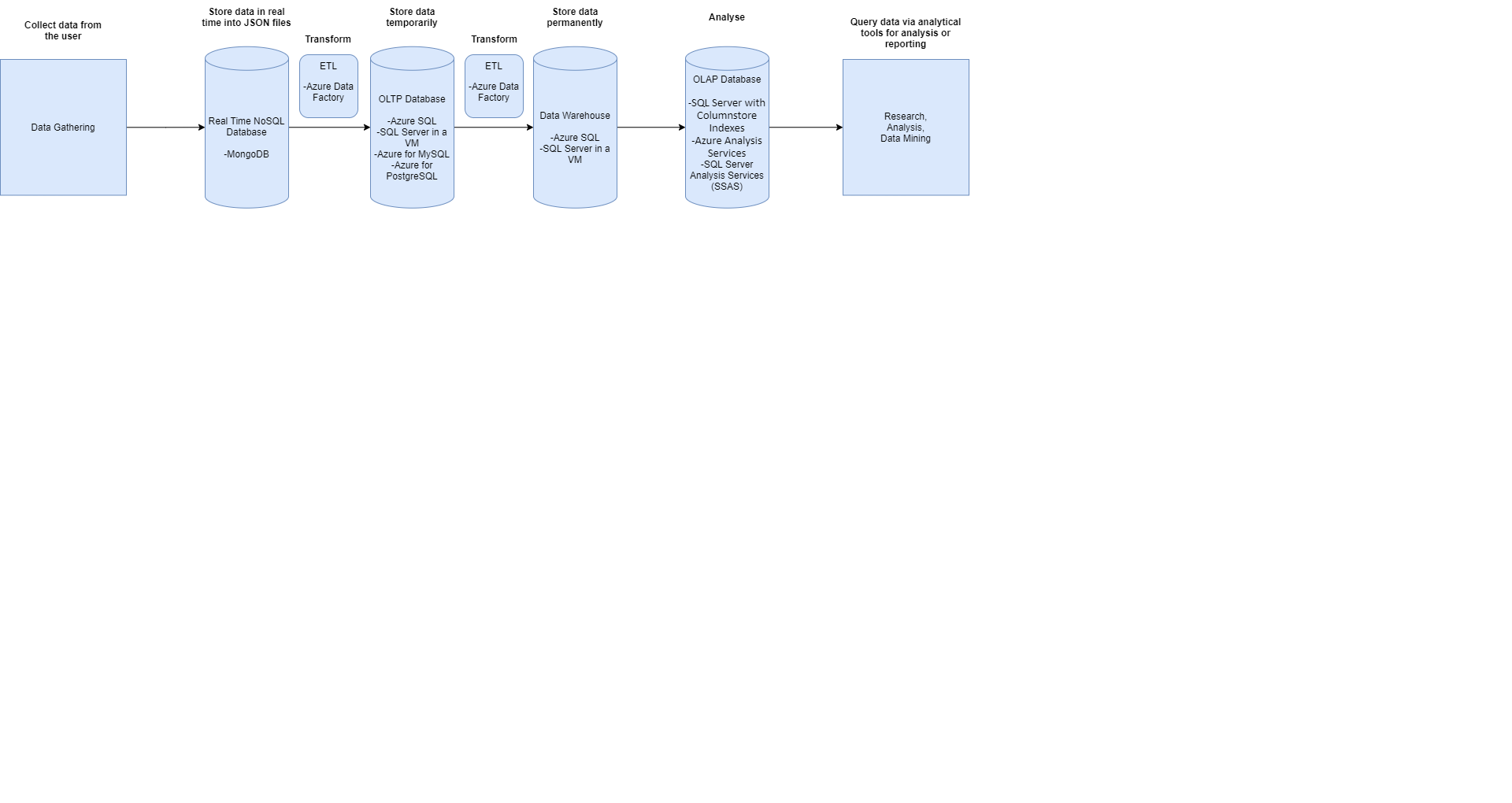
**ETL Data Processing**

**1. Introduction**

Moving data is an essential part of each business process. Sooner or later, we find the need of moving data. Luckily ETL services such as Azure Data Factory exist that can help us to read, write and transform data more easily.

Diagram of the general flow of data:

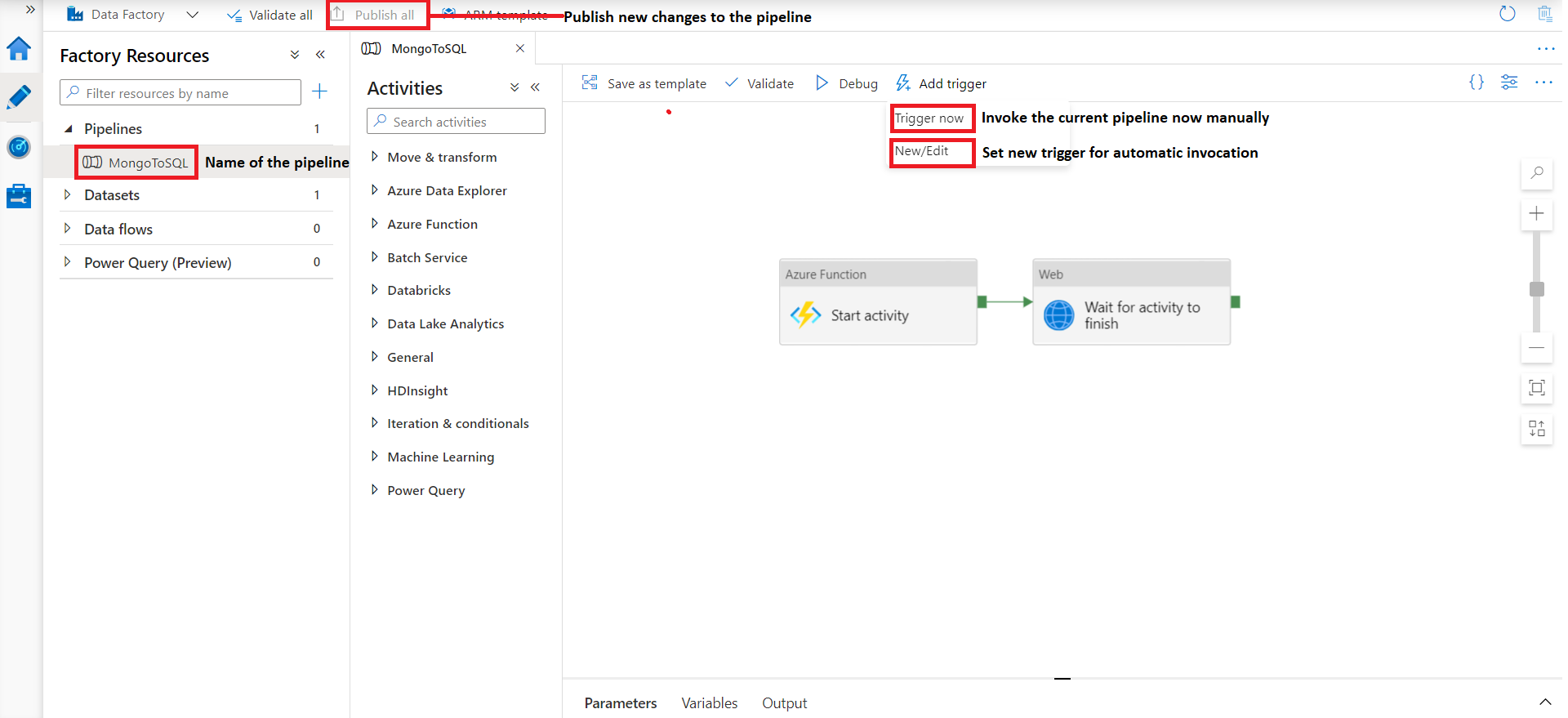


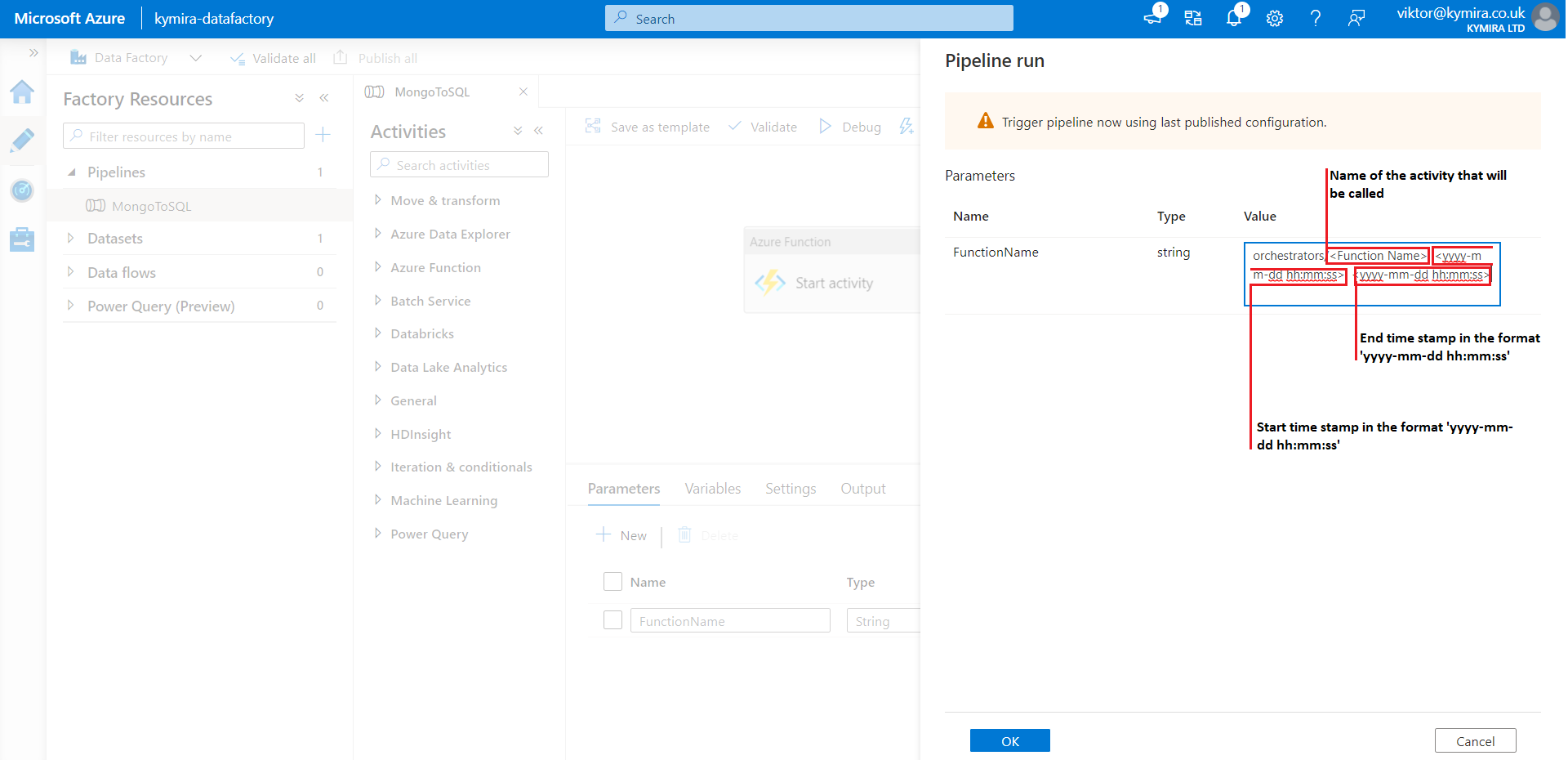
**2. Manual to Azure Data Factory**

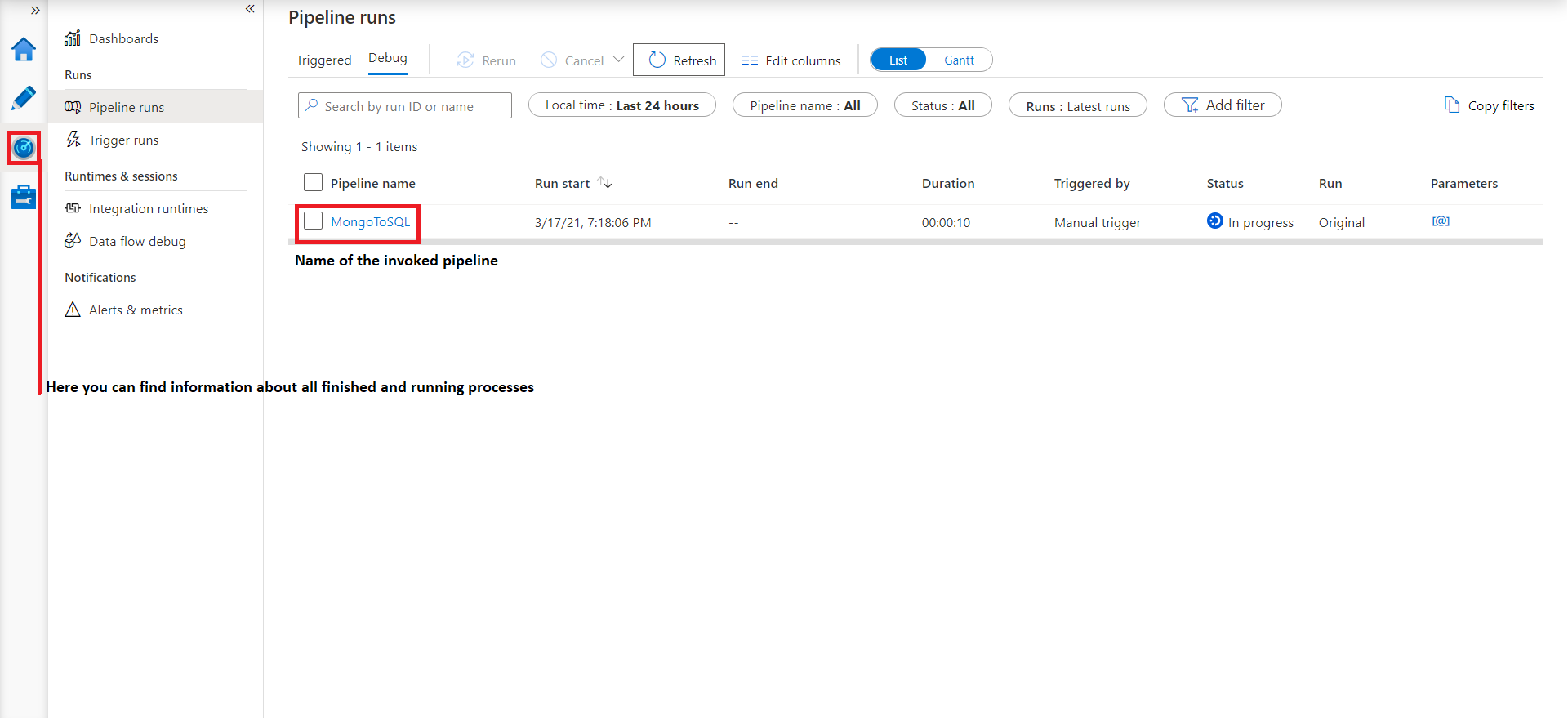
For our current needs I have created a python API that is called from our data factory as a function app. For now, it is configured to process only ECG type of data. The way we choose which activity to call is by providing the