant :

- VCC (Voltage Common Collector) : Connecté au 5V de l'Arduino.
- GND (Ground) : Connecté à la masse (GND) de l'Arduino.
- DATA : Connecté à la broche 3 numérique.

Après avoir effectué le raccordement, l'installation d'Arduino IDE a été réalisée. C'est un environnement de développement intégré permettant d'écrire, de compiler et de téléverser du code sur le microcontrôleur. Cette installation était essentielle pour débuter la programmation du capteur car elle offre une interface simplifiée et des bibliothèques adaptées à l'Arduino Uno. Le code C++ n'a été écrit qu'après cette étape. Je l'ai ensuite modifié à plusieurs reprises afin de corriger les erreurs présentes dans le programme, de le compiler correctement et de l'implémenter sur l'Arduino Uno. Cependant, après plusieurs essais, le microcontrôleur n'affichait aucune température, mais un message d'erreur défini dans le programme.



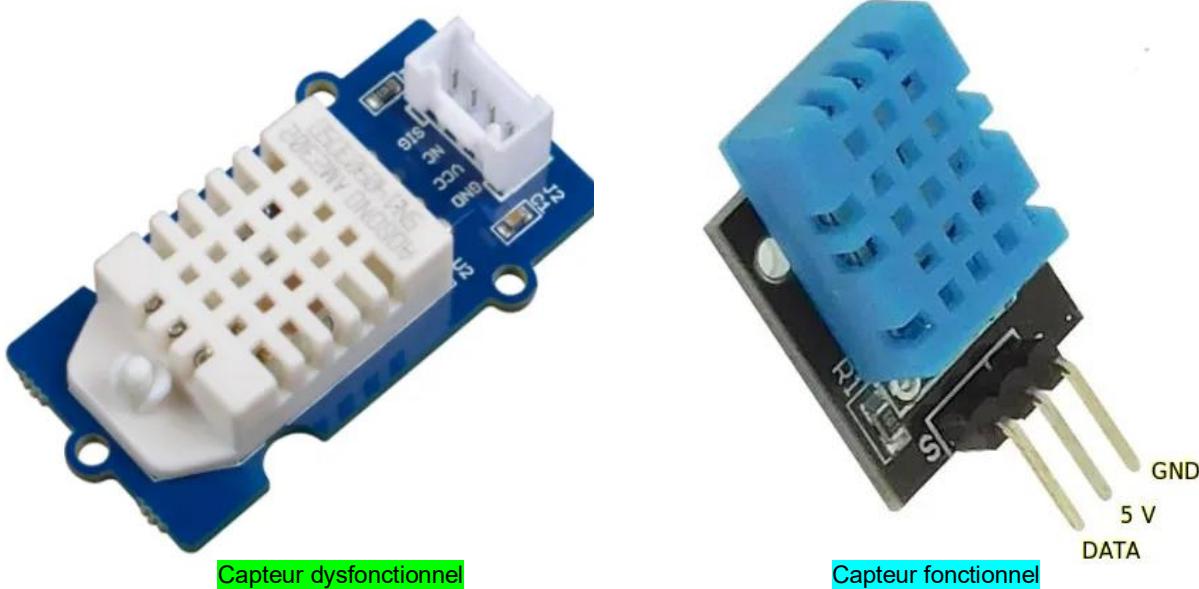
```
Output  Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on

EcEchec de la lecture du capteur DHT11 !
Echec de la lecture du capteur DHT11 !
Echec de la lecture du capteur DHT11 !
```

Message d'erreur lors de la visualisation des données

J'ai ensuite testé le capteur en utilisant le script préexistant dans l'Arduino IDE, intitulé "DHTtester.ino", qui permet de vérifier le bon fonctionnement du capteur. Il s'avérait que le capteur était dysfonctionnel, j'en ai donc demandé un autre au PRN (Professeur Référent Numérique) du lycée. Étant donné que ce deuxième capteur fonctionne, le programme a pu être téléchargé.



Le capteur dysfonctionnel appartient à l'écosystème Grove et dispose d'un connecteur à quatre broches (**VCC** : broche d'alimentation du capteur, **GND** : broche de masse, **DATA** : broche de communication qui transmet les mesures de température et **NC** : ne possède aucune fonction spécifique), facilitant une connexion rapide sans soudure. Ce format est particulièrement adapté aux cartes de développement utilisant le système modulaire Grove, comme celles de Seeed Studio. Tandis que le capteur fonctionnel est un module classique avec trois broches (**GND**, **VCC**, **DATA**), conçu pour être directement branché sur une carte comme Arduino. Il est souvent utilisé pour des projets nécessitant un câblage simple. Ce second capteur est donc beaucoup plus adapté à ce projet.

II. Explication du programme

Dans cette partie, l'explication du programme utilisé est présentée. Des captures d'écran seront affichées et détaillées dans le but d'une totale compréhension.

```
Script_DHT11_officiel.ino
1  #include <DHT.h>
2  #include <SPI.h>
3  #include <Ethernet.h>
4  #include <PubSubClient.h>           // Inclure les bibliothèques
```

Bibliothèques utilisées

Voici les bibliothèques utilisées dans le programme C++ du capteur. Elles permettent d'inclure un ensemble de fonctions, de classes et d'objets nécessaires à la programmation de mon capteur. A quoi servent-elles ?

- **DHT.h** : elle permet de communiquer facilement avec des capteurs de température et d'humidité de type DHT11. Elle fournit des fonctions pour lire les valeurs mesurées par le capteur.
- **SPI.h** : elle sert à gérer la communication SPI (Serial Peripheral Interface), un protocole utilisé pour échanger des données rapidement entre l'Arduino et la carte Ethernet.
- **Ethernet.h** : elle permet au microcontrôleur de se connecter à un réseau local ou à Internet via un module Ethernet. Elle facilite l'obtention d'une adresse IP, l'envoi de requêtes ou la réception de données réseau.
- **PubSubClient.h** : elle est utilisée pour établir une communication MQTT. C'est grâce à elle qu'un appareil peut se connecter à un broker MQTT, publier des messages ou s'abonner à des topics.

```
6  // Configuration du DHT11
7  #define DHTPIN 3                  // Définition de la broche de connexion du capteur DHT11
8  #define DHTTYPE DHT11            // Définition du type de capteur (DHT11)
9  DHT dht(DHTPIN, DHTTYPE);      // Initialisation du capteur DHT11
10
11 // Configuration du réseau Ethernet
12 byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0x6F, 0x4D }; // Adresse MAC unique pour l'Ethernet Shield
13 IPAddress ip(10, 129, 251, 1);                         // Adresse IP de l'Arduino
14 EthernetServer server(80);                            // Serveur web sur le port 80
15
```

Configuration DHT11 et carte Ethernet

Cette image montre la configuration du capteur de température DHT11 (sur quelle broche de l'Arduino il est branché, quelle type de capteur c'est (DHT11) et l'initialisation de celui-ci). Elle montre également l'adresse MAC du shield Ethernet, son adresse ip et le port utilisé lors d'une visualisation des données via une interface web.

```

16 // Configuration du client Ethernet
17 EthernetClient ethernetClient; // Client Ethernet pour se connecter à MQTT
18
19 // Configuration du client MQTT
20 IPAddress mqttServer(10, 129, 251, 253); // Adresse IP du serveur MQTT
21 const int mqttPort = 1883; // Port par défaut du serveur MQTT
22 PubSubClient clientMQTT(ethernetClient); // Initialisation du client MQTT avec EthernetClient
23
24 // Dernière température mesurée
25 float lastTemperature = 0.0;
26
27 // Variables pour contrôler la fréquence des mises à jour MQTT
28 unsigned long lastMQTTSendTime = 0; // Temps de la dernière publication MQTT
29 const unsigned long mqttInterval = 5000; // Intervalle entre chaque publication (5 secondes)

```

Configuration MQTT

Cette partie du programme utilise la carte Ethernet pour lire la température du capteur DHT11. Il se connecte au serveur MQTT (10.129.251.253 :1883). La dernière température lue est stockée dans la variable ‘lastTemperature’. Les données de température sont ensuite publiées sur le serveur MQTT à intervalles réguliers de 5 secondes, grâce aux variables ‘lastMQTTSendTime’ et ‘mqttInterval’.

```

31 void setup() {
32   Serial.begin(9600); // Initialisation de la communication série pour le débogage
33   Serial.println("Initialisation...");
34   dht.begin(); // Initialisation du capteur DHT11
35   Serial.println("Capteur DHT initialisé.");
36
37 // Initialisation de l'Ethernet Shield avec l'adresse MAC et l'adresse IP
38 Ethernet.begin(mac, ip);
39 server.begin();
40 Serial.print("Serveur web démarré sur IP : "); // Afficher l'adresse IP de l'Arduino
41 Serial.println(Ethernet.localIP()); // Affichage de l'adresse IP du serveur dans le moniteur série
42
43 // Configurer le client MQTT
44 clientMQTT.setServer(mqttServer, mqttPort);
45 }

```

Fonction setup()

La fonction `setup()` initialise les différents composants. Elle démarre la communication série à 9600 bauds pour le débogage. Ensuite, elle initialise le capteur DHT11 (`dht.begin()`) et affiche un message de confirmation. L'Ethernet Shield est initialisé avec l'adresse MAC (`mac`) et l'adresse IP (`ip`) définies, puis le serveur web est démarré (`server.begin()`), affichant son adresse IP. Enfin, elle configure l'adresse et le port du serveur MQTT pour le client MQTT (`clientMQTT.setServer()`).

```

47 void loop() {
48     // Assurez-vous que la connexion MQTT est active
49     if (!clientMQTT.connected()) {
50         reconnectMQTT(); // Reconnexion au serveur MQTT si nécessaire
51     }
52     clientMQTT.loop(); // Maintenir la connexion avec le serveur MQTT
53
54     // Vérifiez si le temps d'envoyer les données au MQTT est atteint
55     if (millis() - lastMQTTSendTime > mqttInterval) {
56         sendTemperatureToMQTT(); // Envoyer la température au serveur MQTT
57         lastMQTTSendTime = millis();
58     }
59     handleWebServer();
60 }

```

Fonction principale loop()

