



Event Orchestration

With State Machines, Mass Transit

December 2023

PLATINUM



Octopus Deploy

GOLD

DevBricks^W

Insight^{IT}

PATIENT ZERO[®]



VALID

SILVER

LMG[®]

Seq^{IT}
Machine data for humans

NDC { Sydney }

STANDARD

SixPivot

CHILDCARE

Octopus Deploy

COFFEE

Seq^{IT}
Machine data for humans



DIVERSITY

SixPivot

MEDIA



COMMUNITY



YOW!

Table Of Contents

Event Driven Concepts

Introducing our Use Case

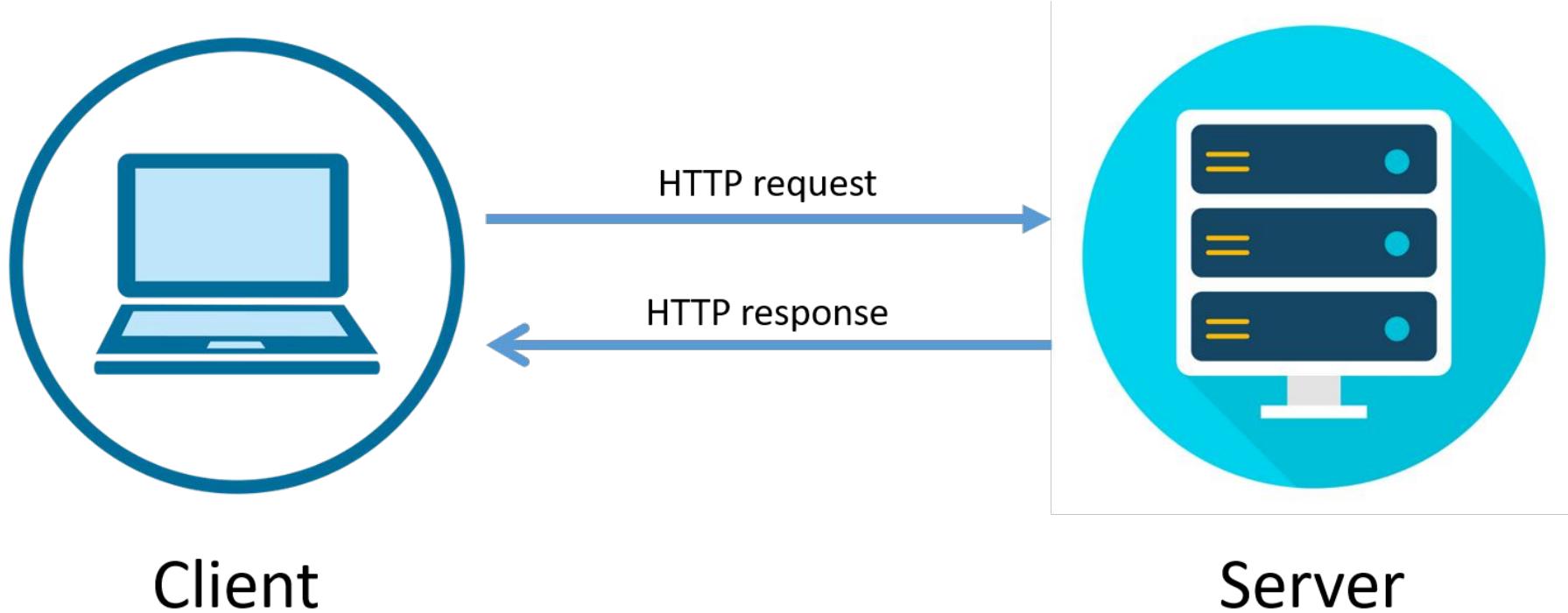
Mass Transit, Sagas and State Machines

Mass Transit Testing

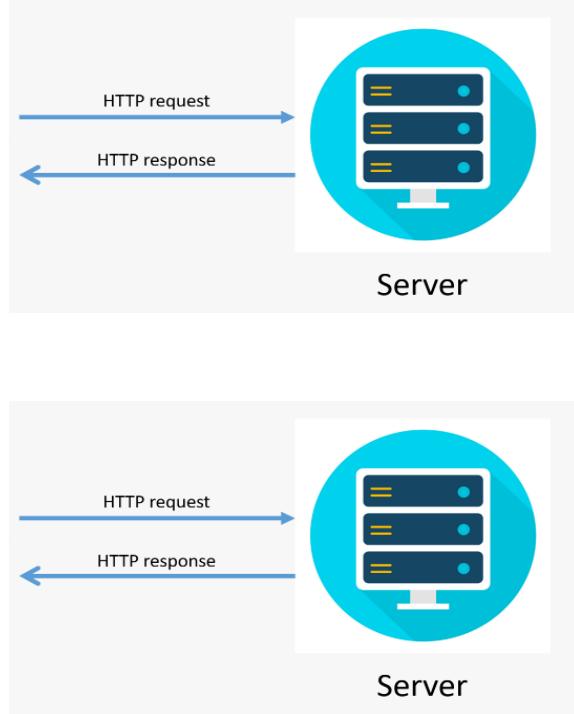
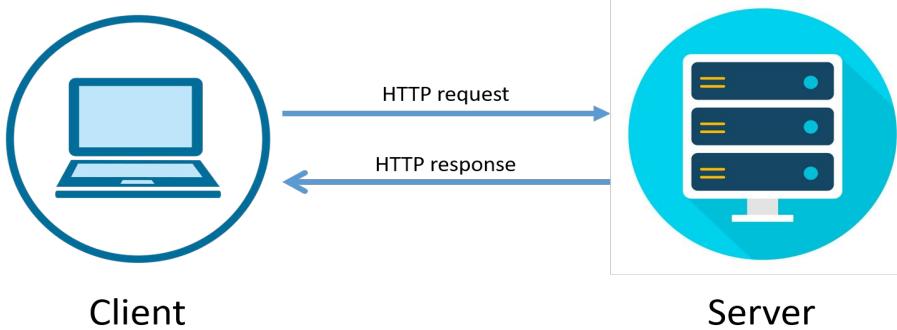
Persistence and Integration Testing

Deploy to AWS with CDK

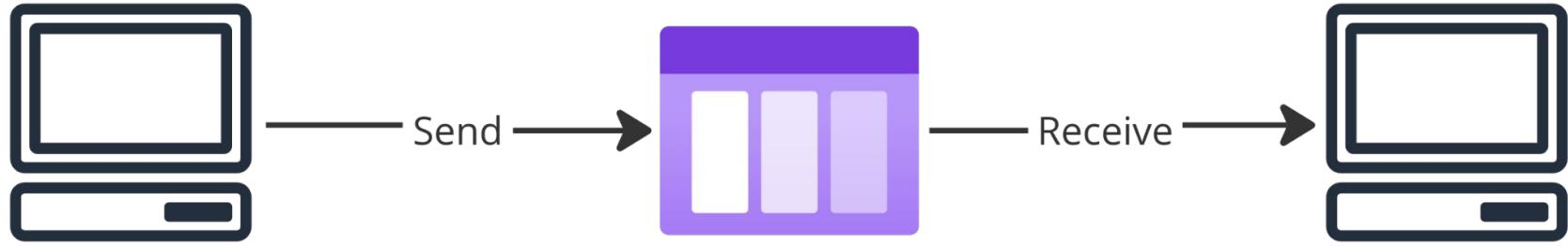
Request - Response Web Servers



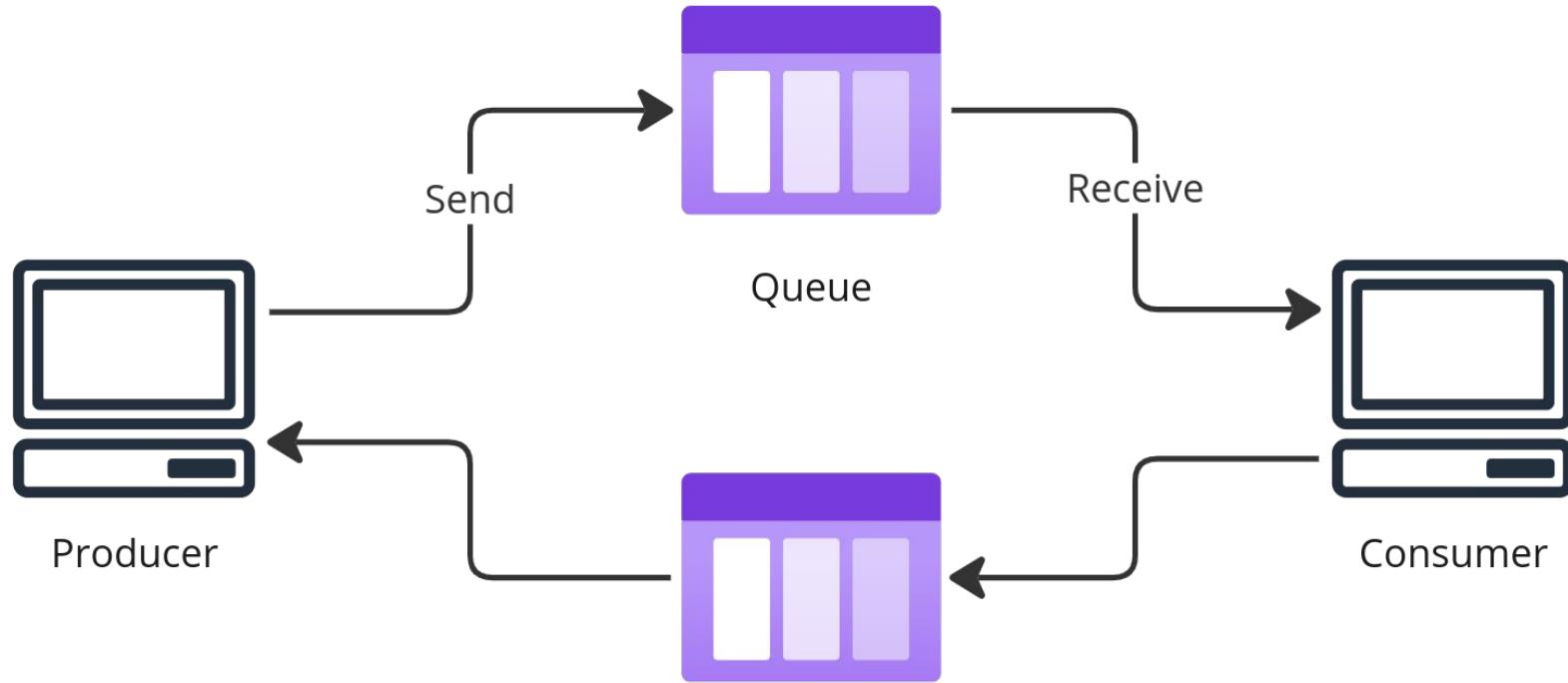
Request - Response Web Servers



Introducing Queues

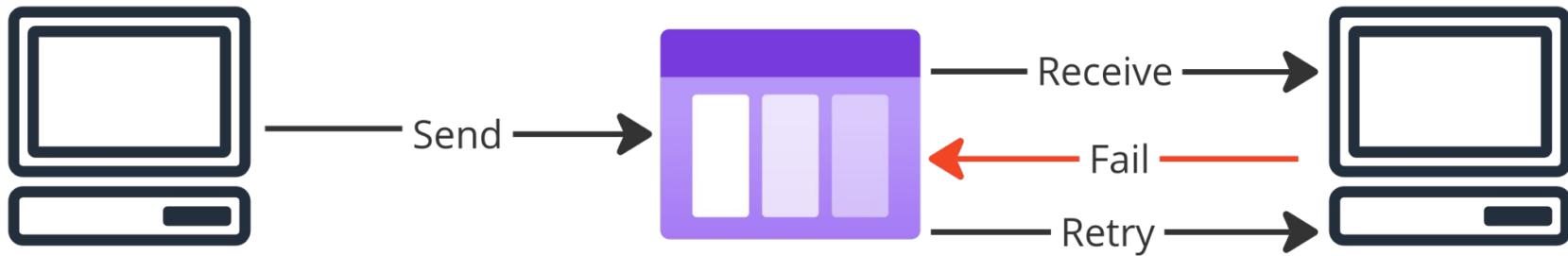


Async Request - Response

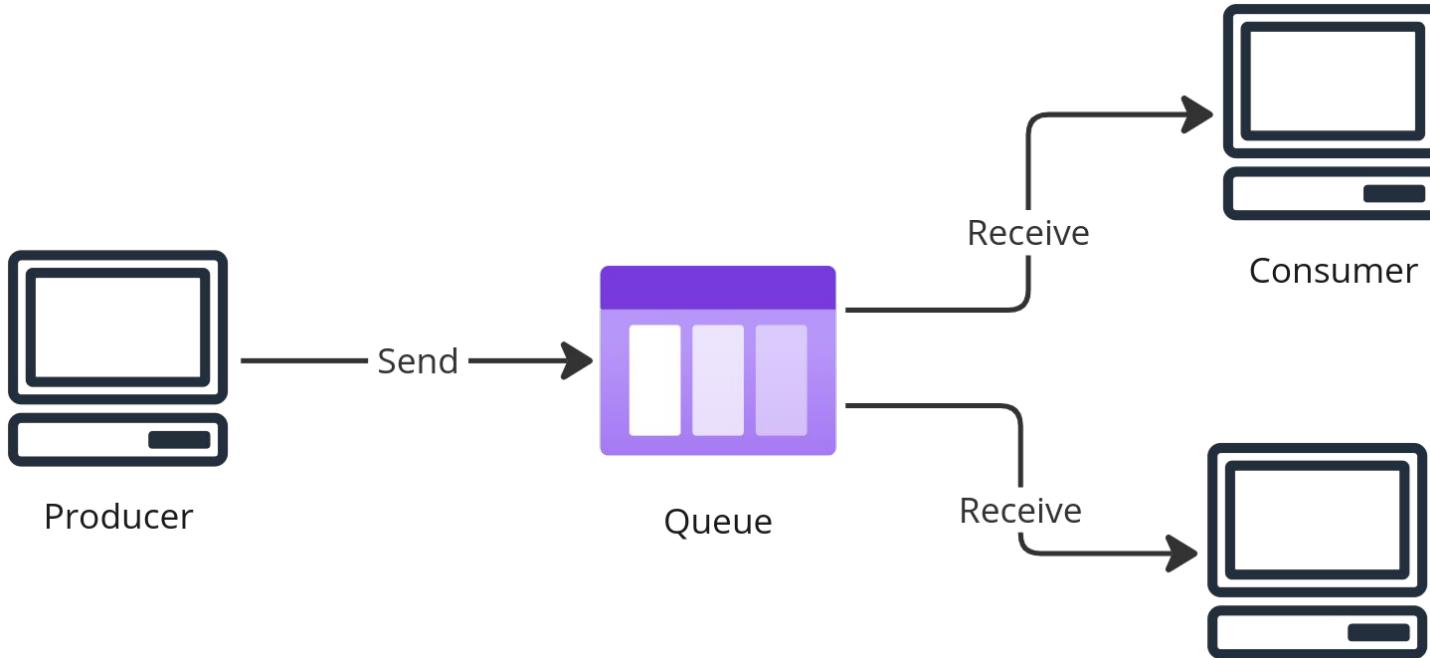


Queues Add Resilience

At-least-once guaranteed delivery

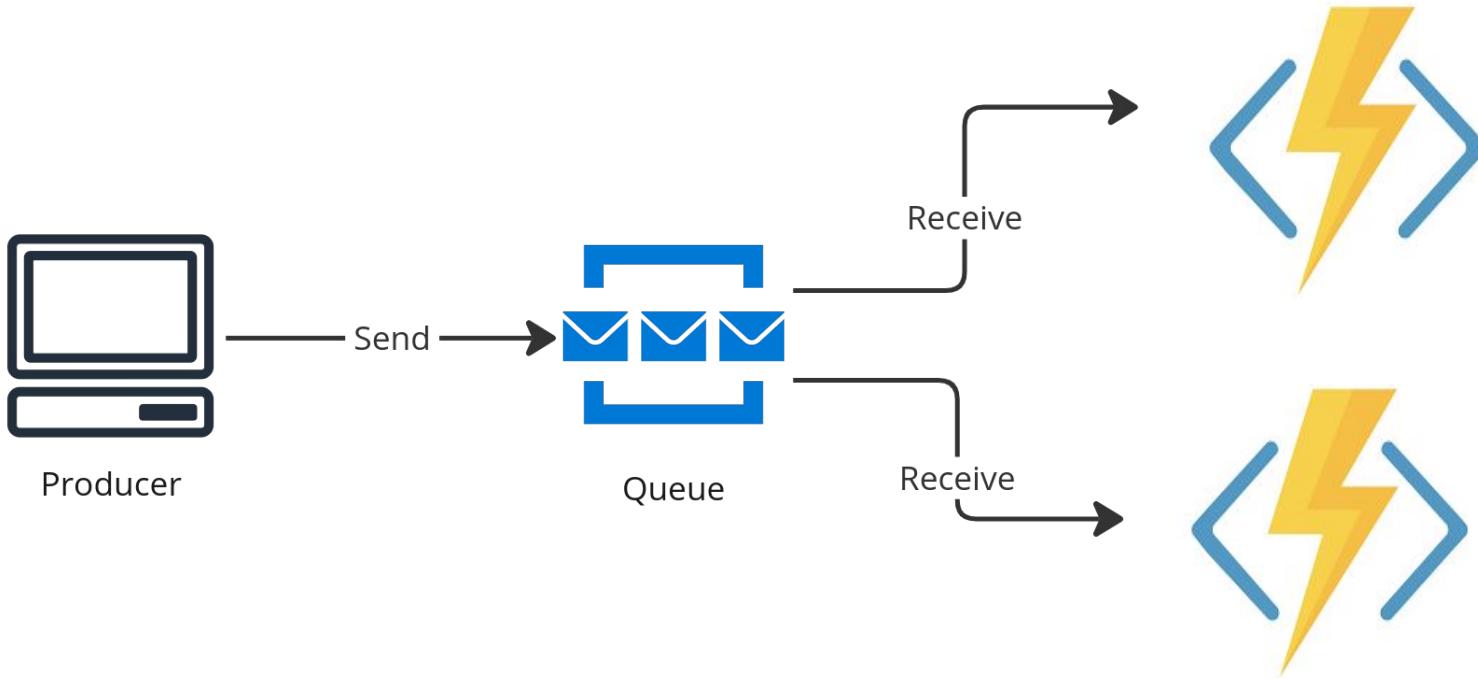


Competing Consumer Pattern



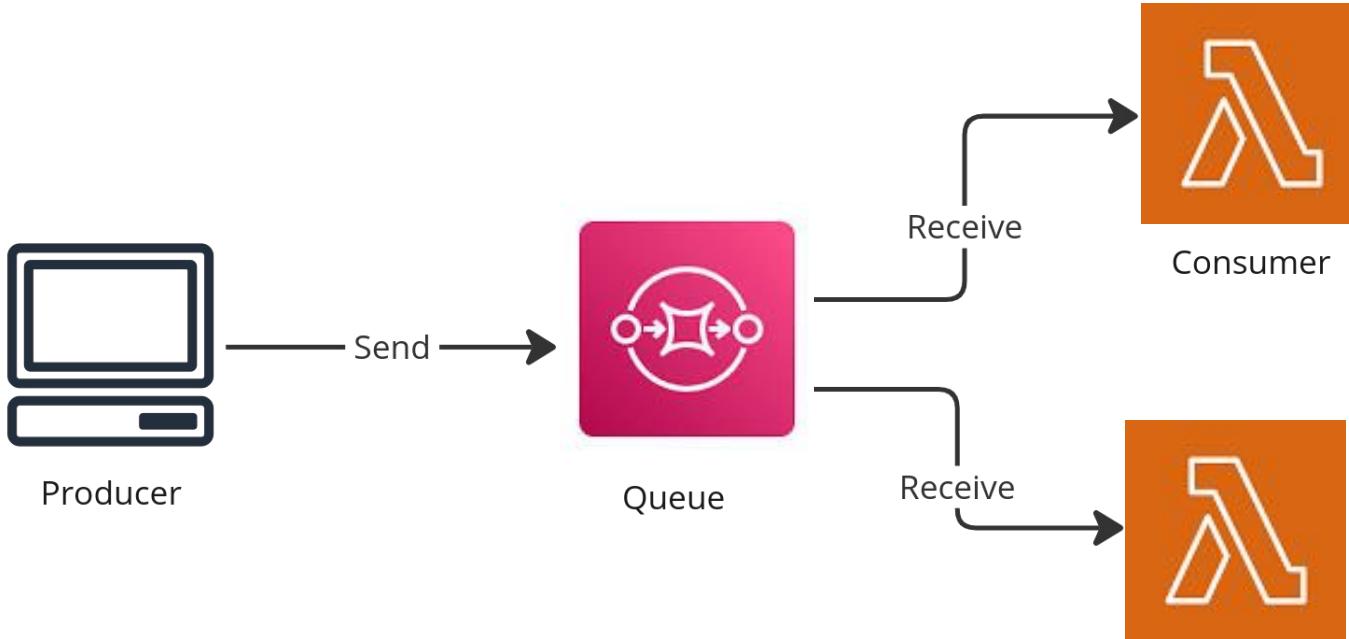
Serverless Compute

Azure Service Bus + Function Apps

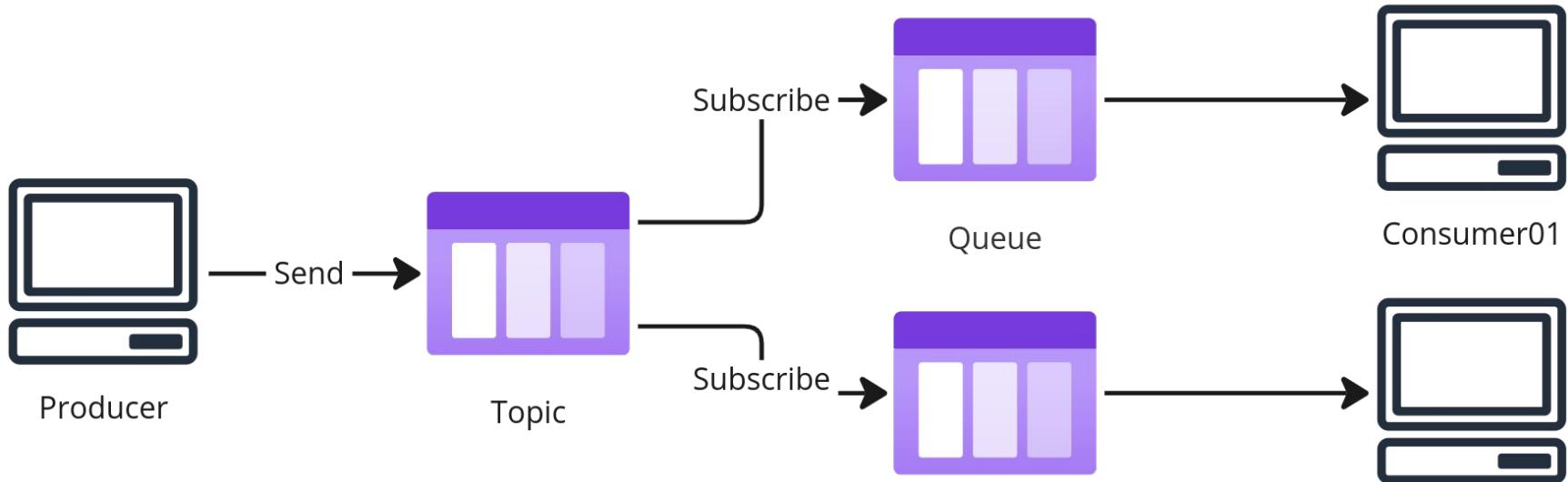


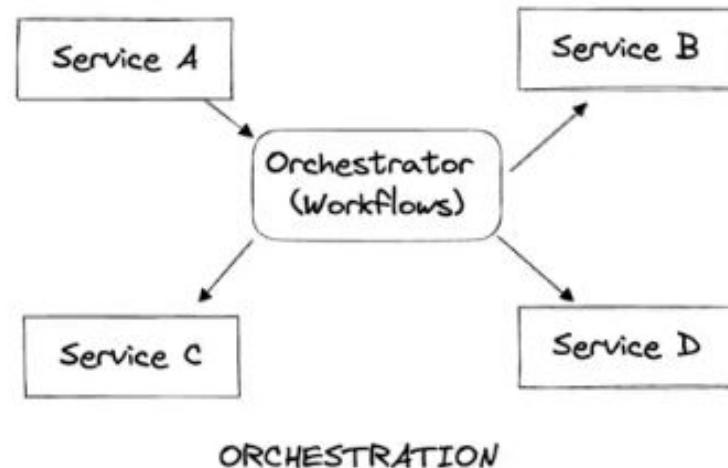
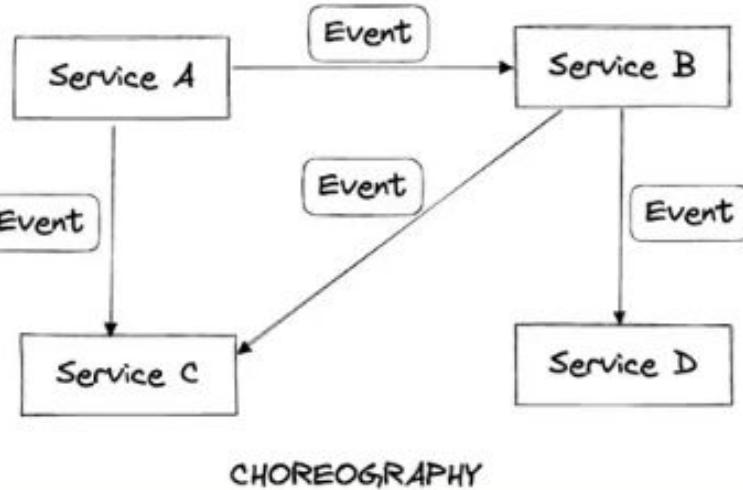
Serverless Compute

AWS SQS + Lambda



Topics or Pub/Sub





Choreography Vs Orchestration

Choreography

- Loosely Coupled
- Stateless
- Flexible
- Difficult Observability

Orchestration

- Tightly Coupled
- Stateful
- Transactional
- Easier Observability

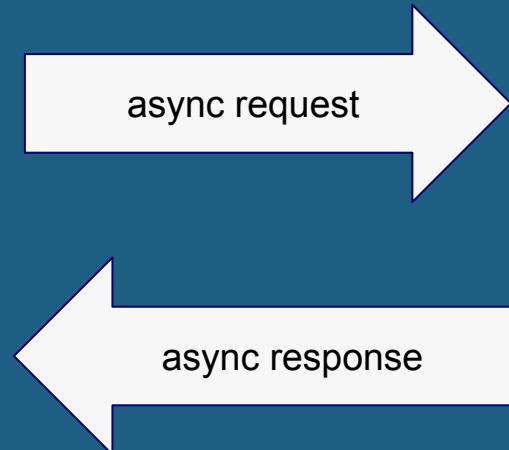
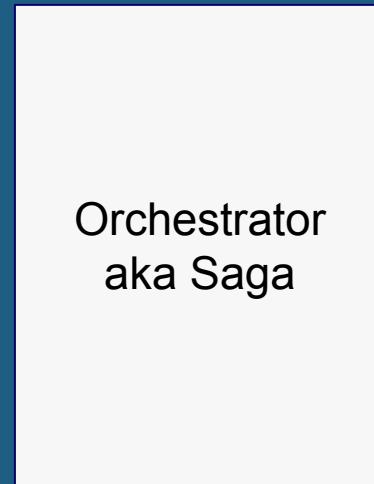
Choreography Vs Orchestration

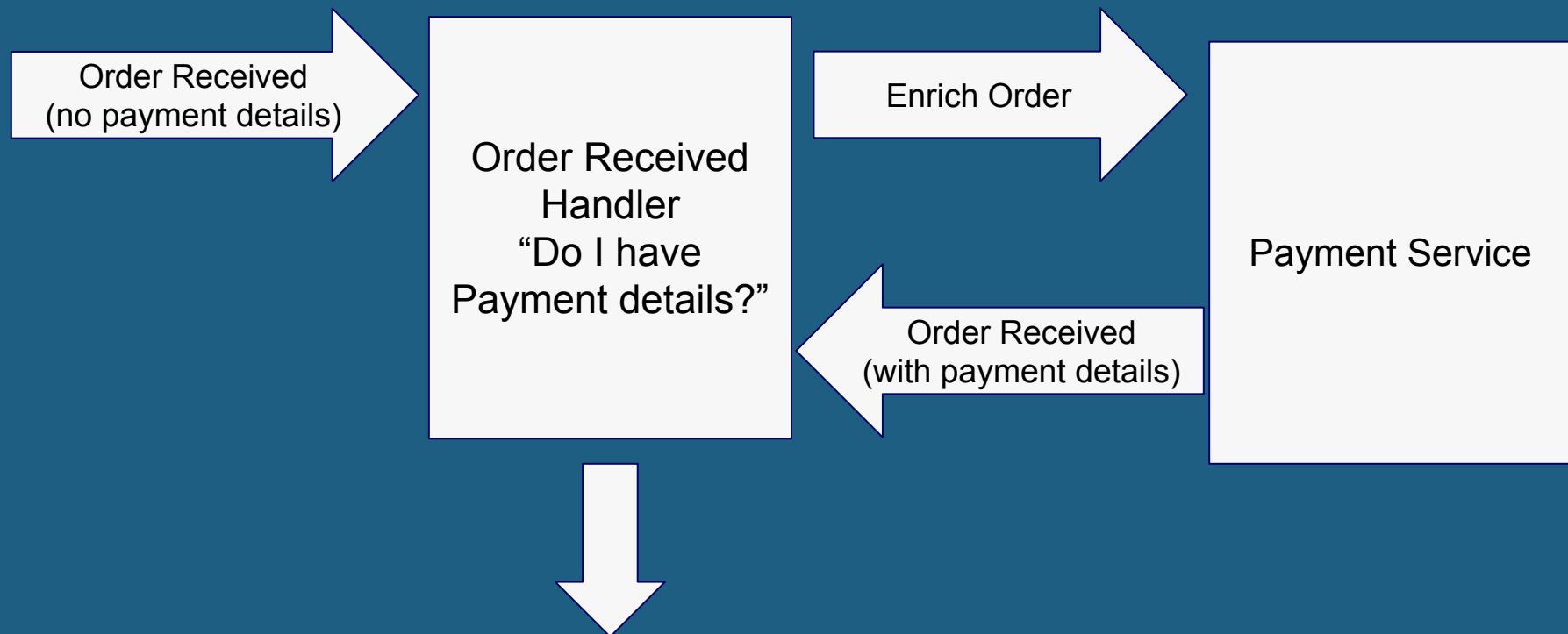
Choreography

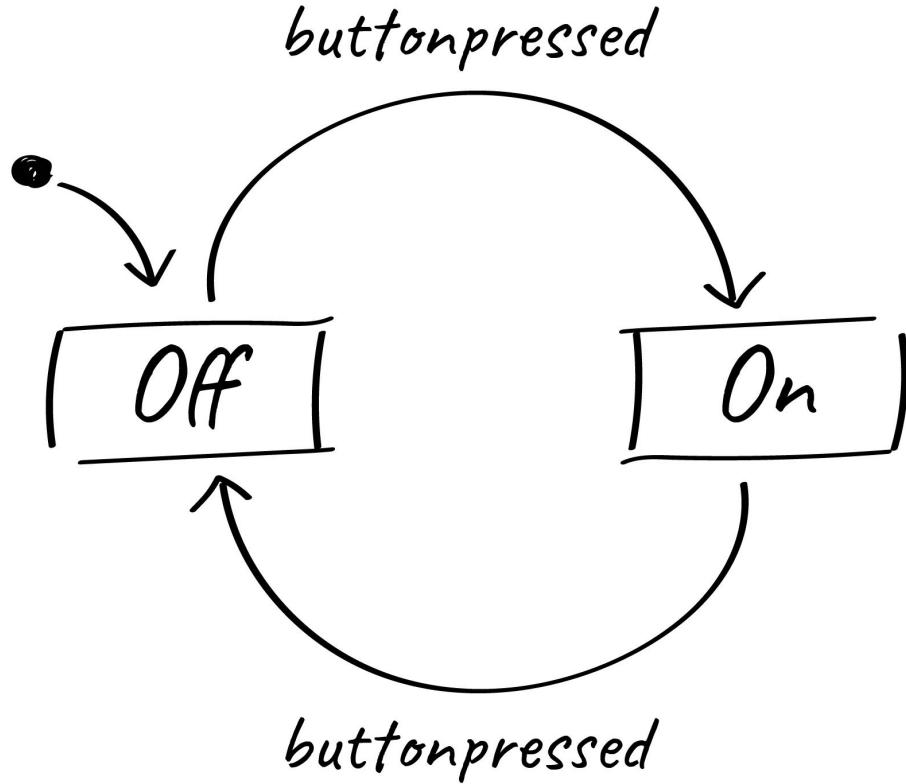
- Loosely Coupled
- Stateless
- Flexible
- Difficult Observability

Orchestration

- Tightly Coupled
- Stateful
- Transactional
- Easier Observability







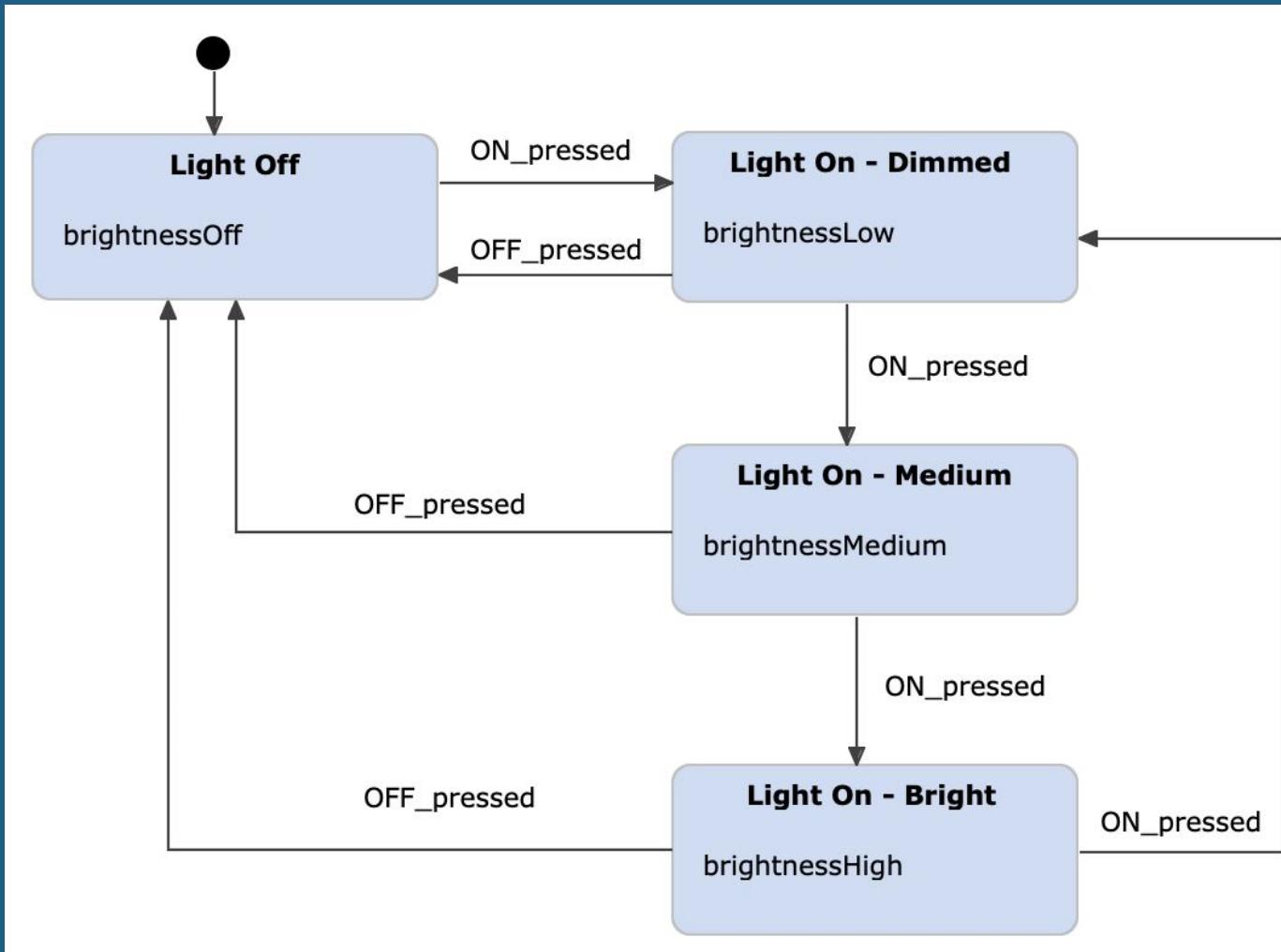


Table Of Contents

Event Driven Concepts

Introducing our Use Case

Mass Transit, Sagas and State Machines

Mass Transit Testing

Persistence and Integration Testing

Deploy to AWS with CDK

A non-trivial use case

Obviously an online trivia game!

- Multiple users
- Multiple Choice Trivia Questions
- The Trivia API
- Assign points based on speed
- Stateful
- Workflow
- Events from multiple sources

```
{  
  "category": "society_and_culture",  
  "id": "622a1c357cc59eab6f94fd14",  
  "correctAnswer": "Karl Marx",  
  "incorrectAnswers": [  
    "Ludwig Wittgenstein",  
    "Socrates",  
    "Friedrich Nietzsche"  
,  
  "question": {  
    "text": "Which philosopher famously said 'Religion is the  
},  
  "tags": [  
    "philosophy",  
    "quotes",  
    "society_and_culture"  
,  
  "type": "text_choice",
```

Trivia Walkthrough

State: LOBBY OPEN

Create Game

Game Data

- CorrelationID
- Name

Trivia Walkthrough



Add Participant

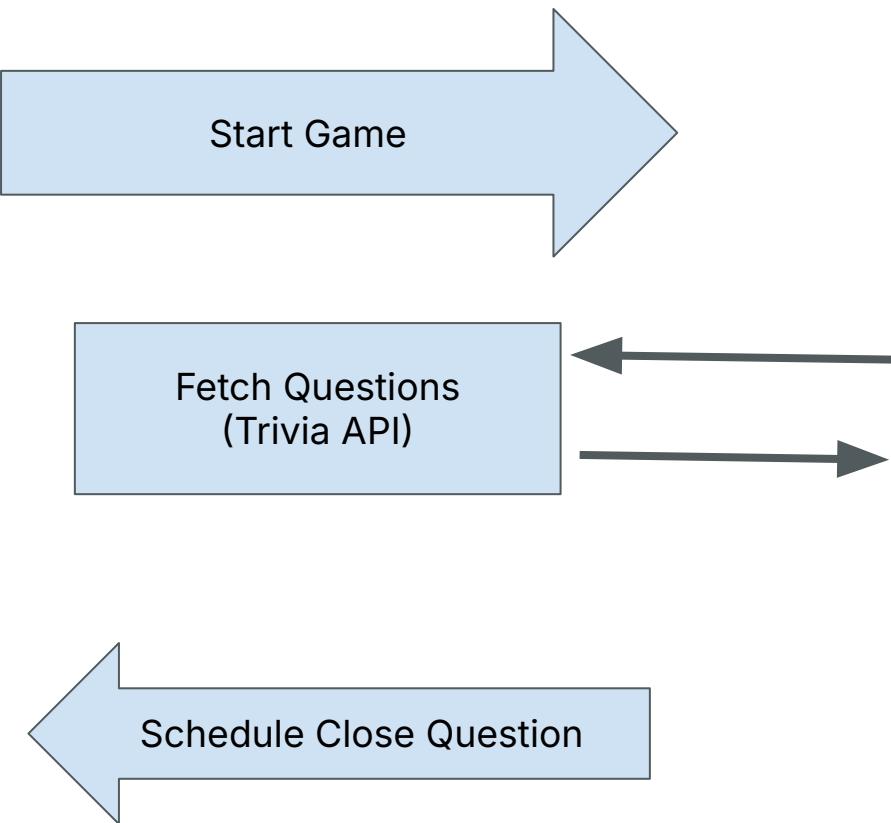
State: LOBBY OPEN

Game Data

- CorrelationID
- Name
- Participants[]

Trivia Walkthrough

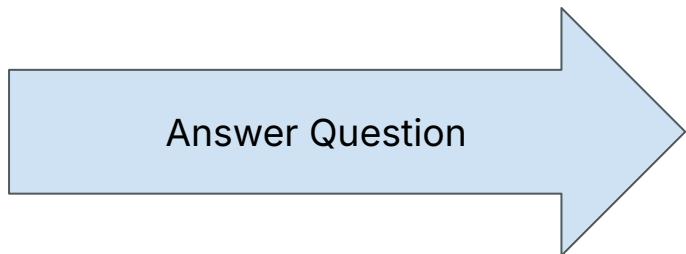
State: QUESTION OPEN



Game Data

- CorrelationID
- Name
- Participants[]
- Questions[]
- CurrentQuestionNo

Trivia Walkthrough



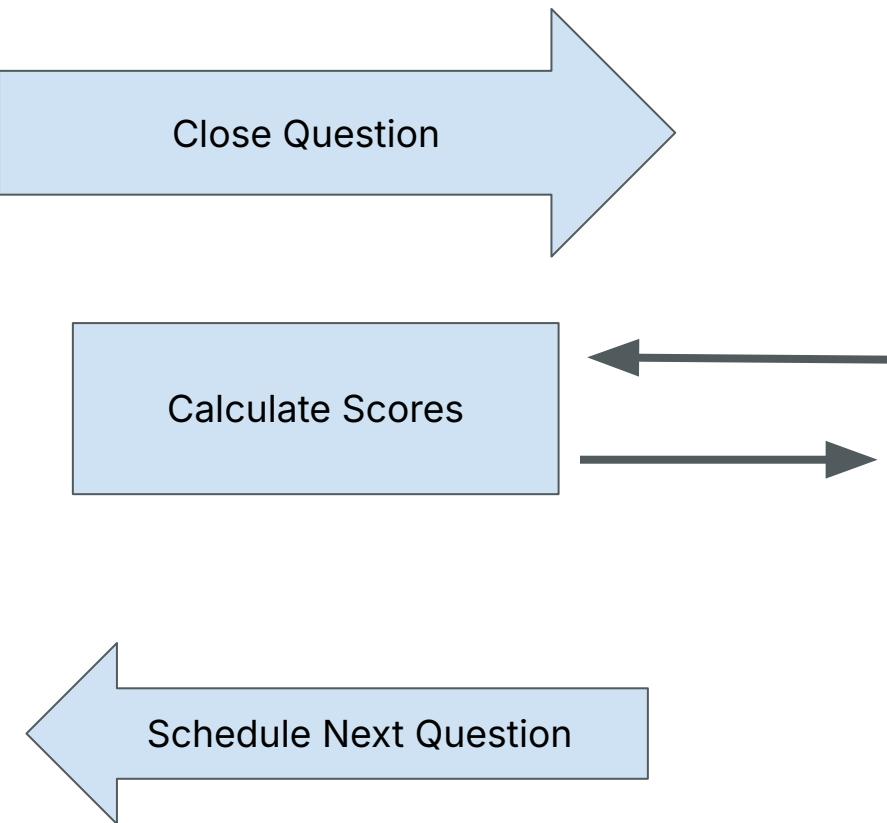
State: QUESTION OPEN

Game Data

- CorrelationID
- Name
- Participants[]
- Questions[]
- CurrentQuestionNo
- ParticipantAnswers[]

Trivia Walkthrough

State: QUESTION RESULT



Game Data

- CorrelationID
- Name
- Participants[]
- Questions[]
- CurrentQuestionNo
- ParticipantAnswers[]
- ParticipantScores[]

Trivia Walkthrough

State: QUESTION OPEN

Next Question

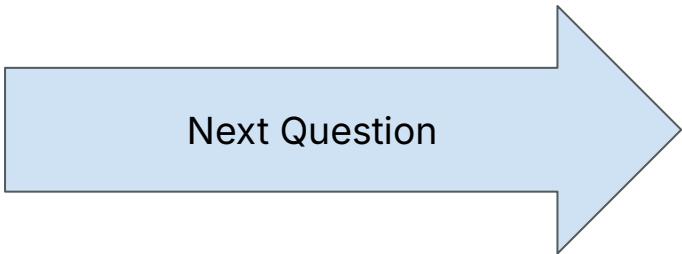
Game Data

- CorrelationID
- Name
- Participants[]
- Questions[]
- CurrentQuestionNo
- ParticipantAnswers[]
- ParticipantScores[]

Schedule Close Question

Trivia Walkthrough

State: FINAL



Next Question

Game Data

- CorrelationID
- Name
- Participants[]
- Questions[]
- CurrentQuestionNo
- ParticipantAnswers[]
- ParticipantScores[]

Table Of Contents

Event Driven Concepts

Introducing our Use Case

Mass Transit, Sagas and State Machines

Mass Transit Testing

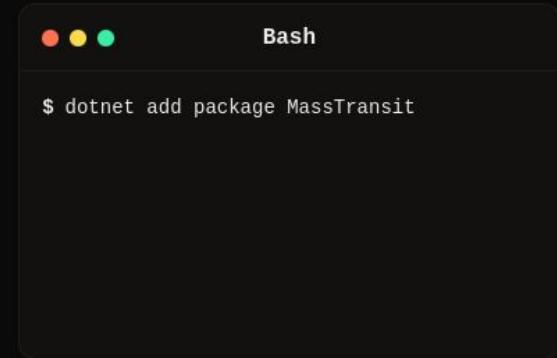
Persistence and Integration Testing

Deploy to AWS with CDK

Easily build reliable distributed applications

MassTransit provides a developer-focused, modern platform for creating distributed applications without complexity.

-  First class testing support
-  Write once, then deploy using RabbitMQ, Azure Service Bus, and Amazon SQS
-  Observability via Open Telemetry (OTEL)
-  Fully-supported, widely-adopted, a complete end-to-end solution

[Learn more](#)[Support →](#)

A terminal window titled "Bash" with three colored dots (red, yellow, green) at the top. It contains the command: \$ dotnet add package MassTransit



```
// Add services to the container.
builder.Services.AddHttpClient();
builder.Services.AddMassTransit(x :IBusRegistrationConfigurator =>
{
    x.SetKebabCaseEndpointNameFormatter();

    x.UsingAmazonSqs(configure: (context, cfg) =>
    {
        cfg.Host("ap-southeast-2", configure: h :IAmazonSqsHostConfigurator =>
        {
            h.AccessKey(builder.Configuration["user-access-key"]);
            h.SecretKey(builder.Configuration["user-secret"]);
            h.Scope("brendan-trivia", scopeTopics: true);
        });
        cfg.ConfigureEndpoints(context);
    });
});
```

```
// Add services to the container.
builder.Services.AddHttpClient();
builder.Services.AddMassTransit(x :IBusRegistrationConfigurator =>
{
    x.SetKebabCaseEndpointNameFormatter();

    x.UsingAmazonSqs(configure: (context, cfg) =>
    {
        cfg.Host("ap-southeast-2", configure: h :IAmazonSqsHostConfigurator =>
        {
            h.AccessKey(builder.Configuration["user-access-key"]);
            h.SecretKey(builder.Configuration["user-secret"]);
            h.Scope("brendan-trivia", scopeTopics: true);
        });
        cfg.ConfigureEndpoints(context);
    });
});
```

```
// Add services to the container.
builder.Services.AddHttpClient();
builder.Services.AddMassTransit(x :IBusRegistrationConfigurator =>
{
    x.SetKebabCaseEndpointNameFormatter();

    x.UsingAmazonSqs(configure: (context, cfg) =>
    {
        cfg.Host("ap-southeast-2", configure: h :IAmazonSqsHostConfigurator =>
        {
            h.AccessKey(builder.Configuration["user-access-key"]);
            h.SecretKey(builder.Configuration["user-secret"]);
            h.Scope("brendan-trivia", scopeTopics: true);
        });
        cfg.ConfigureEndpoints(context);
    });
});
```

```
// Add services to the container.
builder.Services.AddHttpClient();
builder.Services.AddMassTransit(x :IBusRegistrationConfigurator =>
{
    x.SetKebabCaseEndpointNameFormatter();

    x.UsingAmazonSqs(configure: context, cfg =>
    {
        cfg.Host("ap-southeast-2", configure: h :IAmazonSqsHostConfigurator =>
        {
            h.AccessKey(builder.Configuration["user-access-key"]);
            h.SecretKey(builder.Configuration["user-secret"]);
            h.Scope("brendan-trivia", scopeTopics: true);
        });
        cfg.ConfigureEndpoints(context);
    });
});
```

Producers - Send

```
public async Task SendOrder(ISendEndpointProvider sendEndpointProvider)  
{  
    var endpoint = await sendEndpointProvider.GetSendEndpoint("queue:submit-order");  
    await endpoint.Send(new SubmitOrder { OrderId = "123" });  
}
```

Producers - Publish

```
await _publishEndpoint.Publish(message: new CreateGame()  
{  
    Name = name,  
    CorrelationId = id,  
    CreatedAt = DateTime.UtcNow  
}); // Task
```

Producers - Schedule

```
var scheduler = scope.ServiceProvider.GetRequiredService<IMessageScheduler>();  
|  
| await scheduler.SchedulePublish<SendNotification>(  
|     DateTime.UtcNow + TimeSpan.FromSeconds(30), new()  
| {  
|     EmailAddress = "frank@nul.org",  
|     Body = "Thank you for signing up for our awesome newsletter!"  
|});
```

Producers - Request/Response

```
IRequestClient<CheckOrderStatus> _client;

[HttpGet("{orderId}")]
public async Task<IActionResult> Get(string orderId, CancellationToken cancellationToken)
{
    var response = await _client.GetResponse<OrderStatusResult>(new { orderId }, cancellationToken);
    return Ok(response.Message);
}
```

Consumer

```
class SubmitOrderConsumer :  
    IConsumer<SubmitOrder>  
{  
    public async Task Consume(ConsumeContext<SubmitOrder> context)  
    {  
        await context.Publish<OrderSubmitted>(new  
        {  
            context.Message.OrderId  
        });  
    }  
}
```

Consumer: Fetch Questions

```
public class FetchQuestionsConsumer : IConsumer<FetchQuestions>
{
    private readonly HttpClient _httpClient;
```

```
public async Task Consume(ConsumeContext<FetchQuestions> context)
{
    var response = await _httpClient.GetAsync("https://the-trivia-api.com/v2/questions?li");
    var apiQuestions = JsonSerializer.Deserialize<IList<ApiQuestion>>(await response.Content.Read
    var index = 0;
    await context.RespondAsync(message: new QuestionsFetched()
    {
        CorrelationId = context.Message.CorrelationId,
        Questions = apiQuestions.Select(q: ApiQuestion => new Question()
        {
            Content = q.question,
            Category = q.category,
            Type = q.type
        })
    })
}
```

Consumer Saga: Stateful Class, Consumer

```
public class OrderSaga :  
    ISaga,  
    InitiatedBy<SubmitOrder>  
{  
    public Guid CorrelationId { get; set; }  
  
    public DateTime? SubmitDate { get; set; }  
    public DateTime? AcceptDate { get; set; }  
  
    public async Task Consume(ConsumeContext<SubmitOrder> context)  
    {  
        SubmitDate = context.Message.OrderDate;  
    }  
}
```

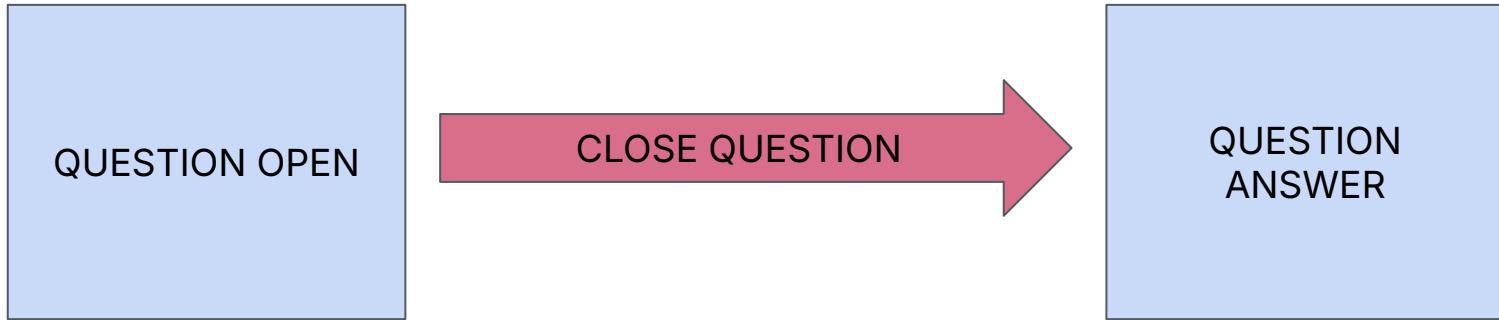
Consumer Saga: Stateful Class, Consumer

```
public class OrderSaga :  
    ISaga,  
    InitiatedBy<SubmitOrder>  
{  
    public Guid CorrelationId { get; set; }  
  
    public DateTime? SubmitDate { get; set; }  
    public DateTime? AcceptDate { get; set; }  
  
    public async Task Consume(ConsumeContext<SubmitOrder> context)  
    {  
        SubmitDate = context.Message.OrderDate;  
    }  
}
```

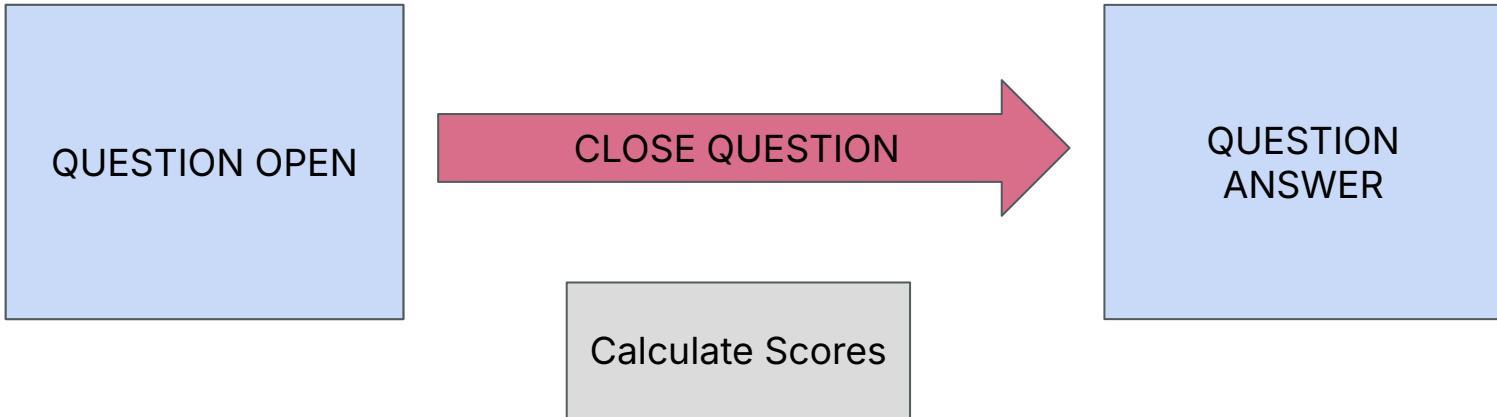
Consumer Saga: Stateful Class, Consumer

```
public class OrderSaga :  
    ISaga,  
    InitiatedBy<SubmitOrder>  
{  
    public Guid CorrelationId { get; set; }  
  
    public DateTime? SubmitDate { get; set; }  
    public DateTime? AcceptDate { get; set; }  
  
    public async Task Consume(ConsumeContext<SubmitOrder> context)  
    {  
        SubmitDate = context.Message.OrderDate;  
    }  
}
```

State Machines



State Machines



State Machine Saga

```
public class GameState: SagaStateMachineInstance
{
        ↗ 4 usages
    public Guid CorrelationId { get; set; }

        ↗ 13 usages
    public string? CurrentState { get; set; }

        ↗ 19 usages
    public int? CurrentQuestionIndex { get; set; } = null;

        ↗ 12 usages
    public IList<Participant> Participants { get; set; } = new List<Participant>();

        ↗ 13 usages
    public IList<Question> Questions { get; set; } = new List<Question>();
}
```

State Machine Saga

```
public class GameStateMachine : MassTransitStateMachine<GameState>
{
    public GameStateMachine()
    {
        InstanceState(x:GameState => x.CurrentState);
    }
}
```

```
Initially(params activities: When(CreateGame) // EventActivity)
    .TransitionTo(LobbyOpen));
```

Concise DSL for State Transitions

```
During(LobbyOpen,|  
    When(AddParticipant)  
        .Then(ctx => ctx.Saga.Participants.Add(ctx.Message.Participant!)),  
    When(StartGame)  
        .Request(FetchQuestionsRequest, ctx =>  
            new FetchQuestions(){ CorrelationId = ctx.CorrelationId!.Value})  
        .TransitionTo(FetchQuestionsRequest!.Pending)  
);
```

Concise DSL for State Transitions

```
During(LobbyOpen |  
    When(AddParticipant)  
        .Then(ctx => ctx.Saga.Add(ctx.Message.Participant!)),  
    When(StartGame)  
        .Request(FetchQuestionsRequest, ctx =>  
            new FetchQuestions(){ CorrelationId = ctx.CorrelationId!.Value})  
        .TransitionTo(FetchQuestionsRequest!.Pending)  
);
```

State

Concise DSL for State Transitions

```
During(LobbyOpen,
    When(AddParticipant)
        .Then(ctx => ctx.Saga
    When(StartGame)
        .Request(FetchQuestionsRequest, ctx =>
            new FetchQuestions(){ CorrelationId = ctx.CorrelationId!.Value})
        .TransitionTo(FetchQuestionsRequest!.Pending)
);

```

Event

Concise DSL for State Transitions

```
During(LobbyOpen,
    When(AddParticipant)
        .Then(ctx => ctx.Saga.Participants.Add(ctx.Message.Participant!)),
    When(StartGame)
        .Request(FetchQuestions)
        .Then(ctx =>
            new FetchQuestion { CorrelationId = ctx.CorrelationId!.Value })
        .TransitionTo(FetchQuestionsRequest!.Pending)
);
```

.Then
Do something locally

Concise DSL for State Transitions

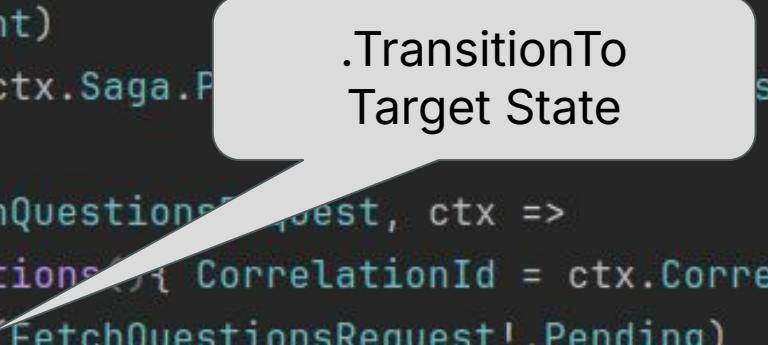
```
During(LobbyOpen,
    When(AddParticipant)
        .Then(ctx => ctx.Saga.Participants.Add(
            new Participant{ CorrelationId = ctx.CorrelationId!.Value })
        .Request.Request
        .Raise async req/response
    ),
    When(StartGame)
        .Request.Request(FetchQuestionsRequest, ctx =>
            new FetchQuestions(){ CorrelationId = ctx.CorrelationId!.Value })
        .TransitionTo(FetchQuestionsRequest!.Pending)
);

```

Concise DSL for State Transitions

```
During(LobbyOpen,
    When(AddParticipant)
        .Then(ctx => ctx.Saga.P
    When(StartGame)
        .Request(FetchQuestionsRequest, ctx =>
            new FetchQuestionsRequest{ CorrelationId = ctx.CorrelationId!.Value})
        .TransitionTo(FetchQuestionsRequest!.Pending)
);

```



.TransitionTo
Target State

sage.Participant!)),

Concise DSL for State Transitions

```
When(CloseCurrentQuestion)
    .Then(ctx =>
{
    ctx.Saga.Questions[ctx.Saga.CorrelationId].QuestionClosed = DateTime.UtcNow;
    CalculateScores(ctx.Saga);
})
.Schedule(NextQuestionScheduler,
    ctx => ctx.Init<NextQuestion>(new NextQuestion(){CorrelationId = ctx.Saga.CorrelationId}),
    ctx => TimeSpan.FromSeconds(ctx.Saga.ShowResultTimeSeconds))
.TransitionTo(QuestionResult)
```

.Schedule

Schedule a future event

Declare Items to use in State Machine

4 usages

```
public State LobbyOpen { get; private set; }
```

6 usages

```
public State QuestionOpen { get; private set; }
```

7 usages

```
public State QuestionResult { get; private set; }
```

Declare Items to use in State Machine

☒ 1 usage

```
public Event<CreateGame> CreateGame { get; set; }
```

☒ 1 usage

```
public Event<AddParticipant> AddParticipant { get; set; }
```

☒ 1 usage

```
public Event<StartGame> StartGame { get; set; }
```

☒ 6 usages

```
public Request<GameState, FetchQuestions, QuestionsFetched> FetchQuestionsRequest {
```

☒ 2 usages

```
public Schedule<GameState, CloseCurrentQuestion> CloseCurrentQuestionScheduler { get;
```

...

Container Config - Assembly Scan

```
.AddMassTransit(x =>
{
    var mtAssembly = (typeof(CreateGame).Assembly);
    x.AddConsumers(mtAssembly);
    x.AddSagaStateMachines(mtAssembly);
    x.AddSagas(mtAssembly);
    x.AddActivities(mtAssembly);
```

Table Of Contents

Event Driven Concepts

Introducing our Use Case

Mass Transit, Sagas and State Machines

Mass Transit Testing

Persistence and Integration Testing

Deploy to AWS with CDK

In Memory Providers

```
x.SetInMemorySagaRepository();
x.UsingInMemory((context, cfg) =>
{
    cfg.ConfigureEndpoints(context);
});
```

Mass Transit Test Harness

[Fact]

By Brendan Richards

```
public async Task ShouldFetchQuestions()
{
    await using var provider = new ServiceCollection()
        .AddHttpClient()
        .AddMassTransitTestHarness(x =>
    {
        x.AddConsumer<FetchQuestionsConsumer>();
    })
    .BuildServiceProvider(true);

    var harness = provider.GetRequiredService<ITestHarness>();
    await harness.Start();
}
```

Test Harness - retrieve state

```
var sagaHarness = harness.GetSagaStateMachineHarness<GameStateMachine, GameState>();
```

```
await harness.Bus.Publish(new CreateGame())
{
    CorrelationId = sagaId
});
(await harness.Consumed.Any<CreateGame>()).ShouldBeTrue();
```

```
var instance =
    sagaHarness.Sagas.ContainsInState(sagaId, sagaHarness.StateMachine,
        sagaHarness.StateMachine.LobbyOpen);

instance.ShouldNotBeNull();
instance.CurrentState.ShouldBe("LobbyOpen");
```

Saga - Custom Initial State

```
harness.AddSagaInstance<GameState>(sagaId, s =>
{
    s.CurrentState = "QuestionOpen";
    s.Participants = Participants;
    s.Questions = MockFetchQuestionsQuestionsConsumer.Questions;
    s.CurrentQuestionIndex = 0;
});
```

Test our whole game engine

```
✓ ? () Sagas (12 tests) Success
  ✓ GameSagaTests (10 tests) Success
    ✓ CreateGameShouldLobbyOpen Success
    ✓ GivenFetchQuestions_WhenQuestionsFetched_QuestionOpen_AndCloseQuestionScheduled Success
    ✓ GivenLastQuestionOpen_WhenCloseCurrentQuestion_QuestionResult_AndNextQuestionScheduled_AndFinal Success
    ✓ GivenLobbyOpen_WhenAddParticipant_Success Success
    ✓ GivenLobbyOpen_WhenGetParticipantState_NoCurrentQuestion Success
    ✓ GivenLobbyOpen_WhenStartGame_FetchQuestionsPending Success
    ✓ GivenQuestionOpen_WhenCloseCurrentQuestion_QuestionResult_AndNextQuestionScheduled Success
    ✓ GivenQuestionOpen_WhenCloseCurrentQuestion_ScoresApplied Success
    ? GivenQuestionOpen_WhenQuestionAnswered_Success Success
    ✓ GivenQuestionOpenWithIncorrectAnswers_WhenCloseCurrentQuestion_ScoresApplied Success
```

Table Of Contents

Event Driven Concepts

Introducing our Use Case

Mass Transit, Sagas and State Machines

Mass Transit Testing

Persistence and Integration Testing

Deploy to AWS with CDK



Marten

.NET Transactional Document DB and Event Store on PostgreSQL

Pragmatic library to boost your developer
productivity!



Configure Marten

```
var services = new ServiceCollection()
    .AddHttpClient();
services.AddMarten(opt => opt.Connection(_postgreSqlContainer.GetCon
services.AddMassTransit(x =>
{
    x.SetMartenSagaRepositoryProvider();
    x.AddSagaRepository<GameState>()
        .MartenRepository();
})
```

Welcome to Testcontainers for .NET!

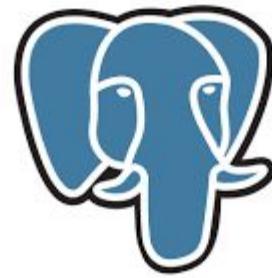


Install the NuGet dependency

```
dotnet add package Testcontainers
```

Run the Hello World container

```
// Create a new instance of a container.  
var container = new ContainerBuilder()  
    // Set the image for the container to "testcontainers/helloworld:1.1.0".  
    .WithImage("testcontainers/helloworld:1.1.0")  
    // Bind port 8080 of the container to a random port on the host.  
    .WithPortBinding(8080, true)  
    // Wait until the HTTP endpoint of the container is available.  
    .WithWaitStrategy(Wait.ForUnixContainer().UntilHttpRequestIsSucceeded(r => r.ForI  
    // Build the container configuration.  
    .Build();
```



Postgre^{SQL}

About

Testcontainers for .NET is a library to support tests with throwaway instances of Docker containers for all compatible .NET Standard versions. The library is built on top of the .NET Docker remote API and provides a lightweight implementation to support your test environment in all circumstances.

Choose from existing pre-configured modules and start containers within a second, to support and run your tests. Or create your own container images with Dockerfiles and run your containers and tests immediately afterward.

PosgreSqlContainer

```
using Testcontainers.PostgreSQL;
✉ Brendan Richards *
public class MartenPersistenceForGameStateTests: IAsyncLifetime
{
    private readonly PostgreSQLContainer _postgreSQLContainer = new PostgreSQLBuilder().Build();
    ✉ Brendan Richards
    public Task InitializeAsync()
    {
        return _postgreSQLContainer.StartAsync();
    }

    ✉ Brendan Richards
    public Task DisposeAsync()
    {
        return _postgreSQLContainer.DisposeAsync().AsTask();
    }
}
```

Arrange

[Fact]

• Brendan Richards *

```
public async Task ShouldPersistGameState()
```

```
{}
```

```
    var services = new ServiceCollection()
        .AddHttpClient();
    services.AddMarten(opt => opt.Connection(_postgreSqlContainer.GetConnectionString()));
    services.AddMassTransit(x =>
    {
        x.SetMartenSagaRepositoryProvider();
        x.AddSagaRepository<GameState>()
            .MartenRepository();
```

Act

```
var busControl = provider.GetRequiredService<IBusControl>();
await busControl.StartAsync();
var bus = provider.GetRequiredService<IBus>();

await bus.Publish(new CreateGame() { CorrelationId = Guid.NewGuid() });
await busControl.StopAsync();
```

Assert

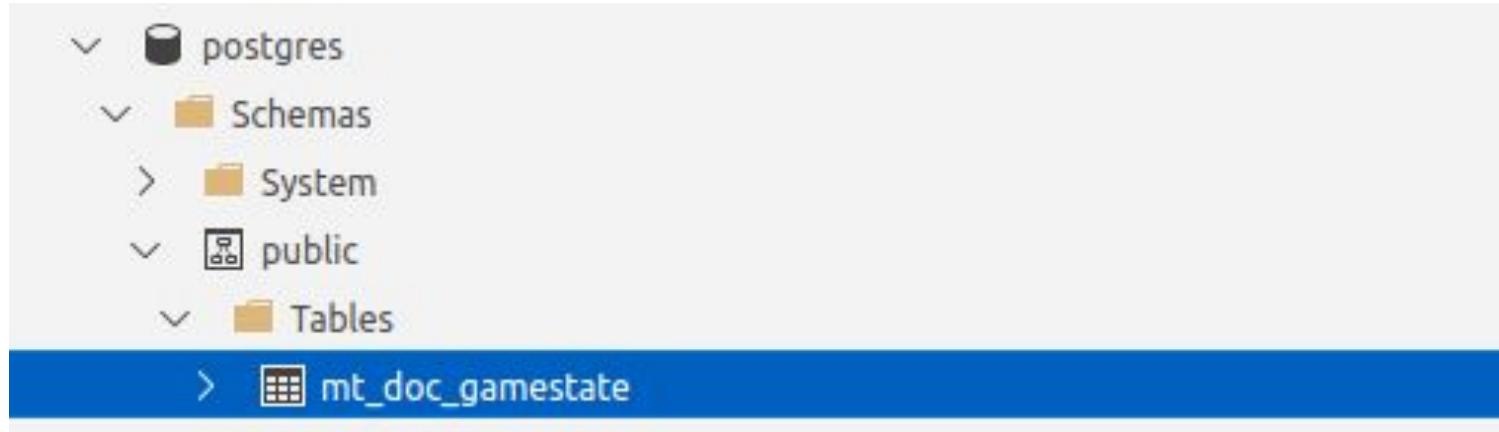
```
// assert record created
await using DbConnection connection = new NpgsqlConnection(_postgreSqlContainer.GetConnectionString("Default"));
await using DbCommand command = new NpgsqlCommand();
await connection.OpenAsync();
command.Connection = connection;
command.CommandText = "SELECT COUNT(id) from mt_doc_gamestate";
var count = await command.ExecuteScalarAsync();
count.ShouldBe(1);
```

Test Containers

The screenshot shows a sidebar with icons for Docker (two overlapping squares), CONTAINERS (a ship), and a magnifying glass. The main area has a header 'DOCKER' and a '...' button. Below it, 'CONTAINERS' is expanded, showing 'Individual Containers' which contains two entries: 'postgres:15.1 romantic_agnesi - Up About a minute' and 'testcontainers/ryuk:0.5.1 testcontainers-ryuk-ee9466fa-7766-42...'. The 'postgres' container is green, indicating it is running.

- DOCKER
- ...
- CONTAINERS
 - Individual Containers
 - postgres:15.1 romantic_agnesi - Up About a minute
 - testcontainers/ryuk:0.5.1 testcontainers-ryuk-ee9466fa-7766-42...

Postgres DB - Azure Data Studio



Postgres DB - Azure Data Studio

Run Cancel

Disconnect Change Connection

postgres



```
1 select id
2 ,data
3 ,mt_last_modified
4 ,mt_version
5 ,mt_dotnet_type
6 from public.mt_doc_gamestate
7 LIMIT 1000
```

Results Messages

	id	data	mt_last_modified	mt_version	mt_dotnet_type
1	950ecc38-060c-4534-a95a-aa...	{"Scores": [], "Questions": ...}	2023-11-30T09:35:09.826149...	018c1f95-07d0-430f-a415-64...	MttApplication.Sagas.GameS...

Persisted Gamestate Json

```
{  
  "Scores": [],  
  "Questions": [],  
  "Responses": [],  
  "CurrentState": "LobbyOpen",  
  "Participants": [],  
  "CorrelationId": "950ecc38-060c-4534-a95a-aa48995c14ee",  
  "AnswerTimeSeconds": 10,  
  "HasAnotherQuestion": false,  
  "CurrentQuestionIndex": null,  
  "ShowResultTimeSeconds": 5,  
  "NextQuestionSchedulerTokenId": null,  
  "CloseCurrentQuestionSchedulerTokenId": null  
}
```

Mass Transit Recap

- Library for working with async messaging
- Consumers
- Send, Publish, Schedule, Request
- Sagas
 - Stateful - Correlation ID
 - Orchestrate multiple consumers
- State Machine Sagas
 - Add State Machine workflow syntax
- Saga Caveats
 - Load and save full state with every message handled
 - Contention on parallel executions
 - Add Participant and Question Response

Table Of Contents

Event Driven Concepts

Introducing our Use Case

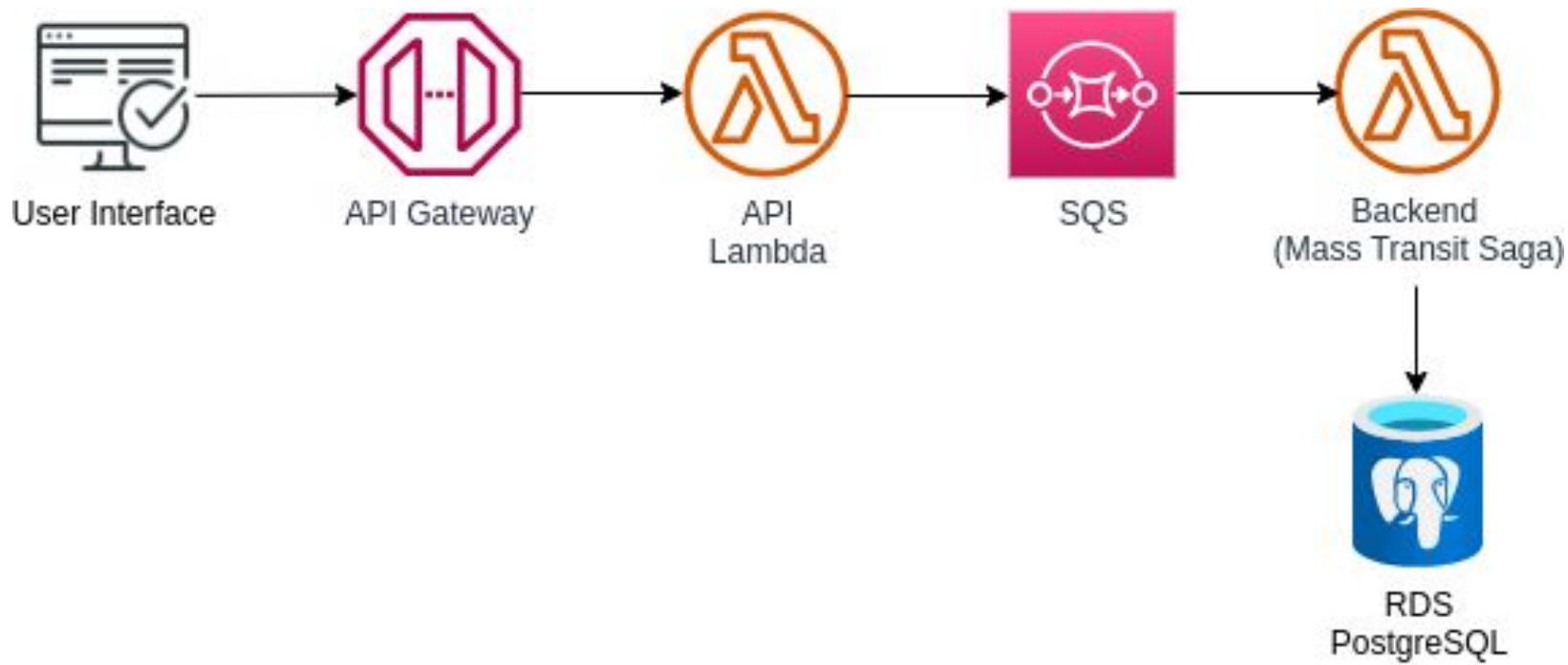
Mass Transit, Sagas and State Machines

Mass Transit Testing

Persistence and Integration Testing

Deploy to AWS with CDK

Architecture



Servers

/prod



Game



GET /Game/create



Parameters

[Cancel](#)

Name	Description
------	-------------

name <small>string</small> <small>(query)</small>	<input type="text" value="my test game"/>
---	---

[Execute](#)

Responses

Code	Description	Links
------	-------------	-------

200	Success	No links
-----	---------	--------------------------

Asp.NET Core Web Application

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

if (builder.Environment.IsDevelopment())
{
    builder.Configuration.AddUserSecrets<Program>();
}
```

```
var app = builder.Build();
```

```
app.Run();
```

AddAWSLambdaHosting

```
builder.Services.AddAWSLambdaHosting(LambdaEventSource.RestApi);

var app = builder.Build();
```

Configure Services

```
builder.Services.AddMassTransit(x =>
{
    x.SetKebabCaseEndpointNameFormatter();
    x.UsingAmazonSqs((context, cfg) =>
    {
        cfg.Host("ap-southeast-2", h =>
        {
            h.AccessKey(builder.Configuration["user-access-key"]);
            h.SecretKey(builder.Configuration["user-secret"]);
            h.Scope("brendan-trivia", true);
        });
        cfg.ConfigureEndpoints(context);
    });
});
builder.Services.AddControllers();
builder.Services.AddSwaggerGen();
```

Configure HTTP Pipeline

```
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();

app.MapGet("/", () => "Welcome to running ASP.NET Core Minimal API on AWS Lambda");

app.UseSwagger();
app.UseSwaggerUI();

app.Run();
```

MVC Controller Class

```
[ApiController]
[Route("[controller]")]
[ 2 usages  ↳ Brendan Richards ]
public class GameController
{
    private readonly IPublishEndpoint _publishEndpoint;
    private readonly ILogger<GameController> _log;
[ ↳ Brendan Richards ]
    public GameController(IPublishEndpoint publishEndpoint, ILogger<GameController> log)
    {
        _publishEndpoint = publishEndpoint;
        _log = log;
    }
}
```

Create Game

```
[HttpGet("create")]
👤 Brendan Richards

public async Task<ActionResult<Guid>> CreateGame(
    [FromQuery] string name)
{
    _log.LogInformation("Publish CreateGame");
    var id = Guid.NewGuid();
    await _publishEndpoint.Publish(new CreateGame()
    {
        Name = name,
        CorrelationId = id,
        CreatedAt = DateTime.UtcNow
    });
    return id;
}
```

Infra-as-code Options

- CloudFormation
- Terraform
- CDK - Cloud Development Kit
 - Write code in your language
 - Compile to CF stack and deploy

CDK - .NET Lambda

```
public class InfraStack: Stack
{
    [1 usage  ↵ Brendan Richards *]
    internal InfraStack(Construct scope, string id, IStackProps props = null) : base(scope, id, props)
    {
        var apiLambda = new Function(this, "MttApiLambda", new FunctionProps
        {
            Runtime = Runtime.DOTNET_6,
            MemorySize = 1024,
            LogRetention = RetentionDays.ONE_DAY,
            Handler = "MttApi",
            Code = Code.FromAsset("../", new Amazon.CDK.AWS.S3.Assets.AssetOptions // parent path to incl
        });
    }
}
```

.NET Lambda Bundling Options

```
Bundling = new BundlingOptions()
{
    Image = Runtime.DOTNET_6.BundlingImage,
    User = "root",
    OutputType = BundlingOutput.ARCHIVED,
    WorkingDirectory = "/asset-input/MttApi/src/MttApi",
    Command = new string[]
    {
        "/bin/sh",
        "-c",
        " dotnet tool install -g Amazon.Lambda.Tools" +
        " && dotnet build" +
        " && dotnet lambda package --output-package /asset-output/function.zip"
    }
}
```

better.

Permissions

```
// publish to sns topics prefixed with brendan-trivia
apiLambda.Role?.AddToPrincipalPolicy(new PolicyStatement(new PolicyStatementProps()
{
    Effect = Effect.ALLOW,
    Actions = new []{ "sns:Publish" },
    Resources = new []{ $"arn:aws:sns:{this.Region}:{this.Account}:brendan-trivia*" }
)));
// list all topics under account
apiLambda.Role?.AddToPrincipalPolicy(new PolicyStatement(new PolicyStatementProps()
{
    Effect = Effect.ALLOW,
    Actions = new []{ "sns>ListTopics" },
    Resources = new []{ $"arn:aws:sns:{this.Region}:{this.Account}:*" }
}));
```

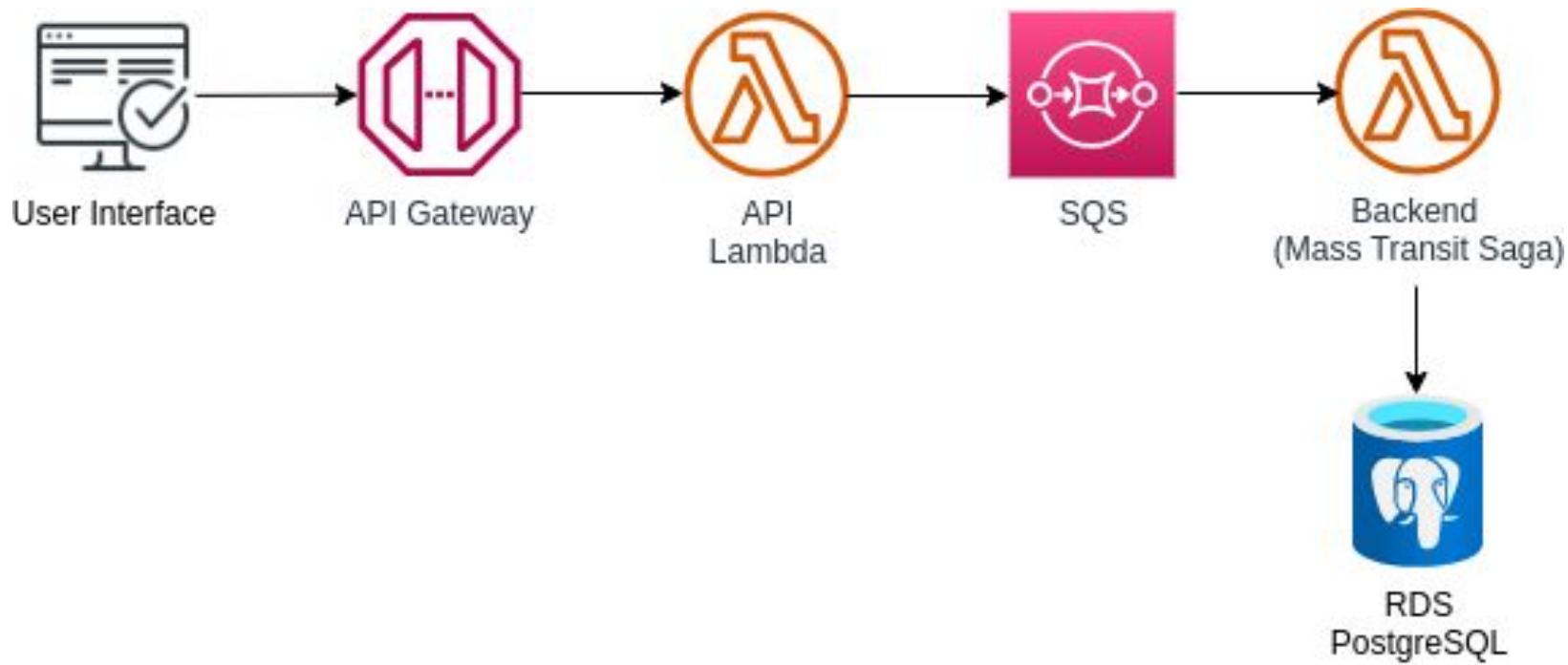
API Gateway

```
//Proxy all request from the root path "/" to Lambda
var restAPI = new LambdaRestApi(this, "Endpoint", new LambdaRestApiProps
{
    Handler = apiLambda,
    Proxy = true,
});
new CfnOutput(this, "apigwtarn", new CfnOutputProps { Value = restAPI.ArnForExecuteApi() });
```

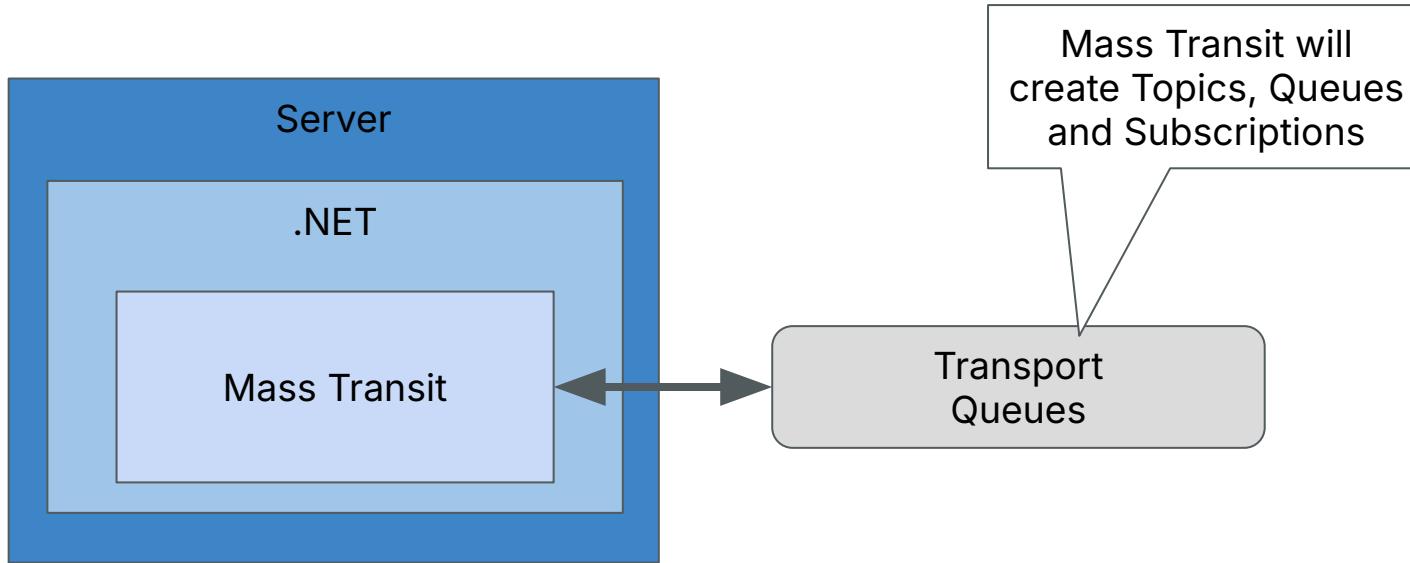
CDK Deploy

```
- q > ➜ ~/code/talks/mass-transit-trivia/mass-transit/MttInfra ➜ ✘ main +2 !9  
- cdk deploy "brendan-trivia-stack" --profile digio-sandbox
```

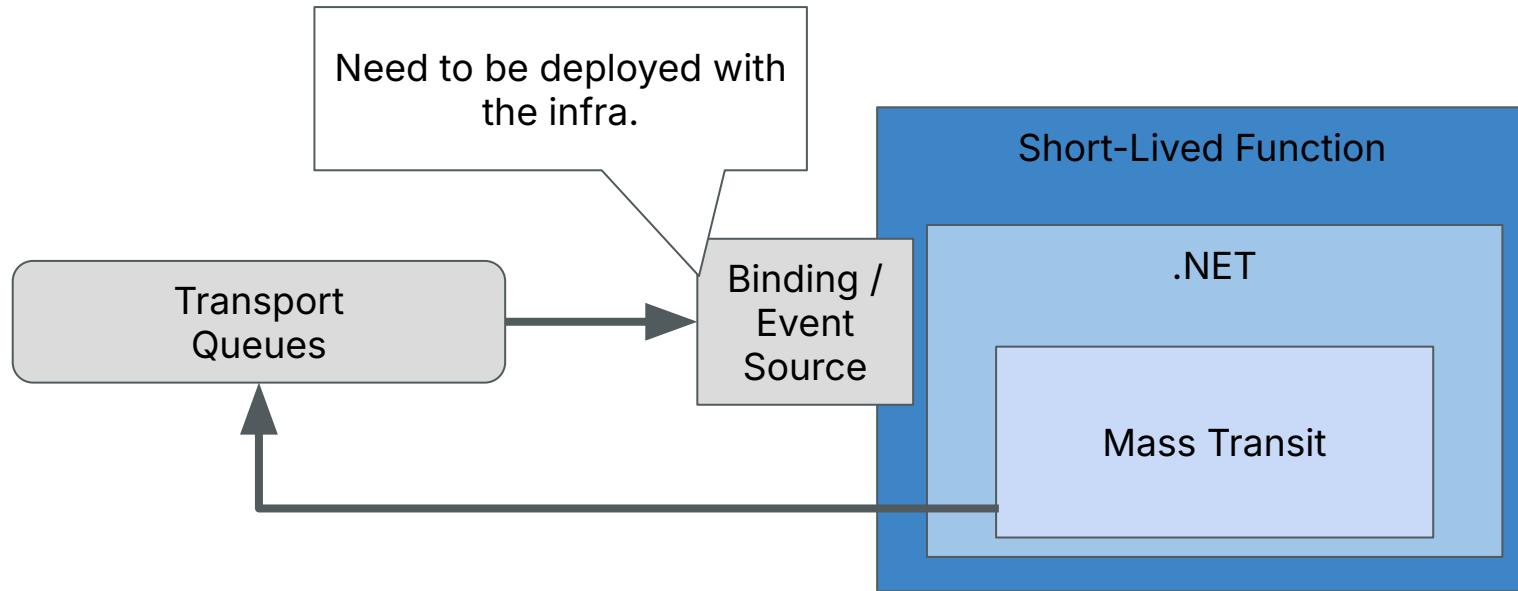
Architecture



Mass Transit Easy Path



Mass Transit + Serverless



```
x.UsingAmazonSqs((context, cfg) =>
{
    cfg.Host("ap-southeast-2", h =>
    {
        h.AccessKey(config["user-access-key"]);
        h.SecretKey(config["user-secret"]);
        h.Scope("brendan-trivia", true);
        h.EnableScopedTopics();
    });
    cfg.DeployTopologyOnly = true;
    cfg.ConfigureEndpoints(context);
});

Console.WriteLine("Deploy Topology...");
await busControl.DeployAsync(source.Token);
```

Topics (18)

[Edit](#)[Delete](#)[Publish message](#)[Create topic](#) brendan-trivia X

12 matches.

< 1 >

	Name	Type	ARN
○	brendan-trivia_MassTransit_Contracts-RequestTimeoutExpired--Mtt...	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Ma...
○	brendan-trivia_MassTransit-Fault--MttApplication_Contracts-Create...	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Ma...
○	brendan-trivia_MassTransit-Fault--MttApplication_Contracts-FetchQ...	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Ma...
○	brendan-trivia_MttApplication_Contracts-AddParticipant	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Mtt...
○	brendan-trivia_MttApplication_Contracts-AnswerQuestion	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Mtt...
○	brendan-trivia_MttApplication_Contracts-CloseCurrentQuestion	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Mtt...
○	brendan-trivia_MttApplication_Contracts>CreateGame	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Mtt...
○	brendan-trivia_MttApplication_Contracts-FetchQuestions	Standard	arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_Mtt...

game-state

SNS subscriptions (13) [Info](#)



Search subscriptions

Subscription ARN

- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MassTransit_Contracts-RequestTimeoutExpired--MttApplication_Contracts-FetchQuestions--:aee49471-408f-46fa-83ff-3210
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MassTransit-Fault--MttApplication_Contracts-FetchQuestions--:7c375e3e-2238-454a-80f8-12fd51fee309
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-AddParticipant:66782c0a-a3b7-49ef-a955-c82d07c90e3e
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-AnswerQuestion:2b072c9c-c8fc-4e2e-965e-7fc04966ea46
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-CloseCurrentQuestion:e513cd72-b3ea-4522-8fb5-622920b5023a
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts>CreateGame:792bf767-248e-4eae-898c-9d340bf7adeb
- arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-GetParticipantState:12434745-18d1-414b-a3cd-484f2e97aab4

game-state

Access policy (Permissions) Info

Define who can access your queue.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "4e1299232fd0420687dcbbc483c39ebf",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "sns.amazonaws.com"  
            },  
            "Action": "sns:SendMessage",  
            "Resource": "arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-CloseCurrentQuestion",  
            "Condition": {  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:sns:ap-southeast-2:864141050364:brendan-trivia_MttApplication_Contracts-CloseCurrentQuestion"  
                }  
            }  
        }  
    ]  
}
```

CDK - Reference existing Queue

```
var gameStateQueue =  
    Queue.FromQueueArn(scope: this, id: "game-state", queueArn: $"arn:aws:sqs:{this.Region}"  
  
var backendLambda = new Function(scope: this, id: "MttBackendLambda", new FunctionProps  
{  
    Runtime = Runtime.DOTNET_6,  
    MemorySize = 1024,  
    LogRetention = RetentionDays.ONE_DAY,  
    Handler = "MttBackend::MttBackend.Function::SQSHandler",  
    Vpc = dbVpc,  
    Code = Code.FromAsset(path: "../", new Amazon.CDK.AWS.S3.Assets.AssetOptions // pare  
      
    backendLambda.AddEventSource(new SqsEventSource(gameStateQueue));
```

Lambda Receive Message

```
public class Function
{
    private readonly Task<IServiceProvider> _buildServiceProviderTask;
    private readonly AwsSecrets _awsSecrets;

     Brendan Richards
    public Function()
    {
        _awsSecrets = new AwsSecrets();
        _buildServiceProviderTask = BuildServiceProvider();
    }

     Brendan Richards
    public async Task SQSHandler(SQSEvent evnt, ILambdaContext context)
    {
```

better.

2023-12-01T08:56:18.296Z 86f9cd25-75a5-56db-af82-90c6d444693b info SQS Handler Got Message:

```
{  
    "messageId": "08000000-b963-5aae-0dbc-08dbf24b5ffa",  
    "requestId": null,  
    "correlationId": "dcaeba1d-7453-4969-a5b4-1908419c382c",  
    "conversationId": "08000000-b963-5aae-0d64-08dbf24b5ffc",  
    "initiatorId": null,  
    "sourceAddress": "amazonsqs://ap-southeast-2/brendan-trivia/169_MttApi_bus_byyyyyf3cppk7uyhbdp9r14zfn?  
durable=false&autodelete=true",  
    "destinationAddress": "amazonsqs://ap-southeast-2/brendan-trivia_MttApplication_Contracts-CreateGame?type=topic",  
    "responseAddress": null,  
    "faultAddress": null,  
    "messageType": [  
        "urn:message:MttApplication.Contracts>CreateGame"  
    ],  
    "message": {  
        "correlationId": "dcaeba1d-7453-4969-a5b4-1908419c382c",  
        "name": "test026",  
        "createdAt": "2023-12-01T08:56:14.9485775Z"  
    },  
    "expirationTime": null,  
    "sentTime": "2023-12-01T08:56:14.974918Z",  
    "headers": {},  
    "host": {  
        "machineName": "169",  
        "processName": "MttApi",  
        "processId": 8,  
        "assembly": "MttApi",  
        "assemblyVersion": "1.0.0.0",  
        "frameworkVersion": "6.0.22",  
        "massTransitVersion": "8.1.2.0",  
        "operatingSystemVersion": "Unix 4.14.255.327"  
    }  
}
```

works for the better.

2023-12-01T08:56:18.296Z 86f9cd25-75a5-56db-af82-90c6d444693b info SQS Handler Got Message:

```
{  
    "messageId": "08000000-b963-5aae-0dbc-08dbf24b5ffa",  
    "requestId": null,  
    "correlationId": "dcaeba1d-7453-4969-a5b4-1908419c382c",  
    "conversationId": "08000000-b963-5aae-0d64-08dbf24b5ffc",  
    "initiatorId": null,  
    "sourceAddress": "amazonsqs://ap-southeast-2/brendan-trivia/169_MttApi_bus_byyyyyf3cppk7uyhbdp9r14zfn?  
durable=false&autodelete=true",  
    "destinationAddress": "amazonsqs://ap-southeast-2/brendan-trivia_MttApplication_Contracts-CreateGame?type=topic",  
    "responseAddress": null,  
    "faultAddress": null,  
    "messageType": [  
        "urn:message:MttApplication.Contracts>CreateGame"  
    ],  
    "message": {  
        "correlationId": "dcaeba1d-7453-4969-a5b4-1908419c382c",  
        "name": "test026",  
        "createdAt": "2023-12-01T08:56:14.9485775Z"  
    },  
    "expirationTime": null,  
    "sentTime": "2023-12-01T08:56:14.974918Z",  
    "headers": {},  
    "host": {  
        "machineName": "169",  
        "processName": "MttApi",  
        "processId": 8,  
        "assembly": "MttApi",  
        "assemblyVersion": "1.0.0.0",  
        "frameworkVersion": "6.0.22",  
        "massTransitVersion": "8.1.2.0",  
        "operatingSystemVersion": "Unix 4.14.255.327"  
    }  
}
```

works for the better.

Table Of Contents

Event Driven Concepts

Introducing our Use Case

Mass Transit, Sagas and State Machines

Mass Transit Testing

Persistence and Integration Testing

Deploy to AWS with CDK



Brendan Richards (He/Him)

Lead Software Engineer @ Mantel Group



brendan-nobadthing / **mass-transit-trivia**

PLATINUM



Octopus Deploy

GOLD

DevBricks^W

Insight^{IT}

PATIENT ZERO[®]



VALID

SILVER

LMG[®]

Seq[®]
Machine data for humans

NDC { Sydney }

STANDARD

SixPivot

CHILDCARE

Octopus Deploy

COFFEE

Seq[®]
Machine data for humans



DIVERSITY

SixPivot

MEDIA



COMMUNITY



YOW!



Please give
me feedback!







Add client logo here. Set the height of the logo to the same height as this rectangle. Please remove rectangle once done.

Presentation template

Mantel Group base template

August 2023



Presentation template

**Mantel Group base
template**

August 2023



- 900+ team members
- Offices across Australia and New Zealand
- Voted Best Place to Work in 2021 and 2022
- Operating at over 140 clients

Mantel group



- 900+ team members
- Offices across Australia and New Zealand
- Voted Best Place to Work in 2021 and 2022
- Operating at over 140 clients

Layouts for presentation

With Mantel Group logo and tagline



Single row heading

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Single row heading

Awesome subheading

Marshmallow roll sweet roll topping
shortbread toffee. Chocolate bar
marshmallow sweet chocolate carrot
cake. Croissant cake caramels
gummi bears icing. Powder brownie
ice cream candy donut liquorice
muffin jelly marshmallow roll.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet roll topping
shortbread toffee. Chocolate bar
marshmallow sweet chocolate carrot
cake. Croissant cake caramels
gummi bears icing. Powder brownie
ice cream candy donut liquorice
muffin jelly marshmallow roll.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Single row heading

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Caption

Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit. Pudding muffin ice cream carrot cake jelly jelly-o toffee.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Caption

Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit. Pudding muffin ice cream carrot cake jelly jelly-o toffee.

Single row heading

Awesome subheading

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Single row heading

Awesome subheading

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Single row heading

Awesome subheading

Layouts for presentation

Without Mantel Group logo and tagline

Single row heading

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit pudding.

Single row heading

Awesome subheading

Marshmallow roll sweet roll topping
shortbread toffee. Chocolate bar
marshmallow sweet chocolate carrot
cake. Croissant cake caramels
gummi bears icing. Powder brownie
ice cream candy donut liquorice
muffin jelly marshmallow roll.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet roll topping
shortbread toffee. Chocolate bar
marshmallow sweet chocolate carrot
cake. Croissant cake caramels
gummi bears icing. Powder brownie
ice cream candy donut liquorice
muffin jelly marshmallow roll.

Heading spanning two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping
shortbread toffee. Chocolate bar
marshmallow sweet chocolate carrot
cake. Croissant cake caramels
gummi bears icing. Powder brownie
ice cream candy donut liquorice
muffin jelly marshmallow roll.

Single row heading

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Caption

Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit. Pudding muffin ice cream carrot cake jelly jelly-o toffee.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar marshmallow sweet chocolate carrot cake. Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll.

Caption

Croissant cake caramels gummi bears icing. Powder brownie ice cream candy donut liquorice muffin jelly marshmallow roll. Jelly fruitcake donut chupa chups jujubes. Muffin marshmallow roll pudding caramels biscuit. Pudding muffin ice cream carrot cake jelly jelly-o toffee.

Single row heading

Awesome subheading

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Caption

Marshmallow roll
sweet roll topping
shortbread toffee.
Chocolate bar
marshmallow sweet
chocolate carrot cake.
Croissant cake.

Single row heading

Awesome subheading

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Heading spanning over two rows

Awesome subheading

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar
marshmallow sweet.

Single row heading

Awesome subheading

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Caption

Marshmallow roll sweet
roll topping shortbread
toffee. Chocolate bar.

Heading spanning over two rows

Awesome subheading

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Caption

Marshmallow roll sweet roll topping shortbread toffee. Chocolate bar.

Single row heading

Awesome subheading

Single row heading

Awesome subheading