
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

FWM310

Building a multi-Region serverless application with AWS AppSync

Ozioma Uzoegwu (he/him/his)

Senior Solutions Architect
Amazon Web Services

@iam_tessot

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some housekeeping items

QR codes are sprinkled throughout



We'll let you know when a buildout is complete for photos

FWM310

300 level with a mix of 200- and 400-level content

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introduction to GraphQL and AWS AppSync

Why multi-Region deployment for AWS AppSync

Architecture patterns

Operating your multi-Region AWS AppSync API

Demo

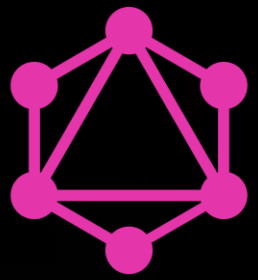
Resources

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to GraphQL

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to GraphQL



GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data

Describe your data

```
type Todo {
   id: ID!
   name: String
   description: String
   priority: Int
   status: TodoStatus
}
```

Ask for what you want

```
query ListTodoQuery {
   listTodos {
      todos {
         name
      }
   }
}
```

Get predictable results

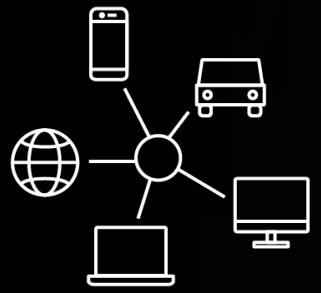
```
{
   “data” : {
      “listTodos” : {
         “todos”: [{“name”: “Eat Lunch”}]
      }
   }
}
```

aws

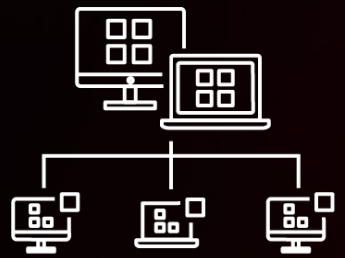
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to GraphQL

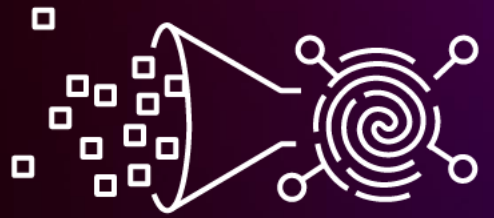
GRAPHQL API PROVIDES THESE BENEFITS TO API BUILDERS AND CONSUMERS



Eliminate over-fetching and under-fetching



Decouple frontend and backend dev team

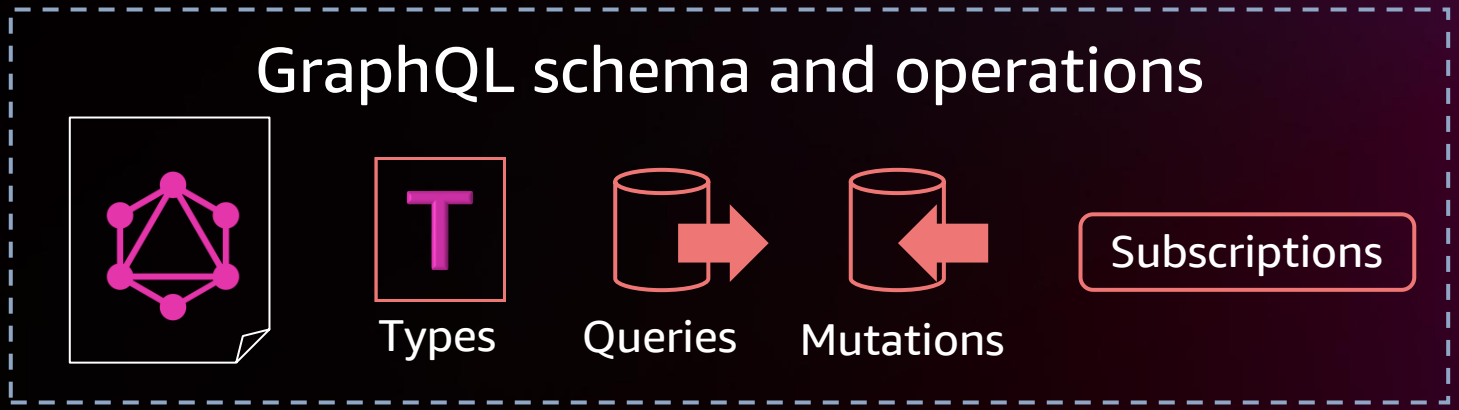


Build engaging apps with Pub/Sub APIs

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to GraphQL




```
type Todo {
  id: ID!
  name: String
  description: String
  priority: Int
  status: TodoStatus
}
```

```
enum TodoStatus{
  done
  pending
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to GraphQL

Queries

```
query GetTodo {
  getTodo(id: “1”) {
    id
    name
  }
}
```

Mutations

```
mutation CreateTodo {
  createTodo(input: {…}) {
    id
    name

    priority
  }
}
```

Subscriptions

```
subscription OnCreateTodo {
  onCreateTodo {
    id
    name
    description
  }
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to AWS AppSync

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

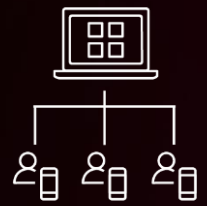
Introduction to AWS AppSync



AWS AppSync is a serverless **GraphQL** and **Pub/Sub API** service that simplifies building modern web and mobile applications



Single data API



Real-time data experiences

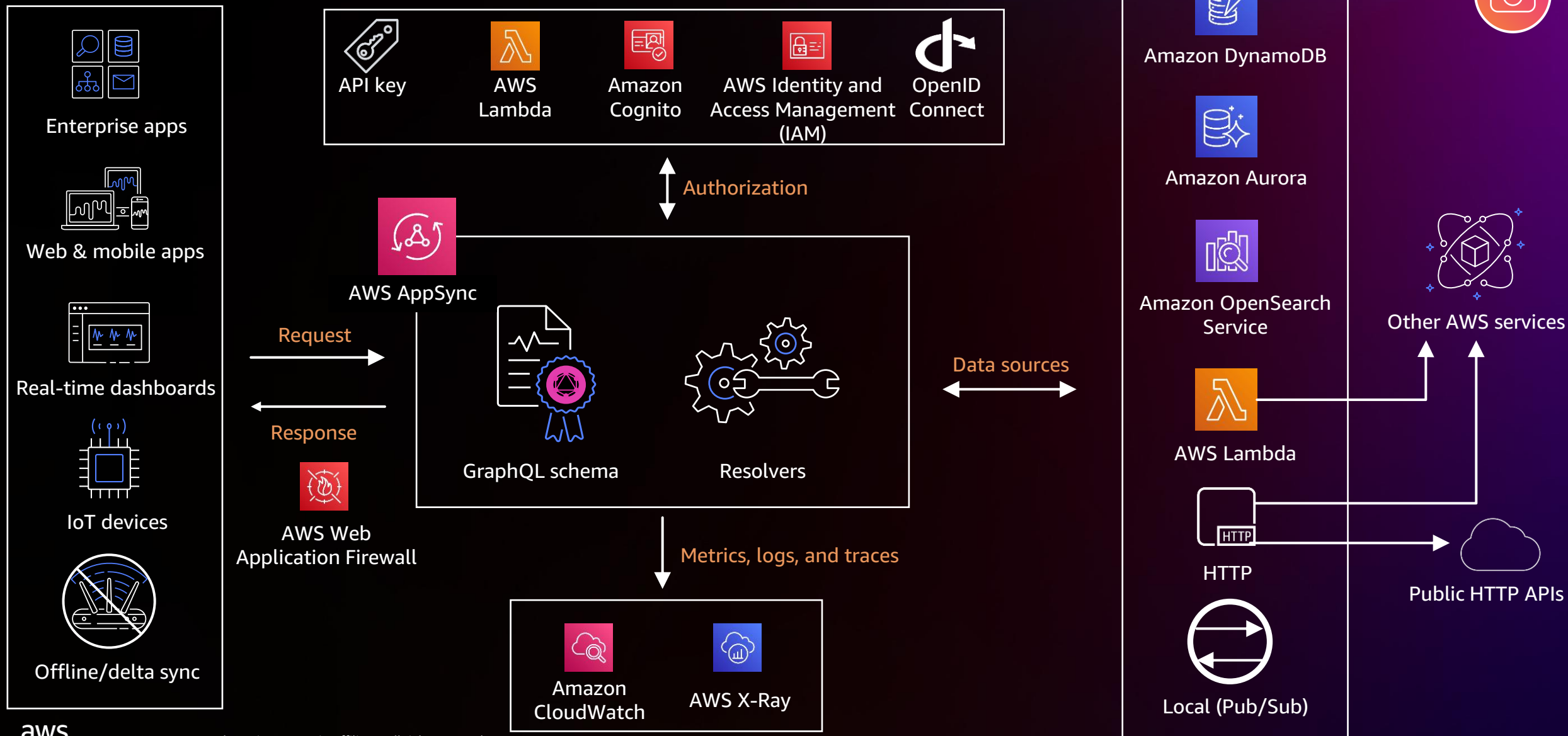


100% serverless

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How AWS AppSync works



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

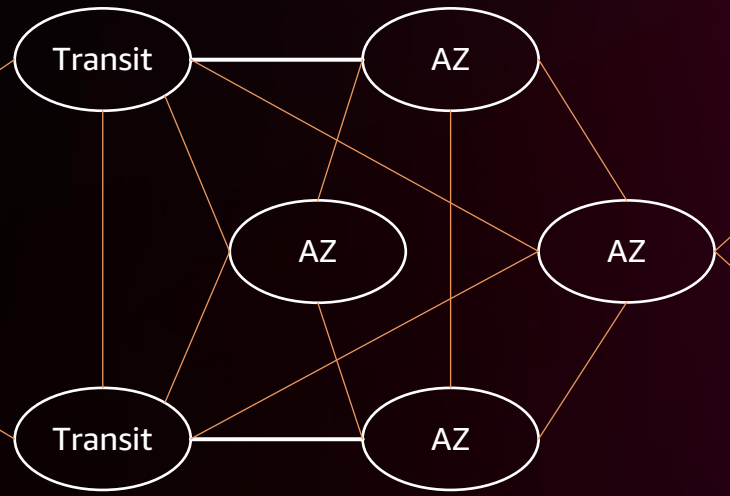
Why multi-Region deployment for AWS AppSync

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

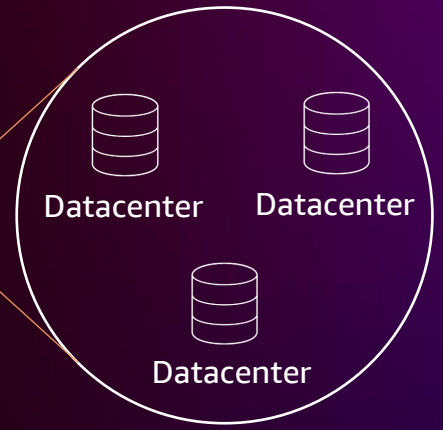
AWS Global Infrastructure



AWS Region



AWS Availability Zone (AZ)



- Redundant power, networking, and connectivity
- High availability, fault tolerance, and scalability
- Better isolation and protection from natural disaster

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



AWS AppSync is serverless

AWS is responsible for the **high availability and resilience** of your AWS AppSync API within a Region

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

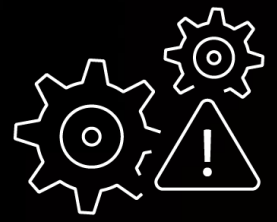
To go multi-Region, or not to go multi-Region?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

To go multi-Region, or not to go multi-Region?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why NOT go multi-Region?



Complexity



Cost



Data replication



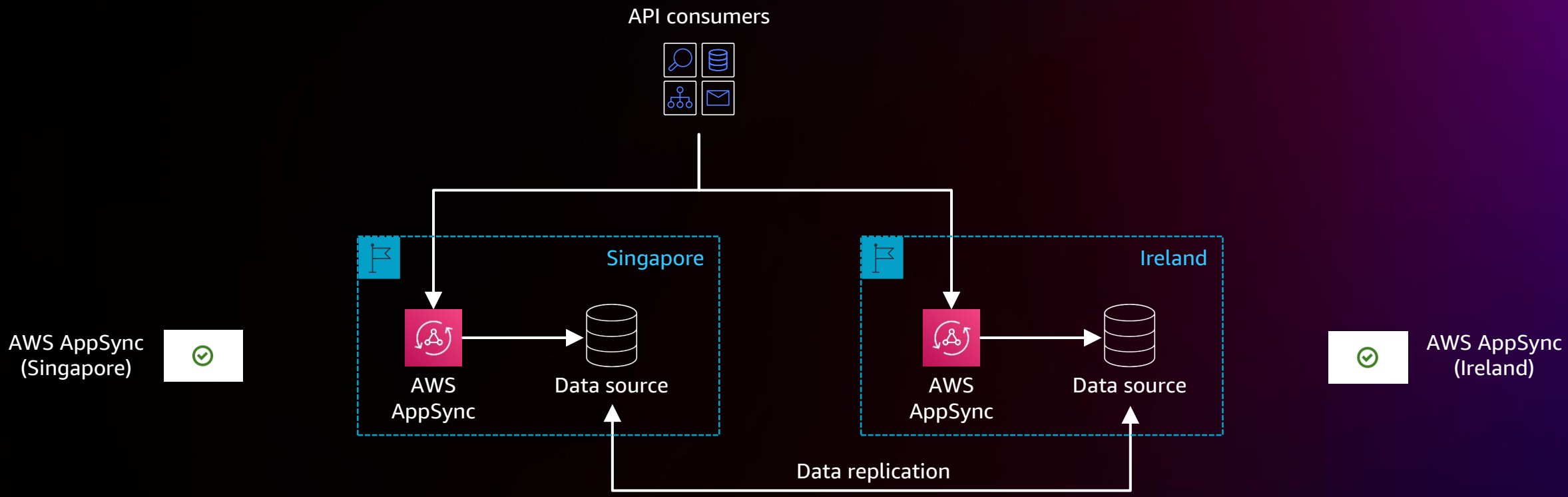
Testing

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why go multi-Region?

