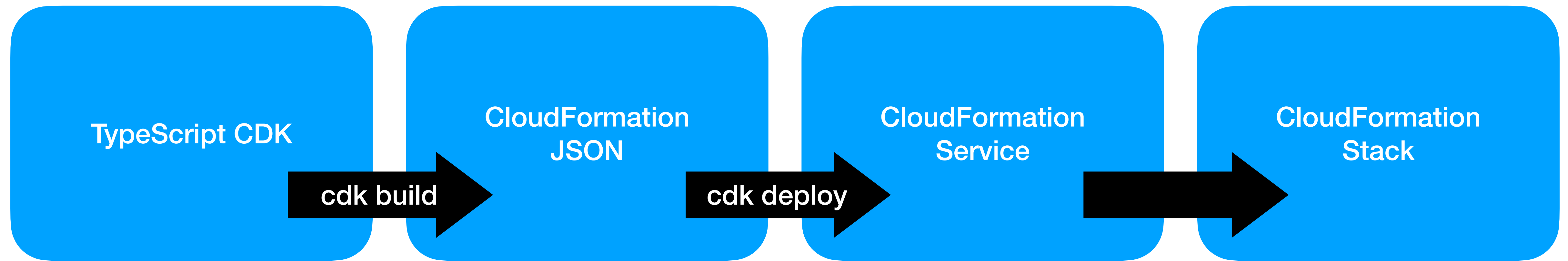


# Infrastructure as Code with AWS CDK and TypeScript

Ted Patrick

# Use the TypeScript CDK

All language SDKs are generated from TypeScript CDK



The key to this workflow is identity.

Name construct instances **very carefully!!**

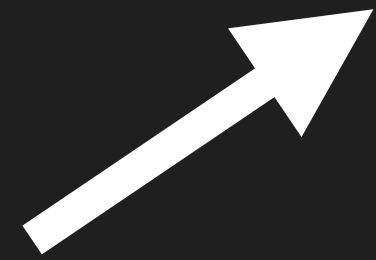
Example: Changing a construct instance name will force it to delete and recreate it.

```
import * as s3 from "aws-cdk-lib/aws-s3";
```

```
const mr_s3_transcripts_prod = new s3.Bucket(  
  this,  
  "mr-s3-transcripts-prod",  
  {  
    bucketName: "mr-s3-transcripts-prod",  
    removalPolicy: cdk.RemovalPolicy.RETAIN  
  }  
);
```

```
import * as s3 from "aws-cdk-lib/aws-s3";
```

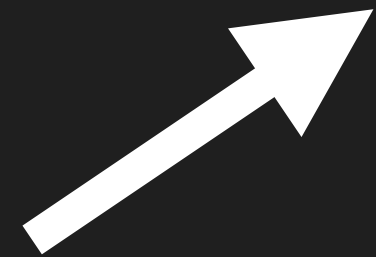
Construct Import



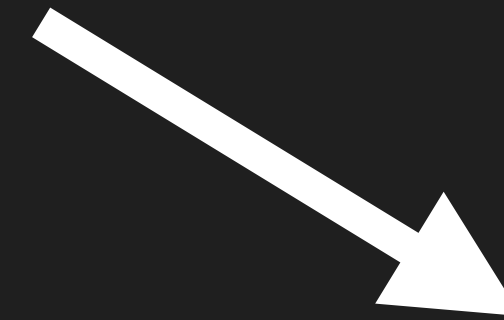
```
const mr_s3_transcripts_prod = new s3.Bucket(  
  this,  
  "mr-s3-transcripts-prod",  
  {  
    bucketName: "mr-s3-transcripts-prod",  
    removalPolicy: cdk.RemovalPolicy.RETAIN  
  }  
);
```

```
import * as s3 from "aws-cdk-lib/aws-s3";
```

Construct Import



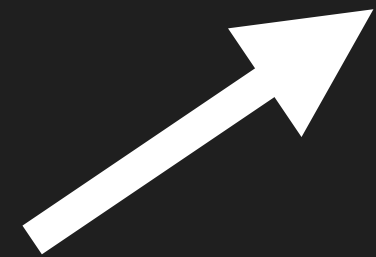
Construct Class



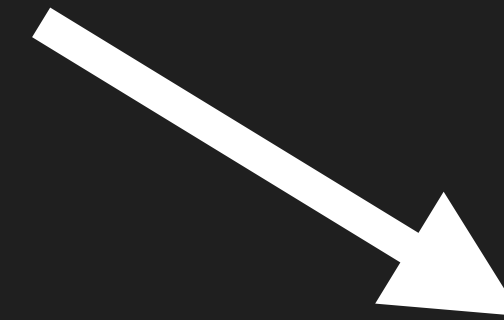
```
const mr_s3_transcripts_prod = new s3.Bucket(  
  this,  
  "mr-s3-transcripts-prod",  
  {  
    bucketName: "mr-s3-transcripts-prod",  
    removalPolicy: cdk.RemovalPolicy.RETAIN  
  }  
);
```

```
import * as s3 from "aws-cdk-lib/aws-s3";
```

