



# Building applications in the Cloud using Go and the AWS CDK

*Vincent Lesierse*  
*Solutions Architect, AWS*

Abstractions

AWS CDK, Pulumi

Generators

Troposphere, GoFormation

Declarative

CloudFormation, Terraform

Scripted

`#!/bin/bash`

Manual

Wikis, Playbooks, Console

# AWS Cloud Development Kit



**Familiar**  
Your language  
Just classes and methods



**Tool Support**  
AutoComplete  
Inline documentation



**Abstraction**  
Sane defaults  
Reusable classes

```
vpc := awsec2.NewVpc(stack, jsii.String("MyVPC"), &awsec2.VpcProps{MaxAzs: jsii.Number(2)})
cluster := ecs.NewCluster(stack, jsii.String("Cluster"), &ecs.ClusterProps{Vpc: vpc})
serv := ecs.NewService(stack, jsii.String("Service"), &ecs.ServiceProps{Cluster: cluster,
    DomainName *string `json:"domainName"`
    // The Route53 hosted zone for the domain, e.g. "example.com.".
    &ecs.ApplicationLoadBalancedFargateServiceProps{
        field DomainName *string
        (awsecspatterns.ApplicationLoadBalancedFargateServiceProps).DomainName on pkg.go.dev
    },
    The domain name for the service, e.g. "api.example.com.". Experimental.
    DomainName: jsii.String("api.example.com."),
},
)

scaling := service.Service().AutoScaleTaskCount(&awsapplicationautoscaling.EnableScalingProps{
    jsii.Number(2),
    jsii.Number(5),
})

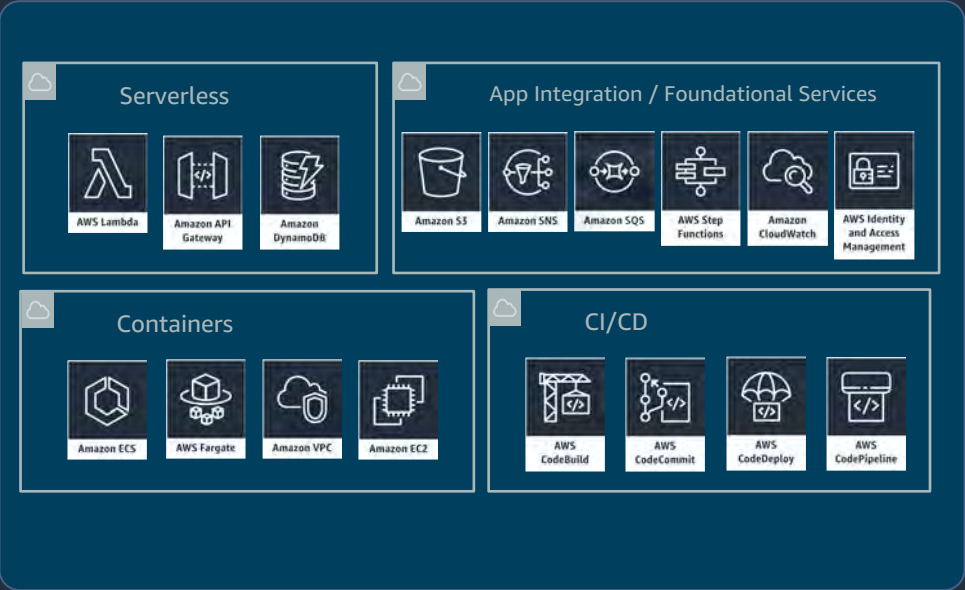
scaling.ScaleOnCpuUtilization(jsii.String("CPUScaling"), &awsecs.CpuUtilizationScalingProps{
    TargetUtilizationPercent: jsii.Number(50),
    ScaleInCooldown:          awscdk.Duration_Seconds(jsii.Number(60)),
    ScaleOutCooldown:         awscdk.Duration_Seconds(jsii.Number(60)),
})
```



