RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS GTI791 PROJETS SPÉCIAUX EN GÉNIE DES TI

SIMON GERVAIS-QUIBLAT
GERS28029406

Professeur-superviseur
Prof. Alain April

Montréal, 12 avril 2019
Hiver 2019

Remerciements

Tout d’abord, j’aimerais remercier mon professeur-superviseur Alain April pour l’opportunité de pouvoir porter ce projet à terme et aider mon regroupement étudiant.

Par la suite, j’aimerais remercier les anciens membres du Lan ETS de m’avoir intégré au projet et aux membres actuels de m’avoir fait confiance. L’expérience que j’ai pu acquérir en étant membre de ce regroupement étudiant m’a permis d’améliorer mes compétences relatives aux communications et à la gestion de projet.

Gestion de l’affichage numérique

Simon Gervais-Quiblat
GERS28029406

RÉSUMÉ

Suite à l’acquisition de plusieurs écrans d’affichage numérique par le Lan ETS, un nouveau besoin est apparu soit le contrôle de l’affichage de ces écrans à distance pendant l’évènement. Ce projet comporte trois cas d’utilisations. CU1 contrôler l’affichage à distance, CU2 modifier l’affichage de plusieurs écrans simultanément, CU3 maintenir l’affichage en cas de panne du réseau. Ce projet est réalisé à l’aide de la plateforme AWS IoT et d’un Raspberry Pi. Le système sera prêt pour l’édition 2020 du Lan ETS.

Table des Matières

Introduction 10~~9~~

Chapitre 1: OBJECTIF 11~~10~~
- 1.1 Gestion de l'affichage centralisé 11~~10~~
- 1.2 Facilité de modification 11~~10~~
- 1.3 Persistant aux pannes réseaux 11~~10~~

Chapitre 2: Affichage numérique 12~~11~~

Chapitre 3: Restrictions 13~~12~~
- 3.1 Requis réseautique 13~~12~~
- 3.2 Requis matériel 13~~12~~
- 3.3 Type de média 14~~13~~
- 3.4 Budget disponible 14~~13~~

Chapitre 4: AWS 15~~14~~
- 4.1 AWS IoT 15~~14~~
  - 4.1.1 Objets 15~~14~~
    - 4.1.1.1 Objet 15~~14~~
    - 4.1.1.2 Groupe d'objets 15~~14~~
    - 4.1.1..3 Type 16~~15~~
  - 4.1.2 Agent de messages 16~~15~~
  - 4.1.3 Authentification et autorisation 17~~16~~
  - 4.1.4 Shadow d'un objet 17~~16~~
- 4.2 AWS S3 17~~16~~
- 4.3 AWS Lambda 18~~17~~
- 4.4 Tarification 18~~17~~

Chapitre 5: Raspberry Pi 20~~19~~

Chapitre 6: Intégration Raspberry Pi et AWS 21~~20~~
- 6.1 Installation des environnements 21~~20~~
  - 6.1.1 AWS 21~~20~~
  - 6.1.2 Raspberry Pi 21~~20~~
- 6.2 Diagrammes de flux de données 21~~20~~
  - 6.2.1 Modification d'une page web hébergée sur S3 21~~21~~
  - 6.2.2 Modification du type d'affichage d'un objet IoT 22~~21~~

Chapitre 7: État du développement 23~~22~~

Recommandation 23~~23~~

Conclusion 25~~24~~

Bibliographie 26~~25~~

LISTE DES TABLEAUX

Tableau 1 Tarification AWS IoT Core 17
Tableau 2 Tarification Amazon S3 18
Tableau 3 Tarification AWS Lambda 18

LISTE DES FIGURES

Figure 1 Entrées des écrans d'affichage 12
Figure 2 Architecture AWS IoT 14
Figure 3 Diagramme publish-subscribe 15
Figure 4 Division Type 16
Figure 5 Kit de démarrage 19
Figure 6 Modification d'une page web 21
Figure 7 Modification du type d'affichage 21

