



AWS Cloud Development Kit (CDK)

A modern way to build your AWS infrastructure

Randy Ridgley

Principal Solutions Architect

Abstractions

AWS CDK, Pulumi

Generators

Troposphere, GoFormation

Declarative

CloudFormation, Terraform

Scripted

`#!/bin/bash`

Manual

Wikis, Playbooks, Ask-Bob-he-knows

AWS Cloud Development Kit



Familiar

Your language

Just classes and methods



Tool Support

AutoComplete

Inline documentation



Abstraction

Sane defaults

Reusable classes

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    domainName
    domainZone
  }
}

// Setup AutoScaling policy
const scaling = service.service.autoScaleTask
scaling.scaleOnCpuUtilization('CpuScaling',
  targetUtilizationPercent: 50,
  scaleInCooldown: Duration.seconds(60),
  scaleOutCooldown: Duration.seconds(60)
});
```

(property) patterns.NetworkLoadBalancedServiceBaseProps.domainName?: string | undefined

The domain name for the service, e.g.

"api.example.com."

@default

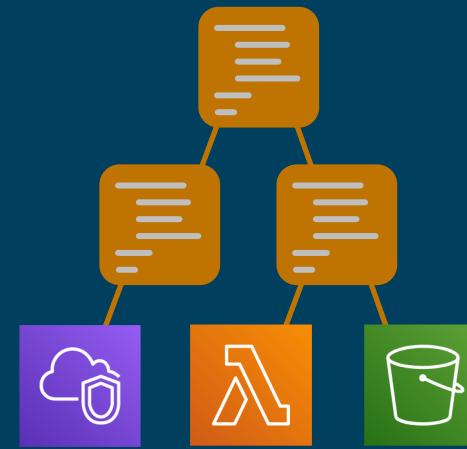
- No domain name.



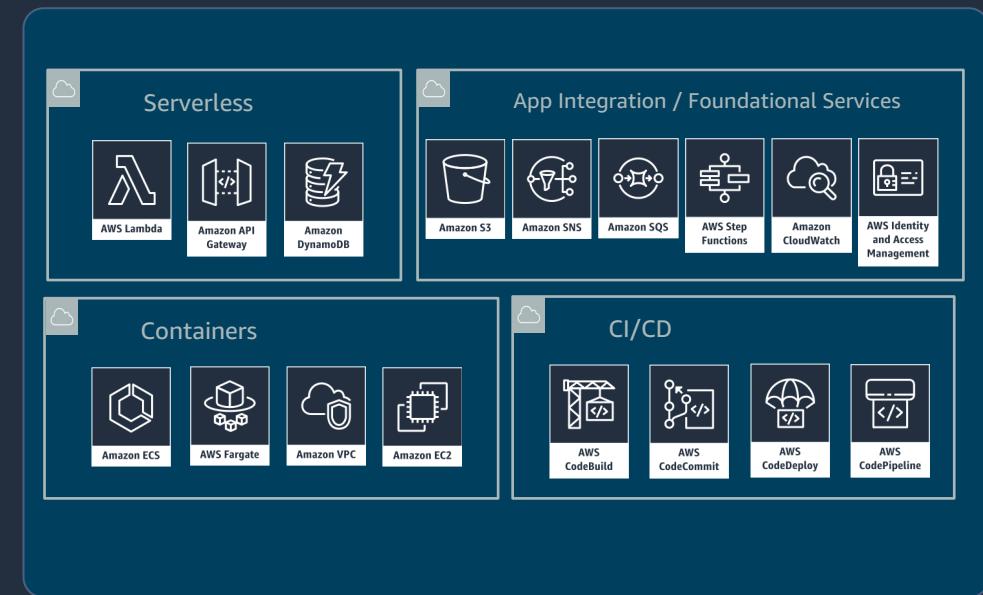
= GO
COMING SOON!

Main Components

App
Stacks
Resources



Core Framework



AWS Construct Library

The screenshot shows a terminal window titled 'cdk-app' running on a Mac OS X system. It displays the output of the 'cdk diff' command, specifically the 'IAM Statement Changes' section. The table lists a single change where a new IAM policy is added to allow sending messages to an SNS topic. The terminal also shows the full JSON configuration for the AWS Lambda function's IAM role.

Resource	Effect	Action	Principal	Condition
+\$ {CdkAppQueue.Arn}	Allow	sqs:SendMessage	Service:sns.amazonaws.com	"ArnEquals": {"aws:SourceArn": "\$ {CdkAppTopic}"}

(NOTE: There may be security-related changes not in this list. See <https://github.com/aws/aws-cdk/issues/1299>)

Conditions