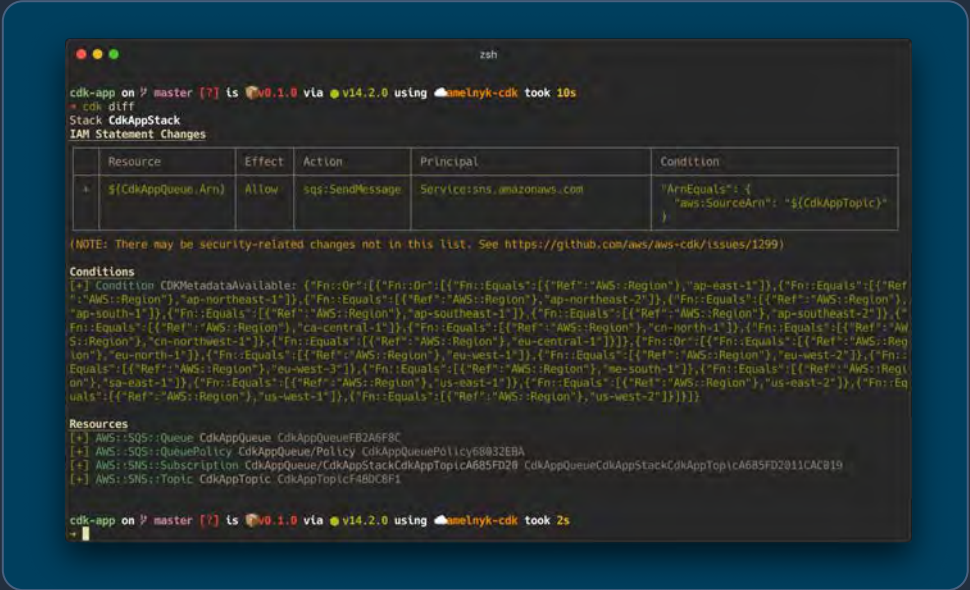
# Main Components



Core Framework



AWS Construct Library



AWS CDK CLI

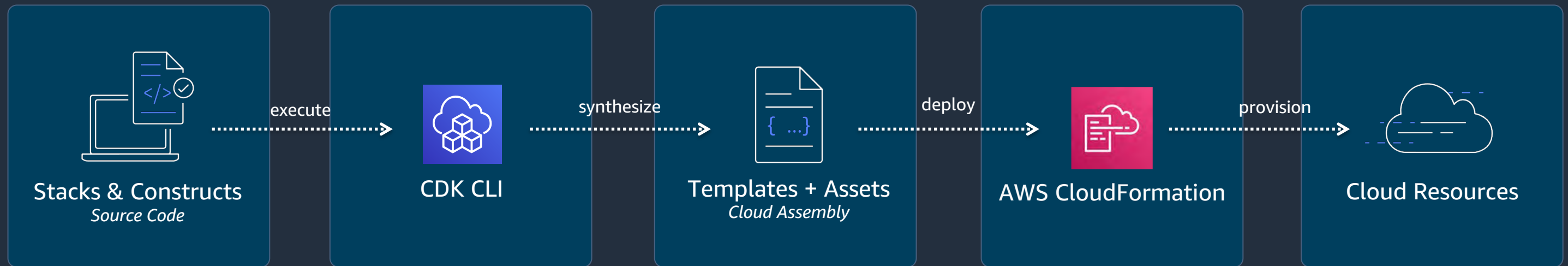







# CDK Example

```
vpc := awsec2.NewVpc(stack, jsii.String("MyVPC"), &awsec2.VpcProps{MaxAzs: jsii.Number(2)})
cluster := awsecs.NewCluster(stack, jsii.String("Cluster"), &awsecs.ClusterProps{Vpc: vpc})

service := awsecspatterns.NewApplicationLoadBalancedFargateService(stack, jsii.String("SampleApp"),
    &awsecspatterns.ApplicationLoadBalancedFargateServiceProps{
        Cluster: cluster,
        TaskImageOptions: &awsecspatterns.ApplicationLoadBalancedTaskImageOptions{
            Image: awsecs.AssetImage_FromRegistry(jsii.String("amazon/amazon-ecs-sample"), nil),
        },
    },
)
```

# Development Workflow



 `cdk init app --language=go` // create new project  
 `go build` // build project  
 `cdk synth` // create templates and assets  
 `cdk diff` // check what will change  
 `cdk deploy` // push changes to the cloud

