Louisville .NET Meetup – April 2018

From Zero to Serverless

Thanks to our Sponsors

MODIS

TEKsystems

Our people make IT possible.

MUSIC CITY TECH

JET BRAINS

TechSmith®

Upcoming Meetups

- May 17
  - Does it NEED to be a PWA?
    - Tara A. Manicsic
- June 21
  - Azure's Cosmos DB from a Developer's Perspective
    - Mike Schlegel
- July 19
  - Develop Couchbase Apps on Microsoft Azure
    - Matt Groves

Other Groups to Check Out

- JSLou
- Louisville Tech Ladies
- Louisville Tech Leaders
- IT Happy Hour

Upcoming Local Events

- Louisville Global Azure Bootcamp
  - April 21
- Louisville DevOps Global Bootcamp
  - June 16

Upcoming Online Events

- Microsoft Build (build.microsoft.com)
  - May 7 – 9
- Google I/O (events.google.com/io)
  - May 8 – 10

Upcoming Regional Events

- CodeStock – Knoxville, TN
  - April 20 - 21
- Stir Trek – Columbus, OH
  - May 4
- Music City Tech – Nashville
  - May 31 – June 2
- Pittsburg TechFest 2018
  - June 2
- Beer City Code – Grand Rapids, MI
  - June 22 – 23
- That Conference – Wisconsin Dells, WI
  - August 6 – 8
- Scenic City Summit 2018 – Chattanooga
  - August 17
- CoderCruise 2018 – The Bahamas
  - August 30 – September 3

BJ's ®
RESTAURANT
BREWHOUSE

Chad Green

From Zero to Serverless

Louisville .NET Meetup
April 19, 2018

⍰ Who is Chad Green

- Data & Solutions Architect at ProgressiveHealth
- Community Involvement
  - Code PaLOUsa Conference Chair
  - Louisville .NET Meetup Organizer
  - Louisville Tech Leaders Meetup Co-Organizer
  - Louisville Tech Ladies Committee Member
- Contact Information
  - ⍰ chadgreen@chadgreen.com
  - ⍰ chadgreen.com
  - ⍰ ChadGreen
  - ⍰ ChadwickEGreen

Our Agenda

What is Serverless Computing

Serverless Options

Azure Functions

Code Demonstrations

What is Serverless Computing

From Zero to Serverless

The evolution of application platforms

On-Premises

What media should I use to keep **backups**?

What is the right **size** of **servers** for the business needs?

How do I **deploy** new **code** to my **servers**?

What happens in case of **server hardware** failure?

How can I increase **server** utilization?

Which packages should be on my **server?**

Who **monitors** my **App**?

How often should I backup my **server**?

What size of **servers** should I **buy?**

How can I **scale** my app?

Are my **servers** in a secure location?

Do I need a secondary **network connection**?

Who has **physical** access to my **servers**?

Which **Operating System** should I use?

What **happens** if the power goes out?

How many **servers** do I need?

Who **monitors** my **servers?**

Do I need a **UPS**?

What **storage** do I need to use?

How often should I **patch** my **servers**?

It takes how long to **provision** a new **server**?

How can I dynamically configure my app?

The evolution of application platforms

IaaS

What is the right **size** of servers for my business needs?

How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my application?

How often should I **patch** my **servers**?

How often should I backup my **server**?

Which packages should be on my **server**?

How do I **deploy** new **code** to my **server**?

Which **Operating System** should I use?

Who **monitors** my application?

The evolution of application platforms

PaaS

What is the right **size** of servers for my business needs?

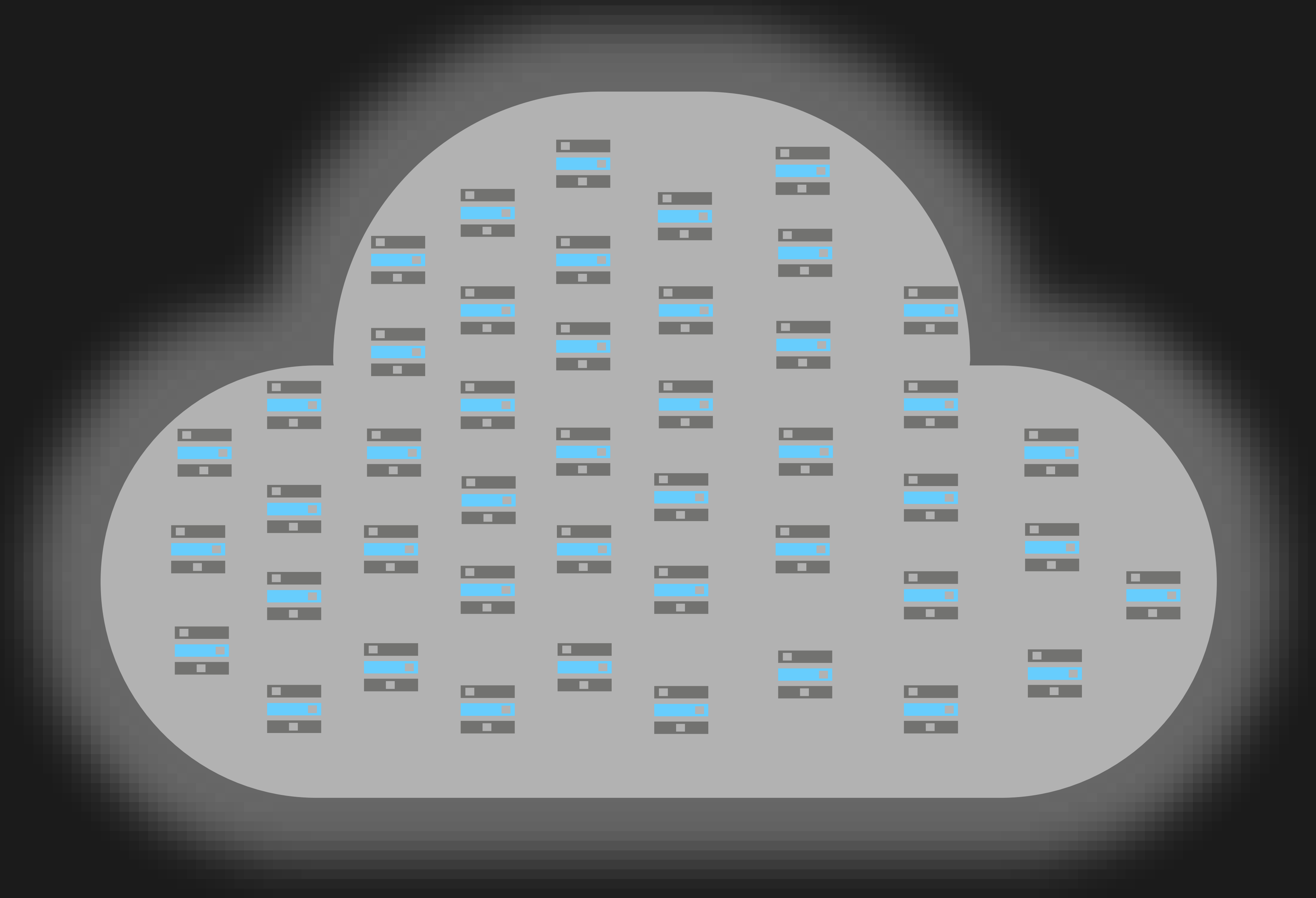
How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my application?

The evolution of application platforms

Serverless



The platform for next generation applications

What is Serverless?

Area #1 Backend as a Service (BaaS)

- Applications that significantly or fully depend on services (in the cloud) to manage server-side logic and state

Area #2 Functions as a Service (FaaS)

- Application run in stateless compute containers that are event-triggered, ephemeral, and fully managed by a 3rd party

What is Serverless?

Abstraction of Servers

Event-Driven/Instant Scale

Micro-Billing

Benefits of Serverless

Manage apps not servers

Reduced DevOps

Faster Time to Market

Challenges of Serverless Architecture

Complexity

Managing a monolithic application as a single unit is more straightforward than managing a fleet of purpose-built functions and the dependencies between them.

Organizational Support

It’s a non-trivial consideration for some to move to a serverless paradigm.

No Runtime Optimization

By its very nature, you mostly do not have control over the execution environment for the workload.

Serverless Options

From Zero to Serverless

Serverless Options

- ~~Zimki~~
- Google Cloud Functions
- Amazon Lambda
- IBM Openwhisk
- Auth0 WebTask

Azure Serverless

Functions

Execute your code based on events you specify

Logic Apps

Design workflows and orchestrate processes

Event Grid

Manage all events that can trigger code or logic

Database

Storage

Security

IoT

Analytics

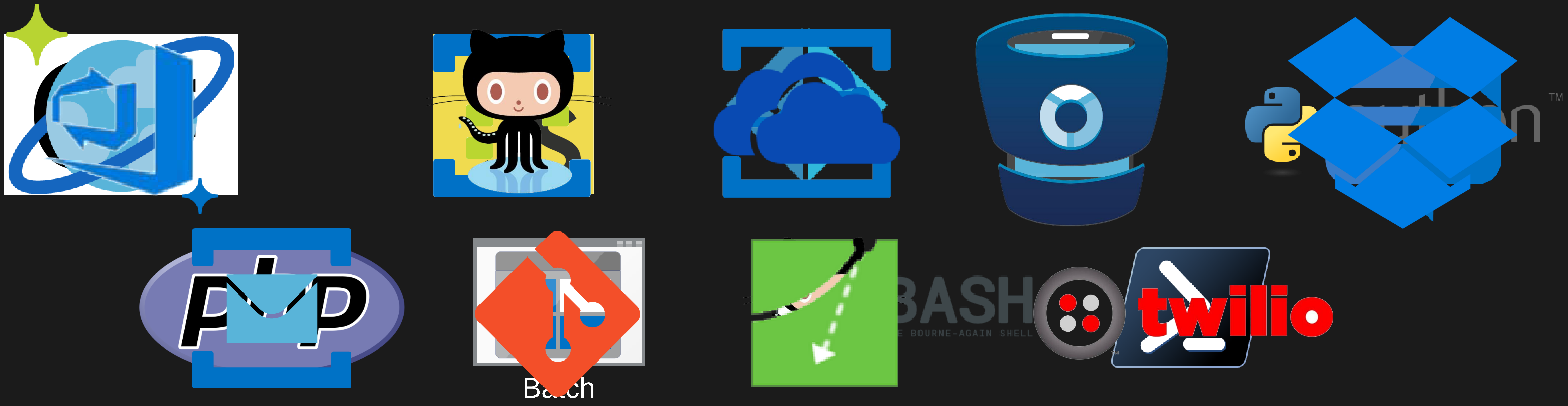
Intelligence

Azure Functions

From Zero to Serverless

Features of Azure Functions

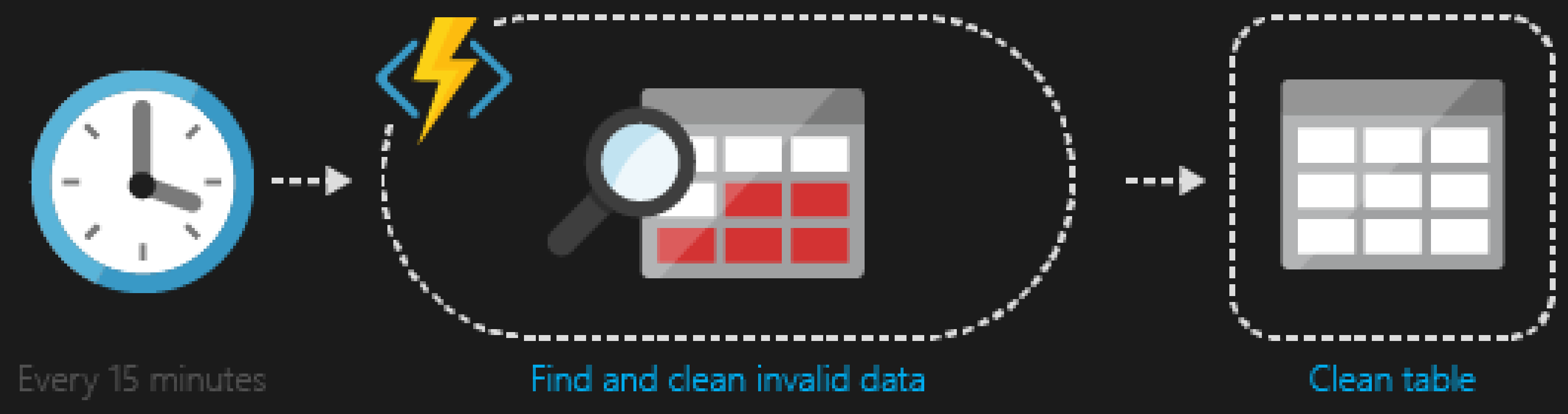
- Choice of language
- Pay-per-use pricing model
- Bring your own dependencies
- Integrated security
- Simplified integration
- Flexible development
- Open-source



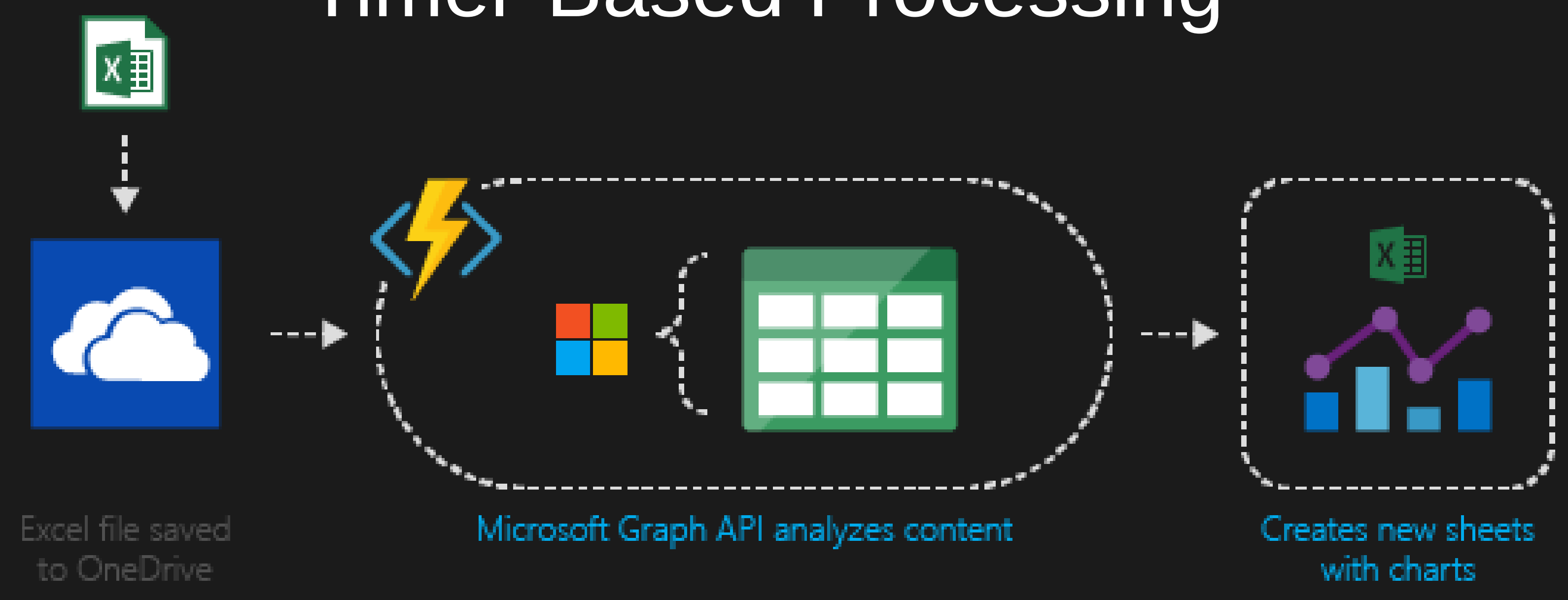
What can you do with Functions

- HTTPTrigger
- TimerTriger
- GitHub webhook
- Generic webhook
- CosmosDBTrigger
- BlobTrigger
- QueueTrigger
- EventHubTrigger
- ServiceBusQueueTrigger
- ServiceBusTopicTrigger

