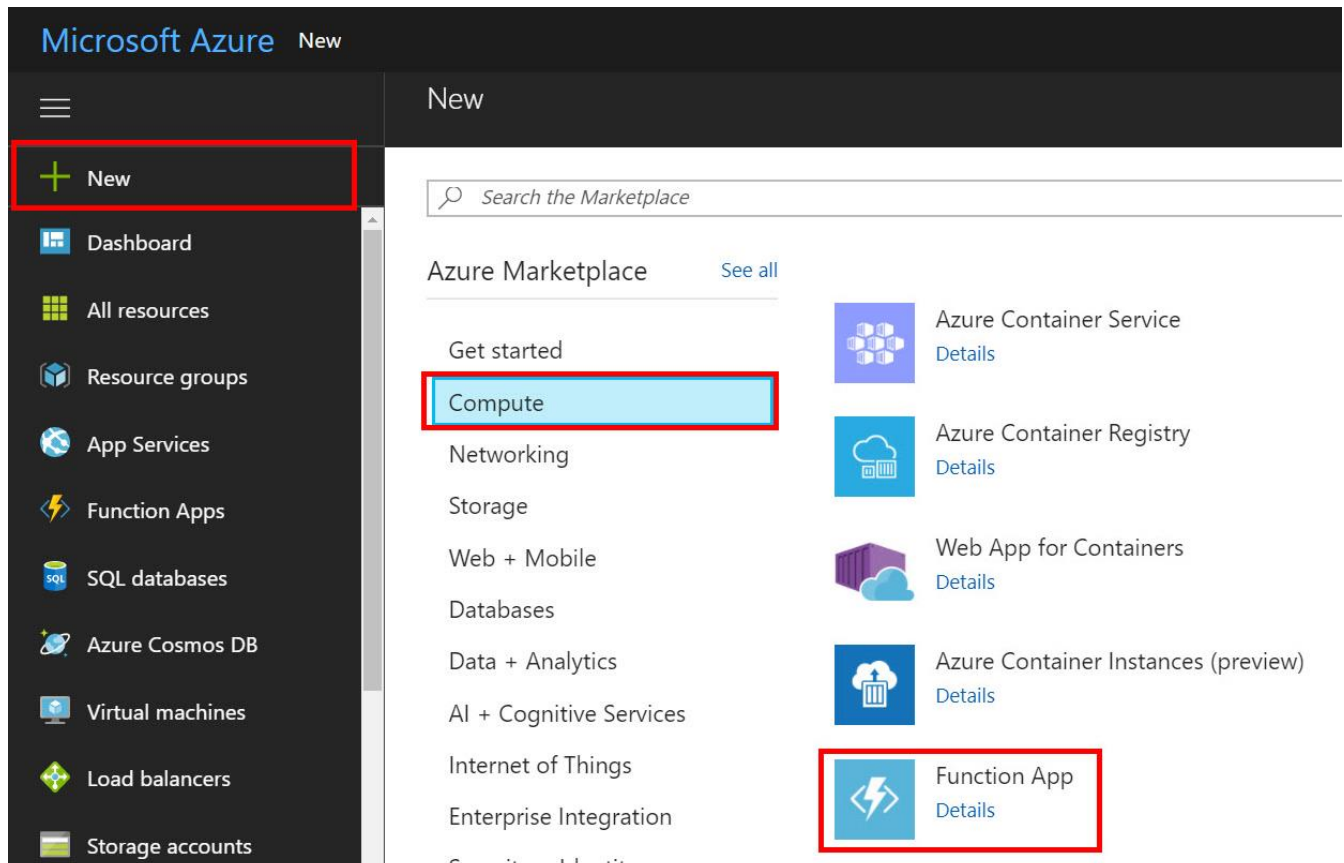


Azure Function App – Blob Trigger

Step 1: Open Microsoft Azure Portal & Login with Azure subscription credentials

<https://portal.azure.com>

Step 2: Click on +New -> Compute -> Function App



Step 3: Enter Function App Details

Function App

Create

* App name

azurefunappdemo

.azurewebsites.net

* Subscription

MSDN Platforms

* Resource Group ⓘ

☒ Create new

☐ Use existing

FUNAPPRG

* Hosting Plan ⓘ

Consumption Plan

* Location

Southeast Asia

* Storage ⓘ

☒ Create New

☐ Select Existing

azurefunappdemoa639

Application Insights ⓘ

On

Off

☒ Pin to dashboard

Create

Automation options

App Name: azurefunappdemo

Subscription: Choose any of the Subscription

Resource Group: Create New resource.

