

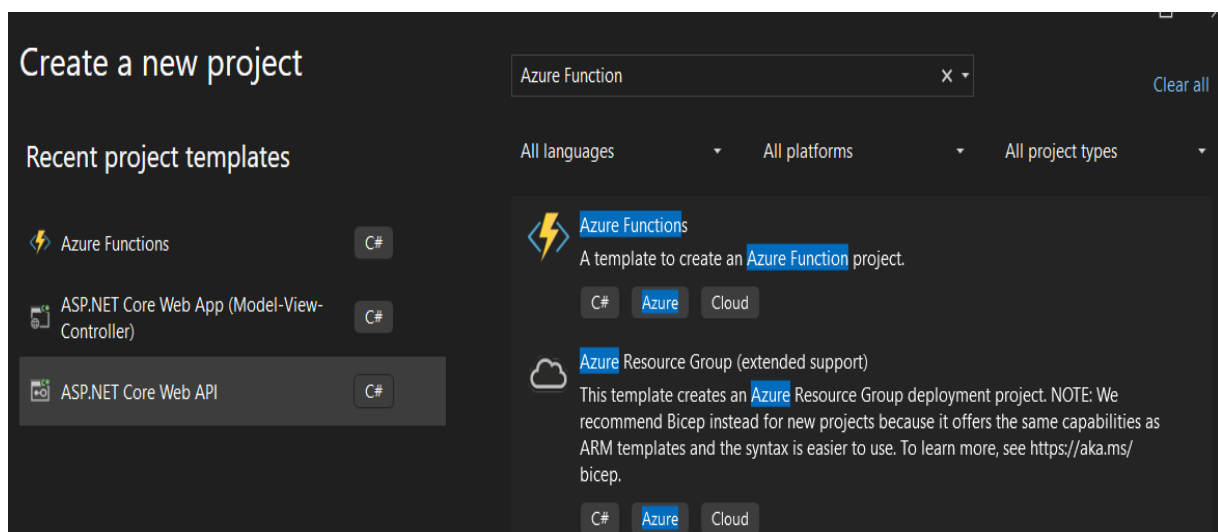
Azure Function Apps: Creation, Deployment, and Durable Functions with .NET

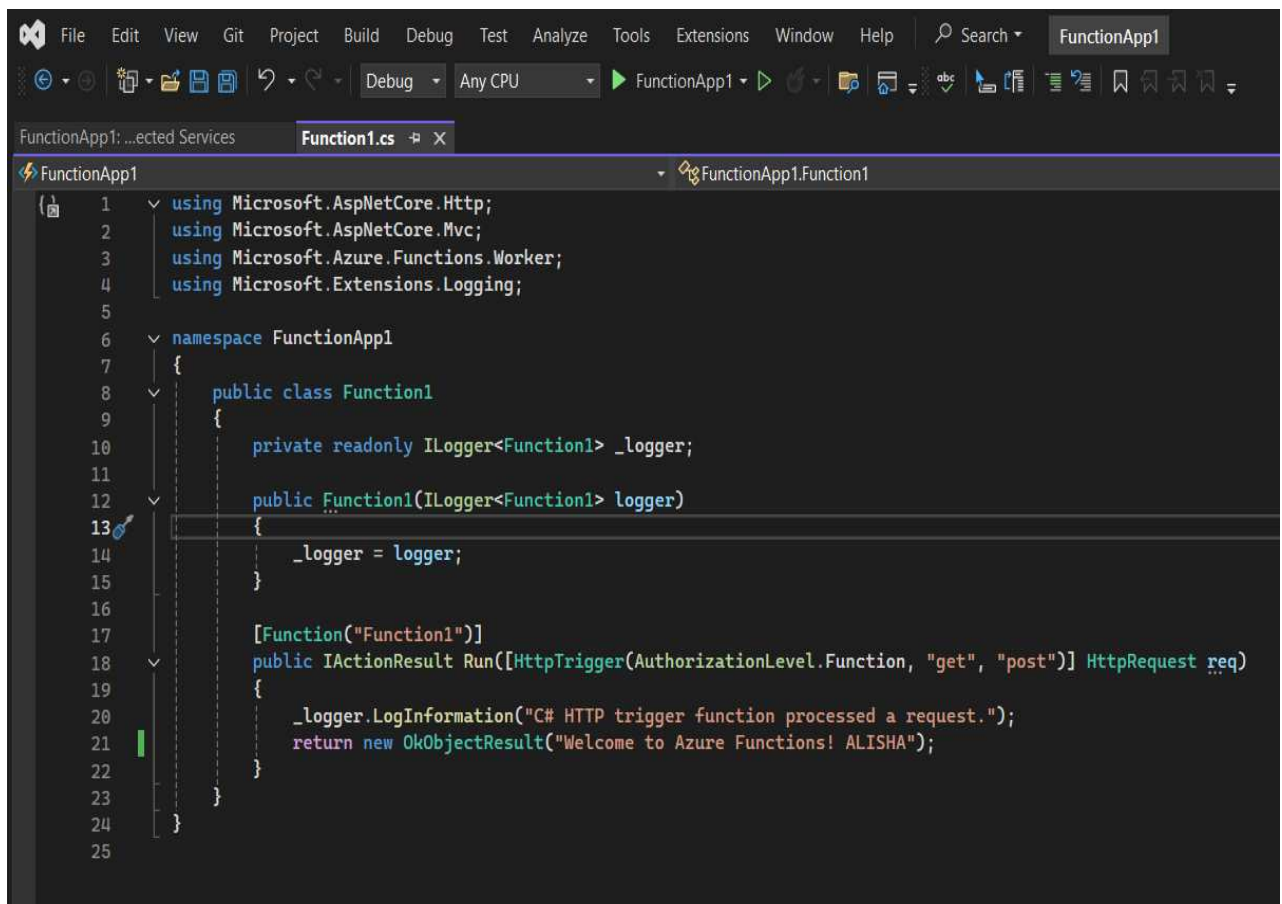
Creating and Deploying an Azure Simple Function App Using .NET and Visual Studio

