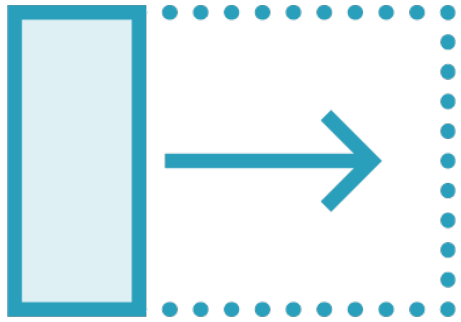
Azure Durable Functions Fundamentals

INTRODUCING DURABLE FUNCTIONS

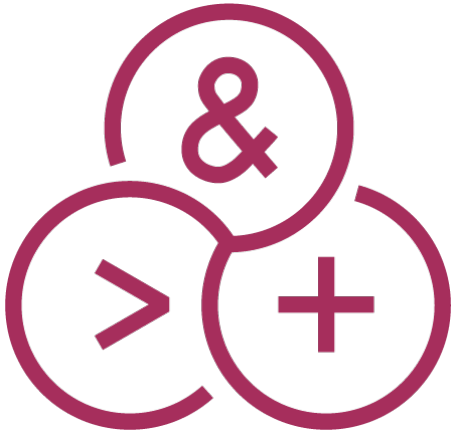
Mark Heath
MICROSOFT MVP

@mark_heath www.markheath.net

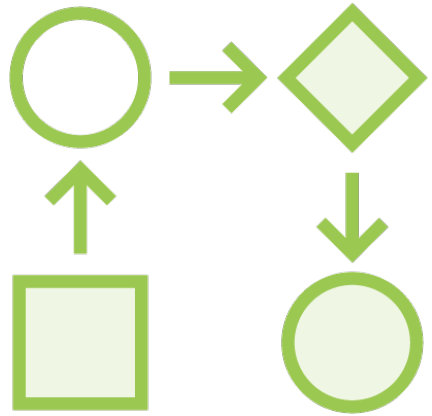
What Are “Durable Functions”?



An extension to Azure Functions

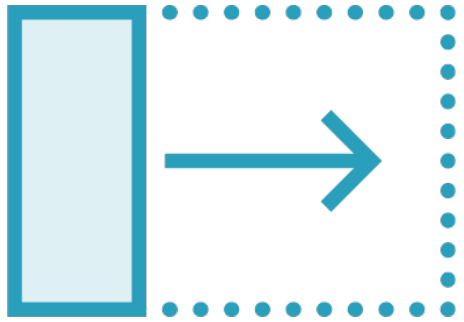


Write “stateful” functions in a “serverless” environment

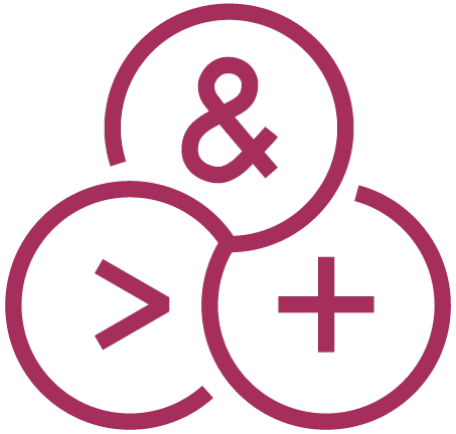


Define workflows in code

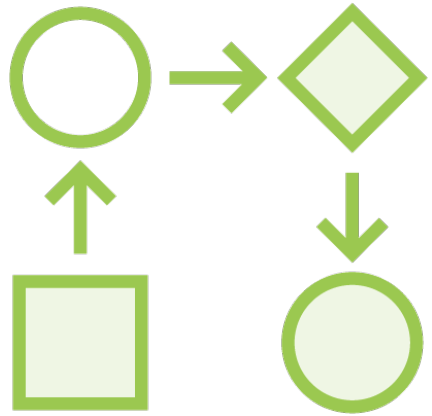
What Are "Durable Functions"?



An extension to
Azure Functions



Write "stateful"
functions in a
"serverless"
environment



Define workflows
in code

What Is Azure Functions?

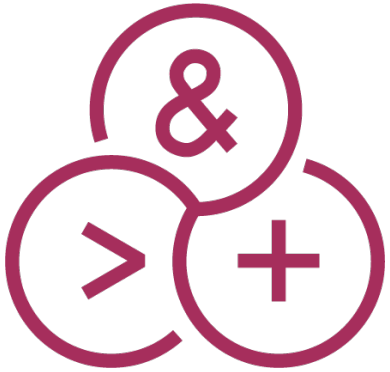
"Functions as a Service"
(FaaS)

Triggers

Timer

Queue Message

HTTP Request



Languages

C#, F#

JavaScript

Java

Bindings

Blob Storage

CosmosDb

SendGrid

Azure Functions Fundamentals

by Mark Heath

Discover how Azure Functions allows you to easily write serverless code in your language of preference to handle events at scale, with minimal overhead and cost.