name of the activity as an argument when invoking the pipeline in ADF. The name of the activity is as follows:

* **ProcessECG**: works with the ECG data

We also need to provide two other arguments. The start time stamp in the format ‘yyyy-mm-dd hh:mm:ss’ and the end time stamp in the same format.



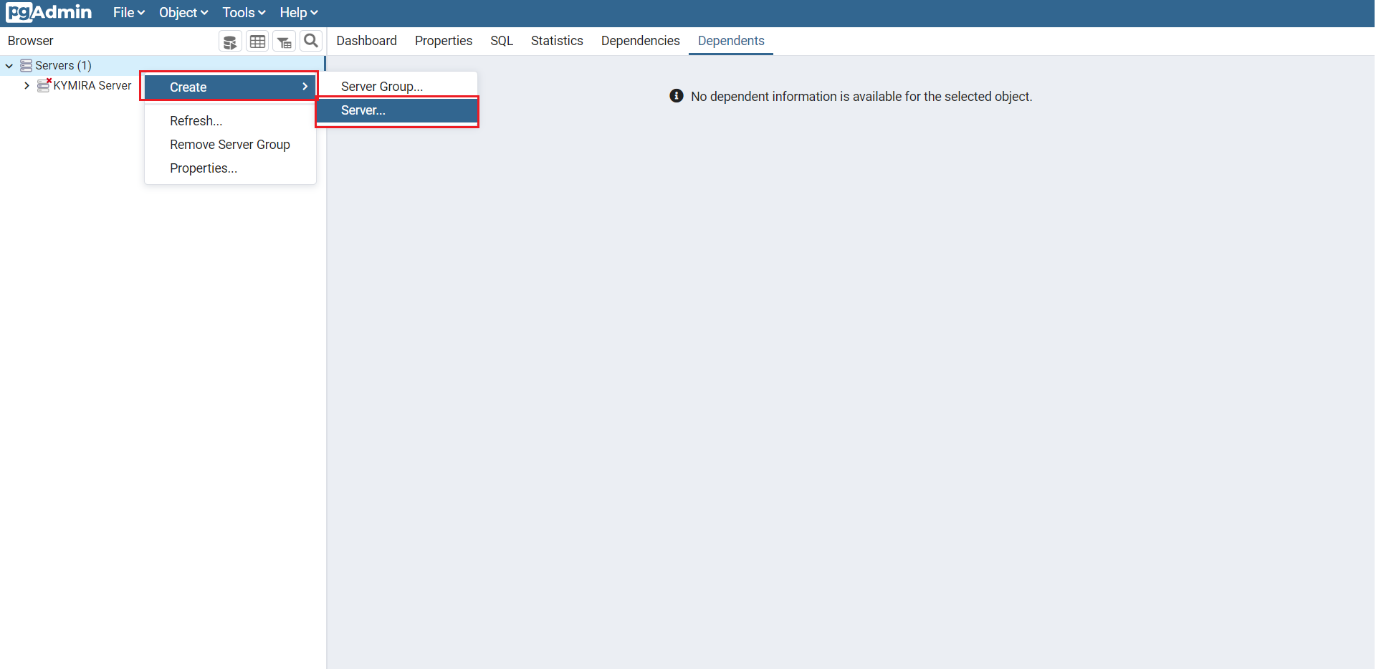
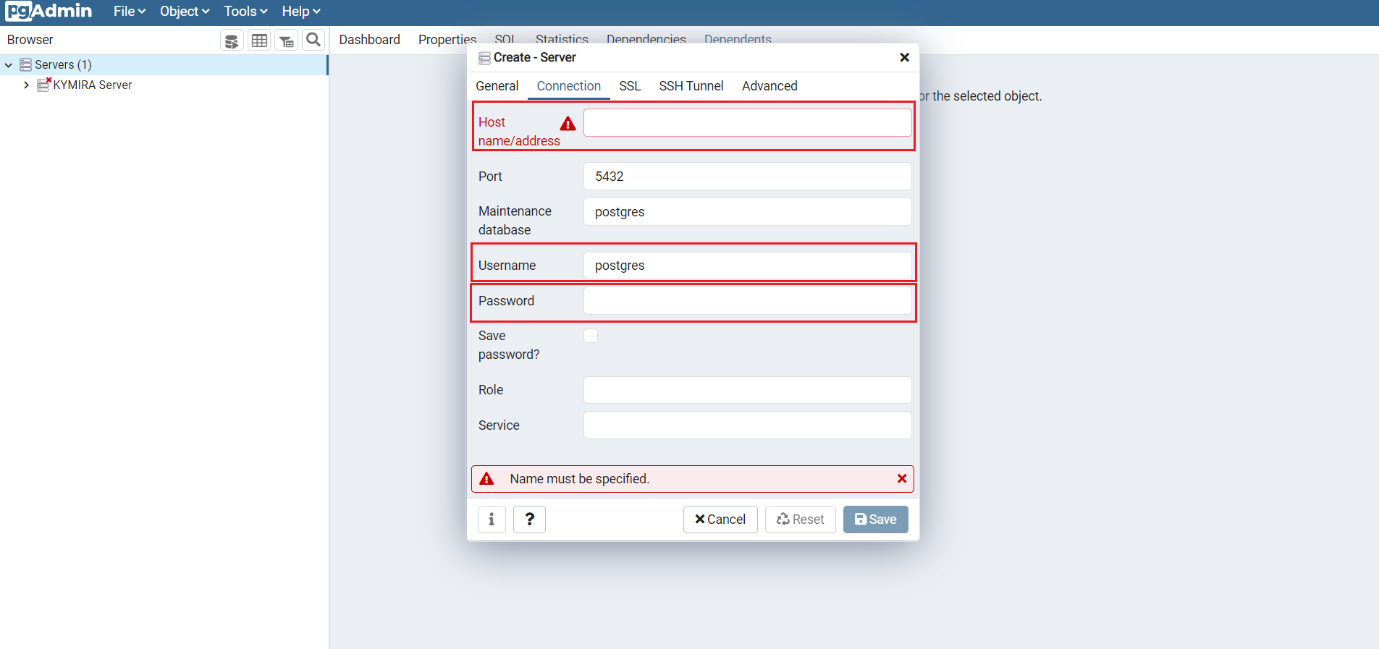




