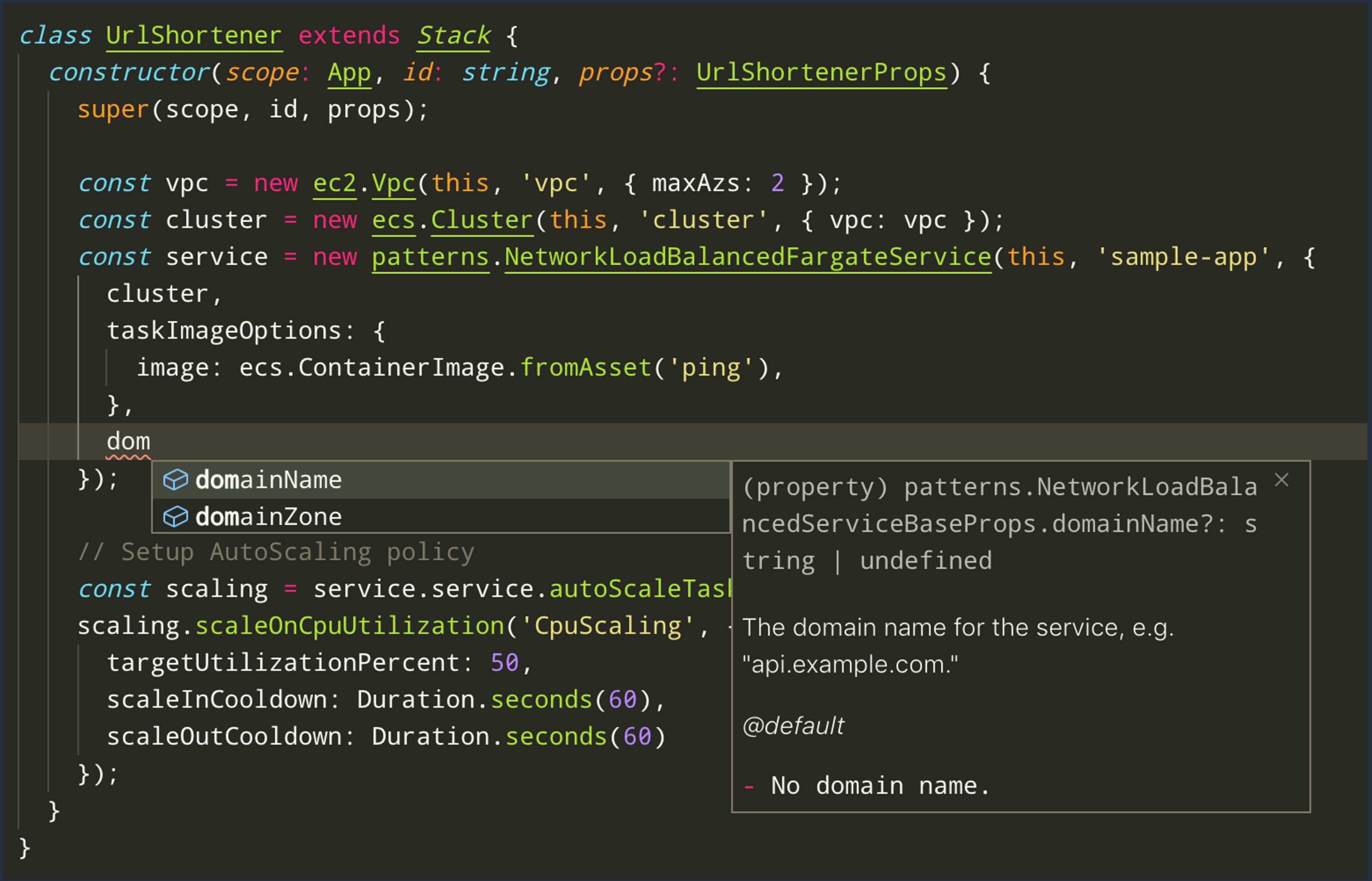
AWS Cloud Development Kit

CDK for Complex Enterprise Applications

Eric Z. Beard
Rico Huijbers

AWS Cloud Development Kit (CDK)

A multi-language software development framework for modeling cloud infrastructure as reusable components



```
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.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling', 
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

```
(property) patterns.NetworkLoadBala
ncedServiceBaseProps.domainName?: s
tring | undefined
```

The domain name for the service, e.g. "api.example.com."

@default

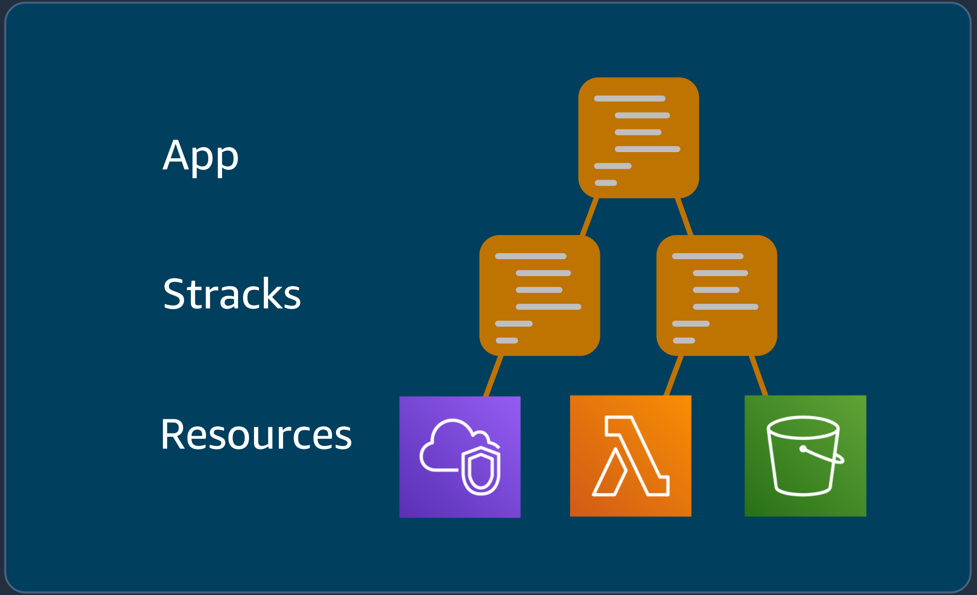
- No domain name.

Familiar
Your language
Just code

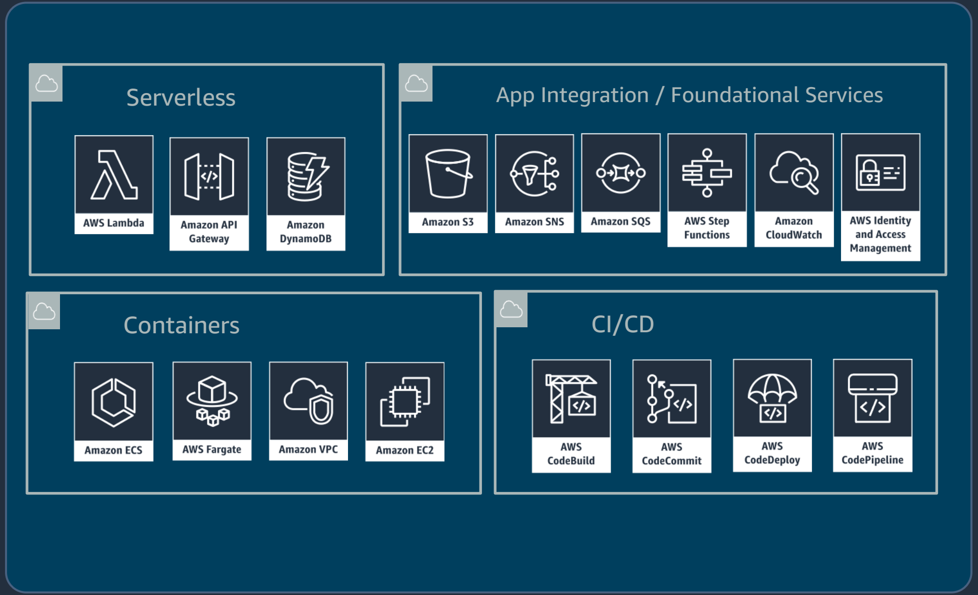
Tool Support
AutoComplete
Inline documentation

Abstraction
Sane defaults
Reusable classes

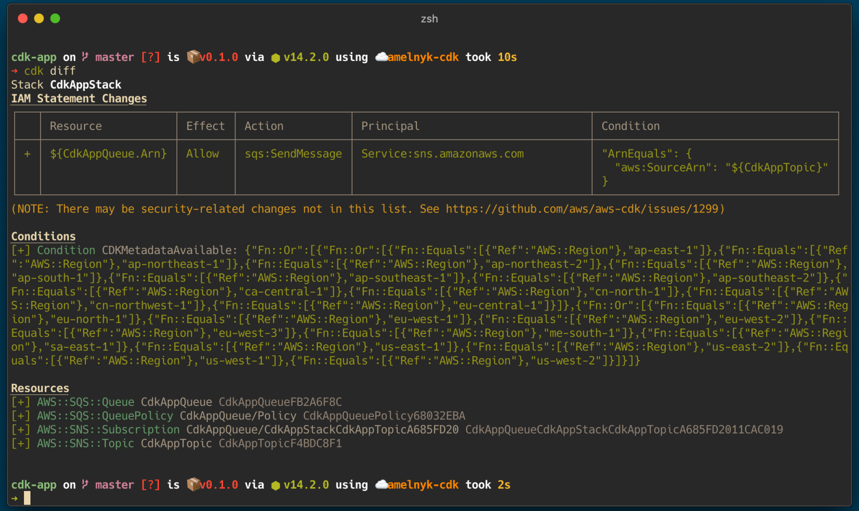
Main Components



Core Framework



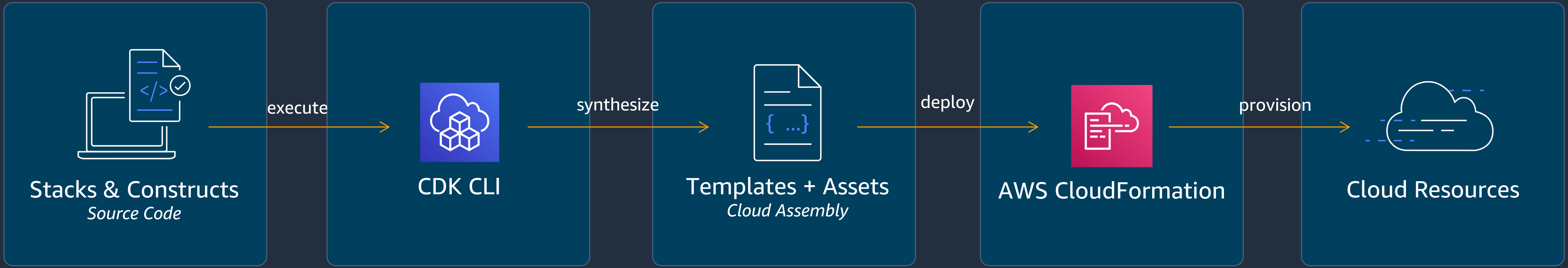
AWS Construct Library



