Building AI-Powered Applications with Azure Serverless Functions

Commit your Code 2025

Speaker: **Naveen Chatlapalli**

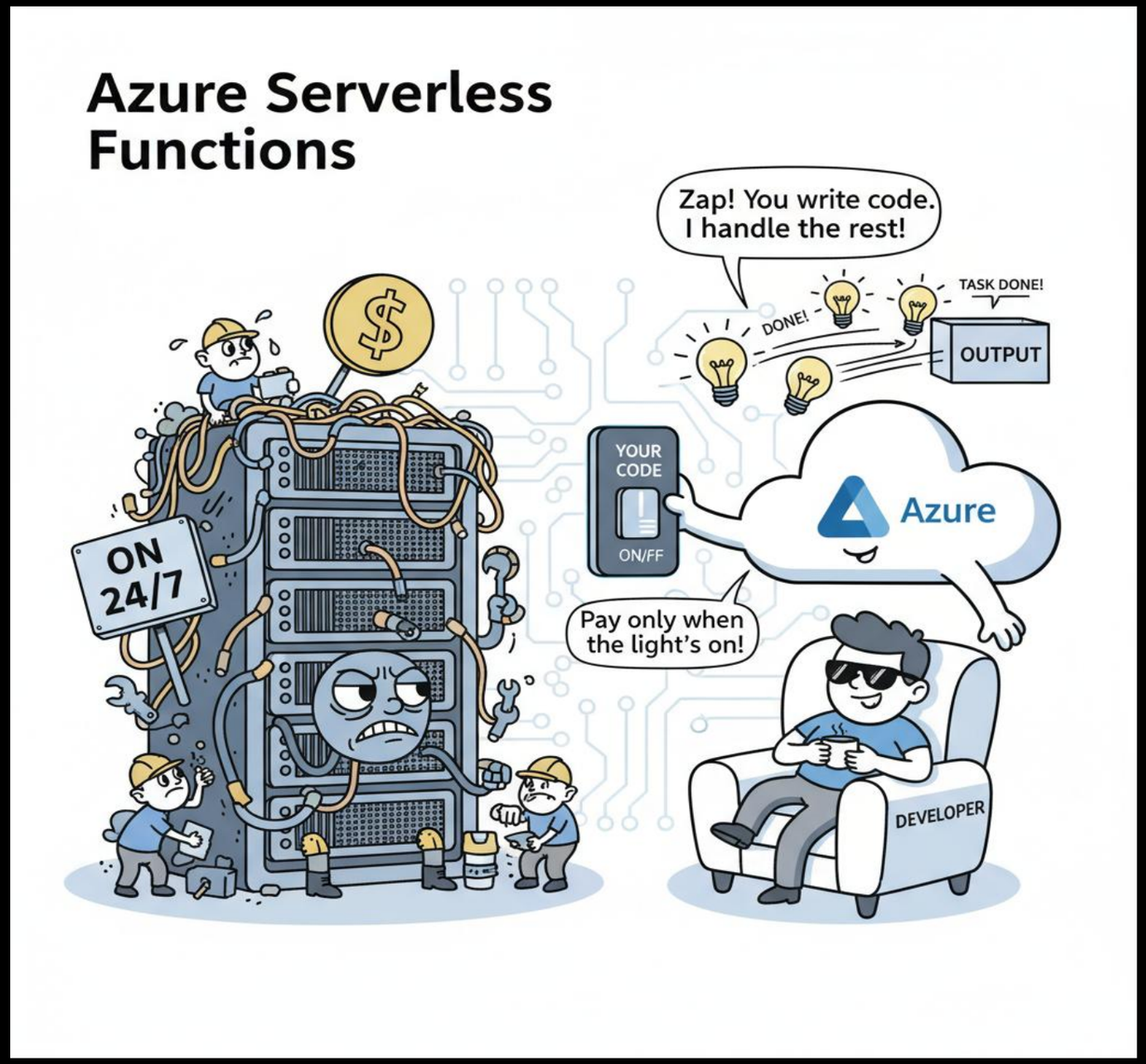
ASHLING

September 26, 2025

Agenda

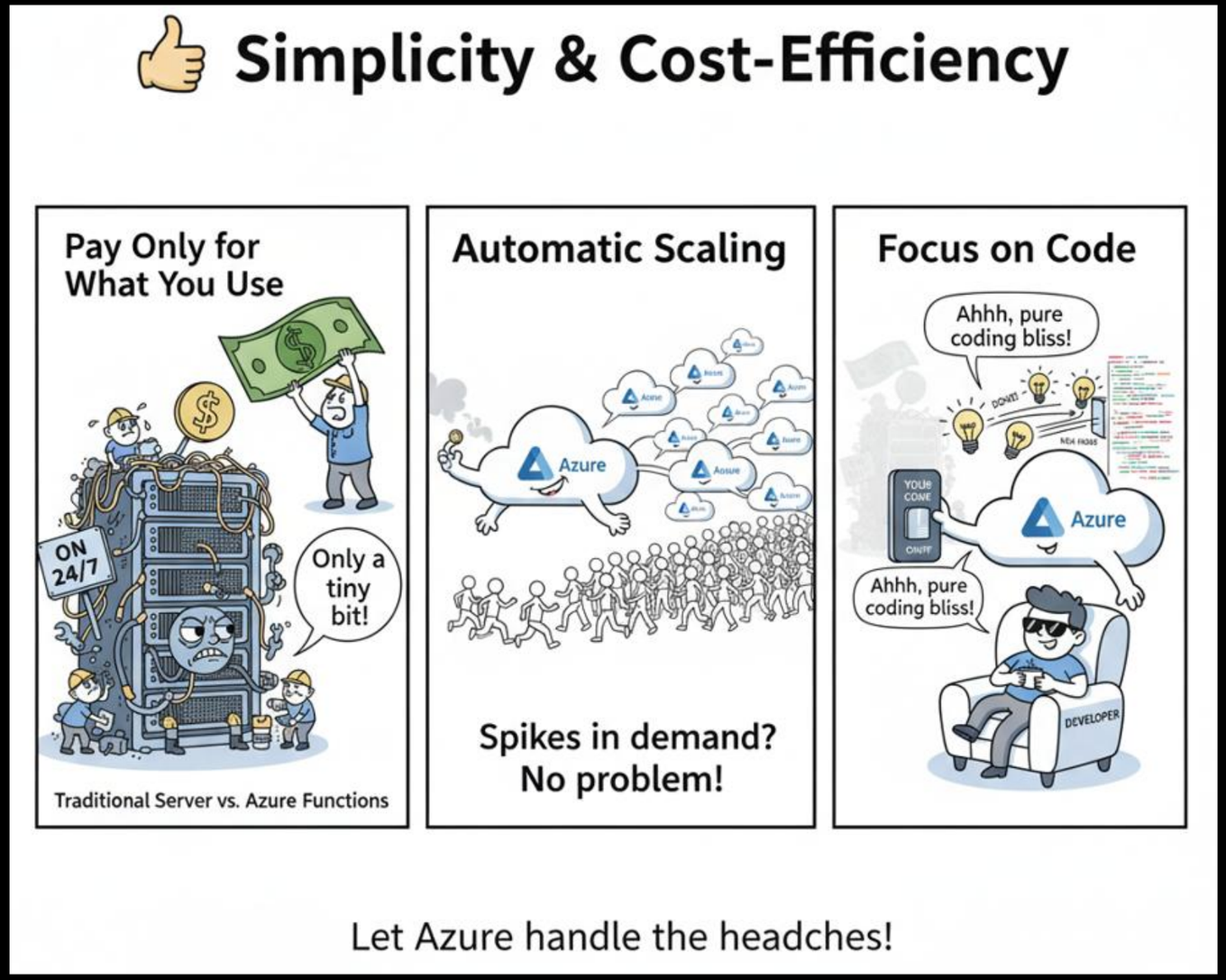
- What + Why + How Serverless
- Azure Functions 101
