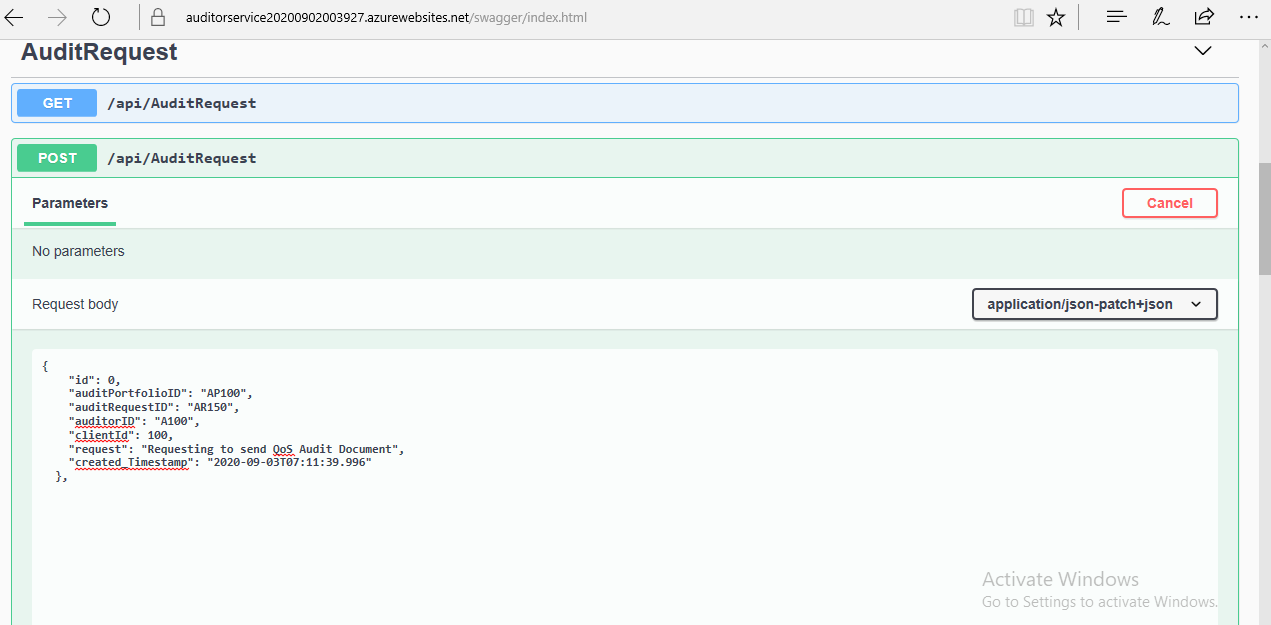
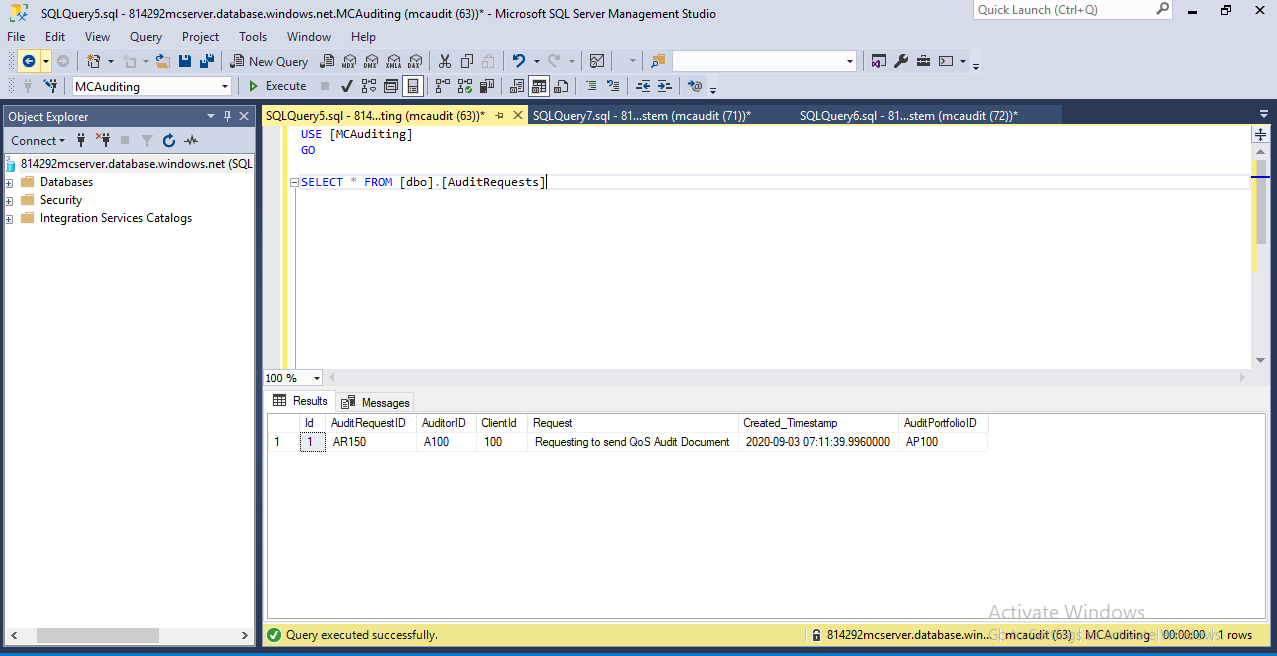
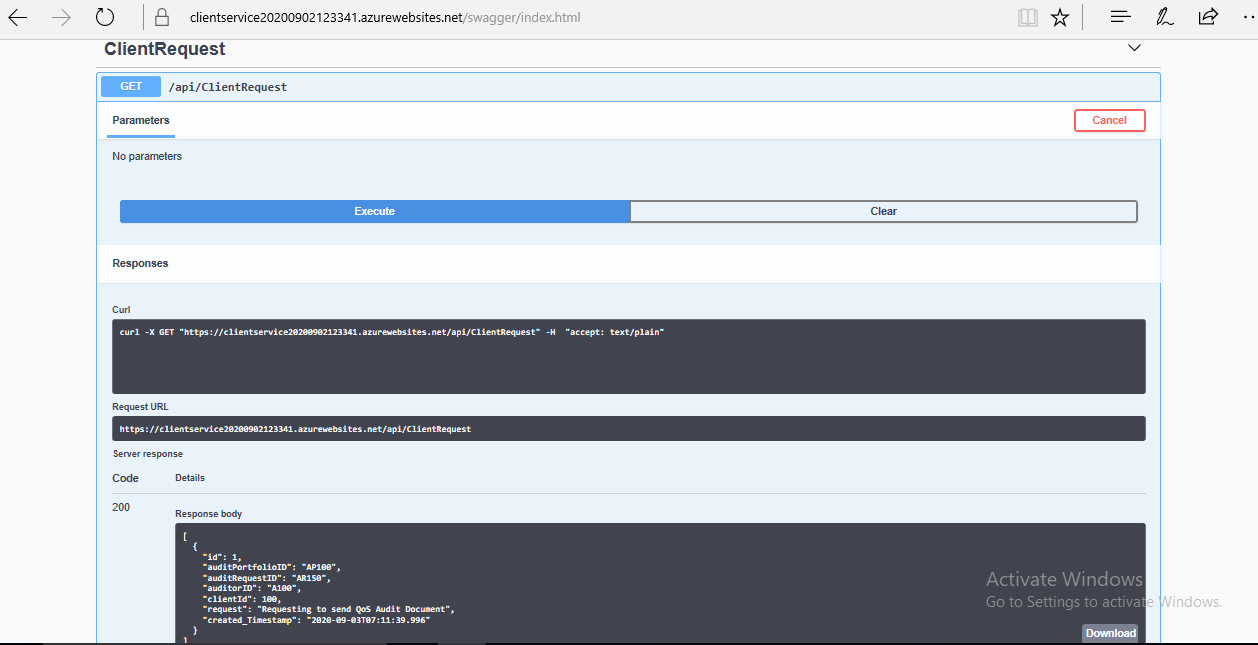
Audit send the request to Client