```
zsh

cdk-app on master [?] is v0.1.0 via v14.2.0 using amelnyk-cdk took 10s
→ cdk diff
Stack CdkAppStack
IAM Statement Changes
┌───┬──────────────────────┬────────┬─────────────────┬─────────────────────────────┬─────────────────────────────────────┐
│   │ 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::Or":[{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-east-1"]},{"Fn::Equals":[{"Ref
":"AWS::Region"},"ap-northeast-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-northeast-2"]},{"Fn::Equals":[{"Ref":"AWS::Region"},
"ap-south-1"]},{"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":"AW
S::Region"},"cn-northwest-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-central-1"]}]},{"Fn::Or":[{"Fn::Equals":[{"Ref":"AWS::Reg
ion"},"eu-north-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-west-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-west-2"]},{"Fn::
Equals":[{"Ref":"AWS::Region"},"eu-west-3"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"me-south-1"]},{"Fn::Equals":[{"Ref":"AWS::Regi
on"},"sa-east-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"us-east-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"us-east-2"]},{"Fn::Eq
uals":[{"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 CdkAppQueueCdkAppStackCdkAppTopicA685FD2011CAC019
[+] AWS::SNS::Topic CdkAppTopic CdkAppTopicF4BDC8F1


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

AWS CDK CLI

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 your account
```

Construct Levels

L3+	Purpose-built constructs	Opinionated abstractions
L2	AWS Constructs	High level service constructs
L1	CloudFormation Resources	Automatically generated

CDK Philosophy – From This...

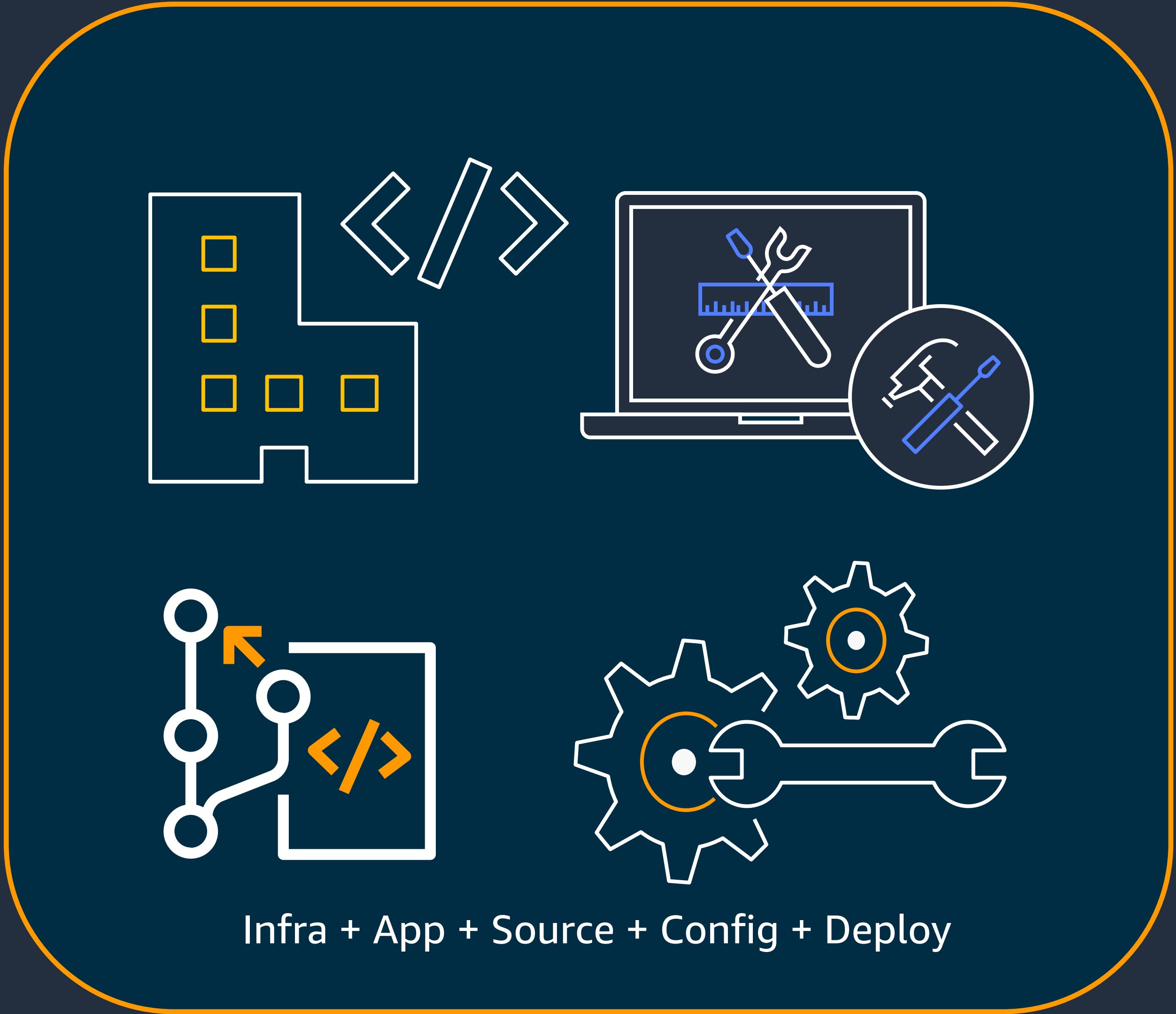
Infrastructure

Application Code

Source Control

Config and Deployment

CDK Philosophy – To This


Infra + App + Source + Config + Deploy

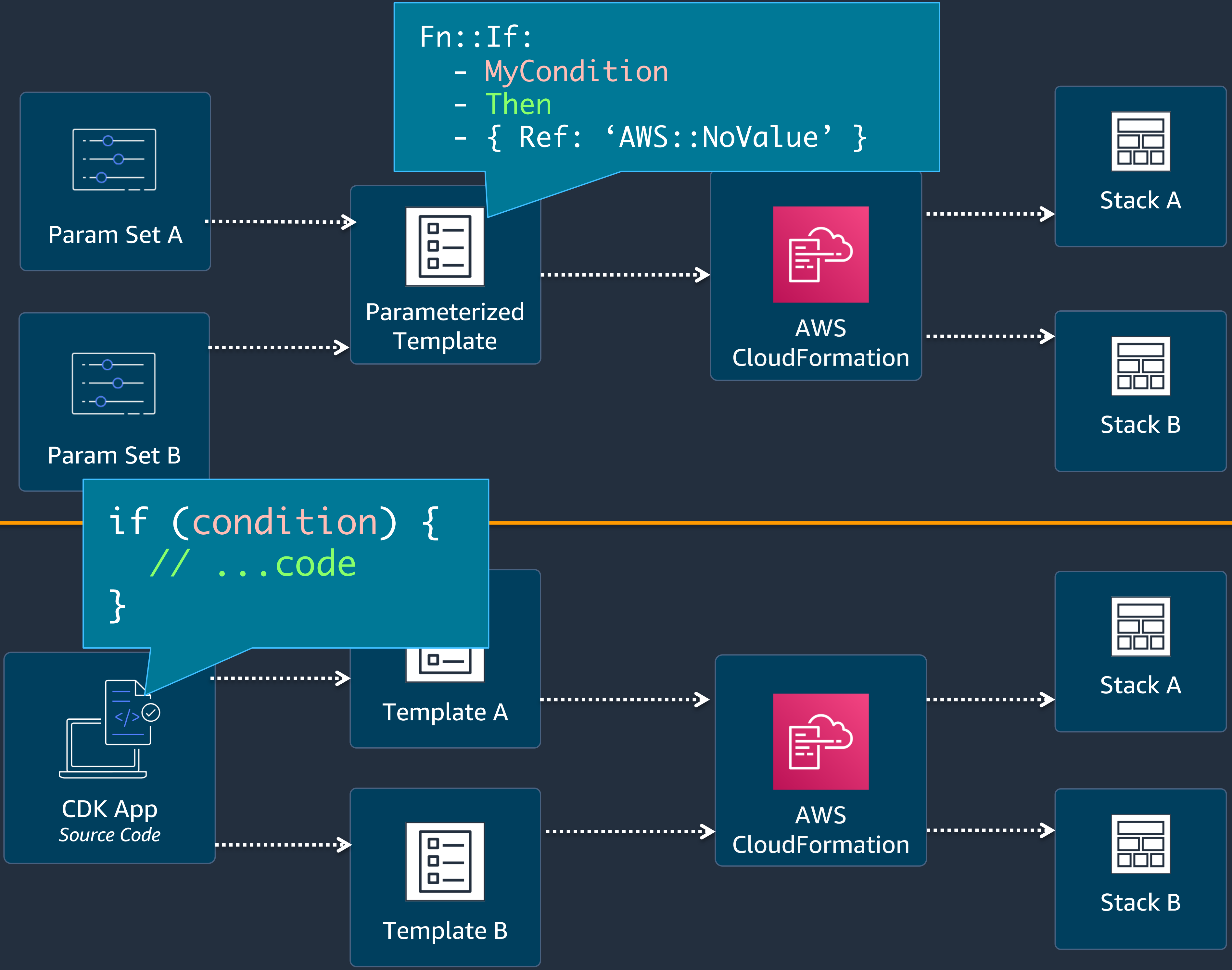
Paradigm Shift

CloudFormation

Parameters and intrinsic functions

CDK

Typed OO language: loops, conditions, inheritance, etc