Start Course | Bookmark | Add to Channel | Live mentoring

Table of contents | Description | Transcript | Exercise files | Discussion | Recommended

This course is part of: Microsoft Azure for Developers Path

Expand all

Course Overview 2m 3s

Course author

Mark Heath

Mark Heath is a software developer based in Southampton, England, working for NICE Systems as a software architect creating cloud based digital evidence management systems for the police. He is the...

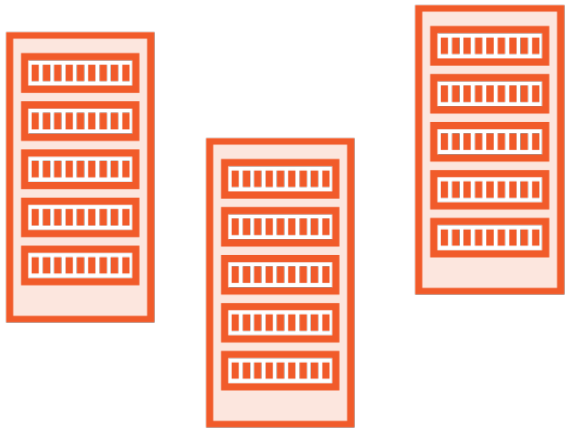
Course info

Level	Intermediate
Rating	★★★★★ (97)
My rating	★★★★★
Duration	3h 15m
Released	28 Jan 2017

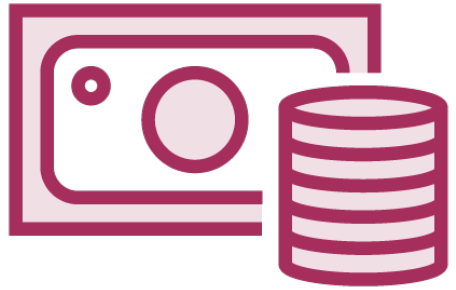
Share course

What Is “Serverless”?

Deploy without having to worry about infrastructure



Automatic scaling



Consumption-based pricing model

Building Serverless Applications in Azure

by Mark Heath

Over the years serverless has become a buzzword, but what does it look like to build via a serverless architecture? This course will teach you how to build serverless applications in Azure, from implementing web hosting to deployment and monitoring.

Start Course | Bookmark | Add to Channel | Live mentoring

Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check

Recommended

Expand all

Course author

Mark Heath

Mark Heath is a software developer based in Southampton, England, working for NICE Systems as a software architect creating cloud based digital evidence management systems for the police. He is the...

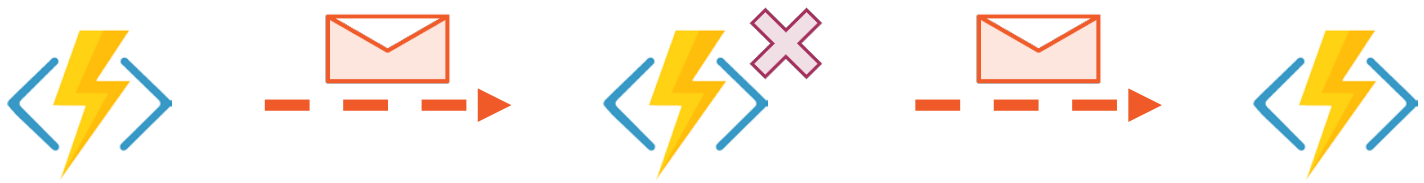
Course info

Level	Beginner
Rating	★★★★½ (19)
My rating	★★★★★
Duration	4h 7m
Released	15 Aug 2017

Share course

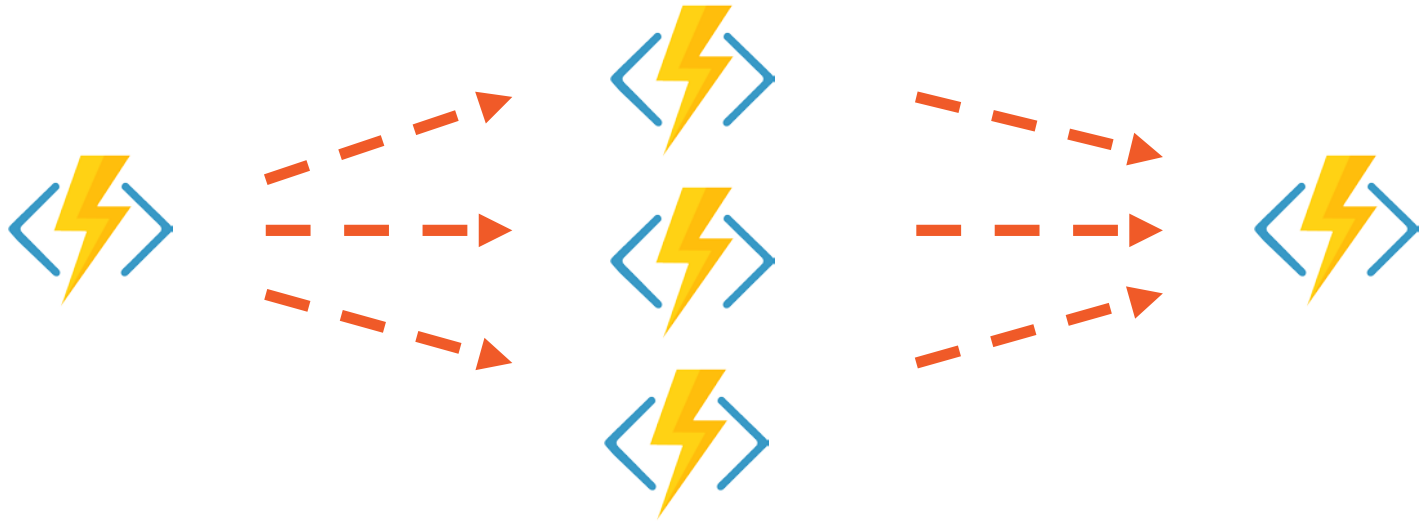
Why Durable Functions?

