



# CDKテスト入門

Tomoya Kudo  
Prototyping Engineer

[kudtomoy@amazon.co.jp](mailto:kudtomoy@amazon.co.jp)

# CDK のテストと assertions モジュール

- CDKはインフラのコードに対してテストが書ける
- 新しい assertions モジュールは全言語に対応
  - CDKv1は @aws-cdk/assertions
  - CDKv2 は aws-cdk-lib/assertions

## @aws-cdk/assertions module

Language	Package
 .NET	Amazon.CDK.Assertio
 Java	software.amazon.aws
 Python	aws_cdk.assertions
 TypeScript	@aws-cdk/assertions

## aws-cdk-lib.assertions module

Language	Package
 .NET	Amazon.CDK.Assertions
 Go	github.com/aws/aws-cdk-go/awscdk/v2/assertions
 Java	software.amazon.awscdk.assertions
 Python	aws_cdk.assertions
 TypeScript	aws-cdk-lib » assertions

# CDK のテストの流れ

1. テスト対象の**Stack**のインスタンスを作成

2. CloudFormationテンプレートを生成

3. Assertする

A. Snapshot Testing

B. Fine-grained Assertions

```
import { Stack } from 'aws-cdk-lib';
import { Template } from 'aws-cdk-lib/assertions';

const stack = new Stack(/* ... */);
// ...
const template = Template.fromStack(stack);
```

↑ CDK StackからCFn生成

```
const templateJson = '{ "Resources": ... }'; /* The CloudFormation template
const template = Template.fromString(templateJson);
```

↑ 文字列からCFn取り込み

# Snapshot Testing

- 生成されたCFnテンプレートを以前のものと比較し、両者に差分があるかをテストする手法
- 各言語のSnapshot Pluginを利用
  - TypeScriptならJest、PythonならSyrupy、Javaならjson-snapshotなど

```
test( name: 'Snapshot test', fn: () => {  
  const app = new App()  
  const stack = new Practice1StV2Stack(app, id: 'Practice1StV2Stack', props: {})  
  const template = Template.fromStack(stack)  
  
  expect(template.toJSON()).toMatchSnapshot()  
})
```

# Fine-grained Assertions

- 生成されたCFnのリソースやアウトプットが、期待通りに生成されているかテストする手法
- 主に**Assertions**ライブラリの機能を利用
  - **Matcher**、**Capture**など

```
// Functionが2つ作られていることを確認する
template.resourceCountIs( type: 'AWS::Lambda::Function', count: 2)

// Resource Matching & Retrieval
// Propertiesが正しいことを確認する
template.hasResourceProperties( type: 'AWS::DynamoDB::Table', props: {
  "AttributeDefinitions": [
```

# CDKのテストまとめ

- **Snapshot Testing**

- CDKフレームワークのバージョンアップや、CDKアプリケーションのリファクタによるデグレを防ぐために有用
- ほとんど手間もかからないのでこちらから始めるのがオススメ

- **Fine-grained Assertions**

- いわゆる一般的なテスト
- 自作Constructが正しく動作するかなどを検証
- 詳しくは公式ドキュメントやBlogを参照
- <https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.assertions-readme.html>

# 本日のまとめ

- CDKの概要とプロトタイピングチームの活用事例や参考コードへのアクセス方法について紹介いたしました
- CDKの使い方をゼロからライブコーディングでお見せしました
- CDKのテストについて紹介いたしました

