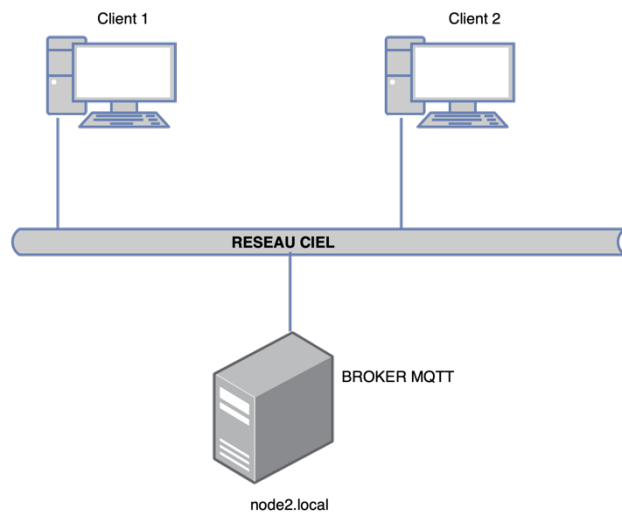


# TP PROTOCOLE MQTT

## Test des abonnements et publications MQTT avec Mosquitto :

Un broker MQTT, nommé « node2.local », est installé et configuré sur le réseau. Cette partie consiste à utiliser deux clients MQTT : l'un en tant que « publisher » et l'autre en tant que « subscriber ».



1. Vérifier que mosquitto est lancé sur chaque client :

```
systemctl status mosquitto
```

2. Choisissez un topic et un message de votre choix :

- Côté subscriber (client 1) :

```
mosquitto_sub -h node2.local -t your/topic
```

- Côté publisher (client 2):

```
mosquitto_pub -h node2.local -t your/topic -m "message"
```

3. Lancer wireshark sur les deux clients

```
sudo wireshark
```

4. Visualiser les requêtes pour les commandes précédentes en appliquant un filtre mqtt

## Le wildcard

un "wildcard" (ou caractère générique) est un caractère spécial utilisé dans les topics pour s'abonner à plusieurs topics à la fois. MQTT propose deux wildcards : + et #.

+ (Wildcard à un niveau) :

Le caractère + peut remplacer un seul niveau de hiérarchie dans un topic.

Par exemple, si vous vous abonnez au topic maison/+/lumière, vous recevrez des messages de maison/salon/lumière, maison/cuisine/lumière, etc., mais pas de maison/salon/lumière/plafond.

# (Wildcard multi-niveaux) :

Le caractère # peut remplacer plusieurs niveaux de hiérarchie dans un topic.

Il doit toujours être utilisé à la fin d'un topic.

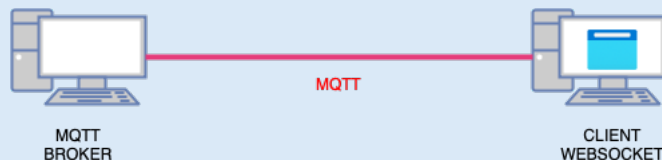
Par exemple, si vous vous abonnez au topic maison/#, vous recevrez des messages de tous les topics qui commencent par maison/, comme maison/salon/lumière, maison/cuisine/thermostat, maison/jardin/arrosage, etc.

5. Tester l'abonnement wildcard multi-niveaux :

```
mosquitto_sub -h node2.local -t '#'
```

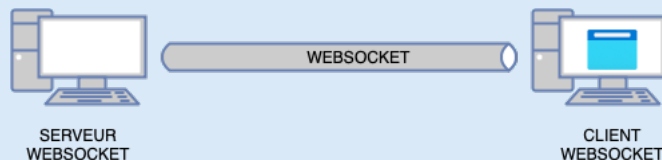
## MQTT sur WEBSOCKET

MQTT côté navigateur :



Dans un navigateur web, vous ne pouvez pas directement utiliser MQTT sans passer par WebSockets en raison des restrictions de sécurité et des capacités des navigateurs. Les navigateurs ne peuvent pas ouvrir des connexions TCP arbitraires (comme requis pour MQTT natif) pour des raisons de sécurité ; ils sont limités à HTTP/HTTPS et WebSockets pour la communication réseau.

Rappel sur les websockets

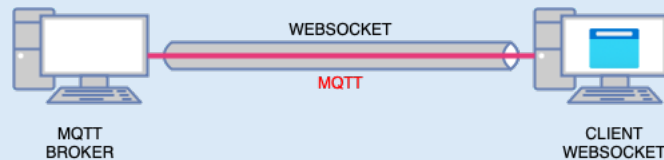


On peut définir une WebSocket comme un canal, mais pour être plus précis, c'est un canal de communication bidirectionnel ouvert sur une seule connexion TCP. Les WebSockets permettent une communication en temps réel entre le client (par exemple, un navigateur web) et le serveur. Contrairement au modèle traditionnel de requête-réponse HTTP, où chaque échange nécessite

une nouvelle connexion, une connexion WebSocket reste ouverte, permettant aux données de circuler librement et instantanément dans les deux sens tant que la connexion reste active.

