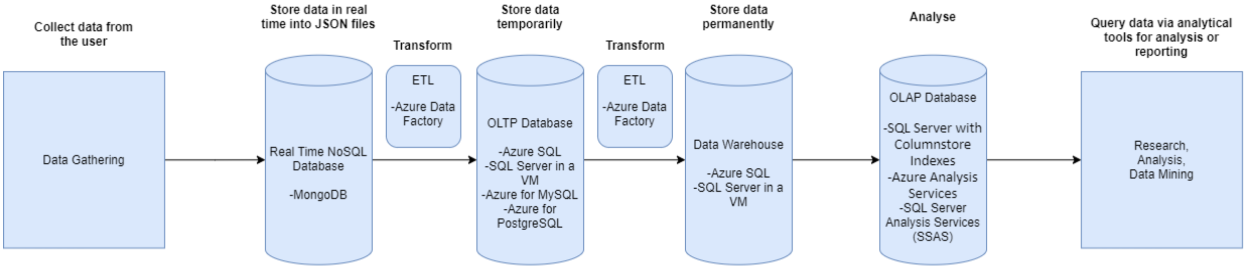
ETL Data Processing

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:

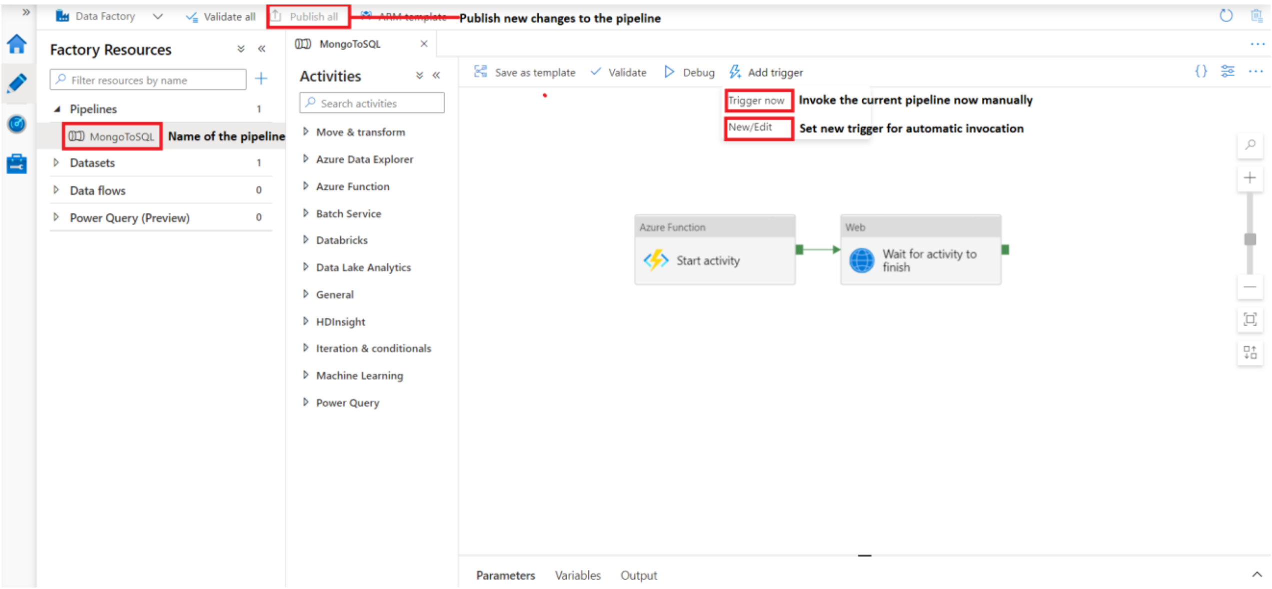


2. Manual to Azure Data Factory

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

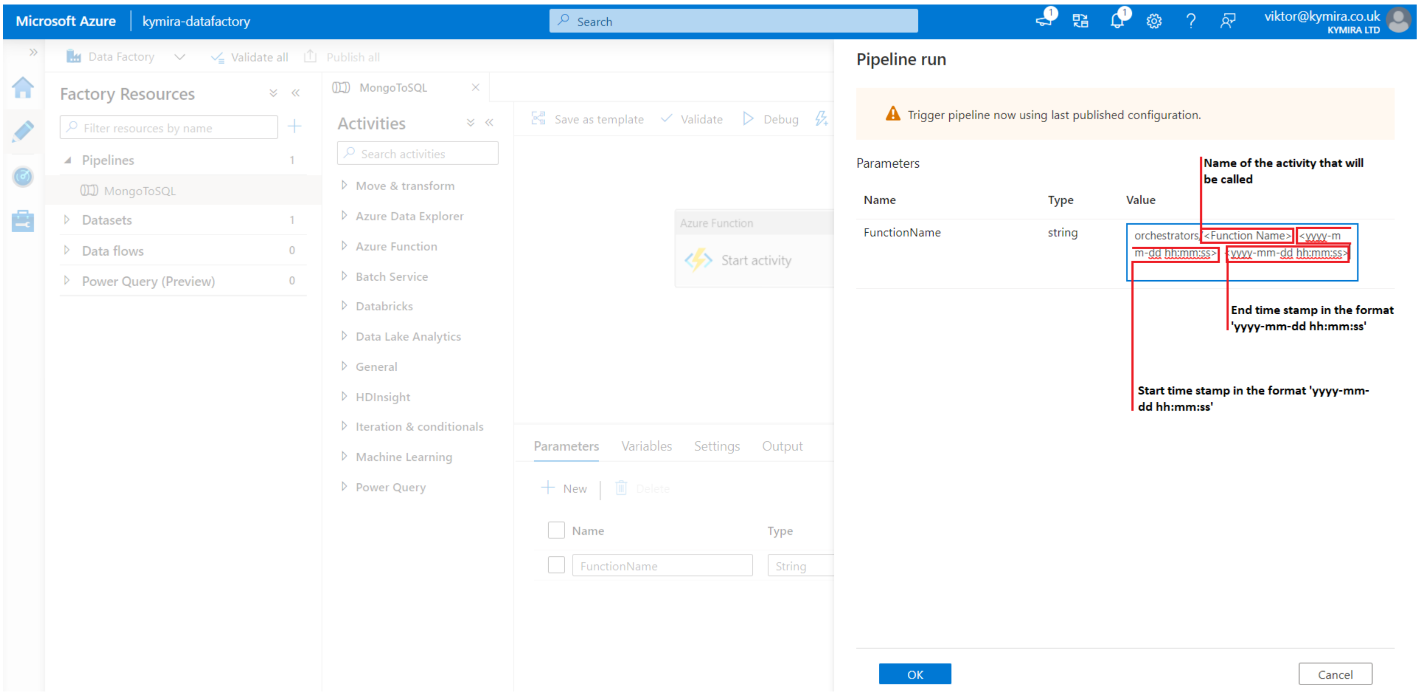
We also need to provide two other arguments. The start time stamp in the format ‘yyyy-mm-dd hh:mm:ss’ and the end time stamp in the same format.


