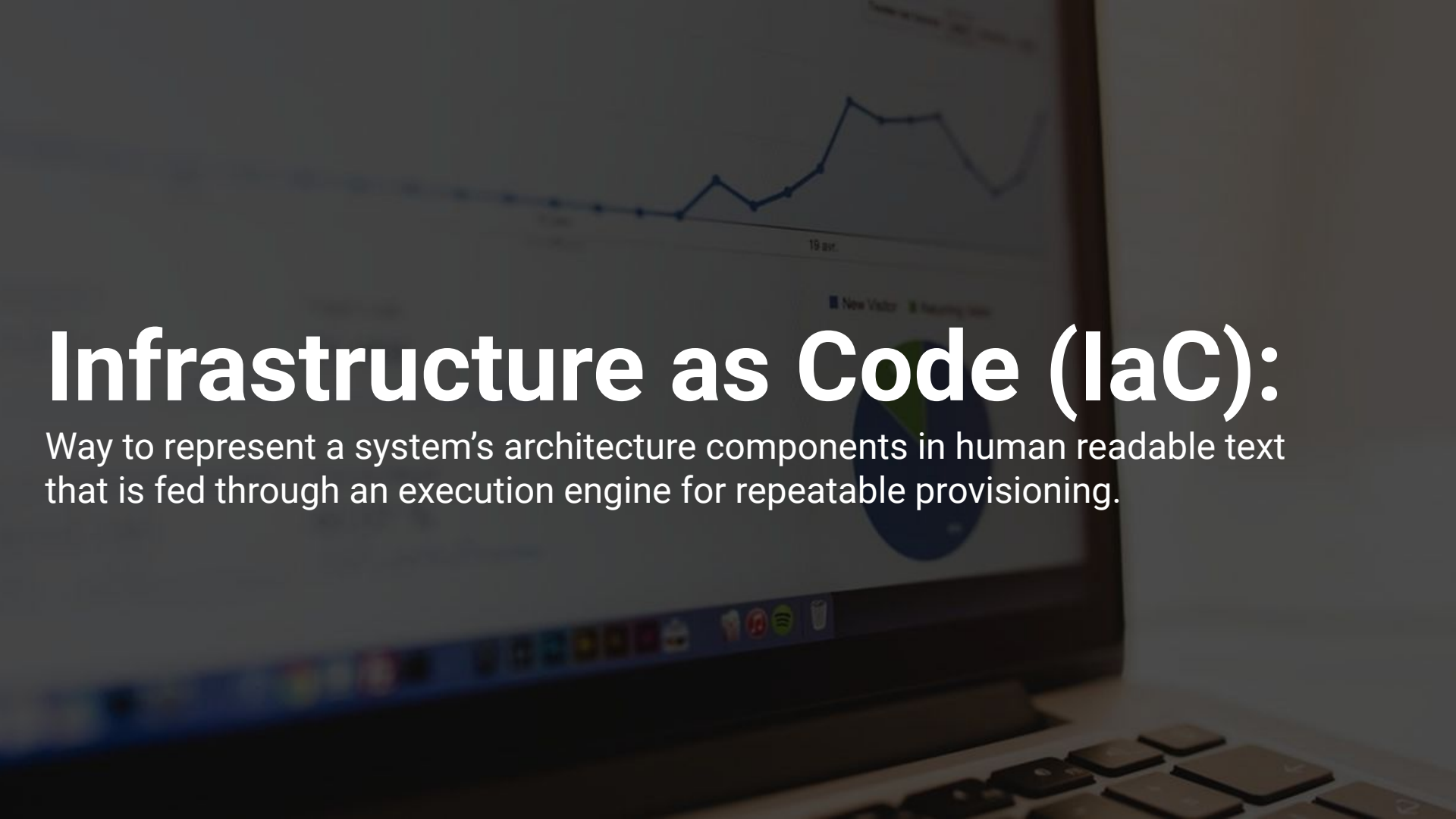
Modern Infrastructure as Code

AWS Cloud Development Kit (CDK)

Presented by Adam McQuistan



Infrastructure as Code (IaC):

Way to represent a system’s architecture components in human readable text that is fed through an execution engine for repeatable provisioning.

