Project 1

Azure Function Listening to a Queue

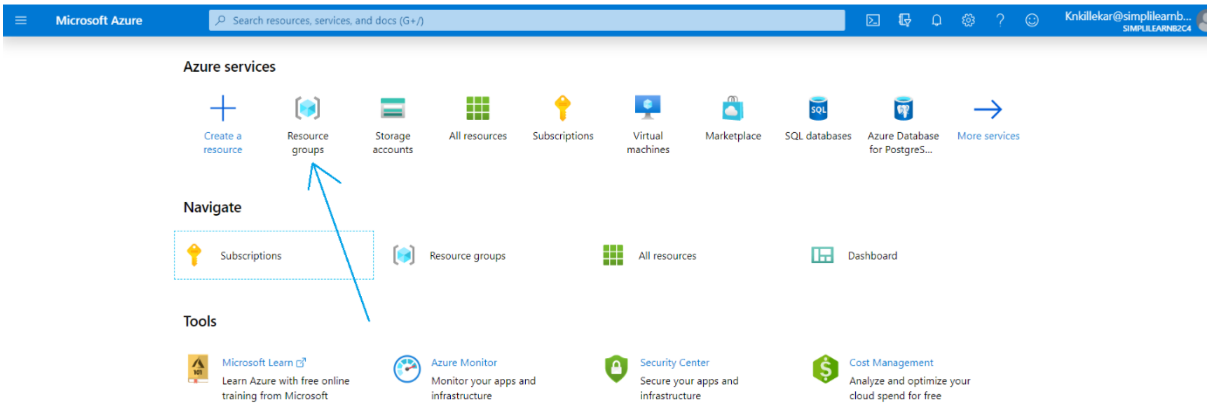
The InGen Soft decided to bring serverless functionality into their application workload using the Azure function and Storage queue. The idea behind this is that the appearance of the message inside a Queue shall trigger the Function execution. For validation, this task has been assigned to a developer to test this functionality using Visual Studio 2017.

The main tasks for this exercise are as follows:

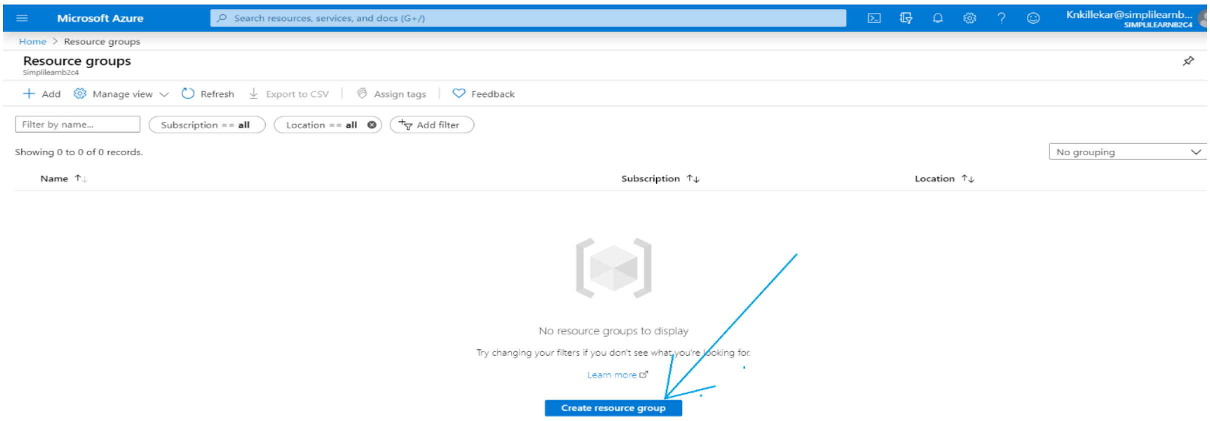
1. Log in to the Azure Portal and create the below services:
   - Resource Group
   - Storage Account
   - Input Queue
2. Create an Azure function using Visual Studio which also includes:
   - Queue Binding
   - Deployment from visual studio

Task 1:

Step 1: Log on to the Azur portal and select the resource group as shown in window below.

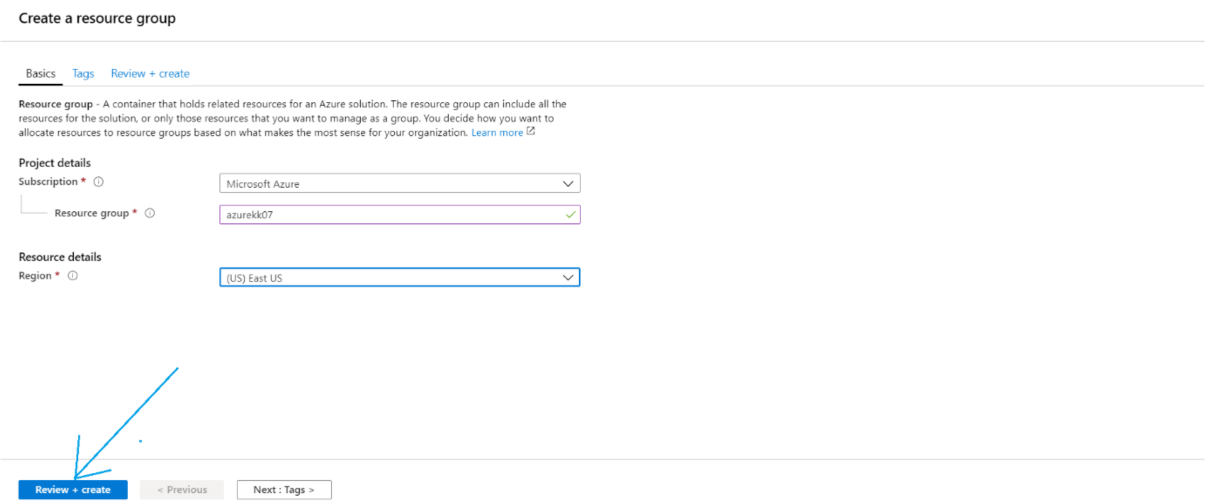


The following window will open and click on **"Create resource group"** as shown in window below.



Next, now enter the following details as shown in the window below and click on **“Review + create”**.

- **Resource group:** azurekk07.
- **Region :** East US.



Create a resource group

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

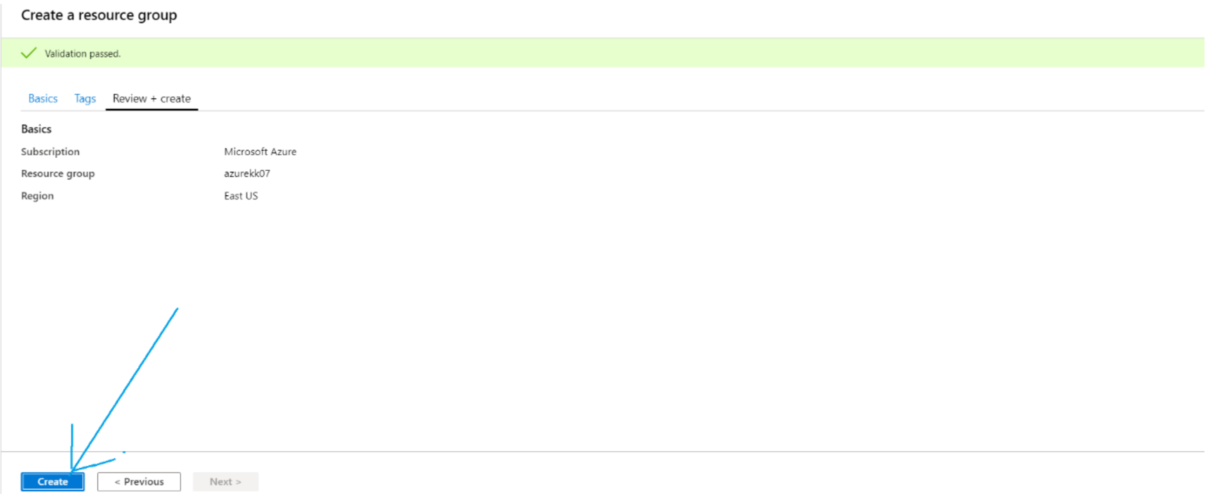
Subscription * : Microsoft Azure

Resource group * : azurekk07

Resource details

Region * : (US) East US

Once the validation is passed as shown below click on **“Create”** to create the resource group.



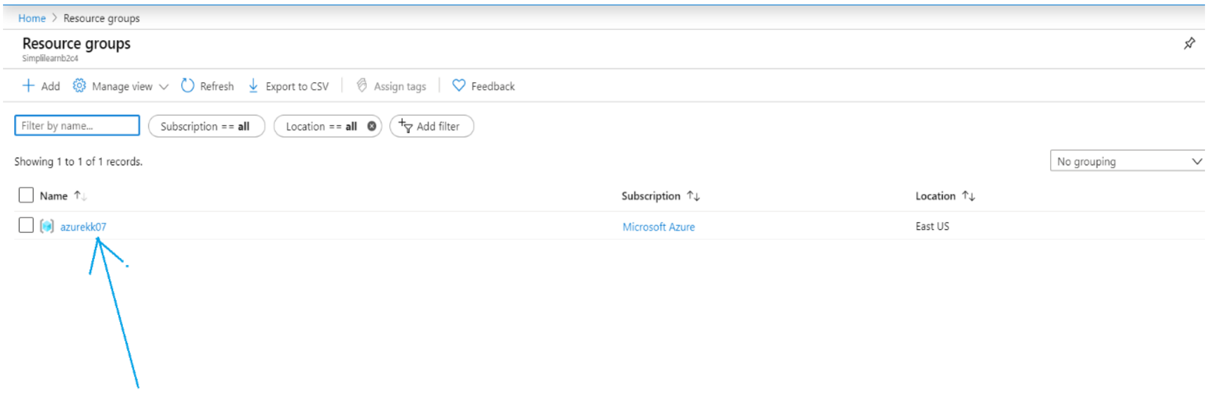
Create a resource group

Validation passed.

Basics

Subscription	Microsoft Azure
Resource group	azurekk07
Region	East US

The resource group is created successfully as shown in the window below.