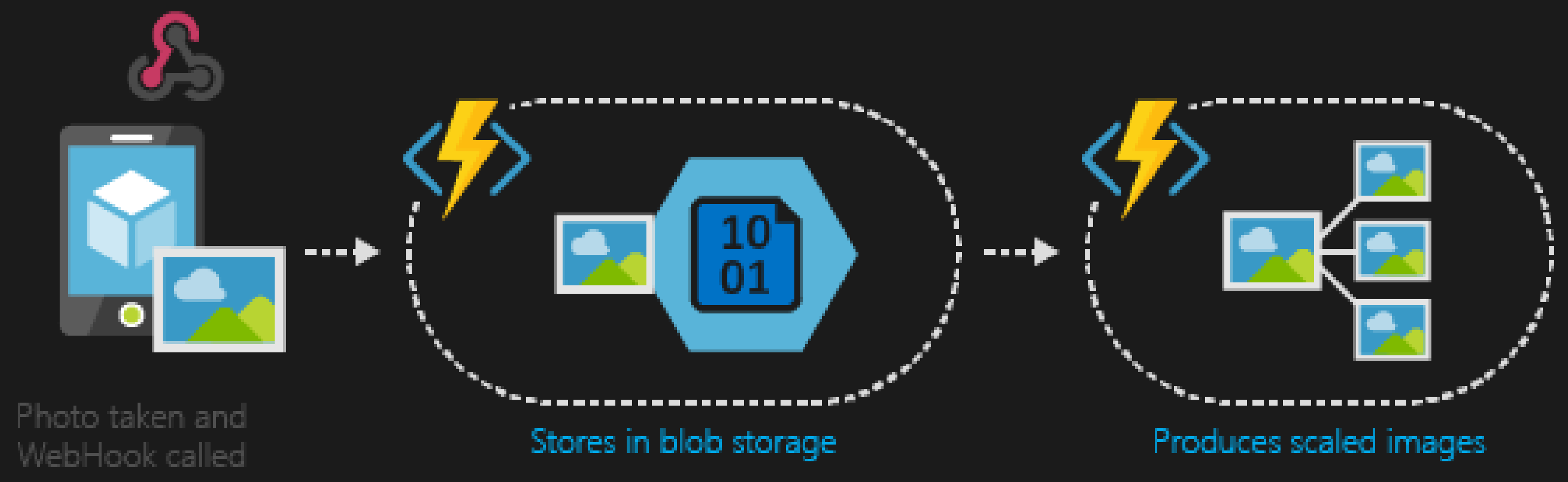
What can you do with Functions



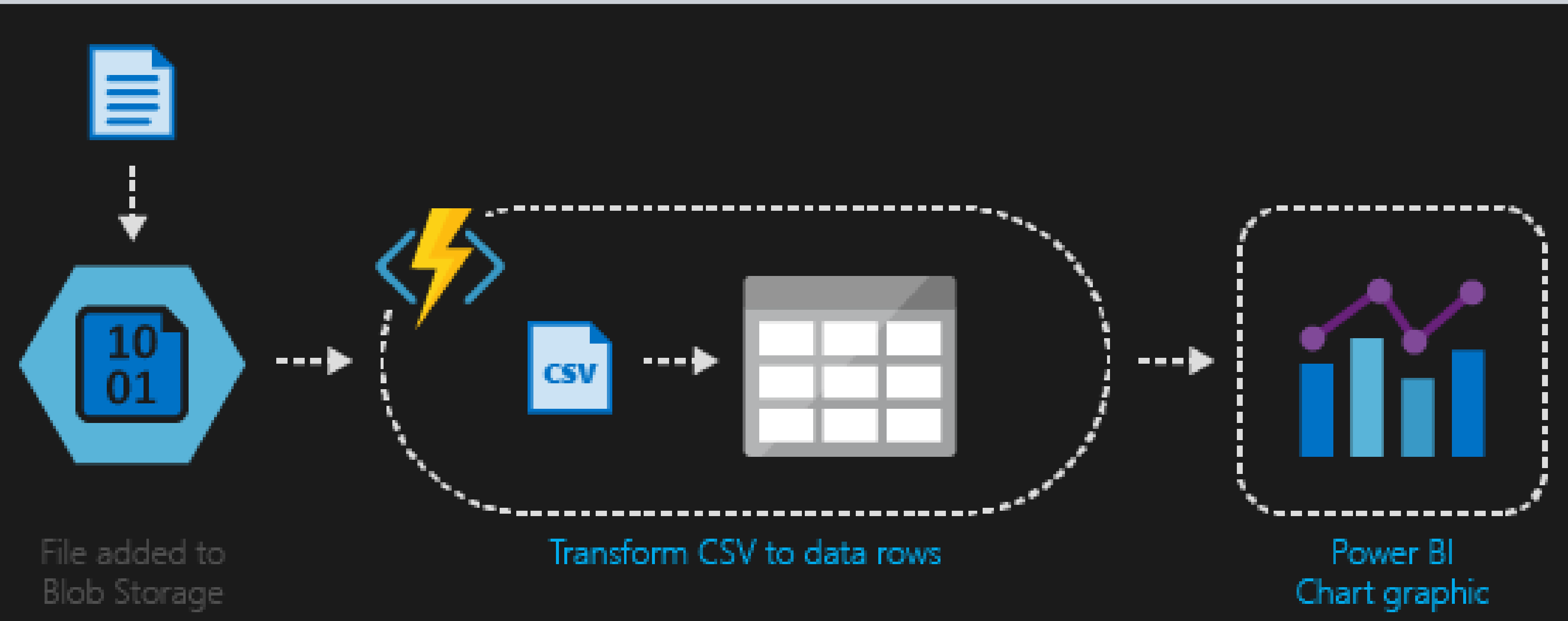
Timer-Based Processing


Excel file saved to OneDrive
Microsoft Graph API analyzes content
Creates new sheets with charts

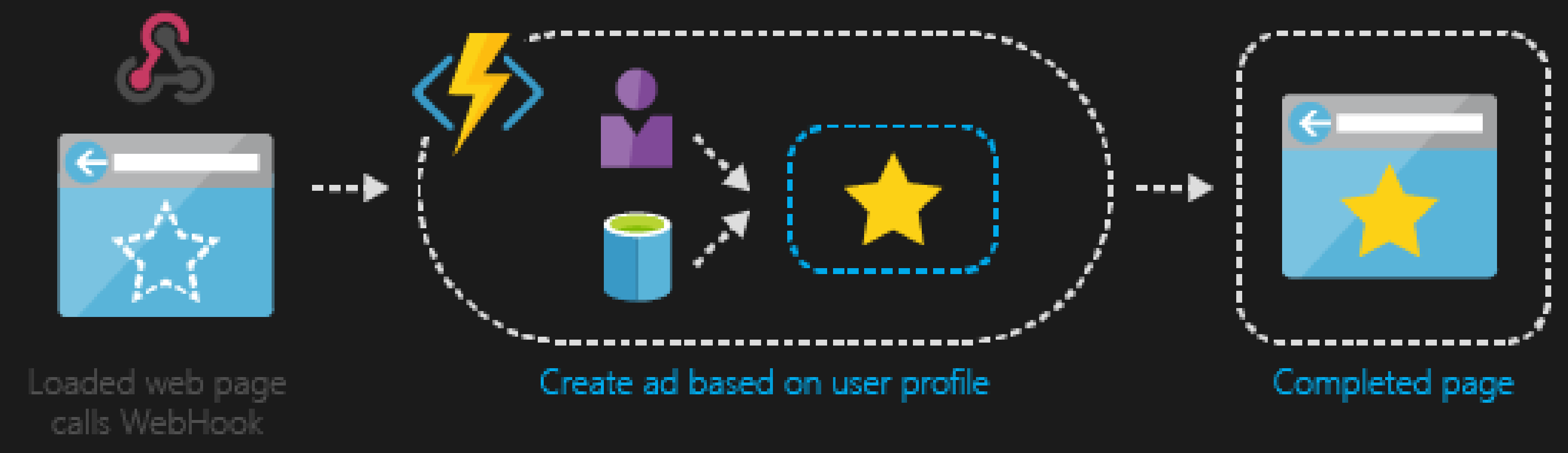
SaaS Event Processing



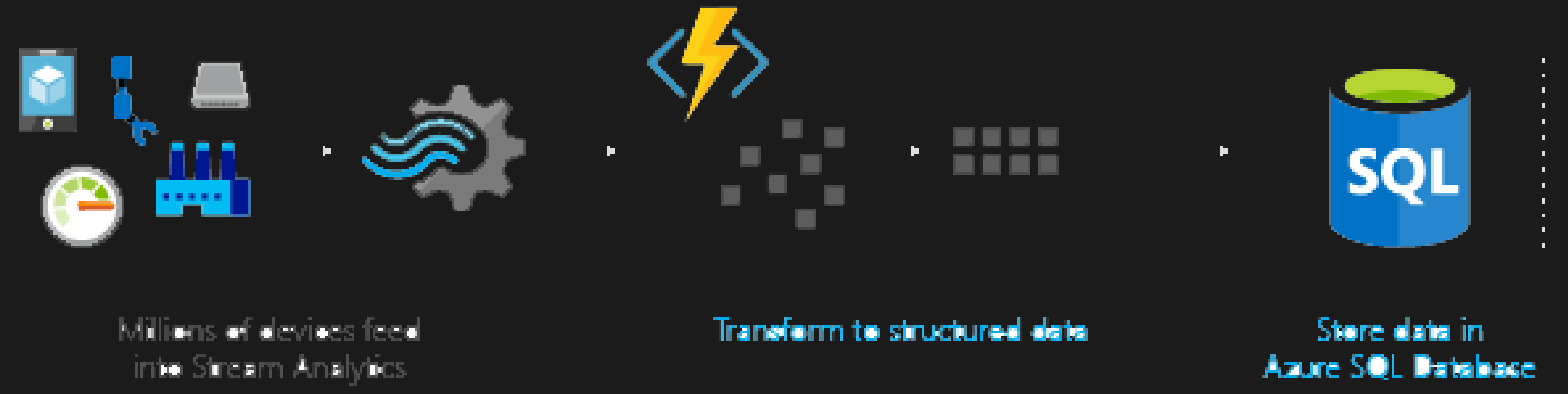
Serverless Mobile Back Ends



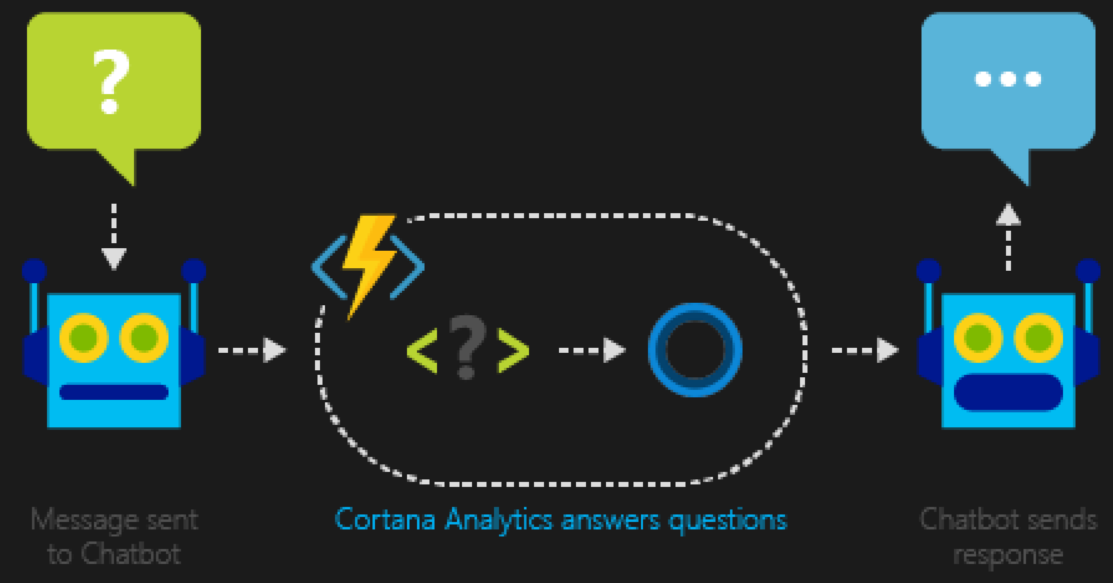
Azure Service Event Processing


Loaded web page calls WebHook
Create ad based on user profile
Completed page

Serverless Web Application Architectures


SQL
Millions of devices feed into Stream Analytics
Transform to structured data
Store data in Azure SQL Database

Real-Time Stream Processing



Real-Time Bot Messaging

Azure Functions Pricing

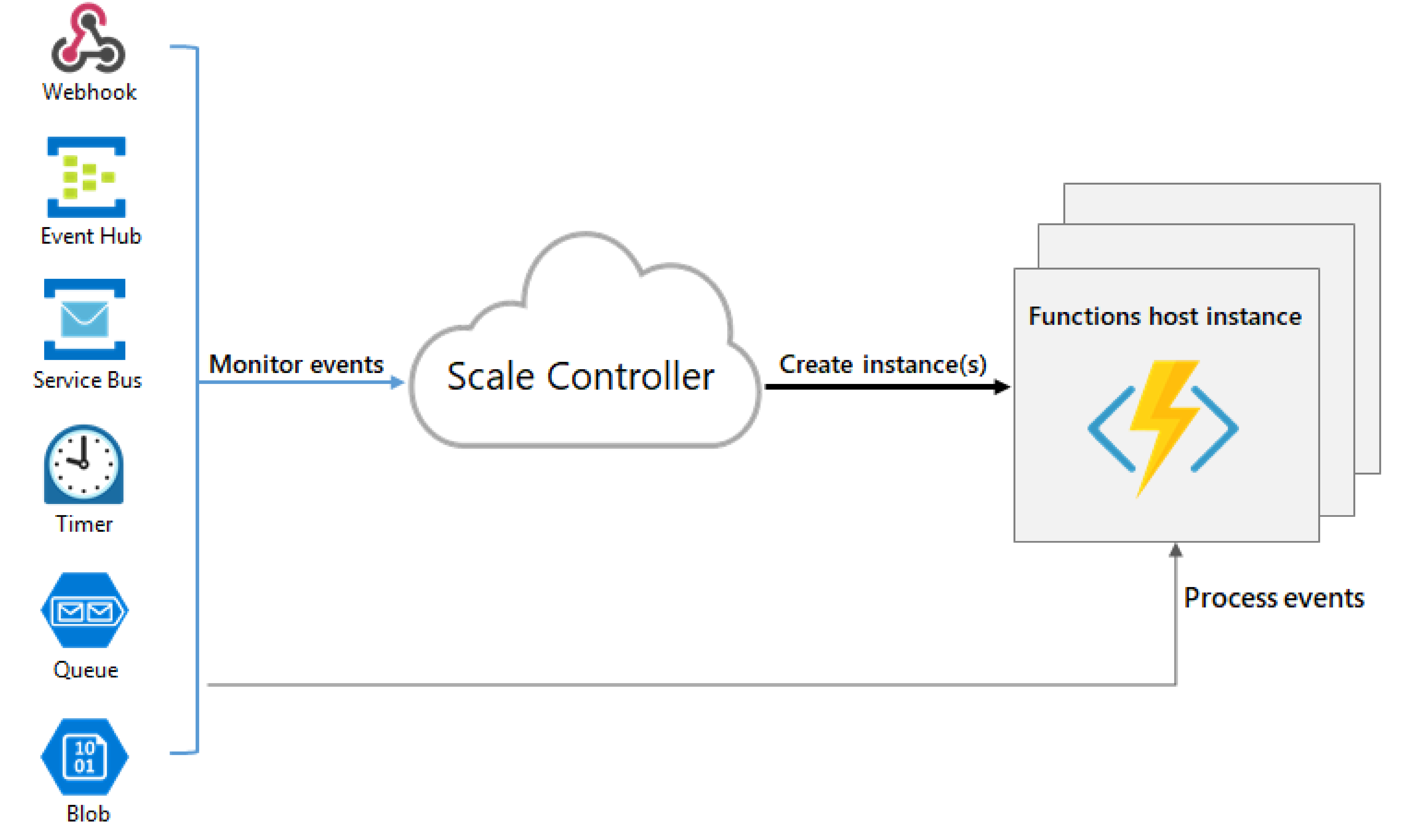
Consumption Plan

- Takes care of everything but your code
- Pay only when your functions are running
- Scale out automatically