Function Chaining Workflow

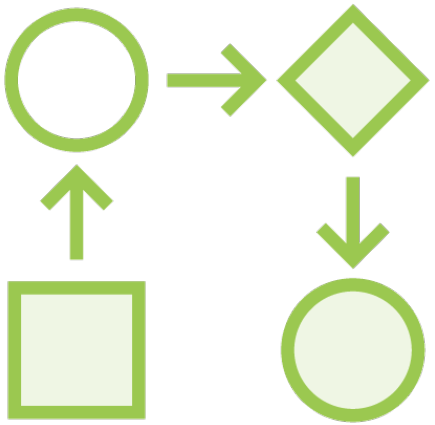


Function 1 Function 2 Function 3

Fan-out Fan-in Workflow



Durable Functions Basics



Define workflows in code

- Parallel execution
- Error handling
- Easily understood “Orchestrator Function”

Supports many workflow patterns

- Waiting for human interaction

Solves the state problem

- Tracks workflow progress

Durable Functions Benefits

1. **Define workflows in code**
   - Easy to understand the big picture
   - Good separation of concerns
2. **Easy to implement complex workflows**
   - Fan-out and fan-in
   - Wait for human interaction
3. **Consolidate exception handling**
4. **Check on progress or cancel workflows**
5. **Manage state for you**

Key Durable Function Concepts

Orchestrator functions

- Define the workflow
- Triggers “activity” functions
- Sleeps during activities

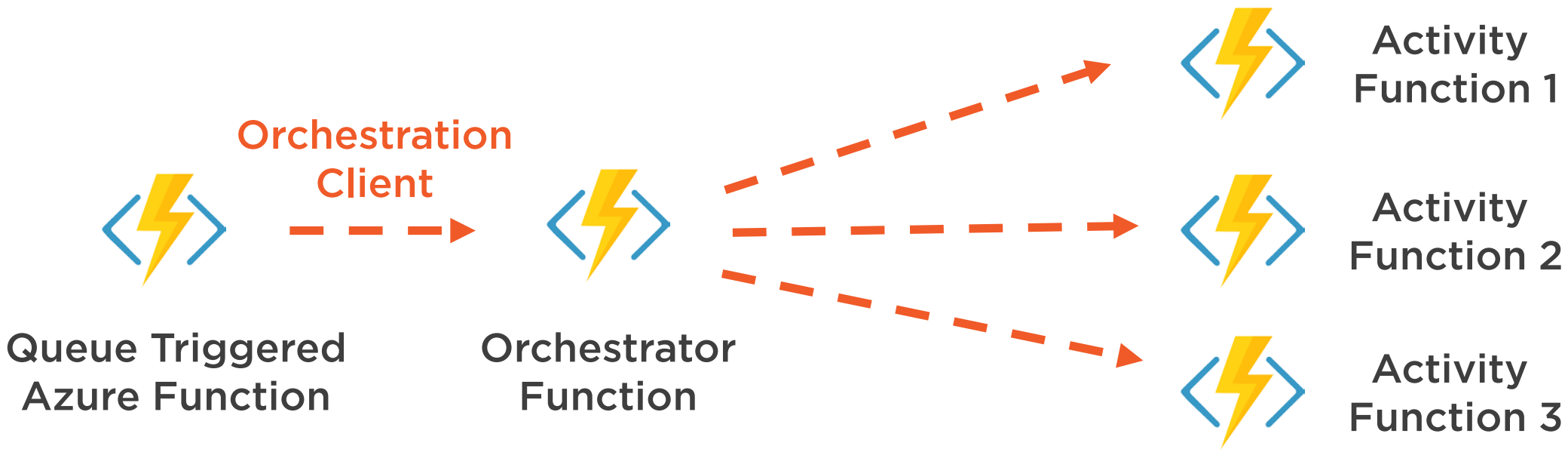
Activity functions

- Performs a single step in a workflow
- Can receive and return data

Starting “orchestrations”

