

Azure Function App

Naman Moolri

CEQ-525

Introduction

Azure Functions is a serverless compute service that enables user to run event-triggered code without having to provision or manage infrastructure. Being as a trigger-based service, it runs a script or piece of code in response to a variety of events.

Azure Functions can be used to achieve decoupling, high throughput, reusability and shared. Being more reliable, it can also be used for the production environments.

The following features are included with Azure Functions:

- Choice of language - C#, Node.js, Python, F#, PHP, batch, bash, Java, or any executable.
- Pay-per-use pricing model.
- Bring your own
- Integrated security
- Code-less integration
- Flexible development

Azure Function Triggers

Triggers are what cause a function to run. A trigger defines how a function is invoked and a function must have exactly one trigger. Triggers have associated data, which is often provided as the payload of the function.

Types of Azure Function Triggers

- **Timer Trigger**
This trigger is called on a predefined schedule. We can set the time for execution of the Azure Function using this trigger.
- **HTTP Trigger**
This trigger gets fired when the HTTP request comes.
- **Queue Trigger**
This trigger gets fired when any new messages come in an Azure Storage Queue.
- **Blob Trigger**

This trigger will get fired when a new or updated blob is detected. The blob contents are passed on as input to the function.

- **Event Hub Trigger**

This trigger is used for the application instrumentation, the user experience, workflow processing, and the Internet of Things (IoT). This trigger will get fired when any events are delivered to an Azure Event Hub.

- **Generic Webhook**

This trigger gets fired when the Webhook HTTP requests come from any service that supports Webhooks.

- **GitHub Webhook**

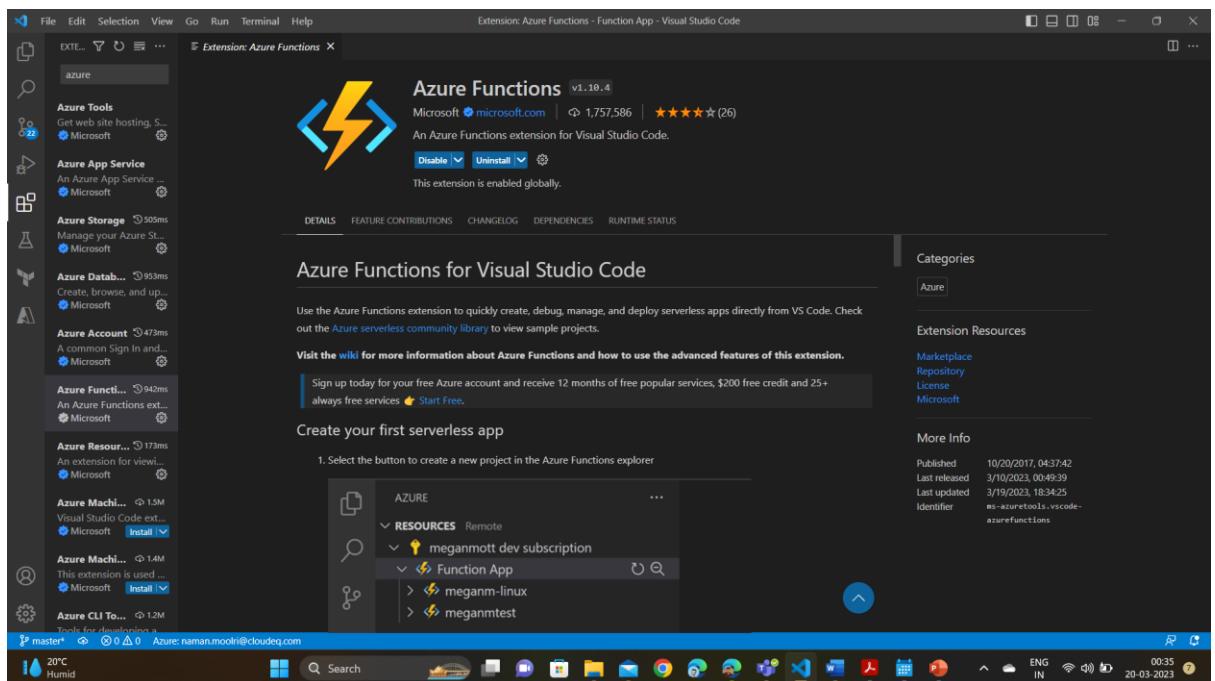
This trigger is fired when an event occurs in your GitHub repositories. The GitHub repository supports events such as Branch created, delete branch, issue comment, and Commit comment.

- **Service Bus Trigger**

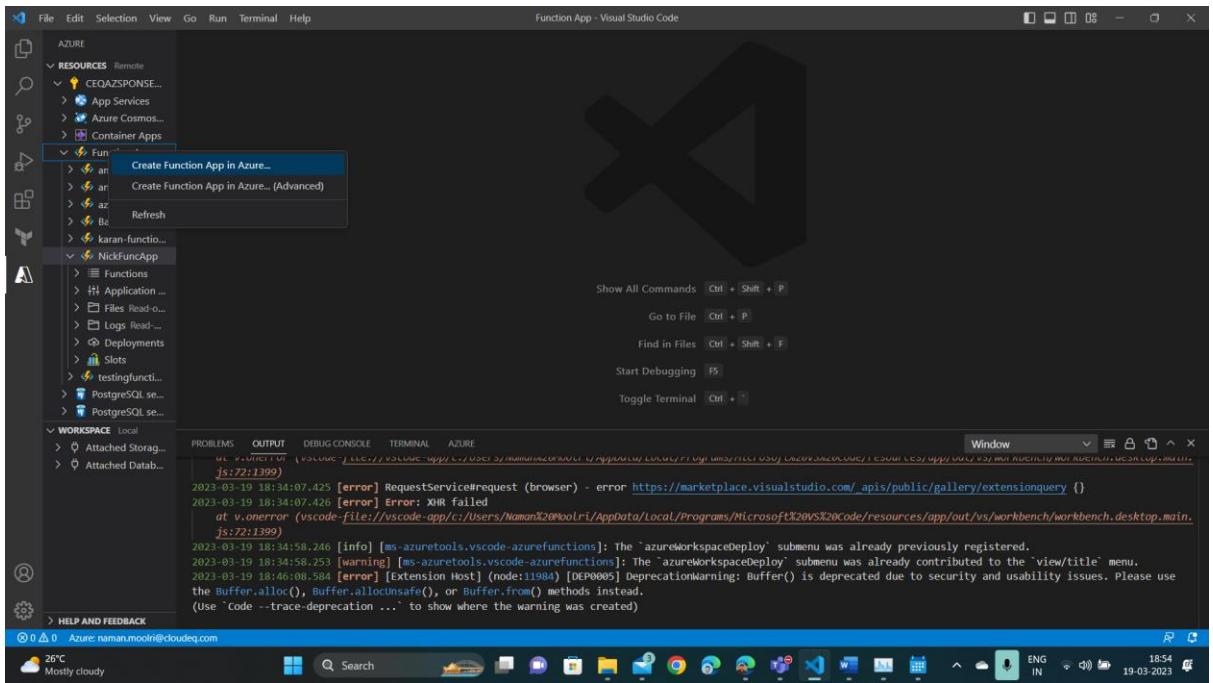
This trigger is fired when a new message comes from a service bus queue or topic.

