h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève
F. Vannel, S. Fourquier

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences
Western Switzerland

Course Project
IoT - HES-SO Master
2018 - 2019 Winter Semester
The complete IoT project

Objectifs

Le projet suivant a pour objectif de créer un système IoT à même de pouvoir être déployé à large échelle. La solution actuellement développée a permis de comprendre le principe de la technologie LoRa et l'utilisation de la solution de gestion de réseau **The Things Network (TTN)**. La visualisation des données sous forme graphique a été gérée par une solution simple à mettre en oeuvre nommée **mydevices**. Celle-ci est parfaite pour un démonstrateur (ou *proof of concept*) mais ne permet pas de gérer facilement une flotte de plusieurs capteurs.

Plusieurs solutions existent proposant des outils pour la gestion de grosses flottes de capteurs et également de plusieurs projets ou clients. Nous allons dans ce projet utiliser la solution commerciale de Amazon : **AWS (Amazon Web Service)**

Rendu du travail

Le rendu de ce projet est à fournir pour le jeudi 10 janvier 2019. Il s'agit d'un travail réalisé au maximum par groupe de deux. A cette date le matériel devra être rendu

A la fin de ce projet, vous devez :

1. écrire un bref rapport décrivant l'architecture complète de votre système (y compris l'architecture avec les différents outils AWS utilisés) et avec des captures d'écran des parties les plus importantes de votre solution AWS.
2. Faire une vidéo démontrant votre projet et incluant les détails visible sur les différentes interfaces utilisé (console de Waspmonte, console de TTN, différents outils de AWS, etc...). Si le temps le permet une démonstration à un professeur ou assistant pourra également être faite.
3. Préparer un ZIP avec tous les codes écrits, le rapport et la vidéo et le déposer sur cyberlearn au plus tard le 17 janvier.

Description du projet

Vous devez mettre en place un système capturant des données depuis plusieurs capteurs et permettant de visualiser sur une page web les données reçues de ces capteur (température, accélérations, ...). La gestion d'une flotte de capteur est important dans ce projet et les sources

de données seront diverses. Au travers d'un algorithme ou d'une page web il sera possible de simuler un actionneur, par exemple l'allumage d'une LED sur un des devices.

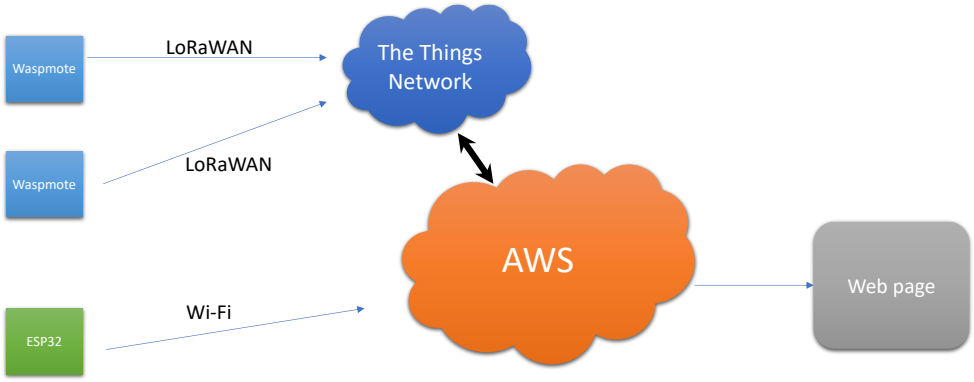

FIGURE 1. *Architecture globale du projet*

Mise en place du projet

1 Relier The Things Network à Amazon Web Services

Il faut disposer d'un compte sur Amazon AWS. Bien que la plupart des services que vous utiliserez seront normalement gratuit, vous pouvez demander un crédit en suivant le guide sur cyberlearn. Attention le crédit d'argent n'est attribué qu'après quelques jours.

Le tutoriel proposé par **The Things Network** vous permettra de réaliser l'intégration de votre capteur avec AWS, puis par la suite la gestion de tous ceux-ci et leur ajout à partir de AWS et non plus TTN :

`https://www.thethingsnetwork.org/docs/applications/aws/`

Si vous souhaitez aller un peu plus loin dans des exemples d'utilisation de services Amazon avec **TTN** il existe également ce tutoriel qui vous permettra d'utiliser d'autre services :

`https://github.com/Klarrio/ttn-aws-workshop`

Attention : Il est important en suivant ces tutoriels de bien faire attention à la **région** que vous avez sélectionnée. Pour information un service **AWS** déployé dans une région x est par défaut invisible dans une autre région y. Donc choisissez une région et restez fidèle à celle-ci !

Par ailleurs vous pourriez ne pas être en mesure de voir certains services ou fonctionnalitées à cause de droits insuffisants. L'outil **IAM** est la clé de votre salut ou de vos problèmes. Apprenez à bien l'utiliser.

2 Un capteur Wi-Fi

Nous vous invitons à ajouter un capteur connecté en direct sur le cloud **AWS**, par exemple un capteur de température connecté directement à internet par du Wi-Fi.

2.1 Le ESP32

Amazon fournit le code à déployer dans le micro-contrôleur pour permettre de connecter directement le *device* de façon sécurisée au Service **IoT de AWS**.

Cependant avant tout il faudra travailler avec un utilisateur dédié à l'ESP32 car dans un but de sécurité du système il n'est pas permis d'utiliser votre `username AWS` mais un `user` qui ne disposera que des droits nécessaires pour le capteur.

2.2 La création d'un user pour le ESP32

— Allez dans le service IAM et créez un utilisateur avec une access key ID et une secret access key.
— Depuis AWS Management Console access lui attacher pour le moment la policies *AdministratorAccess*
— Recopiez dans un fichier l'Access key ID et le Secret access key. Vous devrez l'utiliser dans le code de votre firmware de l'ESP32.
— Il faudra désormais utiliser cette utilisateur également sur votre console AWS. Deconnectez-vous puis connectez-vous avec ce nouvel user.