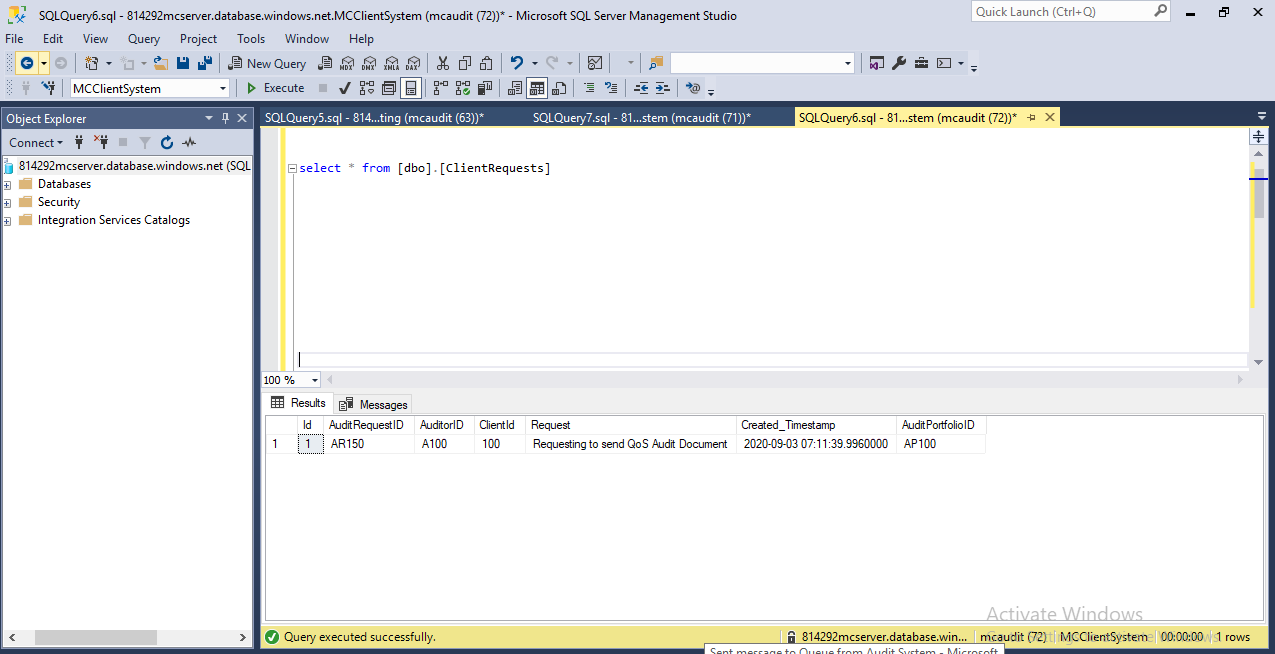
Data in the Audit database table :



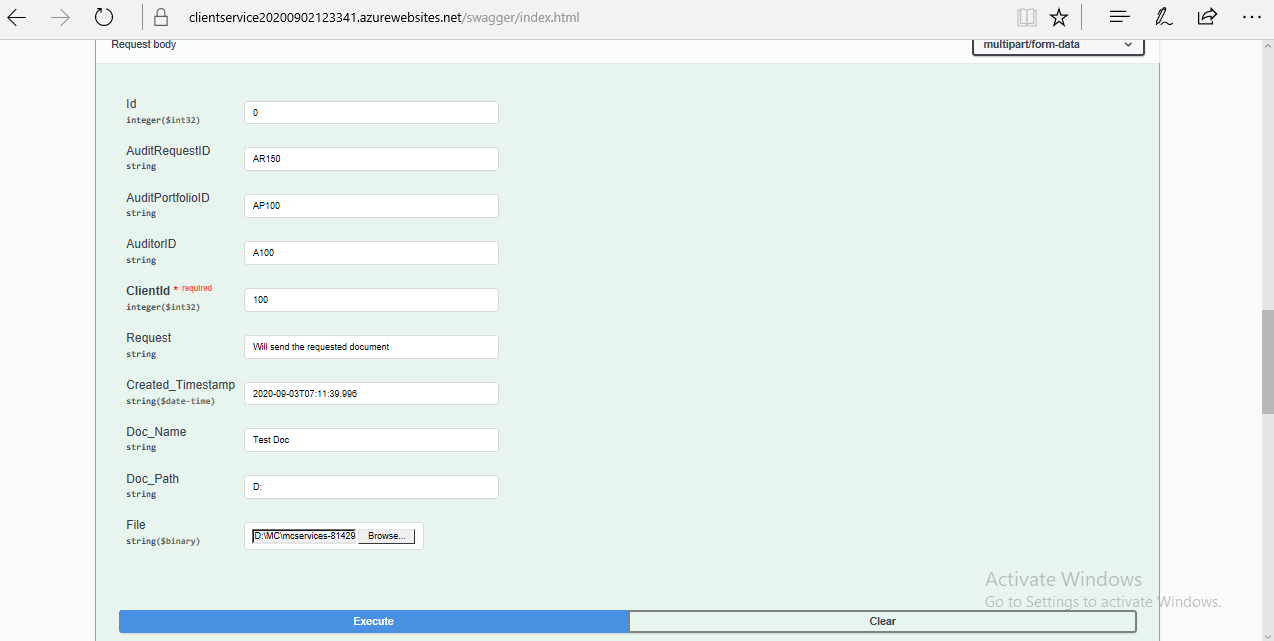
Client Service Get request result :

