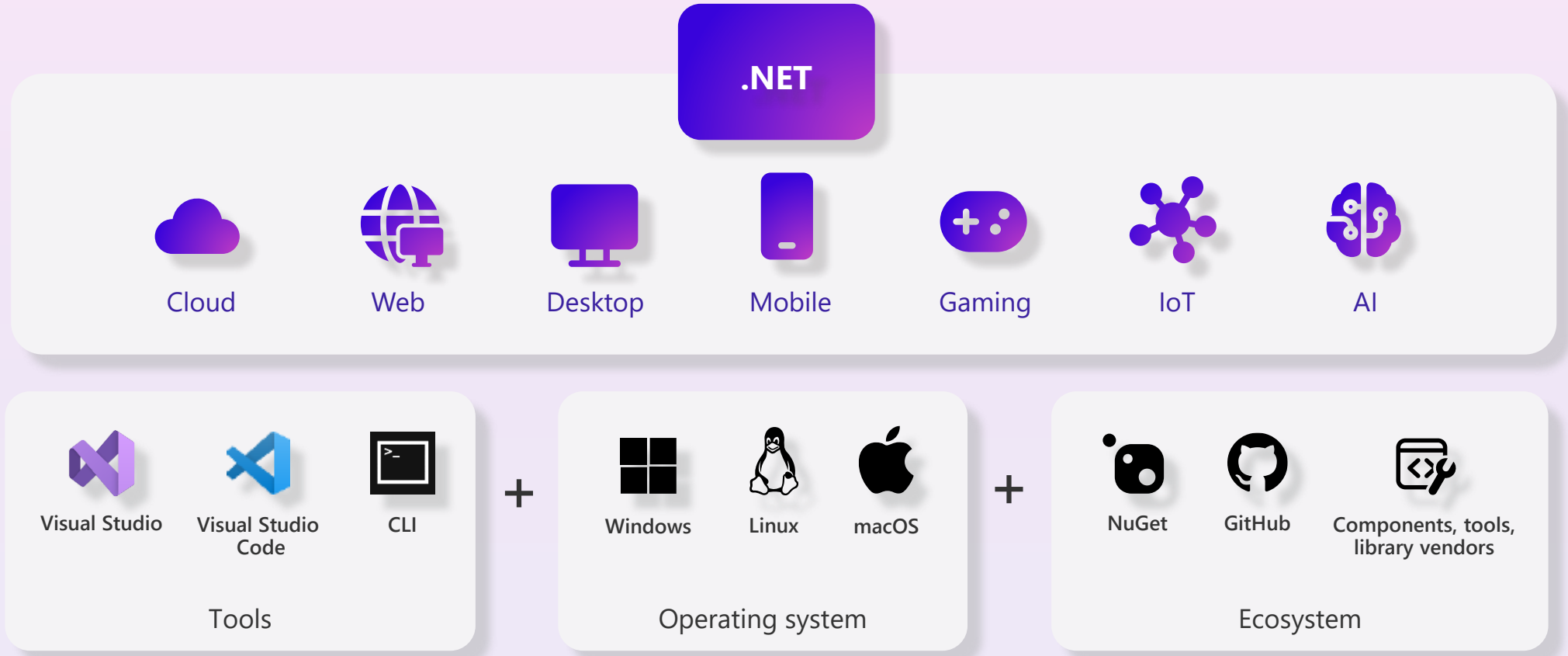
Introduction to

.NET Aspire

James Montemagno
Program Manager, Microsoft
/@JamesMontemagno
/@JamesMontemagno

Build anything with a unified platform



Cloud Native Apps

The CNCF Cloud Native Definition says…

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

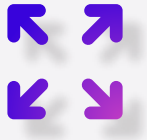
These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

Source: https://github.com/cncf/foundation/blob/main/charter.md

Every App Needs

Observability

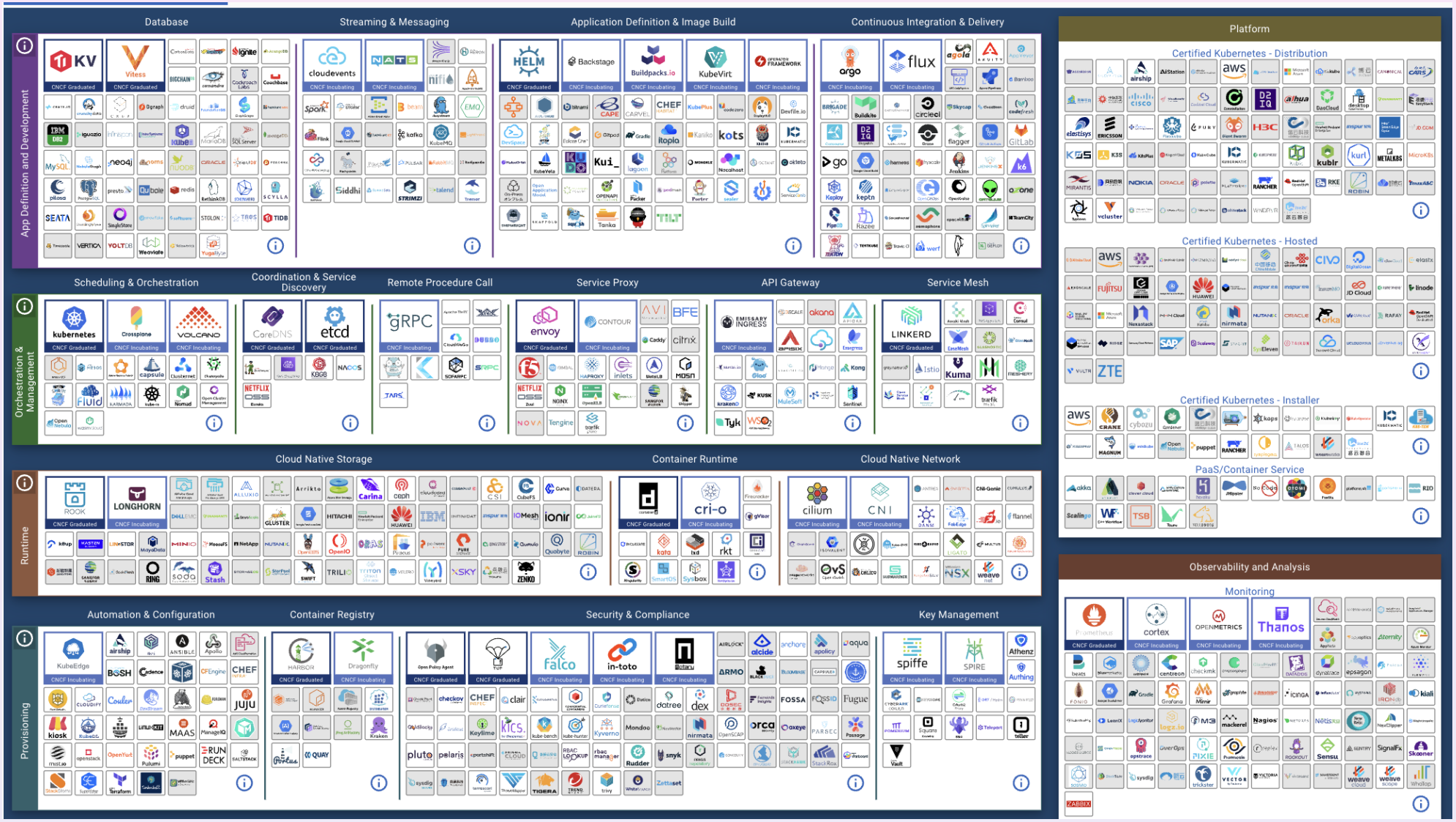
Resiliency



Scalability

Manageability

CNCF Cloud Native Landscape



landscape.cncf.io

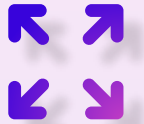
.NET 8 Includes

Observability

- **Built in metrics with dimensions**
- **DI integration for metrics**
- **Better Logging support** (faster, can object serialization)
- **Enrichment**
- **Redaction**
- **Testing fakes for Logging & Metrics**

Resiliency

- **New Polly based resiliency packages**
- **SignalR Stateful Reconnect**



Scalability

- **AOT** (increased density)
- **Performance**
- **Chiseled Ubuntu**

Manageability

- **Certificate auto-rotation support in Kestrel**

It’s Still Not Easy 😟

Complex

Getting Started Is Hard

Too Many Choices

No Paved Path

.NET Aspire

A cloud ready stack for building observable, production ready, distributed applications

.NET Aspire

What does that mean? Who is it for? Should I use it?

.NET Aspire

Works with ALL scale of applications and grows with you!

Northern Mountains

https://www.northernmountains.net

NORTHERN MOUNTAINS

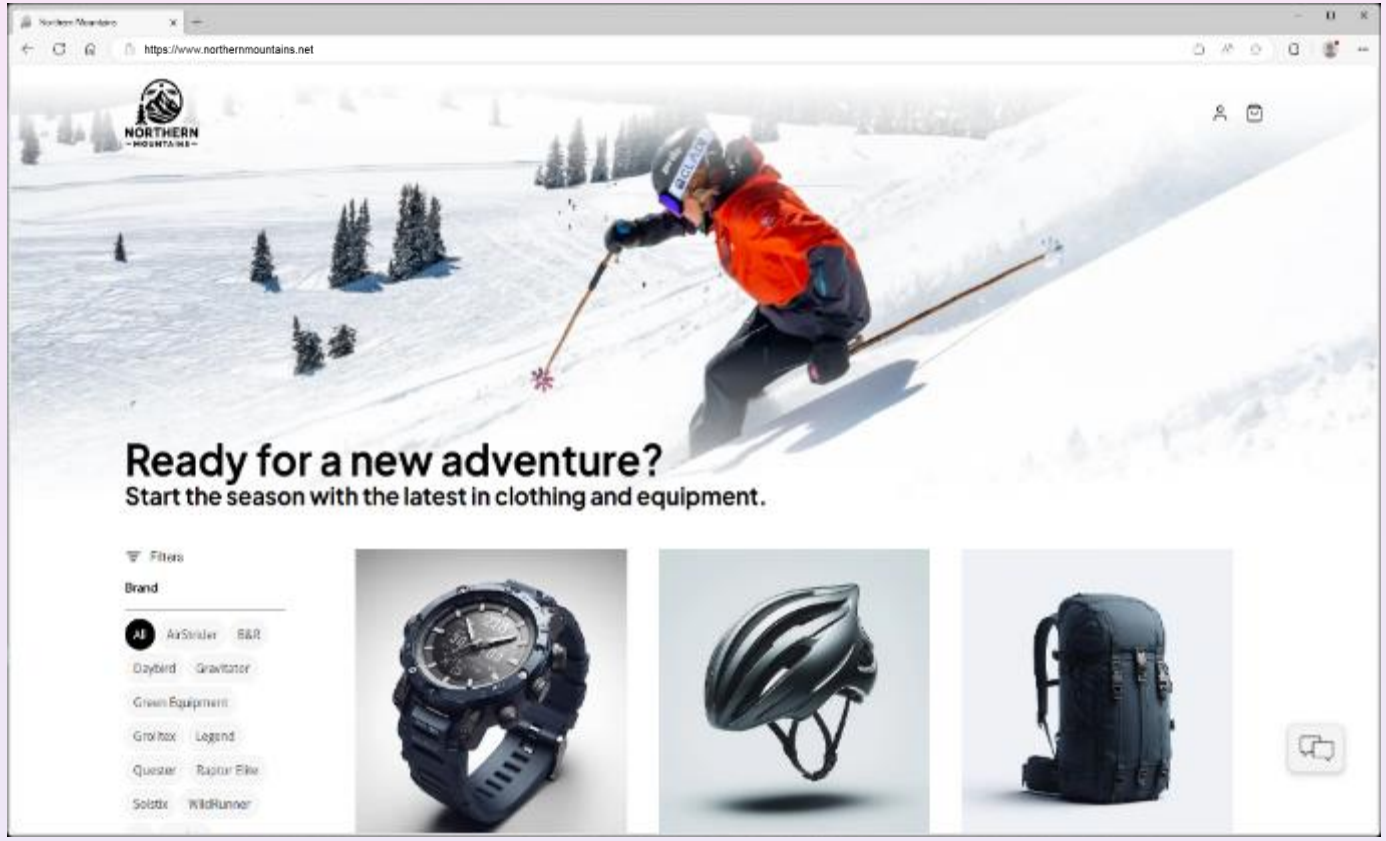
Ready for a new adventure?

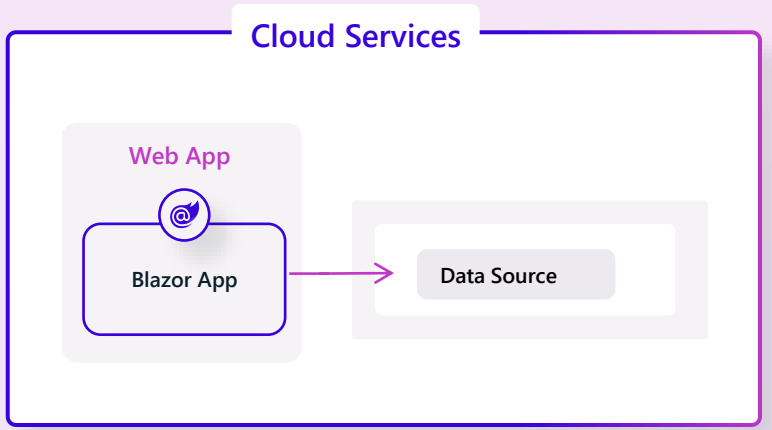
Start the season with the latest in clothing and equipment.

