Part A :

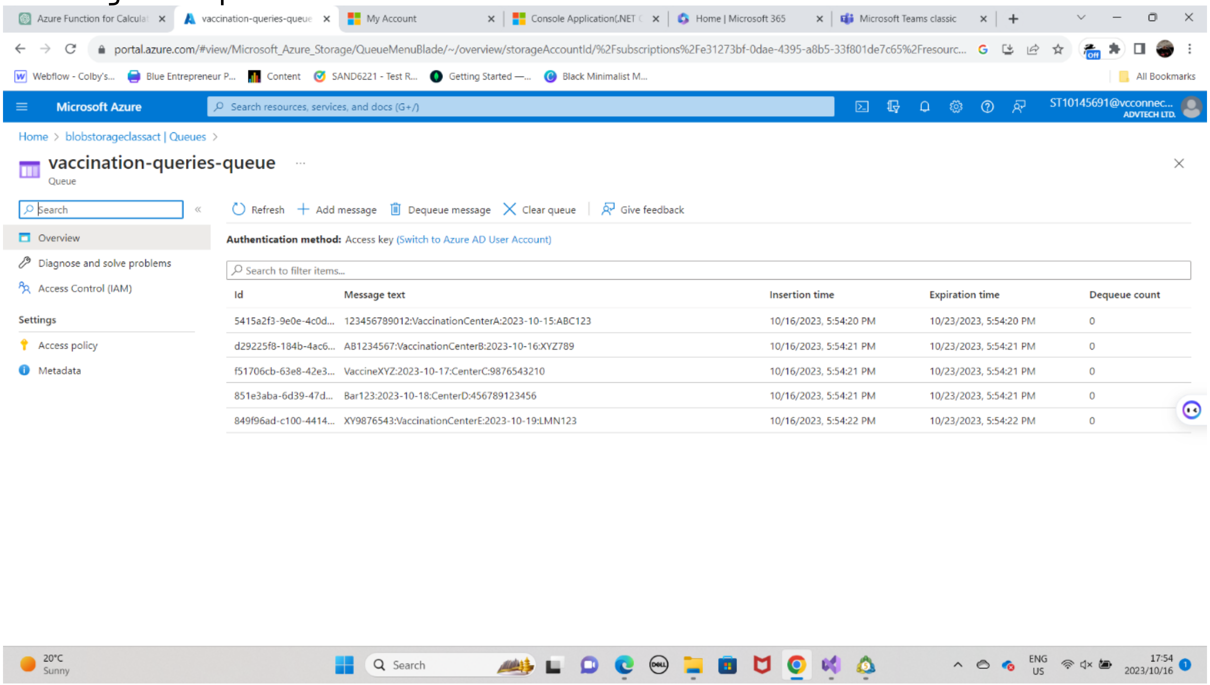
```
using Azure.Storage.Queues;
using ConsoleApp1;
using System;
using System.Numerics;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    internal class QueueService
    {
        private readonly QueueClient _queueClient;

        public QueueService(string connectionString)
        {
            _queueClient = new QueueClient(connectionString, "vaccination-
queries-queue");
        }

        public async Task SendMessageToQueueAsync(string message)
        {
            await _queueClient.SendMessageAsync(message);
            Console.WriteLine("Message sent to the queue: " + message);
        }
    }
}
```

```
{
  "ConnectionStrings": {
    "AzureStorage":
"DefaultEndpointsProtocol=https;AccountName=blobstorageclassact;AccountKey=t+P267
TW4YSi445r50FubSAA4ualopQlTakCPQXge1WS0/qTyKNj7n/Wmqn8EQwNbdZVaH0APMo++AStnYr+Dw=
=;EndpointSuffix=core.windows.net"

  }
}
```

```
using ConsoleApp1;
using System;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            string connectionString =
"DefaultEndpointsProtocol=https;AccountName=blobstorageclassact;AccountKey=t+P267
TW4YSi445r50FubSAA4ualopQlTakCPQXge1WS0/qTyKNj7n/Wmqn8EQwNbdZVaH0APMo++AStnYr+Dw=
=;EndpointSuffix=core.windows.net"; // Replace with your actual connection string
            var queueService = new QueueService(connectionString);

            // Send the first example message (South African ID format)
            string message1 = "123456789012:VaccinationCenterA:2023-10-
15:ABC123";
            await queueService.SendMessageToQueueAsync(message1);

            // Send the second example message (Passport Number format)
            string message2 = "AB1234567:VaccinationCenterB:2023-10-16:XYZ789";
            await queueService.SendMessageToQueueAsync(message2);
```

```
                // Send the third example message (Different Format 1)
                string message3 = "VaccineXYZ:2023-10-17:CenterC:9876543210";
                await queueService.SendMessageToQueueAsync(message3);

                // Send the fourth example message (Different Format 2)
                string message4 = "Bar123:2023-10-18:CenterD:456789123456";
                await queueService.SendMessageToQueueAsync(message4);

                // Send the fifth example message (Another Passport Number format)
                string message5 = "XY9876543:VaccinationCenterE:2023-10-19:LMN123";
                await queueService.SendMessageToQueueAsync(message5);
            }
        }
}
```