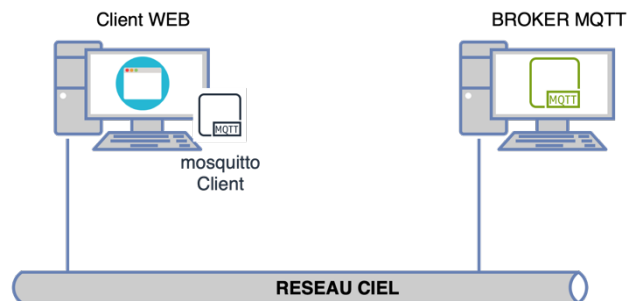
#### MQTT sur websocket :

MQTT sur WebSocket est utilisé pour permettre aux clients web (navigateurs) d'interagir avec un serveur MQTT. Les navigateurs ne supportent pas directement le protocole MQTT sur TCP, donc MQTT sur WebSocket est utilisé comme une couche de transport pour permettre cette communication.



#### Test du MQTT sur websocket :

*Utiliser un PC en broker avec mosquitto. L'autre PC publiera des messages sur le topic newton/formation avec une commande mosquitto. L'abonnement se fait depuis une page web en mqtt sur websocket.*



6. Editer le fichier de configuration du broker :

```
sudo /etc/mosquitto/mosquitto.conf
```

7. Ajouter en fin de fichier les lignes suivantes :

```
listener 9001
protocol websockets
```

8. Relancer le service mosquito :

```
systemctl restart mosquitto
```

9. Télécharger la page web html.

<https://github.com/bouhenic/FormationIOT/tree/main/Tpmqtt/index.html>

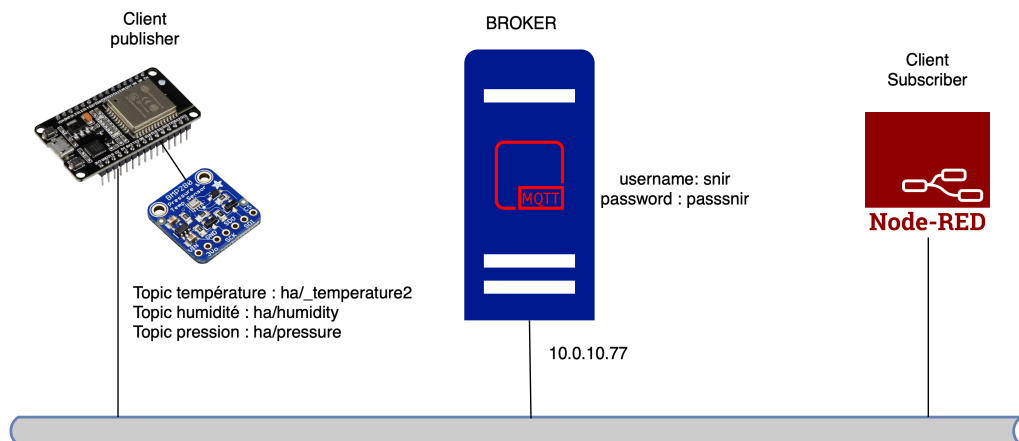
10. Modifier le code (adresse IP du broker et topic).

11. Ouvrir la page web depuis un navigateur.

12. Publier un message sur le topic configuré.

13. Vérifier l'affichage du message sur la page web.

Abonnement à un topic en utilisant NodeRed :



*Un ESP32 équipé d'un capteur BME280 publie 3 mesures (température, humidité et pression) vers un broker mqtt installé sur le réseau. Cette partie consiste à s'abonner aux topics à partir d'une application Node-Red.*

14. Tester depuis Node-Red un abonnement au topics listés précédemment. On utilisera un nœud « mqtt in » et un nœud « débogueur ».



**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server localhost:1883

Topic ha/\_temperature2

QoS 2

Output auto-detect (string or buffer)

Name Name

On saisit le topic

**Edit mqtt in node > Edit mqtt-broker node**

Delete Cancel Update

**Properties**

Name Name

**Connection** Security Messages

Server 10.0.10.77 Port 1883

☐ Enable secure (SSL/TLS) connection

Client ID Leave blank for auto generated

Keep alive time (s) 60 ☒ Use clean session

☐ Use legacy MQTT 3.1 support

On saisit ici les paramètres du broker

On saisit ici les secrets d'accès au broker

**Properties**

Name Name

**Connection** Security Messages

Username snir

Password .....