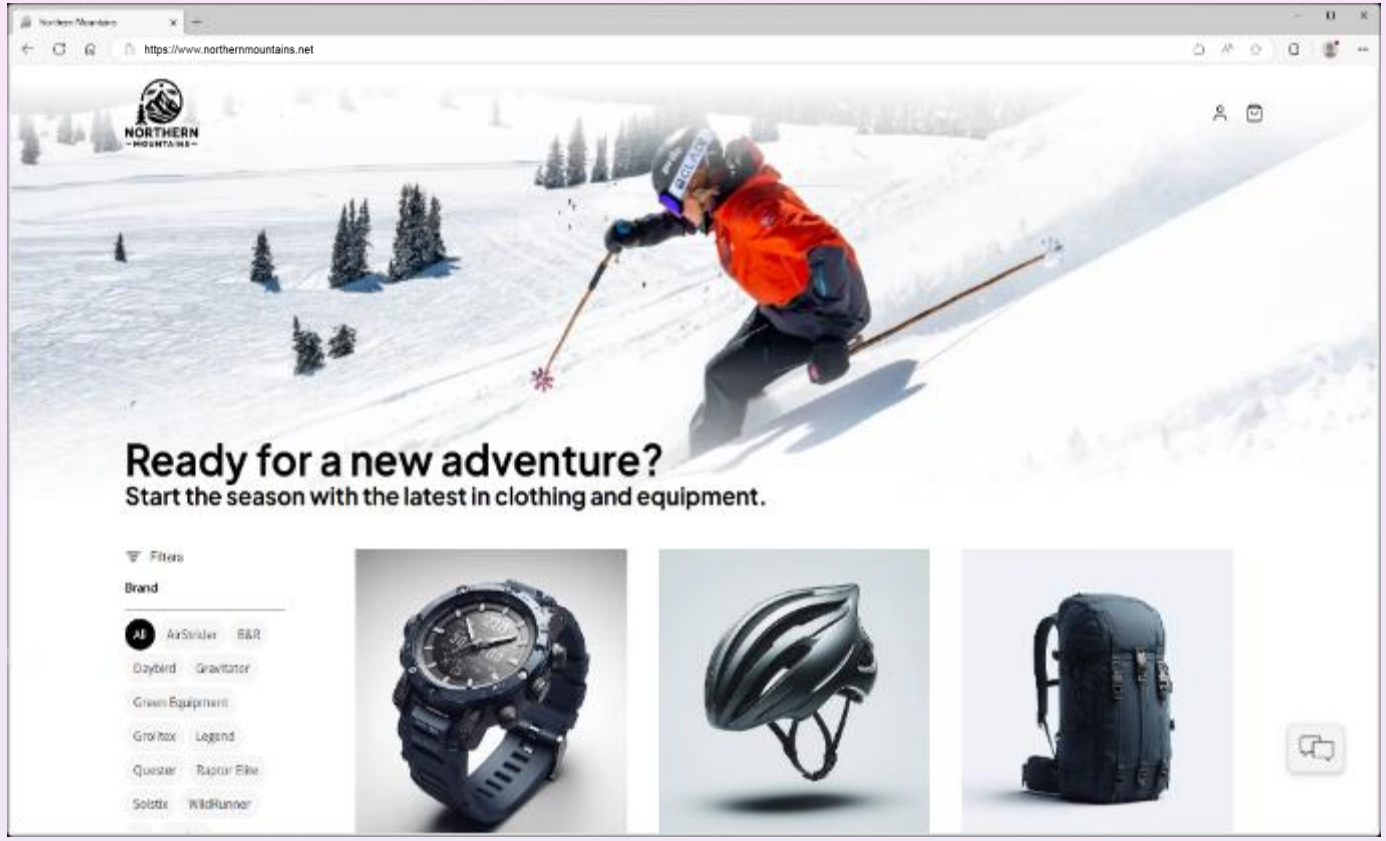
Filters

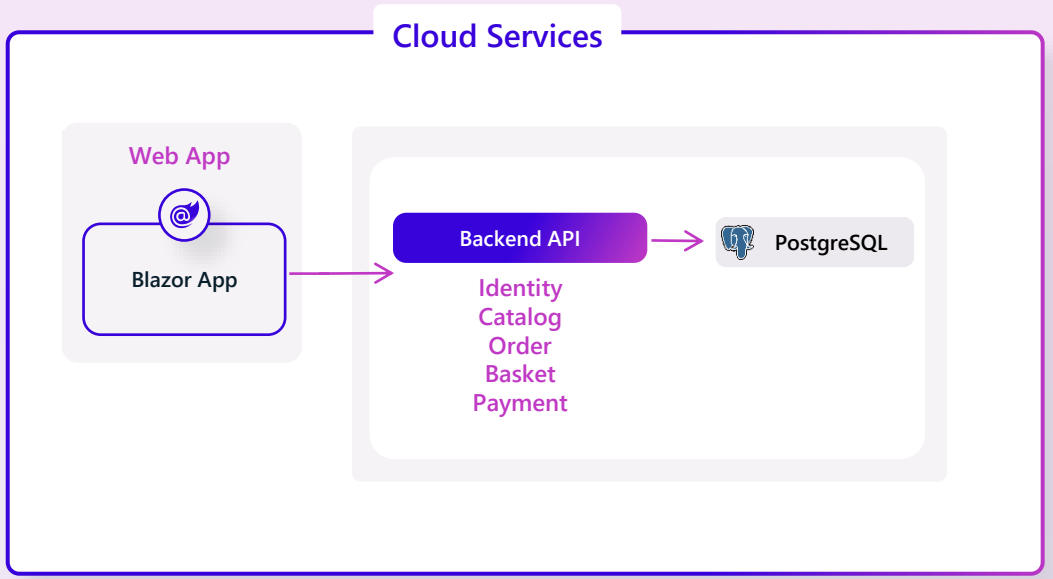
Brand

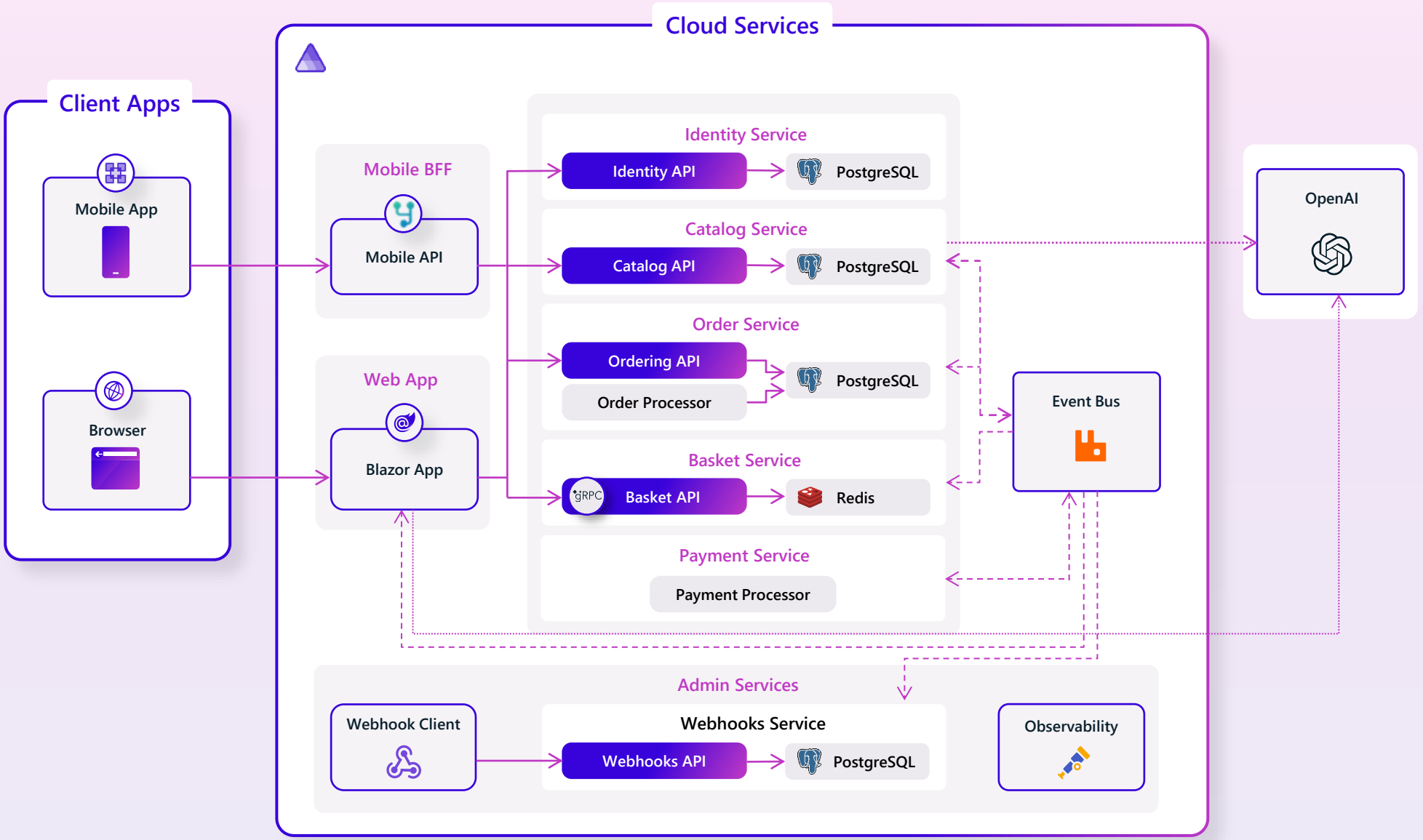
All | AirStrider | B&R | Daybird | Gravitator | Green Equipment | Grolltex | Legend | Quester | Raptor Elite | Solstix | WildRunner


https://www.northernmountains.net
NORTHERN
MOUNTAINS
Ready for a new adventure?
Start the season with the latest in clothing and equipment.
Filters
Brand


Cloud Services
Web App
Blazor App
Data Source


https://www.northernmountains.net
NORTHERN
MOUNTAINS
Ready for a new adventure?
Start the season with the latest in clothing and equipment.
Filters
Brand


Cloud Services
Web App
Blazor App
Backend API
PostgreSQL
Identity
Catalog
Order
Basket
Payment


Cloud Services
Client Apps
Mobile App
Browser
Mobile BFF
Mobile API
Web App
Blazor App
Identity Service
Identity API
PostgreSQL
Catalog Service
Catalog API
PostgreSQL
Order Service
Ordering API
Order Processor
PostgreSQL
Basket Service
gRPC
Basket API
Redis
Payment Service
Payment Processor
OpenAI
Event Bus
Admin Services
Webhook Client
Webhooks Service
Webhooks API
PostgreSQL
Observability

.NET Aspire

Your building blocks for cloud native development

- Smart Defaults
- Developer Dashboard
- Orchestration
- Service Discovery
- Integrations
- Deployment

.NET Aspire

Your building blocks for cloud native development

- Smart Defaults
- Developer Dashboard
- Orchestration
- Service Discovery
- Integrations
- Deployment

DEMO

Adding .NET Aspire to Existing .NET Apps

.NET Aspire

Your building blocks for cloud native development

- Smart Defaults
- Developer Dashboard
- Orchestration
- Service Discovery
- Integrations
- Deployment

.NET Aspire Service Defaults

Observability

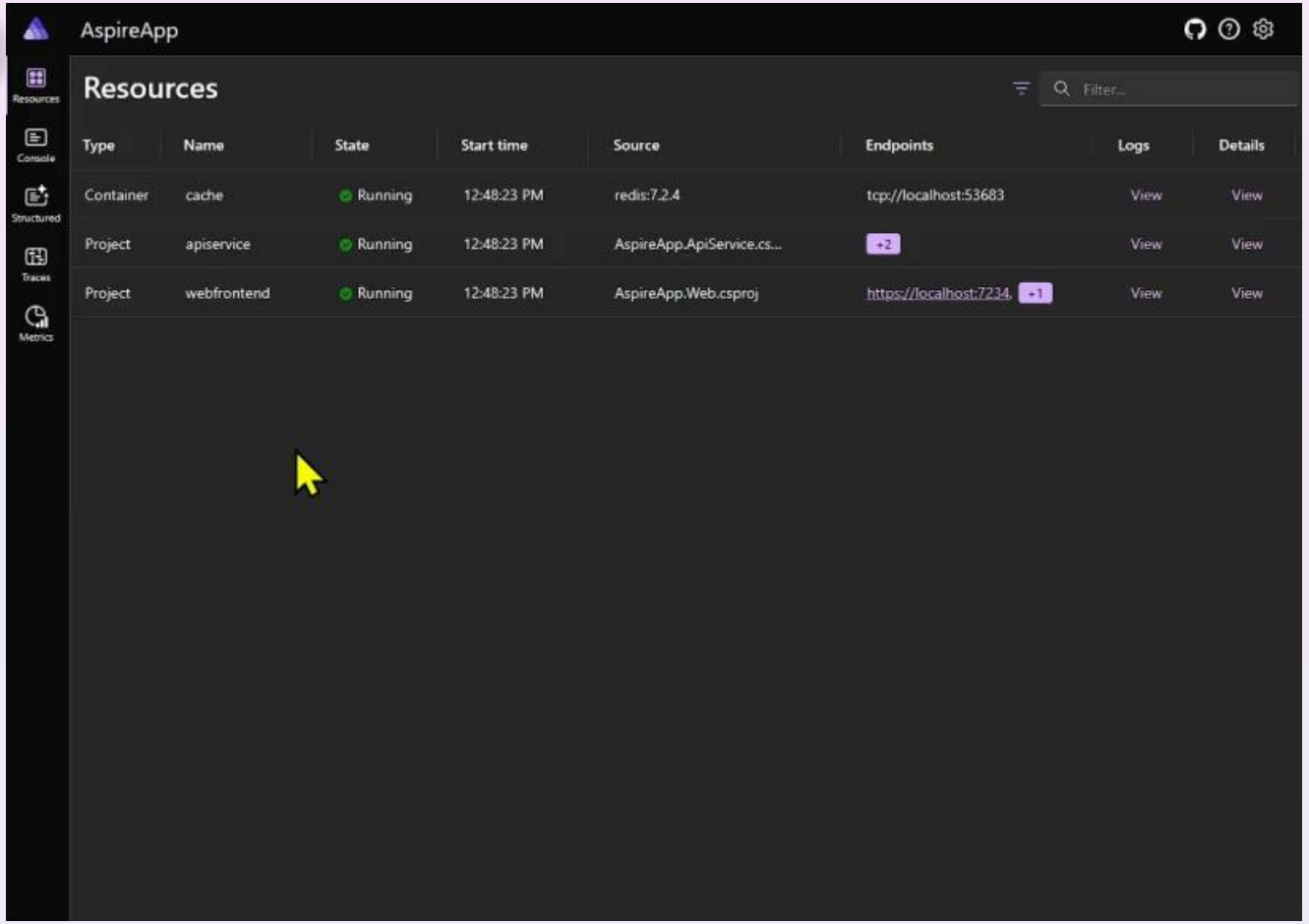
Resiliency

Health Checks

.NET Aspire

Developer Dashboard

- Structured Logs
- Metrics
- Distributed Traces
- Dependencies



.NET Aspire

App Host - Orchestration

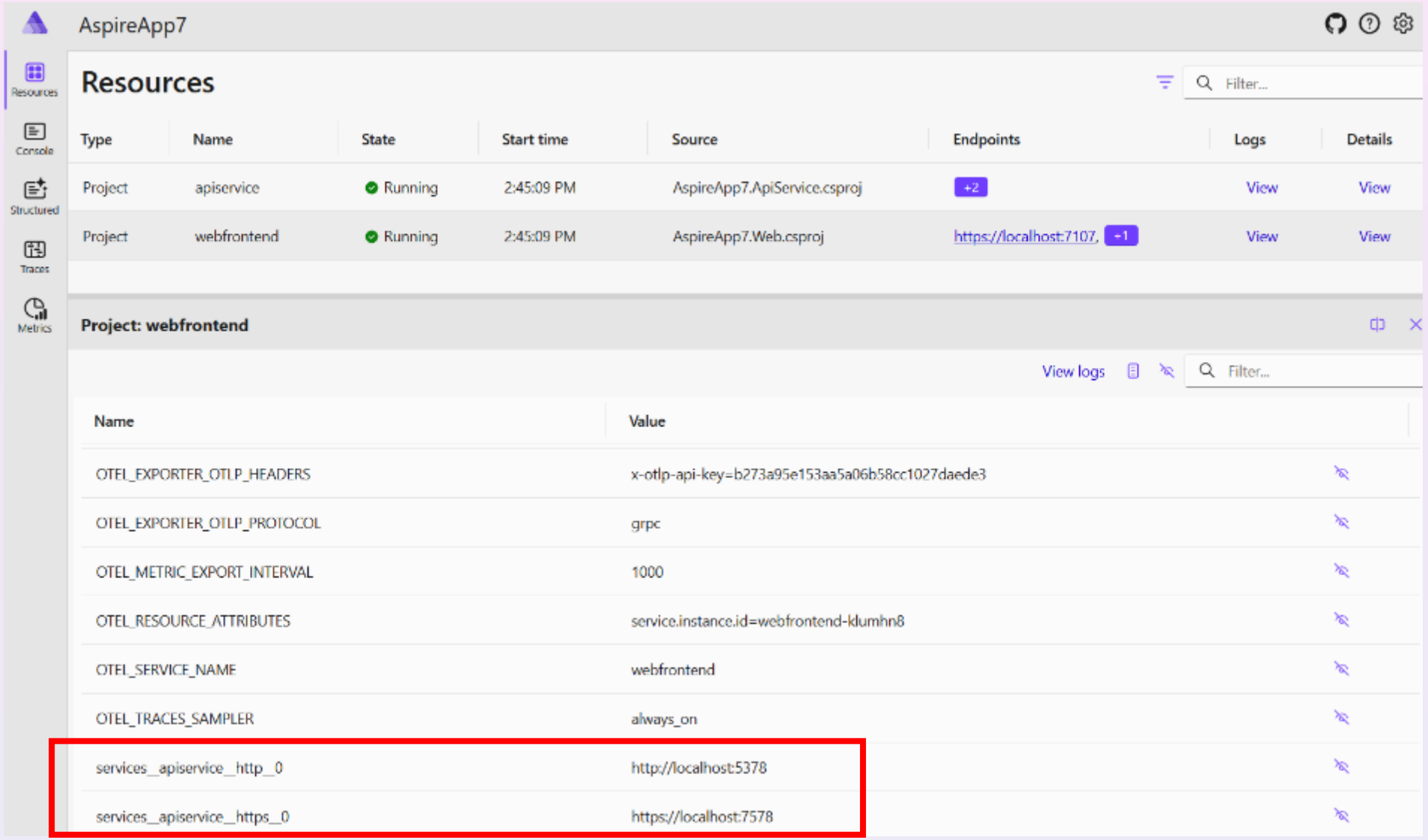
```
var builder = DistributedApplication.CreateBuilder(args);

builder.AddProject<Projects.ApiService>("apiservice");

builder.AddProject<Projects.Web>("webfrontend");

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

Service Discovery

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


var apiService = builder.AddProject<Projects.ApiService>("apiservice");

builder.AddProject<Projects.Web>("webfrontend")
       .WithReference(apiService);

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

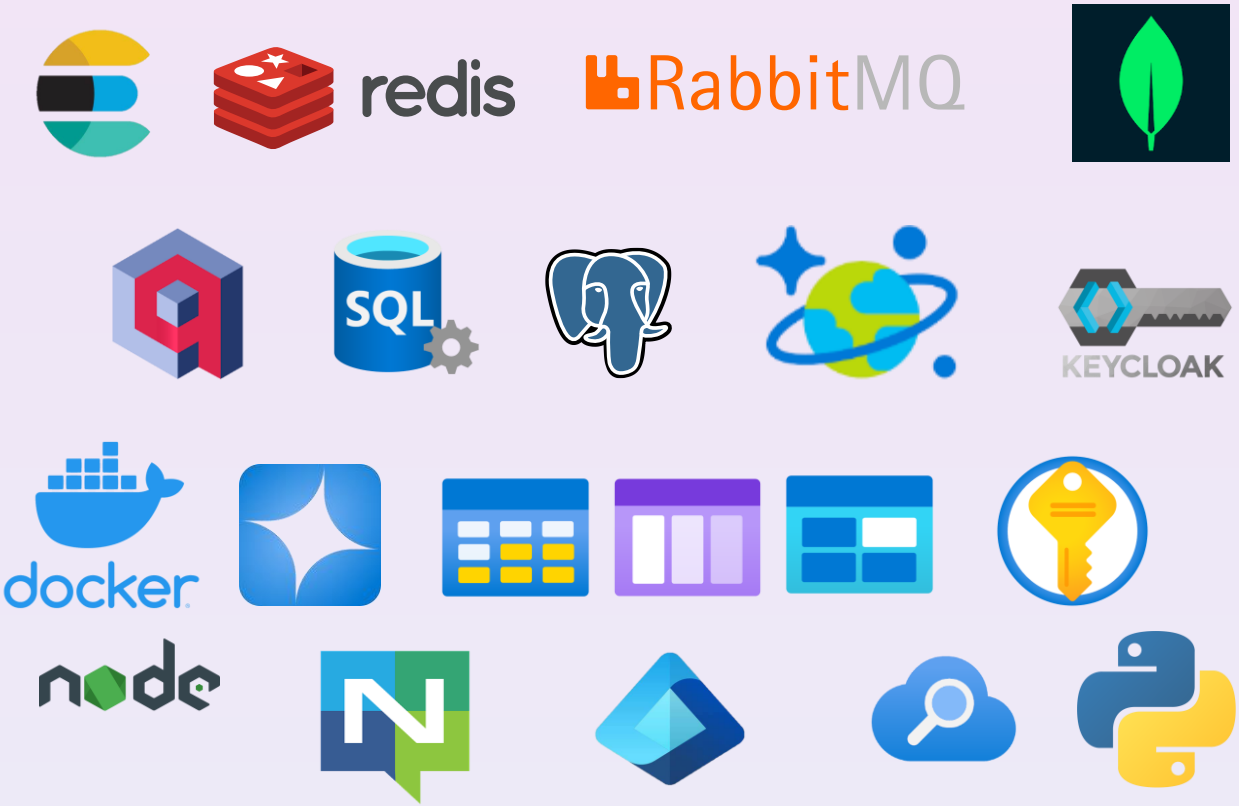
```
builder.Services.AddHttpClient<WeatherApiClient>(client =>
