



Create serverless applications with Azure Functions

Teerasej Jiraphatchandej, Microsoft Certified Trainer, teerasej@nextflow.in.th



“Pon”

Teerasej Jiraphatchandej

www.nextflow.in.th



Why Azure Function?









Where's the code?

- Required environment setup
- Fix resources
- Operate 24/7
- Create the project to run a few tasks





Azure Functions

1. Compute service on the cloud

Environment ready



2. Flexible hosting plan

**Consumption, Premium,
Dedicated**





Scale.







3. Event-based

**Focus on process & data
, like robots in a factory**





Lab 1: Create a function app in Azure portal

**Choose the best
Azure Service to
automate your
business**





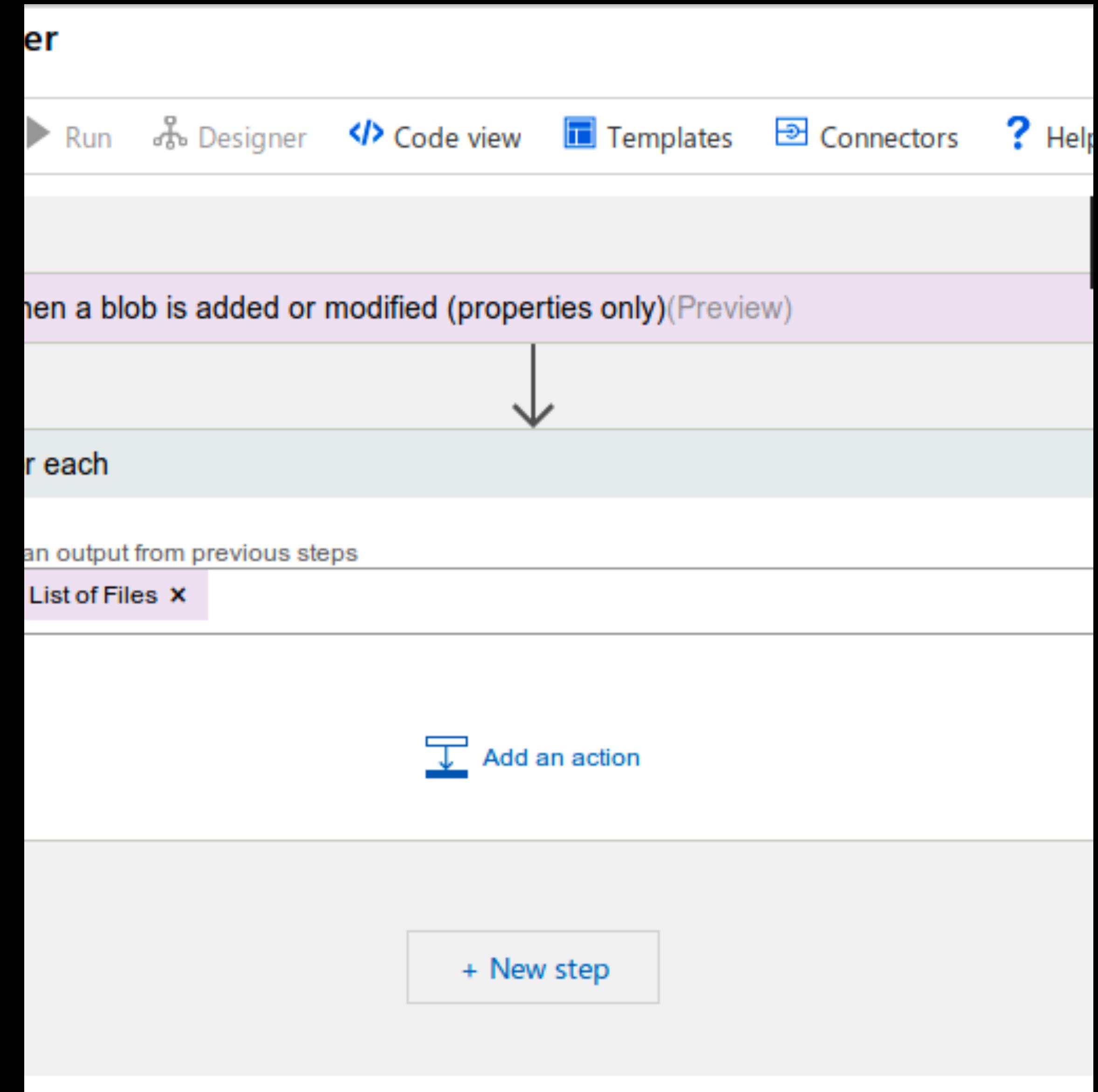
4 technologies



2 Design-first tools
2 Code-first tools

1. Logic Apps

- For developer & IT Pro
- Design UI & Code
- Advanced integration
- Can be added to Devops & Git



