

Introduction to



.NET Aspire

**James Montemagno**

Program Manager, Microsoft

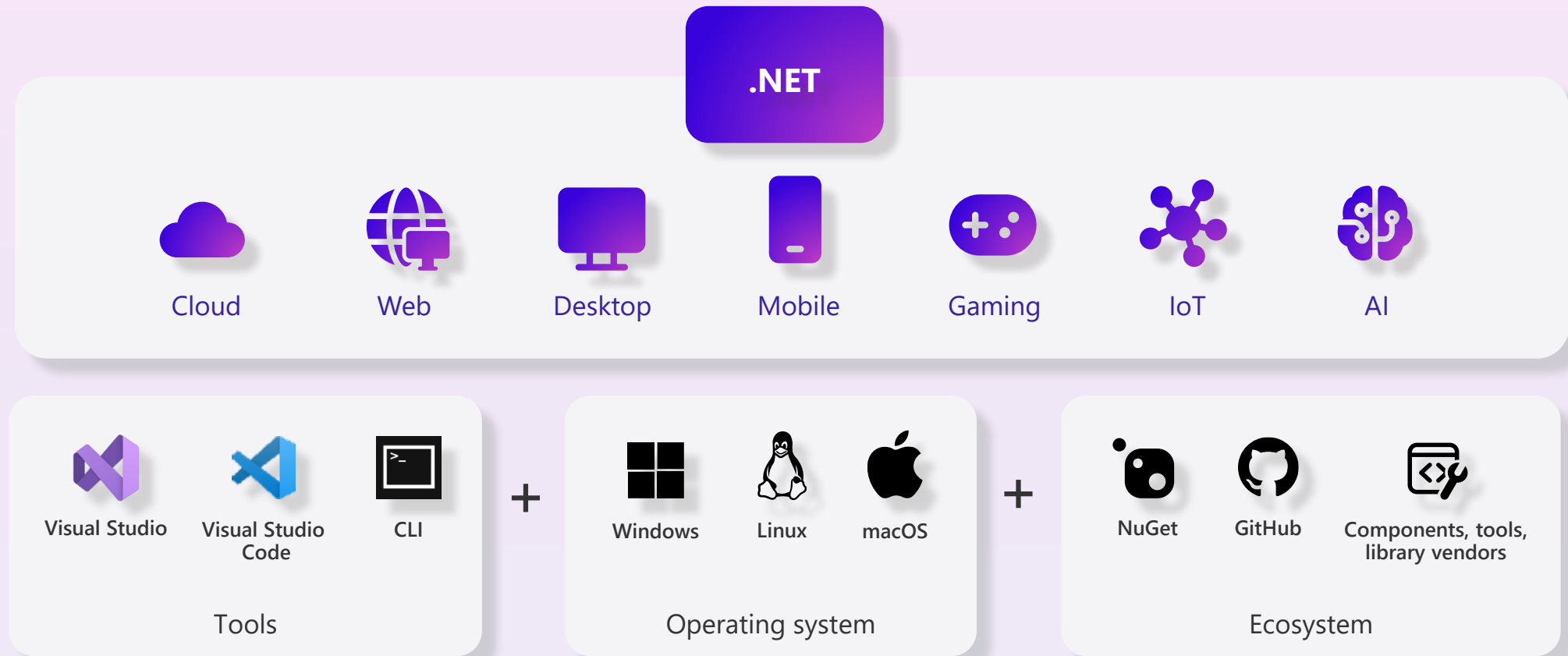


/@JamesMontemagno



/@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

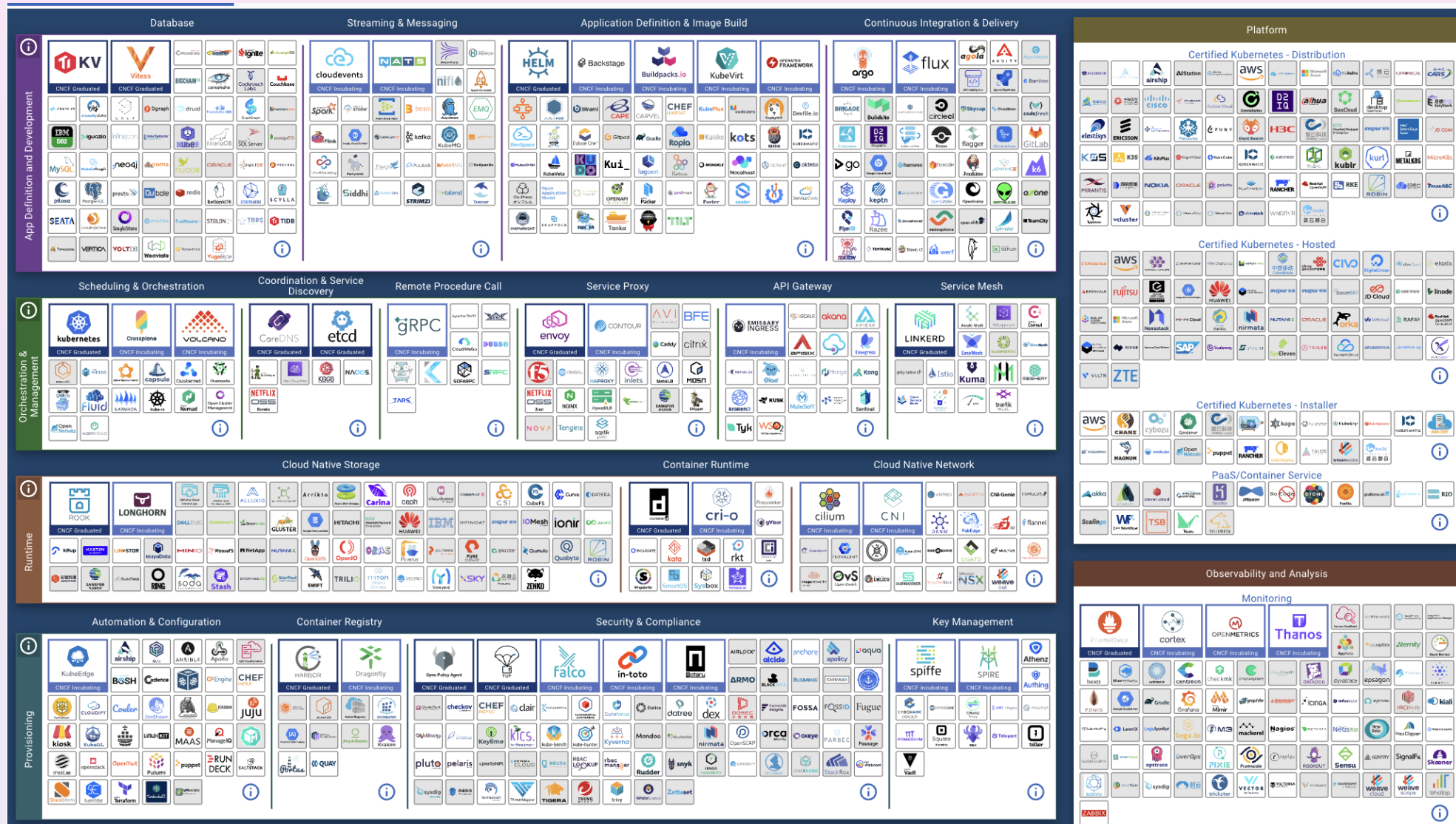


Scalability



Manageability

