**Azure Functions**

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 code that matters most to you, in the most productive language for you, and Azure Functions handles the rest.

Functions provides a comprehensive set of event-driven [triggers and bindings](https://learn.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings) that connect your functions to other services without having to write extra code.

With Functions, you write your function code in your preferred language using your favorite development tools and then deploy your code to the Azure cloud. Functions provides native support for developing in [C#, Java, JavaScript, PowerShell, Python](https://learn.microsoft.com/en-us/azure/azure-functions/supported-languages), plus the ability to use [more languages](https://learn.microsoft.com/en-us/azure/azure-functions/functions-custom-handlers), such as Rust and Go.

**Scenarios**

We often build systems to react to a series of critical events. Whether you're building a web API, responding to database changes, processing event streams or messages, Azure Functions can be used to implement them.

**What are Durable functions**

Durable Functions is a feature of [Azure Functions](https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview) that lets you write stateful functions in a serverless compute environment. The **extension** lets you define stateful workflows by writing [**orchestrator functions**](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-orchestrations) and stateful entities by writing [entity functions](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-entities) using the Azure Functions programming model. Behind the scenes, the extension manages state, checkpoints, and restarts for you, allowing you to focus on your business logic.

The following sections describe typical application patterns that can benefit from Durable Functions:

* [Function chaining](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#chaining)
* [Fan-out/fan-in](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#fan-in-out)
* [Async HTTP APIs](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#async-http)
* [Monitoring](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#monitoring)
* [Human interaction](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#human)
* [Aggregator (stateful entities)](https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview?tabs=in-process%2Cnodejs-v3%2Cv1-model&pivots=csharp#aggregator)

**Function chaining**

In the function chaining pattern, a sequence of functions executes in a specific order. In this pattern, **the output of one function is applied to the input of another**

**function**. The use of queues between each function ensures that the system stays durable and scalable, even though there is a flow of control from one function to the next.

**Fan out/fan in**

In the fan out/fan in pattern, you execute multiple functions in parallel and then wait for all functions to finish. Often, some aggregation work is done on the results that are returned from the functions.

**Prerequisites**

* Make sure that you have the latest version of the [Azure Functions Core Tools](https://learn.microsoft.com/en-us/azure/azure-functions/functions-run-local).
* Durable Functions require an Azure storage account. You need an Azure subscription.
* Make sure that you have version 3.1 or a later version of the [.NET Core SDK](https://dotnet.microsoft.com/download) installed.