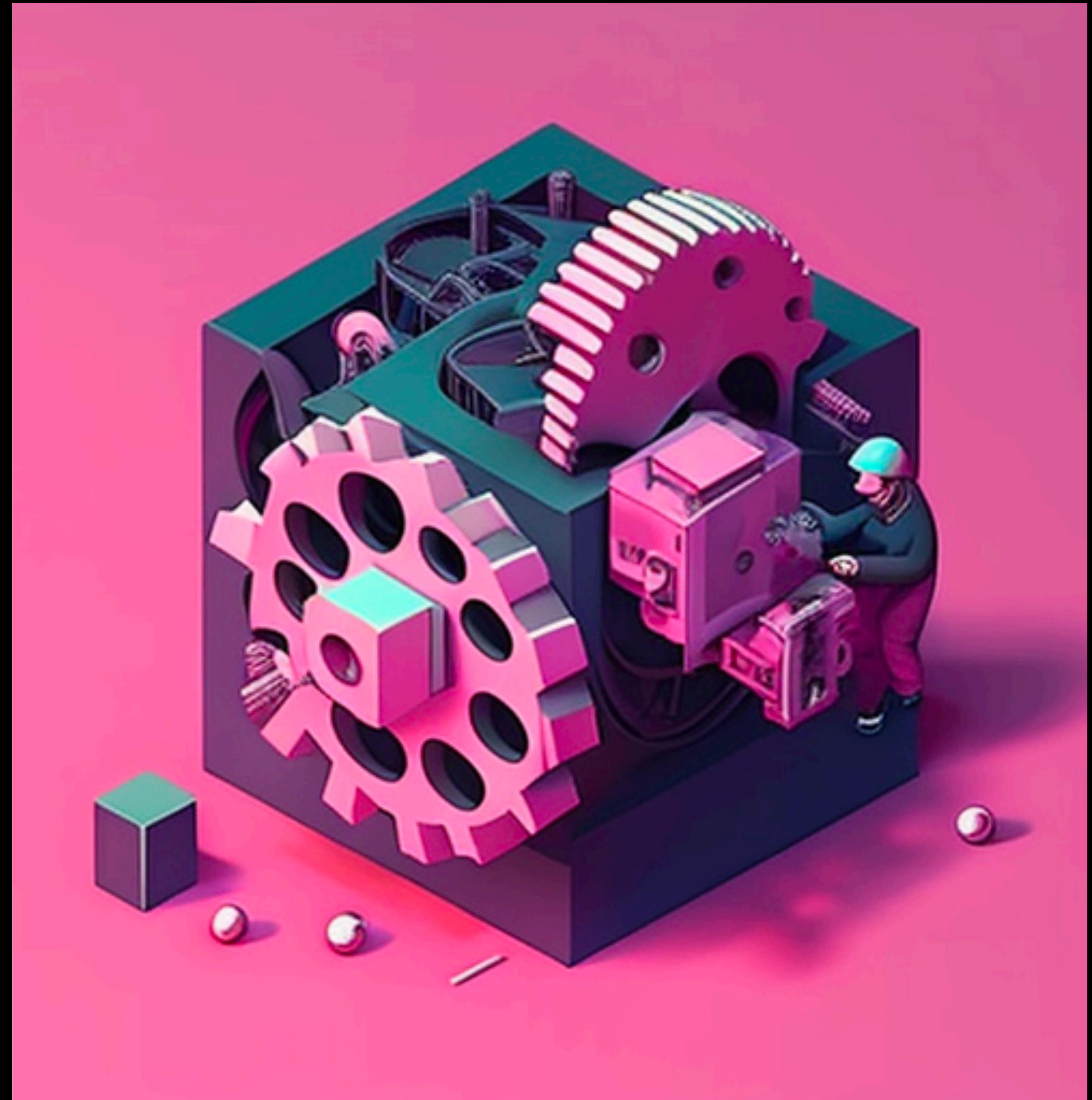
2. Power Automate

- Office worker & business analyst
- Self-service workflow
- GUI only
- Include testing & production environment

The screenshot shows the Microsoft Power Automate Flow interface. On the left, there's a navigation sidebar with options: Home, Approvals, My flows, Templates, Connectors, Data (with a dropdown arrow), and Learn. The main area is titled "Send a customized email when a new file is added". It consists of two main sections: "When a file is created (properties)" and "Get my profile (V2)". The first section has fields for "Site Address" (PhotoDojo) and "Library Name" (Documents). The second section is for "Get my profile (V2)" and notes "No default parameters required." There's also a link to "Show advanced options". At the bottom, there's another section for "Send an email" with a field for "To".

3. Webjobs

- Part of Azure App Service
- Scheduled/Manual
- C# Support



4. Azure Functions

- Auto Scaling
- Dev & Test in browser
- Pay per use
- Support C#, Java, JavaScript, Powershell, etc.