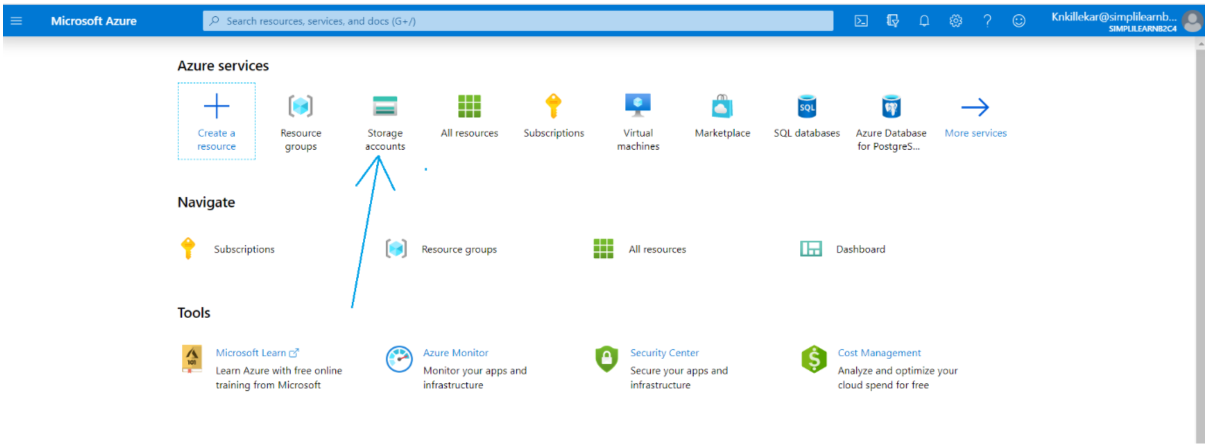
Resource groups

Simplilearnb2c4

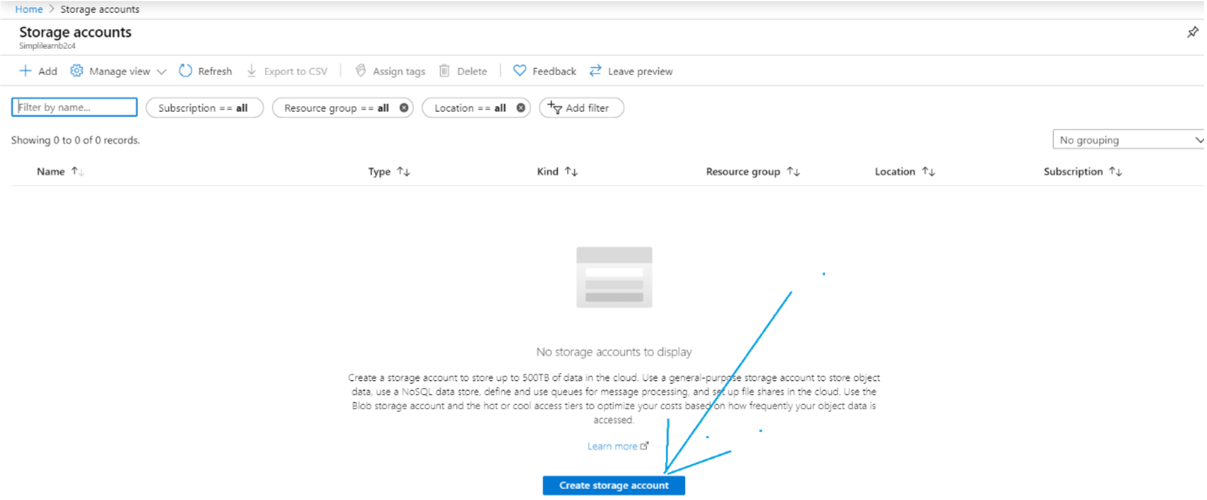
Showing 1 to 1 of 1 records.

Name	Subscription	Location
azurekk07	Microsoft Azure	East US

Step 2: Now, I will create storage account. To create storage account click on the **“Storage accounts”** as shown in the window below.



The following window will open and click on **“Create storage account”** as shown in window below.



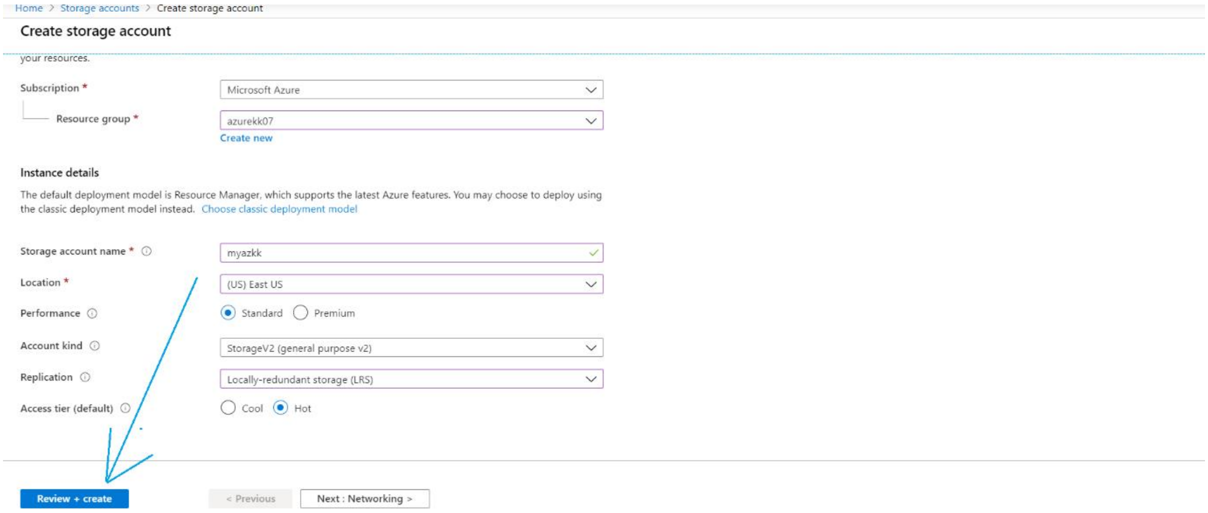
Showing 0 to 0 of 0 records.

No storage accounts to display

Create a storage account to store up to 500TB of data in the cloud. Use a general-purpose storage account to store object data, use a NoSQL data store, define and use queues for message processing, and set up file shares in the cloud. Use the Blob storage account and the hot or cool access tiers to optimize your costs based on how frequently your object data is accessed.

Now the enter the following details as shown in the window below and click on **“Review + create”**.

- **Resource group:** azurekk07 (select the resource group created in step 1).
- **Storage account name:** myazkk.
- **Location:** East US.
- **Performance:** standard.
- **Account kind:** StorageV2 (general purpose v2).
- **Replication:** local-redundant storage(LRS).
- **Access tier (default):** Hot.



Home > Storage accounts > Create storage account

Create storage account

your resources.

Subscription * Microsoft Azure

Resource group * azurekk07

Create new

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name * myazkk

Location * (US) East US

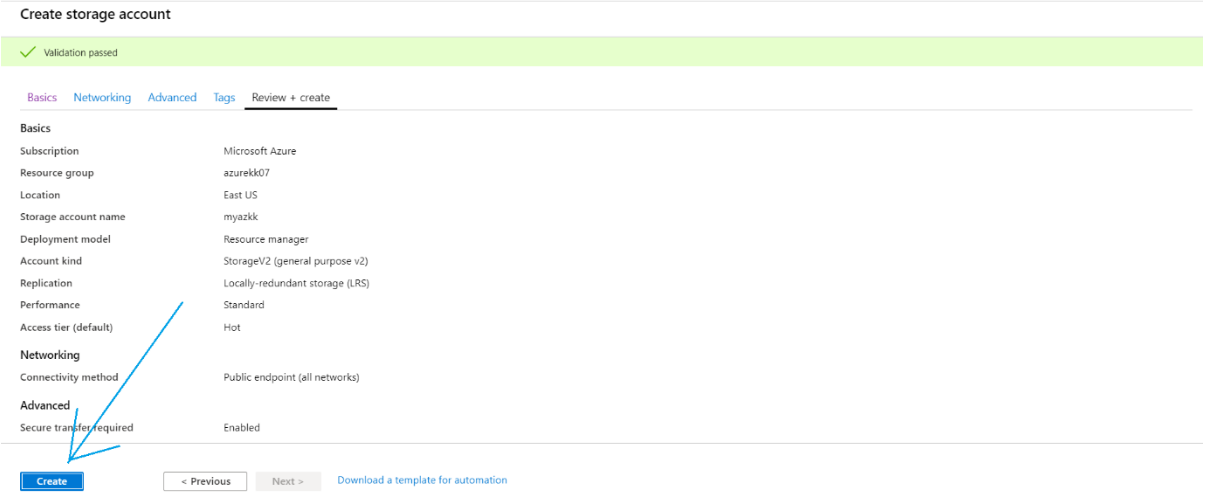
Performance ● Standard ○ Premium

Account kind StorageV2 (general purpose v2)

Replication Locally-redundant storage (LRS)

Access tier (default) ○ Cool ● Hot

The validation is passed as shown in the window below and click on **“Create”** to create the storage account.



Create storage account

✓ Validation passed

Basics

Subscription	Microsoft Azure
Resource group	azurekk07
Location	East US
Storage account name	myazkk
Deployment model	Resource manager
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)
Performance	Standard
Access tier (default)	Hot

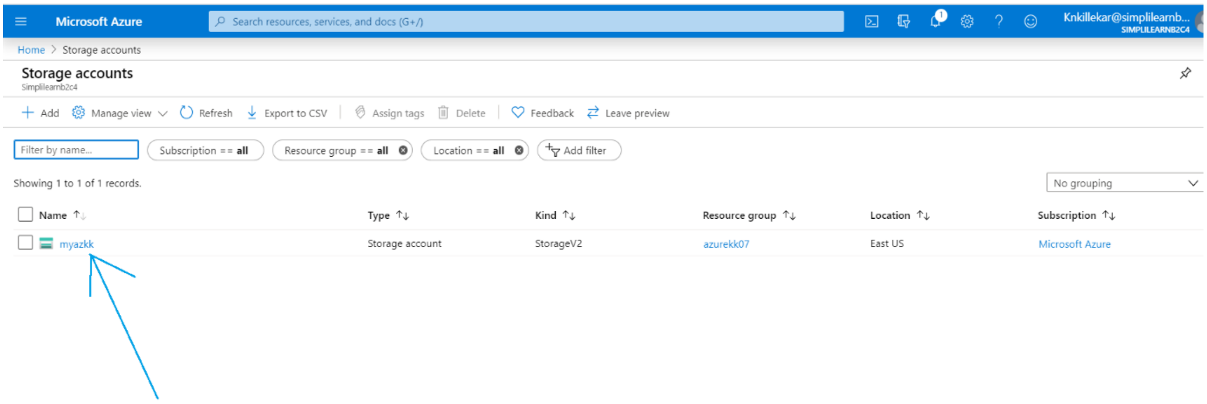
Networking

Connectivity method	Public endpoint (all networks)

Advanced

Secure transfer required	Enabled

The storage account **“myazkk”** is created successfully as shown in the window below.



Microsoft Azure
Search resources, services, and docs (G+/)
Knkillekar@simplilearnb...
SIMPLILEARNB2C4
Home > Storage accounts

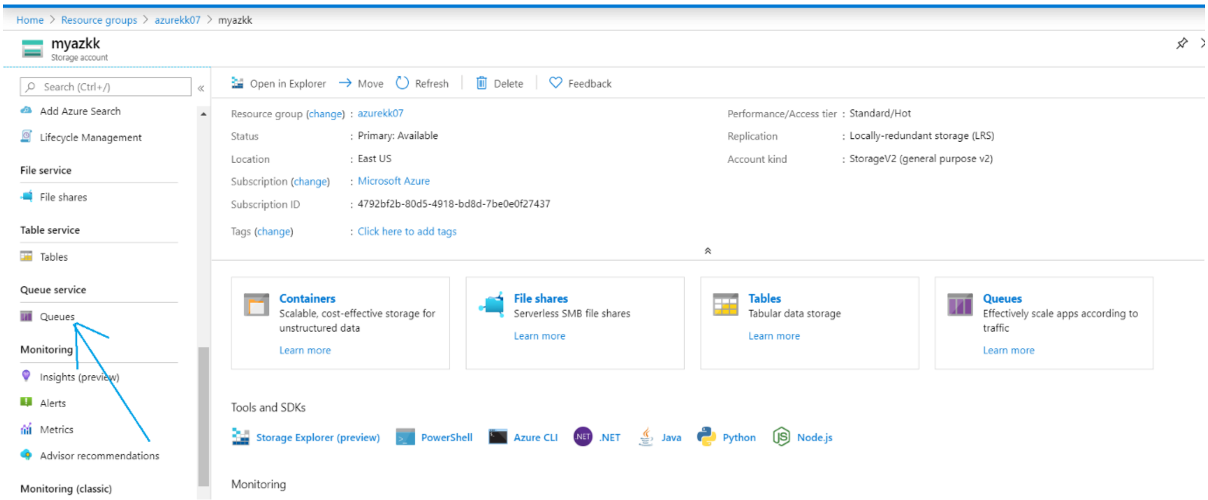
Storage accounts

Simplilearnb2c4

Showing 1 to 1 of 1 records.

Name	Type	Kind	Resource group	Location	Subscription
myazkk	Storage account	StorageV2	azurekk07	East US	Microsoft Azure

Step 3: To create the input queue , go to storage account and select **“Queues”** from Queue services as shown in the window below.



Resource group (change) : azurekk07
Status : Primary: Available
Location : East US
Subscription (change) : Microsoft Azure
Subscription ID : 4792bf2b-80d5-4918-bd8d-7be0e0f27437
Tags (change) : Click here to add tags

Performance/Access tier : Standard/Hot
Replication : Locally-redundant storage (LRS)
Account kind : StorageV2 (general purpose v2)

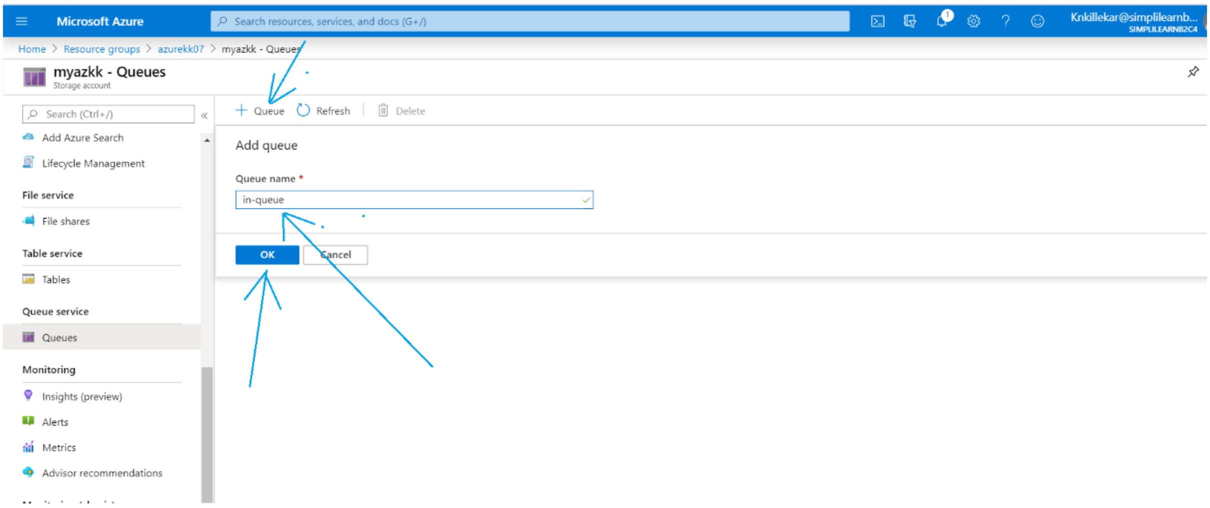
Containers Scalable, cost-effective storage for unstructured data Learn more

File shares Serverless SMB file shares Learn more

Tables Tabular data storage Learn more

Queues Effectively scale apps according to traffic Learn more

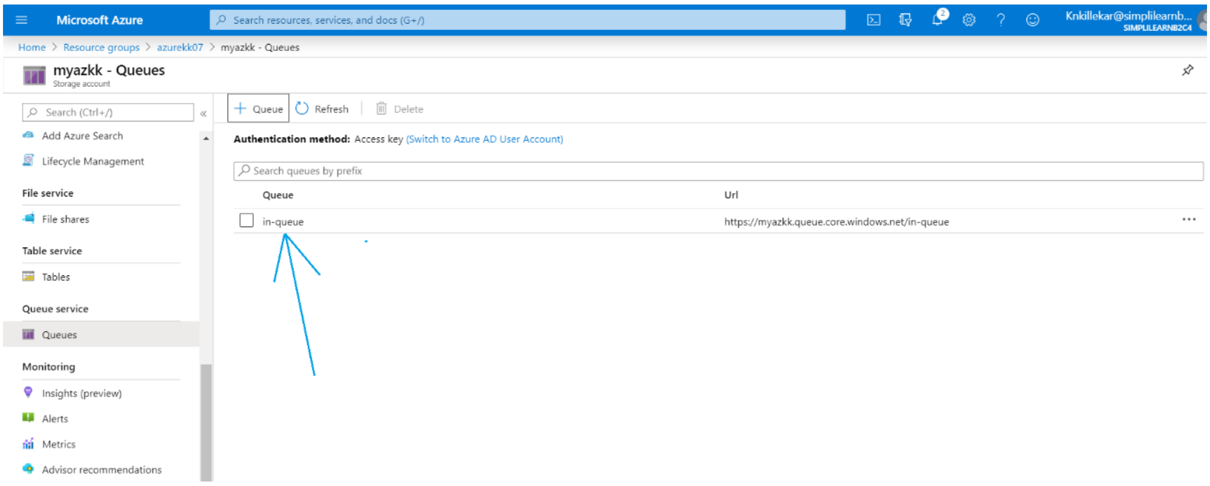
Now click on **“Queue”** and enter the queue name as **“in-queue”** as shown in the window below and click on **“ok”.**



Add queue

Queue name * in-queue

The **in-queue** is successfully created as shown in the window below.



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

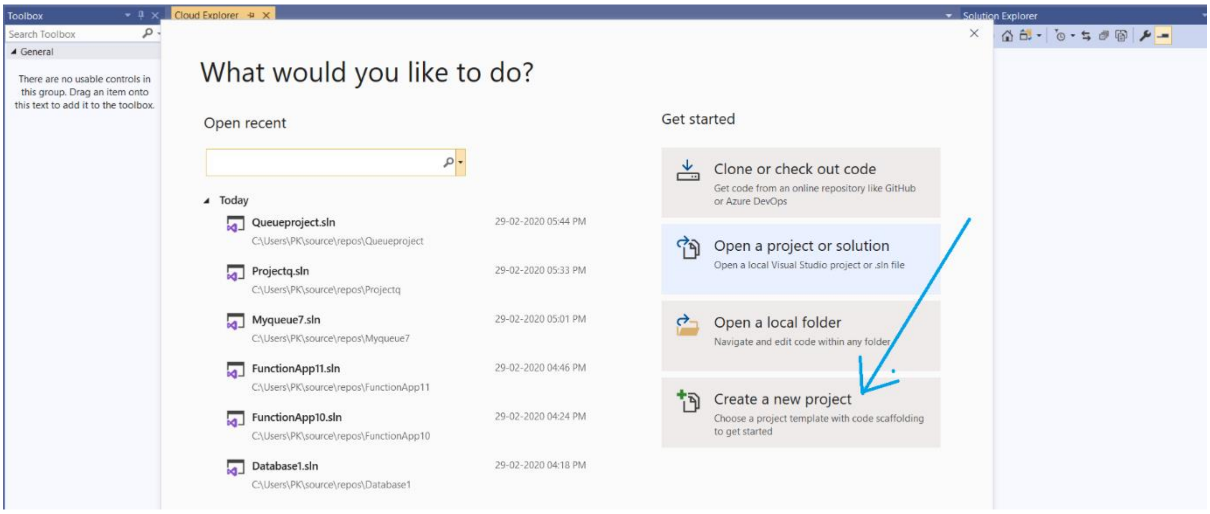
Search queues by prefix

Queue	Url
□ in-queue	https://myazkk.queue.core.windows.net/in-queue

Task 2: Create an Azure function using Visual Studio which also includes:

- Queue Binding
- Deployment from visual studio

Step 1: Open visual studio tool and click on **" Create a new project"** as shown in the window below.



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

What would you like to do?

Open recent

Today

Queueproject.sln 29-02-2020 05:44 PM
C:\Users\PK\source\repos\Queueproject

Projectq.sln 29-02-2020 05:33 PM
C:\Users\PK\source\repos\Projectq

Myqueue7.sln 29-02-2020 05:01 PM
C:\Users\PK\source\repos\Myqueue7

FunctionApp11.sln 29-02-2020 04:46 PM
C:\Users\PK\source\repos\FunctionApp11

FunctionApp10.sln 29-02-2020 04:24 PM
C:\Users\PK\source\repos\FunctionApp10

Database1.sln 29-02-2020 04:18 PM
C:\Users\PK\source\repos\Database1

Get started

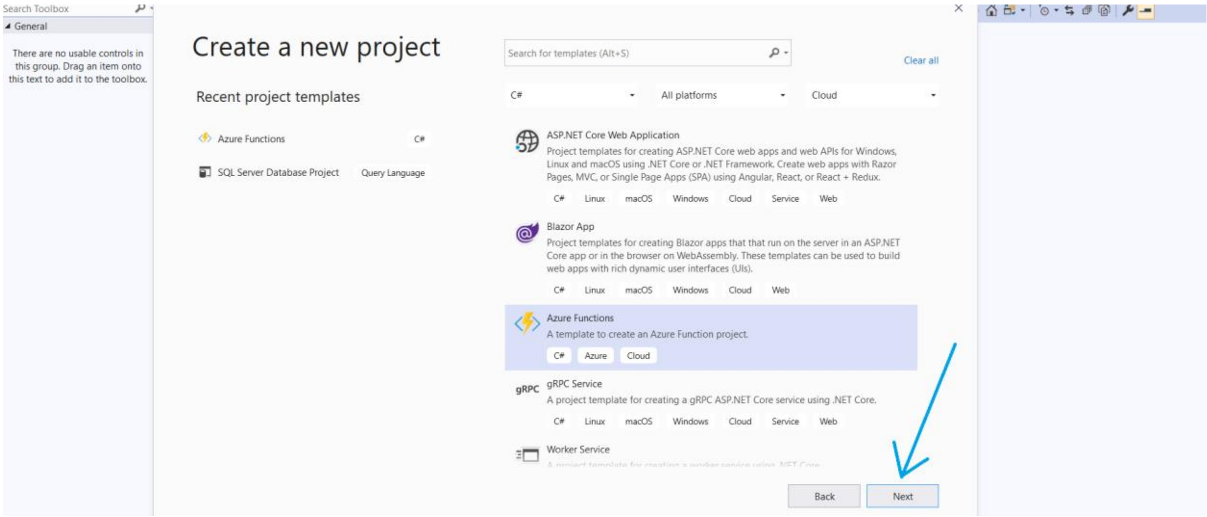
Clone or check out code
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Now select **"Azure functions"** as template and click on **"Next"** as shown in the window below.



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Create a new project

Recent project templates

Azure Functions C#
SQL Server Database Project Query Language

Search for templates (Alt+S)

C# | All platforms | Cloud

ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web

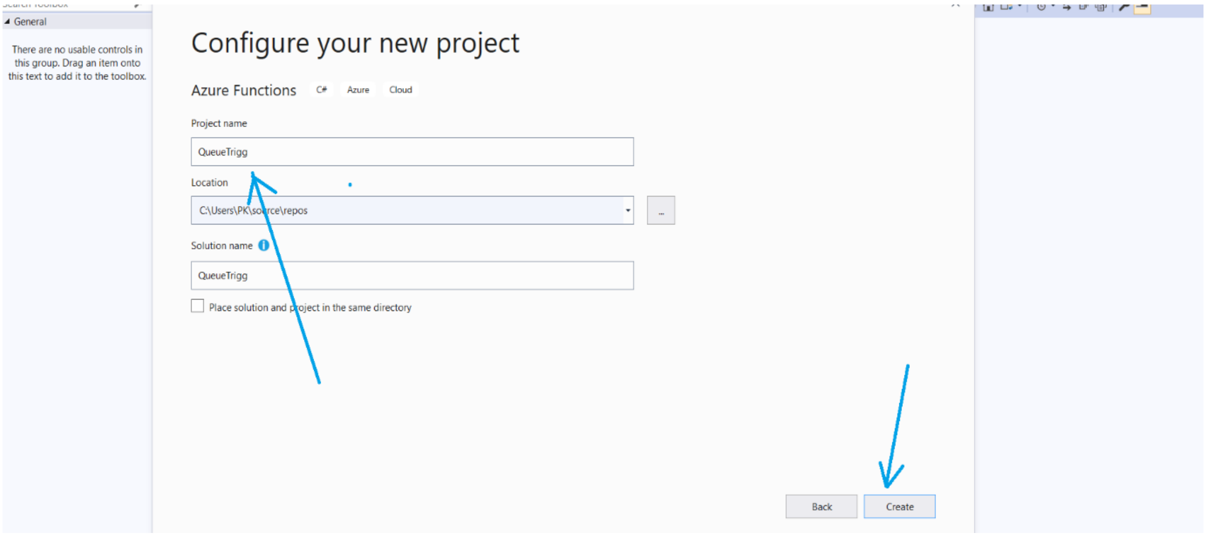
Blazor App
Project templates for creating Blazor apps that that run on the server in an ASP.NET Core app or in the browser on WebAssembly. These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web

Azure Functions
A template to create an Azure Function project.
C# Azure Cloud

gRPC Service
A project template for creating a gRPC ASP.NET Core service using .NET Core.
C# Linux macOS Windows Cloud Service Web

Worker Service
A project template for creating a worker service using .NET Core

Next, enter the project name as **“QueueTrigg”** as shown in the window below and click on **“Create”** .



Configure your new project

Azure Functions C# Azure Cloud

Project name: QueueTrigg

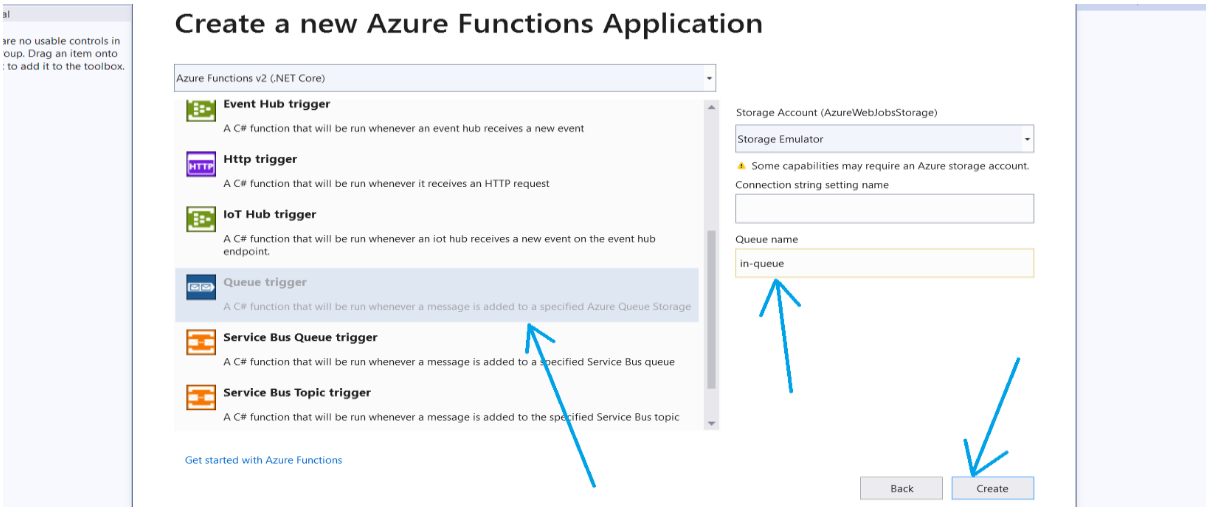
Location: C:\Users\PK\source\repos

Solution name: QueueTrigg

□ Place solution and project in the same directory

General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
...
Back
Create

Now, select **“Queue trigger”** and enter the Queue name **as “in-queue”** as shown in the window below and click on **“Create”.**



Create a new Azure Functions Application

Azure Functions v2 (.NET Core)

Event Hub trigger
A C# function that will be run whenever an event hub receives a new event

Http trigger
A C# function that will be run whenever it receives an HTTP request

IoT Hub trigger
A C# function that will be run whenever an iot hub receives a new event on the event hub endpoint.

Queue trigger
A C# function that will be run whenever a message is added to a specified Azure Queue Storage

Service Bus Queue trigger
A C# function that will be run whenever a message is added to a specified Service Bus queue

Service Bus Topic trigger
A C# function that will be run whenever a message is added to the specified Service Bus topic

Storage Account (AzureWebJobsStorage): Storage Emulator

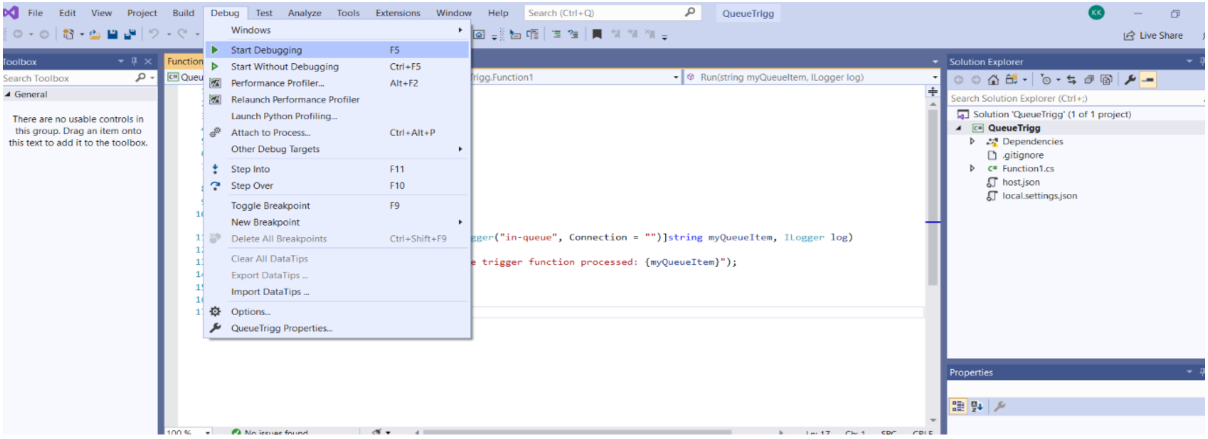
⚠ Some capabilities may require an Azure storage account.

Connection string setting name:

Queue name: in-queue

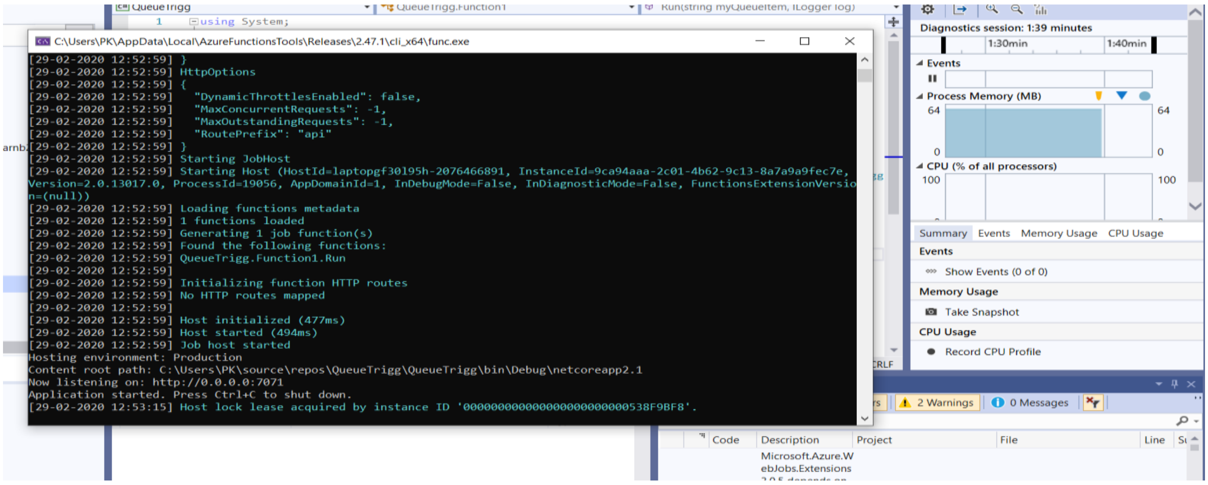
al
are no usable controls in
oup. Drag an item onto
to add it to the toolbox.
Get started with Azure Functions
Back
Create

Step 2: Now go to debug option and click on **“Start Debugging”** as shown in the window below.



```
gger("in-queue", Connection = "")]string myQueueItem, ILogger log)
e trigger function processed: {myQueueItem}");
```

After debugging, a Azure function tool window will pop up as shown below.

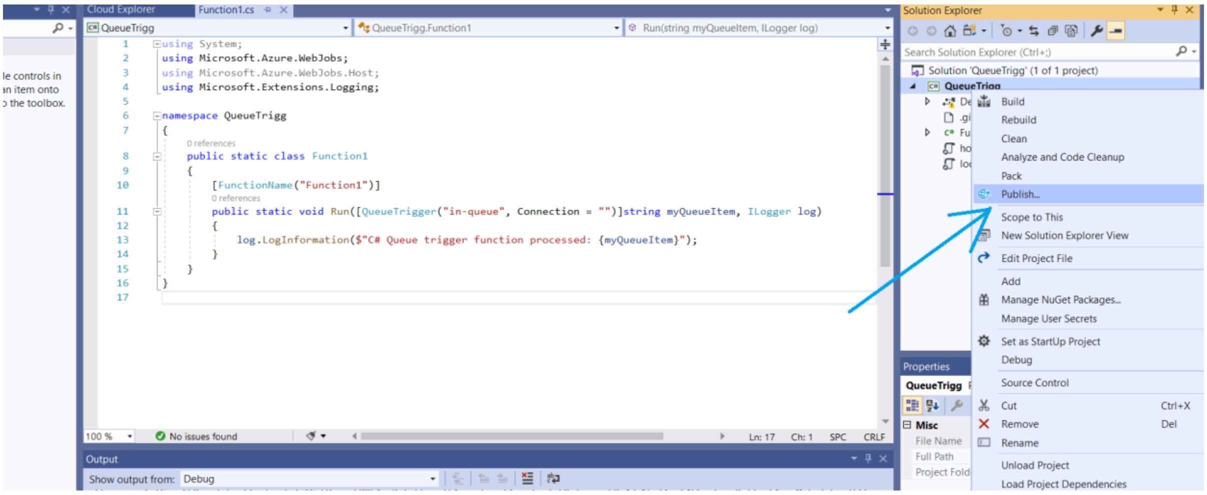


C:\Users\PK\AppData\Local\AzureFunctionsTools\Releases\2.47.1\cli_x64\func.exe

```
[29-02-2020 12:52:59] }
[29-02-2020 12:52:59] HttpOptions
[29-02-2020 12:52:59] {
[29-02-2020 12:52:59]   "DynamicThrottlesEnabled": false,
[29-02-2020 12:52:59]   "MaxConcurrentRequests": -1,
[29-02-2020 12:52:59]   "MaxOutstandingRequests": -1,
[29-02-2020 12:52:59]   "RoutePrefix": "api"
[29-02-2020 12:52:59] }
[29-02-2020 12:52:59] Starting JobHost
[29-02-2020 12:52:59] Starting Host (HostId=laptopgf30195h-2076466891, InstanceId=9ca94aaa-2c01-4b62-9c13-8a7a9a9fec7e,
Version=2.0.13017.0, ProcessId=19056, AppDomainId=1, InDebugMode=False, InDiagnosticMode=False, FunctionsExtensionVersio
n=(null))
[29-02-2020 12:52:59] Loading functions metadata
[29-02-2020 12:52:59] 1 functions loaded
[29-02-2020 12:52:59] Generating 1 job function(s)
[29-02-2020 12:52:59] Found the following functions:
[29-02-2020 12:52:59] QueueTrigg.Function1.Run
[29-02-2020 12:52:59]
[29-02-2020 12:52:59] Initializing function HTTP routes
[29-02-2020 12:52:59] No HTTP routes mapped
[29-02-2020 12:52:59]
[29-02-2020 12:52:59] Host initialized (477ms)
[29-02-2020 12:52:59] Host started (494ms)
[29-02-2020 12:52:59] Job host started
Hosting environment: Production
Content root path: C:\Users\PK\source\repos\QueueTrigg\QueueTrigg\bin\Debug\netcoreapp2.1
Now listening on: http://0.0.0.0:7071
Application started. Press Ctrl+C to shut down.
[29-02-2020 12:53:15] Host lock lease acquired by instance ID '000000000000000000000000538F9BF8'.
```

Now stop debugging by going to debug option and select **"stop debugging".**

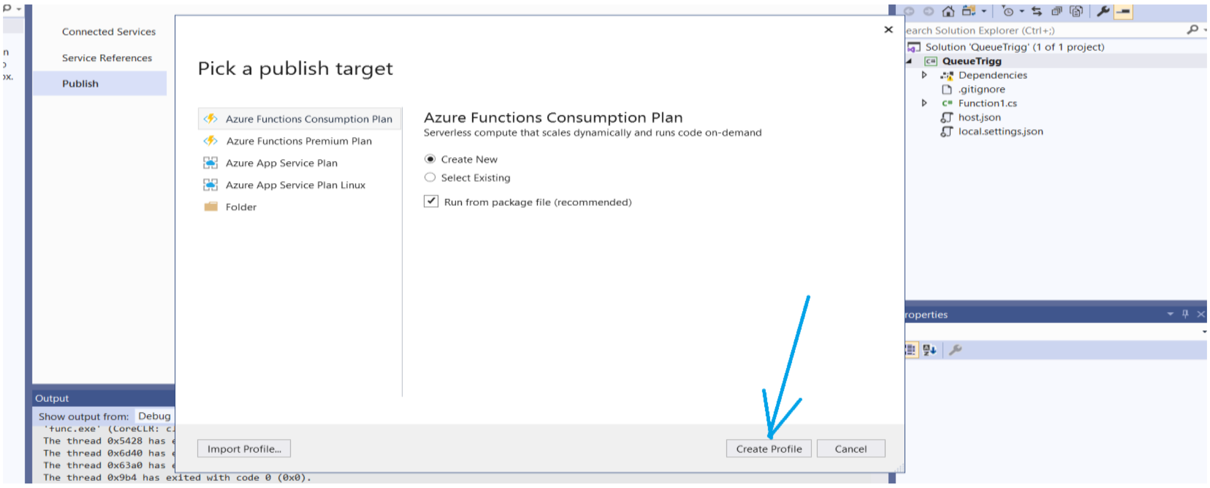
Step 3: Next, to deploy function to azure portal, right click on **"QueueTrigg"** in the solution explorer and select **"Publish"** option as shown in the window below.



```
using System;
using Microsoft.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using Microsoft.Extensions.Logging;

namespace QueueTrigg
{
    0 references
    public static class Function1
    {
        [FunctionName("Function1")]
        0 references
        public static void Run([QueueTrigger("in-queue", Connection = "")]string myQueueItem, ILogger log)
        {
            log.LogInformation($"C# Queue trigger function processed: {myQueueItem}");
        }
    }
}
```

Now keep the default settings and select **"Create Profile"** as shown in the window below.



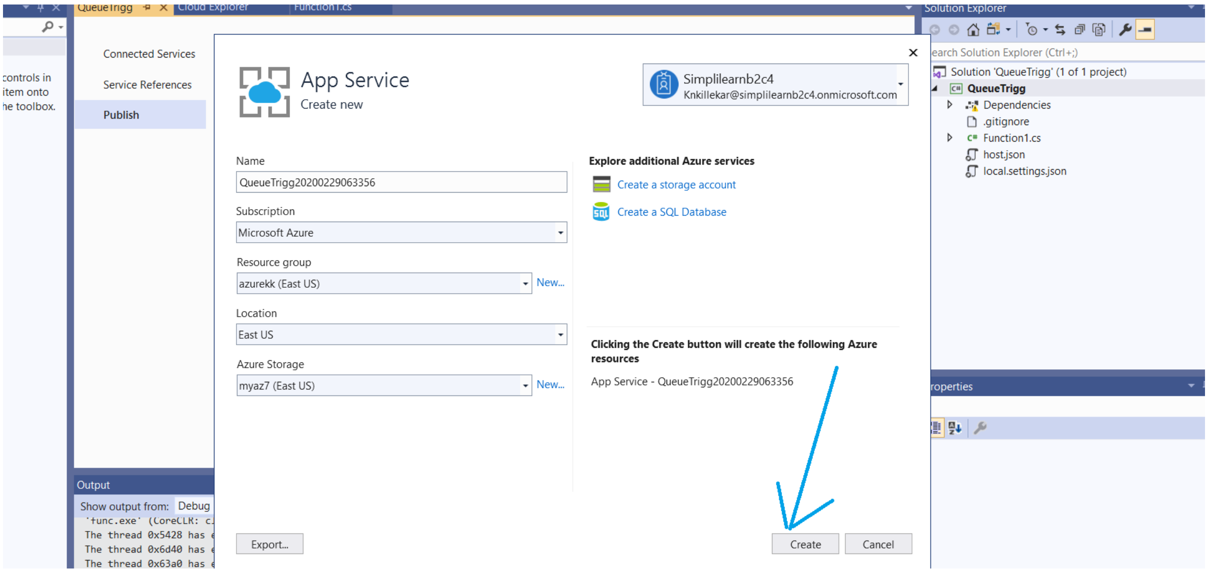
Pick a publish target

Azure Functions Consumption Plan
Serverless compute that scales dynamically and runs code on-demand

● Create New
○ Select Existing
☑ Run from package file (recommended)

```
 'func.exe' (CoreCLR: c
The thread 0x5428 has
The thread 0x6d40 has
The thread 0x63a0 has
The thread 0x9b4 has exited with code 0 (0x0).
```

Now enter the details as shown in the window below and click on “ **Create”.**



App Service
Create new

Simplilearnb2c4
Knkillekar@simplilearnb2c4.onmicrosoft.com

Name: QueueTrigg20200229063356

Subscription: Microsoft Azure

Resource group: azurekk (East US) New...

Location: East US

Azure Storage: myaz7 (East US) New...

Explore additional Azure services

Create a storage account

Create a SQL Database

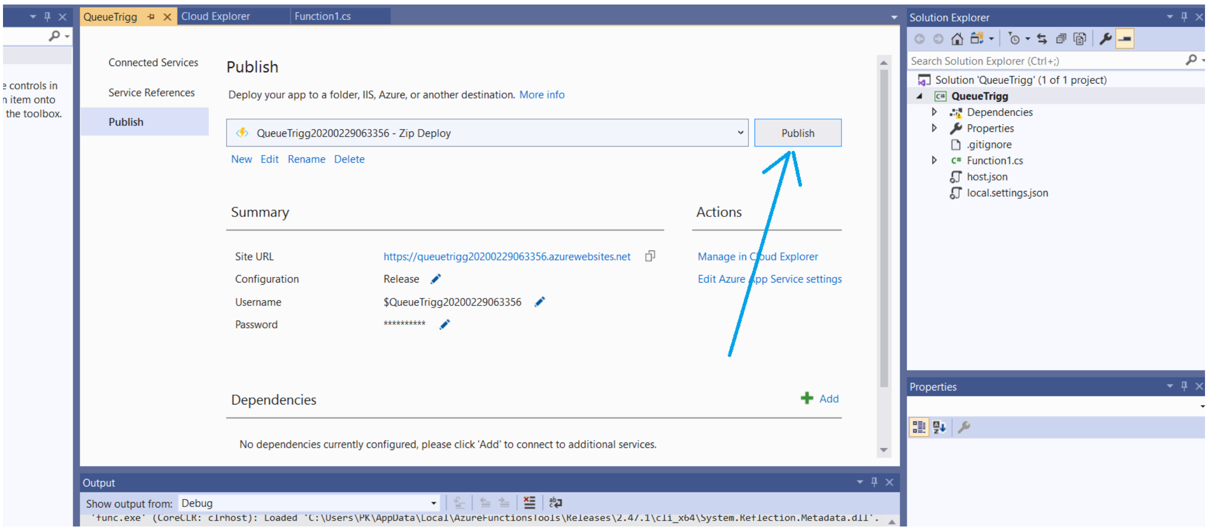
Clicking the Create button will create the following Azure resources

App Service - QueueTrigg20200229063356

Show output from: Debug

```
'func.exe' (CoreCLR: c
The thread 0x5428 has
The thread 0x6d40 has
The thread 0x63a0 has
```

Next, click on **“Publish”** to deploy the app as shown in the window below.



Publish

Deploy your app to a folder, IIS, Azure, or another destination. More info

QueueTrigg20200229063356 - Zip Deploy

Summary

Site URL: https://queuetrigg20200229063356.azurewebsites.net

Configuration: Release

Username: $QueueTrigg20200229063356

Password: **********

Actions

Manage in Cloud Explorer

Edit Azure App Service settings

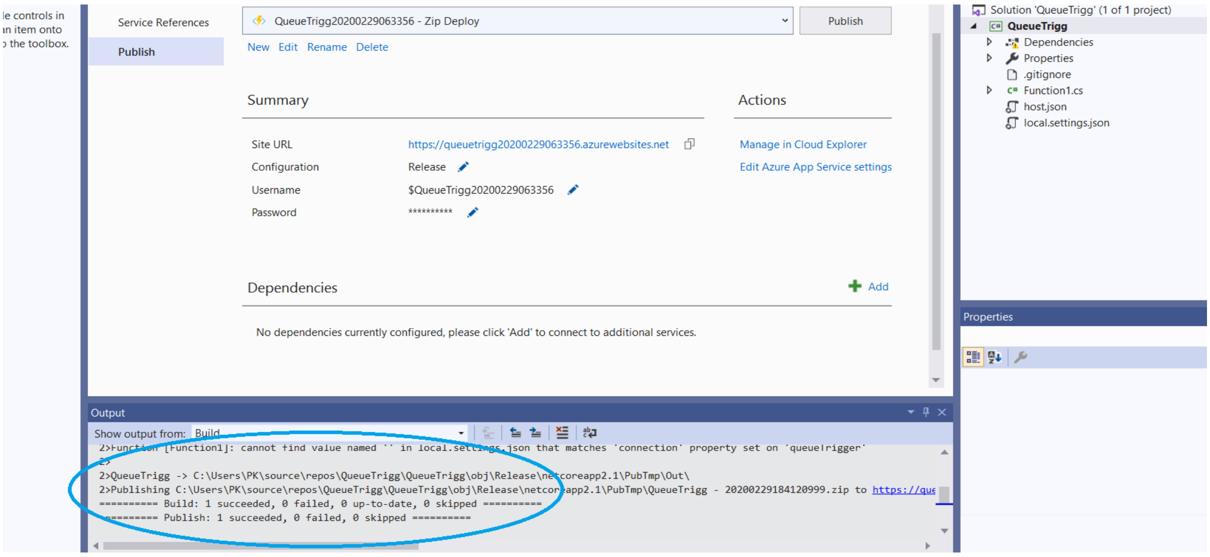
Dependencies

No dependencies currently configured, please click 'Add' to connect to additional services.

Show output from: Debug

```
'func.exe' (CoreCLR: clrhost): Loaded 'C:\Users\PK\AppData\Local\AzureFunctionsTools\Releases\2.47.1\cli_x64\System.Reflection.Metadata.dll'.
```

The following window shows that app is deployed successfully.



QueueTrigg20200229063356 - Zip Deploy

Summary

Site URL https://queuetrigg20200229063356.azurewebsites.net
Configuration Release
Username $QueueTrigg20200229063356
Password *********

Actions

Manage in Cloud Explorer
Edit Azure App Service settings

Dependencies

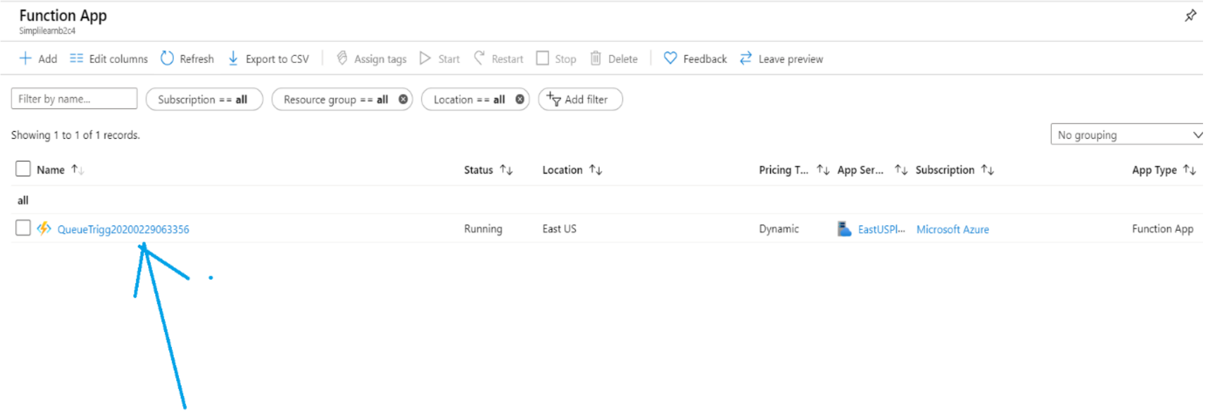
No dependencies currently configured, please click 'Add' to connect to additional services.

Output

Show output from: Build

```
2>Function [Function1]: cannot find value named '' in local.settings.json that matches 'connection' property set on 'queueTrigger'
2>
2>QueueTrigg -> C:\Users\PK\source\repos\QueueTrigg\QueueTrigg\obj\Release\netcoreapp2.1\PubTmp\Out\
2>Publishing C:\Users\PK\source\repos\QueueTrigg\QueueTrigg\obj\Release\netcoreapp2.1\PubTmp\QueueTrigg - 20200229184120999.zip to https://que
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Publish: 1 succeeded, 0 failed, 0 skipped ==========
```

Step 4: Now go to the Azure portal and go to function section , we see that the function **“QueueTrigg”** is created as shown in the window below.



Function App

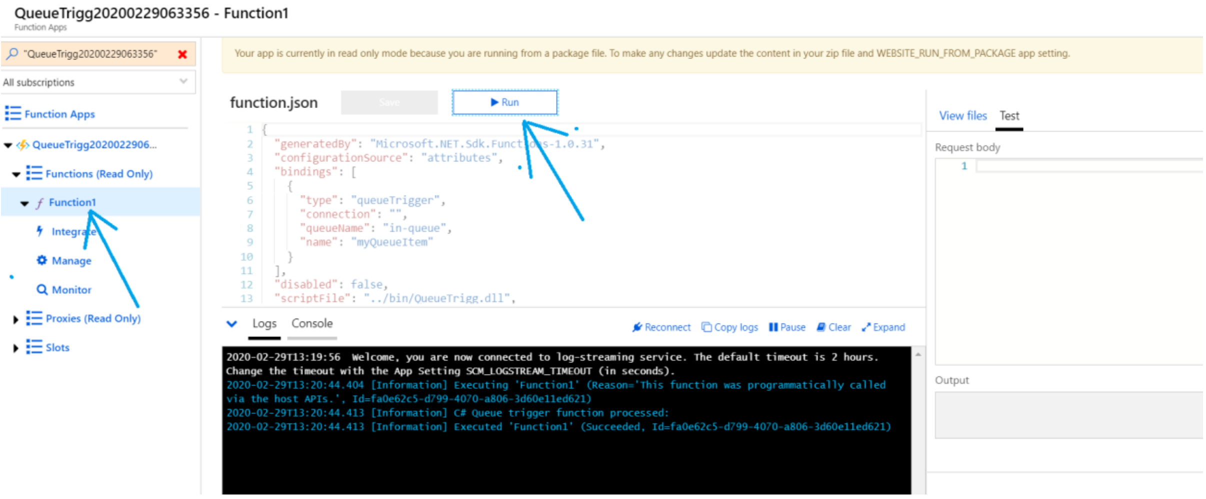
Simplilearnb2c4

Filter by name... Subscription == all Resource group == all Location == all

Showing 1 to 1 of 1 records.

Name	Status	Location	Pricing T...	App Ser...	Subscription	App Type
all						
QueueTrigg20200229063356	Running	East US	Dynamic	EastUSPl...	Microsoft Azure	Function App

Now click on **“QueueTrigg”** , the following window will appear and click on **Function1** and click **“run”.**



QueueTrigg20200229063356 - Function1

Function Apps

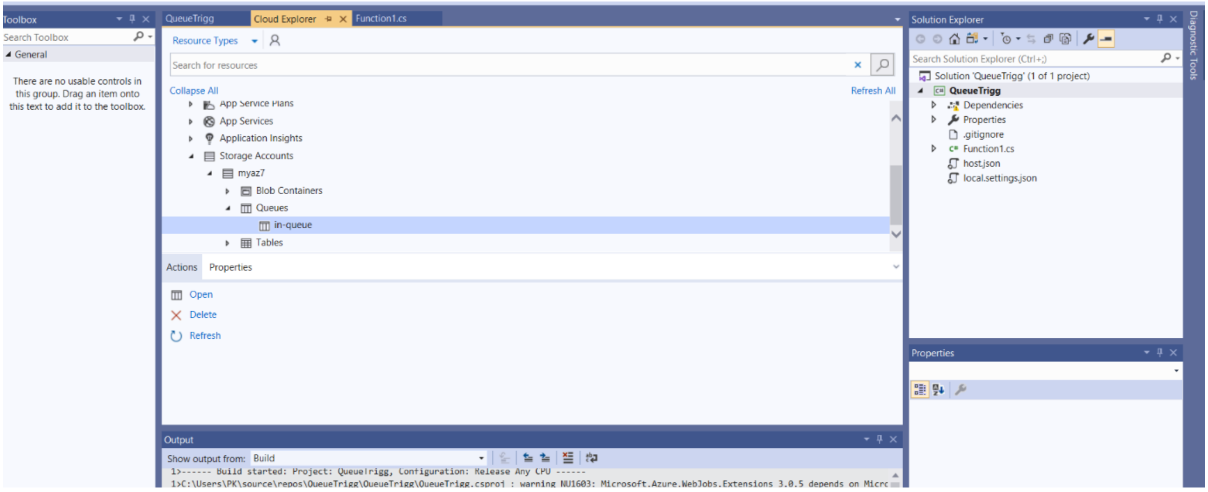
Your app is currently in read only mode because you are running from a package file. To make any changes update the content in your zip file and WEBSITE_RUN_FROM_PACKAGE app setting.

function.json

```
{
  "generatedBy": "Microsoft.NET.Sdk.Functions-1.0.31",
  "configurationSource": "attributes",
  "bindings": [
    {
      "type": "queueTrigger",
      "connection": "",
      "queueName": "in-queue",
      "name": "myQueueItem"
    }
  ],
  "disabled": false,
  "scriptFile": "../bin/QueueTrigg.dll",
```

```
2020-02-29T13:19:56  Welcome, you are now connected to log-streaming service. The default timeout is 2 hours.
Change the timeout with the App Setting SCM_LOGSTREAM_TIMEOUT (in seconds).
2020-02-29T13:20:44.404 [Information] Executing 'Function1' (Reason='This function was programmatically called
via the host APIs.', Id=fa0e62c5-d799-4070-a806-3d60e11ed621)
2020-02-29T13:20:44.413 [Information] C# Queue trigger function processed:
2020-02-29T13:20:44.413 [Information] Executed 'Function1' (Succeeded, Id=fa0e62c5-d799-4070-a806-3d60e11ed621)
```

Step 5: Now go back to visual studio tool, go to view and select **"Cloud explorer"** option. Select the storage account, the following window will appear.

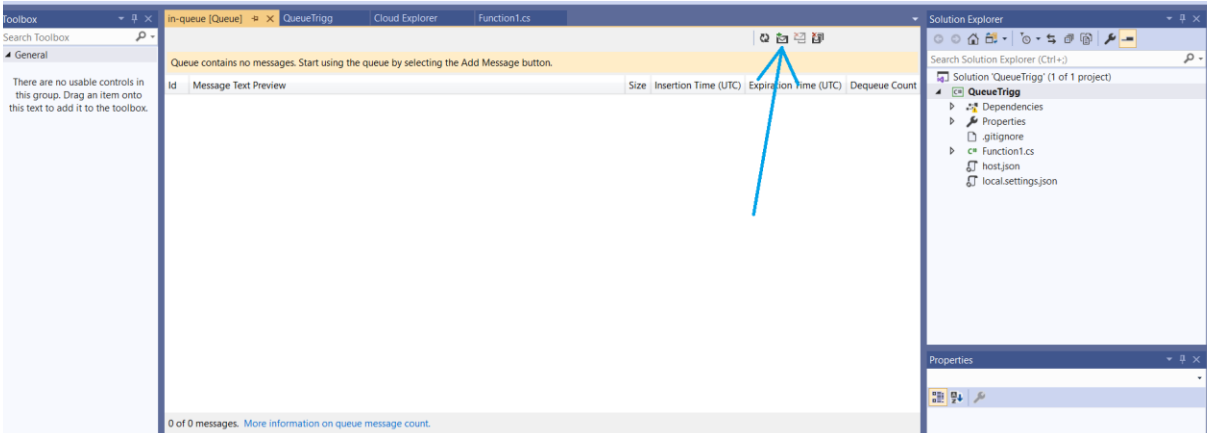


There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Show output from: Build

```
1>------ Build started: Project: QueueTrigg, Configuration: Release Any CPU ------
1>C:\Users\PK\source\repos\QueueTrigg\QueueTrigg\QueueTrigg.csproj : warning NU1603: Microsoft.Azure.WebJobs.Extensions 3.0.5 depends on Micrc
```

Now, double click on **"in-queue"** , the following window will appear and select the message option as shown below.

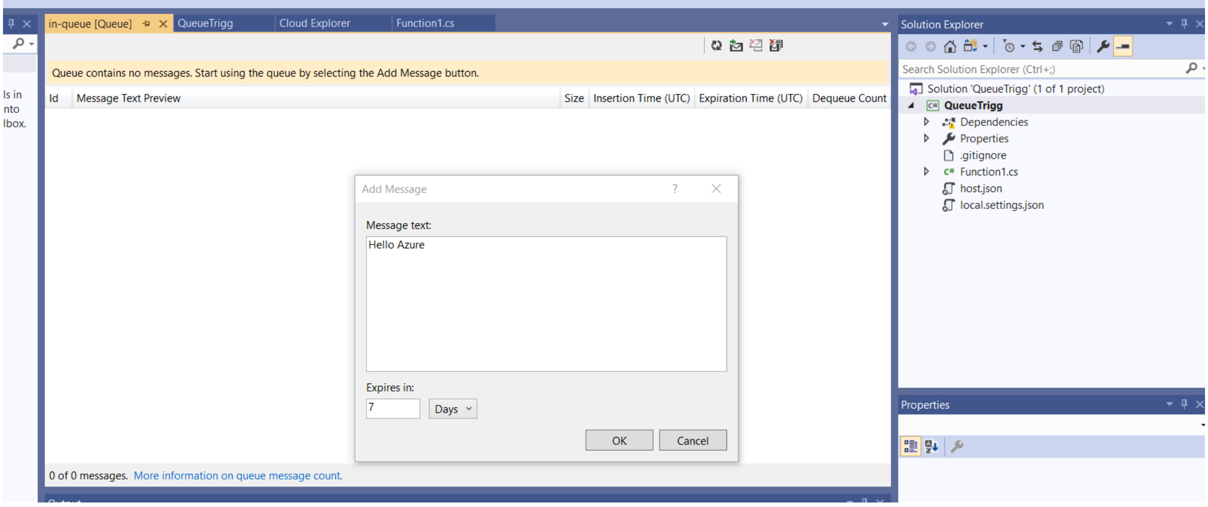


There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Queue contains no messages. Start using the queue by selecting the Add Message button.

0 of 0 messages. More information on queue message count.

Next, add the message in the window as shown below and click "ok".



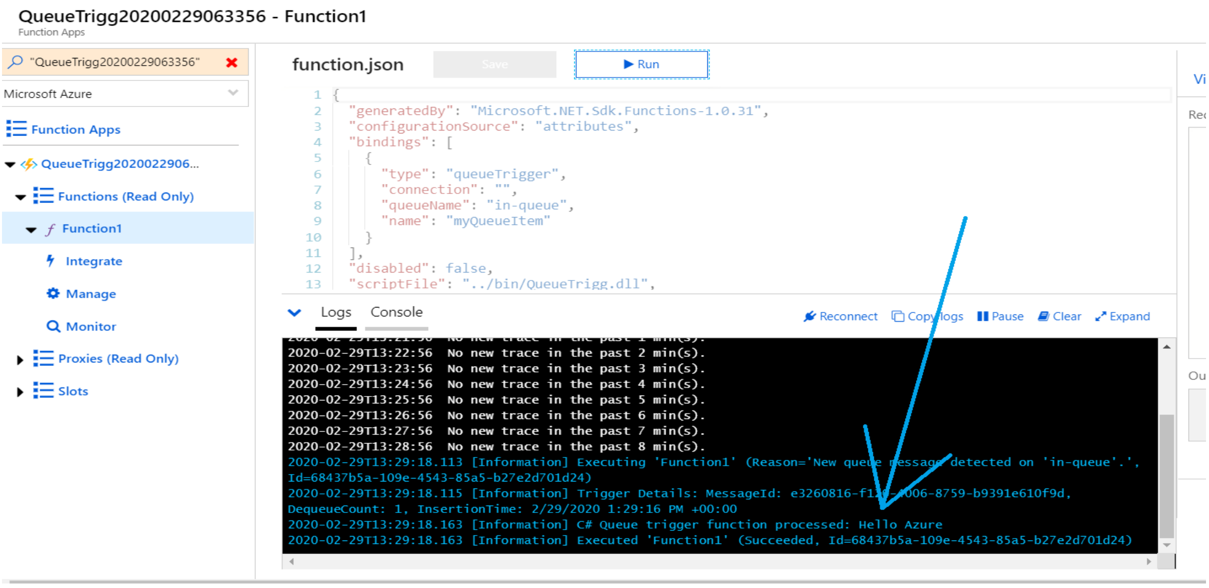
Queue contains no messages. Start using the queue by selecting the Add Message button.

Message text: Hello Azure

Expires in: 7 Days

0 of 0 messages. More information on queue message count.

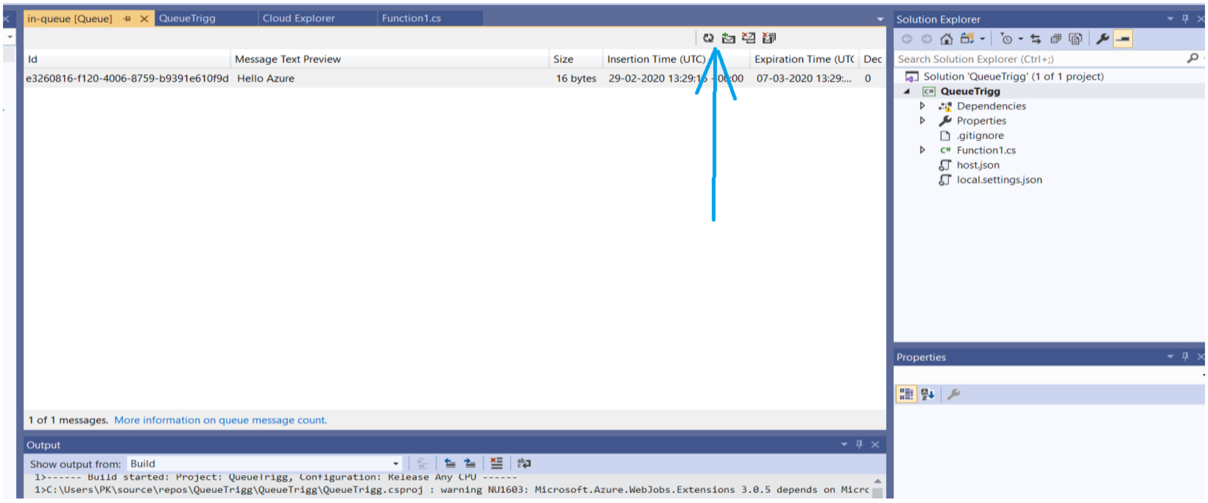
Now go back to azure portal , we can see that the message appears in the log window as shown in the window below.



```
{
  "generatedBy": "Microsoft.NET.Sdk.Functions-1.0.31",
  "configurationSource": "attributes",
  "bindings": [
    {
      "type": "queueTrigger",
      "connection": "",
      "queueName": "in-queue",
      "name": "myQueueItem"
    }
  ],
  "disabled": false,
  "scriptFile": "../bin/QueueTrigg.dll",
```

```
2020-02-29T13:22:56  No new trace in the past 2 min(s).
2020-02-29T13:23:56  No new trace in the past 3 min(s).
2020-02-29T13:24:56  No new trace in the past 4 min(s).
2020-02-29T13:25:56  No new trace in the past 5 min(s).
2020-02-29T13:26:56  No new trace in the past 6 min(s).
2020-02-29T13:27:56  No new trace in the past 7 min(s).
2020-02-29T13:28:56  No new trace in the past 8 min(s).
2020-02-29T13:29:18.113 [Information] Executing 'Function1' (Reason='New queue message detected on 'in-queue'.', Id=68437b5a-109e-4543-85a5-b27e2d701d24)
2020-02-29T13:29:18.115 [Information] Trigger Details: MessageId: e3260816-f120-4006-8759-b9391e610f9d, DequeueCount: 1, InsertionTime: 2/29/2020 1:29:16 PM +00:00
2020-02-29T13:29:18.163 [Information] C# Queue trigger function processed: Hello Azure
2020-02-29T13:29:18.163 [Information] Executed 'Function1' (Succeeded, Id=68437b5a-109e-4543-85a5-b27e2d701d24)
```

Now go back to **"Queue"** tab and refresh the window as shown in the window below.



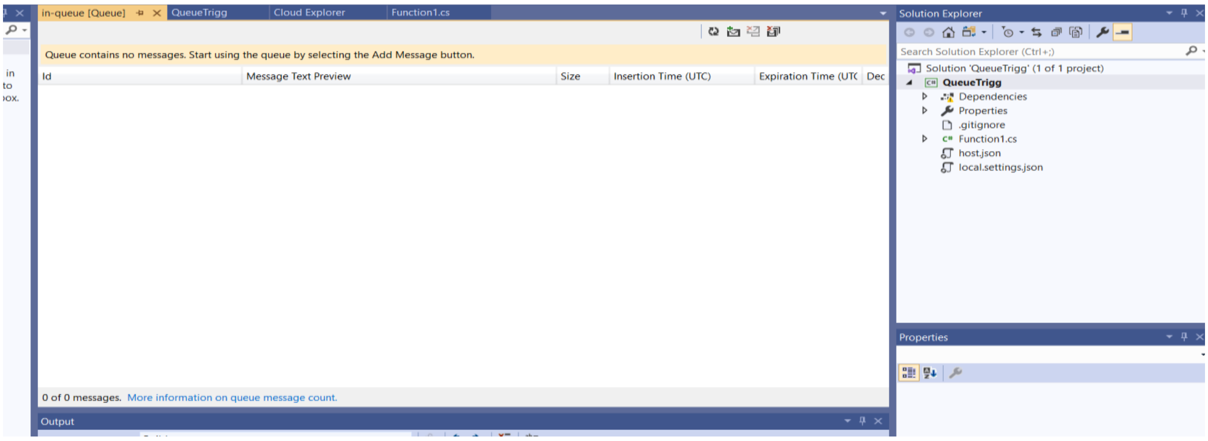
Id	Message Text Preview	Size	Insertion Time (UTC)	Expiration Time (UTC	Dec
e3260816-f120-4006-8759-b9391e610f9d	Hello Azure	16 bytes	29-02-2020 13:29:16 +00:00	07-03-2020 13:29:...	0

1 of 1 messages. More information on queue message count.

Show output from: Build

```
1>------ Build started: Project: QueueTrigg, Configuration: Release Any CPU ------
1>C:\Users\PK\source\repos\QueueTrigg\QueueTrigg\QueueTrigg.csproj : warning NU1603: Microsoft.Azure.WebJobs.Extensions 3.0.5 depends on Micrc
```

When we click on refresh option, the message from the queue **" in-queue"** disappears as shown in the window below.



Queue contains no messages. Start using the queue by selecting the Add Message button.

Id	Message Text Preview	Size	Insertion Time (UTC)	Expiration Time (UTC	Dec

0 of 0 messages. More information on queue message count.

*********************************END************************************