La fonction `loop()` est la boucle principale du programme. Elle vérifie en permanence si le client MQTT est connecté (`!clientMQTT.connected()`) et tente une reconnexion si ce n'est pas le cas en appelant `reconnectMQTT()`. Ensuite, `clientMQTT.loop()` est appelée pour maintenir la connexion avec le serveur MQTT et gérer les messages entrants/sortants. Elle vérifie également si l'intervalle de temps défini (`mqttInterval`) depuis le dernier envoi MQTT est écoulé ; si oui, elle appelle `sendTemperatureToMQTT()` pour publier la température et met à jour le temps du dernier envoi. Enfin, elle appelle `handleWebServer()` pour gérer les requêtes HTTP entrantes.

```

62 // Fonction pour envoyer la température au serveur MQTT
63 void sendTemperatureToMQTT() {
64     // Lecture de la température depuis le capteur DHT11
65     float temp = dht.readTemperature(); // Récupère la température mesurée
66
67     if (!isnan(temp)) { // Vérifie que la température est une valeur valide
68         lastTemperature = temp; // Stocke la dernière température valide
69
70         // Convertir la température en chaîne de caractères pour l'envoi MQTT
71         char tempStr[8]; // Tableau pour stocker la chaîne formatée
72         dtostrf(temp, 2, 2, tempStr); // Convertit le float en string (1 chiffre avant la virgule, 2 après)
73
74         // Publier la température sur le topic "temp" du serveur MQTT
75         if (clientMQTT.publish("temp", tempStr)) { // Envoie la température
76             Serial.println("Température de la salle serveur envoyée à MQTT : " + String(temp) + " °c");
77         } else {
78             Serial.println("Erreur lors de l'envoi de la température."); // Affiche une erreur si l'envoi échoue
79         }
80     } else {
81         Serial.println("Échec de la lecture du capteur DHT11 !"); // Affiche une erreur si la lecture est invalide
82     }
83 }

```

Envoie données vers MQTT

La fonction `sendTemperatureToMQTT()` lit la température du capteur DHT11 (`dht.readTemperature()`). Elle vérifie si la lecture est valide (`!isnan(temp)`) et, si c'est le cas, stocke la température dans `lastTemperature`. Elle convertit ensuite cette température en une chaîne de caractères (`tempStr`) avec deux chiffres après la virgule. Enfin, elle tente de publier cette chaîne sur le topic "temp" du serveur MQTT et affiche un message de succès ou d'erreur dans le moniteur série. Si la lecture du capteur échoue, un code d'erreur spécifique est affiché.

```

85 // Fonction pour reconnecter le client MQTT si déconnecté
86 void reconnectMQTT() {
87     while (!clientMQTT.connected()) { // Boucle tant que le client MQTT n'est pas connecté
88         Serial.print("Tentative de connexion MQTT..."); // Affiche un message de tentative
89
90         String clientId = "arduinoClient-"; // Préfixe pour l'ID client MQTT
91         clientId += String(random(0xffff), HEX); // Ajoute une partie aléatoire pour éviter les conflits d'ID
92
93         if (clientMQTT.connect(clientId.c_str())) { // Tente de se connecter avec l'ID généré
94             Serial.println("Connecté au serveur MQTT."); // Affiche un message si la connexion réussit
95         } else {
96             Serial.print("Échec de la connexion, état : "); // Affiche une erreur si la connexion échoue
97             Serial.println(clientMQTT.state()); // Affiche le code d'état de la tentative
98             delay(5000); // Attend 5 secondes avant de réessayer
99         }
100    }
101 }
102 }
```

Fonction reconnectMQTT()

La fonction `reconnectMQTT()` tente de reconnecter le client MQTT au serveur tant que la connexion n'est pas établie (`while (!clientMQTT.connected())`). Elle affiche un message de tentative de connexion. Un ID client unique est généré en ajoutant une partie aléatoire à la chaîne "arduinoClient-". La fonction essaie ensuite de se connecter au serveur MQTT avec cet ID. En cas de succès, un message de connexion réussie est affiché. En cas d'échec, un message d'erreur indiquant l'état de la connexion est affiché, et le programme attend 5 secondes avant de réessayer de se connecter.

```

103 // Fonction pour gérer les requêtes HTTP et afficher une interface web
104 void handleWebServer() {
105     // Vérifier si un client est connecté
106     EthernetClient client = server.available();
107     if (client) {
108         Serial.println("Client connecté !");
109
110         // Envoi de l'en-tête HTTP 200 OK pour indiquer que la connexion a réussi
111         client.println("HTTP/1.1 200 OK");
112         client.println("Content-Type: text/html; charset=UTF-8"); // Définition du type de contenu HTML
113         client.println("Connection: close"); // Indiquer que la connexion se ferme après la réponse
114         client.println();
115
116         // Génération du contenu HTML affiché sur la page web
117         client.println("<html>");
118         client.println("<head>");
119         client.println("<title>Données du capteur DHT11</title>");
120         client.println("<style>");
121         client.println("body { background-color: #191998; font-family: Arial, sans-serif; text-align: center; padding: 50px; }");
122         client.println("h1 { font-size: 45px; color: #FFFFFF; }");
123         client.println("p { font-size: 40px; color: #FFFFFF; }");
124         client.println("</style>");
125         client.println("</head>");
126         client.println("<body>");
127         client.println("<h1>Données du capteur DHT11</h1>");
128         client.print("<p>Il fait actuellement ");
129         client.print(lastTemperature); // Utilisation de la variable globale lastTemperature
130         client.println(" °C dans la salle serveur du lycée Louis COUFFIGNAL</p>");
131         client.println("</body>");
132         client.println("</html>");
133
134         delay(1); // Pause pour s'assurer que le client a bien reçu les données
135         client.stop(); // Déconnexion du client
136         Serial.println("Client déconnecté.");
137     }
138
139     delay(1000); // Attente de 1 seconde avant la prochaine lecture du capteur
140 }
```

Programmation du serveur web local

La fonction handleWebServer() gère les requêtes HTTP entrantes. Elle vérifie d'abord si un nouveau client s'est connecté (server.available()). Si c'est le cas, elle affiche un message de connexion dans le moniteur série. Ensuite, elle envoie l'en-tête HTTP "200 OK" indiquant que la requête a réussi, ainsi que le type de contenu "text/html" et l'instruction de fermer la connexion après l'envoi. Elle génère ensuite le contenu HTML de la page web, affichant un titre et la température actuelle stockée dans la variable lastTemperature. Après avoir envoyé le contenu, elle introduit un court délai et ferme la connexion avec le client (client.stop()), affichant un message de déconnexion dans le moniteur série. Enfin, un délai d'une seconde est introduit avant de passer à la prochaine reprise de la boucle principale.



Interface web du capteur

Cette image illustre l'interface web générée par le programme Arduino. Elle permet de visualiser les données collectées même en cas de non fonctionnement du serveur.

Une fois les appareils reliant le capteur installés et le microcontrôleur correctement programmé pour la détection et l'envoi des températures, un cahier de recette servant de documentation a été rédigé. Il regroupe chaque étape de la mise en place de la partie capteur de température.

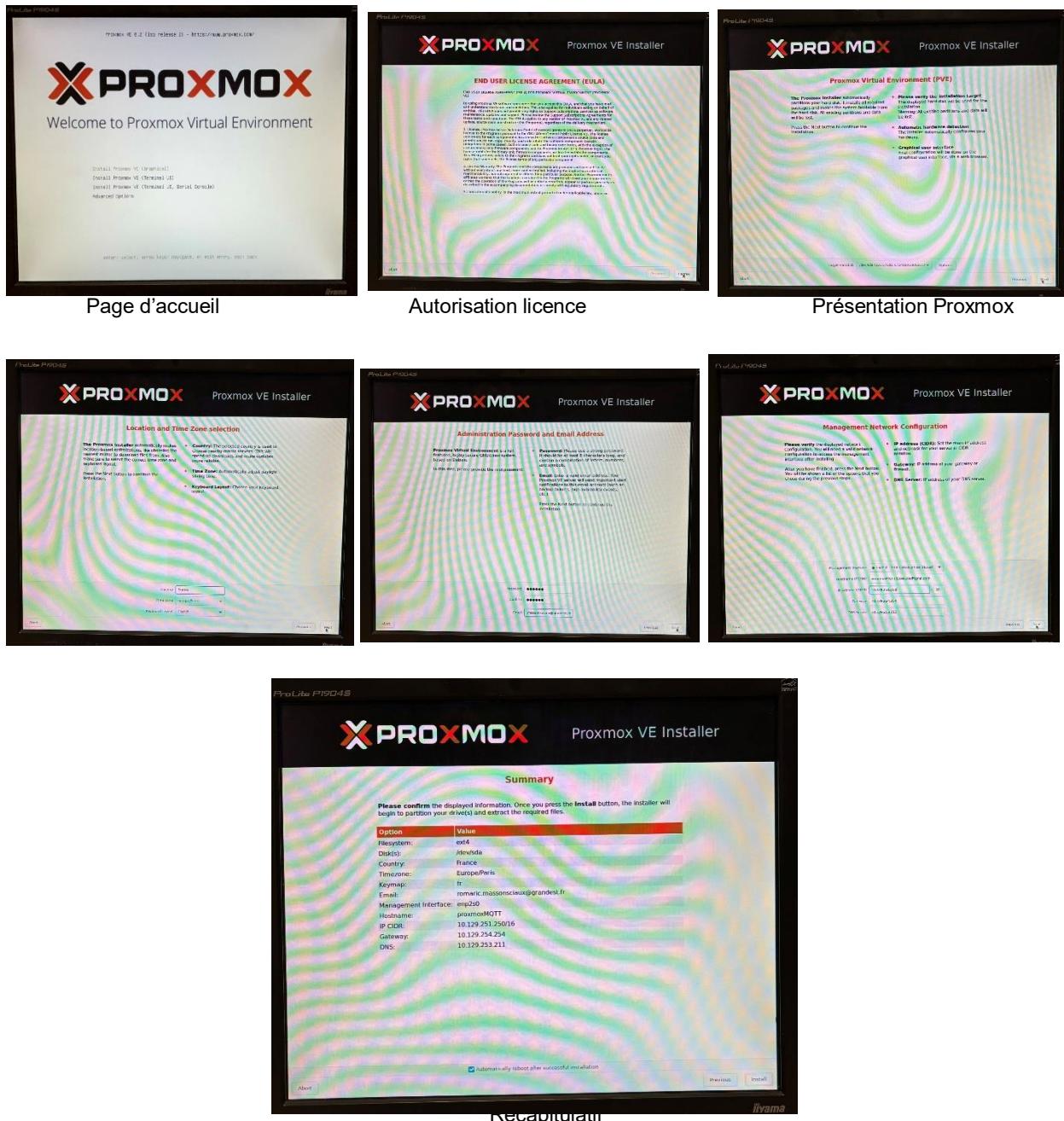
¹Le fichier de programmation du capteur ‘Script_DHT11_officiel.ino’ est disponible sur le GitHub dédié.