    {
        client.BaseAddress = new("https+http://apiservice");
    });
```

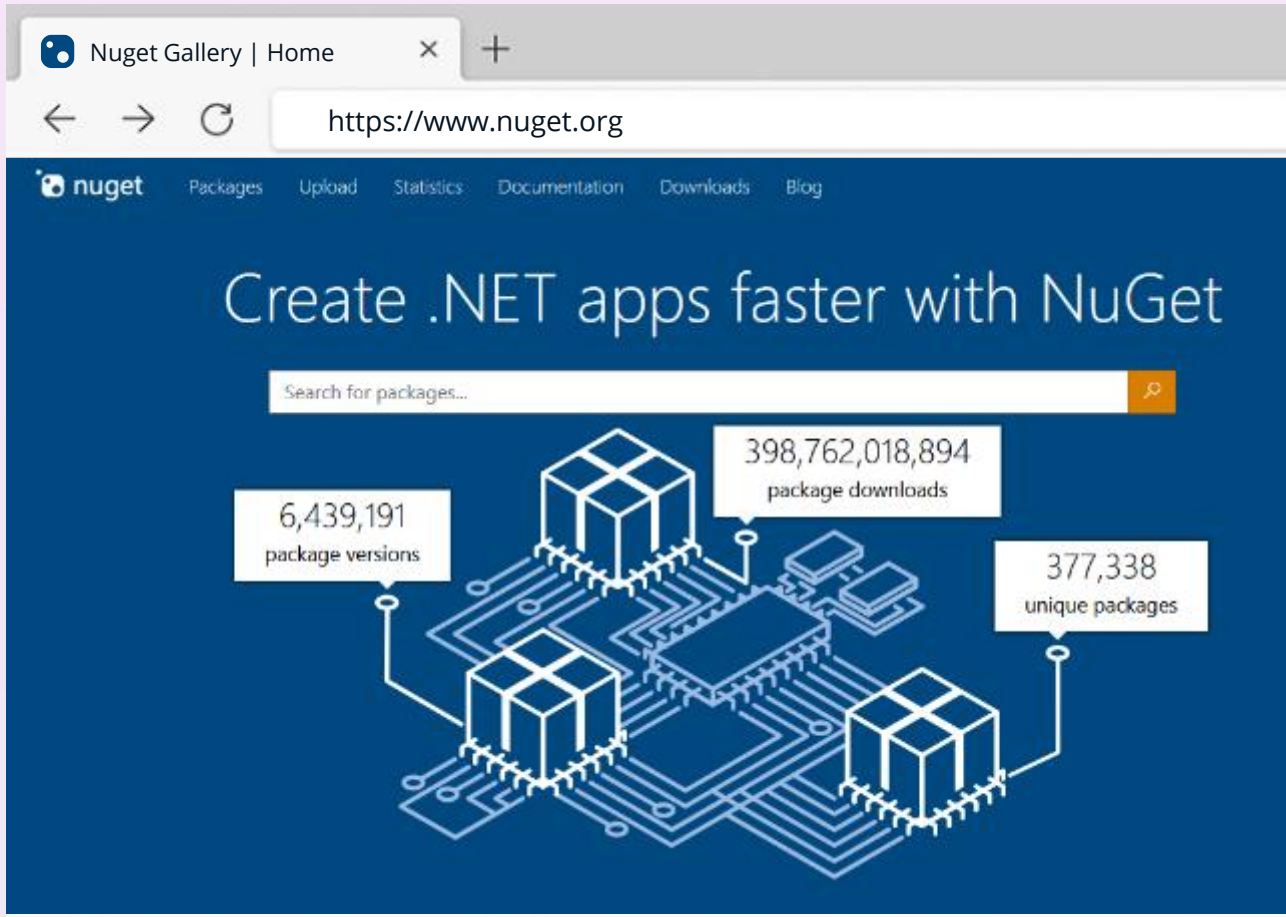
There are still connection strings 😀



.NET Aspire

Integrations





.NET Aspire

Your building blocks for cloud native development

Smart Defaults

Developer Dashboard

Orchestration

Service Discovery

Integrations

Deployment

Deploying Distributed Applications

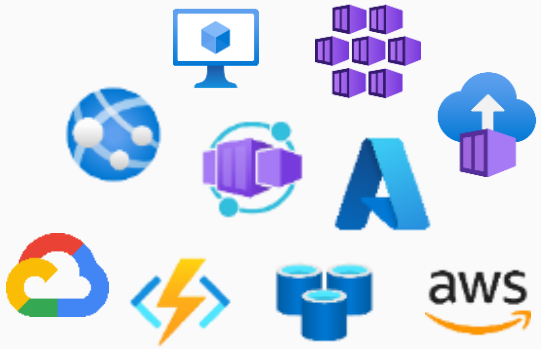
Every Apps Is Different

Tons of Choices

IT & Security Requirements

Complex

Deployment Options with .NET Aspire



How You Do It Today!

AWS CDK

Azure Dev CLI

Visual Studio

This is ALL Optional!

DEMO

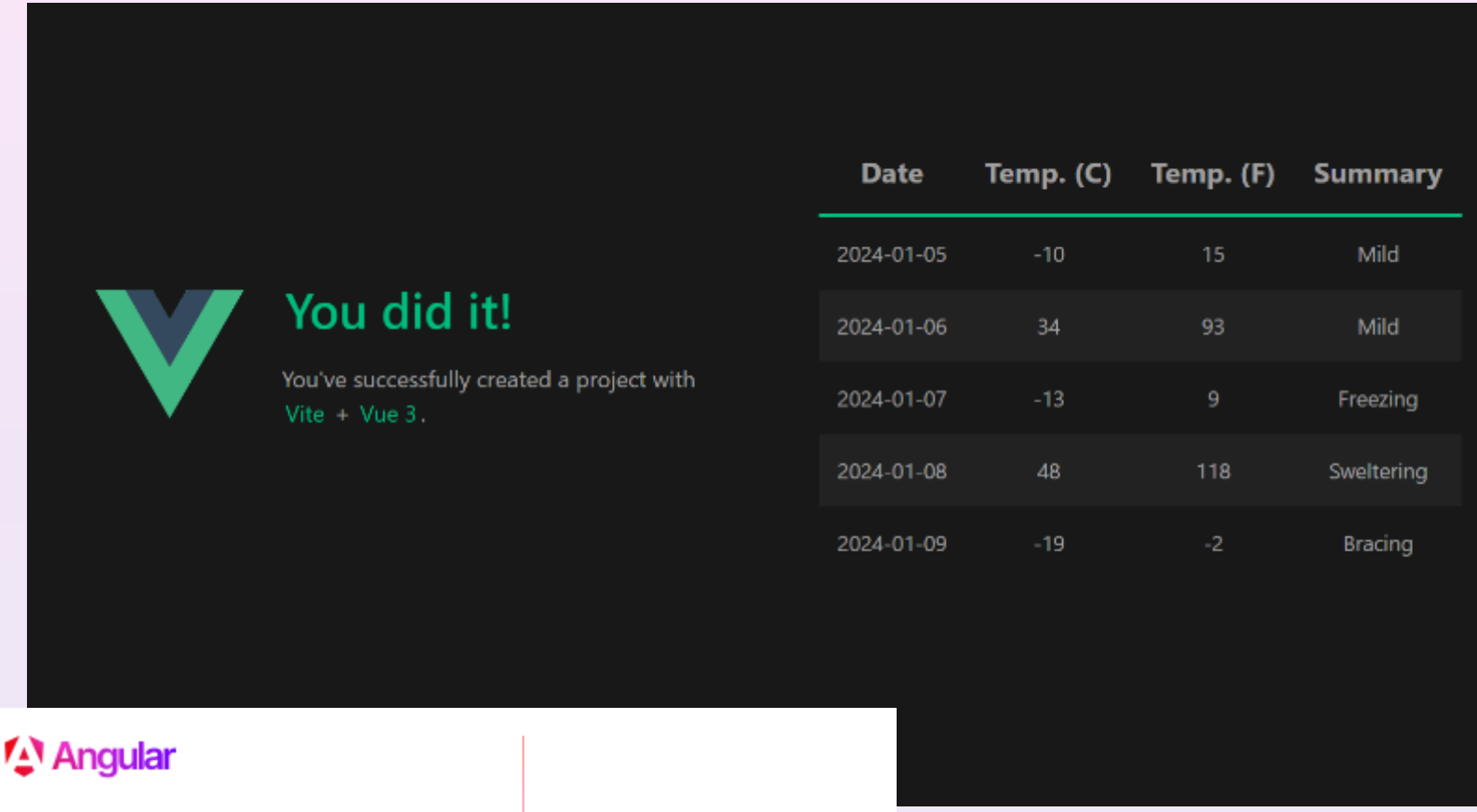
Let’s deploy this thing!

.NET Aspire

Going Beyond .NET!

Add to any app

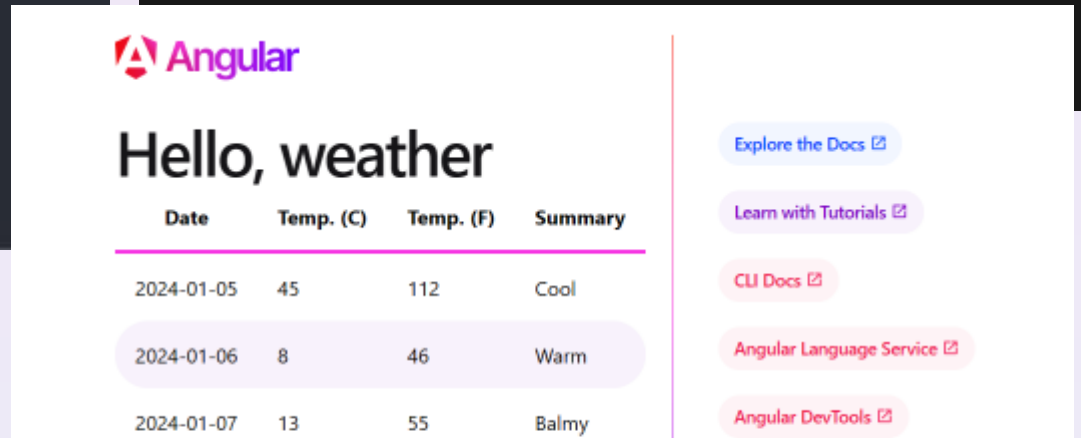
Date	Temp. (C)	Temp. (F)	Summary
2024-01-05	-8	18	Mild
2024-01-06	33	91	Cool
