Working with Azure Blob trigger

Step-1

Download Microsoft Azure Storage Explorer.

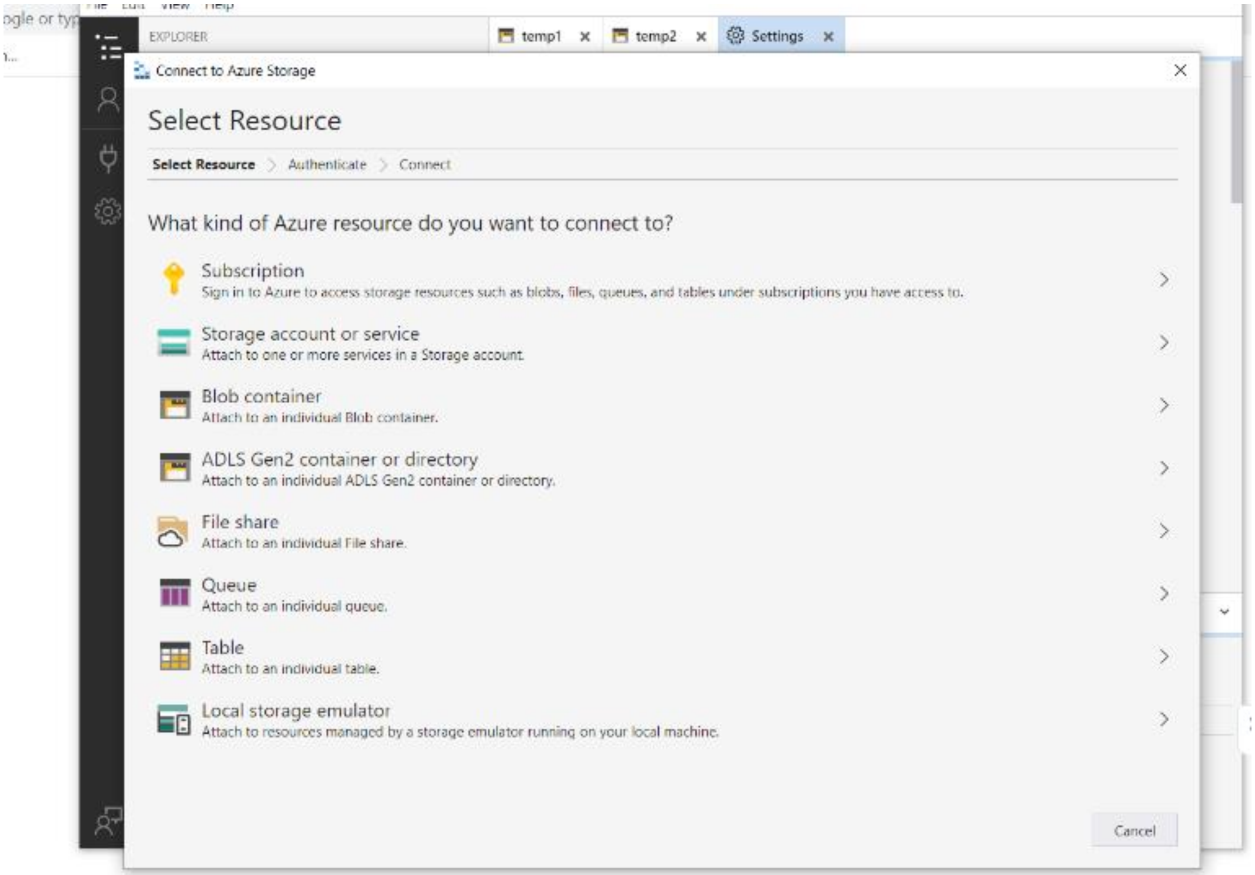
Src: https://azure.microsoft.com/en-us/features/storage-explorer/

Step-2

Configure Storage explorer by clicking storage account and service.

Here I used my student account.

Create 2 blob containers(temp1 and temp2) in Azure Blob containers.



Select Resource

Select Resource > Authenticate > Connect

What kind of Azure resource do you want to connect to?

Subscription
Sign in to Azure to access storage resources such as blobs, files, queues, and tables under subscriptions you have access to.

Storage account or service
Attach to one or more services in a Storage account.

Blob container
Attach to an individual Blob container.

ADLS Gen2 container or directory
Attach to an individual ADLS Gen2 container or directory.

File share
Attach to an individual File share.

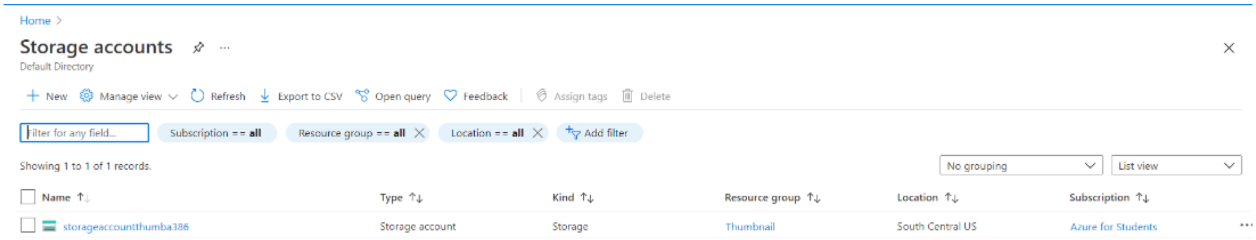
Queue
Attach to an individual queue.

Table
Attach to an individual table.

Local storage emulator
Attach to resources managed by a storage emulator running on your local machine.

Step-3:

By clicking new button, create new storage account for storing images.