**3. SQL database**

All processed entries are written to PostgreSQL database. To access the data, you will need to download pgAdmin4 from here: <https://www.postgresql.org/ftp/pgadmin/pgadmin4/v5.2/windows/> and connect to the server. To achieve that you will need the name of the server, the admin’s name, and the admin’s password. Both the name of the server and the name of the admin can be found in the overview section when you open PostgreSQL in Azure. If you can not remember the password, you can always reset it by clicking the ‘reset password’ button.

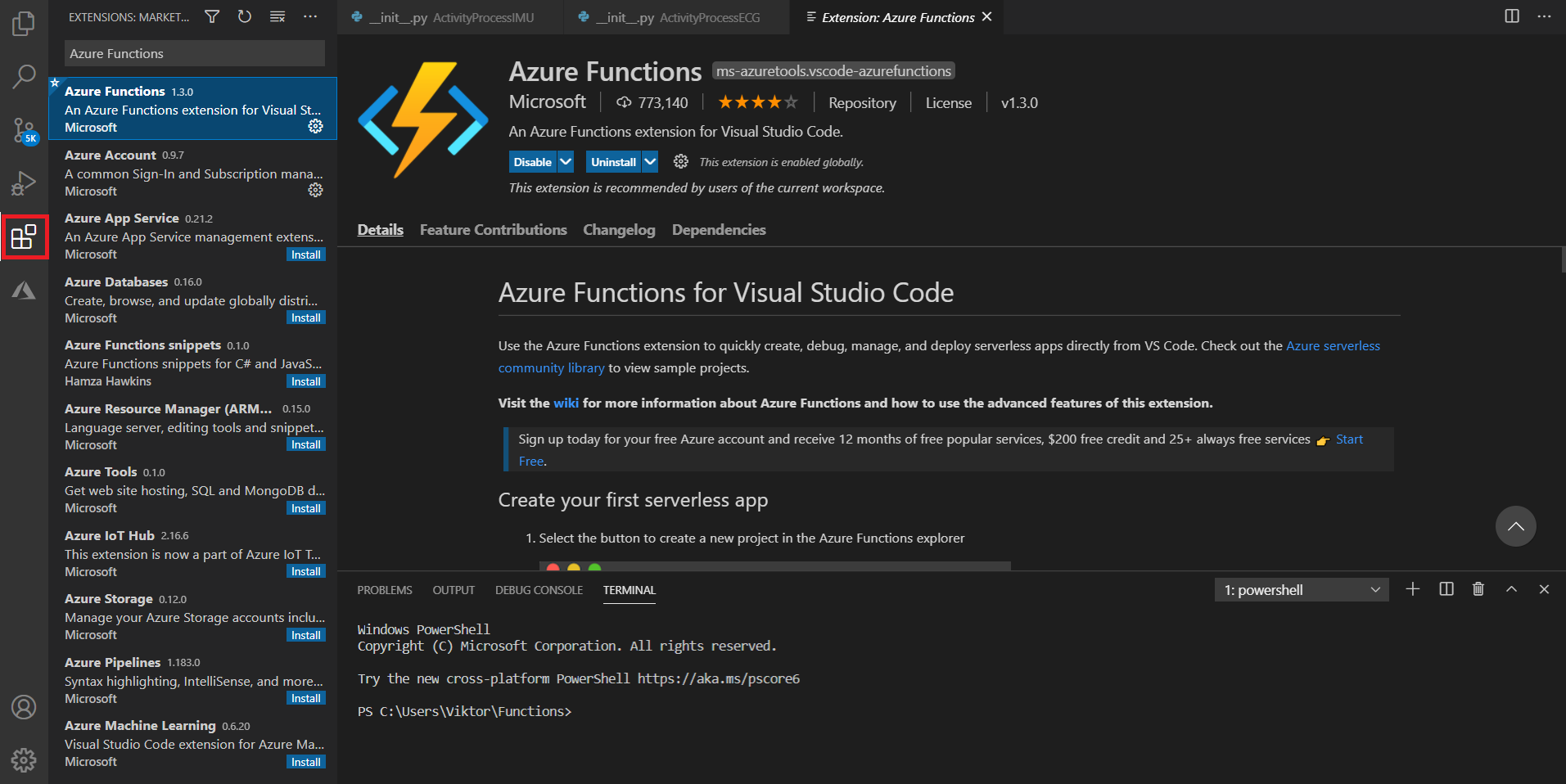
I emphasize that the name of the admin needs to be in the following format: <admin\_name>@<server\_name>. Once you connect to the server you will gain access to all the schemas and tables that are inside.



