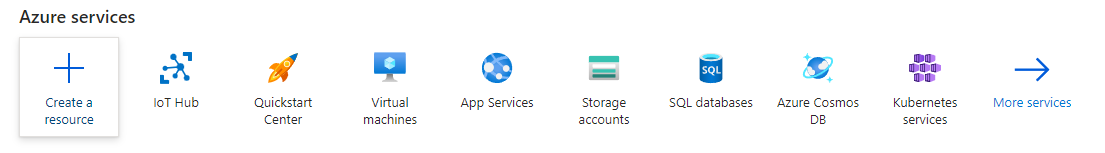
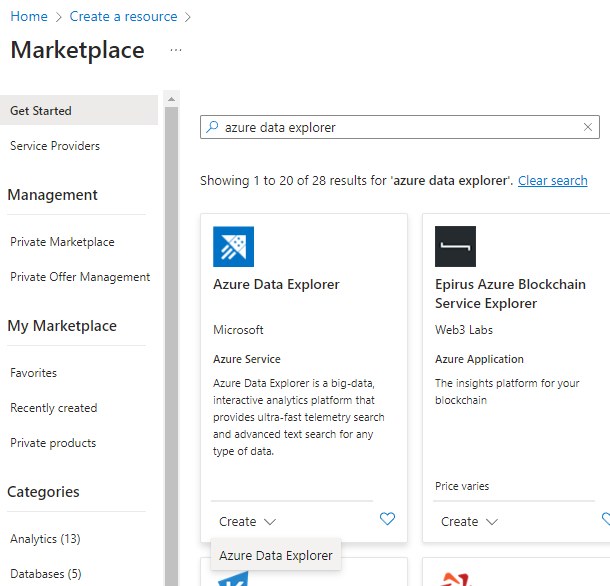
Rapport TP2

Partie 1 – Azure

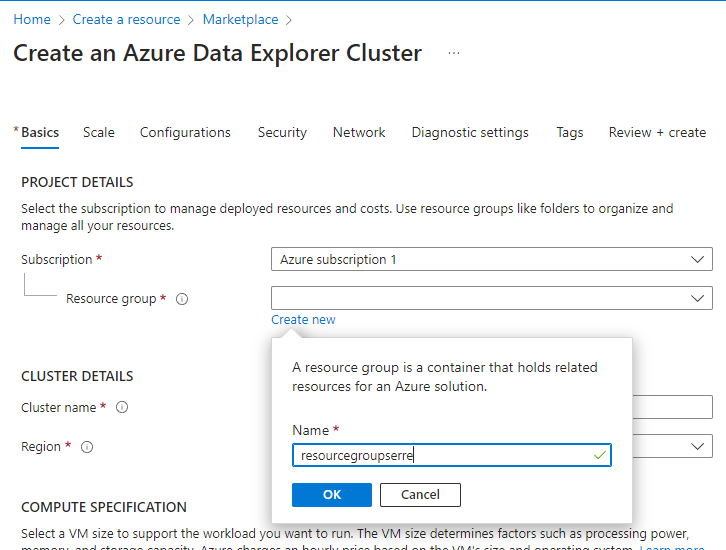
1. Création d’un Cluster
   1. Tout d’abord, il a fallu créer un Cluster en allant à la [page principale du portail Azure](https://portal.azure.com/#home) et en cliquant sur “Create a ressource”.