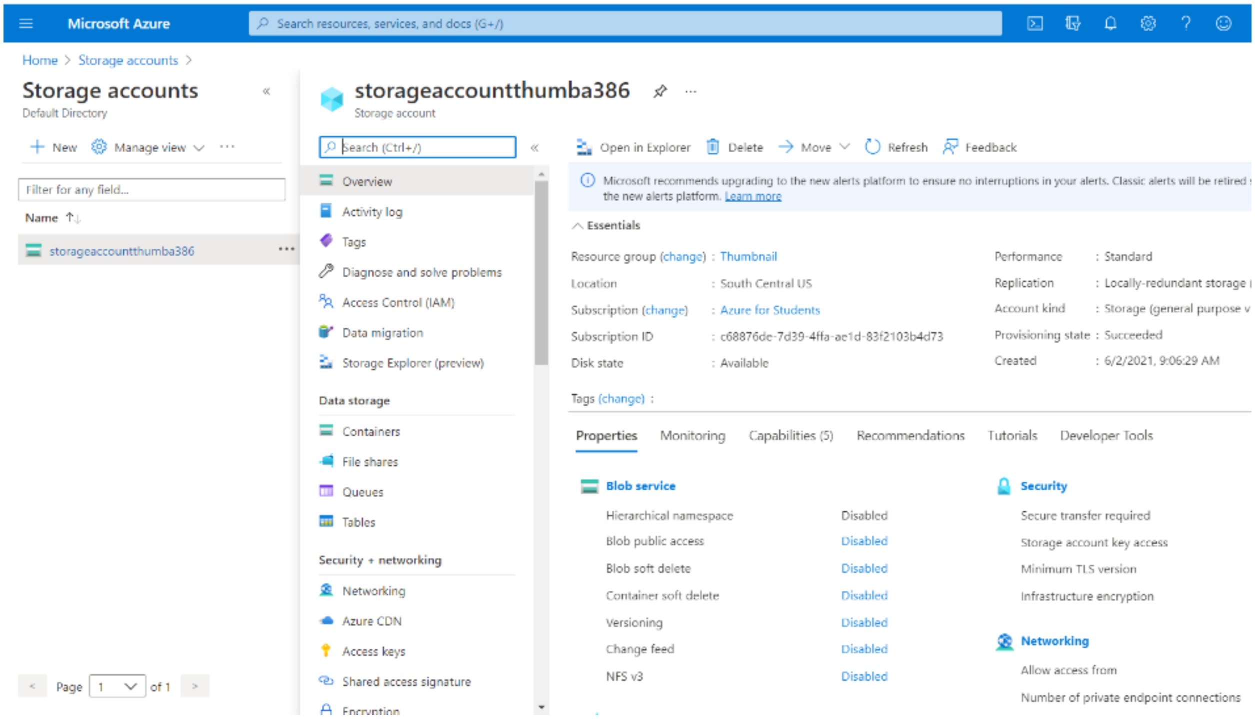


Showing 1 to 1 of 1 records.

Name ↑↓	Type ↑↓	Kind ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
storageaccountthumba386	Storage account	Storage	Thumbnail	South Central US	Azure for Students

Step-4:

You can see your storage account after creating account.



Microsoft recommends upgrading to the new alerts platform to ensure no interruptions in your alerts. Classic alerts will be retired the new alerts platform. Learn more

Essentials

Resource group (change) : Thumbnail
Location : South Central US
Subscription (change) : Azure for Students
Subscription ID : c68876de-7d39-4ffa-ae1d-83f2103b4d73
Disk state : Available

Performance : Standard
Replication : Locally-redundant storage
Account kind : Storage (general purpose v
Provisioning state : Succeeded
Created : 6/2/2021, 9:06:29 AM

Tags (change) :

Blob service

Hierarchical namespace	Disabled
Blob public access	Disabled
Blob soft delete	Disabled
Container soft delete	Disabled
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled

Security

Secure transfer required
Storage account key access
Minimum TLS version
Infrastructure encryption

Networking

Allow access from
Number of private endpoint connections

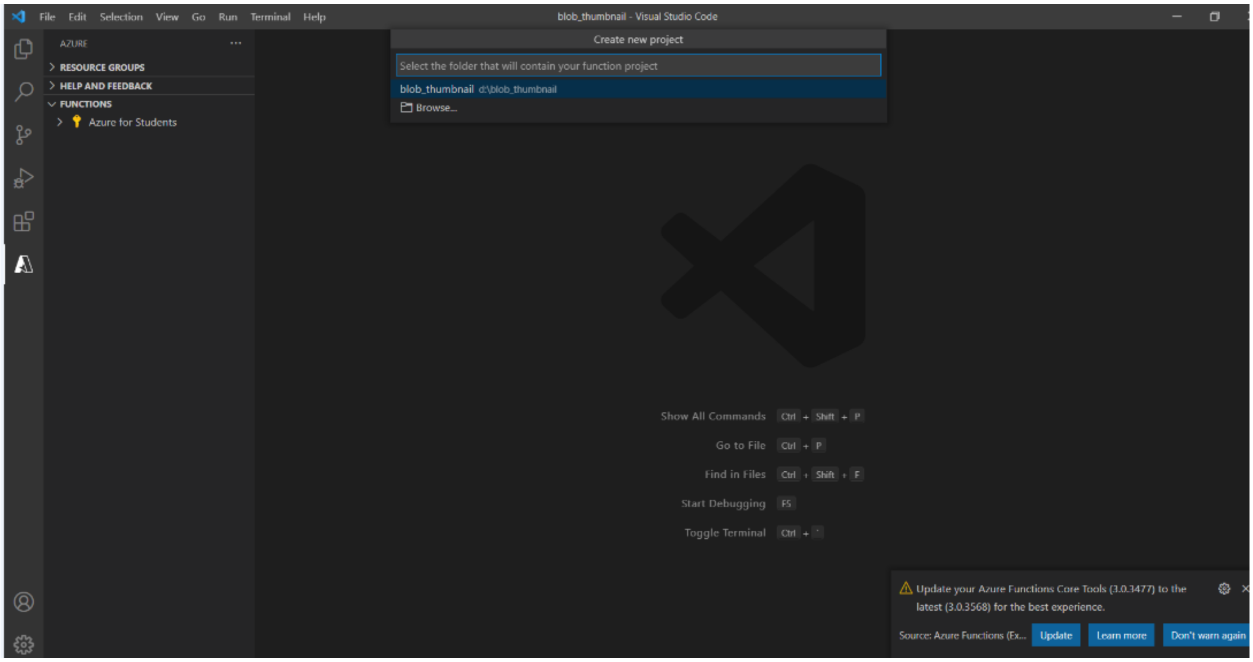
Step-5:

Now working with VS studio.