7. Proxmox VE (Virtual Environnement)

a. Installation et configuration

Le serveur Proxmox a été installé et configuré sur un poste d'une ancienne dotation informatique, le HP Pro Desk 400 G2.5 auquel a été rajouté 8Gb de RAM (2 barrettes de 4Gb, DDR3, 12800U) ainsi qu'un SSD de 240GB. Cela peut paraître peu mais fera parfaitement l'affaire.

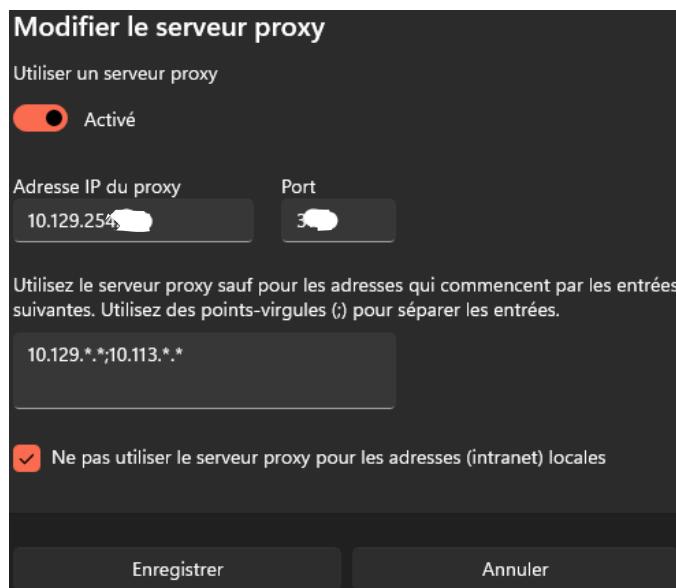
Une fois le poste branché électriquement, relié au réseau du lycée et dont la clé bootable créée avec Rufus contient l'ISO d'un serveur Proxmox est connecté au serveur, nous pouvons booter sur cette clé en pressant F12. Voici les différentes étapes :



b. Interface web de Proxmox

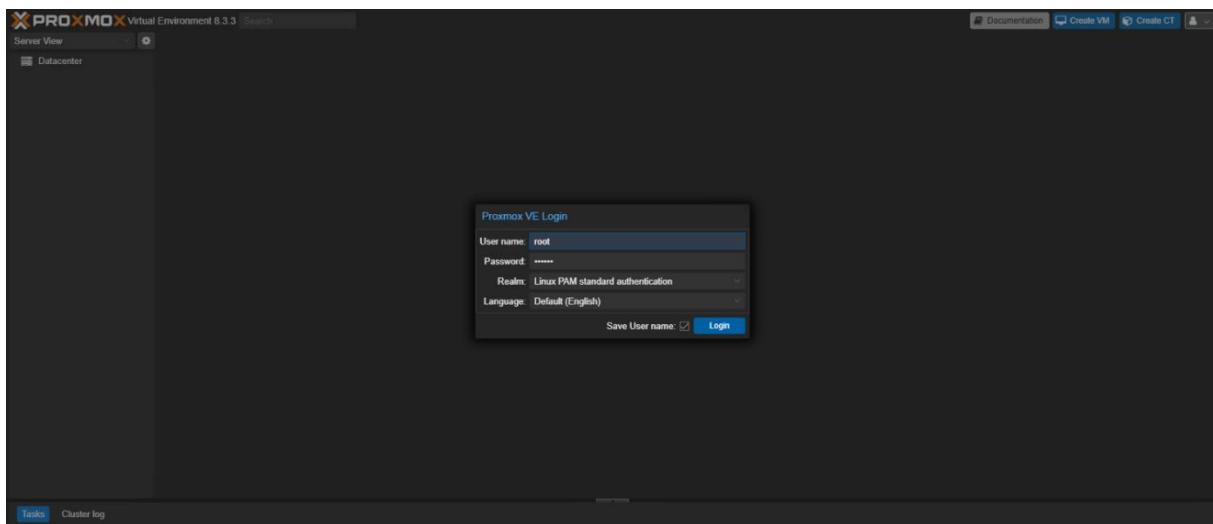
Une fois le serveur Proxmox installé et configuré sur un poste, il est accessible via son interface web en passant par un poste présent sur le même réseau local. Pour cela, il suffit d'ouvrir un navigateur et d'entrer l'adresse IP locale du serveur Proxmox suivie du port 8006 (port par défaut), <https://10.129.251.250:8006>. Cette interface permet d'administrer les machines virtuelles, gérer les ressources du serveur et configurer les différents services. Il est important d'accepter le certificat SSL auto-signé lors de la première connexion si aucun certificat valide n'a été installé. Il est important lors de la connexion de correctement configurer le serveur proxy ou de le désactiver.

Un proxy agit comme un intermédiaire entre un client et un serveur pour améliorer la sécurité, l'anonymat et les performances du réseau. Il permet de masquer l'adresse IP, filtrer les accès, et accélérer le chargement des pages grâce au caching.



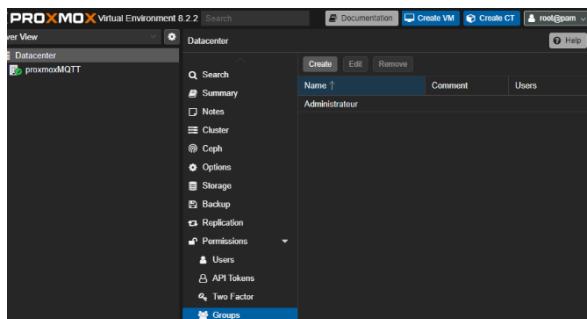
Paramètre proxy

Une fois cela fait, on peut atteindre l'interface de Proxmox en se connectant avec le compte root et le mot de passe mis en place lors de la configuration.

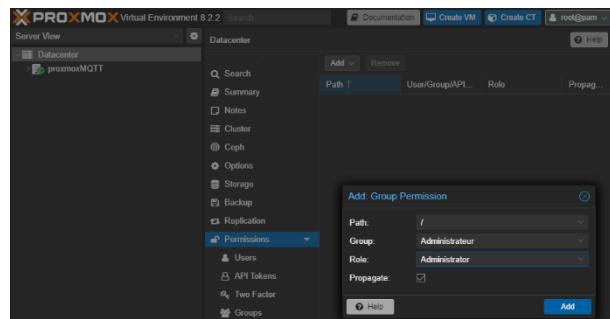


Connexion avec le compte root

Après cela, il est nécessaire de créer des groupes, de définir leurs permissions et d'y ajouter les utilisateurs avec leurs propres mots de passe. Ensuite, les groupes doivent être associés aux permissions correspondantes avant d'y intégrer les utilisateurs pour leur attribuer les droits d'accès appropriés.



Création de groupes



Ajout de permission au groupe

User name:	anthony@pam	First Name:	
Group:	Administrateur	Last Name:	
Expire:	2026-07-31	E-Mail:	anthony.ketterer@grandest.
Enabled:	<input checked="" type="checkbox"/>		
Comment:	Créateur du projet		

Advanced OK

Création d'utilisateurs

Mon compte expirera le 31 juillet 2026. Je ne serais pas indéfiniment présent dans l'entreprise et cette date correspond approximativement à la fin de mon contrat de licence informatique car il est prévu que la Région GRANDEST me renouvelle mon contrat.

Une fois le serveur Proxmox correctement installé et configuré pour le lancement de machines virtuelles, un cahier de recettes servant de documentation a été rédigé. Il regroupe chaque étape de la mise en place de Proxmox.

8. La configuration de la machine virtuel (VM)

Pour configurer une VM, il faut d'abord définir un stockage approprié, sélectionner le système d'exploitation et allouer les ressources nécessaires (**ISO** : International Organization for Standardization, **CPU** : Central Processing Unit, **RAM** : Random Access Memory, disque...). Proxmox permet d'ajouter des périphériques, de paramétriser le réseau en mode bridge ou NAT et d'activer les sauvegardes automatiques. L'authentification et la gestion des accès garantissent une administration sécurisée.

L'ISO mis en place sur cette VM est une Debian que je possépais déjà dans mon ordinateur lors des travaux pratiques de 1^{ère} année, donc inutile d'en télécharger une depuis internet.

	Memory	8.00 GiB
	Processors	2 (1 sockets, 2 cores) [vcpus=2]
	BIOS	Default (SeaBIOS)
	Display	Default
	Machine	Default (i440fx)
	SCSI Controller	VirtIO SCSI single
	CD/DVD Drive (ide2)	local:iso/debian-12.7.0-amd64-netinst.iso,media=cdrom,size=631M
	Hard Disk (scsi0)	local-lvm:vm-100-disk-0,iothread=1,size=82G
	Network Device (net0)	virtio=E\24:11:1B:1C:B3,bridge=vmbr0,firewall=1
	Unused Disk 0	local-lvm:vm-100-disk-1

Configuration de la VM

L'image ISO **debian-12.7-amd64-netinst.iso** (631 Mo) est montée en tant que lecteur CD/DVD depuis le stockage local pour l'installation du système. Le disque principal (vm-100-disk-0) est un stockage SCSI de 82 Go sur **local-lvm**, avec l'option **io_thread=1** pour optimiser les performances d'E/S. La VM dispose d'une interface réseau VirtIO, associée au bridge vmbr0, avec l'adresse MAC et un pare-feu activé. Enfin, un disque non utilisé (**vm-100-disk-1**) est disponible sur **local-lvm** mais n'est pas encore attaché à la VM.