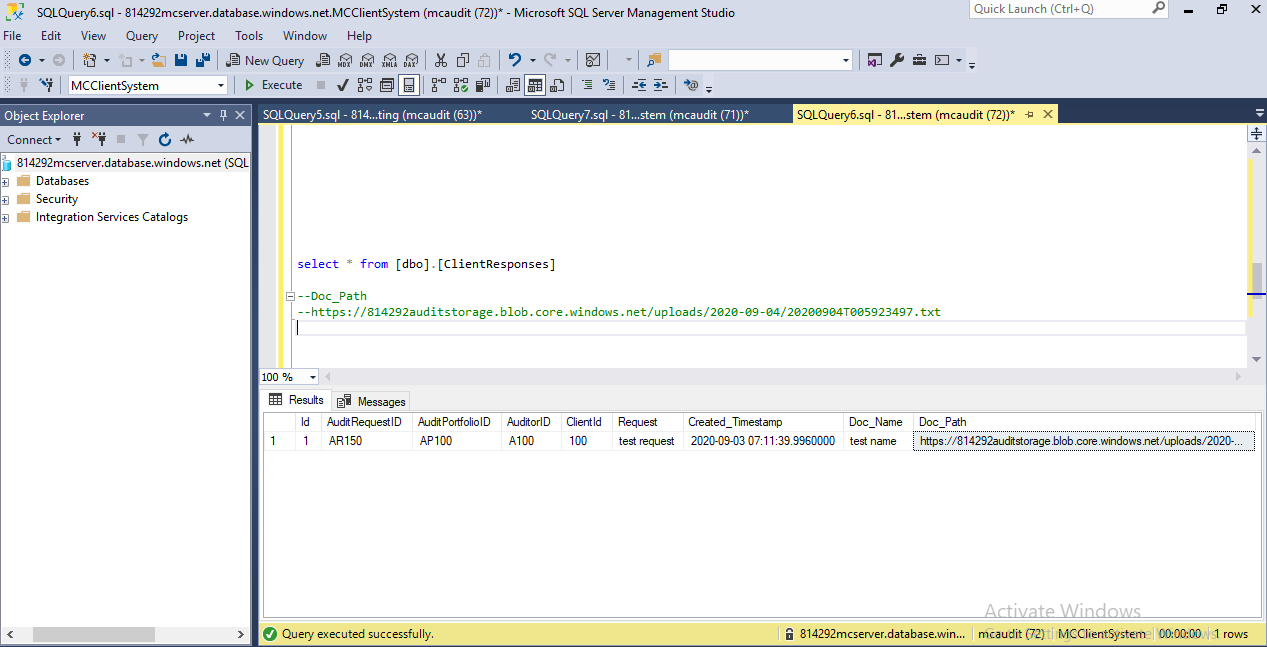


Data in the Client database :

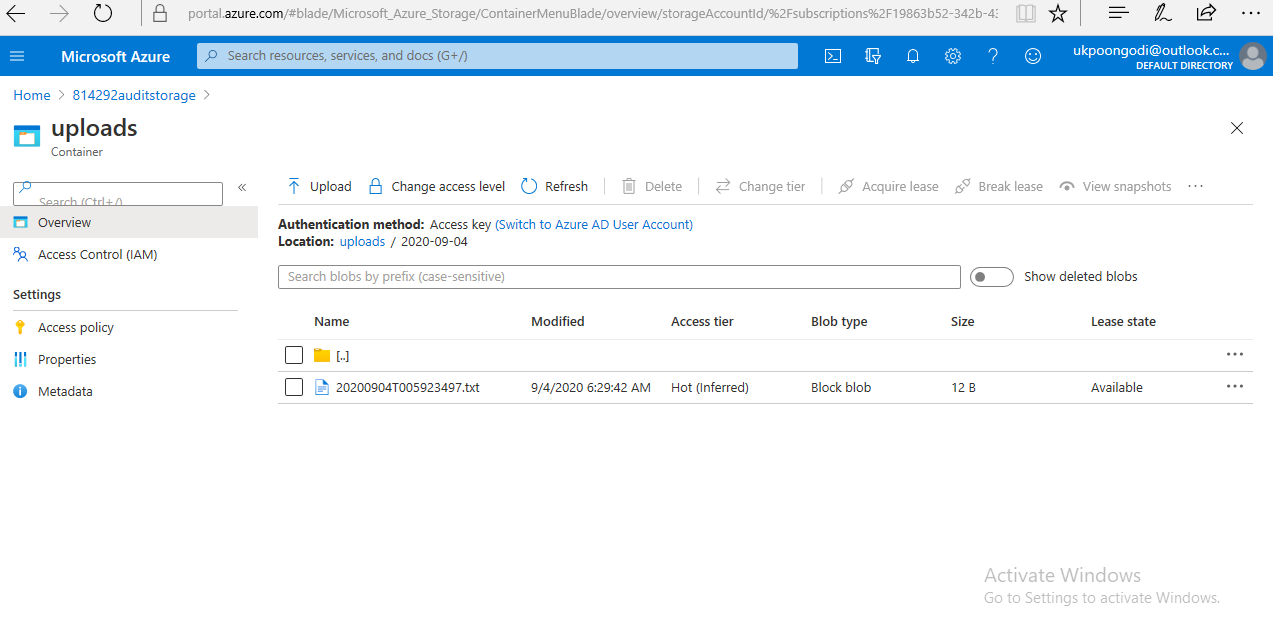


Uploading the file from Client in the Blob storage :