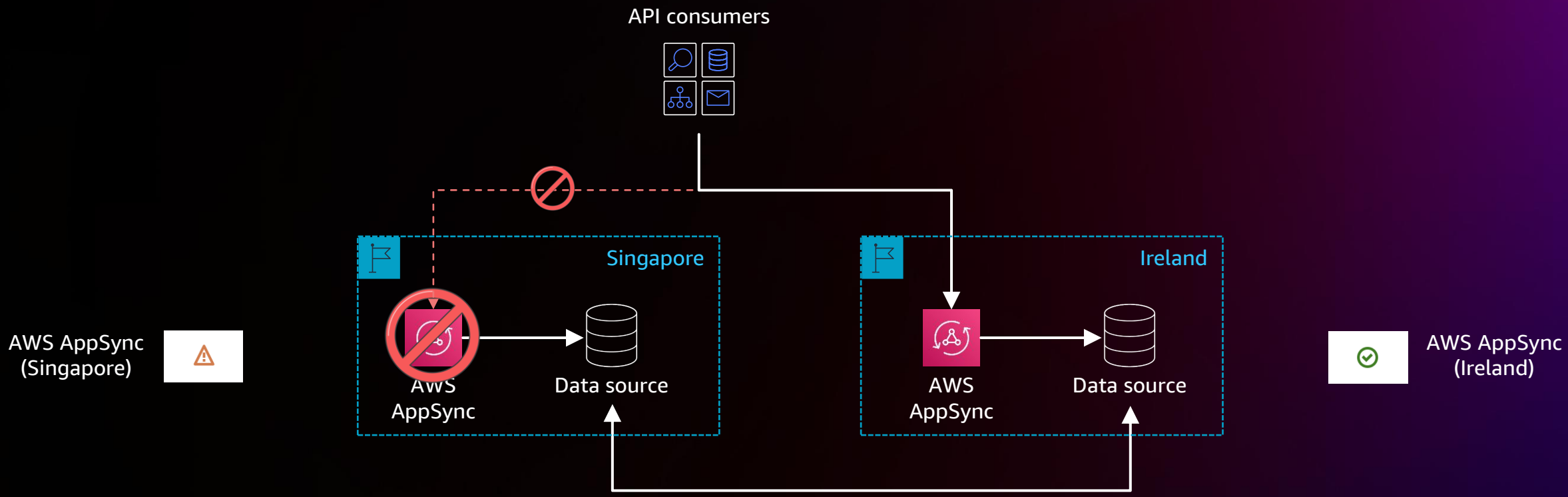
1. Business continuity/disaster recovery strategy



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why go multi-Region?

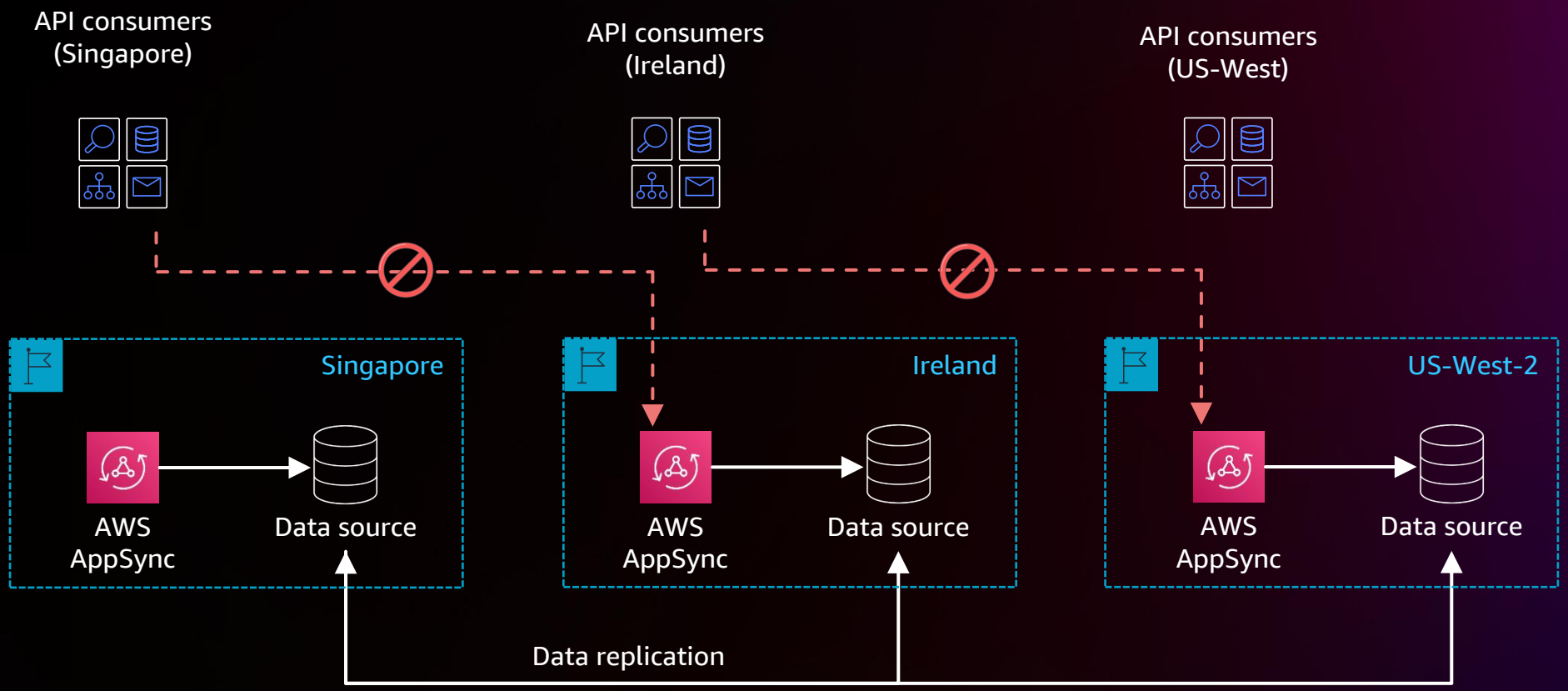
1. Business continuity/disaster recovery strategy



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why go multi-Region?

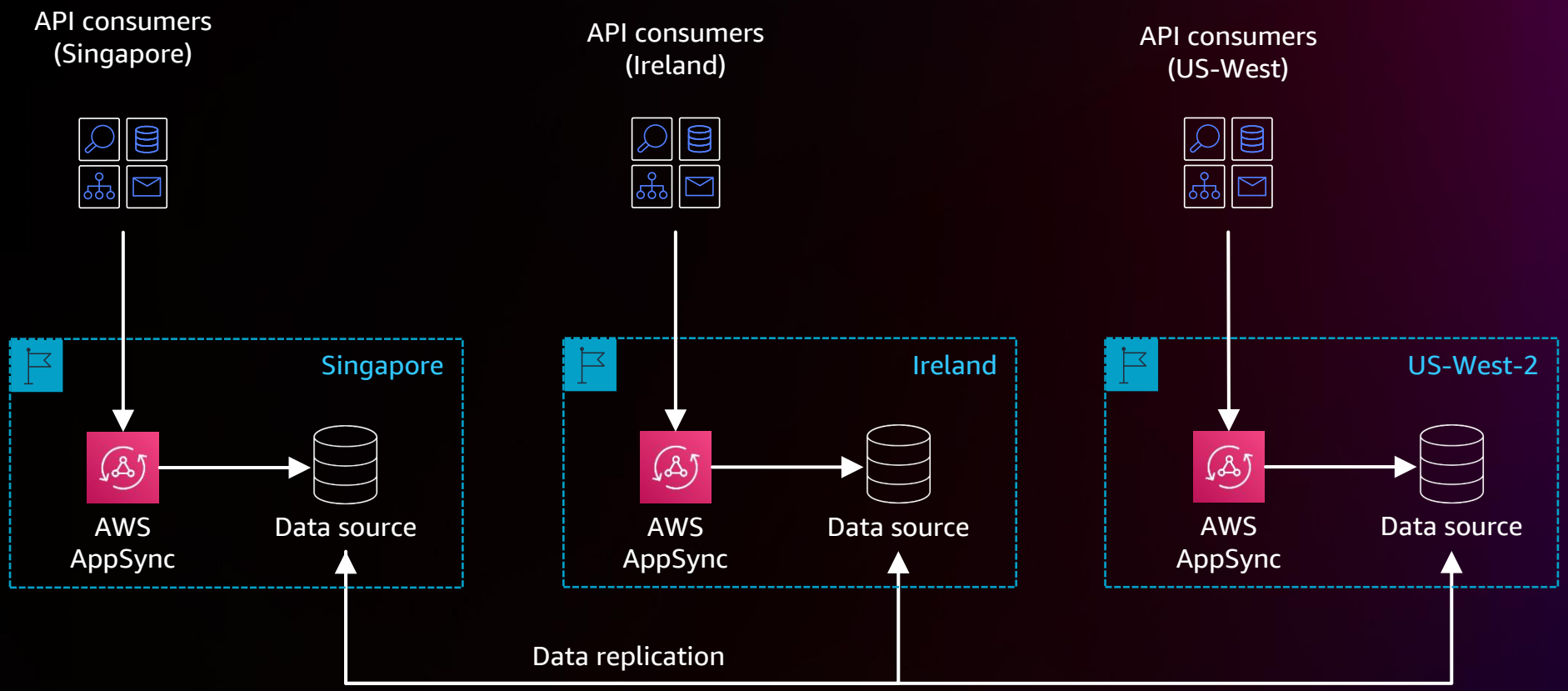
2. Improve latency for geographically dispersed users



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why go multi-Region?

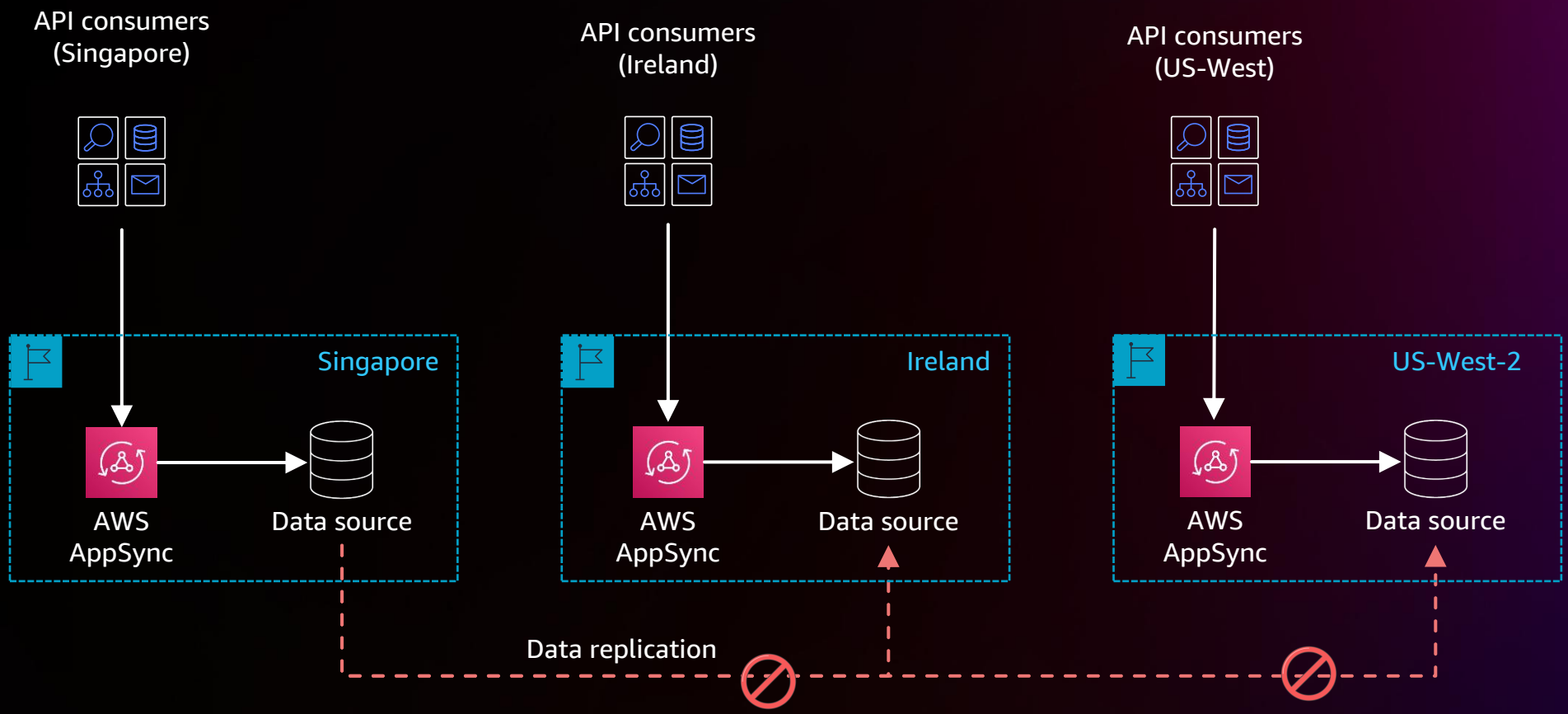
2. Improve latency for geographically distributed users



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why go multi-Region?