Une fois connecté sur la VM, il faut faire quelques configurations essentielles. Renseigner l'adresse ip, le masque de sous-réseau, le DNS, ainsi que la passerelle. Tout cela se trouve dans le fichier suivant : /etc/network/interfaces

```
GNU nano 7.2                                     /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

#ens18
auto ens18
iface ens18 inet static
    address 10.129.251.253
    netmask 255.255.0.0
    gateway 10.129.254.254
    dns-nameservers 10.129.253.211 10.129.253.212
```

Configuration réseau

On peut également renseigner le domaine et les serveurs DNS (Domain Name System) dans ce fichier : /etc/resolv.conf

```
GNU nano 7.2                                     /etc/resolv.conf
# Generated by NetworkManager
search lycee.couffignal.dom lan
nameserver 10.129.253.211
nameserver 10.129.253.212
```

Configuration domaine

Vérification de l'obtention du l'adresse ip une fois 'network.service' redémarré :

```
root@SRV-MQTT:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether b2:24:11:1b:1c:b3 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.129.251.253/16 brd 10.129.255.255 scope global ens18
        valid_lft forever preferred_lft forever
        inet6 fe80::be24:11ff:fe1b:1cb3/64 scope link
            valid_lft forever preferred_lft forever
```

Une fois cette configuration effectuée et un test ICMP (Internet Control Message Protocol) de tous les appareils nécessaires au bon fonctionnement du projet (capteur, Proxmox, VM, passerelle et serveur DNS), le service Mosquitto peut être mis en place.

9. Le protocole MQTT (Message Queuing Telemetry Transport)

a. Explication du protocole

MQTT est un protocole créé en 1999 par Andy Stanford-Clark travaillant chez IBM et Arlen Nipper d'Europet qui permet à des appareils de communiquer entre eux en envoyant et recevant des messages via un serveur central (broker). C'est un dispositif extrêmement léger, il est très utilisé pour les objets connectés (IoT), tels que les capteurs ou les systèmes de maison intelligente (domotique). Au lieu qu'un appareil envoie un message directement à un autre, il le publie sur un topic (thème), et tous les appareils abonnés à ce topic recevront le message.

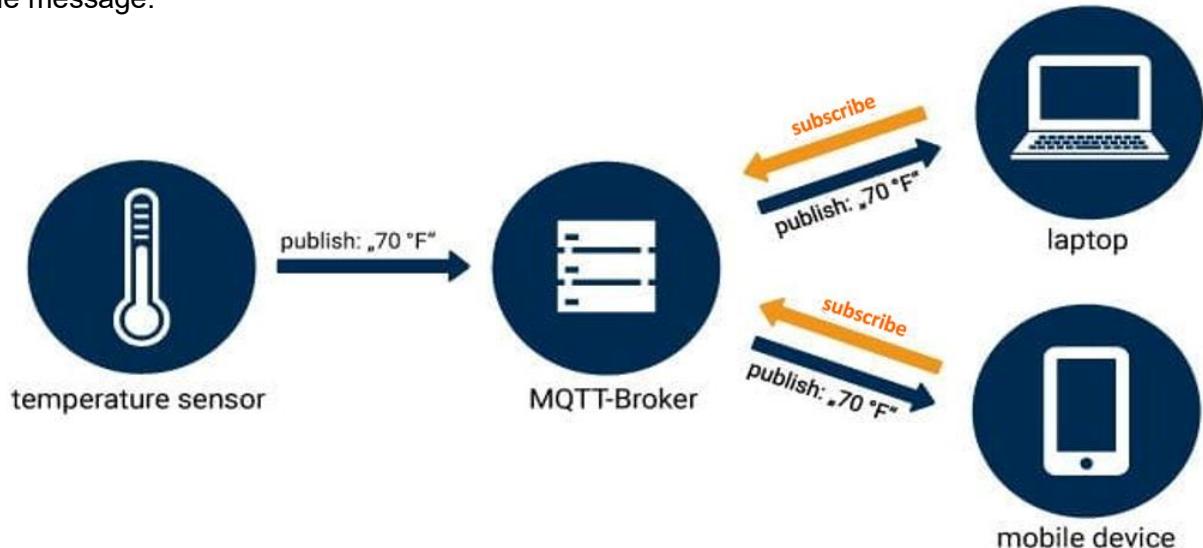


Schéma de fonctionnement MQTT

MQTT est souvent préféré à d'autres protocoles de communication, comme HTTP, car il est optimisé pour les réseaux instables ou à faible bande passante. Contrairement à HTTP, qui fonctionne en mode requête/réponse, MQTT utilise un système publish/subscribe, permettant aux appareils de recevoir automatiquement les mises à jour sans les interroger en permanence. Il consomme aussi moins d'énergie et de données, ce qui est essentiel pour les objets connectés fonctionnant sur batterie. De plus, il offre des niveaux de qualité de service (QoS) pour garantir la transmission des messages, même en cas de coupure de connexion.

- **QoS 0 ("Au plus une fois")** : Le message est envoyé sans confirmation de réception. Il peut être perdu si la connexion est interrompue. C'est le mode le plus rapide et léger.
- **QoS 1 ("Au moins une fois")** : Le message est garanti d'être livré au moins une fois, mais il peut être reçu plusieurs fois en cas de retransmission.
- **QoS 2 ("Exactement une fois")** : Le message est assuré d'être livré une seule fois grâce à un échange de plusieurs confirmations, mais cela augmente le trafic réseau.

b. Installation et configuration du protocole MQTT

Initialement, avant de procéder à l'installation des paquets logiciels mosquitto et mosquitto-clients, il est impératif de mettre à jour la VM via la commande 'apt update'. Après avoir effectué cette étape, l'installation des paquets peut être réalisée en utilisant la commande 'apt install mosquitto mosquitto-clients -y'. La commande renseignée signifie que les modules vont s'installer en prenant par défaut la valeur "yes" si des choix sont proposés.

Ensuite une simple configuration est nécessaire.

```
GNU nano 7.2                                     /etc/mosquitto/mosquitto.conf *
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

# Default MQTT Listener
listener 1883
allow_anonymous true                                Configuration mosquito
```

- **pid_file /run/mosquitto/mosquitto.pid**
Définit le fichier PID (Process ID) pour suivre le processus Mosquitto.
- **persistence true**
- **persistence_location /var/lib/mosquitto/**
Active la persistance des messages MQTT et sauvegarde les données MQTT dans /var/lib/mosquitto/ pour une récupération après redémarrage.
- **log_dest file /var/log/mosquitto/mosquitto.log**
Enregistre les logs dans /var/log/mosquitto/mosquitto.log.
- **include_dir /etc/mosquitto/conf.d**
Charge les fichiers de configuration supplémentaires présents dans /etc/mosquitto/conf.d/.
- **listener 1883**
Écoute sur le port 1883 (communication non sécurisée).
- **allow_anonymous true**
Autorise les connexions anonymes.

c. Tests du protocole MQTT

Maintenant que le service mosquitto est installé, correctement configuré et que le serveur et le capteur peuvent communiquer ensemble, il nous faut d'abord tester le service en envoyant sur un topic ('temp') un message. Ce test ne prend pas encore en compte les données envoyées par le capteur de température.

```
root@SRV-MQTT:~# mosquitto_pub -h 10.129.251.253 -t temp -m "résultat obtenu"
root@SRV-MQTT:~#
```

```
root@SRV-MQTT:~# mosquitto_sub -h 10.129.251.253 -t temp
résultat obtenu
```

Commande mosquitto_pub & mosquitto_sub

- **mosquitto_pub**

C'est l'outil en ligne de commande fourni avec Mosquitto pour publier un message MQTT sur un broker (serveur MQTT).

- **mosquitto_sub**

C'est l'outil en ligne de commande fourni avec Mosquitto pour s'abonner à un topic et recevoir les messages.

Il est complémentaire de mosquitto_pub.

- **-h 10.129.251.253**

C'est l'adresse IP ou le nom du serveur où se trouve le broker Mosquitto.

- **-t temp**

C'est le sujet sur lequel tu publies le message.

- **-m "résultat obtenu"**

C'est le contenu envoyer sur le topic spécifié. Il est également possible d'envoyer aucun message lors de la publication (-m " ").

```
^Cuser_mqtt@SRV-MQTT:~$ mosquitto_pub -h localhost -t temp -m "Température actuelle de la salle serveur du lycée Louis COUFFIGNAL" -r
user_mqtt@SRV-MQTT:~$ mosquitto_sub -h localhost -t temp
Température actuelle de la salle serveur du lycée Louis COUFFIGNAL
18.20
18.20
18.20
18.20
```

Détection de la température du capteur

Cette fois, le capteur transmet les données collectées au serveur MQTT sur le sujet 'temp'. Comme le serveur détient les données, il s'abonne auprès de lui-même pour pouvoir les afficher dans le terminal.

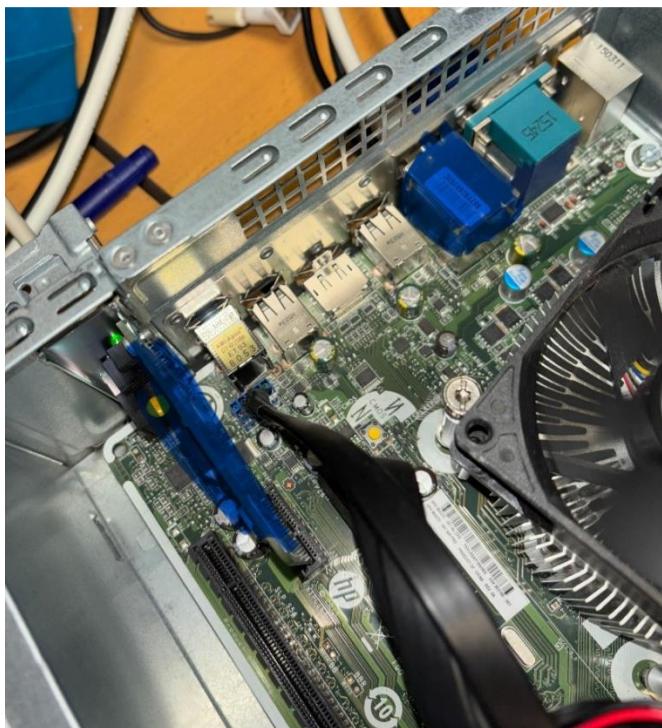
10. Mise en place du VPN de la région GrandEst

L'installation d'un VPN dans une entreprise assure une protection pour les liaisons réseau entre l'utilisateur et les ressources internes de la société. Ceci établit un canal sécurisé qui défend les informations échangées des interceptions ou des cyberattaques. De plus, le VPN offre aussi la possibilité d'accéder aux services internes (serveurs, bases de données, fichiers partagés) comme si on était réellement sur place dans les locaux de l'entreprise.