```
Fn::If:
  - MyCondition
  - Then
  - { Ref: 'AWS::NoValue' }
```

```
if (condition) {
  // ...code
}
```

Best Practices, Part One

the organization

Best Practice - Cloud Center of Excellence (CCoE)

A Cloud Center of Excellence (CCoE)

is a corporate group or team

that leads other employees and the organization as a whole

in cloud adoption, migration and operation.

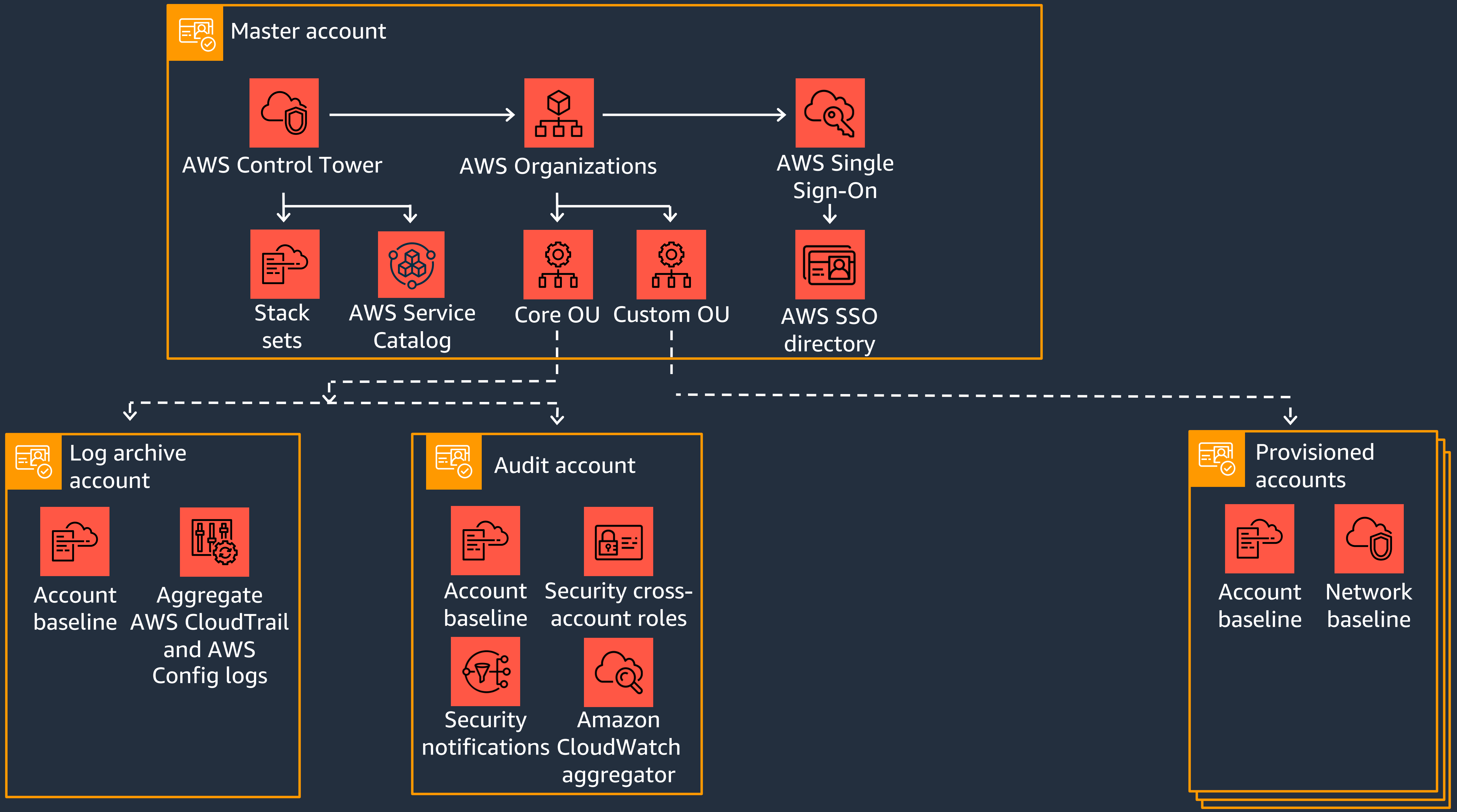
Known by many names:

- Cloud Competency/Capability Center
- Cloud Infrastructure Team
- Cloud Strategy & Enablement Team
- Cloud Tiger Team
- DevOps Team

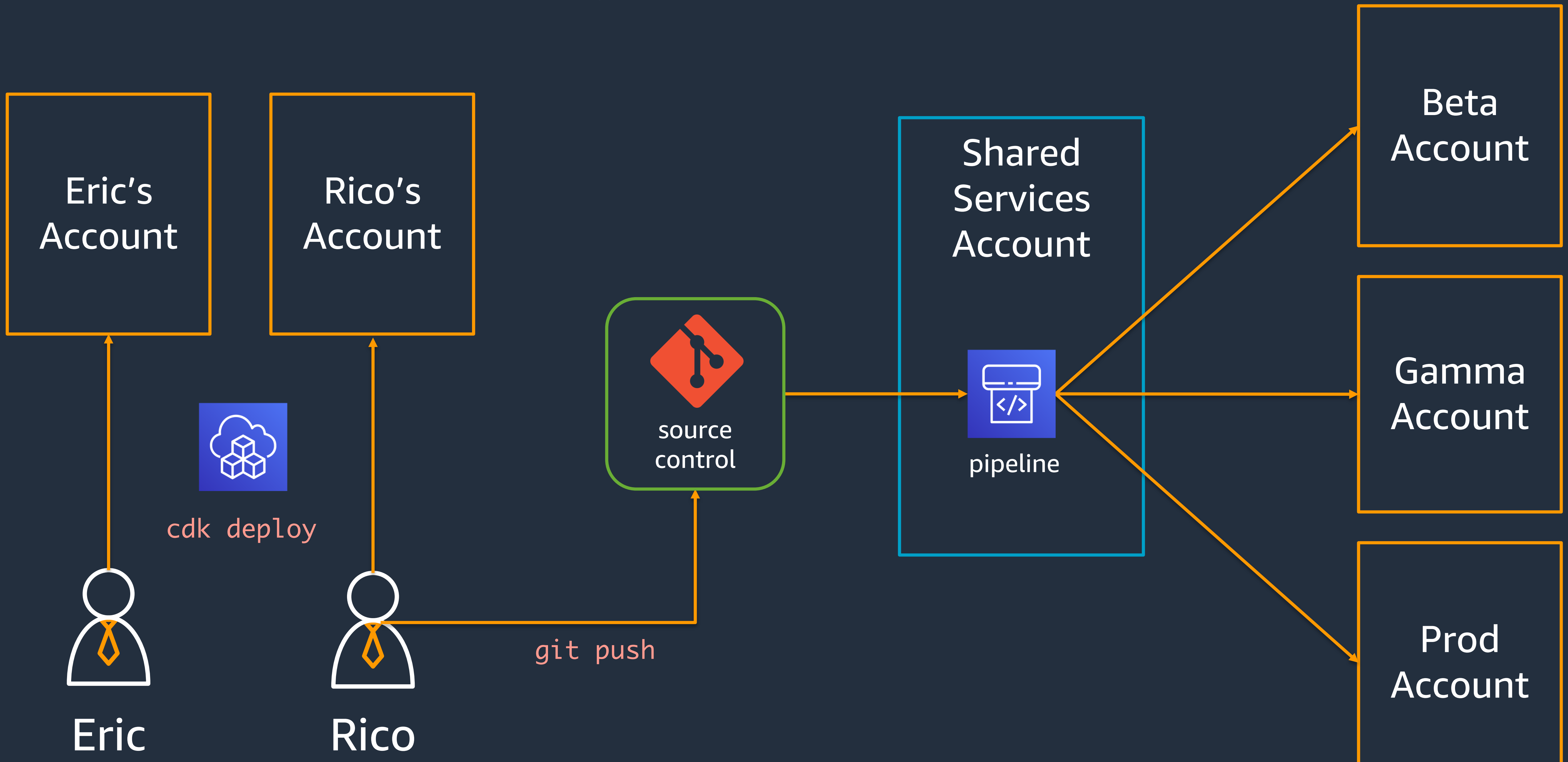
Realm of CCoE:

Research
Evangelize
Apply
Lead
Mentor

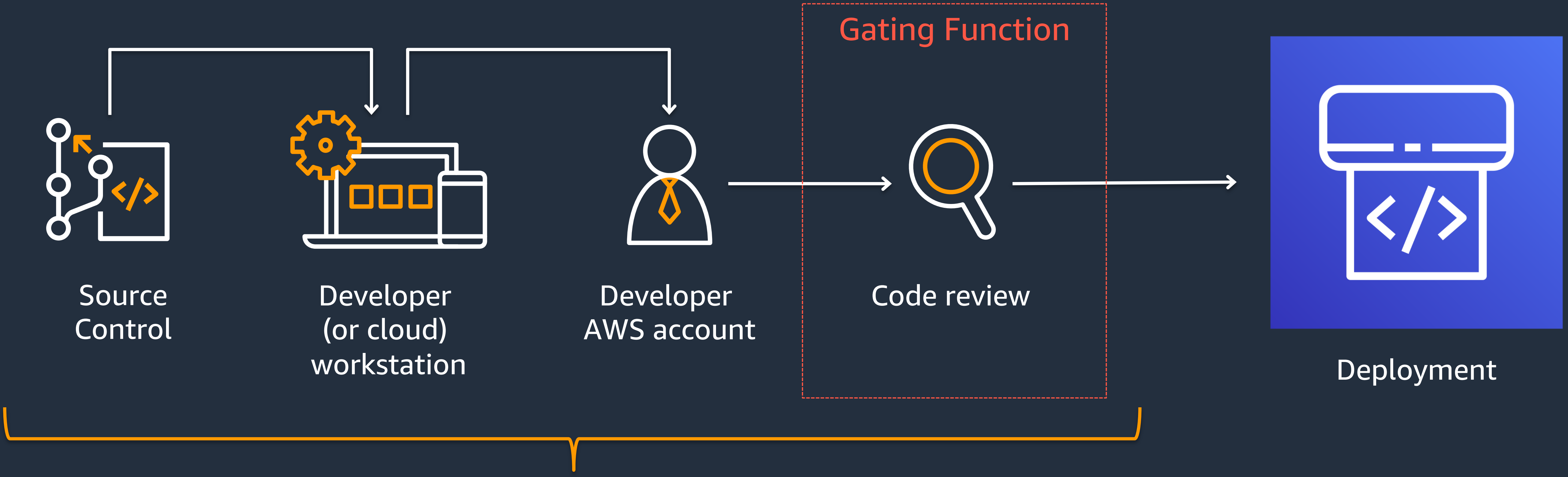
Set up an AWS landing zone



Best Practice – Deploy to multiple accounts

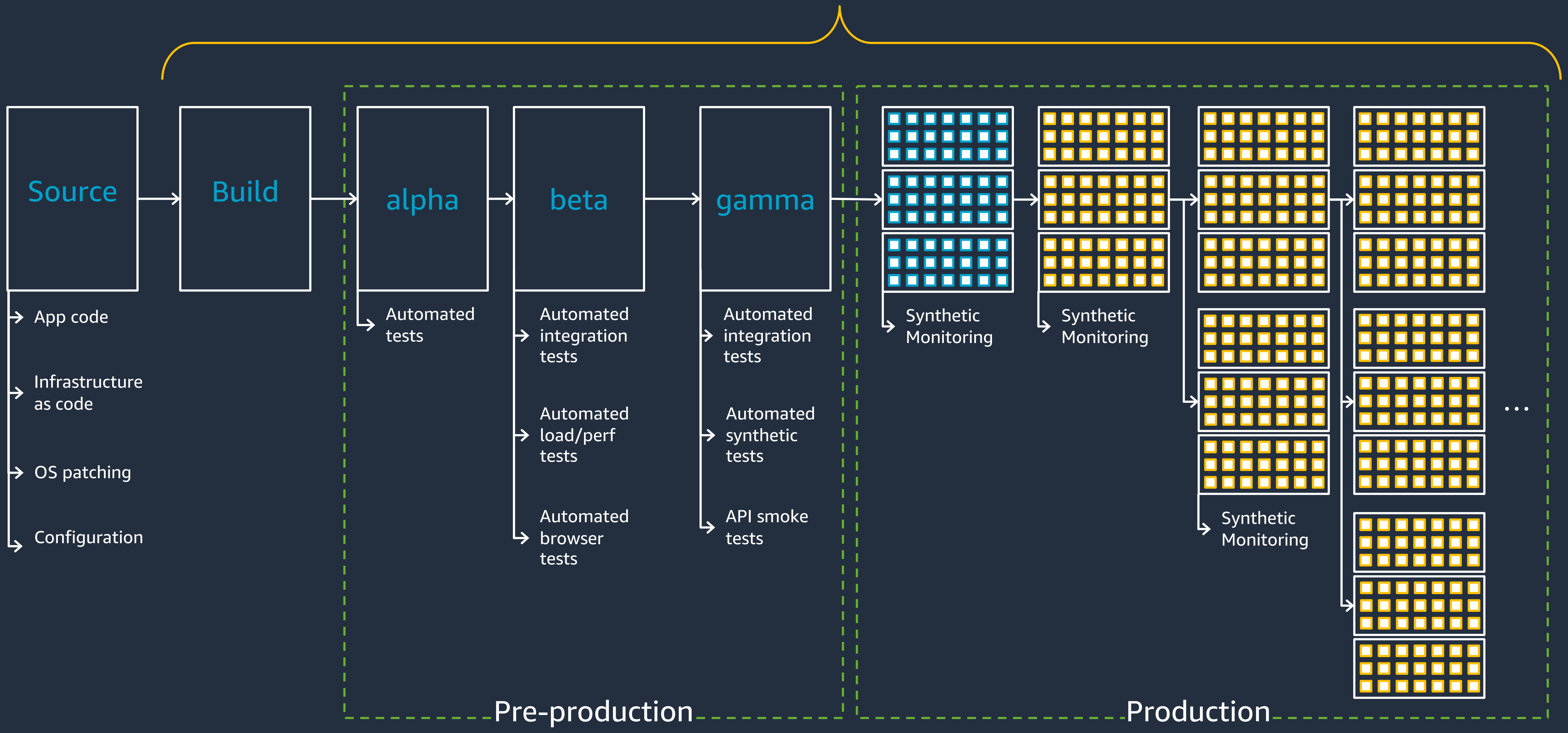


Best Practice – Use code reviews as a gate



This part should be fast

Best Practice – Fully Automate Deployments



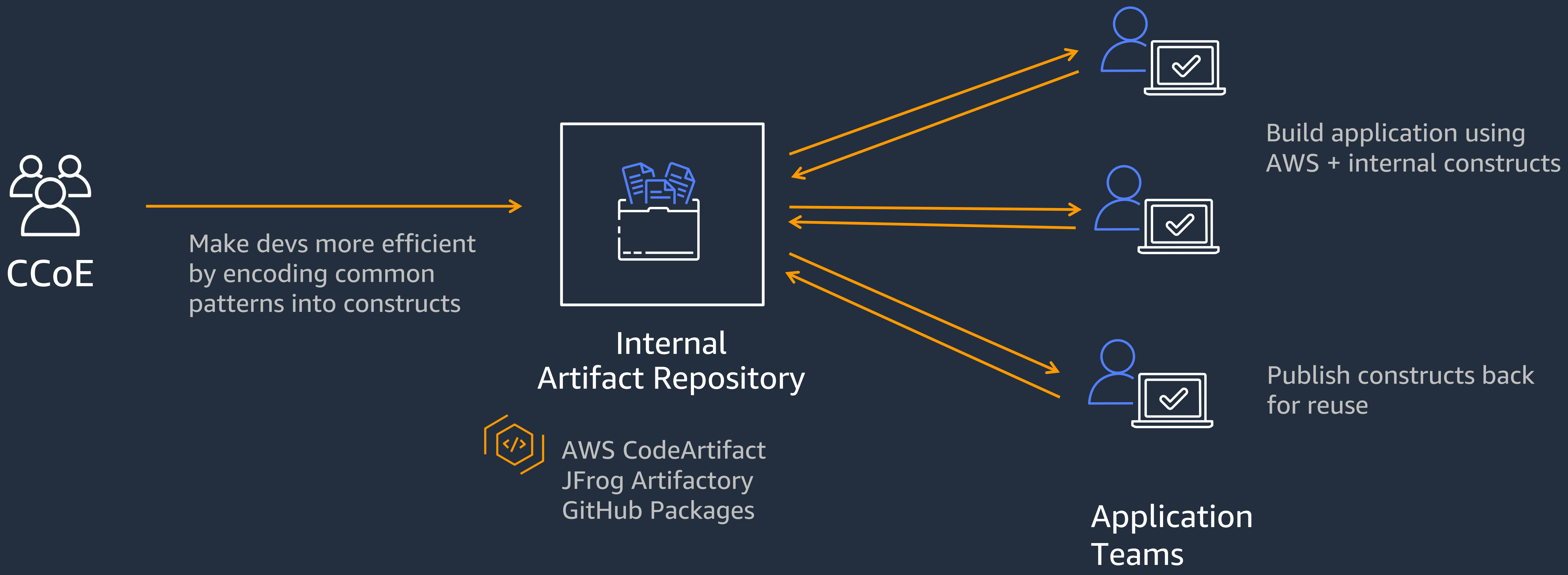
Best Practice – Measure Everything

Business metrics	Operational metrics	Input goals	Enablement
Growth	Errors	Features	Principal reviews
Usage	Throttling	Use cases	Security training
Feedback	Failed deployments	Performance	Ops training
	Performance		

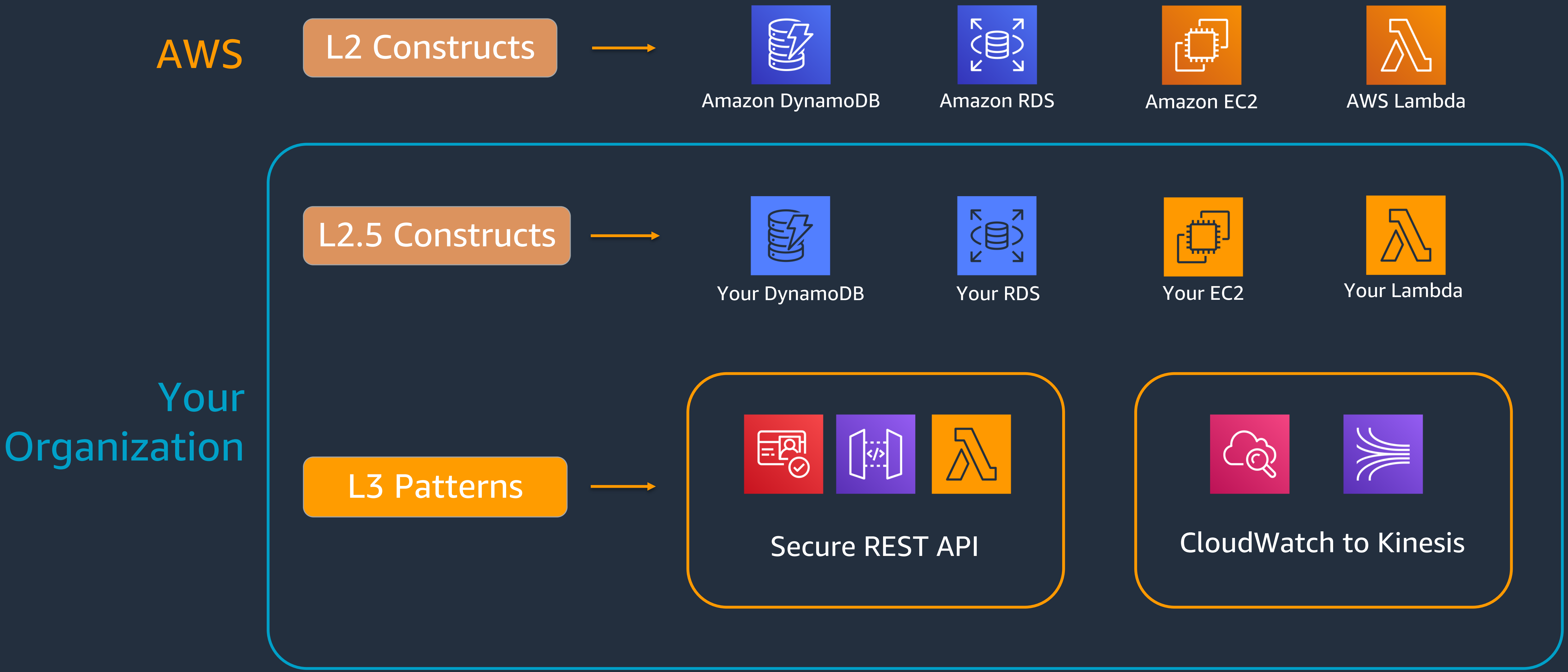
Best Practices, Part Two

CDK

Construct libraries


CCoE
Make devs more efficient by encoding common patterns into constructs
Internal Artifact Repository
AWS CodeArtifact
JFrog Artifactory
GitHub Packages
Build application using AWS + internal constructs
Publish constructs back for reuse
Application Teams

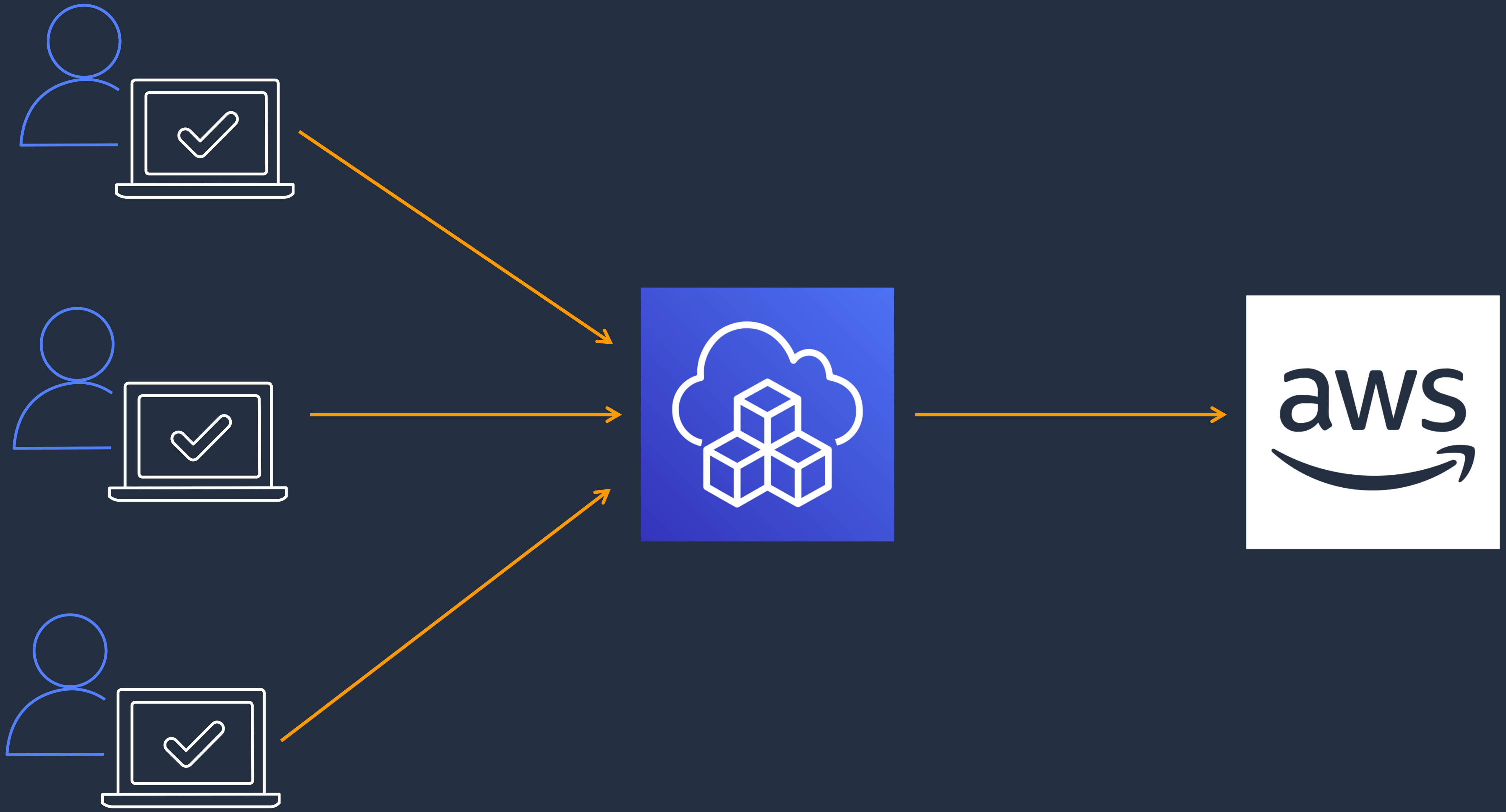
