

	<b>C09 Installer un réseau informatique</b> Protocole MQTT Systèmes d'exploitation	<b>BTS CIEL 1</b>
	<b>C05 Concevoir un système informatique</b> Infrastructures logicielles : centralisées, décentralisées ou réparties	<b>TP</b>

# TP : Mise en œuvre de MQTT

Objectifs .....	1
Architecture .....	2
Mise en place .....	2
Installation du client MQTT .....	2
Client : s'abonner à un topic.....	3
Client : publier sur un topic.....	4
Mise en œuvre de MQTT en réseau .....	5
Utilisation de MQTT sur le cloud Adafruit.....	6
Présentation du cloud Adafruit .....	6
Feeds : injecter des données .....	7
Dashboard : tableau de visualisation .....	8

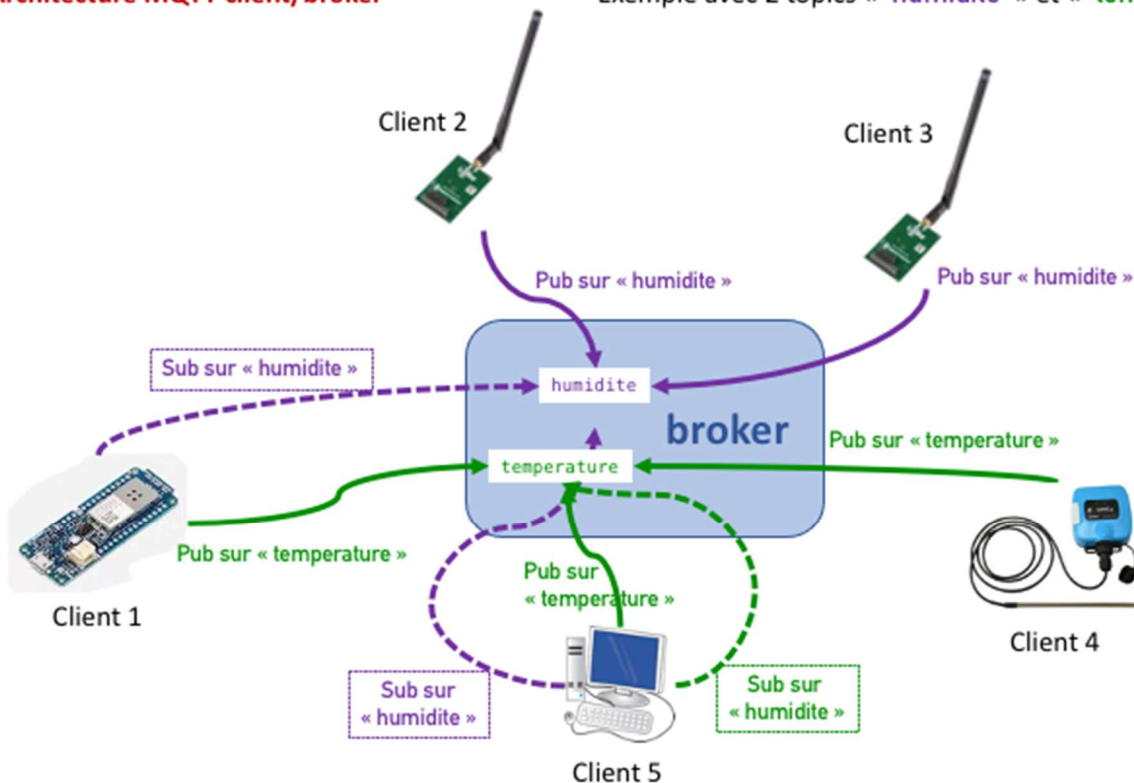
## Objectifs

- Mettre en œuvre une architecture MQTT en local et sur un réseau distant.
- Utiliser le protocole MQTT en ligne de commande (VM Unix)
- Utiliser une authentification MQTT sur un broker distant
- Créer un tableau de visualisation sur le cloud grâce aux requêtes MQTT

# Architecture

## Architecture MQTT client/broker

Exemple avec 2 topics « humidite » et « temperature »



- le **broker** centralise les abonnements et redirige les messages publiés
- le **client** peut publier et/ou s'abonner en s'adressant à un broker

## Mise en place

Pour ce TP

- Votre VM Debian sera un **client** dans l'architecture MQTT
- le **broker** MQTT sera installé et configuré par l'enseignant(e)<sup>1</sup>

MQTT est un protocole, il existe plusieurs logiciels (gratuits ou payants) qui implémentent le protocole en version **client** ou **broker**. Pour la mise en œuvre de MQTT, nous utiliserons [Mosquitto](#), une implémentation simple du protocole MQTT en ligne de commande.

