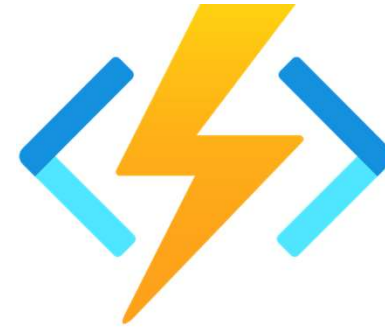




# Azure Function App

# Agenda



What are Function Apps and its usage?

What are the different triggers we can have in Function App?

Which are the Coding languages supported in function App?

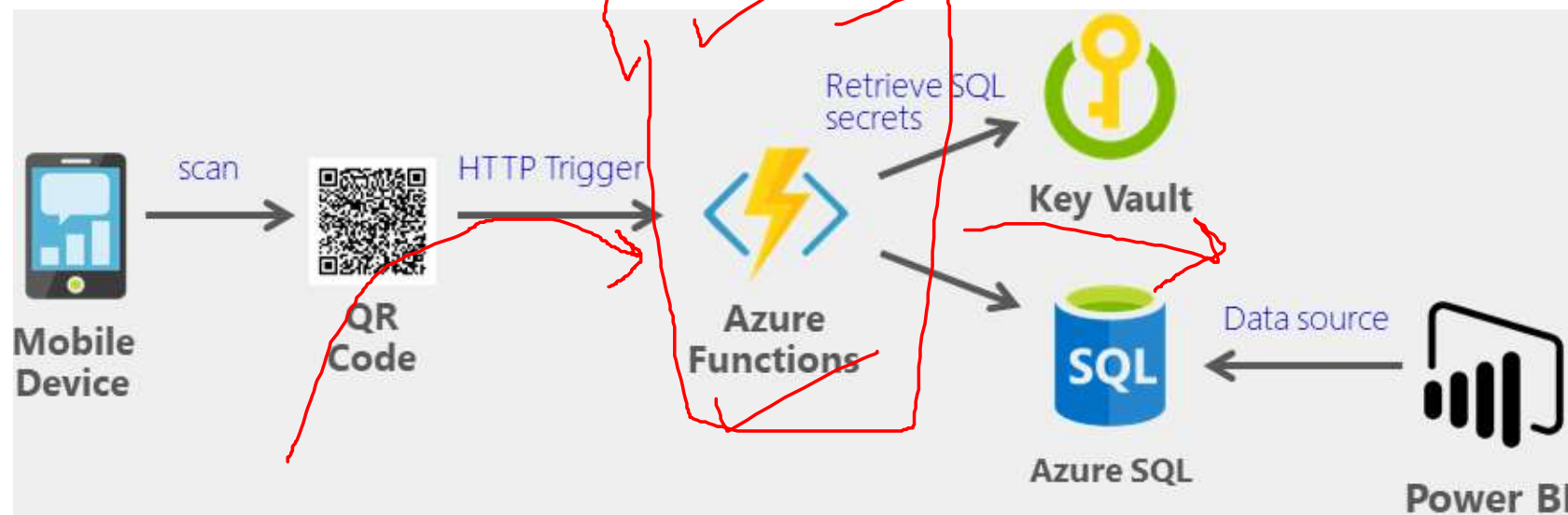
Pros and Cons of Function app when it comes of time limit/Data size

What are durable functions. Which scenarios it can be used?

