

Building and deploying 3-tier Web Applications with AWS CDK

Basil Udoudoh
Head of Technology – Scaleup

About Me

- Hey y'all, I'm Basil Udoudoh, the Head of Technology of Scaleup, a startup founded on helping diverse-owned enterprise suppliers scale to \$100 million in revenue and beyond.
- I'm an experienced technology leader with over 16 years of leading dynamic engineering team and creating scalable and secure web, mobile , and Cloud-based applications on AWS and GCP
- In my spare time you can find me trying to keep up with my lovely wife and kids, tinkering with any number of side technology projects, and rooting for every Atlanta sports team!

About Y'all

- How many of you fall into the following categories?
 - Software Engineers/Architects
 - DevOps Engineers
 - SREs
 - Cloud Engineers/Architect/Administrators
- What Cloud Platforms do you work with?
 - AWS
 - GCP
 - Azure
 - Oracle
- What Infrastructure as Code Tools do you currently use?
 - CloudFormation
 - Terraform

Let's get started!

OK good intro, now what is AWS CDK?

AWS CDK, or **Cloud Development Kit** is an IAC abstraction layer on top of **Amazon CloudFormation** that allows you to define and provision cloud infrastructure using popular programming languages instead of configuration.

Supported Languages



So, Why should I use it?

AWS CDK brings the following (and more) benefits to your infrastructure provisioning pipeline/processes:

- Programmatic Logic
- Uniformity with rest of codebase
- Robust development/management tools
- Testability
- Modularity
- I/O

Sounds good! How does CDK work?

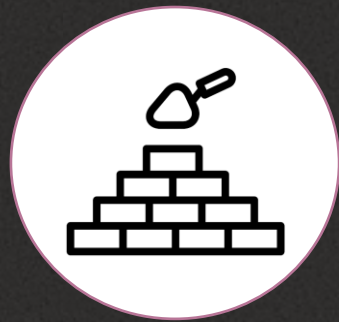
- CDK creates multiple layers of abstraction to model cloud resources
 - Constructs
 - Stacks
 - Apps
- Those resources, once modeled, are then **transpiled** into CloudFormation templates, and deployed into an AWS account.

Abstraction Layer 1 – Construct

- Constructs are the basic building block of CDK. They represent one or more cloud service resource and can have varying levels of abstraction to help provision resources easier.
- Think discrete services like **S3 buckets** and **EC2 instances**
- In CDK, constructs are pre-provisioned classes that can be imported into a project for use.
- A CDK Project can have one or more constructs (as many as you need for as many services are in your cloud stack)

Abstraction Layer 1 – Construct

There are 3 types of Constructs:



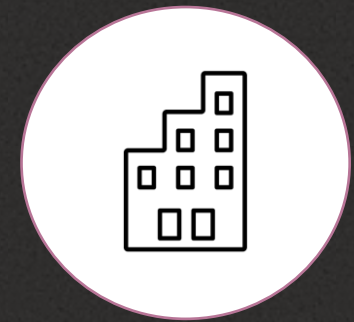
Level 1 Constructs

Also known as **CFN Resources**, these are lower-level constructs of individual resources that basically mirror CloudFormation resources (no abstraction at all)



Level 2 Constructs

Also known as **Curated Constructs**, these are single resources, thoughtfully pre-configured by the CDK team to match common defaults and best security practice settings.

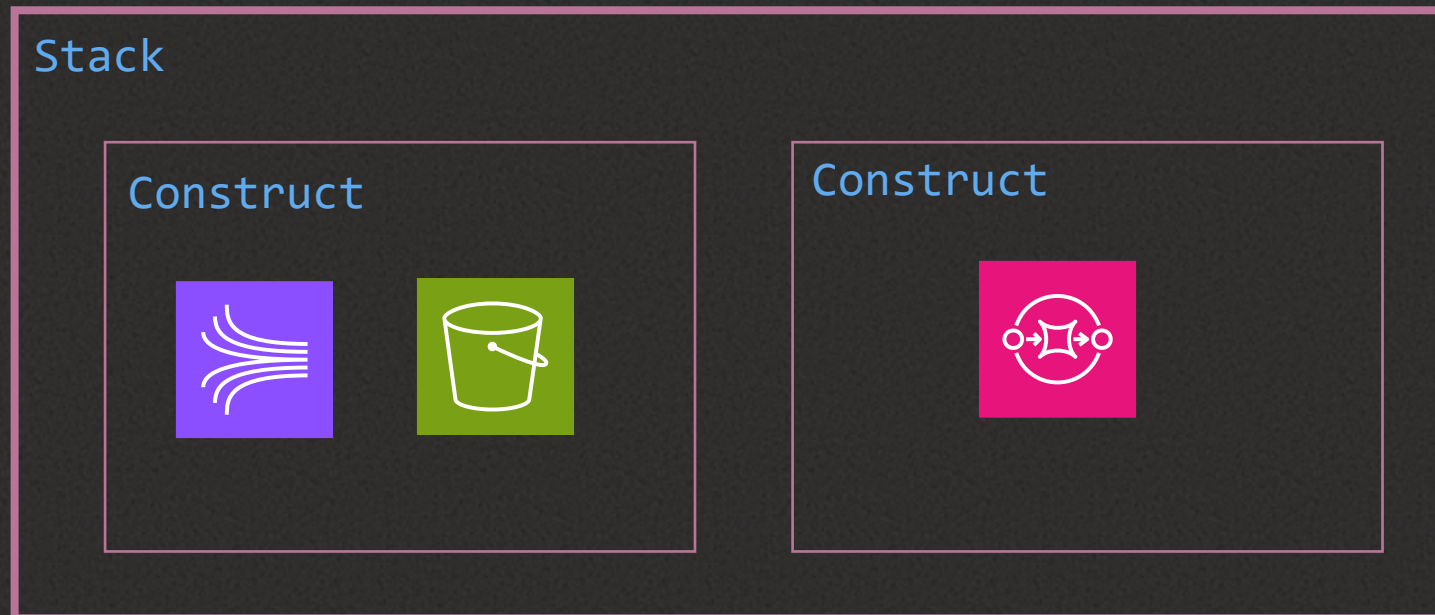


Level 3 Constructs

Also known as **Patterns**, these are higher-level abstractions of one or more resources, created by the CDK team, the Open-Source Community, or you....

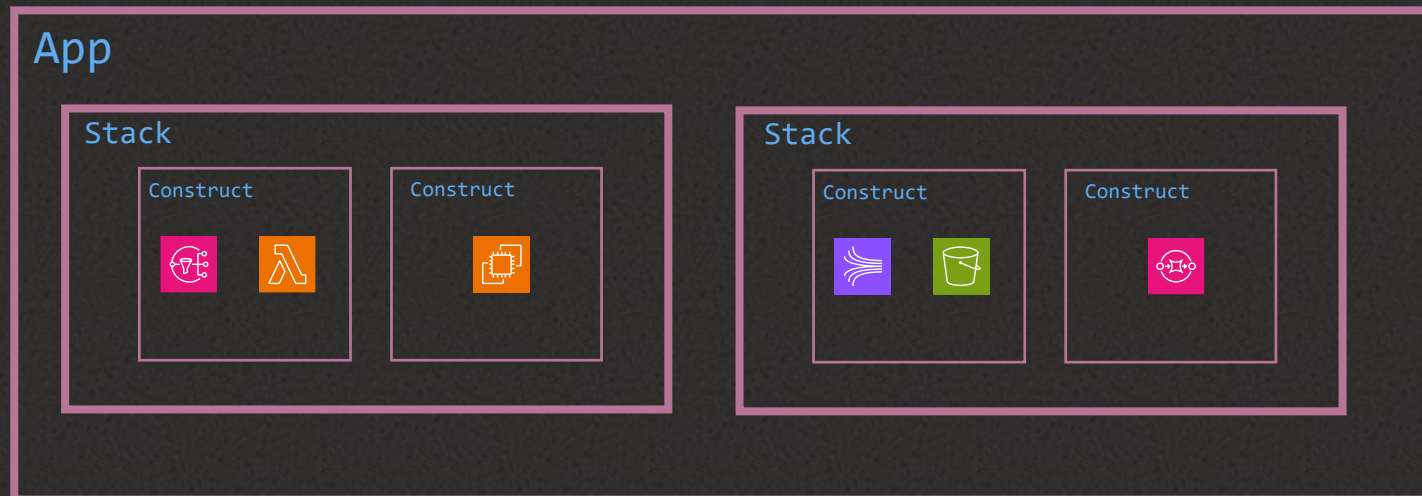
Abstraction Layer 2 – Stacks

- A CDK Stack is a collection of one or more constructs that are provisioned and deployed together in CloudFormation.
- A stack is managed as a single unit, allowing you to deploy, sync, and rollback several constructs at the same time
- A CDK Project can have one or more stacks



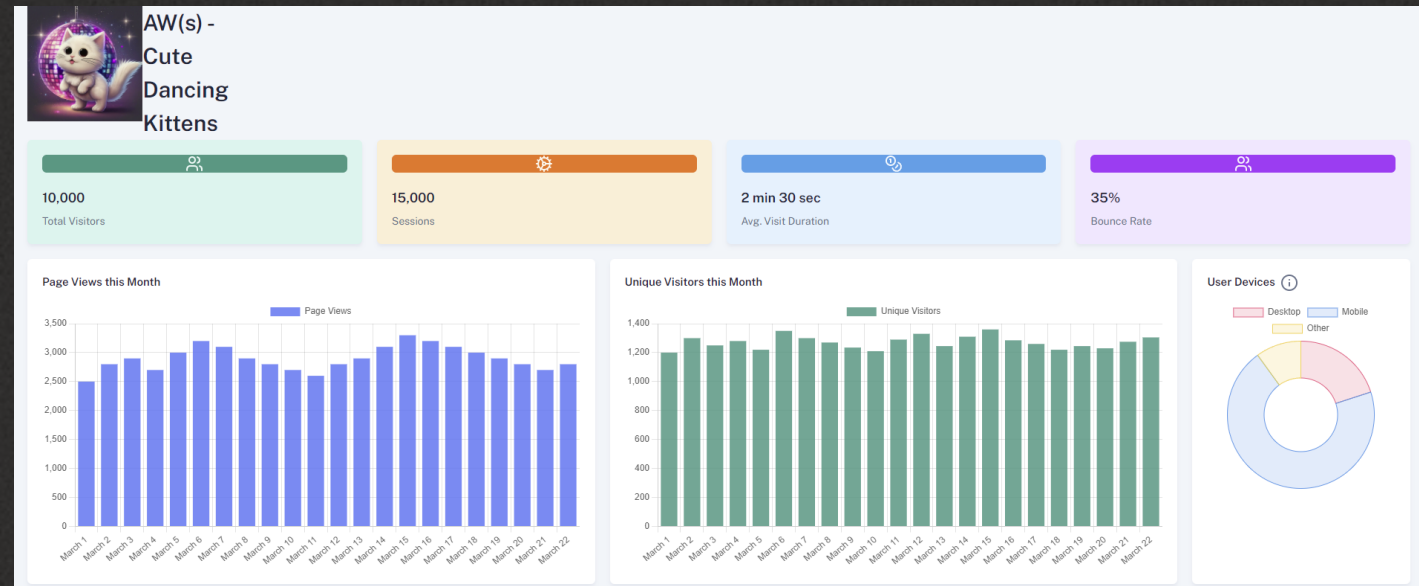
Abstraction Layer 3 – Apps

- A CDK App is a collection of one or more stacks that are provisioned and deployed together in CloudFormation.
- Apps allow stacks to be deployed interdependently – they can refer to each other's constructs and those constructs will be deployed in correct order to maintain this interdependence
- A CDK Project can have only one app

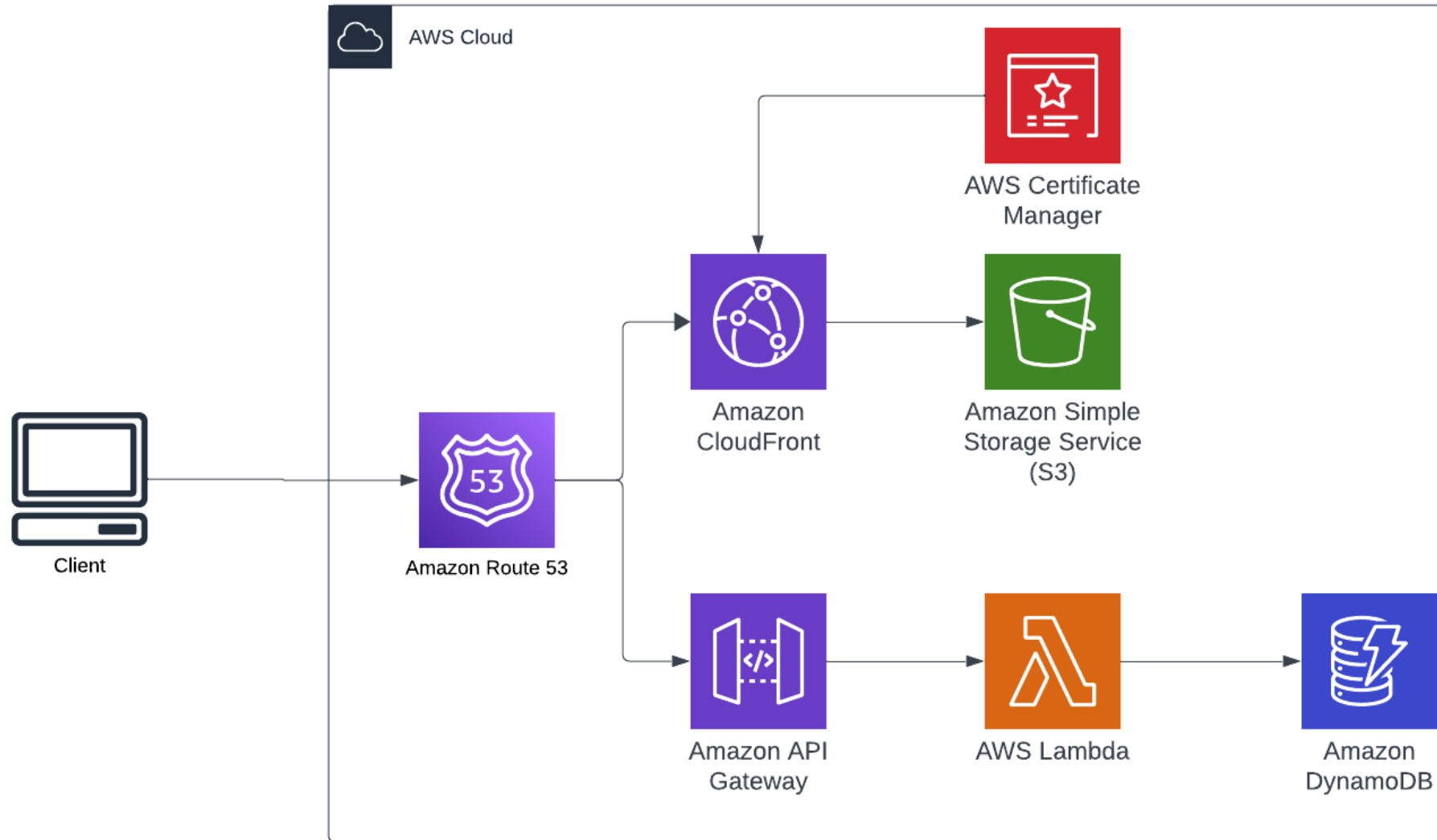


Let's Build Something!

The Other CDK – Cute Dancing Kittens!



Cute Dancing Kittens! – Our Architecture



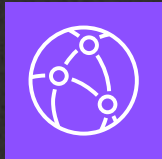
Architectural Components



Amazon Route 53 provides DNS resolution for web app and API layer



AWS Lambda provides serverless compute resources for our API requests



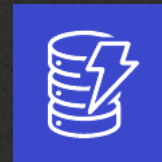
Amazon CloudFront is a CDN that provides fast, global delivery of web files.



Amazon S3 provides object storage, which can also be used to host a static website.



Amazon API Gateway handles API (in this case, REST) communication between clients and underlying resources.



DynamoDB is a NoSQL (Key/Value) database that we will be using to store dancing kitty analytics data



AWS Certificate Manager provisions and manages SSL/TLS certificates, which allow secure connection to our web application/API.

Let's look at the code!

Checkout the finished product at
<https://aws-cute-dancing-kittens.com/>

Enough kittens! How do I get started with CDK?

- CDK has the following requirement:
 - NodeJS (14.15.0 or later, no matter the language)
 - AWS CLI and programmatic access to your AWS Account
 - AWS CDK CLI (can be downloaded via Node Package Manager)
- Once the CLI (the CDK one) is installed, you can create a CDK project in the language of your choice from the command line

```
$ cdk init app --language typescript
```

Important Points on CDK

- CDK has support for a wide array of commonly used languages
- It brings the benefits of modern programming languages into infrastructure provisioning
- Multiple layers of abstraction – Constructs, Stacks, and Apps
- Really easy to get started (as long as you can get Nodejs, you can get CDK)

CDK Learning/Building Resources

- AWS CDK Documentation - <https://docs.aws.amazon.com/cdk/>
- AWS Solutions Constructs - <https://docs.aws.amazon.com/solutions/latest/constructs/welcome.html>
- Construct Hub - <https://constructs.dev/>
- This code repository (presentation and dancing kittens included) - <https://github.com/budoudoh/aws-cdk-webapp>

QR Code for repository →



Thank you!