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ARN	Amazon Resource Name
AWS	Amazon Web Services
IAM	Identity Access Management
IoT	Internet of Things
JSON	JavaScript Object Notation
Lambda	Service d'exécution de code sans serveur d'AWS.
MQTT	Message Queuing Telemetry Transport est le nom d'un protocole de communication.
OTG	USB On -The-Go
Participants	Joueurs ou visiteurs du Lan ETS
PC	Personal Computer Désigne autant une tour ou un laptop dans le contexte de ce document.
S3	Service de stockage d'AWS
SDK	Software Development Kit
UTF-8	Universal Coded Character Set Transformation Format - 8 bit

Introduction

Lors de la session d'automne 2018, le Lan ETS a reçu une commandite d'une trentaine d'écrans d'affichage numérique pour être utilisé lors de leurs évènements. Ces écrans allaient être déployés un peu partout dans le Palais des congrès de Montréal pour transmettre de l'information et diriger les visiteurs du Lan ETS 2019. En date de la validation du projet spécial, le Lan ETS ne possédait pas de système fonctionnel pour contrôler l'ensemble des écrans à distance et n'était pas intéressé à payer pour un tel système. Ce projet propose une solution à la nouvelle problématique de la gestion du contenu des écrans d'affichage numérique afin d'éviter des coûts récurrents pour une licence d'une solution professionnelle.

Chapitre 1: OBJECTIF

L'objectif de ce projet est l'élaboration et l'implémentation d'un système de gestion de contenu des écrans d'affichage numérique qui seront déployées au Lan ETS. Cet objectif a été divisé en trois facettes importantes qui devront être prises en compte lors de l'élaboration de la solution.

1.1 Gestion de l'affichage centralisé

Considérant le grand nombre d'écrans qui seront déployés lors de l'évènement, le système doit permettre de faire des modifications à l'affichage à partir d'un seul endroit. Il serait idéal que le système permette de regrouper les écrans dans des catégories d'affichage pour pouvoir faire une modification qui s'appliquerait à l'ensemble des écrans d'une catégorie.

1.2 Facilité de modification

En prenant en compte le nombre croissant de membres de l'équipe exécutive, de la grande différence de profil et d'expérience des membres du regroupement ainsi que la diversité des ordinateurs portables utilisés par ces mêmes membres; la solution doit être facile à utiliser pour un utilisateur ayant un minimum de connaissances, et ce, peu importe l'OS utilisé, par l'usager.

1.3 Persistant aux pannes réseau

Le réseau du Lan ETS fut extrêmement stable lors des dernières éditions. Ceci étant dit, il est impératif que l'affichage demeure fonctionnel en cas de panne du réseau. Pour être considérés comme étant fonctionnels en temps de panne, les écrans doivent continuer d'afficher leur contenu tel que prévu avant la panne et se mettre à jour, si nécessaire, une fois le réseau restauré.

Chapitre 2: Affichage numérique

Les écrans d'affichage numérique sont présents partout dans nos lieux publics. Que ce soit au restaurant, dans les transports en commun ou les centres commerciaux; leur nombre est en constante augmentation. Ces écrans sont normalement adressables et permettent d'afficher du texte, des images ou des animations pour transmettre de l'information, servir de divertissement ou pour faire de la publicité à des publics ciblés. De plus, grâce aux avancées technologiques, il est de plus en plus commun de pouvoir interagir avec un de ces panneaux par le biais d'un écran tactile, des capteurs ou en utilisant une application sur son téléphone intelligent. C'est en les côtoyant au quotidien que cette technologie est devenue usuelle et mieux comprise du grand public. C'est pour cette raison que le Lan ETS s'est initialement intéressé à une telle plateforme pour transmettre de l'information à ses participants. Le fait de pouvoir modifier de façon instantanée le contenu, la réduction du coût récurrent d'impression, la vaste gamme de médias supportés et la possibilité d'interaction avec le public n'étaient que des points positifs à comparer de l'affichage traditionnel. Ceci étant dit, l'utilisation de ces appareils avait en soi des restrictions qui sont présentées lors du prochain chapitre.