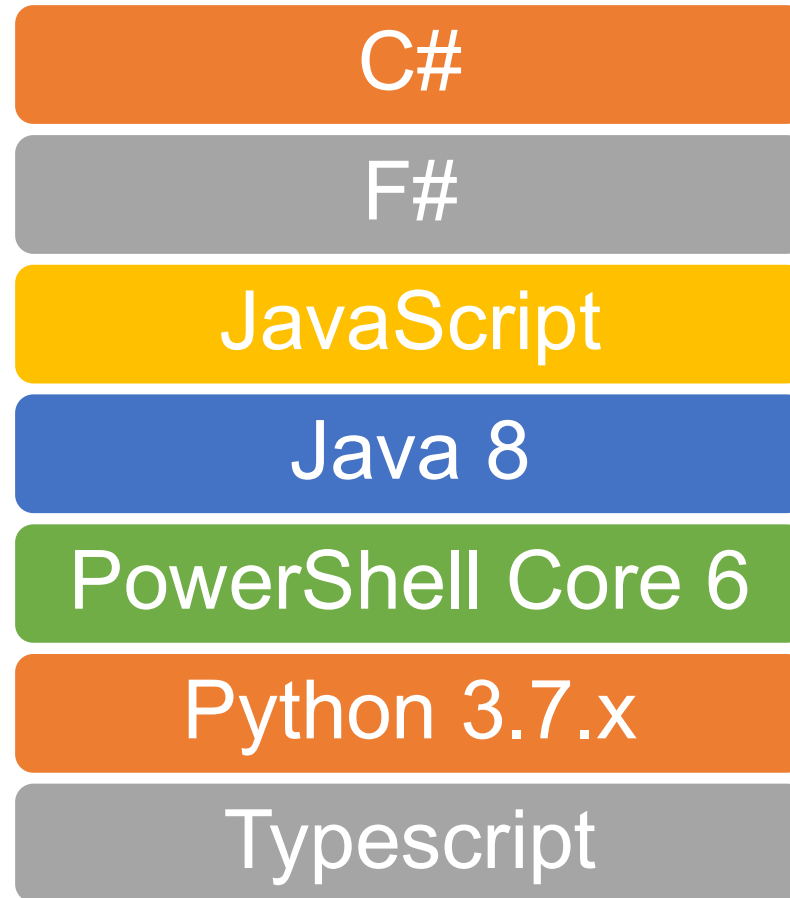
What are Kudu in Function App?

# What are Function Apps and its usage

- Can author and execute snippets of code in the cloud
- No hassle of managing the servers
- It lets you run small pieces of code
- Doesn't have to worry about the infrastructure of the platform



# Supported Languages



# Azure Functions Advantages

Pay as you go model

No infrastructure to maintain

No costs when functions aren't running

Supports variety of Languages

Easy Integration with Other Azure services

Trigger based executions

Bring your own dependencies

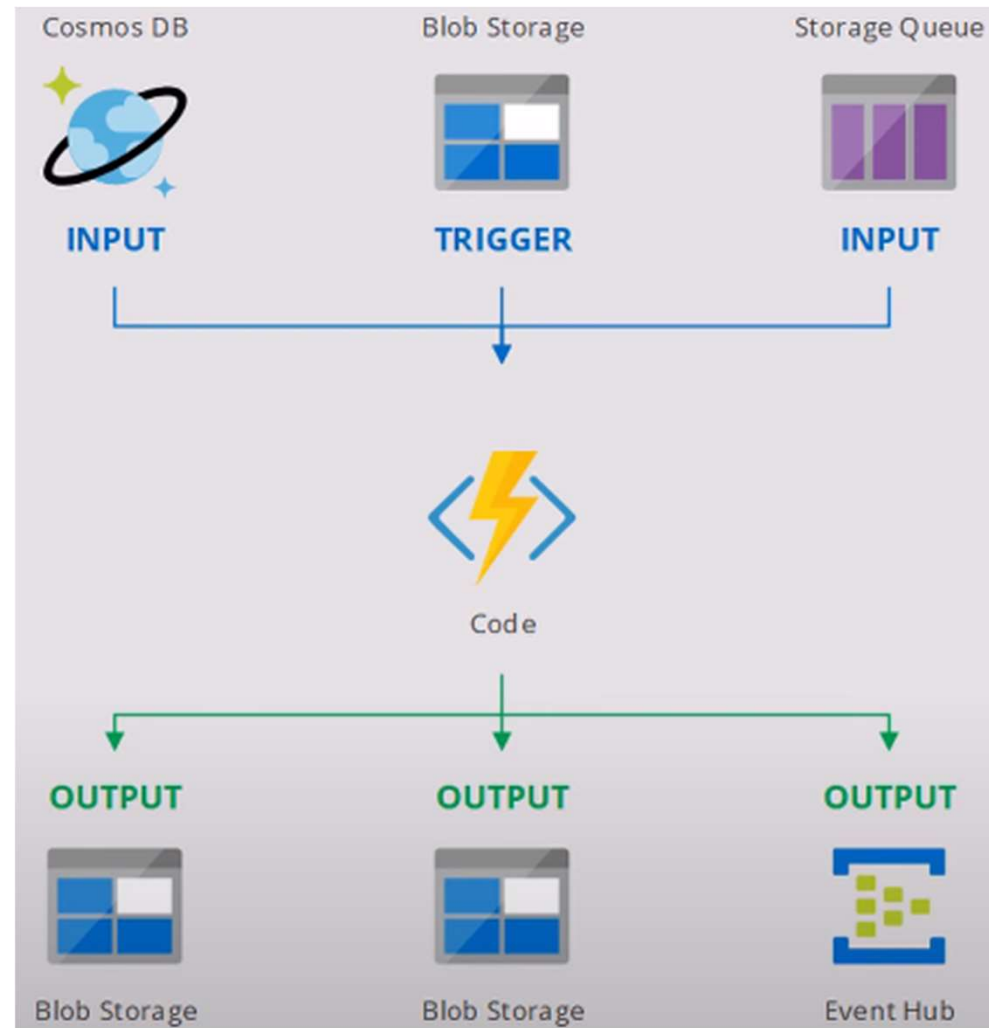
Infinitely scalable

Reduced latency

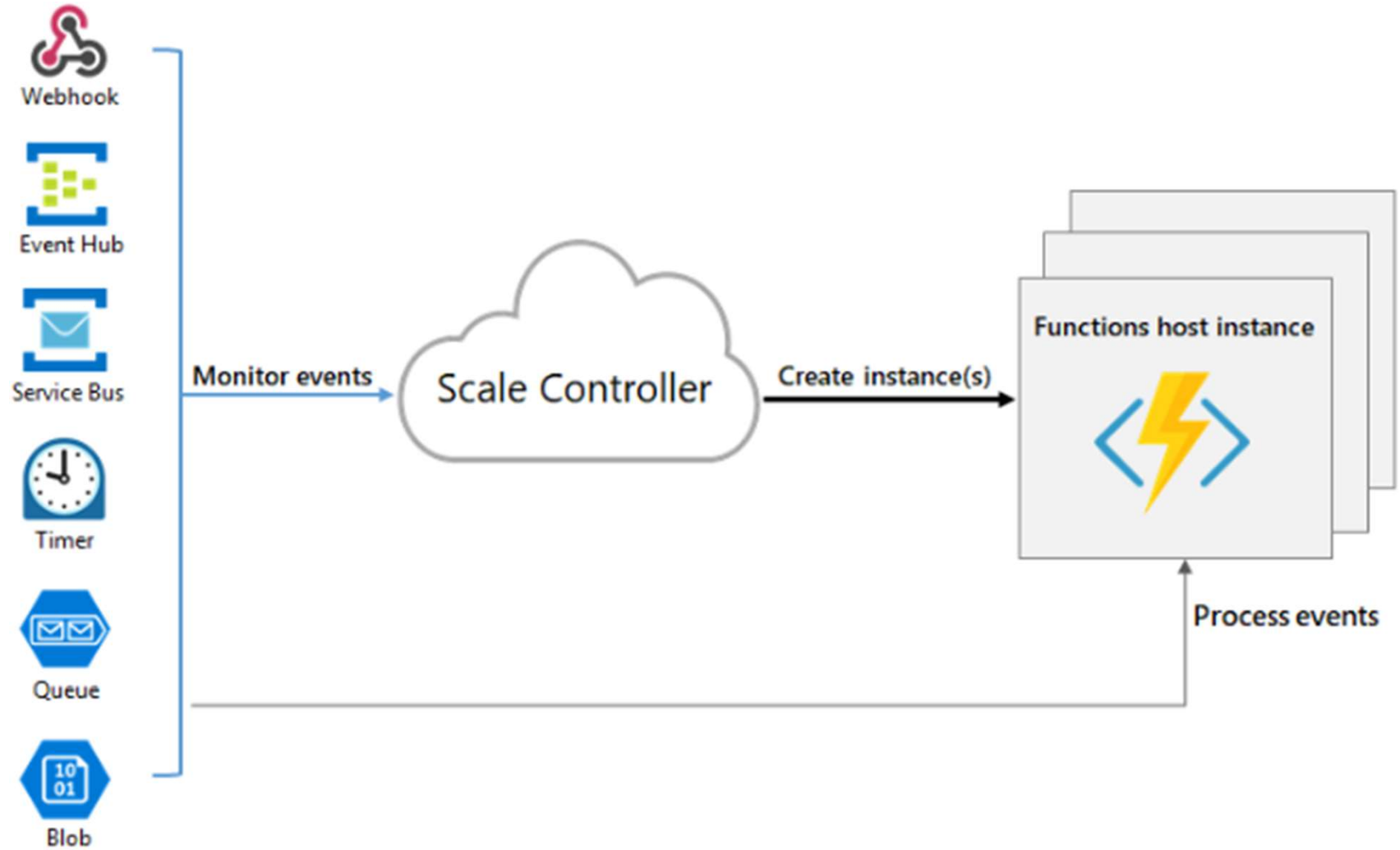
# Azure Function App: Hands-on

- Create our first Function App

# How Azure Functions Works



# Scale up and Scale Down





# Pricing

- Consumption plan is billed based on per-second resource consumption and executions
- For more details, please refer:
  - <https://azure.microsoft.com/en-in/pricing/details/functions/>

# Azure Function: Hands-on

- Create a Function in Azure Function App

# Azure Functions Drawbacks

## Security issues

- Function event data injection
- Insecure serverless deployment configuration
- Inadequate monitoring and logging of functions
- Insecure third-party dependencies
- DDoS attacks

## Vendor lock-in

- Migrating to another is difficult.

## Debugging is more difficult

- Possible, but it's not a simple task, and it can eat up lots of time and resources.

## Cold starts

- Sometimes a delay in the execution of a function, as much as 3 seconds, which can adversely impact some types of applications.

# Azure Functions Drawbacks

- Function app timeout duration

Plan	Runtime Version	Default	Maximum
Consumption	1.x	5	10
Consumption	2.x	5	10
Consumption	3.x	5	10
Premium	1.x	30	Unlimited
Premium	2.x	30	Unlimited
Premium	3.x	30	Unlimited
App Service	1.x	Unlimited	Unlimited
App Service	2.x	30	Unlimited
App Service	3.x	30	Unlimited

# Overview of Triggers

A trigger is what invokes a function to run

A trigger defines how a function is invoked