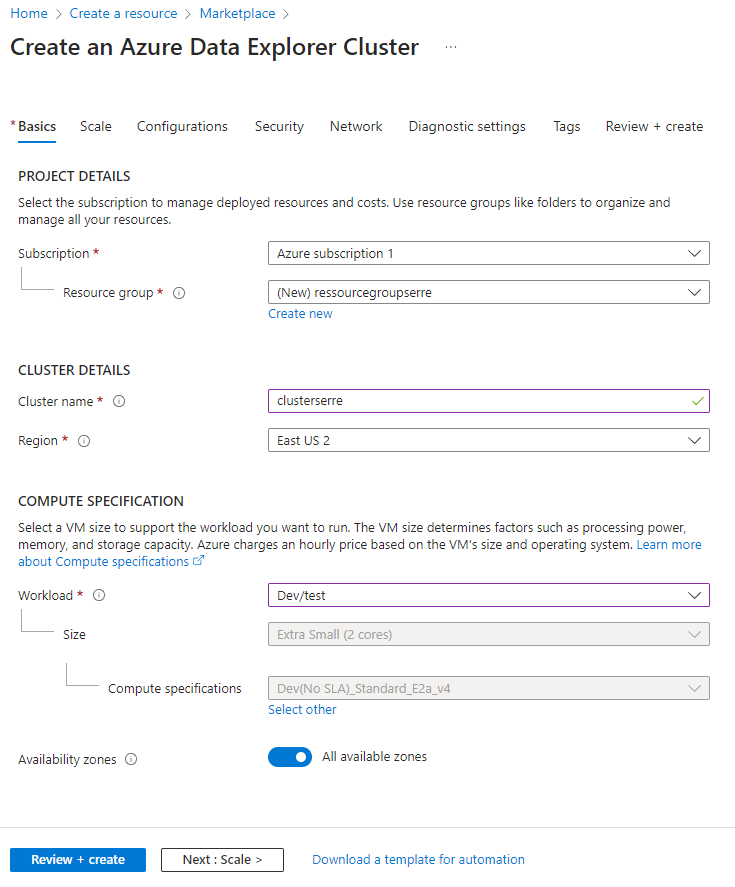
* 1. Ensuite j'ai cherché pour “azure data explorer” dans la barre de recherche. Ensuite j'ai mis mon curseur sur “Create” pour ensuite sélectionner “Azure Data Explorer”.



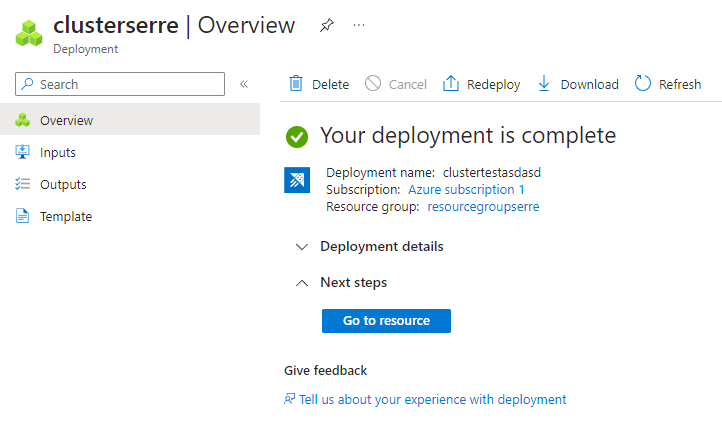
* 1. J’ai créé un “Resource group” au nom de “resourcegroupserre” pour regrouper toutes les composantes nécessaires au fonctionnement du systèmes de télémétrie.



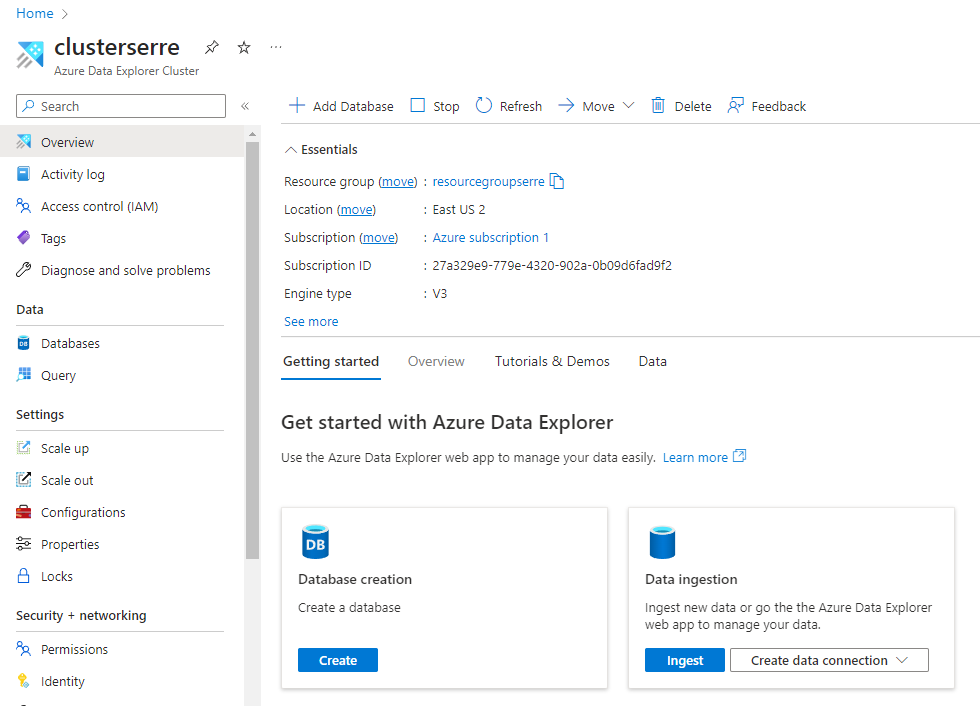
* 1. J’ai continué avec la création du “Azure Data Explorer Cluster” en lui donnant “clusterserre” comme nom. Je choisis aussi la valeur “East US 2” pour l’option “Region” et “Dev/test” pour “Workload”.