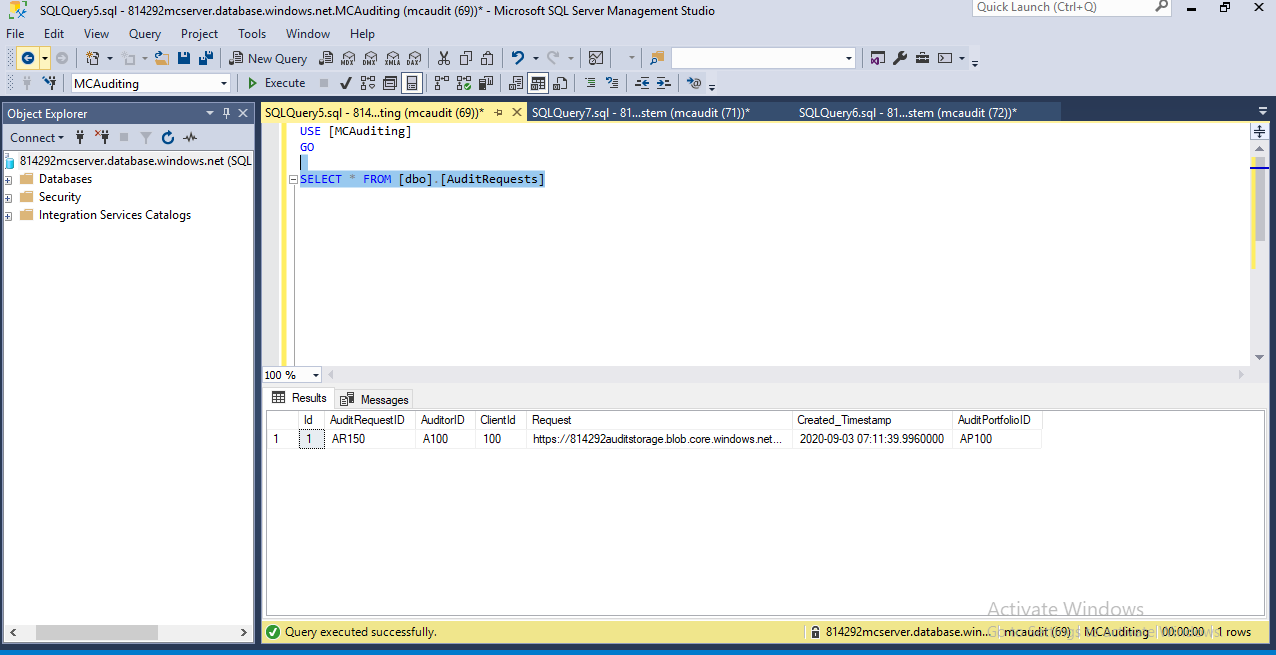




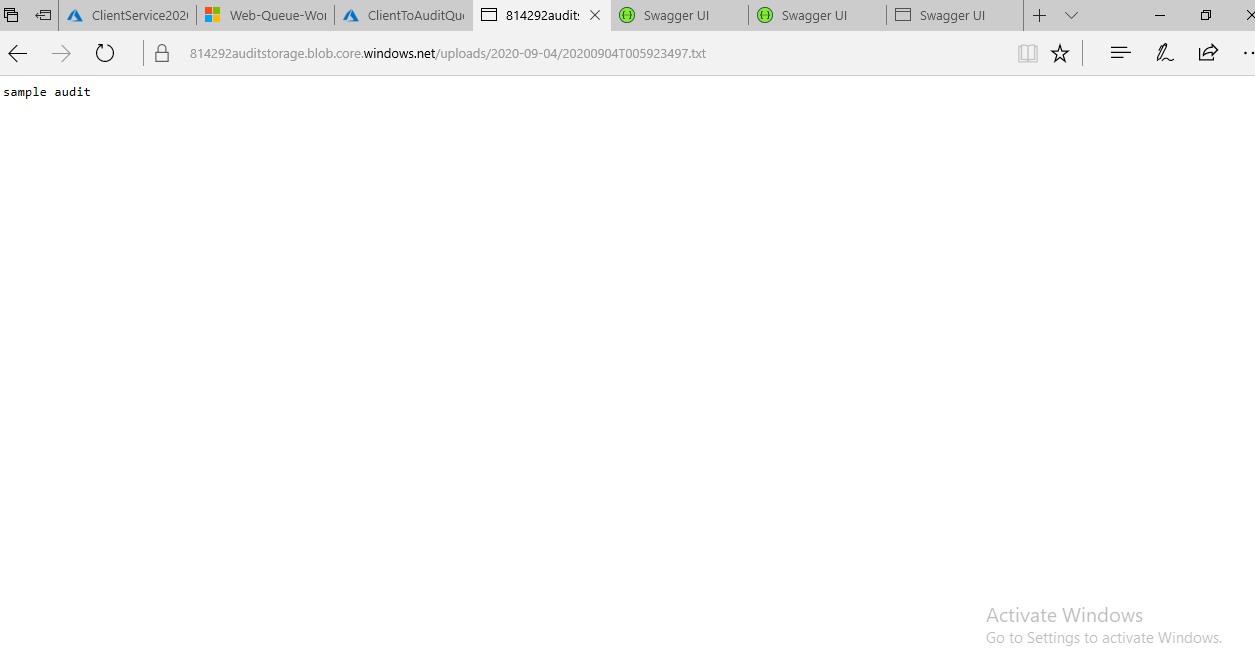
Uploaded file from Azure Blob storage :



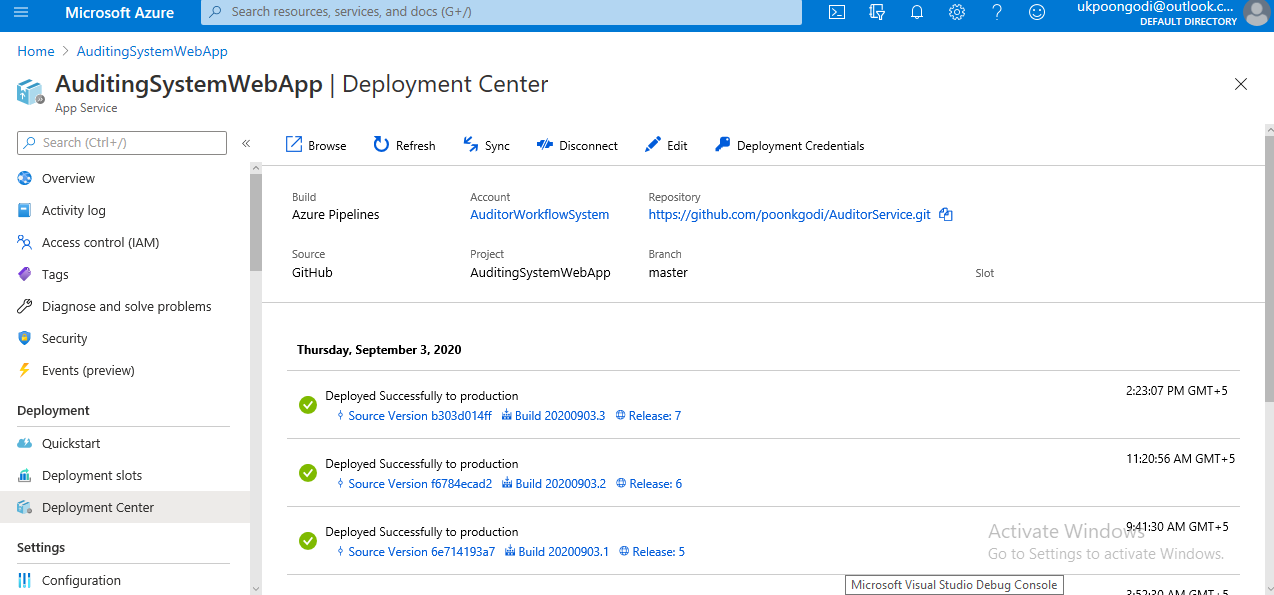
Uploaded Link sending from client from Audit through Service bus Queue using Function App :



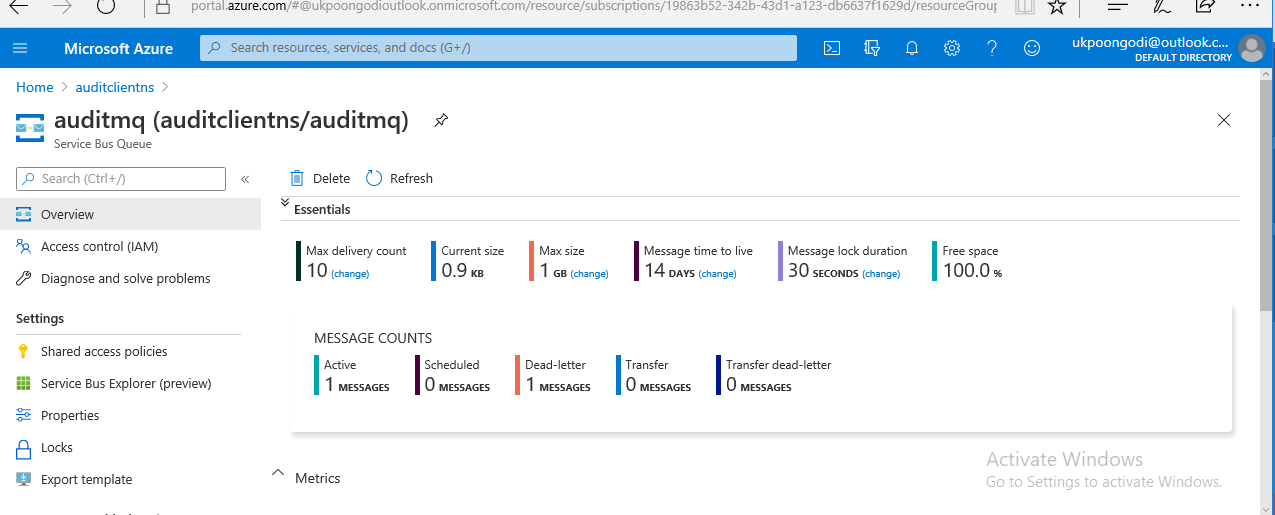
Auditor can download the file from Blob using the URI :



Implemented CI/CD



Sent message to Queue from Audit System :



<https://814292auditstorage.blob.core.windows.net/uploads/2020-09-01/20200901T113030741.txt>

**Command line instructions**

You can also upload existing files from your computer using the instructions below.

