



.net aspire

by Sinan Nar



who am I

consultant

have 8+ yrs exp

have 3 rabbits

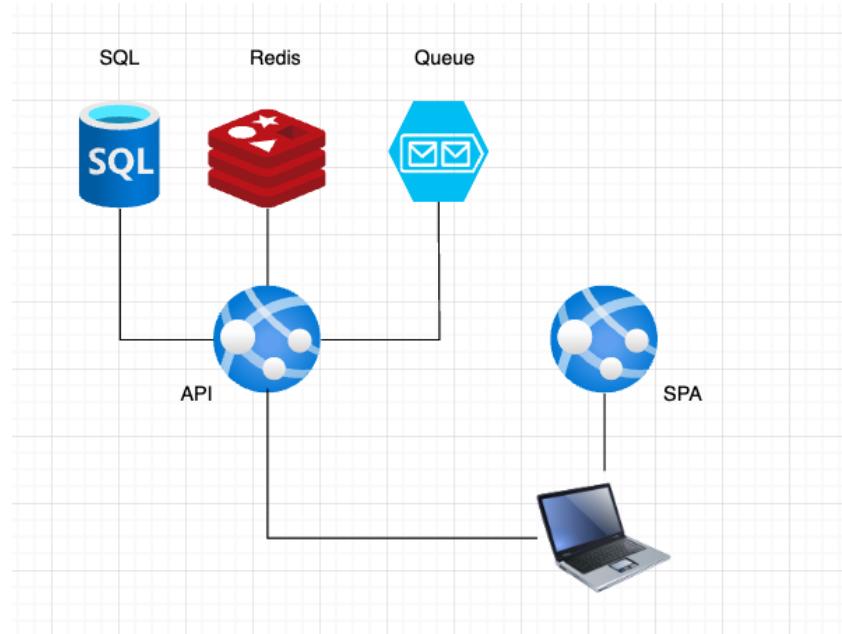


	Microsoft Certified: AI Fundamentals — June 2025
	GitHub Copilot — May 2025
	MongoDB Associate Developer — Nov 2024
	Microsoft Certified: Azure AI Engineer Associate — Jun 2024
	Microsoft Certified: Azure Solution Architect Expert — Mar 2023
	Microsoft Certified: DevOps Engineer Expert — Dec 2022
	Microsoft Certified: Azure Developer Associate — Mar 2022
	Microsoft Certified: Azure Administrator Associate — Feb 2022
	Microsoft Certified: Azure Fundamentals — Jul 2020
	Microsoft Certified Solutions Developer: App Builder — Aug 2019
	Microsoft Certified Solutions Associate: Web Applications — Mar 2019
	Microsoft Certified Professional — Sep 2018



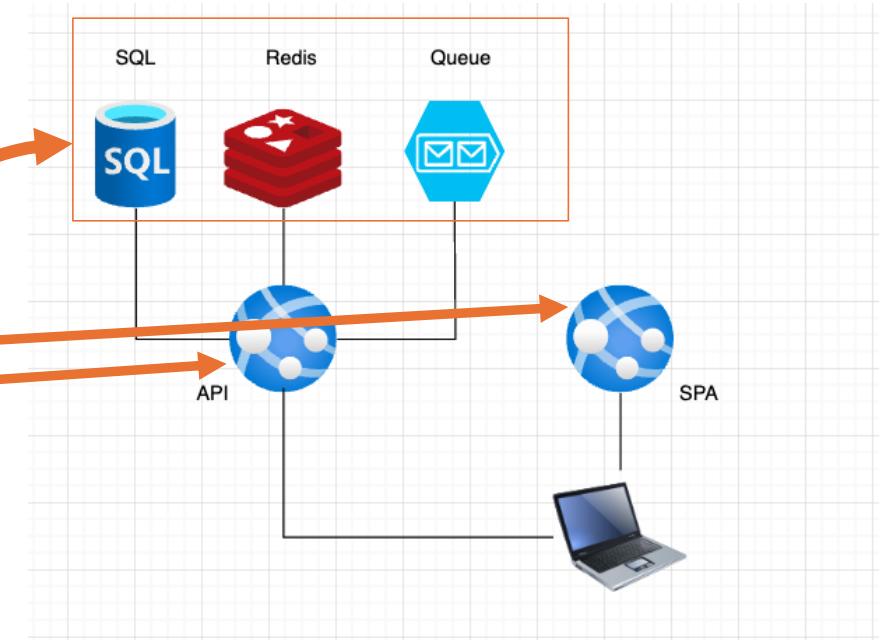


agenda



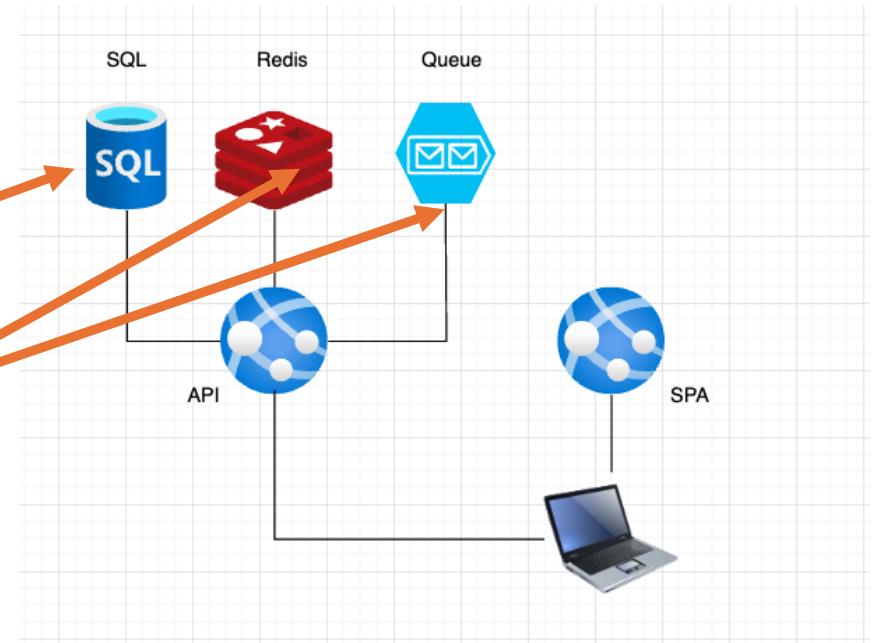
before aspire(folders)

```
✓ src-old
  > angular
  > dotnet
  ✓ infra
    > azurite
  docker-compose.yaml
```



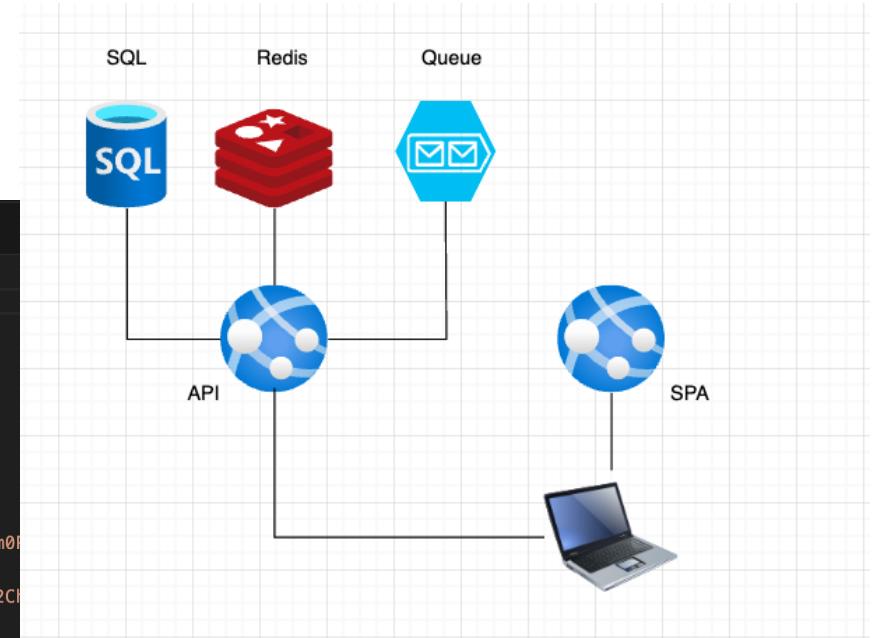
before aspire(w docker)

```
src-old > infra > docker-compose.yaml > name
  1  name: 'demo-infra'
  2  > Run All Services
  3  services:
  4  > Run Service
  5    mssql:
  6      image: mcr.microsoft.com/mssql/server:2022-latest
  7      container_name: mssql
  8      environment: --
  9      ports: --
 10     volumes:
 11       - mssql_data:/var/opt/mssql
 12
 13
 14
 15
 16  > Run Service
 17  azurite:
 18    image: mcr.microsoft.com/azure-storage/azurite
 19    container_name: azurite
 20    ports: --
 21    volumes: --
 22    command: "azurite --blobHost 0.0.0.0 --queueHost 0.0.0.0 --tableHost 0.0.0.0 --skipApiVersionCheck"
 23
 24
 25
 26
 27
 28  > Run Service
 29  redis:
 30    container_name: redis
 31    image: redis:latest
 32    ports:
 33      - "6379:6379"
 34    volumes:
 35      - redis_data:/data
 36
 37  volumes:
 38    mssql_data:
 39    redis_data:
 40    azurite_data:
```