Install the extensions **Azure Account** for logging into your azure account via VS code.

Install the extension **Azure Functions** for creating basic template of azure function and running it via VS Code.

Create a new project and give its name.



blob_thumbnail - Visual Studio Code

Create new project

Select the folder that will contain your function project

blob_thumbnail d:\blob_thumbnail

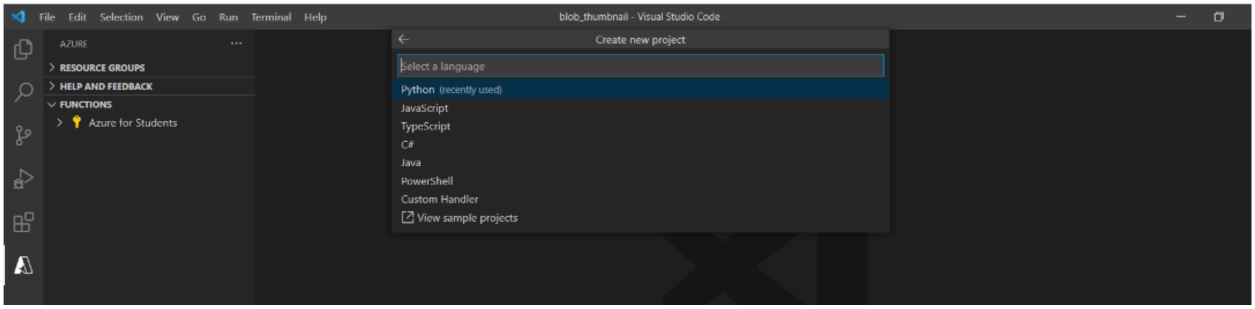
Browse...

Update your Azure Functions Core Tools (3.0.3477) to the latest (3.0.3568) for the best experience.

Source: Azure Functions (Ex...

Step-6:

Now Select select preferred language(python for this project).



blob_thumbnail - Visual Studio Code

Create new project

Select a language

Python (recently used)

JavaScript

TypeScript

C#

Java

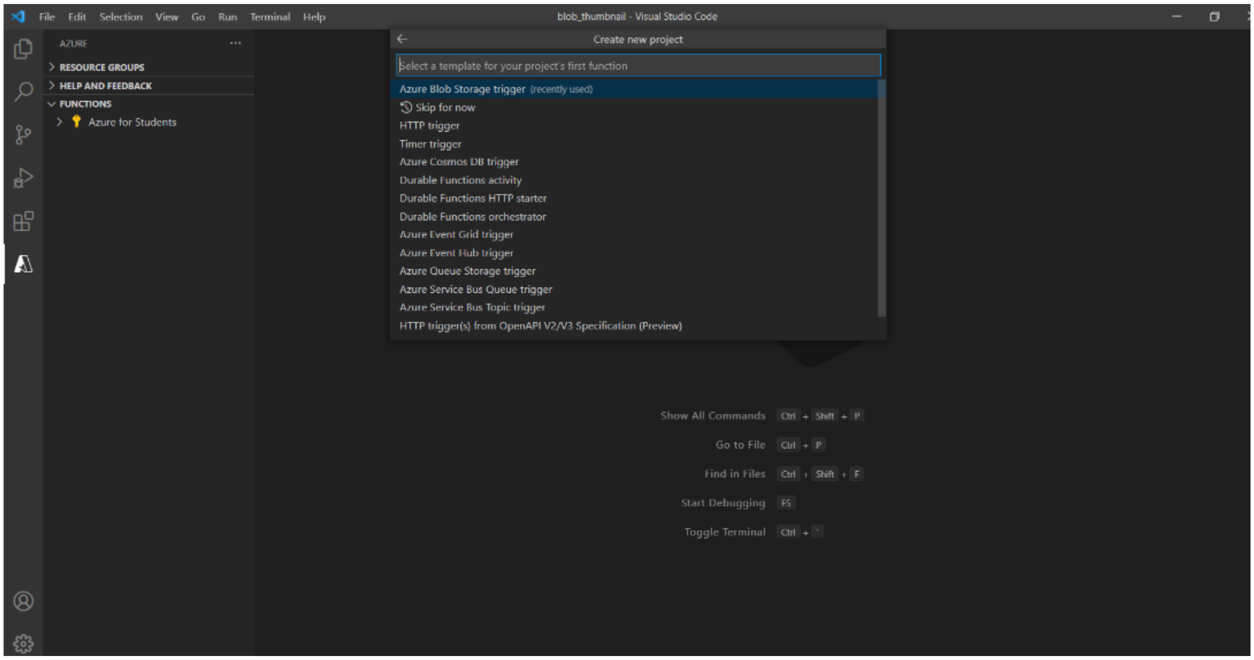
PowerShell

Custom Handler

View sample projects

Step-7:

Select Template for the project function, select **Blob storage Trigger** for our usecase.