```
[+] Condition CDKMetadataAvailable: {"Fn::Or": [{"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-east-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-south-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-northeast-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-northeast-2"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-southeast-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ap-southeast-2"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "ca-central-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "cn-north-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "cn-northwest-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "eu-central-1"]}], {"Fn::Or": [{"Fn::Equals": [{"Ref": "AWS::Region"}, "eu-north-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "eu-west-3"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "me-south-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "sa-east-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "us-east-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "us-east-2"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "us-west-1"]}, {"Fn::Equals": [{"Ref": "AWS::Region"}, "us-west-2"]}]}]}
```

Resources

```
[+] AWS::SQS::Queue CdkAppQueue CdkAppQueueFB2A6F8C
[+] AWS::SQS::QueuePolicy CdkAppQueue/Policy CdkAppQueuePolicy68032EBA
[+] AWS::SNS::Subscription CdkAppQueue/CdkAppStackCdkAppTopicA685FD20 CdkAppQueueCdkAppStackCdkAppTopicC4BDC8F1
[+] AWS::SNS::Topic CdkAppTopic CdkAppTopicF4BDC8F1
```

cdk-app on master [?] is v0.1.0 via v14.2.0 using camelnyk-cdk took 2s

AWS CDK CLI

Write Code

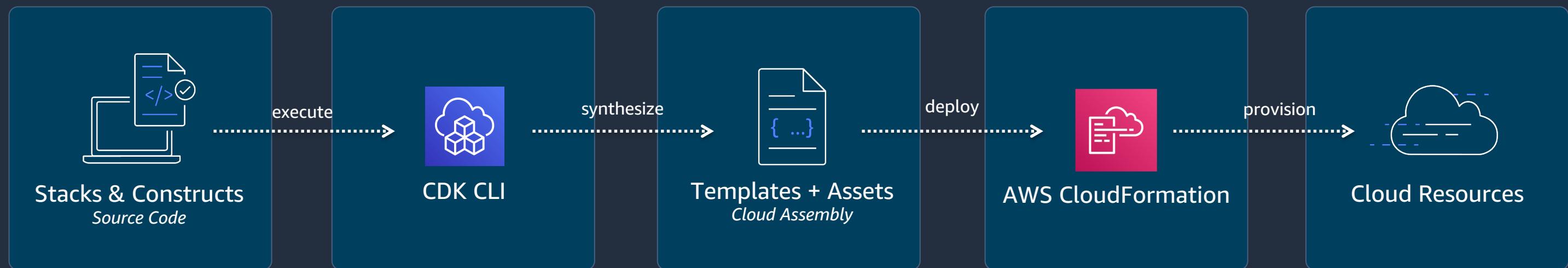
```
1 import { Construct } from "@aws-cdk/core";
2 import { Bucket, BucketEncryption, BucketProps } from "@aws-cdk/aws-s3";
3
4 export class EncryptedBucket extends Construct {
5   constructor(scope: Construct, id: string, props?: BucketProps) {
6     super(scope, id);
7
8     let newProps: BucketProps = { ...props };
9     if (
10       !props ||
11       props?.encryption === undefined ||
12       props?.encryption === BucketEncryption.UNENCRYPTED
13     ) {
14       newProps.encryption = BucketEncryption.KMS_MANAGED;
15     }
16     new Bucket(this, `${id}-bucket`, newProps);
17   }
18 }
19
```

Test Code