- OrchestrationClient binding

Example Durable Function Workflow



Durable Functions State Storage

Durable Functions uses Azure Storage

Storage Queues

- Messages to trigger the next function

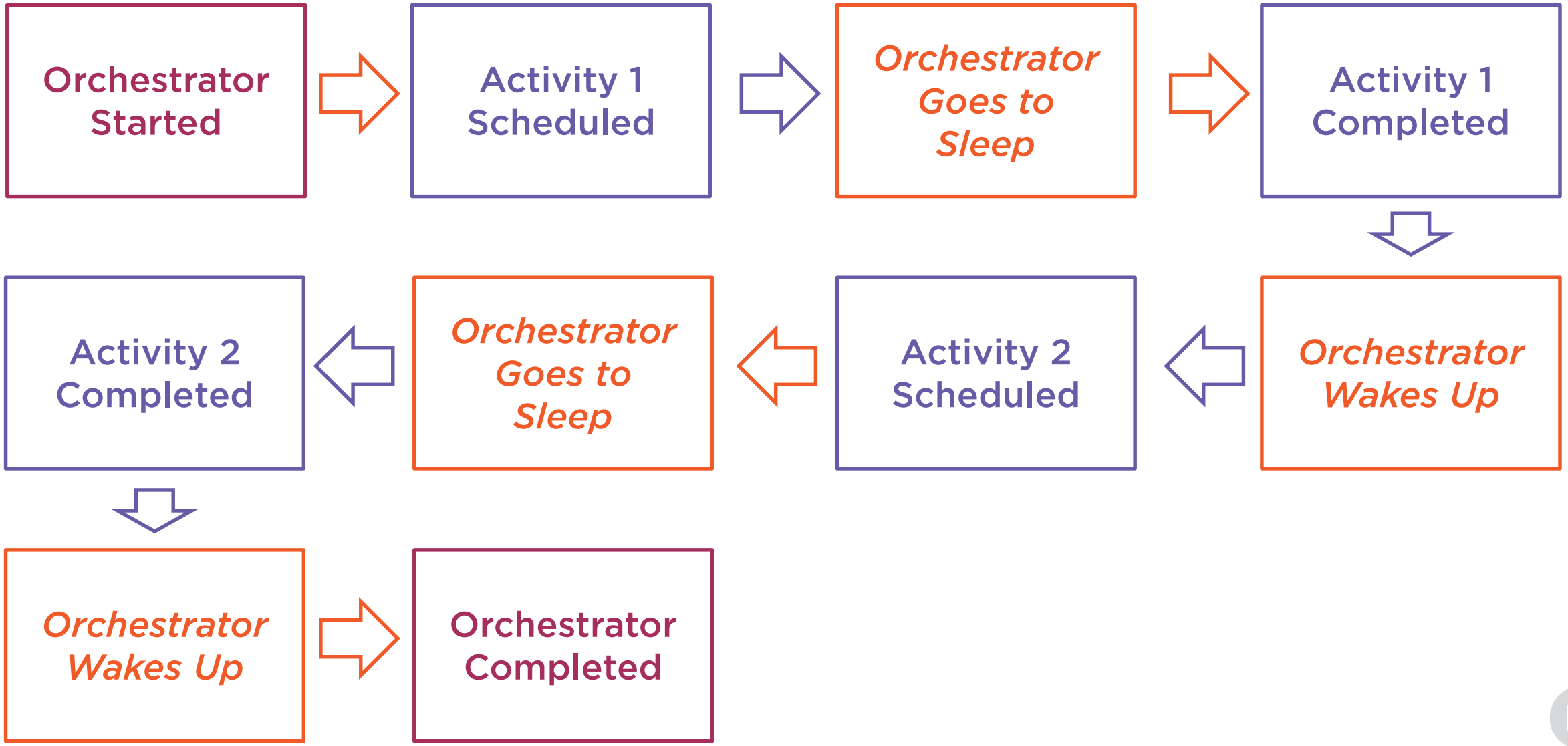
Storage Tables

- Store state of orchestrations

Event sourcing

- Never update rows, only append new ones
- Store full execution history

Event Sourcing Example



Durable Functions Storage Account

You provide the connection string

- Look inside with Azure Storage Explorer

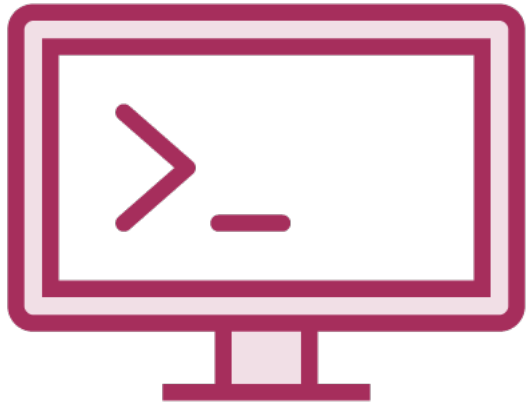
“Task Hub”

- The storage used by Durable Functions
- You can use multiple task hubs
- They can share a storage account

Developing Durable Functions

Azure Portal

Great for experimenting



Command Line

Great for cross platform

Visual Studio 2017

Rich development experience

https://mattchenderson.github.io/azure-durable-functions-azure-portal

Visual Studio | Visual Studio IDE | Features | Offerings | Downloads | Support | Subscriber Access | Free Visual Studio

Visual Studio Community

A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services.