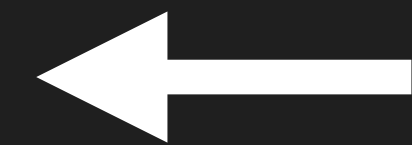
Construct Import



Construct Class



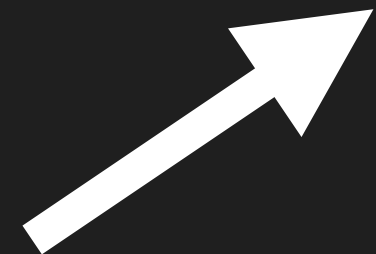
```
const mr_s3_transcripts_prod = new s3.Bucket(  
  this,  
  "mr-s3-transcripts-prod",  
  {  
    bucketName: "mr-s3-transcripts-prod",  
    removalPolicy: cdk.RemovalPolicy.RETAIN  
  }  
);
```



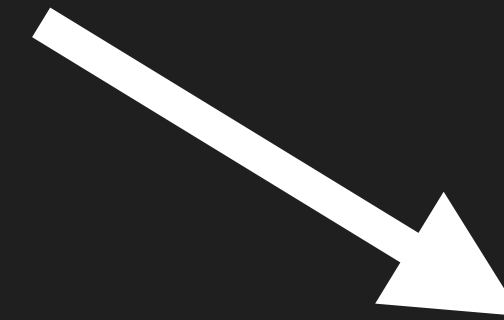
Construct Identity

```
import * as s3 from "aws-cdk-lib/aws-s3";
```

Construct Import

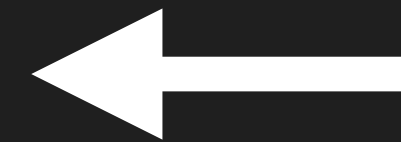


Construct Class



```
const mr_s3_transcripts_prod = new s3.Bucket(  
  this,  
  "mr-s3-transcripts-prod",
```

Construct Identity



Construct Data



```
{  
  bucketName: "mr-s3-transcripts-prod",  
  removalPolicy: cdk.RetentionPolicy.RETAIN  
}  
);
```



# DynamoDB Table

```
const mr_ddb_prod = new dynamodb.Table(this, "mr-ddb-prod", {
  partitionKey: { name: "pk", type: dynamodb.AttributeType.STRING },
  sortKey: { name: "sk", type: dynamodb.AttributeType.STRING },
  timeToLiveAttribute: "ttl",
  tableName: "mr-ddb-prod",
  billingMode: cdk.aws_dynamodb.BillingMode.PAY_PER_REQUEST,
  removalPolicy: cdk RemovalPolicy.RETAIN,
  deletionProtection: true,
});
```

# CloudWatch LogGroup

```
const eventsLogGroup = new logs.LogGroup(this, "mr-log-group-events", {  
    logGroupName: `/mr-events`,  
    retention: logs.RetentionDays.ONE_YEAR,  
    removalPolicy: cdk.RemovalPolicy.RETAIN,  
});
```

# Eventbridge EventBus & Rule

```
const mr_prod_event_bus = new events.EventBus(this, "mr-prod-event-bus", {
  EventBusName: "mr-prod",
});

new events.Rule(this, `mr-sqs-report-rule`, {
  EventBus: mr_prod_event_bus,
  eventPattern: {
    source: ["mr"],
    detailType: [
      "mr.report.summary",
      "mr.report.action_items",
      "mr.report.chapters",
      "mr.report.question",
    ],
  },
  targets: [new targets.SqsQueue(mr_prod_report_sqs)],
});
```

# Aurora Postgres RDS Cluster (reader/writer)

```
// Create an Aurora Serverless RDS Cluster
const db2 = new rds.DatabaseCluster(this, 'mr-db-2', {
  engine: rds.DatabaseClusterEngine.auroraPostgres({ version: rds.AuroraPostgresEngineVersion.VER_15_5 }),
  writer: rds.ClusterInstance.serverlessV2('writer'),
  credentials: rds.Credentials.fromUsername('adminuser', { password: DB_ROOT_PASSWORD }),
  readers: [
    rds.ClusterInstance.serverlessV2('reader', {
      scaleWithWriter: true,
    }),
  ],
  storageEncrypted: true,
  securityGroups: [rdsSecurityGroup],
  serverlessV2MinCapacity: Number( INFRA_SCALE_MIN_DB ),
  serverlessV2MaxCapacity: Number( INFRA_SCALE_MAX_DB ),
  vpcSubnets: {
    subnetType: ec2.SubnetType.PRIVATE_WITH_EGRESS,
  },
  vpc,
  backup: {
    retention: cdk.Duration.days(7),
  }
});
```

## References to existing infra

```
const domainCert = new apigv2.DomainName(  
  this,  
  'ac-app-events-acm-cert',  
  {  
    domainName: `events.${NETWORK_DOMAIN}`,  
    certificate: acm.Certificate.fromCertificateArn(  
      this,  
      'cert',  
      APP_EVENT_ACM_ARN  
    ),  
  }  
);
```

Docs: <https://docs.aws.amazon.com/cdk/api/v2/>

But... The *best* docs are here:

[https://github.com/aws/aws-cdk/tree/main/  
packages/aws-cdk-lib](https://github.com/aws/aws-cdk/tree/main/packages/aws-cdk-lib)