Azure Function #1

Hello Jonah, we often hear about Functions, can you tell us what it is?

Sure, Jules

Azure Function is a **fully-managed serverless** compute service on Microsoft Azure that allows you to write **event-driven solutions** in the Cloud. For example, an image uploaded to an Azure Blob storage or a new data added to an Azure Cosmos DB database.

What exactly is **serverless**?

Serverless computing is a method where a cloud provider, like Azure, provides backend services on a consumption (Pay-As-You-Go) pricing model. This means developers working with serverless development **gain more productivity** buy focusing on writing the logic in the code instead of managing and maintaining the infrastructure and servers.

Interesting!

Absolutely, and in addition **Azure Function** integrates perfectly with other Azure services and event external APIs and providers. Function can be deployed on a **Linux or Windows environment**. Even **Containers**!

One of the advantages of Functions is t**he number of supported languages and Frameworks**: .Net, Node.js, Java, PowerShell, Python, ...

Today, 2 deployment modes are available:

The **Code** mode, we just need to choose a Framework and the version.

Container mode which allows you to instantiate a docker container with the image of your choice.

I really like the principle of **serverless** which takes up more and more space.

next

Azure Function #2

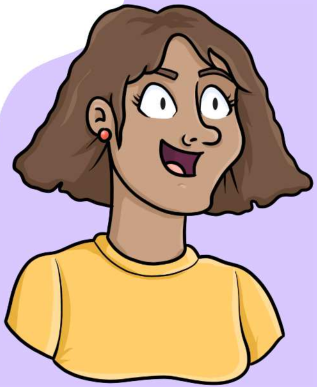
But then, where do you store your code?

Excellent question.

The **code**, but also the configuration of your Function is saved in a **storage account**, which offers **persistent storage**.

Basically, the standard Azure Function is stateless which is not ideal if you want to create a serverless workflow that requires to keep its state.

But there is an extension, **Durable Function**, which allows you to write Functions while managing the state (**statefull**), very useful when you have a workflow of functions that are complex. For example, function chaining, waiting for an external event, or a human-interaction, etc.



As there are 2 deployment modes, how are the associated resources managed?

The resources depend on the hosting plan you choose:

The **Consumption** or **Serverless** plan where the resources are provided directly by Microsoft, very useful when your needs are not systematic or very short in duration. In addition, you only pay for the resources consumed during execution.

The **Dedicated Plan**, where you provision a pool of CPU and Memory resources, as for WebApps. This is handy when executions are more frequent with longer execution times. Useful when you want to run your functions within an App Service Plan or need predictive scaling and cost management.

And finally, the **Premium plan**, where more powerful instances are provided by Microsoft. This is especially useful when Functions run continuously or almost continuously, and you benefit from additional functionality, like VNET integration.

next

Azure Function #3

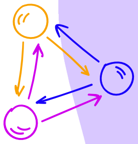
And the billing will depend on the plan you choose?

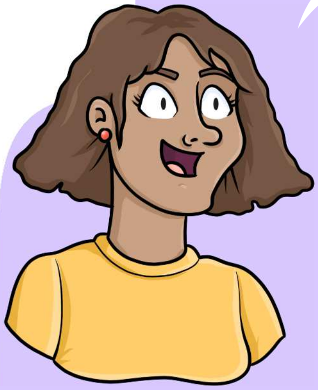
Yes, that's why you have to study well and select the one that best suits your **use case.**

Keep in mind that the Consumption plan is not supported for Container mode.

You do well to point it out!

Regarding the development of Functions, you can do it on the IDE of your choice like Visual Studio but **Visual Studio Code** is perfect for this with the different extensions it offers and its cross-platform.





What about integration with other Azure services?

Azure Function is often used with Azure WebApp or with **Azure Logic Apps** which is also serverless.

A Function can be triggered in several ways. The 1st by a simple **HTTP/S request** on its URL, which makes integration with Azure services very simple, but also services hosted outside of Azure.

A Function can also be triggered at regular intervals via the configuration of a **schedule**. But also be triggered by **Event Grid** or a different service **queue**.

A simple integration with several services that supports several triggers, as well as different frameworks, the product team has done a great job!

This explains its **popularity** in the Azure ecosystem.

next

And in terms of **security**, I assume that Function relies on the services offered by Azure?

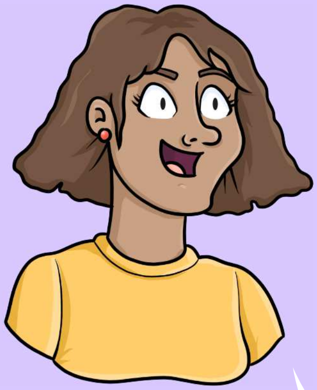
Obviously, it is perfectly integrated with the **Microsoft Defender for Cloud** solution.

Even if **Azure Function** manages HTTP requests, it is strongly recommended to favor **HTTPS requests**.

In addition, you can even **limit access** with **API keys**. Like in Azure Functions, you set your authentication type to the following: Anonymous, Function (API keys) and Admin which requires a master key.

And I guess there is no integration problem with **Azure Key Vault**?!

Exactly, which is very handy when you need to store sensitive information like API keys, passwords, certificates, connection strings, etc...



You can also integrate it with **Application Insight** to retrieve **logs**, **performance metrics** but also **errors**. In addition to this, you can also monitor the function app running through the Azure Monitor metrics.

With the **Premium plan**, you can even deploy the Function in multiple **Availability Zones** (**AZ**).

You made us discover a service that I will hasten to deepen, because it will respond to a multitude of use cases that we have with our customers.

La Minute CLOUD de JULES & LÉA

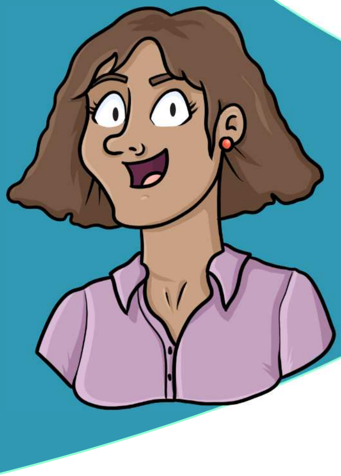
Thank you !

If you want to continue **learning** in a fun way about the **Azure ecosystem**, and not miss any of our illustrations ...

GROW UNA

... Feel free to subscribe on LinkedIn at:

https://aka.ms/grow-una



If you like our work, please share it ;o)

See you soon !