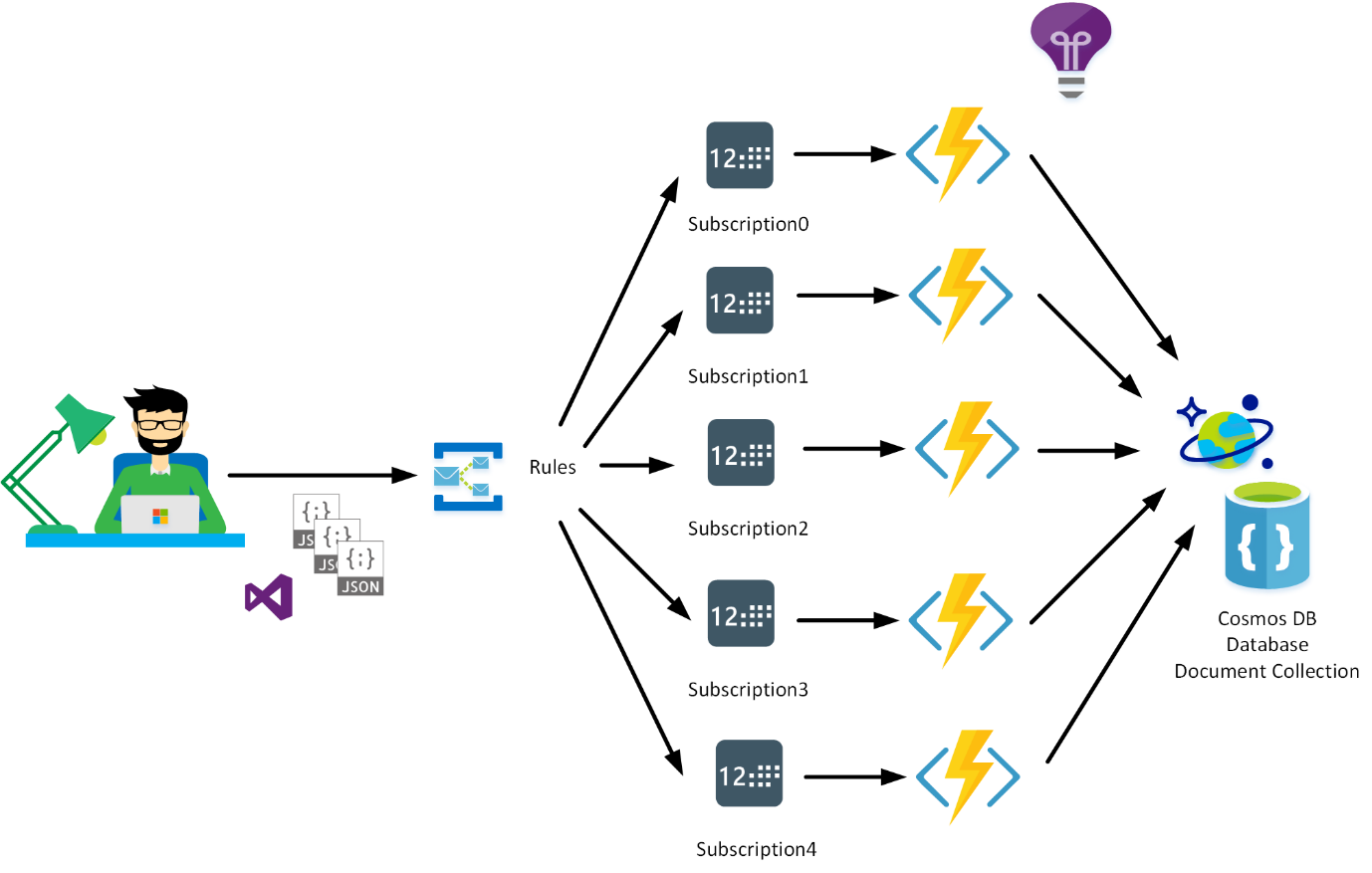
**Deploy a flask app configured with Azure functions configured with Azure MongoDB API in Cosmos DB**

**Introduction**

Flask Application is Deployed in Azure App Service and database is hosted in Azure MongoDb API of Cosmos DB. Azure Functions are acting as a middleware and mantaining end to end connections b/w them.