File Edit Selection View Go Run Terminal Help
blob_thumbnail - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Azure for Students
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `

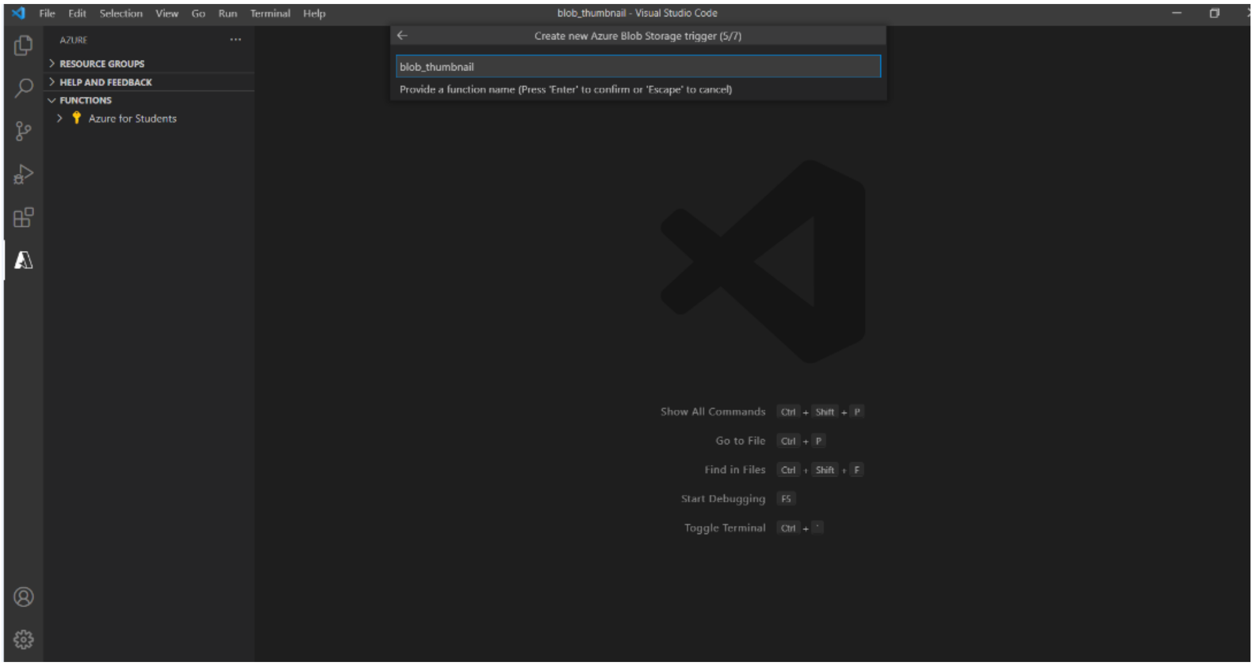
Create new project

Select a template for your project's first function

Azure Blob Storage trigger (recently used)
Skip for now
HTTP trigger
Timer trigger
Azure Cosmos DB trigger
Durable Functions activity
Durable Functions HTTP starter
Durable Functions orchestrator
Azure Event Grid trigger
Azure Event Hub trigger
Azure Queue Storage trigger
Azure Service Bus Queue trigger
Azure Service Bus Topic trigger
HTTP trigger(s) from OpenAPI V2/V3 Specification (Preview)

Step-8:

Provide the name of the function.


File Edit Selection View Go Run Terminal Help
blob_thumbnail - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Azure for Students
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `

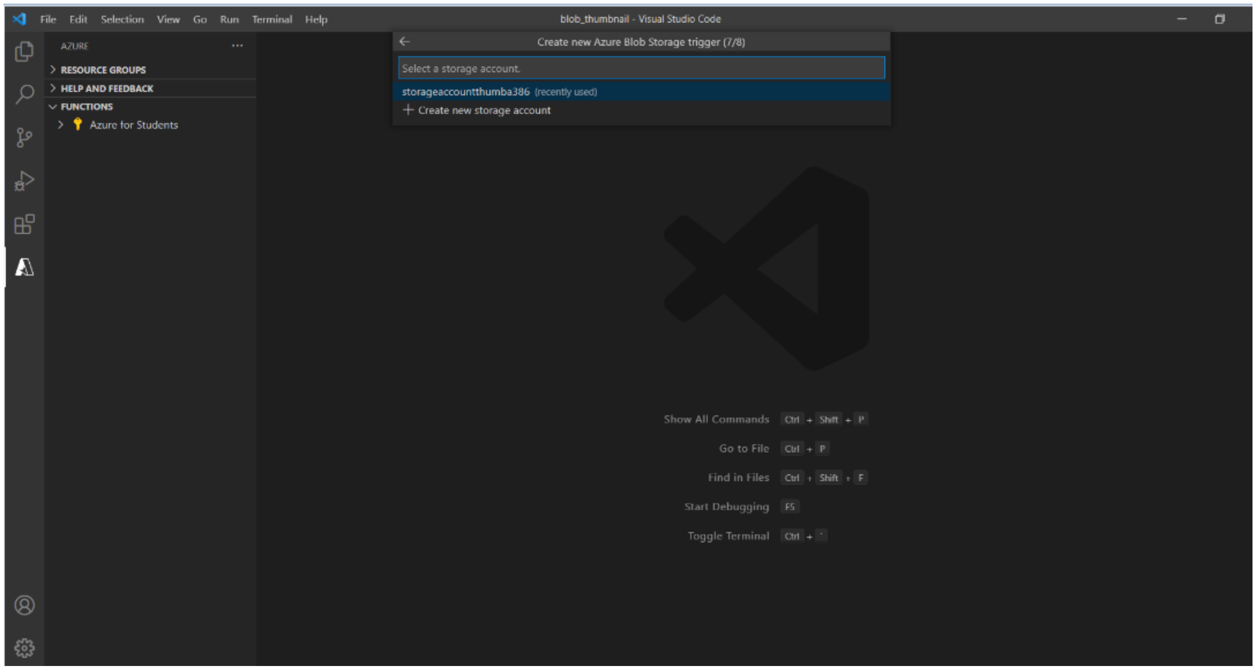
Create new Azure Blob Storage trigger (5/7)

blob_thumbnail

Provide a function name (Press 'Enter' to confirm or 'Escape' to cancel)

Step-9:

Select the storage account that we have already created. We can also create a new storage account if we do not already have one.


File Edit Selection View Go Run Terminal Help
blob_thumbnail - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Azure for Students
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + `

Create new Azure Blob Storage trigger (7/8)

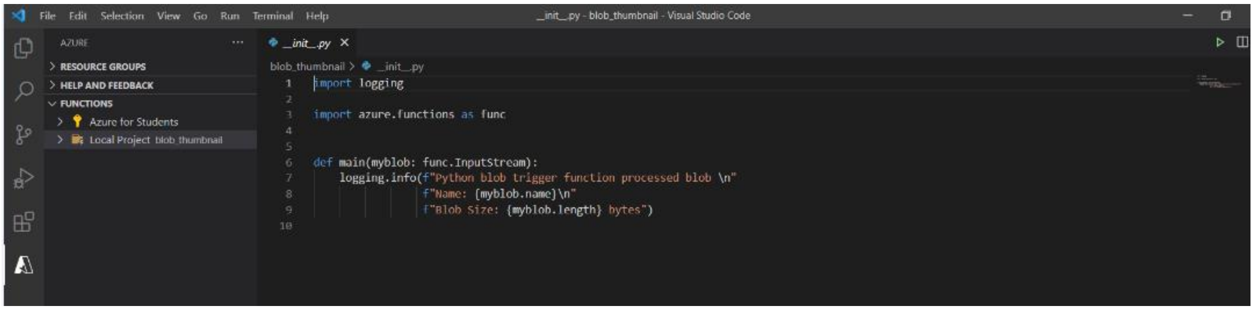
Select a storage account.

storageaccountthumba386 (recently used)

+ Create new storage account

Step-10:

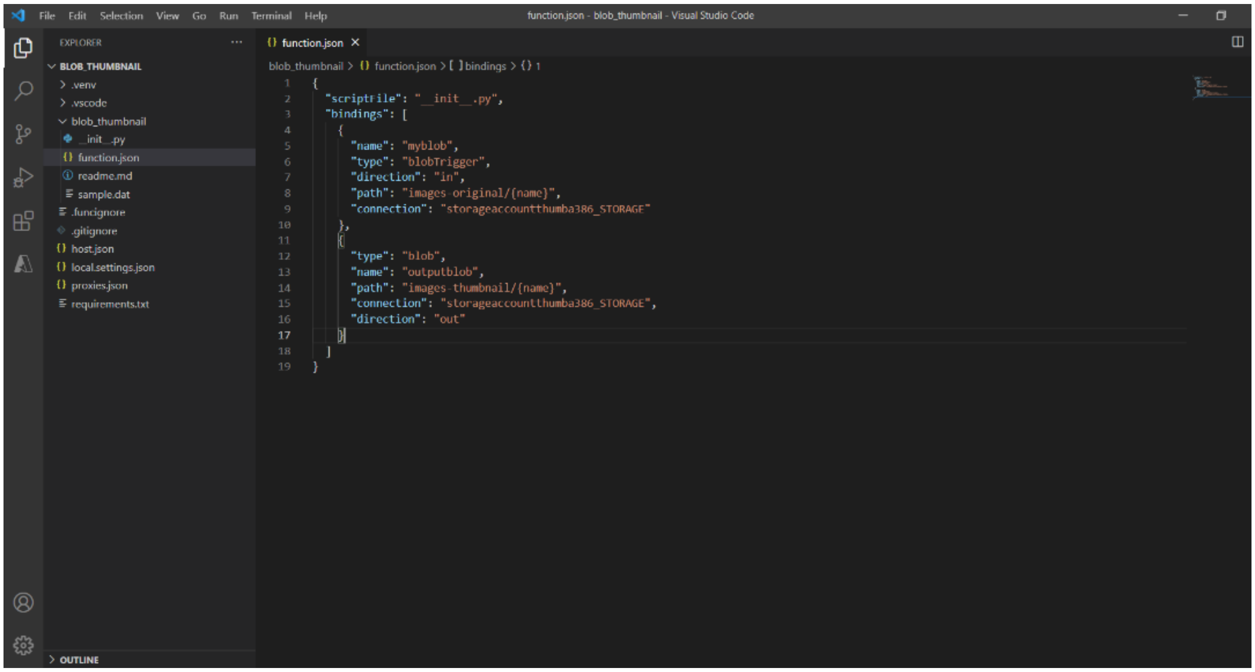
Now in __init__.py file import the azure.function. Have create 2 storages temp1 and temp2.


File Edit Selection View Go Run Terminal Help
__init__.py - blob_thumbnail - Visual Studio Code
AZURE
RESOURCE GROUPS
HELP AND FEEDBACK
FUNCTIONS
Azure for Students
Local Project blob_thumbnail
__init__.py
blob_thumbnail > __init__.py

```python
import logging

import azure.functions as func


def main(myblob: func.InputStream):
    logging.info(f"Python blob trigger function processed blob \n"
                 f"Name: {myblob.name}\n"
                 f"Blob Size: {myblob.length} bytes")

```

Step-11:

Now set the properties for input and output in the function.json file.

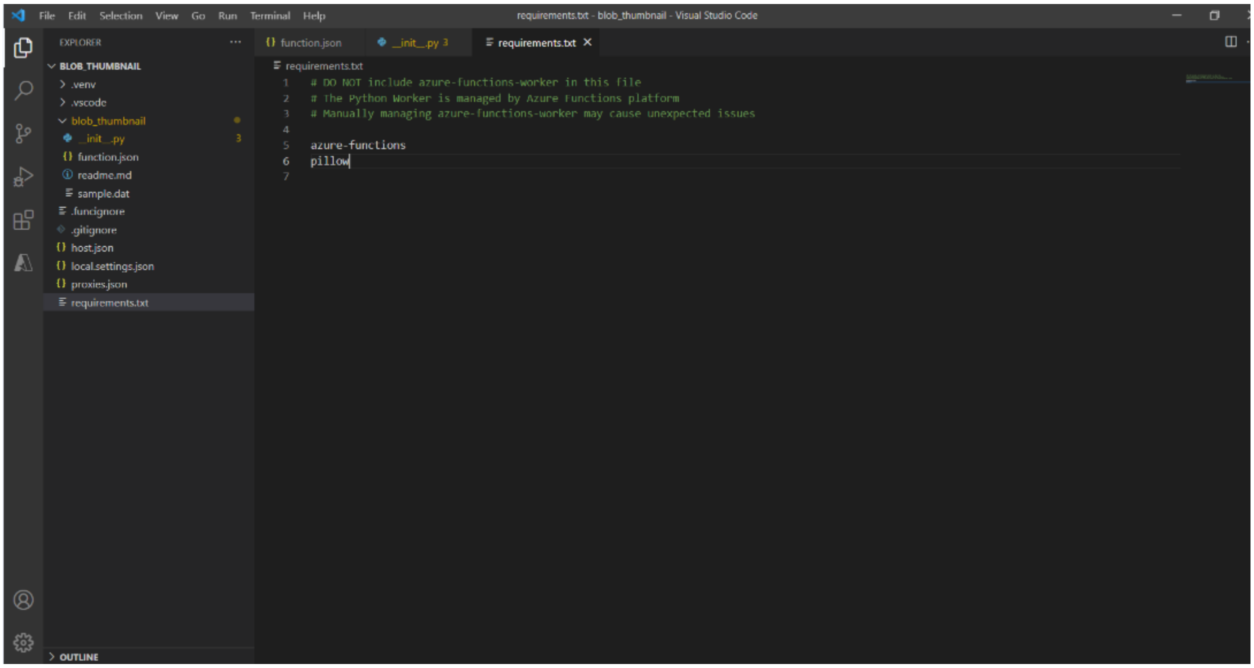


```json
{
  "scriptFile": "__init__.py",
  "bindings": [
    {
      "name": "myblob",
      "type": "blobTrigger",
      "direction": "in",
      "path": "images-original/{name}",
      "connection": "storageaccountthumba386_STORAGE"
    },
    {
      "type": "blob",
      "name": "outputblob",
      "path": "images-thumbnail/{name}",
      "connection": "storageaccountthumba386_STORAGE",
      "direction": "out"
    }
  ]
}
```

Step-12:

Add the required pip package names in Requirement.txt file.

Install them using pip install -r Requirement.txt.



```
# 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
pillow
```

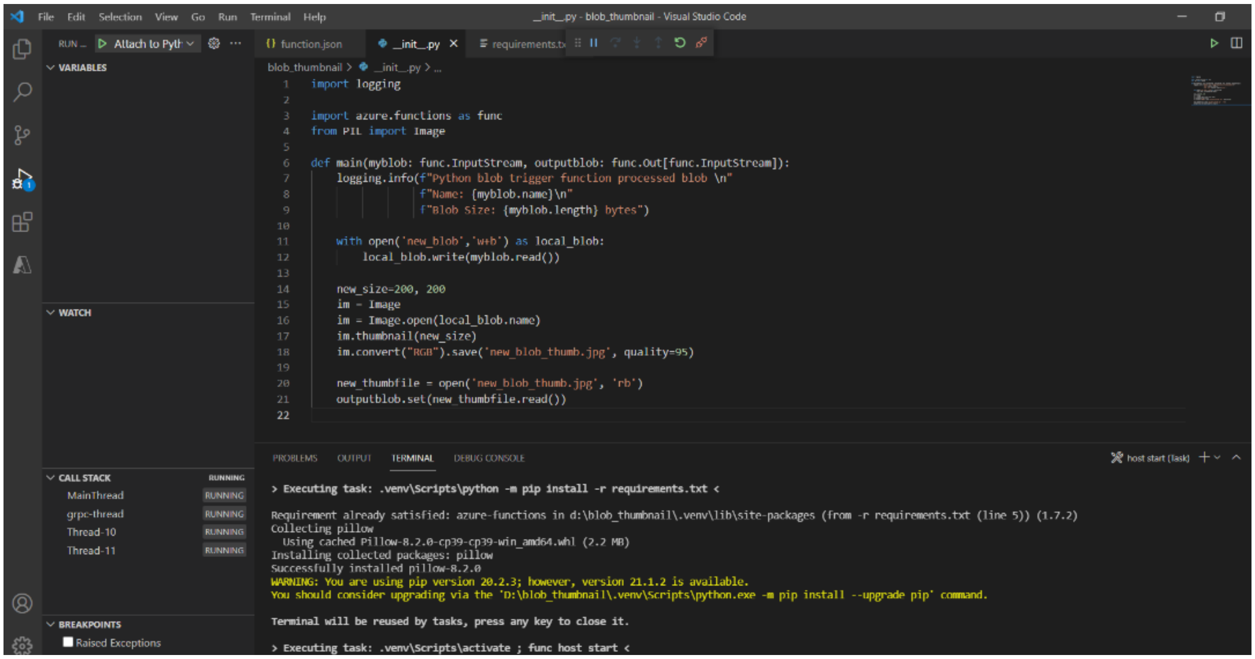
Step-13:

Add the following code to __init__.py file.

This function will be triggered whenever a file will be uploaded to temp1 blob container in our azure blob containers.

This function creates a thumbnail from an image and automatically saves it to temp2 container in our azure blob containers.

We can do any other image processing as per our buisness requirement inside the main() function.



```python
import logging

import azure.functions as func
from PIL import Image

def main(myblob: func.InputStream, outputblob: func.Out[func.InputStream]):
    logging.info(f"Python blob trigger function processed blob \n"
                 f"Name: {myblob.name}\n"
                 f"Blob Size: {myblob.length} bytes")

    with open('new_blob','w+b') as local_blob:
        local_blob.write(myblob.read())

    new_size=200, 200
    im = Image
    im = Image.open(local_blob.name)
    im.thumbnail(new_size)
    im.convert("RGB").save('new_blob_thumb.jpg', quality=95)

    new_thumbfile = open('new_blob_thumb.jpg', 'rb')
    outputblob.set(new_thumbfile.read())
```

```
> Executing task: .venv\Scripts\python -m pip install -r requirements.txt <

Requirement already satisfied: azure-functions in d:\blob_thumbnail\.venv\lib\site-packages (from -r requirements.txt (line 5)) (1.7.2)
Collecting pillow
  Using cached Pillow-8.2.0-cp39-cp39-win_amd64.whl (2.2 MB)
Installing collected packages: pillow
Successfully installed pillow-8.2.0
WARNING: You are using pip version 20.2.3; however, version 21.1.2 is available.
You should consider upgrading via the 'D:\blob_thumbnail\.venv\Scripts\python.exe -m pip install --upgrade pip' command.

Terminal will be reused by tasks, press any key to close it.

> Executing task: .venv\Scripts\activate ; func host start <
```

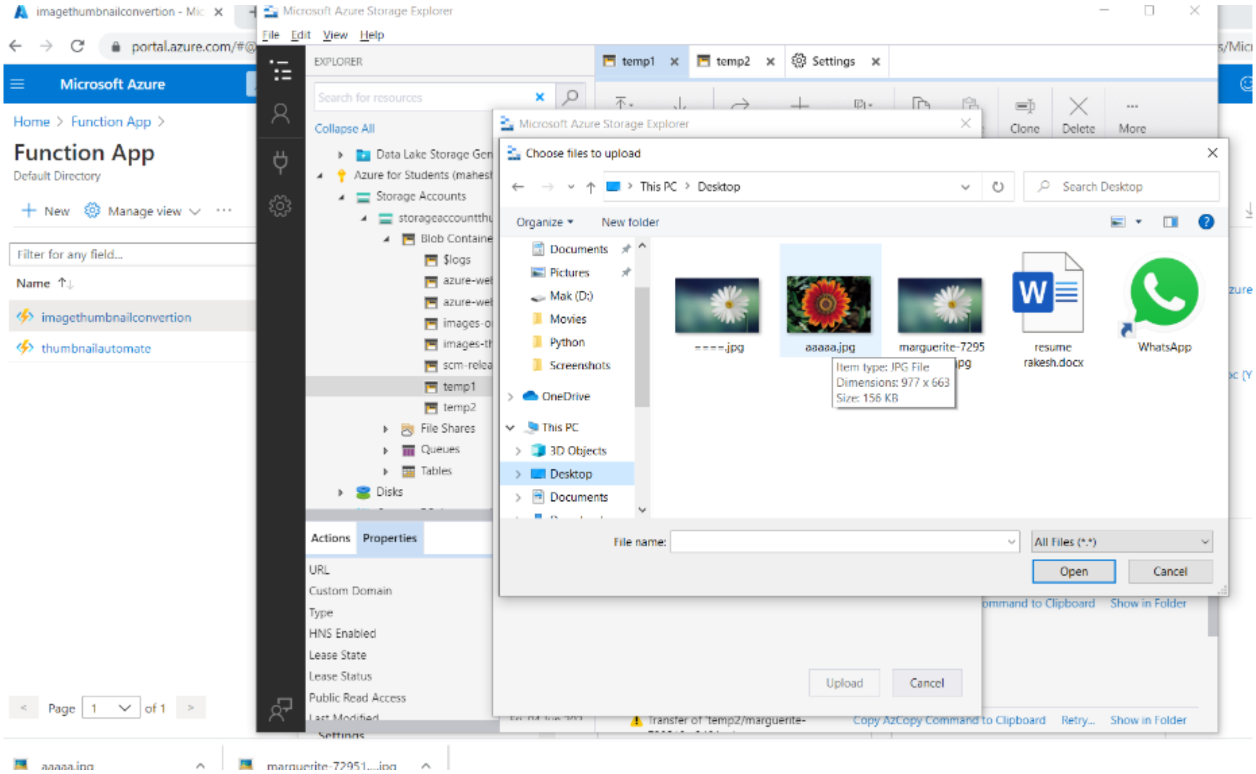
Step 14:

Press **F5** or run the command **func host start** in terminal to run our program.

Step-15:

Let’s test out program.

In Azure portal Upload an image in temp1 storage container.