Benefits of IaC

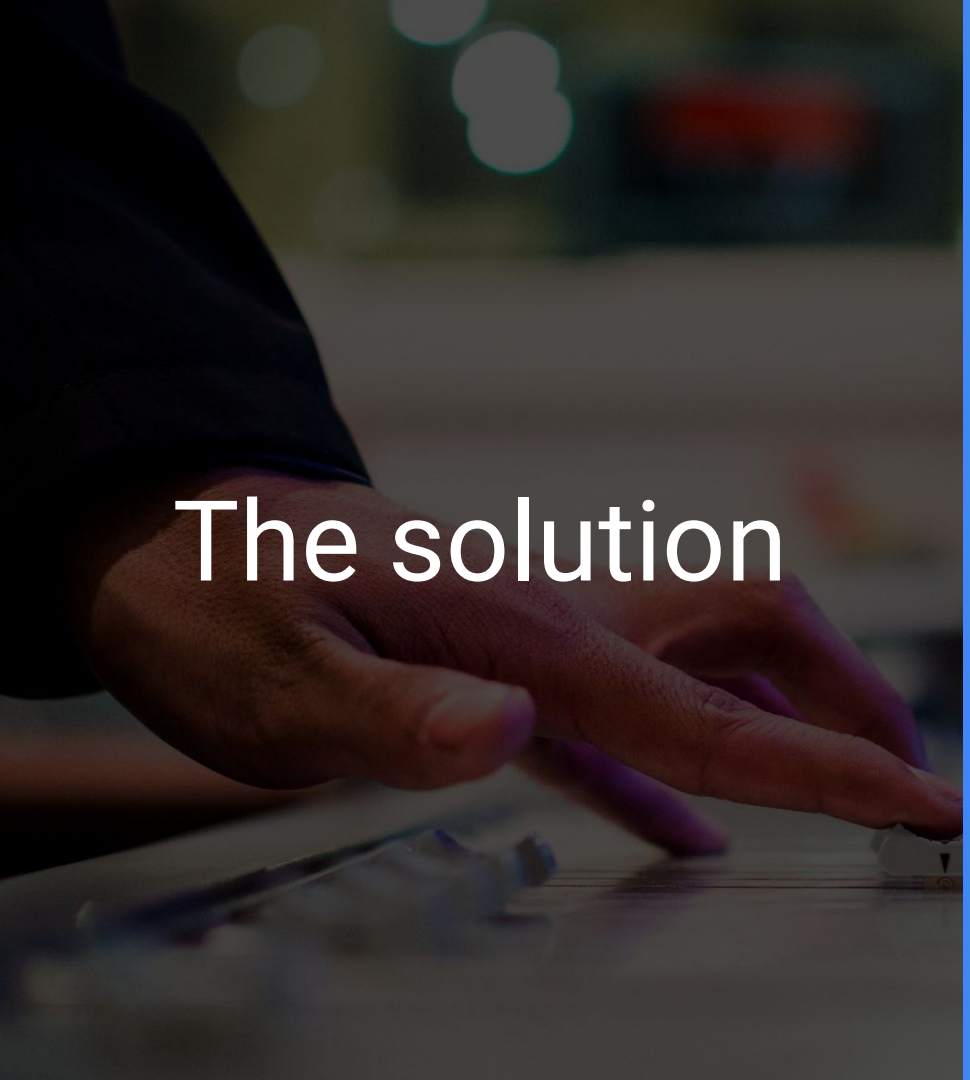
Draws from Software Engineering best practices

- Code as source of truth & documentation of Infrastructure
- Consistency via automation
- Accelerated environment provisioning
- Version controlled, auditable and peer reviewed

Challenges of IaC

Many technologies have successfully executed on the promises of IaC like Terraform, Ansible, Chef, Puppet, …

All require a Domain Specific Language (DSL) and specialized skills



The solution

Cloud Development Kit

- Abstraction over Cloud Formation with interfaces in common high-level programming languages and using OOP paradigm

AWS CDK

Technology Components

NodeJS based CLI utility

- Initialize CDK projects
- Synthesize CDK project to Cloud Formation Stacks
- Deploy synthesized stacks

Construct library

- OO classes representing cloud services & resources
- Base implementation in Typescript then translated to other languages via JSII (Python, Java, C#, Go, JavaScript)

Getting Started with CDK

Step 1

Install CDK CLI

```
npm install -g aws-cdk
```

Step 2

Initialize CDK App

```
cdk init app -l typescript
```

Step 3

Instantiate resource constructs

```
new s3.Bucket(this, "MyBucket");
```

Step 4

Build project

```
npm run build
yarn build
```

Step 5

Synthesize to Cloud Formation Stack(s)

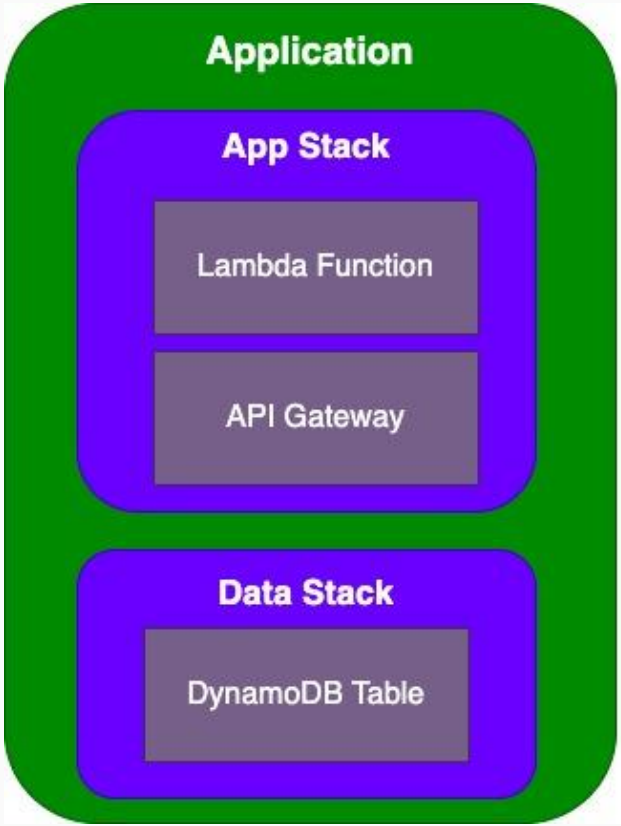
```
cdk synth
```

Step 6

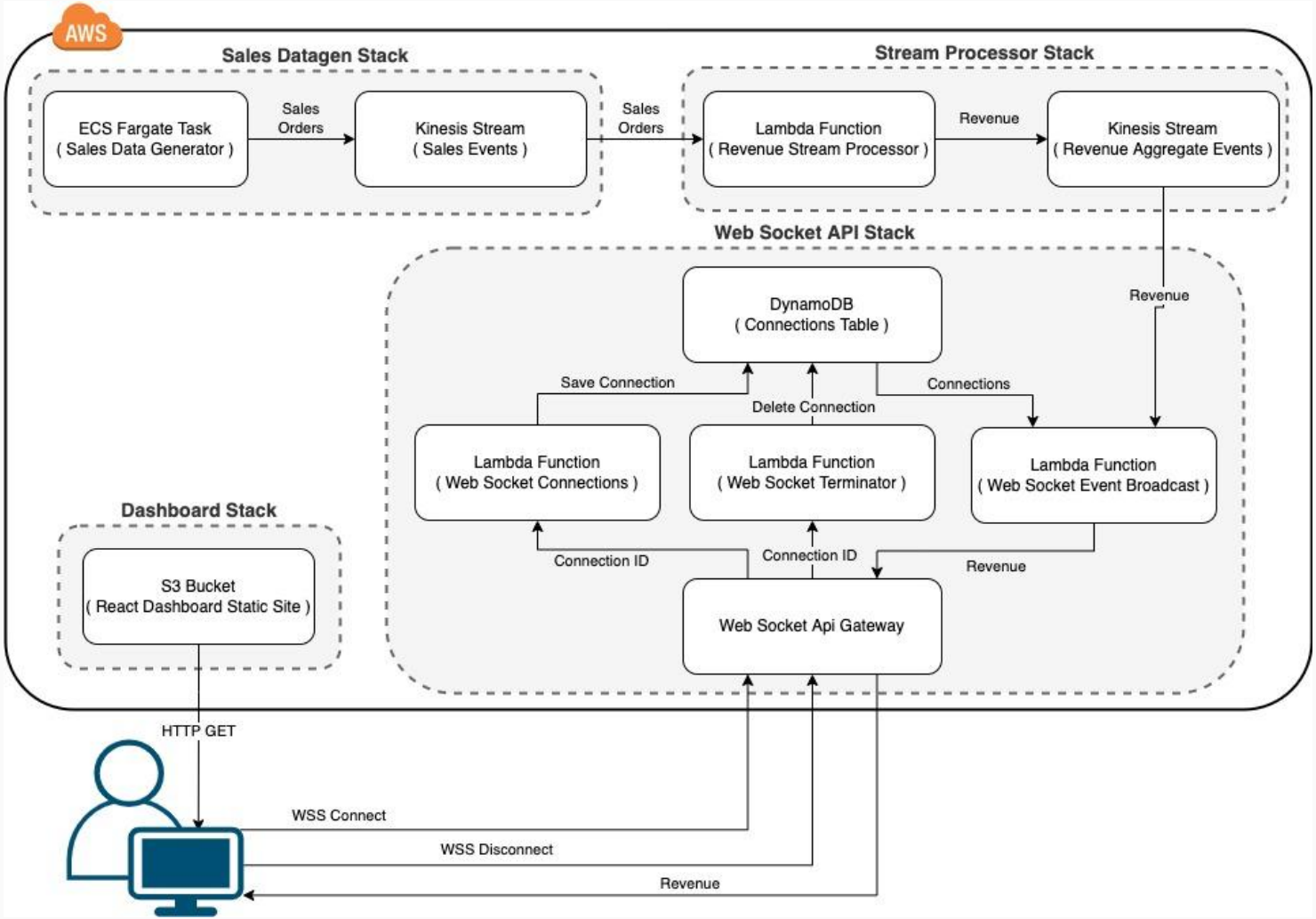
Deploy stacks

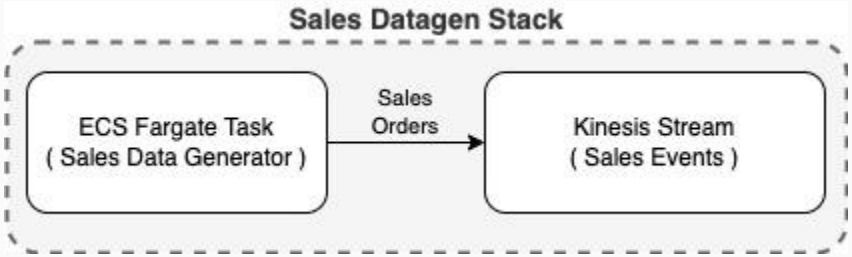
```
cdk deploy
```

AWS CDK Class Types and Relationships



AWS IaC Abstraction Hierarchy


AWS
Sales Datagen Stack
ECS Fargate Task
(Sales Data Generator)
Sales
Orders
Kinesis Stream
(Sales Events)
Sales
Orders
Stream Processor Stack
Lambda Function
(Revenue Stream Processor)
Revenue
Kinesis Stream
(Revenue Aggregate Events)
Web Socket API Stack
DynamoDB
(Connections Table)
Revenue
Save Connection
Delete Connection
Connections
Lambda Function
(Web Socket Connections)
Lambda Function
(Web Socket Terminator)
Lambda Function
(Web Socket Event Broadcast)
Connection ID
Connection ID
Revenue
Web Socket Api Gateway
Dashboard Stack
S3 Bucket
(React Dashboard Static Site)
HTTP GET
WSS Connect
WSS Disconnect
Revenue

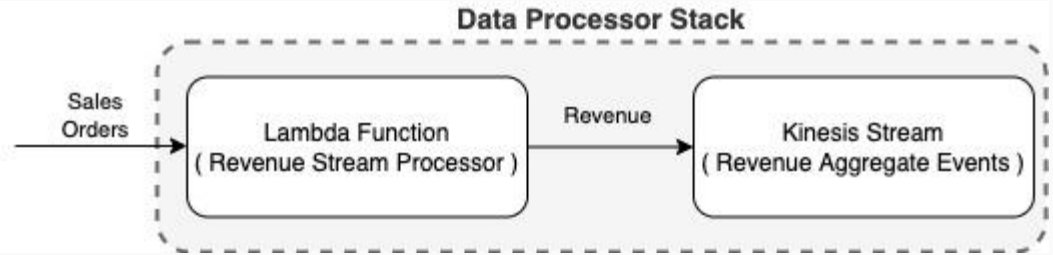


Constructs

- Stream (L2)
- Cluster (L2)
- Role (L2)
- FargateTaskDefinition (L2)
- FargateService (L2)

Deployed Resources

- Kinesis Stream
- VPC and Route Table
- 2 Public Subnets with Internet Gateway
- 2 Private Subnets with Nat Gateways
- Fargate Cluster
- ECS Fargate Task Definition
- ECS Fargate Service
- ECS Task with IAM Roles & Policies
- ECR Container Image
- CloudWatch Log Groups

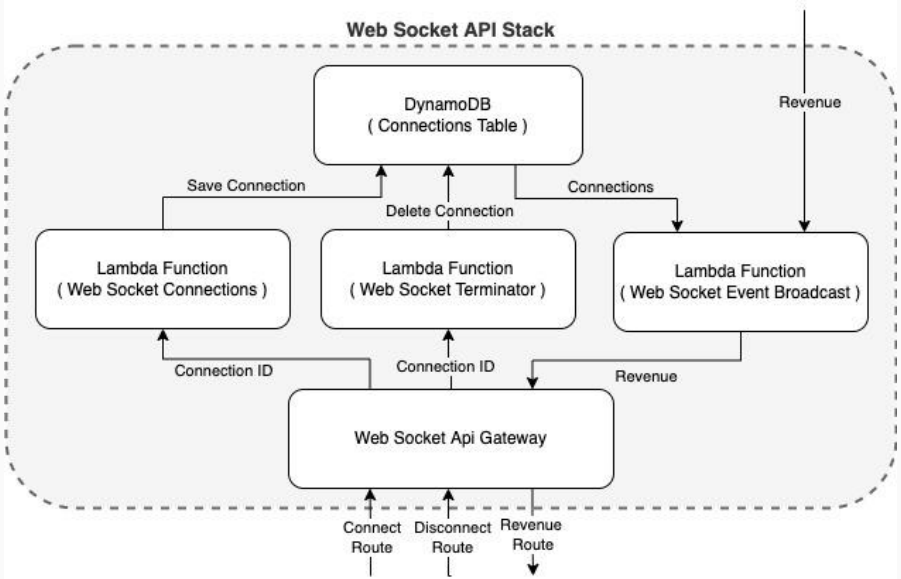


Constructs

- Stream (L2)
- NodejsFunction (L3)

Deployed Resources

- Kinesis Stream
- Lambda Function with IAM Role & Policy
- Lambda / Kinesis Event Source Mapping

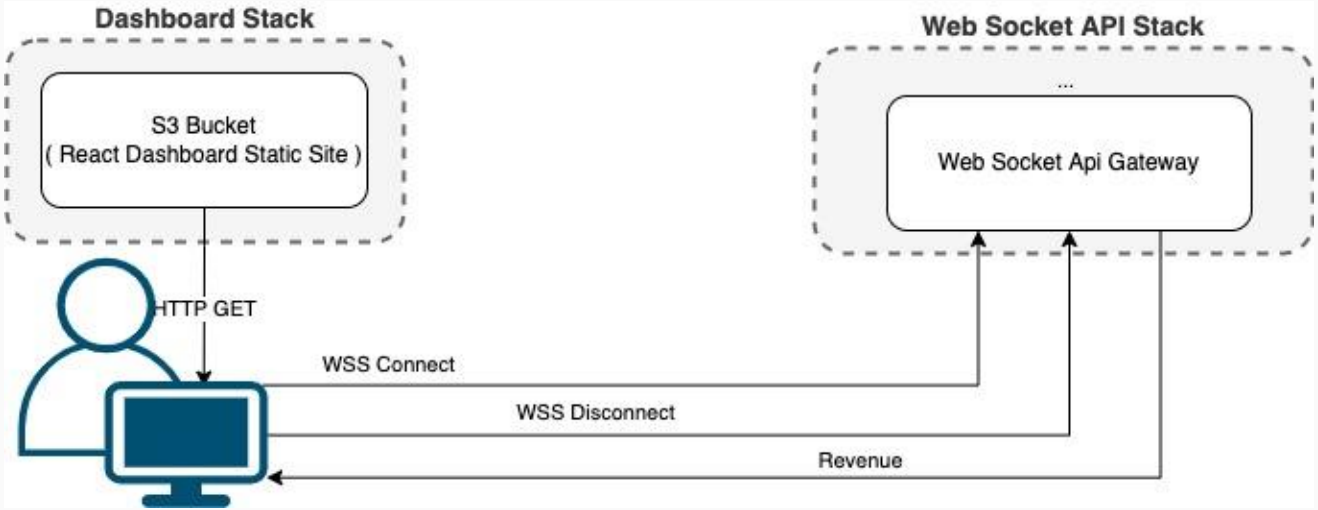


Constructs

- Table (L2)
- NodejsFunction (L3)
- WebsocketApi (Custom L2)
  - CfnApi
  - CfnDeployment
  - CfnStage
  - CfnIntegration
  - CfnRoute

Deployed Resources

- Dynamo DB Table
- Lambda Function with IAM Role & Policy (3)
- Lambda / Kinesis Event Source Mapping
- Api Gateway
- Api Gateway Deployment
- Api Gateway Route
- Api Gateway Integration



Constructs

- Bucket (L2)
- BucketDeployment (L3)

Deployed Resources

- S3 Bucket
- Lambda Function for Deploying Static Site

Q & A / Thank You!

Contact Info

https://www.linkedin.com/in/adammcquistan/

adam.mcquistan@thecodinginterface.com

Slides, Demo Code, and YouTube

https://github.com/amcquistan/nebcode-cdk

https://youtu.be/6yYpuVXWoPk

AWS CDK Docs

https://docs.aws.amazon.com/cdk/api/v2/