Azure Data Factory – Using an Azure Function for more Complex Interaction with an API

Overview

Using an Azure Function for more Complex Interaction with an API

Setting up an Azure Function to Pull from GA API

Prerequisites

During setup in Lab 0, you should have installed:

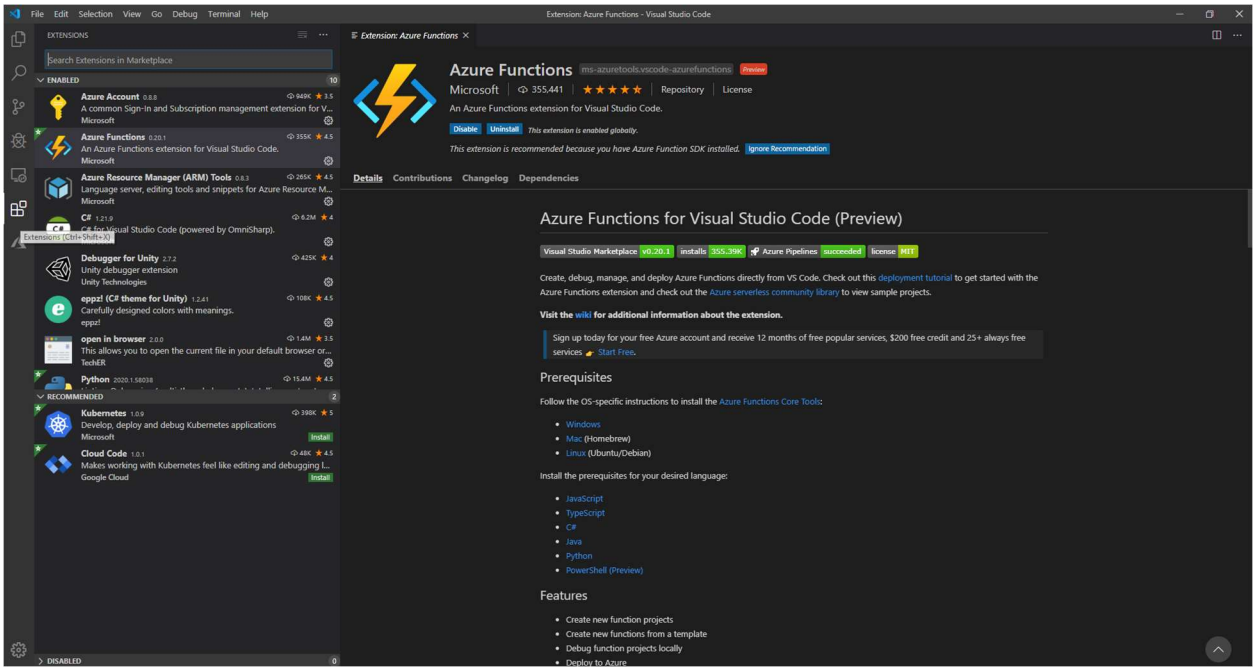
Visual Studio Code (https://code.visualstudio.com/),

Node.js (https://nodejs.org/en/download/),

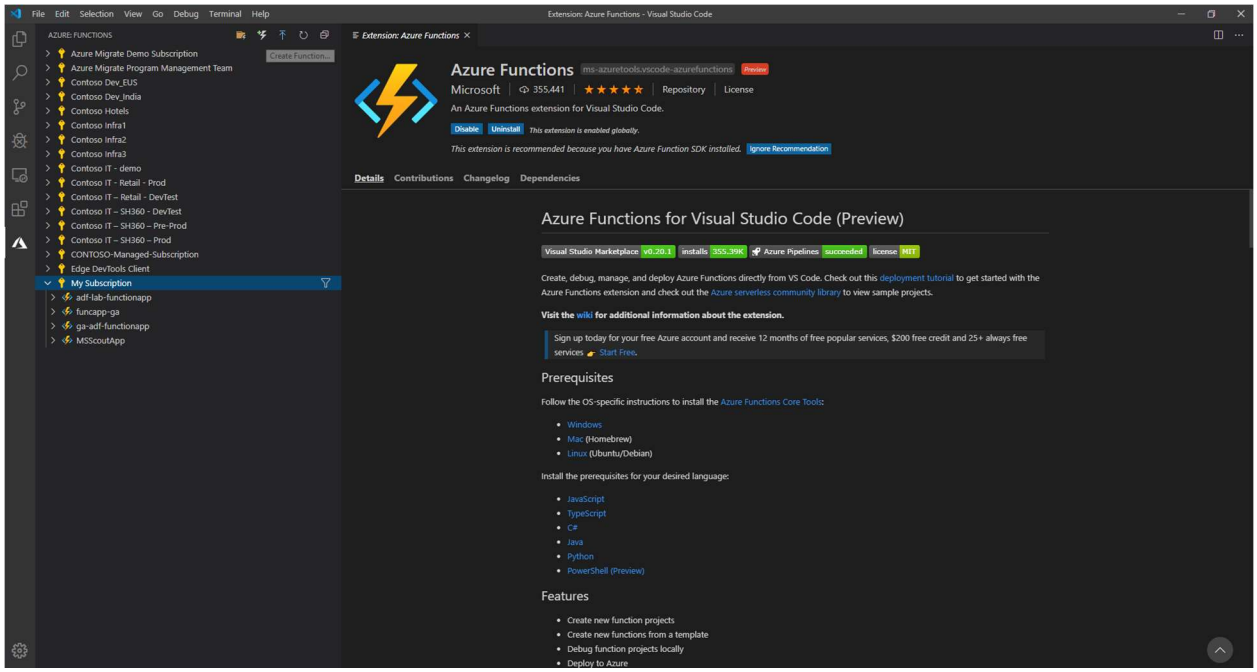
Azure Functions Core Tools (https://docs.microsoft.com/en-us/azure/azure-functions/functions-run-local?tabs=windows#install-the-azure-functions-core-tools)

Connect Visual Studio Code to your Azure Account

Open Visual Studio Code and in the Extensions tab on the right hand panel, search for and install the Azure Functions Extension.



Once installed (you may need to reload Visual Studio Code) you will see a new Azure Functions tab in the right hand panel. Navigate to it.

