

AWS CDKのコンセプト 4

© 2021.04.30

Identifiers

識別子は作成されたスコープ内で一意である必要がある。AWS CDKアプリ内でグローバルの一意である必要はない。

Construct IDs

`id` は作成されるスコープ内で一意である必要がある。

以下の例では、スタックのインスタンスを2つ作成し、それぞれ異なる `id` を設定している。

```
from aws_cdk.core import App, Construct, Stack, StackProps
from aws_cdk import aws_s3 as s3

class MyStack(Stack):

    def __init__(self, scope: Construct, id: str, **kwargs):

        super().__init__(scope, id, **kwargs)
        s3.Bucket(self, "MyBucket")

app = App()
MyStack(app, 'Stack1')
MyStack(app, 'Stack2')
```

Paths

`App` クラスのルート階層から見たコンストラクタの場所。

例えば、上記のMyStackで作成された2つのスタックには、それぞれS3バケットがあるが、パスは `Stack1/MyBucket` と `Stack2/MyBucket` となる。

```
path = my_construct.node.path
```

Unique IDs

CloudFormationはテンプレート内のすべてのLogical IDが一意である必要があるため、CDKは各コンストラクタに一意の識別子を生成する。識別子は、パスをすべて連結し、末尾に8桁のハッシュを追加する。

```
uid = Names.unique_id(my_construct)
```

例として、上記の `Stack2/MyBucket` のLogical IDは `Stack2MyBucket4DD88B4F` となる。

また人が読むには適さないが、addressという、42桁の16進文字で構成された識別子もある。

```
addr = my_construct.node.addr
```

Tokens

トークンはアプリのライフサイクルの内、後で取得できる値を表す。

例えば、CDKアプリで定義されたS3バケットの名前は、CloudFormationテンプレートが合成されたときに取得できる。それまでの `bucket.bucketName` の値は以下になる。

```
${TOKEN[Bucket.Name.1234]}
```

トークンを実際のS3バケットの名前の値として、渡すことができる。

```
bucket = s3.Bucket(self, "MyBucket")

fn = lambda_.Function(stack, "MyLambda",
    environment=dict(BUCKET_NAME=bucket.bucket_name))
```

トークンを含むか否かにかかわらず、Json形式の任意データを出力したい場合、メソッド [stack.toJsonString](#) を使用する。

```
stack = Stack.of(self)
string = stack.to_json_string(dict(value=bucket.bucket_name))
```

Parameters

基本的にAWS CDKアプリでCloudFormationテンプレートのパラメータを使用することは推奨されない。

例外として、AWS CDKアプリで生成されたテンプレートを他のチームに使用するか、AWSサービスに使用するかなどの際は必要となる。

```
upload_bucket_name = CfnParameter(self, "uploadBucketName", type="String",
    description="The name of the Amazon S3 bucket where uploaded files will be stored.")
```

```
bucket = Bucket(self, "myBucket",
    bucket_name=upload_bucket_name.value_as_string)
```

```
cdk deploy MyStack --parameters uploadBucketName=UpBucket --parameters downloadBucketName=DownBucket
```

```
cdk deploy MyStack YourStack --parameters MyStack:uploadBucketName=UploadBucket --parameter
```

Tagging

タグ「key」に値「value」を設定。

```
Tags.of(my_construct).add("key", "value")
```

タグ「key」を削除。

```
Tags.of(my_construct).remove("key")
```

CDKはタグの追加と削除を再帰的に行う。競合がある場合、最も優先度が高いオペレーションが勝つ。

```
Tags.of(my_construct).add("key", "value", priority=300)
```

Optional properties

`apply_to_launched_instances`、`false`に設定することでAuto Scaling Group内で起動されたインスタンスにタグが設定されない。

`include_resource_types` / `exclude_resource_types`、CloudFormationのリソースタイプをもとに、一部リソースのタグのみを操作する。

```
from aws_cdk.core import App, Stack, Tags

app = App()
the_best_stack = Stack(app, 'MarketingSystem')

# Add a tag to all constructs in the stack
Tags.of(the_best_stack).add("StackType", "TheBest")

# Remove the tag from all resources except subnet resources
Tags.of(the_best_stack).remove("StackType",
                                exclude_resource_types=["AWS::EC2::Subnet"])
```

```
Tags.of(the_best_stack).add("StackType", "TheBest",
                              include_resource_types=["AWS::EC2::Subnet"])
```