```
1 import { EncryptedBucket } from "../lib/index";
2 import { App, Stack } from "@aws-cdk/core";
3 import "@aws-cdk/assert/jest";
4 import { BucketEncryption } from "@aws-cdk/aws-s3";
5
6 test("Does not allow for unencrypted buckets", () => {
7   const mockApp = new App();
8   const stack = new Stack(mockApp, "testing-stack");
9
10  new EncryptedBucket(stack, "testing", {
11    encryption: BucketEncryption.UNENCRYPTED,
12  });
13
14  expect(stack).toHaveResource("AWS::S3::Bucket", {
15    BucketEncryption: {
16      ServerSideEncryptionConfiguration: [
17        {
18          ServerSideEncryptionByDefault: {
19            SSEAlgorithm: "aws:kms",
20          },
21        },
22      ],
23    },
24  });
25});
```

Development Workflow



- 📚 `cdk init` // create new project
- 🔧 `npm run build` // build project
- 🧬 `cdk synth` // create templates and assets
- 🔍 `cdk diff` // check what will change
- 🚀 `cdk deploy` // push changes to the cloud

CDK Constructs



```
new s3.CfnBucket(this, "myBucket", {bucketName: "my-bucket"});
```

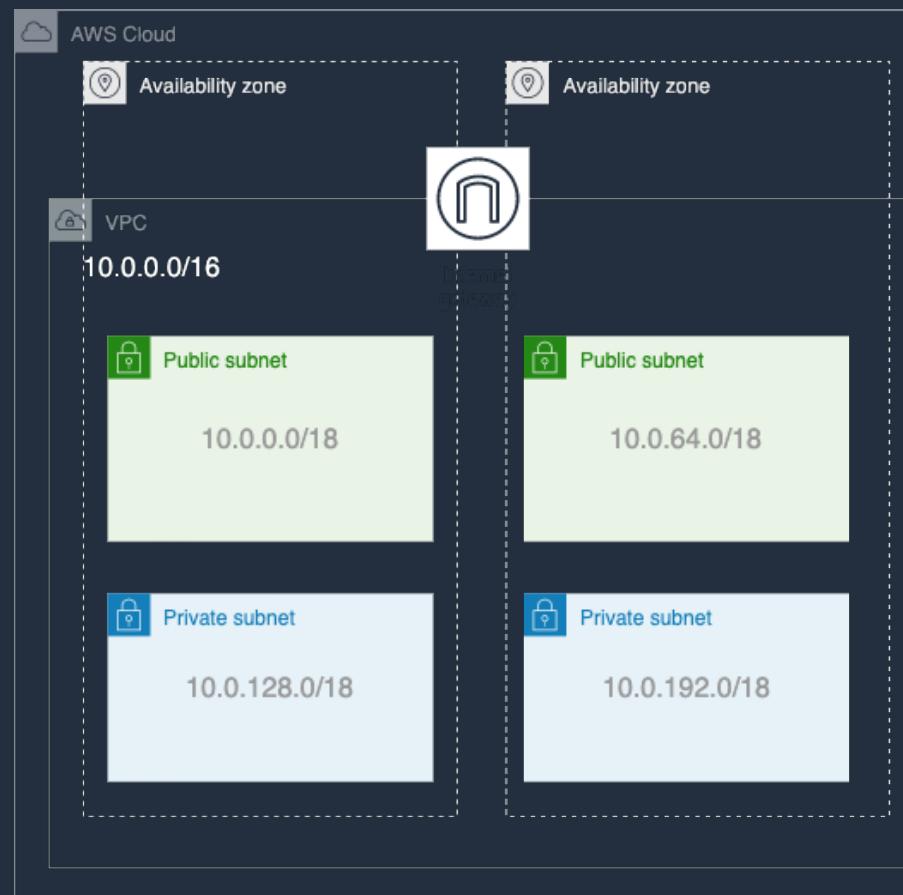
 cdk synth



- Generated mappings from CloudFormation Specification
- abc.CfnXyz → AWS::ABC::XYZ CloudFormation Resource
- ec2.CfnInstance → AWS::EC2::Instance
- kms.CfnKey → AWS::KMS::Key



```
Resources:  
  myBucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      BucketName: my-bucket
```

 cdk synth

- Ready-to-use VPC setup
- 65536 IPs split equally between 4 subnets
- If you provide a region → adjusted to 3 AZs
- Everything is optional, change any parameter
- Sane default values



```
● ● ●

export class CdkAppStack extends cdk.Stack {
  constructor(scope: cdk.App, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

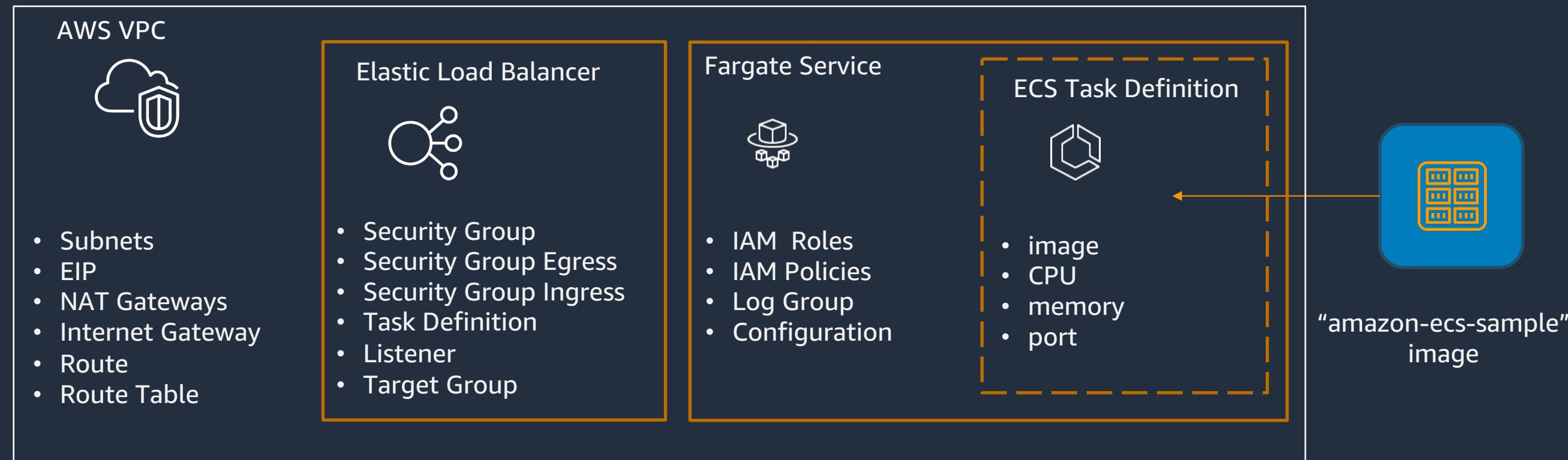
    let table = new ddb.Table(this, "table", {
      partitionKey: { name: "id", type: ddb.AttributeType.STRING },
    });
    let inputBucket = new s3.Bucket(this, "outputBucket");
    const myLambdaFunction = new lambda.Function(this, "myFunction", {
      code: Code.fromAsset("lambda_folder"),
      handler: "function_file.function_name",
      runtime: lambda.Runtime.PYTHON_3_8,
    });

    inputBucket.grantRead(myLambdaFunction);
    table.grantWriteData(myLambdaFunction);
  }
}
```

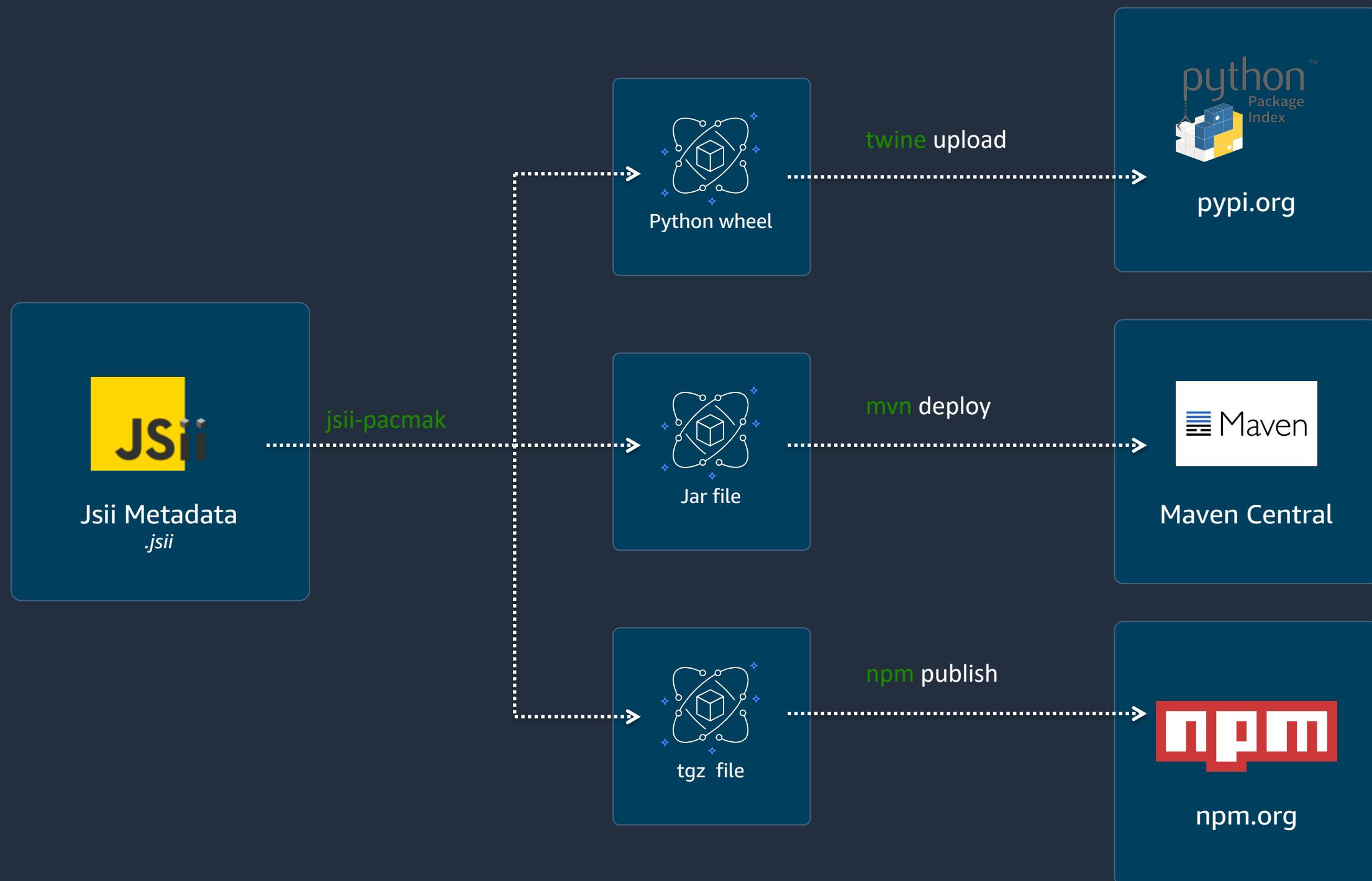
An orange arrow points from the `grantWriteData` line in the code back to the **Amazon DynamoDB** box in the diagram.



```
new ecs_patterns.ApplicationLoadBalancedFargateService(this, "FargateService", {  
    cluster,  
    taskImageOptions: {  
        image: ecs.ContainerImage.fromRegistry("amazon/amazon-ecs-sample"),  
    },  
});
```



Package & Publish



Emerging ecosystem

Define composable configuration models through code



AWS Solutions Constructs

Vetted, configurable, higher-level infrastructure-as-code patterns **provided by AWS Solutions Architecture group**

aws.amazon.com/solutions/constructs

Deeper dive

Turbocharge the AWS CDK with AWS Solutions Constructs

CDK Patterns

A catalog of higher-level constructs **provided by CDK community members**



cdkpatterns.com
@CDKPATTERNS



CDK Patterns

Awesome CDK

Central “**awesome list**” for CDK open source projects, guides, blogs, and resources

github.com/kolomied/awesome-cdk



awesome

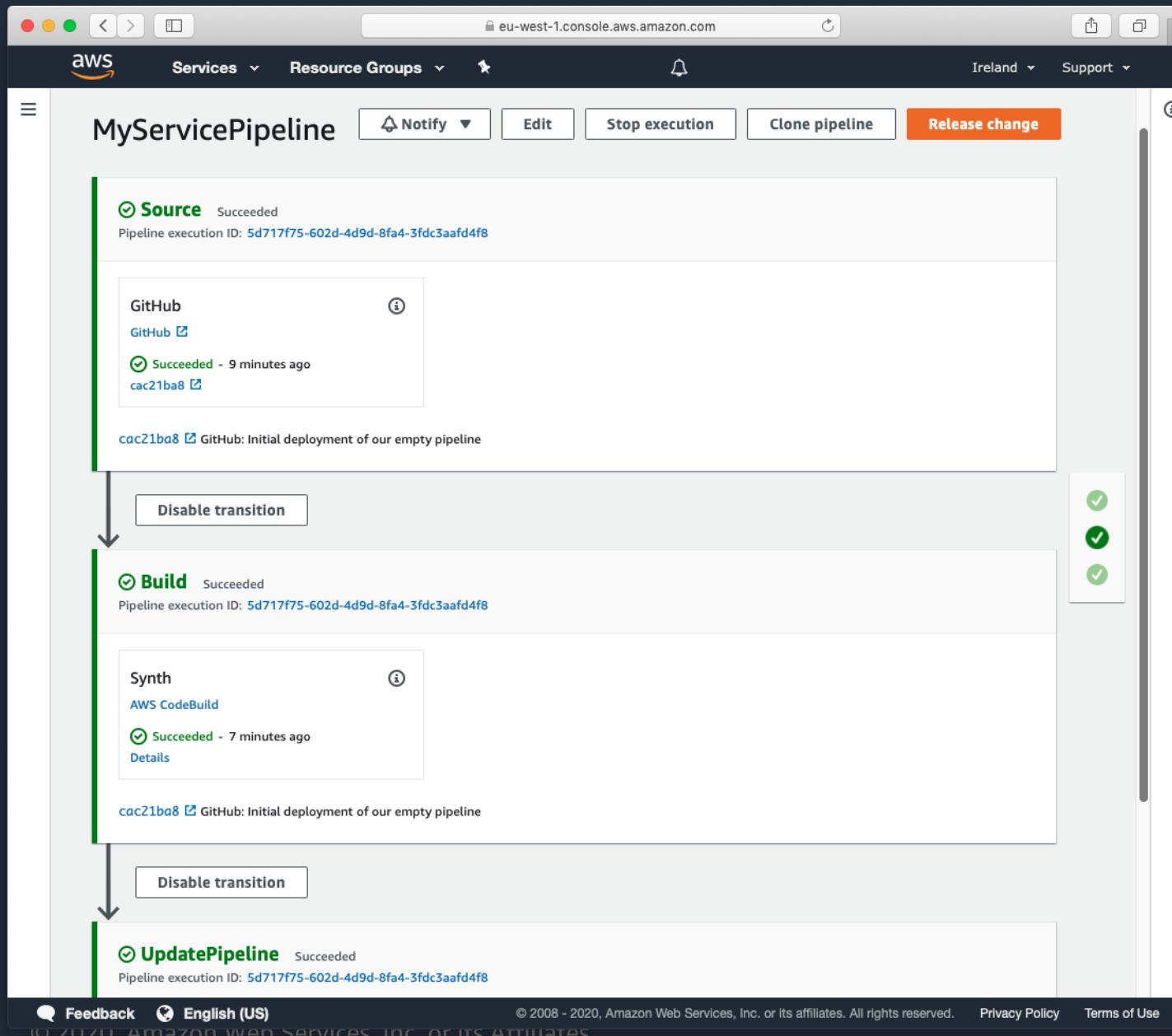
AWS Construct Library

Constructs provided by the **AWS CDK team** to encapsulate a single service's configuration interface

aws.amazon.com/cdk

AWS CDK Pipelines

CONTINUOUS DELIVERY FOR AWS CDK APPLICATIONS



- Model continuous delivery pipelines as part of your infrastructure code
- Pipelines are self modifying as you push your AWS CDK code to origin
- Easily model cross-account and cross-region pipeline configurations

Deeper dive
AWS Online Tech Talk:
Enhanced CI/CD with AWS CDK



CDK for Terraform (cdktf)

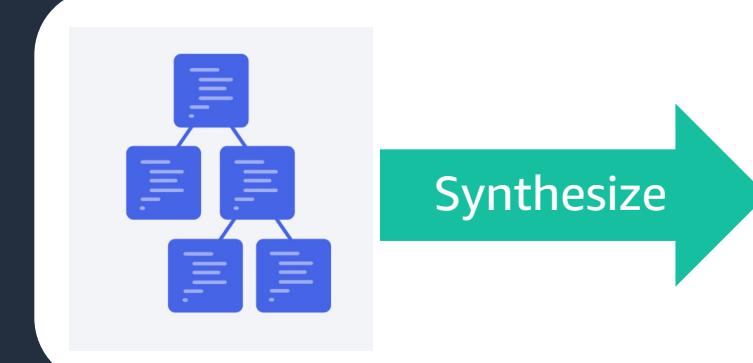
```
import { Construct } from 'constructs';
import { App, TerraformStack } from 'cdktf';
import { AwsProvider } from './gen/providers/aws';
import { Instance } from './gen/providers/aws/instance';

class HelloTerraform extends TerraformStack {
  constructor(scope: Construct, name: string) {
    super(scope, name);

    new AwsProvider(this, 'aws', {
      region: 'us-east-1'
    });

    new Instance(this, 'Hello', {
      ami: "ami-2757f631",
      instanceType: "t2.micro"
    });
  }
}

const app = new App();
new HelloTerraform(app, 'hello-terraform');
app.synth();
```



Deeper dive

HashiCorp webinar: Provision Infrastructure Using CDK for Terraform with Python & TypeScript



CDK for Kubernetes (cdk8s)



cdk8s-welcome > TS main.ts > MyChart > constructor

```
1 import { Construct } from 'constructs';
2 import { App, Chart } from 'cdk8s';
3 import { WebService } from './lib/web-service';
4
5 class MyChart extends Chart {
6   constructor(scope: Construct, name: string) {
7     super(scope, name);
8
9     new WebService(this, 'aws-webapp', {
10       image: 'aws/webapp:1.6',
11       WebService(scope: Construct, ns: string, options:
12         WebServiceOptions): WebService
13
14     }) port?
15   }
16 } #endregion Folding Region End...
17 #region Folding Region Start...
18 class Class Definition ...
19 const Constructor (Type...
20 new Do-While Statement...
21 app. Do-While Statement...
22 dowhile Log error to the ...
23 error For Loop (TypeScr...
24 for For-Each Loop usi...
25 foreach For-In Loop (Type...
26 forin For-In Loop (Type...
27 forof For-Of Loop (Type...
28 function Function Statement...
```

cdk8s-welcome > dist > ! cdk8swelcome.k8s.yaml > {} spec > {} template > {} spec > [] co

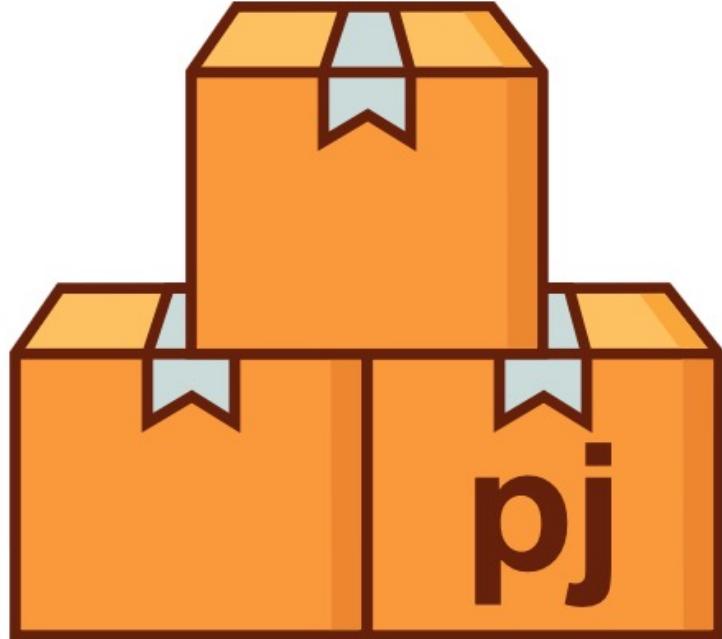
```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: cdk8s-welcome-aws-webapp-service-2325fc6b
5 spec:
6   ports:
7     - port: 80
8       targetPort: 8080
9     selector:
10      app: cdk8swelcomeawswebapp83EC99A6
11      type: LoadBalancer
12
13 apiVersion: apps/v1
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```