- Navigate to [Azure Portal](#)
- Search for **Function App**, click **Create** and **Configure**.

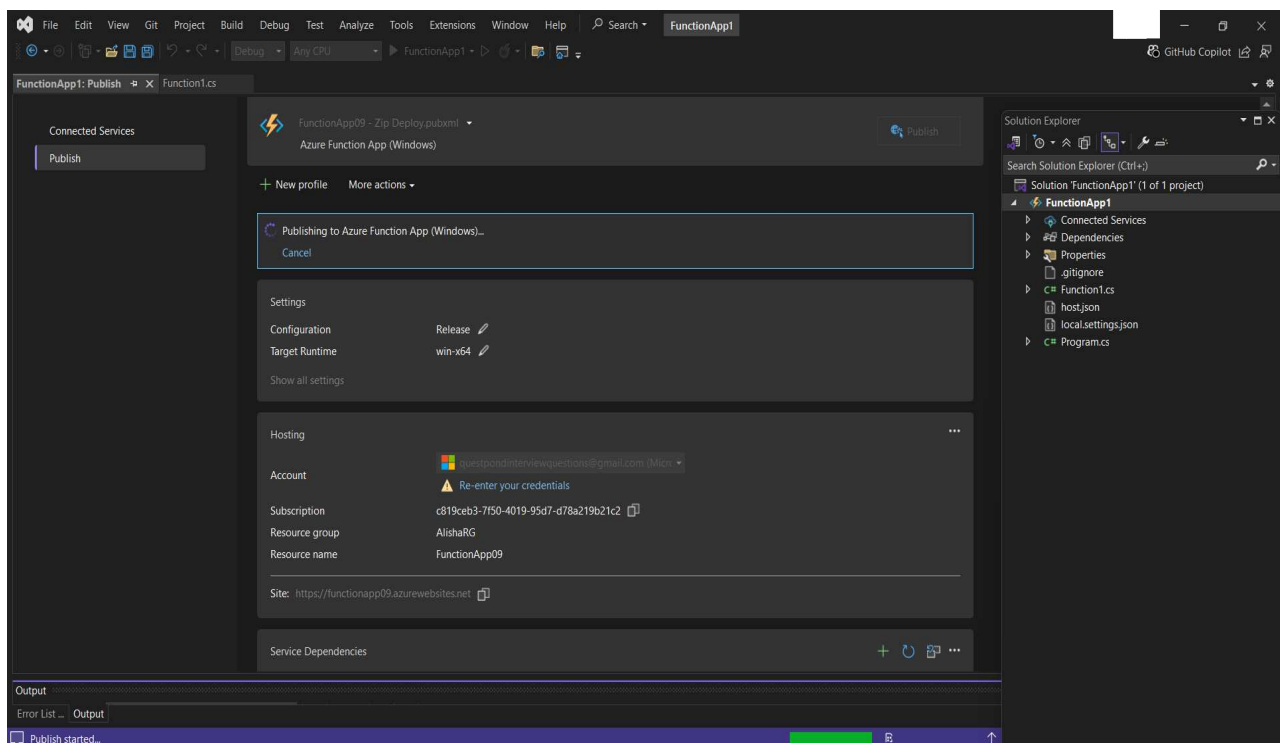
The screenshot shows the 'Create Function App (Consumption)' page in the Azure Portal. The page is titled 'Create Function App (Consumption)' and includes a breadcrumb trail: 'Home > Function App > Create Function App >'. Below the title, there is a section for 'Project Details' with a description: 'Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' The 'Subscription' field is set to 'Pay-As-You-Go (c819ceb3-7f50-4019-95d7-d78a219b21c2)' and the 'Resource Group' is set to '(New) AlishaRG'. There is a 'Create new' link for the resource group. The 'Instance Details' section includes the 'Function App name' field set to 'FunctionApp09', a checkbox for 'Try a secure unique default hostname (preview)' which is checked, and a link to 'More about this update'. The 'Runtime stack' is set to '.NET', the 'Version' is '8 (LTS), isolated worker model', the 'Region' is 'Central India', and the 'Operating System' is 'Windows'. At the bottom, there are three buttons: 'Review + create', '< Previous', and 'Next : Storage >'. The 'Review + create' button is highlighted in blue.

