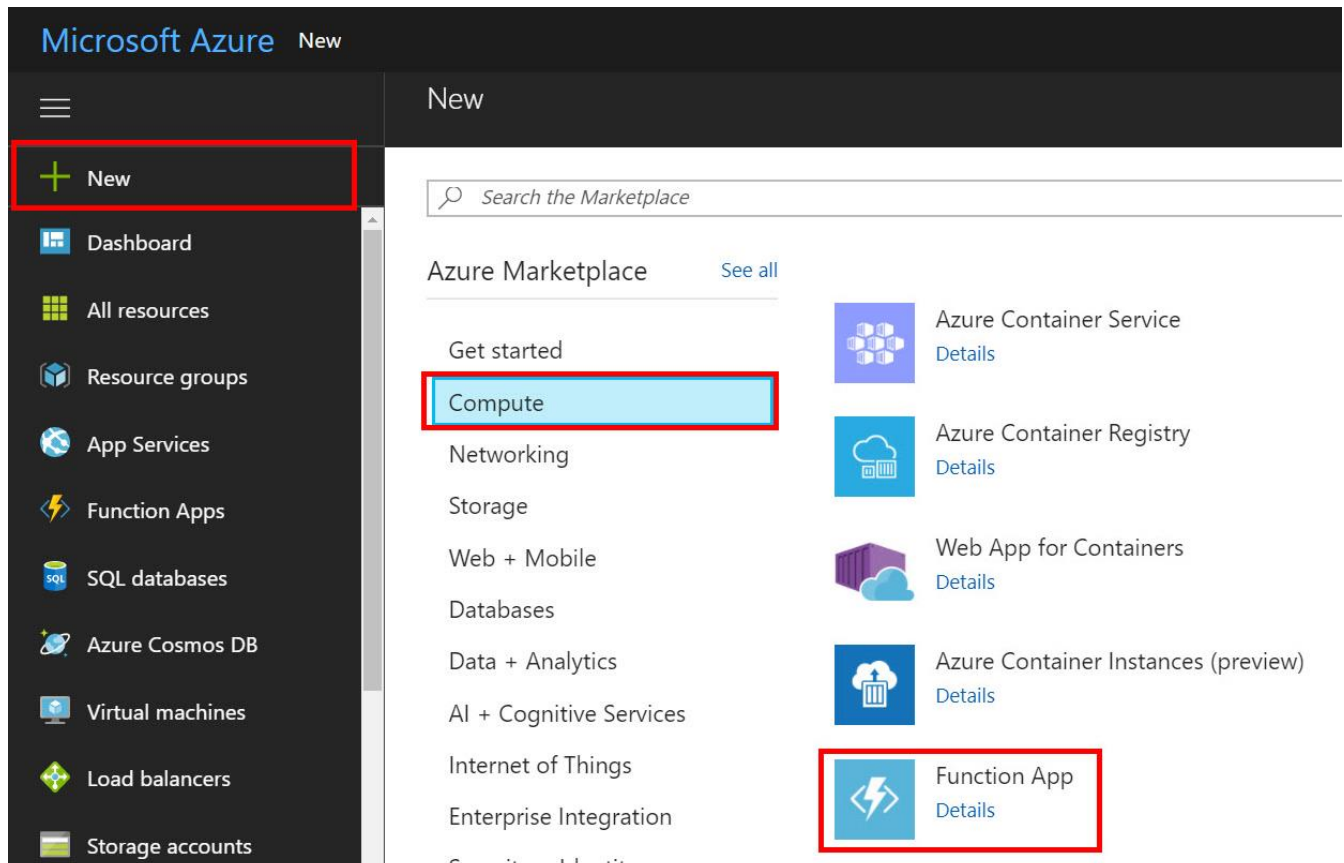


Azure Function App – TimerTrigger

Step 1: Open Microsoft Azure Portal & Login with Azure subscription credentials

<https://portal.azure.com>

Step 2: Click on +New -> Compute -> Function App



Step 3: Enter Function App Details

Function App

Create

* App name

azurefunappdemo

.azurewebsites.net

* Subscription

MSDN Platforms

* Resource Group ⓘ

☒ Create new

☐ Use existing

FUNAPPRG

* Hosting Plan ⓘ

Consumption Plan

* Location

Southeast Asia

* Storage ⓘ

☒ Create New

☐ Select Existing

azurefunappdemoa639

Application Insights ⓘ

On

Off

☒ Pin to dashboard

Create

Automation options

App Name: azurefunappdemo

Subscription: Choose any of the Subscription

Resource Group: Create New resource.