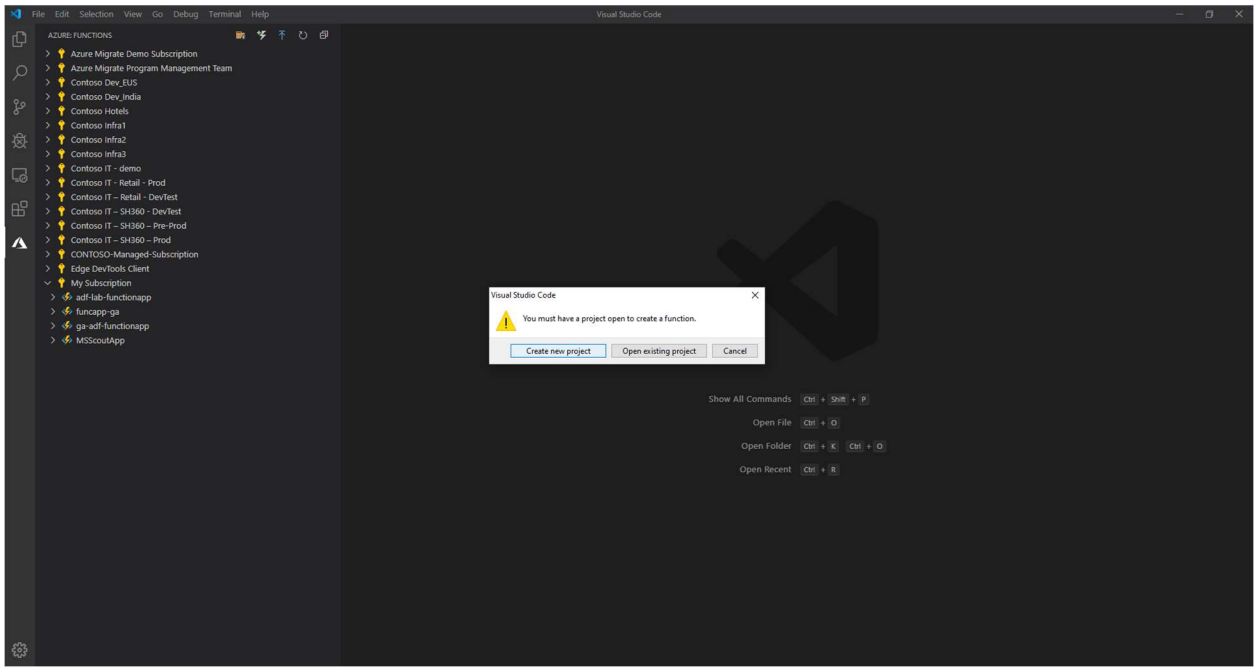


In the Azure Functions tab you will be prompted to connect your Azure account and be able to sign in. Once signed in, you will be able to see your subscriptions listed.

Create your Function

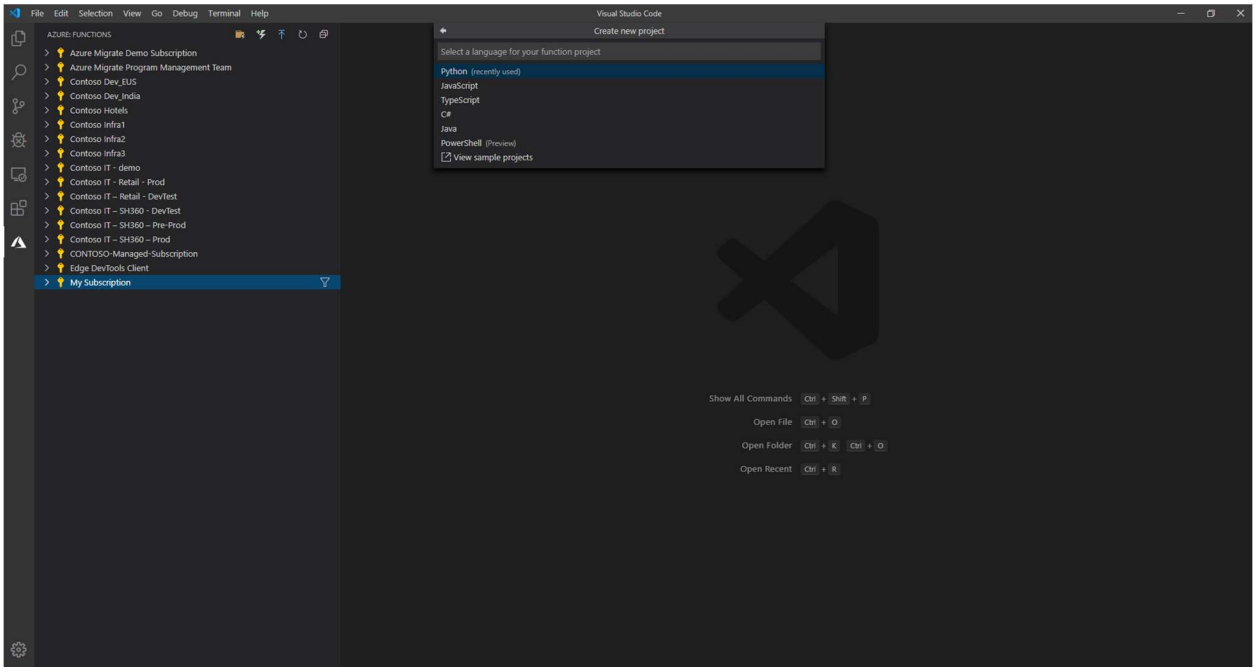
At the top of the Azure Functions panel click the Create Function button 

When prompted, select Create new project.

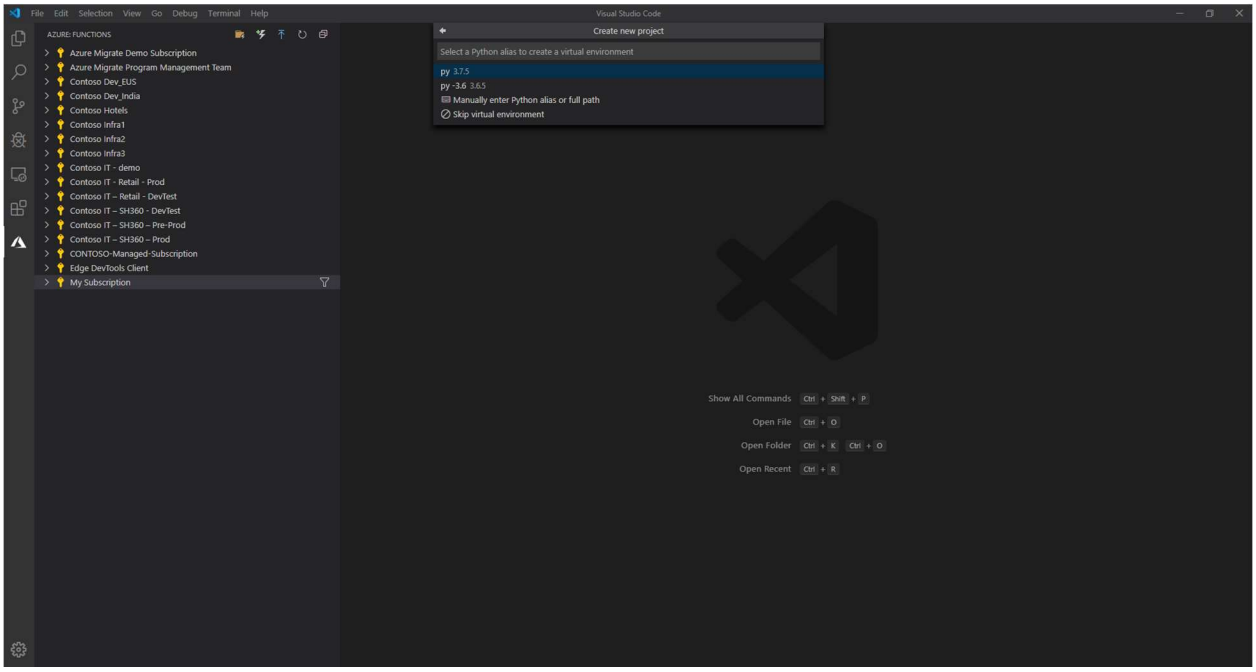


Choose a name for the folder (eg. Azure-Data-Factory-REST-Lab) your project will reside in and create it.

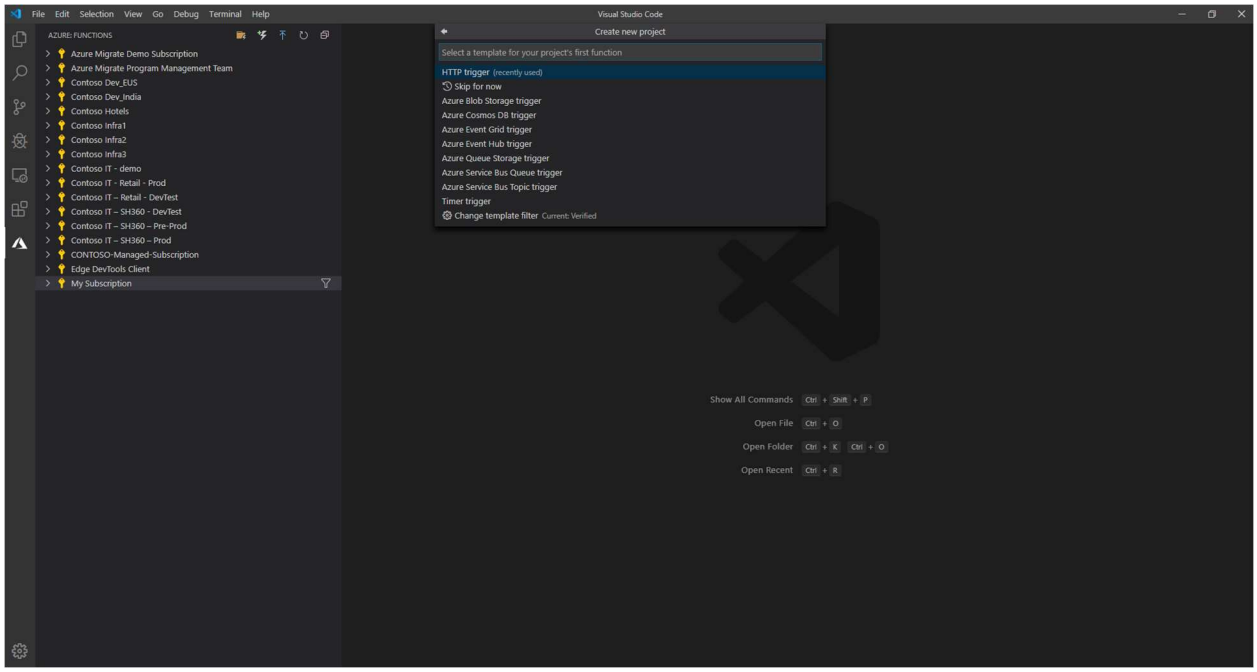
When prompted, select Python as the language for your function project.



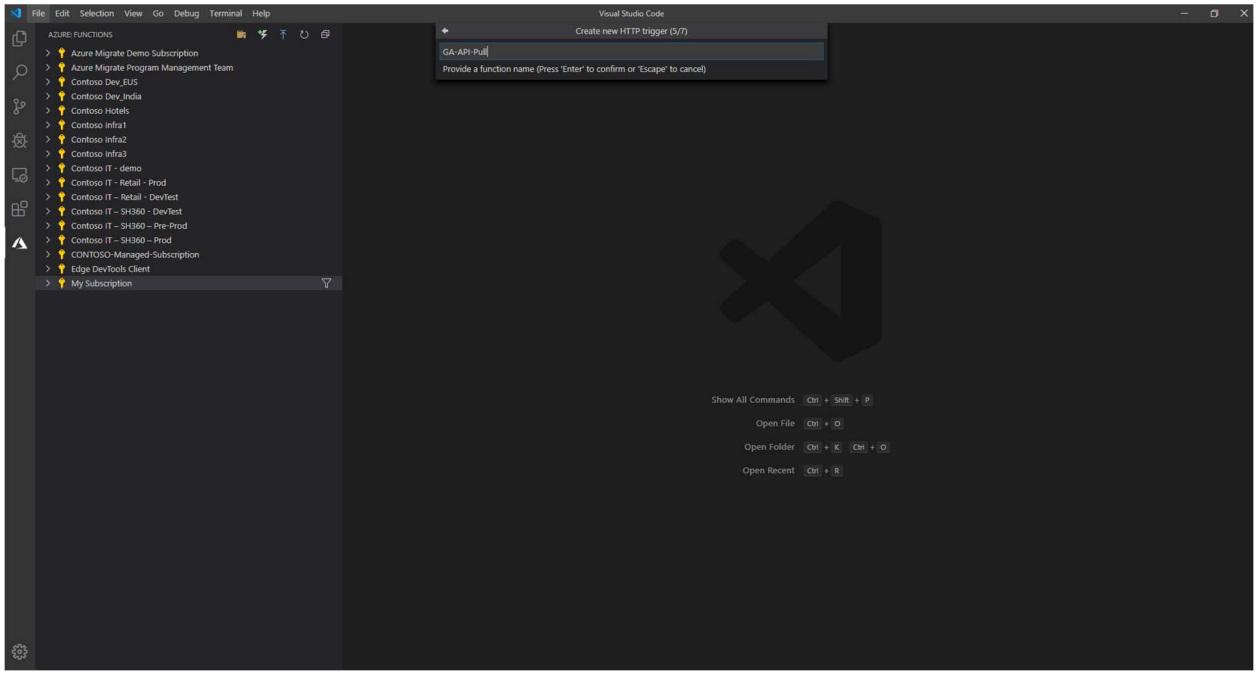
If prompted to select a Python Alias, choose 3.7.5



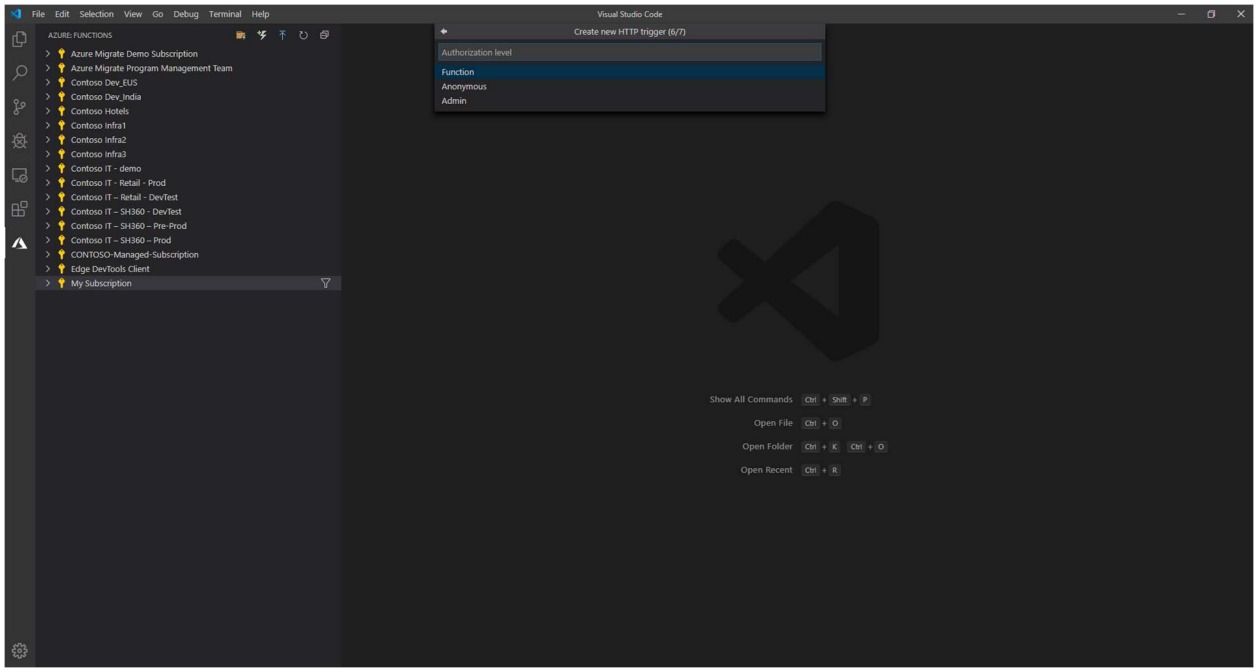
Next, for the template selection, choose to use the HTTP trigger.



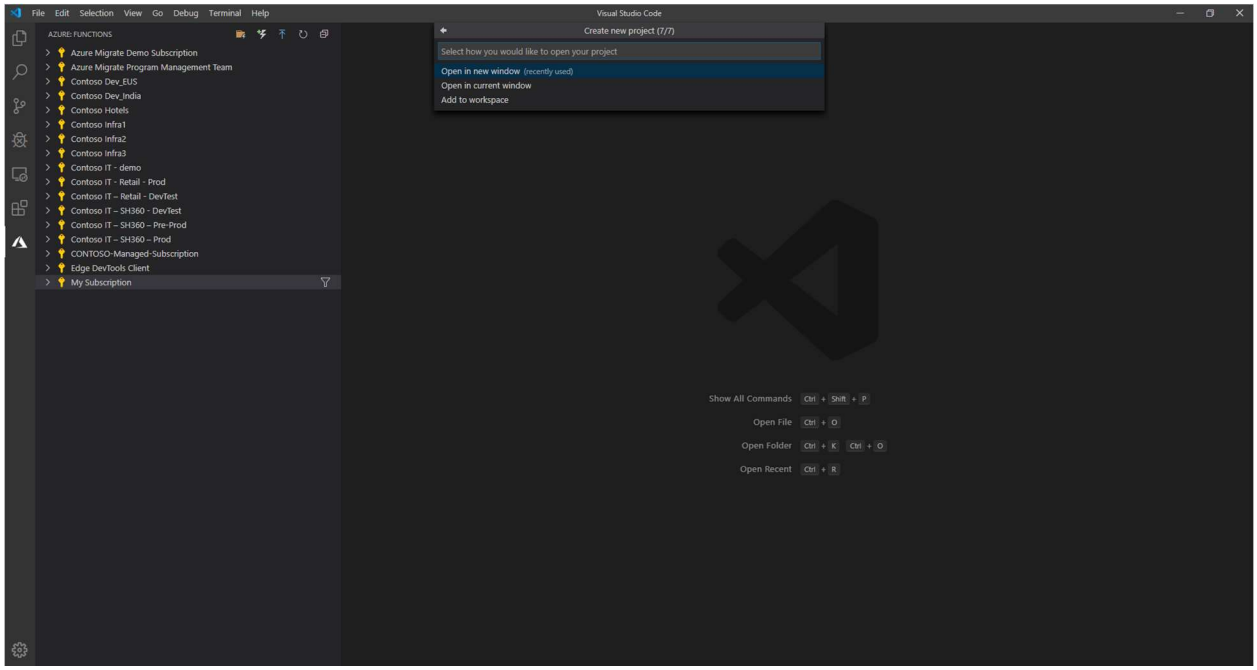
Next select a name for your Function (eg. GA-API-Pull)



For Authorization level, select Function

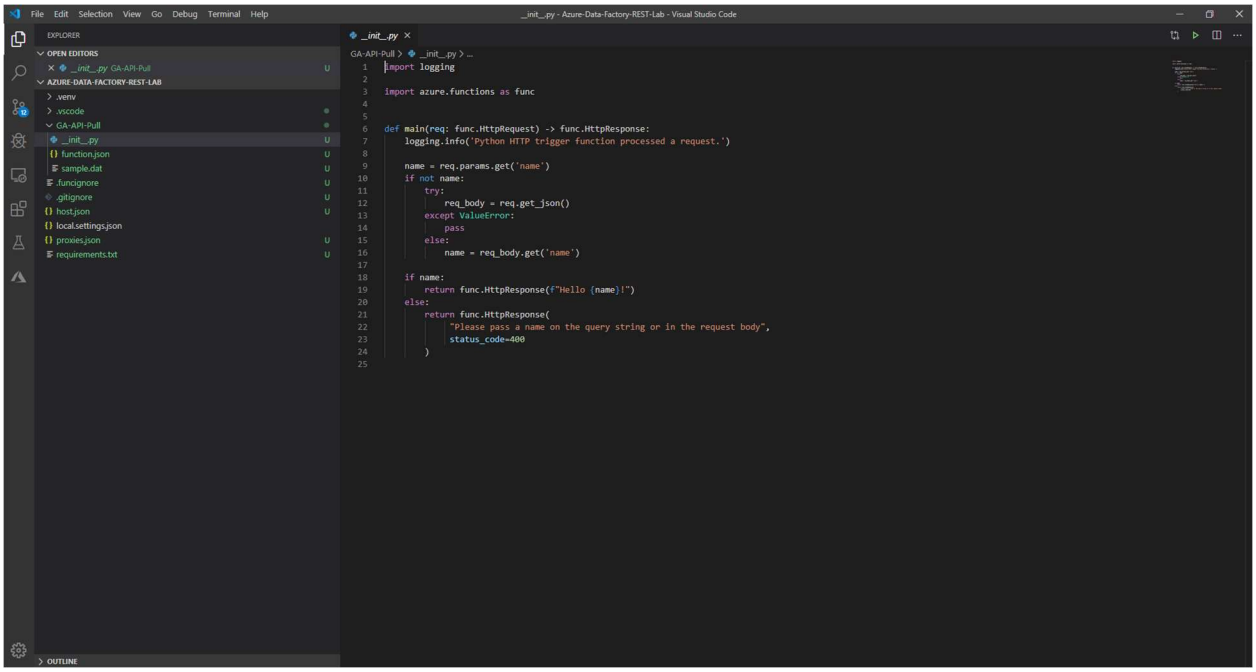


Finally, choose to open your project in your current window.

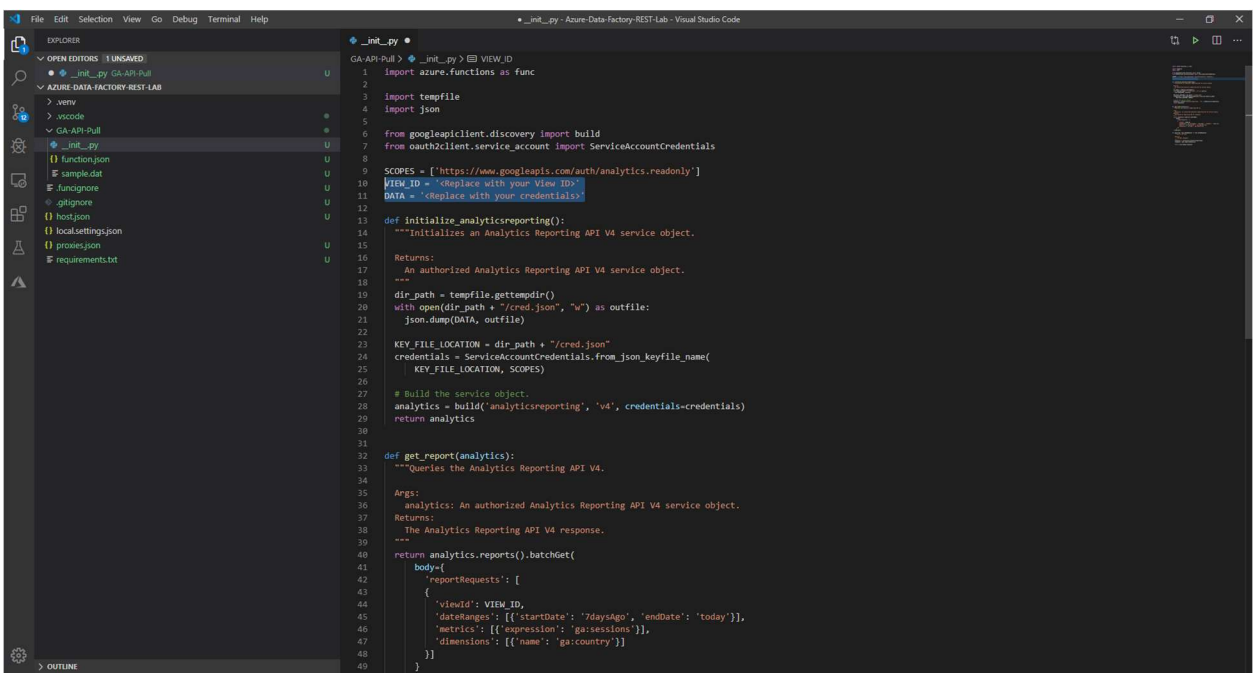


Add your custom code from the lab's github repository.