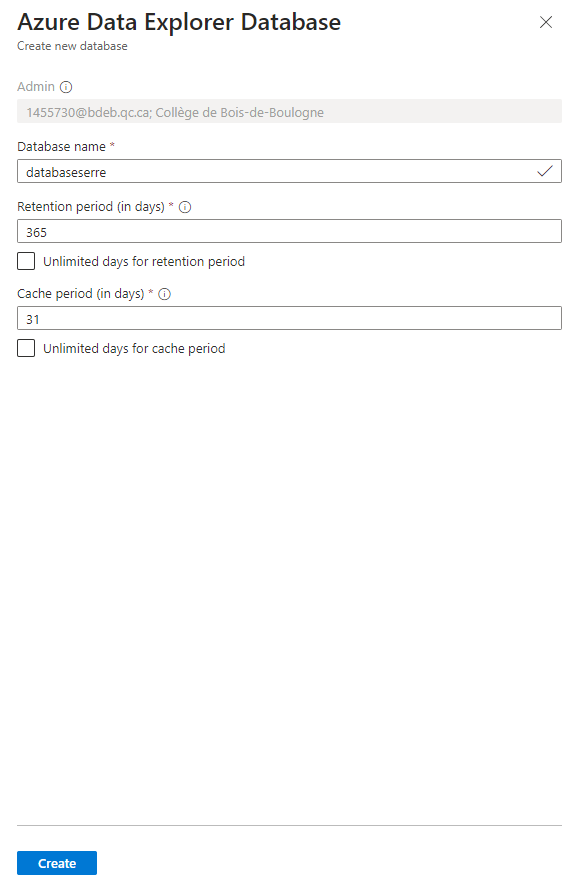
* 1. Une fois présenté avec la page suivante, j’ai cliqué sur “Go to resource”.



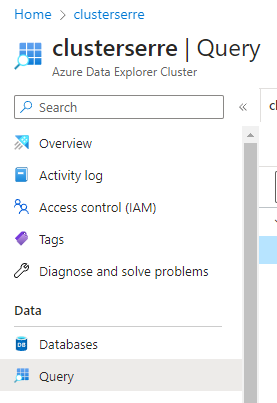
1. Création d'une base de données
   1. Une fois sur la page appartenant au Cluster, j’ai cliqué sur “Create” dans la case “Database creation”.



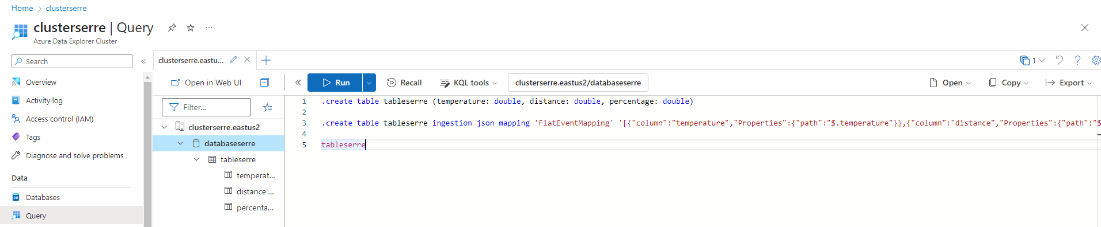
* 1. J’ai donné “databaseserre” comme nom à ma base de données, 365 jours de rétention de données et une période de 31 jours pour la cache. J’ai terminé en pesant sur “Create”.