Mise en place d'un poste servant à la connexion VPN :

Pour cette mise en place, il faut dans un premier temps choisir le matériel sur lequel nous voulons, par la suite, nous connecter à distance. Dans ce cas, le matériel choisi est un poste de la même référence que le serveur MQTT (HP ProDesk 400 G2.5).

Il est nécessaire d'installer une deuxième carte réseau dans ce dernier afin qu'une soit utilisée pour la connexion à distance et l'autre pour accéder au réseau pédagogique (local) du lycée.



La carte réseau connectée à la carte mère sera utilisée pour le réseau local, tandis que la seconde sera ajoutée pour la connexion VPN.

Cartes réseaux du poste VPN

Après l'installation des deux cartes réseau, la configuration de leurs adresses IP doit être effectuée via : Panneau de configuration > L'ensemble des Panneaux de configuration > Connexions réseau.

Adresse IP de la carte réseau locale est : 10.129.251.2/16

Adresse IP du réseau VPN : 10.113.172.X/24

⚠ Pour des raisons de sécurité, la partie hôte du réseau VPN ne sera pas révélée.

a. Configuration des interfaces du switch VPN

Cette configuration permet un passage des trames sur le VLAN par défaut (VLAN 1, non affiché mais bien configuré sur les interfaces). Quant au BPDU Protection (Bridge Protocol Data Unit), il s'agit d'un mécanisme de sécurité intégré au Spanning Tree Protocol (STP) qui empêche l'injection non autorisée de BPDUs sur des ports où ils ne sont pas censés être reçus.

Les BPDUs sont des messages échangés entre switches via le protocole STP, permettant de construire une topologie logique sans boucle en identifiant les chemins redondants et en désactivant automatiquement les ports inutiles, assurant ainsi la stabilité et la résilience du réseau.

Il existe principalement deux types de protection :

```
#  
interface GigabitEthernet1/0/10  
  stp edged-port  
  stp port bpdu-protection enable  
#
```

Interface locale

```
#  
interface GigabitEthernet1/0/2  
  port access vlan 511  
  stp edged-port  
  stp port bpdu-protection enable  

```

Interface VPN

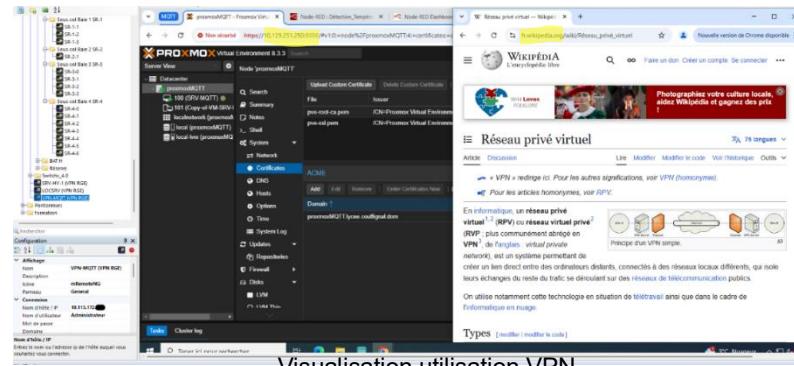
- **BPDU Guard**, qui met un port en état err-disabled s'il reçoit un BPDU alors qu'il est configuré en PortFast*
- **BPDU Filter**, qui bloque l'envoi et la réception des BPDUs sur un port spécifique.

Ces fonctionnalités permettent de prévenir les erreurs de configuration et les attaques pouvant compromettre la stabilité du réseau en modifiant la topologie STP de manière indésirable.

*PortFast : STP impose un délai (30 à 50 secondes) avant qu'un port ne devienne opérationnel, ce qui peut poser problème pour des équipements nécessitant un accès réseau rapide (exemple : DHCP). PortFast réduit ce délai, améliorant ainsi le temps de connexion des périphériques.

b. Test de la mise en place du VPN

Pour vérifier cette configuration, il faut de se connecter avec mRemote (ou autres logiciels utilisant RDP (Remote Desktop Protocol)). Comme montré sur l'image, la connexion VPN a été établie et l'ordinateur a accès au réseau local (le Proxmox en 10.129.251.250:8006) et à internet (fr.wikipedia.org)



Visualisation utilisation VPN

11. Node-Red

Node-RED est un outil de développement open source, conçu pour faciliter la programmation d'applications en connectant des périphériques matériels, des APIs et des services en ligne à l'aide d'une interface graphique intuitive. Basé sur Node.js, il permet de créer des flux de données sous forme de diagrammes où chaque nœud représente une fonction ou une action spécifique. Node-RED est particulièrement adapté aux projets liés à l'IoT, à l'automatisation ou à l'intégration de systèmes hétérogènes. Il est apprécié pour sa prise en main rapide, sa flexibilité et sa capacité à s'intégrer facilement dans des environnements industriels ou domotiques.

L'installation de Node-Red se fait via le terminal du serveur 'sudo npm install -g --unsafe-perm node-red'. Après son installation il faut, pour des raisons de sécurité, mettre en place au moins un identifiant et un mot de passe afin de se connecter à l'interface de Node-Red (<http://localhost:1880>). Pour ce faire, il faut modifier 'adminAuth' présent dans le fichier '/home/nom_user/.node-red/settings.js' comme ceci :

```
GNU nano 7.2                                     /home/user_mqtt/.node-red/settings.js
* property can be used. See https://nodered.org/docs/security.html for details.
*/
adminAuth: {
  type: "credentials",
  users: [{username: "Administrateur",
    password: "$2y$08$dTFI3CMZxwtAebIWyOKCrO.0TVyzvszcEkmJubV009df4E.uIC3Xu",
    permissions: "*"
  }]
},
```

Identifiant et mot de passe Node-Red

- **Type : "credentials"**,

Cela indique que l'on utilise une authentification basée sur un nom d'utilisateur et un mot de passe haché

- **username: "Administrateur"**,

Spécifie le nom d'utilisateur à entrer pour se connecter.

- **password: "\$2y\$08\$dTFI3CMZw...uIC3Xu"**,

Mot de passe chiffré (hashé) avec bcrypt. Ce n'est pas le mot de passe en clair mais une version sécurisée. Node-RED compare ce hash avec celui généré à partir du mot de passe saisi lors de la connexion.

- **permissions: "*"**

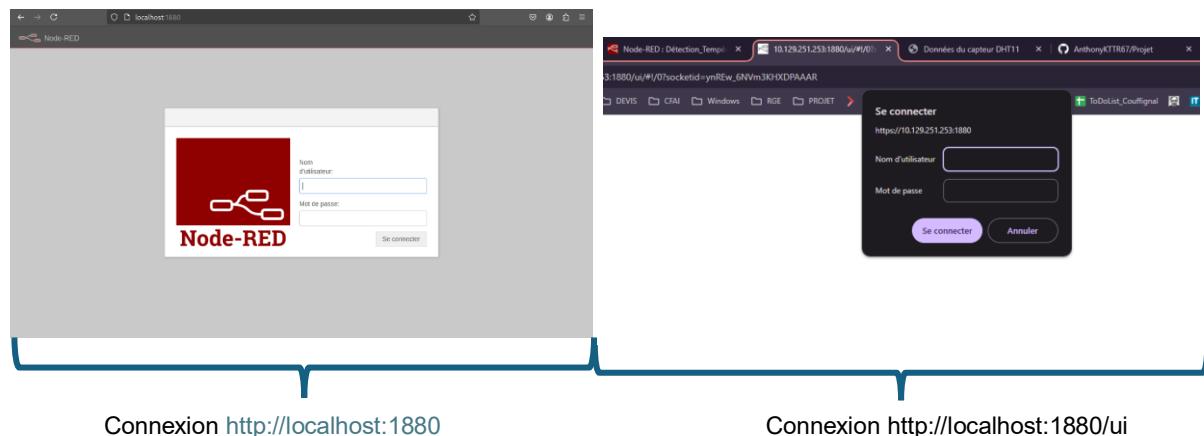
Attribue tous les droits à cet utilisateur : accès à l'interface, aux paramètres, au déploiement, etc.

Il faut également mettre un autre identifiant et mot de passe pour visualiser les données sur l'interface du dashboard (<http://localhost:1880/ui>) afin que l'utilisateur qui accède au dashboard ne puisse pas modifier les nœuds.

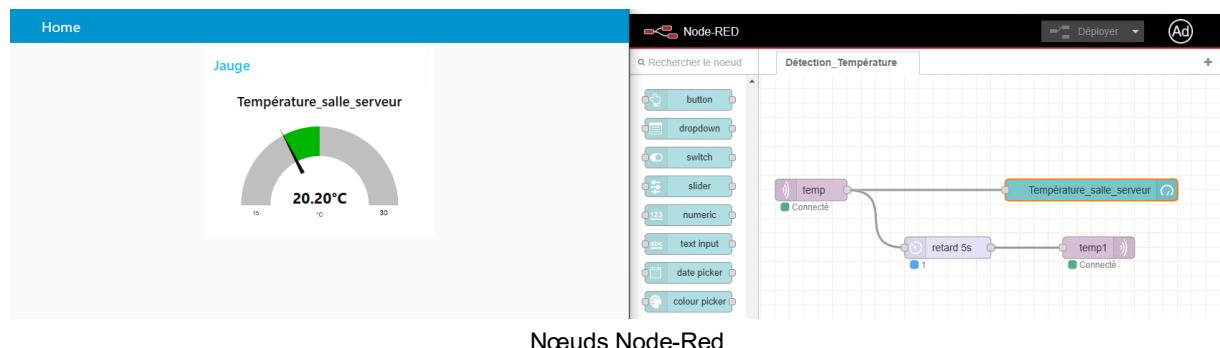
```
httpNodeAuth: {user: "Administrateur", pass: "$2a$12$bH9KN9HCCi.c@JwPQX3nH.vtYOR0SSPSCpdJAKVohc8a01xm82v26"},  
//httpStaticAuth: {user: "user", pass: "$2a$08$z2WtXTja0f81pzD4shMyOCNYz2Z6dNbM6tl8sJogEN0McxWV9DN."},
```

Identifiant et mot de passe dashboard Node-Red

Grâce à cette configuration, la connexion sécurisée via mot de passe est désormais active après le redémarrage de Node-Red ('sudo node-red').