Ex. FUNAPPRG

Hosting Plan: Consumption Plan

Location: Choose nearest location

Storage: Create New Storage Account

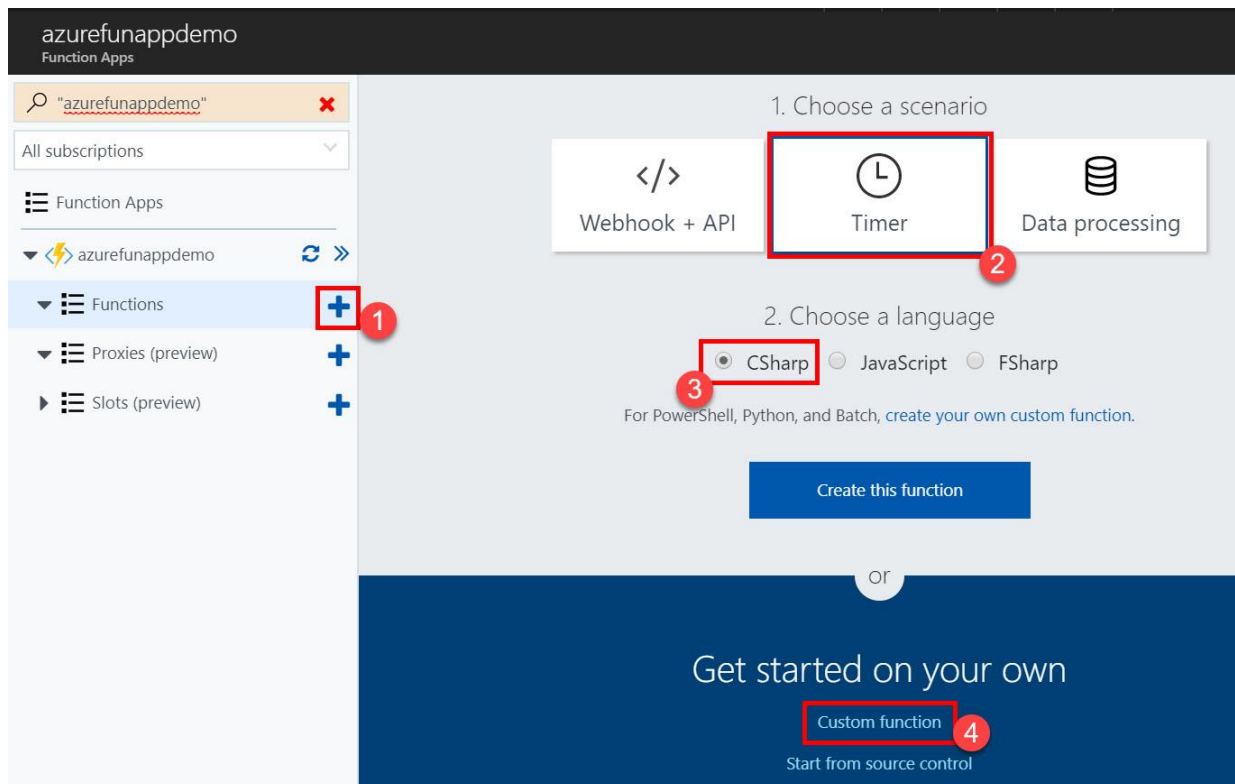
Step 4: Wait for couple of seconds to create function app

