
AWS CDK test Synth

Release 0.0.2

Alessandra Bilardi

Feb 22, 2022

CONTENTS:

1	Getting started	3
1.1	Prerequisites	3
1.2	Installation	3
1.3	Change Log	3
1.4	License	4
2	Usage	5
2.1	Example	6
3	Development	7
3.1	Run tests	7
3.2	Template saved	7
4	Indices and tables	9

This package contains the classes for managing the testing of your AWS CDK code.

GETTING STARTED

AWS CDK test Synth package is implemented for helping you to test your AWS CDK code.

The goal is to cover the missing of an [official testing infrastructure](#) for **Python** as exists for **TypeScript** using **Jest**.

It is part of the [educational repositories](#) to learn how to write standard code and common uses of the TDD.

1.1 Prerequisites

This package is useful for testing your [AWS CDK](#) code, so you probably have everything you need installed already.

1.2 Installation

If you want to use this package into your code, you can install by python3-pip:

```
pip3 install aws_cdk_test_synth
python3
>>> import aws_cdk_test_synth.test_synth as TestSynth
>>> help(TestSynth)
```

Read the documentation on [readthedocs](#) for

- Usage
- Development

1.3 Change Log

See [CHANGELOG.md](#) for details.

1.4 License

This package is released under the MIT license. See [LICENSE](#) for details.

USAGE

The class reads the yaml template saved, synthesizes the new template and compares between them.

You have to import the class **TestSynth** instead of **unittest.TestCase** package and your **TestYourClass** class has to extend it.

```
from tests.test_synth import TestSynth
from sample.yourclass import YourClass
class TestYourClass(TestSynth):
```

The method named **__init__** has to call the class named **TestSynth** with the relative path of yaml template saved.

```
def __init__(self, *args, **kwargs):
    TestSynth.__init__(self, 'tests/yourclass.yaml', *args, **kwargs)
```

There is the method named **synth** with the initialization of **YourClass**

```
def synth(self, app):
    YourClass(app, id="name-of-your-stack")
```

If you have to manage more **YourClass**, you can use the method **synthesizes**

```
def your_synth(self, app):
    YourClass(app, id="name-of-your-stack")

def test_your_synth(self):
    self.synthesizes('your_synth')
```

If you have to manage more templates, you can use the method **load_template**

```
def your_synth(self, app):
    YourClass(app, id=self.id)

def test_your_synth(self):
    self.id = "name-of-your-synth"
    self.load_template('tests/yoursynth.yaml')
    self.synthesizes('your_synth')
```

2.1 Example

When you create your **stack.py** file, you can create your **test_stack.py** like the example in the `tests` folder of this repository. In that example, you can find

- **s3_stack.py**, the example of your stack implementation
- **test_s3_stack.py**, the example of your unittest class
- **s3.yaml**, the template saved

When you run the unittest (see the [Development](#) Section),

- before, you have to create a file empty named **s3.yaml**
- the first time, you have to fill it with the first version
- the times after, if the templates saved and new are different, so the unittest fails, you can evaluate if you have to save a new version or fix your change

You can find other examples in the repositories below:

- [aws-simple-pipeline/tests/test_pipeline_stack.py](#)
- [aws-tool-comparison/cdk/python/tests/test_*.py](#)

DEVELOPMENT

This package is used only for testing your AWS CDK code, so it is useful for helping you to implement your **TDD**. If you want to use this package with your AWS CDK code, you can find an example in the [tests](#) folder of this repository.

3.1 Run tests

```
cd aws-cdk-test-synth/  
pip3 install --upgrade -r requirements.txt  
python3 -m unittest discover -v
```

3.2 Template saved

If the unittest fails, the class prints the yaml files. So, if you will have implemented more unittests classes, it will be simple to run one unittest at a time:

```
cd aws-cdk-test-synth/  
python3 -m unittest discover -v -p test_s3_stack.py
```

And if you will want to compare what it is different with the template saved, you can save the new template:

```
cd aws-cdk-test-synth/  
python3 -m unittest discover -v -p test_s3_stack.py > tests/s3.yaml.2
```

This approach is comfortable for running your program for comparison, like [Meld](#), `diff` or `fc`:

```
cd aws-cdk-test-synth/  
diff tests/s3.yaml tests/s3.yaml.2
```


INDICES AND TABLES

- `genindex`
- `modindex`
- `search`