Pour visualiser par node-red les données de température récoltées par le capteur, il est indispensable de configurer les nœuds de manière adéquate.



Le nœud mqtt in nommé 'temp' récupère les données. Ces données sont ensuite envoyées directement vers une jauge graphique (image de gauche) nommée Température_salle_serveur. En parallèle, un second chemin ajoute un délai de 5 secondes grâce au nœud delay, avant de republier les données via un second nœud mqtt out. Cette architecture simple et modulaire permet à la fois la visualisation directe, le traitement différé, et la rediffusion des données selon les besoins. L'architecture sera légèrement modifiée par la suite lors de l'ajout du service SMTP.

a. Certificats OpenSSL Node-Red

Des certificats SSL/TLS ont été configurés en localhost:1880 afin de sécuriser la connexion à Node-Red. Dans le cadre de ce projet d'étude, j'ai choisi d'utiliser OpenSSL plutôt que Let's Encrypt afin de maîtriser l'ensemble du processus de création et de gestion de certificats TLS. OpenSSL permet de générer manuellement des certificats auto-signés ou signés par une autorité interne, ce qui est particulièrement adapté à mon environnement local.

Dans un premier temps, afin d'organiser notre serveur, la création d'un dossier (/home/user_mqtt/.node-red/certs) est préférable pour y enregistrer les futures clés de certificats. Il faut par la suite générer une clé (nred-key.key) de 2048 bits dans ce dossier.

```
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# openssl genrsa -out nred-key.crt 2048
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# ls
nred-key.crt
```

Génération d'une clé privée

Cette commande OpenSSL sert à générer une CSR (Certificate Signing Request), c'est-à-dire une demande de signature de certificat, en utilisant la clé privée déjà existante.

```
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# openssl req -new -sha256 -key nred-key.crt -out nred-csr.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:GRANDEST
Locality Name (eg, city) []:STRASBOURG
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NODERED
Organizational Unit Name (eg, section) []:NODERED
Common Name (e.g. server FQDN or YOUR name) []:127.0.0.1
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# ls
nred-csr.csr  nred-key.crt
```

Génération d'une clé CSR

Cette commande sert à auto-signer la demande de certificat avec la clé privée, ce qui crée un certificat auto-signé.

```
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# openssl x509 -req -in nred-csr.csr -signkey nred-key.crt -out nred-cert.crt -days 1095
Certificate request self-signature ok
subject=C = FR, ST = GRANDEST, L = STRASBOURG, O = NODERED, OU = NODERED, CN = 127.0.0.1
root@SRV-MQTT:/home/user_mqtt/.node-red/certs# ls
nred-cert.crt  nred-csr.csr  nred-key.crt
```

Signature du certificat

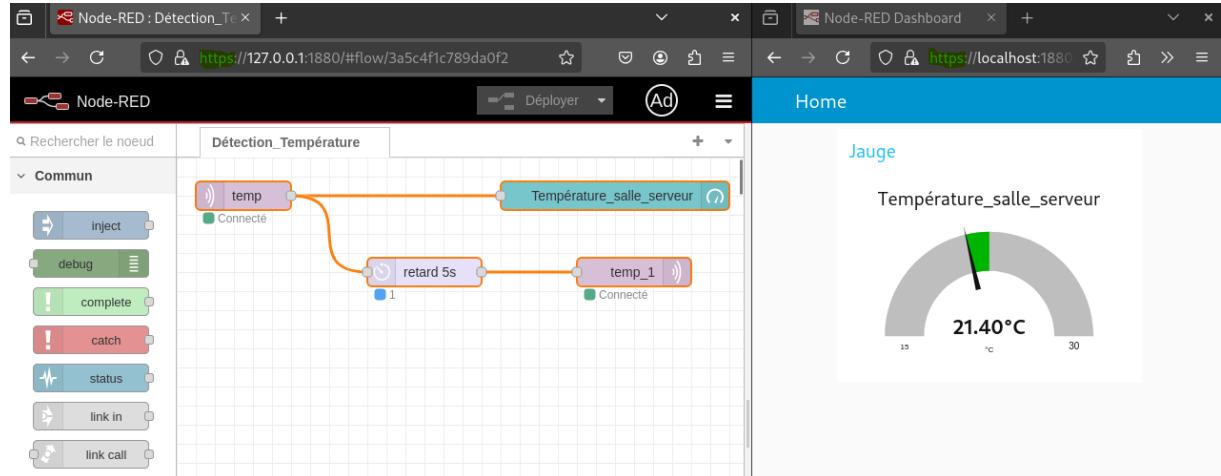
Une fois les certificats créés et la CA (autorité de certification) auto-signé, il suffit d'ajouter le chemin de ces certificats dans le module 'https' du même fichier que lors de l'ajout de mot de passe (/home/user_mqtt/.node-red/settings.js)

```
GNU nano 7.2                               /home/user_mqtt/.node-red/settings.js
*/
/** Option 1: static object */
https: {
  key: require("fs").readFileSync('/home/user_mqtt/.node-red/certs/nred-key.crt'),
  cert: require("fs").readFileSync('/home/user_mqtt/.node-red/certs/nred-cert.crt')
},
```

Ajout chemin SSL

b. Test des certificats Node-Red

Après le redémarrage de Node-Red via le fichier source (node-red -s /home/user_mqtt/.node-red/settings.js), lors de la connexion à son interface web, on peut constater que Node-Red a bien ses certificats actif.



Connexion sécurisée à Node-Red

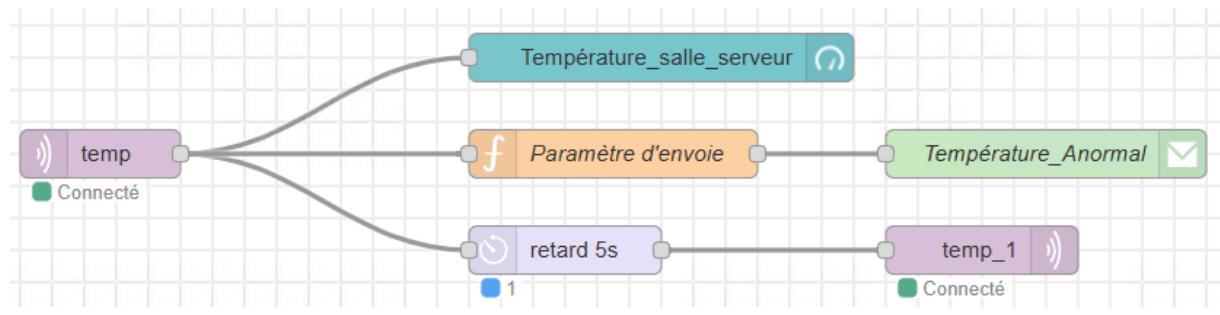
Une fois les certificats correctement installé et configuré pour le lancement sécurisé de Node-Red, un cahier de recettes servant de documentation a été rédigé. Il regroupe chaque étape de la mise en place de ces derniers.

12. Simple Mail Transfer Protocol

Le protocole SMTP (Simple Mail Transfer Protocol) est un protocole standard utilisé pour l'envoi de courriels électroniques sur Internet. Il fonctionne principalement en mode push, c'est-à-dire qu'il permet à un client de transférer des e-mails vers un serveur de messagerie ou entre serveurs. Fonctionnant sur le port 25 par défaut (ou 587/465 en version sécurisée), SMTP joue un rôle essentiel dans la transmission des messages électroniques, que ce soit pour un usage personnel ou professionnel.

Mise en place du service SMTP via Node-Red

Pour procéder à l'installation du service SMTP, il suffit de se connecter à l'interface de Node-Red puis de cliquer sur les trois barres horizontales en haut à droite et de se rendre dans 'gérer la palette'. Dans le module 'installer', écrivez 'node-red-node-email' pour démarrer l'installation.



Nœud SMTP rajouté

Le nœud SMTP 'Température_Anomal' ainsi que une fonction nommée 'Paramètre d'envoie' a été introduit dans Node-Red. Il faut désormais les configurer correctement pour que les mails s'envoient uniquement lorsque la température n'est pas entre 17°C et 24°C (<17°C ou >24°C).

Configuration du nœud 'Température Anormal' :

Propriétés SMTP

Nom : Nom donné au nœud,

To : Adresse e-mail du destinataire.

Server : Adresse du serveur SMTP utilisé, ici celui de Microsoft Office 365.

Port : Port standard utilisé pour le SMTP avec **STARTTLS** (sécurisation de la connexion après l'établissement initial).

Auth type : Type d'authentification utilisée, ici simple login/mot de passe.

Userid : Identifiant utilisé pour s'authentifier sur le serveur SMTP.

Password : Mot de passe associé au compte e-mail.

TLS option : Active la vérification du certificat TLS du serveur pour assurer une communication sécurisée.

Ce script Javascript, placé dans le nœud de fonction, permet d'envoyer une alerte par e-mail si la température sort d'une plage normale (< à 17°C ou > à 24°C). Il commence par récupérer l'état de la dernière alerte envoyée à l'aide d'une variable de flux nommée lastAlert, puis vérifie si la température est anormale. Si c'est le cas et qu'aucune alerte n'a encore été envoyée, il prépare un message e-mail avec un objet (topic) et un texte d'alerte (payload), destinés à l'adresse de l'administrateur. L'e-mail n'est envoyé qu'une seule fois tant que la température reste anormale. Lorsque la température revient à un niveau normal, l'état d'alerte est réinitialisé, permettant d'envoyer une nouvelle alerte si besoin. Cela évite l'envoi répété d'e-mails pour une même anomalie.

programmation de la fonction d'envoi de mail

Une fois le service SMTP correctement installé et configuré pour l'envoie de mail lors d'anomalie via Node-Red, un cahier de recettes servant de documentation a été rédigé. Il regroupe chaque étape de la mise en place de ce service.