Each function in Azure Functions must have only one trigger

Triggers usually have associated data, which is nothing but the payload that triggered the function

# Example of function

```
run.csx Save Run </> Get function URL

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 public static async Task<IActionResult> Run(HttpRequest req, string inputBlob, ILogger log)
9 {
10     log.LogInformation("C# HTTP trigger function processed a request.");
11
12     string name = req.Query["name"];
13
14     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
15     dynamic data = JsonConvert.DeserializeObject(requestBody);
16     name = name ?? data?.name;
17
18     string greetingMessage = inputBlob;
19     if(greetingMessage == null)
20         greetingMessage = "Hello ";
21
22     return name != null
23         ? (ActionResult)new OkObjectResult(greetingMessage + name)
24         : new BadRequestObjectResult("Please pass a name on the query string or in the request body");
25 }
26
```

1) Triggers

2) Input

4) Your actual codes

3) Output

# Type of Triggers

- A series of templates is available to get you started with key scenarios including:
  - HTTP: Run code based on HTTP requests
  - Timer: Schedule code to run at predefined times
  - Azure Cosmos DB: Process new and modified Azure Cosmos DB documents
  - Blob storage: Process new and modified Azure Storage blobs
  - Queue storage: Respond to Azure Storage queue messages
  - Event Grid: Respond to Azure Event Grid events via subscriptions and filters
  - Event Hub: Respond to high-volumes of Azure Event Hub events
  - Service Bus Queue: Connect to other Azure or on-premises services by responding Service Bus queue messages
  - Service Bus Topic: Connect other Azure services or on-premises services by responding to Service Bus topic messages

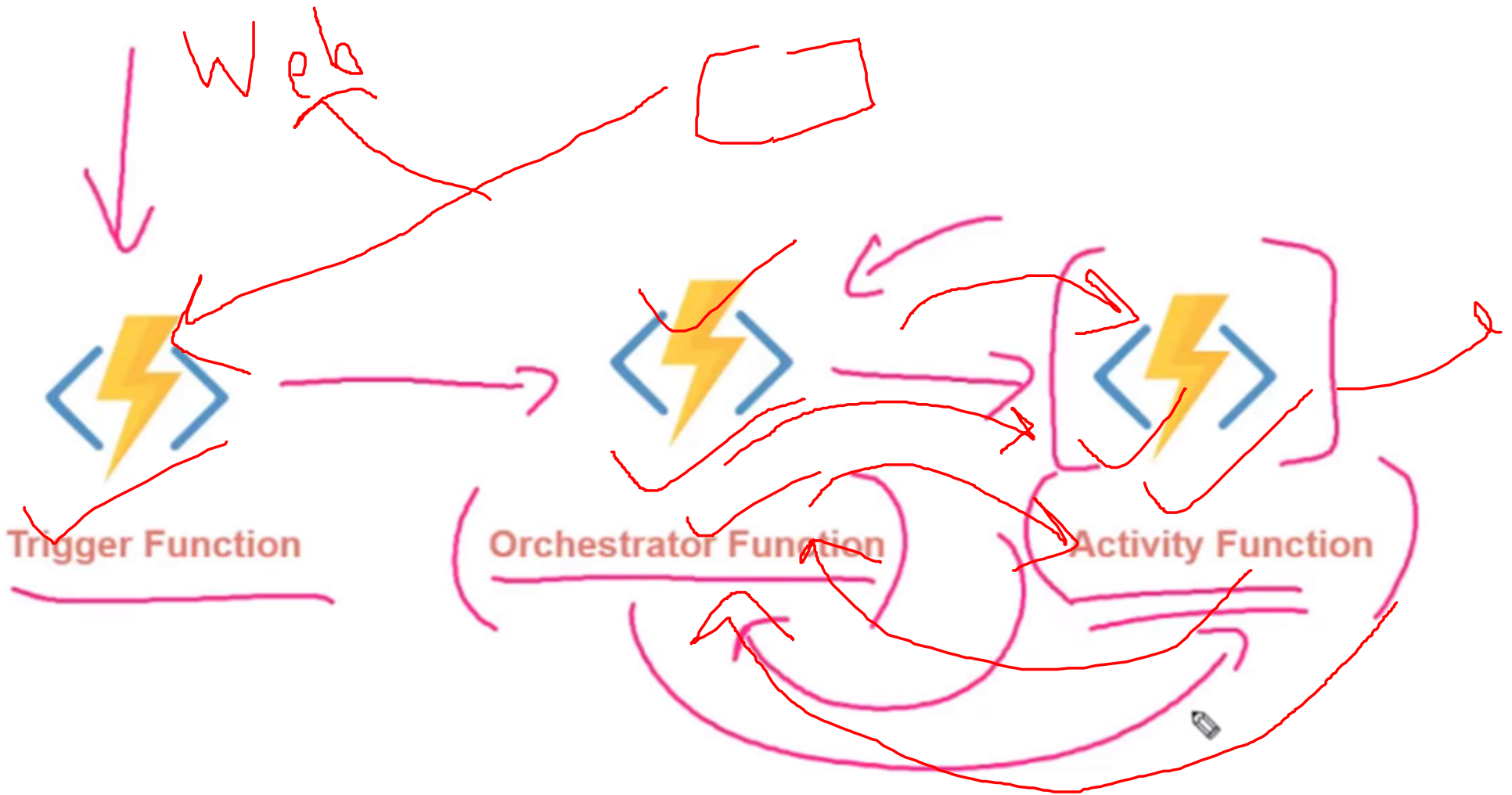
# Triggers: Hands-on



# Azure Durable Functions

- An extension of Azure Function that is used to write Stateful Functions
- It consists of Orchestrator function and entity functions that can be defined as a workflow
- As a state-based services, it enables checkpoint and restarts from it.
- Azure Durable Function supports languages like
  - C#
  - JavaScript
  - F#
  - Python
  - PowerShell