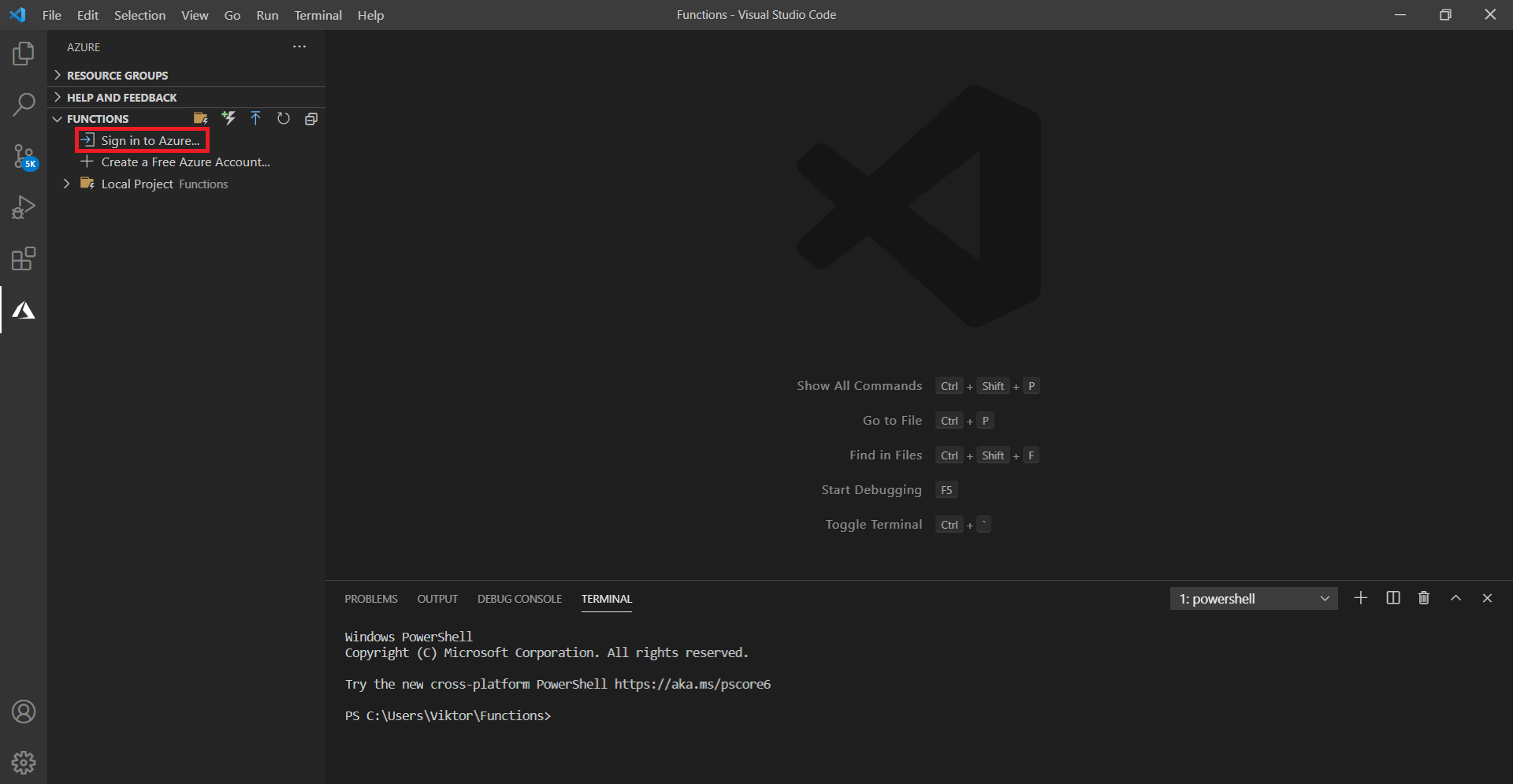
**4. How to add new functionalities to the Function App**

As a said the processing is done by a Function App written in python that is triggered by an HTTP request. Azure does not support the writing of python scripts in the web portal so to work with the application and add new functionalities it needs to be connected to Visual Studio Code. The process is as follows:

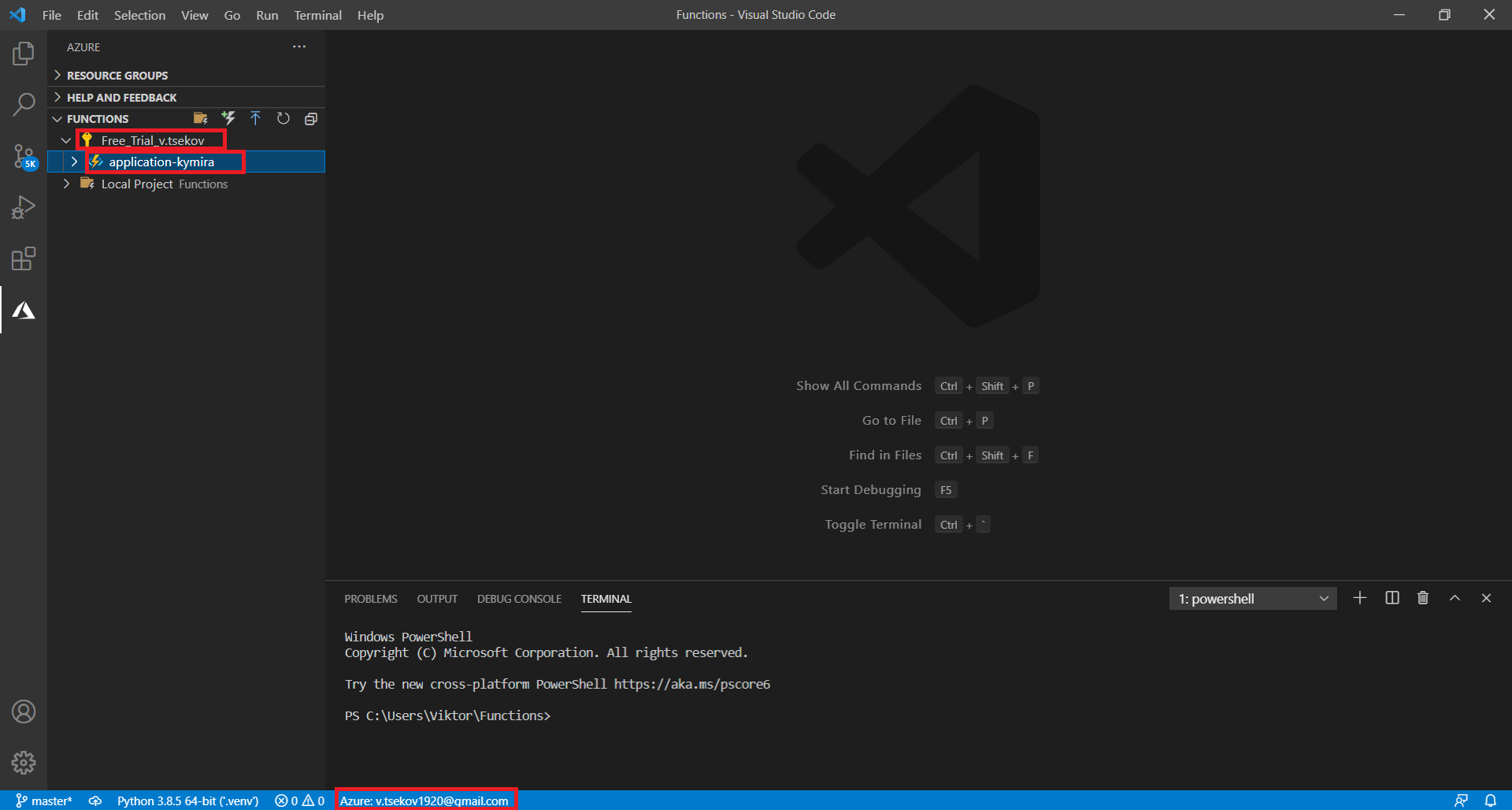
Firstly, you need to go to extensions and add ‘Azure Functions’.