Chapitre 3: Restrictions

3.1 Requis réseautique

Afin de permettre de répondre à l'objectif 1.1, il a été déterminé que le système de gestion d'affichage puisse communiquer avec les écrans à partir d'un même réseau. Par conséquent, il a été convenu que les écrans devaient être connectés au réseau du Lan ETS par le réseau filaire ou par Wifi.

De plus, pour que ces appareils soient joignables à distance par le réseau, il sera nécessaire d'attribuer manuellement une adresse IP statique pour chaque appareil ou faisant le suivi des identifiants uniques de chaque appareil.

3.2 Requis matériel

Les écrans reçus en commandites sont vieux de 2010 et n'ont que deux entrées possibles provenant d'un PC (HDMI et VGA). Les PCs devront donc être compatible avec l'un de ces deux modèles d'entrée de données pour éviter l'achat de convertisseur.

Figure 1 Entrées des écrans d'affichage

Pour le Lan ETS 2019, il a été décidé d'utiliser de vieux PCs qui allaient alimenter les écrans avec les images nécessaires. Ceci étant dit, ces PCs n'étaient pas optimales, surtout lorsqu'on considère vouloir faire de la signalisation dans les grands axes de circulation. On se retrouve avec une tour au sol au milieu du corridor de sécurité ou

sous un écran qui est accroché au mur. Ces machines doivent absolument être câblées, puisqu'elles n'ont pas de carte réseau sans fil. Ceci ajoute le risque de faire trébucher les passants.

La solution optimale doit limiter le nombre de câble pouvant être dans le chemin, être compatible avec les entrées supportées par nos écrans d'affichage et pouvoir se dissimuler derrière l'écran ou utiliser le moins d'espace possible.

3.3 Type de média

Comme pour l'affichage numérique utilisé dans les lieux publics, notre affichage joue le rôle d'informer les visiteurs sur les activités qui auront lieu, l'endroit où ce situent certaines attractions, faire de la publicité pour nos partenaires ou présenter tout autre élément relatif à l'évènement. Il est possible d'atteindre cet objectif en affichant du texte, des images, des vidéos ou un amalgame de ces derniers.

La solution doit pouvoir supporter la manipulation et l'affichage de textes, d'images et de vidéos.

3.4 Budget disponible

Ce projet a vu le jour suivant une commandite d'écran d'affichage numérique industriel. Le fait de pouvoir utiliser ces équipements est une valeur ajoutée à notre évènement, mais n'apporte pas assez pour débloquer un grand budget qui pourrait autrement être utilisé ailleurs.

La solution doit être fonctionnelle, à faible coût et éviter les frais récurrents autant que possible.

Chapitre 4: AWS

Ce chapitre fait un survol des services offerts par AWS qui nous permettront d'atteindre nos objectifs sur le plan logiciel.

4.1 AWS IoT

Le service d'AWS IoT est l'un des plus récents services rendus disponibles. Il permet de communiquer avec des appareils tout en faisant le suivi de leur état. Voici une figure représentant l'architecture d'AWS IoT qui sera couverte plus en détail par la suite.