2024-01-07	-12	11	Freezing
2024-01-08	33	91	Cool
2024-01-09	5	40	Bracing



You did it!

You've successfully created a project with Vite + Vue 3.

Date	Temp. (C)	Temp. (F)	Summary
2024-01-05	-10	15	Mild
2024-01-06	34	93	Mild
2024-01-07	-13	9	Freezing
2024-01-08	48	118	Sweltering
2024-01-09	-19	-2	Bracing



Angular

Hello, weather

Date	Temp. (C)	Temp. (F)	Summary
2024-01-05	45	112	Cool
2024-01-06	8	46	Warm
2024-01-07	13	55	Balmy

Explore the Docs

Learn with Tutorials

CLI Docs

Angular Language Service

Angular DevTools

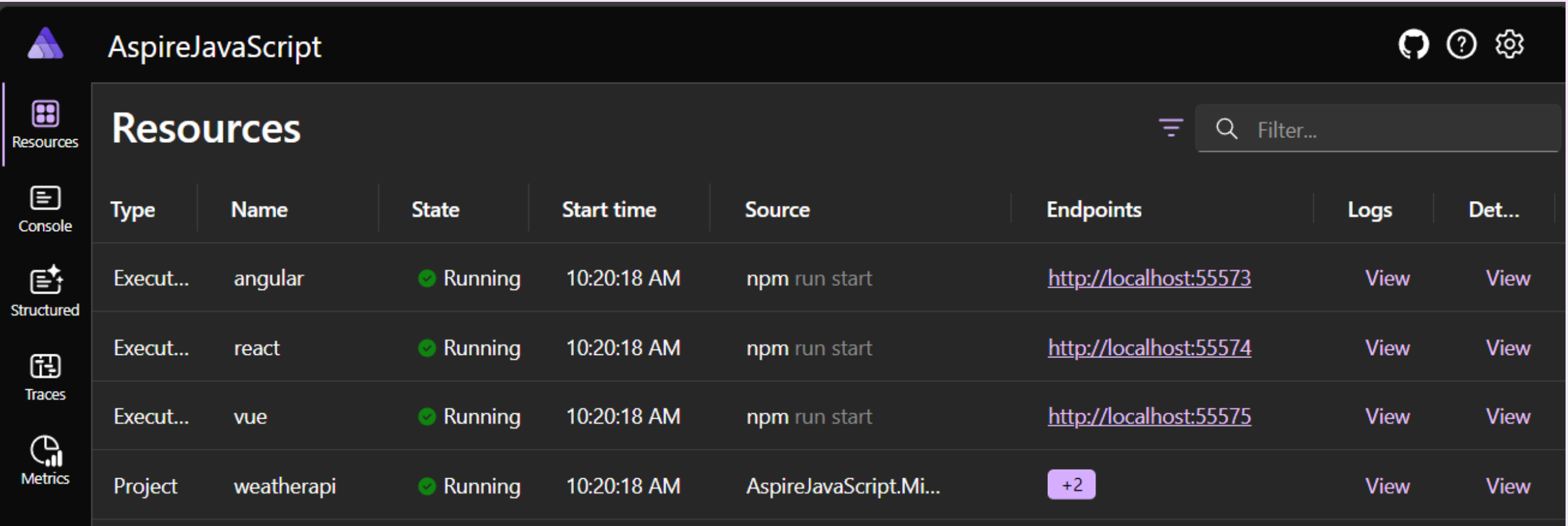
.NET Aspire for Everyone!

AddPythonProject

AddNpmApp

AddNodeApp

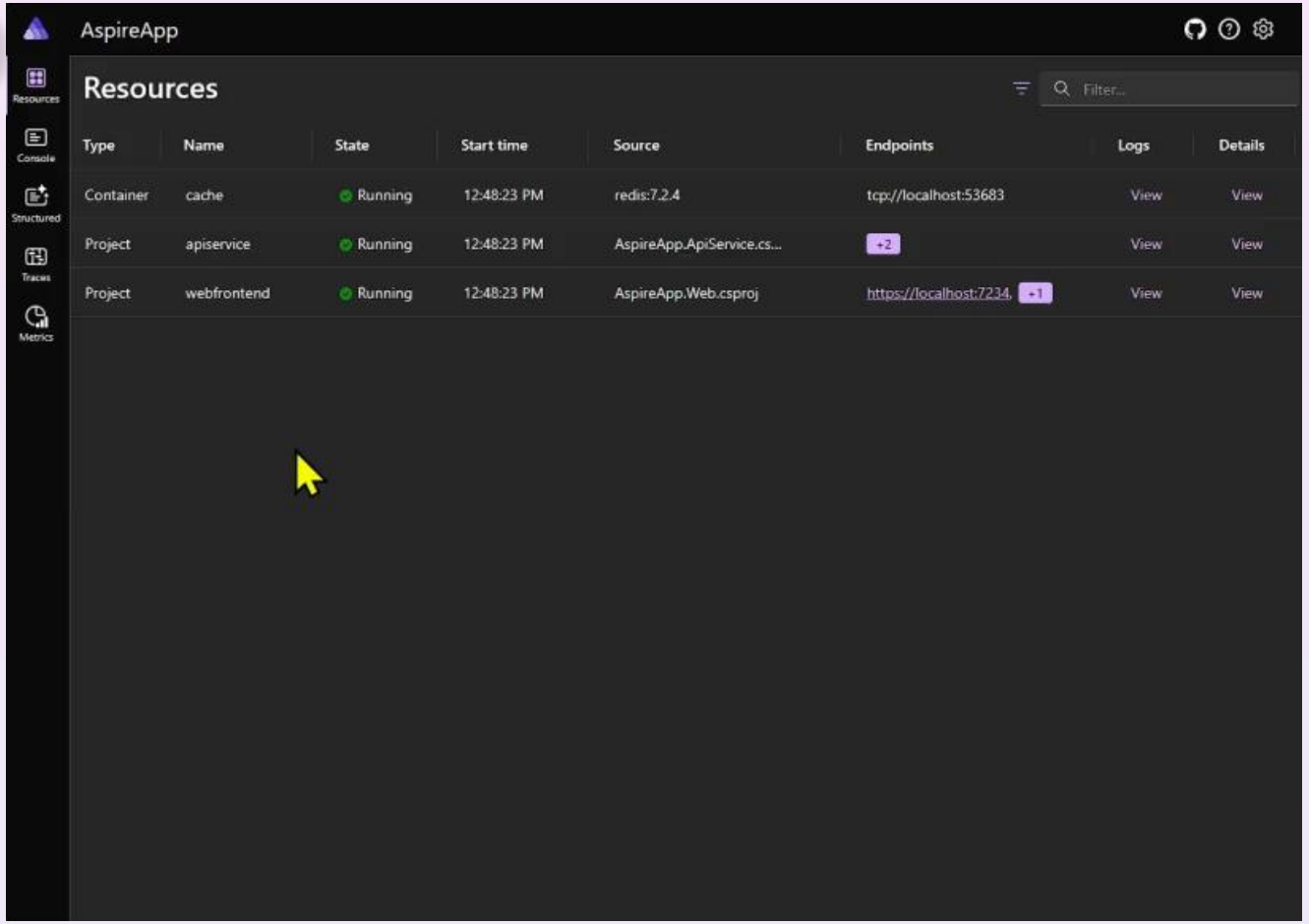
```
builder.AddNpmApp("angular",
                  "../AspireJavaScript.Angular")
    .WithReference(apiService)
    .WithHttpEndpoint(env: "PORT")
    .WithExternalHttpEndpoints()
    .PublishAsDockerFile();