**Attention** : Mosquitto existe en version **broker** ou en version **client**

## Installation du client MQTT

- Démarrer la VM et mettre à jour le système
- Ouvrir une session SSH du PC sur la VM

<sup>1</sup> VM ou Raspberry Pi

- Quelle commande permet d'installer le **client** MQTT Mosquitto sur Debian ? (en cherchant sur Internet)

- Installer le **client Mosquitto** sur votre VM
- Noter l'adresse IP du broker donné par l'enseignant(e)

### Client : s'abonner à un topic

La syntaxe minimale pour s'abonner à un topic est la suivante

```
mosquitto_sub -h <hostname> -t <topic>
```

Options courantes

<b>-h</b> <hostname>	Pour indiquer le broker (hostname ou IP)
<b>-t</b> <topic>	Pour indiquer le topic
<b>-u</b> <username>	<b>username</b> : authentification auprès du broker
<b>-P</b> <password>	<b>password</b> : authentification auprès du broker

Autres options possibles

<b>-d</b>	<b>debug</b> avec cette option les différents messages sont affichés : connexion, déconnexion, publish
<b>-C</b> <xxx>	se déconnecte après xx messages. Exemple : -C 4

Sur le réseau, pour les tests, le broker est configuré pour accepter les connexions anonymes (sans authentification).

- Entrer la commande pour que le **client** s'abonne à un topic '**ciel**' en mode debug

- Observer l'activité sur le terminal
- Observer et analyser l'activité si aucun message n'est publié ? (attendre 2 minutes minimum)

Activité sur le **client** sub (pings réguliers pour se manifester et rester enregistré auprès du broker)

## Client : publier sur un topic

La syntaxe pour publier sur un topic est la suivante :

```
mosquitto_pub -h <hostname> -t <topic> -m <message>
```

Options courantes

<b>-h</b> <hostname>	Pour indiquer le broker (hostname ou IP)
<b>-t</b> <topic>	Pour indiquer le topic
<b>-m</b> <message>	Pour indiquer le message. Exemples -m 42 -m "exemple"

Autres options possibles

<b>-d</b>	debug avec cette option les différents messages sont affichés : connexion, déconnexion, publish
<b>-p</b> <port>	numéro de port (port par défaut si l'option n'est pas utilisée)
<b>-u</b> <username>	username : authentification auprès du broker
<b>-P</b> <password>	password : authentification auprès du broker

- Ouvrir une nouvelle session SSH pour les requêtes de publication
- Vous devez avoir deux terminaux SSH connectés à la VM
  - fenêtre **client** SUB
  - fenêtre **client** PUB
- Écrire la commande permettant de publier les message "test mqtt de Votre Nom" sur le topic "ciel" en mode debug (fenêtre **client** PUB)

- Expliquer les activités affichées sur les terminaux lors de l'envoi d'un message?

activité sur le **client** sub

activité sur le **client** pub

## Mise en œuvre de MQTT en réseau

Pour cette partie vous travaillerez par trinôme.

Notez les noms des étudiants du groupe ainsi que le nom du TOPIC attribué par l'enseignant(e)

Chaque étudiant aura deux fenêtres

- une fenêtre avec le **client** sub
- une fenêtre avec le **client** pub
  
- Chaque étudiant s'abonne à un topic (voir tableau) sur une fenêtre du terminal
- Chaque étudiant publie ponctuellement un nombre sur un topic (voir tableau)

étudiant 1 (nom)	étudiant 2 (nom)	étudiant 3 (nom)
sub topic/temperature	sub à topic/#	sub topic/humidite

- Compléter le tableau avec vos tests

messages reçus 0/N	pub sur topic/humidite	pub sur topic/temperature
sub sur topic/humidite		
sub sur topic/#		
sub sur topic/temperature		