13. Base de données MariaDB & MySQL

Qu'est-ce qu'une base de données et à quoi ça sert ?

Une base de données est un système permettant de stocker, organiser et gérer facilement et simplement des informations (noms, adresses e-mail, données de températures etc). Elle est conçue pour faciliter l'accès, la manipulation et la mise à jour des données de manière rapide et fiable. Dans le cadre d'une application, d'un site web ou d'un système automatisé, la base de données joue un rôle central en assurant la persistance des données. Mettre en place une base de données est essentiel dès lors que l'on souhaite conserver des données de façon organisée et sécurisée, tout en garantissant leur intégrité sur le long terme.

Présentation et utilisation de MariaDB :

MariaDB est un système de gestion de bases de données relationnelles open source, dérivé direct du célèbre MySQL. Il emploie le langage SQL pour faciliter l'établissement, la gestion et l'interaction avec des données structurées en tables. MariaDB, un outil performant et fiable, est compatible avec une multitude de systèmes tels que PHP, Apache2 ou PHPMyAdmin, ce qui facilite son intégration dans des structures web traditionnelles. Il est spécialement conçu pour les projets qui demandent un stockage structuré et sécurisé des données, que ce soient des logs d'utilisateurs, des mesures provenant de capteurs ou des informations liées aux applications. Grâce à sa large communauté, son développement actif et sa licence libre, MariaDB offre une solution flexible et évolutive, aussi bien pour les environnements de test que pour la production. Son utilisation facilite également les échanges entre différentes applications, via des requêtes SQL rapides et efficaces.

Mise en place de la base de données

Pour procéder à l'installation de la base de données il suffit, une fois entrée dans le terminal en mode sudo (root), de démarrer l'installation de MariaDB via la commande bash ‘apt install mariadb-server -y’. Cette commande permet d'installer MariaDB tout en présélectionnant ‘yes’ lorsque des choix y sont proposés.

```
root@SRV-MQTT:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

Installation mysql

Une fois MariaDB installer sur le serveur, il faut installer et configurer MySQL. MySQL permet ainsi sur MariaDB de créer, lire, modifier, supprimer des données et plus encore.

MySQL étant correctement configuré, la création d'une base de données et d'une table va pouvoir être mis en place.

Pour créer une base de données qui permettra par la suite de stocker les valeurs du capteur de température, il faut premièrement entrer dans MariaDB.

```
root@SRV-MQTT:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE capteurs_temp;
Query OK, 1 row affected (0,000 sec)
```

Création base de données

La commande SQL 'CREATE DATABASE capteurs_temp;' permet tout simplement de créer une base de données dont le nom est « capteurs_temp ».

Lorsque l'utilisateur est rentré dans la base de données, il peut créer une table. C'est dans cette table que les données récoltés seront sauvegardées.

```
MariaDB [capteurs_temp]> CREATE TABLE mesures_temperatures (      id INT AUTO_INCREMENT PRIMARY KEY,      timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,      temperature FLOAT NOT NULL );
Query OK, 0 rows affected (0,009 sec)
```

Création table dans DATABASE

- 1. id INT AUTO_INCREMENT PRIMARY KEY,**
 - Crée une colonne id de type entier (INT), qui sert à identifier de manière unique chaque enregistrement dans la table.
 - AUTO_INCREMENT signifie que la valeur de cette colonne sera automatiquement incrémentée à chaque nouvel enregistrement, sans que vous ayez à spécifier manuellement l'ID.
 - PRIMARY KEY indique que cette colonne sera la clé primaire, garantissant qu'aucune valeur en double ne pourra être insérée dans cette colonne.
- 2. timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,**
 - Crée une colonne timestamp de type TIMESTAMP pour enregistrer la date et l'heure à laquelle chaque donnée a été insérée.
 - DEFAULT CURRENT_TIMESTAMP signifie que, si vous ne spécifiez pas de valeur pour cette colonne lors de l'insertion, la date et l'heure actuelles seront automatiquement utilisées.
- 3. valeur FLOAT NOT NULL**
 - Crée une colonne valeur de type FLOAT, qui sera utilisée pour stocker la température mesurée par votre capteur.
 - NOT NULL garantit que cette colonne ne peut pas être vide. Chaque enregistrement doit obligatoirement avoir une valeur de température.

a. Script Python

Après avoir mis en place la table dans la DATABASE, il est maintenant nécessaire de trouver un moyen d'insérer les données dans celle-ci. La décision prise a donc été de les insérer via un script Python. Pour ce faire, des bibliothèques Python (paho-mqtt et mysql-connector-Python) ont été installés.

- **Paho-mqtt** (version 2.1.0) est une bibliothèque Python qui permet de se connecter à un broker MQTT, de publier et recevoir des messages.
- **Mysql-connector-python** (version 9.2.0) est une bibliothèque Python qui permet de se connecter à une base de données MySQL/MariaDB, d'exécuter des requêtes SQL, et de gérer les données depuis un script Python.

```
root@SRV-MQTT:~# python3 -m pip install --break-system-packages paho-mqtt mysql-connector-python
Collecting paho-mqtt
  Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
    67.2/67.2 kB 1.4 MB/s eta 0:00:00
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.2.0-cp311-cp311-manylinux_2_28_x86_64.whl (34.0 MB)
    34.0/34.0 kB 19.7 MB/s eta 0:00:00
Installing collected packages: paho-mqtt, mysql-connector-python
Successfully installed mysql-connector-python-9.2.0 paho-mqtt-2.1.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@SRV-MQTT:~# exit
déconnexion
user_mqtt@SRV-MQTT: $
```

Téléchargement bibliothèques Python

Ce script Python est disponible dans le dossier GitHub.

Il permet d'écouter en continu les messages publiés sur un topic MQTT nommé "temp" via le broker Mosquitto local, puis d'enregistrer automatiquement chaque valeur reçue dans la table mesures_temperatures. Lorsqu'un message est reçu, sa température est décodé et vérifié : s'il s'agit bien d'un nombre float (température valide), le script se connecte à la base de données capteurs_temp et insère la valeur dans la table

```
# ...
# Fonction pour vérifier si une chaîne peut être convertie en float
# ...
def is_float(value):
    try:
        float(value)
        return True
    except ValueError:
        return False

# ...
# Callback appelé lorsqu'un message MQTT est reçu
# ...
def on_message(client, userdata, msg):
    try:
        payload = msg.payload.decode("utf-8") # Décodage du message reçu
        data = payload.split(",") # Séparation des valeurs par virgules (si plusieurs valeurs sont envoyées)

        # Vérifie que la première valeur est un nombre flottant
        if len(data) >= 1 and is_float(data[0]):
            temperature = float(data[0]) # Convertit la valeur en float

            # Connexion à la base de données et insertion de la température
            conn = connect_db()
            cursor = conn.cursor()
            cursor.execute("INSERT INTO mesures_temperatures (temperature) VALUES (%s)", (temperature,))
            conn.commit()

            # Fermeture de la connexion
            cursor.close()
            conn.close()
            print(f"Donnée insérée : {temperature}°C")
    except Exception as e:
        print(f"Erreur lors de l'insertion : {e}")
```

mesures_temperatures. Chaque message reçu déclenche cette action. Le script fonctionne en boucle grâce à mqtt_client.loop_forever(), ce qui lui permet de rester actif en arrière-plan, prêt à traiter les nouvelles données publiées sur le topic.

Script de collecte de température dans la base de données

Lorsque le script Python est sauvegardé dans un dossier spécialisé pour une meilleure organisation, il suffit de le lancer pour le mettre en fonction.

```
root@SRV-MQTT:~# python3 /opt/mqtt_to_mariadb/mqtt_to_mariadb.py
/opt/mqtt_to_mariadb/mqtt_to_mariadb.py:64: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  mqtt_client = mqtt.Client()
Donnée insérée : 20.2°C
Donnée insérée : 20.2°C
Donnée insérée : 20.2°C
Donnée insérée : 20.2°C
```

Lancement script Python

Quand le script 'mqtt_to_mariadb.py' est exécuté dans un terminal local, il affiche les données enregistrées dans la base au moment de la détection.

Dans le but de visualiser les données dans la base, il faut se connecter à MariaDB avec l'utilisateur user-db. La première commande signifie qu'on se connecte à mysql avec l'utilisateur user-db en localhost et que lors de la validation de cette commande il nous faudra entrer le mot de passe de user-db (-p).

