Infrastructure Is Code with the AWS Cloud Development Kit

Richard Boyd
Software Engineer and Developer Advocate
AWS Developer Tools
Agenda

• Our infrastructure management journey
• The AWS Cloud Development Kit (AWS CDK)
• Demo: Build an AWS CDK app
• A lot more to explore
Our infrastructure management journey
🤔 What happens if an API call fails?
🤔 How do I make updates?
🤔 How do I know a resource is ready?
🤔 How do I roll back?

```javascript
require 'aws-sdk-ec2'

e2 = Aws::EC2::Resource.new(region: 'us-west-2')

instance = e2.create_instances({
  image_id: 'IMAGE_ID',
  min_count: 1,
  max_count: 1,
  key_name: 'MyGroovyKeyPair',
  security_group_ids: ['SECURITY_GROUP_ID'],
  instance_type: 't2.micro',
  placement: {
    availability_zone: 'us-west-2a'
  },
  subnet_id: 'SUBNET_ID',
  iam_instance_profile: {
    arn: 'arn:aws:iam::' + 'ACCOUNT_ID' + ':instance-profile/aws-opsworks-ec2-role'
  }
})
```
Resource provisioning engines

Desired state configuration

AWS CloudFormation template (JSON/YAML)

HashiCorp Configuration Language (HCL)

👍 Easy to automate
👍 Reproducible
😊 Configuration syntax
😊 No abstraction, lots of details
Document Object Models (DOMs)

👍 Real code ♥ if statements, for loops, IDE benefits
👍 Desired state
😢 Abstraction is not built-in  Ex: 218 lines of Troposphere for a VPC

Troposphere **Python**
SparkleFormation **Ruby**
GoFormation **Go**

Amazon SQS  Amazon SNS  Amazon DynamoDB  Amazon EC2  Amazon S3  AWS Step Functions

High level

DOMs  Declarative  Scripted  Manual

Low level
AWS Cloud Development Kit (AWS CDK)
AWS CDK application

Stack(s)

Construct
- Amazon SQS
- AWS Lambda

Construct
- Amazon S3
- Amazon DynamoDB

AWS CloudFormation template

Resources

AWS CloudFormation

Componentized
- DOMs
- Declarative
- Scripted
- Manual

High level
Low level
The AWS Construct Library is organized into several modules. They are named like this:

- `aws-xxx`: service package for the indicated service. This package will contain constructs to work with the given service.
- `aws-xxx-1`: a little superscript 1 indicates that this package only contains CloudFormation Resources (for now).
- `aws-xxx-targets`: integration package for the indicated service. This package will contain classes to connect the constructs in the "aws-xxx" package to other AWS services it can work with.
- `xxx`: packages that don't start "aws-" are AWS CDK framework packages.

**Module Contents**

Modules contain the following types:

- **Constructs** - All higher-level constructs in this library.
- **Other Types** - All non-construct classes, interfaces, structs and enums that exist to support the constructs.
- **CloudFormation Resources** - All constructs that map directly onto CloudFormation Resources.
AWS Cloud Development Kit (AWS CDK)

The AWS Cloud Development Kit (AWS CDK) is an open-source software development framework to define cloud infrastructure in code and provision it through AWS CloudFormation.

It offers a high-level object-oriented abstraction to define AWS resources imperatively using the power of modern programming languages. Using the CDK’s library of infrastructure constructs, you can easily encapsulate AWS best practices in your infrastructure definition and share it without worrying about boilerplate logic.

The CDK is available in the following languages:

- JavaScript, TypeScript (GA, Node.js ≥ 10.3.0)
- Python (GA, Python ≥ 3.6)
- Java (Developer Preview, Java ≥ 8 and Maven ≥ 3.5.4)
- .NET (Developer Preview, .NET Core ≥ 2.0)

Developer Guide | CDK Workshop | Getting Started | API Reference | Examples | Getting Help

Developers use the CDK framework in one of the supported programming languages to define reusable cloud components called constructs, which are composed together into stacks, forming a “CDK app”.

They then use the AWS CDK CLI to interact with their CDK app. The CLI allows developers to synthesize artifacts such as AWS CloudFormation Templates, deploy stacks to development AWS accounts and “diff” against a deployed stack to understand the impact of a code change.

The AWS Construct Library includes a module for each AWS service with constructs that offer rich APIs that encapsulate the details of how to use AWS. The AWS Construct Library aims to reduce the complexity and glue-logic required when integrating various AWS services to achieve your goals on AWS.
Demo: Build an AWS CDK app
Demo – Goals

- CDK concepts
- CDK CLI workflow
- AWS construct library
Demo
Demo recap

Get started:

npm install -g aws-cdk

cdk init --language <typescript | java | csharp | ...>
Thank you!

Richard Boyd
@rchrdbyd
rhboyd@amazon.com
Next steps

• Try out the CDK
  • https://cdkworkshop.com
  • https://awslabs.github.io/aws-cdk
  • https://github.com/awslabs/cdk-reinvent

• Engage with us
  • https://github.com/awslabs/aws-cdk
  • https://gitter.im/awslabs/aws-cdk