5 Messages in queue:

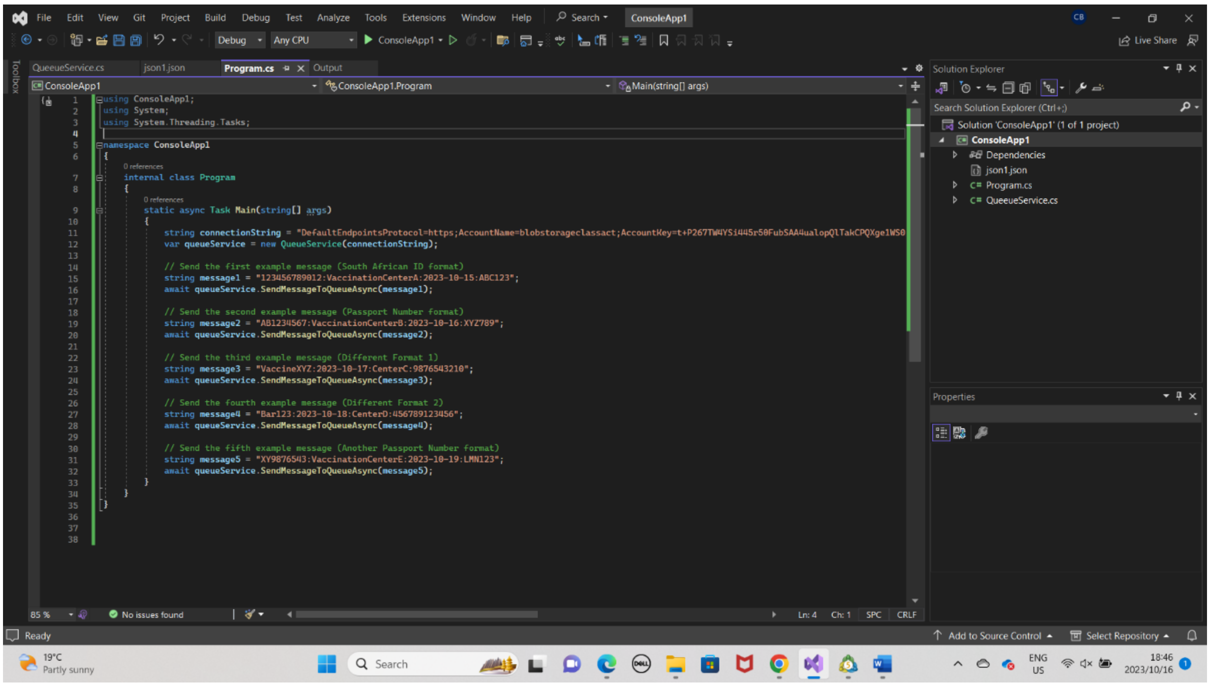


vaccination-queries-queue
Queue

Authentication method: Access key (Switch to Azure AD User Account)

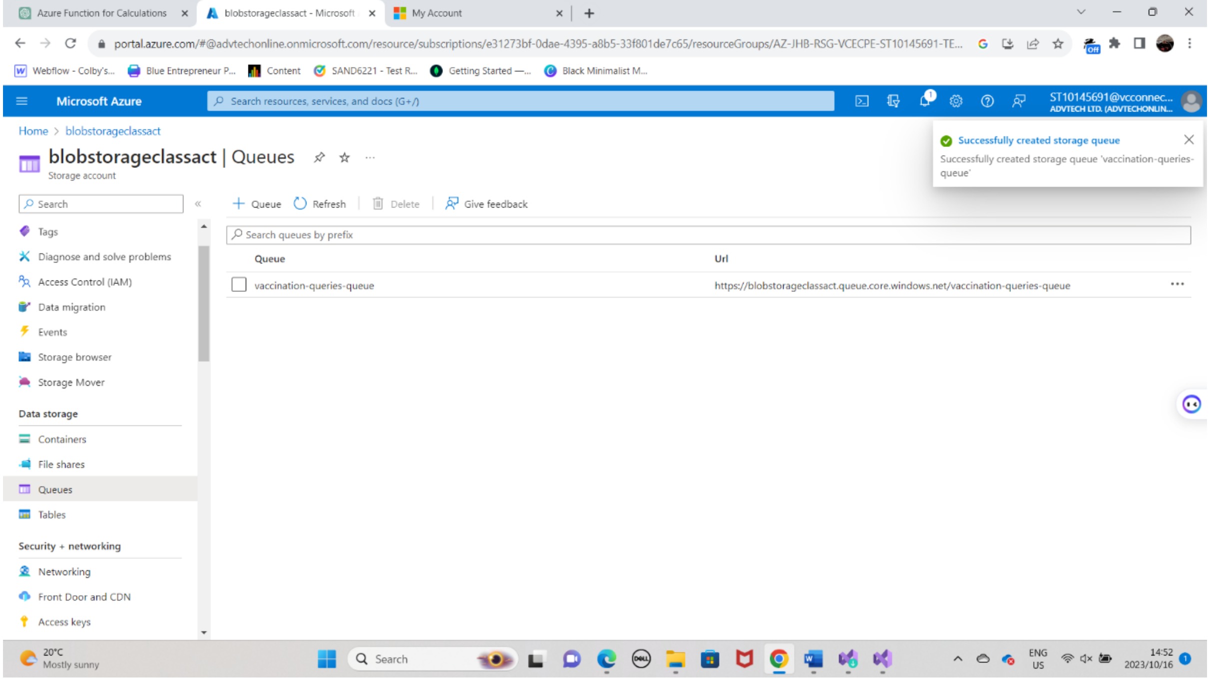
Id	Message text	Insertion time	Expiration time	Dequeue count
5415a2f3-9e0e-4c0d...	123456789012:VaccinationCenterA:2023-10-15:ABC123	10/16/2023, 5:54:20 PM	10/23/2023, 5:54:20 PM	0
d29225f8-184b-4ac6...	AB1234567:VaccinationCenterB:2023-10-16:XYZ789	10/16/2023, 5:54:21 PM	10/23/2023, 5:54:21 PM	0
f51706cb-63e8-42e3...	VaccineXYZ:2023-10-17:CenterC:9876543210	10/16/2023, 5:54:21 PM	10/23/2023, 5:54:21 PM	0
851e3aba-6d39-47d...	Bar123:2023-10-18:CenterD:456789123456	10/16/2023, 5:54:21 PM	10/23/2023, 5:54:21 PM	0
849f96ad-c100-4414...	XY9876543:VaccinationCenterE:2023-10-19:LMN123	10/16/2023, 5:54:22 PM	10/23/2023, 5:54:22 PM	0