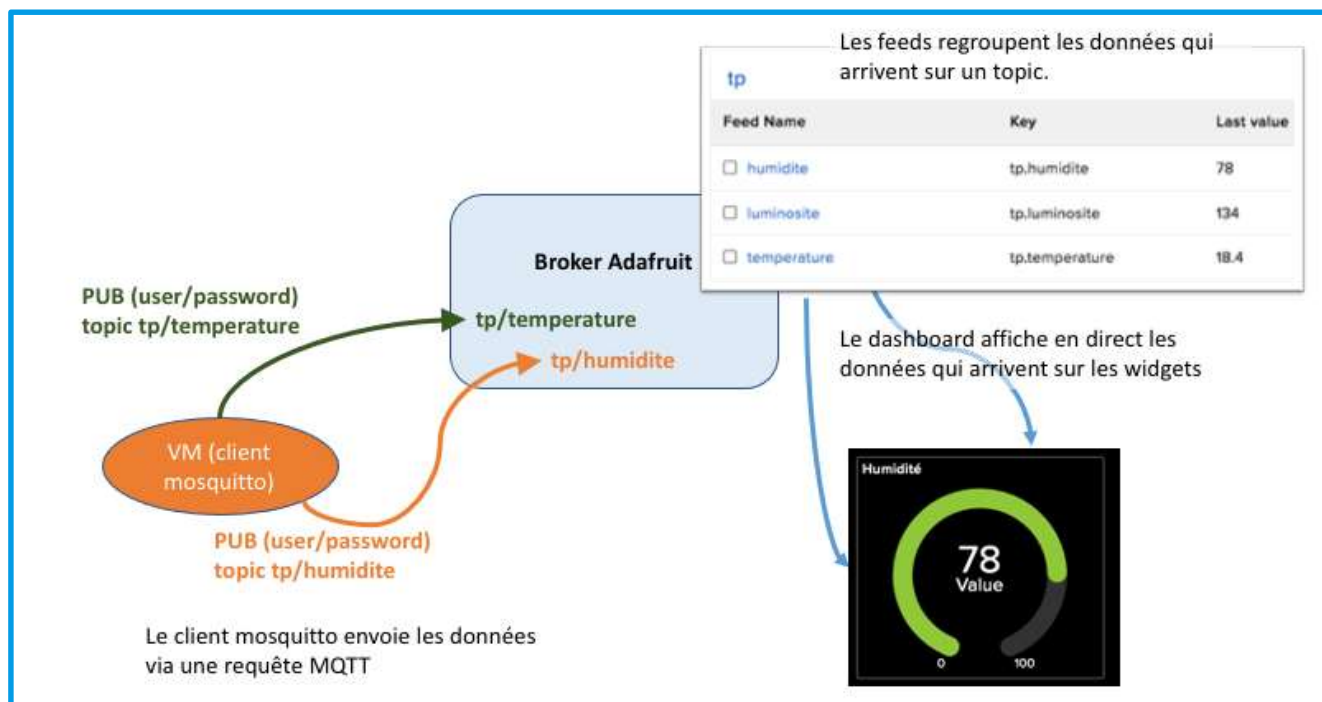
- Conclure sur l'utilisation du #

# Utilisation de MQTT sur le cloud Adafruit

## Présentation du cloud Adafruit

Le cloud Adafruit permet de créer des dashboards (tableaux de bords) à partir de fils de données. Les données peuvent être envoyées de plusieurs façons, parmi lesquelles MQTT.

L'objectif est d'afficher sur un dashboard des données qui arrivent sur votre compte cloud Adafruit, les données étant envoyées via une requête MQTT depuis le client mosquitto.



Quelques éléments :

Dans l'architecture MQTT, io.adafruit.com joue le rôle du **broker**.

**Feed** = un fil de données

un feed regroupe les données qui arrivent sur un **topic**

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> humidite	ciel.humidite	78	less than a minute ago
<input type="checkbox"/> luminosite	ciel.luminosite	134	almost 2 years ago
<input type="checkbox"/> temperature	ciel.temperature	18.4	11 months ago

Le « topic » qui correspond au feed est indiqué dans la propriété « MQTT by Key »

## Edit Feed



### Name

humidite

Maximum length: 128 characters. Used: 8

### Key

humidite

Changing the key will change API URLs and MQTT subscription topics. The only characters we permit are lower case english letters ("a" to "z"), numbers, and dash ("-").

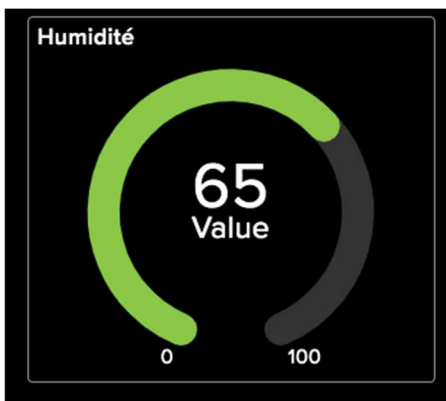
See our [guide to naming things in Adafruit IO](#) for more information about how we handle the formatting of names and keys.

### Current Endpoints

Web <https://io.adafruit.com/ProfTonnerre/feeds/tp.humidite>

API <https://io.adafruit.com/api/v2/ProfTonnerre/feeds/tp.humidite>

MQTT  
by Key [ProfTonnerre/feeds/tp.humidite](#)



**Dashboard** = tableau de bord

Permet d'afficher graphiquement les données qui arrivent sur un feed (topic). Il est possible d'ajouter plusieurs blocs d'affichage (widgets) pour un même feed

Les données d'un feed peuvent être ajoutées sur plusieurs dashboards.

## Feeds : injecter des données

- Créer un compte gratuit sur [io.adafruit.com](https://io.adafruit.com) (en utilisant votre mail du lycée)
- Créer deux feed : **temperature** et **humidite** dans un groupe **tp**

**feed** > **feed info** = informations permettant d'injecter des données sur ce feed (topic)

**Adafruit IO key** = informations user/mot de passe de connexion au **broker**

Compléter le tableau

Paramètre	Nom du paramètre sur Adafruit	Valeur à renseigner
hostname		
user		

password		
topic humidite		
topic temperature		

- Écrire la commande qui permet d'injecter des données sur un des feeds (copie d'écran)

- Vous devez voir les données arriver sur le feed (copie d'écran)

### **Dashboard : tableau de visualisation**

- Créer un Dashboard TP et insérer des widgets de visualisation de vos données. (au minimum une jauge et un graphe). Vous pouvez envoyer des données sur le feed d'un camarade.
- Insérer une copie d'écran