App Service Plan

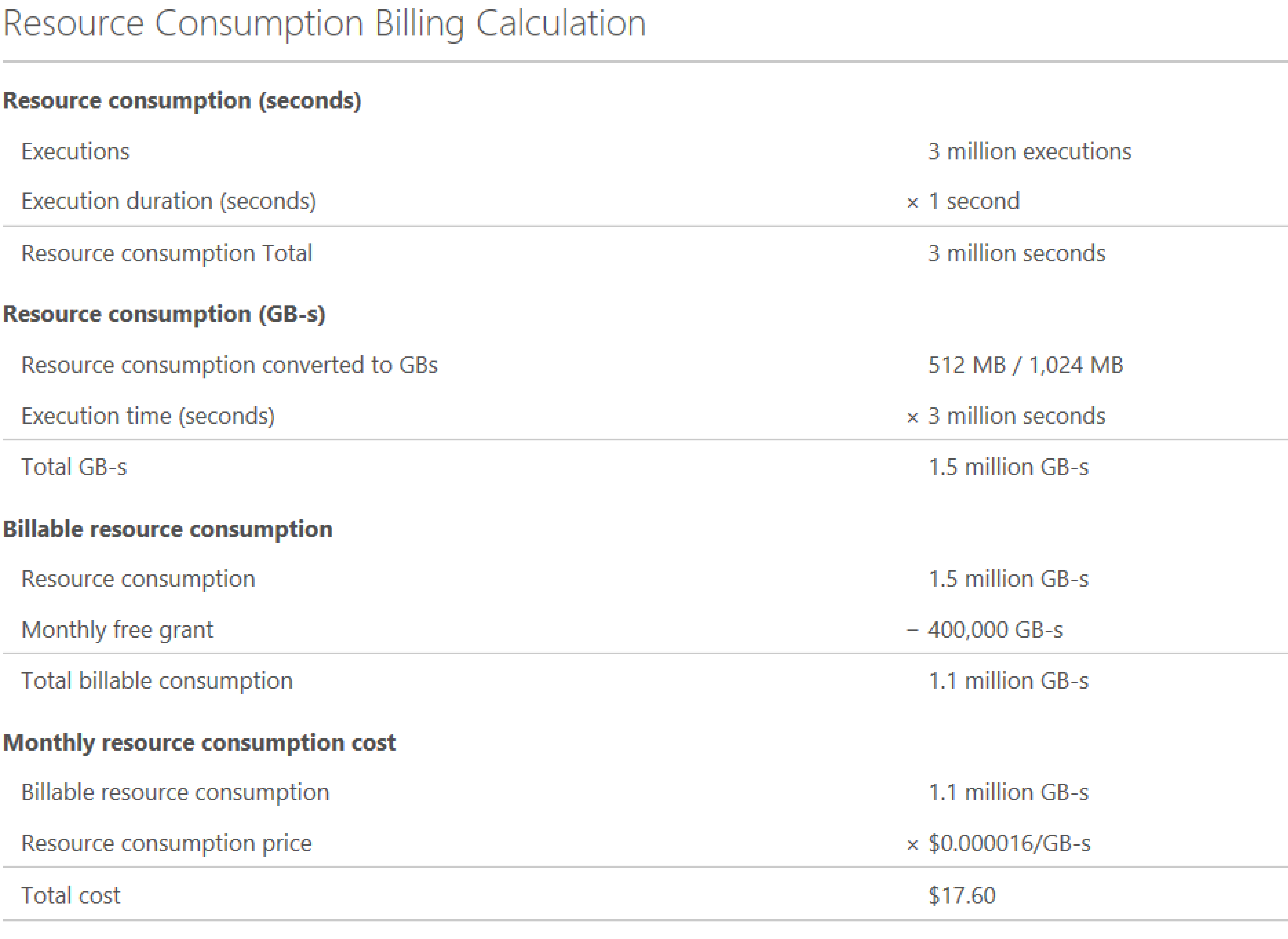
- You pretty much take care of everything
- Consider when:
    - Existing, underutilized VMs
    - Function apps to run continuously
    - More CPU or memory options
    - Run longer than maximum execution time
    - Require features only available on App Service plan
    - Want to run on Linux

How the Consumption plan works



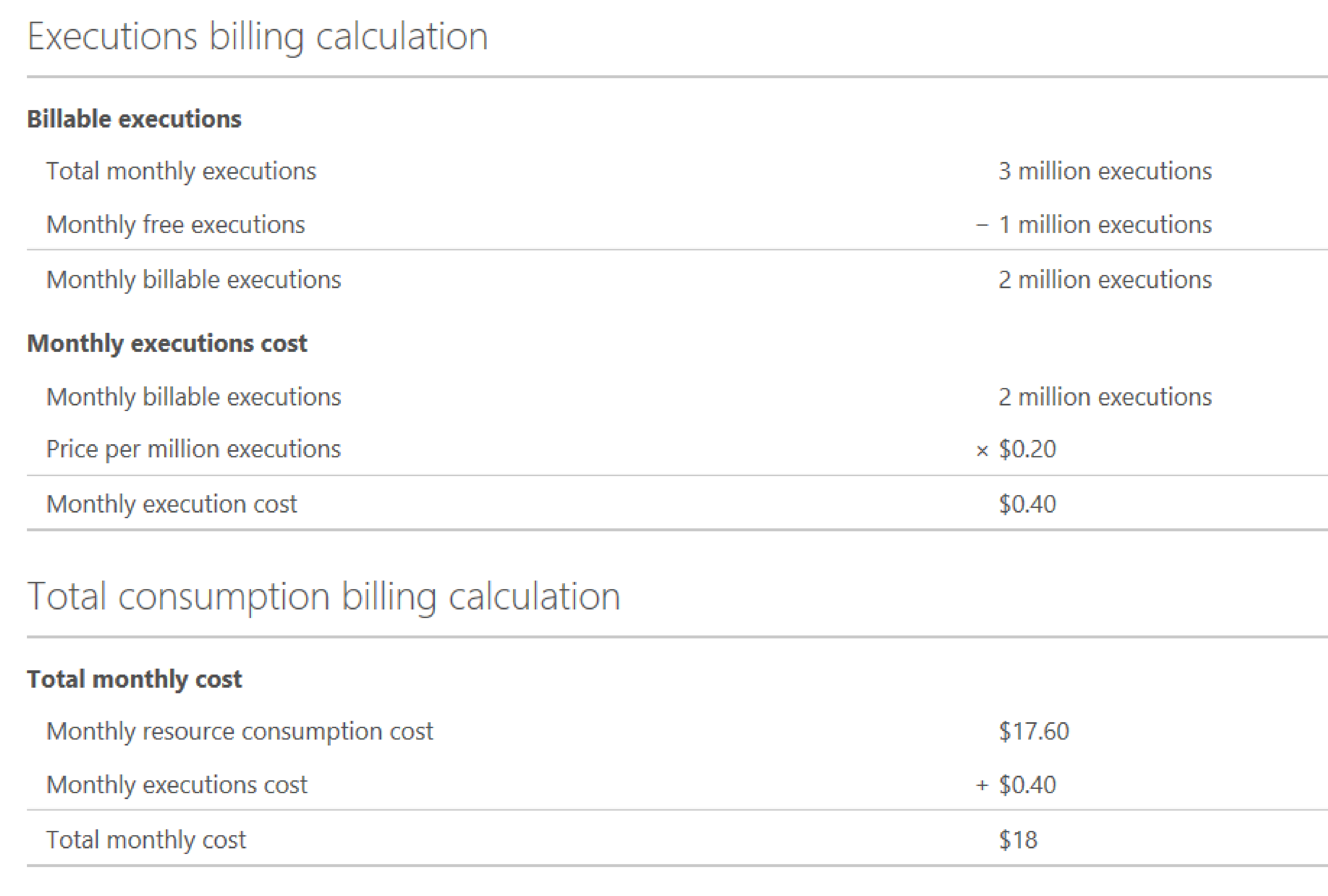
Azure Functions Pricing

- Gigabyte-second (GB-s) – Combination of memory size and execution time
- Executions – Each time a function is executed



Resource Consumption Billing Calculation

Resource consumption (seconds)	
Executions	3 million executions
Execution duration (seconds)	× 1 second
Resource consumption Total	3 million seconds
Resource consumption (GB-s)	
Resource consumption converted to GBs	512 MB / 1,024 MB
Execution time (seconds)	× 3 million seconds
Total GB-s	1.5 million GB-s
Billable resource consumption	
Resource consumption	1.5 million GB-s
Monthly free grant	− 400,000 GB-s
Total billable consumption	1.1 million GB-s
Monthly resource consumption cost	
Billable resource consumption	1.1 million GB-s
Resource consumption price	× $0.000016/GB-s
Total cost	$17.60



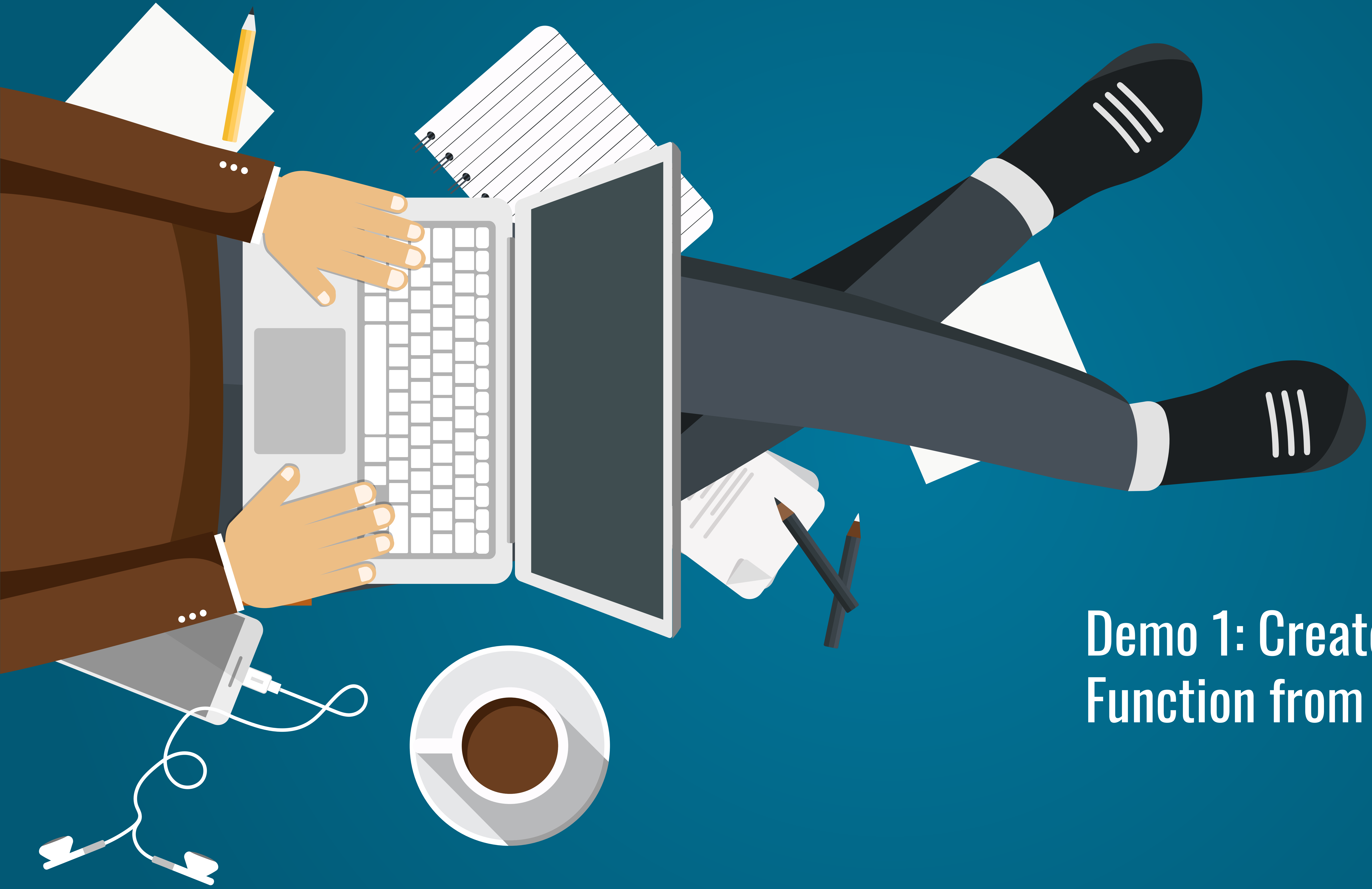
Executions billing calculation

Billable executions	
Total monthly executions	3 million executions
Monthly free executions	− 1 million executions
Monthly billable executions	2 million executions
Monthly executions cost	
Monthly billable executions	2 million executions
Price per million executions	× $0.20
Monthly execution cost	$0.40

Total consumption billing calculation

Total monthly cost	
Monthly resource consumption cost	$17.60
Monthly executions cost	+ $0.40
Total monthly cost	$18