# CDK Constructs

# AWS Construct Library

 Serverless




**AWS Lambda**




**Amazon API Gateway**




**Amazon DynamoDB**


 Application Integration / Foundational Services




**Amazon S3**




**Amazon SNS**




**Amazon SQS**



**AWS Step Functions**



**Amazon CloudWatch**



**AWS Identity and Access Management**

 Containers



**Amazon ECS**



**AWS Fargate**



**Amazon VPC**



**Amazon EC2**

 CI/CD



**AWS CodeBuild**



**AWS CodeCommit**



**AWS CodeDeploy**



**AWS CodePipeline**



# Construct Levels

L3+

Purpose-built constructs

Opinionated abstractions

L2

AWS Constructs

High level service constructs

L1

CloudFormation Resources

Automatically generated

```
awss3.NewCfnBucket(stack, jsii.String("myBucket"), &awss3.CfnBucketProps{ BucketName: jsii.String("my-bucket")})
```



- 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
```

# L2

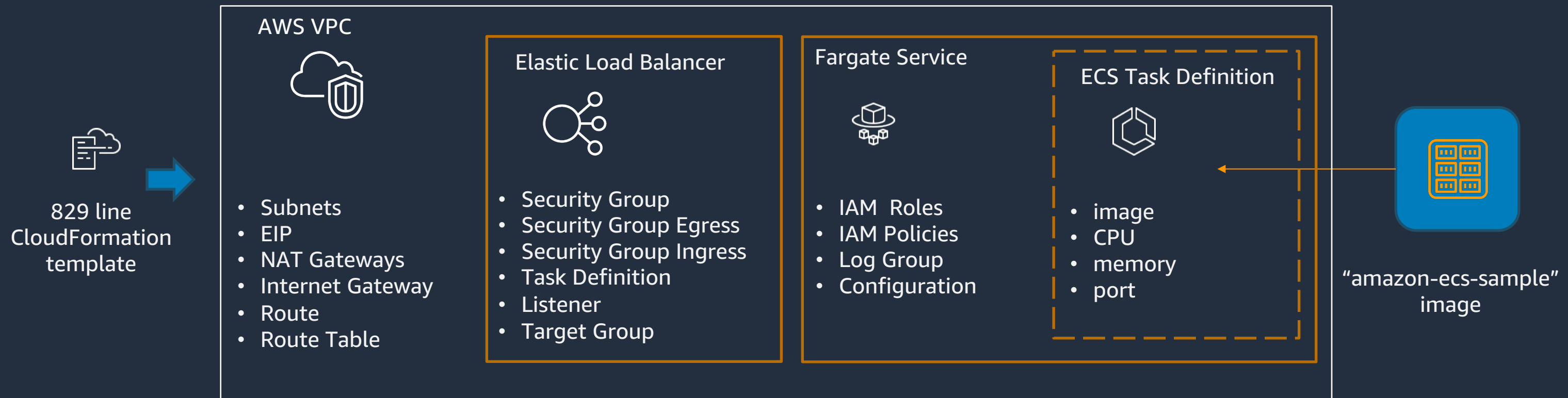
```
awsec2.NewVpc(stack, jsii.String("MyVPC"), &awsec2.VpcProps{ })
```

 cdk synth

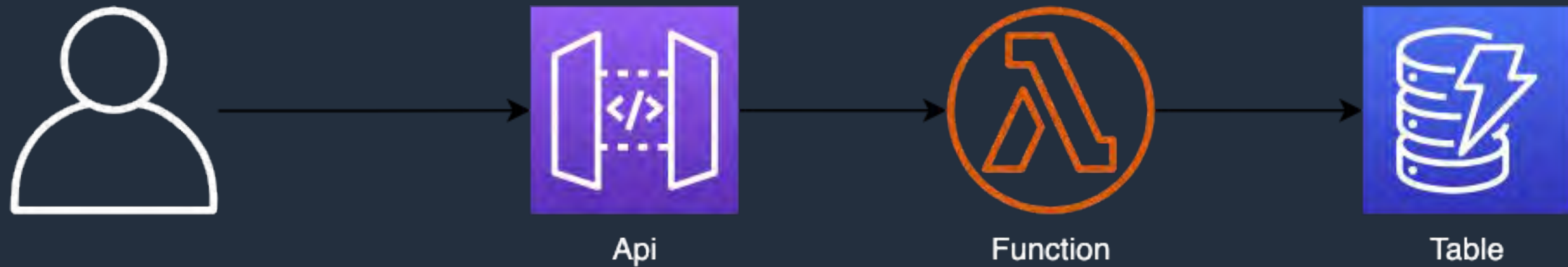


- Ready-to-use VPC setup
- 65536 IPs split equally between 4 subnets
- If you provide a region → adjusted to 3 AZs
- Everything is optional, change any parameter
- Sane default values

```
awsecspatterns.NewApplicationLoadBalancedFargateService(stack, jsii.String("SampleApp"),
    &awsecspatterns.ApplicationLoadBalancedFargateServiceProps{
        Cluster: cluster,
        TaskImageOptions: &awsecspatterns.ApplicationLoadBalancedTaskImageOptions{
            Image: awsecs.AssetImage_FromRegistry(jsii.String("amazon/amazon-ecs-sample"), nil),
        },
    },
)
```



# Deploying a Go application with CDK





# L3 - AWS Solutions Constructs

### Filter AWS Solutions Constructs by:

[Clear all filters](#)

▼ Product

- ☐ Amazon API Gateway
- ☐ Amazon CloudFront
- ☐ AWS CloudTrail
- ☐ Amazon CloudWatch
- ☐ Amazon Cognito
- ☐ Amazon DynamoDB
- ☐ Amazon Elasticsearch
- ☐ Amazon Kinesis
- ☐ AWS Lambda
- ☐ Amazon S3
- ☐ Amazon Simple Notification Service (SNS)
- ☐ AWS Step Functions
- ☐ Amazon Simple Queue Service (SQS)

1-15 (25)

Sort by: Name (A->Z) ▼

API GATEWAY | CLOUDWATCH | DYNAMODBNEW

### API Gateway to DynamoDB

Built by AWS

This AWS Solutions Construct pattern implements an Amazon API Gateway REST API connected to Amazon DynamoDB table.

Publish date

June 2020

API GATEWAY | CLOUDWATCH | LAMBDANEW

### API Gateway to Lambda

Built by AWS

This AWS Solutions Construct pattern implements an Amazon API Gateway REST API connected to an AWS Lambda function pattern.

Publish date

June 2020

API GATEWAY | SQSNEW

### API Gateway to SQS

Built by AWS

This AWS Solutions Construct pattern implements an Amazon API Gateway connected to an Amazon SQS queue pattern.

Publish date

June 2020

CLOUDWATCH | LAMBDANEW

### AWS Events Rule to Lambda

Built by AWS

This AWS Solutions Construct pattern implements an AWS Events rule and an AWS Lambda function.

CLOUDWATCH | STEP FUNCTIONSNEW

### AWS Events Rule to Step Function

Built by AWS

This AWS Solutions Construct pattern implements an AWS Events rule and an AWS Step function.


CLOUDTRAIL | CLOUDWATCH | S3 | STEP FUNCTIONSNEW

### AWS S3 to Step Function

Built by AWS

This AWS Solutions Construct pattern implements an Amazon S3 bucket connected to an AWS Step Function.

© 2021, Amazon Web Services, Inc. or its Affiliates.