https://portal.azure.com/#@advtechonline.onmicrosoft.com/resource/subscriptions/e31273bf-0dae-4395-a8b5-33f801de7c65/resourceGroups/AZ-JHB-RSG-VCECPE-ST10145691-TER/providers/Microsoft.Storage/storageAccounts/blobstorageclassact/overview



```
using ConsoleApp1;
using System;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            string connectionString = "DefaultEndpointsProtocol=https;AccountName=blobstorageclassact;AccountKey=t+P267TW4YSi445r50FubSAA4ualopQlTakCPQXge1WS0
            var queueService = new QueueService(connectionString);

            // Send the first example message (South African ID format)
            string message1 = "123456789012:VaccinationCenterA:2023-10-15:ABC123";
            await queueService.SendMessageToQueueAsync(message1);

            // Send the second example message (Passport Number format)
            string message2 = "AB1234567:VaccinationCenterB:2023-10-16:XYZ789";
            await queueService.SendMessageToQueueAsync(message2);

            // Send the third example message (Different Format 1)
            string message3 = "VaccineXYZ:2023-10-17:CenterC:9876543210";
            await queueService.SendMessageToQueueAsync(message3);

            // Send the fourth example message (Different Format 2)
            string message4 = "Bar123:2023-10-18:CenterD:456789123456";
            await queueService.SendMessageToQueueAsync(message4);

            // Send the fifth example message (Another Passport Number format)
            string message5 = "XY9876543:VaccinationCenterE:2023-10-19:LMN123";
            await queueService.SendMessageToQueueAsync(message5);
        }
    }
}
```

Storage queue created:



portal.azure.com/#@advtechonline.onmicrosoft.com/resource/subscriptions/e31273bf-0dae-4395-a8b5-33f801de7c65/resourceGroups/AZ-JHB-RSG-VCECPE-ST10145691-TE...