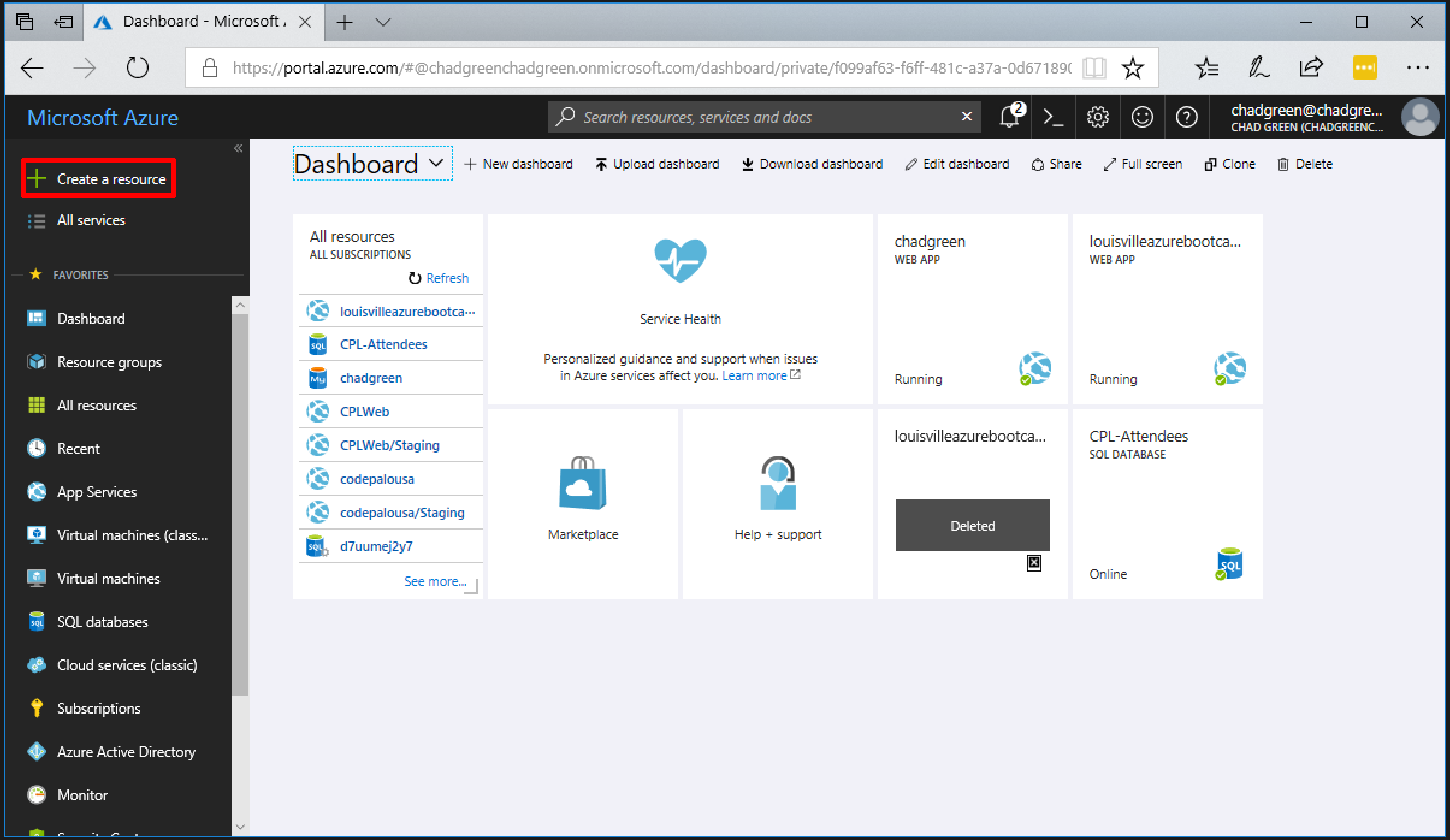
Code Demonstrations



Demo 1: Create an Azure Function from the Portal

Demo 1: Create an Azure Function from the Portal

Log into the Azure Portal – https://portal.azure.com

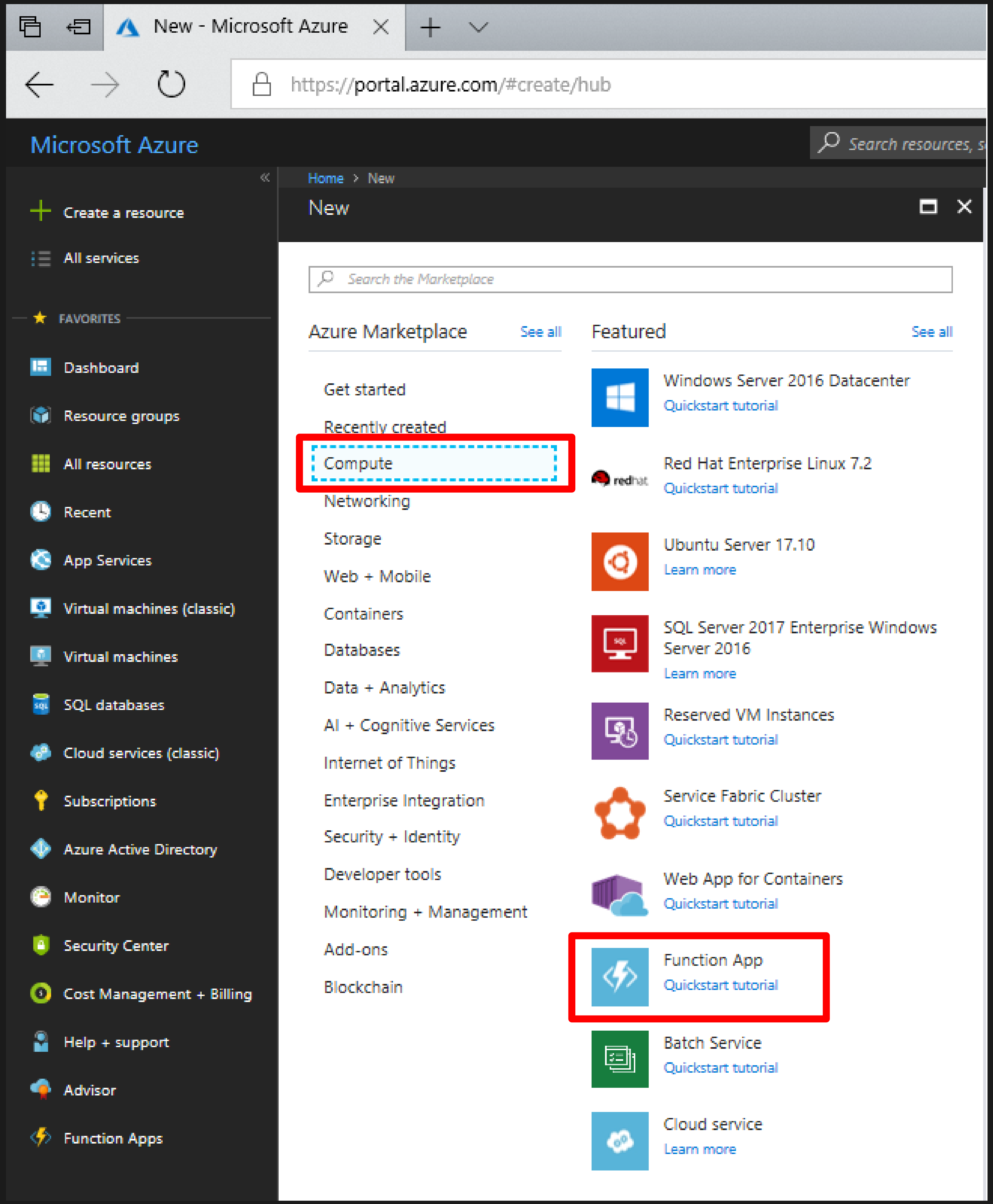


https://portal.azure.com/#@chadgreenchadgreen.onmicrosoft.com/dashboard/private/f099af63-f6ff-481c-a37a-0d67189(

Personalized guidance and support when issues in Azure services affect you. Learn more

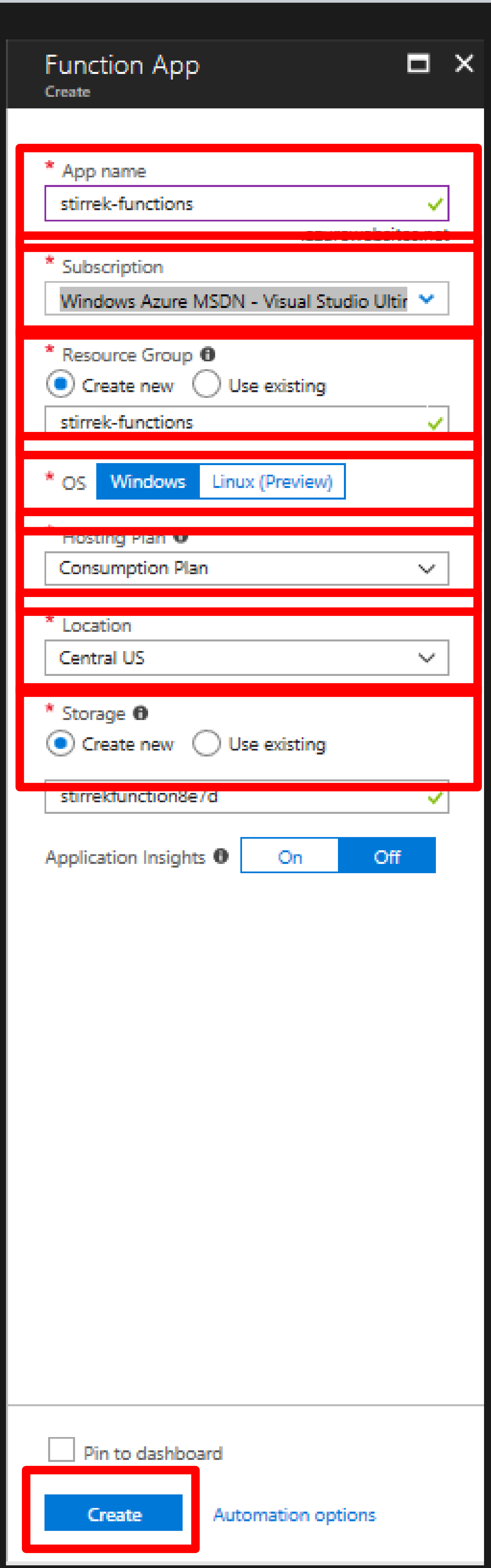
Demo 1: Create an Azure Function from the Portal

Create a function app



Demo 1: Create an Azure Function from the Portal

Create a function app



Function App
Create

* App name
stirrek-functions ✓
.azurewebsites.net

* Subscription
Windows Azure MSDN - Visual Studio Ultir

* Resource Group
● Create new ○ Use existing
stirrek-functions ✓

* OS: [✓ Windows] Linux (Preview)

* Hosting Plan
Consumption Plan

* Location
Central US

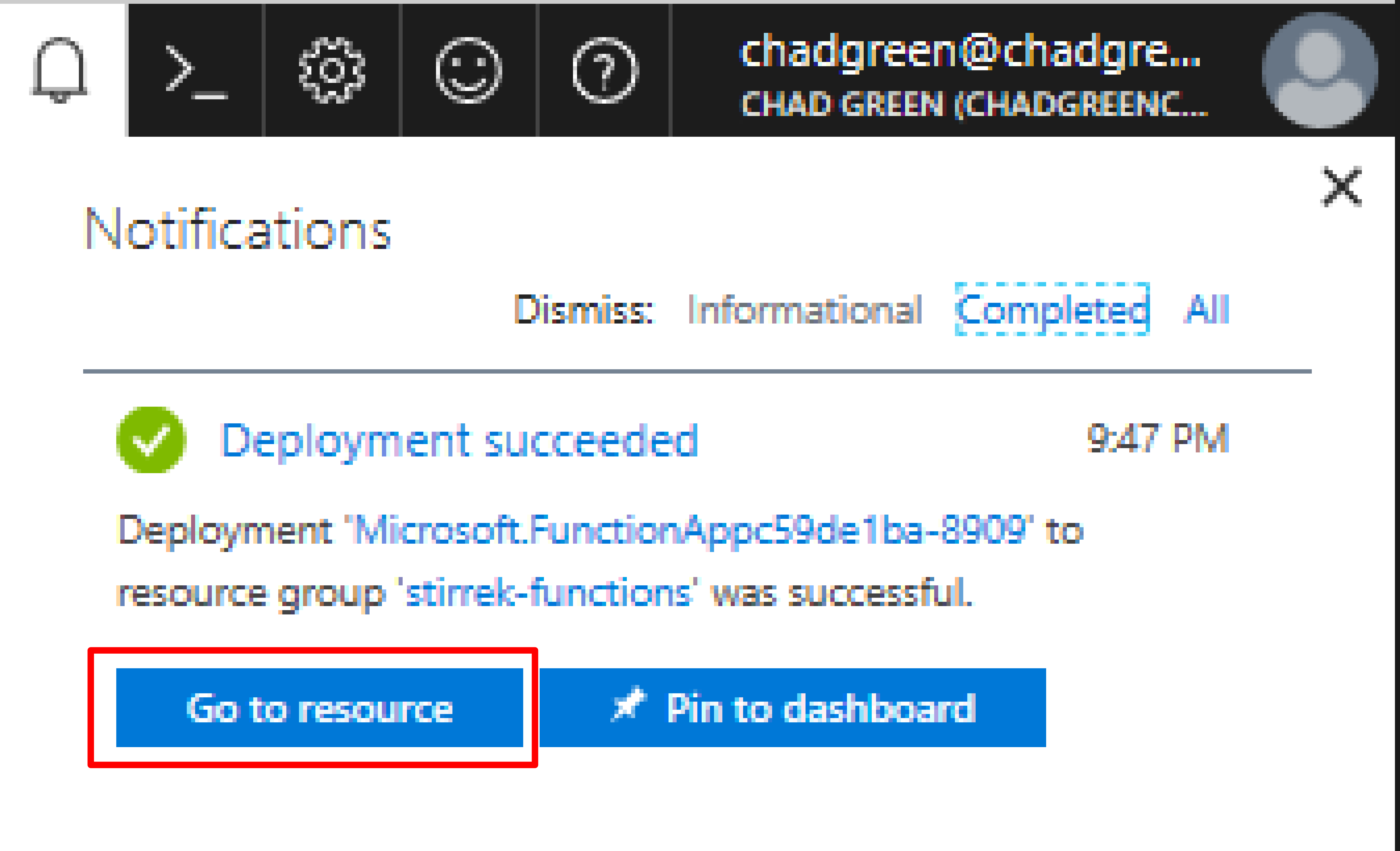
* Storage
● Create new ○ Use existing
stirrekfunction8e7d ✓

Application Insights: On [✓ Off]

□ Pin to dashboard

Demo 1: Create an Azure Function from the Portal

Create a function app



Notifications

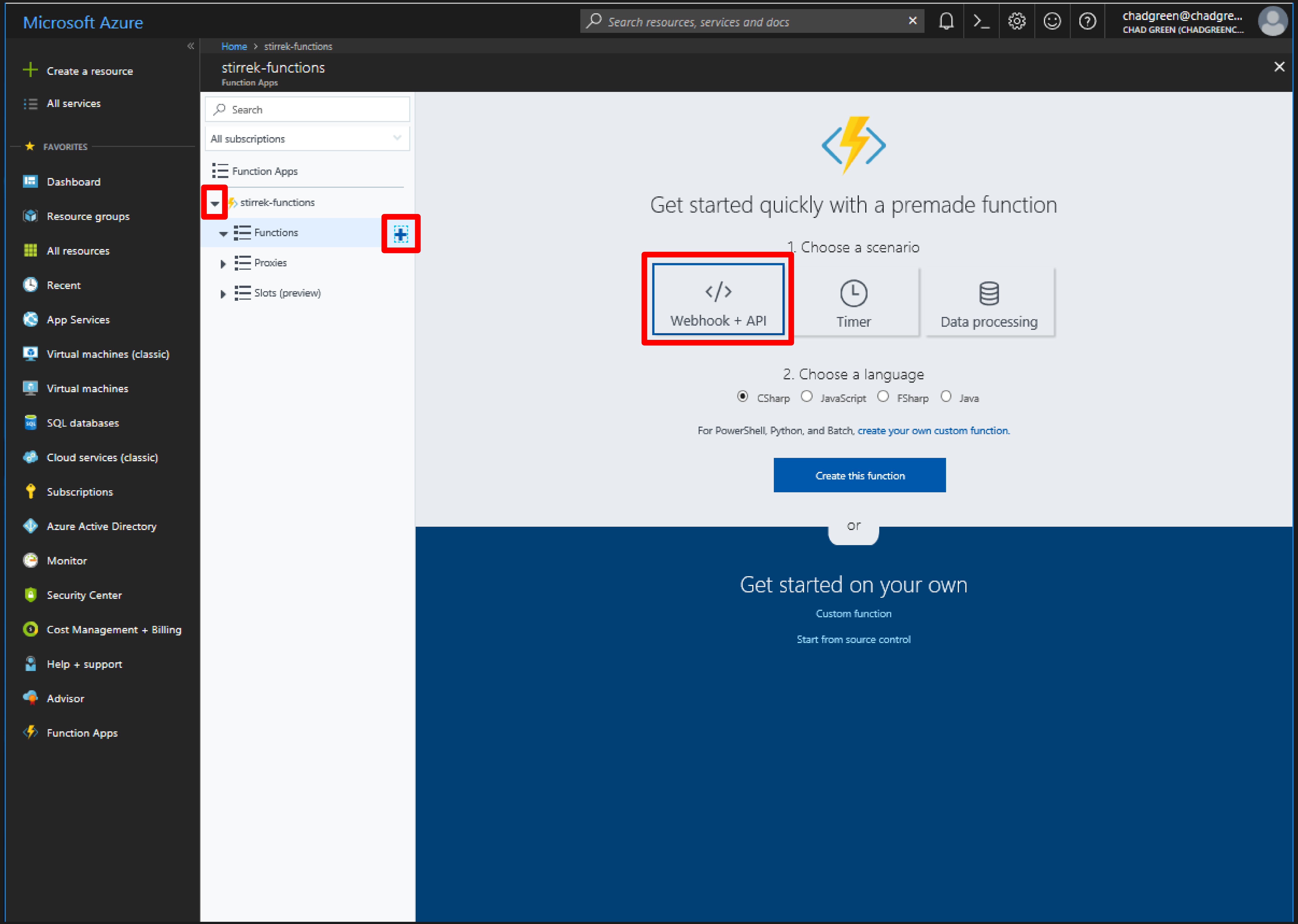
Dismiss: Informational Completed All

Deployment succeeded 9:47 PM

Deployment 'Microsoft.FunctionAppc59de1ba-8909' to resource group 'stimek-functions' was successful.