before aspire(conn str)

```
{ appsettings.json } X  
src-old > dotnet > dotnetwebapi > { appsettings.json } ...  
1  {  
2    "Logging": {  
3      "LogLevel": {  
4        "Default": "Information",  
5        "Microsoft.AspNetCore": "Warning"  
6      }  
7    },  
8    "AllowedHosts": "*"  
9    "ConnectionStrings": {  
10      "Sql": "Server=localhost,1433;Database=AppDbContext;User Id=sa;Password=d3m0r3;MultipleActiveResultSets=true",  
11      "Redis": "localhost:6379",  
12      "Storage": "SharedAccessSignature=sv=2023-01-03&ss=btqf&srt=sco&spr=https%2C&t=1673008800&sig=4d533333333333333333333333333333"  
13    }  
14  }  
15
```



before aspire(running)



```
~/source/JuniorDev-6August2025/src-old/infra git:(main) docker compose -f ./docker-compose.yaml up -d
[+] Running 5/8
  ✓ Network demo-infra_default
  ✓ Volume "demo-infra_azurite_data"
  ✓ Volume "demo-infra_redis_data"
  ✓ Volume "demo-infra_mssql_data"
[+] Running 6/8
  ✓ Network demo-infra_default
  ✓ Volume "demo-infra_azurite_data"
  ✓ Volume "demo-infra_redis_data"
  ✓ Volume "demo-infra_mssql_data"
[+] Running 8/8
  ✓ Network demo-infra_default
  ✓ Volume "demo-infra_azurite_data"
  ✓ Volume "demo-infra_redis_data"
  ✓ Volume "demo-infra_mssql_data"
  ✓ Container redis
  ✓ Container mssql
  ✓ Container azurite
    Started 0.3s
! mssql The requested image's platform (linux/amd64) does not match the host (arm64/v8) and no specific platform was requested 0.0s
```



```
~/source/JuniorDev-6August2025/src-old/dotnet/dotnetwebapi dotnet watch run
CREATE TABLE [TestDatas] (
  [Id] int NOT NULL IDENTITY,
  [Name] nvarchar(max) NOT NULL,
  [Description] nvarchar(max) NOT NULL,
  [CreatedAt] datetime2 NOT NULL,
  [UpdatedAt] datetime2 NOT NULL,
  CONSTRAINT [PK_TestDatas] PRIMARY KEY ([Id])
);
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (19ms) [Parameters=[], CommandType='Text',
INSERT INTO [__EFMigrationsHistory] ([MigrationId], [ProductVersion])
VALUES ('N'20250723105017_Initial', 'N'9.0.5');
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (9ms) [Parameters=[], CommandType='Text',
DECLARE @result int;
EXEC @result = sp_releaseapplock @Resource = '__EFMigrationsLock';
SELECT @result
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5062
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
```



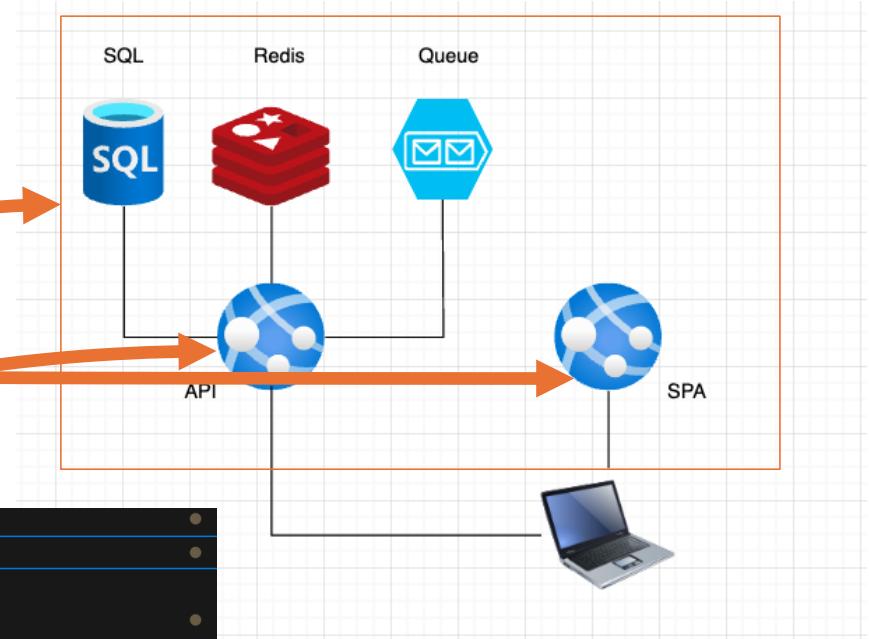
```
v22.15.1 ~/source/JuniorDev-6August2025/src-old/angular npm start
npm start
> angular@0.0.0 start
> ng serve
Component HMR has been enabled, see https://angular.dev/hmr for more info.
Initial chunk files | Names | Raw size
polyfills.js | polyfills | 89.77 kB |
main.js | main | 28.10 kB |
styles.css | styles | 95 bytes |
| Initial total | 117.96 kB
Application bundle generation complete. [0.837 seconds]
Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
→ Local: http://localhost:4200/
→ press h + enter to show help
```



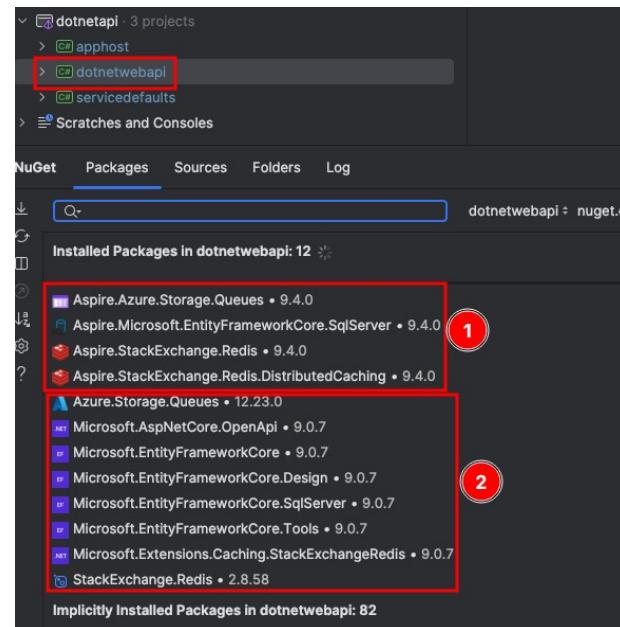
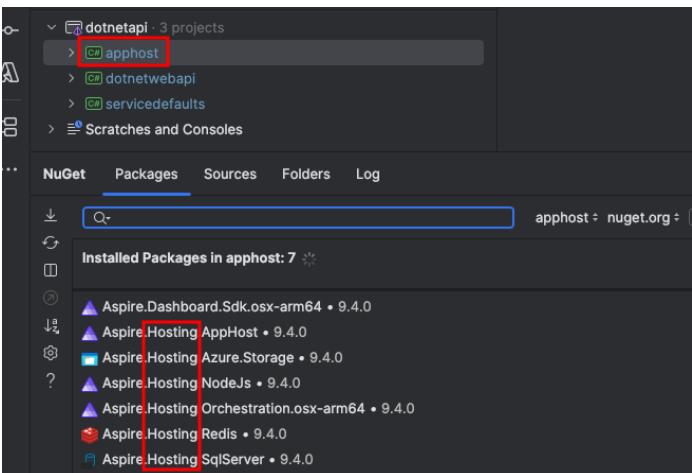
after aspire(folders)

```
✓ src-new
  > angular
  > aspire
  > dotnet
```

```
✓ aspire
  ✓ apphost
    > bin
    > obj
    > Properties
    & .gitignore
    C# AppHost.cs
    & apphost.csproj
    {} appsettings.Development.json
    {} appsettings.json
    ✓ servicedefaults
      > bin
      > obj
      & .gitignore
      C# Extensions.cs
      & servicedefaults.csproj
```



after aspire(nuget)



after aspire(apphost)

```
C# AppHost.cs x C# Program.cs     C# TestController.cs
1  using Projects;
2  var builder = DistributedApplication.CreateBuilder(args);
3
4  var sqlserver :IResourceBuilder<SqlServerServerResource> = builder.AddSqlServer( name: "sqlserver");
5  var appDb :IResourceBuilder<SqlServerDatabaseResource> = sqlserver.AddDatabase( name: "appDb");
6  var redis :IResourceBuilder<RedisResource> = builder.AddRedis( name: "redis");
7  var storage :IResourceBuilder<AzureStorageResource> = builder.AddAzureStorage( name: "storage").RunAsEmulator();
8  var queues :IResourceBuilder<AzureQueueStorageResource> = storage.AddQueues("test-queue");
9
10 var api :IResourceBuilder<ProjectResource> = builder.AddProject<dotnetwebapi>( name: "webapi")
11     .WithEnvironment( name: "ASPIRED_BOOTSTRAP", "true")
12     .WithReference(appDb).WaitFor(appDb)
13     .WithReference(queues).WaitFor(queues)
14     .WithReference(redis).WaitFor(redis);
15
16 var spa :IResourceBuilder<NodeAppResource> = builder.AddNpmApp( name: "spa", workingDirectory: "../../angular")
17     .WithReference(api).WaitFor(api)
18     .WithUrl("http://localhost:4200")
19     .WithHttpEndpoint(env: "PORT");
20
21 builder.Build().Run();
22
```



