Azure Functions

Week 05

Information Processing Techniques

Murtaza Munawar Fazal

Infrastructure as a Service (IaaS)

It provides only a base infrastructure (Virtual machine, Software Define Network, Storage attached). End user have to configure and manage platform and environment, deploy applications on it.

AWS (EC2), GCP (CE), Microsoft Azure (VM) are examples of Iaas.

Software as a Service (SaaS)

It is sometimes called to as “on-demand software”. Typically accessed by users using a thin client via a web browser. In SaaS everything can be managed by vendors: applications, runtime, data, middleware, OSes, virtualization, servers, storage and networking, End users have to use it.

GMAIL is Best example of SaaS. Google team managing everything just we have to use the application through any of client or in browsers. Other examples **SAP**, **Salesforce** .

Platform as a Service (PaaS)

It provides a platform allowing end user to develop, run, and manage applications without the complexity of building and maintaining the infrastructure.

Google App Engine, CloudFoundry, Heroku, AWS (Beanstalk) are some examples of PaaS.

Container as a Service (CaaS)

Is a form of container-based virtualization in which container engines, orchestration and the underlying compute resources are delivered to users as a service from a cloud provider.

Google Container Engine(GKE), AWS (ECS), Azure (ACS) and Pivotal (PKS) are some examples of CaaS.

Function as a Service (FaaS):

It provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure.

AWS (Lamda), Google Cloud Function, Azure Functions are some examples of FaaS

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

Azure Functions

Events

React to timers, HTTP, or events from your favorite Azure services, with more on the way

Code

Author functions in C#, F#, Node.JS, Java, and more

Outputs

Send results to an ever-growing collection of services

Azure Functions can be used…

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 at set times
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

Supported bindings

The following table shows the bindings that are supported in the two major versions of the Azure Functions runtime.

Type	1.x	2.x[1]	Trigger	Input	Output
Blob Storage	✓	✓	✓	✓	✓
Cosmos DB	✓	✓	✓	✓	✓
Event Grid	✓	✓	✓		
Event Hubs	✓	✓	✓		✓
External File[2]	✓			✓	✓
External Table[2]	✓			✓	✓
HTTP	✓	✓	✓		✓
Microsoft Graph Excel tables		✓		✓	✓
Microsoft Graph OneDrive files		✓		✓	✓
Microsoft Graph Outlook email		✓			✓

Type	1.x	2.x[1]	Trigger	Input	Output
Microsoft Graph Events		✓	✓	✓	✓
Microsoft Graph Auth tokens		✓		✓	
Mobile Apps	✓			✓	✓
Notification Hubs	✓				✓
Queue storage	✓	✓	✓		✓
SendGrid	✓	✓			✓
Service Bus	✓	✓	✓		✓
Table storage	✓	✓		✓	✓
Timer	✓	✓	✓		
Twilio	✓	✓			✓
Webhooks	✓		✓		✓

[1] In 2.x, all bindings except HTTP and Timer must be registered. See Register binding extensions.

[2] Experimental — not supported and might be abandoned in the future.

Azure Functions

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.

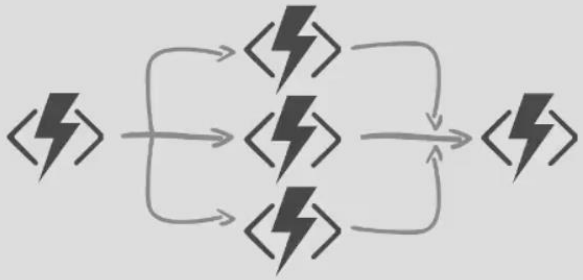
Troubleshoot a function: Use monitoring tools and testing strategies to gain insights into your apps.

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.

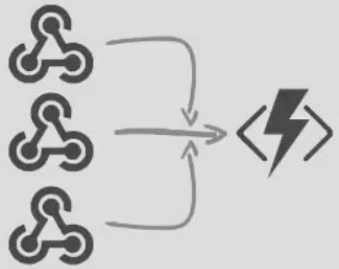
Durable Functions

What's still hard?

Manageable Sequencing
+ Error Handling / Compensation

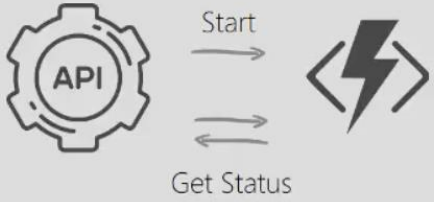


Fanning-out & Fanning-in



External Events Correlation

Flexible Automated Long-running
Process Monitoring



Http-based
Async Long-running APIs

Human Interaction