- **Open Visual Studio** and select **Create a new project**
- Choose **Azure Functions** as the project template and click **Next**





- Right-click the project in **Solution Explorer** and Click **Publish**



FunctionApp09
Function App

Search resources, services, and docs (G+)

Overview

Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Events (preview)
Recommended services (preview)
Functions
App keys
App files
Proxies
Deployment
Settings
Performance
App Service plan
Development Tools

Essentials

Resource group (move) : AlishaRG
Status : Running
Location (move) : Central India
Subscription (move) : Pay-As-You-Go
Subscription ID : c819ceb3-7f50-4019-95d7-d78a219b21c2
Tags (edit) : Add tags

Default domain : functionapp09.azurewebsites.net
Operating System : Windows
App Service Plan : ASP-AlishaRG-8209 (Y1:0)
Runtime version : 4.1037.1.23605

Functions Metrics Properties Notifications (0)

{ } Set up local environment Refresh

Filter by name...

Name	Trigger	Status	Monitor
Function1	HTTP	Enabled	Invocations and more

JSON View

https://portal.azure.com/#blade/WebsitesExtension/FunctionTabMenuBlade/resourceId/%2Fsubscriptions%2Fc819ceb3-7f50-4019-95d7-d78a219b21c2%2Fresourcegroups%2FAlishaRG%2Fproviders%2FMicrosoft.Web%2FSites%2FFunctionApp09%2FFunctions%2FFunction1

- Navigate to **Function App –(Function1)**

Function1 | Code + Test
FunctionApp09

Code + Test Integration Function Keys Invocations Logs Metrics

Save Discard Refresh Test/Run Get function URL Disable Delete Upload Resource JSON Send us your feedback

Editing .NET isolated Function Apps is not supported in the Azure portal. Use your local development environment to edit this Function App.

FunctionApp09 / host.json

```

1 {
2   "version": "2.0",
3   "logging": {
4     "applicationInsights": {
5       "samplingSettings": {
6         "isEnabled": true,
7         "excludedTypes": "Request"
8       },
9       "enableLiveMetricsFilters": true
10    }
11  }
12 }
```

Logs

App Insights Logs Log

Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this function, click on the Log icon in the top right corner of the Code + Test panel.

Test/Run

Input Output

Provide parameters to test the HTTP request. Results can be found in the Output tab.

HTTP method * POST

Key * _master (Host key)

Query parameters

Name	Value

Headers

Name	Value

Body

1

Run Close

- Click on Run to get Output:

Microsoft Azure

Home > FunctionApp09 >

Function1 | Code + Test

FunctionApp09

Code + Test Integration Function Keys Invocations Logs Metrics

Save Discard Refresh Test/Run Get function URL Disable Delete Upload Resource JSON Send us your feedback

FunctionApp09 / host.json

```

1 {
2   "version": "2.0",
3   "logging": {
4     "applicationInsights": {
5       "samplingSettings": {
6         "isEnabled": true,
7         "excludedTypes": "Request"
8       },
9       "enableLiveMetricsFilters": true
10    }
11  }
12 }

```

Logs

App Insights Logs

Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, click on the Logs tab in the left-hand navigation pane.

2025-03-06T18:09:29Z [Information] Executing 'Functions.Function1' (Reason='This function was programmatically called via...')

2025-03-06T18:09:29Z [Information] C# HTTP trigger function processed a request.

2025-03-06T18:09:29Z [Information] Executing OkObjectResult, writing value of type 'System.String'.

2025-03-06T18:09:29Z [Information] Executed 'Functions.Function1' (Succeeded, Id=95e68017-dd1d-4125-9cdb-b63374e5e2b4, Duration=400ms)

Run Close

- Use Azure Monitor to track execution and failures :

Microsoft Azure

Home > FunctionApp09

FunctionApp09 | Logs

Function App

Search

New Query 1*

FunctionApp09 Select scope

Run Time range: Last 24 hours Save Share New alert rule Export Pin to

Tables Queries Functions

Search

Filter Group by: Solution

Collapse all

- exceptions
- pageViews
- performanceCounters
- requests
- traces
- timestamp (datetime)
- message (string)
- severityLevel (int)
- itemType (string)
- customDimensions (dynamic)
- customMeasurements (dynamic)
- operation_Name (string)
- operation_Id (string)

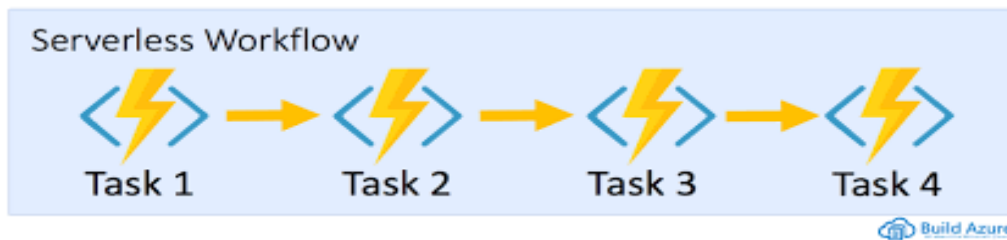
Results Chart

timestamp (UTC)	message
> 3/6/2025, 6:09:28.852 PM	Executed 'Functions.Function1' (Succeeded, Id=95e68017-dd1d-4125-9cdb-b63374e5e2b4, Duration=400ms)
> 3/6/2025, 6:09:28.768 PM	Executing OkObjectResult, writing value of type 'System.String'.
> 3/6/2025, 6:09:28.760 PM	C# HTTP trigger function processed a request.
> 3/6/2025, 6:09:28.490 PM	Executing 'Functions.Function1' (Reason='This function was programmatically called via the host APIs.', Id=95e68017-dd1d-4125-9cdb-b63374e5e2b4, Duration=400ms)
> 3/6/2025, 6:09:08.652 PM	Host Status: { "id": "functionapp09", "state": "Running", "version": "4.1037.1.23605", "versionDetails": "4.1037.1+8ac2126191b938" }
> 3/6/2025, 6:09:08.043 PM	ConcurrencyOptions { "DynamicConcurrencyEnabled": false, "MaximumFunctionConcurrency": 500, "CPUThreshold": 0.8, "SnapshotsEnabled": false }
> 3/6/2025, 6:09:08.023 PM	HttpOptions { "DynamicThrottlesEnabled": true, "EnableChunkedRequestBinding": false, "MaxConcurrentRequests": 100, "MaxOutstandingRequests": 100 }
> 3/6/2025, 6:08:43.037 PM	Host lock lease acquired by instance ID '02a376231b7f4d695b6ff9c87b62ae1b'.

8s 237ms Display time (UTC+00:00) Query details 1 - 8 of 288

Durable Function Apps in Azure

Durable Functions extend Azure Functions by enabling **stateful workflows**.

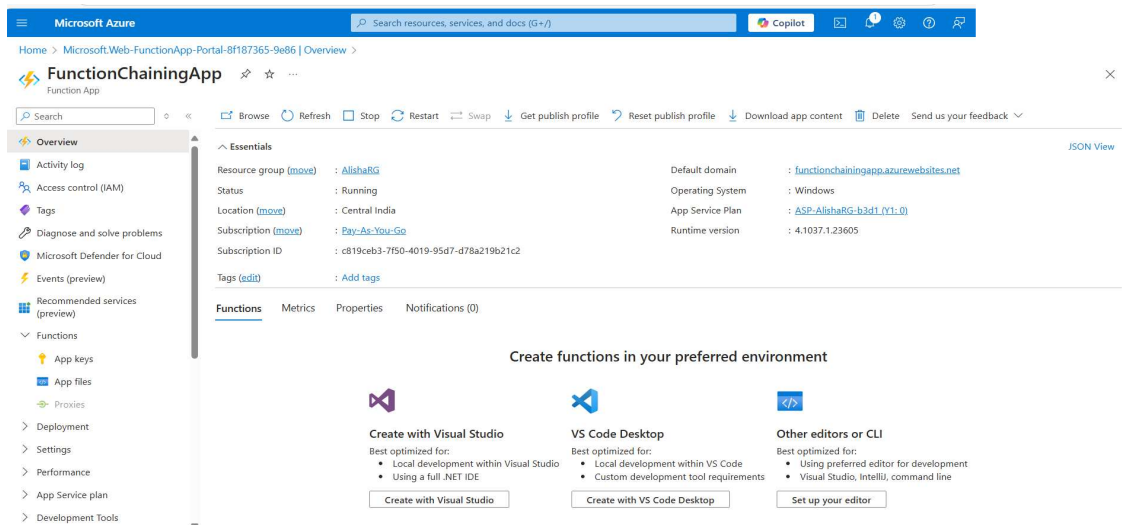


Steps to Implement:

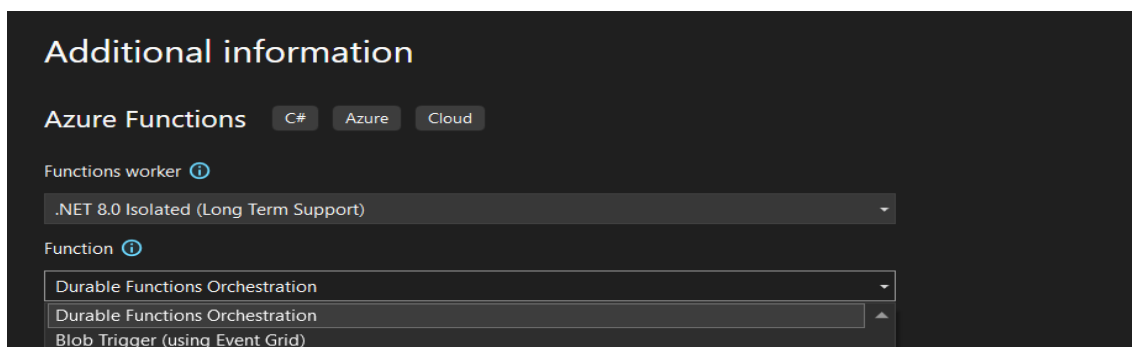
- ✓ **Orchestrator Function** → Executes multiple tasks in **given sequence** and **maintain state** of functions.
- ✓ **Activity Function** → Performs the actual function processing **independently**.
- ✓ **Client Function (Http)** → Triggers the **Orchestrator Function**.

Durable Function Patterns

1. **Function Chaining(Sequential Execution)** : Create a Function app in azure portal:



- Create a Durable orchestrating Function in Visual Studio using .Net runtime:

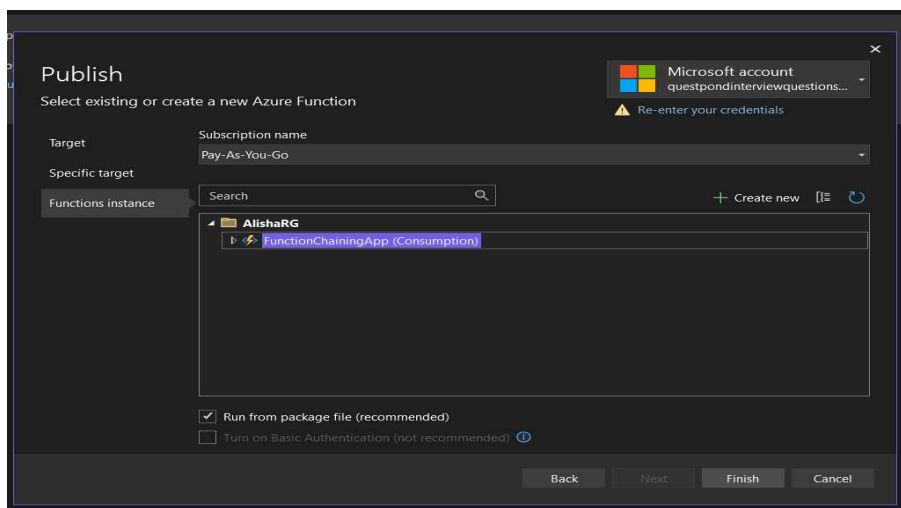



```

5  using Microsoft.Extensions.Logging;
6
7  namespace FunctionChainingApp
8  {
9      public static class Function1
10     {
11         //Orchestration Function
12         [Function(nameof(Function1))]
13         public static async Task<List<string>> RunOrchestrator(
14             [OrchestrationTrigger] TaskOrchestrationContext context)
15         {
16             ILogger logger = context.CreateReplaySafeLogger(nameof(Function1));
17             logger.LogInformation("Saying hello.");
18             var outputs = new List<string>();
19
20             // Durable Functions Activity
21             outputs.Add(await context.CallActivityAsync<string>(nameof(SayHello), "Tokyo"));
22             outputs.Add(await context.CallActivityAsync<string>(nameof(SayHello), "Seattle"));
23             outputs.Add(await context.CallActivityAsync<string>(nameof(SayHello), "London"));
24
25             // returns ["Hello Tokyo!", "Hello Seattle!", "Hello London!"]
26             return outputs;
27         }
28
29         //Activity Function - "SayHello"
30         [Function(nameof(SayHello))]
31         public static string SayHello([ActivityTrigger] string name, FunctionContext executionContext)
32         {
33             ILogger logger = executionContext.GetLogger("SayHello");
34             logger.LogInformation("Saying hello to {name}.", name);
35             return $"Hello {name}!";
36         }
37
38         // starts a new Durable Orchestration instance
39         [Function("Function1_HttpStart")]
40         public static async Task<HttpResponseData> HttpStart(
41             [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post")] HttpRequestData req,
42             [DurableClient] DurableTaskClient client,
43             FunctionContext executionContext)
44         {
45             ILogger logger = executionContext.GetLogger("Function1_HttpStart");
46
47             // Function input comes from the request content.
48             string instanceId = await client.ScheduleNewOrchestrationInstanceAsync(
49                 nameof(Function1));
50
51             logger.LogInformation("Started orchestration with ID = '{instanceId}'.", instanceId);
52
53             // Returns an HTTP 202 response with an instance management payload.
54             // See https://learn.microsoft.com/azure/azure-functions/durable/durable-functions-http-api#start-orchestration
55             return await client.CreateCheckStatusResponseAsync(req, instanceId);
56         }
57     }
58 }