Lab 2: Add Logic to the function app

Execute Azure Functions with Trigger

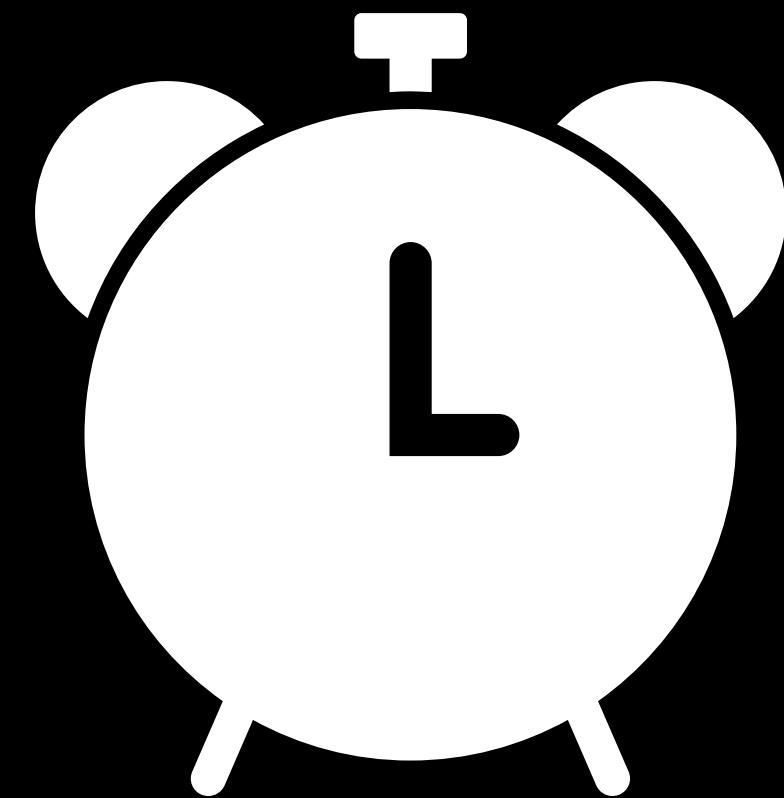




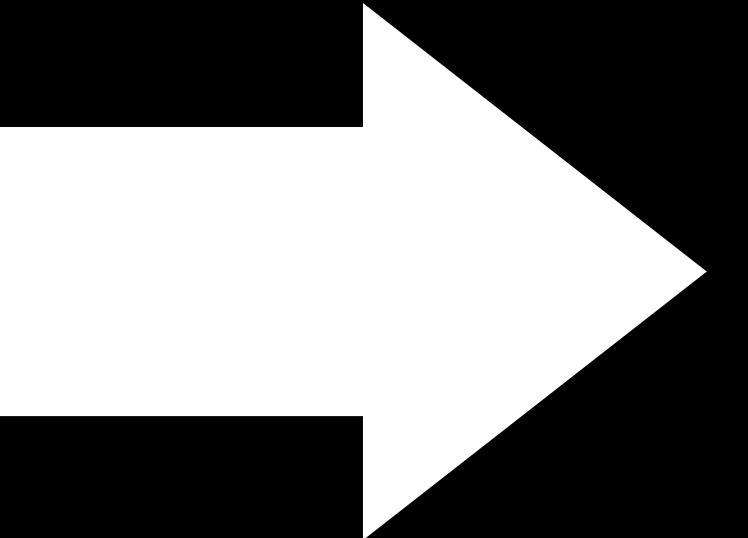
Triggers

Trigger's types

Example



Timer



HTTP



Blob



“Only 1 trigger per function”



Timer

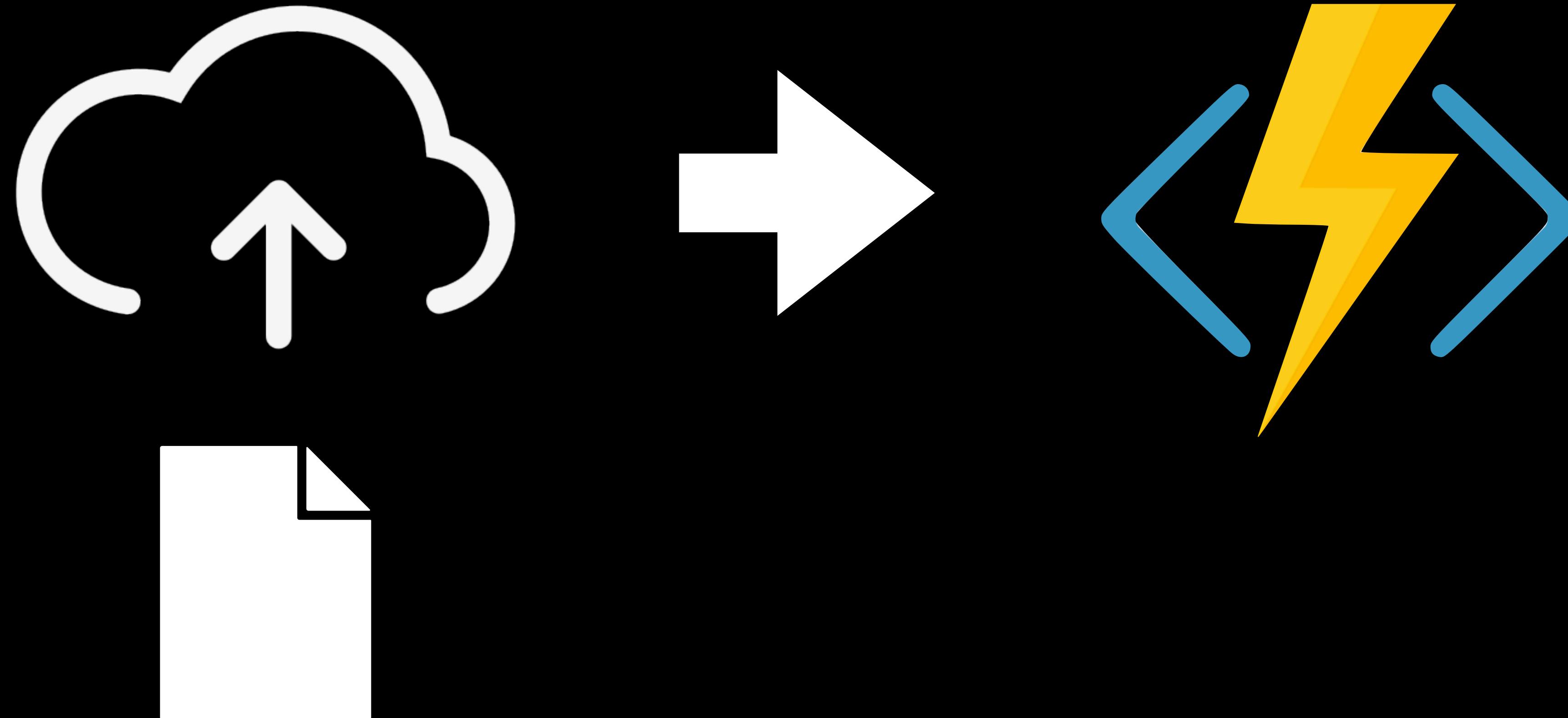
{second} {minute} {hour} {day} {month} {day of the week}