Construct libraries



Construct libraries

Do	*Don't*
✅ Write and publish using jsii	
✅ Vend Constructs	❌ Vend Stacks
✅ Configure with properties	❌ Read environment variables
✅ Unit test your infrastructure	❌ Change ids of stateful resources
✅ Use Constructs for convenience	❌ Use Constructs for compliance

Application Development

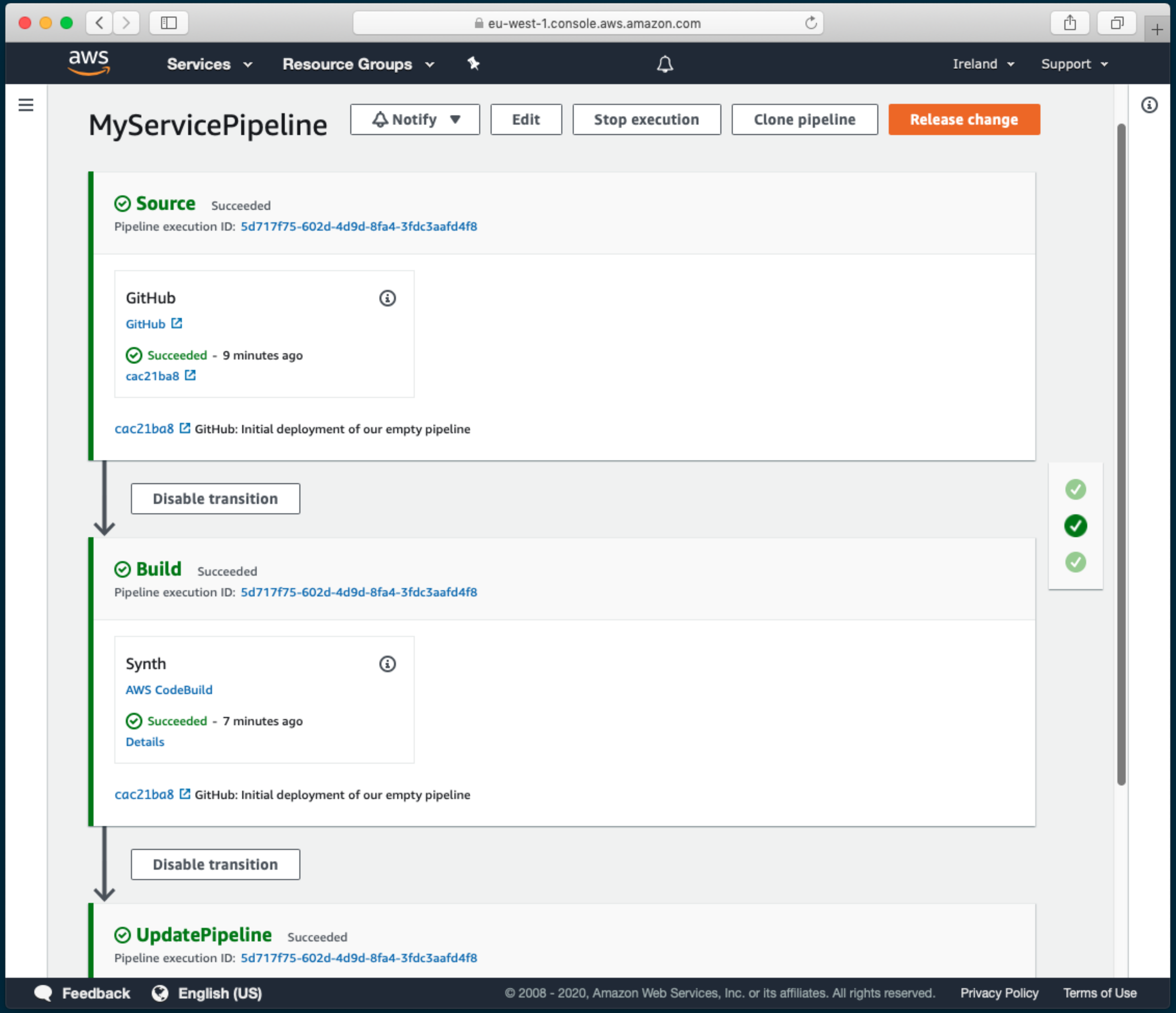


CDK Applications

Do	*Don't*
✅ Parameterize your app for generated resource names	❌ Rely on physical names
✅ Separate into Stacks because of deployment properties	❌ Separate because it *feels* like you should
✅ Make your app deterministic (`cdk.context.json`)	❌ Do network lookups
✅ Allow CDK to manage roles	❌ Import pre-created roles
✅ Model all production Stages	❌ Parameterize app to be a Stage

CDK Pipelines

Continuous delivery for AWS CDK applications



MyServicePipeline

Source Succeeded
Pipeline execution ID: 5d717f75-602d-4d9d-8fa4-3fdc3aafd4f8

GitHub
GitHub
Succeeded - 9 minutes ago
cac21ba8

cac21ba8 GitHub: Initial deployment of our empty pipeline

Build Succeeded
Pipeline execution ID: 5d717f75-602d-4d9d-8fa4-3fdc3aafd4f8

Synth
AWS CodeBuild
Succeeded - 7 minutes ago
Details

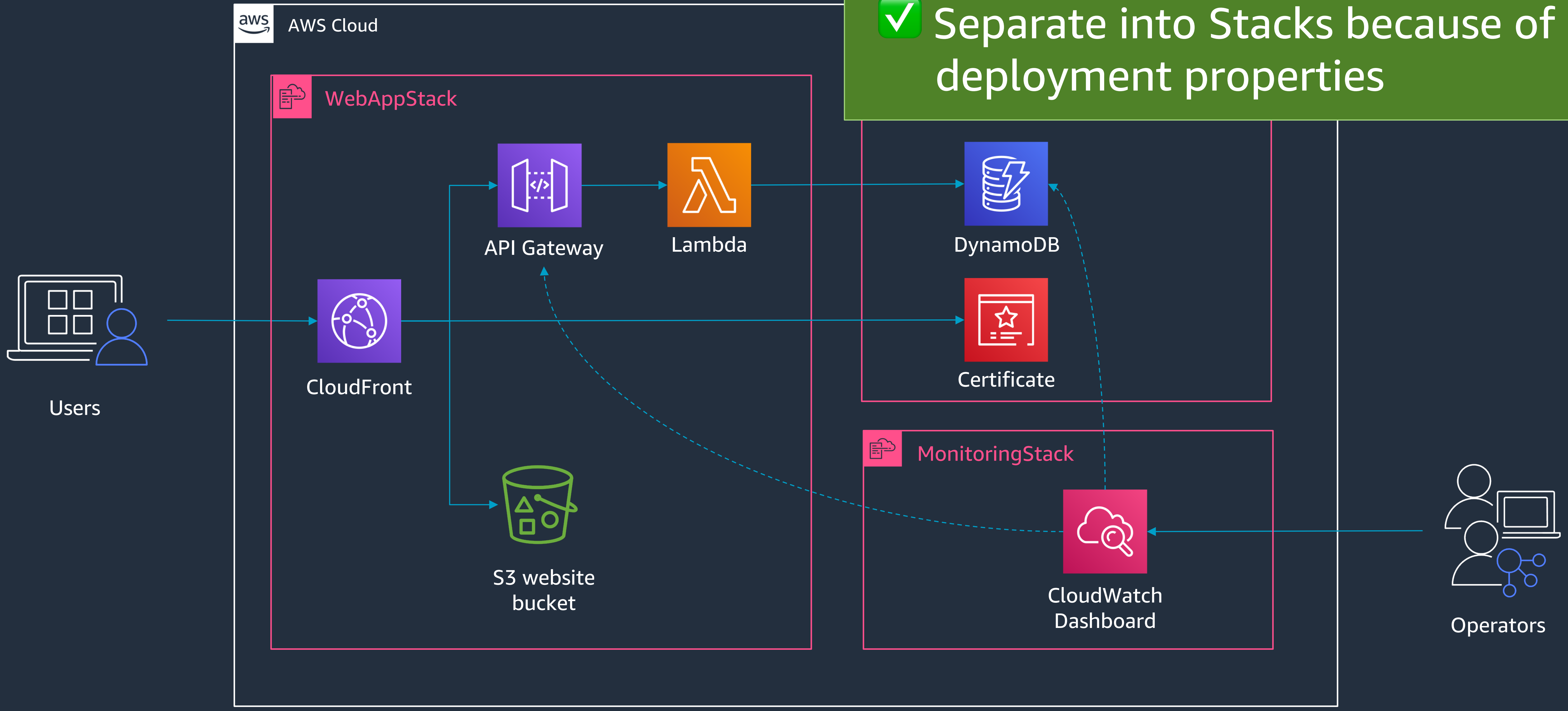
cac21ba8 GitHub: Initial deployment of our empty pipeline

UpdatePipeline Succeeded
Pipeline execution ID: 5d717f75-602d-4d9d-8fa4-3fdc3aafd4f8

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

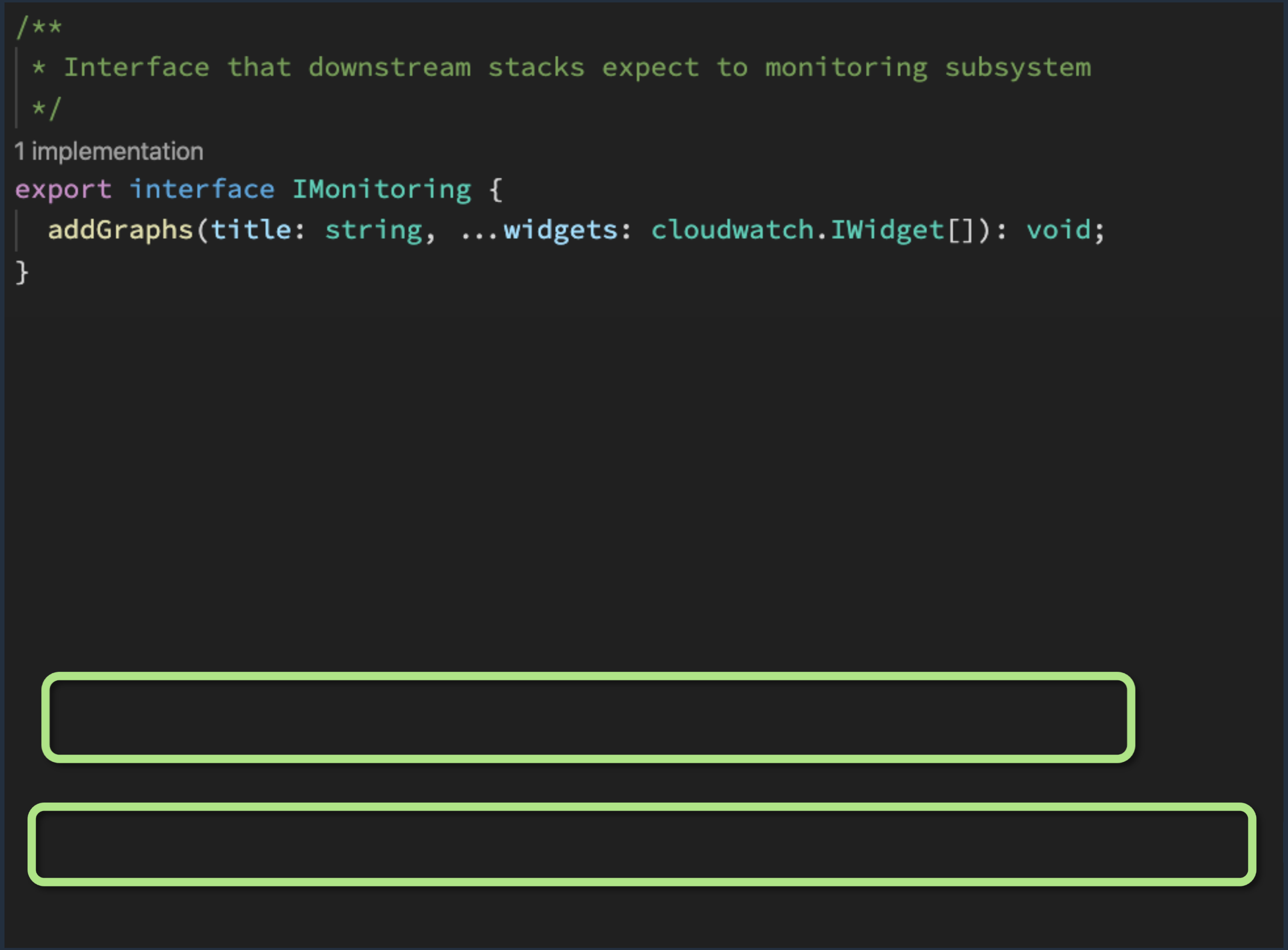
Code Samples

Sample Application – A dynamic website



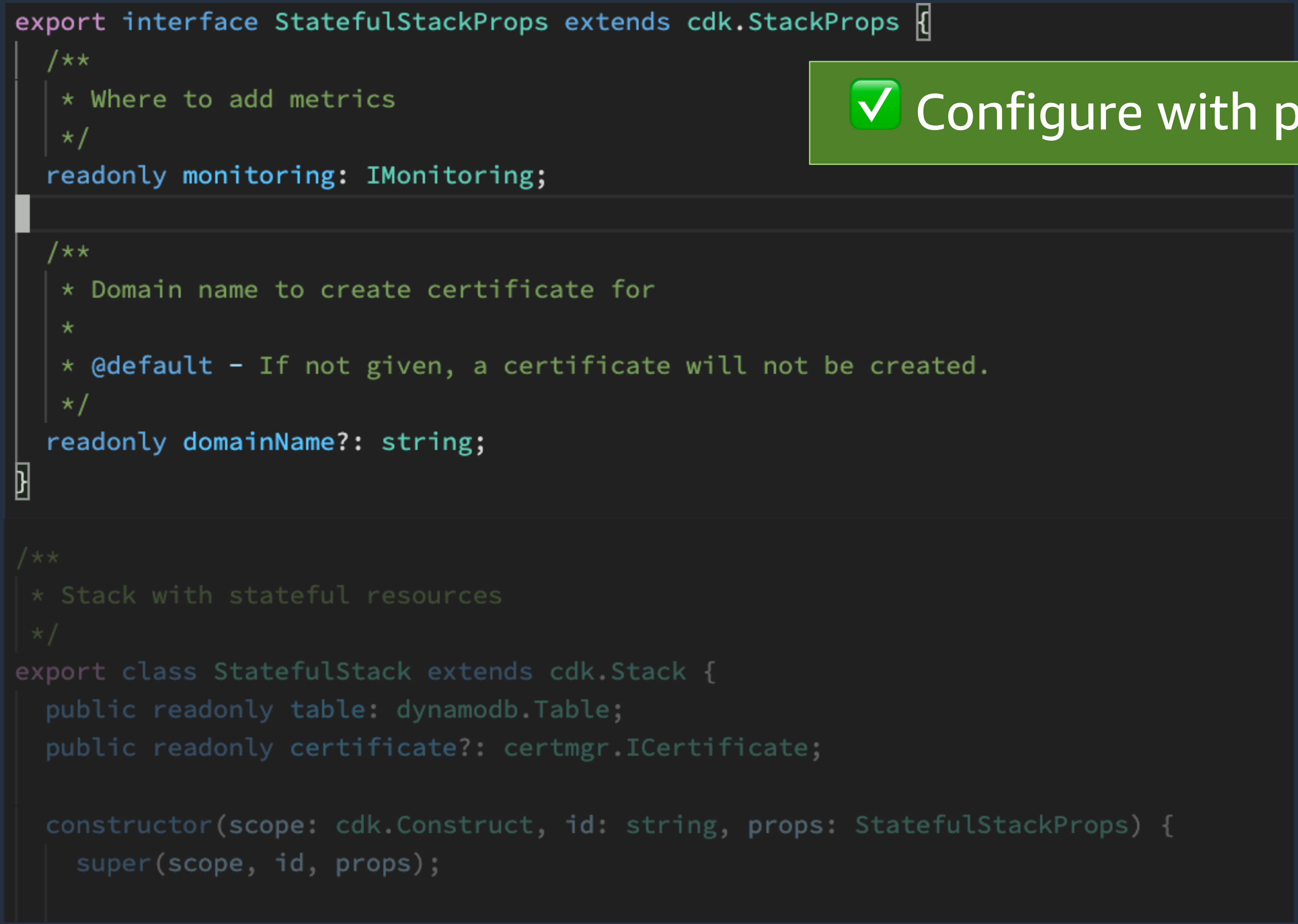
✅ Separate into Stacks because of deployment properties

MonitoringStack