Click on **Add new function** button

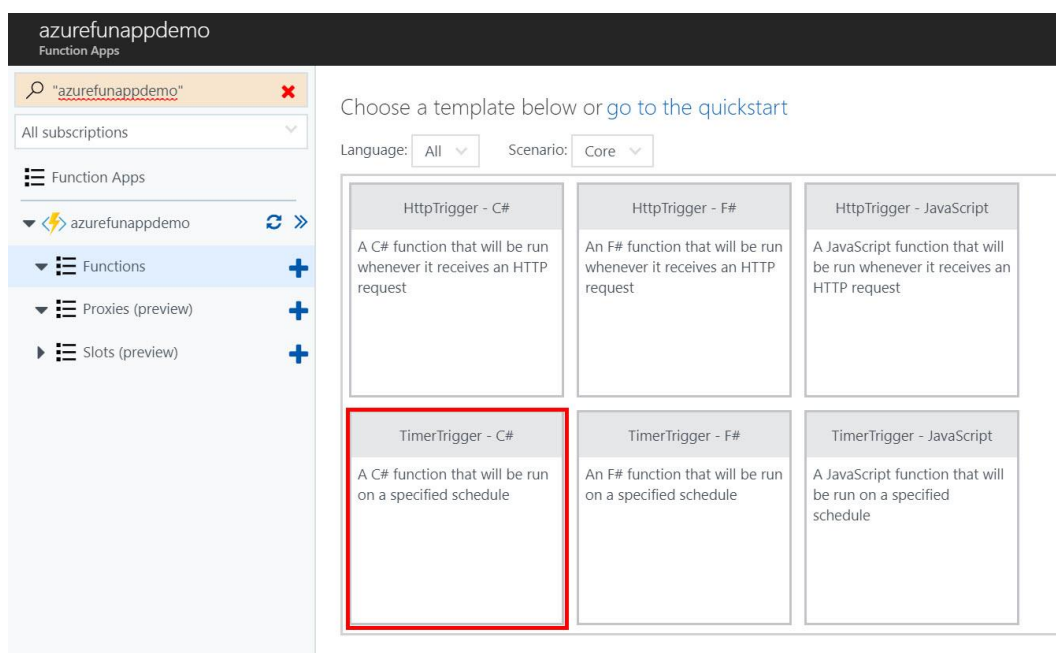
Choose a Scenario: Timer

Choose a language: CSharp

Select **Custom function** option



Select **TimerTrigger – C#** Template



Step 5: Also Change the Schedule option from `0 */5 * * * *` to `*/5 * * * * *` & Click on Create button

azurefunappdemo
Function Apps

Search "azurefunappdemo" ✕

All subscriptions ▾

Function Apps

▼ ⚡ azurefunappdemo ↻ »

▼ ⚡ Functions +

▼ ⚡ Proxies (preview) +

▶ ⚡ Slots (preview) +

A C# function that will be run on a specified schedule

An F# function that will be run on a specified schedule

A JavaScript function that will be run on a specified schedule

Name your function

TimerTriggerCSharp1

Enter a cron expression of the format '{second} {minute} {hour} {day} {month} {day of week}' to specify the schedule. See documentation below for examples.

Schedule ⓘ

*/5 * * * * *

Create

Step 6: Now click on **Run** button

azurefunappdemo - TimerTriggerCSharp1
Function Apps

Search "azurefunappdemo" ✕

All subscriptions ▾

Function Apps

▼ ⚡ azurefunappdemo ↻ »

▼ ⚡ Functions +

▼ ⚡ TimerTriggerCSharp1

⚡ Integrate

⚙ Manage

🔍 Monitor

▼ ⚡ Proxies (preview) +

▶ ⚡ Slots (preview) +

run.csx Save Run

```
1 using System;
2
3 public static void Run(TimerInfo myTimer, TraceWriter
4 {
5     log.Info($"C# Timer trigger function executed at
6 }
7
```

View files Test >

+ Add ↑ Upload 🗑 Delete

TimerTriggerCSharp1

function.json

run.csx

This function app will **run every 5 seconds** because in schedule we have configured that.

azurefunappdemo - TimerTriggerCSharp1
Function Apps

"azurefunappdemo" ✖

All subscriptions

Function Apps

azurefunappdemo

Functions

TimerTriggerCSharp1

Integrate

Manage

Monitor

Proxies (preview)