*/5 * * * *

*/5 * 16 * * *

*/5 * 16 * * Fri

Blob trigger





Lab 3, 4, 5: Execute Azure Function with triggers

Chain Azure Functions together using input and output bindings





Bindings

Bindings component

- Input as a coin
- Output as a soda can
- One or more binding per function
- Less code



Input Binding example

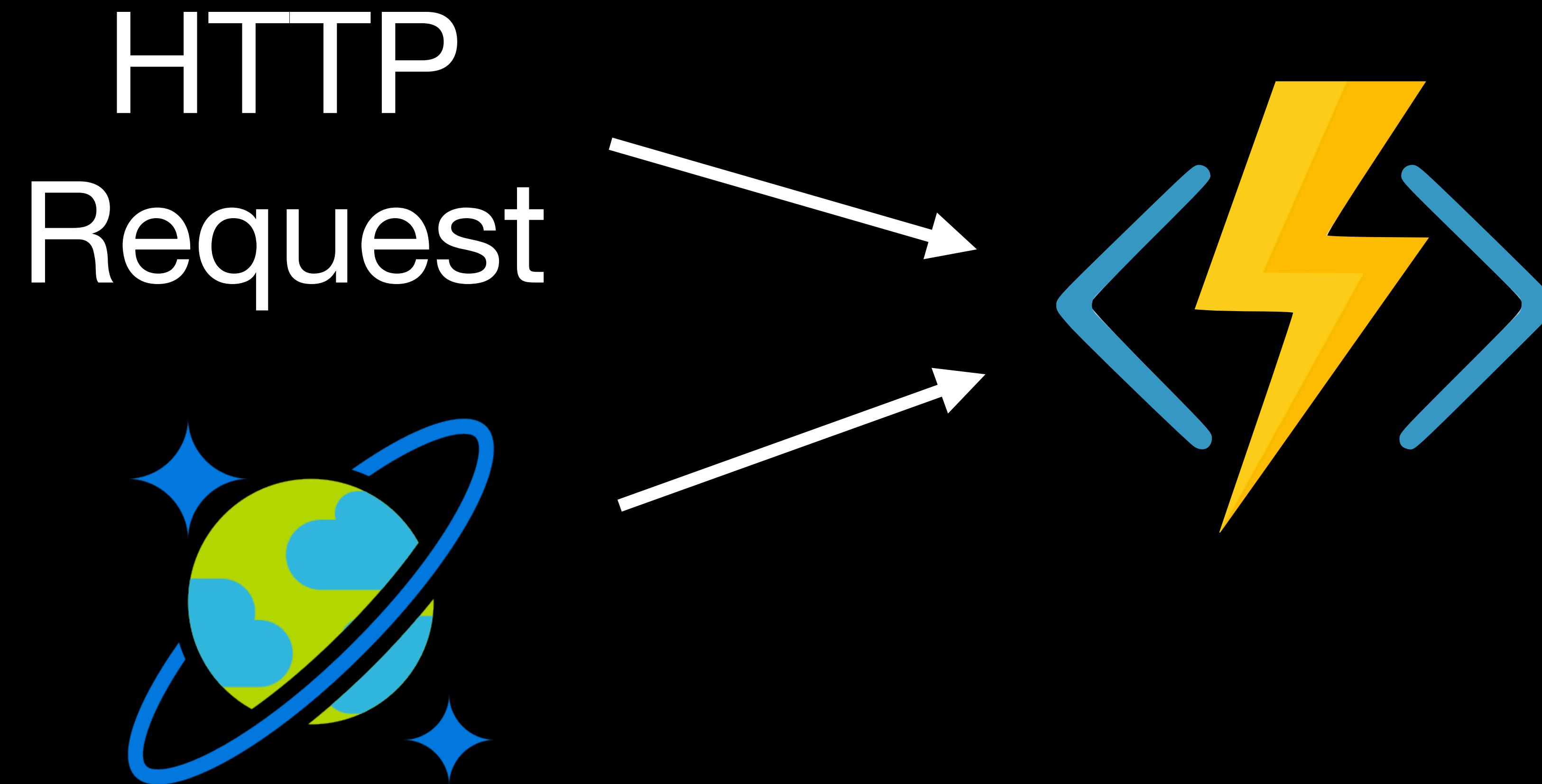
HTTP Request & Cosmos DB

HTTP
Request



Input Binding example

HTTP Request & Cosmos DB

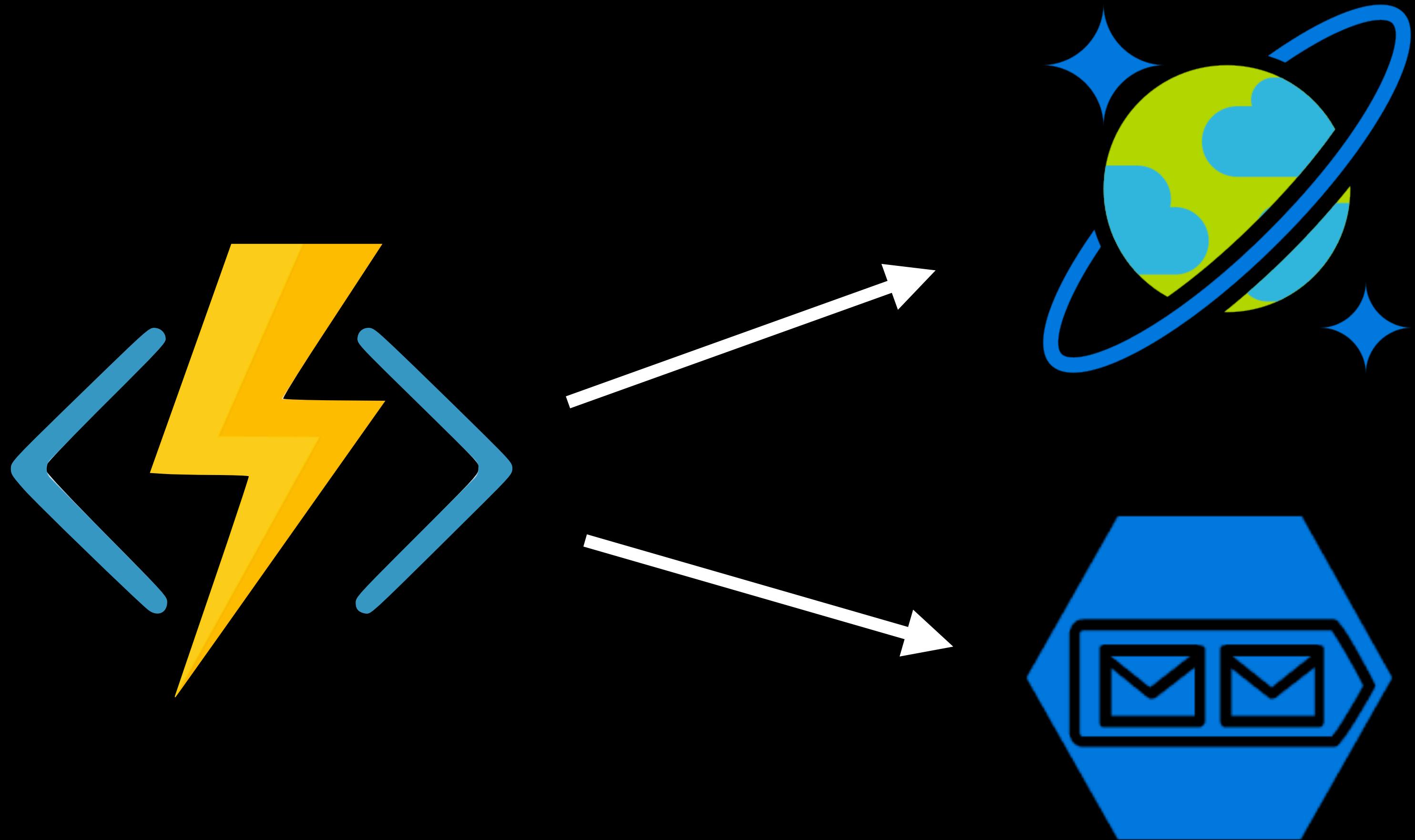




Lab 6, 7: Execute Azure Function with triggers

Output Binding example

Cosmos DB & Storage Queue