Publish new changes to the pipeline

Name of the pipeline

Invoke the current pipeline now manually

Set new trigger for automatic invocation



Pipeline run

Trigger pipeline now using last published configuration.

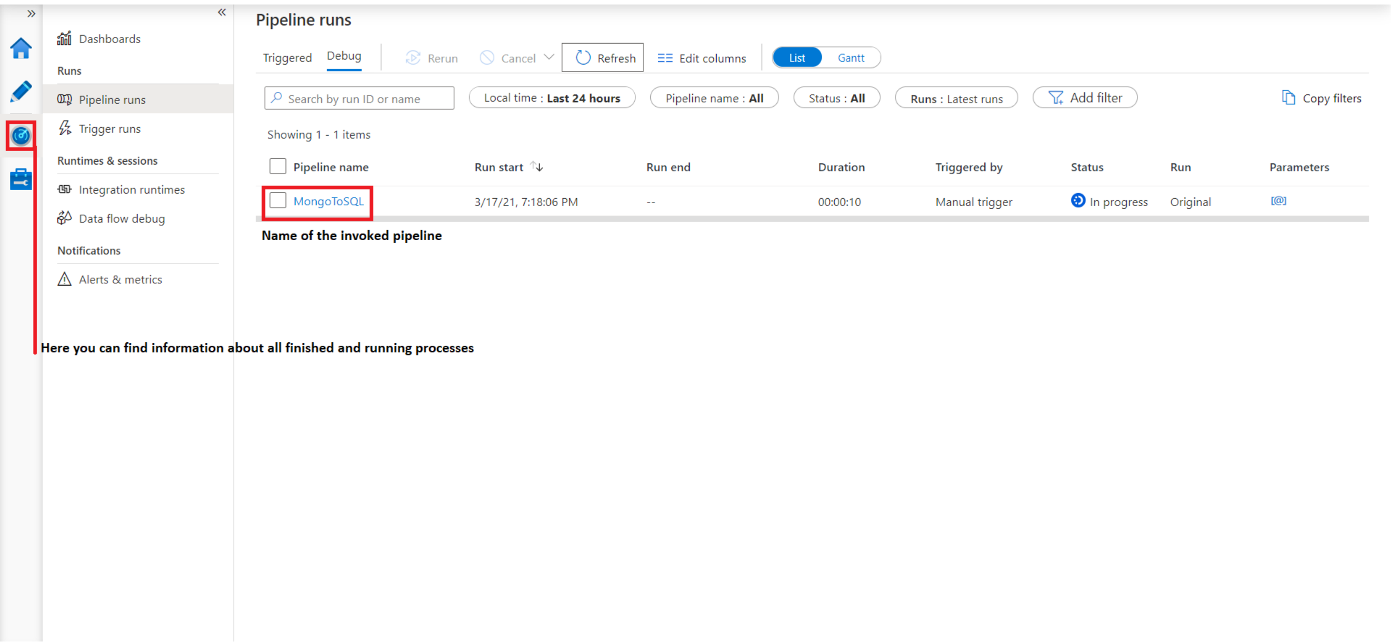
Parameters

Name	Type	Value
FunctionName	string	orchestrators/<Function Name> <yyyy-mm-dd hh:mm:ss> <yyyy-mm-dd hh:mm:ss>