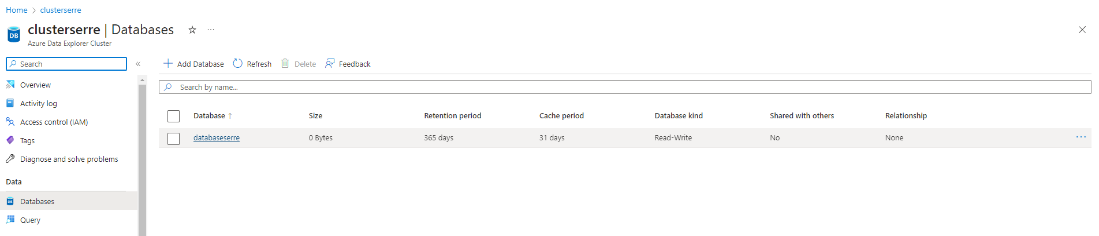
* 1. Une fois la base de données créer. Je suis allé dans la section “Query” dans le menu à gauche pour pouvoir entrer les commandes nécessaires pour créer une table, ses colonnes et son mapping.



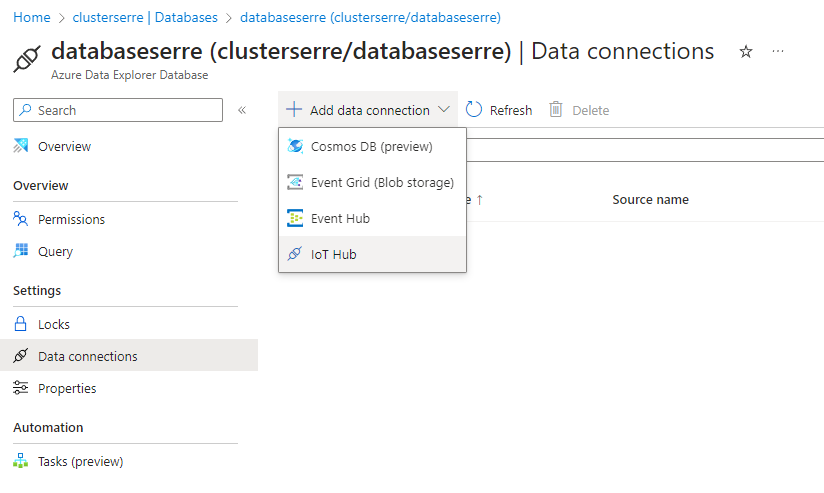
* 1. Une fois les commandes roulées, tous les éléments nécessaires au fonctionnement voulu sont créés dans la base de données. Un fichier \*.kql sera jumeler dans un zip avec ce document contenant les commandes.



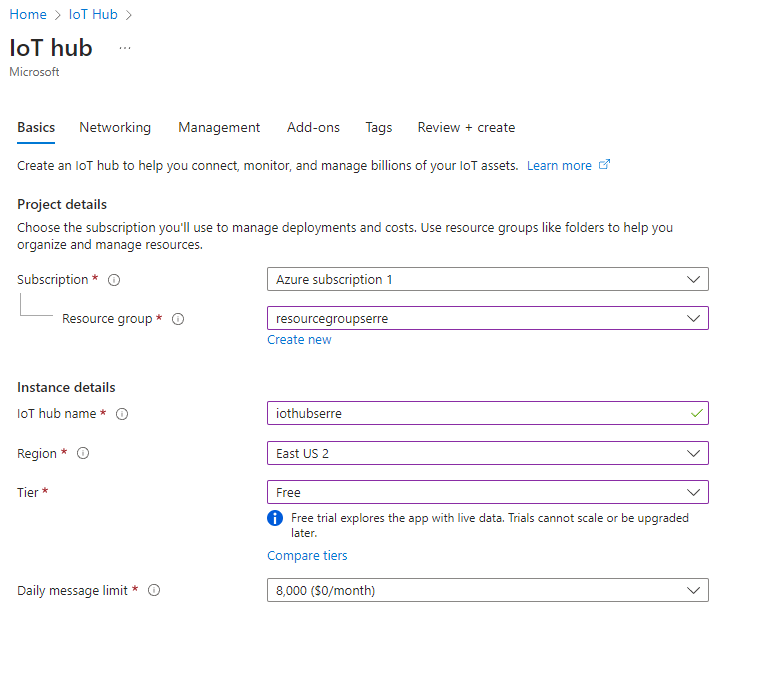
1. Création d’une connexion de données IoT Hub
   1. Une fois que toutes les commandes ont été roule avec succès, je suis allé dans la section “Databases” dans le menu à gauche et j’ai cliqué sur ma base de données (databaseserre).