Deeper dive
Simplifying Kubernetes application management with cdk8s



Projen

projen



Define and maintain complex project configuration through code.

As opposed to existing templating/scaffolding tools, *projen* is not a one-off generator. Synthesized files should never be manually edited (in fact, *projen* enforces that). To modify your project setup, users interact with rich strongly-typed class and execute *projen* to update their project configuration files.

```
const { AwsCdkTypeScriptApp } = require('projen');
const project = new AwsCdkTypeScriptApp({
  cdkVersion: '1.95.2',
  defaultReleaseBranch: 'main',
  name: 'serverless-datalake-builder',

  cdkDependencies: [
    '@aws-cdk/aws-athena',
    '@aws-cdk/aws-cloud9',
    '@aws-cdk/aws-dynamodb',
    '@aws-cdk/aws-ec2',
    '@aws-cdk/aws-glue',
    '@aws-cdk/aws-iam',
    '@aws-cdk/aws-kinesis',
    '@aws-cdk/aws-quicksight',
    '@aws-cdk/aws-s3',
  ], /* Which AWS CDK modules (those that start with "@aws-cdk/") this app uses. */
  // deps: [], /* Runtime dependencies of this module. */
  // description: undefined, /* The description is just a string that helps people understand the purpose of the package. */
  // devDeps: [], /* Build dependencies for this module. */
  // packageName: undefined, /* The "name" in package.json. */
  // projectType: ProjectType.UNKNOWN, /* Which type of project this is (library/app). */
  // release: undefined, /* Add release management to this project. */
});
project.synth();
```

Try it out

Get started

- cdkworkshop.com
- aws.amazon.com/cdk
- [aws-samples/aws-cdk-examples](https://github.com/aws-samples/aws-cdk-examples)

Contribute

- gitter.im/awslabs/aws-cdk
- [aws/aws-cdk](https://github.com/aws/aws-cdk)
- [aws/jsii](https://github.com/aws/jsii)

Community Resources

- cdkpatterns.com
- cdk8s.io
- awsdk.io
- [eladb/awesome-cdk](https://github.com/eladb/awesome-cdk)

The image displays two side-by-side browser windows. The left window shows the 'AWS CDK Intro Workshop :: AW' website at <https://cdkworkshop.com>. It features a dark blue header with the AWS logo and a sidebar with white text links: 'Prerequisites', 'New Project', 'Hello, CDK!', 'Writing constructs', 'Using construct libraries', 'Clean up', and 'Congrats!'. The right window shows the 'awsdk.io' website at <https://awsdk.io>. It has a white background with a large 'CDK Construct Catalog' heading, a cloud icon containing three cubes, and a search bar labeled 'Search packages...'. Below the search bar is a note: 'This is a community project and is not supported by AWS'. Both browser windows have standard OS X-style toolbars at the top.

Demo Time

Katreena Mullican, Senior Solutions Architect – Python CDK

Mark Senerth, Principal Data Engineer – DMED Data Platforms – Java CDK