Name of the activity that will be called

End time stamp in the format 'yyyy-mm-dd hh:mm:ss'

Start time stamp in the format 'yyyy-mm-dd hh:mm:ss'



Pipeline runs

Showing 1 - 1 items

Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run	Parameters
MongoToSQL	3/17/21, 7:18:06 PM	--	00:00:10	Manual trigger	In progress	Original	[@]

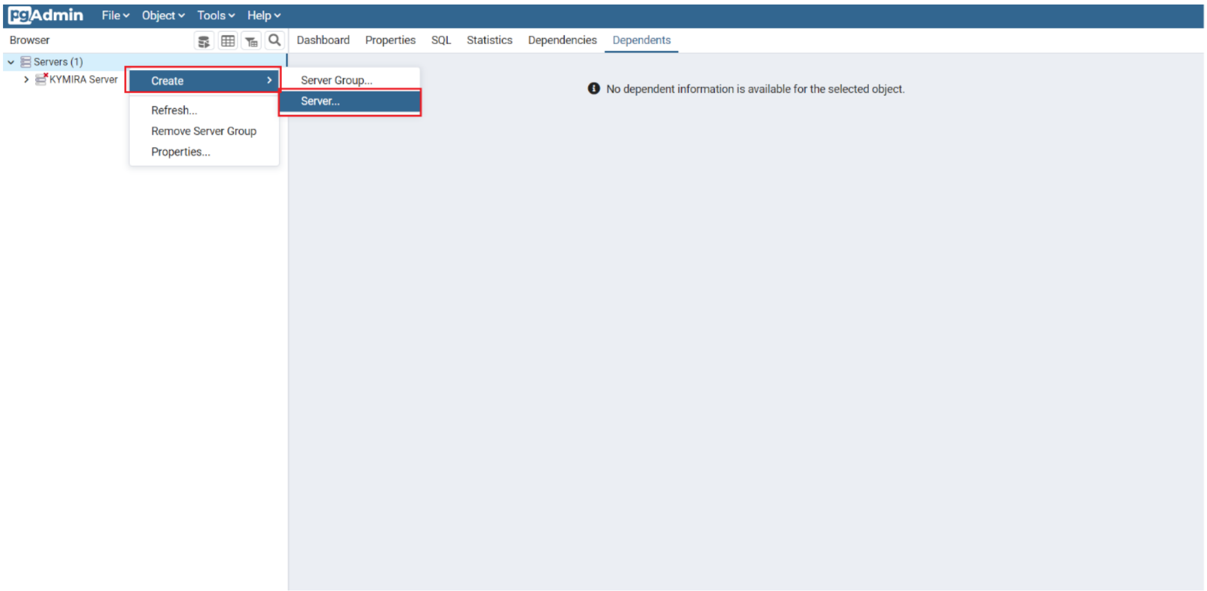
Name of the invoked pipeline

Here you can find information about all finished and running processes

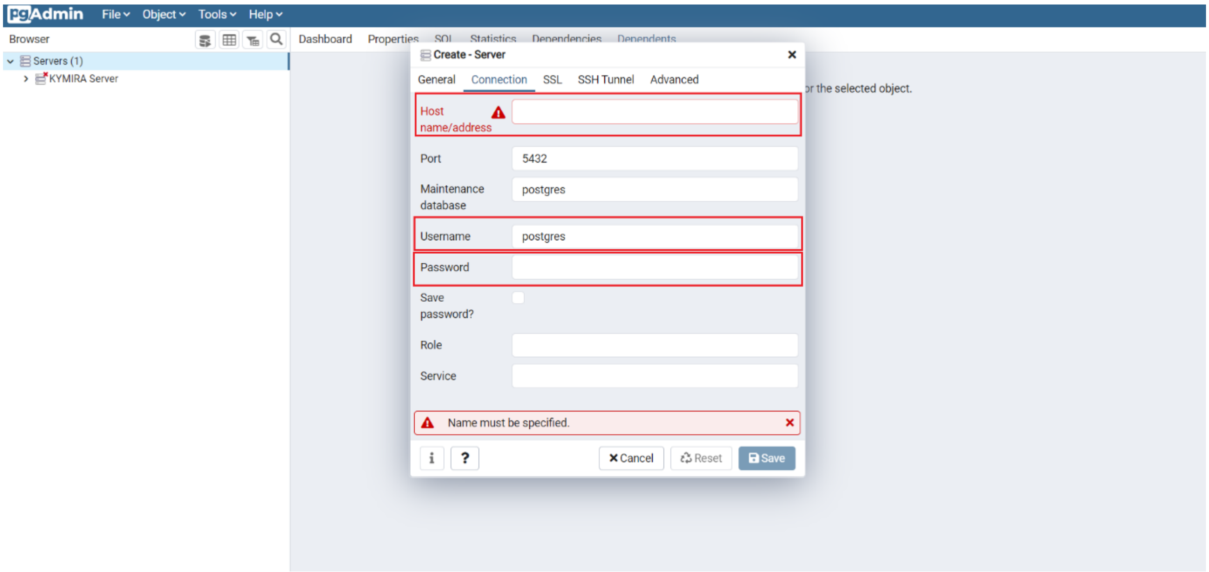
3. SQL database

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.