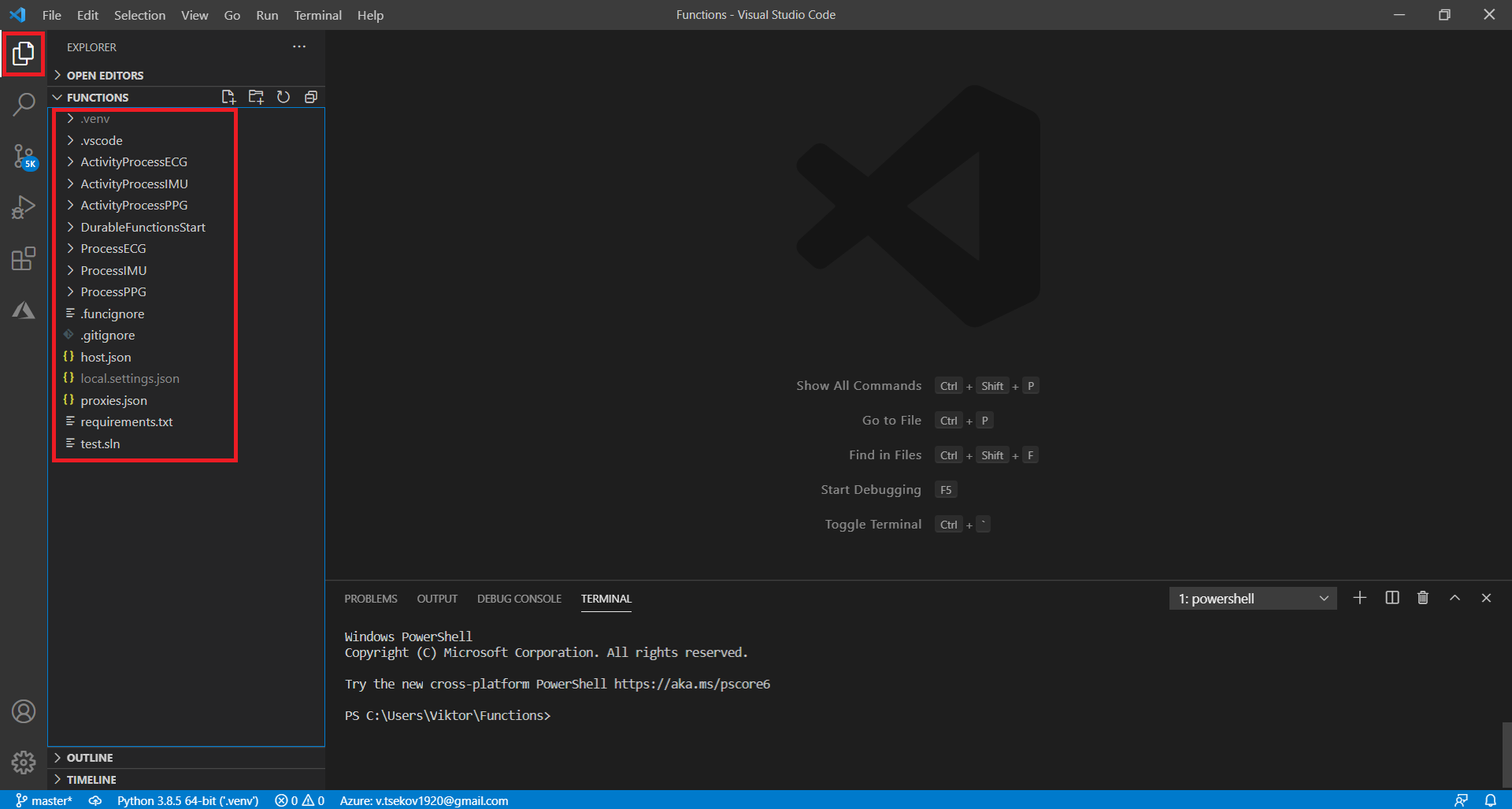
After that you need to click on the azure icon that will appear and login to your azure account.



