Para la conexión con azure desde un computador con Linux descargado (Ubuntu 16.04 LTS), se utiliza el SDK para Python que se recomienda en a pagina de Azure: <https://github.com/azure/azure-iot-sdk-python>. De este SDK se utilizó con éxito los ejemplos para la creación, eliminación, y consulta de dispositivos en un IoTHub previamente creado, siguiendo los pasos de instalación y ejecución que se dan el github.

Sin embargo, en los ejemplos para enviar mensajes a dispositivos creados en el IoTHub se presentan varios errores. De esta manera procedimos a intentar diferentes soluciones, la primera de ellas crear el dispositivo con la herramienta Device Explorer, desde un computador con Windows, y la segunda de ellas crear el dispositivo con los ejemplos del SDK.

En ambos casos se pudo crear con éxito el dispositivo sin el uso del portal de Azure, como se observa en la Imagen 1. No obstante, al intentar enviar un mensaje a manera de prueba (el que viene predeterminado en el ejemplo del SDK), en ambos casos se nos presentó el mismo error que se observa en la Imagen 2. De alguna manera, aún en cuando en el portal de Azure aparecen los dispositivos creados, el código dice no encontrarlos y por ello no puede crear el IoTHandle para enviar el mensaje.

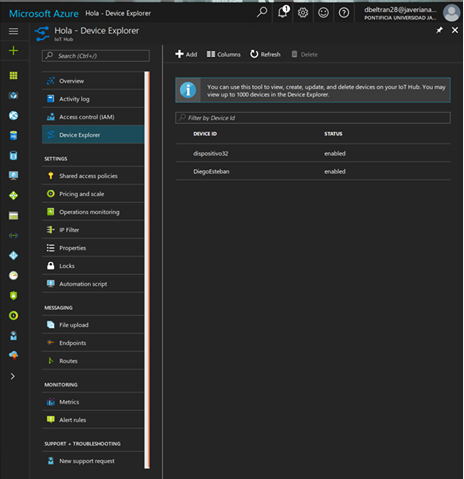


Imagen 1, Portal de Azure con los dispositivos creados y habilitados

De esta manera, accedimos al código del ejemplo y a aquellos que emitían el error en el terminal y tras analizarlo percibimos que el error se encuentra en la función **iothub\_client\_init** del archivo **iothub\_client\_sample.py** como se ve en la Imagen 3. Este archivo, según entendemos de los errores, llama a un archivo llamado **iothub\_client\_ll.c** que hace parte del SDK de C que viene instalado en el SDK de Python. En este archivo, en la linea 354 se puede ver que el error viene de que se esta obteniendo un valor de NULL en el valor del deviceId y por esta razón no lo puede localizar. No hemos podido solucionar este problema, y al correr el comando de “HELP” del archivo, solo nos muestra la manera adecuada de correr el ejemplo pero no nos presenta otras opciones como se observa de igual manera en la imagen 2.



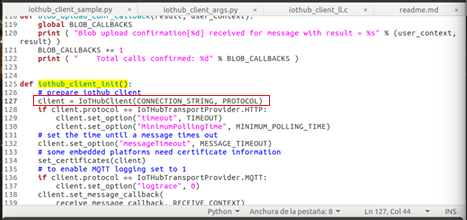
Imagen 2, Ejecución del código junto con los errores obtenidos y la ejecución del comando help.

Imagen 3, Función en la que se presenta el error al ejecutar el código.