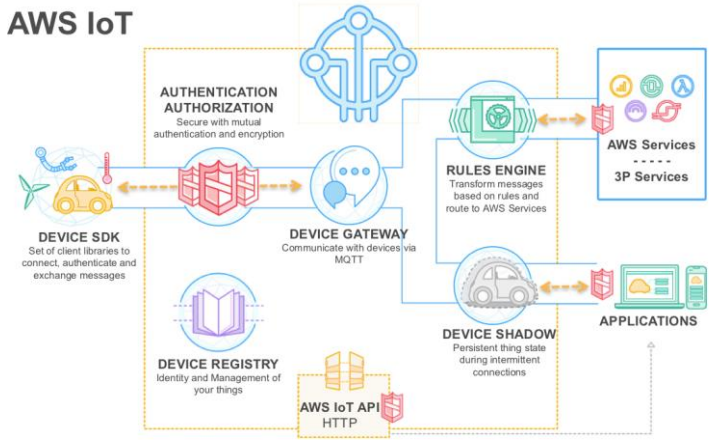

Figure 2 Architecture AWS IoT

4.1.1 Objets

4.1.1.1 Objet

Un objet dans AWS IoT est la représentation virtuelle d'un appareil physique, d'un capteur ou d'une entité logique. Chaque objet est identifiable par son nom (c.-à-d. « *thingName* »), possède un ARN unique et stocke l'ensemble de ses attributs dans le registre sous format JSON.

4.1.1.2 Groupe d'objets

Un groupe d'objet dans AWS IoT permet la gestion centralisée d'un ensemble d'objets de façon simultanée. En plus de contenir des objets, il est possible d'englober d'autres groupes d'objets afin de hiérarchiser ces ensembles, similairement au patron de conception d'héritage. Ceci facilitera la réussite de l'objectif 1.1 en permettant de modifier les attributs d'un groupe pour affecter toutes les entités qui y en descend.

4.1.1.3 Type

Un type est une balise attribuée à un objet qui permet de classer ou identifier rapidement un objet. Cette balise prend la forme d'un ensemble clé et valeur et joue le rôle de métadonnées personnalisées. Les clés et valeurs doivent être composées de caractères Unicode en UTF-8. Un nombre maximal de 50 balises peuvent être attribuées à un même objet ou à un même groupe. Cet attribut permettra de désigner la catégorie d'affichage d'un objet ou d'un groupe.

4.1.2 Agent de messages

La plateforme AWS IoT utilise le protocole MQTT pour procéder à toutes ses communications de façon sécurisée. MQTT est idéal pour des applications IoT pour deux raisons: sa légèreté en perte de batterie et en utilisation de bande passante ainsi que le fait qu'il suit le principe de publication et d'abonnement.

D'abord, ce protocole a été conçu pour répondre au besoin de communiquer avec des appareils éloignés ayant accès à un réseau dont la bande passante est limitée. Son faible coût en données de communication pour garder la connexion avec le serveur actif réduit significativement sa consommation en énergie.

Ensuite, MQTT suit le patron de conception *publish-subscribe*. Tel qu'illustré à la *figure 3* ci-dessous, l'éditeur envoie son message au « topic » sans se soucier des destinataires. Les destinataires quant à eux, s'abonnent au « topic » et ne recevront que les messages de celui-ci. Ceci nous permet de publier des modifications sans avoir à tenir une liste des destinataires à jour, mais simplement de s'assurer que chaque destinataire est abonné au bon « topic ».

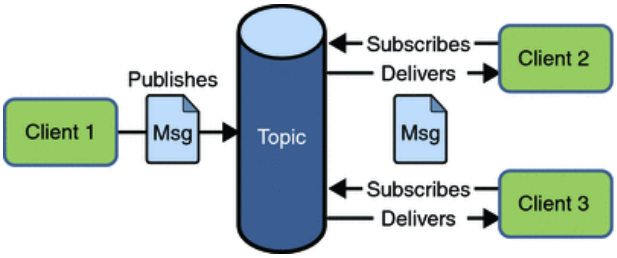

Figure 3 Diagramme *publish-subscribe*

Formatted: Font: Not Bold

4.1.3 Authentification et autorisation

En utilisant AWS IoT, les services de sécurité et d'identité inclus dans AWS sont non seulement disponible, mais requis pour passer par l'agent de messages ou par les autres services d'AWS. Ceci implique que chaque objet soit identifié dans le registre avec les permissions et certificats associés avant de pouvoir aller plus loin. Au besoin, il est possible de mettre en place des mécanismes d'autorisation personnalisée par le biais d'un tiers externe (c.-à-d. OAuth).

4.1.4 Shadow d'un objet

Le *Shadow* d'un objet représente l'état actualisé d'un objet stocké dans le Cloud AWS. Lorsque l'état d'un objet change, son *Shadow* est mis à jour. En cas de perte de connexion, une synchronisation sera effectuée lors du rétablissement de service. La manipulation du *Shadow* se fait par le biais d'un API public unique pour chaque objet. C'est par ce service que nous réussirons à répondre à l'objectif 1.3.

4.2 AWS S3

Le service de stockage d'Amazon S3 permet de stocker des données dans le Nuage. Dans notre cas, nous allons l'utiliser pour héberger des sites web à page unique. Chaque site web représentera un type d'affichage. Pour permettre de modifier un affichage de façon simple et rapide, la disposition sera obtenue grâce aux fichiers HTML et CSS et les animations seront gérées par des scripts en JavaScript.

Par exemple, voici la disposition du contenu de l'affichage des commanditaires

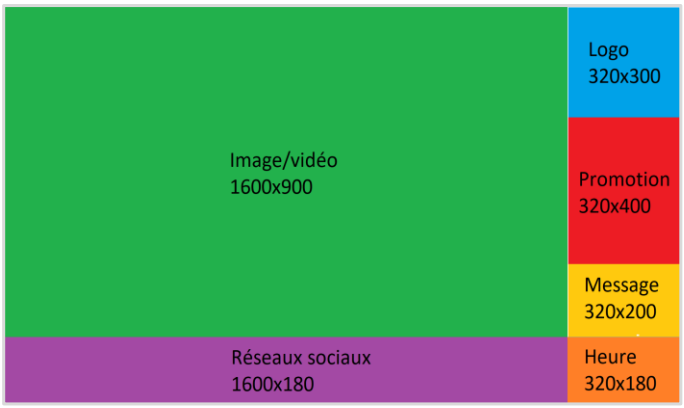

Figure 4 Division Type

4.3 AWS Lambda

Le service d'AWS Lambda permet d'exécuter du code sans serveur. La configuration requiert le téléversement du code à exécuter ainsi que l'identification des éléments déclencheurs pour que le service s'exécute automatiquement.

4.4 Tarification

La majorité des services d'AWS possède un palier d'utilisation gratuit. C'est le cas pour IoT, S3 et Lambda. Pour le calcul de tarification de ces trois services, nous considérons un maximum de 5760 minutes de connexion, soit du jeudi midi au lundi midi, par appareil et un total de 30 appareils. De plus, la tarification prend en compte l'exécution des services dans la zone É.-U. Est (Virginie du Nord).

Tableau 1 Tarification AWS IoT Core

	Utilisation estimée	**Palier gratuit**	**Coût en cas de dépassement**
Frais de connectivité	172 800 min (5760min*30 appareils)	2 250 000 min	0.08 USD/1 000 000 min
Frais de messagerie	39 000 messages (325 messages/appareil/jour * 4 jours * 30 appareils)	500 000 messages	1.00 USD/1 000 000 messages
Frais associés aux shadows d'appareil et au registre	12 000 requêtes (100 requêtes/jour * 4 jours * 30 appareils)	225 000 requêtes	1.25 USD/1 000 000 requêtes
Nombre d'appareils	30	50	*Varie selon la connectivité*

Tableau 2 Tarification Amazon S3

	Utilisation estimée	Palier gratuit	Coût en cas de dépassement
Stockage	72MB (12MB/type d'affichage * 6 types d'affichage)	5 Go	0.023 USD/Go
Nombre de requêtes/mois	1200 requêtes	20 000 requêtes	0.005 USD/1000 requêtes

Tableau 3 Tarification AWS Lambda

	Utilisation estimée	Palier gratuit	Coût en cas de dépassement
Nombre de requêtes/mois	400 requêtes (100 requêtes/jour * 4 jours)	1 000 000 requêtes	0.20 USD/1 000 000 requêtes

La restriction 3.4 est respectée puisque nous réussissons à demeurer à l'intérieur des limites d'utilisation gratuite.

