

This is code. Fork it on github.com/ojacques



CDK

L'infrastructure-as-code avec son langage favori

Olivier JACQUES

AWS ProServe



Les façons de provisionner de l'infrastructure

- **Manuel** : Documents Word, Wikis, demande-a-robert-il-l'a-déjà-fait
- **Scripté** : `#!/bin/bash`
- **Déclaratif** : CloudFormation, Terraform
- **Générateurs** : Troposphere, GoFormation
- **Abstractions** : AWS CDK, Pulumi

This is code. Fork it on github.com/ojacques

AWS Cloud Development Kit



© 2022, Amazon Web Services, Inc. or its Affiliates.

This is code. Fork it on github.com/ojacques

Programmez l'infrastructure en Python, Typescript, Javascript, Java, C#, Go

```
40 // Create random value
41 const pet = new random.Pet(this, "random-name", {
42   length: 2,
43 });
44
45 // Create Lambda executable
46 const asset = new TerraformAsset({
47   path: path.resolve(__dirname, "lambda.zip"),
48   type: AssetType.ARCHIVE,
49 });
50
51 // Create unique S3 bucket
52 const bucket = new aws.s3.S3Bucket(this, "bucket", {
53   bucketPrefix: `learn-cdktf-${name}`,
54 });
55
56 // Upload Lambda zip file to newly created S3 bucket
57 const lambdaArchive = new aws.s3.S3Object(this, "lambda-archive", {
58   bucket: bucket.bucket,
59   key: `${config.version}/${asset.fileName}`,
60   source: asset.path, // returns a posix path
61 });
62
63 // Create Lambda role
64 const role = new aws.iam.IamRole(this, "lambda-exec", {
65   name: `learn-cdktf-${name}-${pet.id}`,
66   assumeRolePolicy: JSON.stringify(lambdaRolePolicy)
67 });
```

constructor S3Bucket(scope: Construct, id: string, config?: aws.s3.S3BucketConfig | undefined): aws.s3.S3Bucket

Create a new `aws_s3_bucket` Resource

@param scope — The scope in which to define this construct

@param id — The scoped construct ID. Must be unique amongst siblings in the same scope

@param options — S3BucketConfig = {}

3 projets






- AWS CDK
- CDK for Terraform
- Cloud Development Kit for Kubernetes - CDK8s

Tous basés sur JSSI

3 cycles

- [Python, TS, JS, Java, C#, Go] => **AWS CDK** => CloudFormation
- [Python, TS, JS, Java, C#, Go] => **CDKTF** => Terraform HCL
- [Python, TS, JS, Java, C#, Go] => **CDK8s** => Kubernetes manifests

Séquence

-  `cdktf init`
-  `npm run build`
-  `cdktf synth`
-  `cdktf diff`
-  `cdktf deploy`

This is code. Fork it on github.com/ojacques



Merci

 @ojacques2