Slots (preview)

run.csx

Save

Run

```
1 using System;
2
3 public static void Run(TimerInfo myTimer, TraceWriter log)
4 {
5     log.Info($"C# Timer trigger function executed at: {DateTime.Now}");
6 }
7
```

Logs

Pause Clear Copy logs Expand

2017-09-20T05:04:45.012 Function started (Id=465e67ba-a922-49e5-b048-7e3574698a85)

2017-09-20T05:04:45.012 C# Timer trigger function executed at: 9/20/2017 5:04:45 AM

2017-09-20T05:04:45.012 Function completed (Success, Id=465e67ba-a922-49e5-b048-7e3574698a85, Du

2017-09-20T05:04:50.012 Function started (Id=f90bb33d-937a-4514-be61-ea629d441e30)

2017-09-20T05:04:50.012 C# Timer trigger function executed at: 9/20/2017 5:04:50 AM

2017-09-20T05:04:50.012 Function completed (Success, Id=f90bb33d-937a-4514-be61-ea629d441e30, Du

Step 7: Click on **Integrate** option.

This option will provide triggers, inputs, outputs, timestamp parameter name, schedule.

azurefunappdemo - TimerTriggerCSharp1
Function Apps

"azurefunappdemo" ✖

All subscriptions

Function Apps

azurefunappdemo

Functions

TimerTriggerCSharp1

Integrate

Manage

Monitor

Proxies (preview)

Slots (preview)

Triggers

Timer (myTimer)

Inputs

+ New Input

Outputs

+ New Output

Advanced editor

Timer trigger ✖ delete

Timestamp parameter name

myTimer

Schedule

*/*5 * * * * *

+ Documentation

Step 8: Click on **Manage** option

This option will provide function state, Host Keys, etc

The screenshot shows the Azure Functions portal for 'azurefunappdemo - TimerTriggerCSharp1'. The left sidebar contains a search bar, a dropdown for 'All subscriptions', and a list of function apps. Under 'azurefunappdemo', the 'Functions' section is expanded, showing 'TimerTriggerCSharp1'. The 'Manage' option is highlighted with a red box. The main area displays the 'Function State' as 'Enabled', a 'Delete function' button, and a table of 'Host Keys (All functions)'. The table has columns for NAME, VALUE, and ACTIONS. The first row is '_master' with a 'Click to show' value and 'Copy' and 'Renew' actions. The second row is 'default' with a 'Click to show' value and 'Copy', 'Renew', and 'Revo' actions. An 'Add new host key' button is at the bottom.

NAME	VALUE	ACTIONS
_master	Click to show	Copy Renew
default	Click to show	Copy Renew Revo

Step 9: Click on **Monitor** option

This option will provide monitoring details

The screenshot shows the Azure Functions portal for 'azurefunappdemo - TimerTriggerCSharp1'. The left sidebar is the same as in Step 8, but the 'Monitor' option is highlighted with a red box. The main area displays monitoring details. At the top, a yellow banner suggests enabling Application Insights. Below this, two boxes show 'Success count since Sep 1st' as 14 and 'Error count since Sep 1st' as 0. The 'Invocation log' section shows a table of function invocations with columns for Function, Status, Details: Last ran, and (duration). The table shows five successful invocations. The 'Invocation details' section shows a 'Parameter' field and a 'Logs' section.

Function	Status	Details: Last ran	(duration)
TimerTriggerCSharp1 (2017-09-20T05:06:5 ...)	✓	a few seconds ago	(0 ms)
TimerTriggerCSharp1 (2017-09-20T05:06:4 ...)	✓	a few seconds ago	(0 ms)
TimerTriggerCSharp1 (2017-09-20T05:06:4 ...)	✓	a few seconds ago	(0 ms)
TimerTriggerCSharp1 (2017-09-20T05:06:3 ...)	✓	a few seconds ago	(0 ms)
TimerTriggerCSharp1 (2017-09-20T05:06:3 ...)	✓	a few seconds ago	(0 ms)