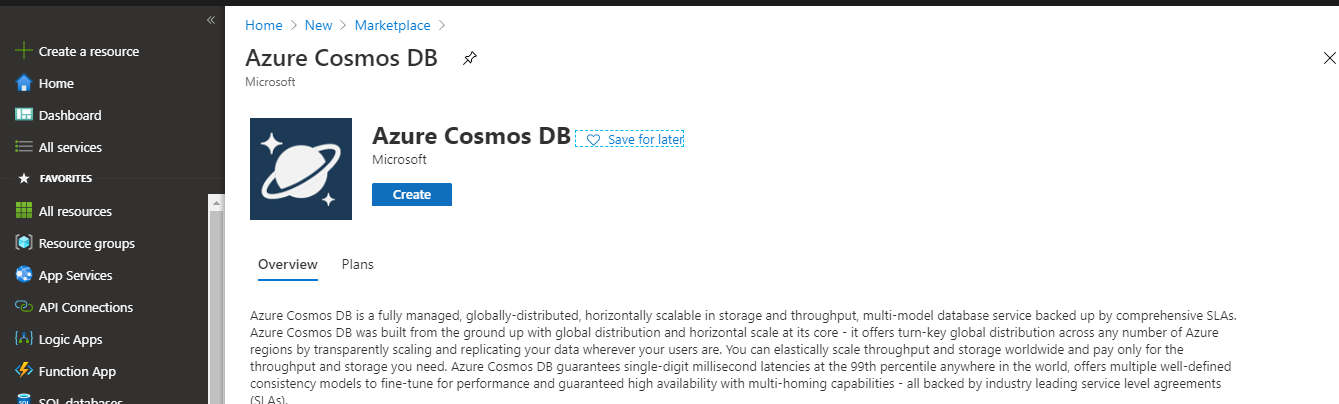


**Prerequisites:**

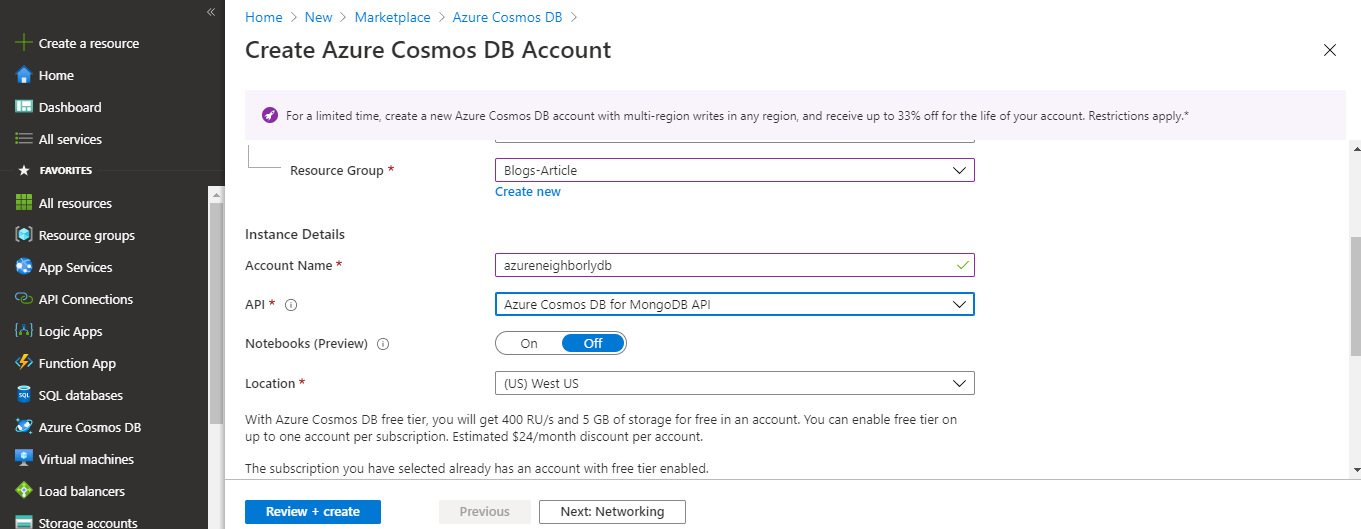
* You need to have an active Azure subscription. If you not have you can sign-up for a free [account](https://azure.microsoft.com/en-in/free/).
* (optional) Studio3T a software installed on your PC in order to manage the Azure MongoDB Database. You can get it from [here](https://studio3t.com/download/).
* (optional) Download [Source Code](https://github.com/DhruvKinger/Neighborly-App-Azure) to follow up with the exercise.

**Task 1**: Setup a Cosmos DB Account with MongoDB API.

* Navigate to the [Azure Portal](https://portal.azure.com/).
* Go to the existing Resource group or create a new one.
* Click on create a new resource. In Azure Market place search Cosmos DB.