# Open Ecosystem

## Awesome CDK

A curated list of awesome projects related to the AWS Cloud Development Kit (CDK).

DISCLAIMER: this is a personal project and not an official AWS project.

### Contents

- Construct Libraries
- Tools
- Training Materials and Sample Code
- Blog Posts & Talks
- Tips & Tricks
- Related Projects

### Construct Libraries

This section includes code libraries in various programming languages.

- [aws-delivlib](#): synthesizes CI/CD pipelines for m...
- [cdk-tweet-queue](#): fills up an SQS queue with t...
- [cdk-dynamo-table-viewer](#): exposes the conten...
- [cdk-tweet-sentiment](#): identify sentiments in tw...
- [cdk-watchful](#): automatic dashboards and alarm...

## CDK-Patterns

@CdkPatterns

Opensource collection of Python/TypeScript with AWS CDK. Check back weekly for new

[cdkpatterns.com](#) Joined January 20

42 Following 365 Followers

### Tweets

Pinned Tweet

**CDK-Patterns @CdkPatterns · Feb**  
Announcing "The EventBridge ATM awesome blog post - [github.com/cdk-patterns/EventBridge-ATM](#) clone, build, deploy [#RoutingRules](#)

The diagram illustrates the EventBridge ATM architecture. It starts with an 'ATM service' (represented by an AWS Lambda icon) which triggers 'ATM transaction events' (represented by a series of pink hexagons). These events are sent to an 'Amazon EventBridge event bus'. From the event bus, the events are processed by 'Rules' (represented by a pink hexagon). The rules then trigger 'Downstream AWS Lambda functions' (represented by an AWS Lambda icon). A feedback loop labeled '3. Not approved' returns from the downstream functions to the event bus.

### Tweets & replies

2 8 43

# Try it out

## Get started

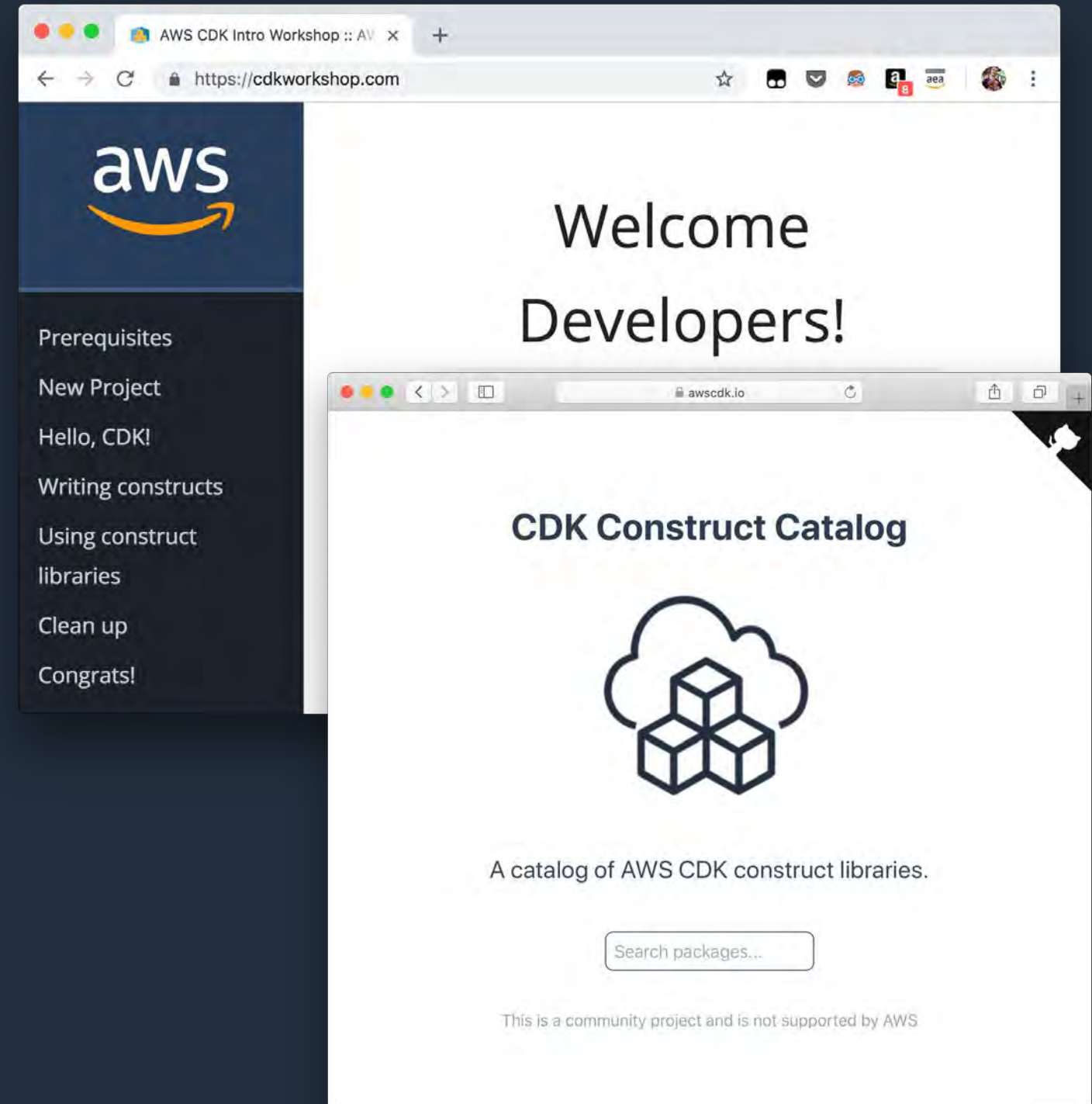
- [cdkworkshop.com](https://cdkworkshop.com)
- [aws.amazon.com/cdk](https://aws.amazon.com/cdk)
- [aws-samples/aws-cdk-examples](https://aws-samples/aws-cdk-examples)

## Contribute

- [aws/aws-cdk](https://aws/aws-cdk)
- [aws/jsii](https://aws/jsii)

## Community Resources

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





# Thank You!

*Vincent Lesierse @vlesierse*