A sidecar is an extra container in your ECS task definition that runs alongside your .NET application container. This sidecar runs the OpenTelemetry Collector (or AWS Distro for OpenTelemetry Collector) and handles exporting telemetry data to your backend (e.g., AWS X-Ray, OTLP, Honeycomb, Dynatrace, etc.).

**Architecture Overview**

+--------------------------+ +-----------------------------+

| .NET App Container | <---> | OTEL Collector Sidecar |

| (.NET with OTEL SDK) | | (Receives OTLP, exports) |

+--------------------------+ +-----------------------------+

| |

| |

OTLP Export Sends to:

(localhost:4317) - AWS X-Ray

- OTLP backend

- Honeycomb, etc.

### Steps to Instrument with Sidecar in ECS

#### 1. **Instrument your .NET app (SDK only)**

Like earlier, modify Program.cs or Startup.cs:

builder.Services.AddOpenTelemetryTracing(builder =>

{

builder

.SetResourceBuilder(ResourceBuilder.CreateDefault().AddService("your-dotnet-service"))

.AddAspNetCoreInstrumentation()

.AddHttpClientInstrumentation()

.AddOtlpExporter(options =>

{

options.Endpoint = new Uri("http://localhost:4317"); // points to sidecar

});

});

This assumes the OTEL Collector is running on localhost:4317 in the same task (i.e., sidecar).

#### 2. **Add OTEL Collector as a Sidecar Container in ECS**

In your ECS task definition, add a second container:

{

"name": "otel-collector",

"image": "amazon/aws-otel-collector:latest",

"essential": true,

"portMappings": [

{

"containerPort": 4317,

"protocol": "tcp"

}

],

"logConfiguration": {

"logDriver": "awslogs",

"options": {

"awslogs-group": "/ecs/otel-collector",

"awslogs-region": "us-west-2",

"awslogs-stream-prefix": "otel"

}

},

"mountPoints": [],

"environment": [

{ "name": "AWS\_REGION", "value": "us-west-2" }

]

}

#### 3. **Configure OTEL Collector**

Mount a config file to the sidecar container. Here's an example config for sending traces to AWS X-Ray:

# otel-collector-config.yaml

receivers:

otlp:

protocols:

grpc:

exporters:

awsxray:

service:

pipelines:

traces:

receivers: [otlp]

exporters: [awsxray]

You can mount this config using a volume in your task definition.

#### 4. **Networking Between App and Sidecar**

Both containers in the ECS task share the localhost network namespace, so your .NET app can call http://localhost:4317 to send OTLP traces to the sidecar.