

AWS CDK Testing – 1.Fine-grained assertions Tests

There are two categories of tests that you can write for AWS CDK apps.

Fine-grained assertions



JEST

- AWS CDK apps uses the AWS CDK's [assertions](#) module and test frameworks like [Jest](#) for TypeScript
- Test **specific aspects** of the generated AWS CloudFormation template - this **property with this value**
- Detect **regressions**.
- **Test-driven development**- write a test first, then make it pass by writing a correct implementation
- Most **frequently** used tests.

AWS CDK 2.69.0

[API Reference](#)
[Python](#)
[Java](#)
[.NET](#)
[Go](#)

API Reference

aws-cdk-lib

aws-cdk-lib.alexa_ask

aws-cdk-lib.assertions

Overview

Classes

Annotations

Capture

Match

Matcher

MatchResult

Template

Structs

MatchCapture

MatchFailure

TemplateParsingOptions

aws-cdk-lib.assets

aws-cdk-lib.aws_accessanalyzer

aws-cdk-lib.aws_acmpca

aws-cdk-lib.aws_amazonmq

aws-cdk-lib.aws_amplify

aws-cdk-lib.aws_amplifyuibuilder

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

Assertions

If you're migrating from the old `assert` library, the migration guide can be found in [our GitHub repository](#).

Functions for writing test asserting against CDK applications, with focus on CloudFormation templates.

The `Template` class includes a set of methods for writing assertions against CloudFormation templates. Use one of the `Template.fromXxx()` static methods to create an instance of this class.

To create `Template` from CDK stack, start off with:

AWS CDK Testing – Fine-grained assertions Tests

Steps to follow for testing using Fine-grained assertions



1. Write the CDK code for AWS Service being created with required properties or attributes
2. Write the CDK test for properties that need to be tested for AWS Infrastructure Service using Assertion module and Jest
3. Run the test using – `npm run test` command
4. AWS CDK validates against the synthesized CloudFormation template in the '`cdk.out`' folder and not the deployed CloudFormation code in the Production environment.

```

TS infra-stack.ts 1, M X
infra > lib > TS infra-stack.ts > InfraStack > constructor
1 import * as cdk from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import * as s3 from 'aws-cdk-lib/aws-s3';
4
5 export class InfraStack extends cdk.Stack {
6   constructor(scope: Construct, id: string, props?: cdk.StackProps) {
7     super(scope, id, props);
8
9     // AWS S3 CDK testing
10
11     // S3 bucket
12     const s3buckettest = new s3.Bucket(this, 'S3bucketTest', {
13       bucketName: 'testdemo18032023',
14       versioned: true
15     });
16   }
17 }
18

```

```

TS infra.test.ts X
section_9 > CDKfinegrainedtesting > infra > test > TS infra.test.ts > ...
1 import * as cdk from 'aws-cdk-lib';
2 import { Template } from 'aws-cdk-lib/assertions';
3 import * as Infra from '../lib/infra-stack';
4
5 // example test. To run these tests, uncomment this file along with the
6 // example resource in lib/infra-stack.ts
7 test('S3 testing', () => {
8   const app = new cdk.App();
9   // WHEN
10   const stack = new Infra.InfraStack(app, 'MyTestStack');
11   // THEN
12   const template = Template.fromStack(stack);
13
14   template.hasResourceProperties('AWS::S3::Bucket', {
15     "BucketName": "testdemo18032023",
16     "VersioningConfiguration": {
17       "Status": "Enabled"
18     }
19   });
20 });
21

```

```

PS C:\Users\ADMIN\Desktop\AWS Publish\Udemy\AWS CDK\AWS\CDK_final_03032023\Section 7\CDK Testing\CDKfinegrainedtesting\infra> npm run test
> infra@0.1.0 test
> jest
PASS test/infra.test.ts (11.847 s)
  ✓ S3 testing (60 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 12.355 s
Ran all test suites.

```

AWS CDK Testing – 2. Snapshot Tests

Snapshot Tests



- Test the **synthesized AWS CloudFormation** template against a **previously stored baseline template**.
- Snapshot tests let you **refactor** freely
- If the changes are **intentional**, you can accept a new **baseline** for future tests.



JEST

Steps to follow for testing using Snapshot Test

1. Write the CDK code for AWS Service being created with required properties or attributes
2. Write the CDK test and define the baseline/master CloudFormation template against which the new templates will be compared.
3. Run the test using – “npm run test” command
4. A baseline CloudFormation template will be generated under the “__snapshots__” folder.
5. When any changes are made to the CDK Code, the synthesized template will be tested against the baseline template
6. You can modify the template if changes are inadvertent or you can accept using “npm test -- -u” command, which will

Creator and Copyright - Rahul Trisal

```
TS snap-test.ts X
section_9 > CDKsnapshottesting > infra > lib > TS snap-test.ts > DeadLetterQueue
1 import * as cdk from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import * as cloudwatch from 'aws-cdk-lib/aws-cloudwatch';
4 import * as sqs from 'aws-cdk-lib/aws-sqs';
5
6 export class DeadLetterQueue extends sqs.Queue {
7   public readonly messagesInQueueAlarm: cloudwatch.IAlarm;
8
9   constructor(scope: Construct, id: string) {
10     super(scope, id);
11
12     // Add the alarm
13     const alarm = new cloudwatch.Alarm(this, 'Alarm', {
14       alarmDescription: 'There are messages in the Dead Letter Queue',
15       evaluationPeriods: 1,
16       threshold: 2,
17       metric: this.metricApproximateNumberOfMessagesVisible(),
18       alarmName: 'snapshottestalarm'
19     });
20   }
21 }
```

```
TS infra.test.ts X
section_9 > CDKsnapshottesting > infra > test > TS infra.test.ts > ...
1 import * as cdk from 'aws-cdk-lib';
2 import { Match, Template } from 'aws-cdk-lib/assertions';
3 import { DeadLetterQueue } from '../lib/snap-test';
4
5
6 describe("DeadLetterQueue", () => {
7   test("matches the snapshot", () => {
8     const stack = new cdk.Stack();
9     new DeadLetterQueue(stack, "DeadLetterQueue");
10
11     const template = Template.fromStack(stack);
12     expect(template.toJSON()).toMatchSnapshot();
13   });
14 });
15
```

```
PS C:\Users\ADMIN\Desktop\AWS Publish\Udemy\AWS CDK\AWS\CDK_final_03032023\Section 7\CDK Testing\CDKsnapshottesting\infra> npm run build
> infra@0.1.0 build
> tsc
PS C:\Users\ADMIN\Desktop\AWS Publish\Udemy\AWS CDK\AWS\CDK_final_03032023\Section 7\CDK Testing\CDKsnapshottesting\infra> npm run test
> infra@0.1.0 test
> jest
```

```
PASS test/infra.test.ts (9.601 s)
  DeadLetterQueue
    ✓ matches the snapshot (132 ms)

  › 1 snapshot written.
Snapshot Summary
  › 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 1 written, 1 total
Time: 10.017 s
Ran all test suites.
```