No dependent information is available for the selected object.



Host name/address:

Port: 5432

Maintenance database: postgres

Username: postgres

Password:

Save password? □

Role:

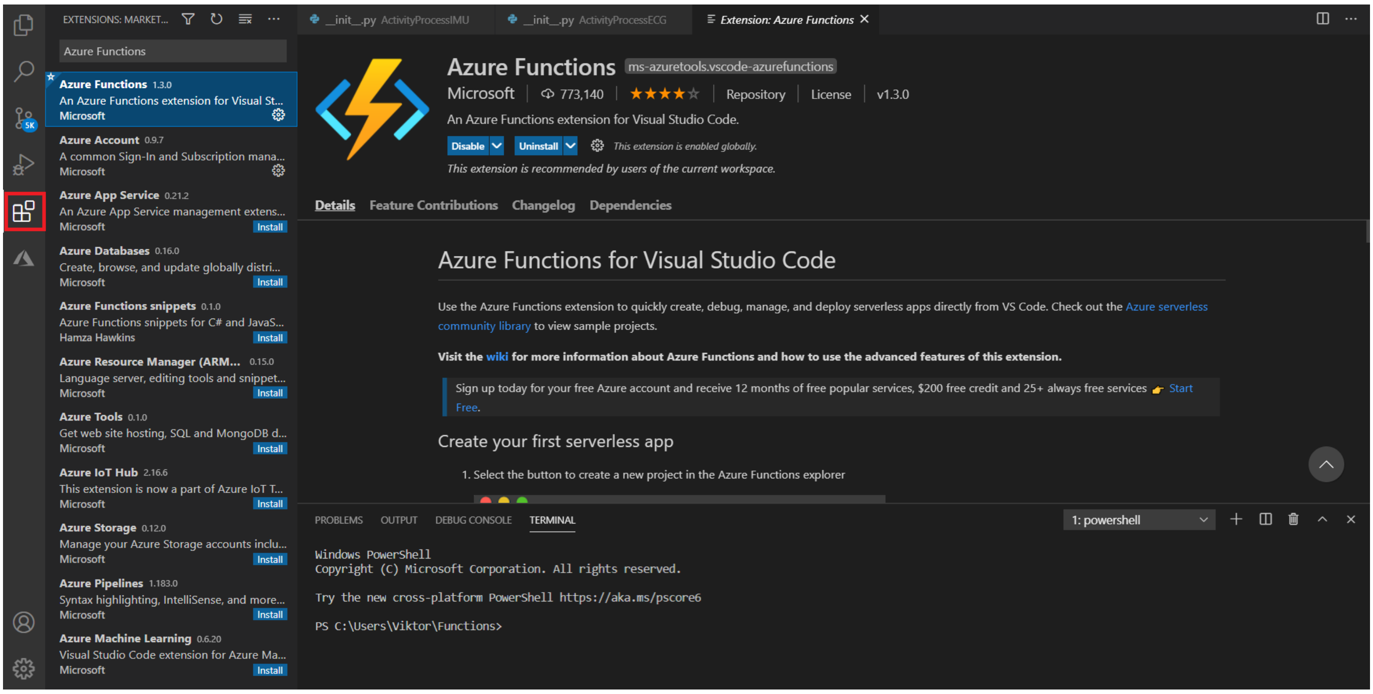
Service:

Name must be specified.

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’.


EXTENSIONS: MARKET...
Azure Functions
Azure Functions 1.3.0 An Azure Functions extension for Visual St... Microsoft
Azure Account 0.9.7 A common Sign-In and Subscription mana... Microsoft
Azure App Service 0.21.2 An Azure App Service management extens... Microsoft Install
Azure Databases 0.16.0 Create, browse, and update globally distri... Microsoft Install
Azure Functions snippets 0.1.0 Azure Functions snippets for C# and JavaS... Hamza Hawkins Install
Azure Resource Manager (ARM... 0.15.0 Language server, editing tools and snippet... Microsoft Install
Azure Tools 0.1.0 Get web site hosting, SQL and MongoDB d... Microsoft Install
Azure IoT Hub 2.16.6 This extension is now a part of Azure IoT T... Microsoft Install
Azure Storage 0.12.0 Manage your Azure Storage accounts inclu... Microsoft Install
Azure Pipelines 1.183.0 Syntax highlighting, IntelliSense, and more... Microsoft Install
Azure Machine Learning 0.6.20 Visual Studio Code extension for Azure Ma... Microsoft Install
__init__.py ActivityProcessIMU
__init__.py ActivityProcessECG
Extension: Azure Functions
Azure Functions ms-azuretools.vscode-azurefunctions
Microsoft 773,140 ★★★★☆ Repository License v1.3.0
Disable Uninstall
Details Feature Contributions Changelog Dependencies
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell

An Azure Functions extension for Visual Studio Code.

This extension is enabled globally.

This extension is recommended by users of the current workspace.

Azure Functions for Visual Studio Code

Use the Azure Functions extension to quickly create, debug, manage, and deploy serverless apps directly from VS Code. Check out the Azure serverless community library to view sample projects.

Visit the wiki for more information about Azure Functions and how to use the advanced features of this extension.

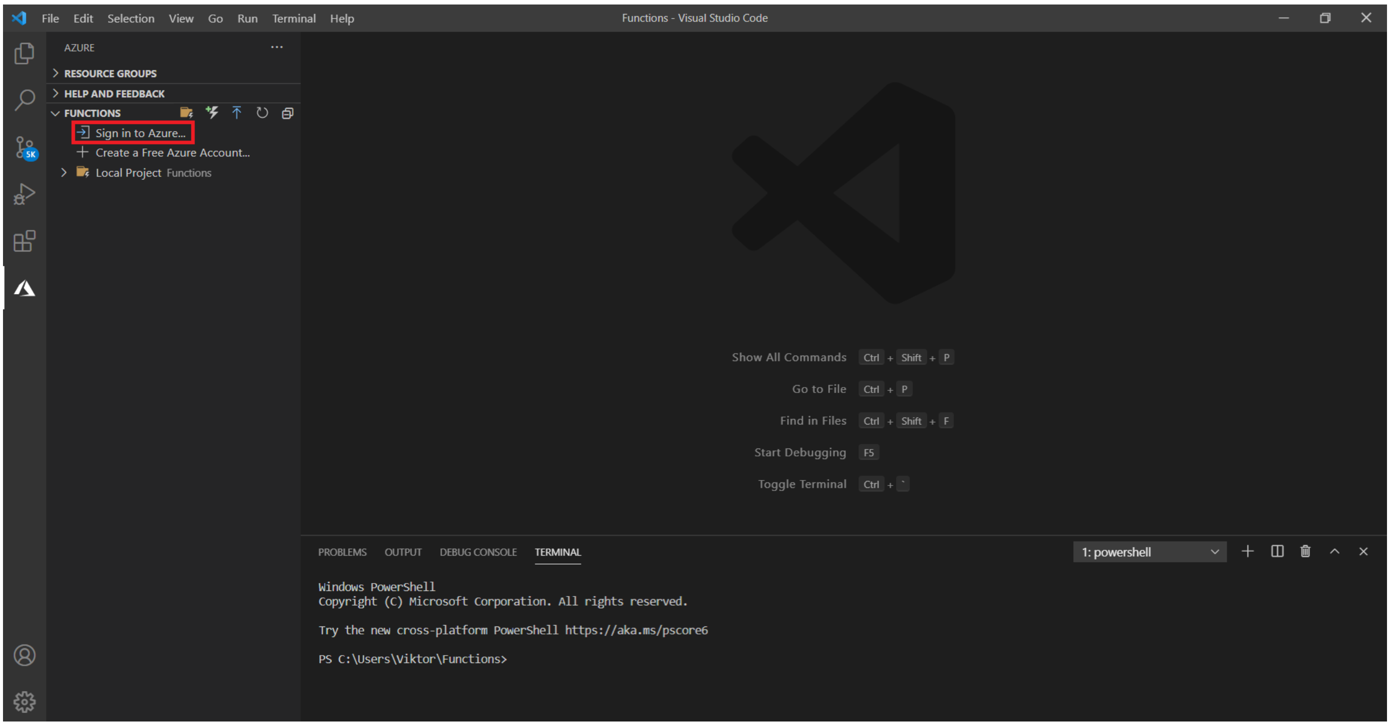
Sign up today for your free Azure account and receive 12 months of free popular services, $200 free credit and 25+ always free services 👉 Start Free.

Create your first serverless app

1. Select the button to create a new project in the Azure Functions explorer

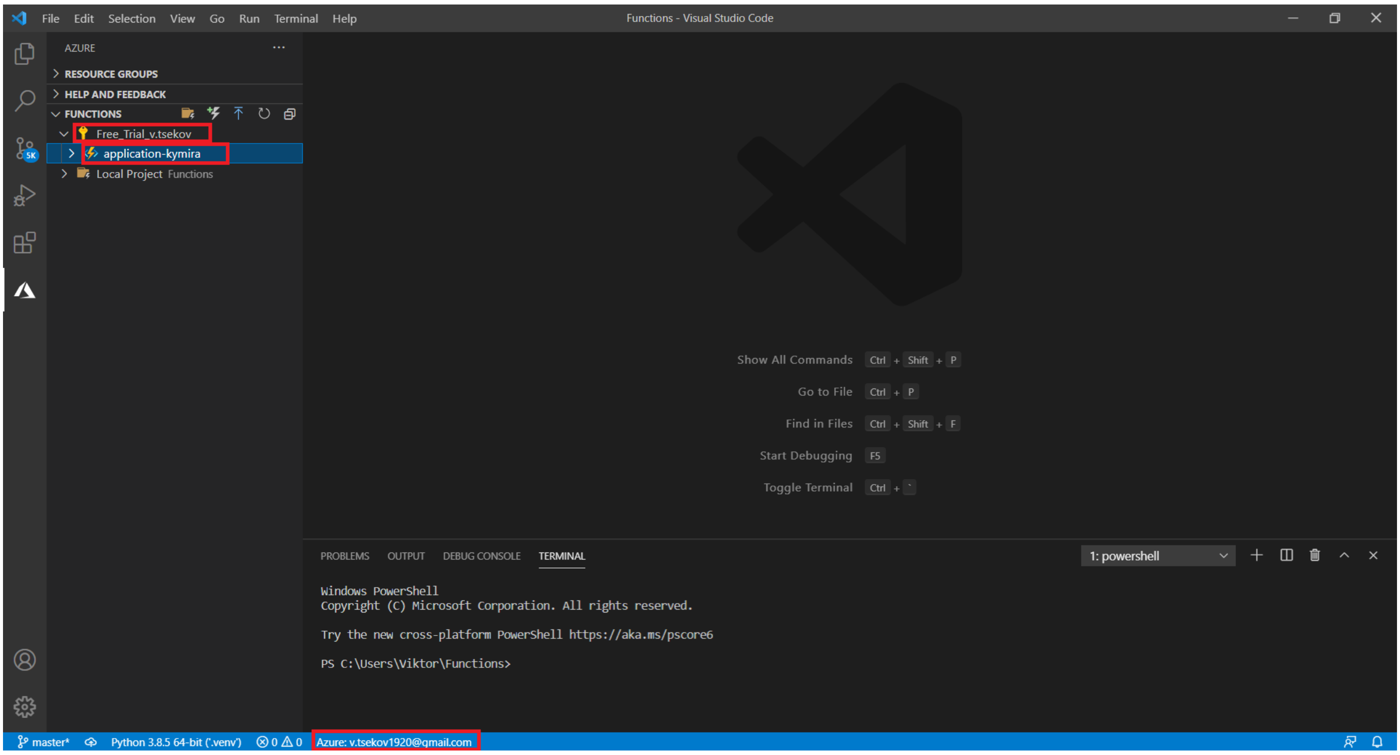
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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


File Edit Selection View Go Run Terminal Help
Functions - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Sign in to Azure...
Create a Free Azure Account...
Local Project Functions
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell

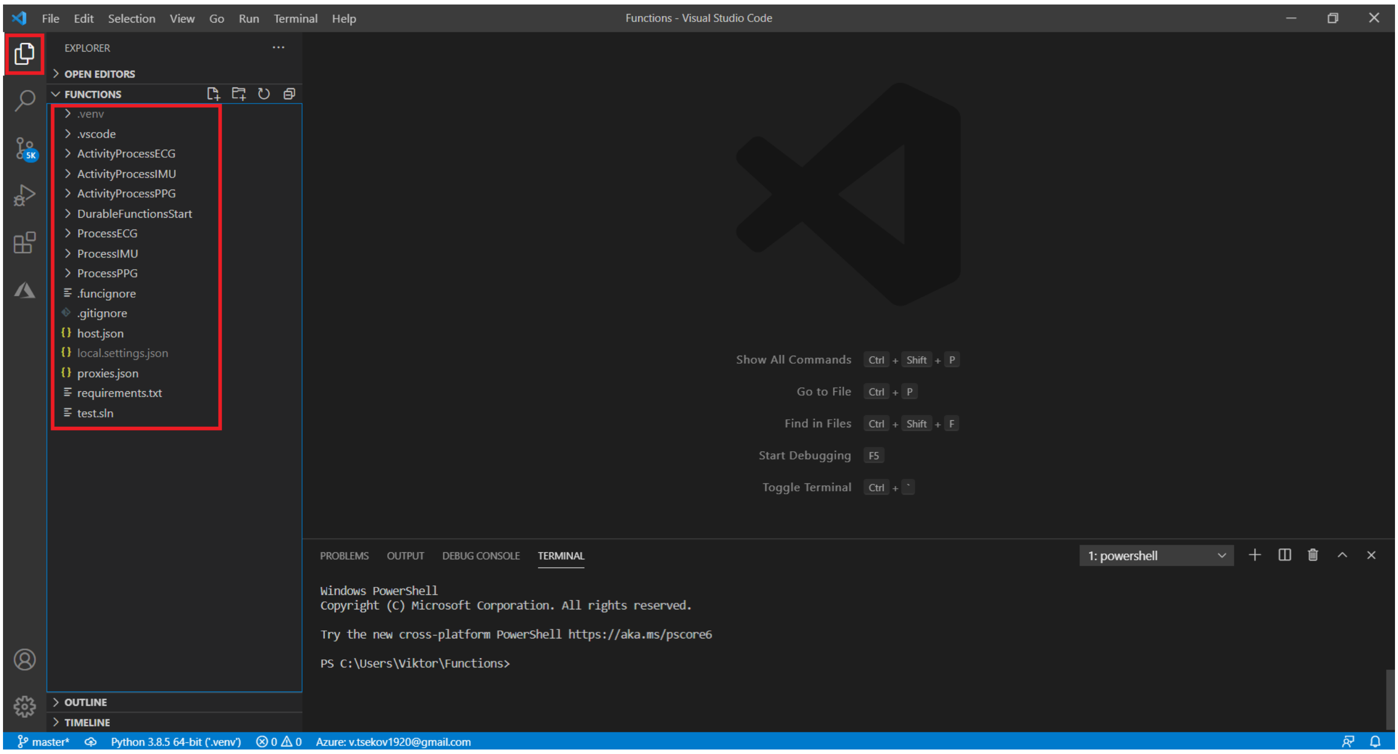
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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.