Ex. FUNAPPRG

Hosting Plan: Consumption Plan

Location: Choose nearest location

Storage: Create New Storage Account

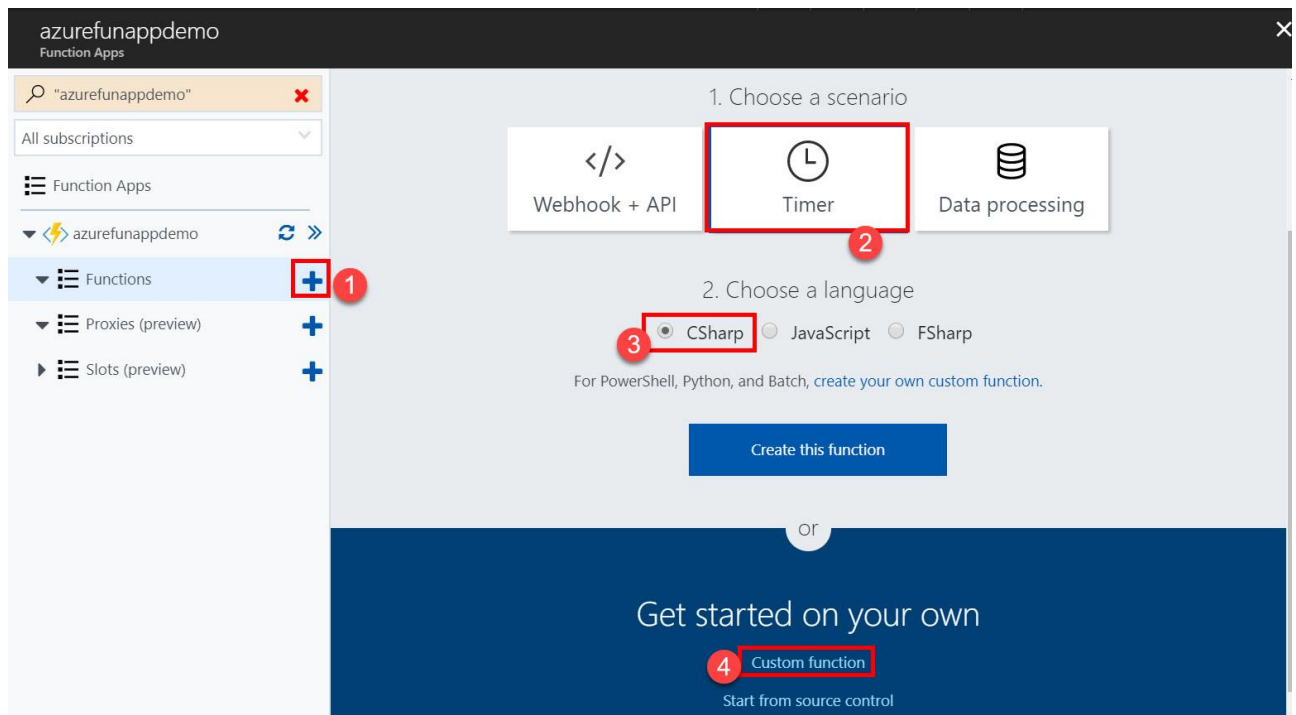
Step 4: Wait for couple of seconds to create function app