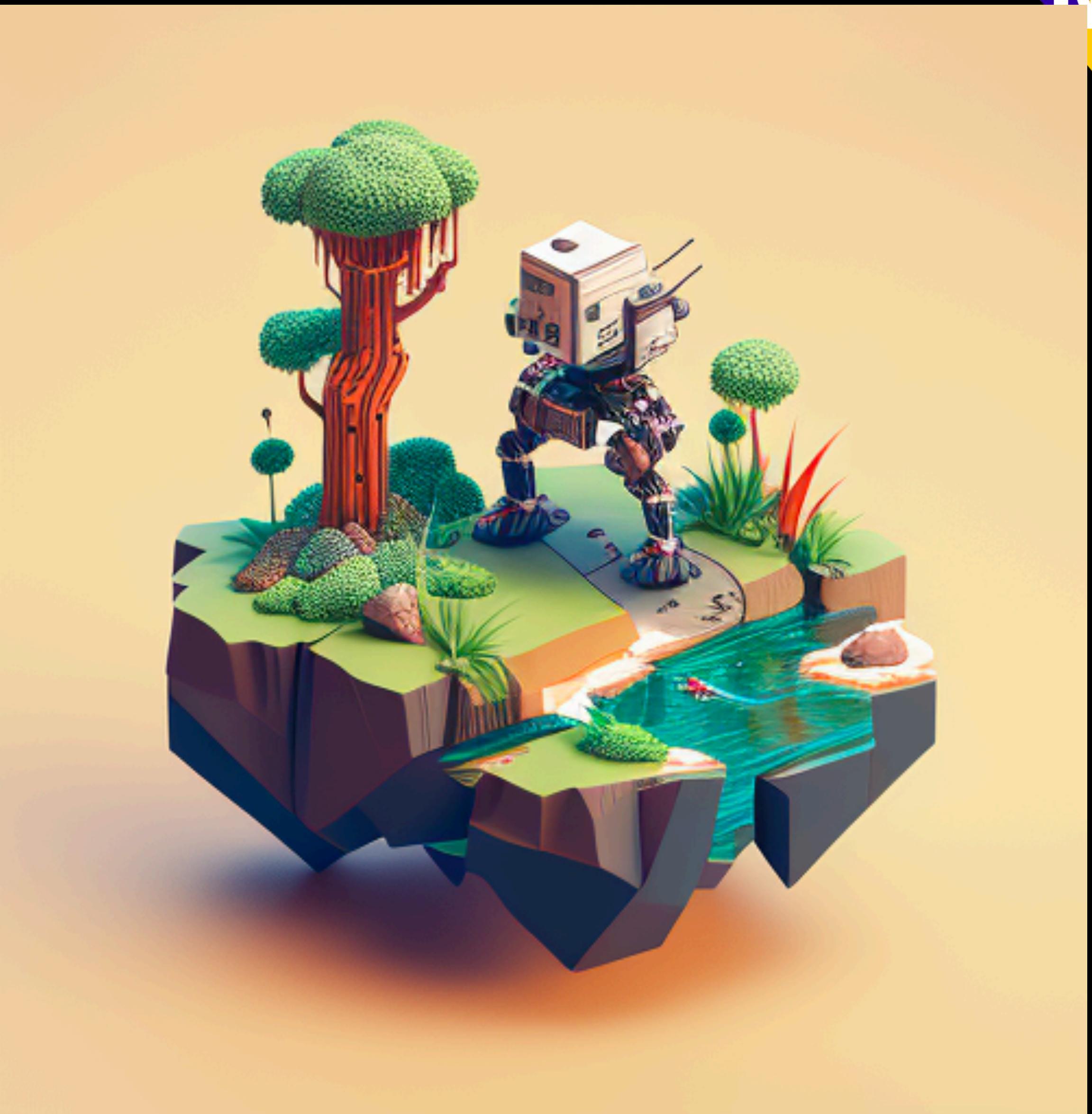
Lab 8:

Execute Azure Function with triggers

Create a long-running serverless workflow with Durable Functions



Why?
Functions are independent.
Stateless.



Durable function

- A function's extension
- Team up multiple functions
 - Chain
 - Orchestrate
- Manage state





Durable Function's type

Client type

- Entry point
- Response to any source



Orchestrator type

- How action will be execute in order
- Support C# & JavaScript



Activity type

- Basic unit
- Perform a task





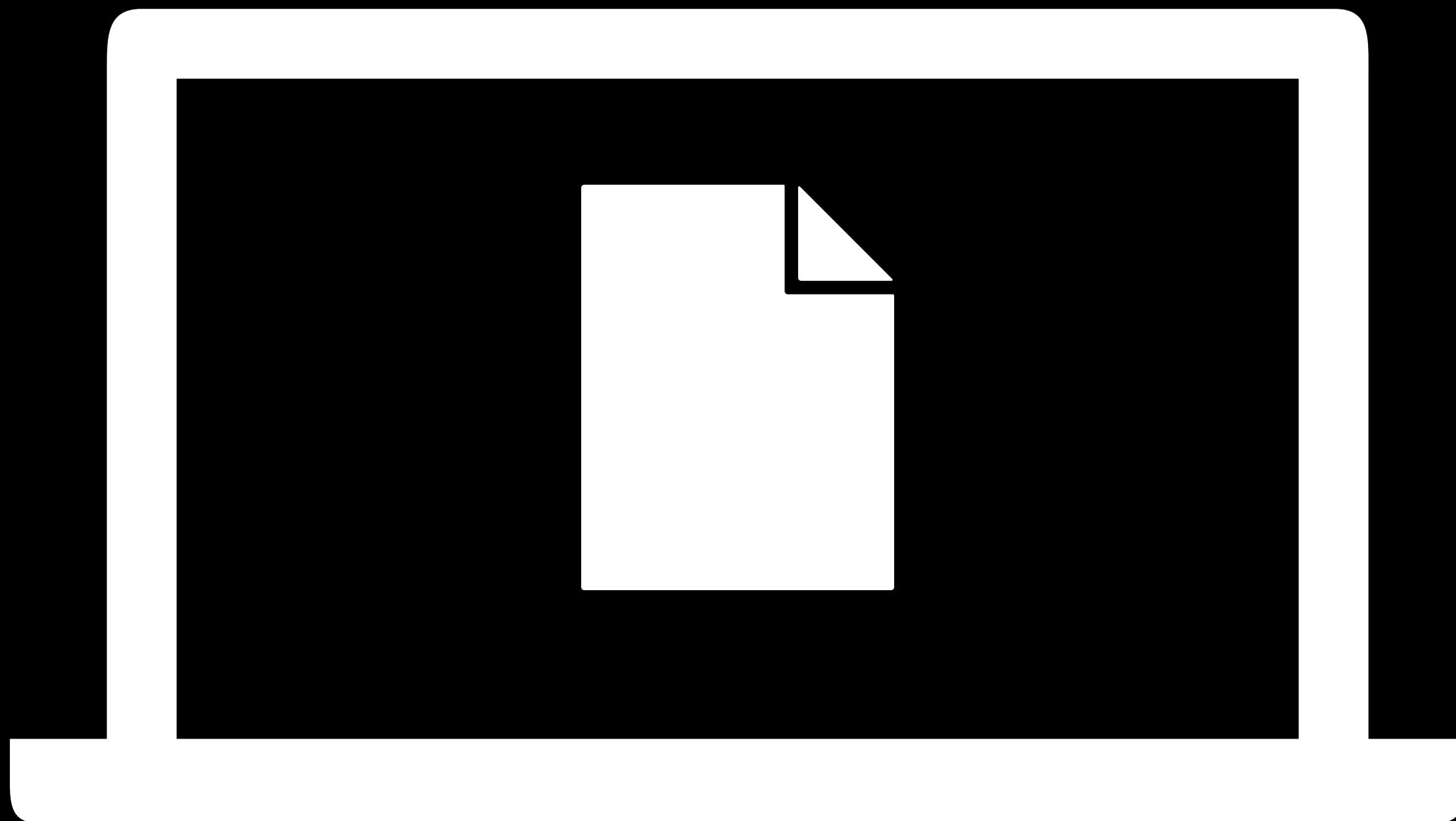
Lab 9: Create a workflow using Durable Functions

Develop, test, and publish Azure Functions by using Azure Functions Core Tools



Develop locally

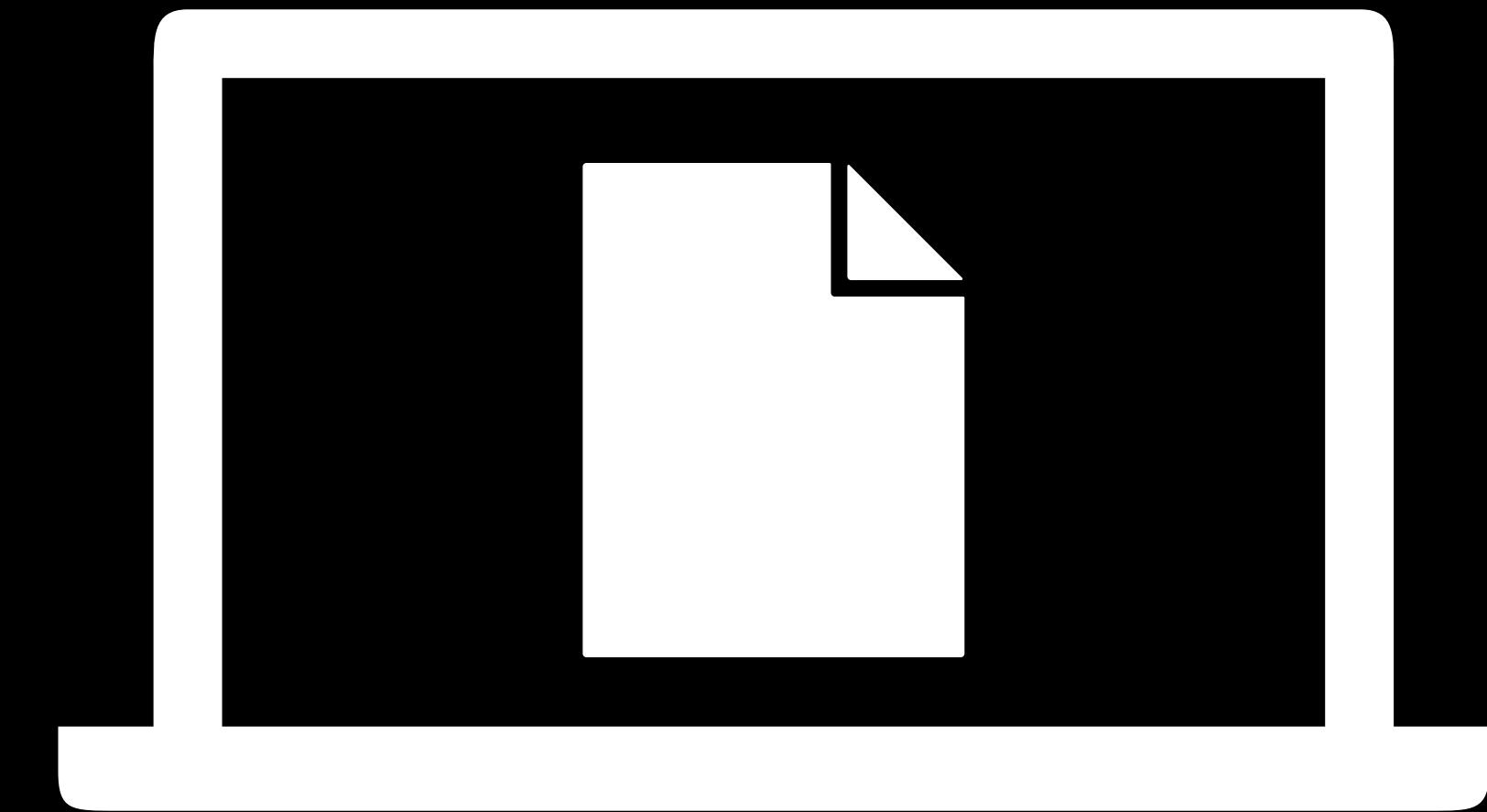
With CLI



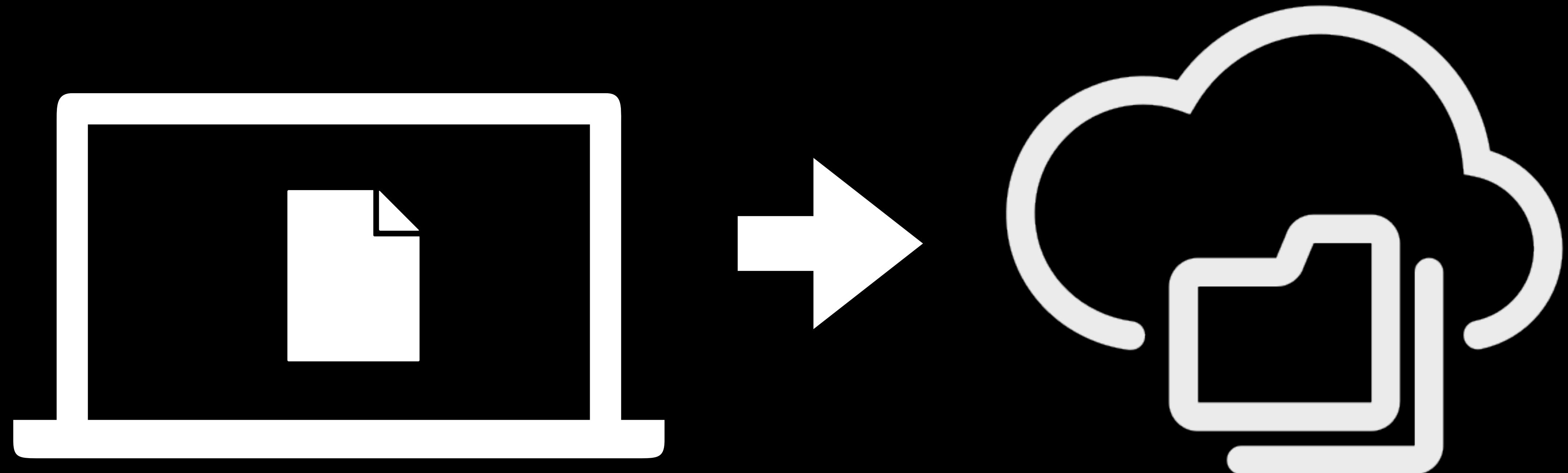
Develop locally

With CLI

- Generate function app
- Generate function from template
- Run function for testing & debug



Publishing With CLI





Lab 10, 11: Develop, test, and publish Azure Functions by using Azure Functions Core Tools

The conclusion





Thank you!

Follow more at:

