**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**

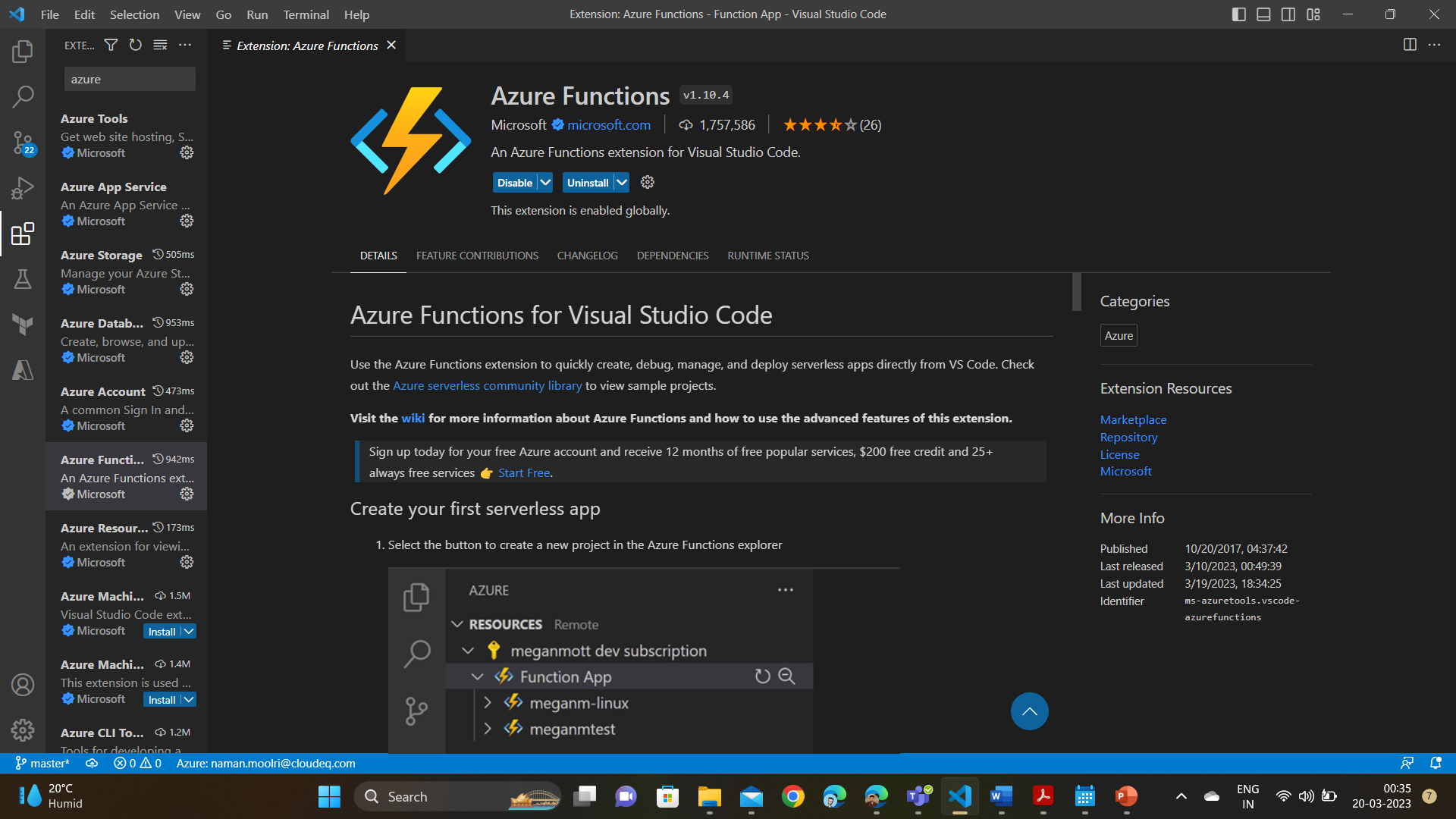
This trigger is fired when an event occurs in your GitHub repositories. The GitHub repository supports events such as Branch created, delete branch, issue comment, and Commit comment.

* **Service Bus Trigger**

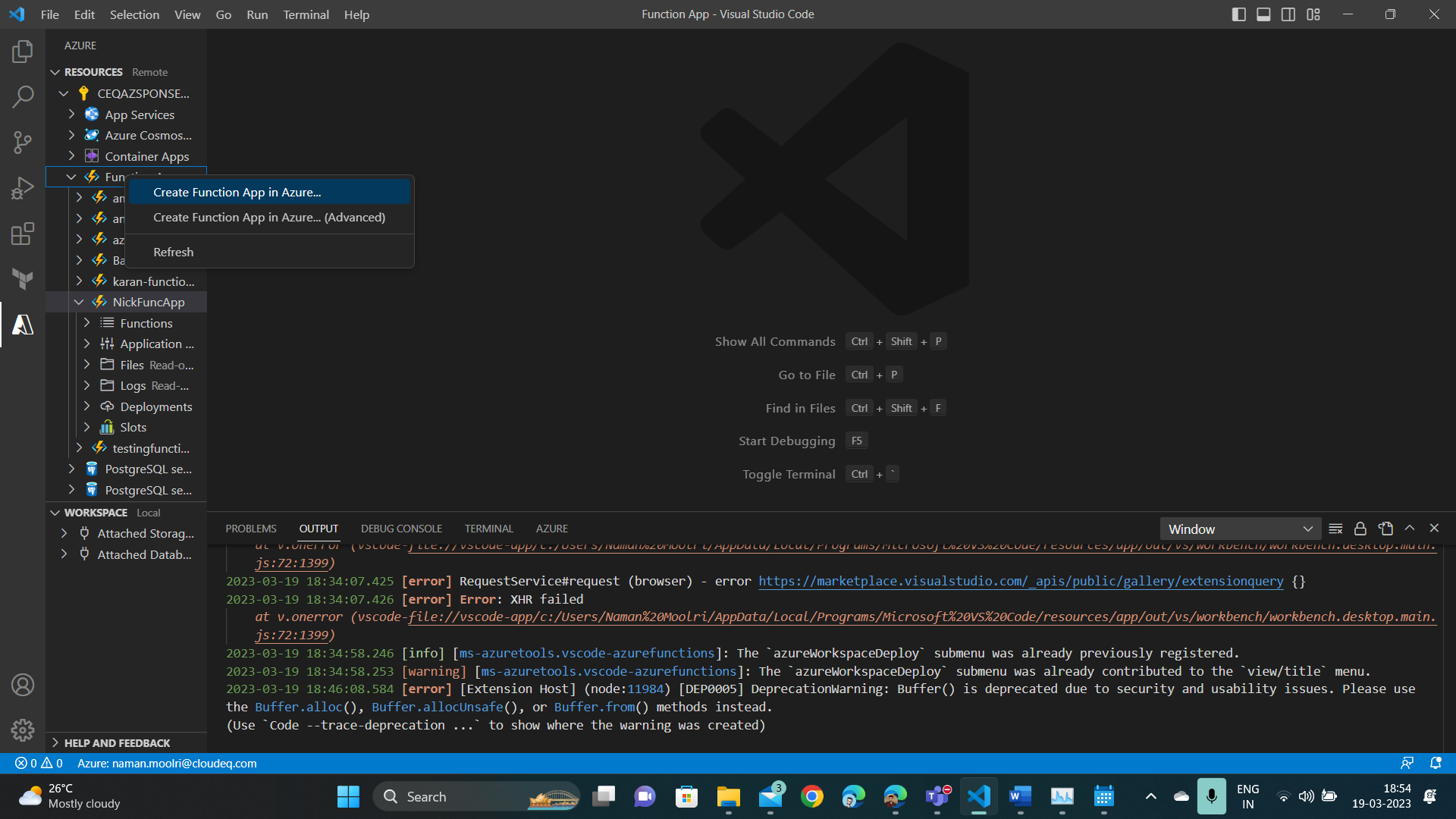
This trigger is fired when a new message comes from a service bus queue or topic.