Home > blobstorageclassact

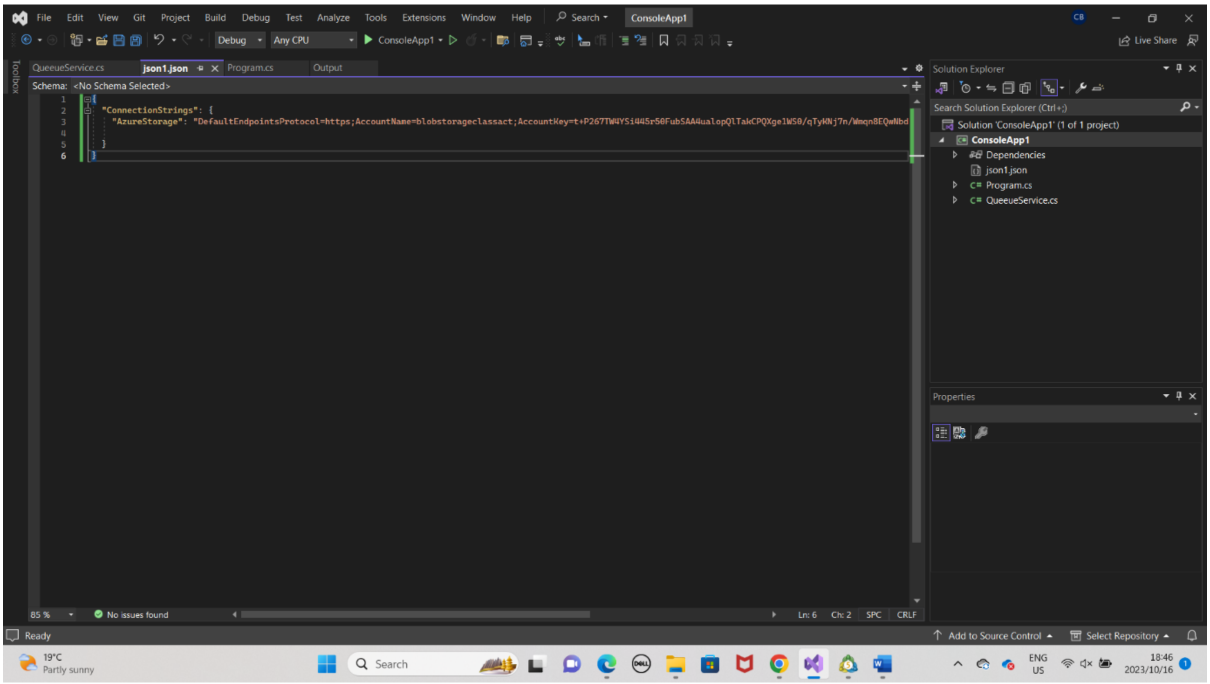
blobstorageclassact | Queues

Storage account

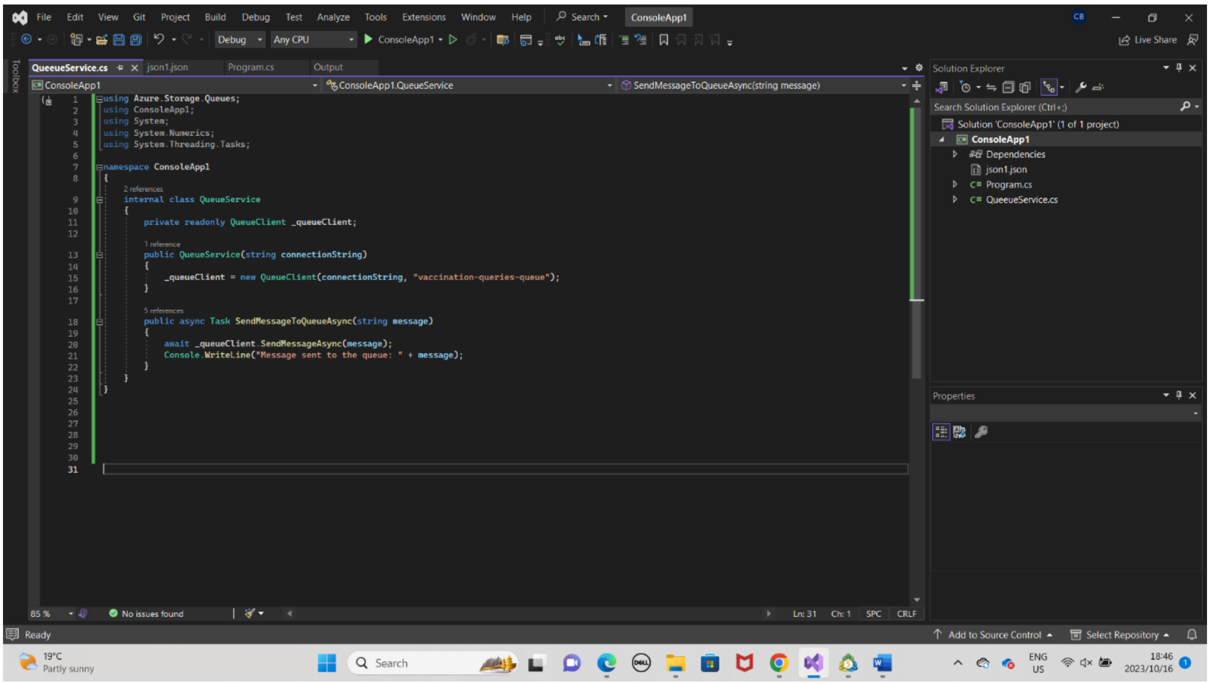
Successfully created storage queue

Successfully created storage queue 'vaccination-queries-queue'

Queue	Url
vaccination-queries-queue	https://blobstorageclassact.queue.core.windows.net/vaccination-queries-queue



```
{
  "ConnectionStrings": {
    "AzureStorage": "DefaultEndpointsProtocol=https;AccountName=blobstorageclassact;AccountKey=t+P267TW4YSi445r50FubSAA4ualopQlTakCPQXge1WS0/qTyKNj7n/Wmqn8EQwNbd

  }
}
```

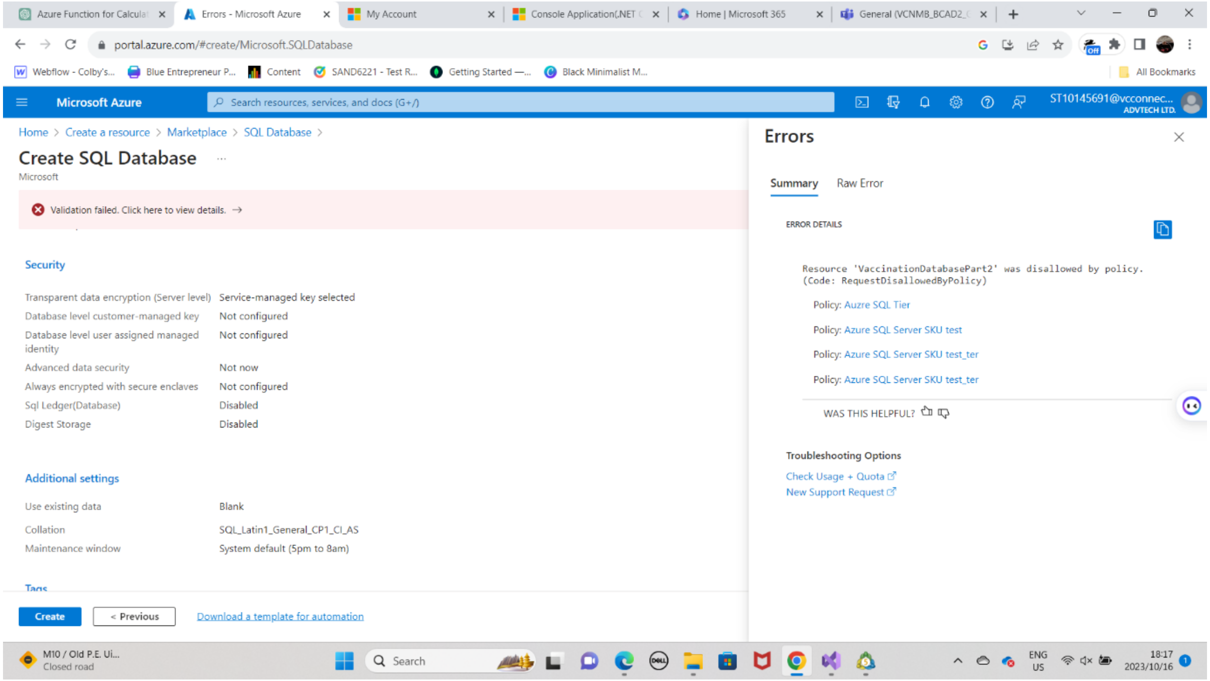


```
using Azure.Storage.Queues;
using ConsoleApp1;
using System;
using System.Numerics;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    2 references
    internal class QueueService
    {
        private readonly QueueClient _queueClient;

        1 reference
        public QueueService(string connectionString)
        {
            _queueClient = new QueueClient(connectionString, "vaccination-queries-queue");
        }

        5 references
        public async Task SendMessageToQueueAsync(string message)
        {
            await _queueClient.SendMessageAsync(message);
            Console.WriteLine("Message sent to the queue: " + message);
        }
    }
}
```

Part B :

Changing from http trigger to queee trigger : function.json file

```
{
  "bindings": [
    {
      "name": "myQueueItem",
      "type": "queueTrigger",
      "direction": "in",
      "queueName": "  vaccination-queries-queue", // Replace with the name of your Azure Storage Queue
      "connection": "AzureWebJobsStorage"
    }
  ]
}
```