Le tutoriel suivant vous guidera sur les autres parties du travail à réaliser :

`https://docs.aws.amazon.com/freertos/latest/userguide/getting_started_espressif.html`

Le code exemple à déployer dans le ESP32 se trouve à présent dans votre disque dur dans le répertoire `/amazon-freertos/demos/common/mqtt/`

2.3 Test

Normalement ce code envoi des message MQTT dans `freertos/demos/echo`

Depuis AWS IoT, vérifiez si ces messages arrivent bien.
— Allez dans Service IoT core, puis test.
— Subscription topic : freertos/demos/echo.
— Verifiez que des messages arrivent bien (redémarrer l'ESP32 si nécessaire).
Si cela fonctionne correctement modifiez le code pour lire un capteur de température.

2.4 Ajout d'un capteur de température

Nous allons brancher le capteur MCP9700A sur 3 GPIO successifs. 2 seront configurés en sortie pour alimenter le capteur et un en entrée analogique pour mesurer la température. Nous choisissons les GPIO 27, 33 et 15. (pin 27 masse de lu capteur, 15 alimentation 3.3V et pin 33 entrée analogique)

Dans le code de l'ESP32, changez le topic MQTT pour par exemple par `"sensor/temp"`

Le code d'exemple actuel s'arrête après une minute. Modifiez-le pour envoyer en continue `for( xX = 0; xX < xIterationsInAMinute; xX++ )`

L'initialisation de votre capteur de température se fait ainsi :

1. Set GPIO pour alimenter le capteur

```
gpio_set_direction(27,GPIO_MODE_OUTPUT);
gpio_set_level(27,0);
gpio_set_direction(15,GPIO_MODE_OUTPUT);
gpio_set_level(15,1);
```

2. Le code permettant de lire la valeur du convertisseur analogique vers numérique :

```
#include <driver/gpio.h>
#include <driver/adc.h>
adc1_config_width(ADC_WIDTH_BIT_12);
adc1_config_channel_atten(ADC1_CHANNEL_5,ADC_ATTEN_DB_11);

float val = adc1_get_raw(ADC1_CHANNEL_5);
val=val*3900/(1<<12)-500; //mv
val/=10; //Celcius degrees
```

3. Modifiez le message MQTT pour envoyer un message du type `"Name":"Inside","temp":24.2`
4. Vérifiez le bon fonctionnement de votre capteur

2.5 Stockage des données dans une base de données

1. Créer une table dynamoDB avec comme `key Name(string)` et `sort key Time(number)`
2. Depuis le service IoT Core, dans l'onglet Action, créez une règle.
3. Le champ *query* permet de filtrer les message MQTT. Nous allons ainsi récupérer les messages depuis le *topic* où sont transmis les message. Par exemple : `SELECT Name AS Name, temp AS temp, timestamp() AS Time FROM 'sensor/t'`
   Cette règle ajoutera automatiquement la date dans Time
4. Choisssez comme action : "`Split message into multiple columns of a database table (DynamoDBv2)`".
5. Sélectionnez votre table et créer un IAM adapté.
6. Si besoin vous pouvez ajouter une `error action` du type "This action will write the message to a file in a S3 bucket." Pour ceci, créez un bucket dans S3. Key sera le nom du fichier créée. Créez un IAM adapté.

A présent les données reçues depuis votre capteur doivent s'enregistrer dans dynamoDB.

2.6 Afficher les données dans un graphique

AWS ne dispose pas d'outil parfaitement adapté à la représentation de données de capteurs. La solution optimale est un site internet hébergé dans leurs services. Entre autre la solution simple consiste à stocker les fichiers de la page web dans un bucket S3 et d'aller récupérer les données à afficher depuis la base de donnée dynamoDB.

A nouveau pour des raisons de sécurité il ne vous sera pas possible d'avoir un script dans une page web qui aura le droit d'accéder à d'autres données que celles qu'il doit afficher.

Cet utilisateur aura besoin d'une access key ID et un secret access key ainsi que d'une policy, dont voici un exemple à utiliser (après avoir changé la ligne ressource)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:BatchGetItem",
                "dynamodb:DescribeTable",
                "dynamodb:GetItem",
                "dynamodb:ListTables",
```

```
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:DescribeReservedCapacity",
                "dynamodb:DescribeReservedCapacityOfferings",
                "dynamodb:ListTagsOfResource",
                "dynamodb:DescribeTimeToLive",
                "dynamodb:DescribeLimits",
                "dynamodb:ListGlobalTables",
                "dynamodb:DescribeGlobalTable",
                "dynamodb:DescribeBackup",
                "dynamodb:ListBackups",
                "dynamodb:DescribeContinuousBackups"
            ],
            "Resource": "arn:aws:dynamodb:us-east-1:103694155724:table/mySensor"
        }
    ]
}
```

Pour héberger un site vous pouvez vous référer à ce tutoriel `https://docs.aws.amazon.com/fr_fr/AmazonS3/latest/dev/HowDoIWebsiteConfiguration.html`

Vous trouverez sur cyberlearn deux fichiers (index.html et index.js) que vous pourrez adapter avec votre utilisateur et votre clé de sécurité afin d'afficher les données sur une courbe.

3 Intégration complète

Concevez une solution à même d'afficher des données provenant de vos divers capteurs tel que représenté à la figure 1. Relisez la section décrivant le projet afin de bien respecter le cahier des charges et ajoutez la gestion d'un actionneur (probablement représenté par l'état d'une LED)

Bon travail et soyez imaginatif sur le but de l'application finale.