```
/**
 * Interface that downstream stacks expect to monitoring subsystem
 */
1 implementation
export interface IMonitoring {
  addGraphs(title: string, ...widgets: cloudwatch.IWidget[]): void;
}
```

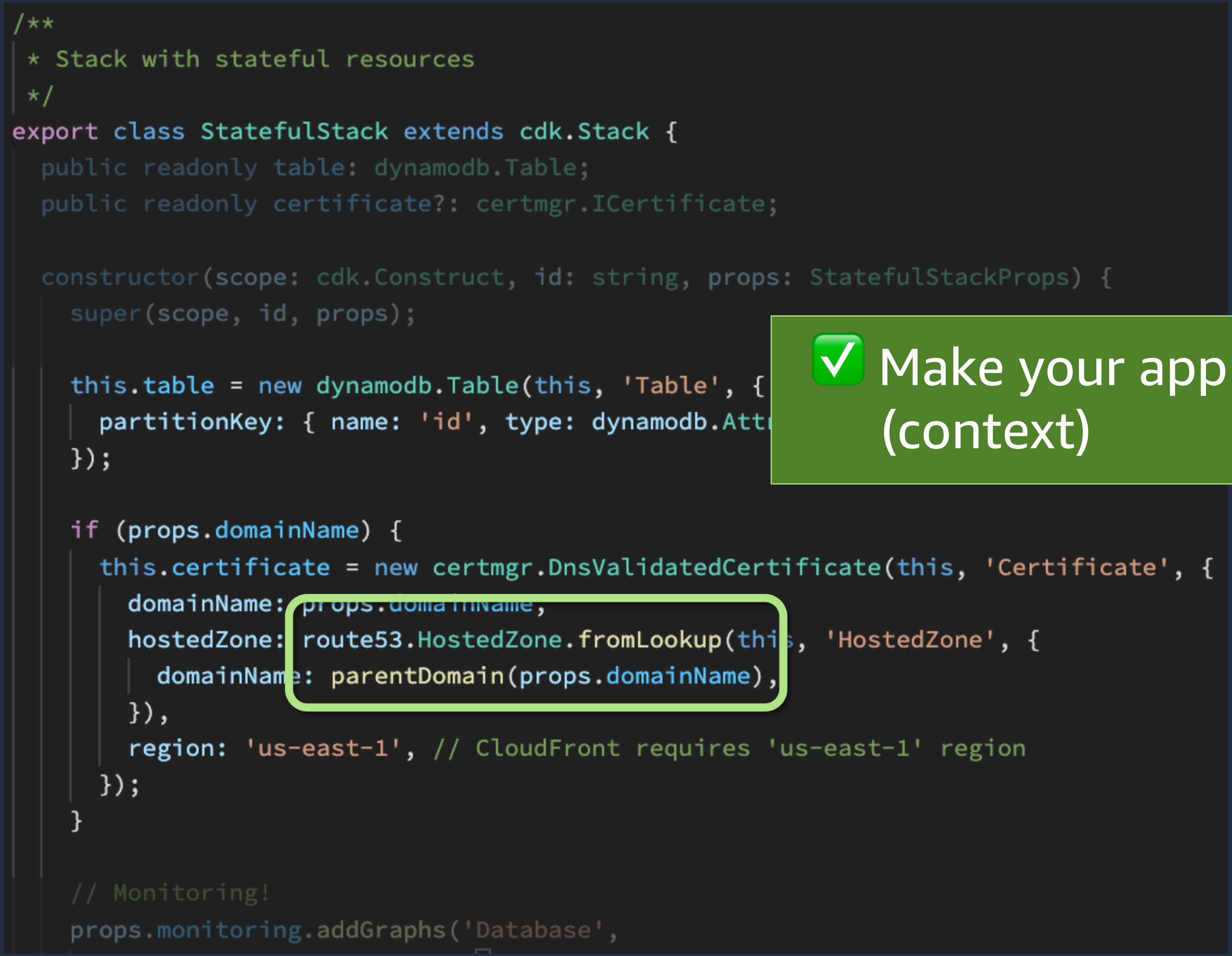
StatefulStack