after aspire(conn str)

```
var sqlserver :IResourceBuilder<SqlServerServerResource> = builder.AddSqlServer( name: "sqlserver");
var appDb :IResourceBuilder<SqlServerDatabaseResource> = sqlserver.AddDatabase( name: "appDb");
var redis :IResourceBuilder<RedisResource> = builder.AddRedis( name: "redis");
var storage :IResourceBuilder<AzureStorageResource> = builder.AddAzureStorage( name: "storage").RunAsEmulator();
var queues :IResourceBuilder<AzureQueueStorageResource> = storage.AddQueues("test-queue");

var api :IResourceBuilder<ProjectResource> = builder.AddProject<dotnetwebapi>( name: "webapi")
    .WithEnvironment( name: "ASPIRED_BOOTSTRAP", "true")
    .WithReference(appDb).WaitFor(appDb)
    .WithReference(queues).WaitFor(queues)
    .WithReference(redis).WaitFor(redis);
```



after aspire(conn str cont.)

```
var isAspiredBootstrap = Environment.GetEnvironmentVariable("ASPIRED_BOOTSTRAP") == "true";
if (isAspiredBootstrap)
{
    builder.AddServiceDefaults();
    builder.AddSqlServerDbContext<AppDbContext>( connectionName: "appDb");
    builder.AddRedisDistributedCache(connectionName: "redis");
    builder.AddAzureQueueServiceClient( connectionName: "test-queue");
    builder.Services.AddSingleton(sp =>
    {
        var qService = sp.GetRequiredService<QueueServiceClient>();
        return qService.GetQueueClient("test-queue");
    });
}
else
{
    var sqlConnectionString = builder.Configuration.GetConnectionString( name: "Sql");
    var redisConnectionString = builder.Configuration.GetConnectionString( name: "Redis");
    var storageConnectionString = builder.Configuration.GetConnectionString( name: "Storage");
    builder.Services.AddDbContext<AppDbContext>(options => options.UseSqlServer(sqlConnectionString));
    builder.Services.AddStackExchangeRedisCache(options => { options.Configuration = redisConnectionString; });
    builder.Services.AddScoped(_ => new QueueClient(storageConnectionString, queueName: "test-queue"));
}
```



after aspire(defaults)



Solution ▾

dotnetapi · 3 projects

- apphost
 - Dependencies
 - Properties
 - .gitignore
 - AppHost.cs
 - appsettings.json
 - appsettings.Development.json
- dotnetwebapi
 - Dependencies
 - Properties
 - Controllers
 - Migrations
 - AppDbContext.cs
 - appsettings.json
 - appsettings.Development.json
 - dotnetwebapi.http
 - Program.cs
 - TestService.cs
- servicedefaults
 - Dependencies
 - .gitignore
 - Extensions.cs

Scratches and Consoles

AppHost.cs

```
5 var builder = WebApplication.CreateBuilder(args);
6 // Allow CORS from any origin
7 builder.Services.AddCors(options =>
8     [...]);
9
10 builder.Services.AddControllers();
11
12 var isAspiredBootstrap = Environment.GetEnvironmentVariable("ASPIRE_BOOTSTRAP");
13 if (isAspiredBootstrap)
14 {
15     builder.AddServiceDefaults();
16     builder.AddSqlServerDbContext<AppDbContext>(connectionName: "test-queue");
17     builder.AddRedisDistributedCache(connectionName: "redis");
18     builder.AddAzureQueueServiceClient<QueueServiceClient>(connectionName: "test-queue");
19     builder.Services.AddSingleton(sp =>
20         {
21             var qService = sp.GetRequiredService<QueueServiceClient>();
22             return qService.GetQueueClient("test-queue");
23         });
24 }
25 else [...]
26
27 builder.Services.AddOpenApi();
28
29 var app = builder.Build();
30 using (var scope = app.Services.CreateScope())
31 {
32     [...]
33
34     app.MapOpenApi();
35     app.MapDefaultEndpoints();
36     app.UseHttpsRedirection();
37     app.UseCors();
38     app.UseAuthorization();
39     app.MapControllers();
40     app.Run();
41 }
```

Extensions.cs

```
1 < > using ...
2
3 // Adds common .NET Aspire services: service discovery, resilience, etc.
4 // This project should be referenced by each service project in your application.
5 // To learn more about using this project, see https://aka.ms/dotnetaspire
6
7 public static class Extensions
8 {
9     private const string HealthEndpointPath = "/health";
10    private const string AlivenessEndpointPath = "/alive";
11
12    public static TBuilder AddServiceDefaults<TBuilder>(this TBuilder builder)
13    {
14        [...]
15    }
16
17    public static TBuilder ConfigureOpenTelemetry<TBuilder>(this TBuilder builder)
18    {
19        [...]
20    }
21
22    public static TBuilder AddOpenTelemetryExporters<TBuilder>(this TBuilder builder)
23    {
24        [...]
25    }
26
27    public static WebApplication MapDefaultEndpoints(this WebApplication app)
28    {
29        [...]
30    }
31 }
```

The code editor shows two files: AppHost.cs and Extensions.cs. AppHost.cs contains logic for building a web application, including adding services like CORS, controllers, and endpoints. It also checks an environment variable ASPIRE_BOOTSTRAP to determine whether to use specific configuration or default values. Extensions.cs is a static class containing extension methods for TBuilder, such as AddServiceDefaults, ConfigureOpenTelemetry, AddOpenTelemetryExporters, and MapDefaultEndpoints. Red arrows point from the highlighted code in AppHost.cs to the corresponding implementation in Extensions.cs.

after aspire(defaults cont.)



```
C# Extensions.cs x
  ↗ Sinan Nar
  ↗ 1 usage ↗ Sinan Nar
16  public static class Extensions
17  {
18      private const string HealthEndpointPath = "/health";
19      private const string AlivenessEndpointPath = "/alive";
20
21      ↗ 1 usage ↗ Sinan Nar
22      public static TBuilder AddServiceDefaults<TBuilder>(this TBuilder builder) where TBuilder : IHostApplicationBuilder
23      {
24          builder.ConfigureOpenTelemetry();
25
26          builder.AddDefaultHealthChecks();
27
28          builder.Services.AddServiceDiscovery();
29
30          builder.Services.ConfigureHttpClientDefaults(http =>
31          {
32              // Turn on resilience by default
33              http.AddStandardResilienceHandler();
34
35              // Turn on service discovery by default
36              http.AddServiceDiscovery();
37          });
38
39          // Uncomment the following to restrict the allowed schemes for service discovery.
40          // builder.Services.Configure<ServiceDiscoveryOptions>(options =>
41          // {
42          //     options.AllowedSchemes = ["https"];
43          // });
44
45          return builder;
46      }
    }
```

after aspire(defaults cont.)

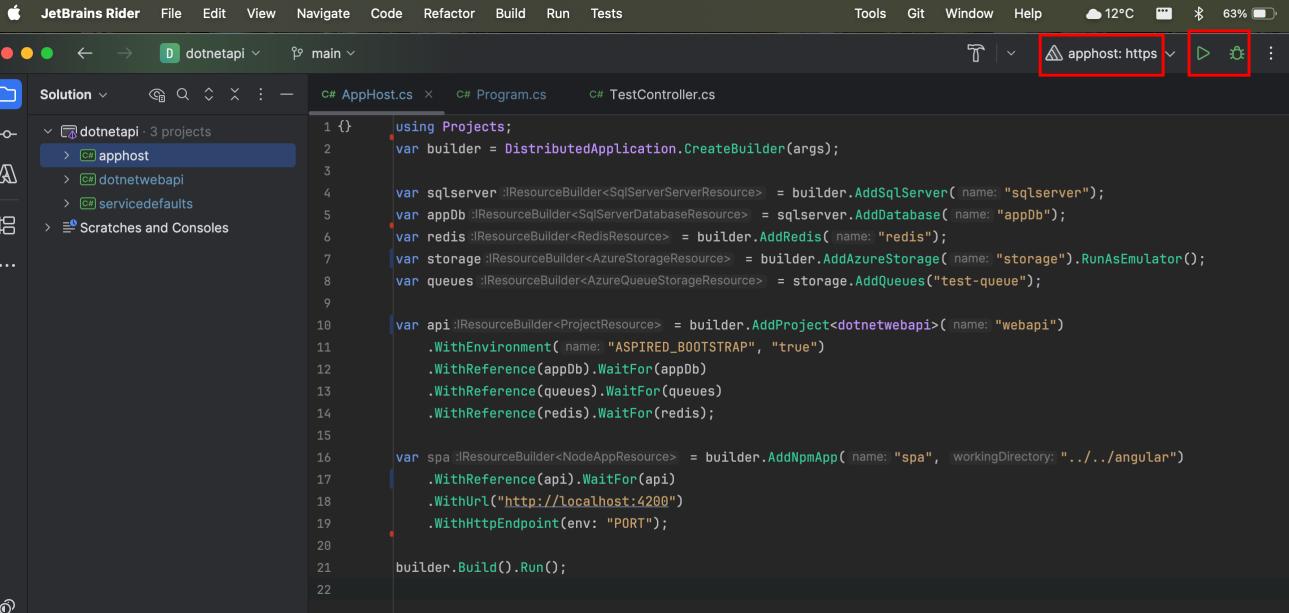
```
1 usage  ↵ Sinan Nar
public static WebApplication MapDefaultEndpoints(this WebApplication app)
{
    // Adding health checks endpoints to applications in non-development environments has security implications.
    // See https://aka.ms/dotnet/aspire/healthchecks for details before enabling these endpoints in non-development environments.
    if (app.Environment.IsDevelopment())
    {
        // All health checks must pass for app to be considered ready to accept traffic after starting
        app.MapHealthChecks( pattern: HealthEndpointPath);

        // Only health checks tagged with the "live" tag must pass for app to be considered alive
        app.MapHealthChecks( pattern: AlivenessEndpointPath, new HealthCheckOptions
        {
            Predicate = r :> r.Tags.Contains("live")
        });
    }

    return app;
}
```