# CNCF Cloud Native Landscape



# .NET 8 Includes



## Observability

**Built in metrics with dimensions**

**DI integration for metrics**

**Better Logging support**  
(faster, can object serialization)

**Enrichment**

**Redaction**

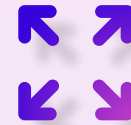
**Testing fakes for Logging & Metrics**



## 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!**



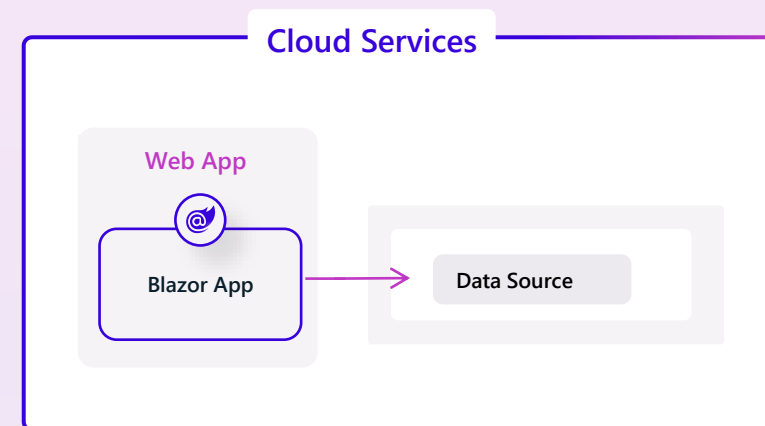
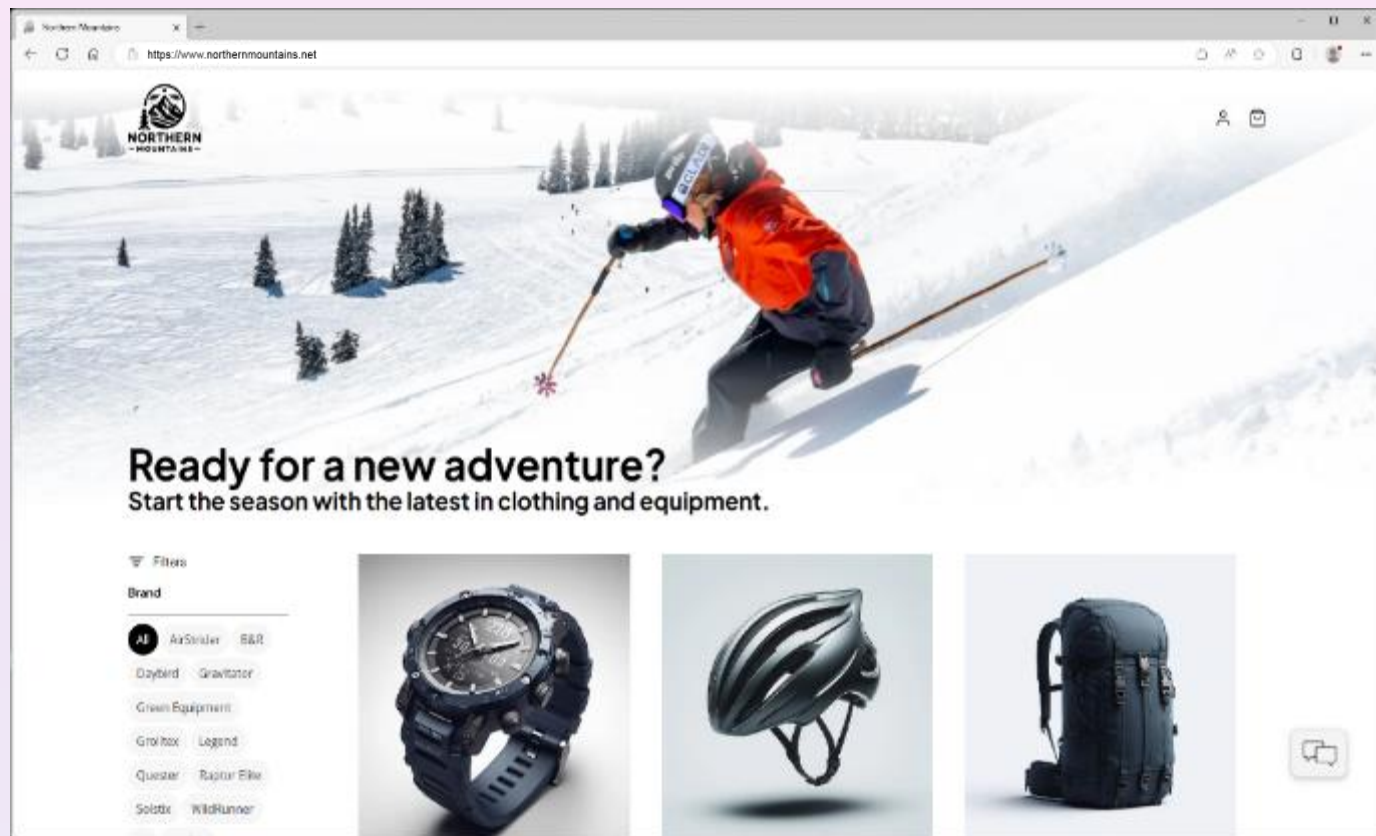
**Ready for a new adventure?**  
Start the season with the latest in clothing and equipment.