Windows | macOS

Download VS Community 2017

Everything you need all in one place

Flexibility

Build apps for any platform

Productivity

Designers, editors, debuggers, profilers, in one single tool

Ecosystem

Access to thousands of extensions

Languages

Code in C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python, and more

https://www.visualstudio.com/vs/community/

Visual Studio Installer

Modifying — Visual Studio Enterprise 2017 — 15.6.6

Workloads | Individual components | Language packs | Installation locations

Windows (3)

- **Universal Windows Platform development** — Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++. [x]
- **.NET desktop development** — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#. [x]
- **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. [x]
- **Azure development** — Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including... [x]
- **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
- .NET Core cross-platform development *
- Individual components
  - NuGet package manager
  - C# and Visual Basic Roslyn compilers
  - Static analysis tools
  - C# and Visual Basic
  - .NET Framework 4.6.1 targeting pack
  - F# language support
  - PowerShell Tools for Visual Studio 2017
  - Testing tools core features

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise Change...

Total space required 4.31 GB

Modify

1.16.1214.412

Visual Studio Installer

Products

Installed

Visual Studio Enterprise 2017 Preview

15.7.0 Preview 4.0

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

Release notes

Modify Launch More

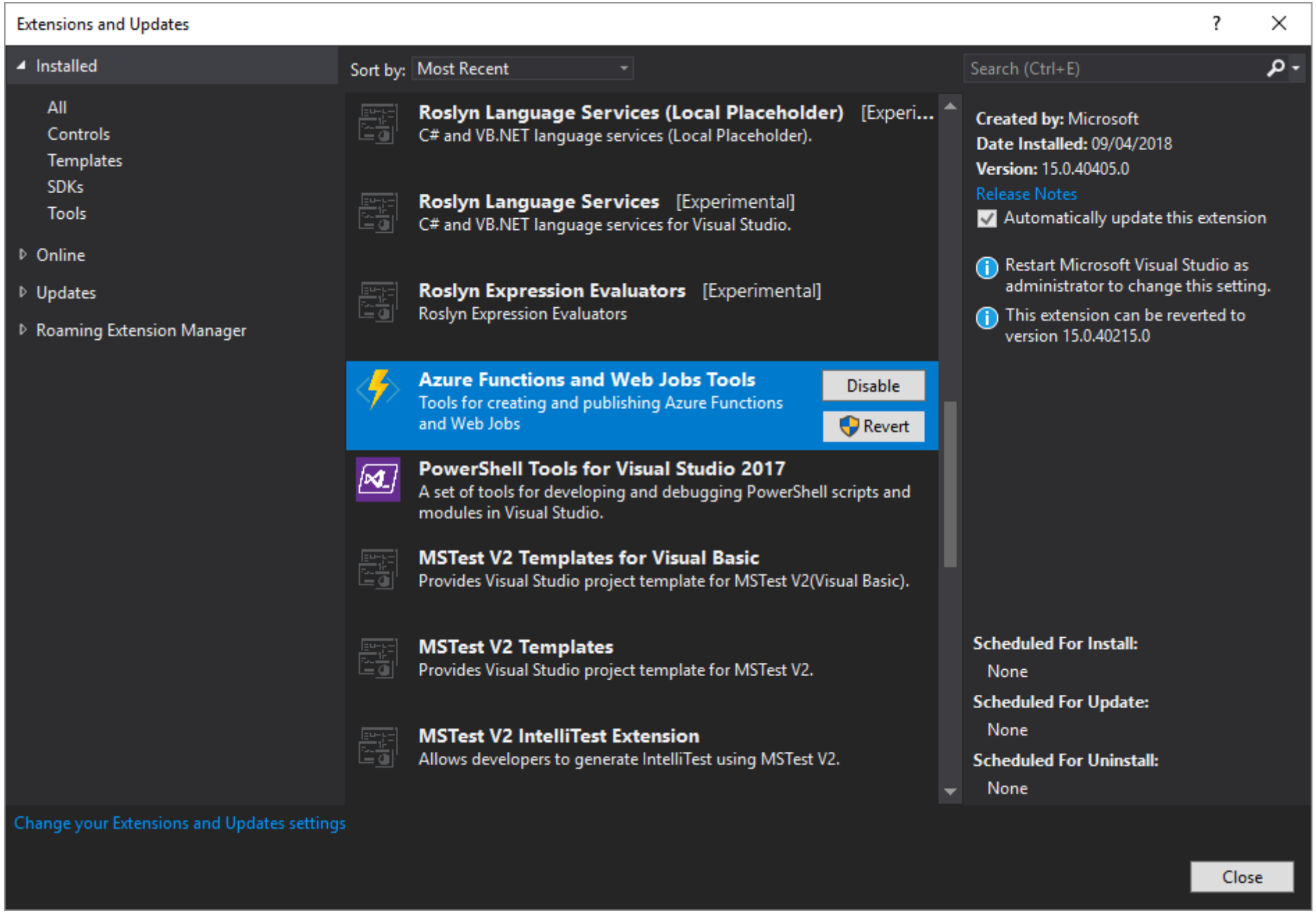