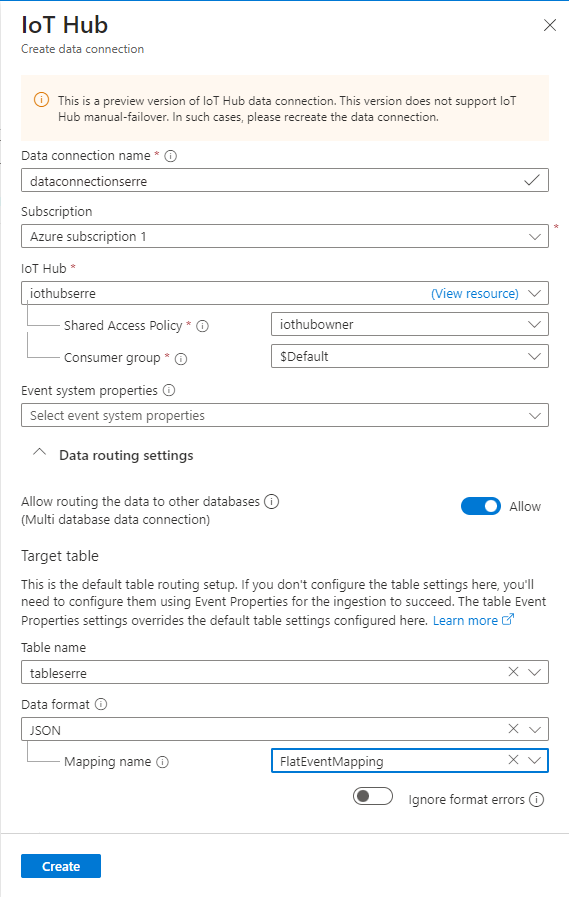
* 1. Encore une fois je vais dans le menu à gauche pour sélectionner la section “Data connections”. Une fois dans la nouvelle page, je rajoute une connexion a un IoT hub que je vais créer en même temps à l’aide du bouton “Add data connection” et “IoT Hub”.



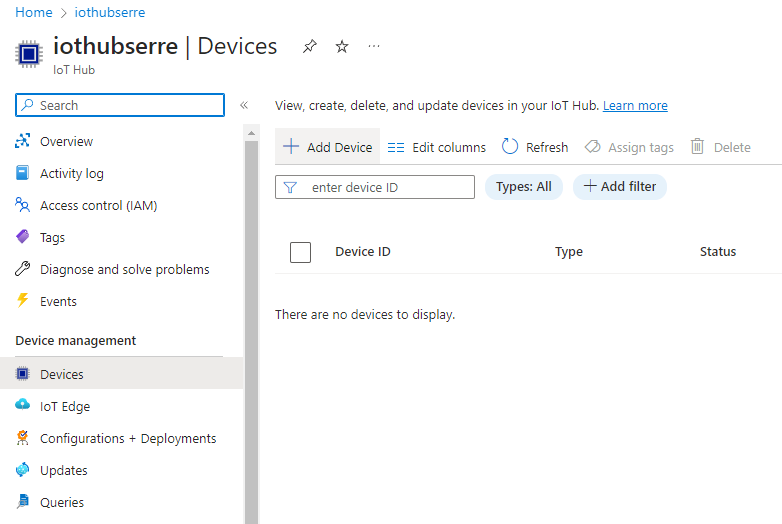
* 1. Dans la page de creation d’un IoT Hub je m’assure encore une fois que je l’associe avec le “Resource group” correct. Je lui donne le nom de “iothubserre” et je sélectionne la même valeur pour “Region” que dans les autres étapes (East US 2). Je choisis aussi la valeur “Free” pour l’option “Tier”.



