Introduction to Azure Functions

11/20/2020 • 2 minutes to read • +22

In this article

Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs. Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.

You focus on the pieces of code that matter most to you, and Azure Functions handles the rest.

We often build systems to react to a series of critical events. Whether you're building a web API, responding to database changes, processing IoT data streams, or even managing message queues - every application needs a way to run some code as these events occur.

To meet this need, Azure Functions provides "compute on-demand" in two significant ways.

First, Azure Functions allows you to implement your system's logic into readily available blocks of code. These code blocks are called "functions". Different functions can run anytime you need to respond to critical events.

Second, as requests increase, Azure Functions meets the demand with as many resources and function instances as necessary - but only while needed. As requests fall, any extra resources and application instances drop off automatically.

Where do all the compute resources come from? Azure Functions provides as many or as few compute resources as needed to meet your application's demand.

Providing compute resources on-demand is the essence of serverless computing in Azure Functions.

Scenarios

In many cases, a function integrates with an array of cloud services to provide feature-rich implementations.

The following are a common, *but by no means exhaustive*, set of scenarios for Azure Functions.

If you want to...	then...
Build a web API	Implement an endpoint for your web applications using the HTTP trigger
Process file uploads	Run code when a file is uploaded or changed in blob storage
Build a serverless workflow	Chain a series of functions together using durable functions
Respond to database changes	Run custom logic when a document is created or updated in Cosmos DB
Run scheduled tasks	Execute code on pre-defined timed intervals
Create reliable message queue systems	Process message queues using Queue Storage, Service Bus, or Event Hubs
Analyze IoT data streams	Collect and process data from IoT devices
Process data in real time	Use Functions and SignalR to respond to data in the moment

As you build your functions, you have the following options and resources available:

- **Use your preferred language**: Write functions in C#, Java, JavaScript, PowerShell, or Python, or use a custom handler to use virtually any other language.
- **Automate deployment**: From a tools-based approach to using external pipelines, there's a myriad of deployment options available.
- **Troubleshoot a function**: Use monitoring tools and testing strategies to gain insights into your apps.
- **Flexible pricing options**: With the Consumption plan, you only pay while your functions are running, while the Premium and App Service plans offer features for specialized needs.

Next Steps

Get started through lessons, samples, and interactive tutorials

Is this page helpful?

Yes No

Recommended content

Getting started with Azure Functions

Take the first steps toward working with Azure Functions.

Azure Functions pricing

Learn how billing works for Azure Functions.

Estimating Consumption plan costs in Azure Functions

Learn how to better estimate the costs that you may incur when running your function app in a Consumption plan in Azure.

Durable Functions Overview - Azure

Introduction to the Durable Functions extension for Azure Functions.

Show more