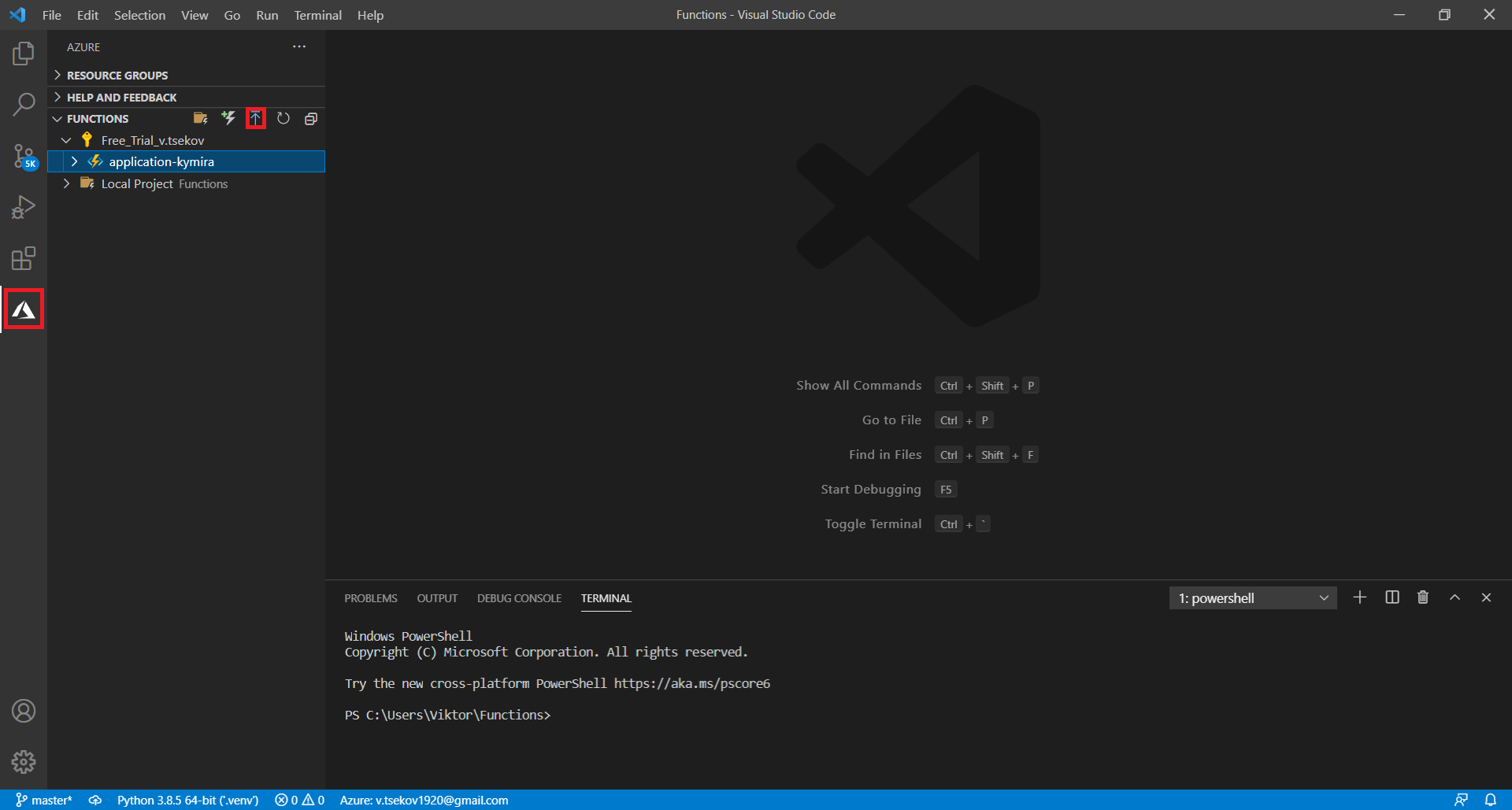
After that you should be able to see your username at the bottom of the screen and all the subscriptions that belong to that account. When opened, the subscription will show all the function applications that are inside it.



But the contents inside the function can only be read. To edit the functions, you need to create a local project and upload the changes to the application later. I have uploaded the application to a repository on GitHub that can be found here: https://github.com/ViktorTsekov/ETL\_Data\_Processing. If you clone the repository and open it in your explorer in Visual Studio Code, you will be able to access the functions and modify them.



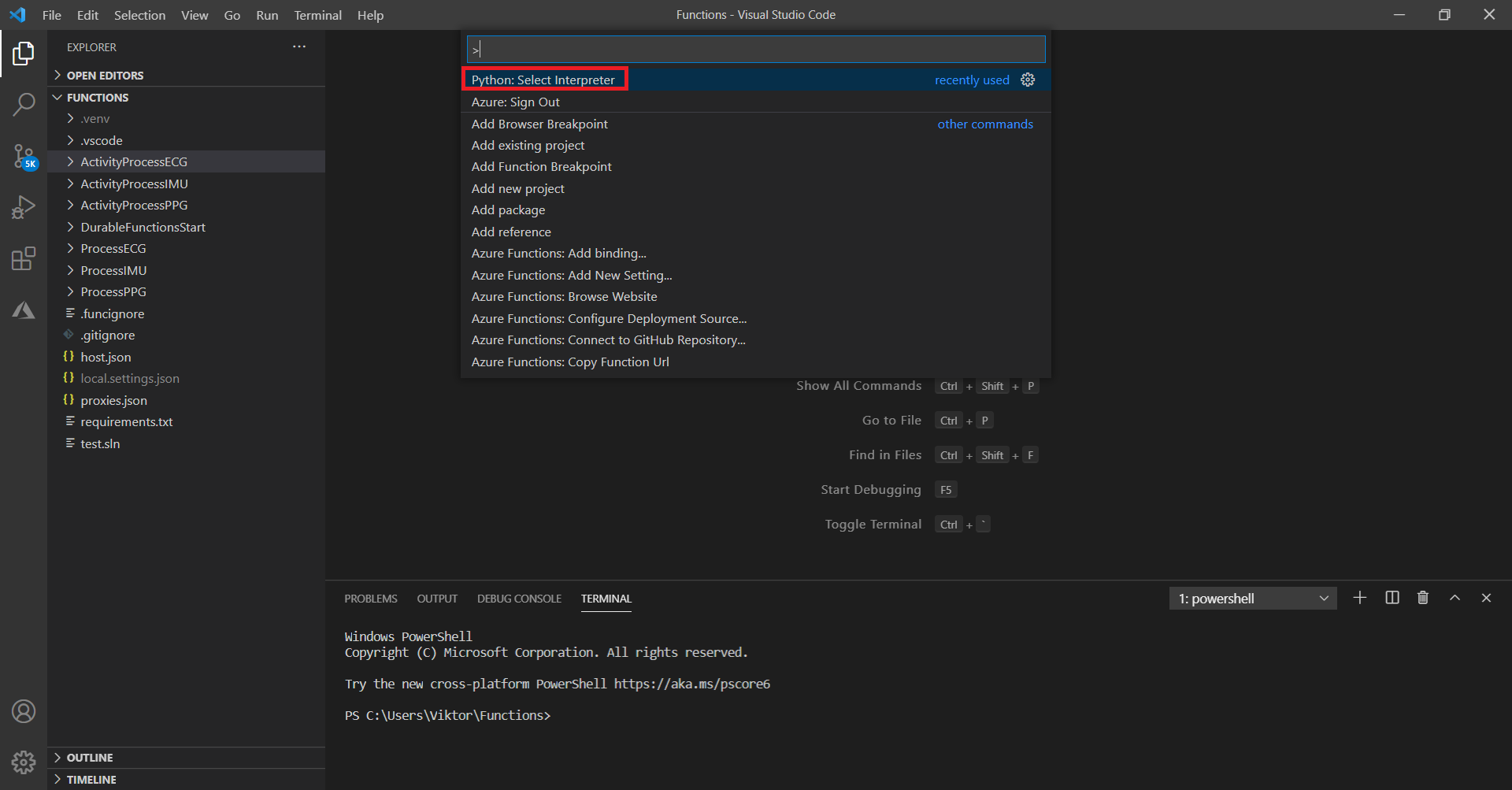
Once you are finished with the modifications you need to go back to the Azure extension and upload them to the cloud for the changes to take effect. That can be done by clicking the blue arrow at the top of the screen.