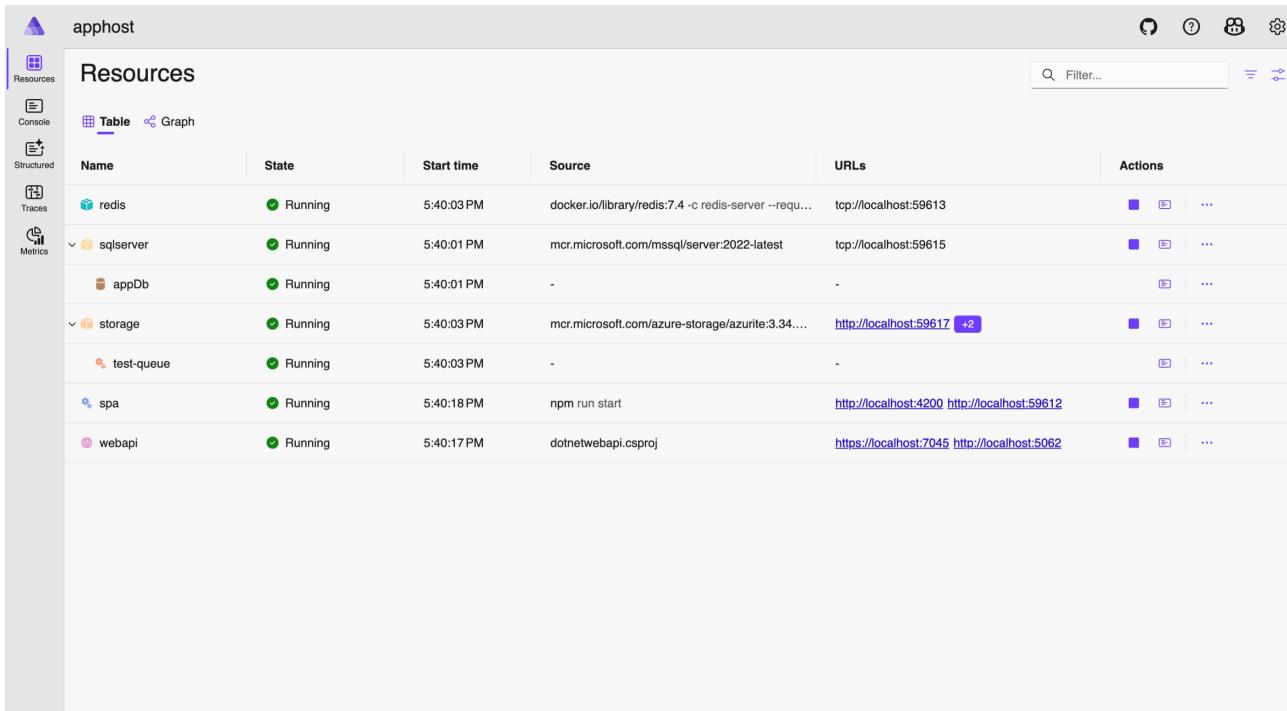
after aspire(running)



```
1 <using Projects;
2 var builder = DistributedApplication.CreateBuilder(args);
3
4 var sqlserver :IResourceBuilder<SqlServerServerResource> = builder.AddSqlServer( name: "sqlserver");
5 var appDb :IResourceBuilder<SqlServerDatabaseResource> = sqlserver.AddDatabase( name: "appDb");
6 var redis :IResourceBuilder<RedisResource> = builder.AddRedis( name: "redis");
7 var storage :IResourceBuilder<AzureStorageResource> = builder.AddAzureStorage( name: "storage").RunAsEmulator();
8 var queues :IResourceBuilder<AzureQueueStorageResource> = storage.AddQueues("test-queue");
9
10 var api :IResourceBuilder<ProjectResource> = builder.AddProject<dotnetwebapi>( name: "webapi")
11     .WithEnvironment( name: "ASPIRED_BOOTSTRAP", "true")
12     .WithReference(appDb).WaitFor(appDb)
13     .WithReference(queues).WaitFor(queues)
14     .WithReference(redis).WaitFor(redis);
15
16 var spa :IResourceBuilder<NodeAppResource> = builder.AddNpmApp( name: "spa", workingDirectory: "../../angular")
17     .WithReference(api).WaitFor(api)
18     .WithUrl("http://localhost:4000")
19     .WithHttpEndpoint(env: "PORT");
20
21 builder.Build().Run();
```



aspire dashboard

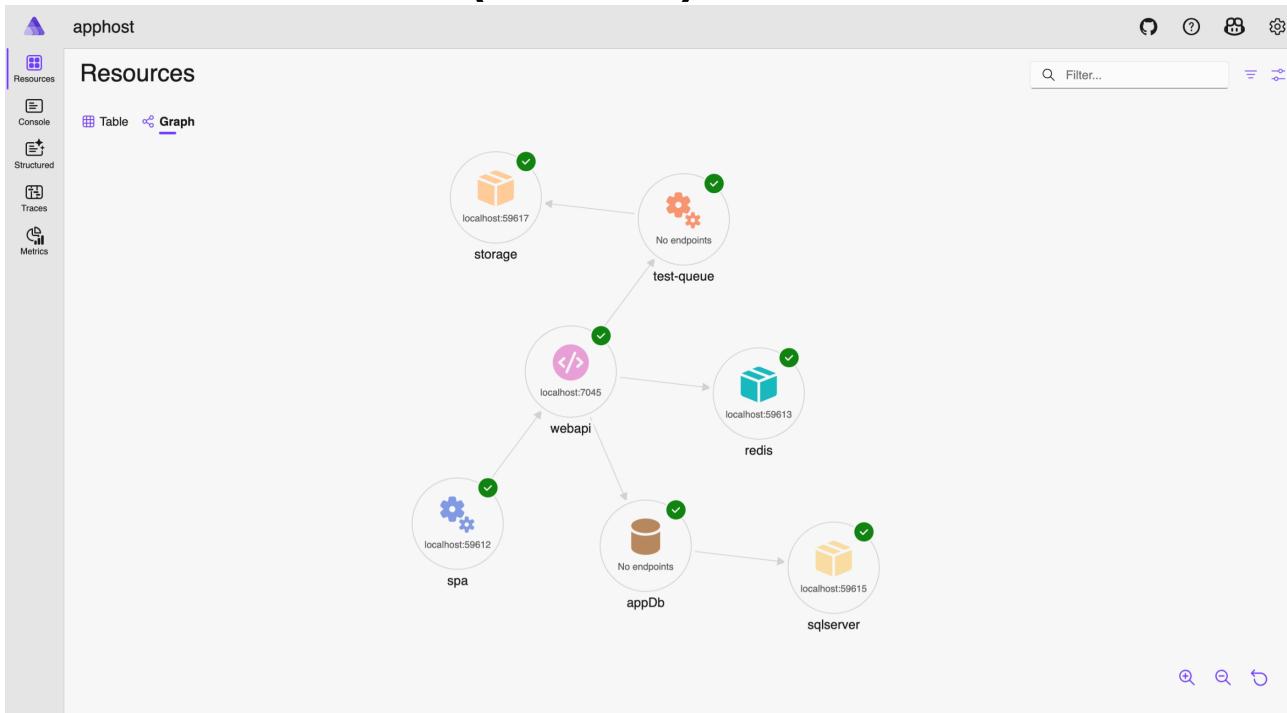


The screenshot shows the aspire dashboard interface for an application named "apphost". The main view is a "Resources" table. The table has columns: Name, State, Start time, Source, URLs, and Actions. There are 7 rows in the table:

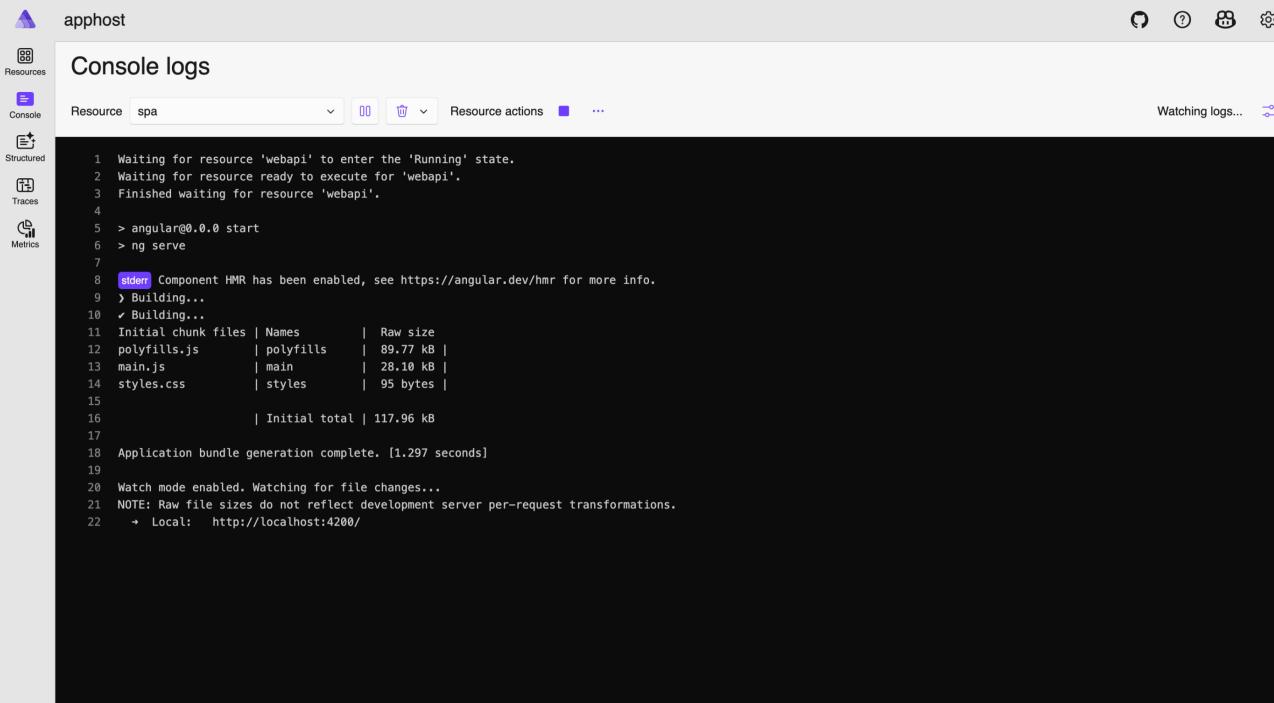
Name	State	Start time	Source	URLs	Actions
redis	Running	5:40:03 PM	docker.io/library/redis:7.4 -c redis-server --requ...	tcp://localhost:59613	[Copy] [Edit] [...]
sqlserver	Running	5:40:01 PM	mcr.microsoft.com/mssql/server:2022-latest	tcp://localhost:59615	[Copy] [Edit] [...]
appDb	Running	5:40:01 PM	-	-	[Copy] [Edit] [...]
storage	Running	5:40:03 PM	mcr.microsoft.com/azure-storage/azurite:3.34....	http://localhost:59617 [+2]	[Copy] [Edit] [...]
test-queue	Running	5:40:03 PM	-	-	[Copy] [Edit] [...]
spa	Running	5:40:18 PM	npm run start	http://localhost:4200 http://localhost:59612	[Copy] [Edit] [...]
webapi	Running	5:40:17 PM	dotnetwebapi.csproj	https://localhost:7045 http://localhost:5062	[Copy] [Edit] [...]



aspire dashboard (cont.)



aspire dashboard (cont.)



The screenshot shows the aspire dashboard interface with the application name "apphost" at the top. On the left, there's a sidebar with icons for Resources, Console (which is selected), Traces, and Metrics. The main area is titled "Console logs" and shows the following log output:

```
1 Waiting for resource 'webapi' to enter the 'Running' state.
2 Waiting for resource ready to execute for 'webapi'.
3 Finished waiting for resource 'webapi'.
4
5 > angular@0.0.0 start
6 > ng serve
7
8 [stderr] Component HMR has been enabled, see https://angular.dev/hmr for more info.
9 ) Building...
10 ✓ Building...
11 Initial chunk files | Names      | Raw size
12 polyfills.js        | polyfills   | 89.77 kB |
13 main.js             | main        | 28.10 kB |
14 styles.css          | styles      | 95 bytes |
15
16           | Initial total | 117.96 kB
17
18 Application bundle generation complete. [1.297 seconds]
19
20 Watch mode enabled. Watching for file changes...
21 NOTE: Raw file sizes do not reflect development server per-request transformations.
22 + Local: http://localhost:4200/
```



aspire dashboard (cont.)



apphost

Console logs

Resource webapi

Structured

Traces

Metrics

Watching logs...

```
72 EXEC sp_who2 @SPID=@@SPID, @LockCount=1, @Err=1, @Status=1, @LoginName='sa', @Session=1
73 SELECT @result
74 info: Microsoft.Hosting.Lifetime[14]
75 Now listening on: https://localhost:59729
76 info: Microsoft.Hosting.Lifetime[14]
77 Now listening on: http://localhost:59730
78 info: Microsoft.Hosting.Lifetime[0]
79 Application started. Press Ctrl+C to shut down.
80 info: Microsoft.Hosting.Lifetime[0]
81 Hosting environment: Development
82 info: Microsoft.Hosting.Lifetime[0]
83 Content root path: /Users/sinannar/source/JuniorDev-6August2025/src-new/dotnet/dotnetwebapi
84 info: dotnetwebapi.Controllers.TestController[0]
85 InsertData called with name: test, description: test
86 info: Microsoft.EntityFrameworkCore.Database.Command[20101]
87 Executed DbCommand (53ms) [Parameters=@p0='?' (DbType = DateTime2), @p1='?' (Size = 4000), @p2='?' (Size = 4000), @p3='?' (DbType = DateTime2)], CommandType='Text', CommandTimeout='30'
88 SET IMPLICIT_TRANSACTIONS OFF;
89 SET NOCOUNT ON;
90 INSERT INTO [TestDatas] ([CreatedAt], [Description], [Name], [UpdatedAt])
91 OUTPUT INSERTED.[Id]
92 VALUES (@p0, @p1, @p2, @p3);
93 info: dotnetwebapi.Controllers.TestController[0]
94 Data inserted with Id: 1
95 info: dotnetwebapi.Controllers.TestController[0]
96 ReadData called with id: 1
97 info: Microsoft.EntityFrameworkCore.Database.Command[20101]
98 Executed DbCommand (13ms) [Parameters=@_p_0='?' (DbType = Int32)], CommandType='Text', CommandTimeout='30'
99 SELECT TOP(1) [t].[Id], [t].[CreatedAt], [t].[Description], [t].[Name], [t].[UpdatedAt]
100 FROM [TestDatas] AS [t]
101 WHERE [t].[Id] = @_p_0
102 info: dotnetwebapi.Controllers.TestController[0]
103 Data retrieved with Id: 1, Name: test02/08/2025 5:41:08 PM +12:00, Description: test02/08/2025 5:41:08 PM +12:00
```

aspire dashboard (cont.)

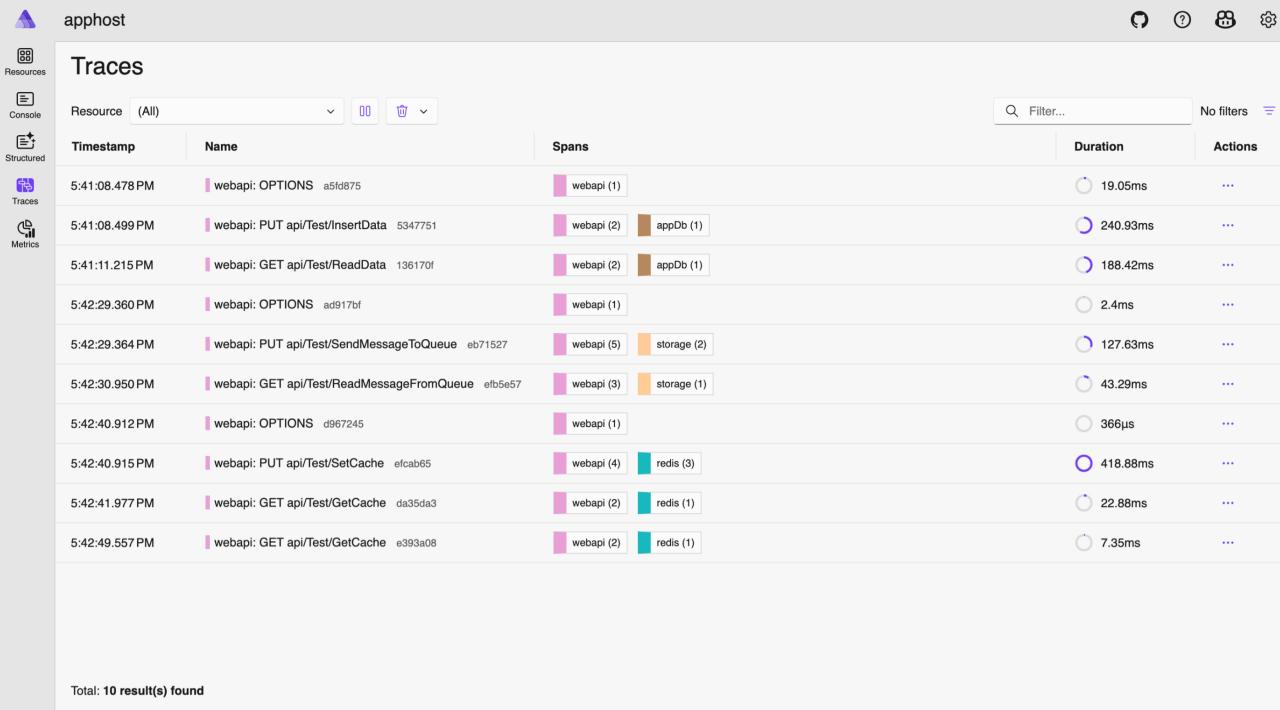


The screenshot shows the 'Structured logs' section of the aspire dashboard for the 'apphost' resource. The interface includes a left sidebar with 'Resources' (selected), 'Console', 'Structured' (selected), 'Traces', and 'Metrics'. The main area has tabs for 'Resource' (All) and 'Level' (All). It features a search bar with 'Explain errors' and 'Filter...', and a 'No filters' button. A table lists log entries with columns for 'Resource', 'Level', 'Timestamp', 'Message', 'Trace', and 'Actions'. The 'Resource' column shows 'webapi' repeated 15 times. The 'Level' column shows 'Information' for most entries and 'Error' for two entries. The 'Timestamp' column shows dates from 6:01:09.211 PM to 6:01:23.904 PM. The 'Message' column contains various log messages, including database command executions and unhandled exceptions. The 'Trace' column provides links to specific trace details, such as 'df56502', 'bc2cdf1', and 'e6c6b6e'. The 'Actions' column contains three-dot ellipsis buttons. At the bottom, it says 'Total: 94 result(s) found'.

Resource	Level	Timestamp	Message	Trace	Actions
webapi	Information	6:01:09.211 PM	Executed DbCommand (5ms) [Parameters=@__p_0=? (DbType = Int32)], CommandType='...', CommandTimeout=30	df56502	...
webapi	Error	6:01:09.252 PM	An unhandled exception has occurred while executing the request.	df56502	...
webapi	Information	6:01:10.137 PM	ReadData called with id: 0	bc2cdf1	...
webapi	Information	6:01:10.144 PM	Executed DbCommand (4ms) [Parameters=@__p_0=? (DbType = Int32)], CommandType='...', CommandTimeout=30	bc2cdf1	...
webapi	Error	6:01:10.178 PM	An unhandled exception has occurred while executing the request.	bc2cdf1	...
webapi	Information	6:01:17.657 PM	ReadData called with id: 0	e6c6b6e	...
webapi	Information	6:01:17.666 PM	Executed DbCommand (6ms) [Parameters=@__p_0=? (DbType = Int32)], CommandType='...', CommandTimeout=30	e6c6b6e	...
webapi	Error	6:01:17.712 PM	An unhandled exception has occurred while executing the request.	e6c6b6e	...
webapi	Information	6:01:20.790 PM	ReadData called with id: 1	060929a	...
webapi	Information	6:01:20.796 PM	Executed DbCommand (3ms) [Parameters=@__p_0=? (DbType = Int32)], CommandType='...', CommandTimeout=30	060929a	...
webapi	Information	6:01:20.811 PM	Data retrieved with Id: 1, Name: test02/08/2025 6:01:05 PM +12:00, Description: test02/08/20...	060929a	...
webapi	Information	6:01:23.879 PM	Request [db5e49cd-8fd5-408e-b517-72600c230f8c] PUT http://127.0.0.1:65010/devstoreacco...	7658c54	...
webapi	Information	6:01:23.904 PM	Response [db5e49cd-8fd5-408e-b517-72600c230f8c] 201 Created (0.0s) Server:Azurite-Qu...	7658c54	...



aspire dashboard (cont.)



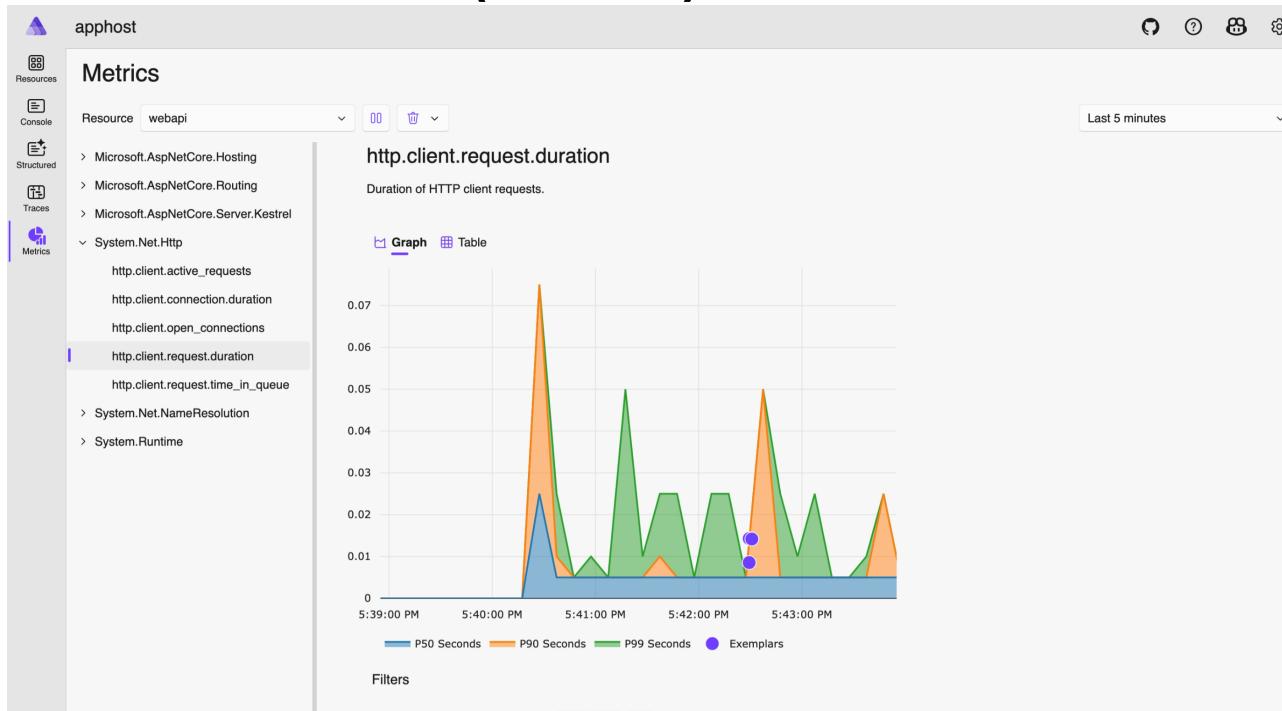
The screenshot shows the aspire Traces dashboard for the apphost service. The interface includes a sidebar with 'Resources' (Console, Structured, Traces, Metrics), a search bar, and filter options. The main area displays a table of traces with columns: Timestamp, Name, Spans, Duration, and Actions.

Timestamp	Name	Spans	Duration	Actions
5:41:08.478 PM	webapi: OPTIONS a5fd875	webapi (1)	19.05ms	...
5:41:08.499 PM	webapi: PUT api/Test/InsertData 5347751	webapi (2) appDb (1)	240.93ms	...
5:41:11.215 PM	webapi: GET api/Test/ReadData 136170f	webapi (2) appDb (1)	188.42ms	...
5:42:29.360 PM	webapi: OPTIONS ad917bf	webapi (1)	2.4ms	...
5:42:29.364 PM	webapi: PUT api/Test/SendMessageToQueue eb71527	webapi (5) storage (2)	127.63ms	...
5:42:30.950 PM	webapi: GET api/Test/ReadMessageFromQueue efb5e57	webapi (3) storage (1)	43.29ms	...
5:42:40.912 PM	webapi: OPTIONS d967245	webapi (1)	366µs	...
5:42:40.915 PM	webapi: PUT api/Test/SetCache efca65	webapi (4) redis (3)	418.88ms	...
5:42:41.977 PM	webapi: GET api/Test/GetCache da35da3	webapi (2) redis (1)	22.88ms	...
5:42:49.557 PM	webapi: GET api/Test/GetCache e393a08	webapi (2) redis (1)	7.35ms	...

Total: 10 result(s) found



aspire dashboard (cont.)



```

1  var builder = DistributedApplication.CreateBuilder(args);
2
3  var postgres = builder.AddPostgres("postgres")
4      .WithPgAdmin()
5      .WithLifetime(ContainerLifetime.Persistent);
6
7  if (builder.ExecutionContext.IsRunMode)
8  {
9      // Data volumes don't work on ACA for Postgres so only add when running
10     postgres.WithDataVolume();
11 }
12
13 var catalogDb = postgres.AddDatabase("catalogdb");
14
15 var basketCache = builder.AddRedis("basketcache")
16     .WithDataVolume()
17     .WithRedisCommander();
18
19 var catalogDbManager = builder.AddProject<Projects.AspireShop_CatalogDbManager>("catalogdbmanager")
20     .WithReference(catalogDb)
21     .WaitFor(catalogDb)
22     .WithHttpHealthCheck("/health")
23     .WithHttpCommand("/reset-db", "Reset Database", commandOptions: new() { IconName = "DatabaseLightning" });
24
25 var catalogService = builder.AddProject<Projects.AspireShop_CatalogService>("catalogservice")
26     .WithReference(catalogDb)
27     .WaitFor(catalogDbManager)
28     .WithHttpHealthCheck("/health");
29
30 var basketService = builder.AddProject<Projects.AspireShop_BasketService>("basketservice")
31     .WithReference(basketCache)
32     .WaitFor(basketCache);
33
34 builder.AddProject<Projects.AspireShop_Frontend>("frontend")
35     .WithExternalHttpEndpoints()
36     .WithUrlForEndpoint("https", url => url.DisplayText = "Online Store (HTTPS)")
37     .WithUrlForEndpoint("http", url => url.DisplayText = "Online Store (HTTP)")
38     .WithHttpHealthCheck("/health")
39     .WithReference(basketService)
40     .WithReference(catalogService)
41     .WaitFor(catalogService);
42
43 builder.Build().Run();

```

1

2

3

4

frontend/program.cs



examples

```
1  using Azure.Provisioning.Storage;
2
3  var builder = DistributedApplication.CreateBuilder(args);
4
5  builder.AddAzureContainerAppEnvironment("env");
6
7  var storage = builder.AddAzureStorage("storage").RunAsEmulator()
8      .ConfigureInfrastructure((infrastructure) =>
9      {
10        var storageAccount = infrastructure.GetProvisionableResources().OfType<StorageAccount>().FirstOrDefault(r => r.BicepIdentifier == "storage")
11        ?? throw new InvalidOperationException($"Could not find configured storage account with name 'storage'");
12
13        // Ensure that public access to blobs is disabled
14        storageAccount.AllowBlobPublicAccess = false;
15    });
16  var blobs = storage.AddBlobs("blobs");
17  var queues = storage.AddQueues("queues");
18
19  var functions = builder.AddAzureFunctionsProject<Projects.ImageGallery_Functions>("functions")
20      .WithReference(queues)
21      .WithReference(blobs)
22      .WaitFor(storage)
23      .WithRoleAssignments(storage,
24          // Storage Account Contributor and Storage Blob Data Owner roles are required by the Azure Functions host
25          StorageBuiltInRole.StorageAccountContributor, StorageBuiltInRole.StorageBlobDataOwner,
26          // Queue Data Contributor role is required to send messages to the queue
27          StorageBuiltInRole.StorageQueueDataContributor)
28      .WithHostStorage(storage);
29
30  builder.AddProject<Projects.ImageGallery_FrontEnd>("frontend")
31      .WithReference(queues)
32      .WithReference(blobs)
33      .WaitFor(functions)
34      .WithExternalHttpEndpoints();
35
36  builder.Build().Run();
```



examples

```
1  var builder = DistributedApplication.CreateBuilder(args);
2
3  var weatherApi = builder.AddProject<Projects.AspireJavaScript_MinimalApi>("weatherapi")
4      .WithExternalHttpEndpoints();
5
6  builder.AddNpmApp("angular", "../AspireJavaScript.Angular")
7      .WithReference(weatherApi) 1
8      .WaitFor(weatherApi)
9      .WithHttpEndpoint(env: "PORT")
10     .WithExternalHttpEndpoints()
11     .PublishAsDockerFile();
12
13 builder.AddNpmApp("react", "../AspireJavaScript.React")
14     .WithReference(weatherApi) 2
15     .WaitFor(weatherApi)
16     .WithEnvironment("BROWSER", "none") // Disable opening browser on npm start
17     .WithHttpEndpoint(env: "PORT")
18     .WithExternalHttpEndpoints()
19     .PublishAsDockerFile();
20
21 builder.AddNpmApp("vue", "../AspireJavaScript.Vue")
22     .WithReference(weatherApi) 3
23     .WaitFor(weatherApi)
24     .WithHttpEndpoint(env: "PORT")
25     .WithExternalHttpEndpoints()
26     .PublishAsDockerFile();
27
28 builder.AddNpmApp("reactvite", "../AspireJavaScript.Vite")
29     .WithReference(weatherApi)
30     .WithEnvironment("BROWSER", "none") 4
31     .WithHttpEndpoint(env: "VITE_PORT")
32     .WithExternalHttpEndpoints()
33     .PublishAsDockerFile();
34
35 builder.Build().Run();
```



examples

```
1  using Microsoft.Extensions.Hosting;
2
3  var builder = DistributedApplication.CreateBuilder(args);
4
5  #pragma warning disable ASPIREHOSTINGPYTHON001 // Type is for evaluation purposes only and is subject to change or removal in future updates. Suppress this diagnostic to proceed.
6  var pythonapp = builder.AddPythonApp("instrumented-python-app", "../InstrumentedPythonProject", "app.py")
7      .WithHttpEndpoint(env: "PORT")
8      .WithExternalHttpEndpoints();
9  #pragma warning restore ASPIREHOSTINGPYTHON001
10
11 if (builder.ExecutionContext.IsRunMode && builder.Environment.IsDevelopment())
12 {
13     pythonapp.WithEnvironment("DEBUG", "True");
14 }
15
16 builder.Build().Run();
```



examples

```
1 var builder = DistributedApplication.CreateBuilder(args);
2
3 var apiService = builder.AddProject<Projects.ClientAppsIntegration_ApiService>("apiservice");
4
5 if (OperatingSystem.IsWindows())
6 {
7     builder.AddProject<Projects.ClientAppsIntegration_WinForms>("winformsclient")
8         .WithReference(apiService)
9         .WaitFor(apiService)
10        .WithExplicitStart()
11        .ExcludeFromManifest();
12
13    builder.AddProject<Projects.ClientAppsIntegration_WPF>("wpfclient")
14        .WithReference(apiService)
15        .WaitFor(apiService)
16        .WithExplicitStart()
17        .ExcludeFromManifest();
18 }
19
20 builder.Build().Run();
```



examples

```
using Microsoft.Extensions.Hosting;

var builder = DistributedApplication.CreateBuilder(args);

var goVersion = builder.AddParameter("goversion", "1.24.2", publishValueAsDefault: true);

var ginapp = builder.AddDockerfile("ginapp", "../ginapp")
    .WithBuildArg("GO_VERSION", goVersion)
    .WithHttpEndpoint(targetPort: 5555, env: "PORT")
    .WithExternalHttpEndpoints();

if (builder.ExecutionContext.IsPublishMode || builder.Environment.IsProduction())
{
    ginapp
        .WithEnvironment("GIN_MODE", "release")
        // Trust all proxies when running behind a reverse proxy. If deploying to an environment
        // without a reverse proxy that ensures X-Forwarded-* headers are not forwarded from clients,
        // this should be removed.
        .WithEnvironment("TRUSTED_PROXYIES", "all");
}

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



examples

```
1 // Licensed to the .NET Foundation under one or more agreements.
2 // The .NET Foundation licenses this file to you under the MIT license.
3
4 var builder = DistributedApplication.CreateBuilder(args);
5
6 var sqlserver = builder.AddSqlServer("sqlserver")
7     .WithLifetime(ContainerLifetime.Persistent);
8
9 var db1 = sqlserver.AddDatabase("db1");
10
11 var migrationService = builder.AddProject<Projects.DatabaseMigrations_MigrationService>("migration")
12     .WithReference(db1)
13     .WaitFor(db1);
14
15 builder.AddProject<Projects.DatabaseMigrations_ApiService>("api")
16     .WithReference(db1)
17     .WaitForCompletion(migrationService);
18
19 builder.Build().Run();
```

Code Blame 21 lines (18 loc) · 916 Bytes

```
1 <Project Sdk="Microsoft.NET.Sdk.Worker">
2
3     <PropertyGroup>
4         <TargetFramework>net8.0</TargetFramework>
5         <Nullable>enable</Nullable>
6         <ImplicitUsings>enable</ImplicitUsings>
7         <!-- Make build happy with generated migrations -->
8         <NoWarn>$ (NoWarn); IDE0161; IDE0005</NoWarn>
9     </PropertyGroup>
10
11     <ItemGroup>
12         <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="8.0.13">
13             <PrivateAssets>all</PrivateAssets>
14             <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
15         </PackageReference>
16         <PackageReference Include="Aspire.Microsoft.EntityFrameworkCore.SqlServer" Version="9.4.0" />
17         <ProjectReference Include=".\\DatabaseMigrations.ApiModel\\DatabaseMigrations.ApiModel.csproj" />
18         <ProjectReference Include=".\\DatabaseMigrations.ServiceDefaults\\DatabaseMigrations.ServiceDefaults.csproj" />
19     </ItemGroup>
20
21 </Project>
```



examples

```
1  using MetricsApp.AppHost.OpenTelemetryCollector;
2
3  var builder = DistributedApplication.CreateBuilder(args);
4
5  var prometheus = builder.AddContainer("prometheus", "prom/prometheus", "v3.2.1")
6      .WithBindMount("../prometheus", "/etc/prometheus", isReadOnly: true)
7      .WithArgs("--web.enable-otlp-receiver", "--config.file=/etc/prometheus/prometheus.yml")
8      .WithHttpEndpoint(targetPort: 9090, name: "http");
9
10 var grafana = builder.AddContainer("grafana", "grafana/grafana")
11     .WithBindMount("./grafana/config", "/etc/grafana", isReadOnly: true)
12     .WithBindMount("./grafana/dashboards", "/var/lib/grafana/dashboards", isReadOnly: true)
13     .WithEnvironment("PROMETHEUS_ENDPOINT", prometheus.GetEndpoint("http"))
14     .WithHttpEndpoint(targetPort: 3000, name: "http");
15
16 builder.AddOpenTelemetryCollector("otelcollector", "../otelcollector/config.yaml")
17     .WithEnvironment("PROMETHEUS_ENDPOINT", $"{prometheus.GetEndpoint("http")}/api/v1/otlp");
18
19 builder.AddProject<Projects.MetricsApp>("app")
20     .WithEnvironment("GRAFANA_URL", grafana.GetEndpoint("http"));
21
22 using var app = builder.Build();
23
24 await app.RunAsync();
```



examples

```
1  var builder = DistributedApplication.CreateBuilder(args);
2
3  var sqlserver = builder.AddSqlServer("sqlserver")
4      .WithDataVolume()
5      .WithLifetime(ContainerLifetime.Persistent);
6
7  var sqlDatabase = sqlserver.AddDatabase("sqldb");
8
9  var postgresServer = builder.AddPostgres("postgresserver")
10     .WithDataVolume()
11     .WithLifetime(ContainerLifetime.Persistent);
12
13 var postgresDatabase = postgresServer.AddDatabase("postgres");
14
15 var blobs = builder.AddAzureStorage("Storage")
16     // Use the Azurite storage emulator for local development
17     .RunAsEmulator(emulator => emulator.WithDataVolume())
18     .AddBlobs("BlobConnection");
19
20 builder.AddProject<Projects.VolumeMount_BazorWeb>("blazorweb")
21     .WithReference(sqlDatabase)
22     .WaitFor(sqlDatabase)
23     .WithReference(postgresDatabase)
24     .WaitFor(postgresDatabase)
25     .WithReference(blobs)
26     .WaitFor(blobs);
27
28 builder.Build().Run();
```



aspire integrations



Integrations

Overview

- > Tutorials
- Apache Kafka
- > Azure
- Elasticsearch
- > Entity Framework Core
- GitHub Models
- Keycloak (Preview)
- Milvus
- MongoDB
- MySQL
- NATS
- > Orleans
- PostgreSQL
- Qdrant
- RabbitMQ service broker
- > Redis
- Seq
- SQL Server
- YARP (Yet Another Reverse Proxy)
- > Community Toolkit
- Aspire.Hosting API reference

Azure

- Overview
- Customize Azure resources
- Local Azure provisioning
- Configure Azure Container Apps environments
- User-assigned managed identity
- Manage role assignments
- Azure App Configuration
- Azure AI Foundry (Preview)
- Azure AI Inference (Preview)
- Azure AI Search
- > Azure Cache for Redis
- Azure Container Registry (Preview)
- Azure Cosmos DB
- [Azure Event Hubs](#)
- Azure Functions (Preview)
- Azure Key Vault
- Azure PostgreSQL
- Azure OpenAI (Preview)
- Azure SignalR Service
- Azure Service Bus
- Azure SQL Database
- > Azure Storage
- Azure Web PubSub
- Aspire.Hosting.Azure API reference
- Aspire.Azure API reference

Community Toolkit

- Overview
- Bun apps
- Dapr
- Deno apps
- Go apps
- Java/Spring
- Node.js extensions
- Python extensions
- Ollama
- Meilisearch
- Rust apps
- SQL Database Projects
- Data API Builder
- EventStore
- > SQLite
- SQL Server Extensions
- MongoDB Extensions
- Redis Extensions
- PostgreSQL Extensions
- MySQL Extensions
- RavenDB



aspire integrations(cont)



The screenshot shows a screenshot of the AWS SDK for .NET (V3) Developer Guide page. The title is "Integrating AWS with .NET Aspire in the AWS SDK for .NET". A callout box highlights a note about version 4 (V4) being released. The main content discusses .NET Aspire's role in building cloud-ready applications and its integration with AWS services like CloudFormation and Lambda. A sidebar on the left lists various developer guide topics, and a right sidebar provides recommended tasks and how-to guides.

AWS SDK for .NET (V3) Developer Guide

[Documentation](#) > [AWS SDK for .NET](#) > [Developer Guide](#)

Integrating AWS with .NET Aspire in the AWS SDK for .NET

[PDF](#) [RSS](#) Focus mode

Version 4 (V4) of the AWS SDK for .NET has been released!
To start using the new version of the SDK, see the [AWS SDK for .NET \(V4\) Developer Guide](#), especially the topic for [Migrating to version 4](#).

.NET Aspire is a new way of building cloud-ready applications. In particular, it provides an orchestration for local environments in which to run, connect, and debug the components of distributed applications. To improve the inner dev loop for cloud ready applications, integrations with .NET Aspire have been created for connecting your .NET applications to AWS resources. These integrations are available through the [Aspire.Hosting.AWS](#) NuGet package.

The following .NET Aspire integrations are available:

- The ability to provision your AWS resources through [AWS CloudFormation](#). This integration is utilized within the .NET Aspire AppHost project.
For more information, see the blog post [Integrating AWS with .NET Aspire](#).
- Installing, configuring, and connecting the AWS SDK for .NET to [Amazon DynamoDB local](#). This integration is utilized within the .NET Aspire AppHost project.
For more information, see the blog post [Integrating AWS with .NET Aspire](#).
- Enable a local development environment for [AWS Lambda](#) functions. This integration is utilized within the .NET Aspire AppHost project.
For more information, see the blog posts [Building and Debugging .NET Lambda applications with .NET Aspire \(Part 1\)](#) and [Building and Debugging .NET Lambda applications with .NET Aspire \(Part 2\)](#).

On this page

[Additional information](#)

Recommended tasks

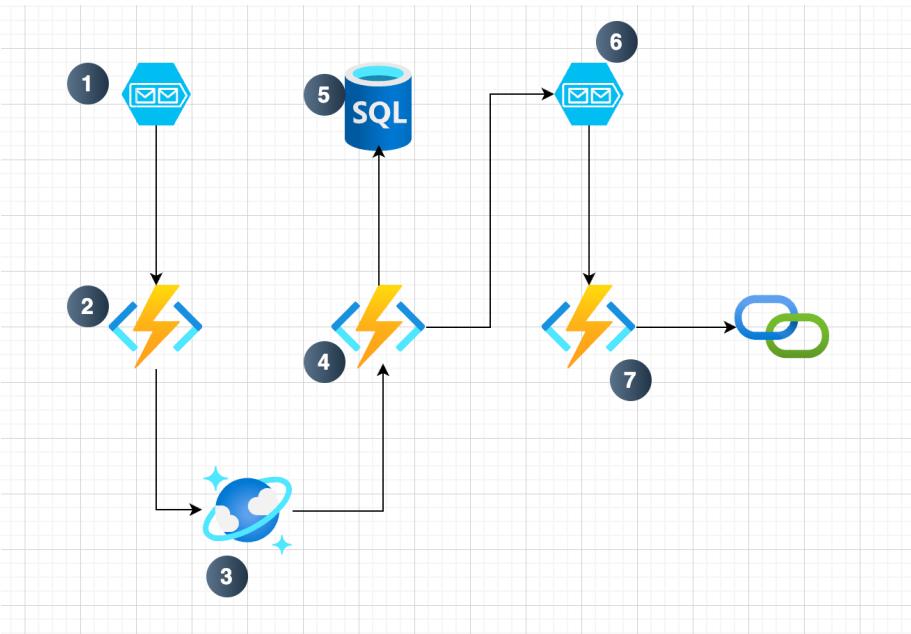
Learn about

- Understand the release history of [DynamoDB local](#)
- Understand AWS OpsWorks features and APIs
- Understand support for other AWS services and configuration
- Understand the AWS SDK for .NET and its key use cases
- Understand how to integrate .NET Aspire with AWS services

How to

- Configure C# Lambda function handlers for .NET runtime

into to .net aspire blog post



thank you for listening



- github.com/dotnet/aspire
- github.com/dotnet/aspire-samples/tree/main/samples
 - AspireShop/AspireShop.AppHost/AppHost.cs
 - AspireWithAzureFunctions/ImageGallery.AppHost/AppHost.cs
 - AspireWithJavaScript/AspireJavaScript.AppHost/AppHost.cs
 - AspireWithPython/AspireWithPython.AppHost/AppHost.cs
 - ClientAppsIntegration/ClientAppsIntegration.AppHost/AppHost.cs
 - ContainerBuild/ContainerBuild.AppHost/AppHost.cs
 - DatabaseMigrations/DatabaseMigrations.AppHost/AppHost.cs
- learn.microsoft.com/en-us/dotnet/aspire/
- github.com/sinannar/JuniorDev-6August2025