Steps to implement HTTP Trigger through VSCode

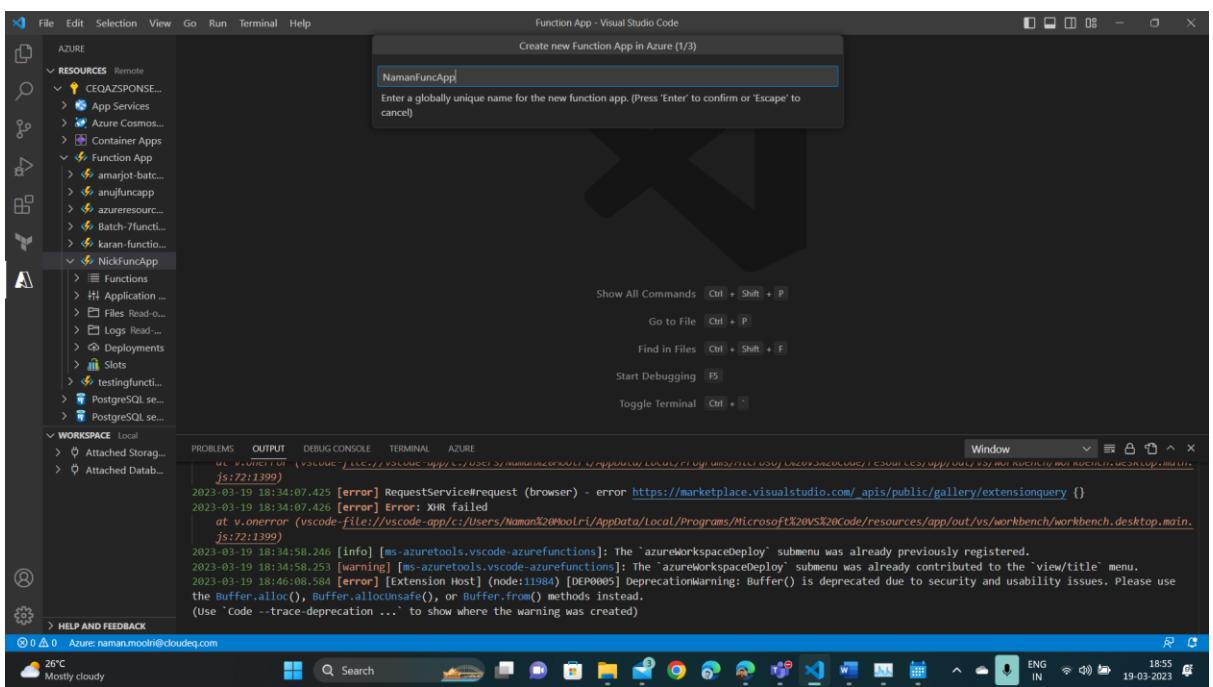
1. **Install Azure Functions and Azure Tools extension on VSCode. Now, connect it to your Azure Account by logging in through Azure Extension.**



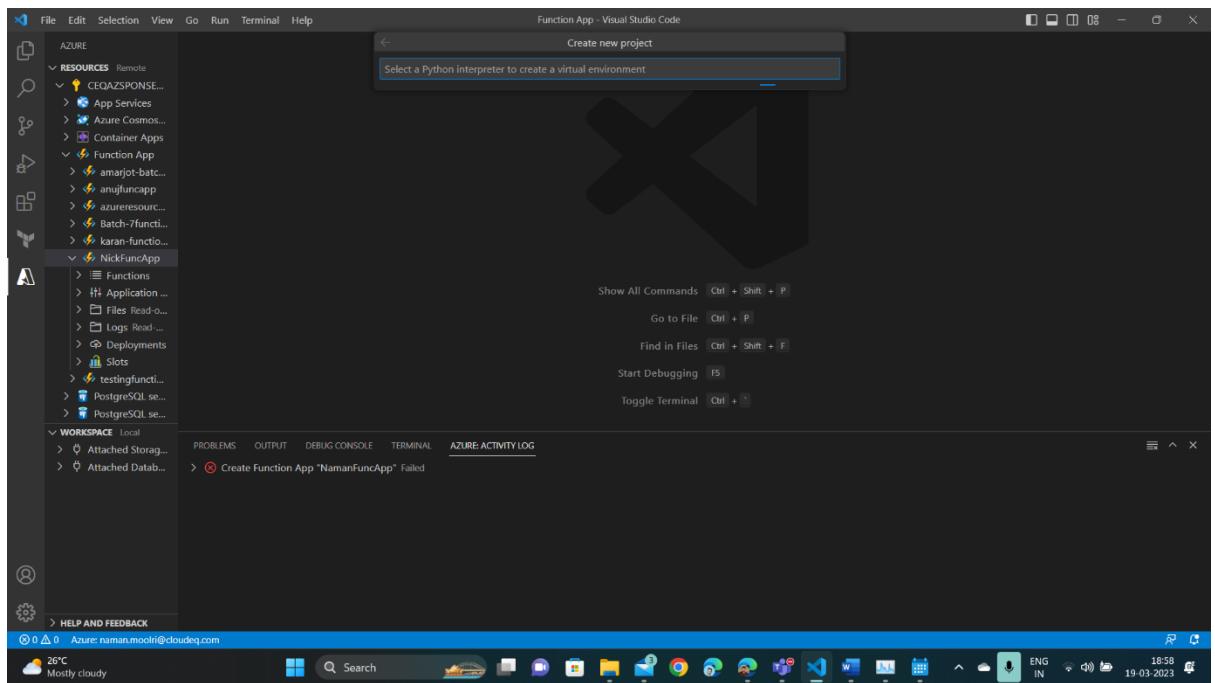
2. Creating an Azure Function Project.



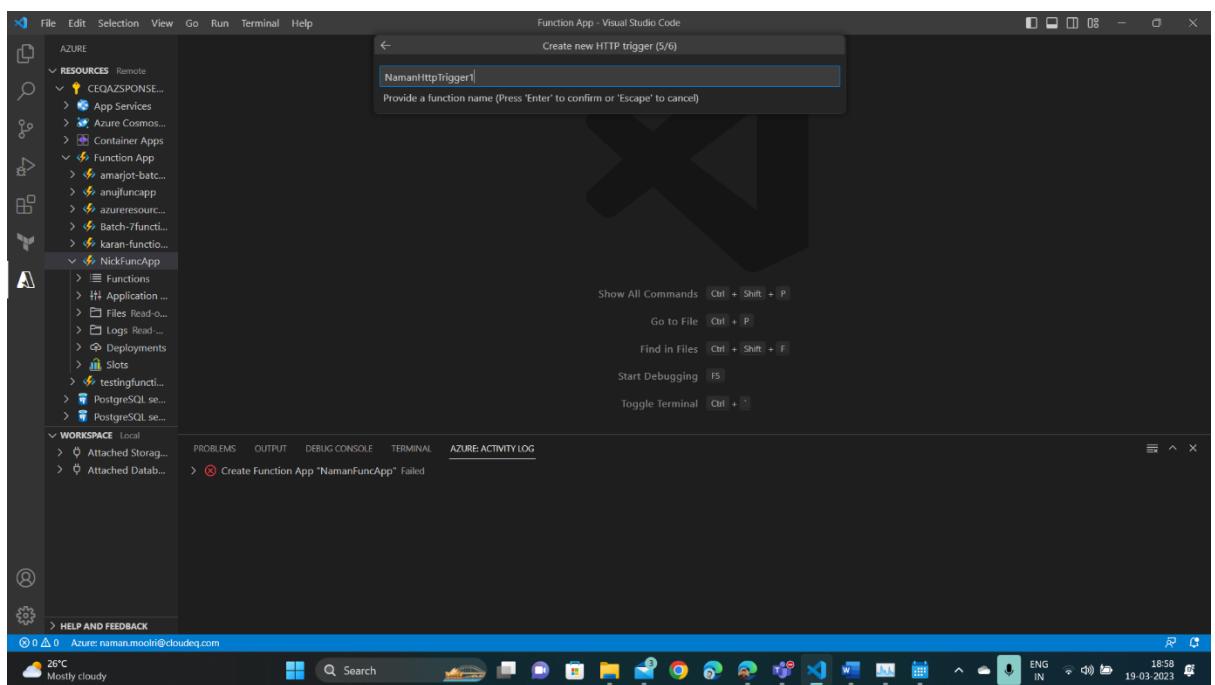
3. Naming our Function App.



4. Selecting the Python Environment.



5. Giving Name to our HTTP Trigger.



6. Our HTTP Trigger is created.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the Azure resources, including App Services, Container Apps, and Function App. The `NickFuncApp` function app is selected.
- Code Editor:** Displays the `function.json` file content:

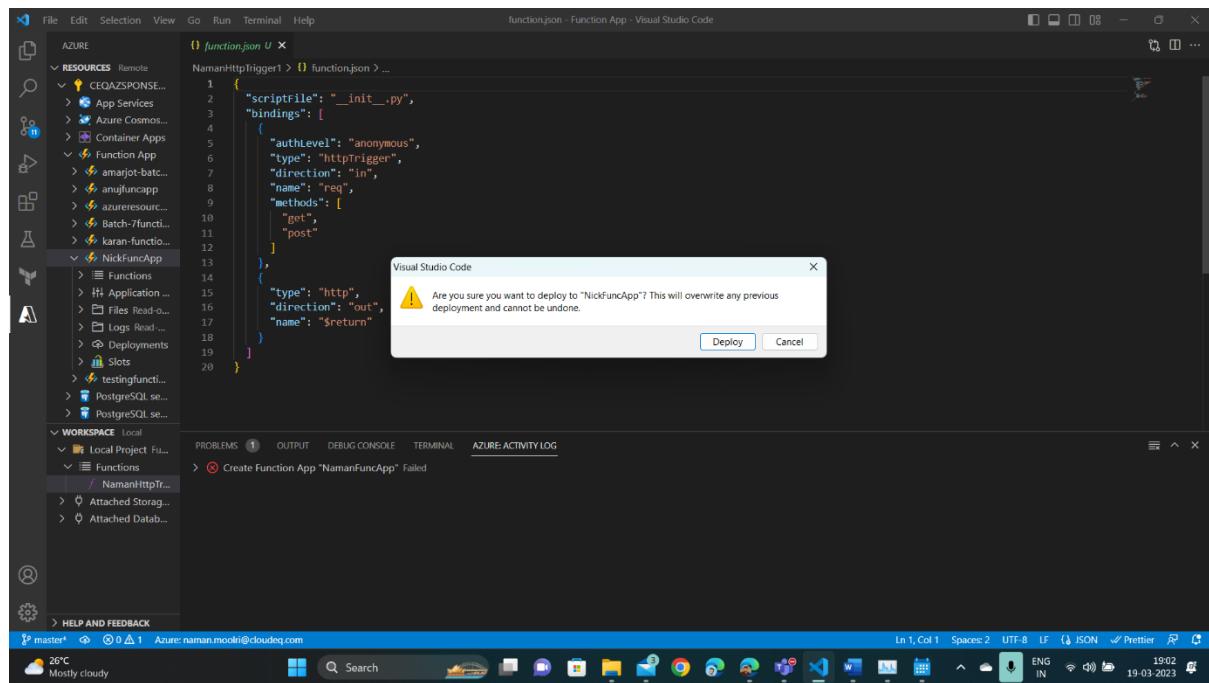
```
1  {  
2      "scriptFile": "__init__.py",  
3      "bindings": [  
4          {  
5              "authLevel": "anonymous",  
6              "type": "httptrigger",  
7              "direction": "in",  
8              "name": "req",  
9              "methods": [  
10                  "get",  
11                  "post"  
12              ]  
13          },  
14          {  
15              "type": "http",  
16              "direction": "out",  
17              "name": "$return"  
18          }  
19      ]  
20  }
```
- Activity Bar:** Shows the Help and Feedback icon.
- Bottom Status Bar:** Displays the current weather (26°C, Mostly cloudy), user information (Azure: naman.machhi@cloudiq.com), and system status (19-03-2023).

7. Deploying to our Function App

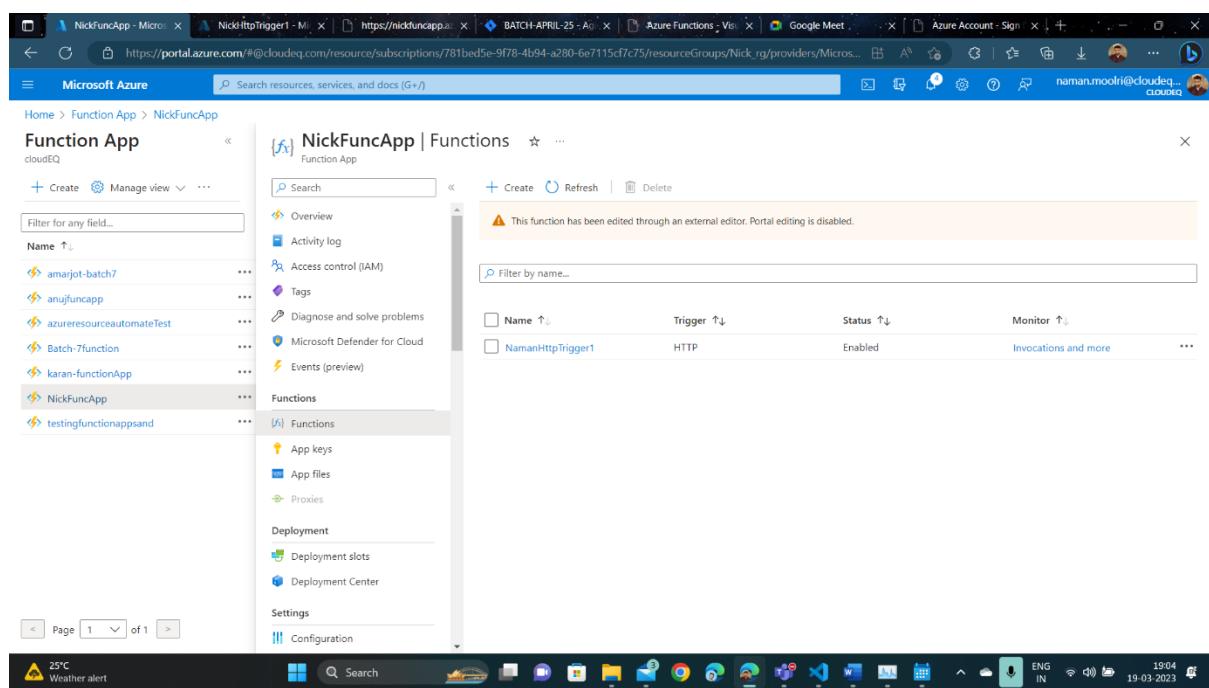
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the Azure resources, including App Services, Container Apps, and Function App. The `NickFuncApp` function app is selected.
- Code Editor:** Displays the `function.json` file content, identical to the previous screenshot.
- Activity Bar:** Shows the Help and Feedback icon.
- Bottom Status Bar:** Displays the current weather (26°C, Mostly cloudy), user information (Azure: naman.machhi@cloudiq.com), and system status (19-03-2023). A context menu is open over the function name, with the "Deploy to Function App..." option highlighted.

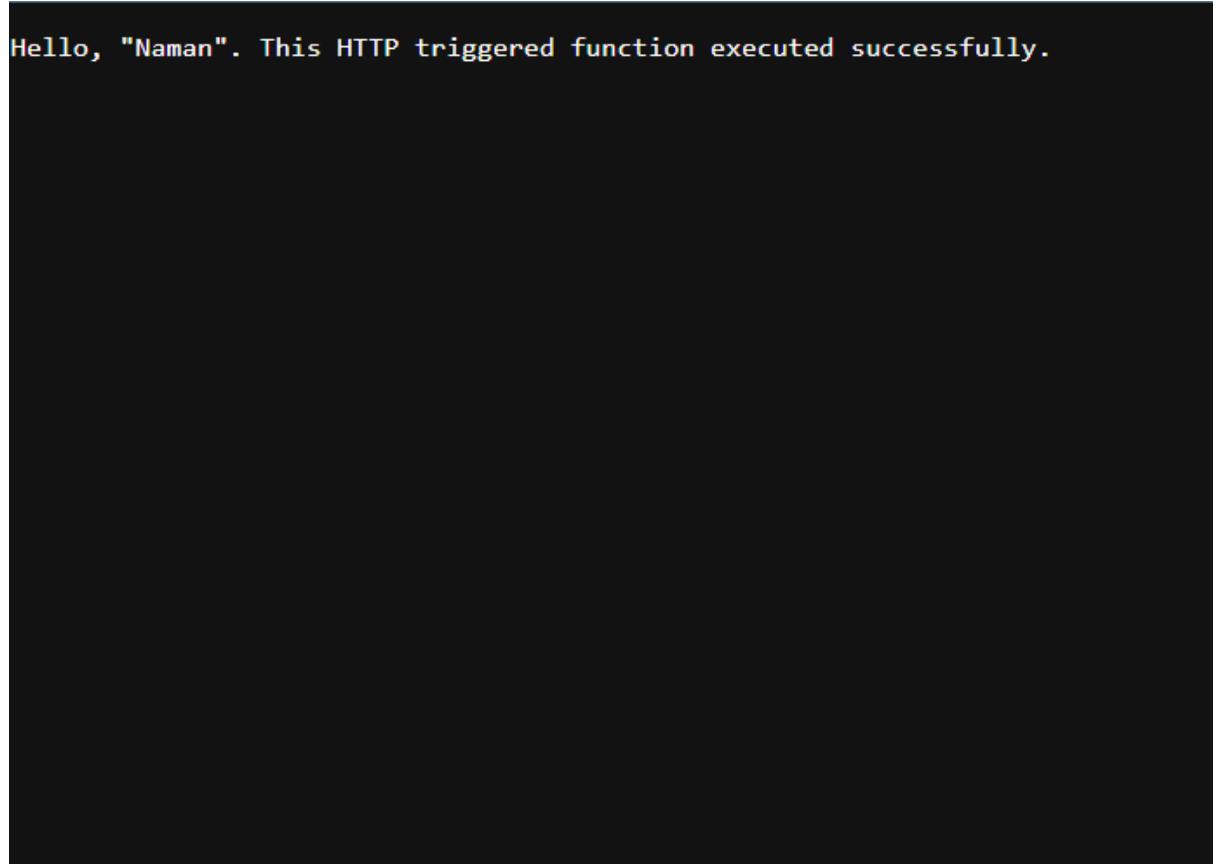
8. Confirming the deployment.



9. HTTP Trigger is deployed in our Function App.



10. Checking the Code, Test and Monitor column for connection.



11. Checking the Logs.

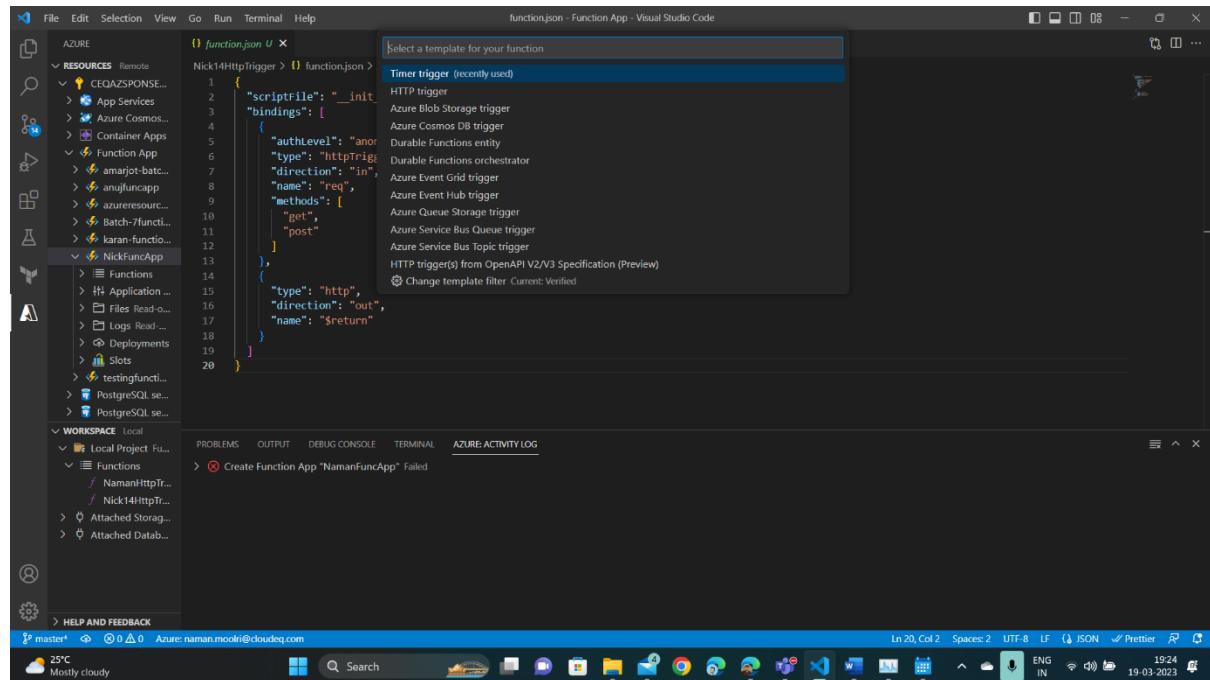
A screenshot of the Microsoft Azure portal showing the logs for the "Nick14HttpTrigger" function. The logs tab is selected. The log output shows several entries indicating successful connections and invocations of the function. Key log entries include:

```
Connected!
2023-03-19T13:51:14Z [Information] Executing 'Functions.Nick14HttpTrigger' (Reason='This function was programmatically called via the host APIs.', Id=84749d54-78b3-413e-9728-0923951481ab)
2023-03-19T13:51:14Z [Verbose] Sending invocation id:84749d54-78b3-413e-9728-0923951481ab on workerId:f15be63e-0126-4063-bb2d-ffb41fac79a8
2023-03-19T13:51:14Z [Information] Python HTTP trigger function processed a request.
2023-03-19T13:51:14Z [Information] Executed 'Functions.Nick14HttpTrigger' (Succeeded, Id=84749d54-78b3-413e-9728-0923951481ab, Duration=50ms)
2023-03-19T13:52:05Z [Information] Executing 'Functions.Nick14HttpTrigger' (Reason='This function was programmatically called via the host APIs.', Id=dfdf8d89-89f3-4405-89bd-e20685db11f7)
2023-03-19T13:52:05Z [Verbose] Sending invocation id:dfdf8d89-89f3-4405-89bd-e20685db11f7 on workerId:f15be63e-0126-4063-bb2d-ffb41fac79a8
2023-03-19T13:52:05Z [Information] Python HTTP trigger function processed a request.
2023-03-19T13:52:05Z [Information] Executed 'Functions.Nick14HttpTrigger' (Succeeded, Id=dfdf8d89-89f3-4405-89bd-e20685db11f7, Duration=12ms)
```

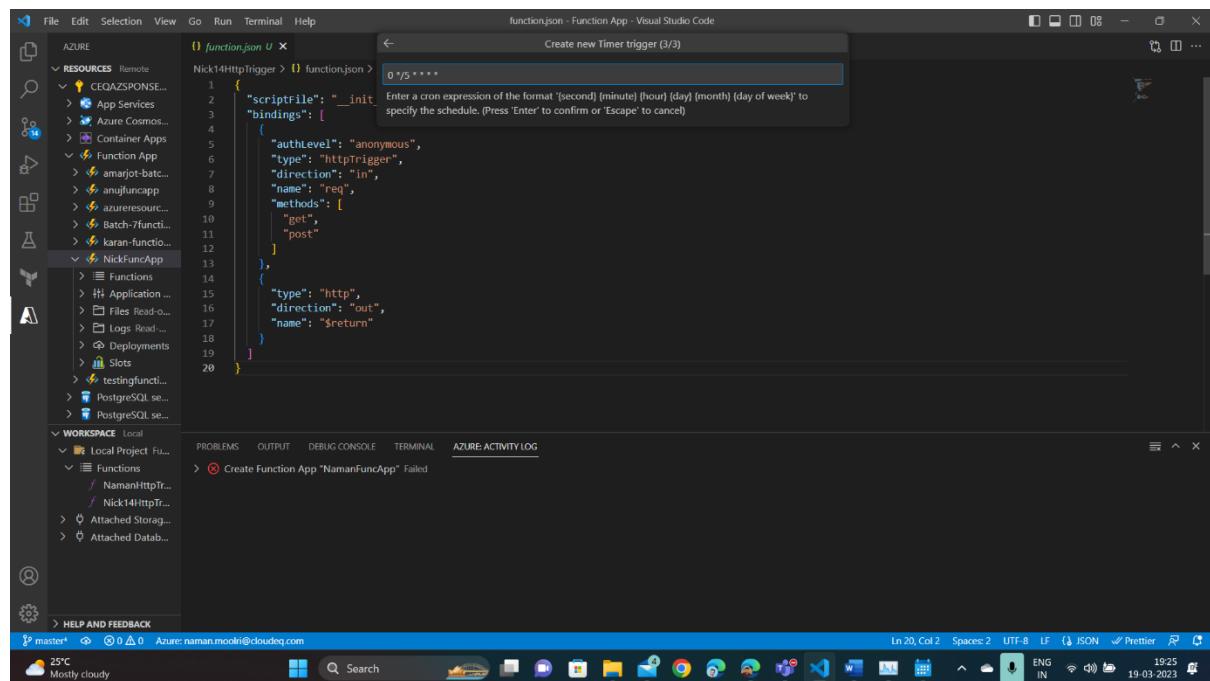
The portal interface includes a navigation bar at the top, a search bar, and various developer tools like Code + Test and Monitor on the left.

Steps to implement Timer Trigger through VSCode

1. Creating a Timer Trigger.



2. Selecting the timeout time by entering the cron expression.



3. Timer Trigger is created.

The screenshot shows the Visual Studio Code interface with the Azure extension loaded. The left sidebar displays the Azure resource tree, including App Services, Container Apps, Function App, and a local project named 'NickFuncApp'. The 'FUNCTIONS' section under 'NickFuncApp' lists three functions: 'NamanHttpTrigger', 'NamanTimer...', and 'Nick14HttpTr...'. The main editor area shows the 'function.json' file for the 'NamanTimer...' function, which contains the following JSON code:

```
1 {  
2     "scriptFile": "__init__.py",  
3     "bindings": [  
4         {  
5             "name": "mytimer",  
6             "type": "timerTrigger",  
7             "direction": "in",  
8             "schedule": "0 */5 * * *" // This line is highlighted in yellow  
9         }  
10    ]  
11 }
```

The status bar at the bottom indicates the file is 1 line long, 1926 characters wide, and was last modified on 19-03-2023 at 19:26. The system tray shows it's 25°C and mostly cloudy.

4. Now deploying our Timer Trigger to our Function App.

The screenshot shows the Visual Studio Code interface with the Azure extension loaded. The left sidebar displays the Azure resource tree, including App Services, Container Apps, Function App, and a local project named 'NickFuncApp'. The 'FUNCTIONS' section under 'NickFuncApp' lists three functions: 'NamanHttpTrigger', 'NamanTimer...', and 'Nick14HttpTr...'. The main editor area shows the 'function.json' file for the 'NamanTimer...' function, which contains the same JSON code as the previous screenshot. A tooltip 'Deploy to Function App...' is visible over the 'NamanTimer...' function entry in the tree view. The status bar at the bottom indicates the file is 1 line long, 1926 characters wide, and was last modified on 19-03-2023 at 19:26. The system tray shows it's 25°C and mostly cloudy.

5. Timer Trigger is deployed in our Function App.

NickFuncApp | Functions

Name	Trigger	Status	Monitor
NamanHttpTrigger1	HTTP	Disabled	Invocations and more
NamanTimerTrigger	Timer	Enabled	Invocations and more
Nick14HttpTrigger	HTTP	Enabled	Invocations and more

6. Checking our Code, Test and Monitor for connection.

NamanTimerTrigger | Monitor

Date (UTC)	Success	Result Code	Duration (ms)	Operation Id
2023-03-19 14:10:00.003	Success	0	44	c74bb9e6de955c137e5e10d17ab8a3c
2023-03-19 14:05:00.002	Success	0	42	23d36e5b1169e645c96660b927e511d

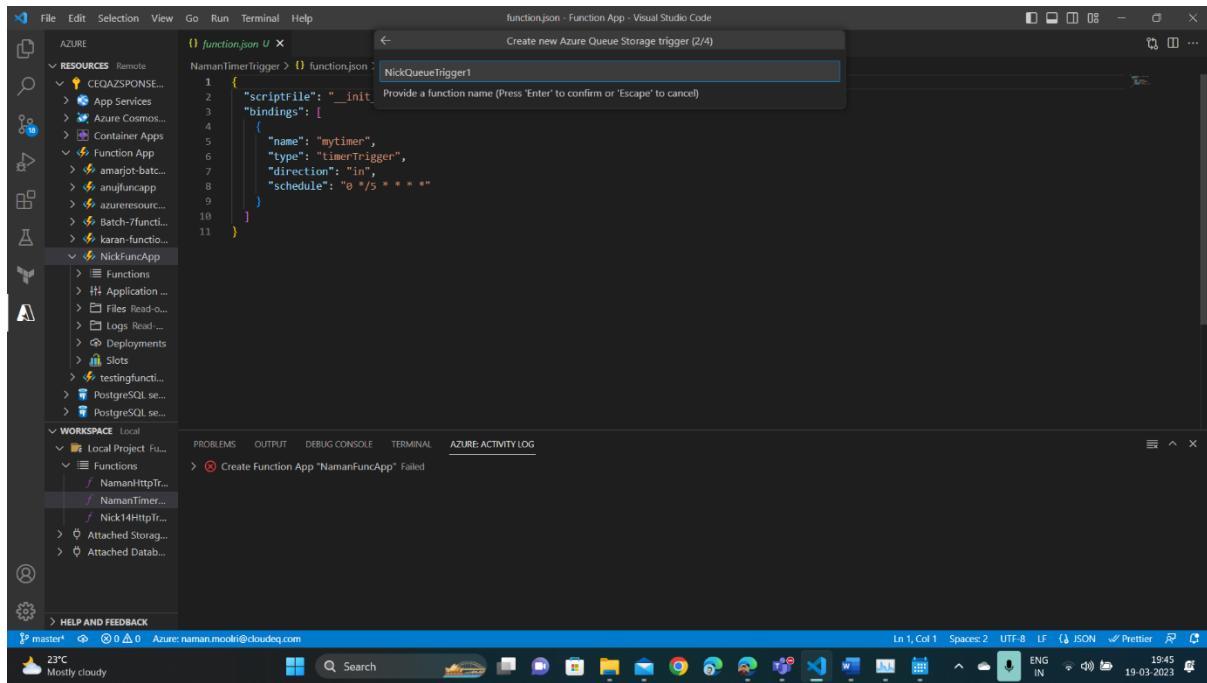
7. Checking Logs.

The screenshot shows the Microsoft Azure portal interface with the URL <https://portal.azure.com/#view/WebsitesExtension/FunctionMenuBlade/~/monitor/resourceId/%2Fsubscriptions%2F781bed5e-9f78-4b94-a280-6e711...>. The main content area displays the logs for the 'NamanTimerTrigger' function. The logs are listed under the 'Logs' tab, with the 'Information' level selected. The logs show the following entries:

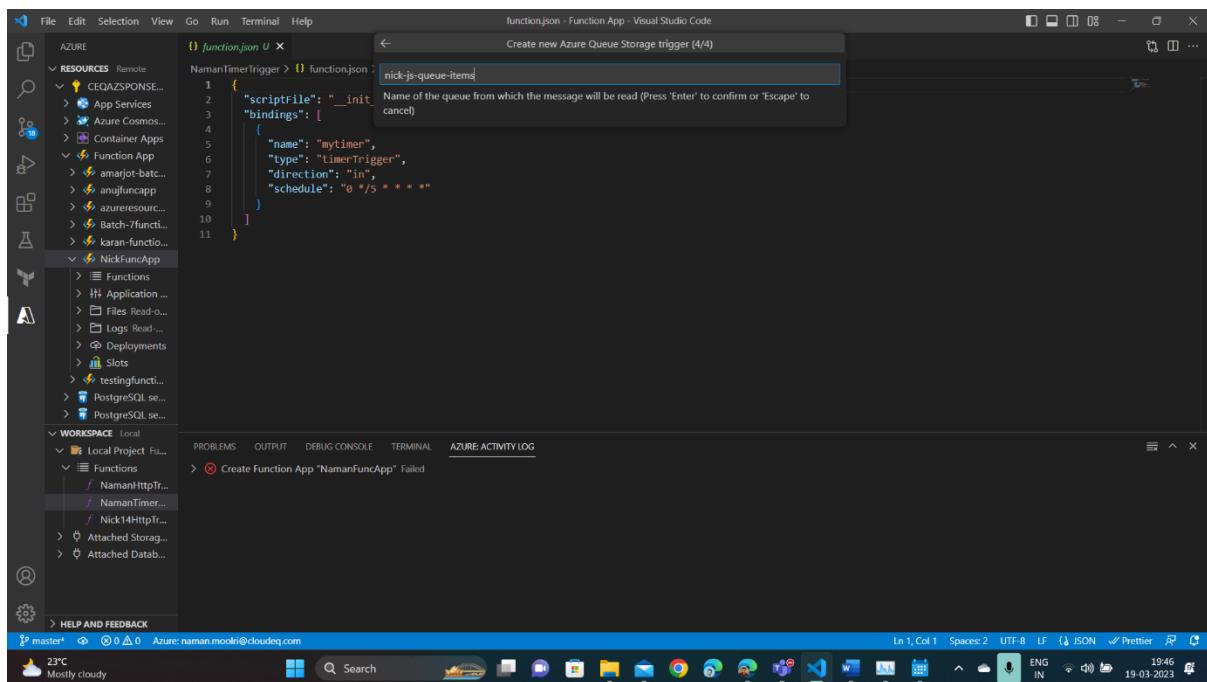
```
Connected!
2023-03-19T14:05:00Z [Information] Executing 'Functions.NamanTimerTrigger' (Reason='Timer fired at 2023-03-19T14:05:00+00:00', Id=9539ac4b-cf06-47e9-b57b-58009af79bf)
2023-03-19T14:05:00Z [Verbose] Sending invocation id:9539ac4b-cf06-47e9-b57b-58009af79bf on workerId:acf6122-7796-47e1-8dbd-f7788f261ea2
2023-03-19T14:05:00Z [Verbose] Posting invocation id:9539ac4b-cf06-47e9-b57b-58009af79bf on workerId:acf6122-7796-47e1-8dbd-f7788f261ea2
2023-03-19T14:05:00Z [Information] Python timer trigger function ran at 2023-03-19T14:05:00.034469+00:00
2023-03-19T14:05:00Z [Information] Executed 'Functions.NamanTimerTrigger' (Succeeded, Id=9539ac4b-cf06-47e9-b57b-58009af79bf, Duration=40ms)
2023-03-19T14:10:00Z [Information] Executing 'Functions.NamanTimerTrigger' (Reason='Timer fired at 2023-03-19T14:10:00+00:00', Id=6da0fe1e-1e4a-4ce0-9a28-955f27d2376d)
2023-03-19T14:10:00Z [Verbose] Sending invocation id:6da0fe1e-1e4a-4ce0-9a28-955f27d2376d on workerId:00f1430e-9301-4aea-a246-0eac035bdead
2023-03-19T14:10:00Z [Verbose] Posting invocation id:6da0fe1e-1e4a-4ce0-9a28-955f27d2376d on workerId:00f1430e-9301-4aea-a246-0eac035bdead
2023-03-19T14:10:00Z [Information] Python timer trigger function ran at 2023-03-19T14:10:00.031589+00:00
2023-03-19T14:10:00Z [Information] Executed 'Functions.NamanTimerTrigger' (Succeeded, Id=6da0fe1e-1e4a-4ce0-9a28-955f27d2376d, Duration=42ms)
```

Steps to implement Queue Trigger through VSCode

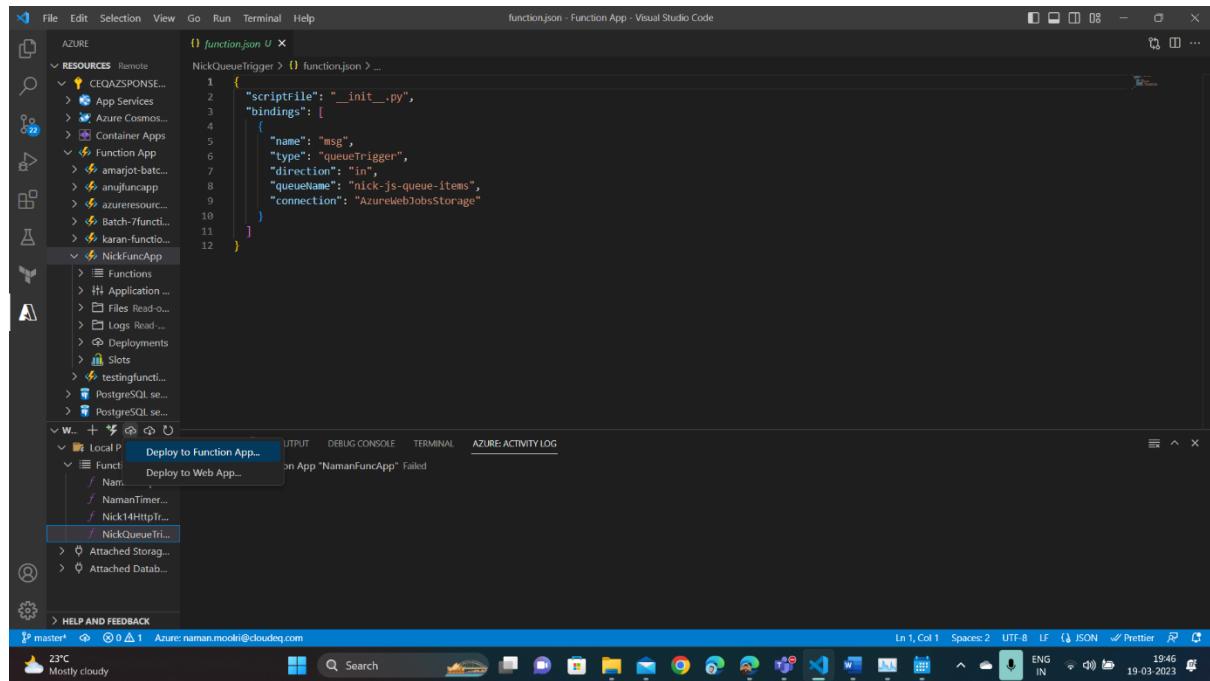
1. Creating a Queue Storage Trigger.



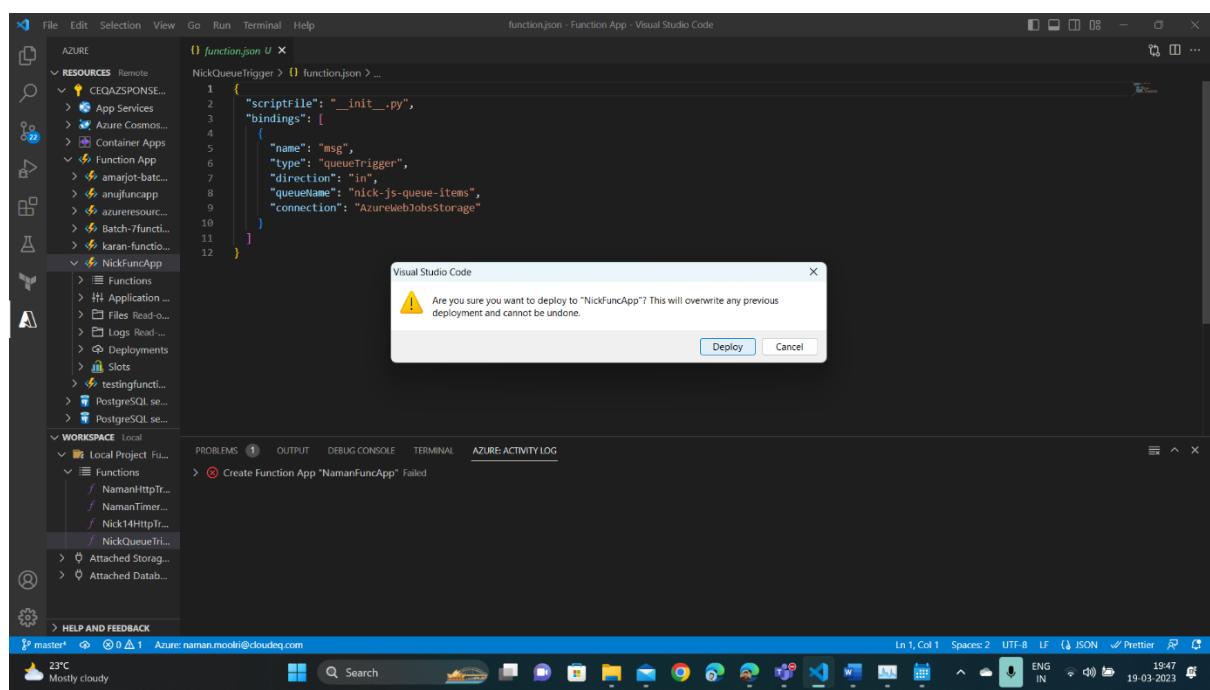
2. Naming the queue from which the message will be read.



3. Deploying Queue Trigger to our Function App.



4. Confirming the deployment.



5. Trigger is deployed in our Function App.

Name	Trigger	Status	Monitor
NamanHttpTrigger1	HTTP	Disabled	Invocations and more
NamanTimerTrigger	Timer	Enabled	Invocations and more
Nick14HttpTrigger	HTTP	Enabled	Invocations and more
NickQueueTrigger	Queue	Enabled	Invocations and more

6. Now, go to our resource group. Then, click the storage resource and create a new storage queue with the name that we provided to our Queue in VSCode for reading messages.

Queue	Url
nick-js-queue-items	https://nickrg898b.queue.core.windows.net/nick-js-queue-items

7. Now, add message to the queue.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is https://portal.azure.com/#view/Microsoft_Azure_Storage/QueueMenuBlade/~/overview/storageAccountId/%2Fsubscriptions%2F781bed5e-9f78-4b94-a280-6e711.... The page title is "nick-js-queue-items". The main content area is titled "Add message to queue". The "Message text" field contains the text "Hi Naman, Welcome to Azure Function App". Below the text input, there is a "Expires in:" dropdown set to "7 Days". There are two checkboxes: "Message never expires" (unchecked) and "Encode the message body in Base64" (checked). At the bottom are "OK" and "Cancel" buttons.



8. Check the Code, Test and Monitor for connection in Functions column.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is <https://portal.azure.com/#view/WebsitesExtension/FunctionMenuBlade/~/monitor/resourceId/%2Fsubscriptions%2F781bed5e-9f78-4b94-a280-6e711...>. The page title is "NickQueueTrigger | Monitor". The left sidebar has "Monitor" selected. The main content area shows "Invocations" tab selected. It displays "Success Count" (1) and "Error Count" (0) over the last 30 days. Below this is the "Invocation Traces" section, which lists a single successful invocation from 2023-03-19 at 14:42:03.144 with an operation ID of f021bc89ef8801204e5494c90bb9feac.



9. Checking the Logs.