File Edit Selection View Go Run Terminal Help
Functions - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Free_Trial_v.tsekov
application-kymira
Local Project Functions
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
master* Python 3.8.5 64-bit ('.venv') ⓧ 0 ⚠ 0 Azure: v.tsekov1920@gmail.com

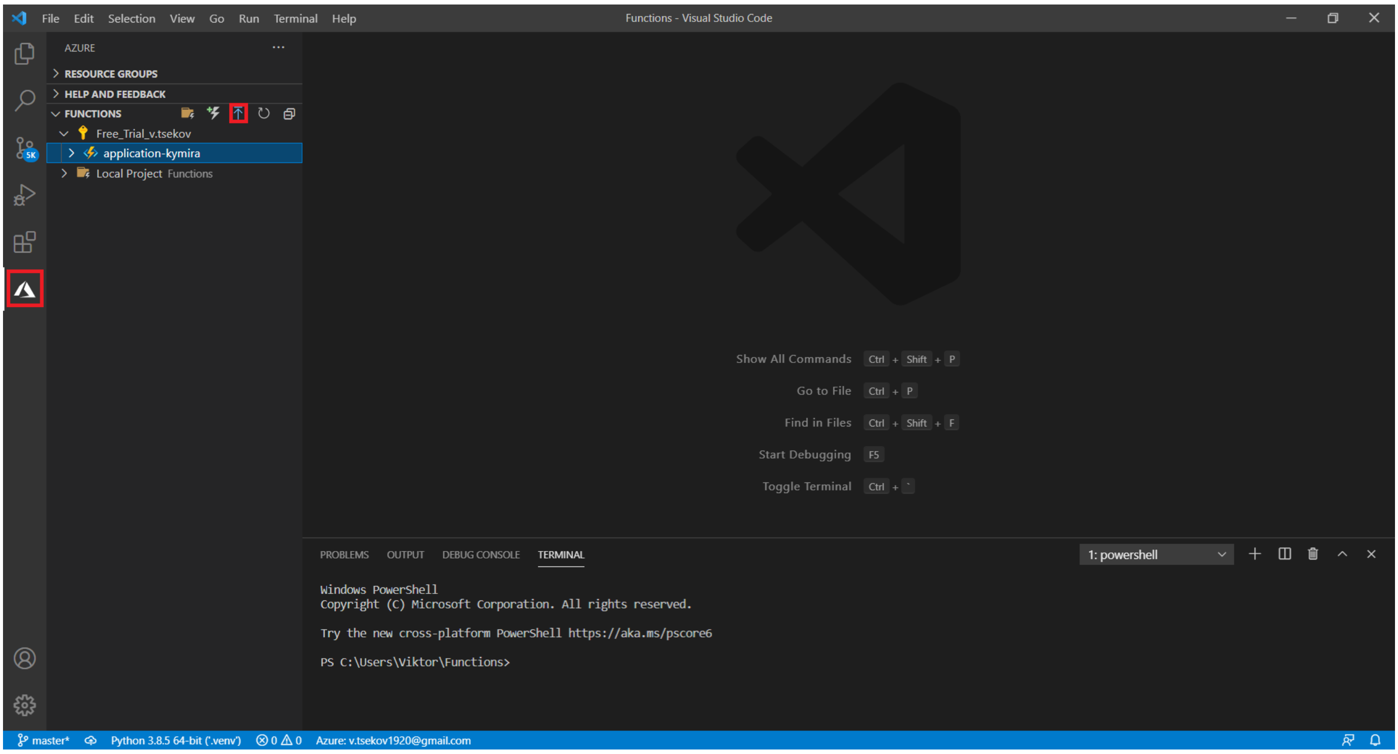
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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.