Visual Studio Enterprise 2017

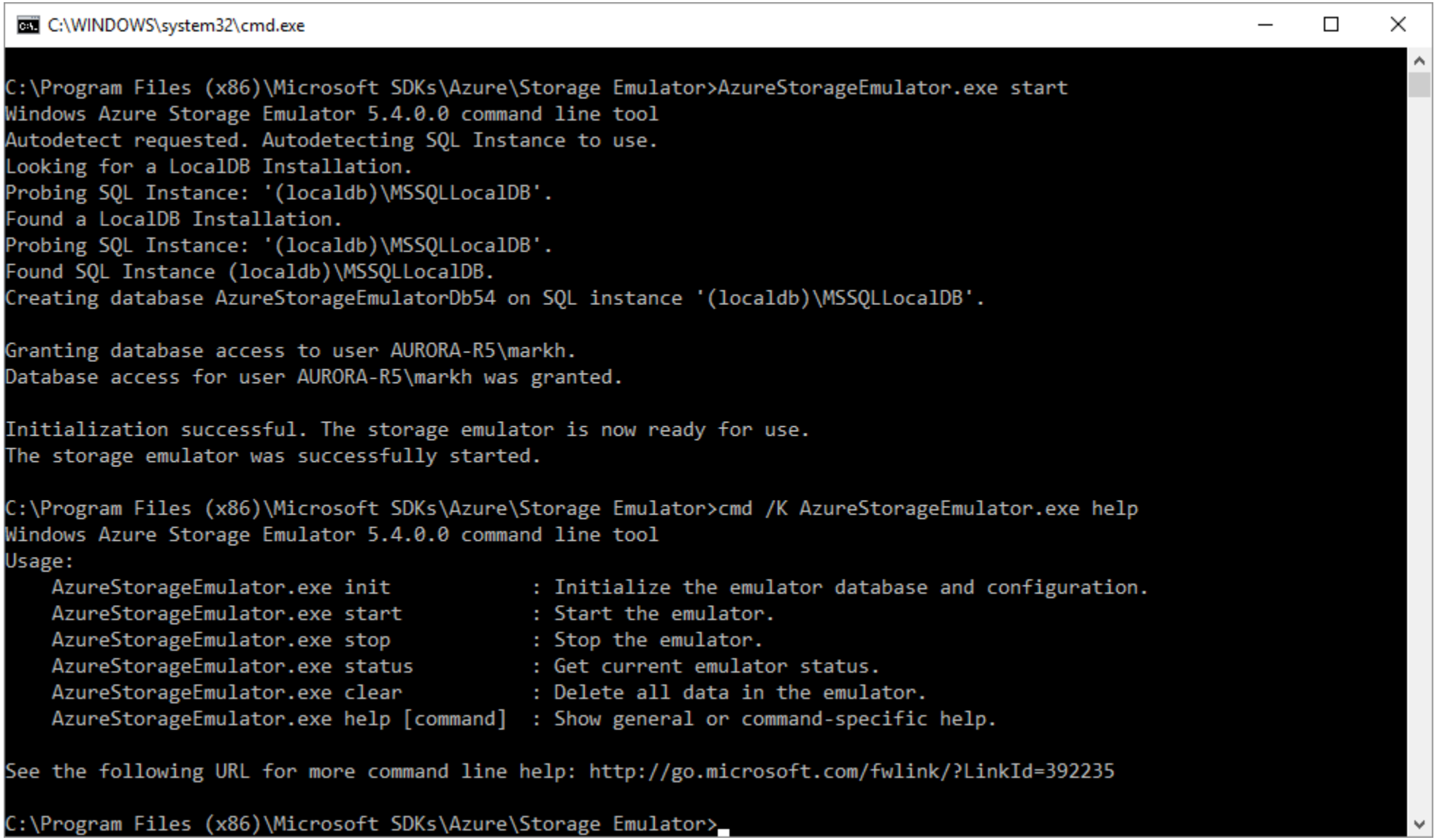
15.6.6

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

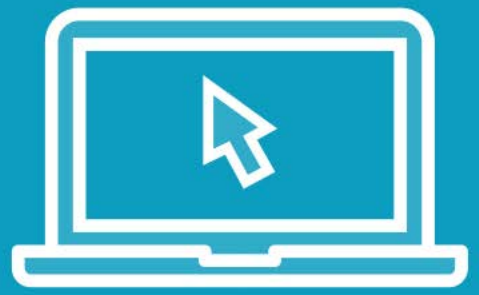
Release notes

Modify Launch More


Extensions and Updates
Installed
All
Controls
Templates
SDKs
Tools
Online
Updates
Roaming Extension Manager
Sort by: Most Recent
Search (Ctrl+E)
Roslyn Language Services (Local Placeholder) [Experi...
C# and VB.NET language services (Local Placeholder).
Roslyn Language Services [Experimental]
C# and VB.NET language services for Visual Studio.
Roslyn Expression Evaluators [Experimental]
Roslyn Expression Evaluators
Azure Functions and Web Jobs Tools
Tools for creating and publishing Azure Functions and Web Jobs
Disable
Revert
PowerShell Tools for Visual Studio 2017
A set of tools for developing and debugging PowerShell scripts and modules in Visual Studio.
MSTest V2 Templates for Visual Basic
Provides Visual Studio project template for MSTest V2(Visual Basic).
MSTest V2 Templates
Provides Visual Studio project template for MSTest V2.
MSTest V2 IntelliTest Extension
Allows developers to generate IntelliTest using MSTest V2.
Created by: Microsoft
Date Installed: 09/04/2018
Version: 15.0.40405.0
Release Notes
Automatically update this extension
Restart Microsoft Visual Studio as administrator to change this setting.
This extension can be reverted to version 15.0.40215.0
Scheduled For Install:
None
Scheduled For Update:
None
Scheduled For Uninstall:
None
Change your Extensions and Updates settings
Close


C:\WINDOWS\system32\cmd.exe
C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>AzureStorageEmulator.exe start