Create SQL Database

Microsoft

Validation failed. Click here to view details. →

Security

Transparent data encryption (Server level)	Service-managed key selected
Database level customer-managed key	Not configured
Database level user assigned managed identity	Not configured
Advanced data security	Not now
Always encrypted with secure enclaves	Not configured
Sql Ledger(Database)	Disabled
Digest Storage	Disabled

Additional settings

Use existing data	Blank
Collation	SQL_Latin1_General_CP1_CI_AS
Maintenance window	System default (5pm to 8am)

Tags

Errors

ERROR DETAILS

```
Resource 'VaccinationDatabasePart2' was disallowed by policy.
(Code: RequestDisallowedByPolicy)
```

Policy: Auzre SQL Tier

Policy: Azure SQL Server SKU test

Policy: Azure SQL Server SKU test_ter

Policy: Azure SQL Server SKU test_ter

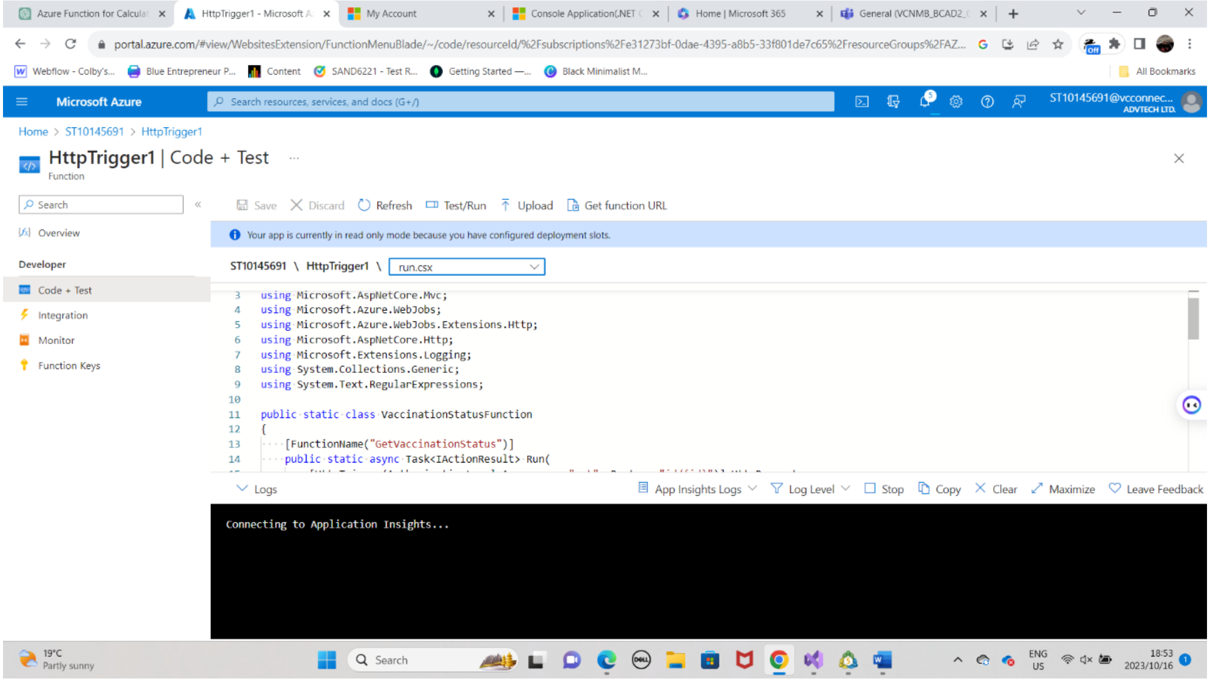
Troubleshooting Options

Check Usage + Quota

New Support Request

Function code:

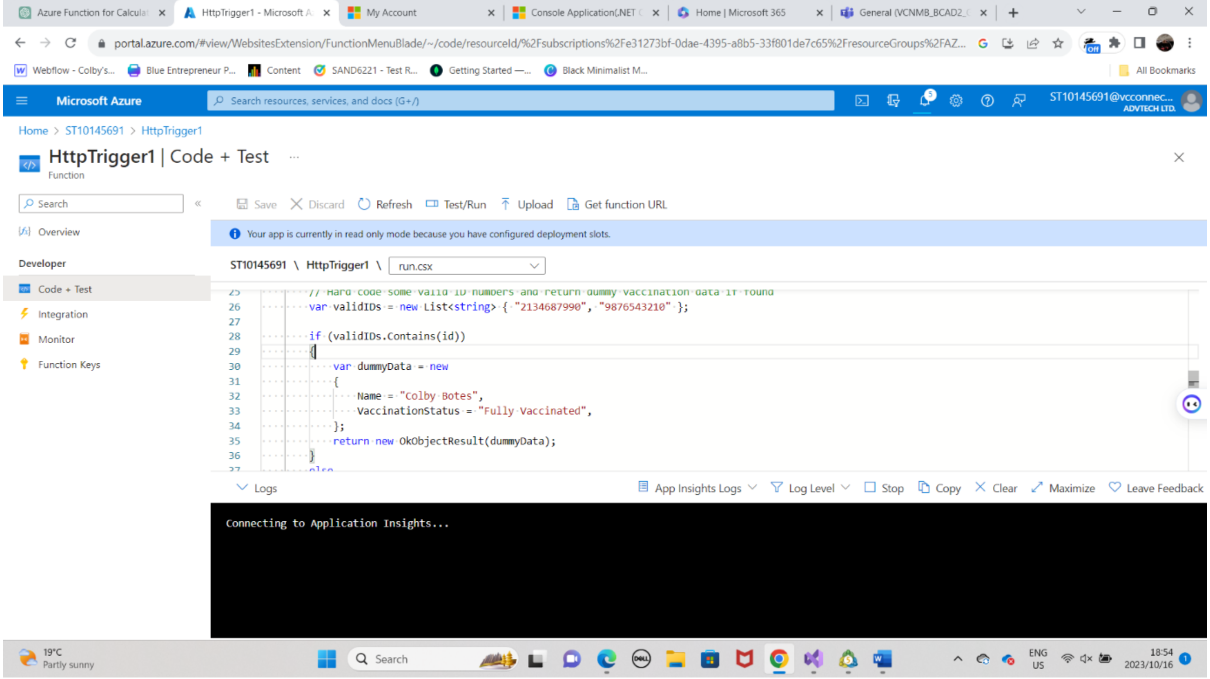
```
using System;

using Microsoft.Azure.WebJobs;

using Microsoft.Extensions.Logging;


public static void ProcessQueueMessage(

    [QueueTrigger("vaccination-queries-queue ")] string myQueueItem, // Match the parameter name with the one in function.json

    ILogger log)

{

    log.LogInformation($"C# Queue trigger function processed: {myQueueItem}");


    // Process the message here

    // Example: Parse the message and insert it into an Azure SQL Database

}
```



Your app is currently in read only mode because you have configured deployment slots.

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Extensions.Http;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Logging;
using System.Collections.Generic;
using System.Text.RegularExpressions;

public static class VaccinationStatusFunction
{
    [FunctionName("GetVaccinationStatus")]
    public static async Task<IActionResult> Run(
```

```
Connecting to Application Insights...
```



Your app is currently in read only mode because you have configured deployment slots.

ST10145691 \ HttpTrigger1 \ run.csx

```
        // Hard code some valid ID numbers and return dummy vaccination data if found
        var validIDs = new List<string> { "2134687990", "9876543210" };

        if (validIDs.Contains(id))
        {
            var dummyData = new
            {
                Name = "Colby Botes",
                VaccinationStatus = "Fully Vaccinated",
            };
            return new OkObjectResult(dummyData);
        }
        else
```

```
Connecting to Application Insights...
```

URL: https://st10145691.azurewebsites.net