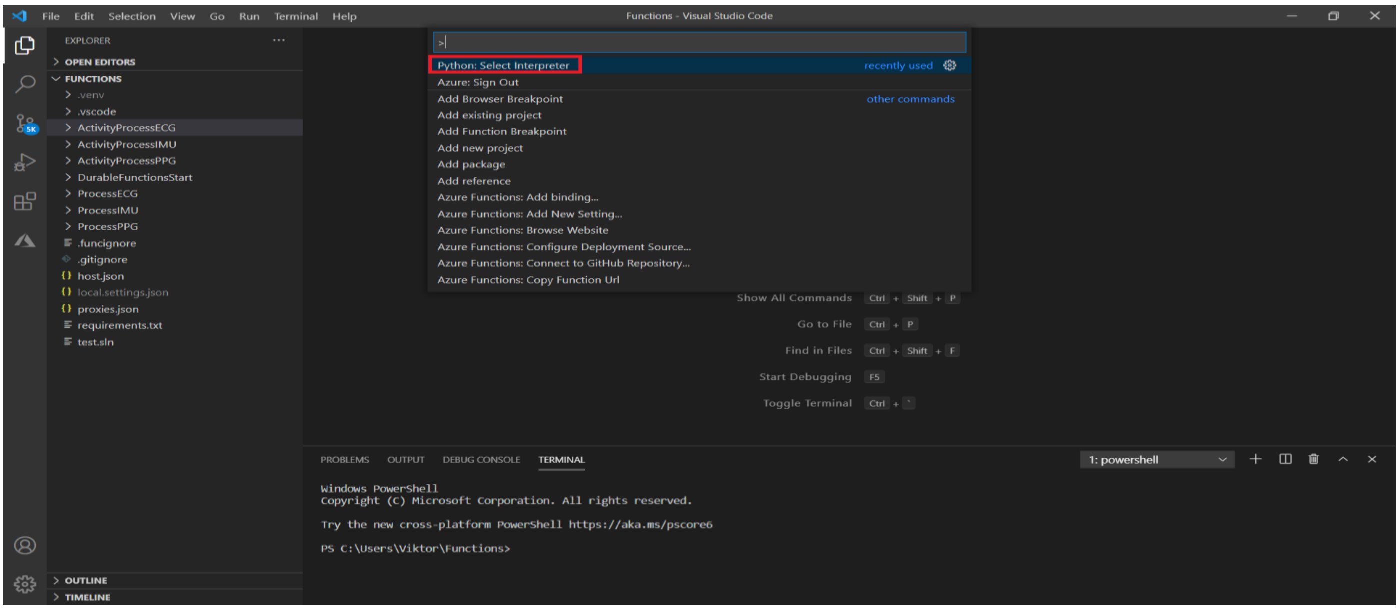

File Edit Selection View Go Run Terminal Help
Functions - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Free_Trial_v.tsekov
application-kymira
Local Project Functions
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
master* Python 3.8.5 64-bit ('.venv') ⊗ 0 ⚠ 0 Azure: v.tsekov1920@gmail.com

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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

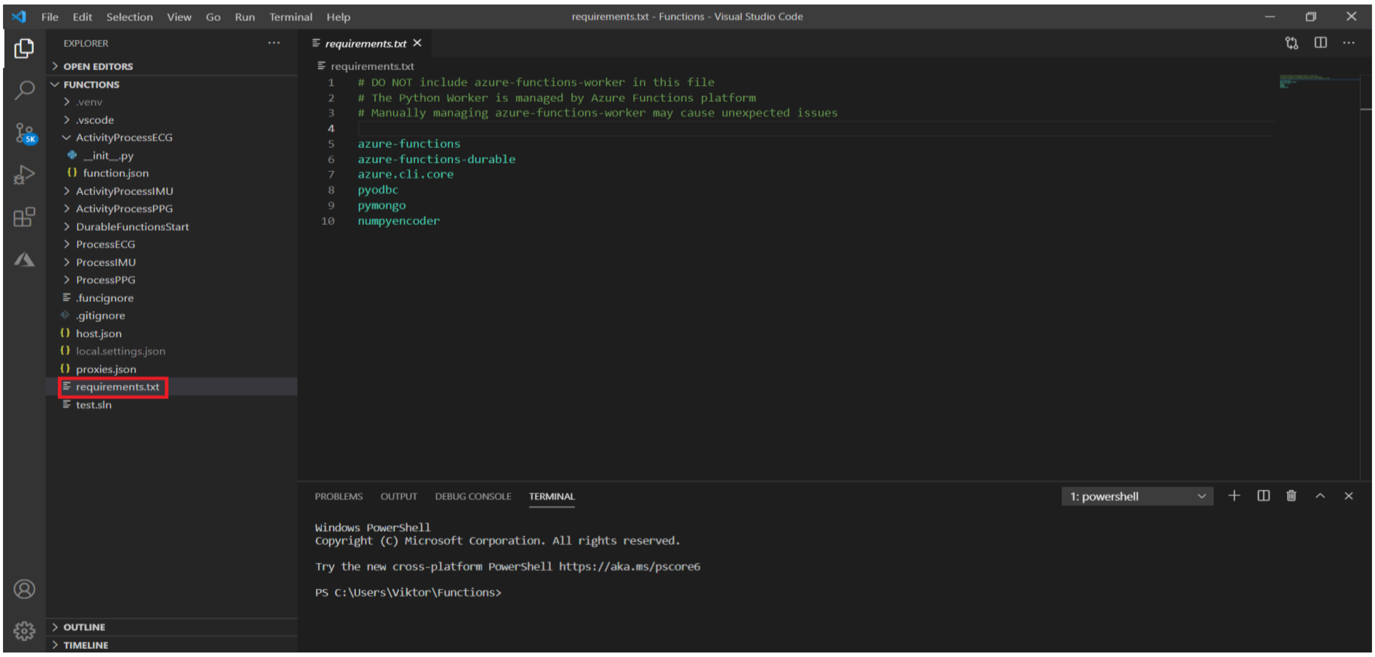
5. Requirements and setup

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.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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.



```
# DO NOT include azure-functions-worker in this file
# The Python Worker is managed by Azure Functions platform
# Manually managing azure-functions-worker may cause unexpected issues

azure-functions
azure-functions-durable
azure.cli.core
pyodbc
pymongo
numpyencoder
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Viktor\Functions>
```

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.