Demo 1: Create an Azure Function from the Portal

Create an HTTP triggered function



Get started quickly with a premade function

1. Choose a scenario

2. Choose a language

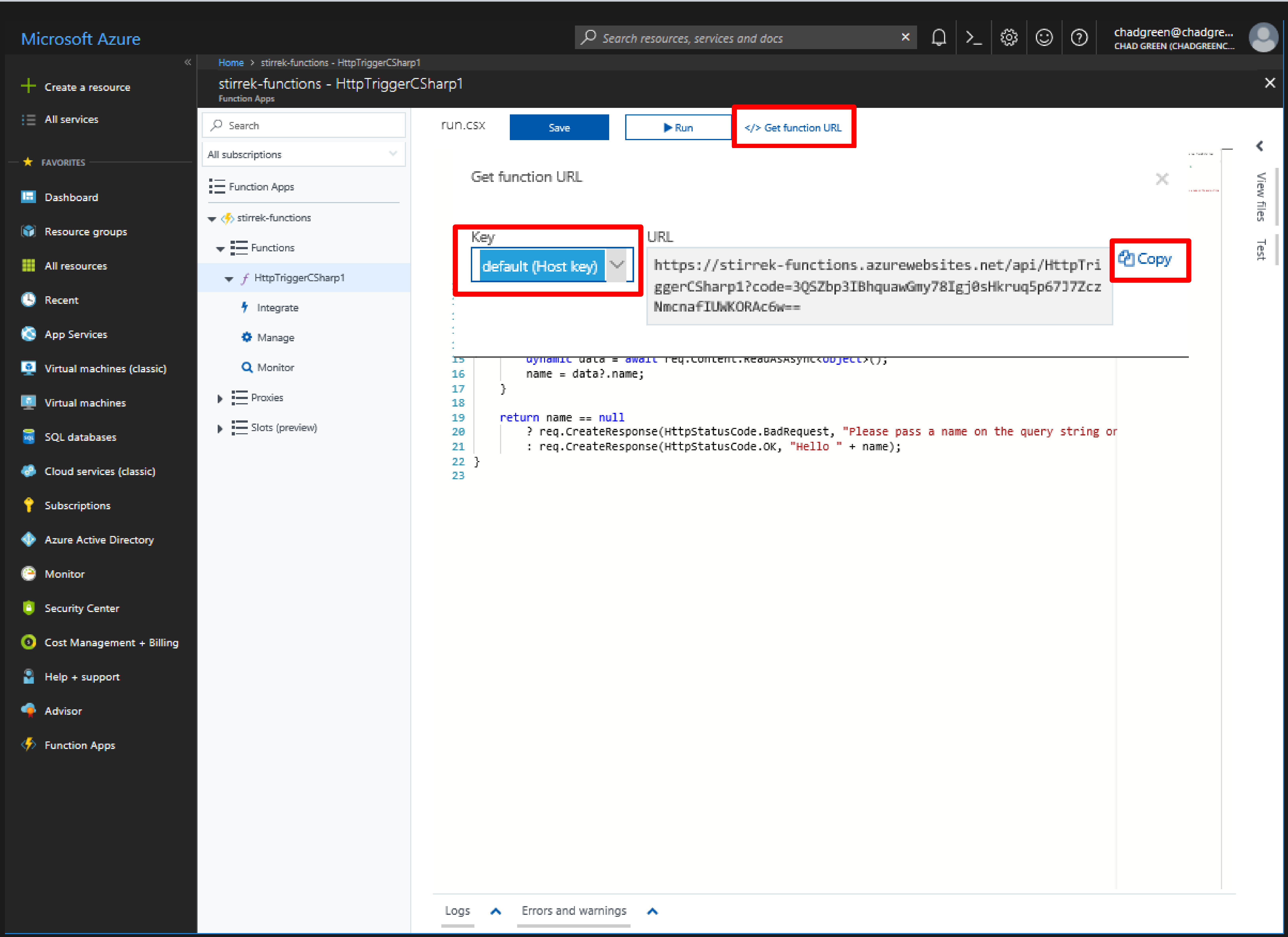
● CSharp ○ JavaScript ○ FSharp ○ Java

For PowerShell, Python, and Batch, create your own custom function.

Get started on your own

Demo 1: Create an Azure Function from the Portal

Test the function



Get function URL

Key: default (Host key)

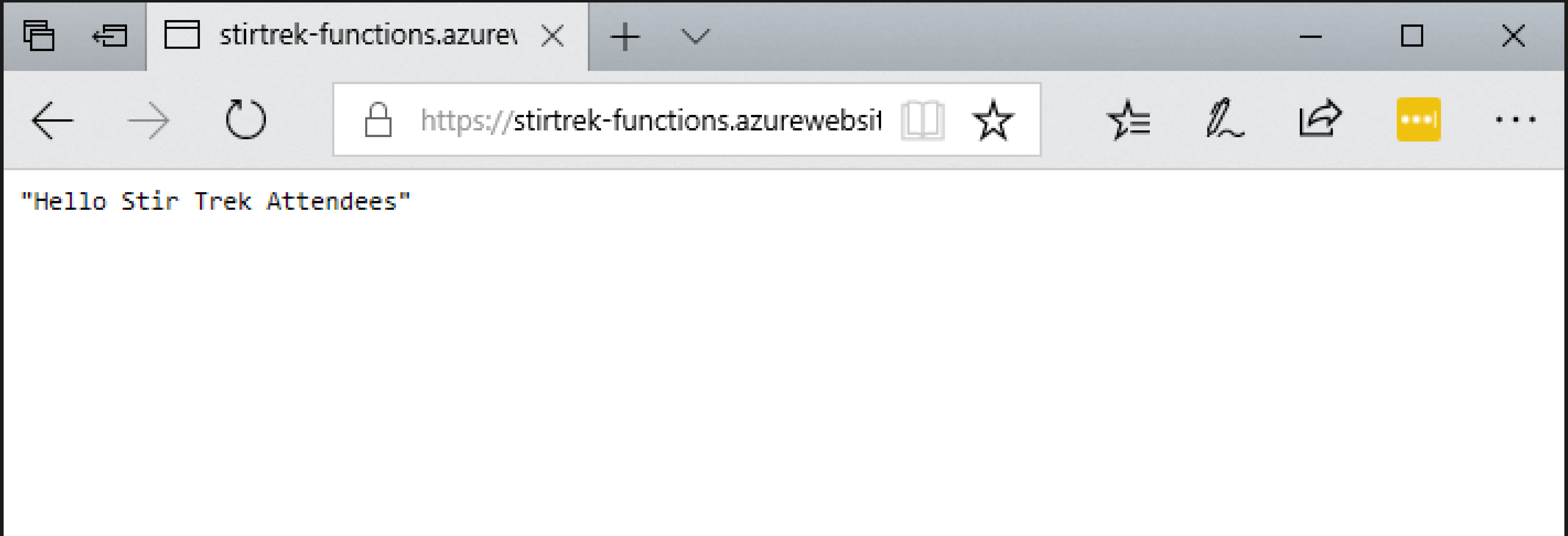
URL: https://stirrek-functions.azurewebsites.net/api/HttpTriggerCSharp1?code=3QSZbp3IBhquawGmy78Igj0sHkruq5p67J7ZczNmcnafIUWKORAc6w==

Copy

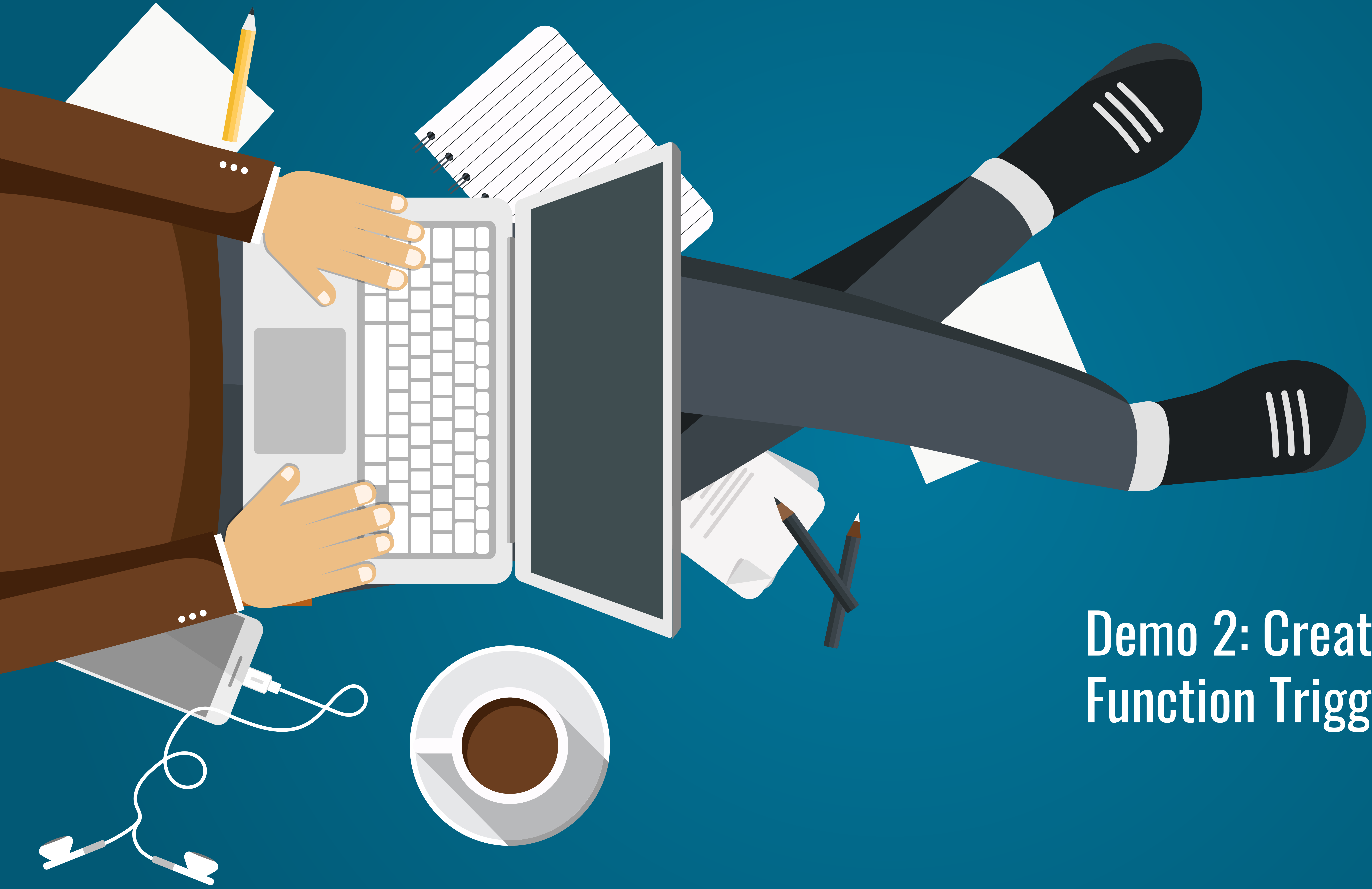
```
        dynamic data = await req.Content.ReadAsAsync<object>();
        name = data?.name;
    }

    return name == null
        ? req.CreateResponse(HttpStatusCode.BadRequest, "Please pass a name on the query string or
        : req.CreateResponse(HttpStatusCode.OK, "Hello " + name);
}
```

Demo 1: Create an Azure Function from the Portal

Test the function



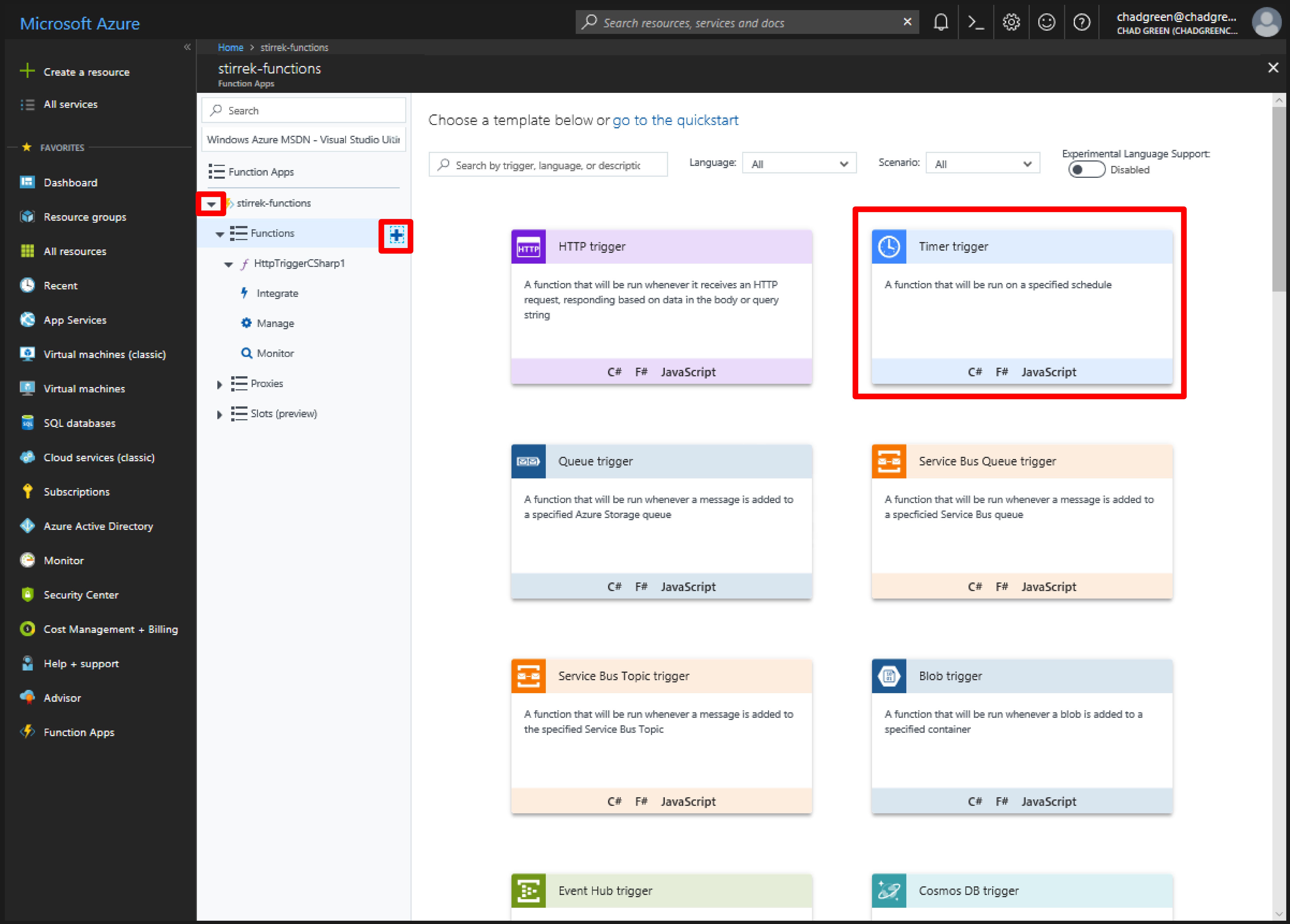
```
"Hello Stir Trek Attendees"
```



Demo 2: Create an Azure Function Triggered by a Timer

Demo 2: Create an Azure Function Triggered by a Timer

Create a timer triggered function



Choose a template below or go to the quickstart

Search by trigger, language, or descriptic

Language: All

Scenario: All

Experimental Language Support: Disabled

HTTP trigger
A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string
C# F# JavaScript

Timer trigger
A function that will be run on a specified schedule
C# F# JavaScript

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

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

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

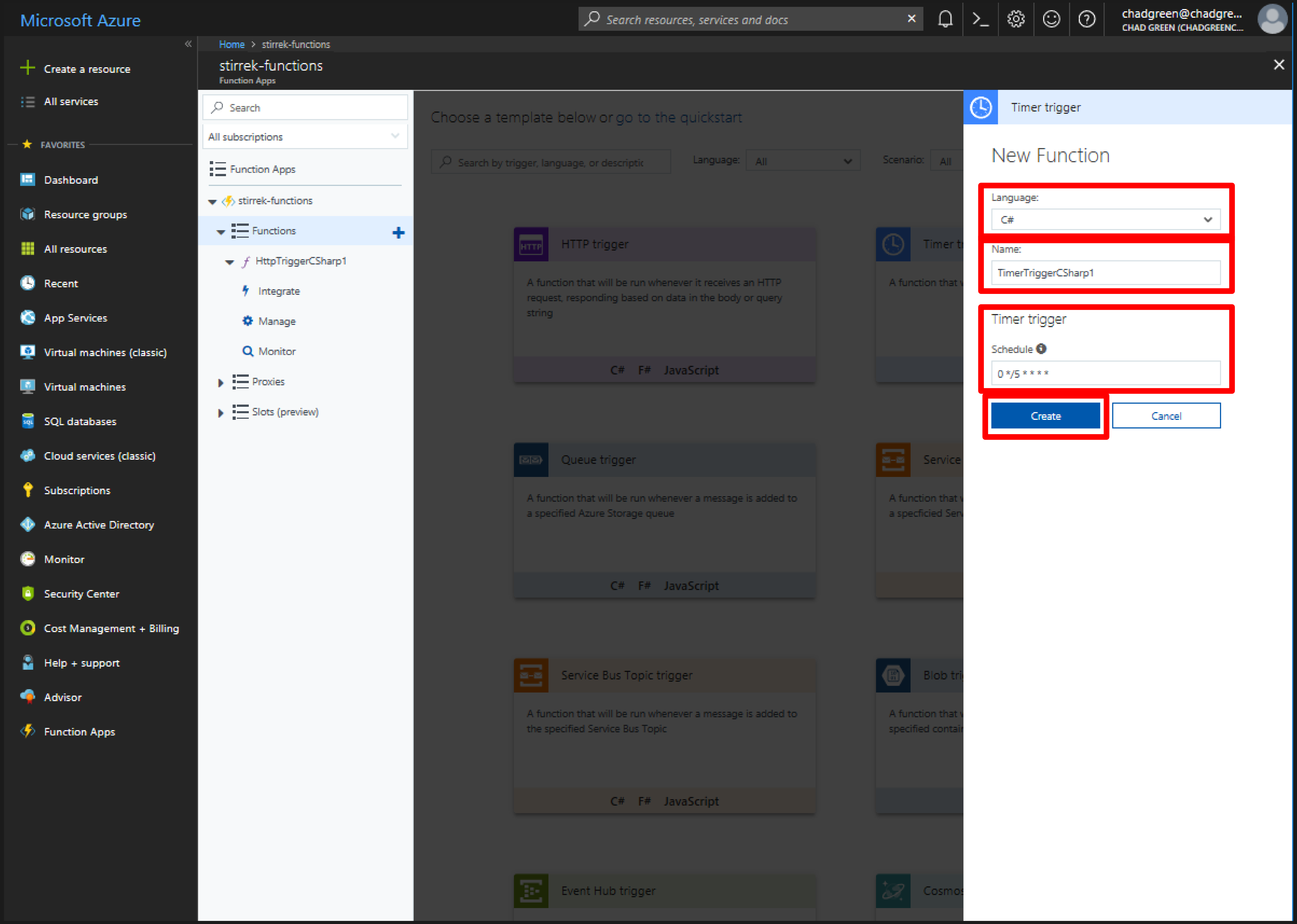
Blob trigger
A function that will be run whenever a blob is added to a specified container
C# F# JavaScript

Event Hub trigger

Cosmos DB trigger

Demo 2: Create an Azure Function Triggered by a Timer

Create a timer triggered function



Choose a template below or go to the quickstart

HTTP trigger: A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string

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

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

New Function

Language: C#

Name: TimerTriggerCSharp1

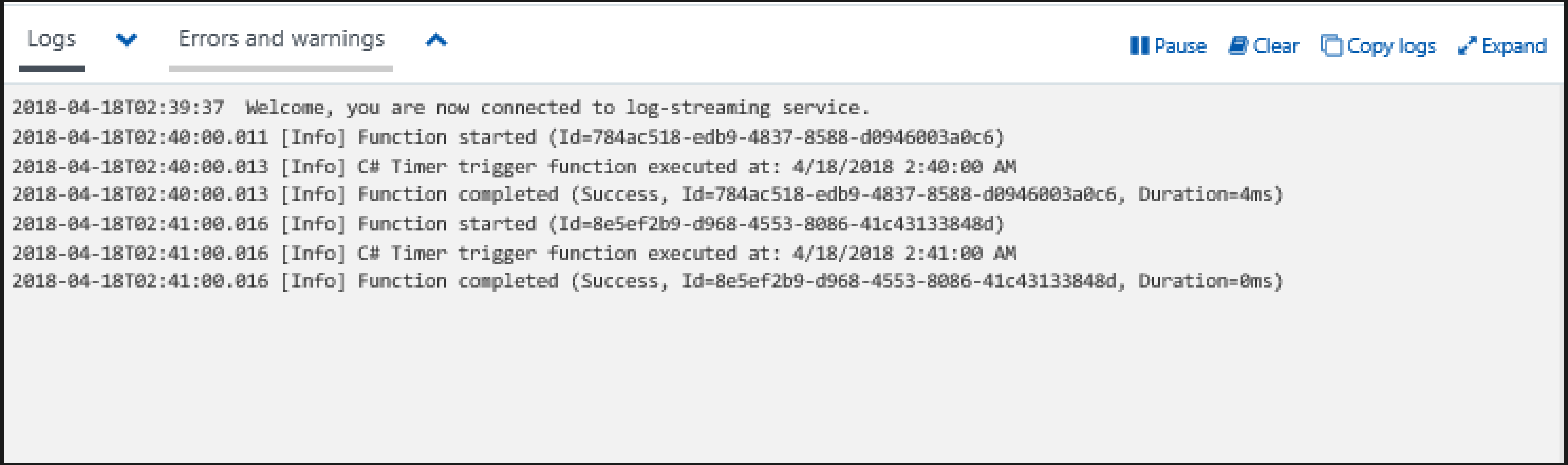
Timer trigger

Schedule: 0 */5 * * * *

Create | Cancel

Demo 2: Create an Azure Function Triggered by a Timer

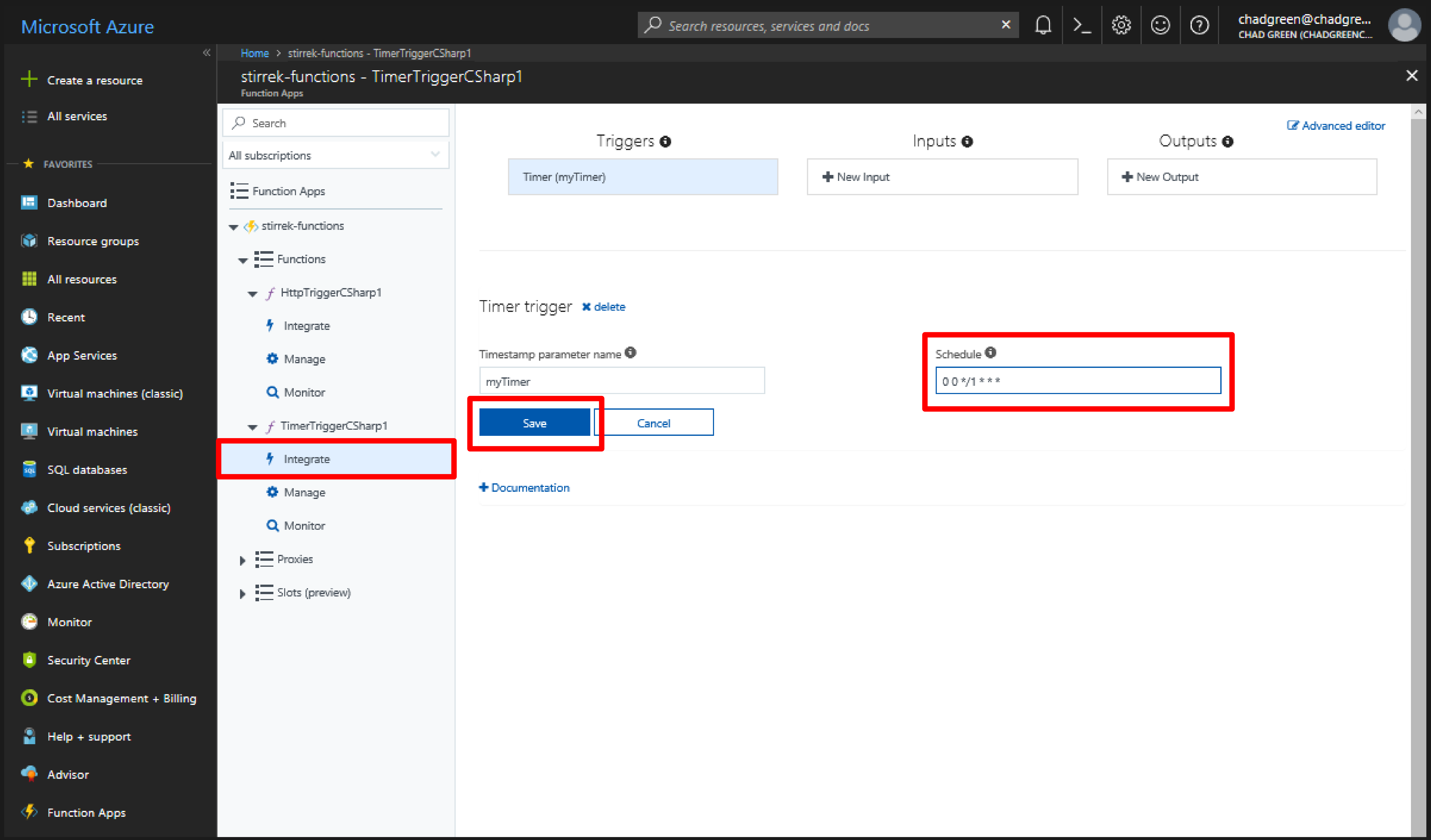
Verify execution



```
2018-04-18T02:39:37  Welcome, you are now connected to log-streaming service.
2018-04-18T02:40:00.011 [Info] Function started (Id=784ac518-edb9-4837-8588-d0946003a0c6)
2018-04-18T02:40:00.013 [Info] C# Timer trigger function executed at: 4/18/2018 2:40:00 AM
2018-04-18T02:40:00.013 [Info] Function completed (Success, Id=784ac518-edb9-4837-8588-d0946003a0c6, Duration=4ms)
2018-04-18T02:41:00.016 [Info] Function started (Id=8e5ef2b9-d968-4553-8086-41c43133848d)
2018-04-18T02:41:00.016 [Info] C# Timer trigger function executed at: 4/18/2018 2:41:00 AM
2018-04-18T02:41:00.016 [Info] Function completed (Success, Id=8e5ef2b9-d968-4553-8086-41c43133848d, Duration=0ms)
```

Demo 2: Create an Azure Function Triggered by a Timer

Update the timer's schedule



Timer trigger

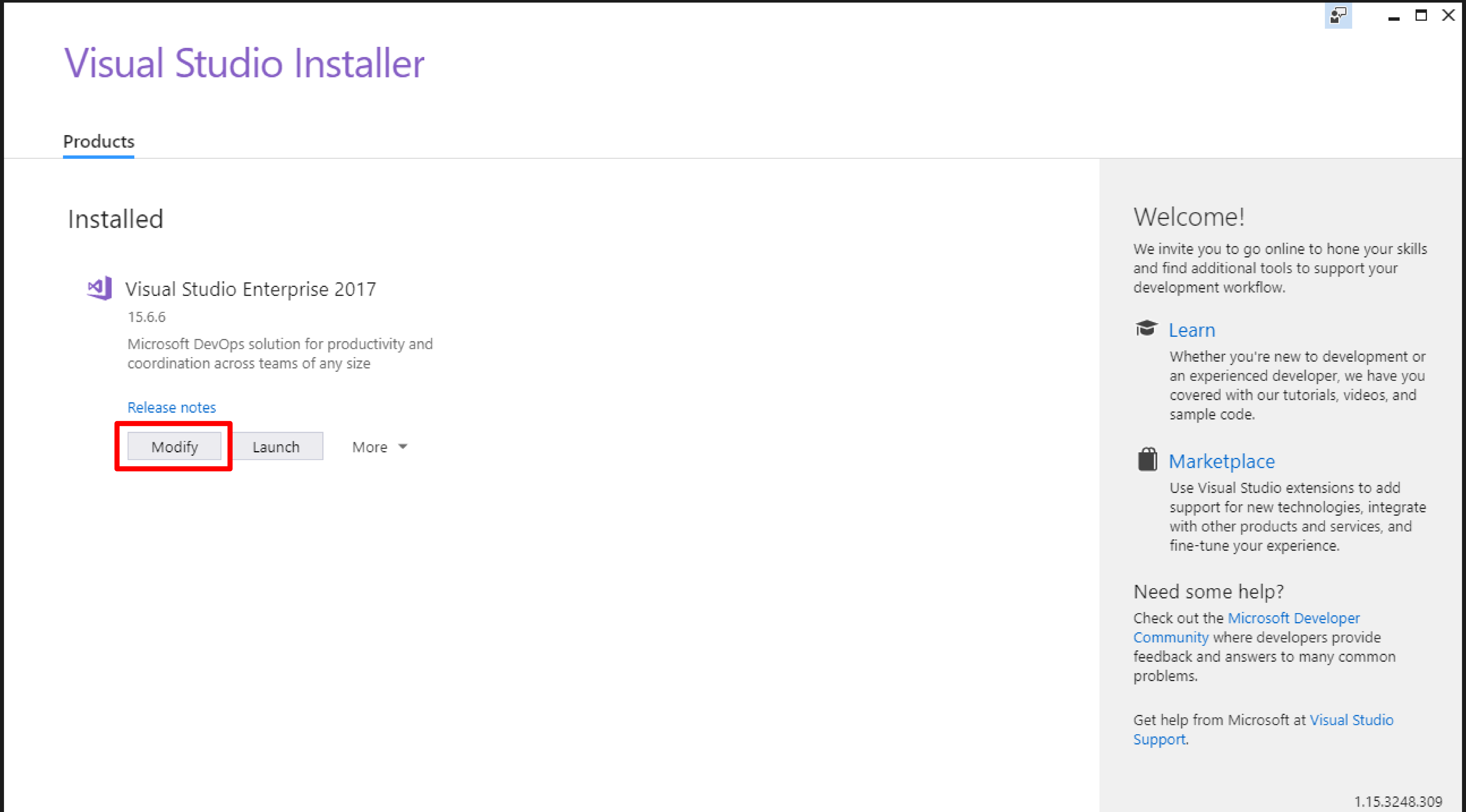
Timestamp parameter name: myTimer

Schedule: 0 0 */1 * * *

Demo 3: Create an Azure Function from Visual Studio

Demo 3: Create an Azure Function from Visual Studio

Install Visual Studio 2017



Visual Studio Installer

Products

Installed

Visual Studio Enterprise 2017

15.6.6

Microsoft DevOps solution for productivity and coordination across teams of any size

Release notes

Modify Launch More ▾

Welcome!

We invite you to go online to hone your skills and find additional tools to support your development workflow.

Learn

Whether you're new to development or an experienced developer, we have you covered with our tutorials, videos, and sample code.

Marketplace

Use Visual Studio extensions to add support for new technologies, integrate with other products and services, and fine-tune your experience.

Need some help?

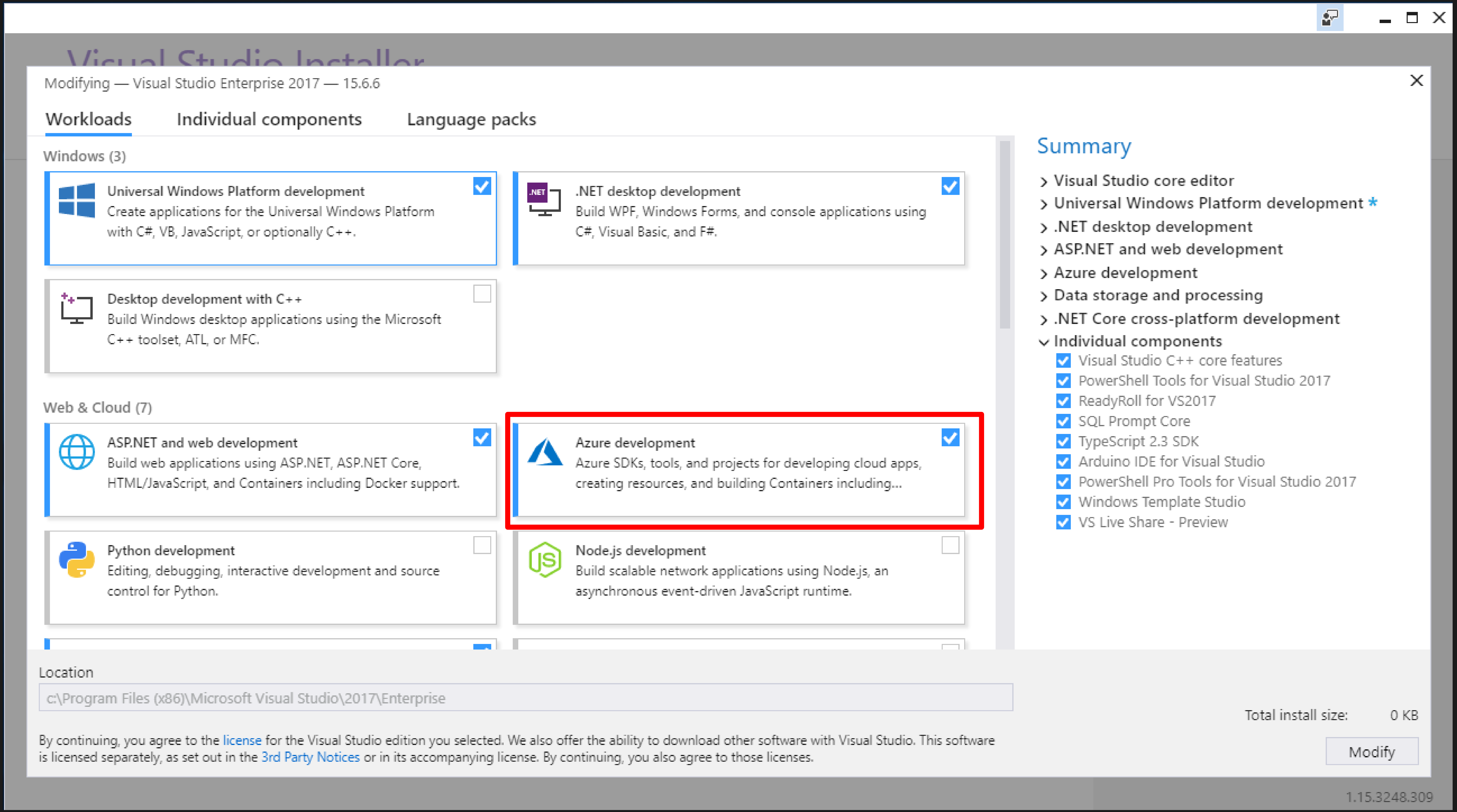
Check out the Microsoft Developer Community where developers provide feedback and answers to many common problems.

Get help from Microsoft at Visual Studio Support.

1.15.3248.309

Demo 3: Create an Azure Function from Visual Studio

Include the Azure development workload



Visual Studio Installer

Modifying — Visual Studio Enterprise 2017 — 15.6.6

Windows (3)

☑ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.

☑ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

□ Desktop development with C++
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.

Web & Cloud (7)

☑ ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.

☑ Azure development
Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...

□ Python development
Editing, debugging, interactive development and source control for Python.

□ Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Summary

- Visual Studio core editor
- Universal Windows Platform development *
- .NET desktop development
- ASP.NET and web development
- Azure development
- Data storage and processing
- .NET Core cross-platform development
- Individual components
  - ☑ Visual Studio C++ core features
  - ☑ PowerShell Tools for Visual Studio 2017
  - ☑ ReadyRoll for VS2017
  - ☑ SQL Prompt Core
  - ☑ TypeScript 2.3 SDK
  - ☑ Arduino IDE for Visual Studio
  - ☑ PowerShell Pro Tools for Visual Studio 2017
  - ☑ Windows Template Studio
  - ☑ VS Live Share - Preview

Location

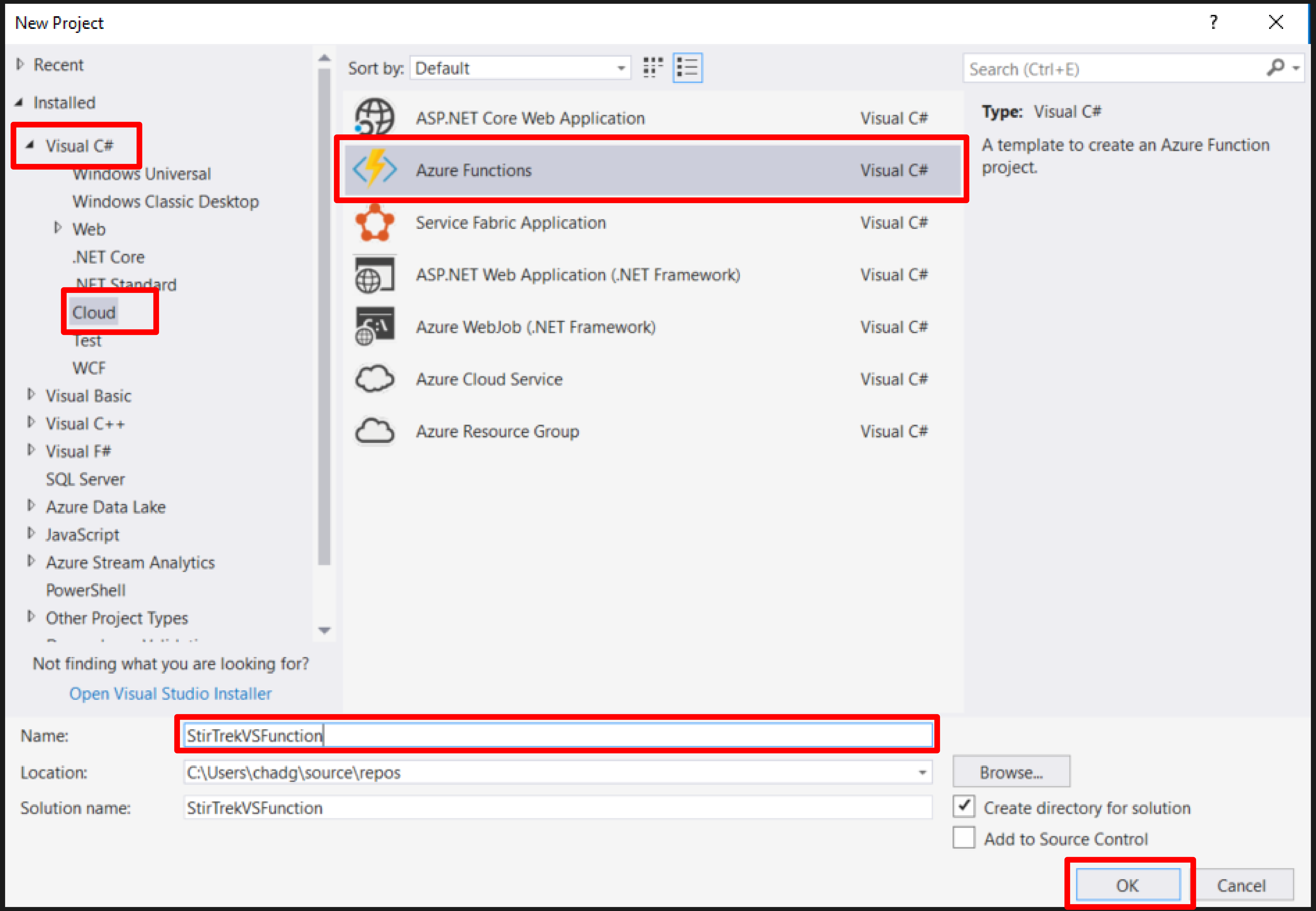
c:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total install size: 0 KB

Demo 3: Create an Azure Function from Visual Studio

Create a function app project


New Project
?
Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual C++
Visual F#
SQL Server
Azure Data Lake
JavaScript
Azure Stream Analytics
PowerShell
Other Project Types
Not finding what you are looking for?
Open Visual Studio Installer
Sort by: Default
Search (Ctrl+E)
ASP.NET Core Web Application Visual C#
Azure Functions Visual C#
Service Fabric Application Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
Azure WebJob (.NET Framework) Visual C#
Azure Cloud Service Visual C#
Azure Resource Group Visual C#
Browse...
OK
Cancel

Type: Visual C#

A template to create an Azure Function project.

Name: StirTrekVSFunction

Location: C:\Users\chadg\source\repos

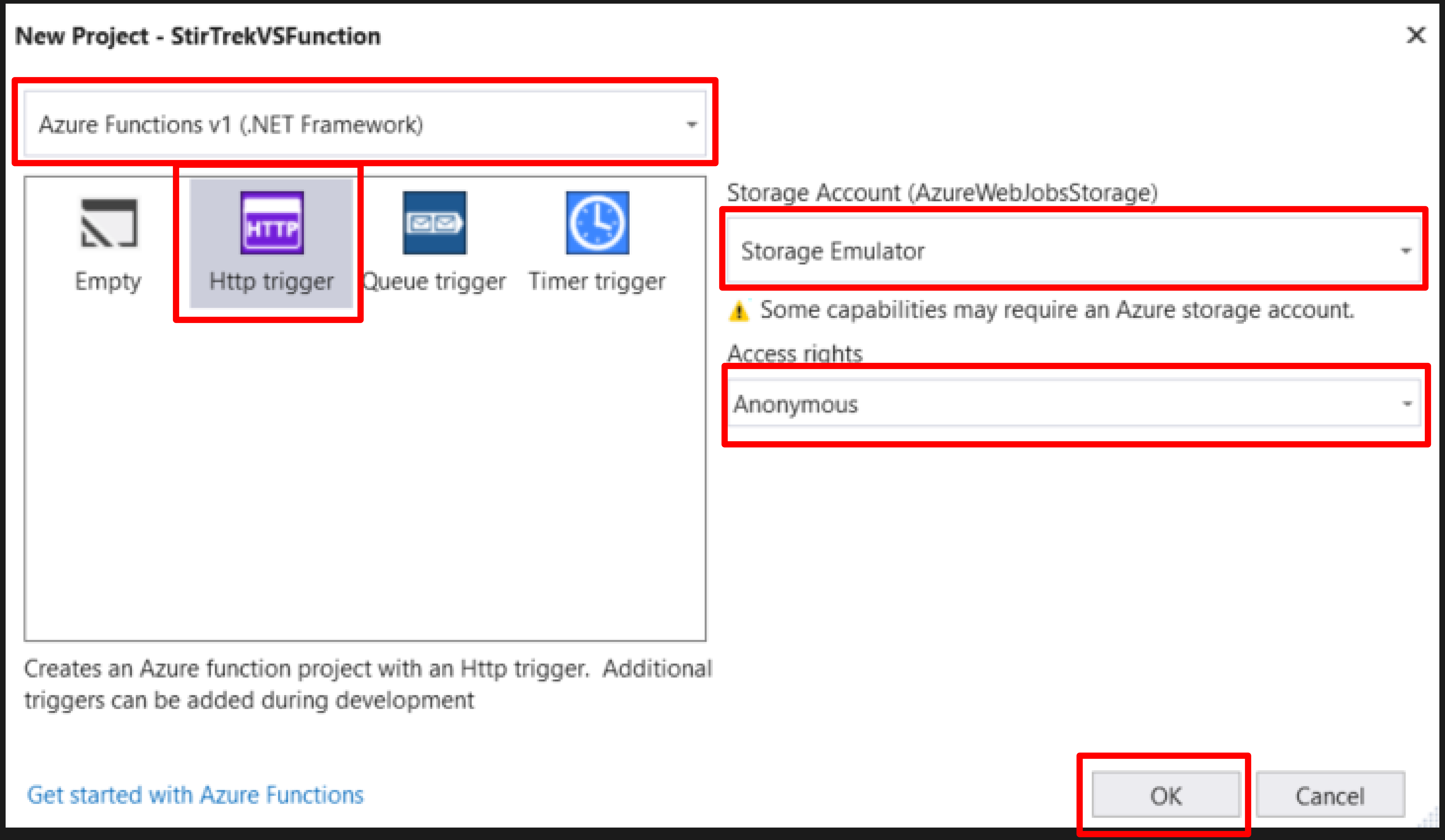
Solution name: StirTrekVSFunction

☑ Create directory for solution

□ Add to Source Control

Demo 3: Create an Azure Function from Visual Studio

Create a function app project



New Project - StirTrekVSFunction

Azure Functions v1 (.NET Framework)