3. Data residency requirements



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture patterns for multi-Region AWS AppSync deployment

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Understand the application requirements

1. What is the **Recovery Time Objective (RTO)** and **Recovery Point Objective (RPO)**?
   How much data can you afford to recreate or lose (RPO) and how quickly must you recover (RTO)?

2. What are the **GraphQL operations** required for the API?
   Do you need to support queries, mutations, and subscriptions or a subset of operations?

3. What are the **data sources** and the **shape of the data** generated?
   Does your application generate and consume or only consume data? Internal or external data sources? Key-value or relational data?

4. What are the application **access patterns**?
   High-throughput/low-latency? Heavy reads? Heavy writes?

5. What is the **consistency model** required?
   Full consistency? Eventual consistency?

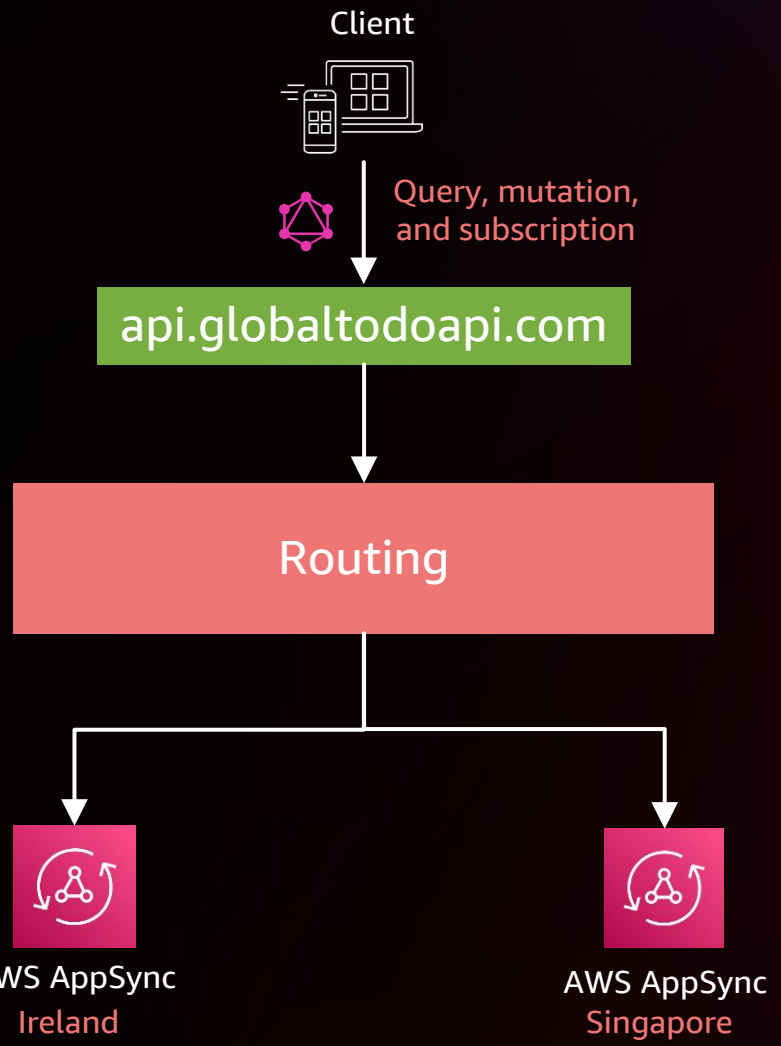
6. How should concurrent updates' **conflicts** be resolved?
   Last writer wins, first writer wins, etc.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-Region AWS AppSync routing via single endpoint

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-Region AWS AppSync routing via single endpoint



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

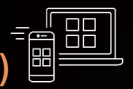
AWS AppSync custom domains

Without AWS AppSync custom domain

AWS AppSync

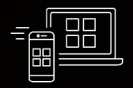


GraphQL endpoint
(for queries and mutation)



https://apiId.appsync-api.us-east-1.amazonaws.com/graphql

Real-time endpoint
(for subscription)



wss://apiId.appsync-realtime-api.us-east-1.amazonaws.com/graphql



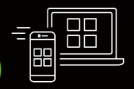
tinyurl.com/569kpxmz

With AWS AppSync custom domain

AWS AppSync



GraphQL endpoint
(for queries and mutation)



https://api.globaltodoapi.com/graphql

Real-time endpoint
(for subscription)



wss://api.globaltodoapi.com /graphql/realtime

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

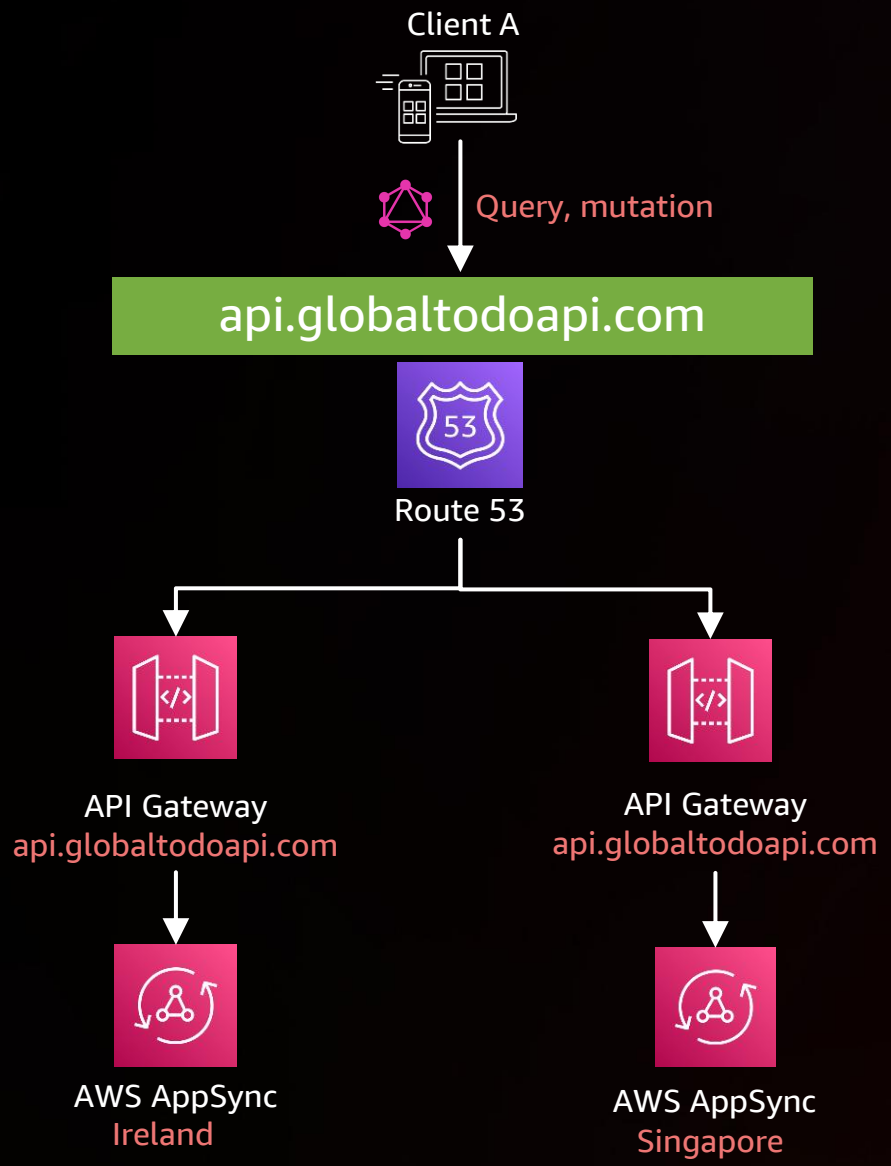
Routing GraphQL requests with single API endpoint

		Query	Mutation	Subscription
Option 1	Routing with Amazon Route 53 and Amazon API Gateway as a regional proxy	✓	✓	✗
Option 2	Routing with Amazon Route 53 and Amazon CloudFront (with AWS Lambda@Edge)	✓	✓	✓

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using Amazon Route 53 and API Gateway



1. Configure **API Gateway as a proxy** to forward request to the AWS AppSync API endpoint in the target Regions

Example API Gateway integration configuration

Route path	/todoapi
Route method	POST
Integration type	HTTP URI
Integration method type	ANY
Integration target	AWS AppSync endpoint

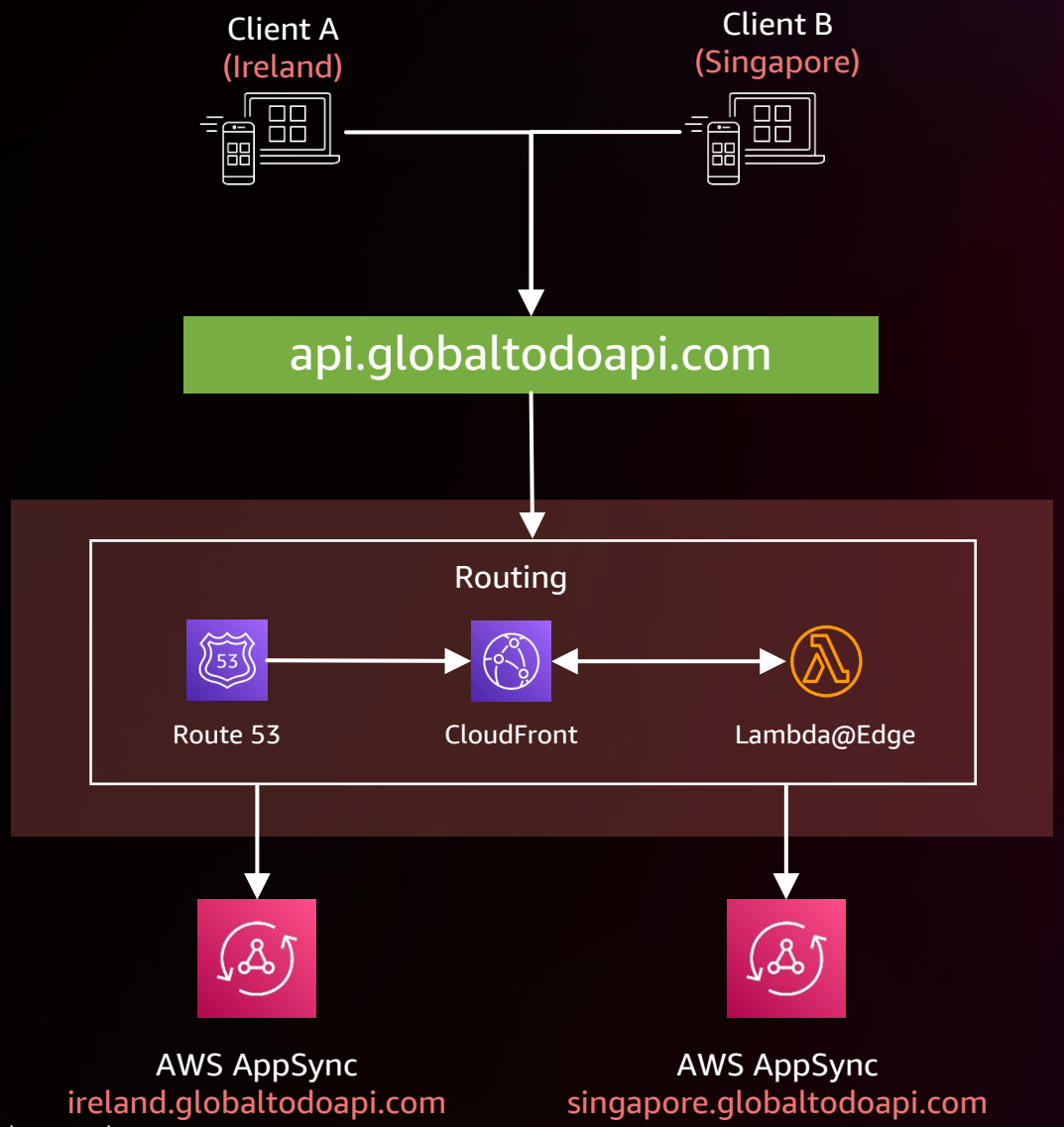
2. Configure **Regional custom domain** on the API Gateway endpoint in the target Regions

3. Create latency/geo-location/weighted record sets in Route 53 to **map the custom domain name to the API Gateway endpoints** in the two Regions

4. Send **queries and mutations** via the single API endpoint

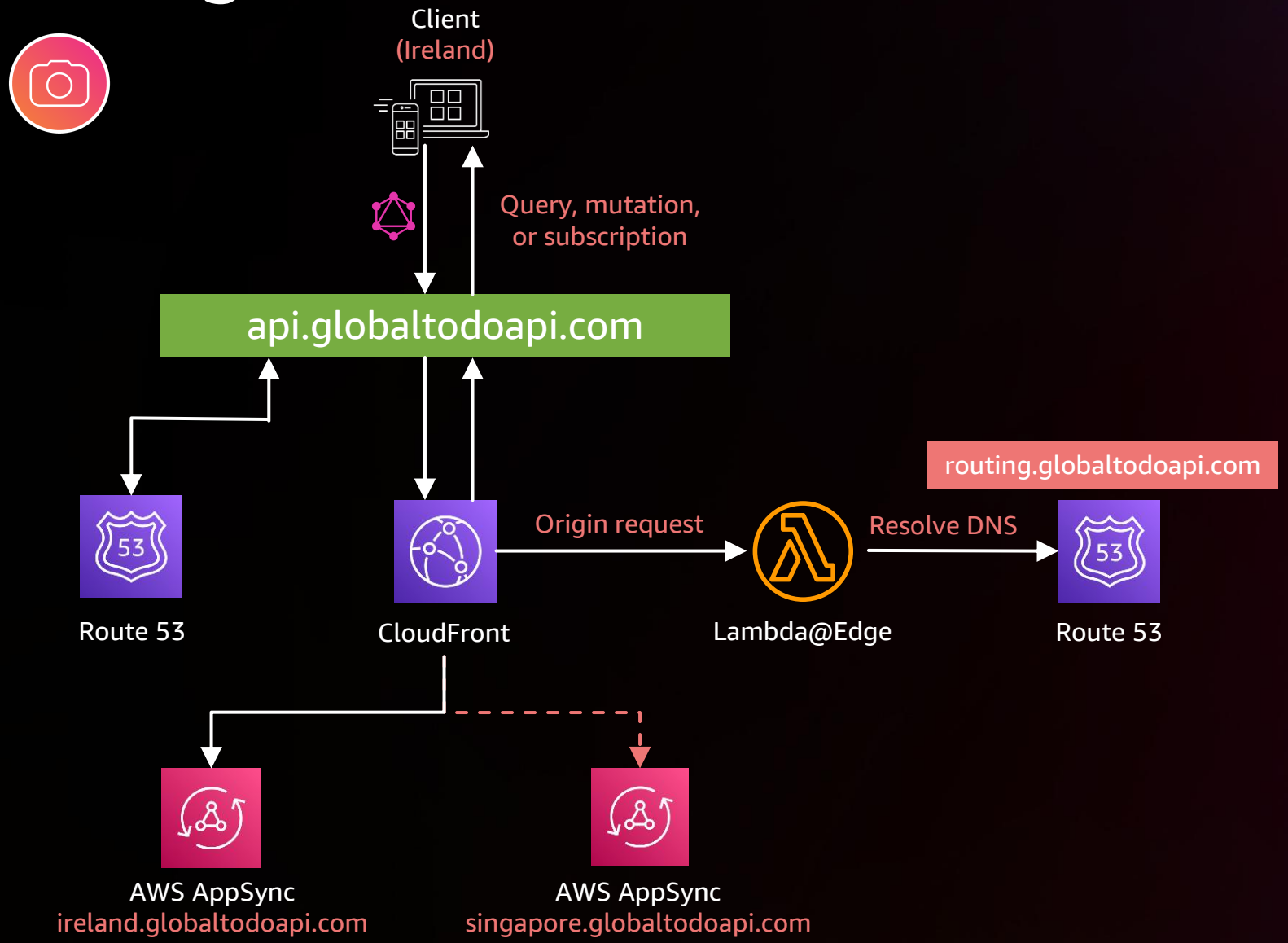
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using Amazon Route 53 and CloudFront



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using Amazon Route 53 and CloudFront



Lambda@Edge function will

1. Resolve the DNS **"routing.globaltodoapi.com"** to return the Region to route request

Example configuration for Ireland Region

Record name	routing.globaltodoapi.com
Type	CNAME
Routing policy	Latency/Geo-location/Weighted
Region	Europe (Ireland)
Value	eu-west-1

2. Update request headers

```
request.origin.custom.domainName
= "ireland.globaltodoapi.com";

request.headers["host"] = [{
 key: "host",
 value: "ireland.globaltodoapi.com"
}];
```

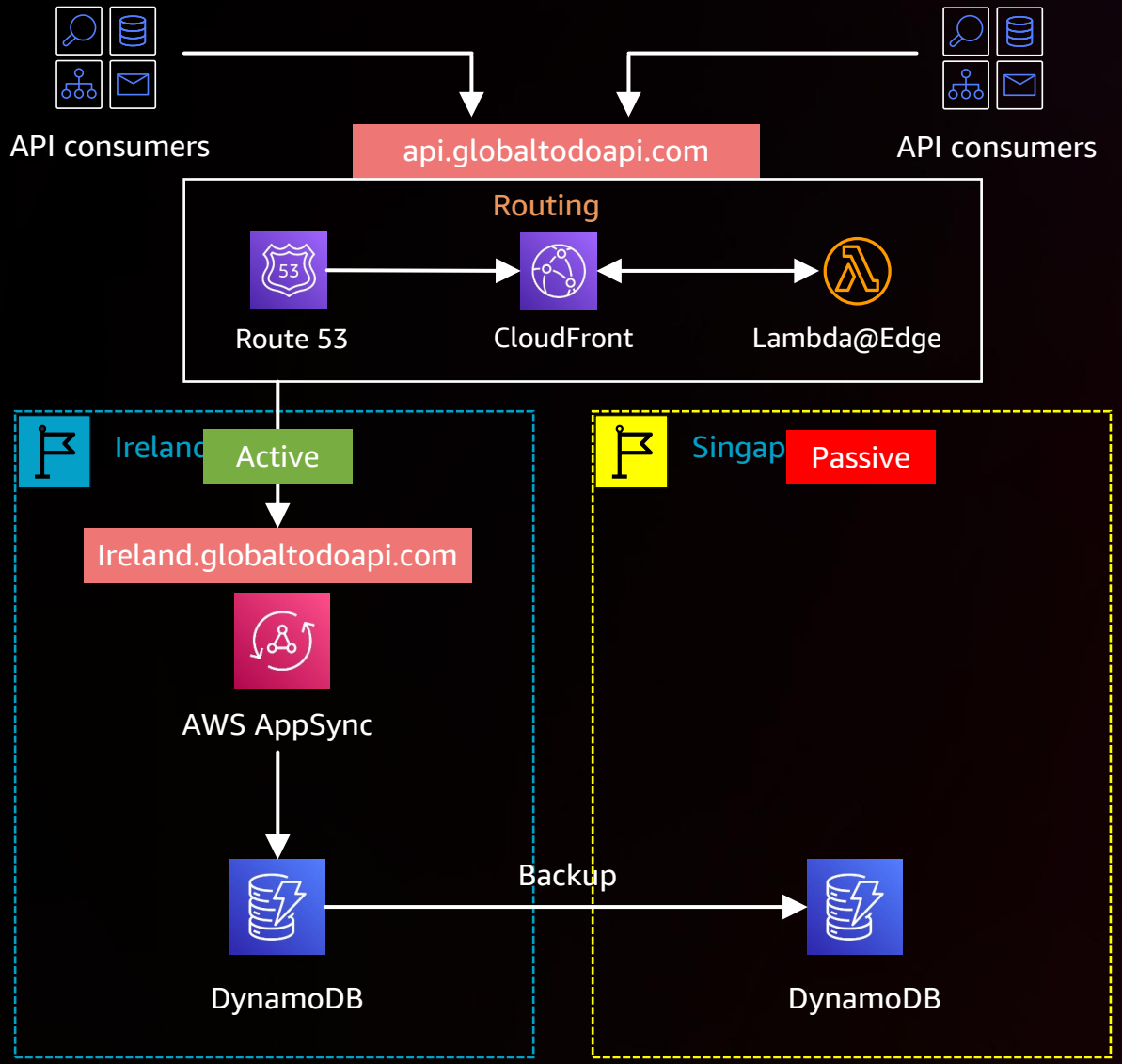
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case 1

Non-critical workload with RTO and RPO from minutes to hours

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

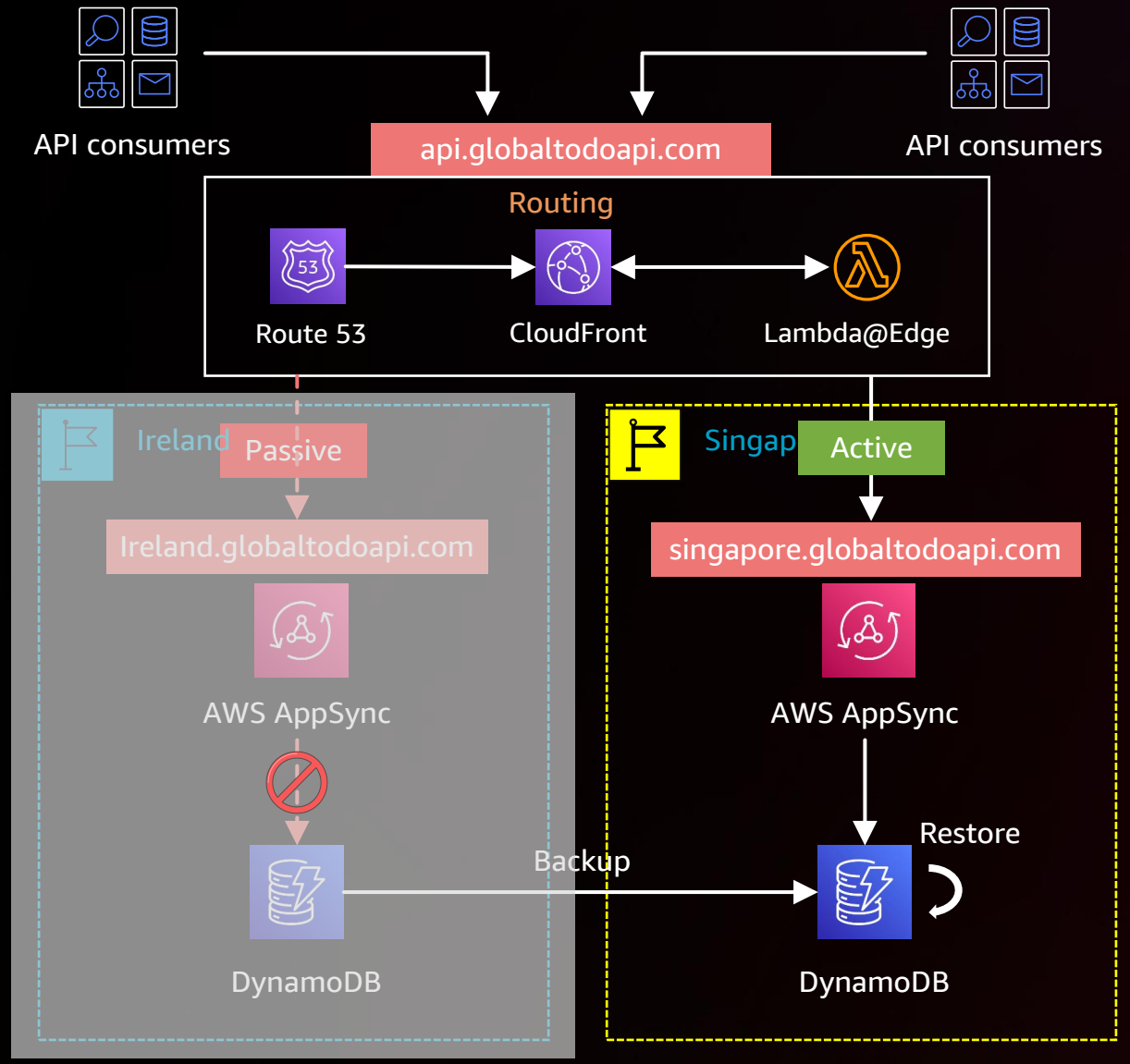
Non-critical workload – Active-passive pattern



1. Setup health checks to monitor the API endpoints
2. Use AWS Backup to set up backup policy on the DynamoDB table
3. Copy the backups to a different AWS Region or AWS account

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Non-critical workload – Active-passive pattern



Cross-Region failover

1. Restore the DynamoDB in the failover Region
2. Deploy the AWS AppSync API and configure custom domain
3. Ensure DNS failover to the new active region

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

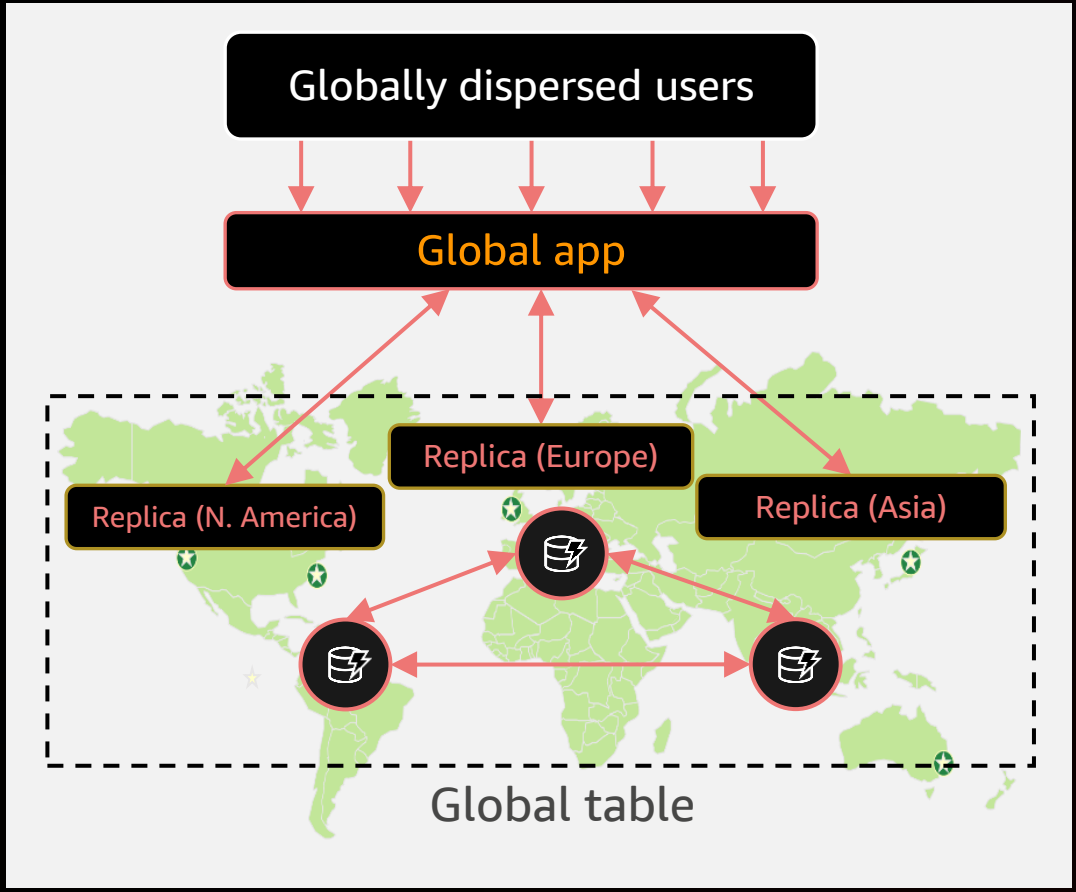
Use case 2

Critical workload with near-real-time RTO and RPO

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DynamoDB global tables

THE FIRST FULLY MANAGED, MULTI-MASTER, MULTI-REGION DATABASE



Build high-performance, globally distributed applications

Low-latency reads and writes to locally available tables

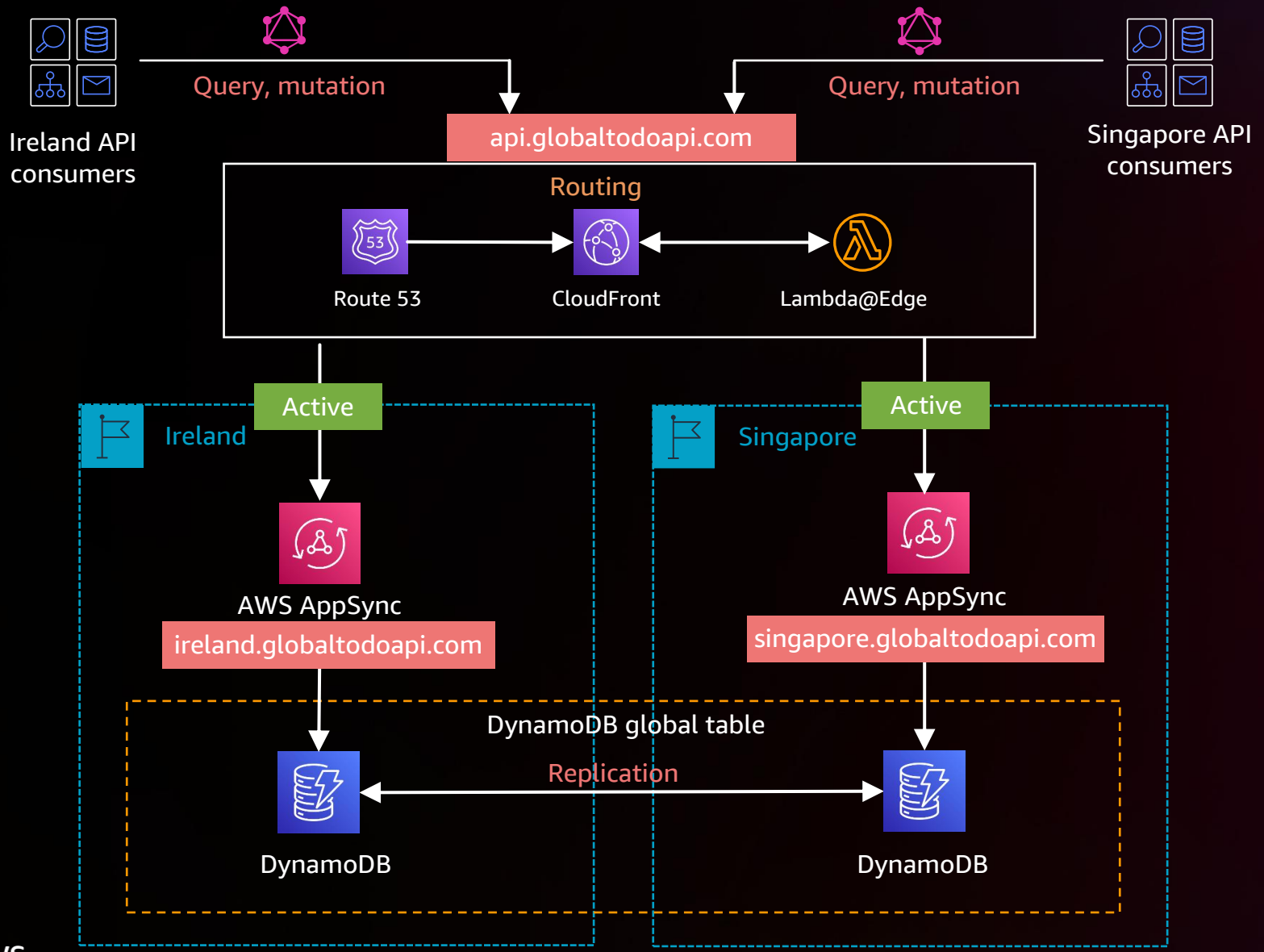
Disaster-proof with multi-Region redundancy

Easy to set up and no application re-writes required

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Critical workload – Active-active pattern

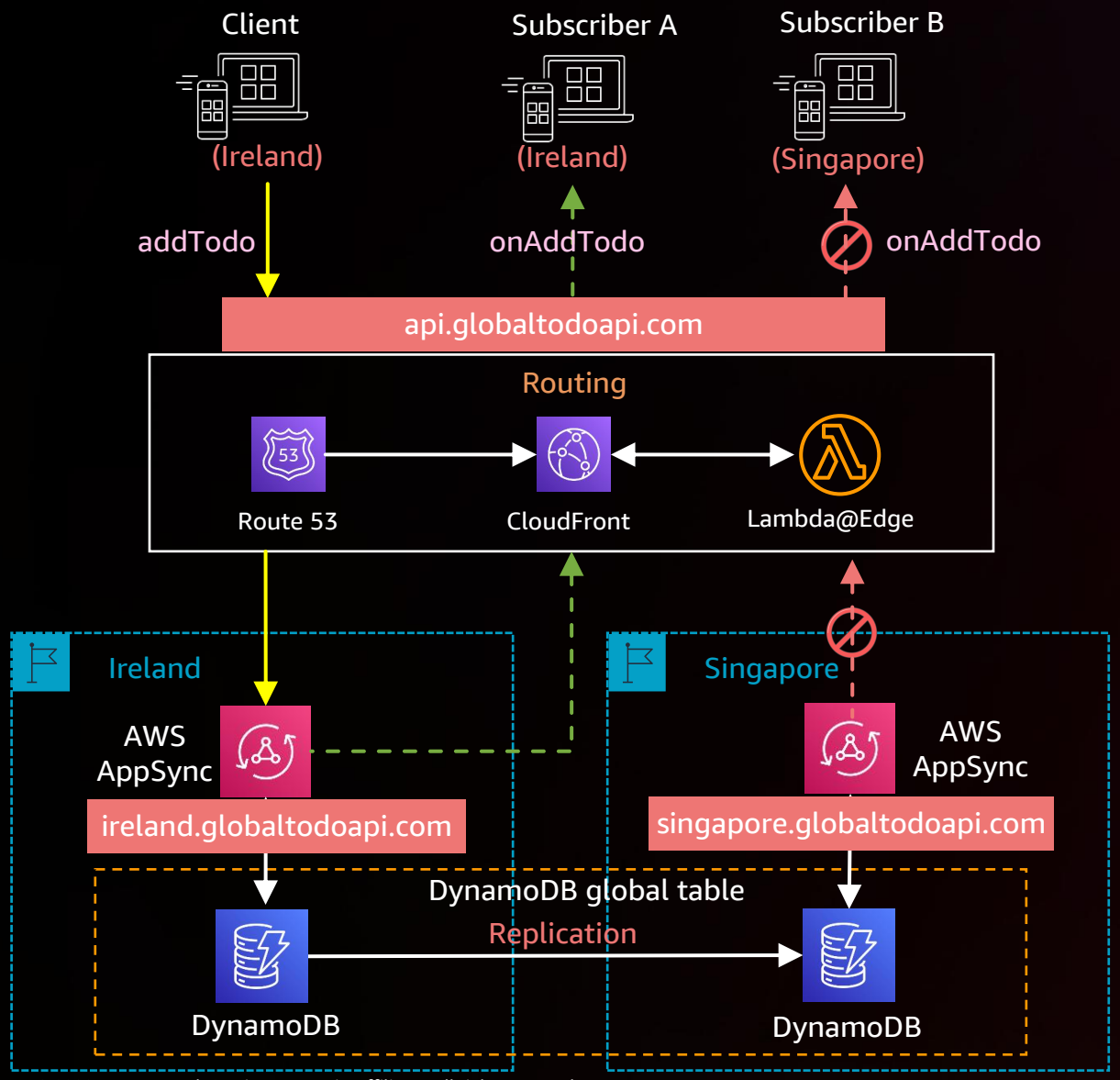




... but how about
GraphQL subscription

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enabling GraphQL subscription in multi-Region




```
type Todo {
   id: ID!
   name: String
   description: String
   priority: Int
   status: TodoStatus
}

type Mutation {
   addTodo(input: AddTodo): Todo
}

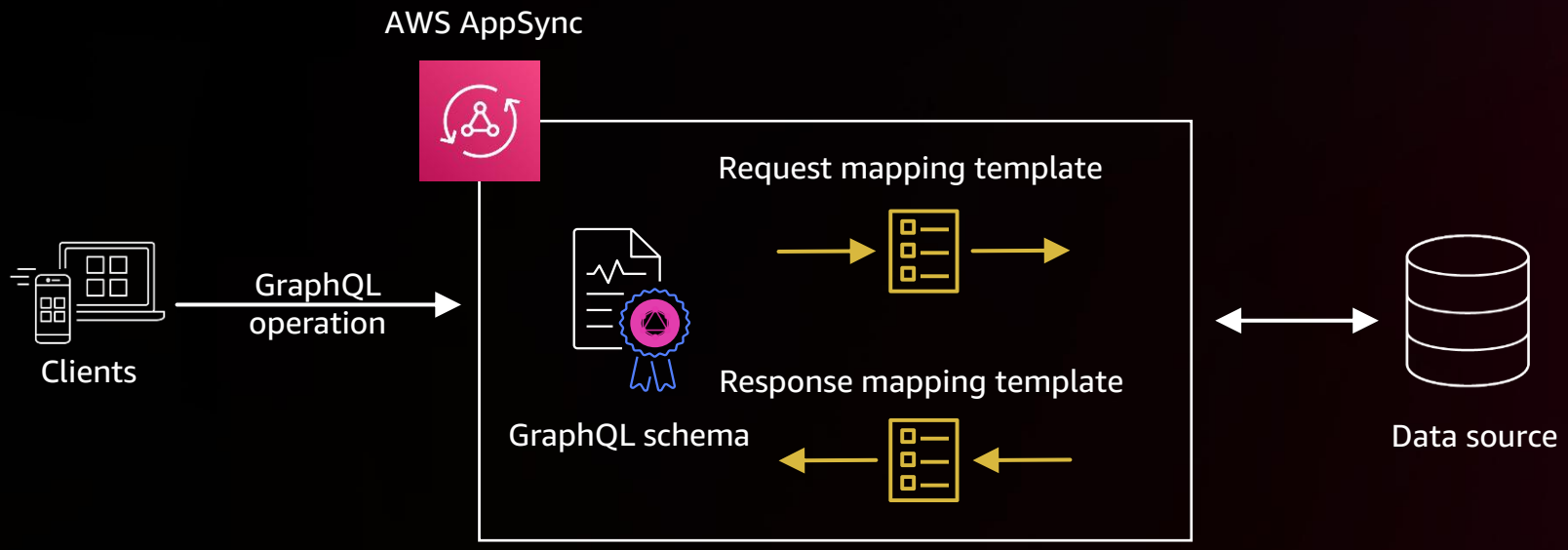
type Subscription {
   onAddTodo(input: AddTodo): Todo
   @aws_subscribe(mutations:["addTodo"])
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enabling GraphQL subscription in multi-Region

AWS APPSYNC RESOLVERS

AWS AppSync GraphQL APIs
Supports JavaScript Resolvers
tinyurl.com/5spm36rx



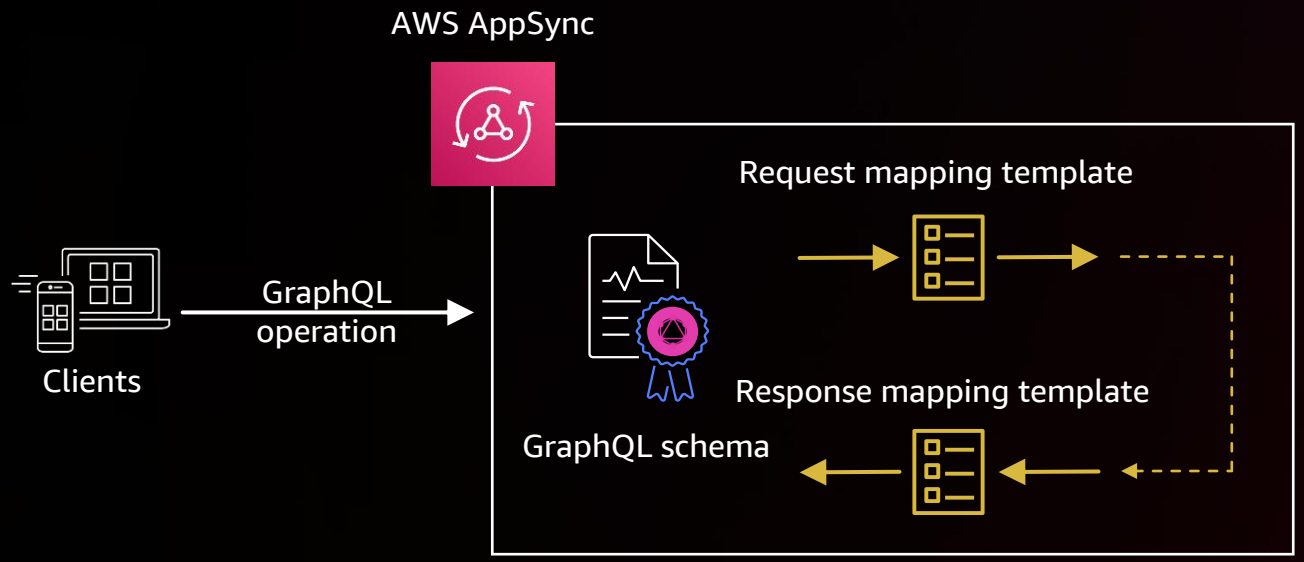
AWS AppSync resolvers

- AWS AppSync receives the request from clients
- Request mapping template transforms and executes the request
- Request is sent to the data source and the data source sends a response
- Response mapping template processes the response before sending to the clients

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enabling GraphQL subscription in multi-Region

USING AWS APPSYNC LOCAL RESOLVERS



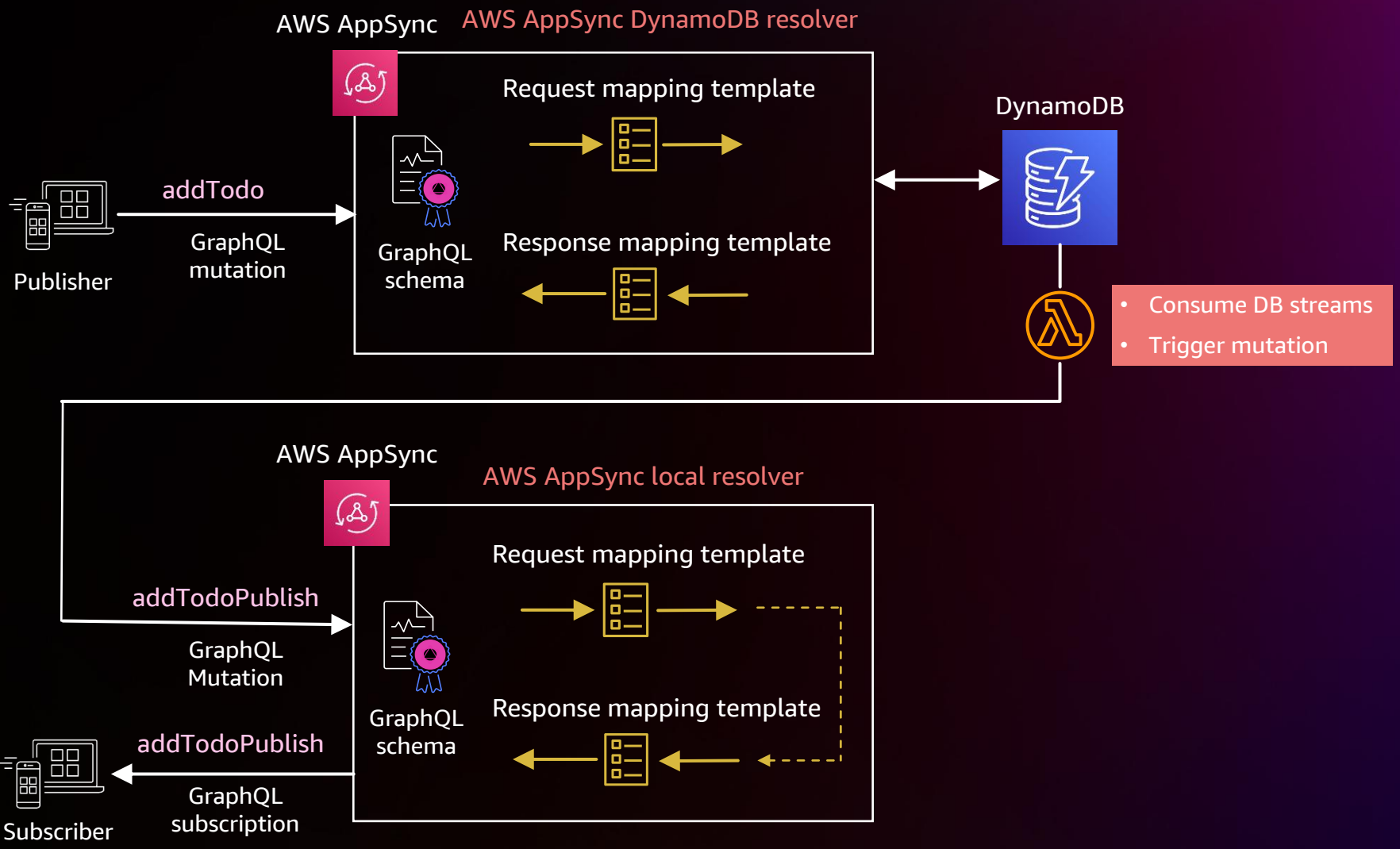
AWS AppSync Local Resolvers

- Forwards the result of request mapping template to the response mapping template
- Field resolution will not leave AWS AppSync
- Popularly used to **publish notification without triggering a data source call**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

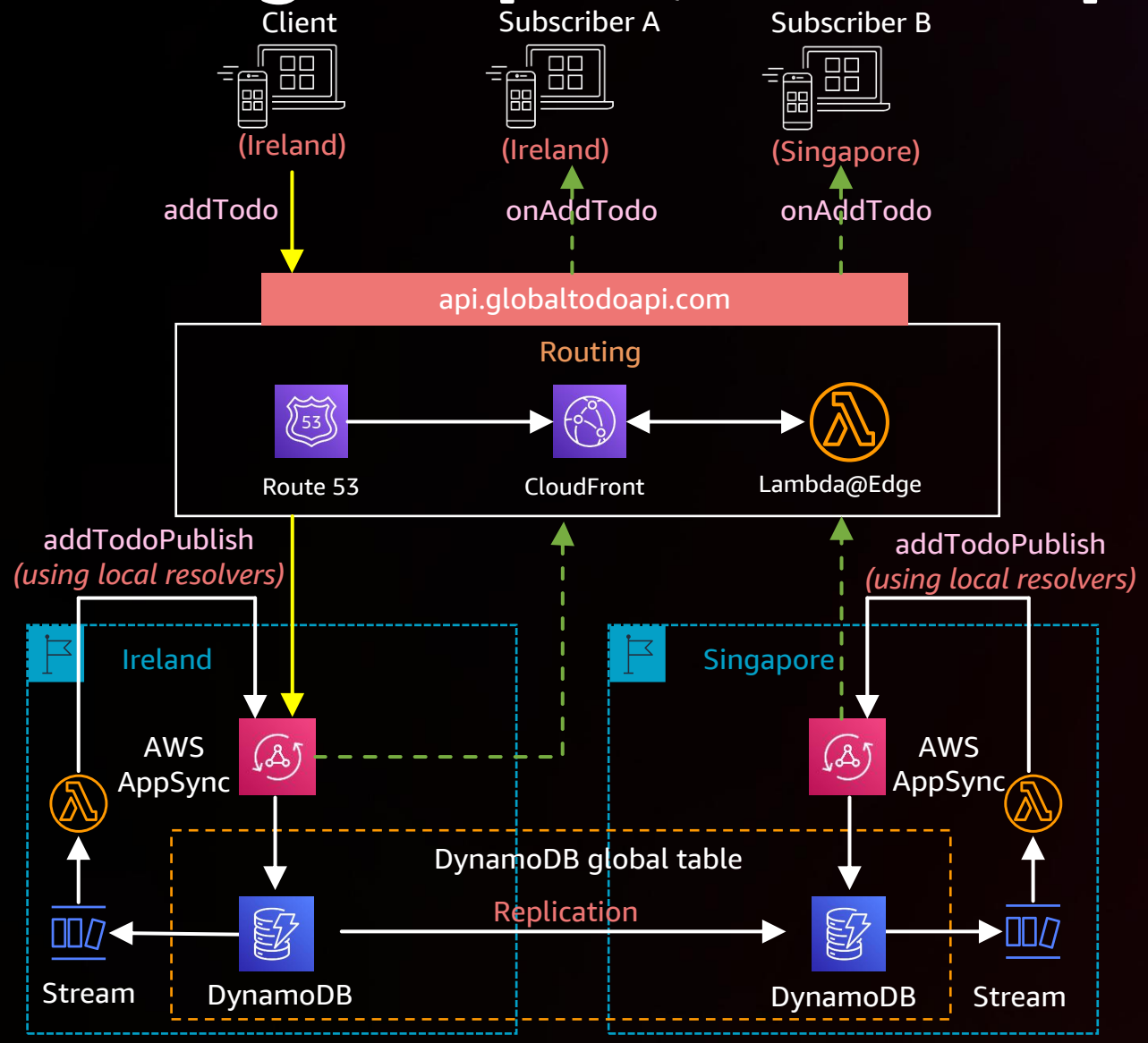
Enabling GraphQL subscription in multi-Region

USING AWS APPSYNC LOCAL RESOLVERS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enabling GraphQL subscription in multi-Region




```
type Todo {
   id: ID!
   name: String
   description: String
   priority: Int
   status: TodoStatus
}

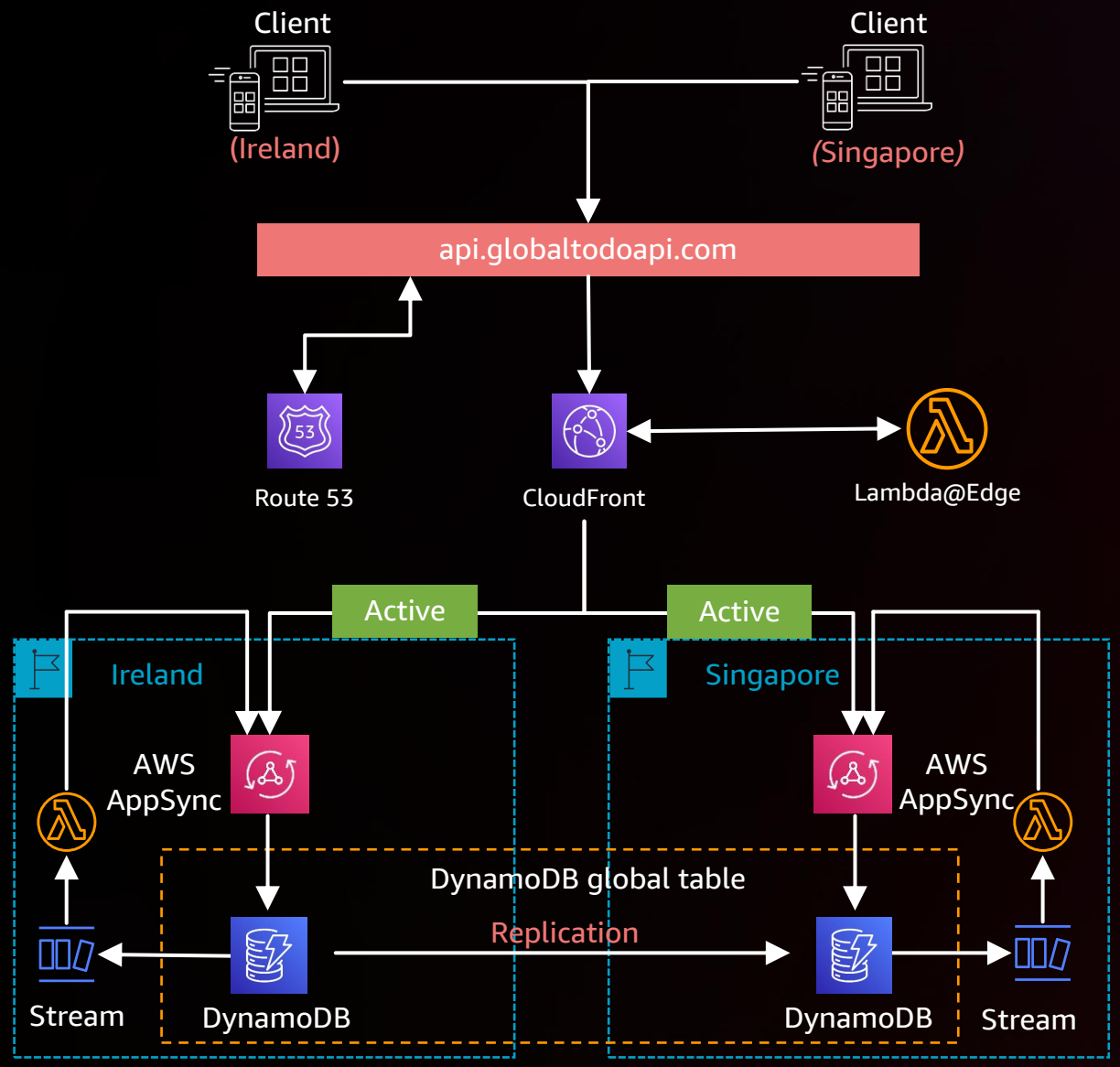
type Mutation {
   addTodo(input: AddTodo): Todo
   addTodoPublish(input: AddTodo): Todo

}

type Subscription {
   onAddTodo(input: AddTodo): Todo
   @aws_subscribe(mutations:["addTodoPublish"])
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Critical workload – Active-active pattern



Key features

- Single global API endpoint
- Supports queries, mutation, and subscriptions
- Enable subscription with data sources that provides change logs
- Configure once without any need to set up and maintain a failover process

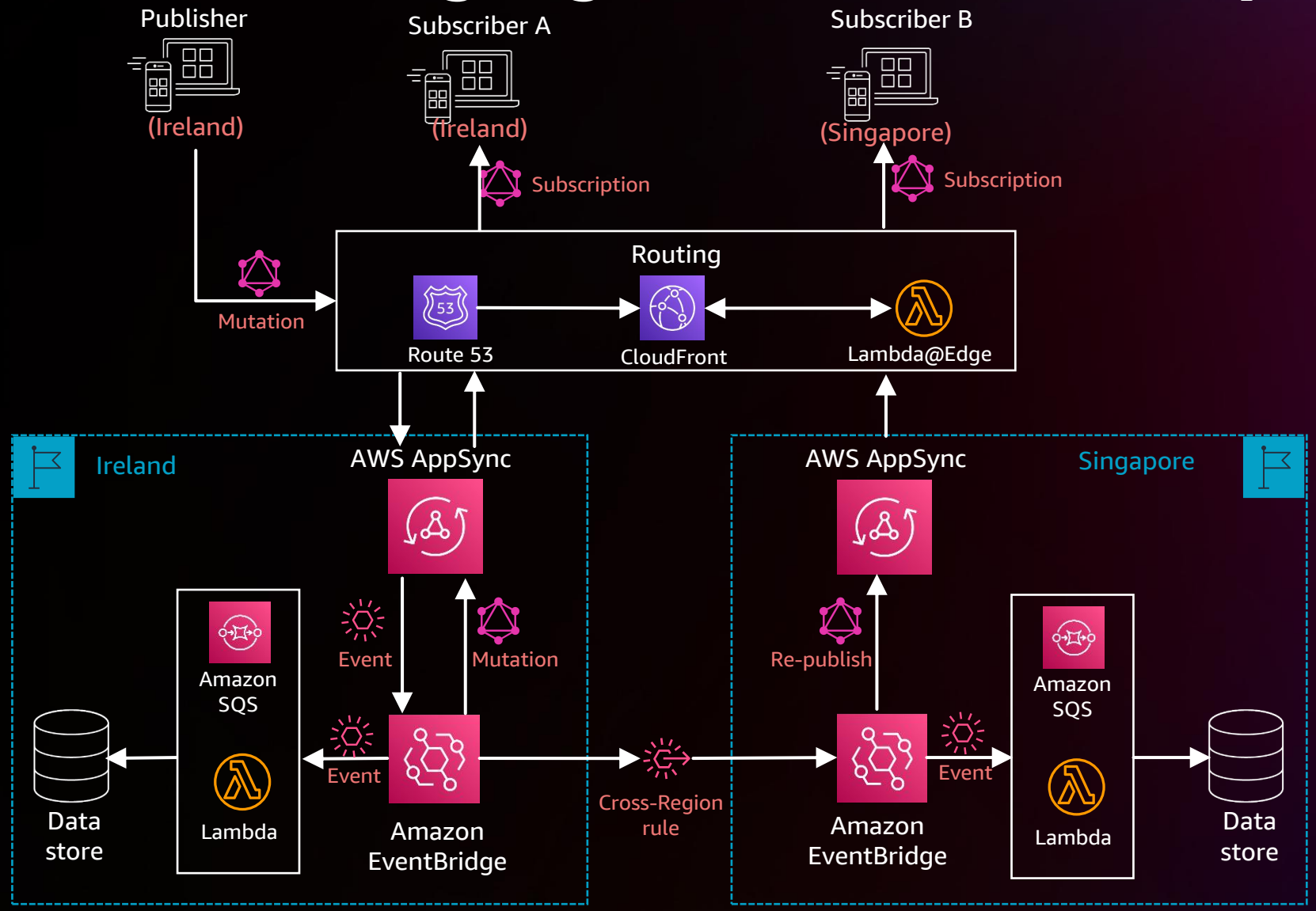
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case 3

Real-time messaging with WebSocket Pub/Sub APIs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Real-time messaging – Active-active pattern



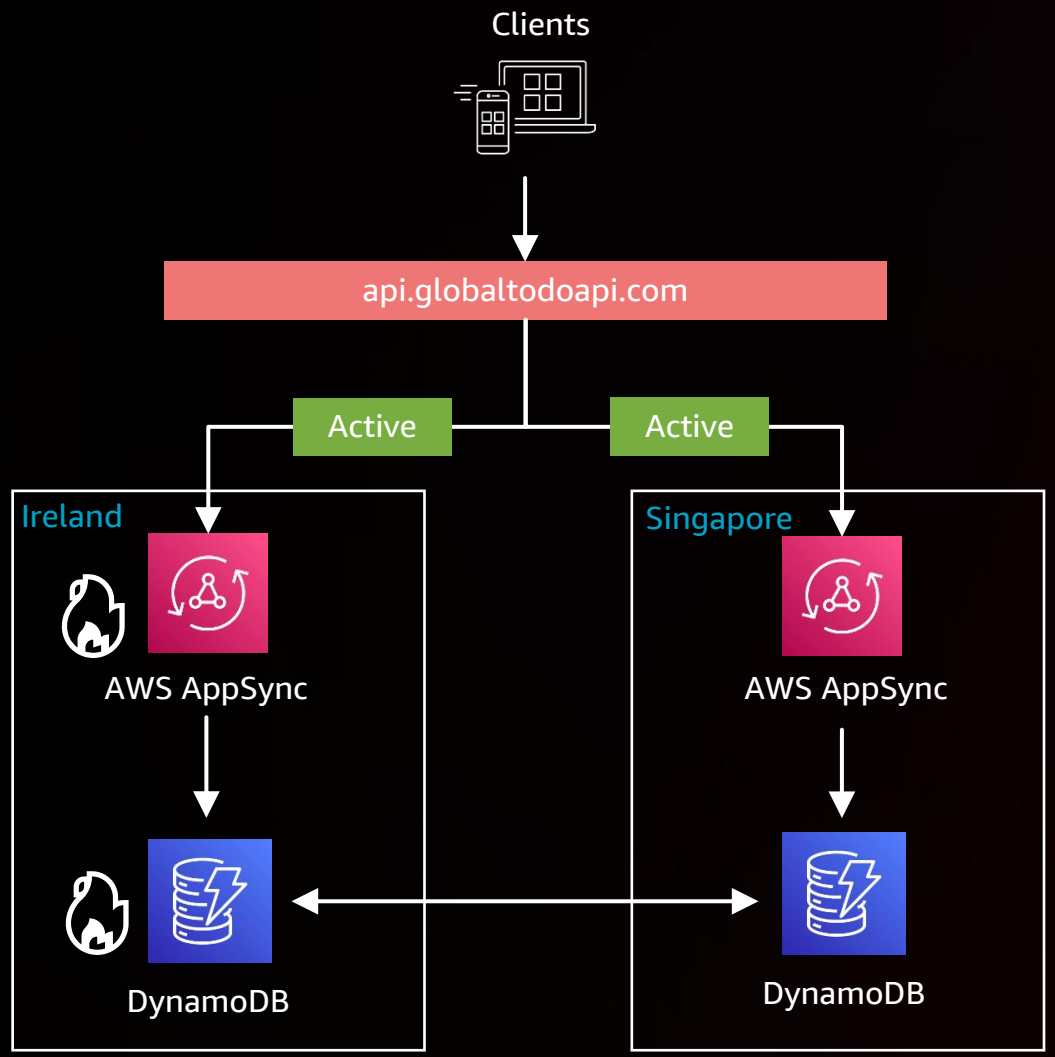
tinyurl.com/yezbrtb3

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Managing regional failures

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint

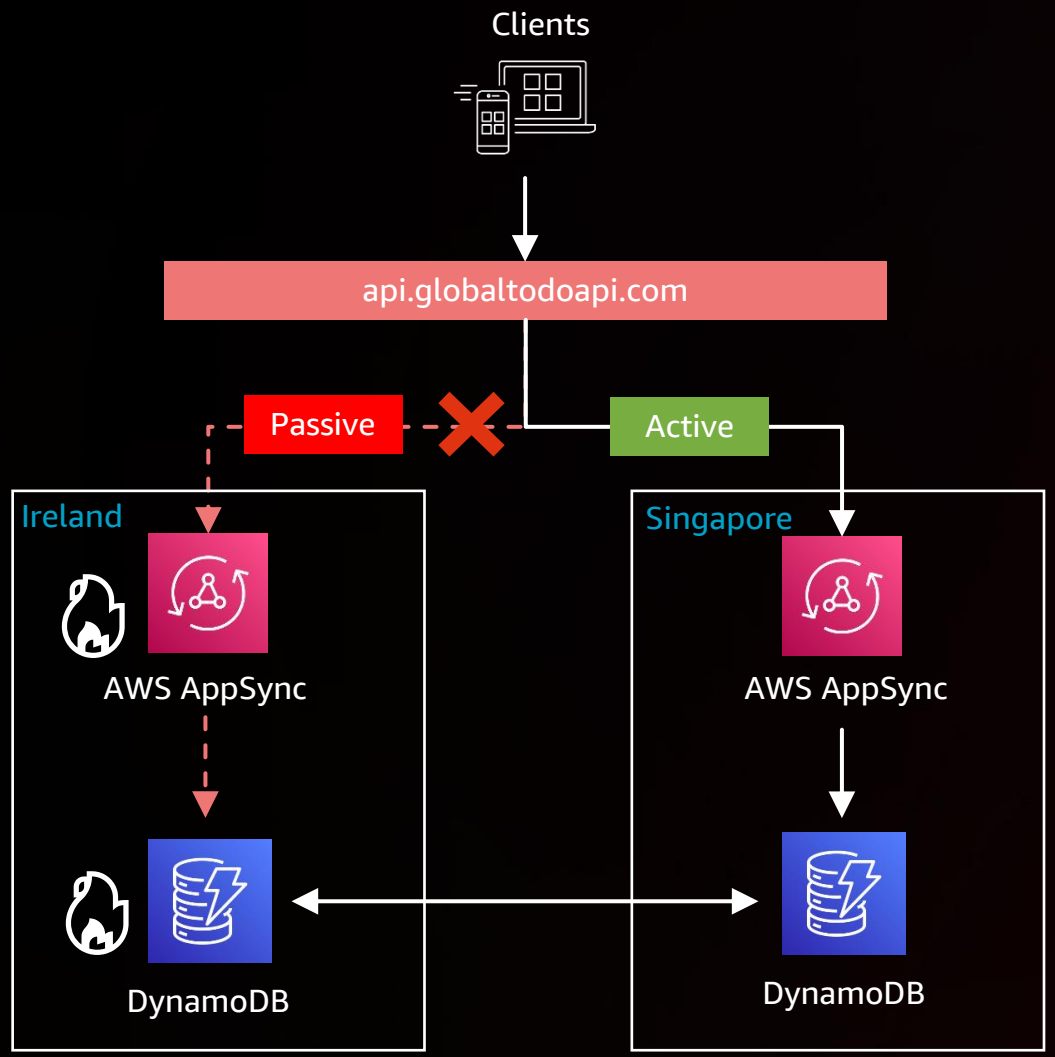


Sources of failure

- Prolonged AWS service outage
- Accidental deletion or changes to configuration
- Security incidents
- Exhausting your AWS service quotas

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint



Sources of failure

- Prolonged AWS service outage
- Accidental deletion or changes to configuration
- Security incidents
- Exhausting your AWS service quotas

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint

USING ROUTE 53 APPLICATION RECOVERY CONTROLLER (ARC) ROUTING CONTROL

tinyurl.com/28f535m9



Route 53 Application Recovery Controller

Gives you insights into whether your applications and resources are ready for recovery, and helps you manage and coordinate failover using **readiness check** and **routing control** features

aws

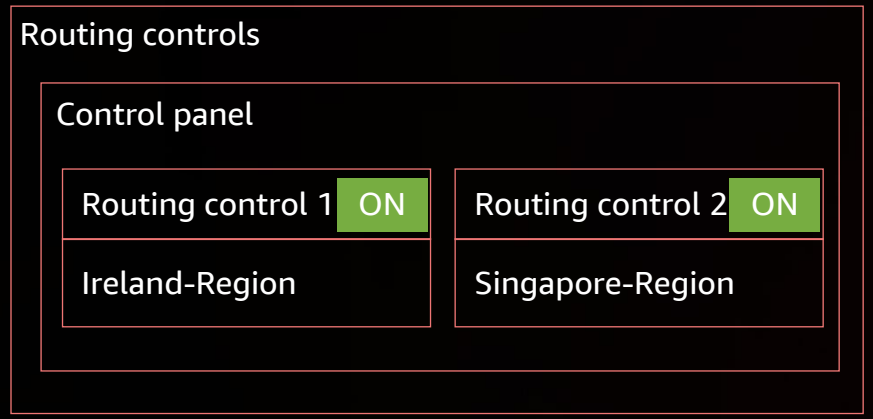
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint

USING ROUTE 53 APPLICATION RECOVERY CONTROLLER (ARC) ROUTING CONTROL

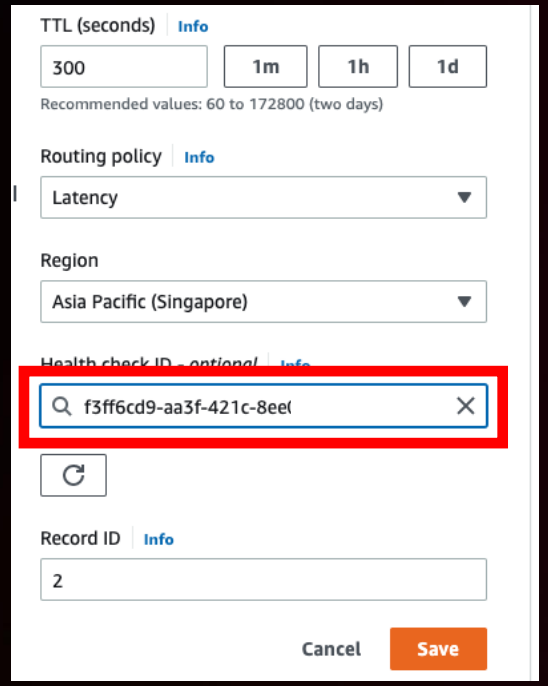
1

Set up routing control in ARC which will create a health check



2

Use routing control ARC health check for DNS routing



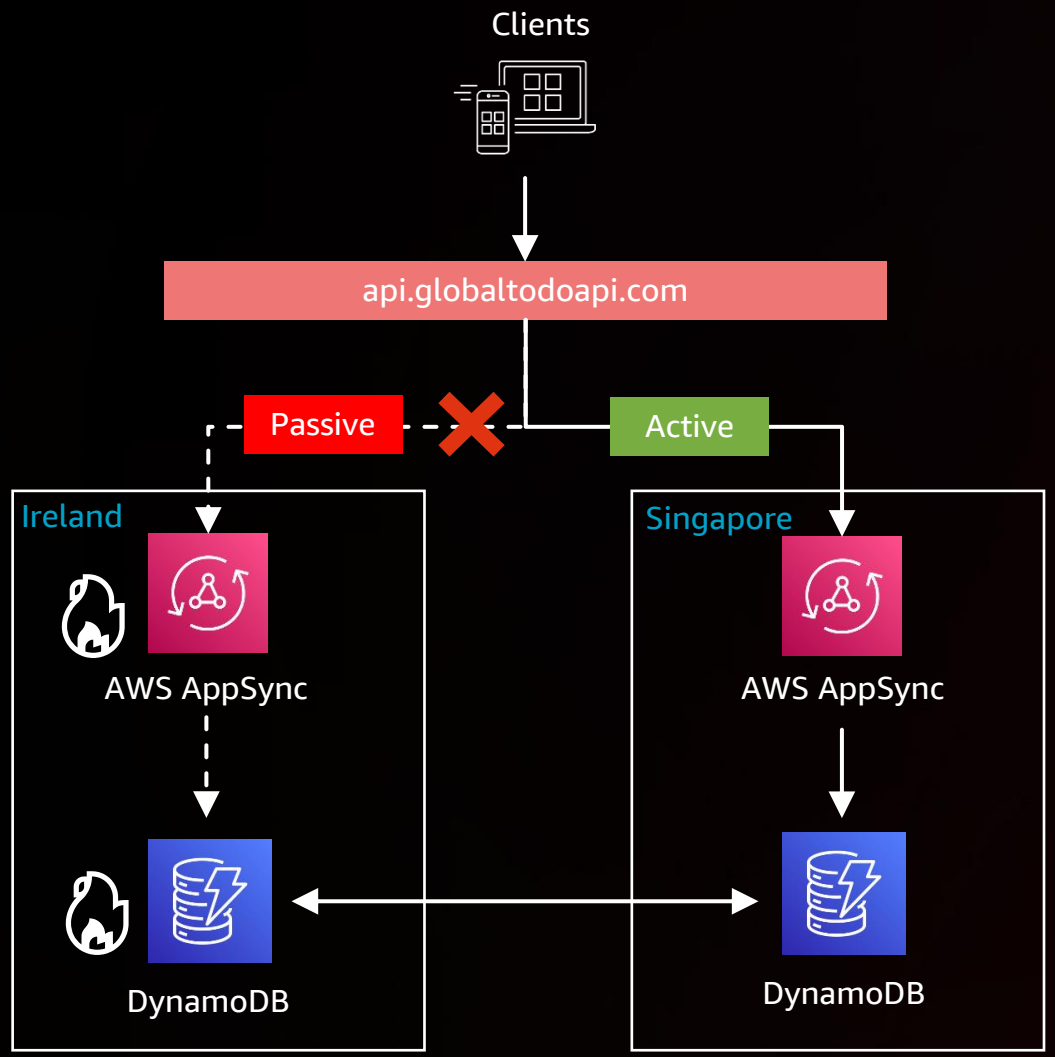
3

Turn routing control switch on/off to control traffic routing

```
aws route53-recovery-cluster \
    update-routing-control-state \
    --routing-control-arn <routing-control-arn>
    --routing-control-state On \
    --region us-west-2 \
    --endpoint-url <regional-cluster-endpoint>
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint



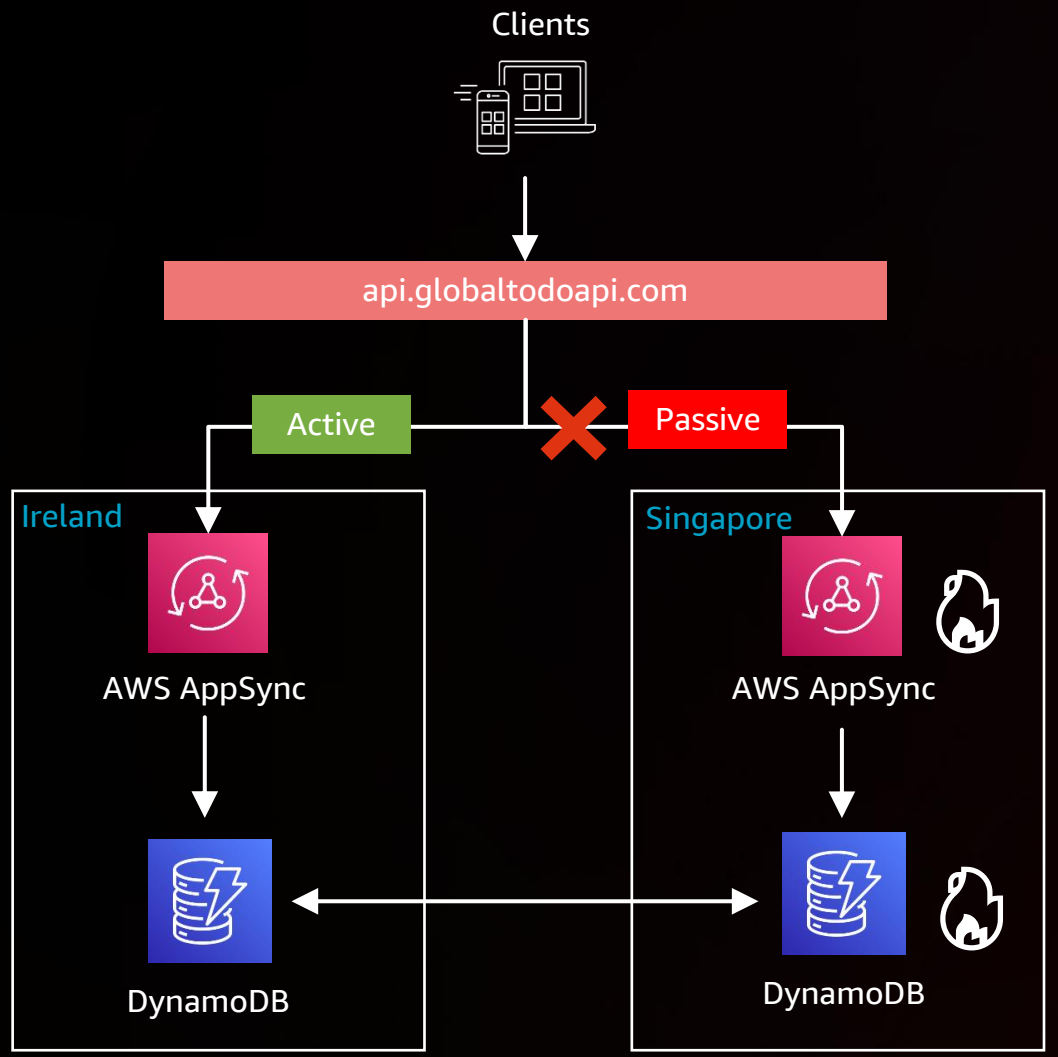
Routing controls

Control panel

Routing control 1	OFF	Routing control 2	ON
Ireland-Region		Singapore-Region	

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Routing to a healthy AWS AppSync endpoint



Routing controls

Control panel

Routing control 1	ON	Routing control 2	OFF
Ireland-Region		Singapore-Region	

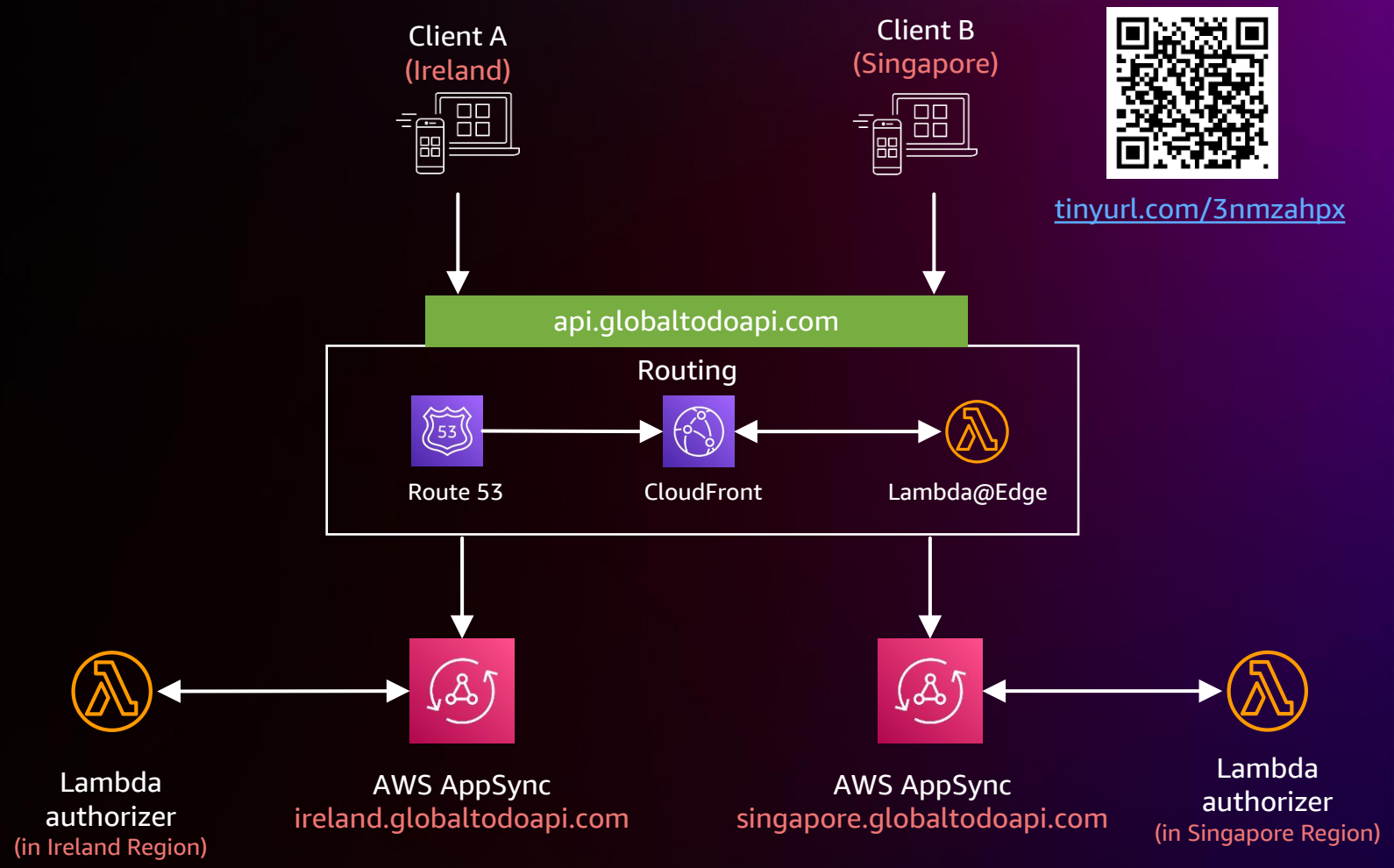
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-Region AWS AppSync API authorization

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-Region AWS AppSync API authorization

Authorization method	Multi-Region support?
API Key	✗
Lambda	✓
Amazon Cognito	✗
IAM	✓
OpenID Connect	?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

1. Deploy the architecture that meets your use case and application requirements
   e.g. Active/Active, Active/Passive, real-time messaging

2. Setup a single endpoint for your multi-region AWS AppSync API
   using Amazon CloudFront, AWS Lambda@Edge and Amazon Route 53

3. Route requests to healthy AWS AppSync endpoint to manage regional failures
   using Amazon Route 53 Application Recovery Controller

4. Ensure your authorization method is resilient to regional failures
   e.g., using IAM, AWS Lambda authorizer or OpenID connect providers

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources



Demo GitHub Repo
https://tinyurl.com/ud97z8cs
GitHub repository for the demo presented during the session



Blog Post – Multi Region deployment of AWS AppSync with Amazon DynamoDB Global Tables
https://tinyurl.com/2nvkzjr3
A blog post on multi-Region deployment of AWS AppSync



Reference architecture for multi-Region deployment of AWS AppSync
https://tinyurl.com/2cwvpxmc
The reference architecture for setting up multi-Region deployment of AWS AppSync



How to GraphQL on AWS
https://aws.amazon.com/graphql/
A set of blogs on how to build and deploy GraphQL API on AWS

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Ozioma Uzoegwu

@iam_tessot

Learn more and get started for free at aws.amazon.com/appsync/



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.