

|  |  |
| --- | --- |
|  | Lab 7: Question from Insem Paper. |
|  |  |
|  | Akshar Panchani ID- 202101522  IT457 Cloud Computing  4/14/24 |

**Question 8:**

Code Used: -

using Azure.Storage.Blobs;

using System;

using System.IO;

using System.Security.Cryptography;

using System.Threading.Tasks;

*class* Program

{

    static async Task Upload\_Large\_FTB(string connection\_str, string filep, string cname, string bname)

    {

        BlobServiceClient blobclient = new BlobServiceClient(connection\_str);

        BlobContainerClient cclient = blobclient.GetBlobContainerClient(cname);

        BlobClient blobClient = cclient.GetBlobClient(bname);

        long blockS = 4 \* 1024 \* 1024; // 4MB block size

        byte[] buff = new byte[blockS];

        using (FileStream fileStream = File.OpenRead(filep))

        {

            long file\_size = fileStream.Length;

            long byteloaded = 0;

            string[] blkIds = new string[(int)Math.Ceiling((double)file\_size / blockS)];

            while (byteloaded < file\_size)

            {

                int bytesRead = await fileStream.ReadAsync(buff, 0, (int)Math.Min(blockS, file\_size - byteloaded));

                using (MD5 md5 = MD5.Create())

                // Calculate MD5 hash of the block

                {

                    byte[] blkHash = md5.ComputeHash(buff, 0, bytesRead);

                    string blkId = Convert.ToBase64String(blkHash);

                    // Upload the block

                    using (MemoryStream mstream = new MemoryStream(buff, 0, bytesRead))

                    {

                        await blobClient.StageBlockAsync(blkId, mstream);

                    }

                    blkIds[byteloaded / blockS] = blkId;

                }

                byteloaded += bytesRead;

            }

            // Commit the list of blocks

            await blobClient.CommitBlockListAsync(blkIds);

        }

        Console.WriteLine("File uploaded successfully.");

    }

    static async Task Main(string[] args)

    {

        string connection\_str = "<your\_connection\_string>";

        string filep = "<path\_to\_your\_file>";

        string cname = "<your\_container\_name>"; // container name

        string bname = "<your\_blob\_name>"; // blob name

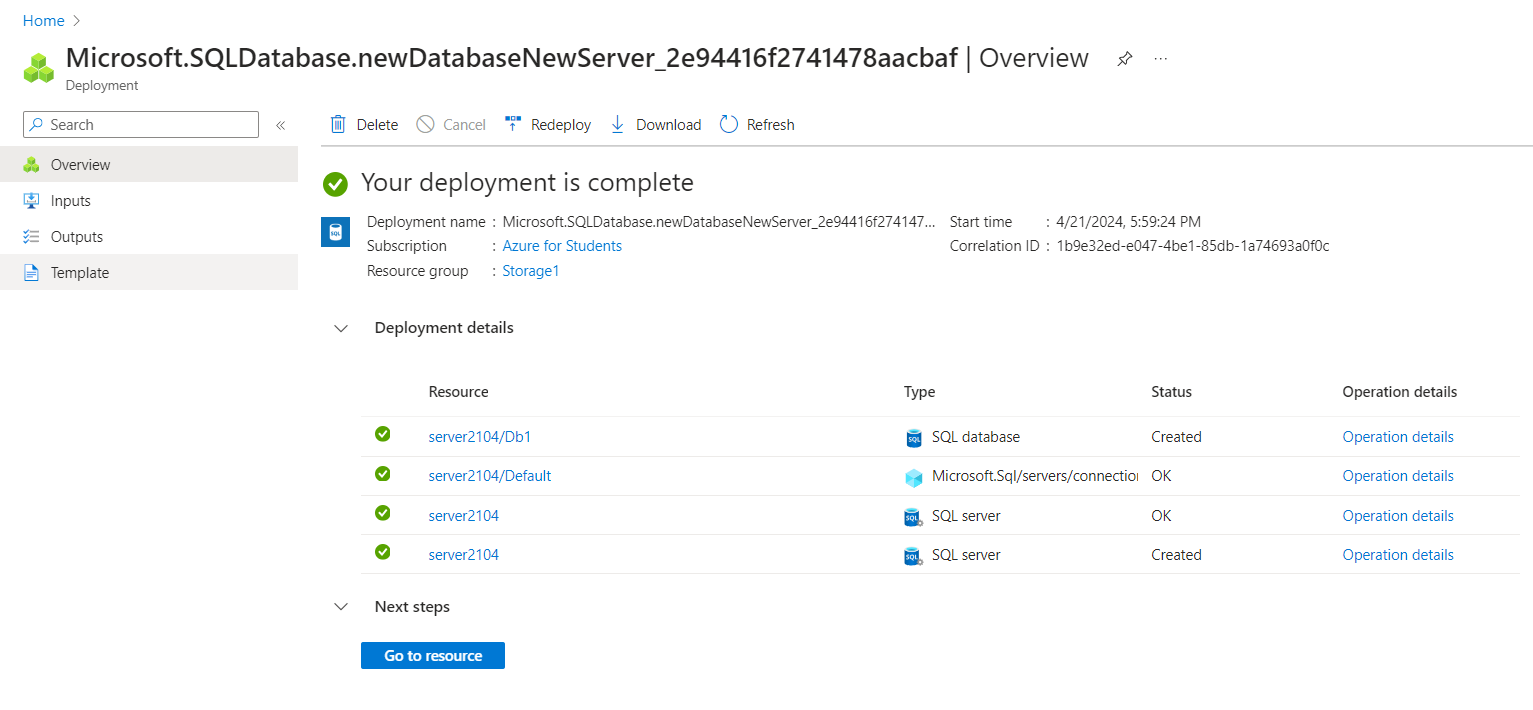
        await Upload\_Large\_FTB(connection\_str, filep, cname, bname);

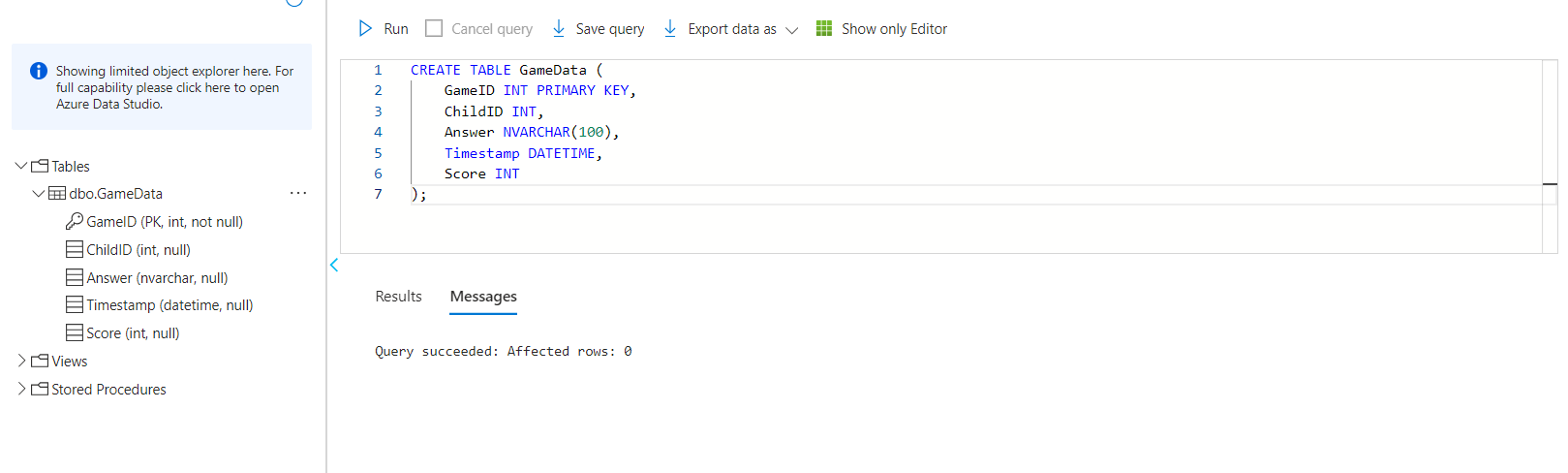
    }

}

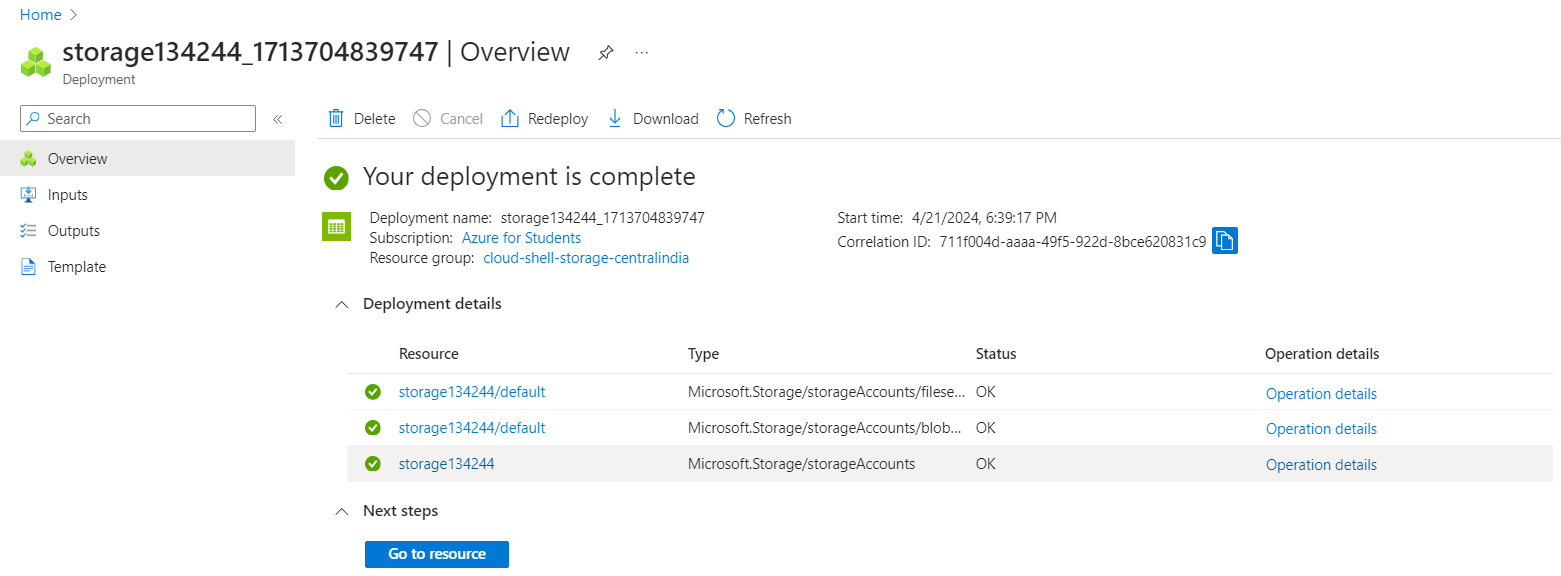
