Cracking API Integration in Power BI: 3 Easy Solutions to Tackle Authentication and Refresh Hurdles

Scheduling refreshes in Power bi services has two main challenges: hosting all your data on the cloud and ensuring connectivity when using Python scripts or pulling data from VPN secured data bases like Redshift. You'll need an AWS instance or a VNet on Azure to use as a bridge to manage these connections. Additionally, Power BI lacks native support for Authorization headers, making Python a valuable tool for seamless data integration without custom connectors.
If you opt for the Python route, consider that it requires a Personal Gateway or an AWS instance/VNet to handle Redshift databases and secure data retrieval. While M code or simple web connections are more cost-effective, the Python option offers flexibility and security at a higher cost. This guide presents tailored solutions to fit various scenarios and budgets, helping you navigate these challenges efficiently!

Find all the code on my repository:
https://github.com/tomasdevelopment/Automation-Flows-with-Python-/tree/main/Power%20Bi%20%2B%20Python/Api%20Request%20with%20Bearer%20Token%20from%20Power%20Bi

Option 1: Using a Python Scrypt with Pandas and Requests and hosting your refresh on aws instance or azure vnet

GET DATA-PYTHON SCRYPT

1) PYTHON SCRIPT INTEGRATION:

Utilize Python scripts within Power BI to handle API requests, including Bearer token authentication for secure data access. This approach offers flexibility in data processing, but it's crucial to be aware of Power BI's limitations: Python scripts are capped at 250,000 rows and have a maximum execution time of 30 minutes. Additionally, if your data source requires secure connections, you'll need to set up a Virtual Network (VNet) on Azure or use a Personal Gateway. For cloud-hosted environments, deploying a Personal Gateway on an AWS EC2 instance, such as a T3 or T3a instance type, provides a virtual machine capable of handling these tasks, with the choice depending on your performance and cost requirements.

YOU WILL NEED:

Python Scripting + Vnet in Azure or AWS Instance

Option 2: Using a Power Query M Code Scrypt

2) M CODE APPROACH:

Leverage M code to transform JSON responses into structured tables. This method involves using the Json.Document(Web.Contents()) function, allowing for a straightforward integration of API data without the need for advanced scripting. It's a powerful alternative when dealing with nested data structures.

MCode Repository:
https://github.com/tomasdevelopment/Automation-Flows-with-Python-/blob/main/Power%20Bi%20%2B%20Python/Api%20Request%20with%20Bearer%20Token%20from%20Power%20Bi/Api%20Request%20with%20Bearer%20token%20using%20Mcode

YOU WILL NEED:

Power BI and AWS instance only of your db is VPN secured.

GET DATA - FROM BLANK QUERY

Option 3: Using get from Web Method

3)GET FROM WEB, DIRECT URL METHOD:

When dealing with APIs that do not require authentication headers, a simple URL input can be used. This approach is quick and efficient, making it suitable for less sensitive data that is publicly accessible or protected by simpler means, such as token-based access in the URL.

YOU WILL NEED:

Power BI MCODE and AWS instance only of your db is VPN secured.

GET DATA - FROM WEB