# Azure Durable Functions



# Azure Durable Functions - Example

- In this example, the values F1, F2, F3, and F4 are the names of other functions in the same function app.
  - `import azure.functions as func`
  - `import azure.durable_functions as df`
  - `def orchestrator_function(context: df.DurableOrchestrationContext):`
  - `x = yield context.call_activity("F1", None)`
  - `y = yield context.call_activity("F2", x)`
  - `z = yield context.call_activity("F3", y)`
  - `result = yield context.call_activity("F4", z)`
  - `return result`
  - `main = df.Orchestrator.create(orchestrator_function)`

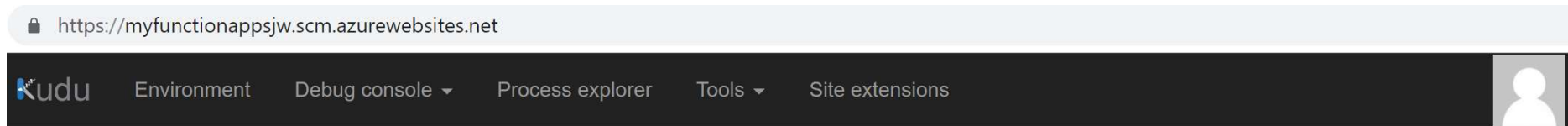
# Durable Function: Hands-on

- Create a Durable Function

# What are Kudu in Function App

- The advanced tools for App Service (also known as Kudu)
- Provide access to advanced administrative features of your function app.
- From Kudu, you manage
  - System information,
  - App settings,
  - Environment variables,
  - Site extensions,
  - HTTP headers, and
  - server variables.
- Accessing your KUDU console is easily done via the following URL:  
<https://dotest.scm.azurewebsites.net>
- Replace dotest with the name of your app name

# What are Kudu in Function App



## Environment



Build	79.11121.3655.0 (935294dece)
Azure App Service	79.0.8598.64 (rd_websites_stable.181217-1712)
Site up time	00.00:03:05
Site folder	D:\home
Temp folder	D:\local\Temp\

## REST API (works best when using a JSON viewer extension)

- [App Settings](#)
- [Deployments](#)
- [Source control info](#)
- [Files](#)
- [Log streaming](#) (use curl, not browser!)
- [Processes and mini-dumps](#)
- [Runtime versions](#)
- Site Extensions: [installed](#) | [feed](#)
- [Web hooks](#)
- WebJobs: [all](#) | [triggered](#) | [continuous](#)
- Functions: [list](#) | [host config](#)

/ + | 3 items



	Name	Modified	Size
 	 data	29/04/2016, 10:28:24 am	
 	 LogFiles	29/04/2016, 10:32:54 am	
 	 site	29/07/2016, 1:50:51 pm	



[Use old console](#)

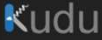

Kudu Remote Execution Console

Type 'exit' then hit 'enter' to get a new powershell process.

Type 'cls' to clear the console

PS D:\home>

# Process Explorer

 Environment Debug console ▾ Process explorer **Tools ▾** Site extensions 

D diagnostic dump

L log stream

W WebJobs dashboard

W Web hooks

D Download deployment script

S Support

er

[Refresh](#)

name	pid	user_name	total_cpu_time	memory	thread_count	properties	profiling	
w3wp.exe	24000	dotest	6 s	121,960 KB	102,816 KB	44	<a href="#">Properties..</a>	<a href="#">Start Profiling</a>
w3wp.exe <span>scm</span>	25440	dotest	7 s	112,996 KB	98,728 KB	49	<a href="#">Properties..</a>	<a href="#">Start Profiling</a>



# Kudo: Hands-on

*Thanks*