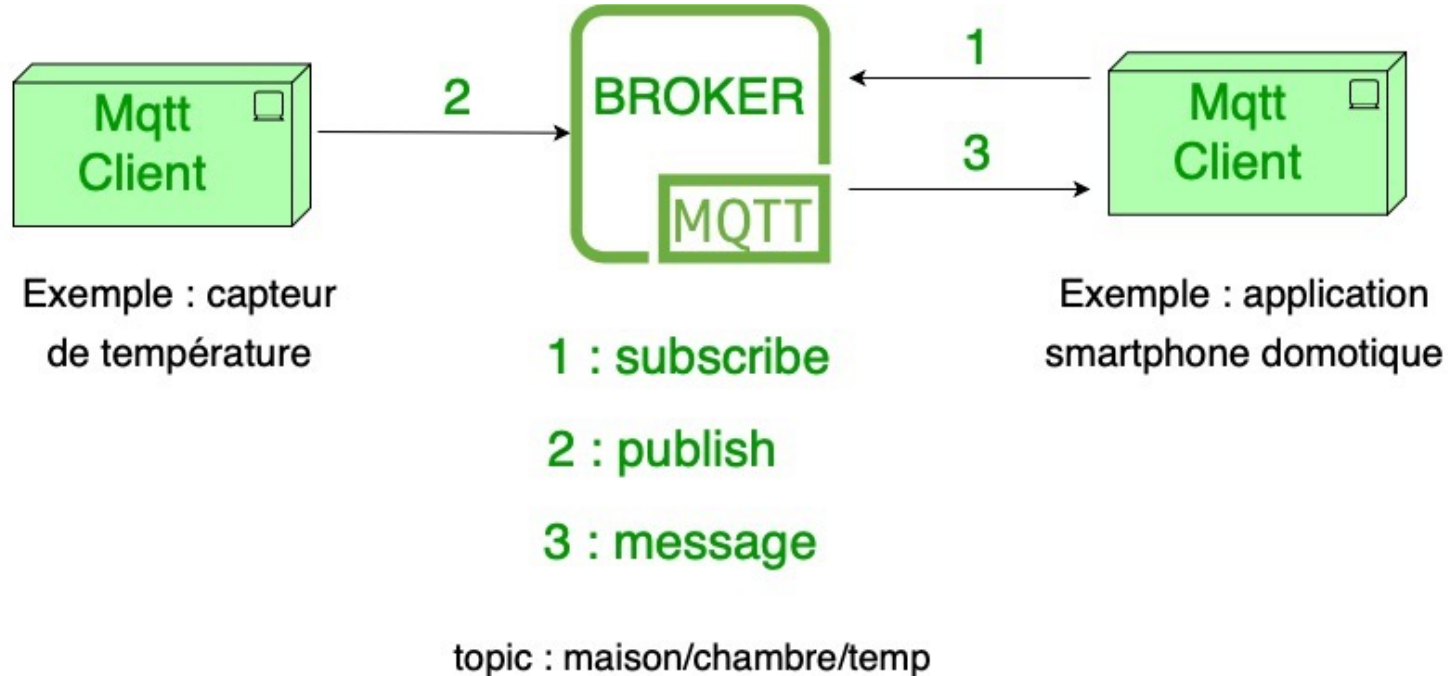


# PROTOCOLE MQTT



Le protocole MQTT est assez basique dans son fonctionnement, c'est d'ailleurs sa force. Il s'agit d'un protocole de messagerie machine to machine.

Concrètement les équipements connectés publient et/ou s'abonnent à un **topic** qui référence les messages et les communique aux abonnés.

Mosquitto est un logiciel open-source qui sert à la fois de broker et de client pour le protocole MQTT (MQ Telemetry Transport), conçu pour la messagerie légère et efficace dans les réseaux IoT (Internet des Objets).

Installer Mosquitto :

```
sudo apt install -y mosquitto mosquitto-clients
```

Pour lancer Mosquitto :

```
systemctl start mosquitto
```

Pour s'abonner à un topic :

```
mosquitto_sub -h localhost -t your/topic
```

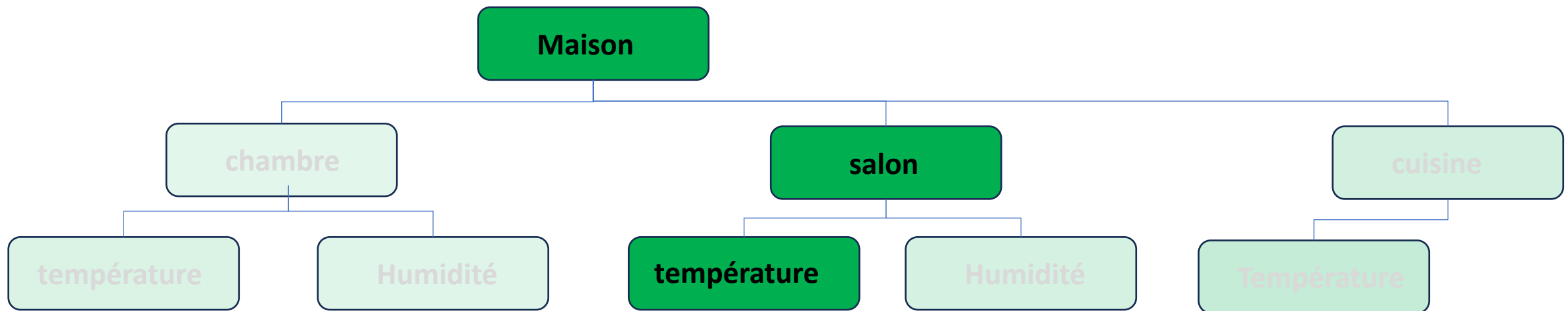
Pour publier un message sur un topic :

```
mosquitto_pub -h localhost -t your/topic -m "Hello MQTT"
```

# LES TOPICS

Les topics sont structurés de manière hiérarchique avec des niveaux séparés par des barres obliques ("/"), ce qui permet une organisation similaire à celle des chemins dans un système de fichiers.

Par exemple, dans un topic tel que "maison/salon/temperature", "maison" est le niveau le plus élevé de la hiérarchie, "salon" est un sous-niveau spécifiant une pièce de la maison, et "temperature" indique le type d'information (dans ce cas, la température du salon).



- Le signe plus ("+") peut être utilisé comme joker pour un seul niveau de topic. Par exemple, "maison/+/temperature" s'abonnerait aux messages de température de toutes les pièces de la maison.
- Le signe dièse ("#") peut être utilisé comme joker pour plusieurs niveaux de topics à partir de sa position dans le topic. Par exemple, "maison/#" s'abonnerait à tous les messages commençant par "maison".

# PROTOCOLE MQTT

C'est un protocole applicatif. Il fonctionne sur TCP. Il utilise le port 1883 et 8883 pour le mqtts.

No.	Time	Source	Destination	Protocol	Info
44	1...	192.168.1.39	3.68.58.135	MQTT	Connect Command
46	1...	3.68.58.135	192.168.1.39	MQTT	Connect Ack
48	1...	192.168.1.39	3.68.58.135	MQTT	Subscribe Request
50	1...	3.68.58.135	192.168.1.39	MQTT	Subscribe Ack

```
> Frame 44: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
> Ethernet II, Src: 78:4f:43:6c:f5:bb (78:4f:43:6c:f5:bb), Dst: 20:66:cf:68:f2:a0 (20:66:cf:68:f2:a0)
> Internet Protocol Version 4, Src: 192.168.1.39, Dst: 3.68.58.135
> Transmission Control Protocol, Src Port: 64301 (64301), Dst Port: mqtt (1883), Seq: 1, Ack: 1, Len: 14
> MQ Telemetry Transport Protocol
```

Échange de trames entre broker et client lors d'un abonnement à un topic :

1. **Connect Command** : requête de connexion au broker.
2. **Connect Ack (Acknowledgement)** : qui est la confirmation du broker que la commande de connexion a été reçue et acceptée.
3. **Subscribe Request** : le client demande à s'abonner à un topic.
4. **Subscribe Ack** : confirmation du broker que la demande d'abonnement a été reçue et que le client est maintenant abonné au topic demandé.

# PROTOCOLE MQTT

No.	Time	Source	Destination	Protocol	Info
✓ 234	40.759562	192.168.1.39	3.66.104.219	MQTT	Ping Request
238	40.775686	3.66.104.219	192.168.1.39	MQTT	Ping Response

- > Frame 238: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
- > Ethernet II, Src: 20:66:cf:68:f2:a0 (20:66:cf:68:f2:a0), Dst: 78:4f:43:6c:f5:bb (78:4f:43:6c:f5:bb)
- > Internet Protocol Version 4, Src: 3.66.104.219, Dst: 192.168.1.39
- > Transmission Control Protocol, Src Port: mqtt (1883), Dst Port: 64204 (64204), Seq: 1, Ack: 3, Len: 2
- > MQ Telemetry Transport Protocol

Les Ping Request et Ping Response dans MQTT sont utilisés pour s'assurer que la connexion entre le client et le broker reste ouverte et que le client est toujours actif. C'est une partie essentielle de la gestion de la session MQTT, particulièrement utile lorsque les connexions doivent être maintenues ouvertes même en l'absence de communication de données régulière.

# PROTOCOLE MQTT

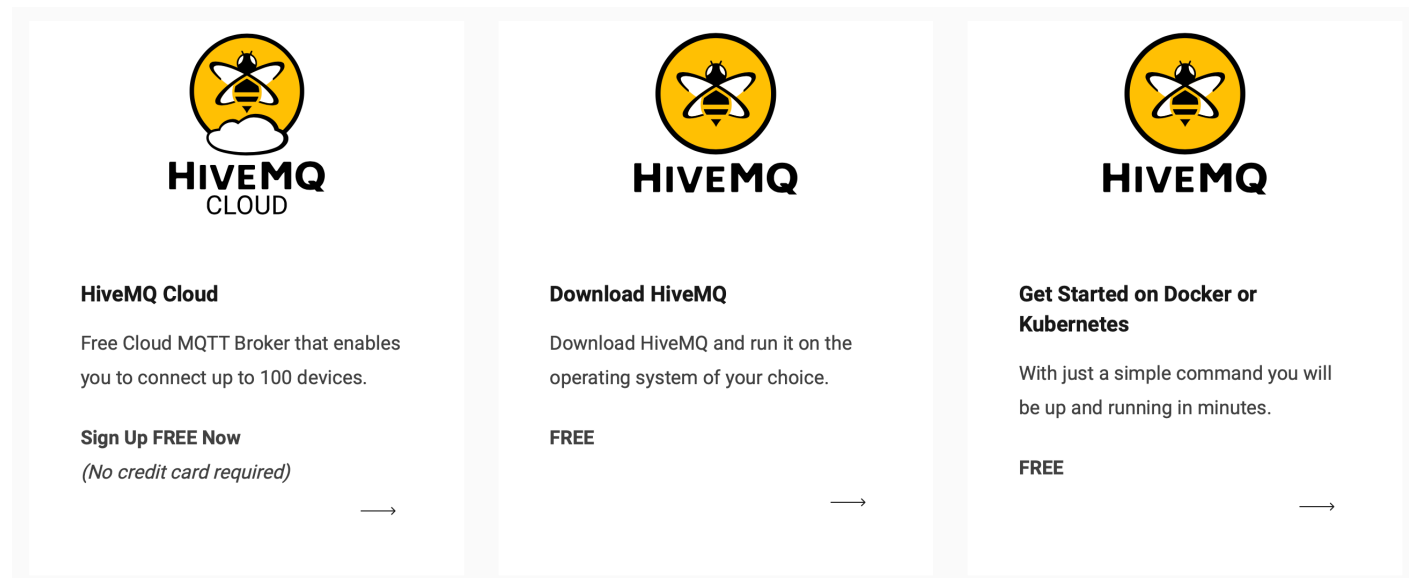
No.	Time	Source	Destination	Protocol	Info
79	1...	192.168.1.39	3.68.58.135	MQTT	Connect Command
81	1...	3.68.58.135	192.168.1.39	MQTT	Connect Ack
83	1...	192.168.1.39	3.68.58.135	MQTT	Publish Message
84	1...	192.168.1.39	3.68.58.135	MQTT	Disconnect Req

> Frame 79: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)  
> Ethernet II, Src: 78:4f:43:6c:f5:bb (78:4f:43:6c:f5:bb), Dst: 20:66:cf:68:f2:a0 (20:66:cf:68:f2:a0)  
> Internet Protocol Version 4, Src: 192.168.1.39, Dst: 3.68.58.135  
> Transmission Control Protocol, Src Port: 64538 (64538), Dst Port: mqtt (1883), Seq: 1, Ack: 1, Len: 14  
> MQ Telemetry Transport Protocol

1. **Connect Command** : c'est la commande envoyée par un client MQTT pour établir une connexion avec un broker MQTT.
2. **Connect Ack (Connect Acknowledgment)** : c'est la confirmation du broker que la connexion a été établie.
3. **Publish Message** : le client MQTT publie un message sur un topic.
4. **Disconnect Req (Disconnect Request)** : c'est la demande de déconnexion envoyée par le client MQTT pour terminer la session de communication avec le broker MQTT.

# HIVEMQ

HiveMQ est une plateforme de messagerie MQTT conçue pour faciliter la communication entre appareils dans les applications Internet des Objets (IoT). Elle fournit un broker MQTT hautement évolutif et performant, capable de gérer des millions de messages et des milliers de clients simultanément.



Plusieurs solutions possibles :

- Utiliser **broker.hivemq.com** pour faire des tests (c'est sans inscription).
- S'inscrire sur le cloud
- Télécharger l'application et l'installer sur une machine.
- Utiliser une image Docker.

# AUTRES OPTIONS MOSQUITTO

```
mosquitto_pub -h <broker> -t your/topic -m "message" -r
```

-r : Cette option est utilisée pour retenir le message. Lorsqu'un message est retenu, le dernier message publié sur un topic sera stocké par le broker et immédiatement envoyé à tout client qui s'abonne à ce topic.

```
mosquitto_rr -h <broker> -t your/topic -e other/topic -m "message"
```

La commande `mosquitto_rr` est une commande du client MQTT Mosquitto qui est utilisée pour envoyer un message MQTT et attendre une réponse, ce qui est un modèle de communication request-response (requête-réponse).

-e other/topic : L'option -e spécifie le topic d'écoute pour la réponse.



# UN PEU DE SÉCURITÉ

Authentification avec mot de passe :

```
mosquitto_pub -h <broker> -u <user> -P <password> -t your/topic -m "message"
```

Connexion chiffrée avec le mqttts (port 8883) avec certificat signé par CA :

```
mosquitto_pub -h <broker> -p 8883 -t your/topic -m "message"
```

Connexion chiffrée avec le mqttts (port 8883) avec certificat auto-signé:

```
mosquitto_pub -h <broker> -p 8883 --cafile /ca.crt -t your/topic -m "message"
```

Connexion chiffrée avec le mqttts (port 8883) avec certificat auto-signé et authentification avec certificat client :

```
mosquitto_pub -h 172.27.0.2 -p 8883 --cafile /ca.crt --cert /client.crt --key /client.key -t "your/topic" -m "message"
```