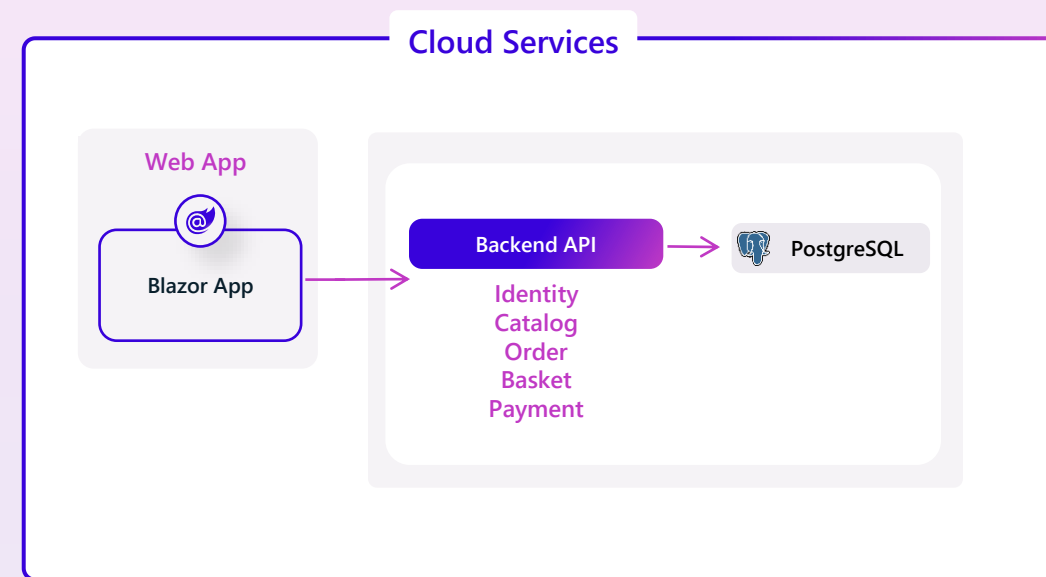
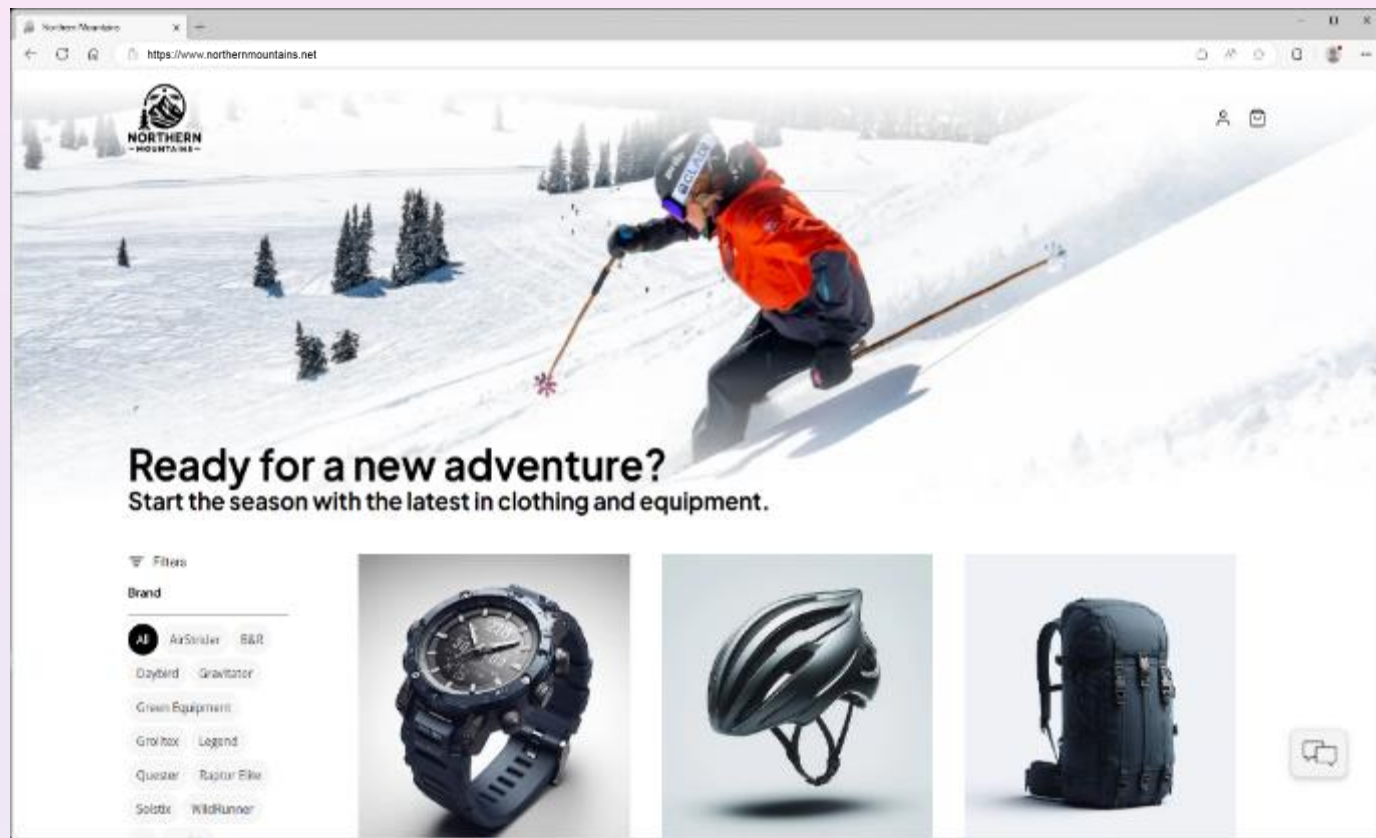
☰ Filters

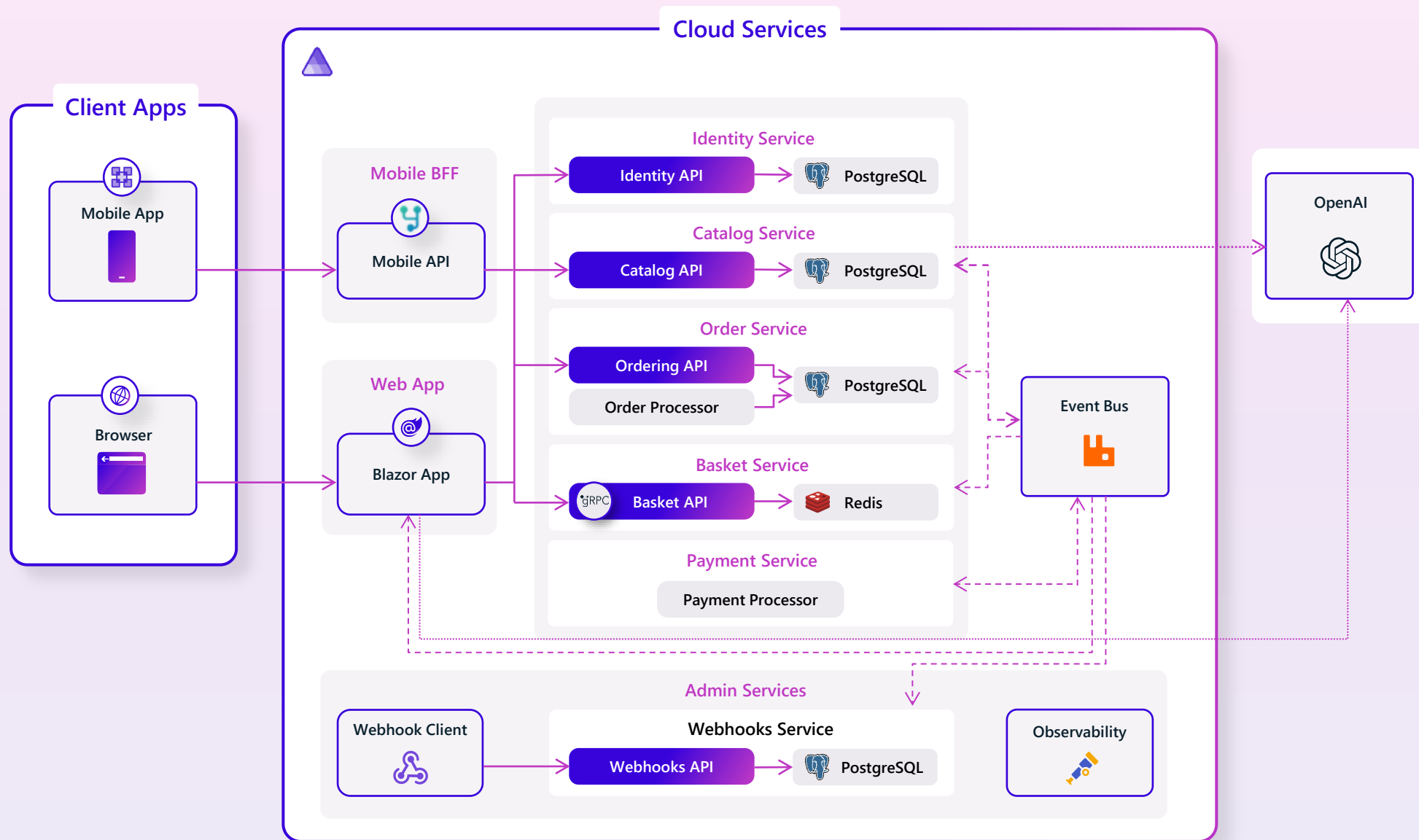
Brand

- All
- AirStrider
- B&R
- Daybird
- Gravitator
- Green Equipment
- Grolitex
- Legend
- Quester
- Raptor Elite
- Solstix
- WildRunner











# .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

The screenshot shows the .NET Aspire Developer Dashboard interface. The title bar reads 'AspireApp'. On the left is a sidebar with icons for Resources, Console, Structured, Traces, and Metrics. The main area is titled 'Resources' and contains a table with the following data:

Type	Name	State	Start time	Source	Endpoints	Logs	Details
Container	cache	Running	12:48:23 PM	redis:7.2.4	tcp://localhost:53683	View	View
Project	apiservice	Running	12:48:23 PM	AspireApp.ApiService.cs...	+2	View	View
Project	webfrontend	Running	12:48:23 PM	AspireApp.Web.csproj	https://localhost:7234 +1	View	View



# .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);

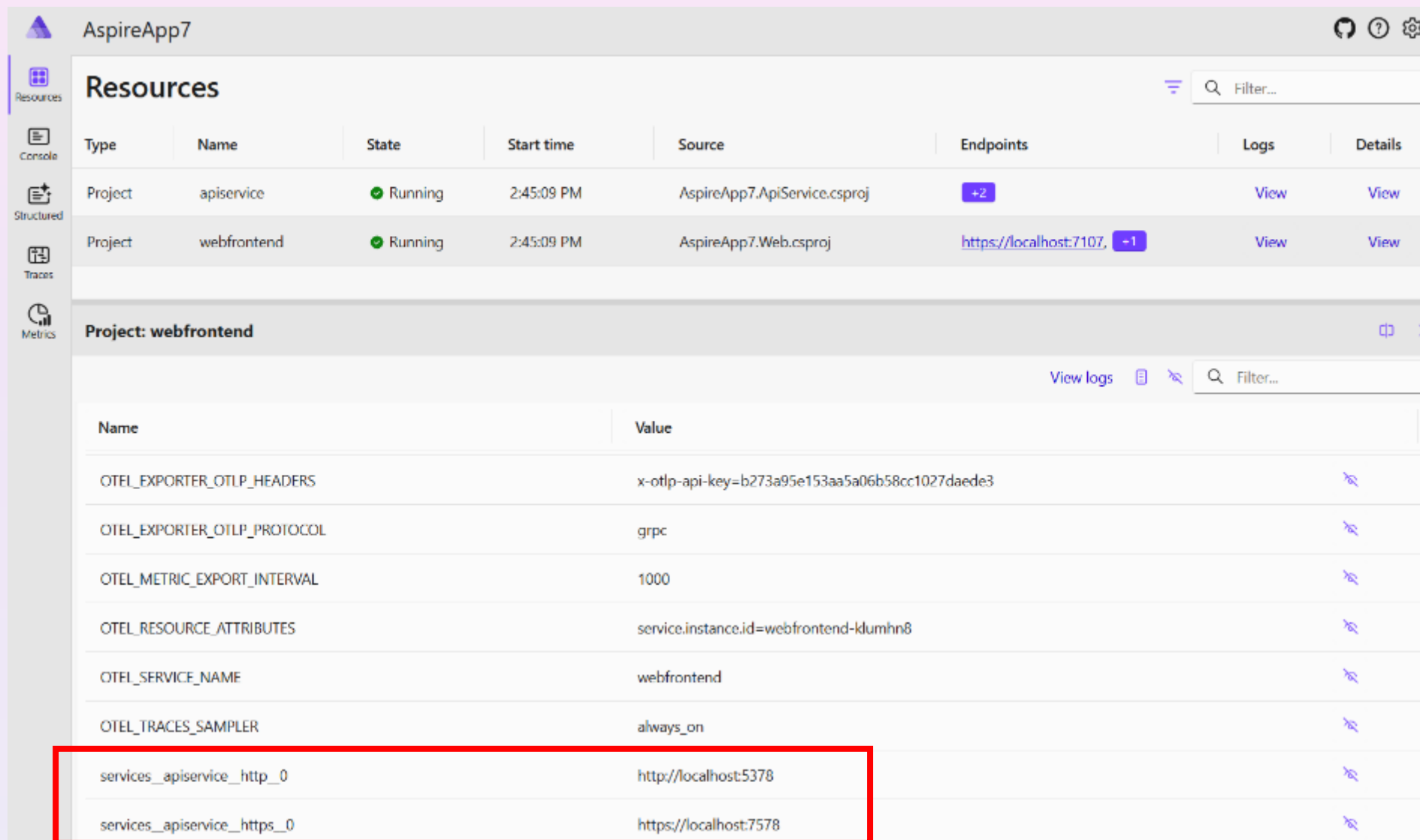
var apiService = builder.AddProject<Projects.ApiService>("apiservice");

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 🤪



The screenshot displays the AspireApp7 dashboard. The 'Resources' section lists two projects: 'apiservice' and 'webfrontend', both in a 'Running' state. The 'webfrontend' project is selected, showing its environment variables. A red box highlights the 'services\_\_apiservice\_\_http\_0' and 'services\_\_apiservice\_\_https\_0' variables, which are connection strings for the API service.

Type	Name	State	Start time	Source	Endpoints	Logs	Details
Project	apiservice	Running	2:45:09 PM	AspireApp7.ApiService.csproj	+2	View	View
Project	webfrontend	Running	2:45:09 PM	AspireApp7.Web.csproj	<a href="https://localhost:7107/">https://localhost:7107/</a> -1	View	View

Name	Value
OTEL_EXPORTER_OTLP_HEADERS	x-otlp-api-key=b273a95e153aa5a06b58cc1027daede3
OTEL_EXPORTER_OTLP_PROTOCOL	grpc
OTEL_METRIC_EXPORT_INTERVAL	1000
OTEL_RESOURCE_ATTRIBUTES	service.instance.id=webfrontend-klumhn8
OTEL_SERVICE_NAME	webfrontend
OTEL_TRACES_SAMPLER	always_on
services__apiservice__http_0	http://localhost:5378
services__apiservice__https_0	https://localhost:7578



# .NET Aspire

## Integrations



redis



RabbitMQ



KEYCLOAK



docker



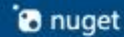
node



NuGet Gallery | Home



https://www.nuget.org



Packages

Upload

Statistics

Documentation

Downloads

Blog

## Create .NET apps faster with NuGet

Search for packages...