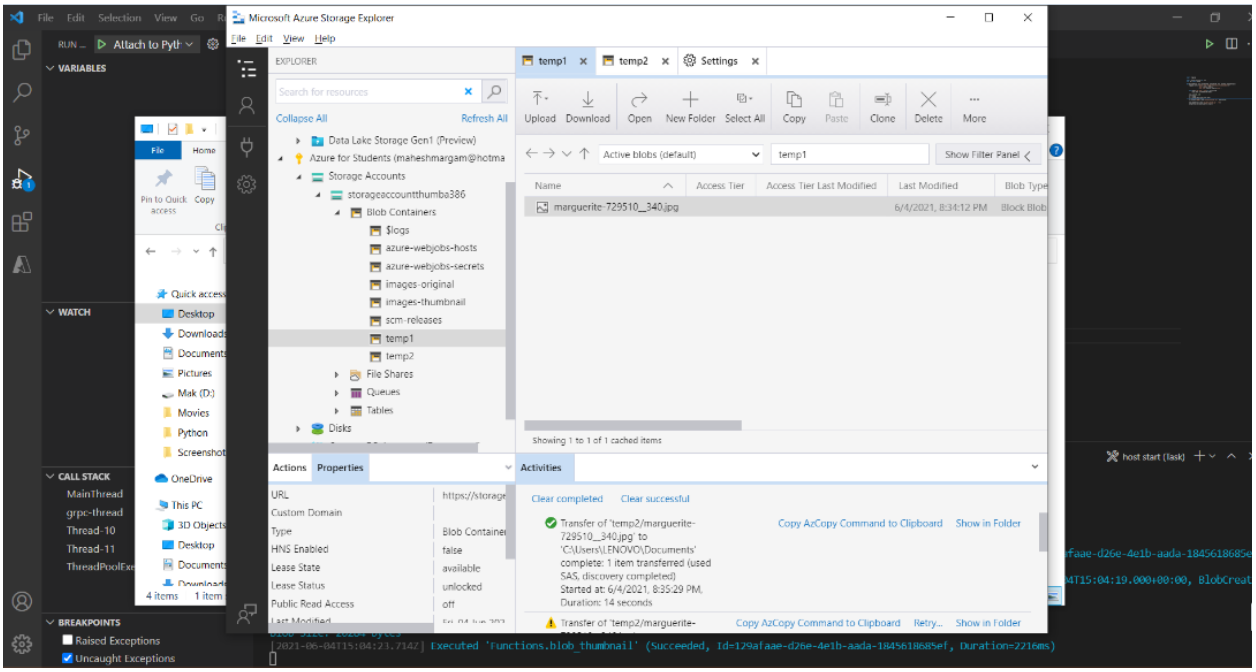
Choose files to upload

This PC > Desktop

----.jpg
aaaaa.jpg
marguerite-7295...jpg
resume rakesh.docx
WhatsApp

Item type: JPG File
Dimensions: 977 x 663
Size: 156 KB

File name:
All Files (*.*)



marguerite-729510__340.jpg 6/4/2021, 8:34:12 PM Block Blob

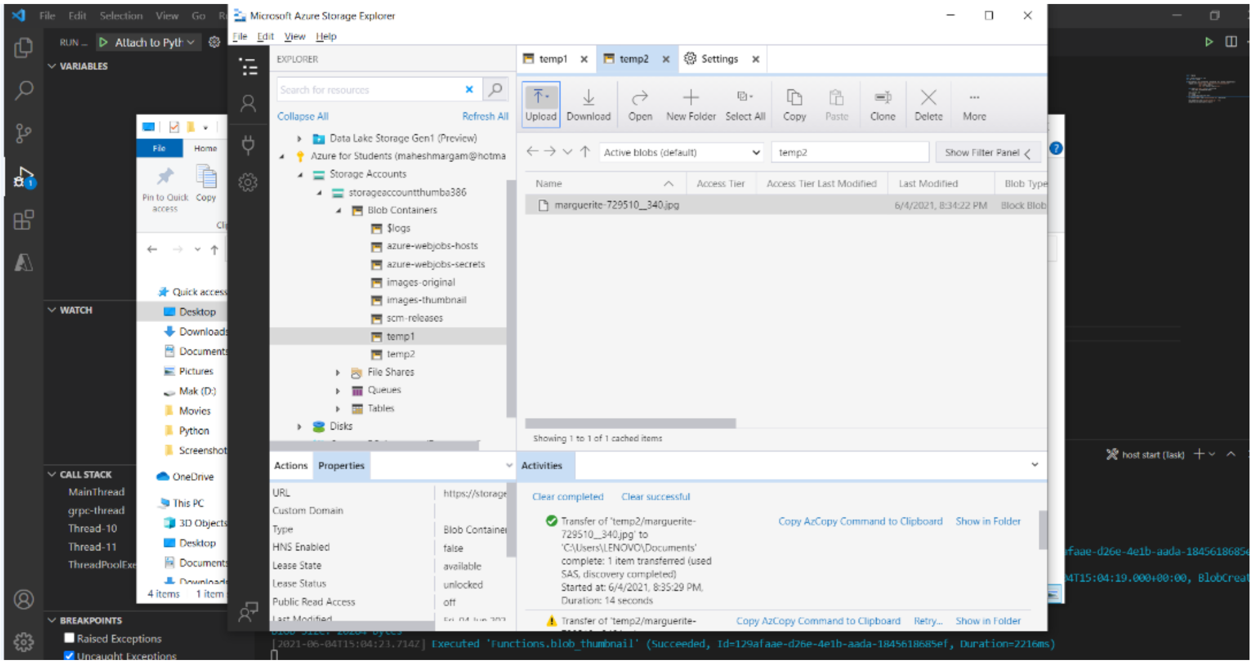
Showing 1 to 1 of 1 cached items

Transfer of 'temp2/marguerite-729510__340.jpg' to 'C:\Users\LENOVO\Documents' complete: 1 item transferred (used SAS, discovery completed)
Started at: 6/4/2021, 8:35:29 PM,
Duration: 14 seconds

```
[2021-06-04T15:04:23.714Z] Executed 'Functions.blob_thumbnail' (Succeeded, Id=129afaae-d26e-4e1b-aada-1845618685ef, Duration=2216ms)
```

Step-16:

Now you can see this trigger activated and temp2 storage also contain that image from temp1 storage.



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type
marguerite-729510__340.jpg			6/4/2021, 8:34:22 PM	Block Blob

Showing 1 to 1 of 1 cached items

URL	https://storage
Custom Domain	
Type	Blob Container
HNS Enabled	false
Lease State	available
Lease Status	unlocked
Public Read Access	off

Transfer of 'temp2/marguerite-729510__340.jpg' to 'C:\Users\LENOVO\Documents' complete: 1 item transferred (used SAS, discovery completed) Started at: 6/4/2021, 8:35:29 PM, Duration: 14 seconds

Transfer of 'temp2/marguerite-

```
[2021-06-04T15:04:23.714Z] Executed 'Functions.blob_thumbnail' (Succeeded, Id=129afaae-d26e-4e1b-aada-1845618685ef, Duration=2216ms)
```

After this we can see in Microsoft Azure Storage Explorer that new file will be uploaded in temp2 container which in the processed image via main() function.