Creates an Azure function project with an Http trigger. Additional triggers can be added during development

Storage Account (AzureWebJobsStorage): Storage Emulator

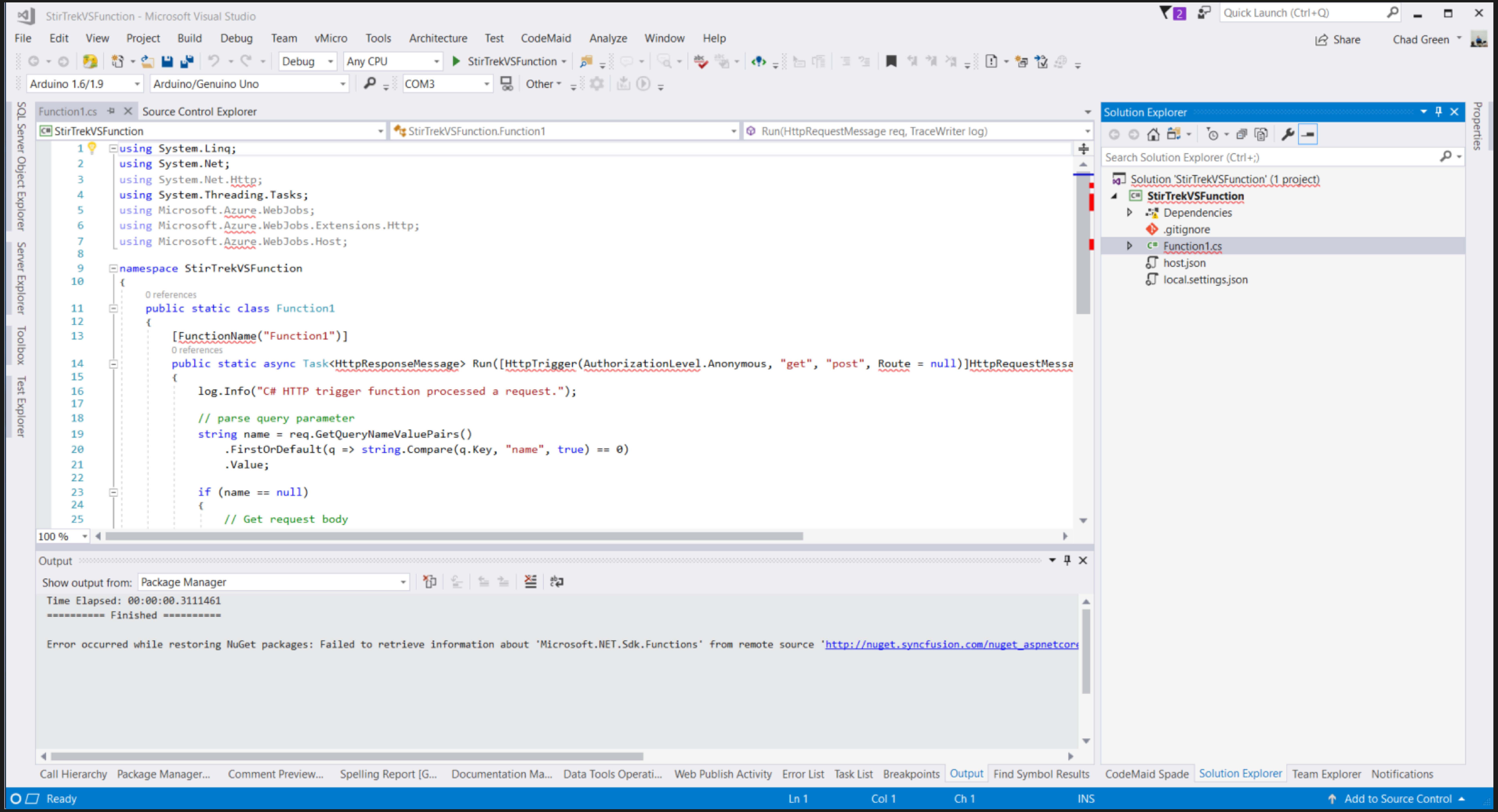
⚠ Some capabilities may require an Azure storage account.

Access rights: Anonymous

Get started with Azure Functions

Demo 3: Create an Azure Function from Visual Studio

Create a function app project

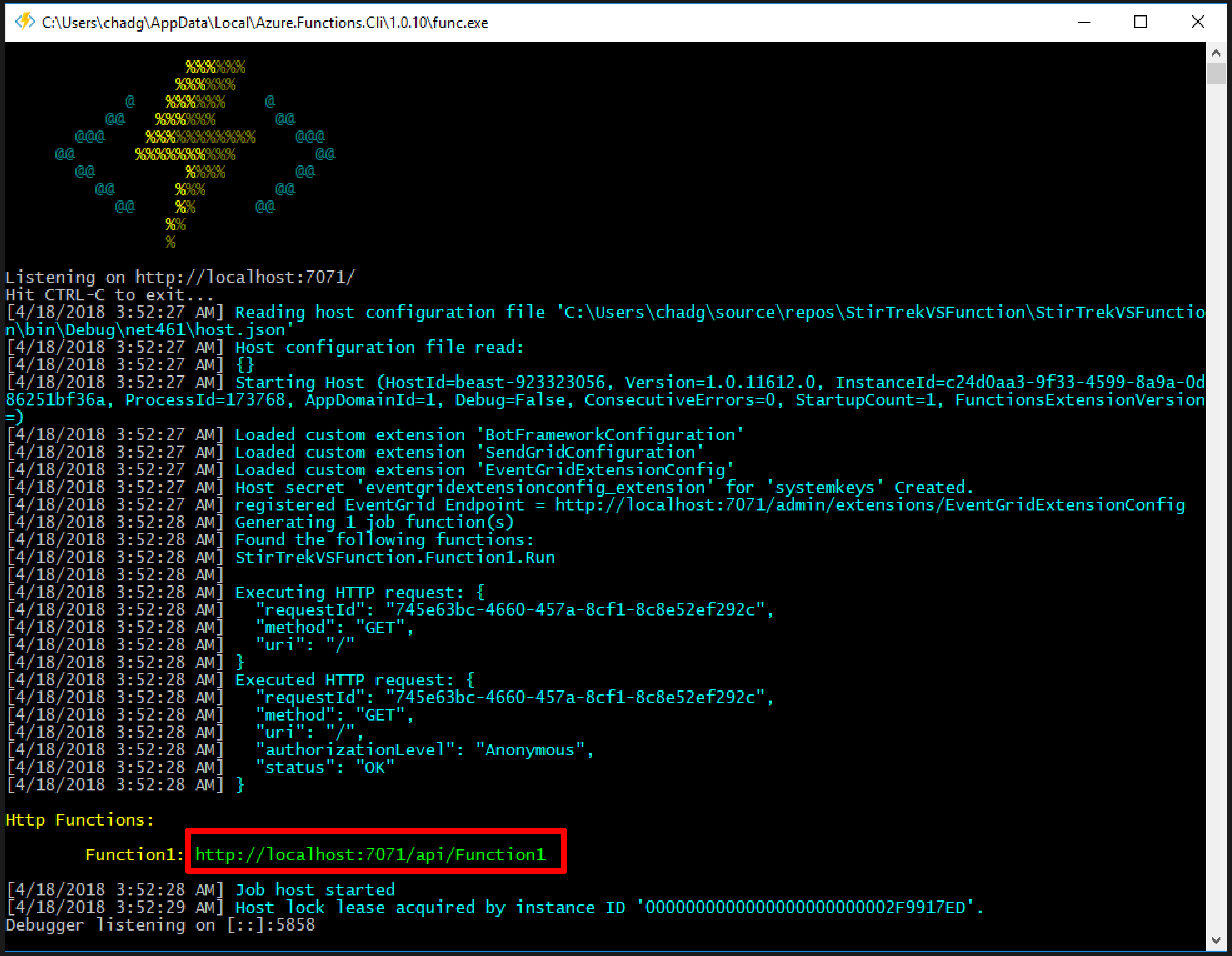


```csharp
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Threading.Tasks;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Extensions.Http;
using Microsoft.Azure.WebJobs.Host;

namespace StirTrekVSFunction
{
    0 references
    public static class Function1
    {
        [FunctionName("Function1")]
        0 references
        public static async Task<HttpResponseMessage> Run([HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)]HttpRequestMessa
        {
            log.Info("C# HTTP trigger function processed a request.");

            // parse query parameter
            string name = req.GetQueryNameValuePairs()
                .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
                .Value;

            if (name == null)
            {
                // Get request body
```

```
Time Elapsed: 00:00:00.3111461
========== Finished ==========

Error occurred while restoring NuGet packages: Failed to retrieve information about 'Microsoft.NET.Sdk.Functions' from remote source 'http://nuget.syncfusion.com/nuget_aspnetcore
```

Demo 3: Create an Azure Function from Visual Studio

Test the function locally

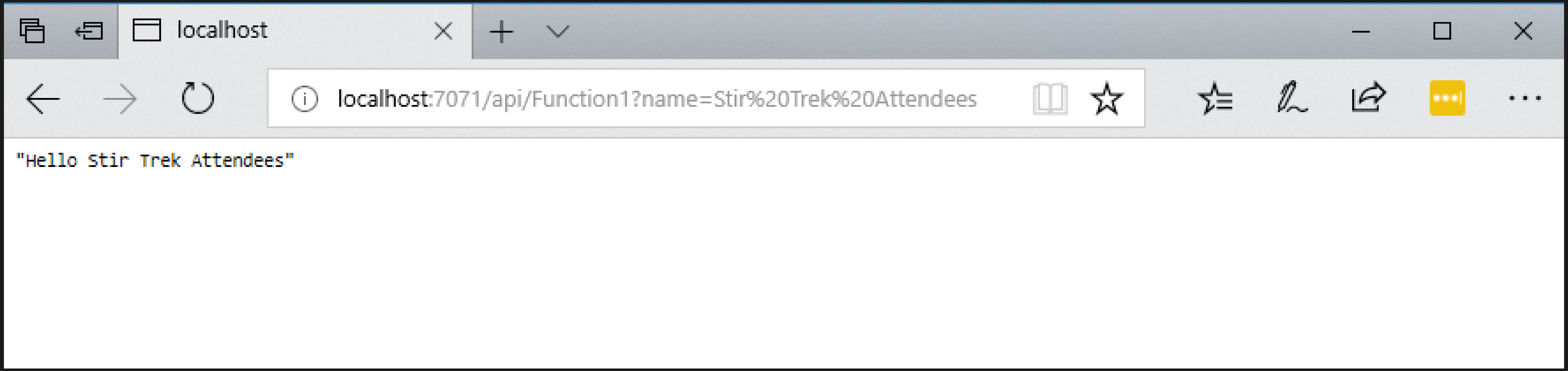


C:\Users\chadg\AppData\Local\Azure.Functions.Cli\1.0.10\func.exe

```
                  %%%%%%
                 %%%%%%
            @   %%%%%%    @
          @@   %%%%%%      @@
       @@@    %%%%%%%%%%%%   @@@
     @@      %%%%%%%%%%        @@
       @@         %%%%       @@
         @@      %%%       @@
           @@    %%      @@
                %%
                %

Listening on http://localhost:7071/
Hit CTRL-C to exit...
[4/18/2018 3:52:27 AM] Reading host configuration file 'C:\Users\chadg\source\repos\StirTrekVSFunction\StirTrekVSFunctio
n\bin\Debug\net461\host.json'
[4/18/2018 3:52:27 AM] Host configuration file read:
[4/18/2018 3:52:27 AM] {}
[4/18/2018 3:52:27 AM] Starting Host (HostId=beast-923323056, Version=1.0.11612.0, InstanceId=c24d0aa3-9f33-4599-8a9a-0d
86251bf36a, ProcessId=173768, AppDomainId=1, Debug=False, ConsecutiveErrors=0, StartupCount=1, FunctionsExtensionVersion
=)
[4/18/2018 3:52:27 AM] Loaded custom extension 'BotFrameworkConfiguration'
[4/18/2018 3:52:27 AM] Loaded custom extension 'SendGridConfiguration'
[4/18/2018 3:52:27 AM] Loaded custom extension 'EventGridExtensionConfig'
[4/18/2018 3:52:27 AM] Host secret 'eventgridextensionconfig_extension' for 'systemkeys' Created.
[4/18/2018 3:52:27 AM] registered EventGrid Endpoint = http://localhost:7071/admin/extensions/EventGridExtensionConfig
[4/18/2018 3:52:28 AM] Generating 1 job function(s)
[4/18/2018 3:52:28 AM] Found the following functions:
[4/18/2018 3:52:28 AM] StirTrekVSFunction.Function1.Run
[4/18/2018 3:52:28 AM]
[4/18/2018 3:52:28 AM] Executing HTTP request: {
[4/18/2018 3:52:28 AM]   "requestId": "745e63bc-4660-457a-8cf1-8c8e52ef292c",
[4/18/2018 3:52:28 AM]   "method": "GET",
[4/18/2018 3:52:28 AM]   "uri": "/"
[4/18/2018 3:52:28 AM] }
[4/18/2018 3:52:28 AM] Executed HTTP request: {
[4/18/2018 3:52:28 AM]   "requestId": "745e63bc-4660-457a-8cf1-8c8e52ef292c",
[4/18/2018 3:52:28 AM]   "method": "GET",
[4/18/2018 3:52:28 AM]   "uri": "/",
[4/18/2018 3:52:28 AM]   "authorizationLevel": "Anonymous",
[4/18/2018 3:52:28 AM]   "status": "OK"
[4/18/2018 3:52:28 AM] }

Http Functions:

        Function1: http://localhost:7071/api/Function1

[4/18/2018 3:52:28 AM] Job host started
[4/18/2018 3:52:29 AM] Host lock lease acquired by instance ID '0000000000000000000000002F9917ED'.
Debugger listening on [::]:5858
```

Demo 3: Create an Azure Function from Visual Studio

Test the function locally



localhost:7071/api/Function1?name=Stir%20Trek%20Attendees

```
"Hello Stir Trek Attendees"
```

Best Practices

From Zero to Serverless

Best Practices

- Functions *should* do one thing
- Functions *should* finish as quickly as possible

- Where to get started
  - Start small, replace 1 API or background processing item
  - Integration is a great place, often it’s a new layer on top of old layers

Summary

From Zero to Serverless

Summary

- With Azure Functions, the focus is on the code and not managing the infrastructure

- Serverless architecture provides the following benefits:
  - Reduced time to market
  - Lower total cost of ownership
  - Pay per execution

Azure Serverless Compute Summary

- It’s now becoming easier than ever to create small, targeted microservice/nanoservice architecture using a variety of services

- Azure provides many services that can help you achieve a low-friction, high-throughput and low-cost solution

- Azure Functions, Logic Apps, Event Grid are just a few in the serverless architecture family

Who is Chad Green

chadgreen@chadgreen.com

chadgreen.com

ChadGreen

ChadwickEGreen

bit.ly/LDN0418

BJ's ®
RESTAURANT
BREWHOUSE