Navigate to your newly created project's '__init__.py' file.

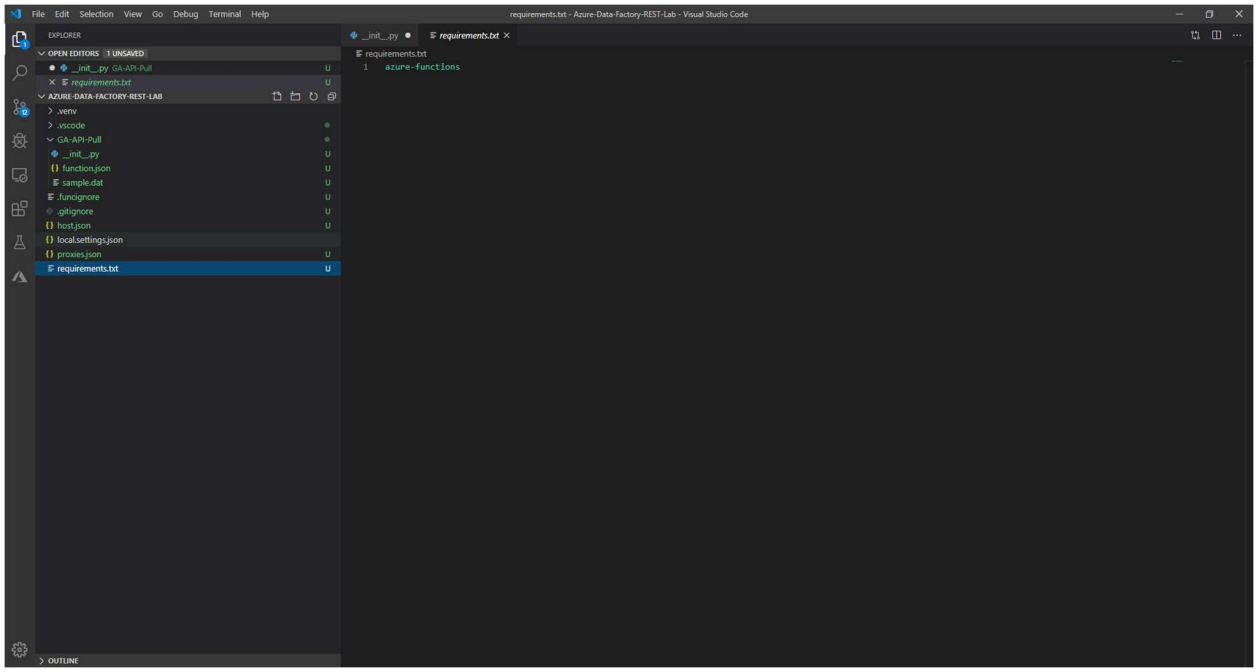


Copy and paste the code from https://raw.githubusercontent.com/Gryczka/Azure-Data-Factory-REST-Lab/master/src/GA-Pull-Function.py and replace the code currently in your __init__.py file.



Your proctors will provide you with a VIEW_ID and DATA to add to the script.

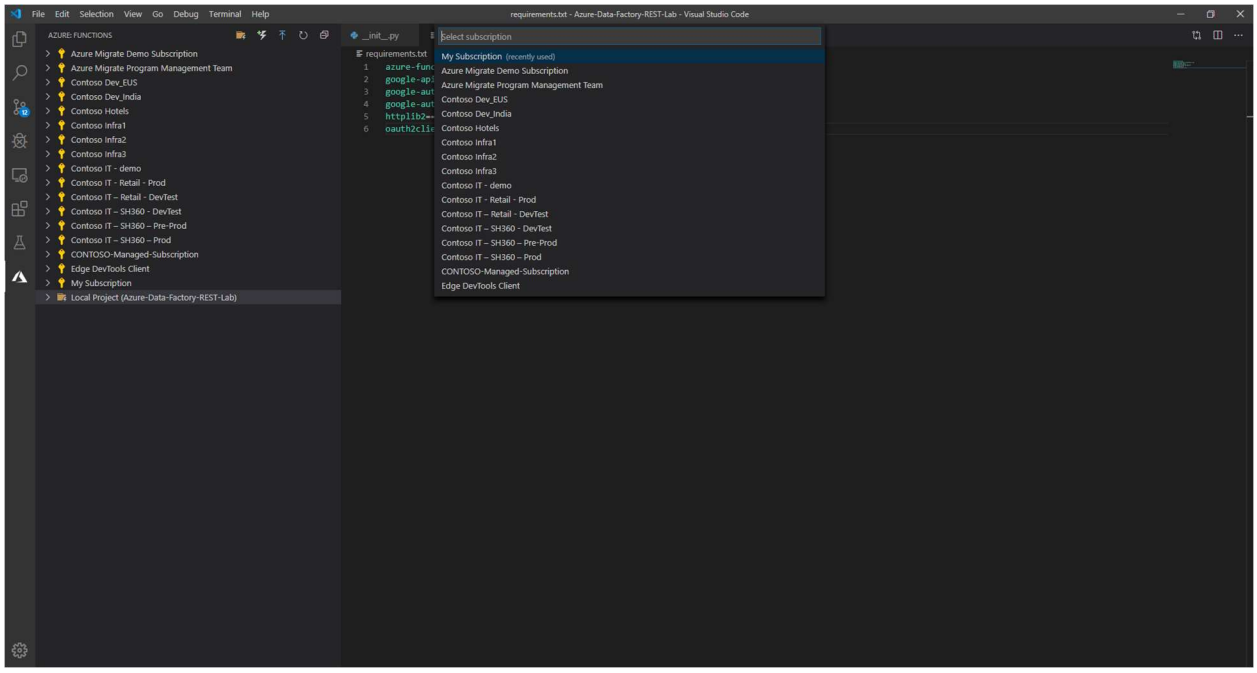
Next, navigate to your requirements.txt file.



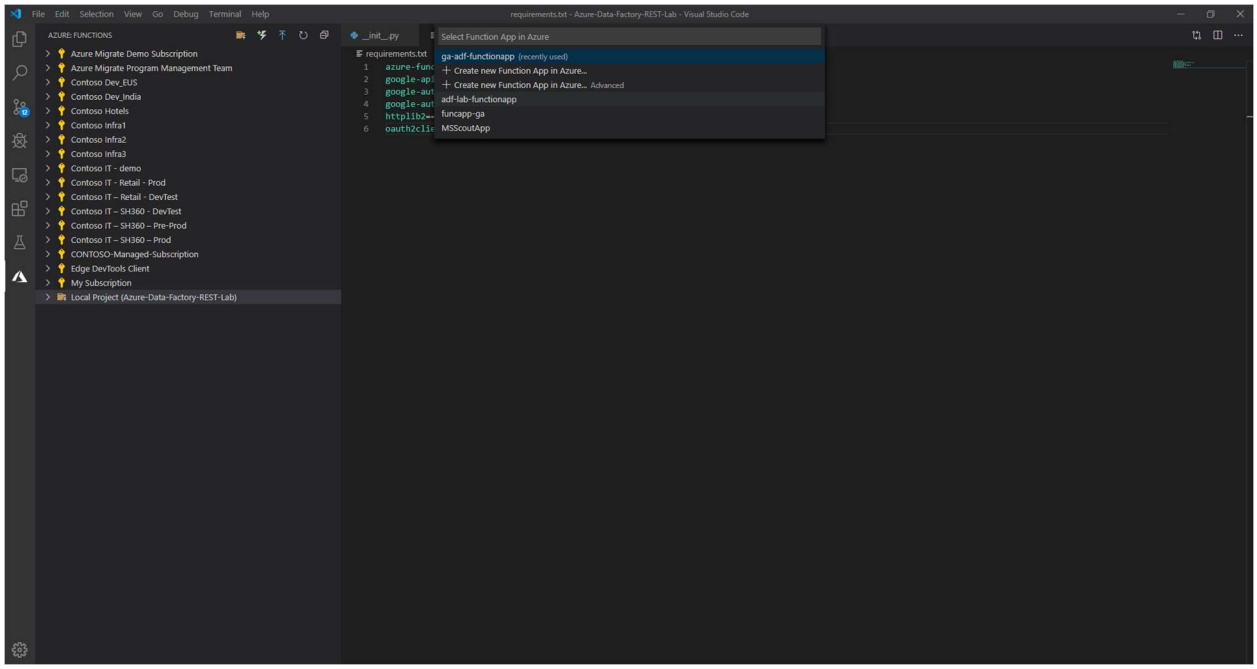
Add the rest of the requirements from https://raw.githubusercontent.com/Gryczka/Azure-Data-Factory-REST-Lab/master/src/requirements.txt and save both files.

Next navigate back to your Azure Functions panel in VS Code, and click the Deploy to Function App button 

You will be prompted to select your subscription.



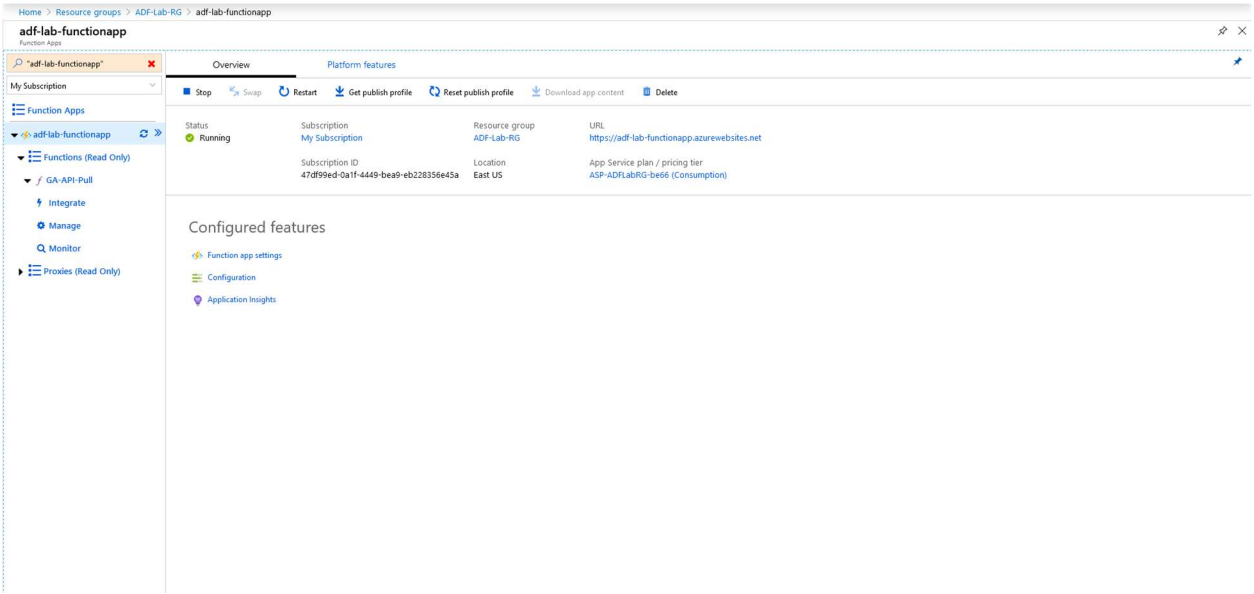
Then select the Function App you created during Lab 0 (adf-lab-functionapp)



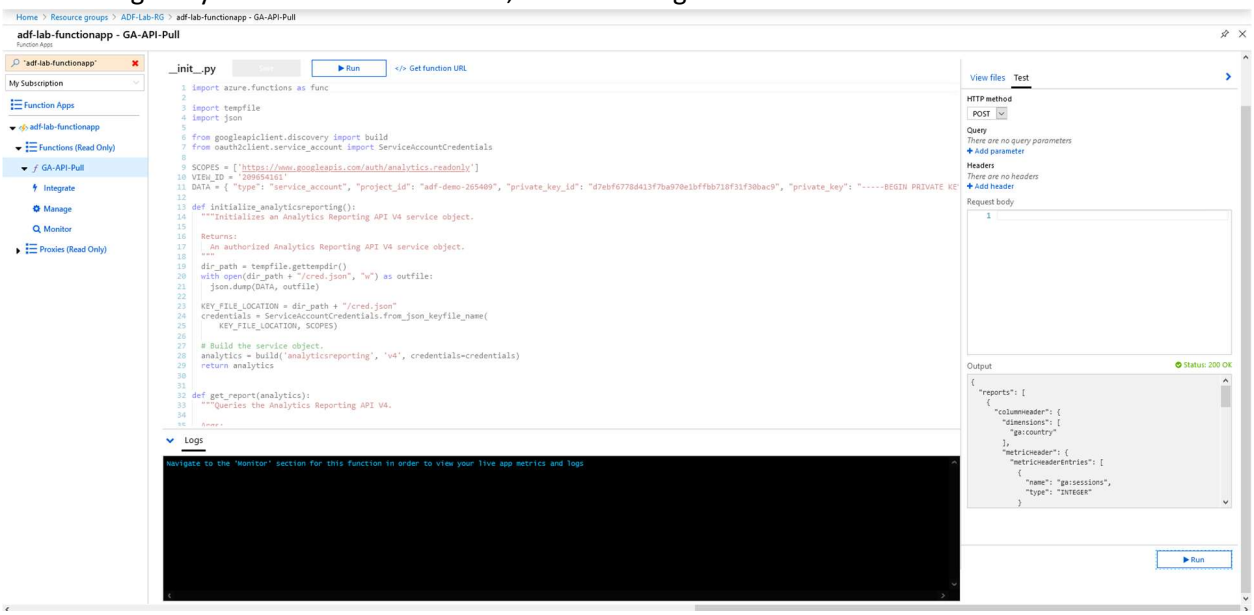
When prompted, confirm that you would like to deploy.

Test your Function in the Portal

In the Azure Portal, navigate to your adf-lab-functionapp



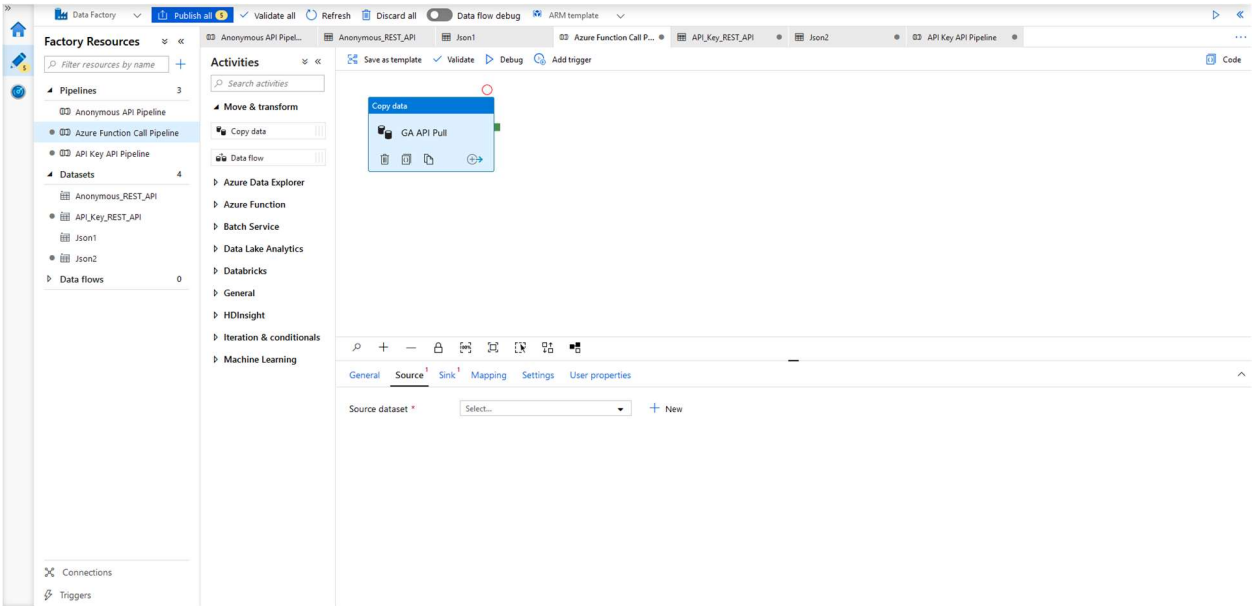
Click through to your GA-API-Pull Function, and on the right side Run a Test



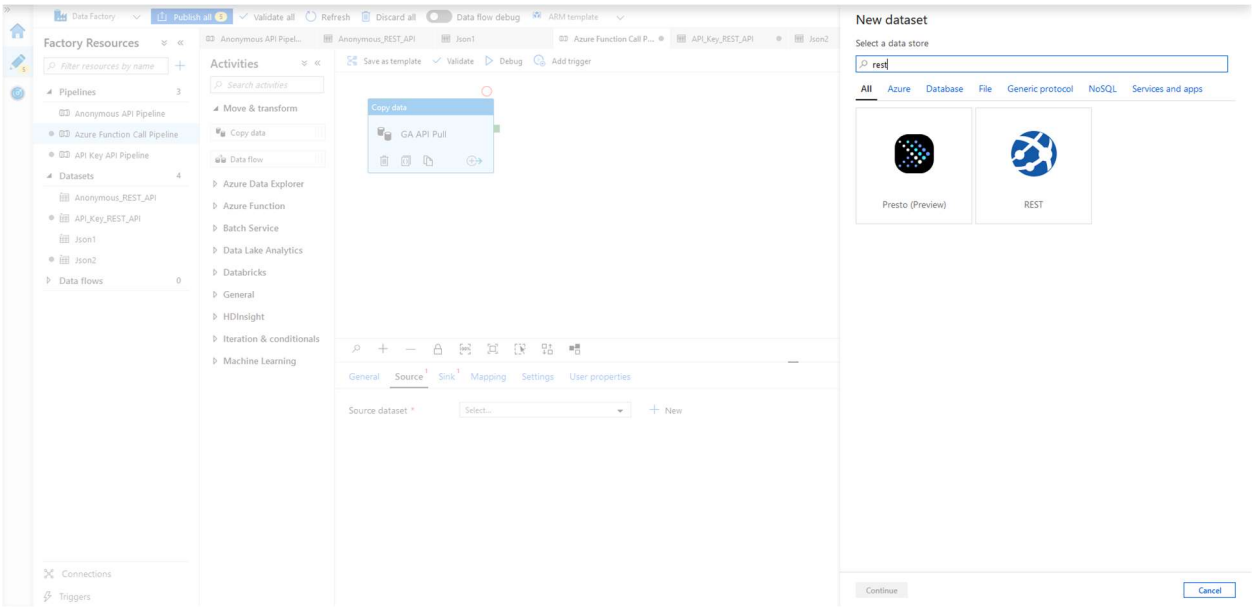
If you received an Output and a Status: 200 OK response, your Function is ready.

Calling the Function from Data Factory

Create a new pipeline to call your Azure Function and set up an activity to Copy Data

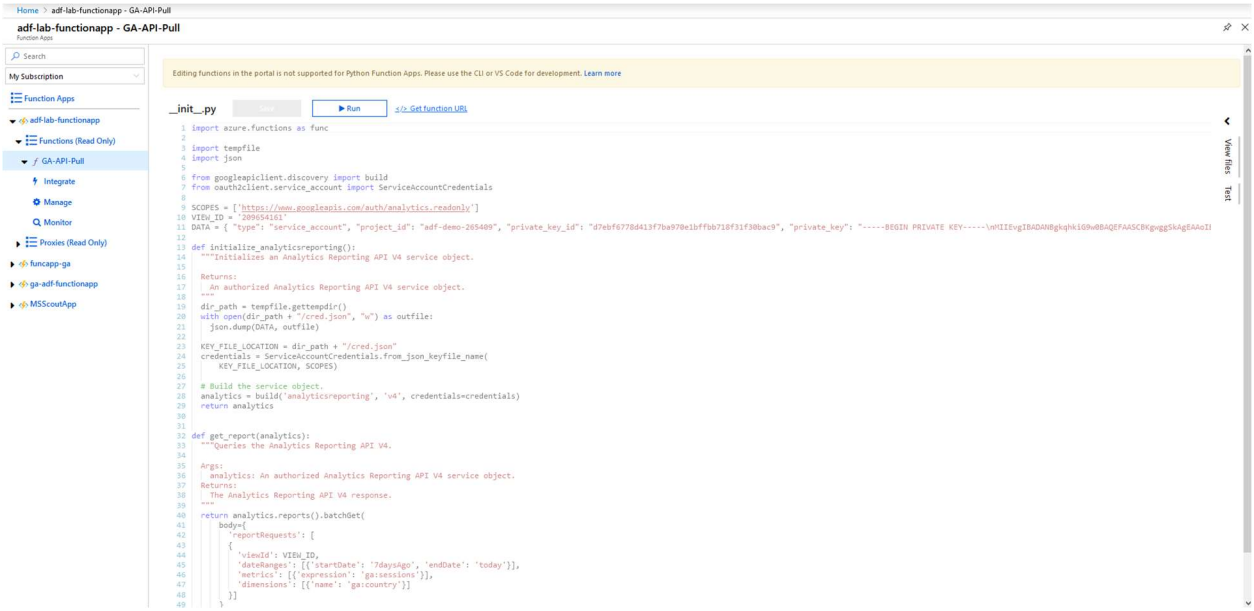


Create a Source dataset for the activity, again choosing REST

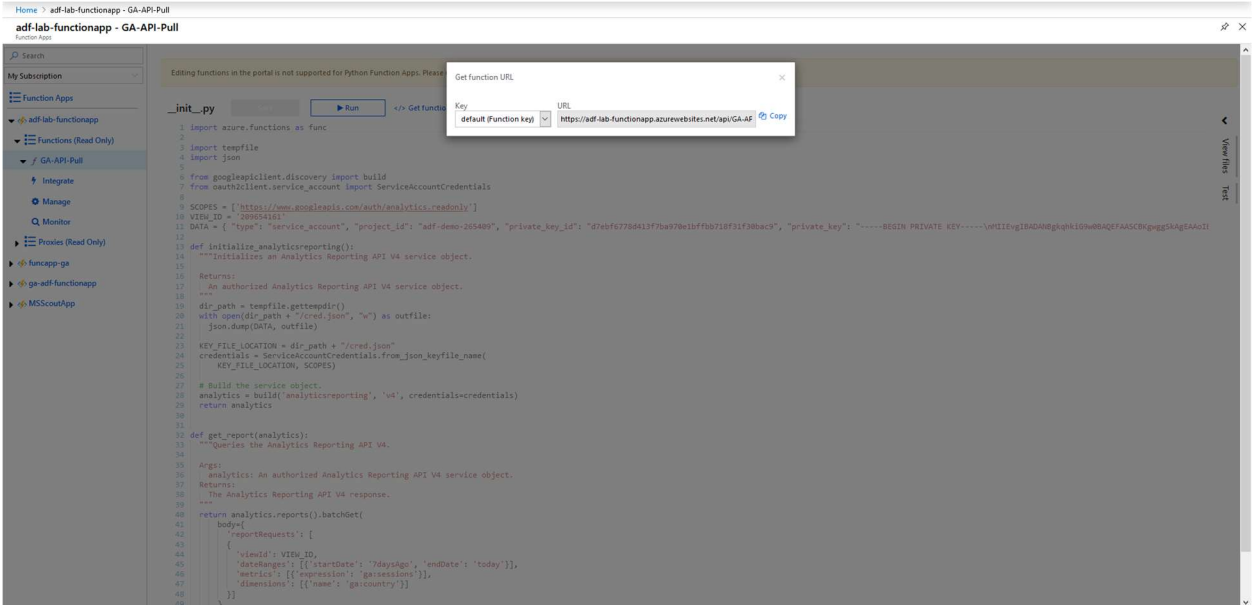


Next open the created rest resource and Create a New Linked Service

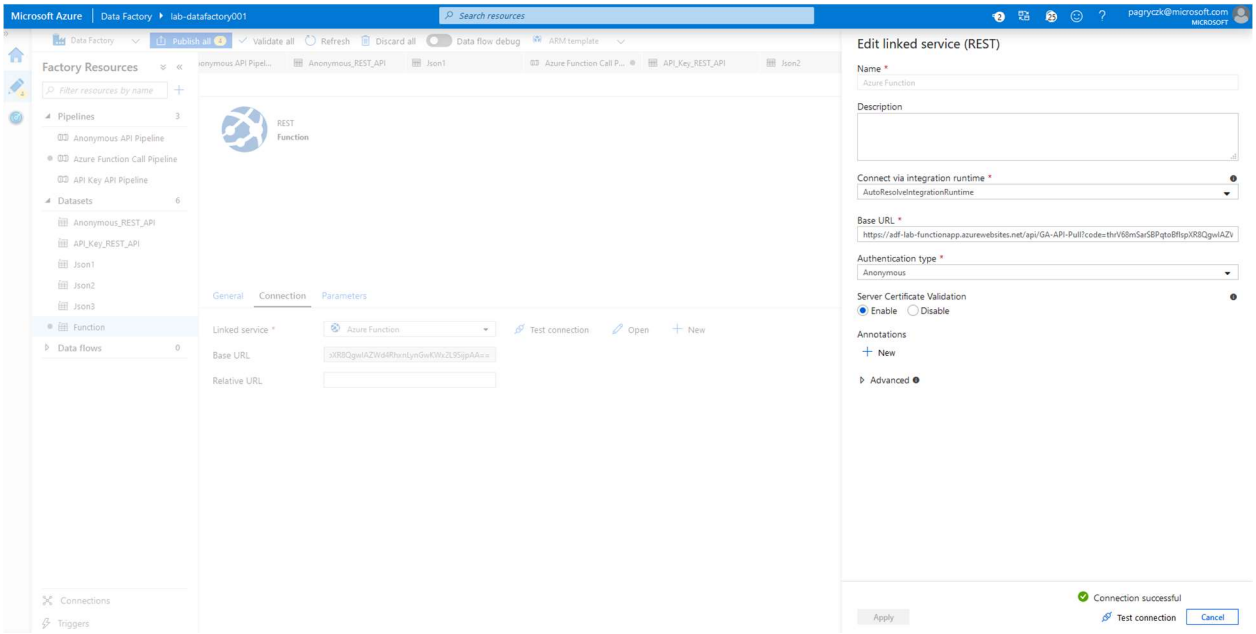
Next, open your Azure Function App in the Azure portal in another tab, and navigate to your function and click on



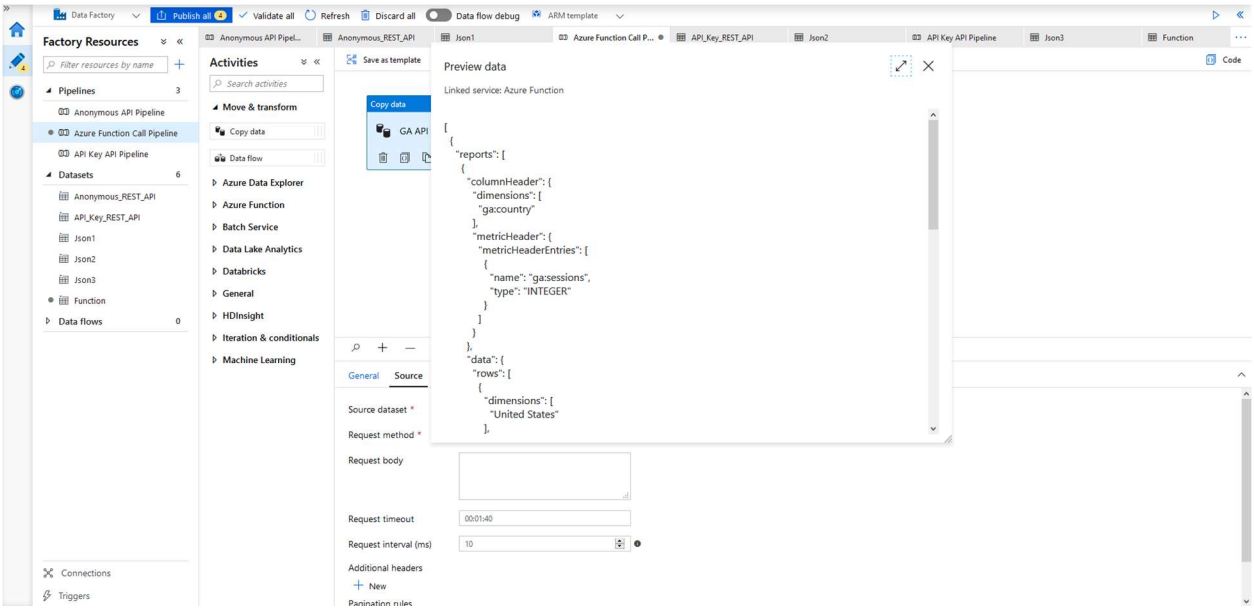
On the popup, click 



Now returning to your Linked Service, paste the copied url in as the Base URL with Anonymous Authentication selected, and test your connection before creating.



At this point you'll be able to preview your data in the copy activity to confirm it is connected.



Finally create a JSON sink to your Data Lake Gen 2 as you did for the APIs in Lab 1.