Windows Azure Storage Emulator 5.4.0.0 command line tool
Autodetect requested. Autodetecting SQL Instance to use.
Looking for a LocalDB Installation.
Probing SQL Instance: '(localdb)\MSSQLLocalDB'.
Found a LocalDB Installation.
Probing SQL Instance: '(localdb)\MSSQLLocalDB'.
Found SQL Instance (localdb)\MSSQLLocalDB.
Creating database AzureStorageEmulatorDb54 on SQL instance '(localdb)\MSSQLLocalDB'.
Granting database access to user AURORA-R5\markh.
Database access for user AURORA-R5\markh was granted.
Initialization successful. The storage emulator is now ready for use.
The storage emulator was successfully started.
C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>cmd /K AzureStorageEmulator.exe help
Windows Azure Storage Emulator 5.4.0.0 command line tool
Usage:
AzureStorageEmulator.exe init : Initialize the emulator database and configuration.
AzureStorageEmulator.exe start : Start the emulator.
AzureStorageEmulator.exe stop : Stop the emulator.
AzureStorageEmulator.exe status : Get current emulator status.
AzureStorageEmulator.exe clear : Delete all data in the emulator.
AzureStorageEmulator.exe help [command] : Show general or command-specific help.
See the following URL for more command line help: http://go.microsoft.com/fwlink/?LinkId=392235
C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>

Demo



Create an Azure Function App

Enable Durable Functions

Summary

Azure Functions

- Many languages, triggers & bindings

Serverless

- Automatic scaling
- Consumption based pricing

Durable Functions

- An Azure Functions extension
- Implement complex workflows
- Define workflows in code
- Check progress and cancel workflows