- AI Integration & Best Practices
- Real-world Demo

What is Serverless



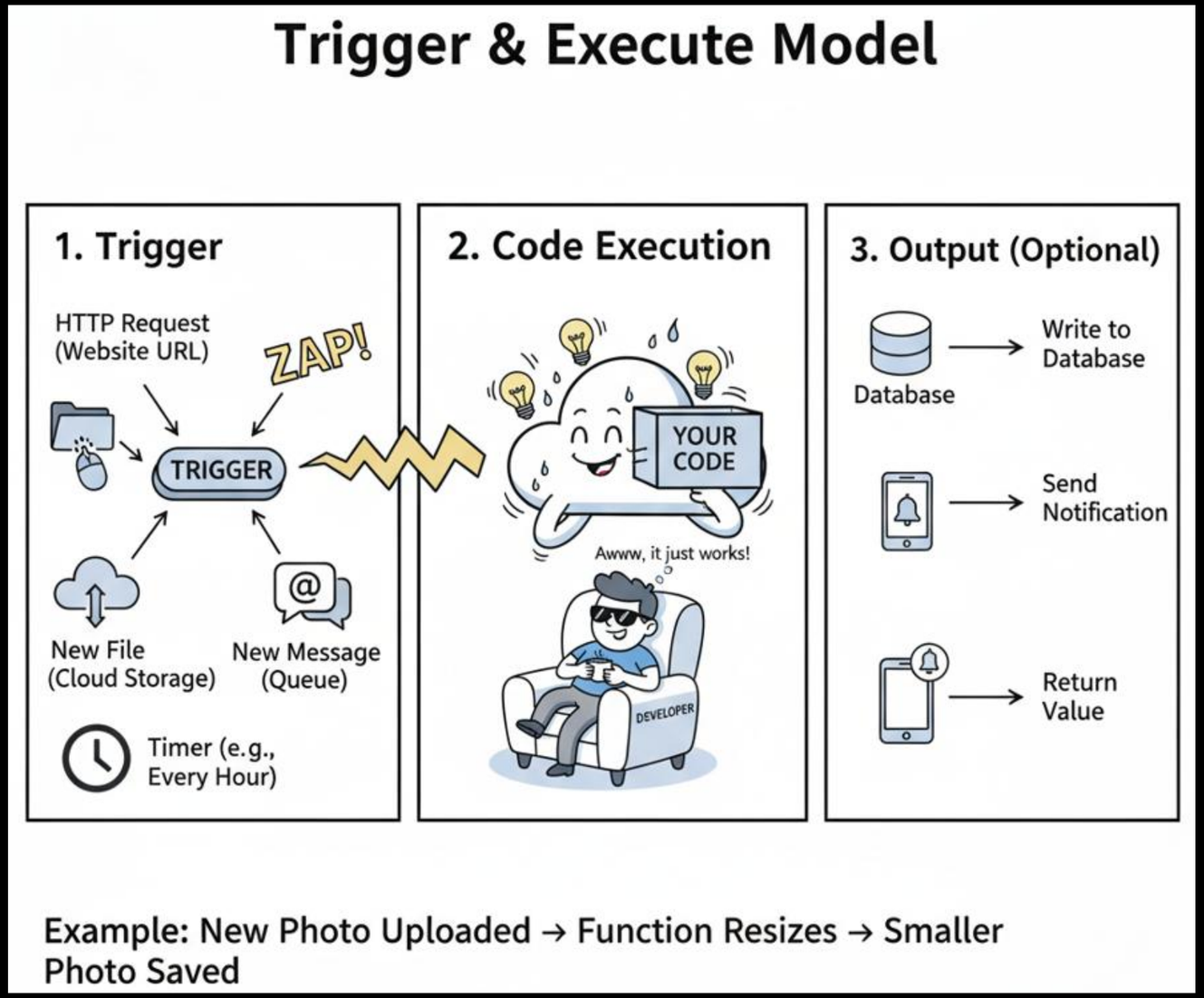
Azure Functions is a **serverless** solution that lets you run code without managing any infrastructure. You simply provide the code for what you want to do, and it only runs when triggered by an event

Why Serverless



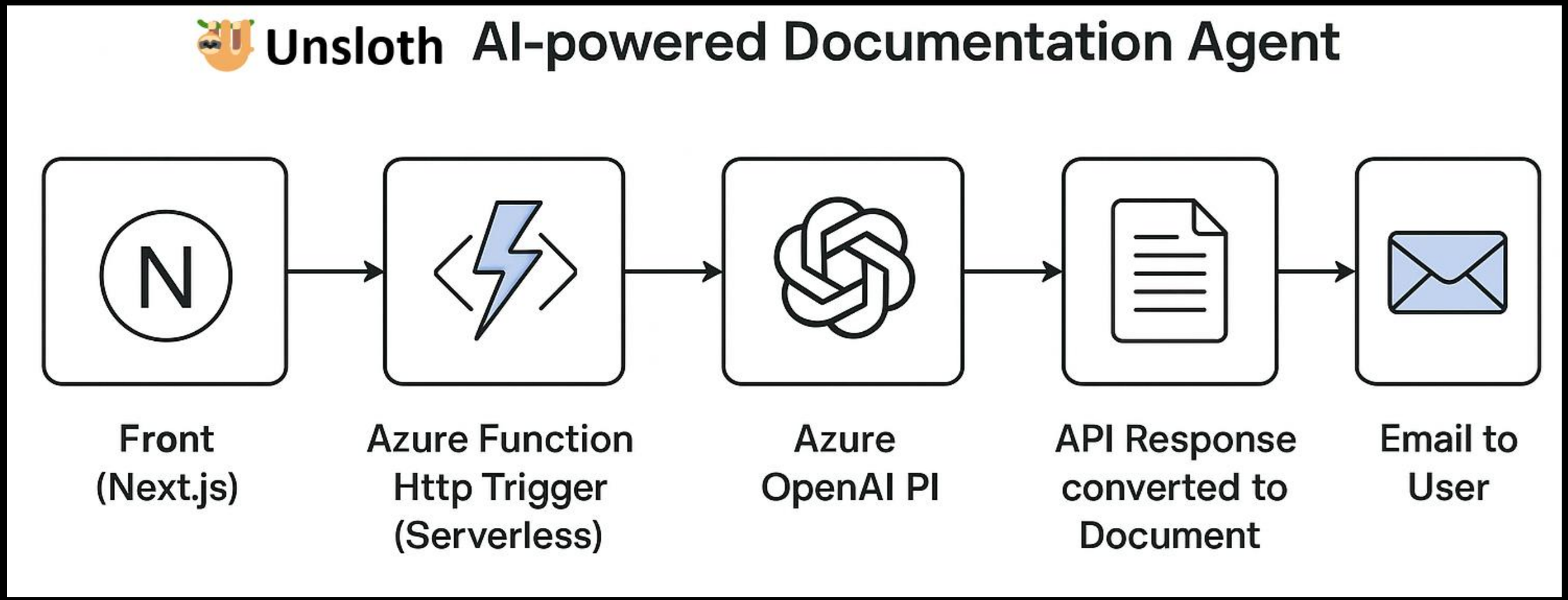
The key benefits of using Azure Functions are **simplicity** and **cost-efficiency**. The service **automatically scales** to handle any workload, you **pay only for what you use**, and developers can **focus entirely on their code** instead of managing servers.

How Serverless

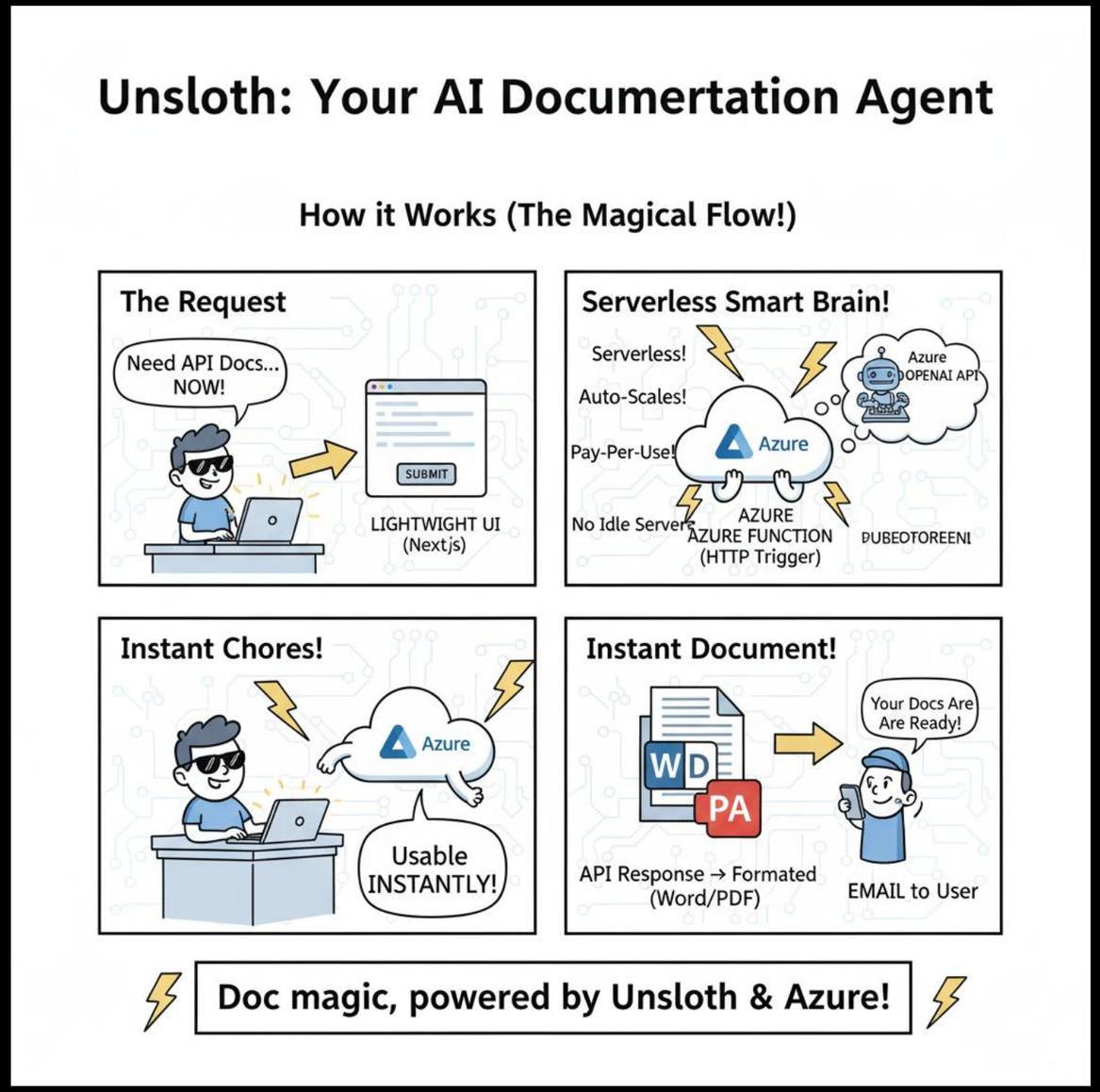


Azure Functions work using a **trigger and execute** model. An event, like a new file being uploaded or a scheduled time, acts as a **trigger** that automatically runs your code. Once executed, your function performs its task and can produce an **output**, like saving data to a database or sending a notification.

Why I chose serverless



Product I developed



- Serverless, scales automatically with user requests.
- Pay-per-use, so cost-efficient (no idle servers).
- Directly integrates with front-end requests via HTTPS — perfect for a request/response pattern.
- Keeps infrastructure minimal (no VM or container overhead).

Unsloth | Inbox - naveen.aifanatic@gmail | Intelligent Automation Solution

unsloth.aifanatic.pro/dashboard

Unsloth

- Dashboard
- Documents
- Feedback
- Settings

Beta

Dashboard Beta

Chrome Extension naveen.chatlapalli@ashlingpartners.com

Document Type

Excel

Report Type

Test Case Document

Project Description

Click the microphone to start recording

Voice Input

Example: 'Login functionality with email and password authentication, social login options, and multi-factor authentication. Includes password reset flow and session management.'

Wanna try a template?

Recipients

Add email addresses separated by semicolons

naveen.aifanatic@gmail.com

Azure Functions Overview

Triggers & Bindings

HTTP, Event Grid, Blob, Timer, and more

Hosting Plans

Consumption, Premium, Flex

Language Support

Python, C#, JavaScript, and others

DevOps Ready

Local development with func CLI, GitHub Actions deployment

Consumption vs Premium vs Flex

Plan	Cost Model	Cold Start	Key Features
Consumption	Pay per execution	Highest	Infinite scale, zero cost when idle
Premium	Reserved instances	Minimal	Always Ready, VNET support, better SLA
Flex	Consumption + burst	Medium	15-min capacity reservation, balanced approach

Event-Driven Architecture

Event Publication

System or custom events trigger workflows automatically

Event Grid Routing

CloudEvents schema standardizes message format

Fan-out Processing

Single event triggers multiple parallel functions

Event Filtering

Process only relevant events based on patterns

Blob Storage Trigger

File Upload Event

Function automatically triggers when files are uploaded to Blob Storage.

Stream Processing

Handle large files efficiently with InputStream binding without full download.

Batch Processing

Process multiple files in parallel with maxBatchSize property settings.

Orchestration

Combine with Durable Functions for complex fan-in/out processing patterns.

Best Practices for Azure Serverless

Performance

Choose right plan. Minimize cold starts. Optimize code with smart caching.

Cost

Pay only when executed. Monitor usage. Keep resources in-region.

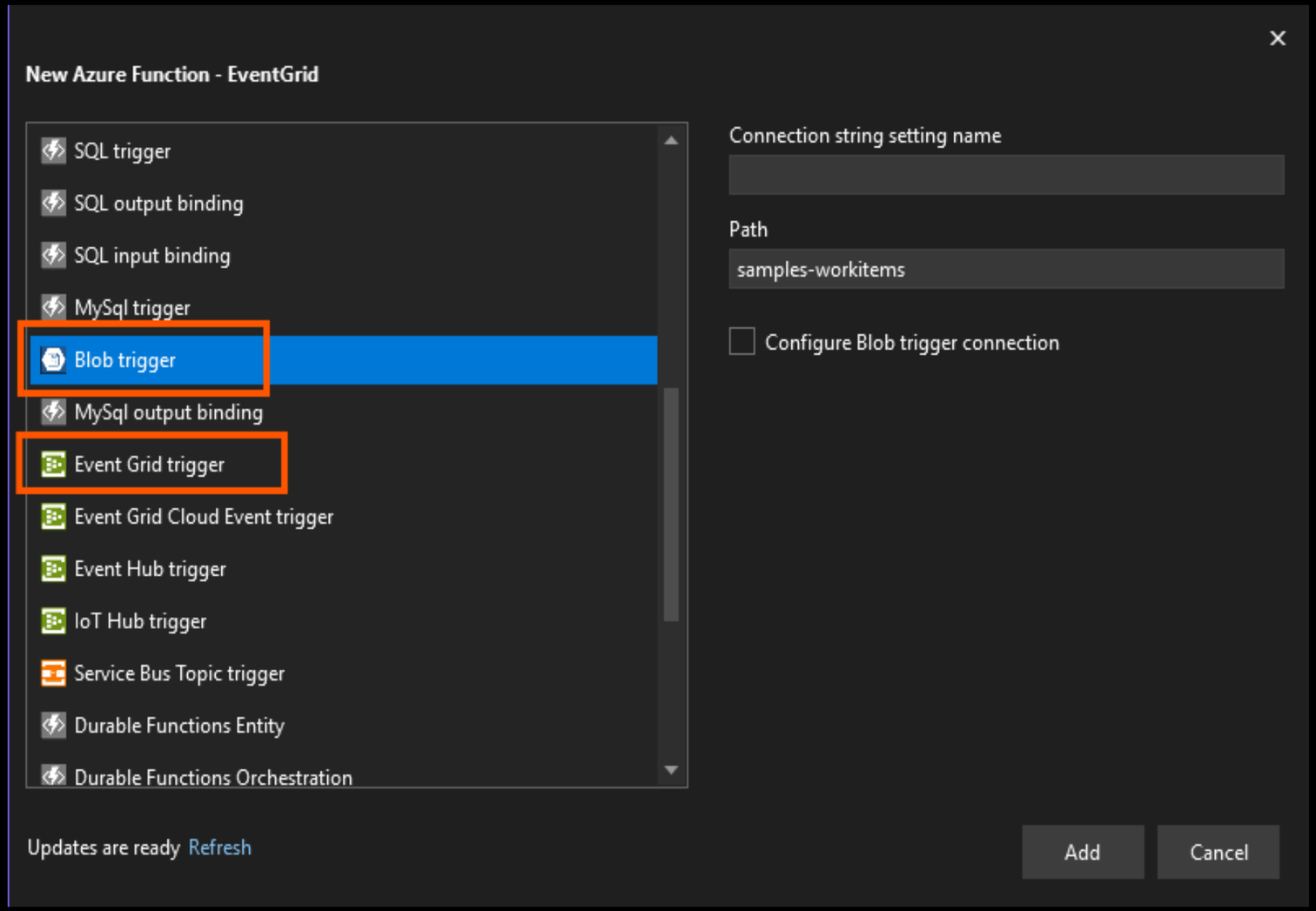
Security

Small, focused functions. Use API Management. Apply least-privilege access.

Real-World Example: IoT sensor data triggering real-time maintenance alerts.

Azure Function – Trigger Types

Add new Azure Function and choose your trigger type

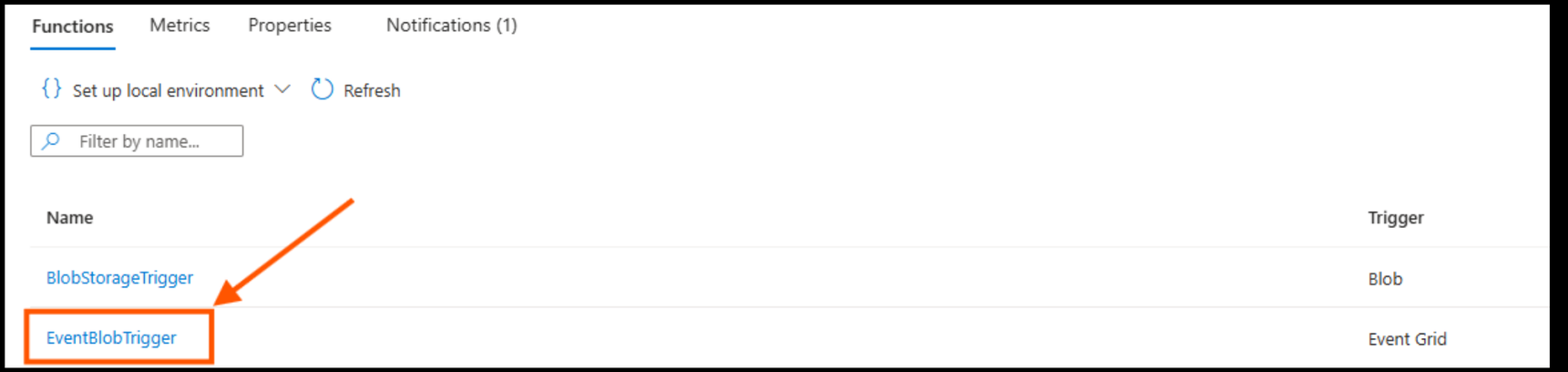


Azure Function – Event Grid Trigger Implementation

C# - Azure Function Class file which identifies the Function Name in Visual Studio

```
[FunctionName("EventBlobTrigger")]
0 references
public static void Run(
    [EventGridTrigger] EventGridEvent eventGridEvent,
    [Queue("azure-webjobs-blobtrigger-az-child-mrocfunction")] ICollector<BlobInfo> outputQueue,
    ILogger log)...
```

After publishing the function, the Azure portal → Functions displays the same matching trigger name

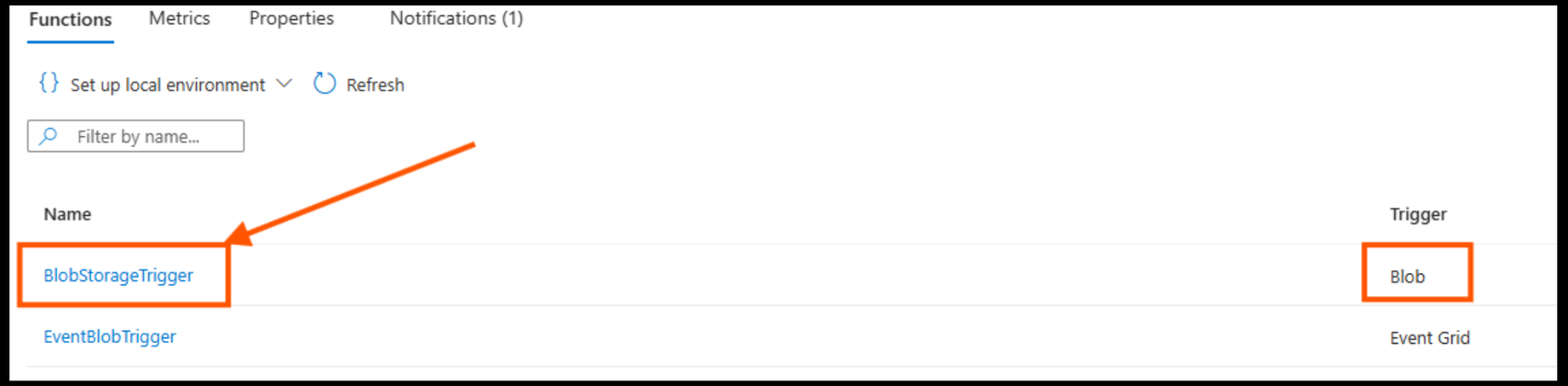


Azure Function – Blob Storage Trigger Implementation

C# - Azure Function Class file which identifies the Function Name in Visual Studio

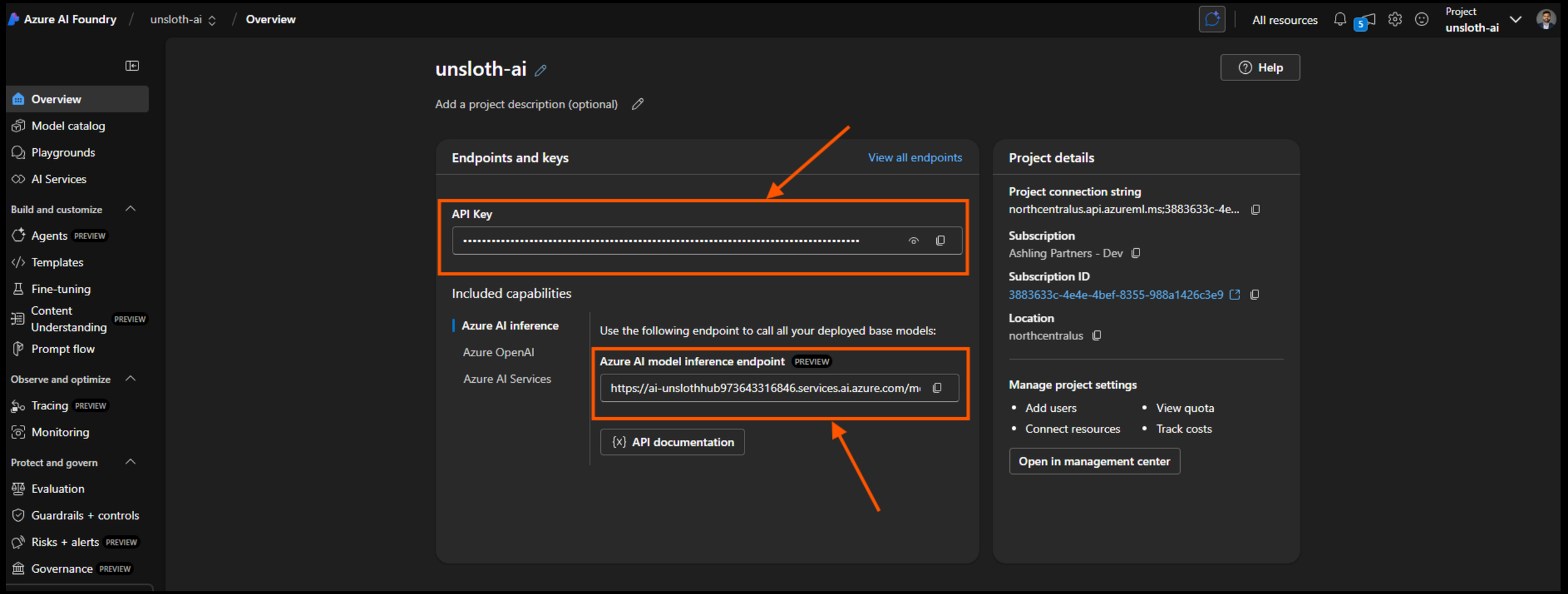
```
[StorageAccount("AzureWebJobsStorage")]
0 references
public class BlobTrigger
{
    [FunctionName("BlobStorageTrigger")]
    0 references
    public static async Task Run([BlobTrigger("%ContainerName%")] Stream myBlob, string name, ILogger log)...
```

After publishing the function the Azure portal → Functions displays the same matching trigger name



Azure AI Foundry - Implementation

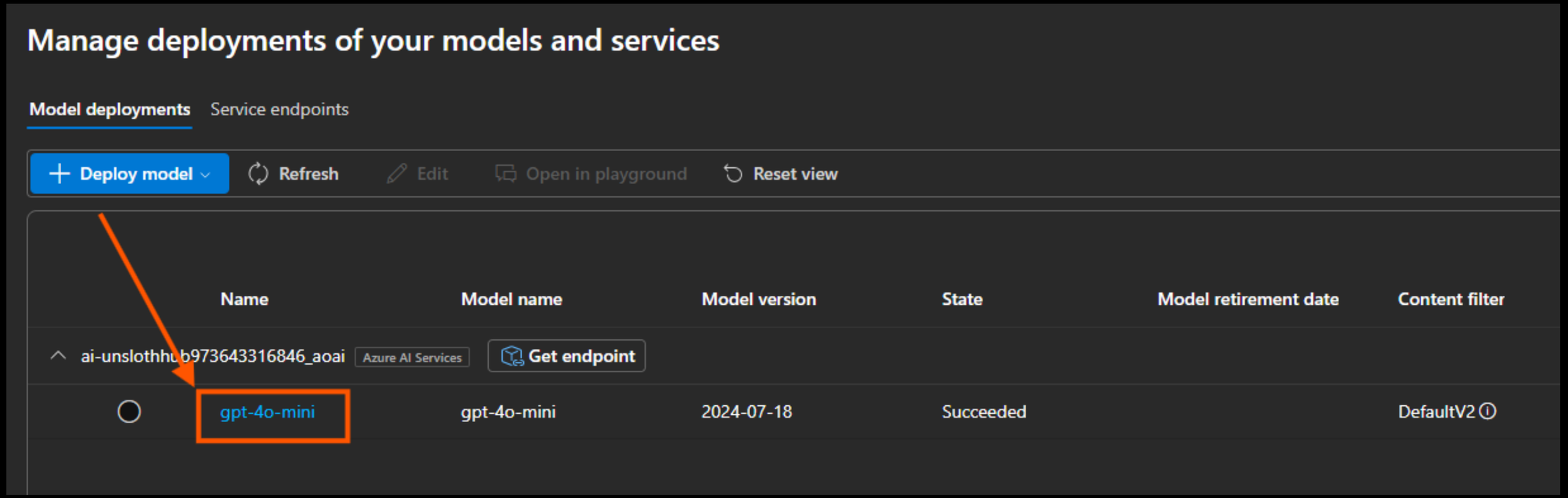
Streamlines the entire AI development process, from exploring and choosing models to deploying and monitoring solutions at scale.



Azure AI Foundry – Deploy AI Model

GPT-4o mini → is the most cost-efficient small model and has vision capabilities. The model has 128K context and an October 2023 knowledge cutoff.

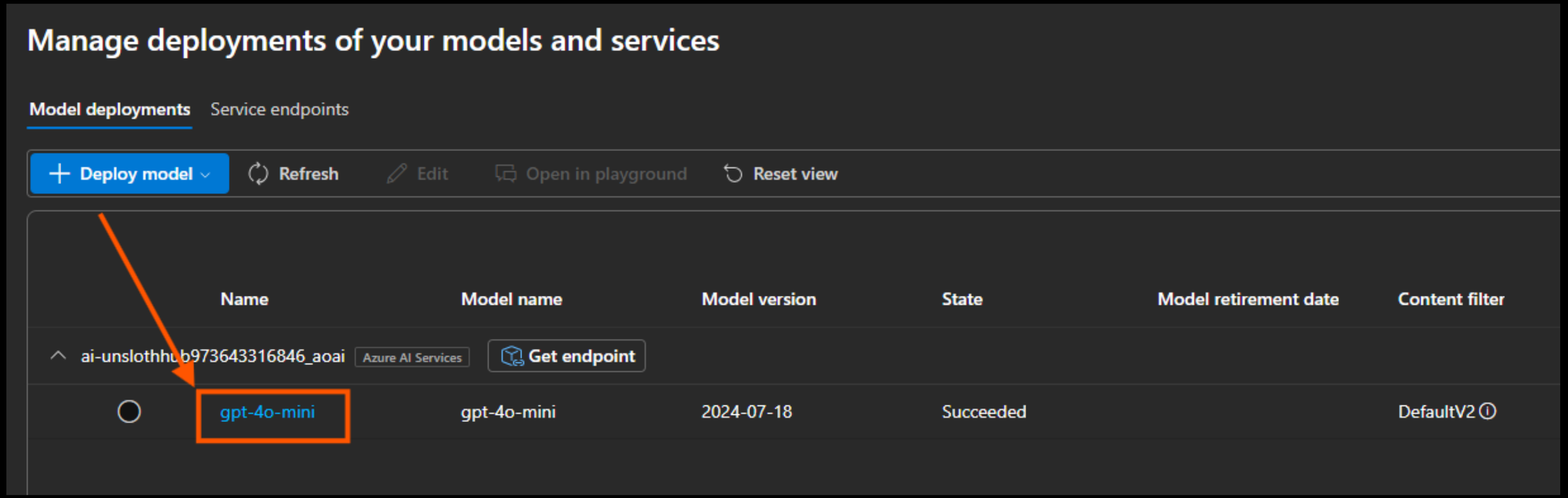
Cost Per 1M Token: Input: **$0.15** → Output: **$0.60**

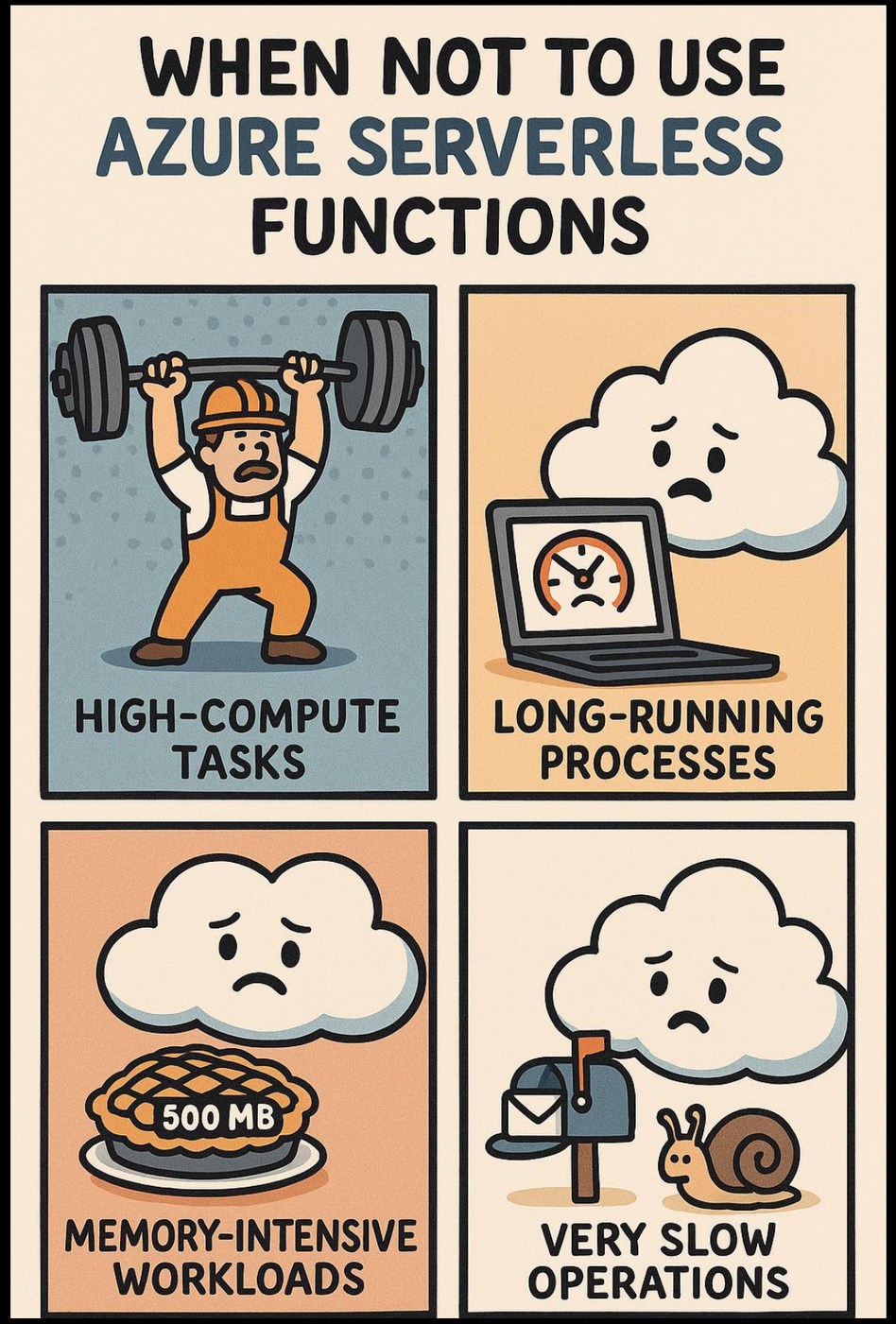


Azure AI Foundry – Deploy AI Model

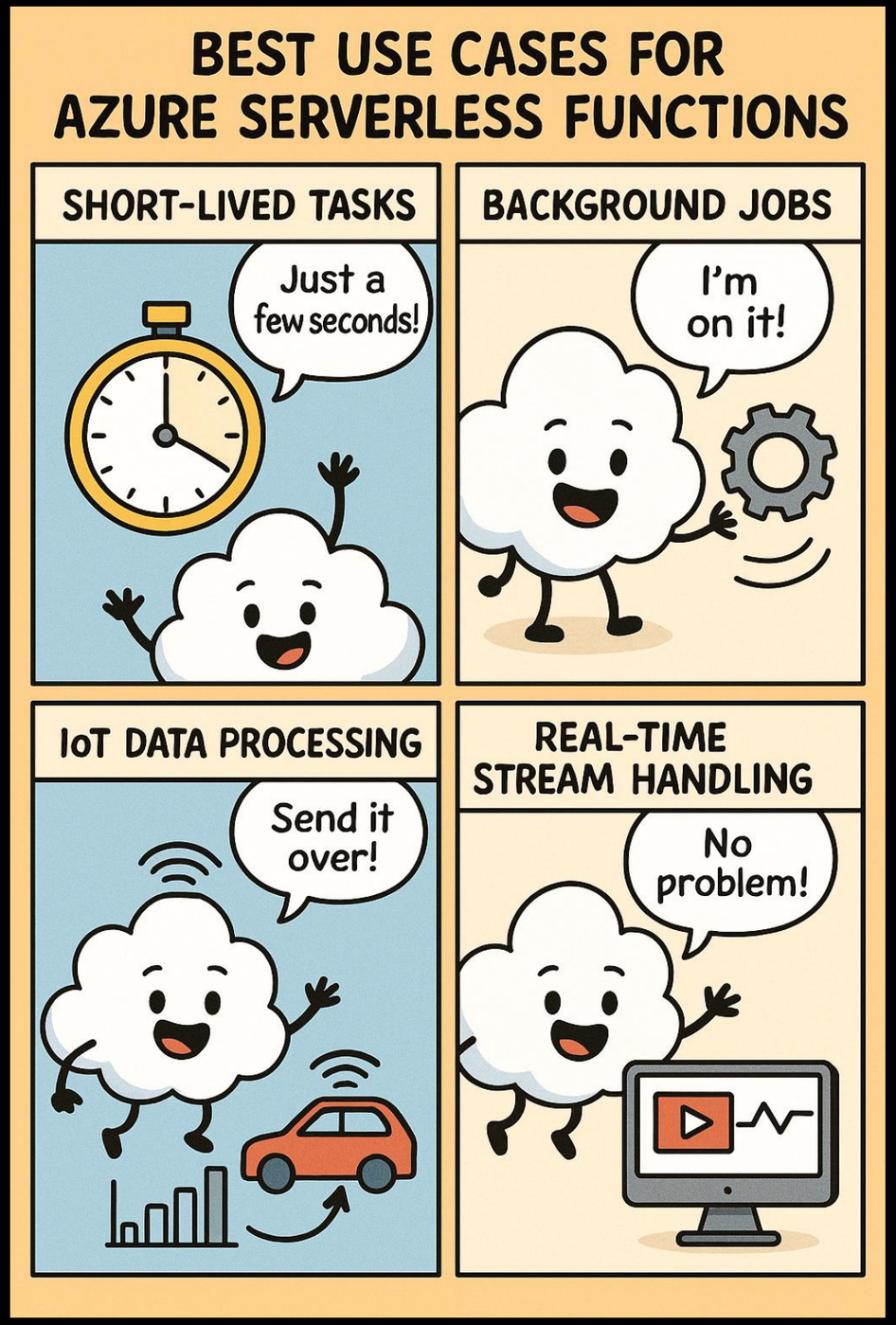
GPT-4o mini → is the most cost-efficient small model and has vision capabilities. The model has 128K context and an October 2023 knowledge cutoff.

Cost Per 1M Token: Input: **$0.15** → Output: **$0.60**





Azure Functions are not ideal for every situation.
You should avoid using them for **long-running processes** or constant **high-compute tasks** that require sustained power. They are also a poor choice for workloads that are very **memory-intensive** or involve slow operations, as serverless platforms have built-in time and resource limits.



Azure Functions are ideal for **short-lived, automated tasks** that run in response to an event. They are perfect for running **background jobs**, such as sending a confirmation email, or for processing continuous streams of data from **IoT devices** and **real-time applications**.

Contact Me!

Website: https://naveen.aifanatic.pro

Email: naveen.aifanatic@gmail.com