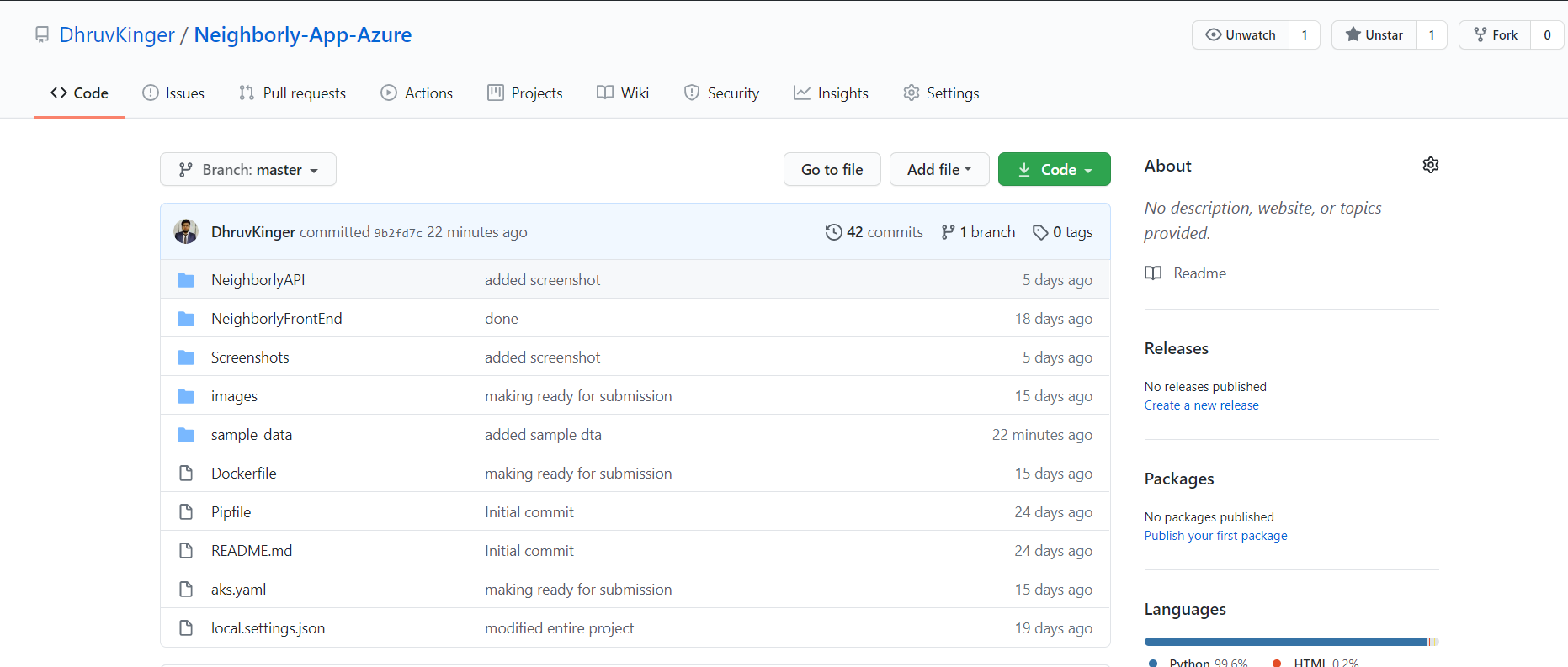
* Now it will prompt you to a page and ask for the required fields. You need to fill in those details.
* Main thing here to consider is you to have choose MongoDB API. Other fields can be filled as you wish.



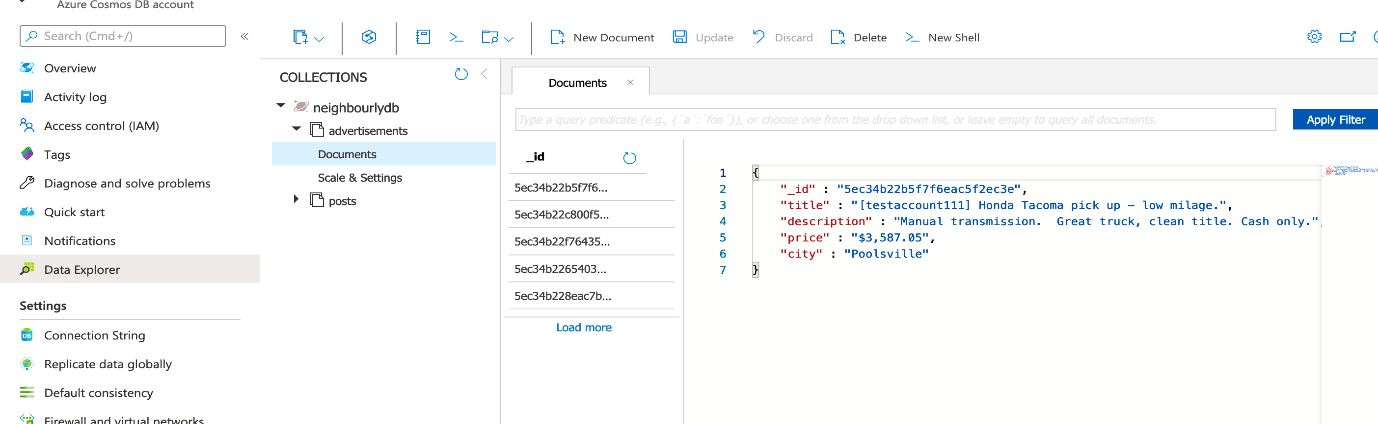
* It can take more than 15 minutes for account creation. Once it is deployed. You can click on data explorer where you can feed in your data.
* In order to import data you can follow 2 methods first one is through Azure Portal and other one is through the Studio 3T.
* Here are the links you can choose any of them.

1. [Azure Notebooks.](https://docs.microsoft.com/en-us/azure/notebooks/access-data-resources-jupyter-notebooks)
2. [Studio 3T.](https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb-mongochef)

* Json data needed to import in the Collections can be found be in the “sample\_data” folder provided in the source code. See Below.



* **Note**: After deployment you can have something like this in Data Explorer.

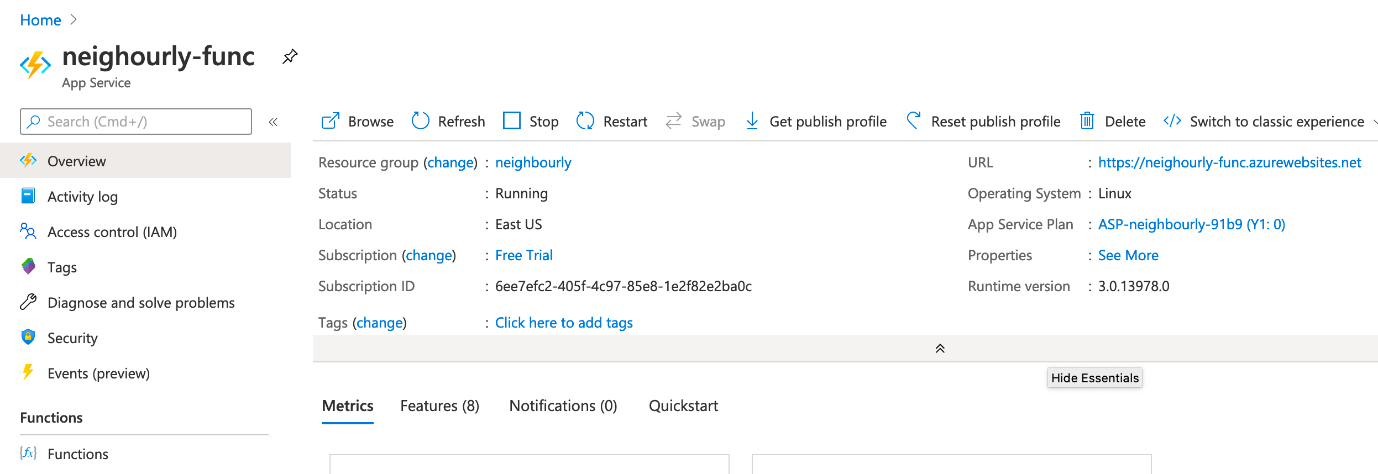


So, our 1st task is completed.

**Task 2**: Create an Azure Function and connect in to Azure Cosmos DB you just created above.

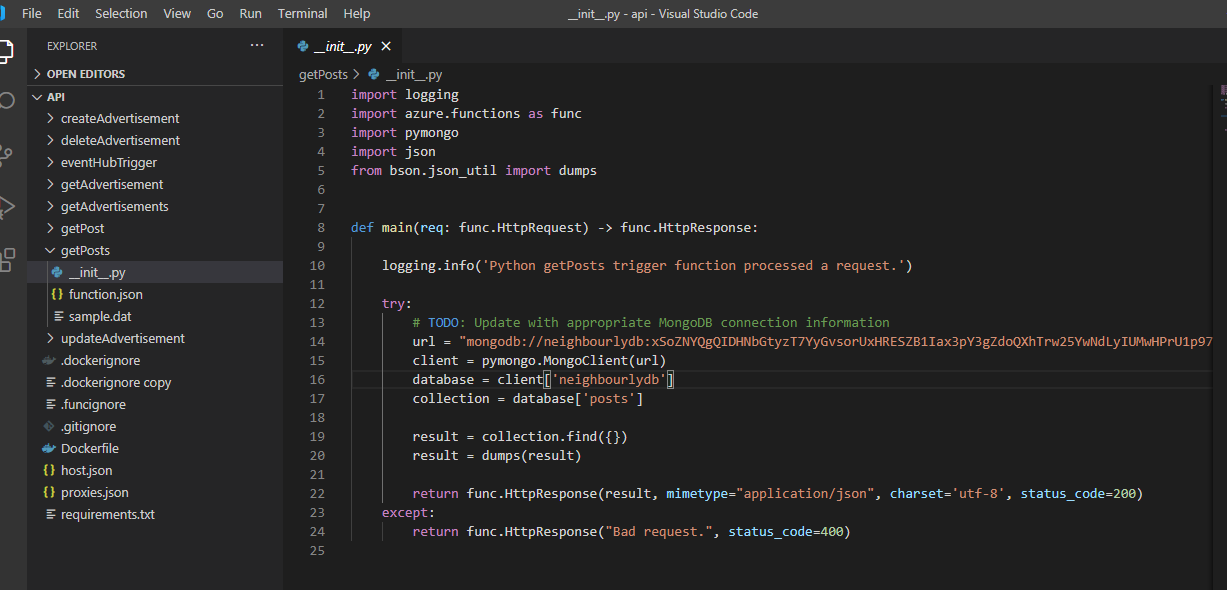
**Note**: You can create and deploy it with Visual Studio Code as well.

* Click on create a new resource in the same resource group and select Function App.
* You can create functions for Http Trigger, Service Bus queue and many more. You can edit them in Azure as well except for python function.
* After function app is created you can have its endpoint.

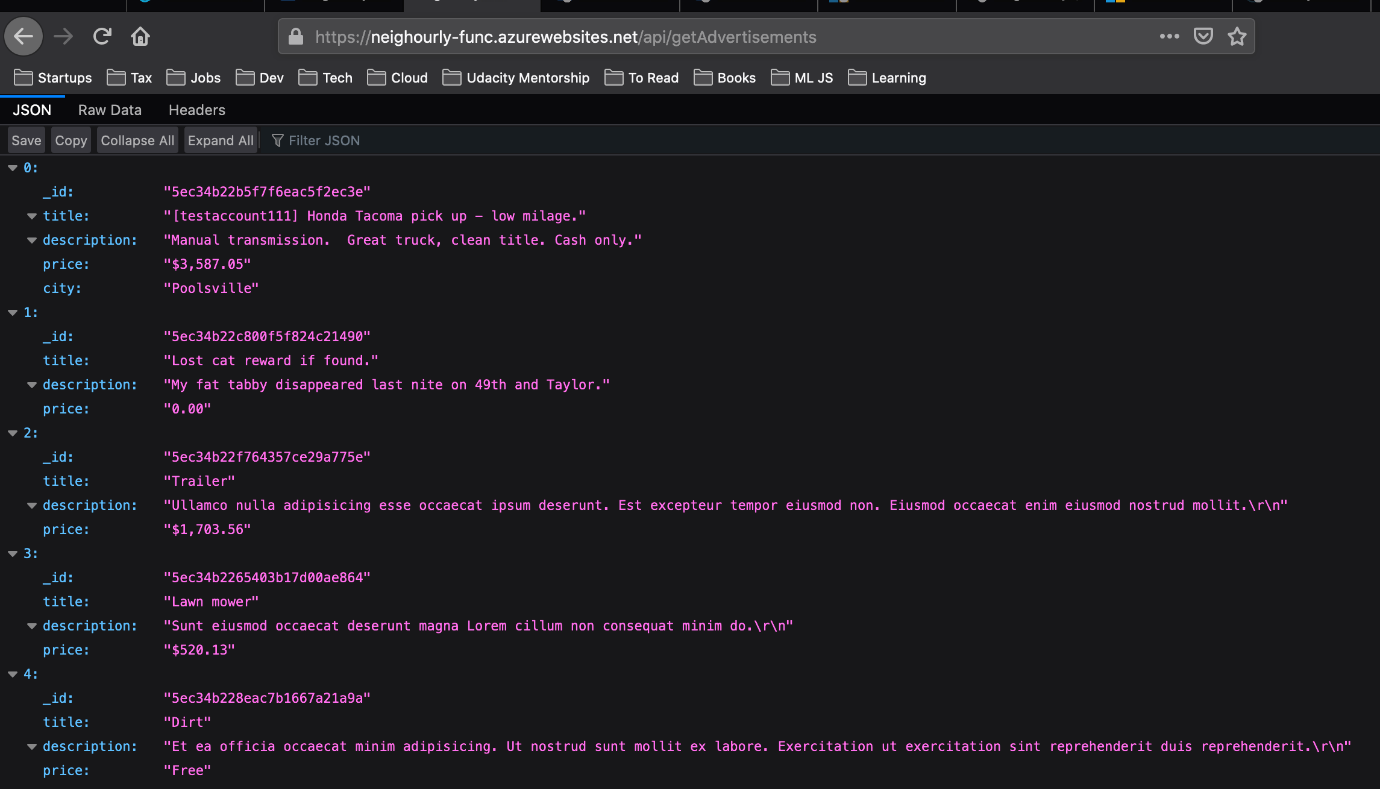


**Note**: You can edit in Visual Studio Code. Now in our case we have to connect it to Mongo DB. So, our Code looks something like this.

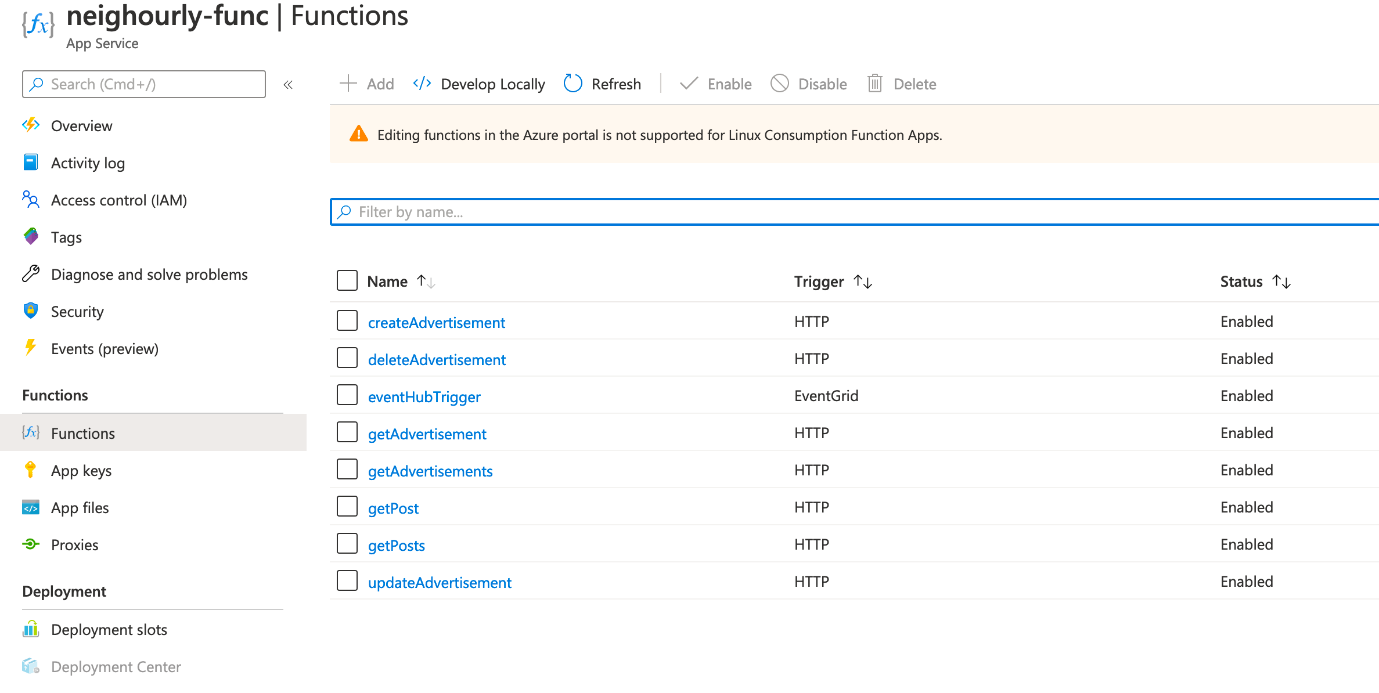
* After creating it we can run it locally for testing purpose and deploy it back to Azure with some changes.
* Here is the [link](https://docs.microsoft.com/en-us/azure/azure-functions/functions-develop-vs-code?tabs=csharp) through which you can take some help.
* Code is provided in the ‘NeighborlyAPI’ Folder of the code provided in Prerequisites.



* You can test it’s endpoint on Postman or you can check it on Browser like (Google Chrome, Mozilla Firefox) as well.



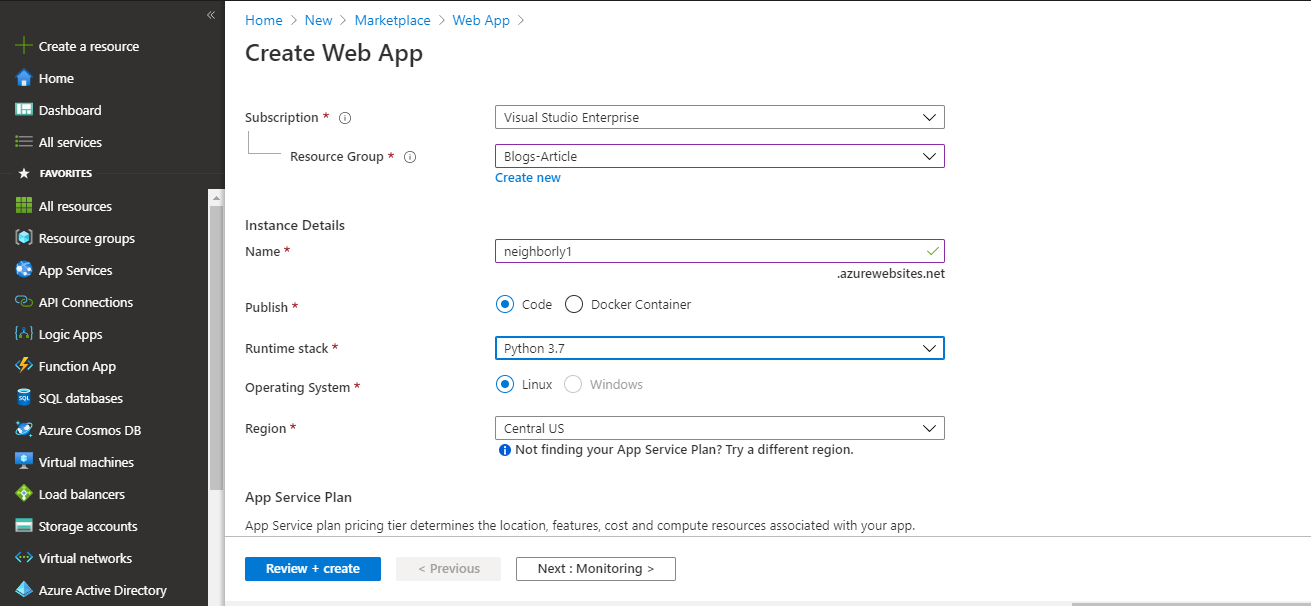
* Similarly you can deploy other functions as well. See how Azure Function looks.



So, our 2nd task is completed as well.

**Task 3**: Deploy a Flask App in Azure after configuring Azure Functions endpoint to it.

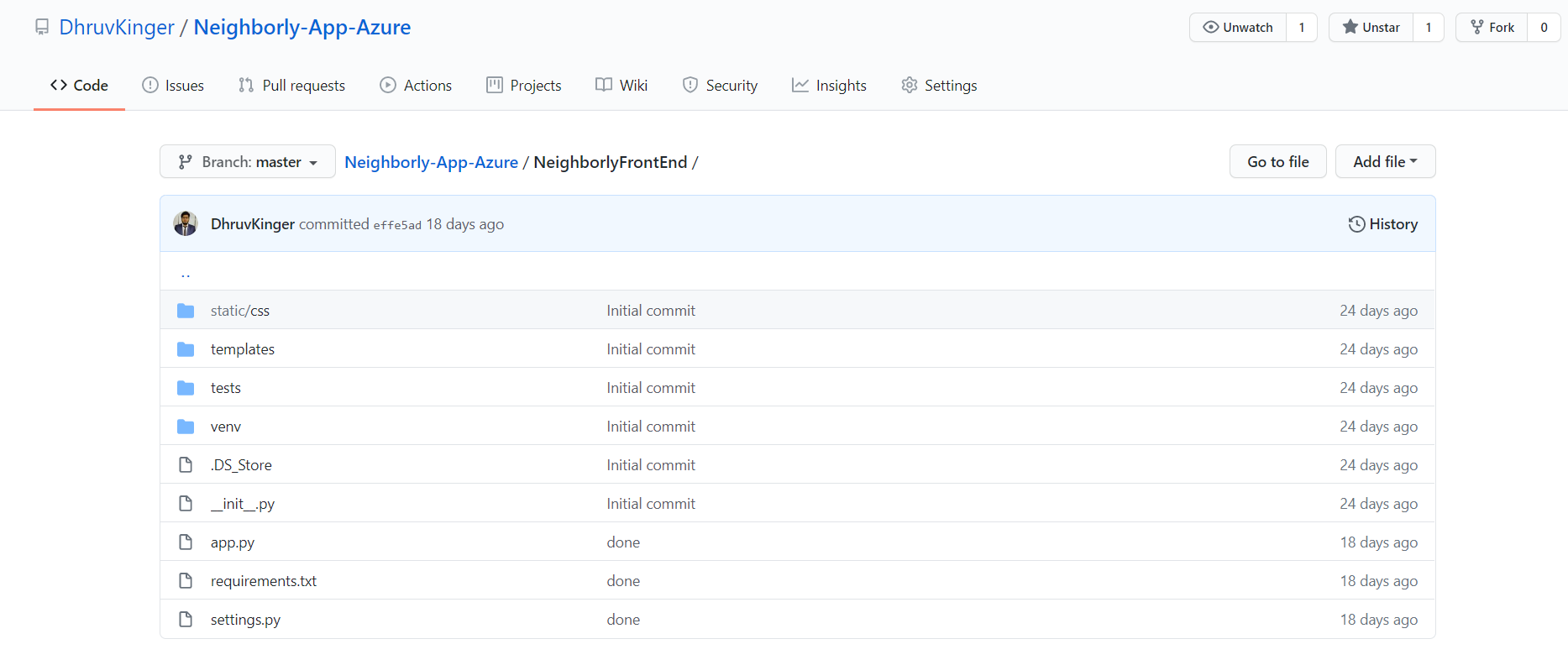
* Create a new resource in the resource group and search for azure web app. Select Create.
* Now fill in the details like App Name, Location, Pricing Tier and App Service Plan.

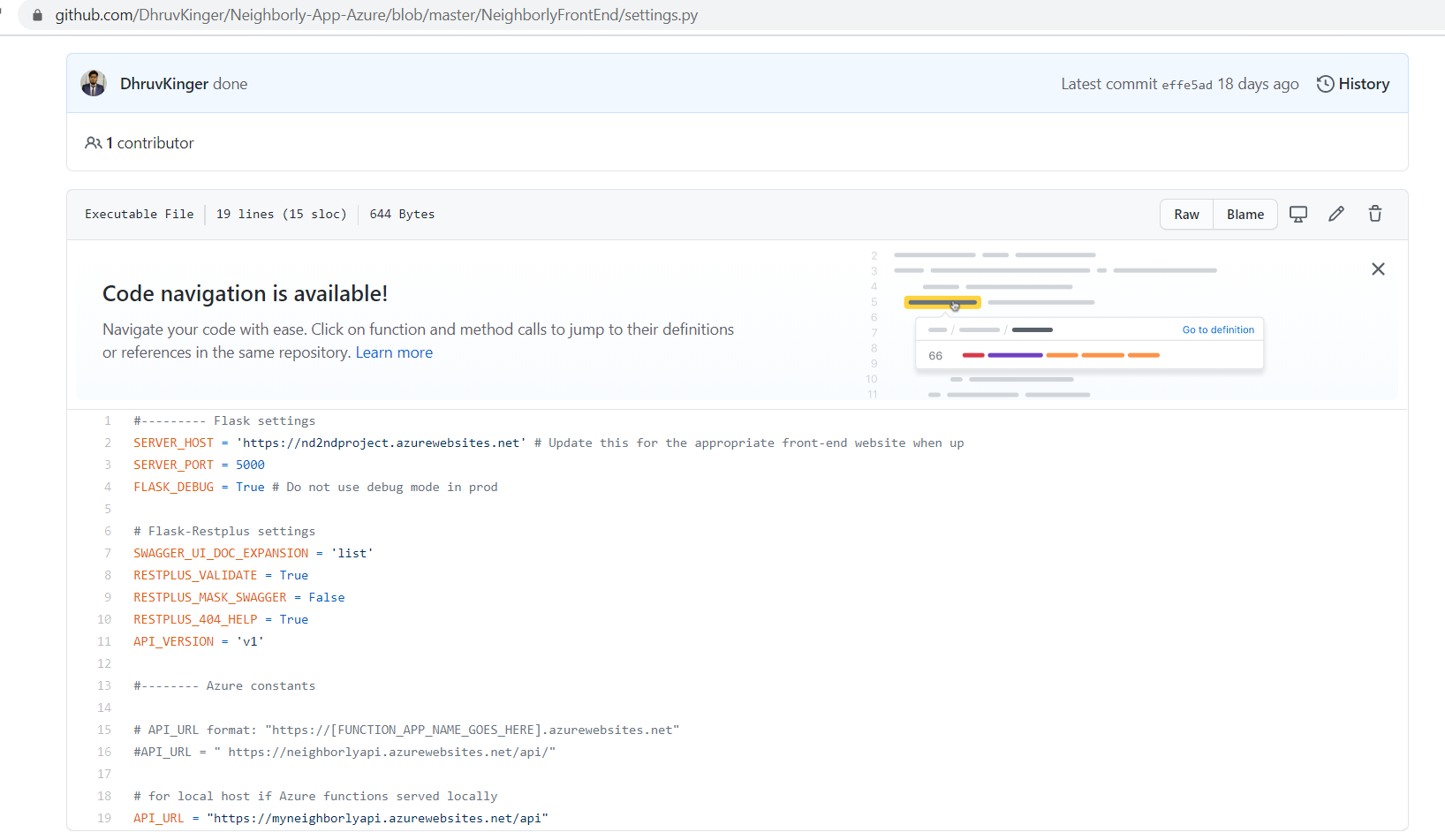


* Now in order deploy code to you can choose any method like FTP as well Git Pipelines.
* You can take some help from [this](https://docs.microsoft.com/en-us/azure/devops/pipelines/targets/webapp?view=azure-devops&tabs=yaml) link.

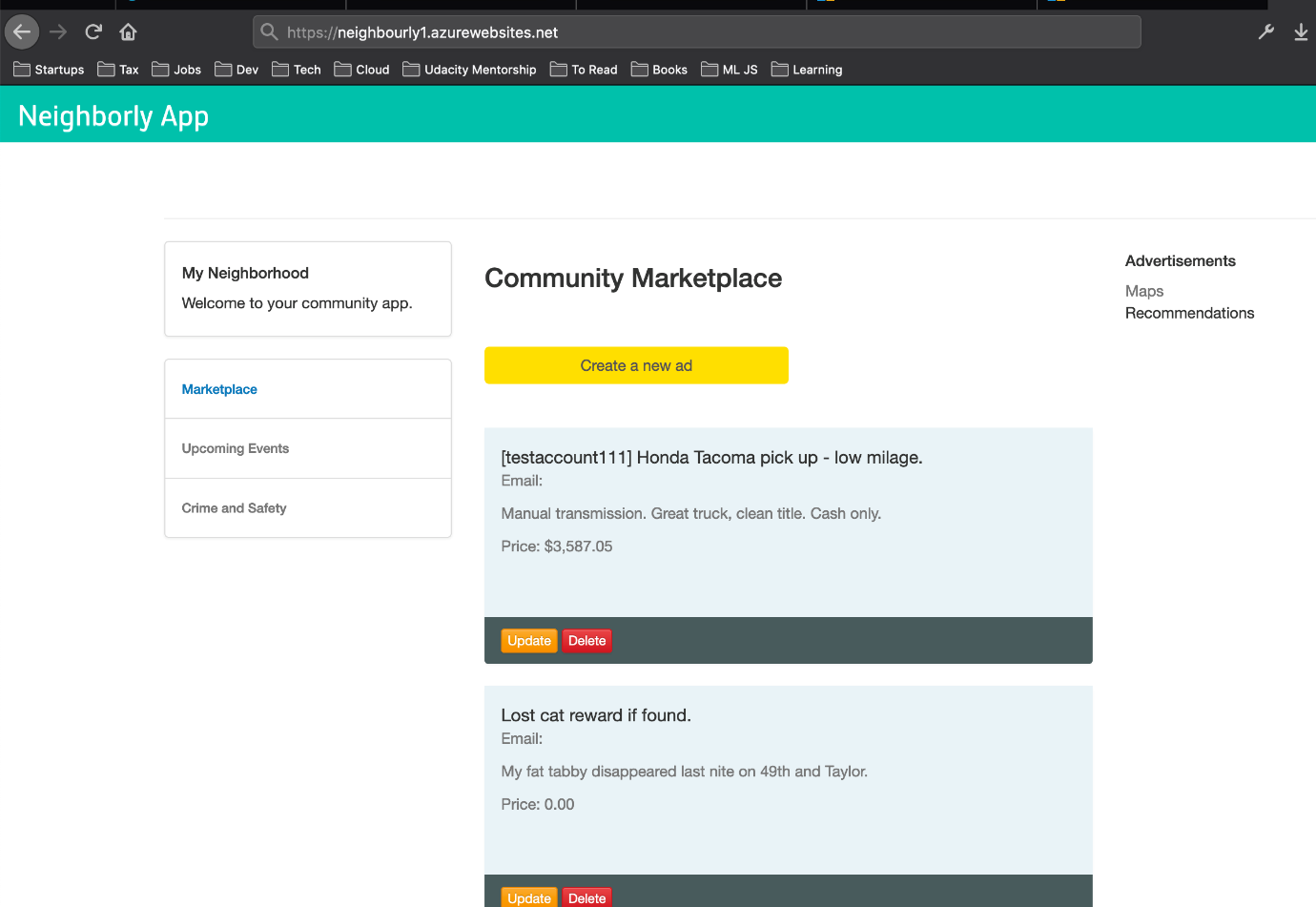
**Note**: In order to connect azure functions to Flask App we have to add functions app connection string to the app.

* Code for the Flask App is the ‘NeighborlyFrontEnd’ Folder in the code provided as mentioned in the Prerequisites.
* Folder Directory is described below. For you to deploy the code changes are required only in the settings.py file.
* SERVER\_HOST=”< name\_of\_your\_web\_app >.azurewebsites.net”
* API\_URL= “< Your\_function\_app\_url >.azurewebsites.net/api”
* Rest everything is fine!





* After deploying code, your app runs something like this.



**Note**: So, we have completed our task 3 as well. Here we are ready with the app running in the browser fetching data and showing images to screen because azure function endpoints are calling them.