**Git global setup**

git config --global user.name "Poongodi"

git config --global user.email "poonkgodi.u@cognizant.com"

**Create a new repository**

git clone http://172.18.2.18/Poonkgodi/test\_project.git

cd test\_project

touch README.md

git add README.md

git commit -m "add README"

git push -u origin master

**Push an existing folder**

cd existing\_folder

git init

git remote add origin http://172.18.2.18/Poonkgodi/test\_project.git

git add .

git commit -m "Initial commit"

git push -u origin master

**Push an existing Git repository**

cd existing\_repo

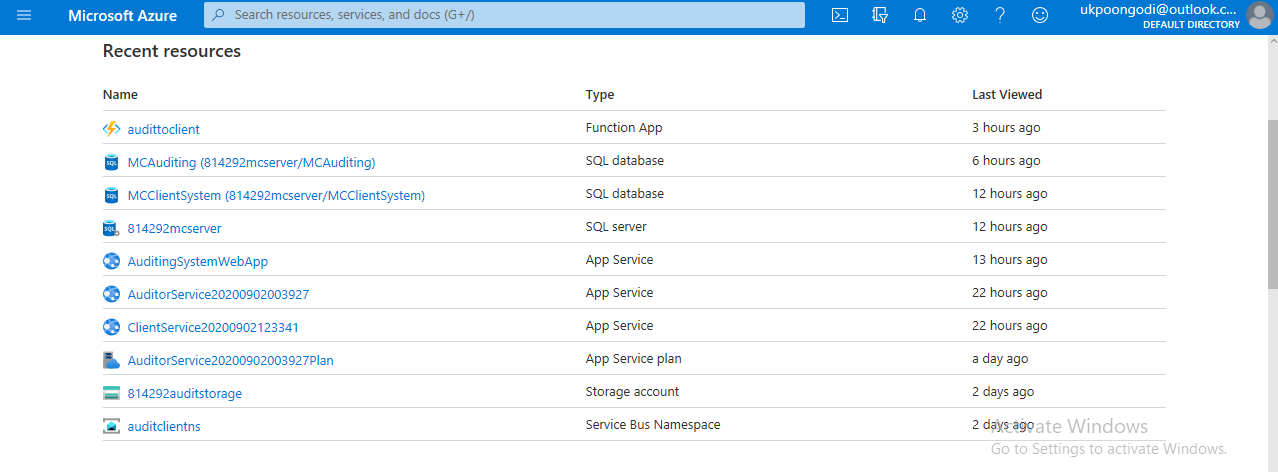
git remote rename origin old-origin

git remote add origin http://172.18.2.18/Poonkgodi/test\_project.git

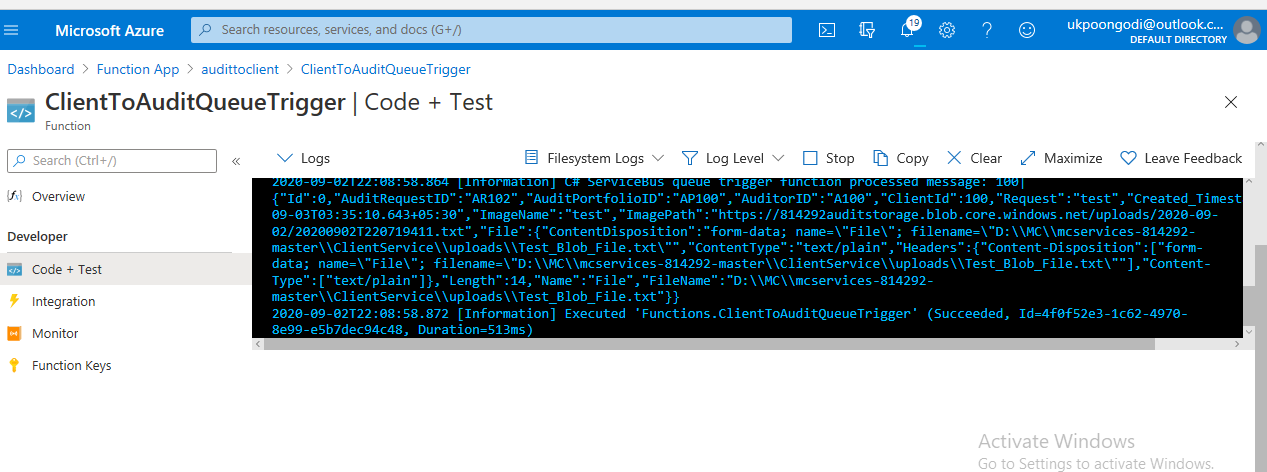
git push -u origin --all

git push -u origin –tags

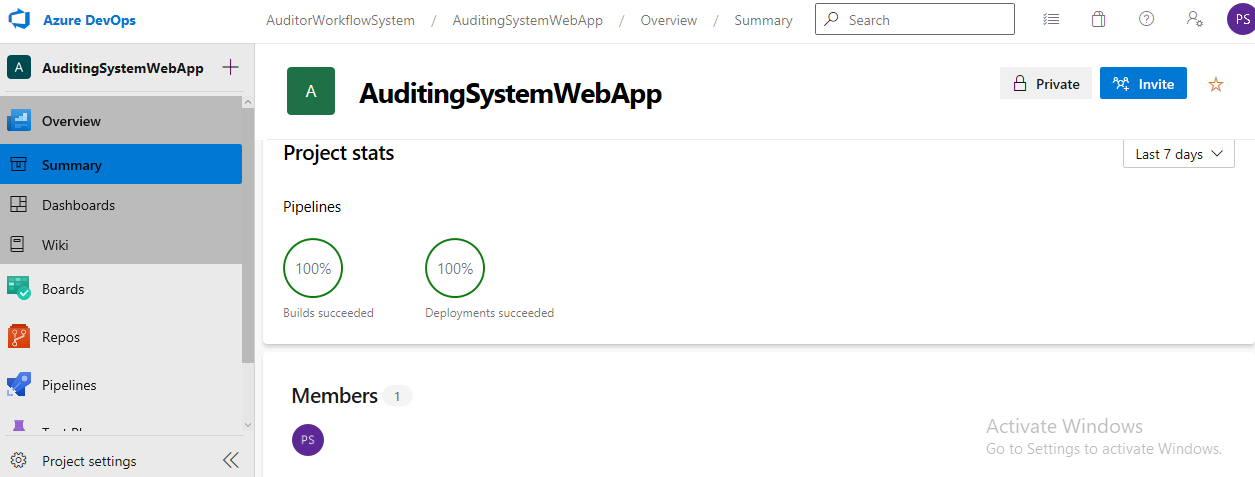
Azure web portal :

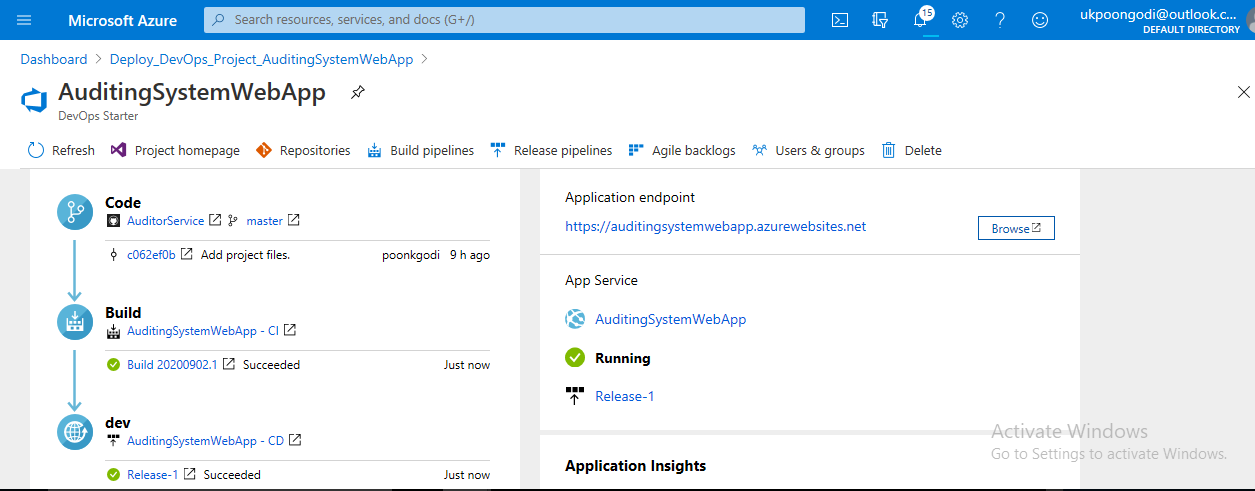


ClienttoAuditQueue through function App service bus Trigger :

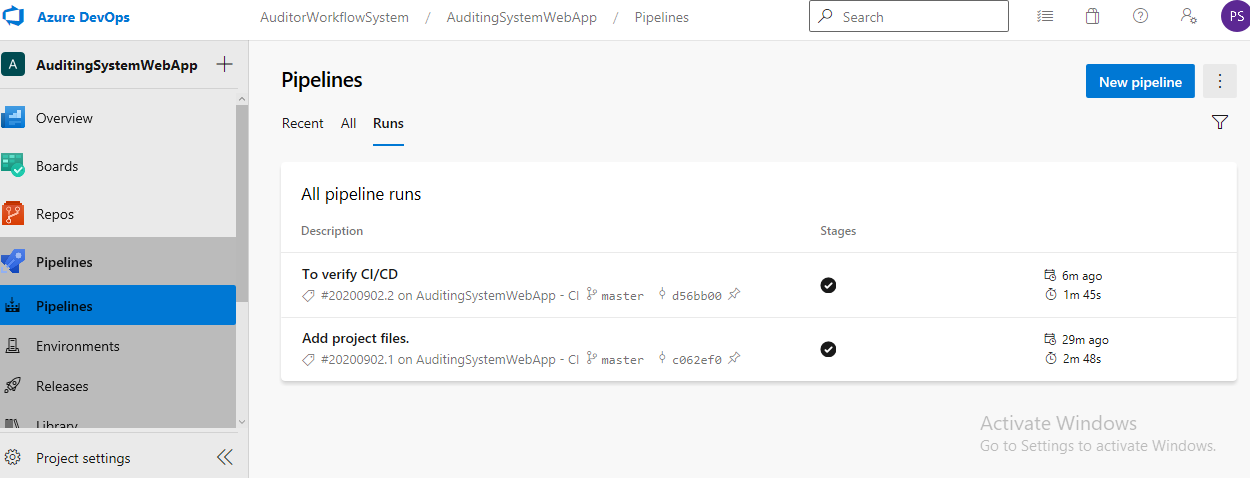


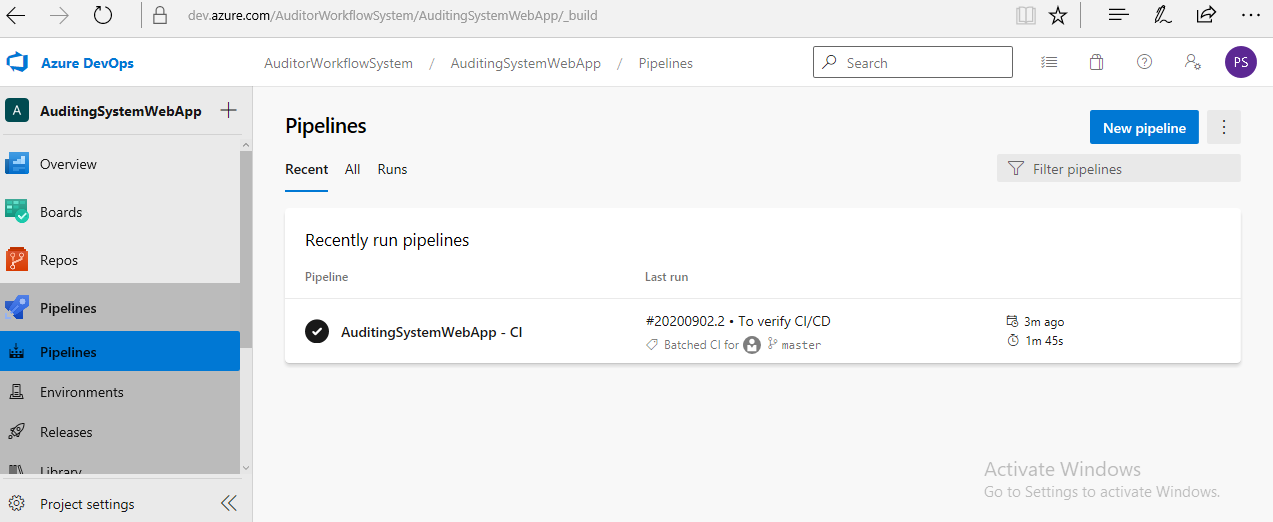
DevOps Starter :





CI/CD Pipeline :





CI/CD Release :

1. Created Resource Group

2. Created Service Bus Namespace

3. Created Queue - Audit and Client

4. Created Function App and mapped Queue to communicate the messages

5. Under Function App - Created 2 Function and mapped the Queue in the ServiceBusQueueTrigger

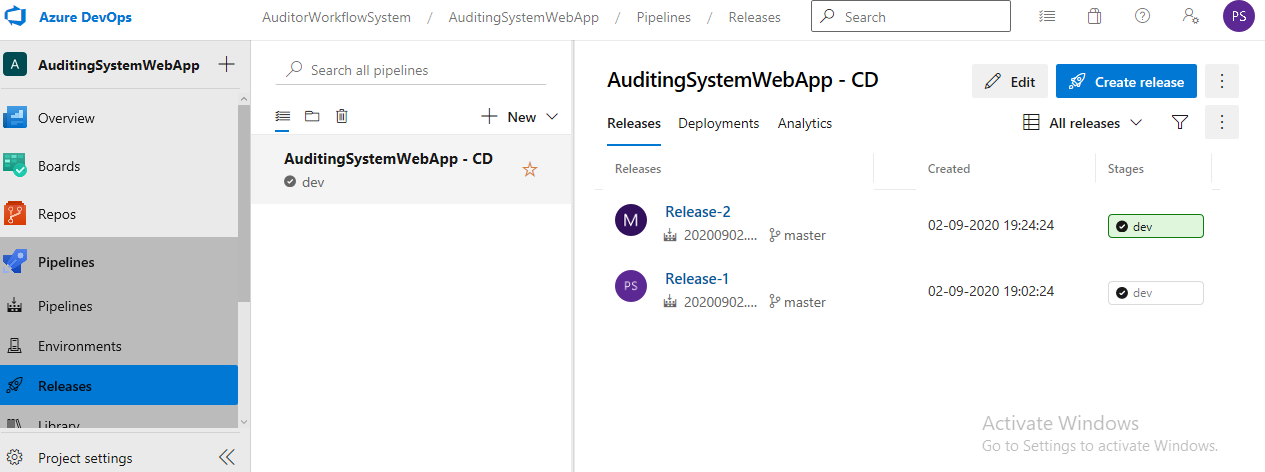
Function App Bridge between Web Apps(Deployed Code) between Client and Audit

6.Created Web APP to deploy the Application

7.Created SQL Databases in Azure - Audit and Client

8.Created Blob Storage to upload and download the files

9.Kept the code into GitHub Repository

10.created DevApps starter to Implemented CI/CD for the Build & Deployment

Implementation Steps :

1. Created Resource Group

2. Created Service Bus Namespace

3. Created Queue - Audit and Client

4. Created Function App and mapped Queue to communicate the messages

5. Under Function App - Created 2 Function and mapped the Queue in the ServiceBusQueueTrigger

Function App Bridge between Web Apps(Deployed Code) between Client and Audit

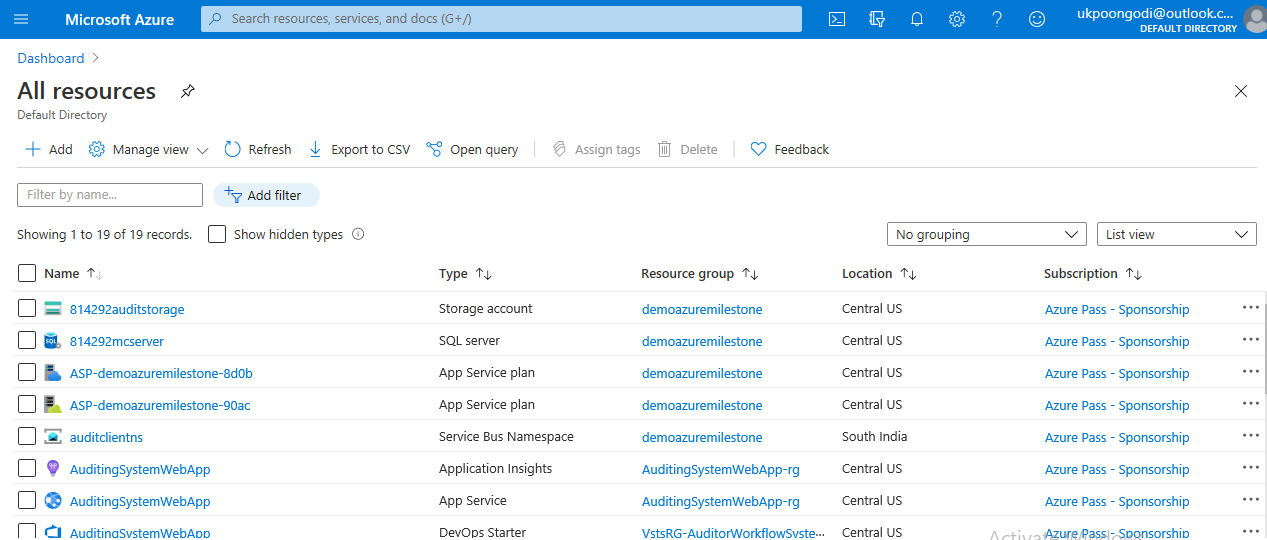
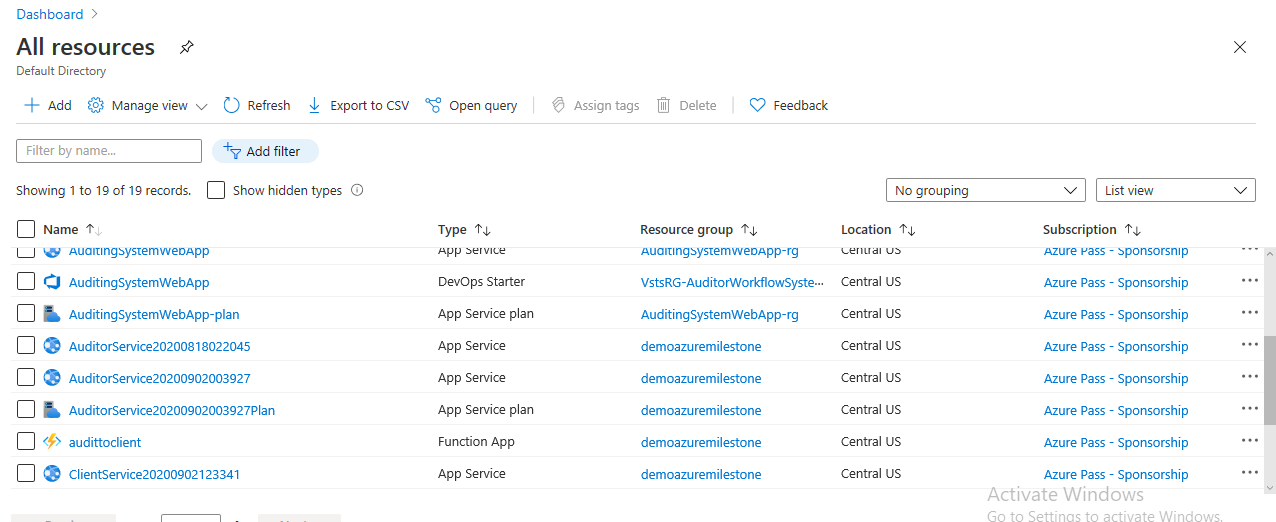
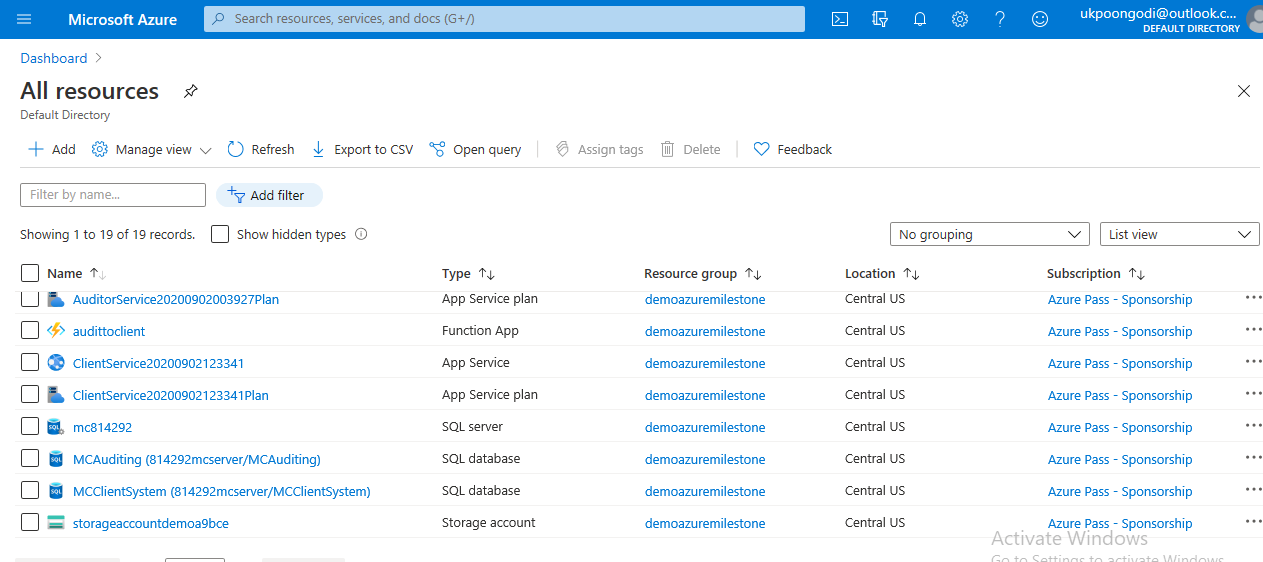
6.Created Web APP to deploy the Application

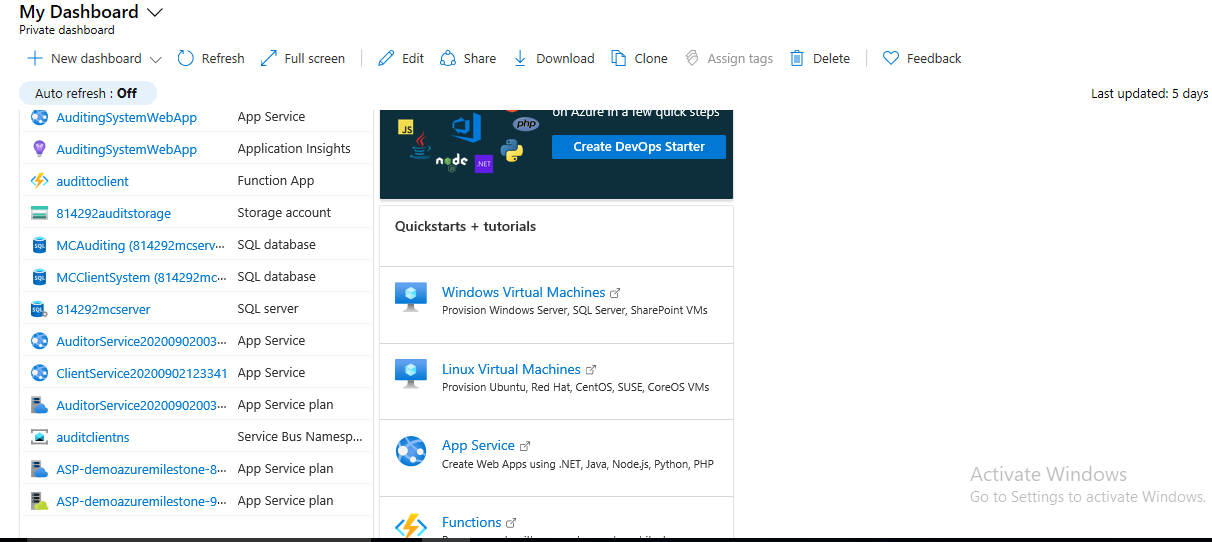
7.Created SQL Databases in Azure - Audit and Client

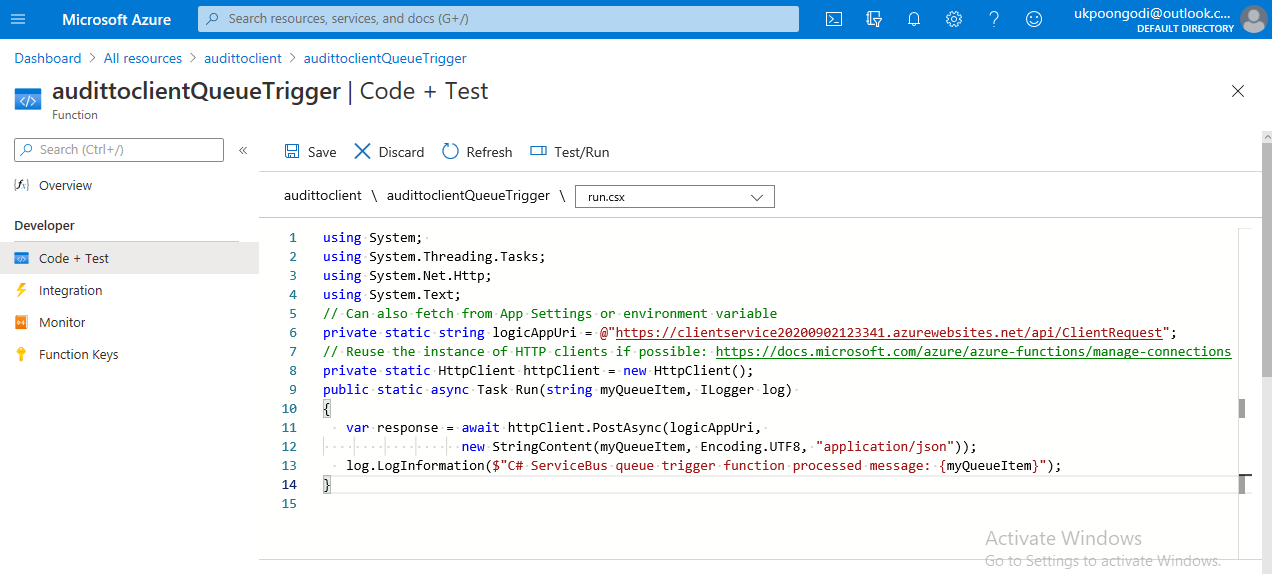
8.Created Blob Storage to upload and download the files

9.Kept the code into GitHub Repository

10.created DevApps starter to Implemented CI/CD for the Build & Deployment

  
Function App 🡪 functions 🡪 control + test



using System;

using System.Threading.Tasks;

using System.Net.Http;

using System.Text;

// Can also fetch from App Settings or environment variable

private static string logicAppUri = @"https://clientservice20200902123341.azurewebsites.net/api/ClientRequest";

// Reuse the instance of HTTP clients if possible: https://docs.microsoft.com/azure/azure-functions/manage-connections

private static HttpClient httpClient = new HttpClient();

public static async Task Run(string myQueueItem, ILogger log)

{

   var response = await httpClient.PostAsync(logicAppUri,

new StringContent(myQueueItem, Encoding.UTF8, "application/json"));

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

}