```
user_mqtt@SRV-MQTT: $ mysql -u user-db -p -h localhost
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 377182
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE capteurs_temp;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [capteurs_temp]> SELECT * FROM mesures_temperatures
    -> LIMIT 20;
+-----+-----+
| id | timestamp           | temperature |
+-----+-----+
| 1  | 2025-03-31 13:18:09 | 19.6        |
| 2  | 2025-03-31 13:18:15 | 19.8        |
| 3  | 2025-03-31 13:18:20 | 19.8        |
| 4  | 2025-03-31 13:18:25 | 19.8        |
| 5  | 2025-03-31 13:18:30 | 19.8        |
| 6  | 2025-03-31 13:18:35 | 19.8        |
| 7  | 2025-03-31 13:18:40 | 19.8        |
| 8  | 2025-03-31 13:18:45 | 19.8        |
| 9  | 2025-03-31 13:18:50 | 19.8        |
| 10 | 2025-03-31 13:18:55 | 19.8        |
| 11 | 2025-03-31 13:19:00 | 19.8        |
```

Visualisation des valeurs insérée dans la base

Lorsqu'on entre dans la base et qu'on sélectionne les données de la table 'SELECT * FROM mesures_temperatures', elle s'affiche dans un tableau comme présent sur l'image.

Une fois la base de données correctement installé et configuré pour le stockage des données de température et leur visualisation, un cahier de recettes servant de documentation a été rédigé. Il regroupe chaque étape de la mise en place de MariaDB.

14. Installation et configuration de PHPMyAdmin

PHPMyAdmin est une application web écrite en PHP qui permet de gérer facilement des bases de données MySQL ou MariaDB depuis une interface graphique accessible via un navigateur (http ou https). Au lieu d'écrire manuellement des commandes SQL en ligne de commande, phpMyAdmin offre des formulaires et des menus pour exécuter des actions comme créer des bases de données, ajouter des tables, modifier des enregistrements, sauvegarder ou importer des données. Il est particulièrement utile pour les administrateurs de bases de données débutants ou pour gérer rapidement des serveurs à distance, tout en restant puissant pour effectuer des opérations complexes.

En premier temps, il faut installer PHPMyAdmin via la commande 'apt install phpmyadmin' en sudo. Un choix sera proposé s'il on veut installer PHPMyAdmin via apache ou Lighttpd. Voici les différents avantages et inconvénients des deux :

Apache

✓ Avantages :

- Très compatible avec phpMyAdmin et la majorité des applications web.
- Large communauté et beaucoup de documentation.
- Modules puissants pour la gestion avancée (authentification, SSL, réécriture d'URL, etc.).

✗ Inconvénients :

- Consomme plus de RAM que Lighttpd, surtout sous forte charge.
- Moins rapide pour servir de nombreux petits fichiers statiques.
- Plus lourd à configurer pour des sites très optimisés.

Lighttpd

✓ Avantages :

- Très léger et rapide, parfait pour des serveurs à ressources limitées.
- Optimisé pour gérer de nombreuses connexions simultanées.
- Facile à installer et à configurer pour un usage simple.

✗ Inconvénients :

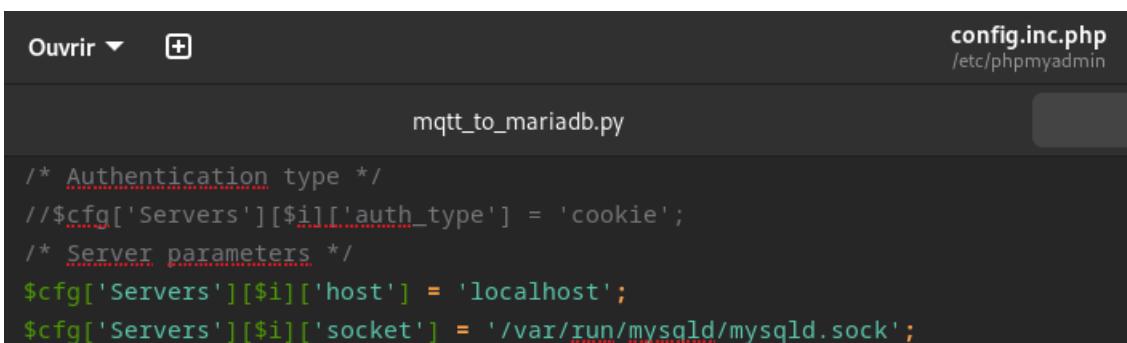
- Moins compatible avec certaines applications complexes prévues pour Apache.
- Moins de modules disponibles comparé à Apache.
- Moins de documentation et de support communautaire.

```
root@SRV-MQTT:~# ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
root@SRV-MQTT:~# a2enconf phpmyadmin
Enabling conf phpmyadmin.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@SRV-MQTT:~# systemctl restart apache2.service
```

Lien symbolique

La commande **ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf** permet de créer un lien symbolique entre le fichier de configuration de phpMyAdmin et le répertoire des configurations disponibles d'Apache. Cette étape est nécessaire pour que le serveur web Apache puisse reconnaître et activer la configuration spécifique à phpMyAdmin. En reliant le fichier sans le dupliquer, cela simplifie la gestion et les mises à jour de la configuration. Le redémarrage d'Apache est nécessaire pour prendre en compte ce changement.

Dans la photo suivante on y voit une partie du fichier config.inc.php utilisé pour configurer phpMyAdmin. Ici, le fichier définit la connexion au serveur MariaDB/MySQL. La variable `$cfg['Servers'][$i]['host']` est configurée sur 'localhost', ce qui signifie que phpMyAdmin tentera de se connecter à une base de données locale. Ensuite, `$cfg['Servers'][$i]['socket']` est spécifiée avec le chemin `/var/run/mysqld/mysqld.sock`, indiquant que la connexion se fera via un socket Unix (et non par TCP/IP).



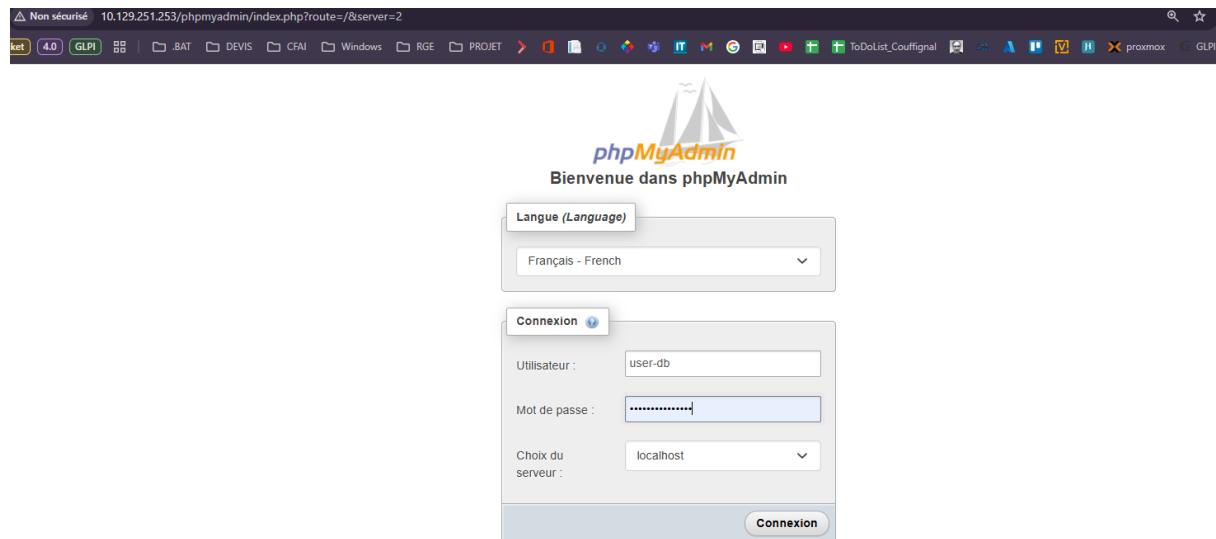
```
config.inc.php
/etc/phpmyadmin

mqtt_to_mariadb.py

/*
 * Authentication type */
// $cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['socket'] = '/var/run/mysqld/mysqld.sock';
```

Connexion PHPMyAdmin/MariaDB

Une fois cette manipulation terminée, la connexion à l'interface web de PHPMyAdmin est possible (<http://localhost/phpmyadmin/>) via le serveur ou (<http://10.129.251.253/phpmyadmin/>) via un poste présent sur le même réseau. La connexion n'est possible que via le socket unix afin d'éviter que des trames passe par le réseau, cela est plus sécurisé.



Connexion à PHPMyAdmin

Lors de cette connexion, il nous est désormais possible de visualiser la base de données mise en place récemment. L'identifiant de la valeur récoltée (id), la date et l'heure à laquelle elle a été récoltée (timestamp) et la température (temperature) peuvent être aperçu dans la table 'mesures_temperatures' sur phpmyadmin.

	id	timestamp	temperature
<input type="checkbox"/>	461816	2025-04-27 15:28:02	19.8
<input type="checkbox"/>	461815	2025-04-27 15:27:56	20.2
<input type="checkbox"/>	461814	2025-04-27 15:27:51	20.2
<input type="checkbox"/>	461813	2025-04-27 15:27:46	20.2
<input type="checkbox"/>	461812	2025-04-27 15:27:40	20.2
<input type="checkbox"/>	461811	2025-04-27 15:27:35	20.2

Visualisation des données via PHPMyAdmin

15. Annexes

La configuration initiale de phpMyAdmin ne constitue qu'une première étape. Pour renforcer la sécurité globale du système, deux documentations complémentaires ont été réalisées. La première traite de la sécurisation des communications entre MariaDB et phpMyAdmin. La seconde détaille la mise en place de certificats OpenSSL (HTTPS) pour phpMyAdmin, afin d'assurer l'authenticité du serveur et la confidentialité des données échangées entre l'utilisateur et l'interface web.

Ces deux démarches permettent de renforcer significativement la protection de l'outil d'administration, en conformité avec les bonnes pratiques en matière de sécurité des systèmes d'information.

Lien vers ces deux documentations :

- [Sécurisation des communications entre MariaDB et PHPMyAdmin](#)
- [Certification OpenSSL PHPMyAdmin \(connexion via `https://localhost/phpmyadmin/`\)](#)

Voici également un lien vers le cahier de recette cité plusieurs fois dans ce rapport :

- [Cahier de recette \(une grande partie des documentations faites s'y trouve\)](#)

Vous pouvez également accéder à ces trois documentation via [easi-training](#).

16. Conclusion

Ce projet avait pour objectif de concevoir et de déployer un système complet de surveillance de la température au sein de la salle serveur du lycée Louis COUFFIGNAL de Strasbourg. La démarche a consisté à installer un capteur capable de remonter en temps réel les mesures de température, à développer une interface de visualisation avec Node-RED, et à configurer un système d'alerte automatique par e-mail en cas de dépassement des seuils prédefinis.

Afin d'assurer la conservation et l'exploitation des données collectées, une base MariaDB a été mise en place, permettant l'archivage structuré des températures. L'intégration sécurisée avec PHPMyAdmin a ensuite facilité l'analyse graphique de ces données historiques, offrant une meilleure visibilité sur l'évolution des conditions environnementales de la salle serveur.

La réalisation de ce projet a permis de mobiliser des compétences variées, allant de l'installation de capteurs et la configuration de serveurs à la sécurisation des communications et à la gestion de bases de données. Au-delà de l'aspect technique, ce travail s'inscrit dans une démarche de prévention des risques et de fiabilisation de l'infrastructure informatique de l'entreprise.

Enfin, ce système reste évolutif. Des perspectives d'amélioration peuvent être envisagées, telles que l'ajout de capteurs complémentaires (comme l'humidité), l'ajout de capteurs dans une grande totalité des baies de brassages afin d'obtenir une visualisation complète du matériel informatique ou encore l'intégration de solutions d'analyse prédictive pour anticiper d'éventuels incidents.