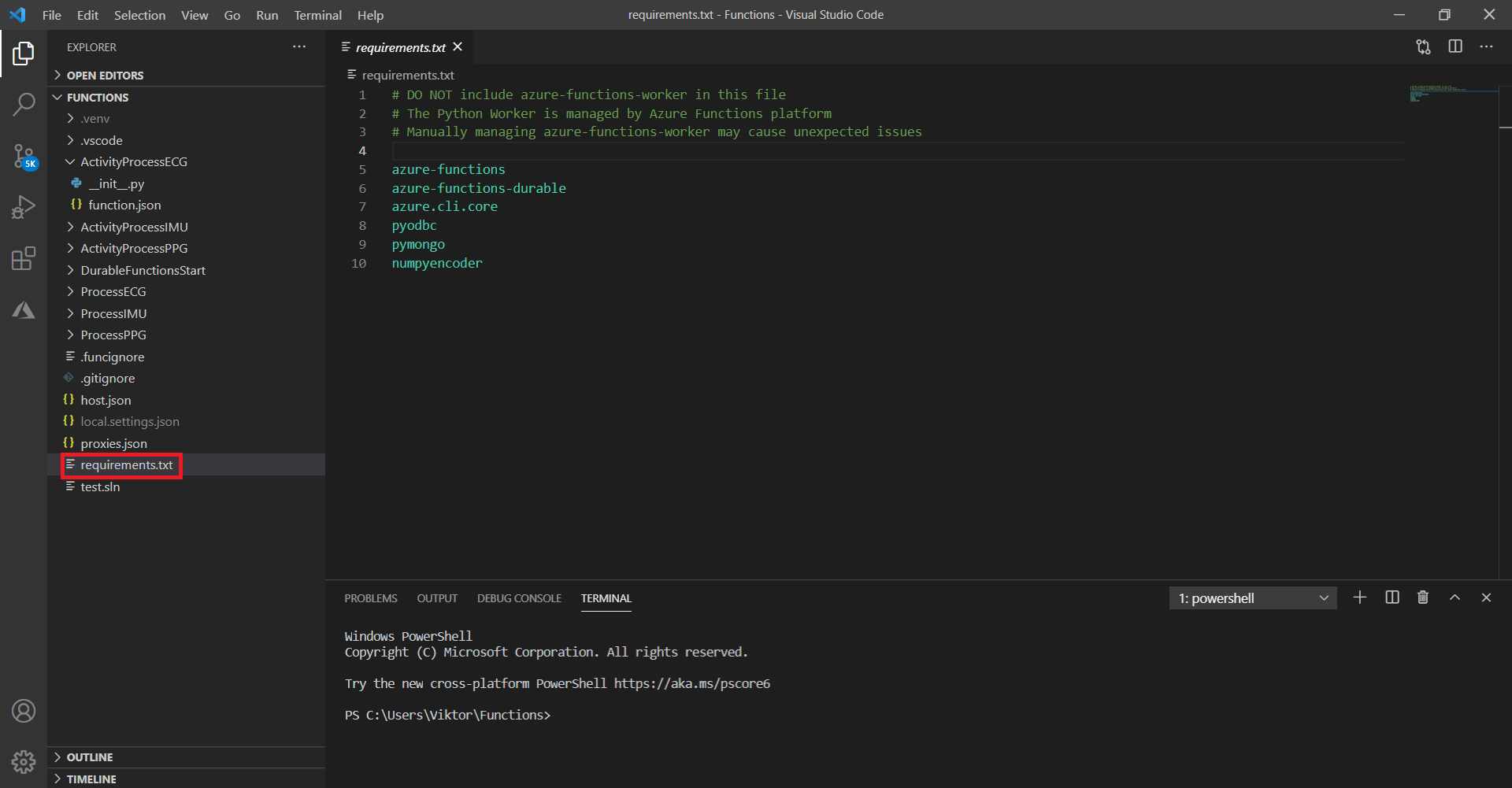


This YouTube video explains very well how to write python scripts in an Azure environment, I strongly recommend it: https://www.youtube.com/watch?v=Fb5pO3-62Nc&ab\_channel=ProgrammerInHouse

**5. Requirements and setup**

A python virtual environment is required to be included in the application to run the scripts we have written. A virtual environment can be created by executing the command ‘python –m venv <name\_of\_VE>’. Also, the newly created environment must be selected as a default interpreter by clicking CTR + SHIFT + P.



But let us say you need to use external libraries such as ‘pyodbc’ or ‘pymongo’. To achieve that you need to include them in the file ‘requirements.txt’ and run the command: ‘python -m pip install -r requirements.txt’. This will install the necessary libraries to the virtual environment that you have selected in the prior step. If that does not work, try specifying explicitly the library path of your environment by typing the command: ‘pip install --target= <library\_path> <package\_name>’. The libraries are contained inside the ‘site-packages’ folder in the virtual environment you have selected.

**6. Conclusion**

The activities in the function app are triggered from our ADF but the processing is done in the Function App itself. Once a function app is created it also creates its own resource group with an App Service Plan inside on which the application is hosted. Azure does not support writing of python scripts in the portal that is why we need to use an external code editor such as Visual Studio Code for that purpose. One important thing to note is since the processing is not done in the Data Factory but in an external app, once a process is started it cannot be stopped. So, it is very important to consider which processes will be run before running them.