```



.NET Aspire

Developer Dashboard

- Structured Logs
- Metrics
- Distributed Traces
- Dependencies

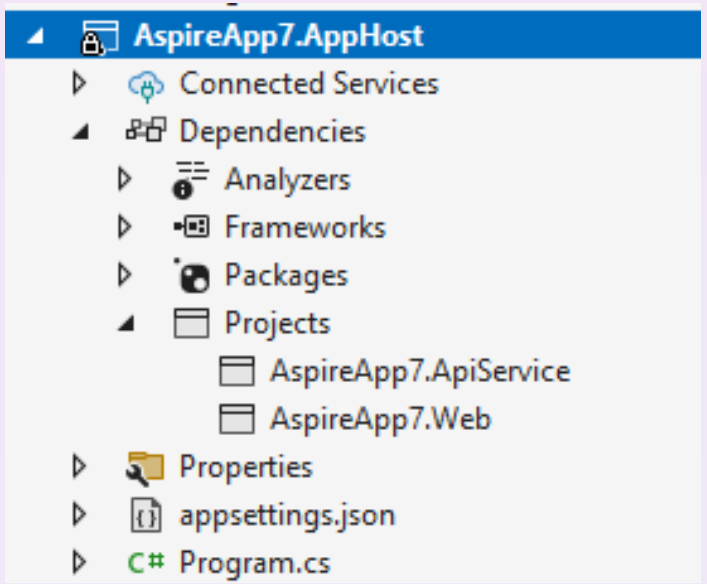


Integrating the Developer Dashboard

Standalone

```
docker run --rm -it `
    -p 18888:18888 `
    -p 4317:18889 -d `
    --name aspire-dashboard `
    mcr.microsoft.com/dotnet/aspire-dashboard:8.0.0
```

.NET Aspire App Host



DEMO

Standalone dashboard

.NET Aspire

A cloud ready stack for building observable, production ready, distributed applications

- Smart Defaults
- Developer Dashboard
- Orchestration
- Service Discovery
- Components
- Deployment

.NET Aspire

Learning Resources

aka.ms/letslearn/dotnet/aspire

.NET Aspire Videos

aka.ms/aspire/videos

Documentation

aka.ms/dotnet-aspire

Engage with team on GitHub

github.com/dotnet/aspire

James Montemagno
Program Manager, Microsoft

/@JamesMontemagno
/@JamesMontemagno