Chapitre 5: Raspberry Pi

Ce chapitre explique le processus de sélection de la solution sur le plan du matériel.

Afin de pouvoir respecter les restrictions du matériel, de médias et de budget, il fallait trouver un appareil qui avait une sortie HDMI ou VGA, qui supportait des images, du texte ou des vidéos en plus d'être peu coûteux. Pour faire suite à la décision du logiciel choisi, l'adoption d'un Raspberry Pi pour compléter notre projet IoT était l'option de prédilection.

Le modèle sélectionné fut le Raspberry Pi Zero W. Le modèle Zéro de Raspberry Pi est le plus nano-ordinateur monocarte disponible sur le marché à ce jour. Le kit de démarrage, en vente pour $34.95 CAD en ligne, comprend le Pi, un mini HDMI et un OTG. Une autre caractéristique importante pour ce projet est le fait que le modèle Zéro se fait aussi en W (Wireless). Par conséquent, ce Pi peut se connecter par Wifi ou Bluetooth dès sa première utilisation, sans avoir à ajouter une carte réseau supplémentaire. Enfin, le kit inclut une carte MicroSD de 8 Gb qui est suffisant pour accueillir l'OS Raspbian.

Figure 5 Kit de démarrage

Avec ce nano-ordinateur sous la main, les trois restrictions de compatibilité, de médias et de coût sont respectées, en plus de permettre d'éliminer le câble Ethernet tout en restant connecté au réseau sans fil du Lan ETS.

Chapitre 6: Intégration Raspberry Pi et AWS

6.1 Installation des environnements

6.1.1 AWS

La première étape dans AWS est de créer un usager dans IAM avec les permissions requises pour exécuter les tâches des trois services utilisés. Ce sont les clés d'accès de cet usager qui seront utilisées plus tard dans le Raspberry Pi. Par la suite, il faut créer les objets dans IoT et leur attribuer un type d'affichage et téléverser les pages web dans des bucket S3 dont l'option d'hébergement de site web statique est activée. Il ne reste plus qu'à créer les deux fonctions Lambda qui se déclenchent respectivement lors d'une modification d'une page web hébergée sur un des bucket S3 et lors d'une modification du type d'affichage d'un objet IoT. Il est nécessaire de téléverser le script associé au transfert des données pour chacune de ces situations dans ces fonctions et d'identifier les éléments déclencheurs.

6.1.2 Raspberry Pi

La préparation du Pi commence par l'installation de Raspbian sur la carte MicroSD. La prochaine étape consiste à configurer le Wifi pour que le Pi ait accès à Internet. Ensuite, il faut installer le SDK de AWS IoT disponible sur GitHub. Une fois le SDK bien installé, il faut ajouter les certificats, les clés d'accès au projet et le script d'affichage au projet. Enfin, il ne reste plus qu'à installer les librairies requises par l'environnement.

6.2 Séquence d'exécution

Dans notre contexte, une fonction Lambda serait déclenchée seulement lorsqu'un changement serait détecté sur un des sites Web hébergés sur S3 où si un objet dans IoT voit son type d'affichage changer. Les deux cas d'utilisation sont illustrés ci-dessous:

6.2.1 Modification d'une page web hébergée sur S3

Figure 6 Modification d'une page web

6.2.2 Modification du type d'affichage d'un objet IoT

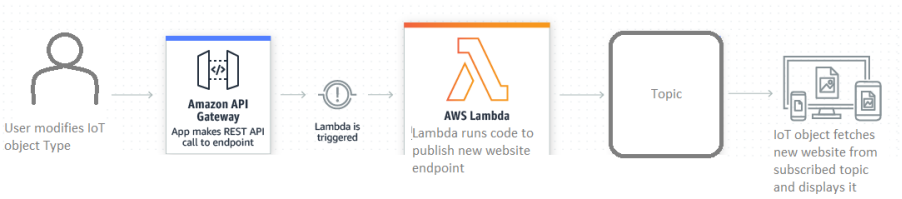

Figure 7 Modification du type d'affichage

Chapitre 7: État du développement

En date de remise du projet, les fonctions Lambda sont fonctionnelles et permettent aux données de compléter la séquence complète telle qu'espérée. Un objet IoT est présentement capable de passer d'un type d'affichage à un autre et le script présent sur le Pi réussit à ouvrir le site web. Toutefois, le navigateur ne s'ouvre pas en mode kiosque et le script n'est pas exécuté automatiquement lors de l'ouverture du Pi. De plus, la modification d'un groupe d'objets n'a pas été testée.

Recommandation

Un système de gestion de contenu du domaine du logiciel libre a été déployé quelques jours avant l'édition du Lan ETS 2019. Son utilisation semble exiger moins de configuration et demeure, à la base, un système développé exactement pour les besoins de ce projet. Avant d'aller de l'avant avec l'implantation à grande échelle du système de gestion proposé dans ce document, je comparerais les deux systèmes en utilisant les mêmes restrictions pour évaluer lequel répond mieux aux objectifs. De plus, je validerais le budget disponible pour réaliser quelconque de ces projets.

Conclusion

En somme, la solution proposée permet de gérer l'affichage des écrans de façon centralisée sur la plateforme AWS. La sélection du type d'affichage est facile à modifier puisque la configuration est simplement spécifiée dans un objet JSON et la disposition des éléments à l'écran est facilement modifiable en éditant le fichier HTML ou CSS. La solution est persistante aux pannes du réseau grâce à la fonctionnalité de *Shadow* d'AWS IoT et les coûts reliés à l'implantation de cette solution demeurent réalistes selon l'état des finances pour l'édition 2020 du Lan ETS. La documentation et les scripts seront remis aux membres du Lan ETS comme référence pour l'implantation de futurs projets IoT.

Bibliographie

Raspberry Pi, Spécification Raspberry Pi Zero W
Internet, consulté le 13 février 2019.
https://www.raspberrypi.org/products/raspberry-pi-zero-w/

AWS IoT Documentation, API Reference
Internet, consulté le 15 février 2019.
https://docs.aws.amazon.com/fr_fr/iot/latest/apireference/Welcome.html

AWS IoT Developer Guide, Developer Guide
Internet, consulté le 16 février 2019.
https://www.amazon.ca/AWS-IoT-Amazon-Web-Services-ebook/dp/B07JBRCWWZ/ref=sr_1_1?s=digital-text&ie=UTF8&qid=1548256387&sr=1-1&keywords=AWS%20IoT%3A%20Developer%20Guide

MQTT, MQTT Documentation
Internet, consulté le 12 mars 2019.
http://mqtt.org/

GitHub, AWS IoT Device SDK for Python
Internet, consulté le 20 mars 2019.
https://github.com/aws/aws-iot-device-sdk-python

AWS IoT, Architecture AWS IoT
Internet, consulté le 10 avril 2019.
https://m.media-amazon.com/images/G/01/DeveloperBlogs/AmazonDeveloperBlogs/legacy/AWS_IoT23._CB520207442_.png

Oracle, Publish/Subscribe Messaging Domain
Internet, consulté le 20 avril 2019.
https://docs.oracle.com/cd/E19575-01/819-3669/bnced/index.html

Canakit, Pi Zero Wireless Starter Kit
Internet, consulté le 20 avril 2019.
https://www.canakit.com/raspberry-pi-zero-wireless.html?src=raspberrypi

AWS IoT Core, Tarification AWS IoT Core

Internet, consulté le 21 avril 2019.
https://aws.amazon.com/fr/iot-core/pricing/

Amazon S3, Tarification Amazon S3
Internet, consulté le 21 avril 2019.
https://aws.amazon.com/fr/s3/pricing/

AWS Lambda, Tarification AWS Lambda
Internet, consulté le 21 avril 2019.
https://aws.amazon.com/fr/lambda/pricing/