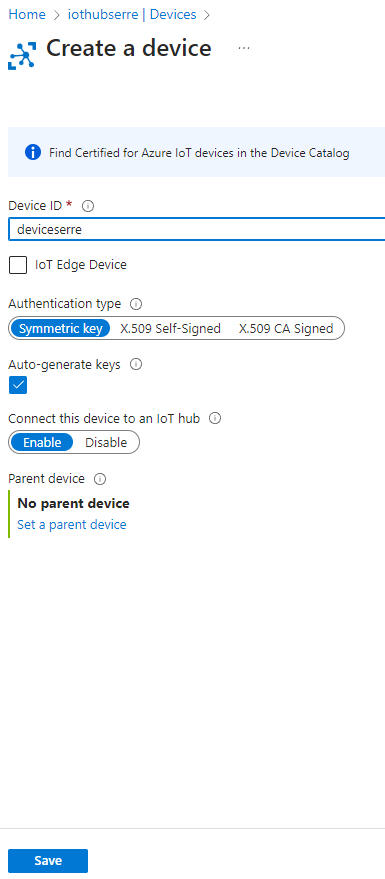
* 1. Une fois l’IoT Hub créer, il faut créer un “Data connection”. Je lui donne le nom de “dataconnectionserre” et j'entre le nom de mon IoT Hub que je viens de créer (iothubserre) dans le champ “IoT Hub”. Je choisis aussi la valeur “iothubowner” pour l’option “Shared Access Policy” et “$Default” pour “Consumer group”. Ensuite j’ai activé l’option “Allow routing the data to other databases” en m’assurant que l’état de l’interrupteur associer était à “Allow”. Finalement, j’ai inscrit le nom de ma table (tableserre) dans le champ “Table name”, choisis le format “JSON” sous “Data format” et choisis le mapping que j’ai créé avec mes commandes KQL (FlatEventMapping) dans le champ “Mapping name”.



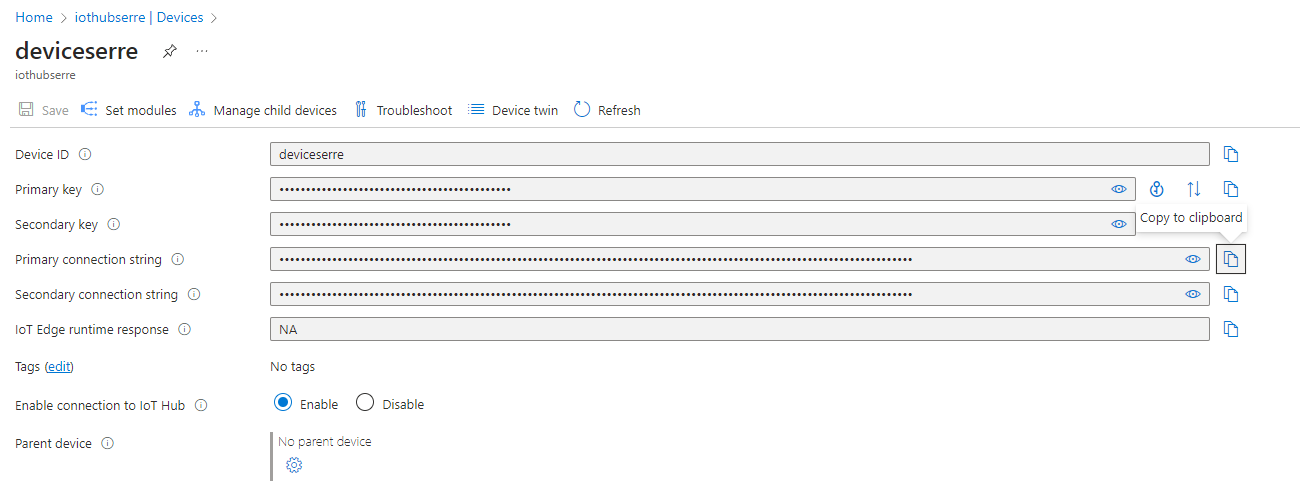
1. Création d’un appareil
   1. Je suis retourné dans la page de mon IoT Hub (iothubserre) pour créer un appareil. Pour faire ceci il a fallu cliquer sur “Devices” dans le menu à gauche et ensuite sur le bouton “Add Device” dans la barre d’options.



* 1. Dans l’onglet de creation d’appareil j’entre le nom voulu qui est “deviceserre”, je m’assure que “Authentication type” est sur “Symmetric key”, le case à cocher associé à “Auto-generate kets” est coché et que “Connect this device to an IoT hub” est sur “Enable”.



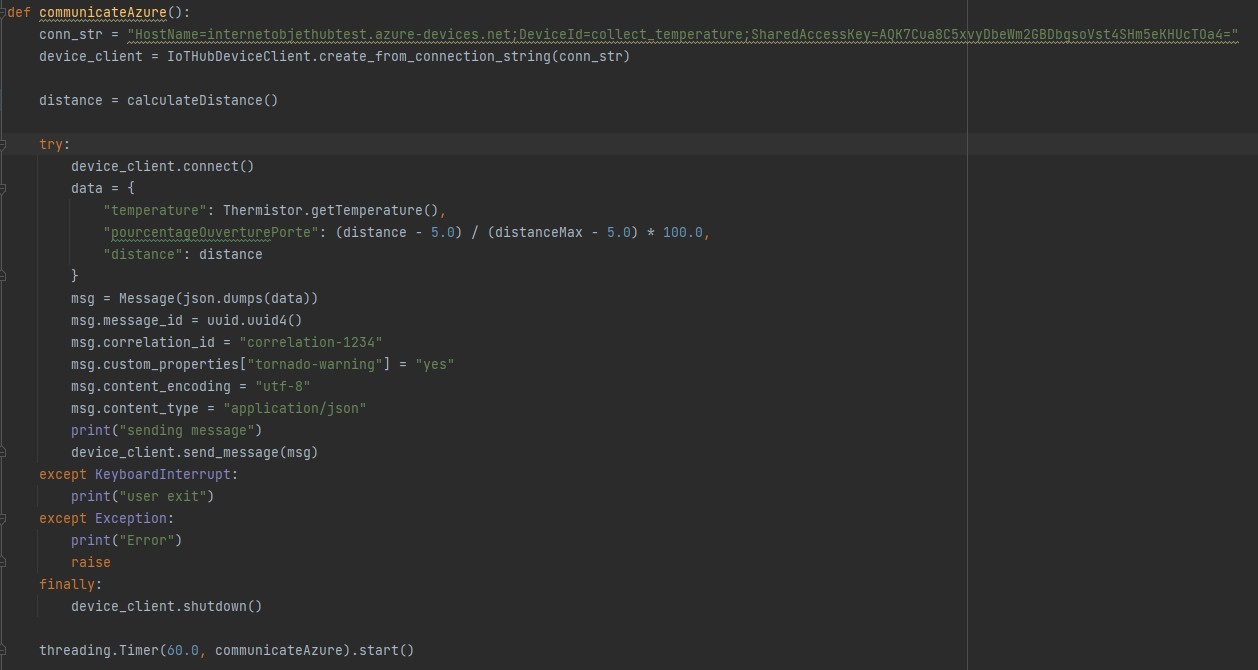
* 1. Une fois l’appareil créer, j’entre dedans en cliquant sur son nom dans la page des appareil (étape 4.a). Je prends en note le “Primary connection string” pour l’utiliser dans mon code. Au niveau de Azure, le travail est maintenant terminé.



1. Vérifier des données entrantes
   1. Pour vérifier si les données entrent et qu’elles sont correct, on peut retourner à la page “Query” de la même manière qu'indiquer à l’étape 2.d et rouler la dernière commande du fichier \*.kql. Cette commande affichera toutes les donnes dans notre table “tableserre”.

Partie 2 – Intégration dans le code

Dans le code, nous implémentons la méthode suivante (il se peut que certains noms de variables soient différents de ceux du code actuel) :



Lorsque cette méthode est appelée, elle exécute une première fois son code, puis lance un thread qui exécutera la méthode à chaque 60 secondes. La base du code de cette méthode est inspirée de la documentation donnée par notre enseignant.