**Question 11: -**

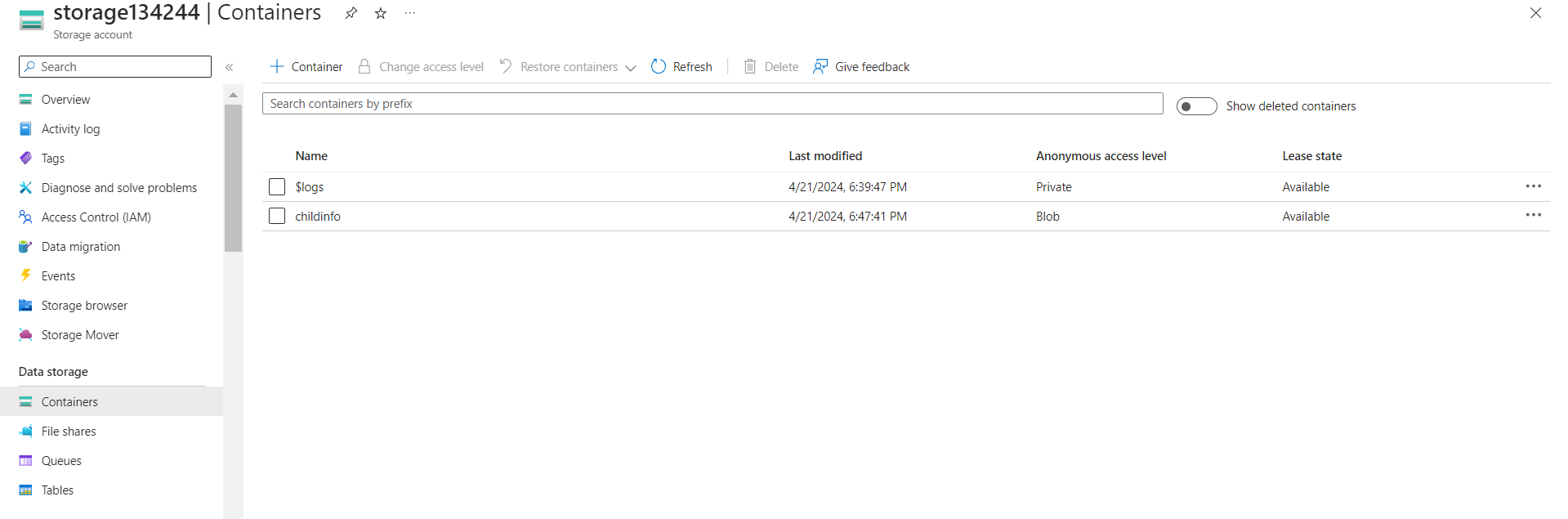
Creating SQL database: -



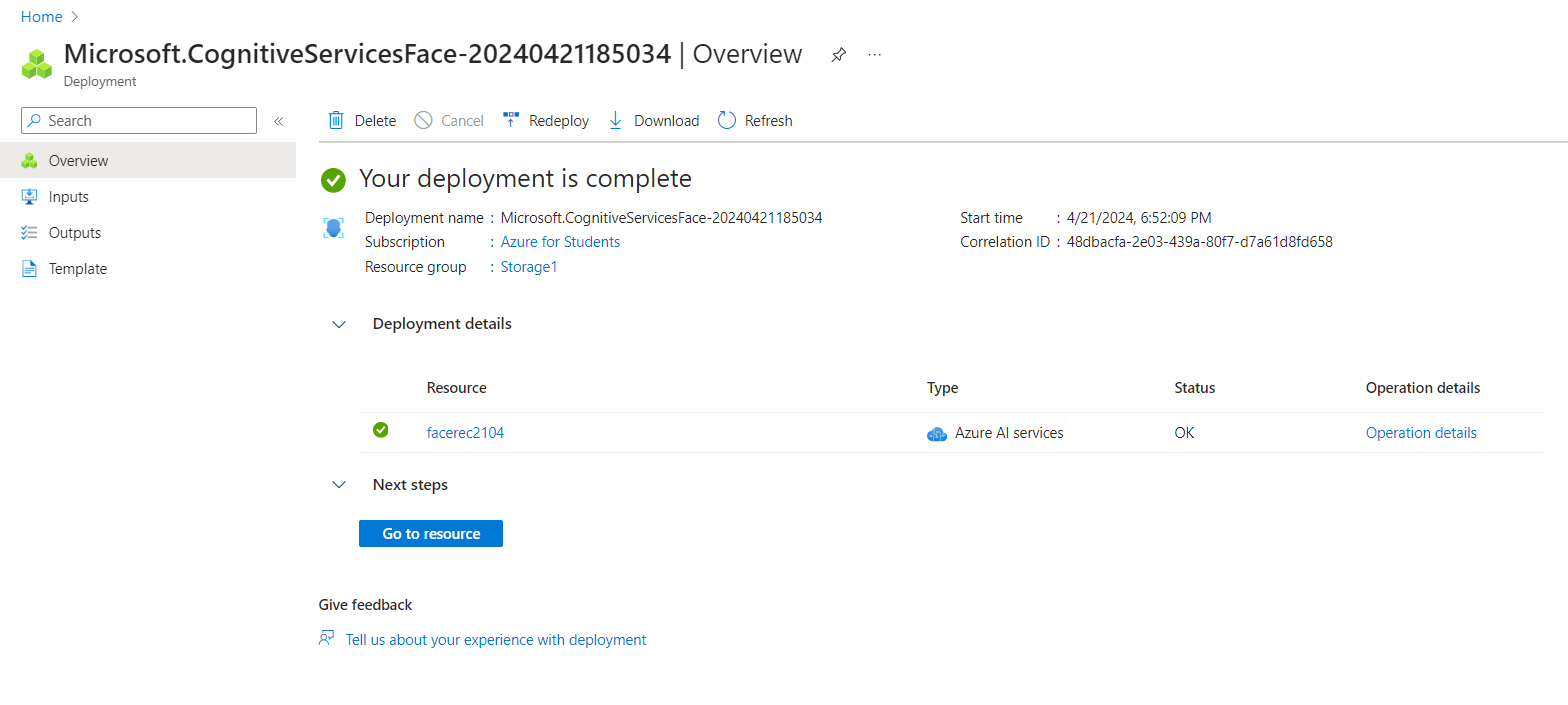


Creating blob storage: -

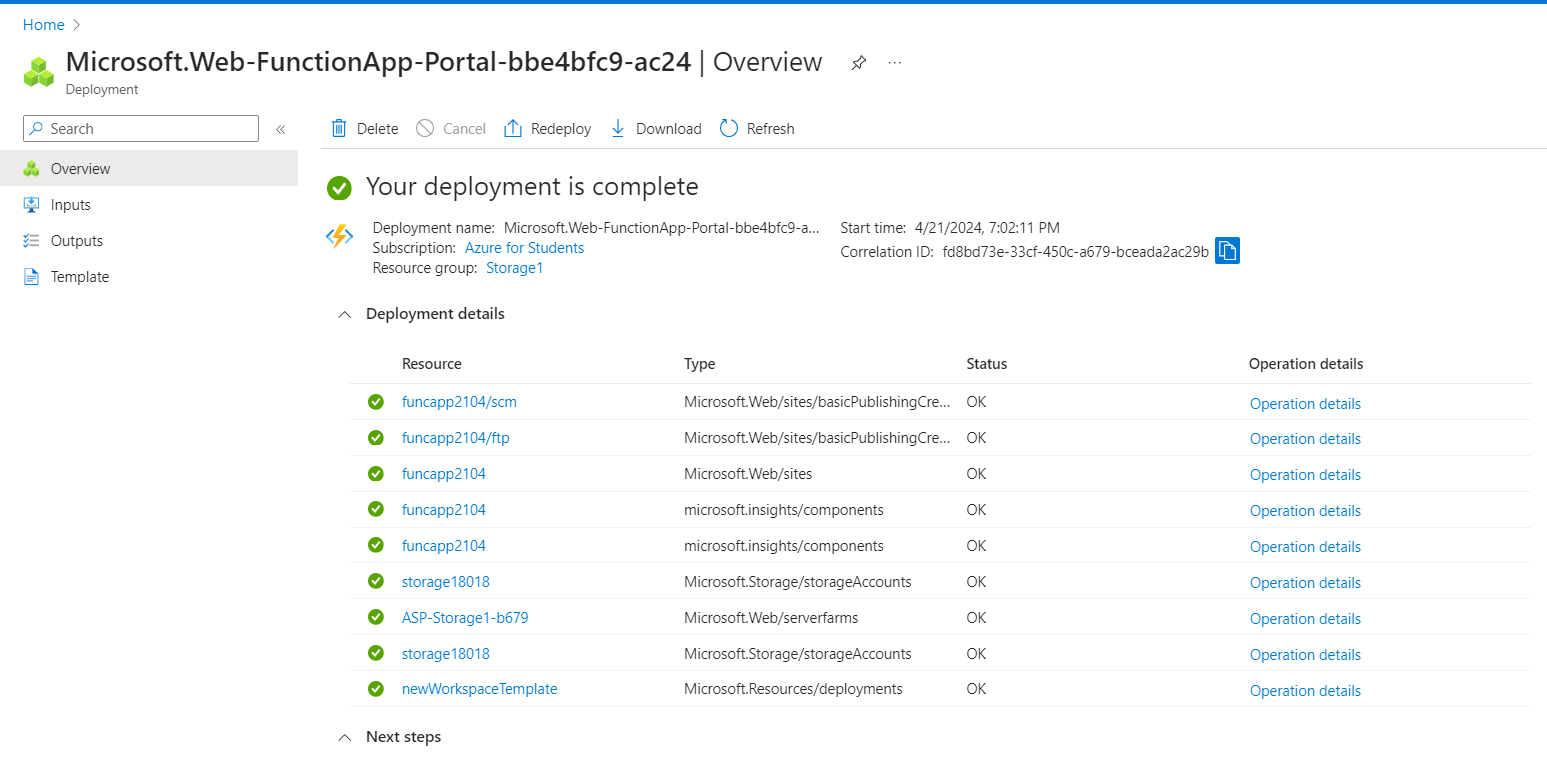




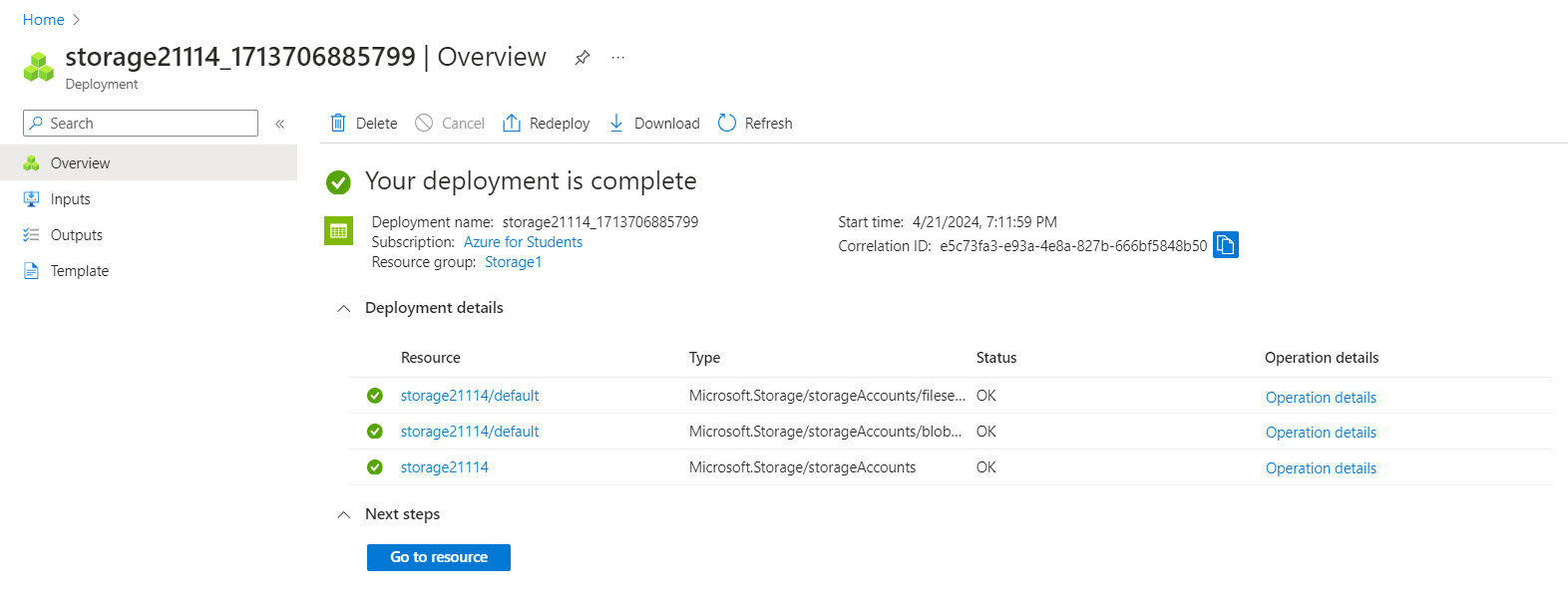
Creating Face API: -

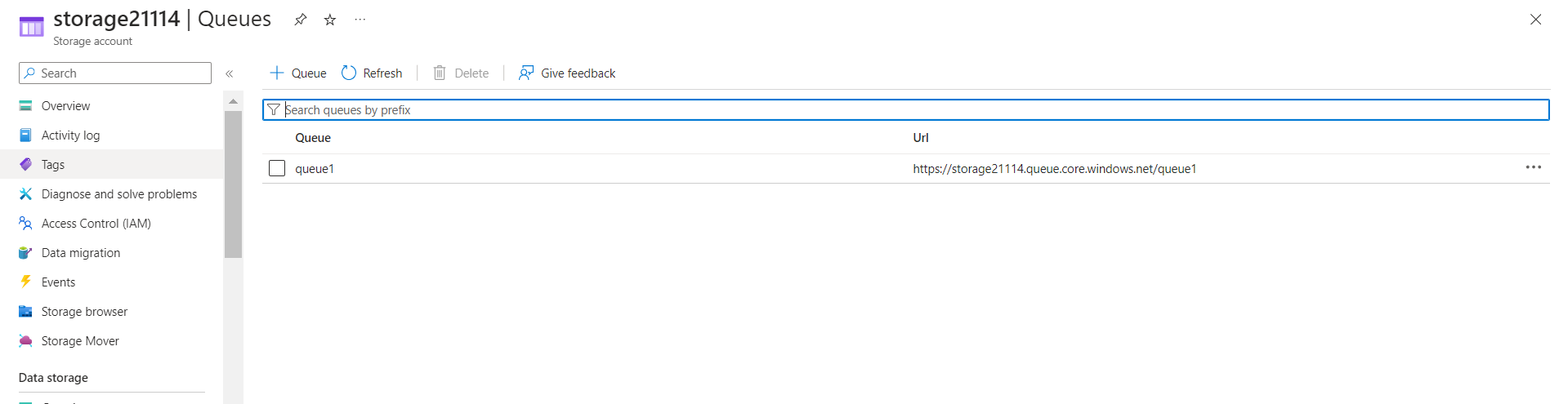


Creating Function App: -

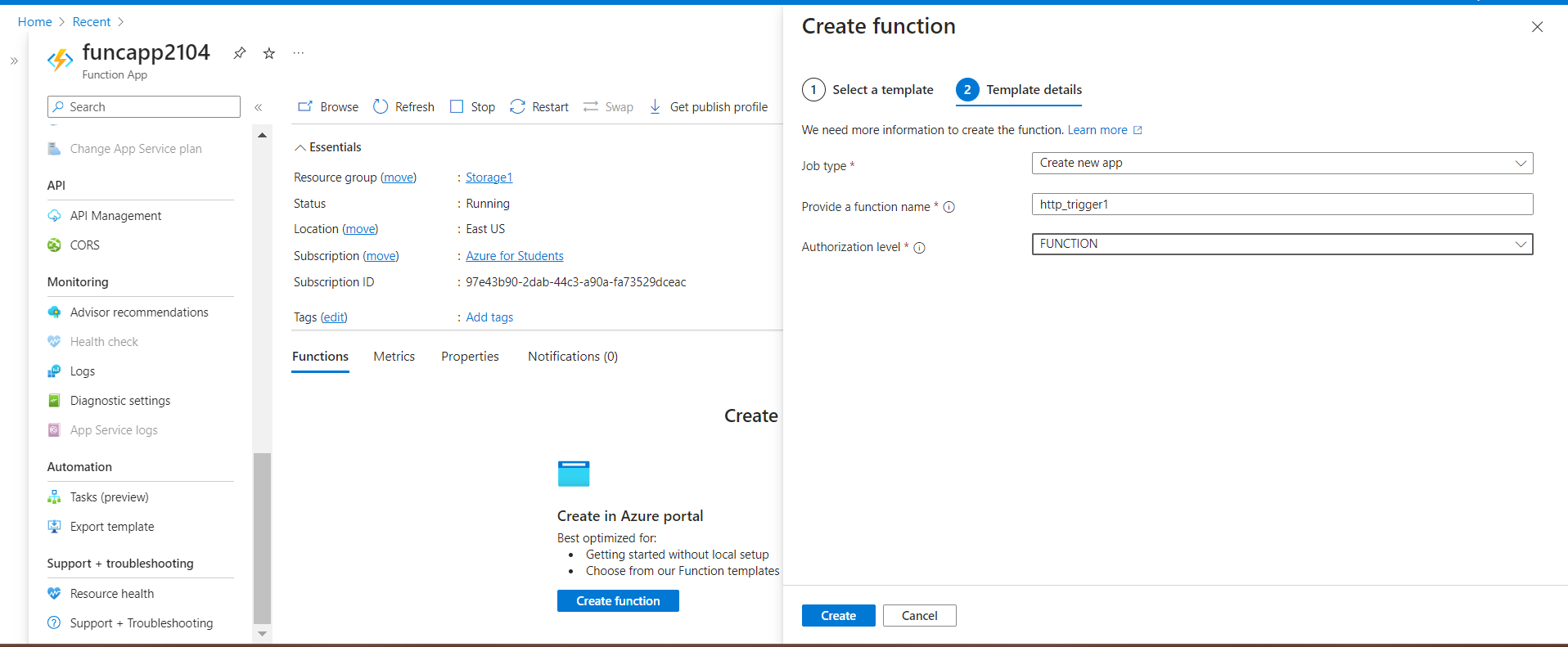


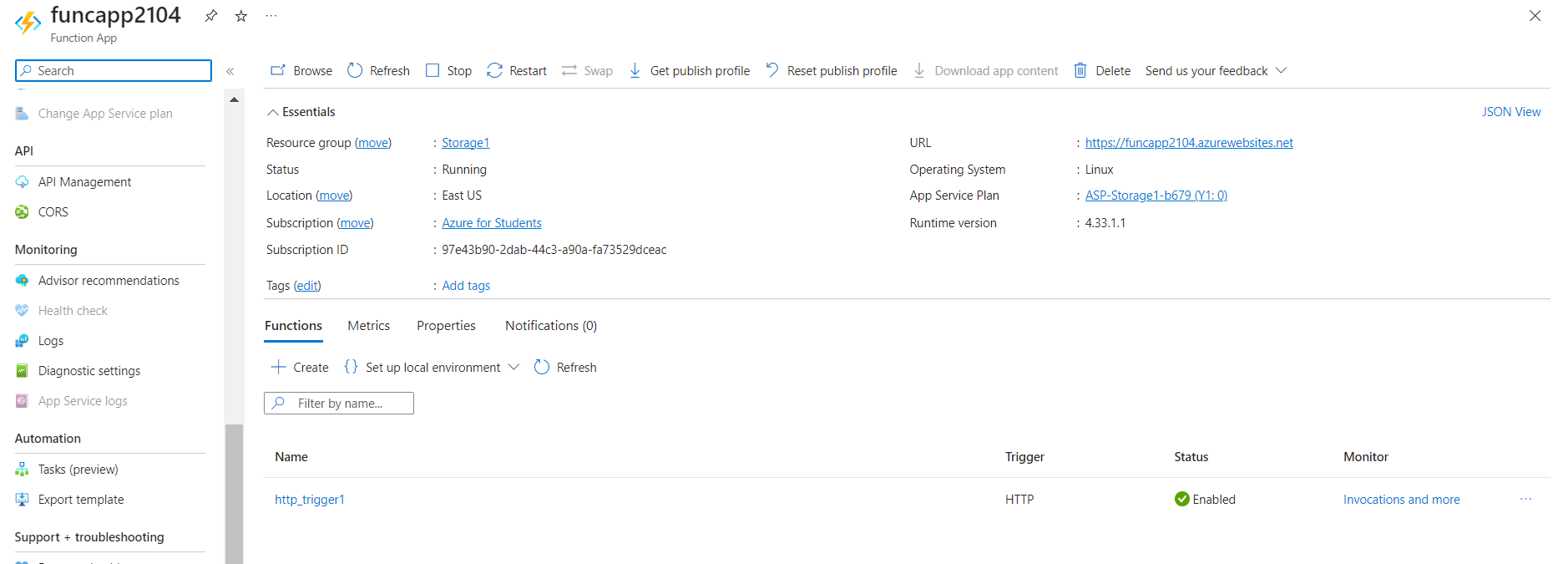
Creating Queue storage: -

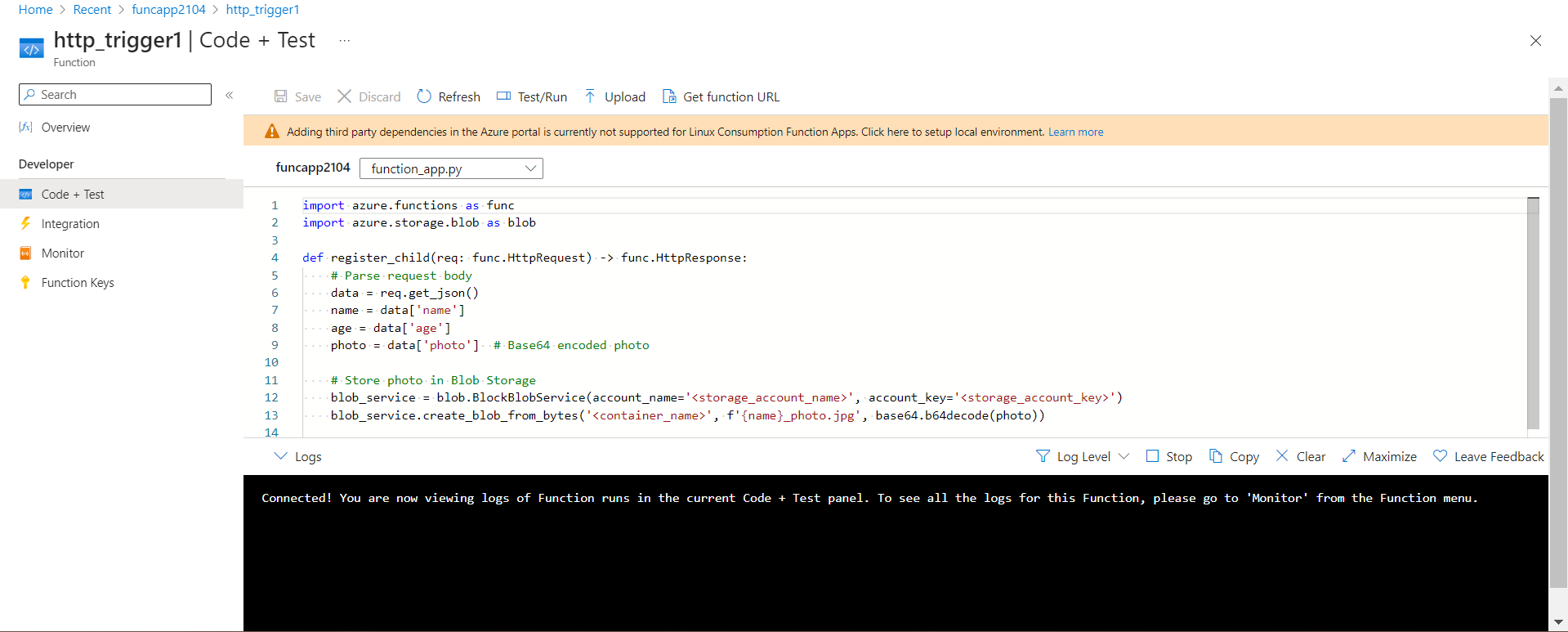




Creating function in function app: -







Result: -