Click on **Add new function** button

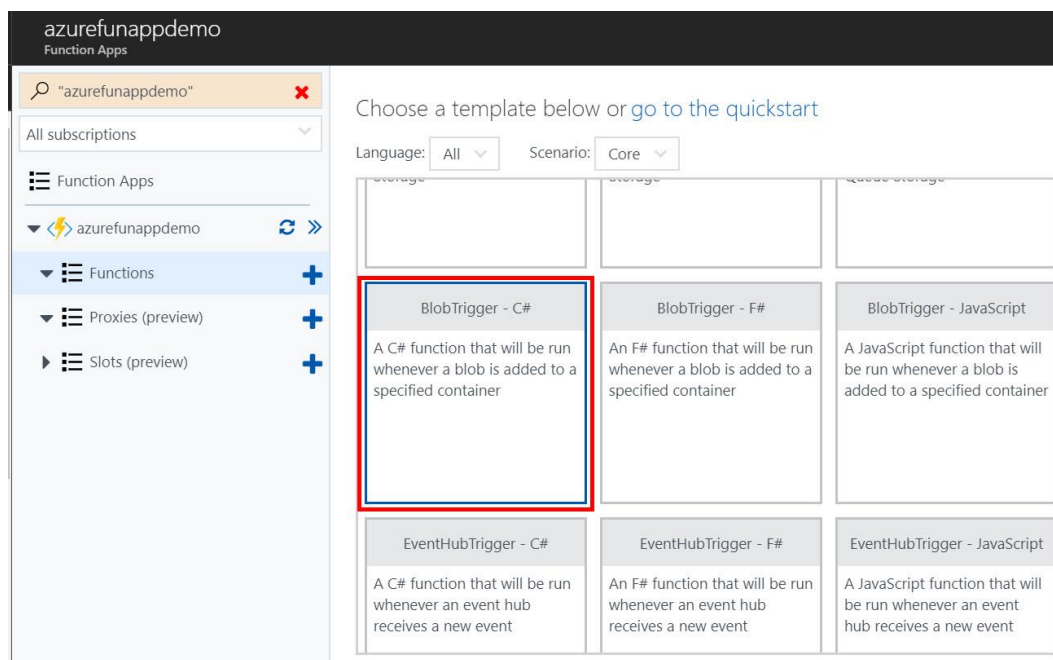
Choose a Scenario: Timer

Choose a language: CSharp

Click on **Custom function** button



Step 5: Select **BlobTrigger - C#**



Scroll down & edit path with **mycontainer**

Path: **mycontainer/{name}**

azurefunappdemo
Function Apps

Search "azurefunappdemo" ✕

All subscriptions ▾

Function Apps

▼ ⚡ azurefunappdemo ↻ »

▼ ☰ Functions +

▼ ☰ Proxies (preview) +

▶ ☰ Slots (preview) +

C#	F#	JavaScript
A C# function that will be run whenever a message is added to a specified Service Bus queue	An F# function that will be run whenever a message is added to a specified Service Bus queue	A JavaScript function that will be run whenever a message is added to a specified Service Bus queue

Name your function

BlobTriggerCSharp1

Azure Blob Storage trigger

Path ⓘ

Storage account connection ⓘ show value new

Create

Click on **new** option of Storage account

azurefunappdemo
Function Apps

Search "azurefunappdemo" ✕

All subscriptions ▾

Function Apps

▼ ⚡ azurefunappdemo ↻ »

▼ ☰ Functions +

▼ ☰ Proxies (preview) +

▶ ☰ Slots (preview) +

C#	F#	JavaScript
A C# function that will be run whenever a message is added to a specified Service Bus queue	An F# function that will be run whenever a message is added to a specified Service Bus queue	A JavaScript function that will be run whenever a message is added to a specified Service Bus queue

Name your function

BlobTriggerCSharp1

Azure Blob Storage trigger

Path ⓘ

Storage account connection ⓘ show value **new**

Create

Choose **Azure Storage Account** from the list

Storage Account

Don't see a storage account? Azure Functions requires a general-purpose storage account that supports Tables, Queues, Files and Blobs.

Create New

azurefunappdemo0e
South Central US

Storage account connection ⓘ show value
AzureWebJobsDashboard ▼

After selecting all option click on **Create** button

azurefunappdemo
Function Apps

"azurefunappdemo" ✕

All subscriptions ▼

Function Apps

▼ ⚡ azurefunappdemo ↻ »

▼ Functions +

▼ Proxies (preview) +

▶ Slots (preview) +

ServiceBusQueueTrigger - C#
A C# function that will be run whenever a message is added to a specified Service Bus queue

ServiceBusQueueTrigger - F#
An F# function that will be run whenever a message is added to a specified Service Bus queue

ServiceBusQueueTrigger - JavaScript
A JavaScript function that will be run whenever a message is added to a specified Service Bus queue

Name your function

BlobTriggerCSharp1

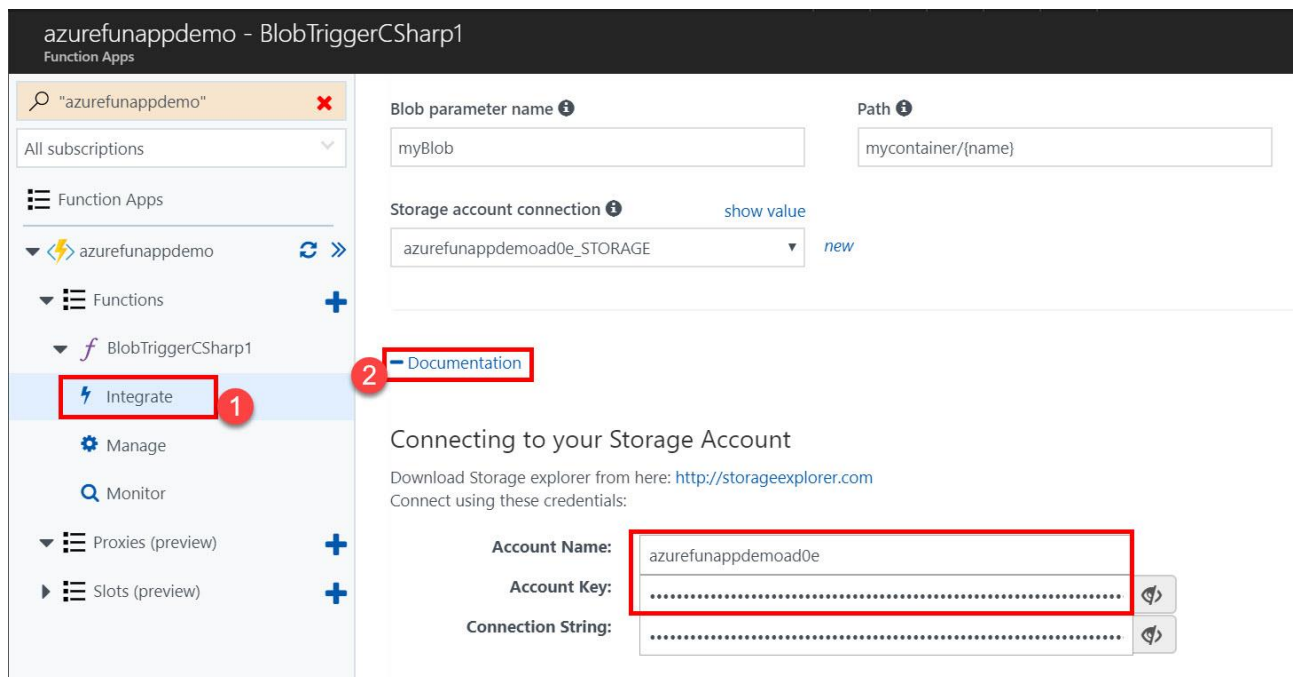
Azure Blob Storage trigger

Path ⓘ mycontainer/{name}

Storage account connection ⓘ show value
azurefunappdemo0e_STORAGE ▼ new

Create

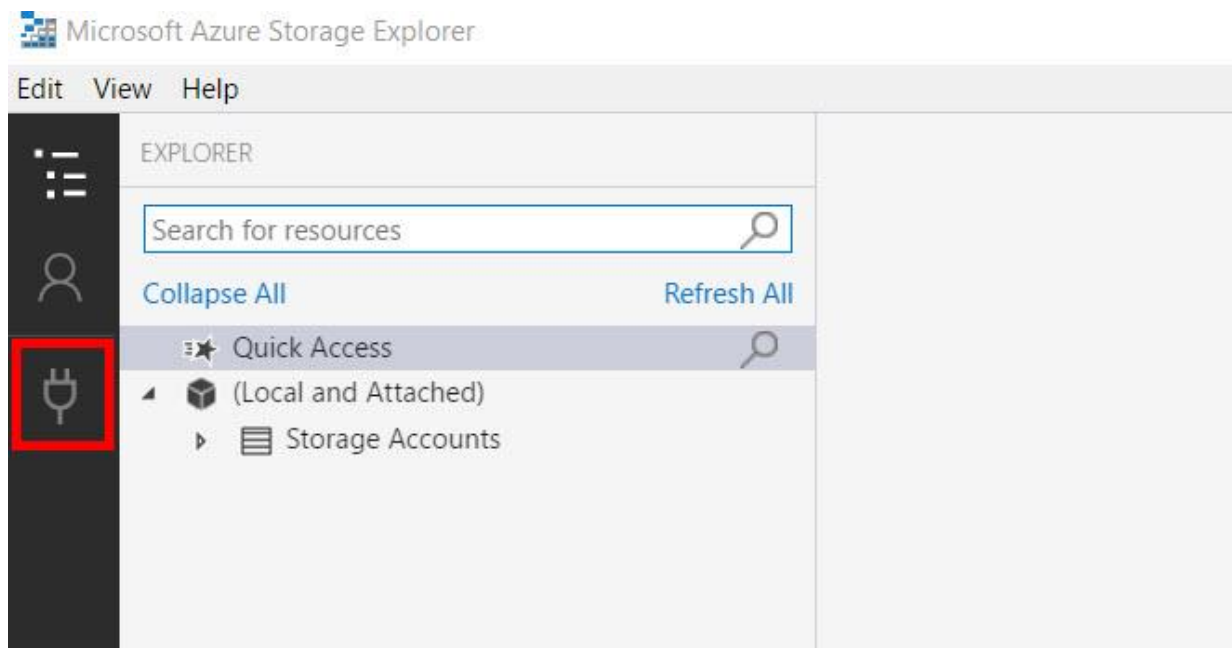
Step 6: Click on **Integrate** option. Expand **Documentation** option & **Copy Account Name & Key**.



Step 7: Download **Microsoft Azure Storage Explorer** from below URL:

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

Install & Start Microsoft Azure Storage Explorer. Click on Pin icon from left side



Step 8: Choose third option: **Use a storage account name and key** & click on **Next** button

Connect to Azure Storage

How do you want to connect to your Storage Account or service?

☐ Add an Azure Account

Azure environment:

Azure

☐ Use a shared access signature (SAS) URI or connection string

☒ Use a storage account name and key

Next Connect Cancel

Step 9: Copy Account Name & Key from Azure Function App to Microsoft Azure Storage Explorer.

Attach External Storage

Enter information to connect to the Microsoft Azure storage account

Account name: azurefunappdemo0e

Account key: glAUctr06A7ONr1Wfs0HEzE7/xdABrLwnQXYpy27VVISjg84dOYcgUNKbXOEh5+mRA+7jR7WVWu2Q-274kCC

Storage endpoints domain: Microsoft Azure Default

core.windows.net

Use HTTP (Not recommended)

Online privacy statement

Back Next Connect Cancel

erCSharp1

Documentation

Connecting to your Storage Account

Download Storage explorer from here: <http://storageexplorer.com>

Connect using these credentials:

Account Name: azurefunappdemo0e

Account Key: glAUctr06A7ONr1Wfs0HEzE7/xdABrLwnQXYpy27VVISjg84dOYcgUNKbXOEh5+mRA+7jR7WVWu2Q-274kCC

Connection String:

You can now view the blobs, queues and tables associated with this storage binding.

Settings for storage blob trigger

- name** : The variable name used in function code for the blob.
- path** : A path that specifies the container to monitor, and optionally a blob name ;
- connection** : The name of an app setting that contains a storage connection string

connection empty, the trigger will work with the default storage connection string function app, which is specified by the AzureWebJobsStorage app setting.

Step 10: Click on **Connect** button.

Connection Summary

The following settings will be used to connect to your storage account:

Account name: azurefunappdemo0e

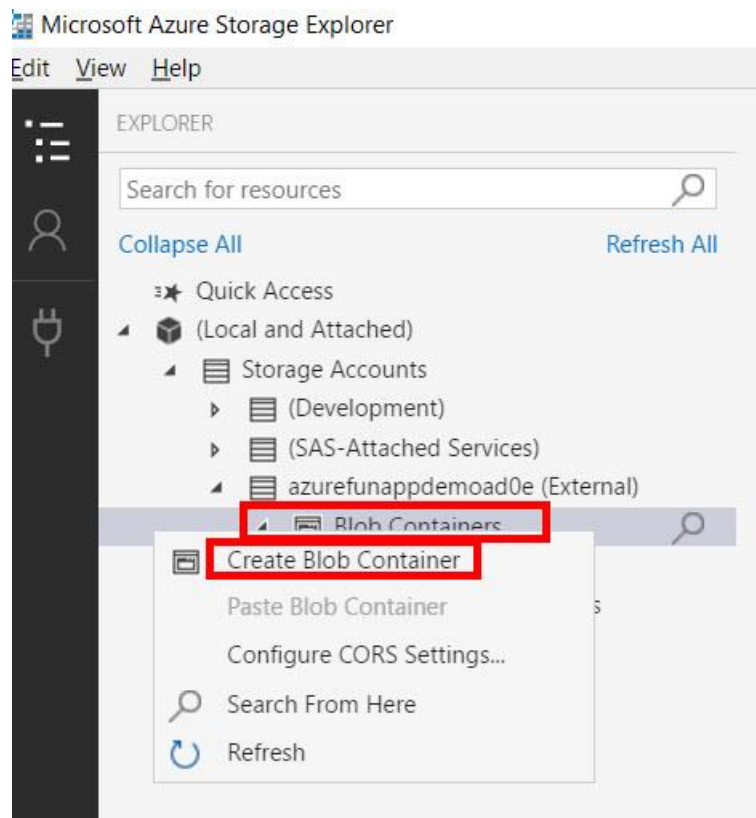
Account key: glAUctr06A7ONr1Wfs0HEzE7/xdABrLwnQXYpy27VVISjg84dOYcgUNKbXO..

Default endpoints protocol: https

Make sure you only connect to resources you trust.

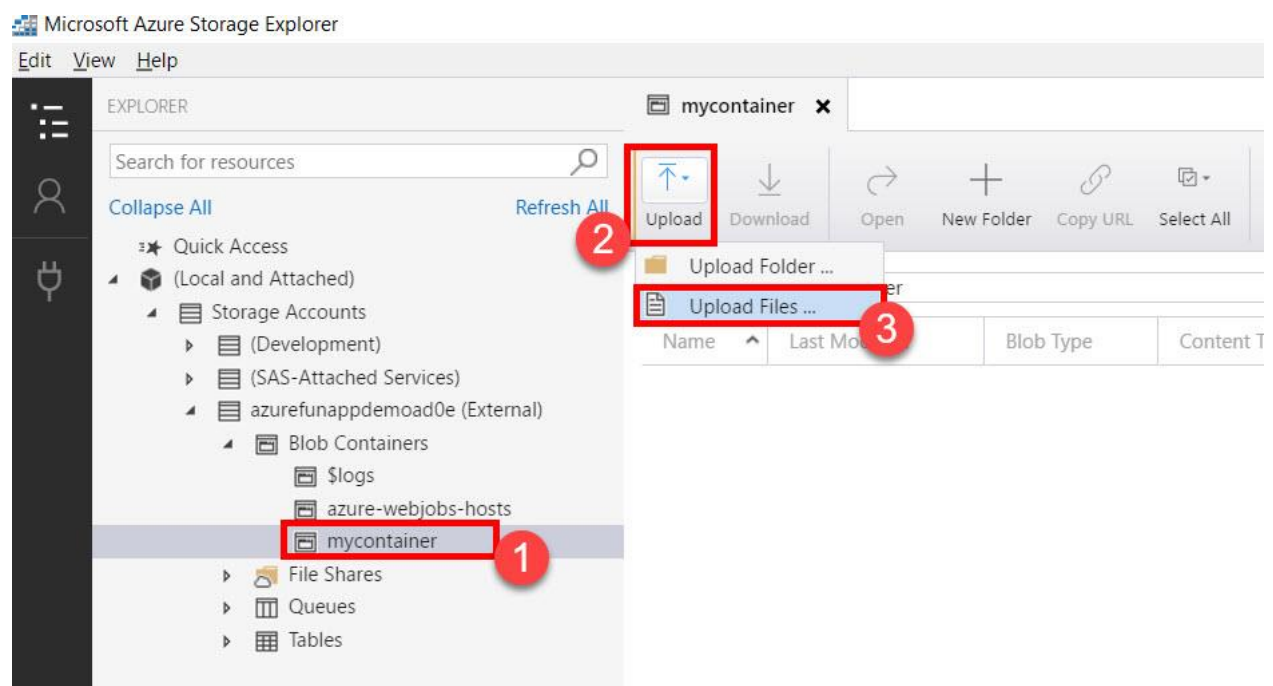
Back Next Connect Cancel

Step 11: Right click on Blob Containers & Create Blob Container.

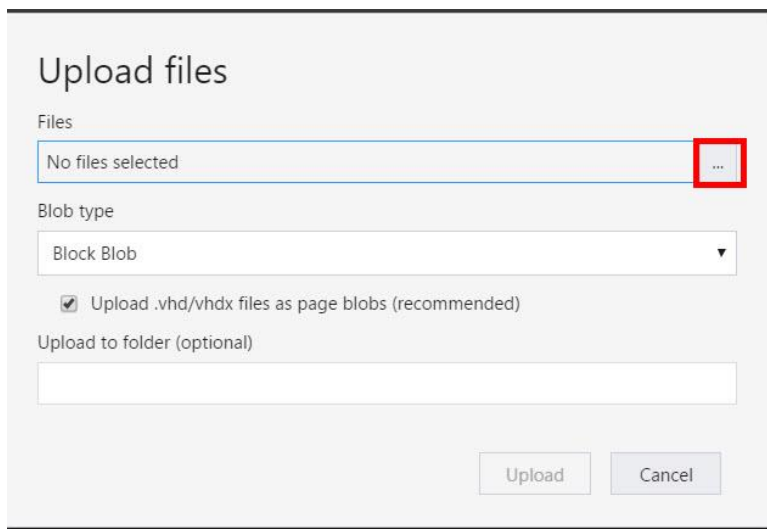


Step 12: Create new Container: mycontainer

Click on **Upload** button & select **Upload Files....**



Step 13: Upload files window will open & click on browse option



Upload files

Files

No files selected

Blob type

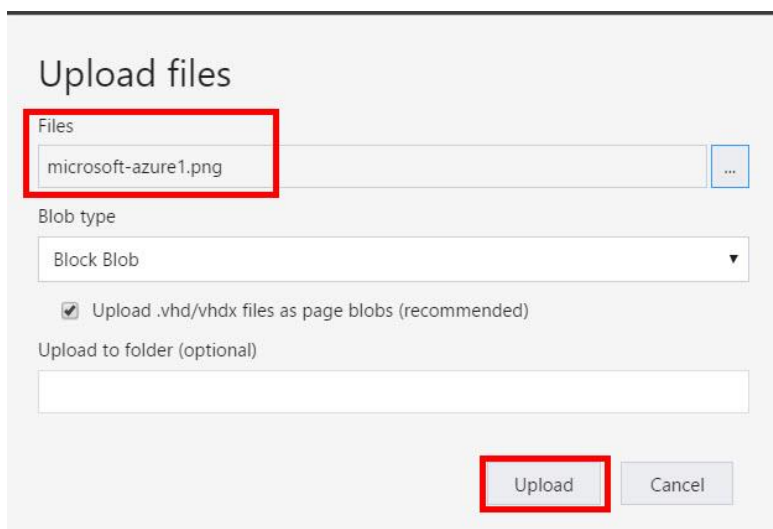
Block Blob

☒ Upload .vhd/vhdx files as page blobs (recommended)

Upload to folder (optional)

Upload Cancel

Upload Image file & click on Upload button



Upload files

Files

microsoft-azure1.png

Blob type

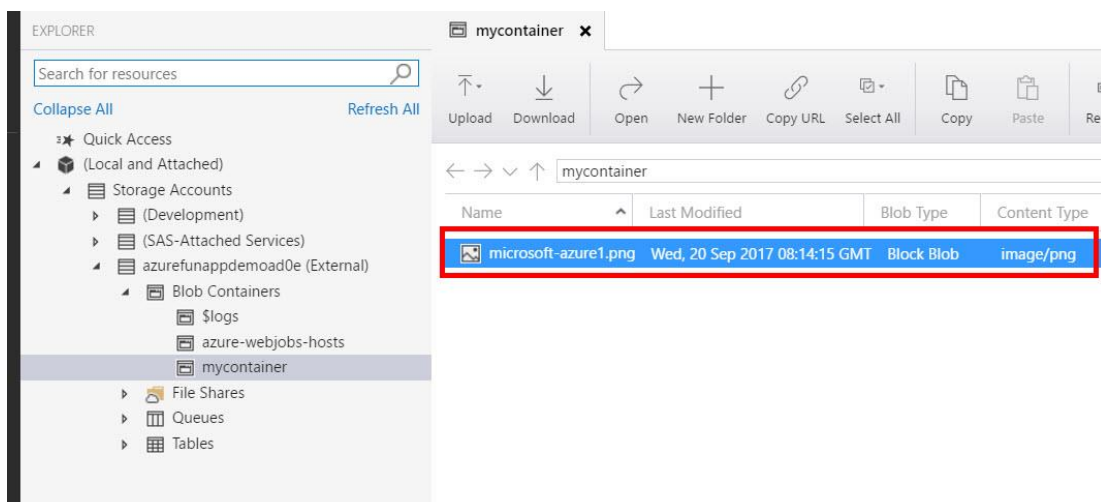
Block Blob

☒ Upload .vhd/vhdx files as page blobs (recommended)

Upload to folder (optional)

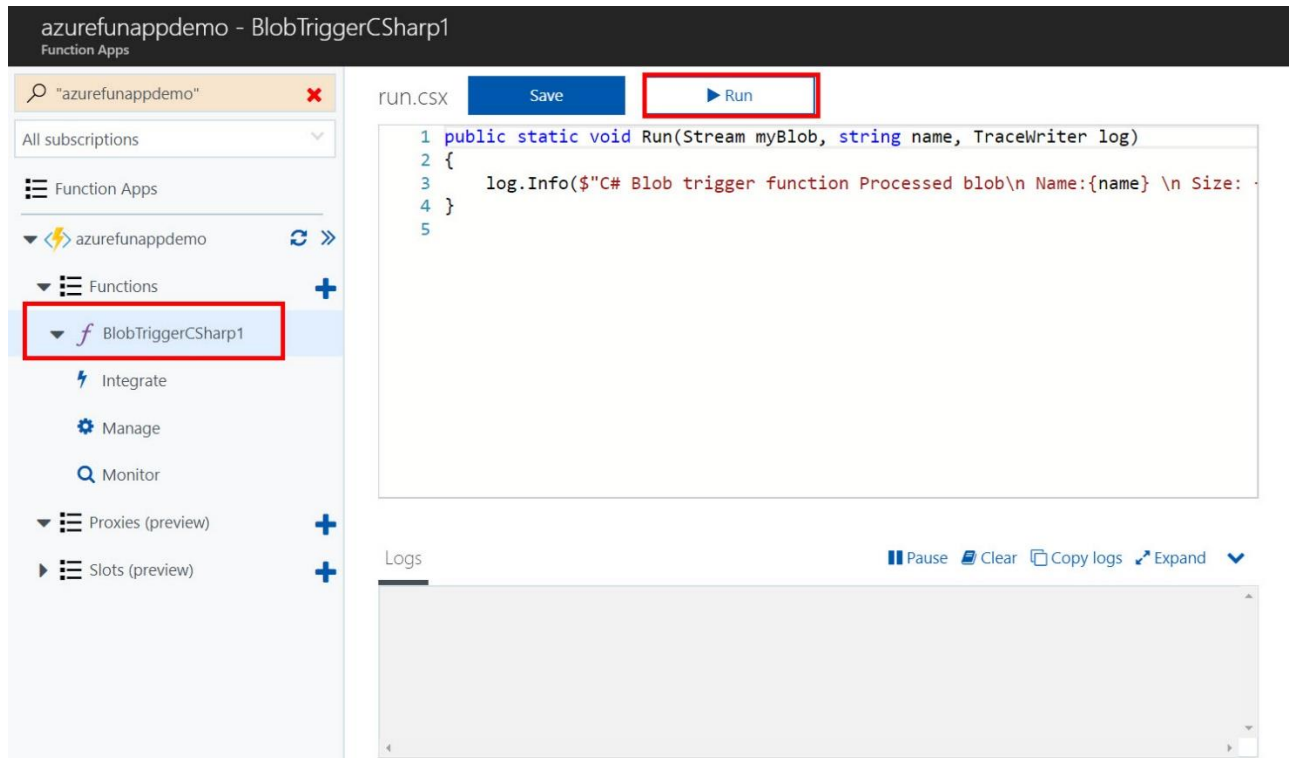
Upload Cancel

Image will successfully have uploaded on container



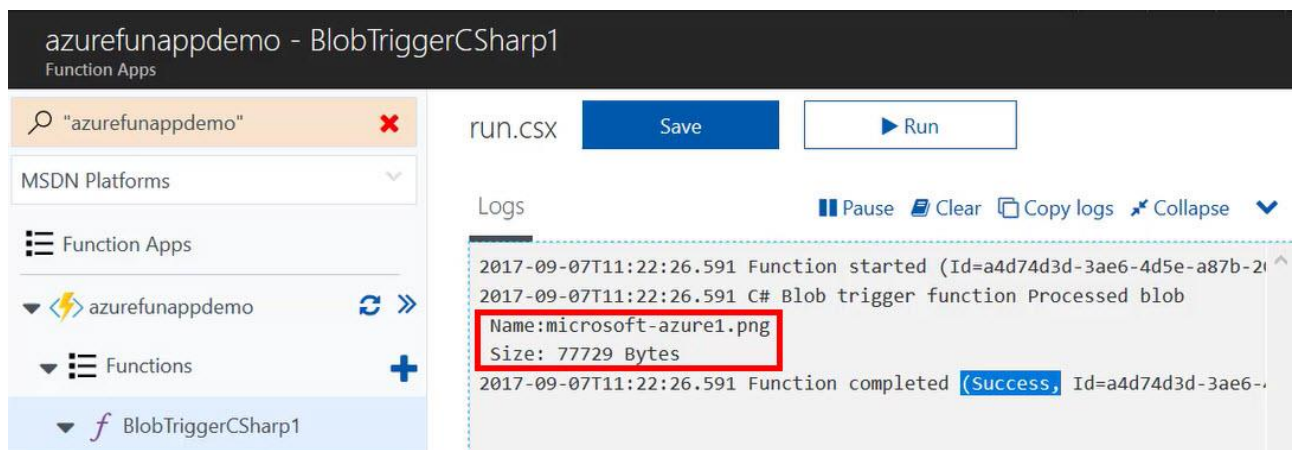
Step 14: Navigate to Azure Function App & click on function: **BlobTriggerCSharp1**

Now click on **Run** button



The screenshot shows the Azure Portal interface for the 'azurefunappdemo' Function App. In the left-hand navigation pane, the 'Functions' section is expanded, and 'BlobTriggerCSharp1' is selected. The main area displays the C# code for the function, 'run.csx', which is a simple Blob trigger function. The 'Run' button is highlighted with a red box. Below the code editor, the 'Logs' section is visible but currently empty.

In Logs, you can find all details



The screenshot shows the same Azure Portal interface, but now the 'Logs' section is expanded and displays the execution details of the 'BlobTriggerCSharp1' function. The logs show the function started at 2017-09-07T11:22:26.591, processed a blob named 'microsoft-azure1.png' with a size of 77729 Bytes, and completed successfully at the same time. The log entry for the blob name and size is highlighted with a red box.