Summary

OrchestrationClient binding

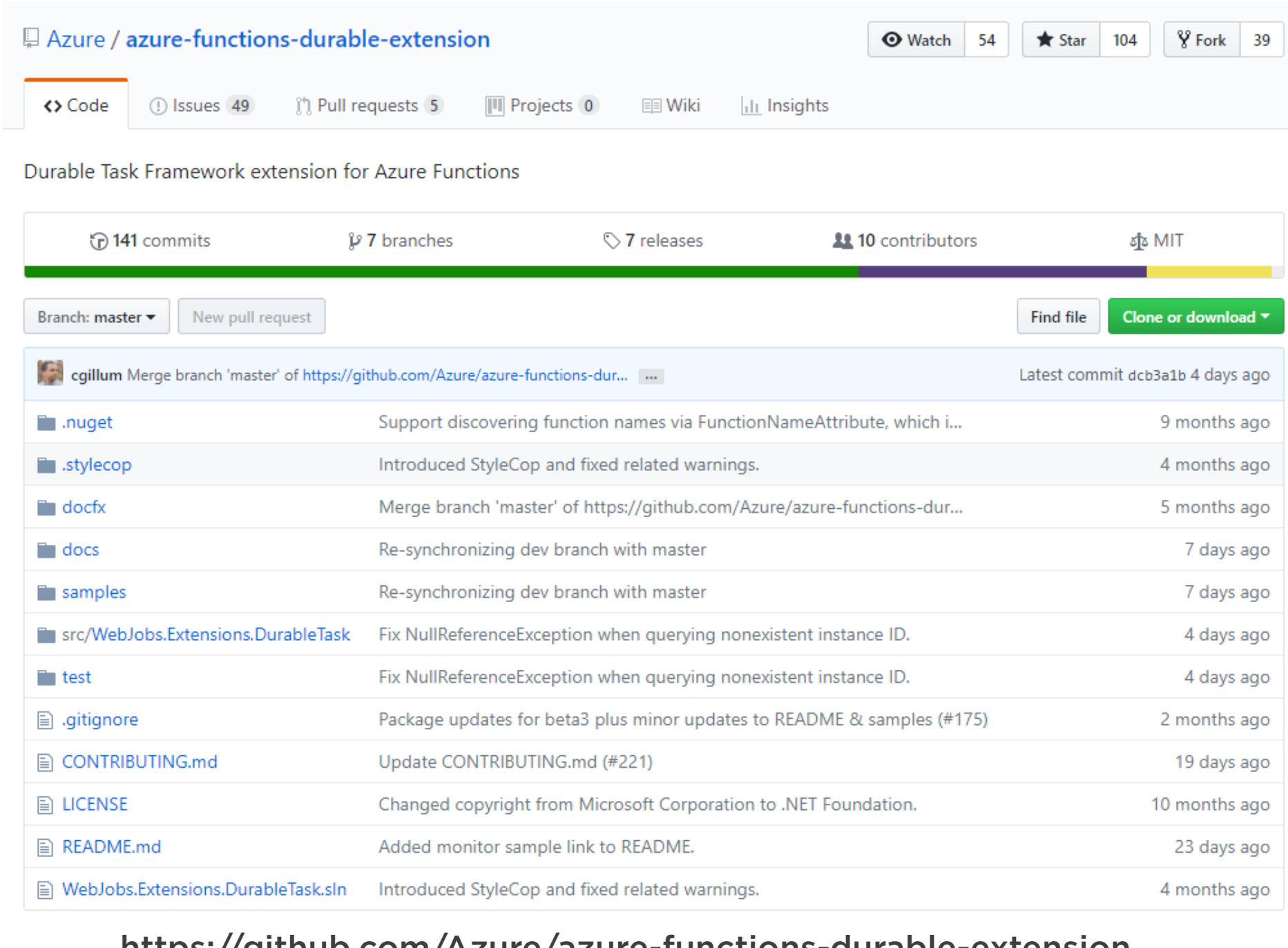
- Start new workflows

Orchestrator function

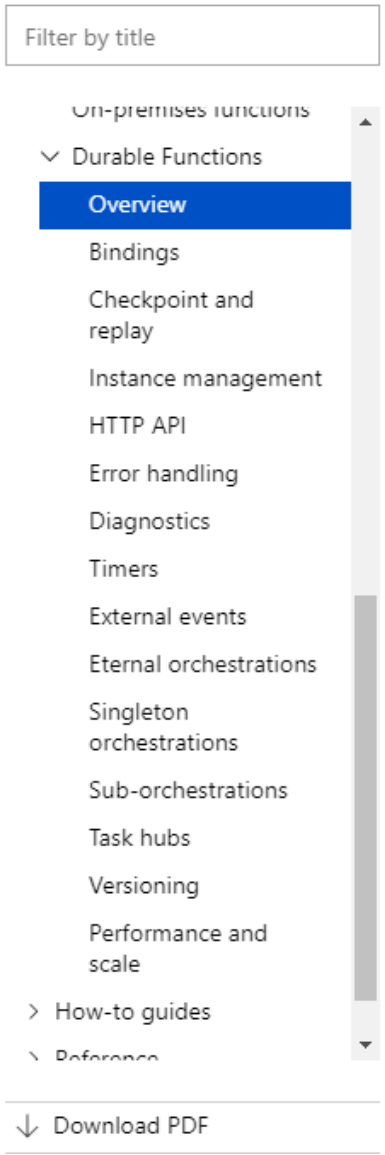
- Manage the workflow
- Calls activity functions
- Sleeps during activities
- Stores state in a “task hub”
- Wakes when activities complete

Development options

- Visual Studio 2017



https://github.com/Azure/azure-functions-durable-extension



Durable Functions overview (preview)

09/29/2017 • 13 minutes to read • Contributors

Durable Functions is an extension of Azure Functions and Azure WebJobs that lets you write stateful functions in a serverless environment. The extension manages state, checkpoints, and restarts for you.

The extension lets you define stateful workflows in a new type of function called an *orchestrator function*. Here are some of the advantages of orchestrator functions:

- They define workflows in code. No JSON schemas or designers are needed.
- They can call other functions synchronously and asynchronously. Output from called functions can be saved to local variables.
- They automatically checkpoint their progress whenever the function awaits. Local state is never lost if the process recycles or the VM reboots.

> **Note**
>
> Durable Functions is in preview and is an advanced extension for Azure Functions that is not appropriate for all applications. The rest of this article assumes that you have a strong familiarity with Azure Functions concepts and the challenges involved in serverless application development.

The primary use case for Durable Functions is simplifying complex, stateful coordination problems in serverless applications. The following sections describe some typical application patterns that can benefit from Durable Functions.

Pattern #1: Function chaining

https://docs.microsoft.com/en-us/azure/azure-functions/durable-functions-overview

Up next ...

Chaining multiple activities in a workflow