**PROJET DE DATA ENGINEERING : STREAMING ET ANALYSE DE DONNEES REELLES AVEC DATA AZURE**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Ie PARTIE : Notions à Maîtriser (Rappel)**

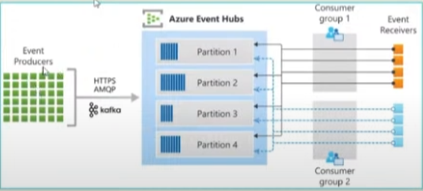
**Micro batching :** Nous extrayons les données en un instant donné

**IIe PARTIE : A PROPOS DU PROJET**

**Ressources à utiliser :**

**Event Hub :** conserve la donnée pendant 1 jour,

**Azure stream analytics :** c’est une ressource qui sert à réaliser un traitement de données sur les donnnées à temps réel. On peut réaliser du SQL dessus pour traiter la donnée en temps réel.



**Producteurs de données** : un script python qui va aller récupérer de la donnée depuis un API par exemple pour les pusher vers Stream analytics qui va jouer le rôle du **consummer** pour analyser ces données en temps réel

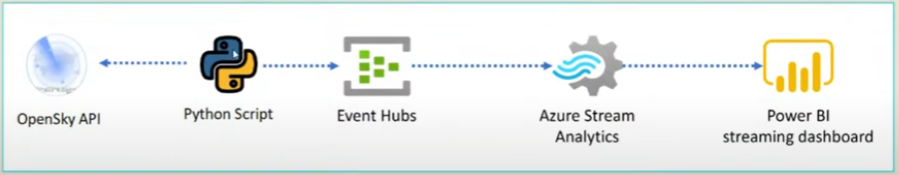


Ensuite à partir du Event hub, on va configurer des tockens pour relier python à notre **event hub, « Shared access policies »**

**Source de données :**

On va récupérer de la donnée depuis un site internet « **opensky-network** » qui est un API open source qui donne les informations sur le vol en temps réels

**ARCHITECTURE DE NOTRE PIPELINE DE DONNEES (PROJET)**

****

**Un script python** qui va récupérer les données **depuis l’API** et **un script python** qui va nous connecter à notre **event hub** pour envoyer à intervalle de temps régulier toute la donné recuperée. Ensuite ces données seront analysées, traitées via **Azure stream Analytics** pour être visualisé **sur Power BI.**

**ETAPES DE REALISATION DU PROJET**

**Etape 1 : Un script Python (Open\_sky.py) pour récupérer les données depuis l’API**

**Module à installer :**

-Librairies : request (pour faire des requêtes avec Python)

**Etape 2 : Un script Python comme Producer (send.py) pour envoyer les donnéesvers Event Hub**

Pour le consumer (send.py), nous avons besoin d’installer le module :

**-Azure.eventhub**

**-pour le module open\_sky. Il faut donc cloner le API depuis son depot git :**

**Tu te places dans le dossier ou se trouve les fichiers python**

**1.Cloner le dépôt**

**git clone** [**https://github.com/openskynetwork/opensky-api.git**](https://github.com/openskynetwork/opensky-api.git)

**2. Naviguer dans le répertoire cloné :**

**cd opensky-api/python**

**3.Installer les dépendances :**

Le nom de votre hub créé

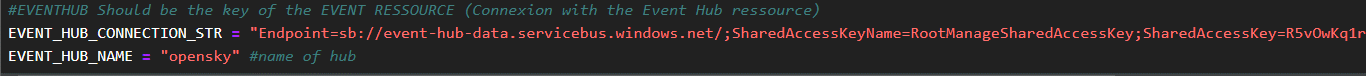
**pip install requests**

* **Configuration de Event HUB**

1-) On va ajouter **un Event hub (opensky).**

**On peut donc avoir plusieurs hubs selon les besoins dans ma ressource Event hub.** Chaque hub a un besoin particulier.

2-) La connexion avec python se fait avec **Stratégies d’accès partagé---RootManageSharedAccessKey**



**Lien microsoft pour comprendre la synchronisation :** [azure.eventhub.aio.EventHubProducerClient class | Microsoft Learn](https://learn.microsoft.com/en-us/python/api/azure-eventhub/azure.eventhub.aio.eventhubproducerclient?view=azure-python#azure-eventhub-aio-eventhubproducerclient-send-batch)

La clé que vous recuperez depuis RootManageSharedAccessKey

Le rôle du **producer (send.py )** est d’envoyer de la donnée à intervalle de temps régulier sur **event hub**.

Donc une fois la configuration terminée :aller dans invite de commade exécuter dans votre répertoire source : **python send.py 10 1** (duration, fréquence)

Donc ça veut dire que nous l’exécutons pendant 10min avec une fréquence de 1 min par exécution.

**Etape 3 : Configuration du Stream-Analityque job (consummer)**

**Le stream analytic** sert à la transformation et au traitement de la donnée **en temps réel. Il va donc consommer les données** de **eventhub** et appliquer une transformation.

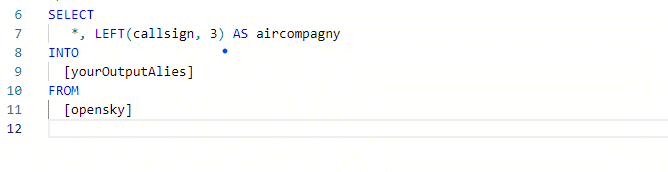
Les composantes importantes du **stream Analytics sont:**

**1)Input (eventhub,…)**

Aller dans topologie de la tâche--Entrées

**Pour la stream Analytic** on a pas besoin de clés (tokken) pour connecter les services, azure utilise les identités managers qui est un service tiers de azure.

2)La partie **Query** (**ou requête**) Pour visualiser **nos données** provenant de notre **eventhub, traiter** avant de visualiser vers Power BI.



From (le nom de ton alias (stream analytic)

**Notion à lire : utilisation des tables statistiques (de reference) en SQL**

**3)OUTPUT : ou est ce que nos données seront envoyées (Power BI,…)**

**Alors vous enregistrer votre sortie sous format que vous voulez. Vous enregistrez vos requetes > Ensuite vous allez dans vue d’ensemble (Overview)> puis vous lancer (commencer le travail pour analyser les données en temps réel )**