6,439,191  
package versions

398,762,018,894  
package downloads

377,338  
unique packages





# .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

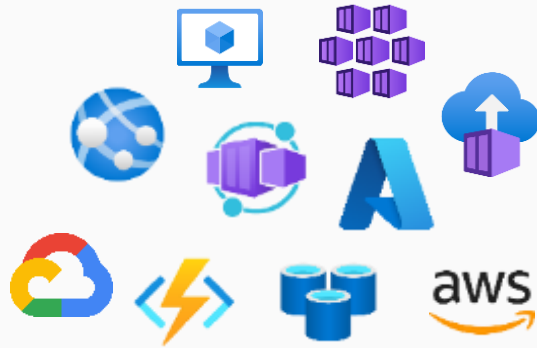


IT & Security  
Requirements



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



## 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();
```


 AspireJavaScript




 Resources

 Console


 Structured





 Traces

 Metrics

# Resources



Type	Name	State	Start time	Source	Endpoints	Logs	Det...
Execut...	angular	 Running	10:20:18 AM	npm run start	<a href="http://localhost:55573">http://localhost:55573</a>	<a href="#">View</a>	<a href="#">View</a>
Execut...	react	 Running	10:20:18 AM	npm run start	<a href="http://localhost:55574">http://localhost:55574</a>	<a href="#">View</a>	<a href="#">View</a>
Execut...	vue	 Running	10:20:18 AM	npm run start	<a href="http://localhost:55575">http://localhost:55575</a>	<a href="#">View</a>	<a href="#">View</a>
Project	weatherapi	 Running	10:20:18 AM	AspireJavaScript.Mi...	+2	<a href="#">View</a>	<a href="#">View</a>



# .NET Aspire

## Developer Dashboard



Structured Logs



Metrics



Distributed Traces



Dependencies

The screenshot shows the .NET Aspire Developer Dashboard interface. The title bar reads 'AspireApp'. On the left is a sidebar with icons for Resources, Console, Structured, Traces, and Metrics. The main area is titled 'Resources' and contains a table with the following data:

Type	Name	State	Start time	Source	Endpoints	Logs	Details
Container	cache	Running	12:48:23 PM	redis:7.2.4	tcp://localhost:53683	View	View
Project	apiservice	Running	12:48:23 PM	AspireApp.ApiService.cs...	+2	View	View
Project	webfrontend	Running	12:48:23 PM	AspireApp.Web.csproj	https://localhost:7234 +1	View	View

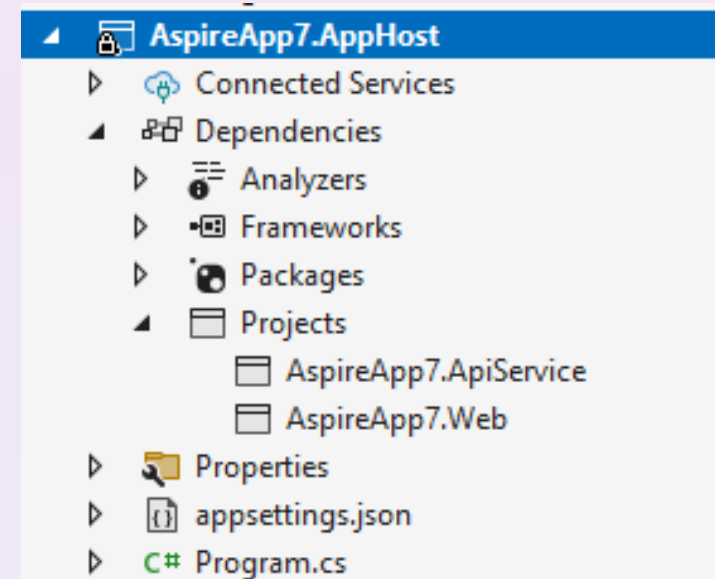


# Integrating the Developer Dashboard

## 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](https://aka.ms/letslearn/dotnet/aspire)

**.NET Aspire Videos**

[aka.ms/aspire/videos](https://aka.ms/aspire/videos)

**Documentation**

[aka.ms/dotnet-aspire](https://aka.ms/dotnet-aspire)

**Engage with team on GitHub**

[github.com/dotnet/aspire](https://github.com/dotnet/aspire)

**James Montemagno**  
Program Manager, Microsoft



/@JamesMontemagno

/@JamesMontemagno