```

- Publish selected Function app on azure :

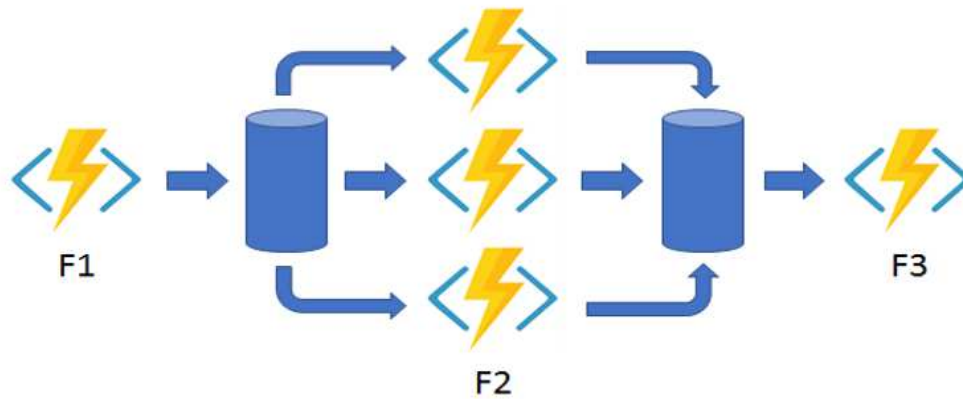


- Navigate to Function1_HttpStart and Test&Run the code:

- Run the URL (StatusQueryGetUri) to get output:

2. Fan-out / Fan-in(Parallel Execution):

- A function triggers multiple parallel executions (fan-out), and later collects their results (fan-in).



FanOut

FanIn

- Create Function App In Azure :

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons. The main content area displays the overview of a Function App named 'FanInFanOutFunctionApp'. The left sidebar contains a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Recommended services (preview), Functions, Deployment, Settings, Performance, App Service plan, Development Tools, API, Monitoring, and Automation. The main content area shows the 'Essentials' section with details about the Function App, including Resource group, Status, Location, Subscription, and Subscription ID. Below this, there are tabs for Functions, Metrics, Properties, and Notifications (0). The 'Functions' tab is selected, and it displays a section titled 'Create functions in your preferred environment' with three options: 'Create with Visual Studio', 'VS Code Desktop', and 'Other editors or CLI'. Each option includes a list of best practices and a button to create or set up the environment.

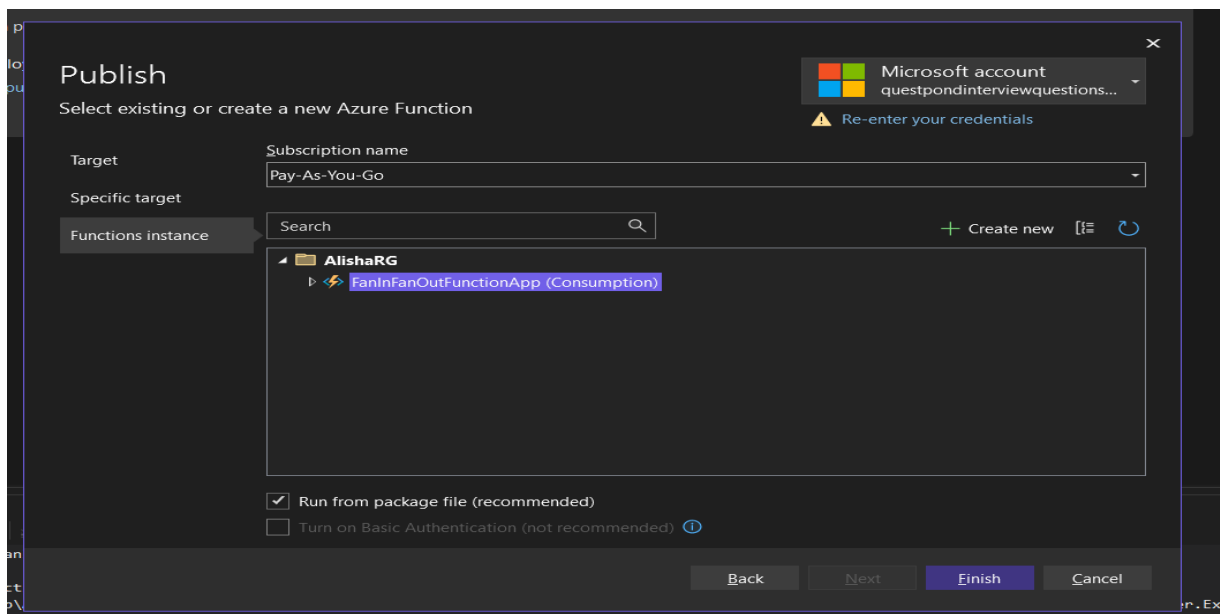
- Create Durable orchestrating Function App in .Net & Publish on Azure:

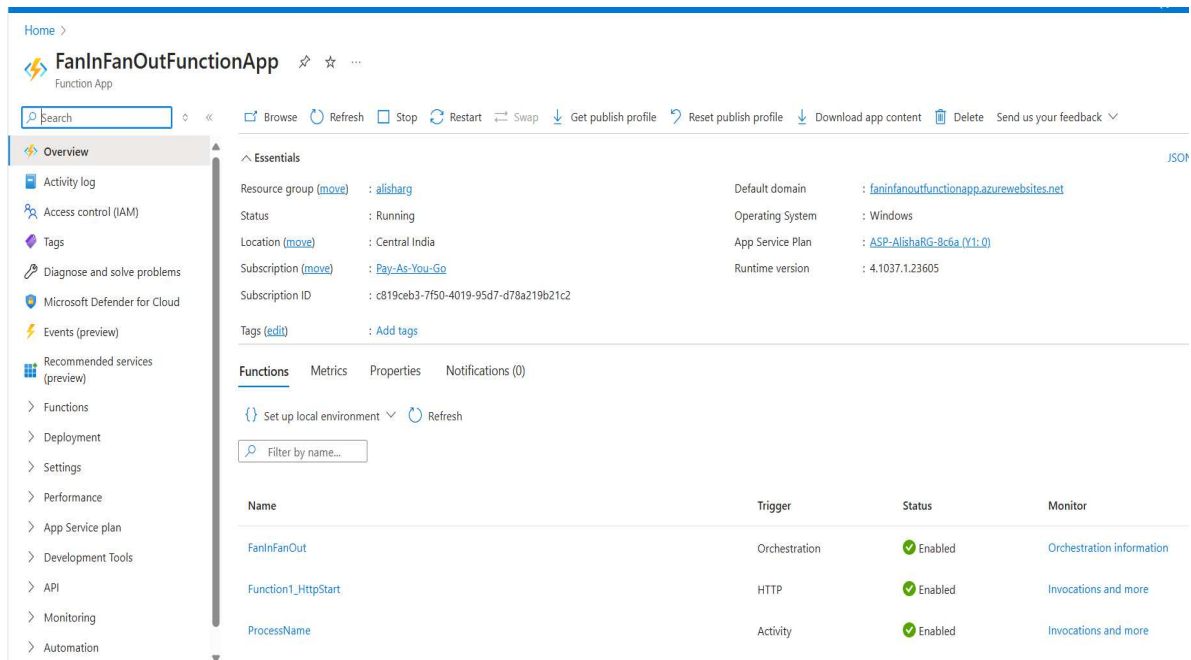
```

Function1.cs
FunctionAppFanInFanOut
FunctionAppFanInFanOut.FanInFanOut

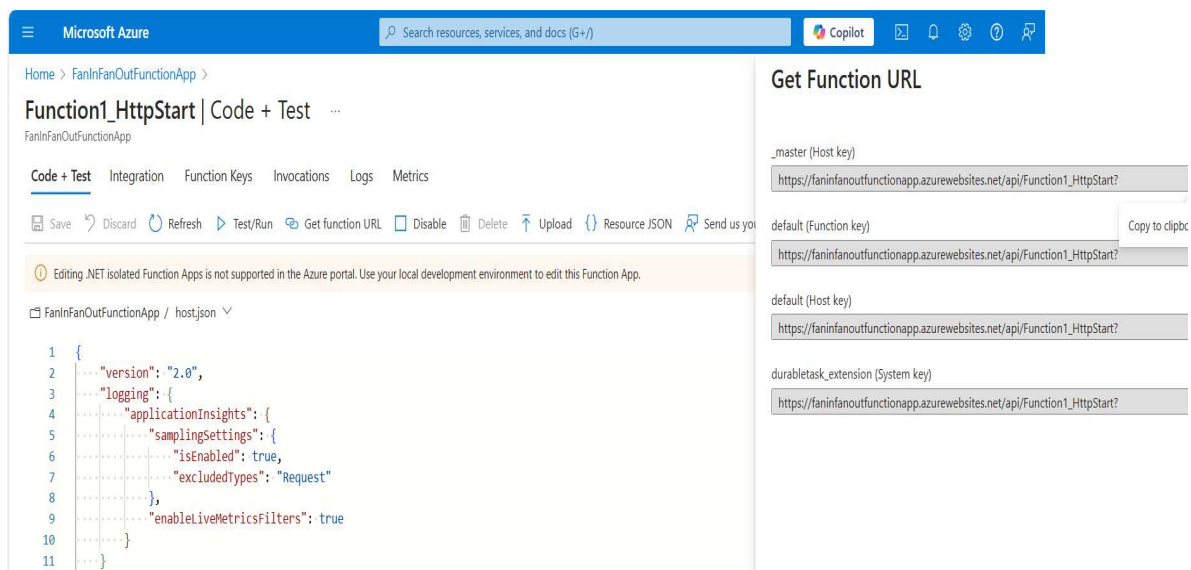
10 {
11     public static class FanInFanOut
12     {
13         /**Orchestrator Function**
14         [Function(nameof(FanInFanOut))]
15         public static async Task<List<string>> RunOrchestrator(
16             [OrchestrationTrigger] TaskOrchestrationContext context)
17         {
18             var names = context.GetInput<List<string>>();
19             var tasks = new List<Task<string>>();
20
21             // Fan-Out: Start multiple activity functions in parallel
22             foreach (var name in names)
23             {
24                 tasks.Add(context.CallActivityAsync<string>("ProcessName", name));
25             }
26
27             // Fan-In: Wait for all activity functions to complete
28             var results = await Task.WhenAll(tasks); // Wait for all name processing
29             return results.ToList();
30         }
31
32         /**Activity Function**
33         [Function(nameof(ProcessName))]
34         public static string ProcessName([ActivityTrigger] string name) => "Hello, " + name + "!";
35
36
37         /**Client Function (HTTP Trigger)**
38         [Function("Function1_HttpStart")]
39         public static async Task<HttpResponseBody> HttpStart(
40             [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post")] HttpRequestData req,
41             [DurableClient] DurableTaskClient client,
42             FunctionContext executionContext)
43         {
44             var names = new List<string> { "John", "Alice", "Bob" };
45
46             // Use ScheduleNewOrchestrationInstanceAsync for .NET Isolated Worker
47             string instanceId = await client.ScheduleNewOrchestrationInstanceAsync(
48                 nameof(FanInFanOut), names);
49
50             // Create response
51             var response = req.CreateResponse(HttpStatusCode.OK);
52             await response.WriteStringAsync($"Started. Instance ID: {instanceId}");
53
54             return response;
55         }
56     }

```





- Navigate to Function1_HttpStart(Client Function):



Test and Run Instructions:

1. Copy the **Master Host Key** and execute the function. This will return a unique **Instance ID** (e.g., 3bf40905f7154865aaa64cbbad4ee8c7).
2. Copy the **Host Function Key** (e.g., gO_FP8_8PJ7qyMMTWJa3DcHQUGdgqVC3AvEC-R-5n9z2AzFu5huhoQ==).
3. Replace the **Instance ID** and **Function Key** in the following URL and open it in a browser to get the output:

https://faninfanoutfunctionapp.azurewebsites.net/runtime/webhooks/durabletask/instances/3bf40905f7154865aaa64cbbad4ee8c7?code=gO_FP8_8PJ7qyMMTWJa3DcHQUGdgqVC3AvEC-R-5n9z2AzFu5huhoQ==

```
Get Function URL - Microsoft Azu x faninfanoutfunctionapp.azureweb x faninfanoutfunctionapp.azureweb x + - □ X
https://faninfanoutfunctionapp.azurewebsites.net/runtime/webhooks/durabletask/ins...
pretty-print □

{"name":"FanInFanOut","instanceId":"3bf40905f7154865aaa64cbbad4ee8c7","runtimeStatus":"Completed","input":
["John","Alice","Bob"],"customStatus":null,"output":["Hello, John!","Hello, Alice!","Hello, Bob!"],"createdTime":"2025-03-
8T17:50:45Z","lastUpdatedTime":"2025-03-8T17:50:45Z"}
```

- Use **Monitor** to track execution and failures:

