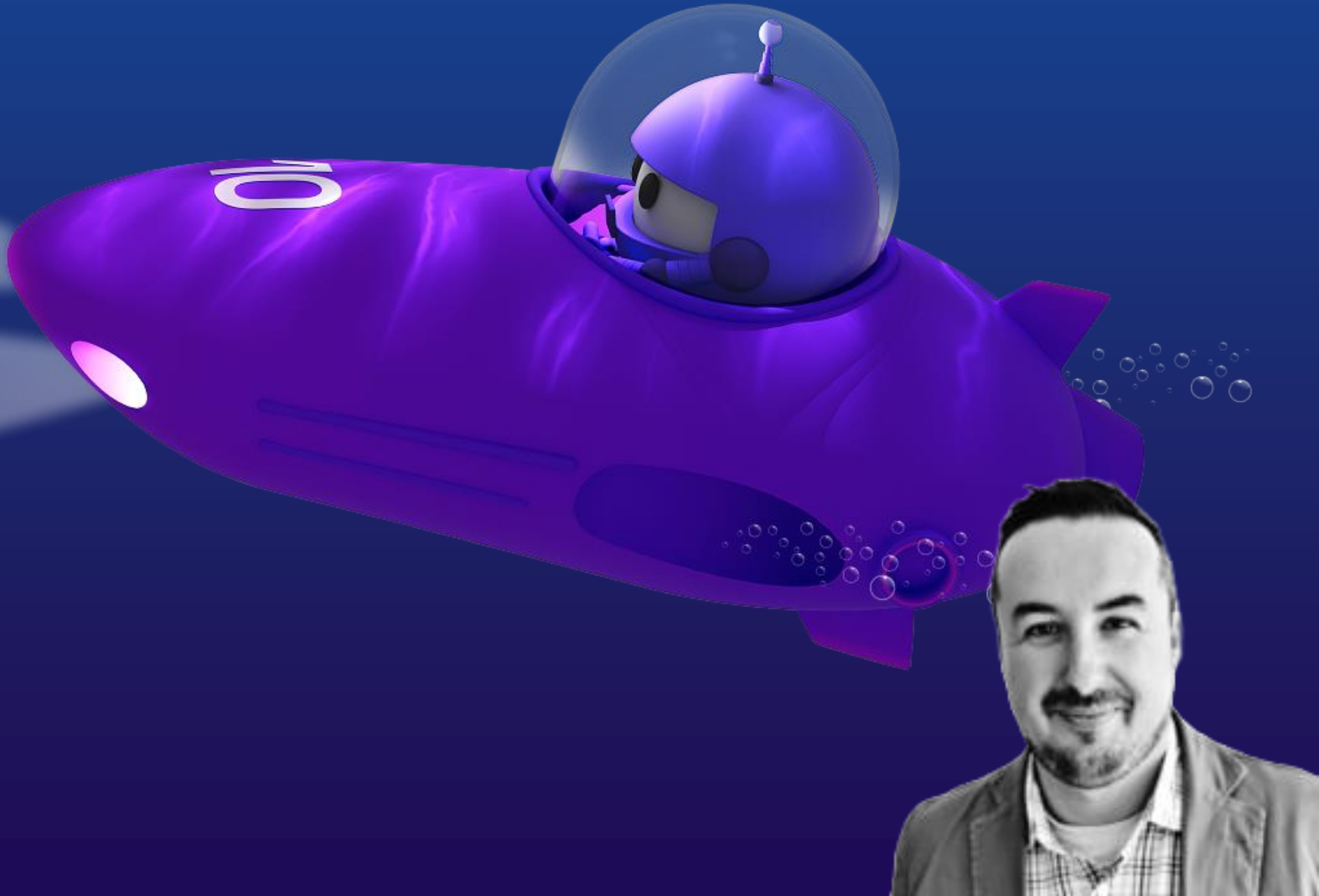
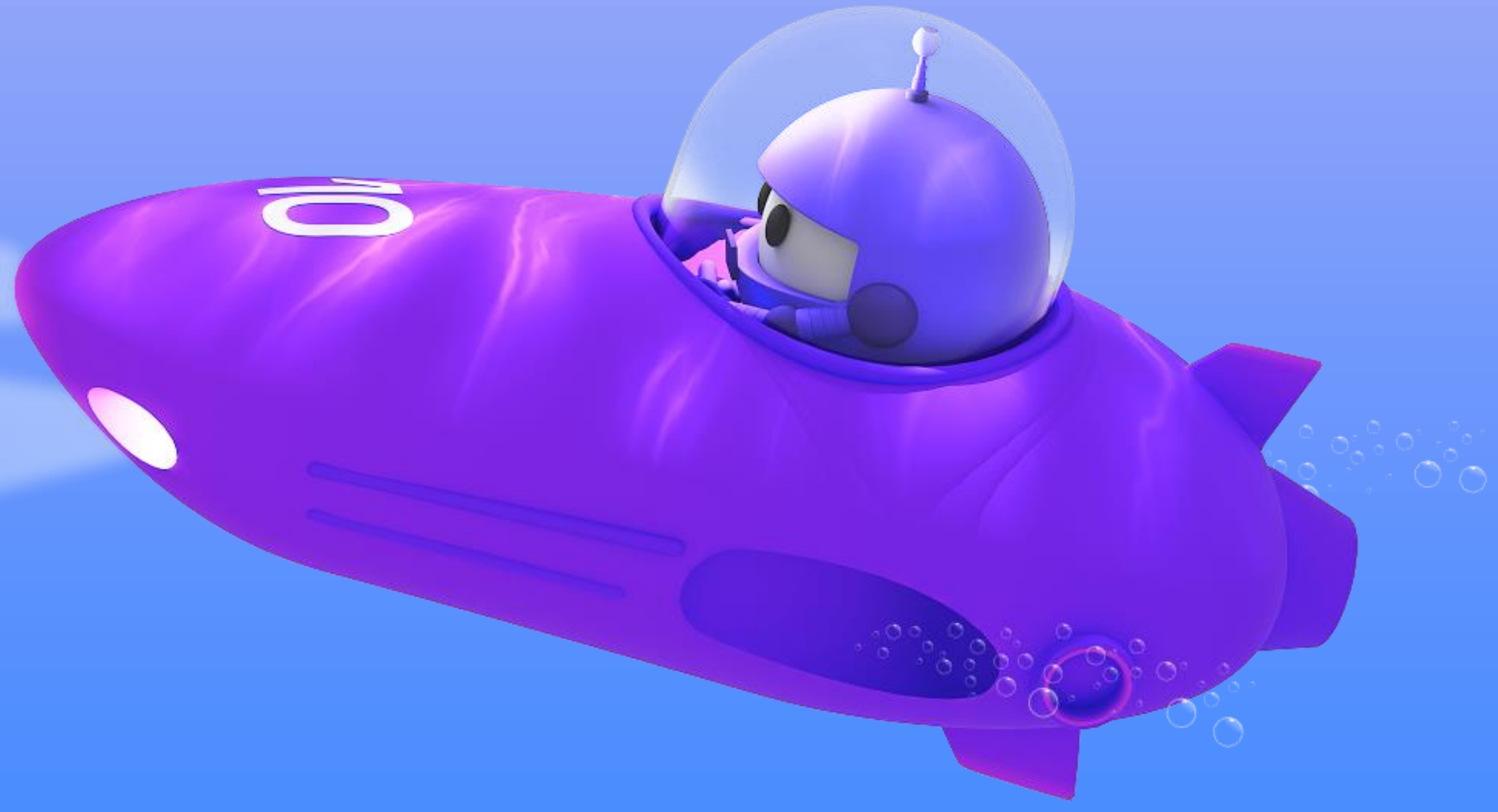
Novità di Aspire

.Net Conf 2025



Aspire
Introduction

.NET



> “
>
> Uno stack di strumenti per la creazione di applicazioni distribuite, moderne, osservabili e pronte per la produzione, sviluppate per il cloud. Fornisce strumenti, modelli e componenti per semplificare lo sviluppo, l'osservabilità e la distribuzione di applicazioni cloud-native.

Generate from AI

Aspire

Main Features

Aspire

Developer Dashboard: Real-time logs, metrics, and resource status for better local diagnostics

Observability: Built-in OpenTelemetry for automatic performance insights

Orchestration: Starts and coordinates multiple services in a distributed app

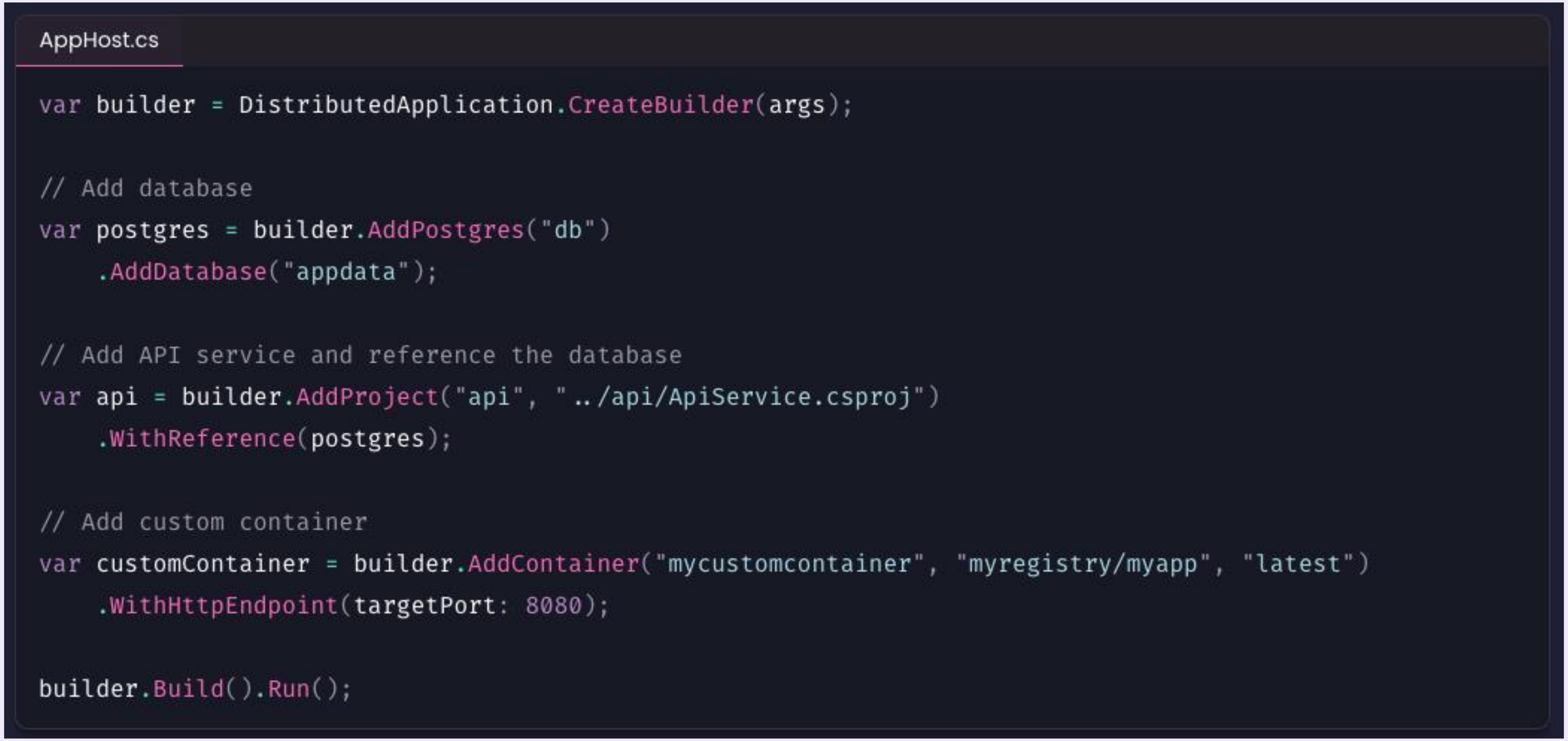
Service Discovery: Enables seamless inter-service communication

Deployment: Works across cloud, Kubernetes, and on-prem environments

Local Development: Unified experience for building and testing before deployment

Aspire

Define your stack in code

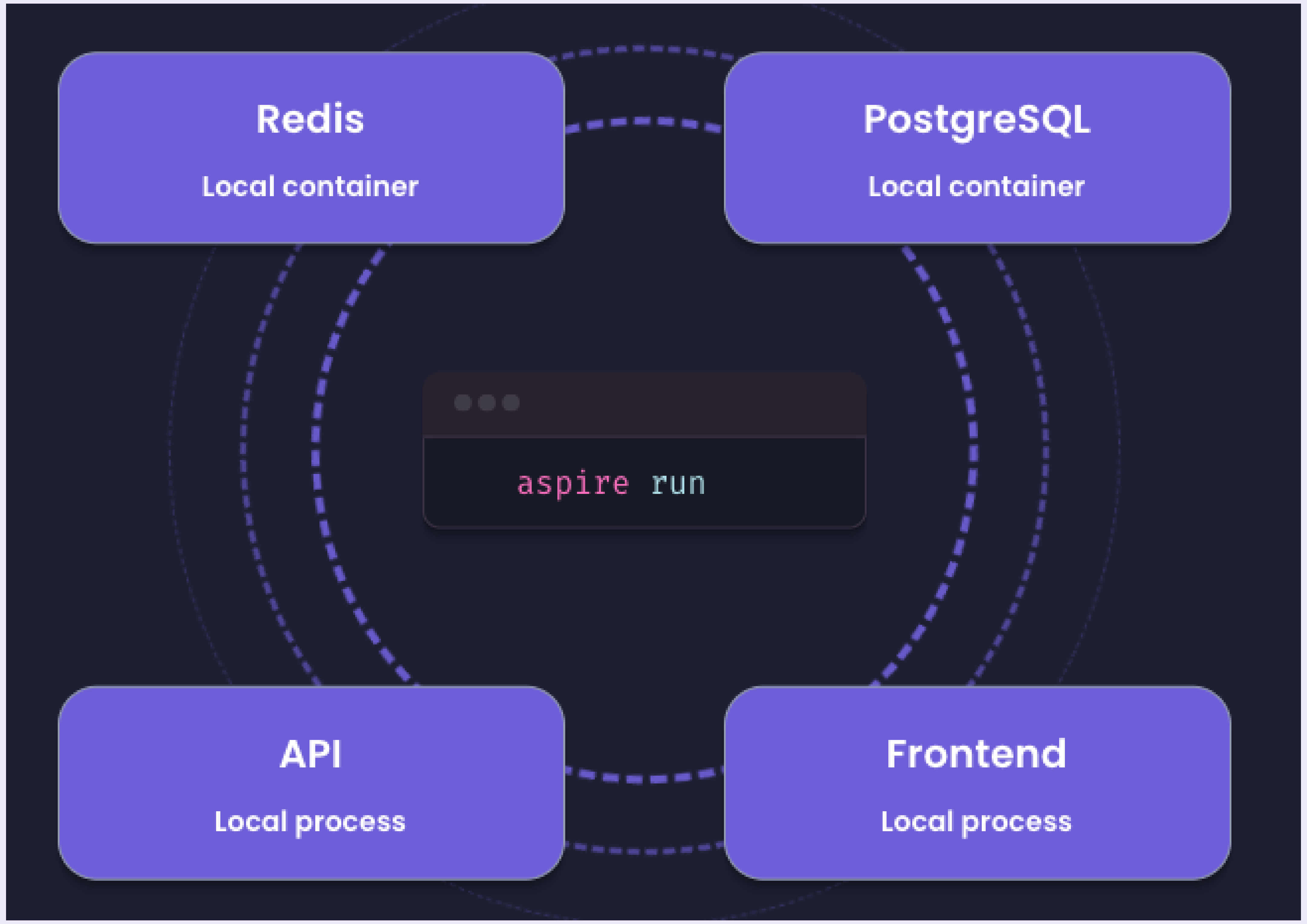


```csharp
var builder = DistributedApplication.CreateBuilder(args);

// Add database
var postgres = builder.AddPostgres("db")
    .AddDatabase("appdata");

// Add API service and reference the database
var api = builder.AddProject("api", "../api/ApiService.csproj")
    .WithReference(postgres);

// Add custom container
var customContainer = builder.AddContainer("mycustomcontainer", "myregistry/myapp", "latest")
    .WithHttpEndpoint(targetPort: 8080);

builder.Build().Run();
```

Aspire

Local-first, production-ready

Local



Redis

Local container

PostgreSQL

Local container

```
aspire run
```

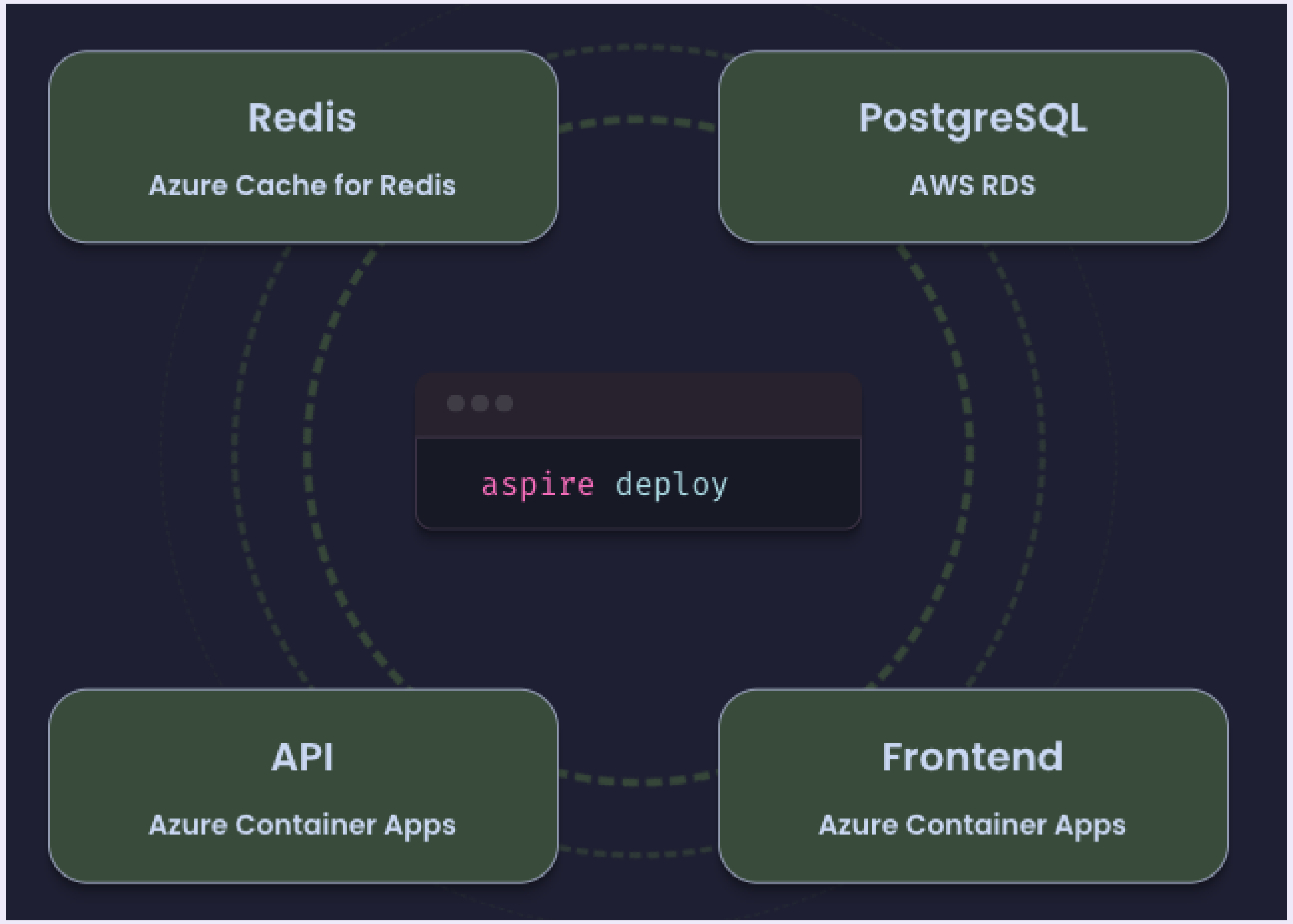
API

Local process

Frontend

Local process

Prod



Redis

Azure Cache for Redis

PostgreSQL

AWS RDS

```
aspire deploy
```

API

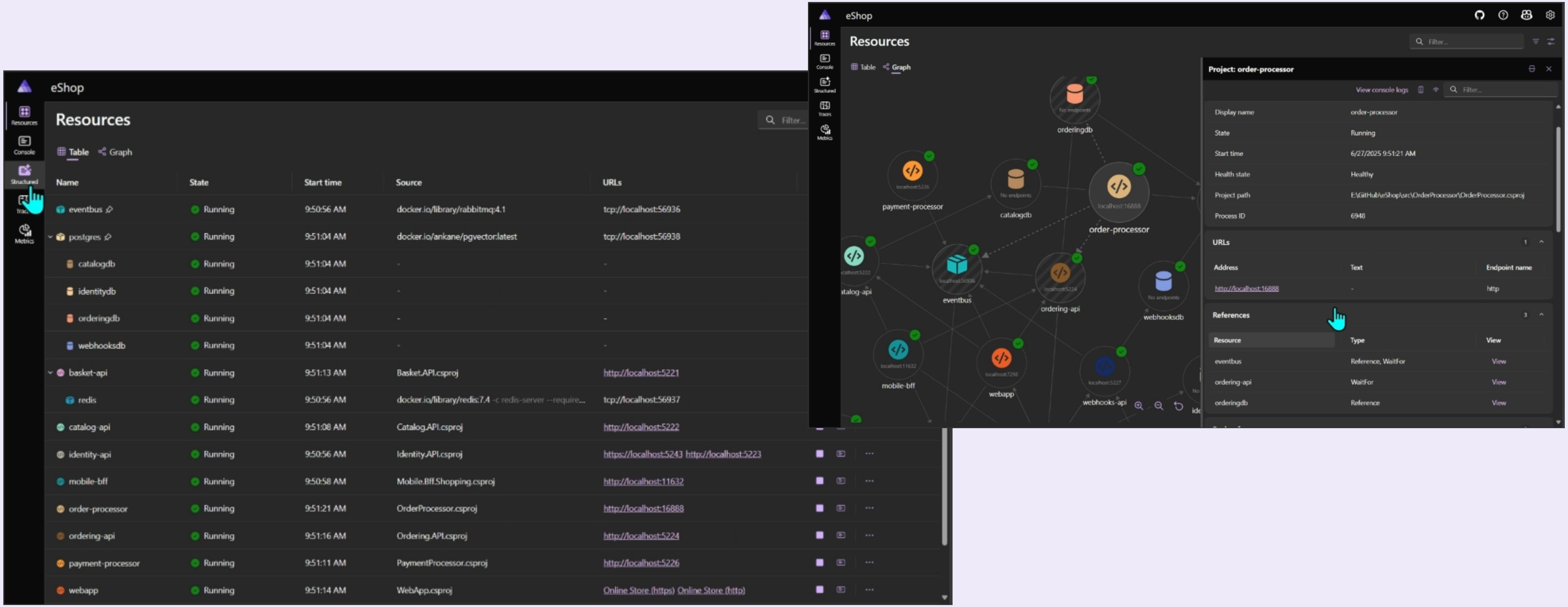
Azure Container Apps

Frontend

Azure Container Apps

Aspire

OpenTelemetry developer dashboard



eShop

Resources

Table Graph

Name	State	Start time	Source	URLs
eventbus	Running	9:50:56 AM	docker.io/library/rabbitmq:4.1	tcp://localhost:56936
postgres	Running	9:51:04 AM	docker.io/ankane/pgvector:latest	tcp://localhost:56938
catalogdb	Running	9:51:04 AM	-	-
identitydb	Running	9:51:04 AM	-	-
orderingdb	Running	9:51:04 AM	-	-
webhooksdb	Running	9:51:04 AM	-	-
basket-api	Running	9:51:13 AM	Basket.API.csproj	http://localhost:5221
redis	Running	9:50:56 AM	docker.io/library/redis:7.4 -c redis-server --require...	tcp://localhost:56937
catalog-api	Running	9:51:08 AM	Catalog.API.csproj	http://localhost:5222
identity-api	Running	9:50:56 AM	Identity.API.csproj	https://localhost:5243 http://localhost:5223
mobile-bff	Running	9:50:58 AM	Mobile.Bff.Shopping.csproj	http://localhost:11632
order-processor	Running	9:51:21 AM	OrderProcessor.csproj	http://localhost:16888
ordering-api	Running	9:51:16 AM	Ordering.API.csproj	http://localhost:5224
payment-processor	Running	9:51:11 AM	PaymentProcessor.csproj	http://localhost:5226
webapp	Running	9:51:14 AM	WebApp.csproj	Online Store (https) Online Store (http)

eShop

Resources

Table Graph

Project: order-processor

View console logs

Display name	order-processor
State	Running
Start time	6/27/2025 9:51:21 AM
Health state	Healthy
Project path	E:\GitHub\eShop\src\OrderProcessor\OrderProcessor.csproj
Process ID	6948

URLs 1

Address	Text	Endpoint name
http://localhost:16888	-	http

References 3

Resource	Type	View
eventbus	Reference, WaitFor	View
ordering-api	WaitFor	View
orderingdb	Reference	View

Aspire

.NET Aspire 8.0

Maggio 2024

General Availability of .NET Aspire.

.NET Aspire 8.2

Agosto 2024

AddDockerfile
AddPythonProject
Community contributions (Keycloack, Elaticsearch, ...)
Metrics exemplars
Span links

.NET Aspire 9.0

Novembre 2024

.NET 9.0 Standard Term Support
Aspire.AppHost.Sdk
Browser telemetry support
WaitFor
Persistent containers
Resource commands
OpenAI (Preview)

.NET Aspire 9.4

Luglio 2025

Aspire CLI
Interaction Service
External service modeling
GitHub Models integration
Azure AI Foundry integration

.NET Aspire 9.5

Settembre 2025

Generative AI visualizer
Multi-resource console logs
Custom resource icons
Trace filtering
OpenAI hosting integration
integration
YARP static files support
Azure App Configuration emulator

Aspire 13.0

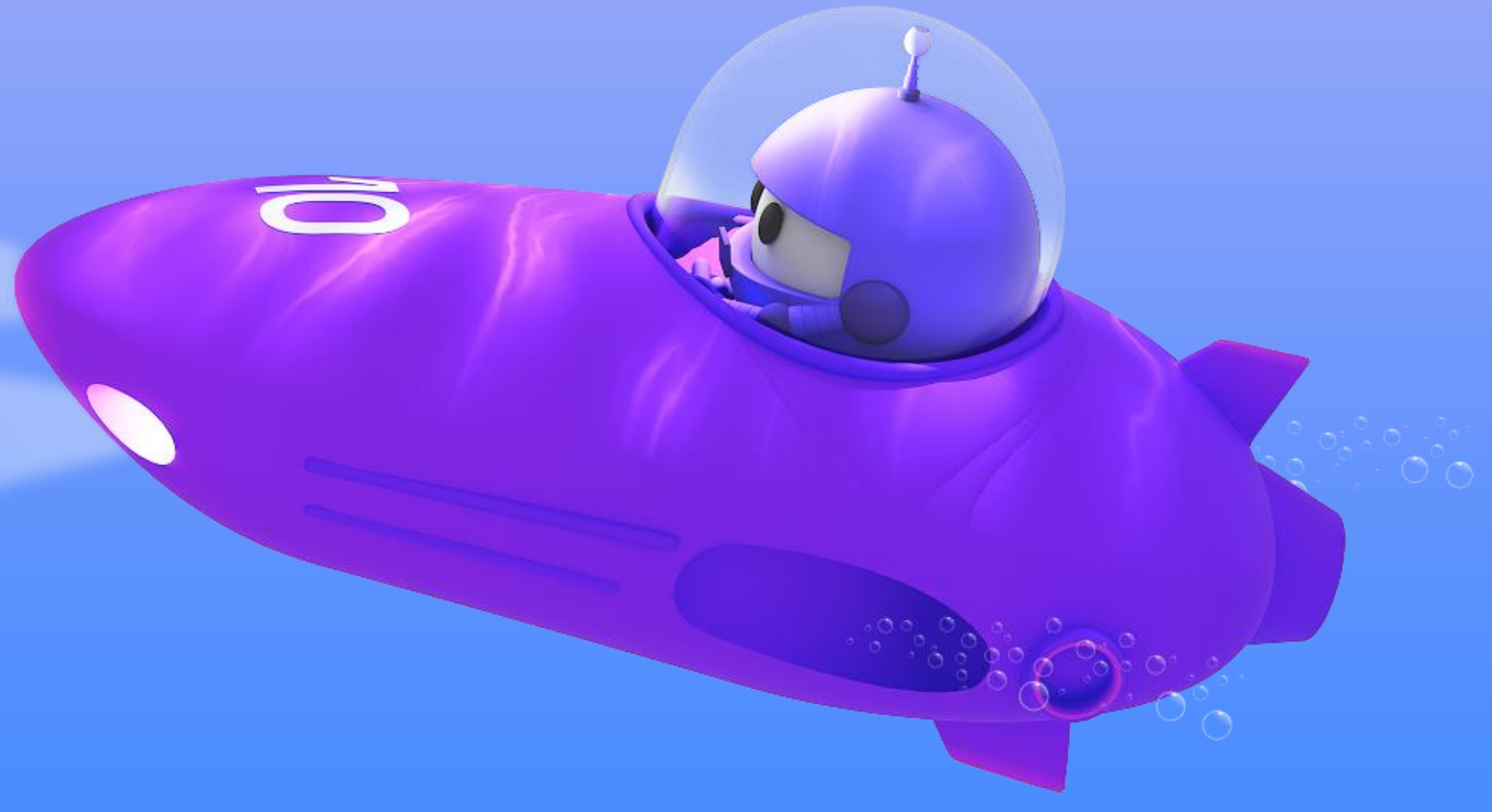
Novembre 2026

Aspire is no longer ".NET Aspire" - it's now simply Aspire
First-class Python support
First-class JavaScript support
Polyglot infrastructure
Container files as build artifacts
aspire do: a new platform for build, publish and deployment pipelines
Modern CLI
VS Code extension

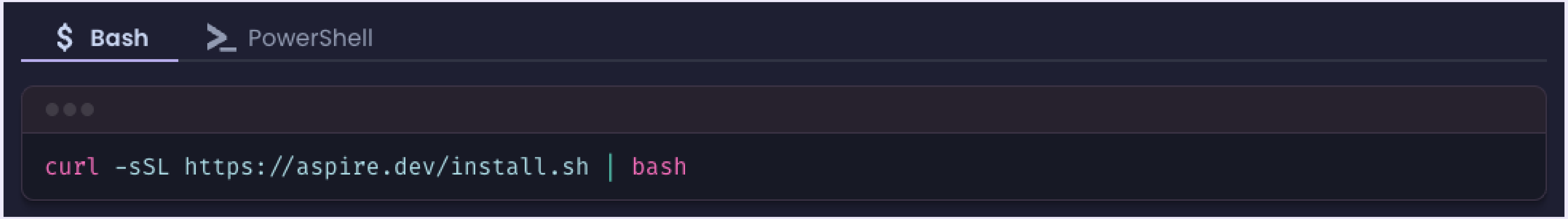
Aspire

Upgrade to Aspire 13.0

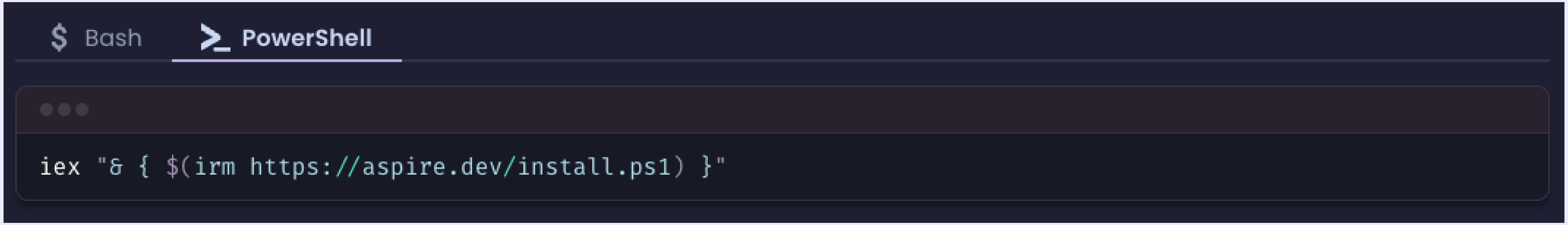
.NET



Update the Aspire CLI to the latest version

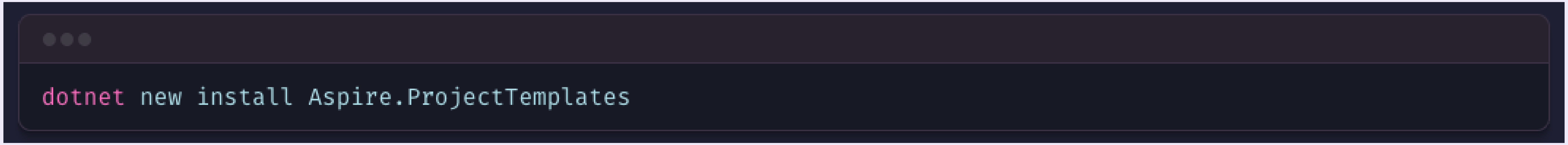


```bash
curl -sSL https://aspire.dev/install.sh | bash
```



```powershell
iex "& { $(irm https://aspire.dev/install.ps1) }"
```

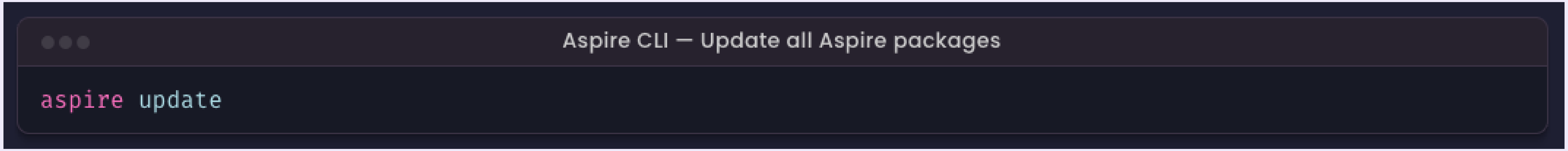
Update your Aspire templates



```
dotnet new install Aspire.ProjectTemplates
```

https://learn.microsoft.com/en-us/dotnet/aspire/get-started/upgrade-to-aspire-13?tabs=bash&pivots=vscode

Update your Aspire project using the **aspire update command**



```
aspire update
```

```
PS C:\Code\eshop>
```

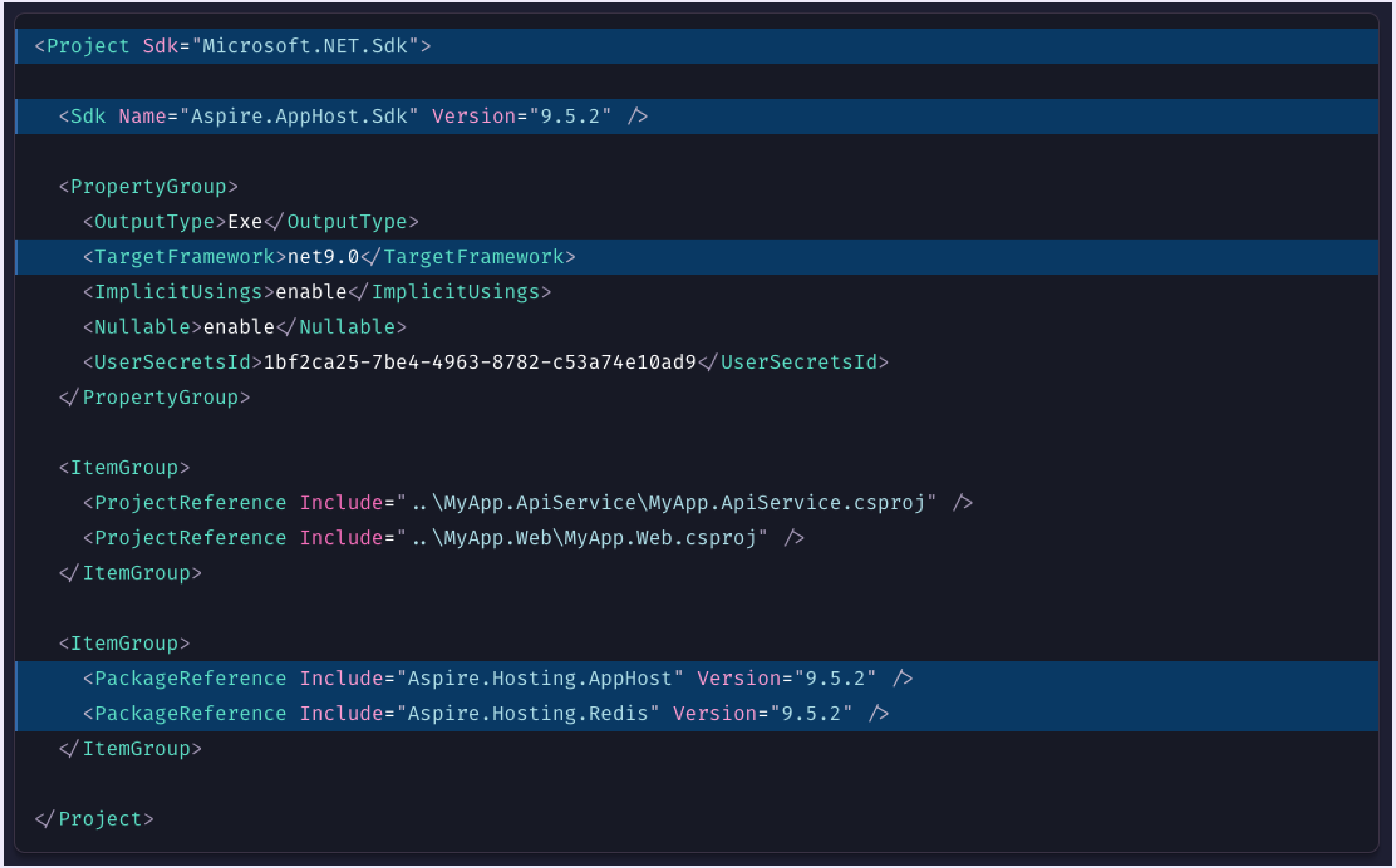
Update the Aspire.AppHost.Sdk version in your AppHost project.

Update all Aspire NuGet packages to version 13.0.

Handle dependency resolution automatically.

AppHost project

Before 9.x

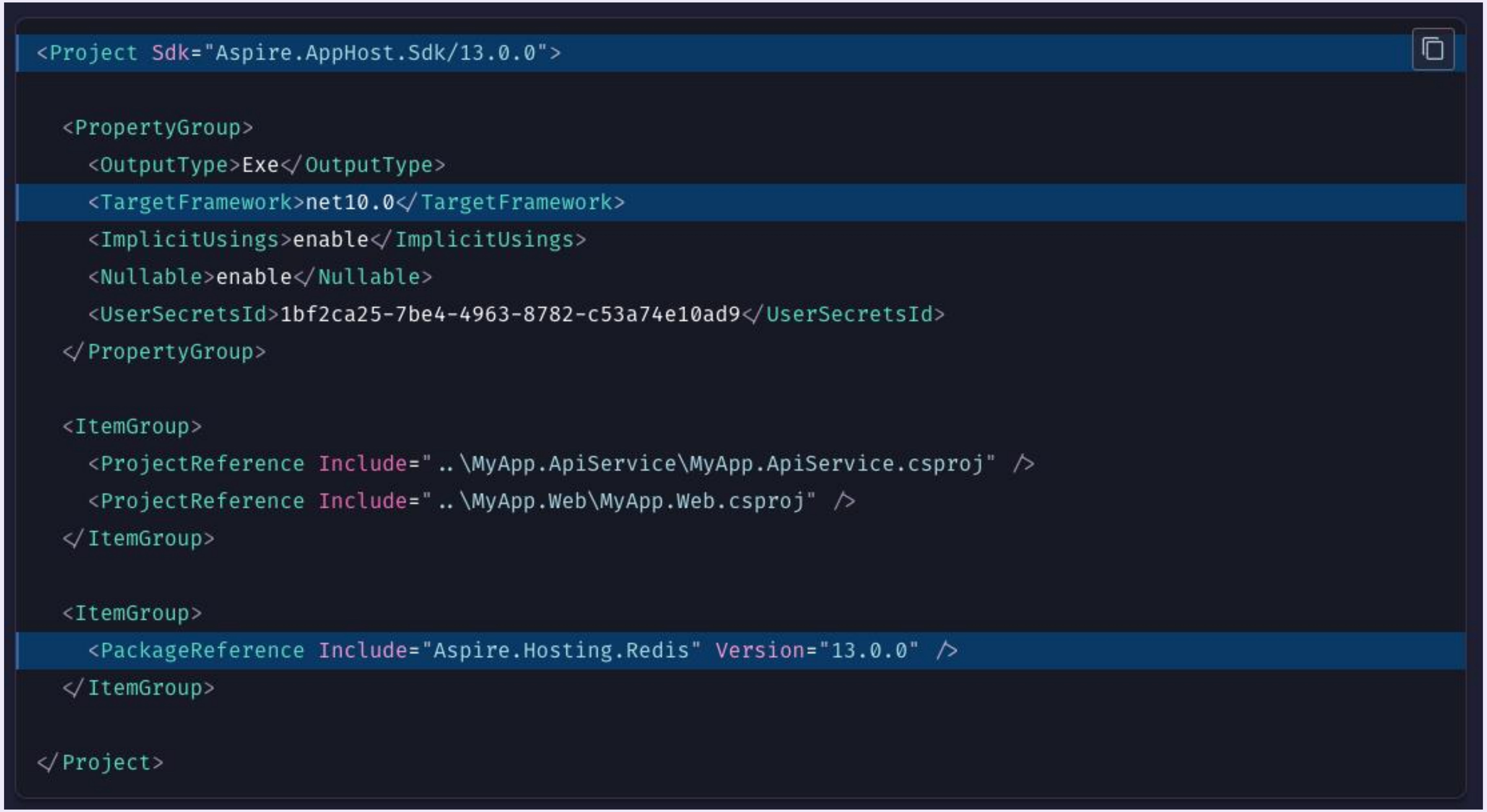


```xml
<Project Sdk="Microsoft.NET.Sdk">

  <Sdk Name="Aspire.AppHost.Sdk" Version="9.5.2" />

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net9.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <UserSecretsId>1bf2ca25-7be4-4963-8782-c53a74e10ad9</UserSecretsId>
  </PropertyGroup>

  <ItemGroup>
    <ProjectReference Include="..\MyApp.ApiService\MyApp.ApiService.csproj" />
    <ProjectReference Include="..\MyApp.Web\MyApp.Web.csproj" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Aspire.Hosting.AppHost" Version="9.5.2" />
    <PackageReference Include="Aspire.Hosting.Redis" Version="9.5.2" />
  </ItemGroup>

</Project>
```

AppHost project

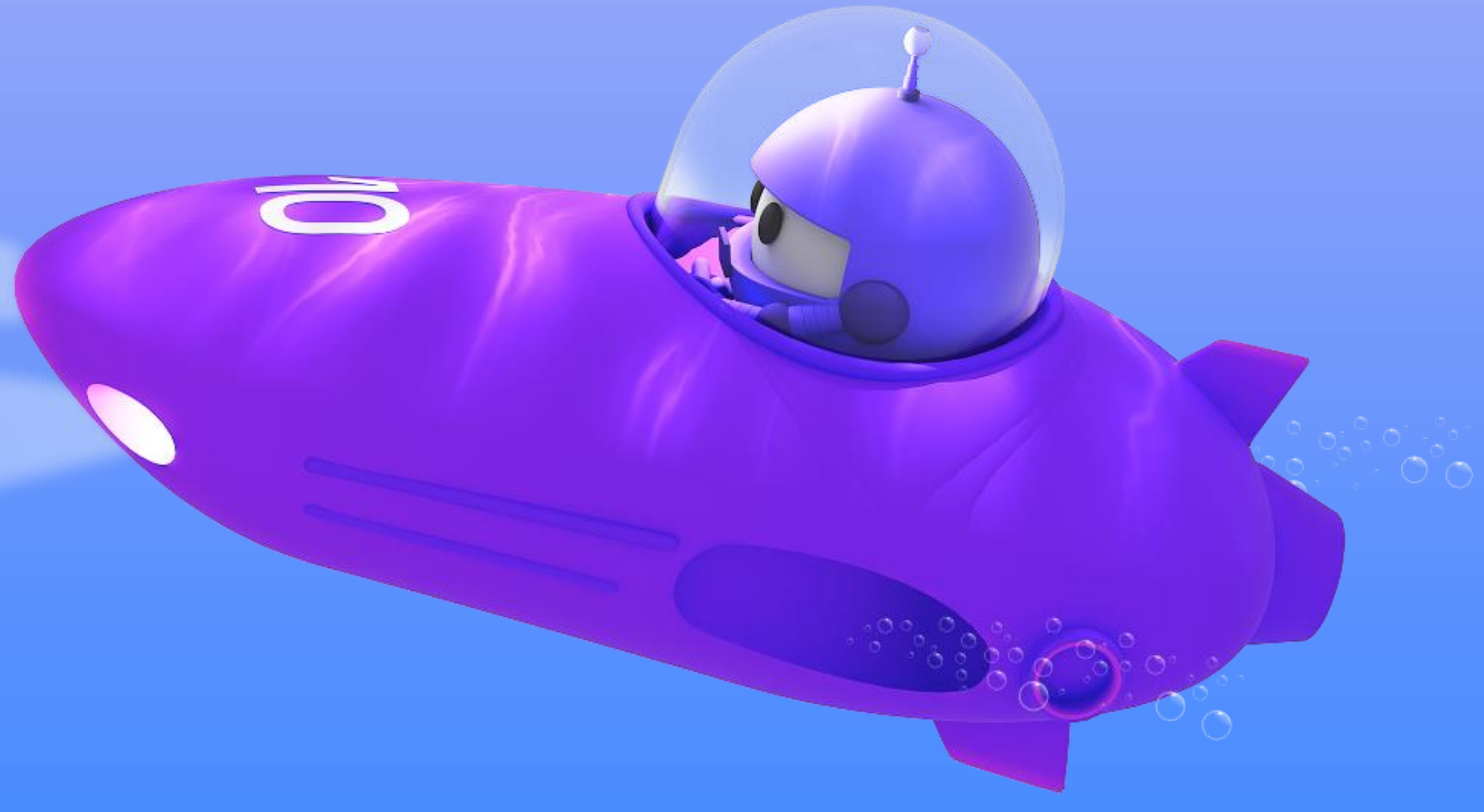
After 13.0

Aspire 13.0 introduces a simplified AppHost project template structure. The SDK now encapsulates the Aspire.Hosting.AppHost package, resulting in cleaner project files



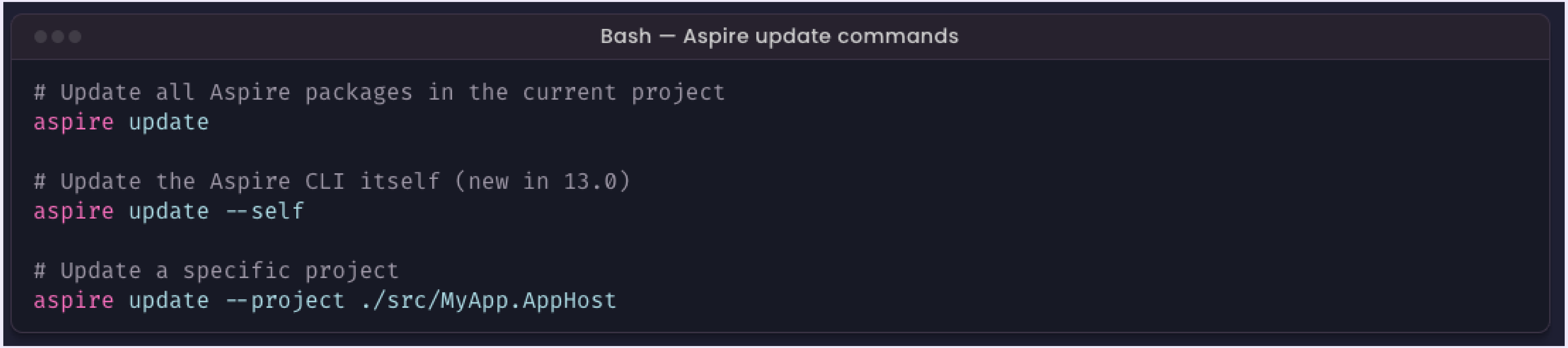
```xml
<Project Sdk="Aspire.AppHost.Sdk/13.0.0">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net10.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <UserSecretsId>1bf2ca25-7be4-4963-8782-c53a74e10ad9</UserSecretsId>
  </PropertyGroup>

  <ItemGroup>
    <ProjectReference Include="..\MyApp.ApiService\MyApp.ApiService.csproj" />
    <ProjectReference Include="..\MyApp.Web\MyApp.Web.csproj" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Aspire.Hosting.Redis" Version="13.0.0" />
  </ItemGroup>

</Project>
```

Aspire 13
What's New



🛠️ CLI and tooling

Aspire update improvements

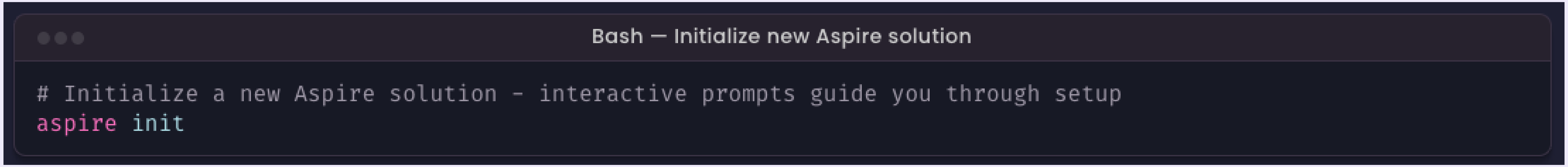


```bash
# Update all Aspire packages in the current project
aspire update

# Update the Aspire CLI itself (new in 13.0)
aspire update --self

# Update a specific project
aspire update --project ./src/MyApp.AppHost
```

CLI self-update: The --self flag allows you to update the Aspire CLI without reinstalling.
Improved reliability: Numerous bug fixes for edge cases in dependency resolution.
Better error handling: Clearer error messages when updates fail.
Performance improvements: Faster package detection and update operations.

🛠️ CLI and tooling

aspire init command



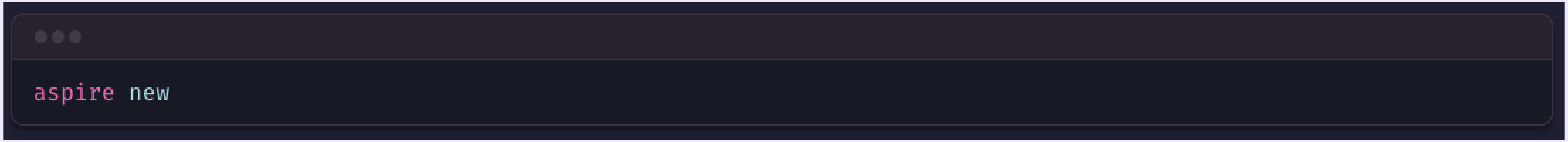
Bash — Initialize new Aspire solution

```bash
# Initialize a new Aspire solution - interactive prompts guide you through setup
aspire init
```

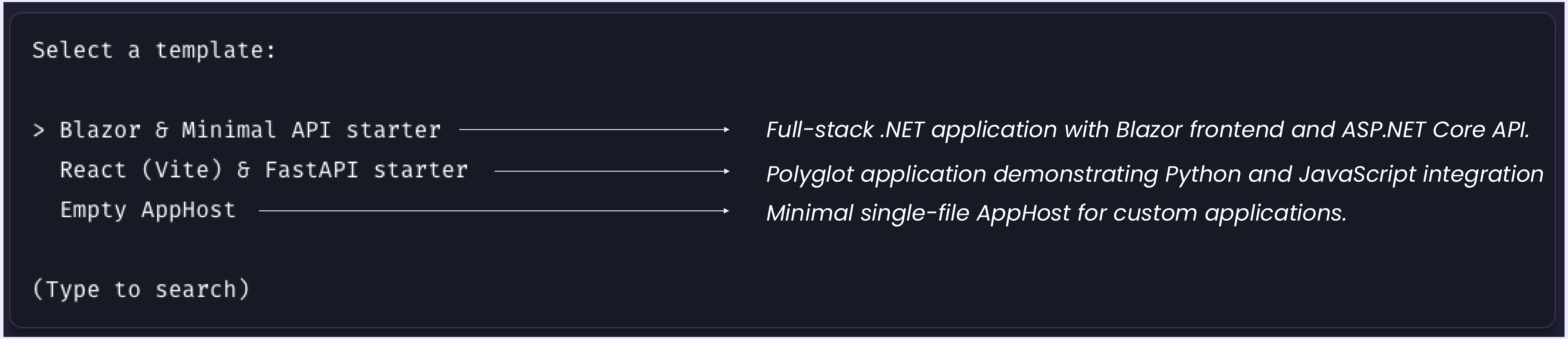
Discover existing solutions: Automatically finds and updates solution files in the current directory.
Create single-file AppHost: If no solution exists, creates a single-file AppHost for quick starts.
Add projects intelligently: Prompts to add projects to your app host.
Configure service defaults: Sets up service defaults referencing automatically.
Setup NuGet.config: Creates package source mappings for Aspire packages.
Manage template versions: Interactively selects the appropriate template version.

🛠️ CLI and tooling

aspire new command



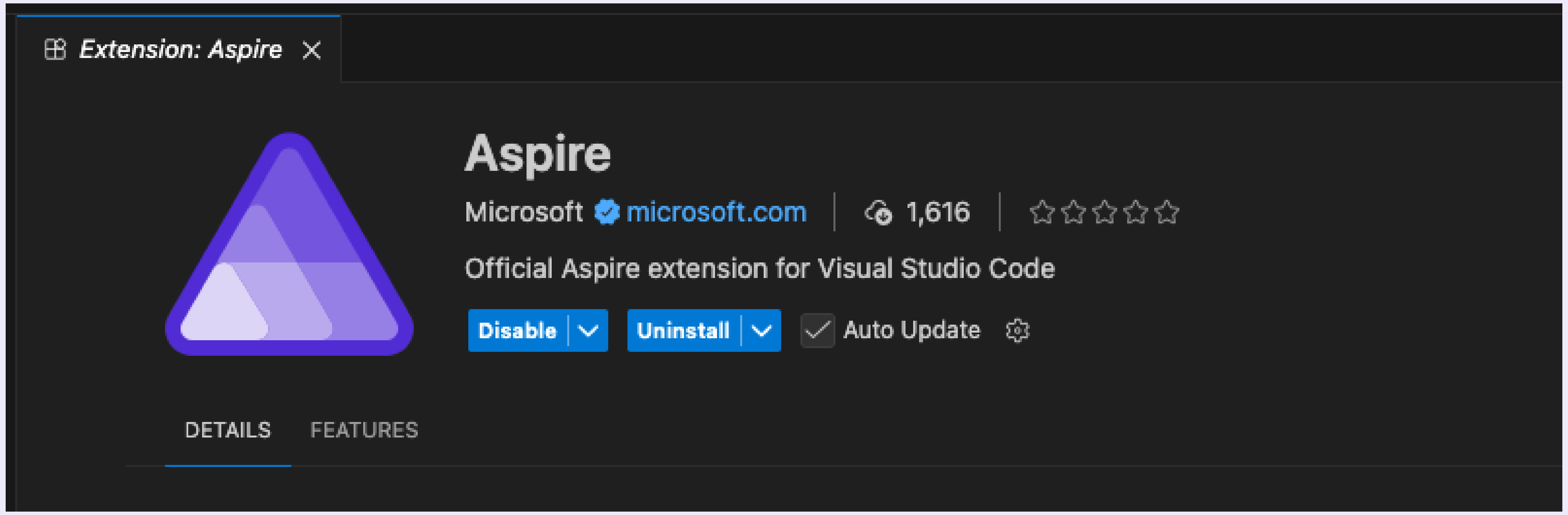
```
aspire new
```



```
Select a template:

> Blazor & Minimal API starter → Full-stack .NET application with Blazor frontend and ASP.NET Core API.
  React (Vite) & FastAPI starter → Polyglot application demonstrating Python and JavaScript integration
  Empty AppHost → Minimal single-file AppHost for custom applications.

(Type to search)
```

🛠️ CLI and tooling

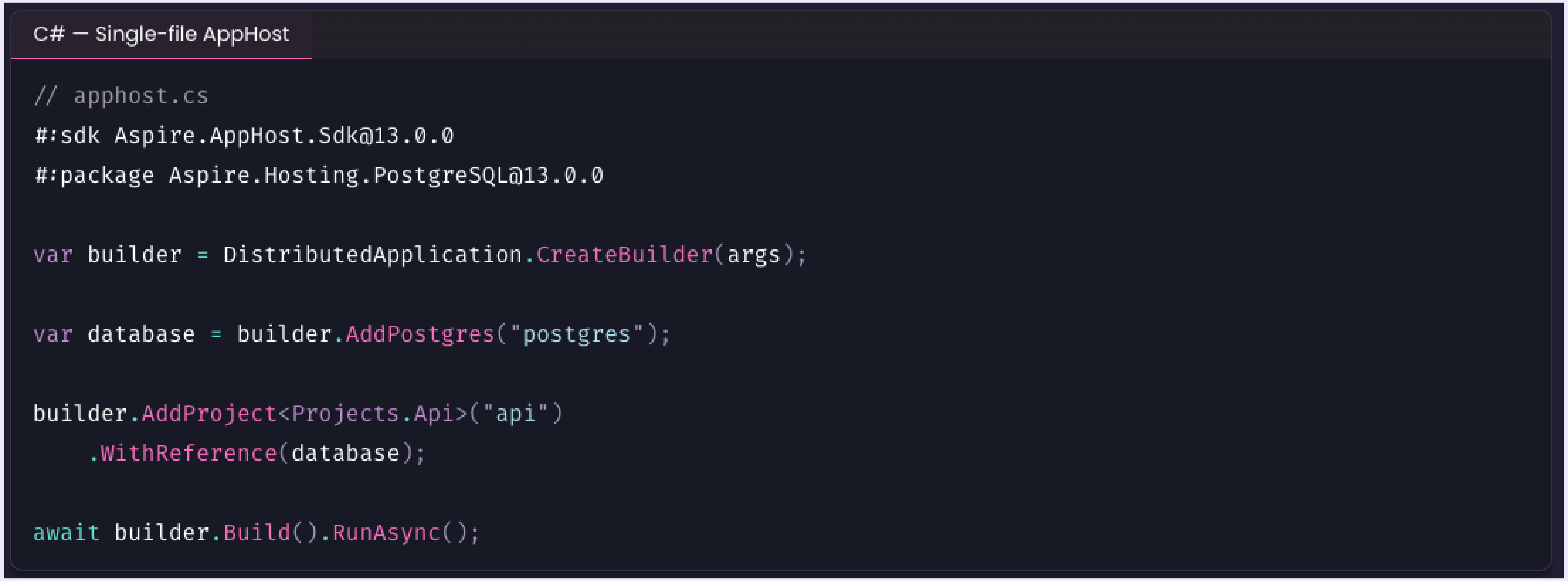
Visual Studio Code extension



Debug Python and C# projects inside VS Code: Launch your apphost using the Aspire debugger to launch and debug any C# and Python resources in your app.
Project creation: Use Aspire: New Aspire project to create new Aspire projects from templates.
Integration management: Use Aspire: Add an integration to add Aspire integrations to your AppHost.
Launch configuration: Use Aspire: Configure launch.json to automatically set up a VS Code launch configuration.
Configuration management: Use Aspire: Manage configuration settings to configure Aspire CLI settings.
Publish and deployment: Use Aspire: Publish deployment artifacts and Aspire: Deploy app commands (preview).

🛠️ CLI and tooling

Single-file
AppHost support



C# — Single-file AppHost

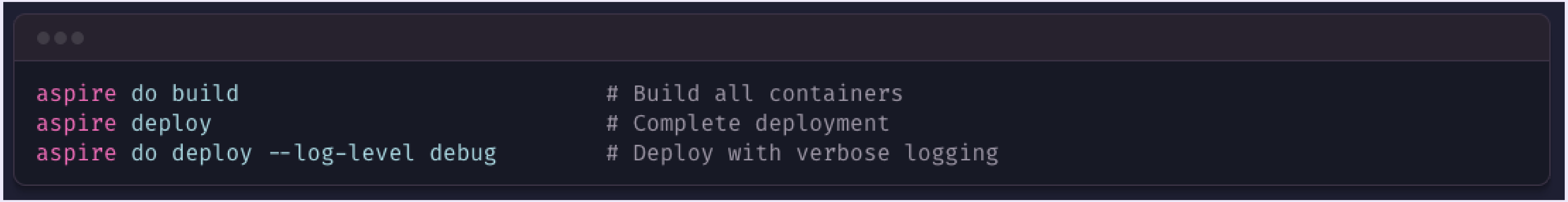
```csharp
// apphost.cs
#:sdk Aspire.AppHost.Sdk@13.0.0
#:package Aspire.Hosting.PostgreSQL@13.0.0

var builder = DistributedApplication.CreateBuilder(args);

var database = builder.AddPostgres("postgres");

builder.AddProject<Projects.Api>("api")
    .WithReference(database);

await builder.Build().RunAsync();
```

No project file needed - just a single .cs file with package references declared using #:package directives.

Demo

🚀 **Deployment improvements**

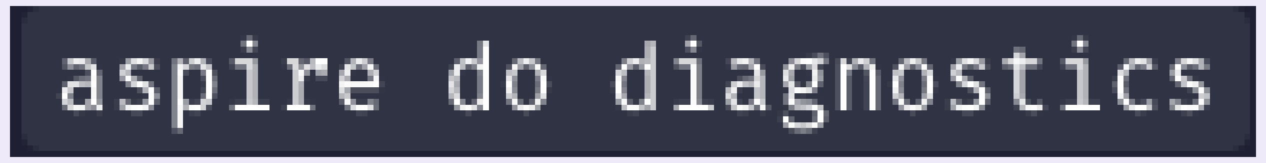
Deployment pipeline reimplementation



```
aspire do build                          # Build all containers
aspire deploy                            # Complete deployment
aspire do deploy --log-level debug       # Deploy with verbose logging
```

Aspire 13.0 completely reimplements the deployment workflow on top of aspire do. This architectural change transforms deployment from a monolithic operation into a composable set of discrete, parallelizable steps.

The new deployment pipeline automatically parallelizes independent operations, dramatically reducing deployment time. Steps like prerequisites, builds, and provisioning run concurrently when dependencies allow.

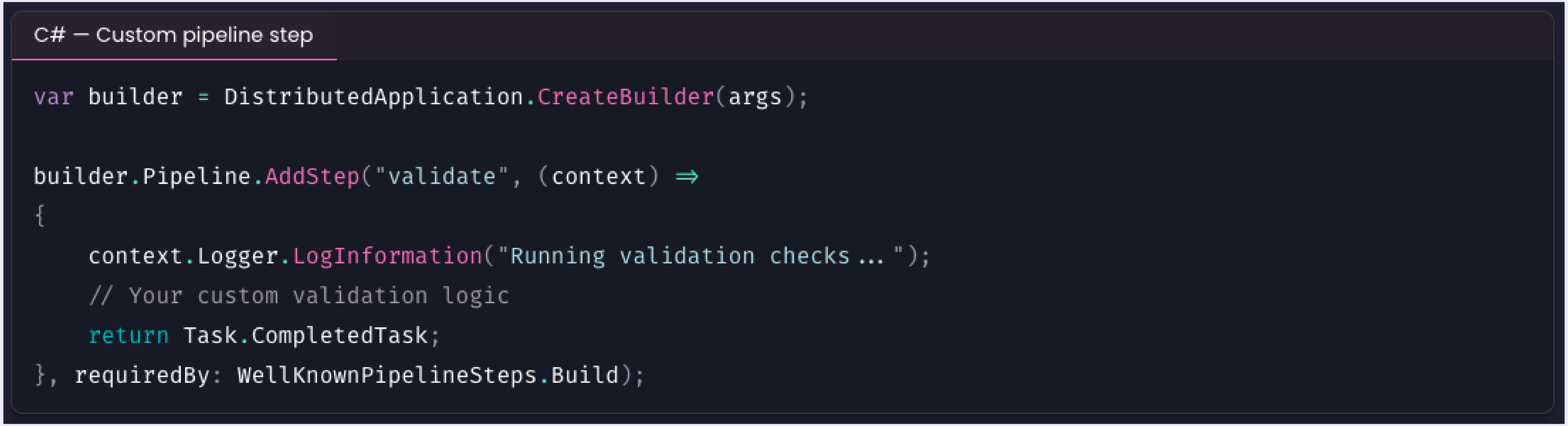


```
aspire do diagnostics
```

To view execution order with parallelization indicators, see step dependencies and resources, simulate “what if” scenarios

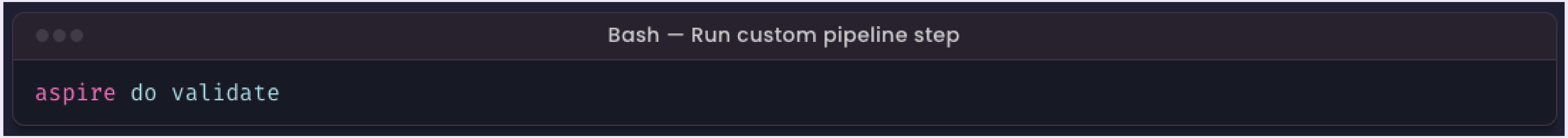
⭐ Major new features

aspire do



C# — Custom pipeline step

```csharp
var builder = DistributedApplication.CreateBuilder(args);

builder.Pipeline.AddStep("validate", (context) =>
{
    context.Logger.LogInformation("Running validation checks...");
    // Your custom validation logic
    return Task.CompletedTask;
}, requiredBy: WellKnownPipelineSteps.Build);
```



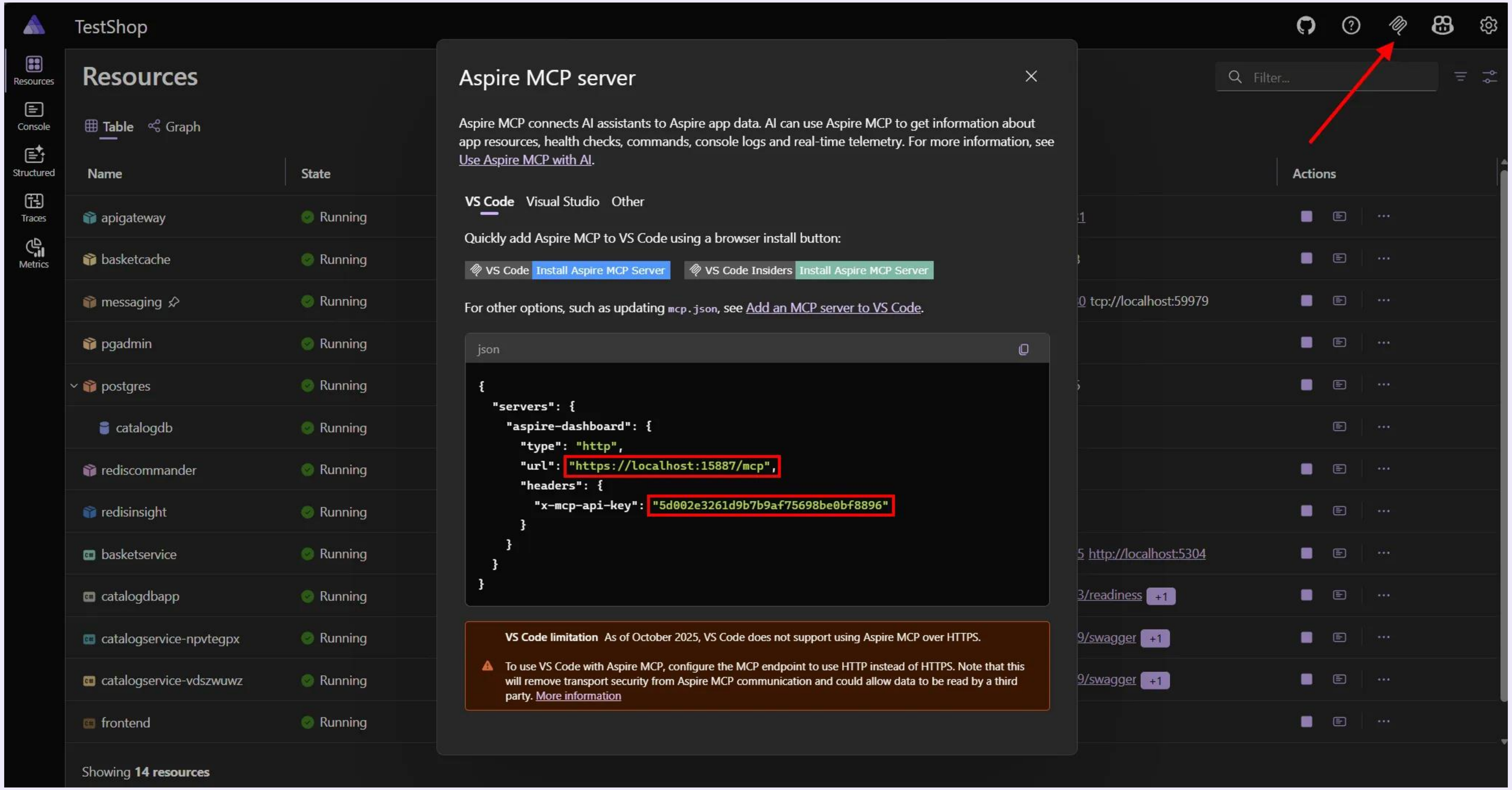
Bash — Run custom pipeline step

```bash
aspire do validate
```

Demo

📊 Dashboard enhancements

Aspire MCP server



Resources

Name	State		Actions
apigateway	Running	1	
basketcache	Running	3	
messaging	Running	0 tcp://localhost:59979	
pgadmin	Running		
postgres	Running	5	
catalogdb	Running		
rediscommander	Running		
redisinsight	Running		
basketservice	Running	5 http://localhost:5304	
catalogdbapp	Running	3/readiness +1	
catalogservice-npvtegpx	Running	9/swagger +1	
catalogservice-vdszwuwz	Running	9/swagger +1	
frontend	Running		

Showing 14 resources

Aspire MCP server

Aspire MCP connects AI assistants to Aspire app data. AI can use Aspire MCP to get information about app resources, health checks, commands, console logs and real-time telemetry. For more information, see Use Aspire MCP with AI.

VS Code Visual Studio Other

Quickly add Aspire MCP to VS Code using a browser install button:

VS Code Install Aspire MCP Server   VS Code Insiders Install Aspire MCP Server

For other options, such as updating `mcp.json`, see Add an MCP server to VS Code.

json

```json
{
  "servers": {
    "aspire-dashboard": {
      "type": "http",
      "url": "https://localhost:15887/mcp",
      "headers": {
        "x-mcp-api-key": "5d002e3261d9b7b9af75698be0bf8896"
      }
    }
  }
}
```

VS Code limitation As of October 2025, VS Code does not support using Aspire MCP over HTTPS.

⚠ To use VS Code with Aspire MCP, configure the MCP endpoint to use HTTP instead of HTTPS. Note that this will remove transport security from Aspire MCP communication and could allow data to be read by a third party. More information

Demo

📦 Integrations

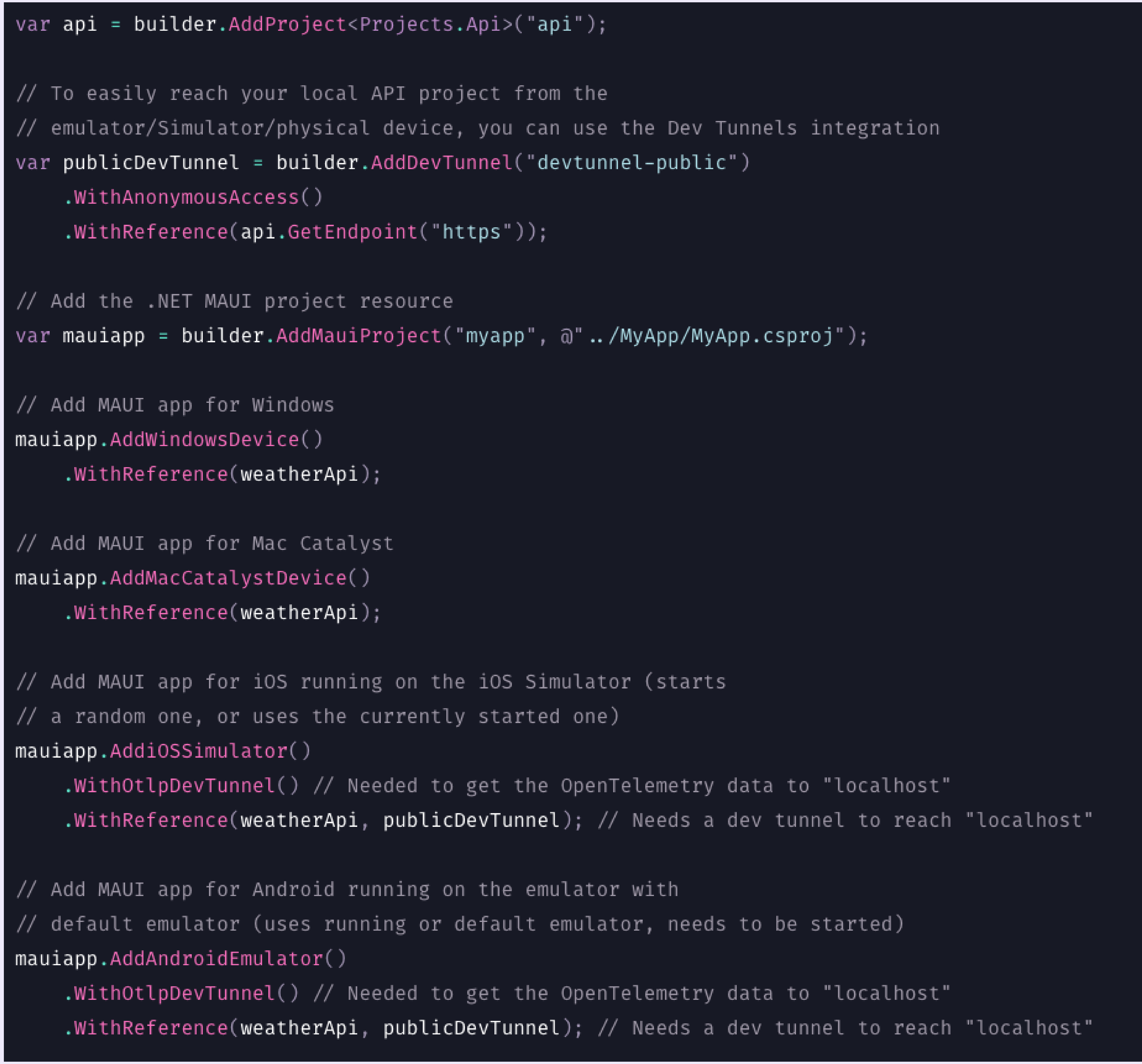
.NET MAUI integration

Platform support: Windows, Mac Catalyst, Android, and iOS

Device registration: Register multiple device instances for testing

Platform validation: Automatically detects host OS compatibility and marks resources as unsupported when needed

Full orchestration: MAUI apps participate in service discovery and can reference backend services



```
var api = builder.AddProject<Projects.Api>("api");

// To easily reach your local API project from the
// emulator/Simulator/physical device, you can use the Dev Tunnels integration
var publicDevTunnel = builder.AddDevTunnel("devtunnel-public")
    .WithAnonymousAccess()
    .WithReference(api.GetEndpoint("https"));

// Add the .NET MAUI project resource
var mauiapp = builder.AddMauiProject("myapp", @"../MyApp/MyApp.csproj");

// Add MAUI app for Windows
mauiapp.AddWindowsDevice()
    .WithReference(weatherApi);

// Add MAUI app for Mac Catalyst
mauiapp.AddMacCatalystDevice()
    .WithReference(weatherApi);

// Add MAUI app for iOS running on the iOS Simulator (starts
// a random one, or uses the currently started one)
mauiapp.AddiOSSimulator()
    .WithOtlpDevTunnel() // Needed to get the OpenTelemetry data to "localhost"
    .WithReference(weatherApi, publicDevTunnel); // Needs a dev tunnel to reach "localhost"

// Add MAUI app for Android running on the emulator with
// default emulator (uses running or default emulator, needs to be started)
mauiapp.AddAndroidEmulator()
    .WithOtlpDevTunnel() // Needed to get the OpenTelemetry data to "localhost"
    .WithReference(weatherApi, publicDevTunnel); // Needs a dev tunnel to reach "localhost"
```

Demo

Aspire 13

Additional innovations

Model enhancements

- Network identifiers
- Interaction service dynamic inputs
- Interaction service custom choices
  Reference and connection improvements
- Connection properties
- Endpoint reference enhancements
- Other app model improvements

Major new features

- Container files as build artifacts
- Dockerfile builder API (Exp.)
- Certificate management

CLI and Toolin

- Automatic .NET SDK installation (Preview)
- Non-interactive mode for CI/CD

Aspire as a polyglot platform

- VS Code debugging support
- Automatic Dockerfile generation
- Python version detection
- Package manager flexibility
- JS Customizing scripts
- Node support
- Polyglot infrastructure

Azure

- Azure tenant selection
- Aspire dashboard in App Service
- Application Insights integration

Deployment improvements

- Deployment state management
- Deployment pipeline reimplementation

Dashboard enhancements

- Interaction services dynamic inputs and comboboxes
- Polyglot language icons
- Improved accent colors
- Health checks last run time

Breaking changes

- Package renames
- Removed APIs
- Obsolete APIs
- Changed signatures
- Major architectural changes

Aspire 13 Resources

What's new in Aspire 13	https://aspire.dev/whats-new/aspire-13
aspire-samples	https://github.com/dotnet/aspire-samples
Community Toolkit	https://github.com/CommunityToolkit/Aspire
Integrations gallery	https://aspire.dev/integrations/gallery

THANK YOU!

Marco Bortolin

email: m.bortolin@hunext.com
twitter: @marcobortolin

https://github.com/bortolin
https://www.linkedin.com/in/marcobortolin

hunext
SOFTWARE