**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.**



1. **Creating an Azure Function Project.**

****

1. **Naming our Function App.**

**Graphical user interface

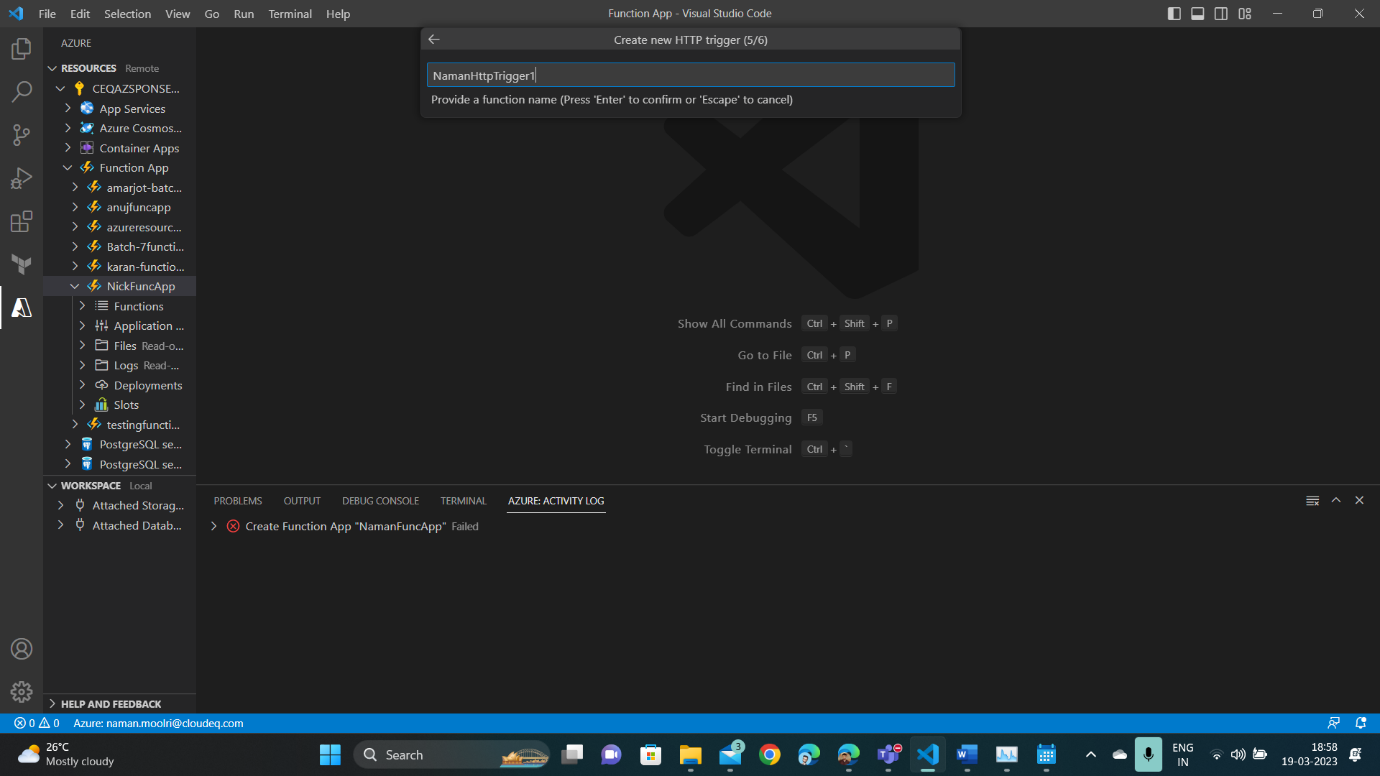
Description automatically generated**

1. **Selecting the Python Environment.**

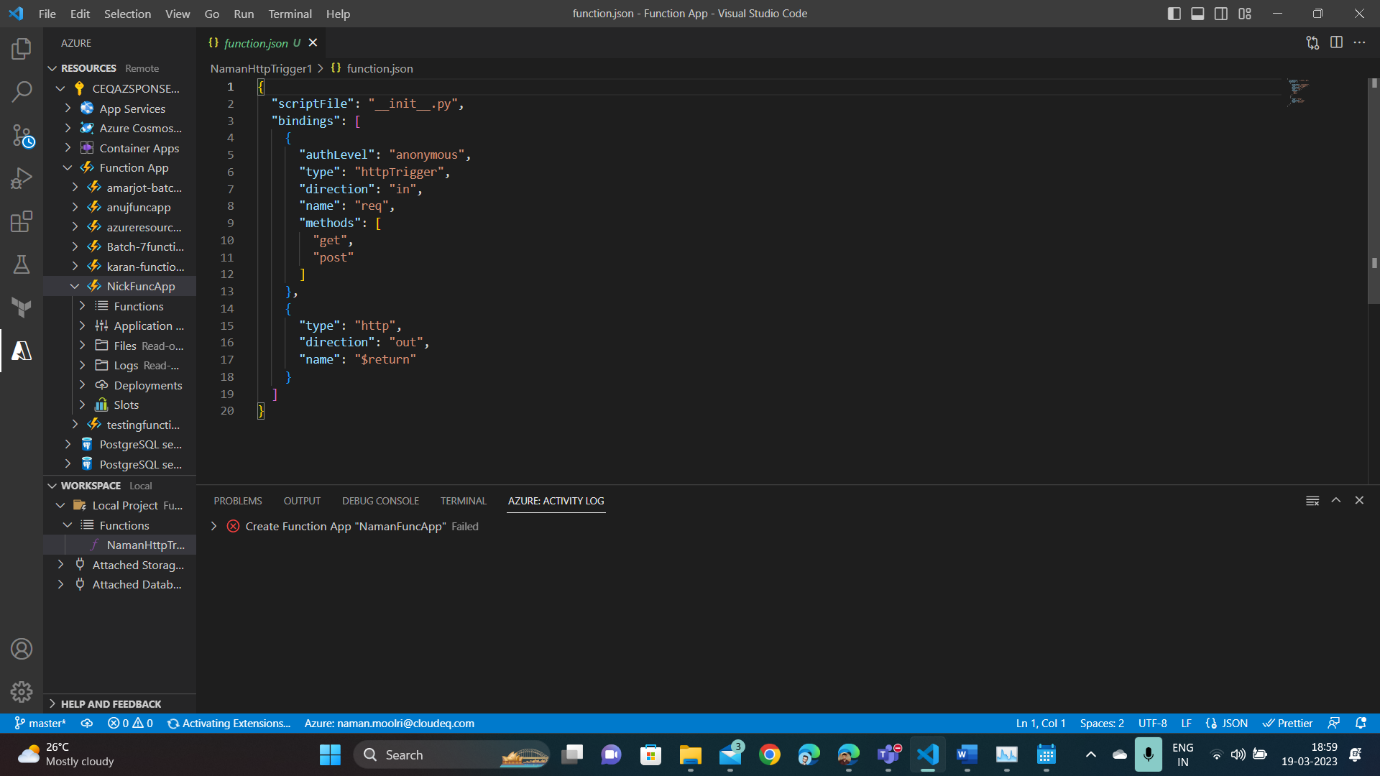
A screenshot of a computer

Description automatically generated

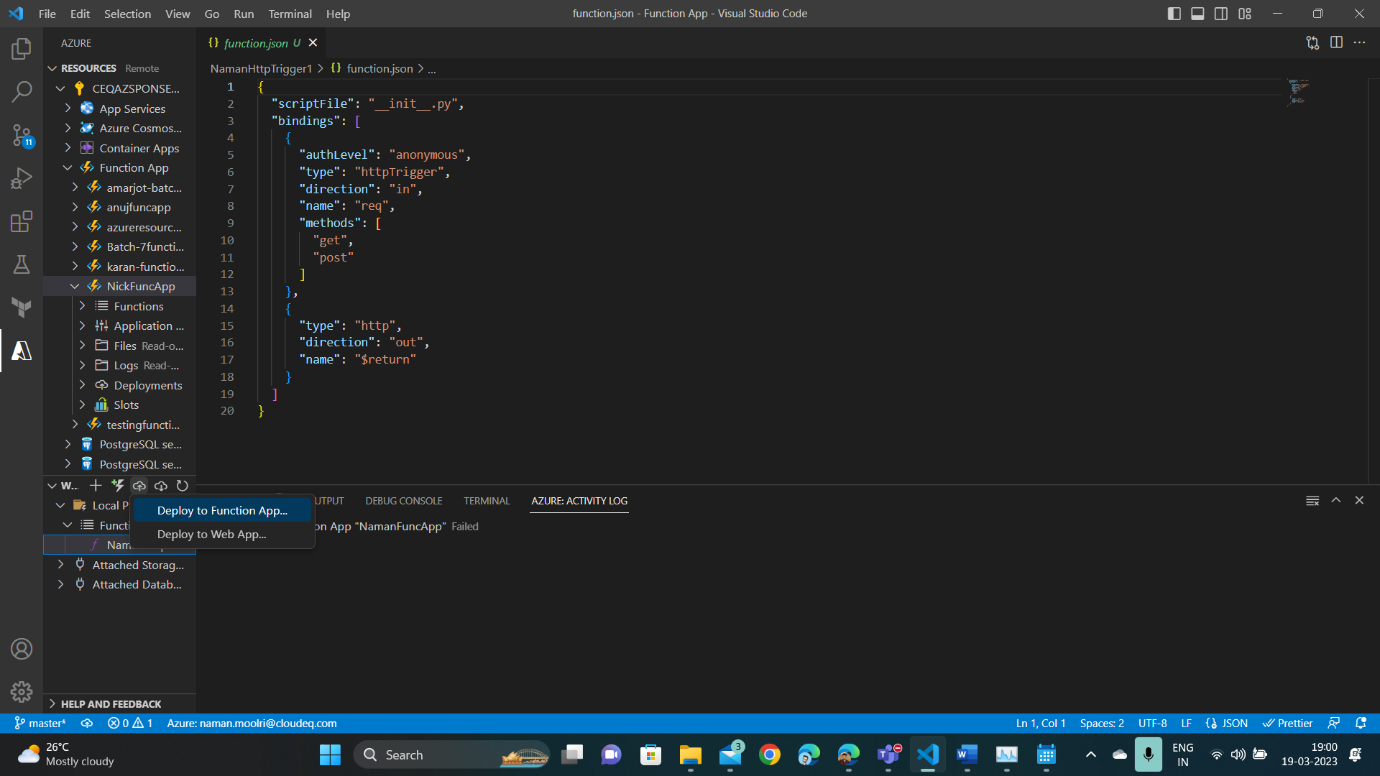
1. **Giving Name to our HTTP Trigger.**



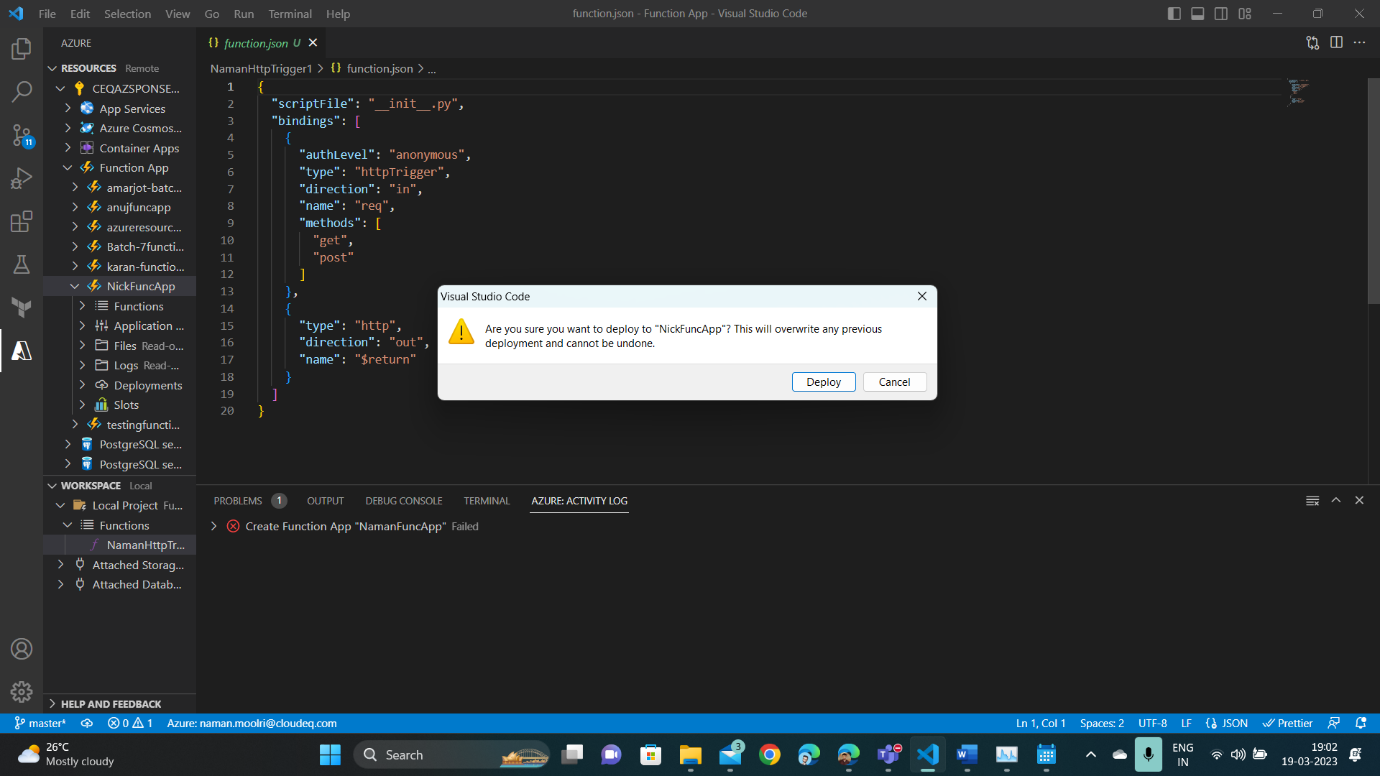
1. **Our HTTP Trigger is created.**

****

1. **Deploying to our Function App**

****

1. **Confirming the deployment.**

****

1. **HTTP Trigger is deployed in our Function App.**

**A screenshot of a computer

Description automatically generated**

1. **Checking the Code, Test and Monitor column for connection.**

**Text

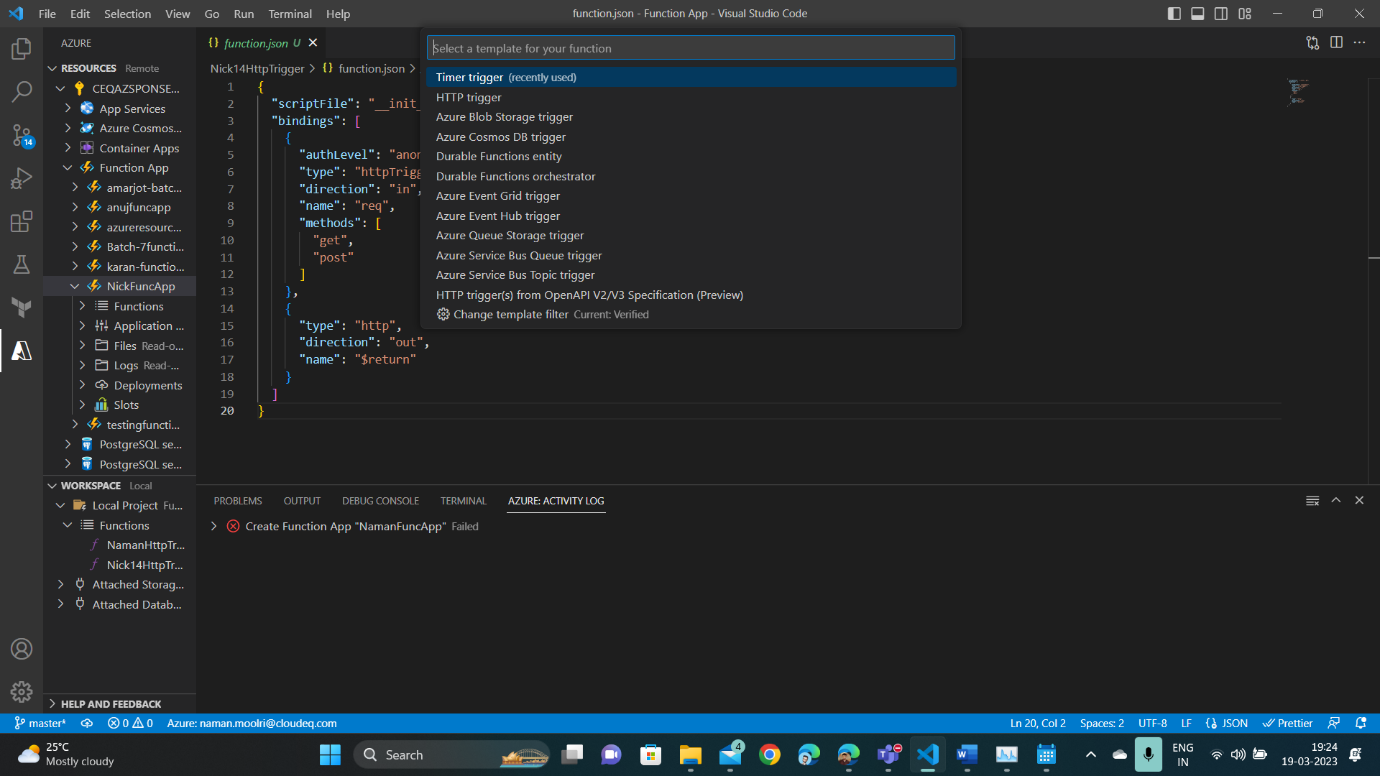
Description automatically generated**

1. **Checking the Logs.**

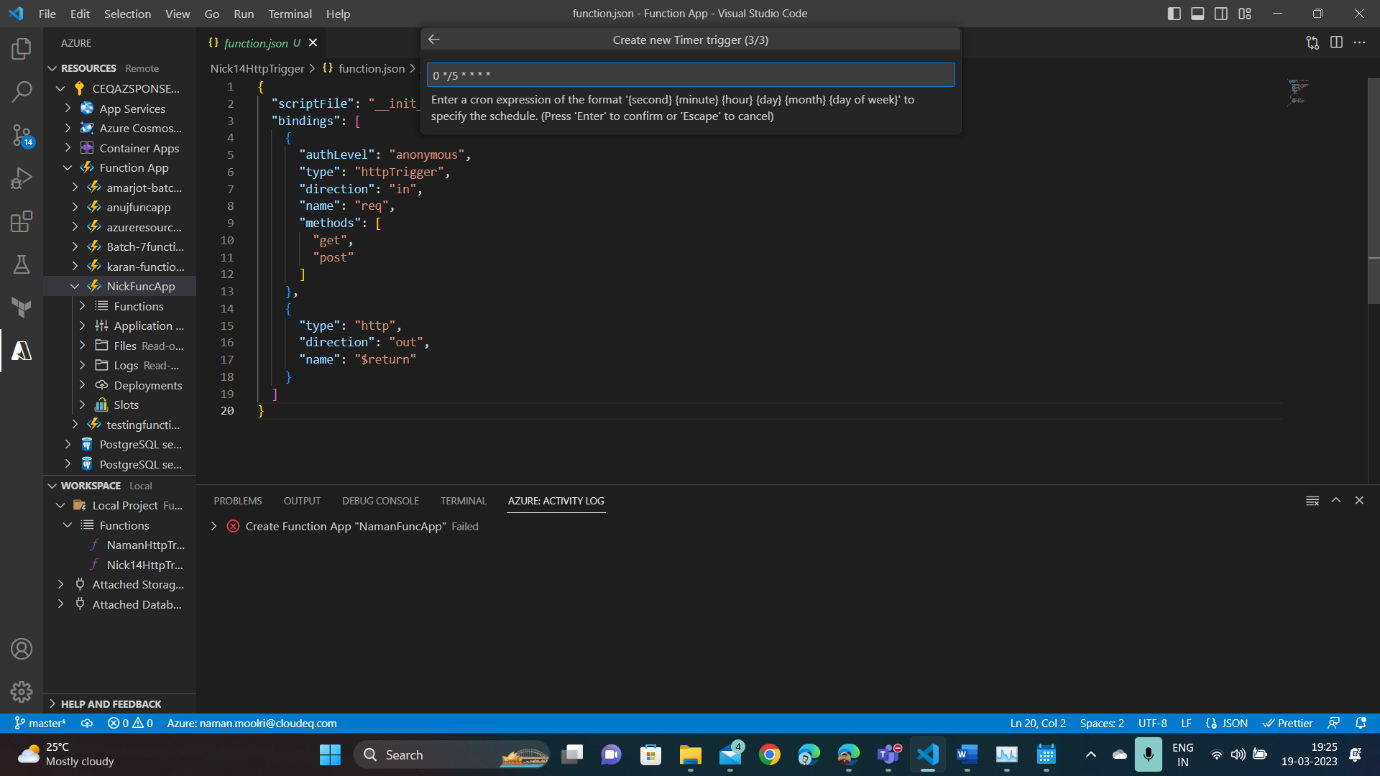
****

**Steps to implement Timer Trigger through VSCode**

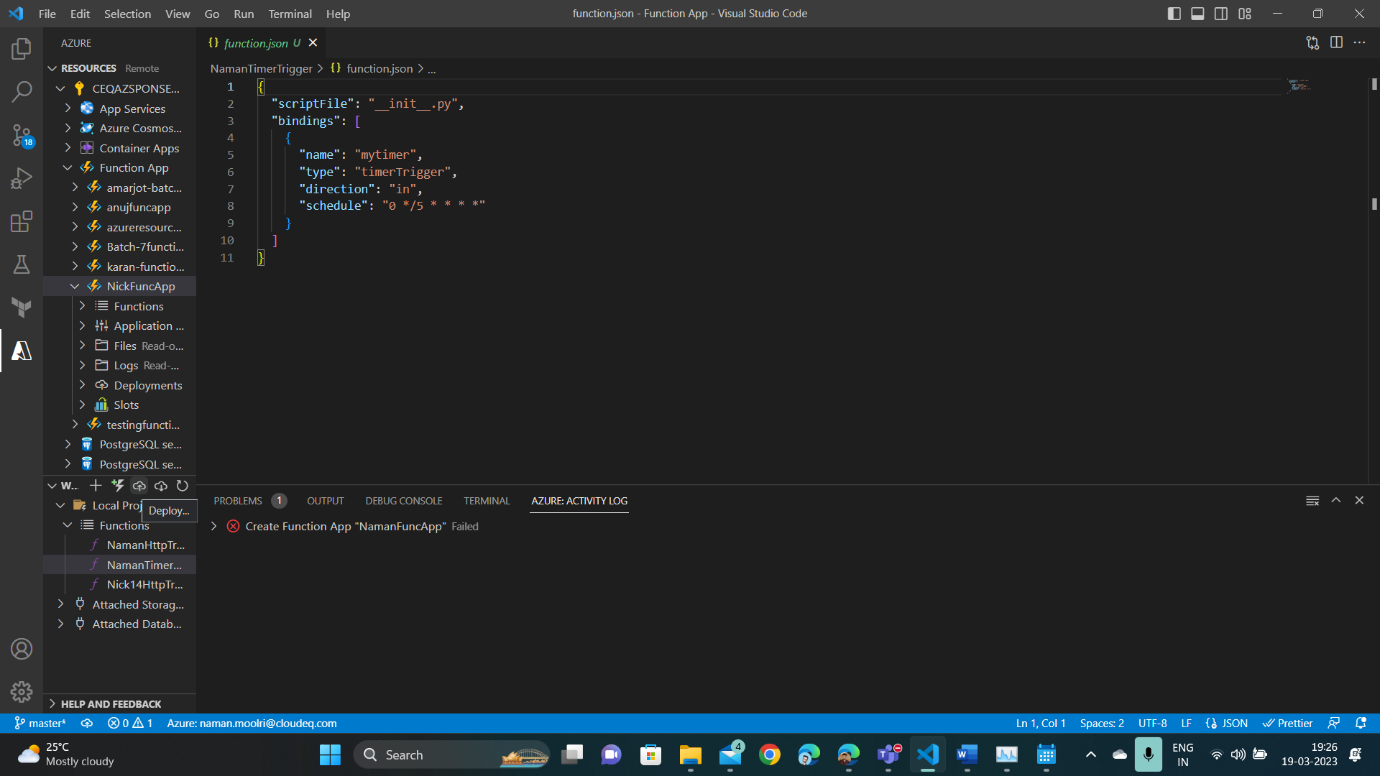
1. **Creating a Timer Trigger.**

****

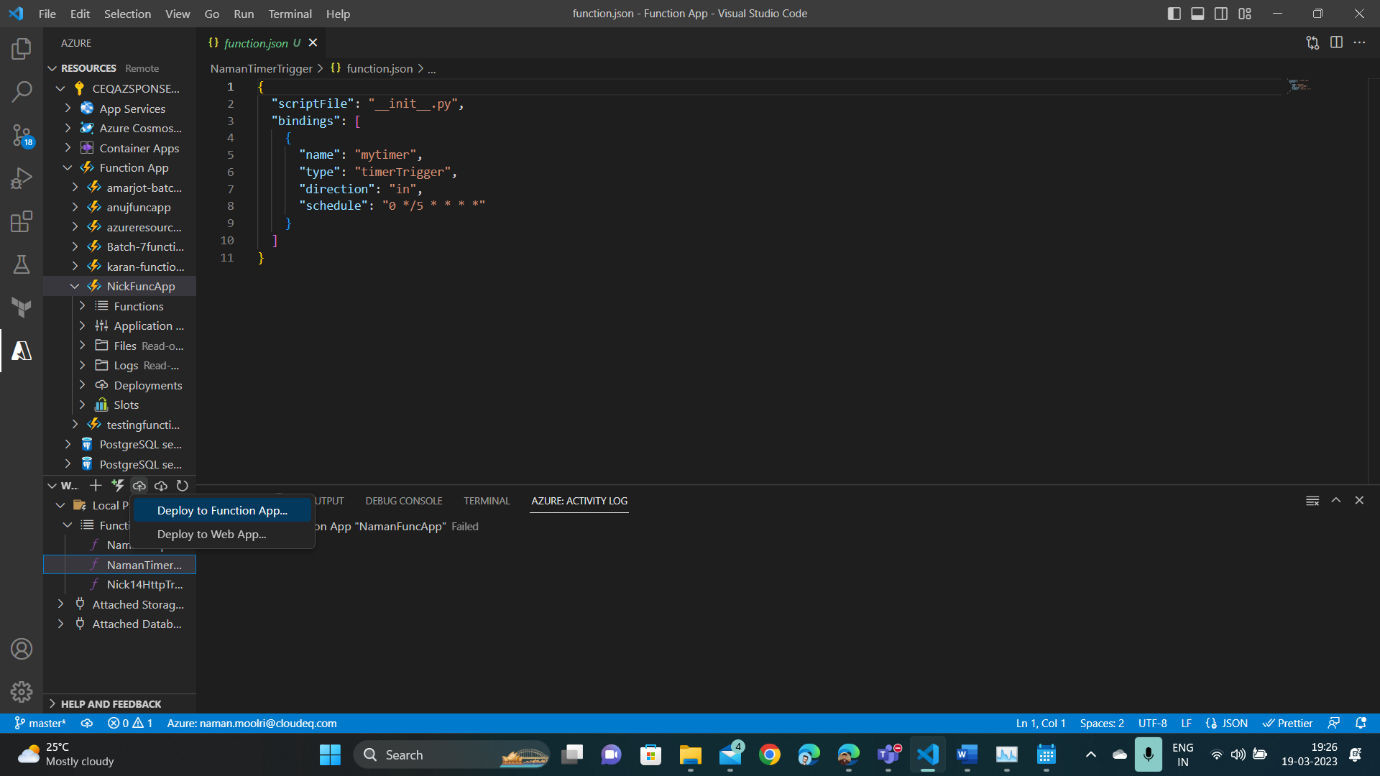
1. **Selecting the timeout time by entering the cron expression.**

****

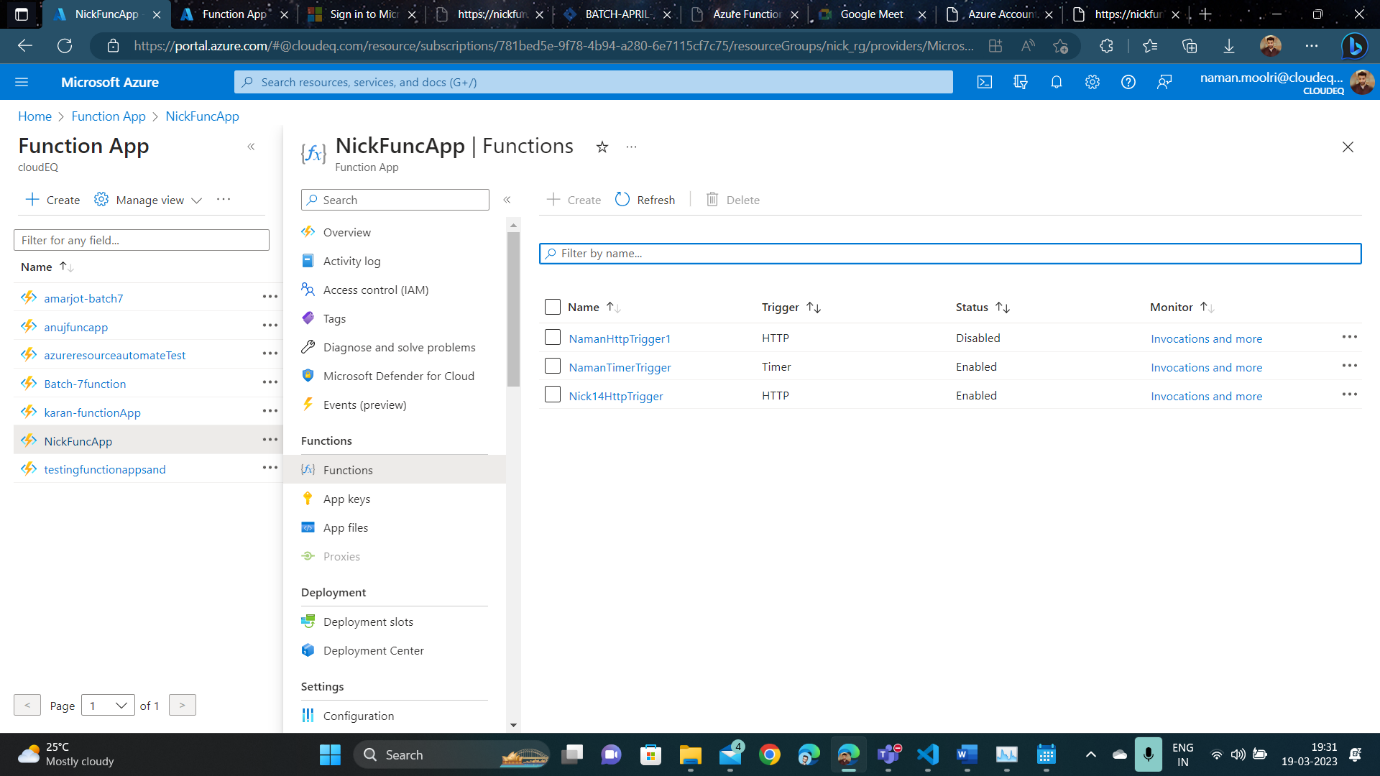
1. **Timer Trigger is created.**

****

1. **Now deploying our Timer Trigger to our Function App.**

****

1. **Timer Trigger is deployed in our Function App.**

****

1. **Checking our Code, Test and Monitor for connection.**

**A screenshot of a computer

Description automatically generated**

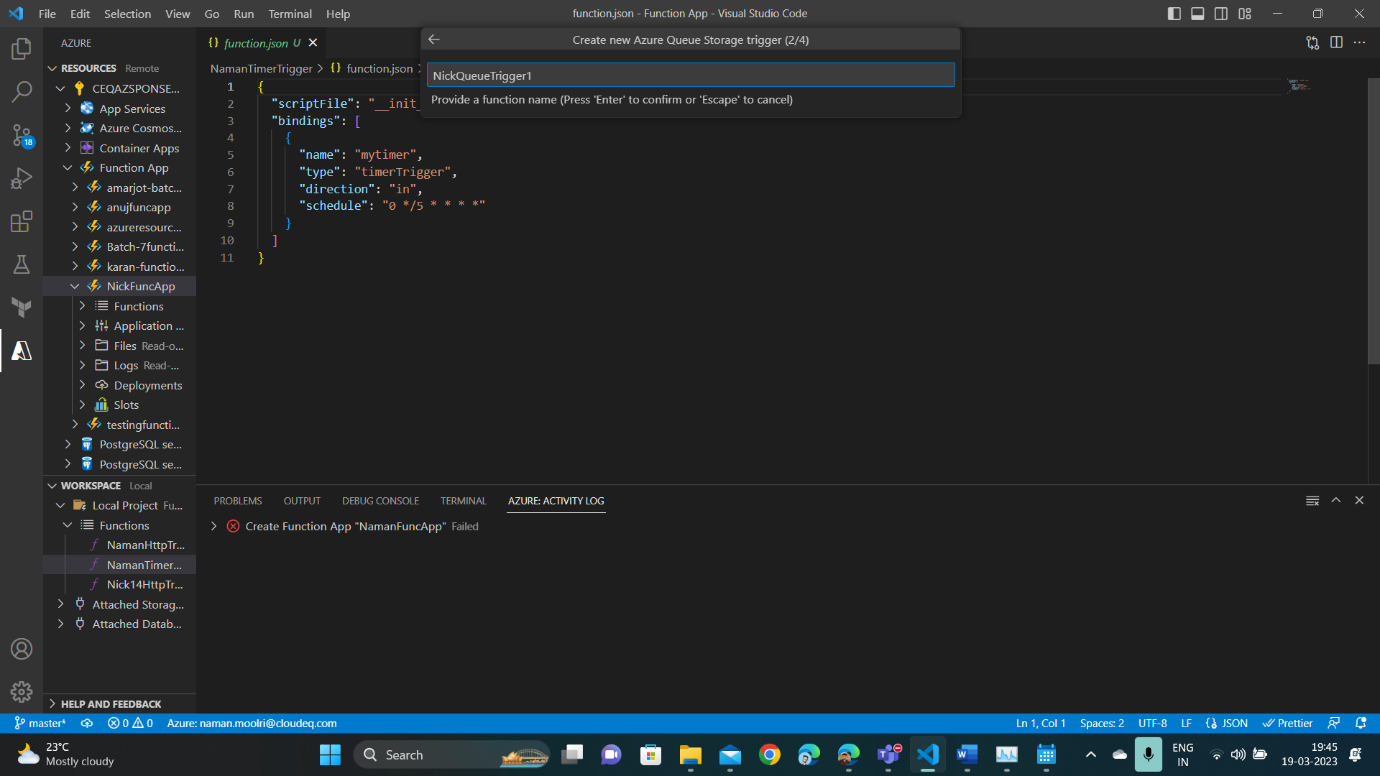
1. **Checking Logs.**

**A screenshot of a computer

Description automatically generated**

**Steps to implement Queue Trigger through VSCode**

1. **Creating a Queue Storage Trigger.**

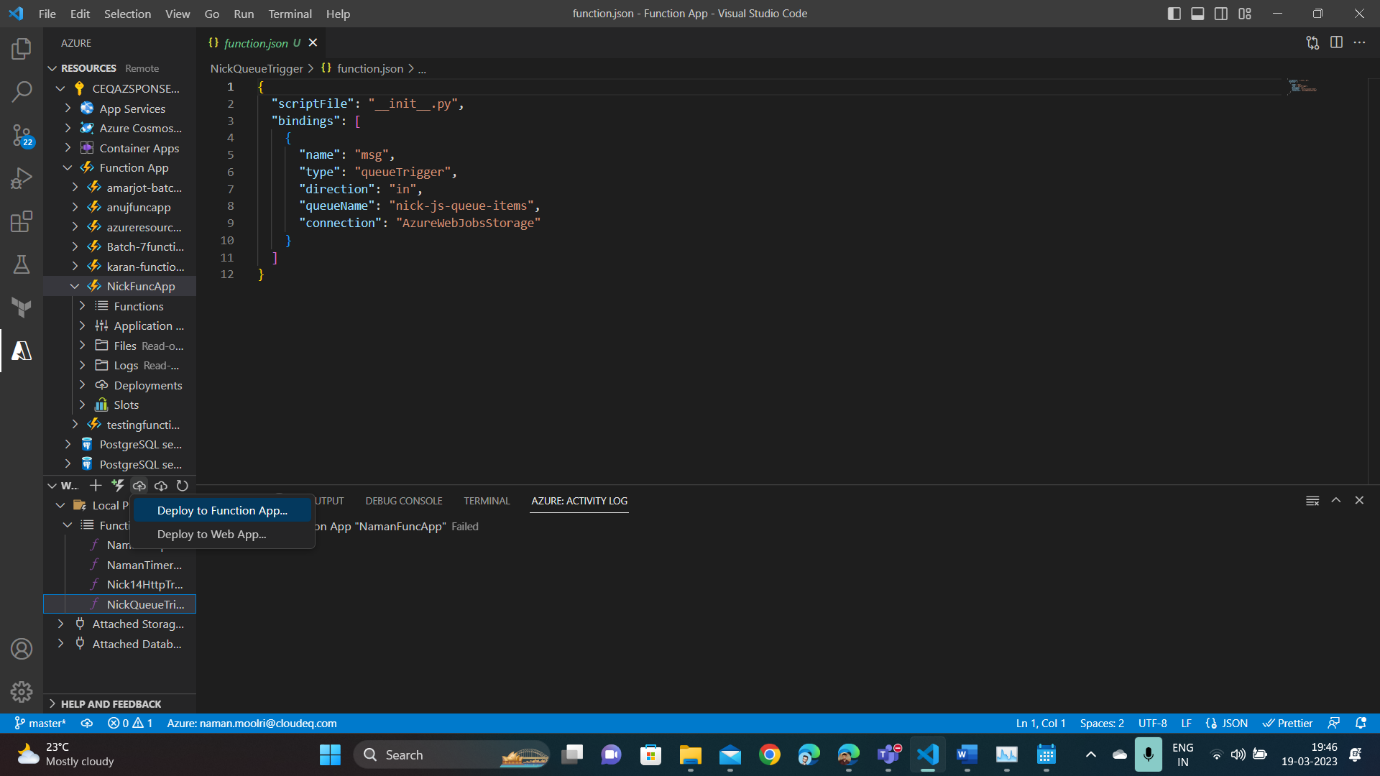
****

1. **Naming the queue from which the message will be read.**

**A screenshot of a computer

Description automatically generated**

1. **Deploying Queue Trigger to our Function App.**

****

1. **Confirming the deployment.**

**A screenshot of a computer

Description automatically generated**

1. **Trigger is deployed in our Function App.**

**A screenshot of a computer

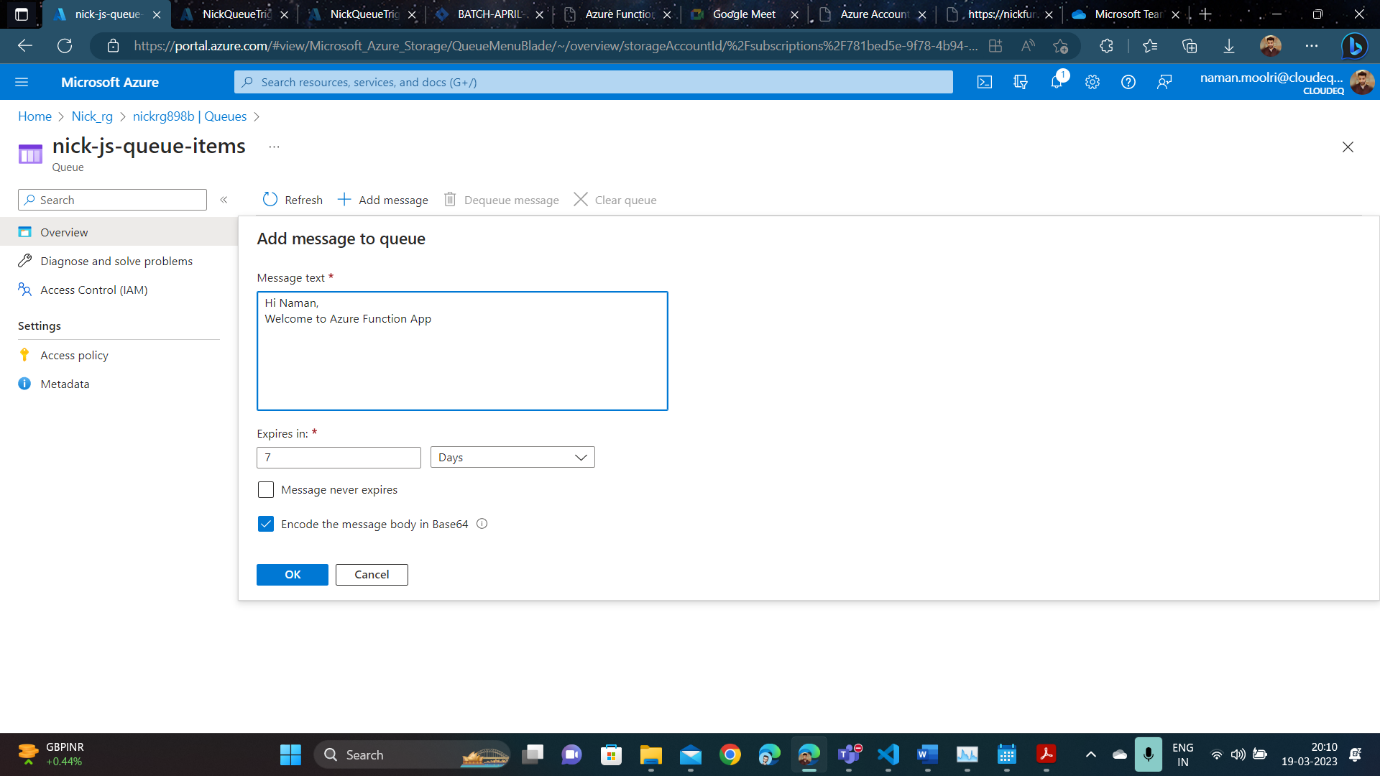
Description automatically generated**

1. **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.**

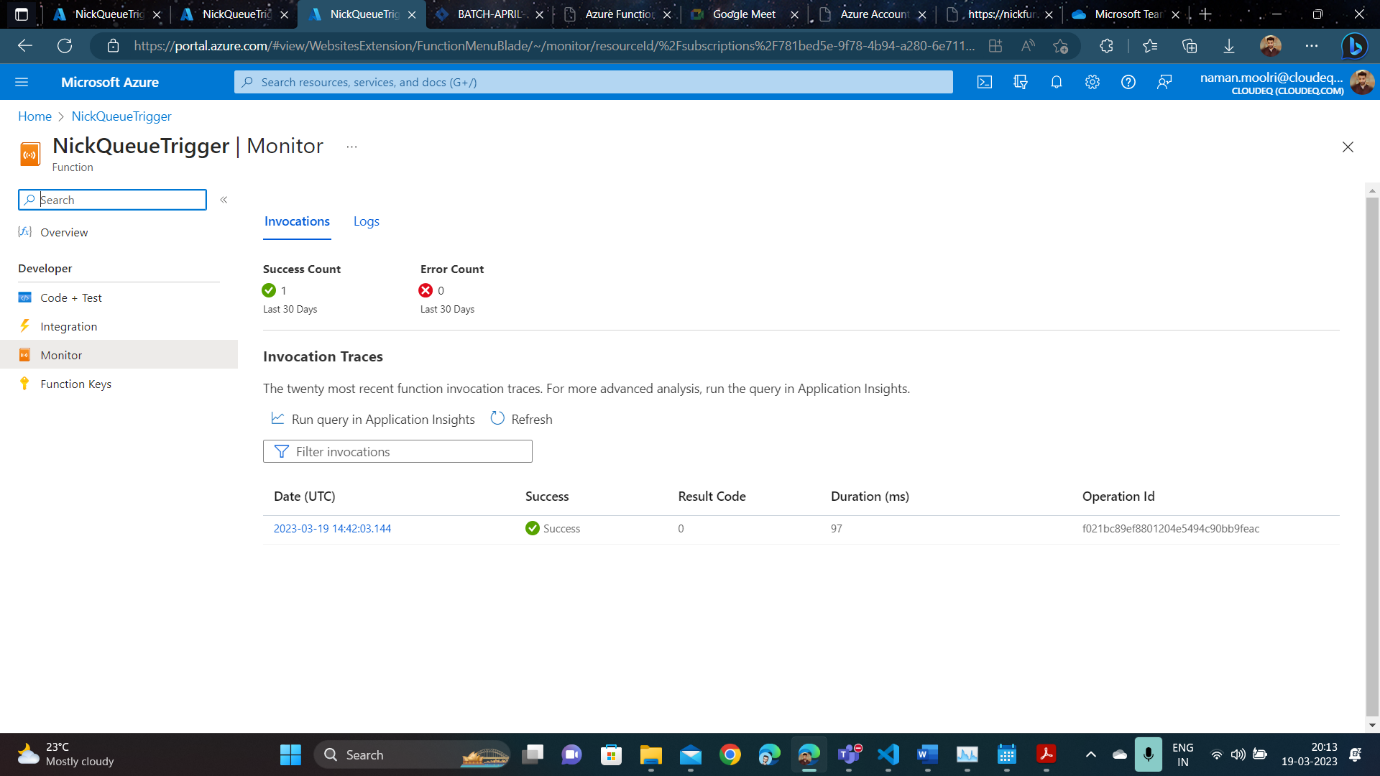
**A screenshot of a computer

Description automatically generated**

1. **Now, add message to the queue.**

****

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

****

1. **Checking the Logs.**

**A screenshot of a computer

Description automatically generated**