```
export interface StatefulStackProps extends cdk.StackProps {
  /**
   * Where to add metrics
   */
  readonly monitoring: IMonitoring;

  /**
   * Domain name to create certificate for
   *
   * @default - If not given, a certificate will not be created.
   */
  readonly domainName?: string;
}

/**
 * Stack with stateful resources
 */
export class StatefulStack extends cdk.Stack {
  public readonly table: dynamodb.Table;
  public readonly certificate?: certmgr.ICertificate;

  constructor(scope: cdk.Construct, id: string, props: StatefulStackProps) {
    super(scope, id, props);
```

✅ Configure with properties

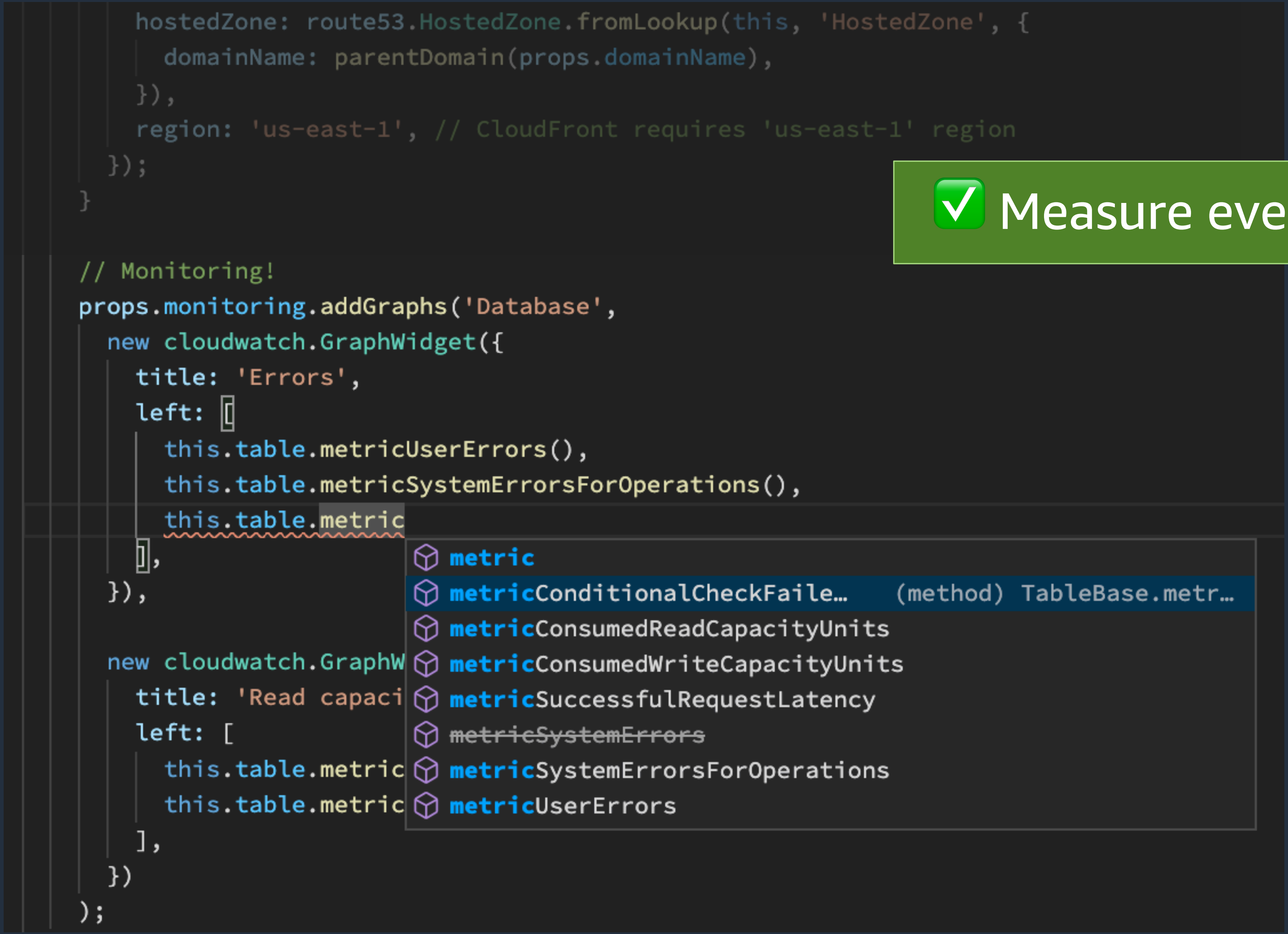
StatefulStack



```
/**
 * Stack with stateful resources
 */
export class StatefulStack extends cdk.Stack {
  public readonly table: dynamodb.Table;
  public readonly certificate?: certmgr.ICertificate;

  constructor(scope: cdk.Construct, id: string, props: StatefulStackProps) {
    super(scope, id, props);

    this.table = new dynamodb.Table(this, 'Table', {
      partitionKey: { name: 'id', type: dynamodb.Attr
    });

    if (props.domainName) {
      this.certificate = new certmgr.DnsValidatedCertificate(this, 'Certificate', {
        domainName: props.domainName,
        hostedZone: route53.HostedZone.fromLookup(this, 'HostedZone', {
          domainName: parentDomain(props.domainName),
        }),
        region: 'us-east-1', // CloudFront requires 'us-east-1' region
      });
    }

    // Monitoring!
    props.monitoring.addGraphs('Database',
```

✅ Make your app deterministic (context)

StatefulStack

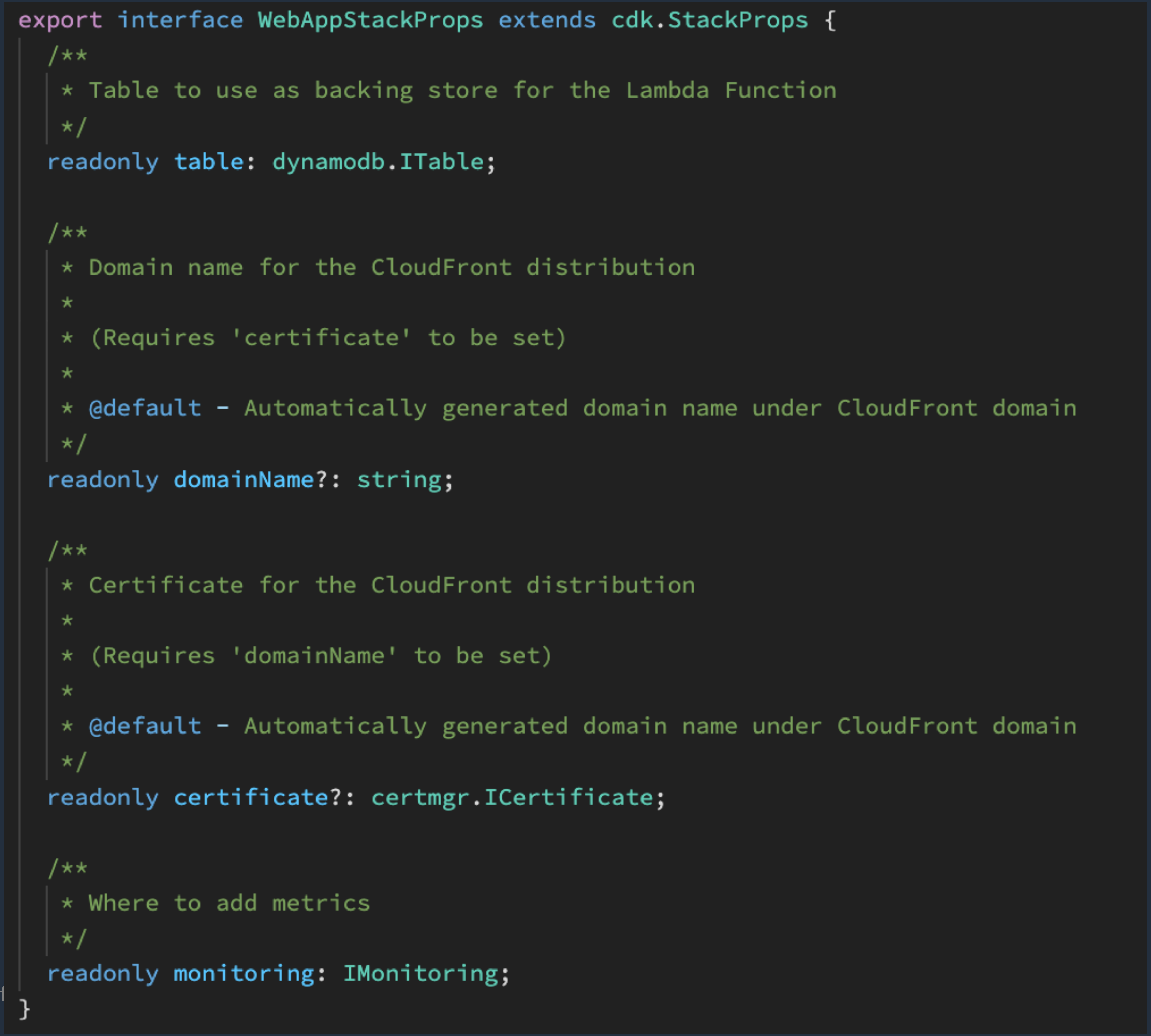


```
      hostedZone: route53.HostedZone.fromLookup(this, 'HostedZone', {
        domainName: parentDomain(props.domainName),
      }),
      region: 'us-east-1', // CloudFront requires 'us-east-1' region
    });
  }

  // Monitoring!
  props.monitoring.addGraphs('Database',
    new cloudwatch.GraphWidget({
      title: 'Errors',
      left: [
        this.table.metricUserErrors(),
        this.table.metricSystemErrorsForOperations(),
        this.table.metric
      ],
    }),

    new cloudwatch.GraphW
      title: 'Read capaci
      left: [
        this.table.metric
        this.table.metric
      ],
    })
  );
```

```
metric
metricConditionalCheckFaile…    (method) TableBase.metr…
metricConsumedReadCapacityUnits
metricConsumedWriteCapacityUnits
metricSuccessfulRequestLatency
~~metricSystemErrors~~
metricSystemErrorsForOperations
metricUserErrors
```

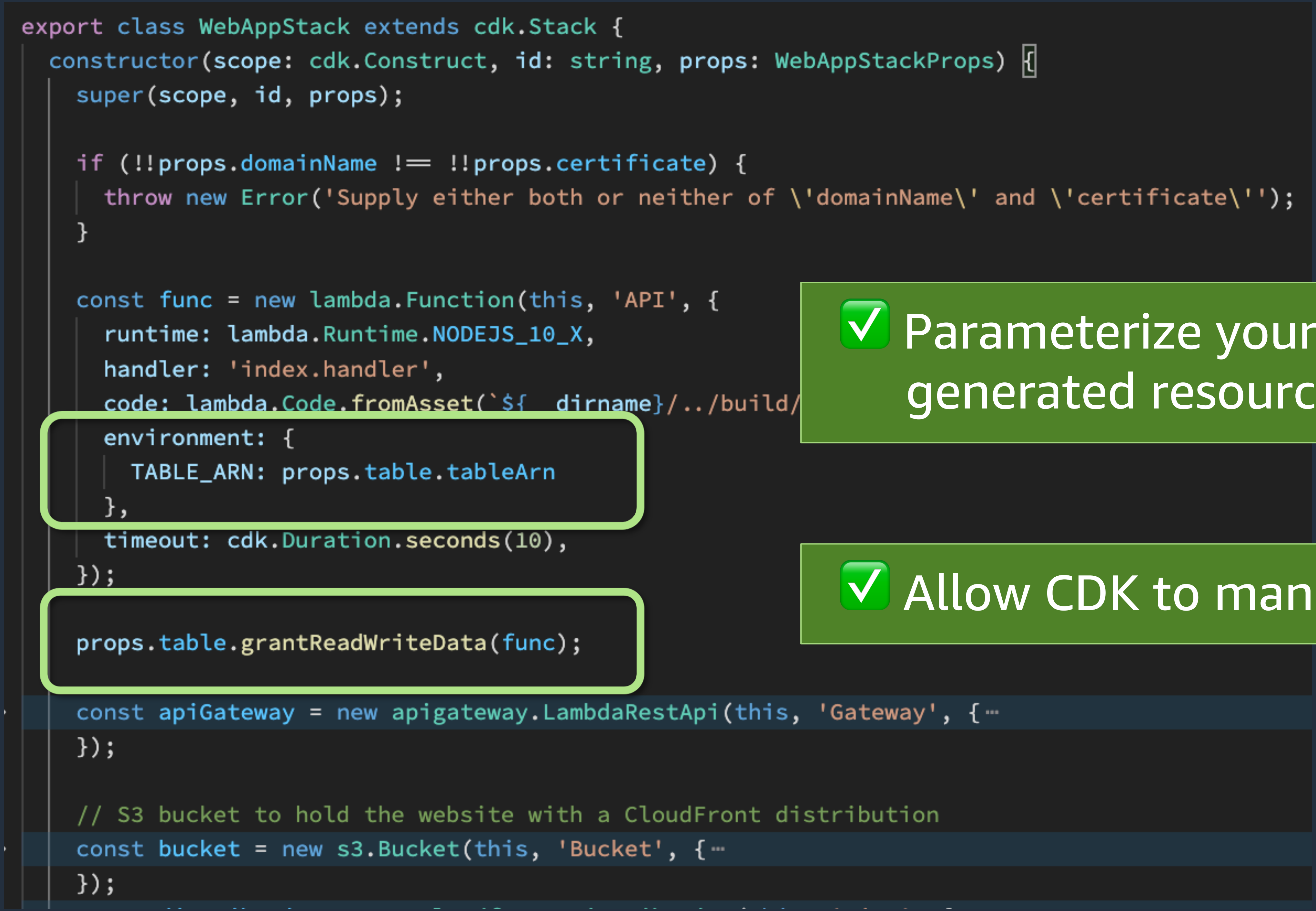
✅ Measure everything

WebAppStack



```
export interface WebAppStackProps extends cdk.StackProps {
  /**
   * Table to use as backing store for the Lambda Function
   */
  readonly table: dynamodb.ITable;

  /**
   * Domain name for the CloudFront distribution
   *
   * (Requires 'certificate' to be set)
   *
   * @default - Automatically generated domain name under CloudFront domain
   */
  readonly domainName?: string;

  /**
   * Certificate for the CloudFront distribution
   *
   * (Requires 'domainName' to be set)
   *
   * @default - Automatically generated domain name under CloudFront domain
   */
  readonly certificate?: certmgr.ICertificate;

  /**
   * Where to add metrics
   */
  readonly monitoring: IMonitoring;
}
```

WebAppStack

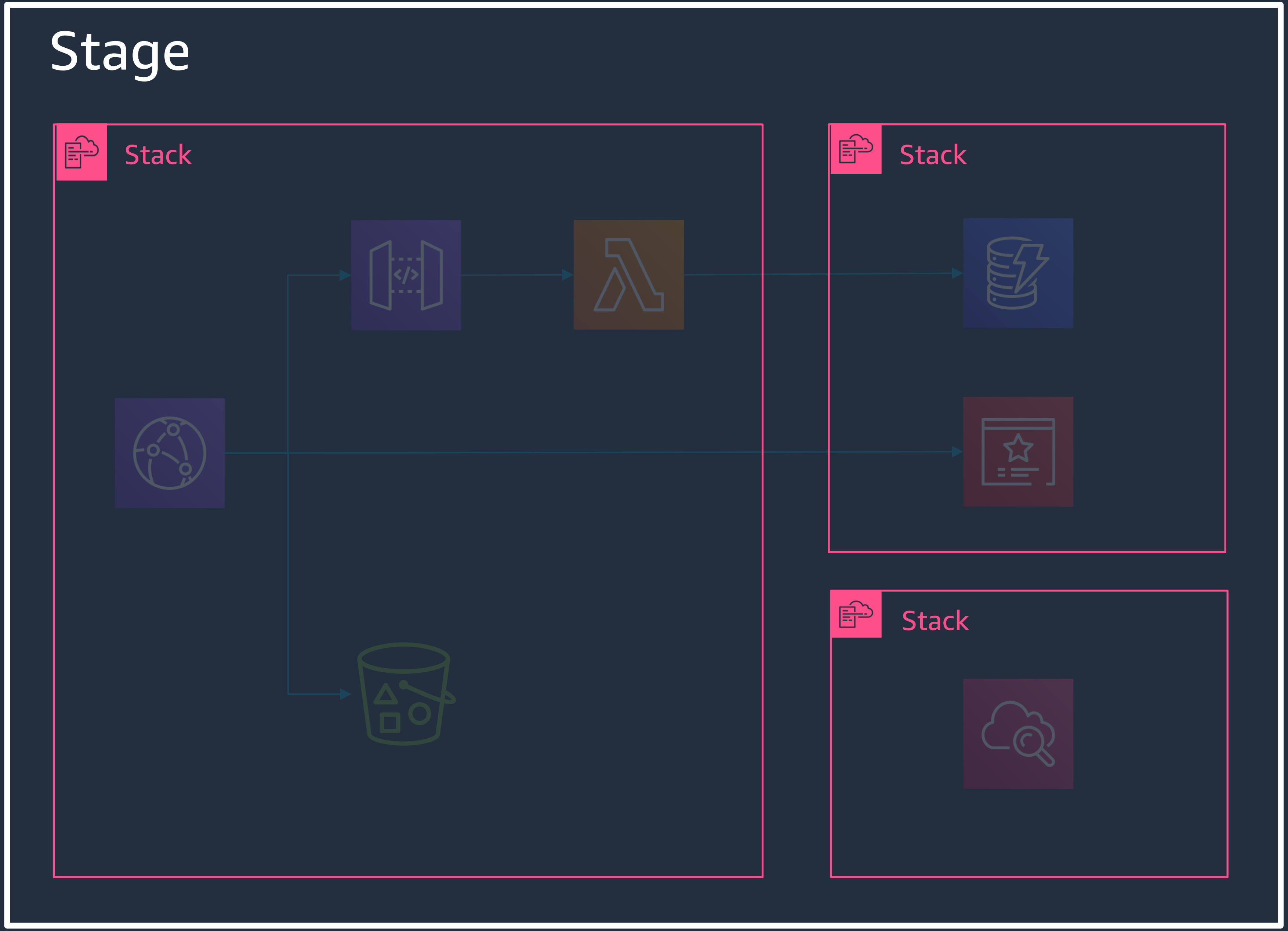


```
export class WebAppStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props: WebAppStackProps) {
    super(scope, id, props);

    if (!!props.domainName !== !!props.certificate) {
      throw new Error('Supply either both or neither of \'domainName\' and \'certificate\'');
    }

    const func = new lambda.Function(this, 'API', {
      runtime: lambda.Runtime.NODEJS_10_X,
      handler: 'index.handler',
      code: lambda.Code.fromAsset(`${__dirname}/../build/
      environment: {
        TABLE_ARN: props.table.tableArn
      },
      timeout: cdk.Duration.seconds(10),
    });

    props.table.grantReadWriteData(func);

    const apiGateway = new apigateway.LambdaRestApi(this, 'Gateway', {…
    });

    // S3 bucket to hold the website with a CloudFront distribution
    const bucket = new s3.Bucket(this, 'Bucket', {…
    });
```

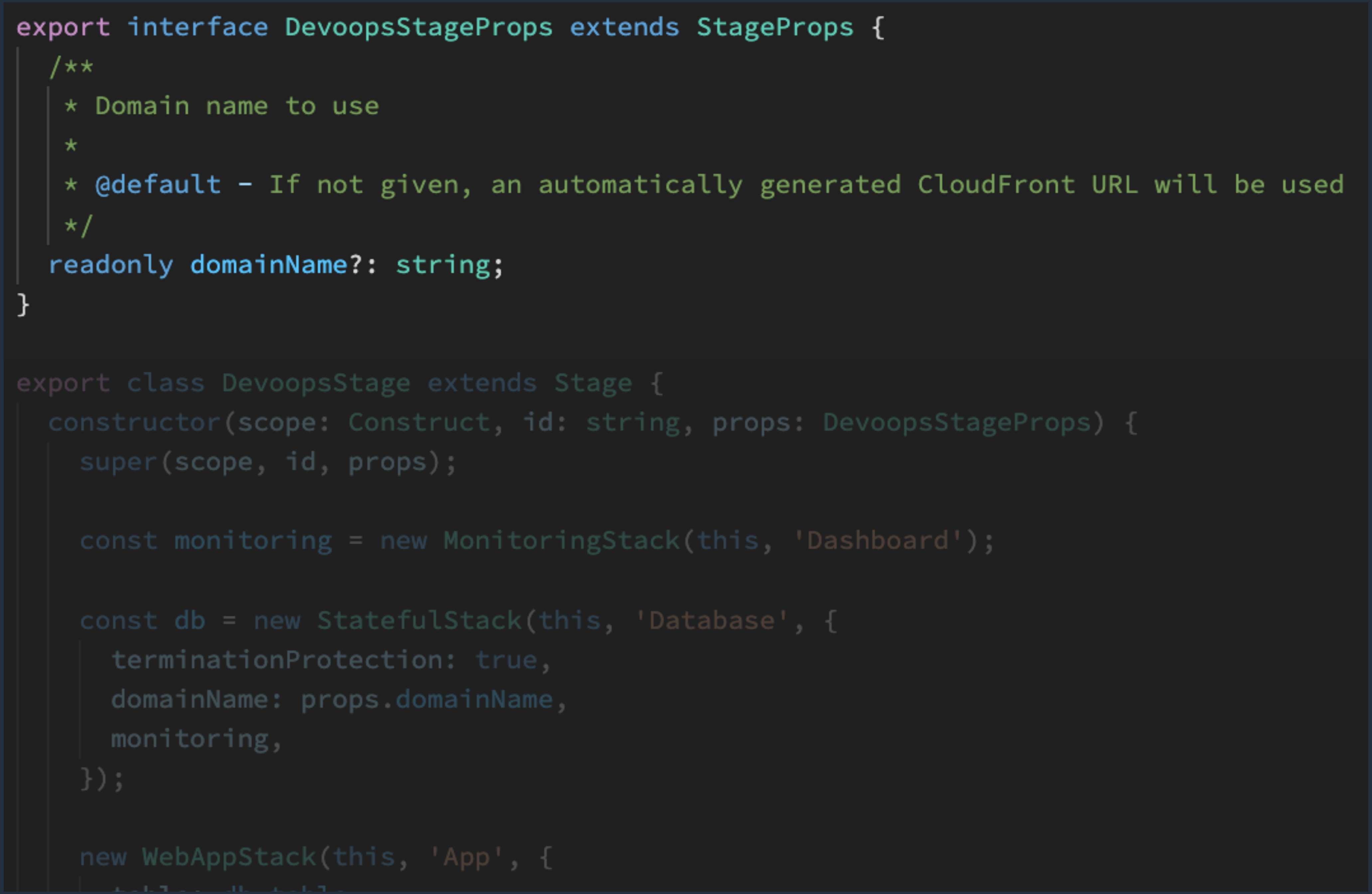
✅ Parameterize your app for generated resource names

✅ Allow CDK to manage roles

Stage

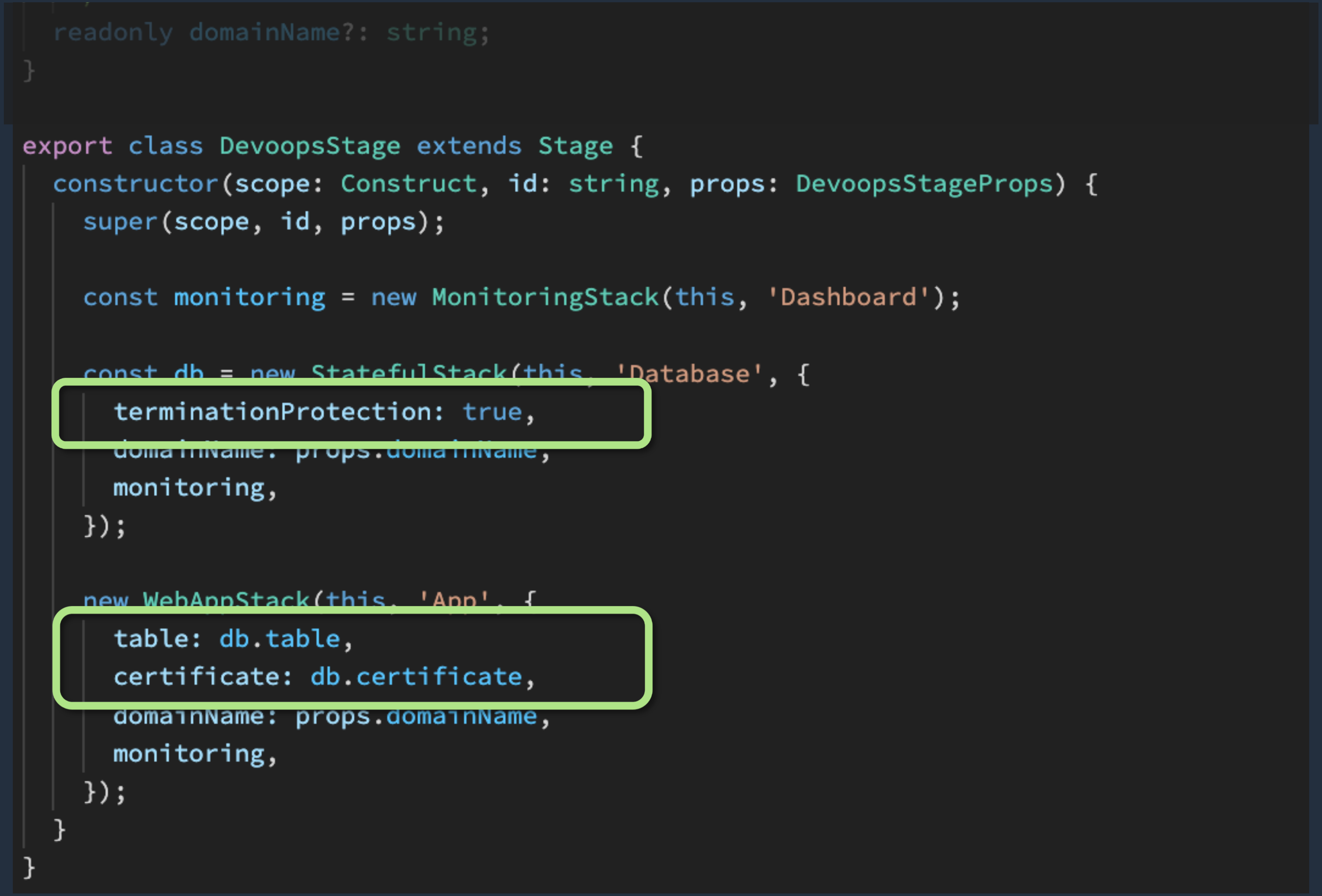


Stage



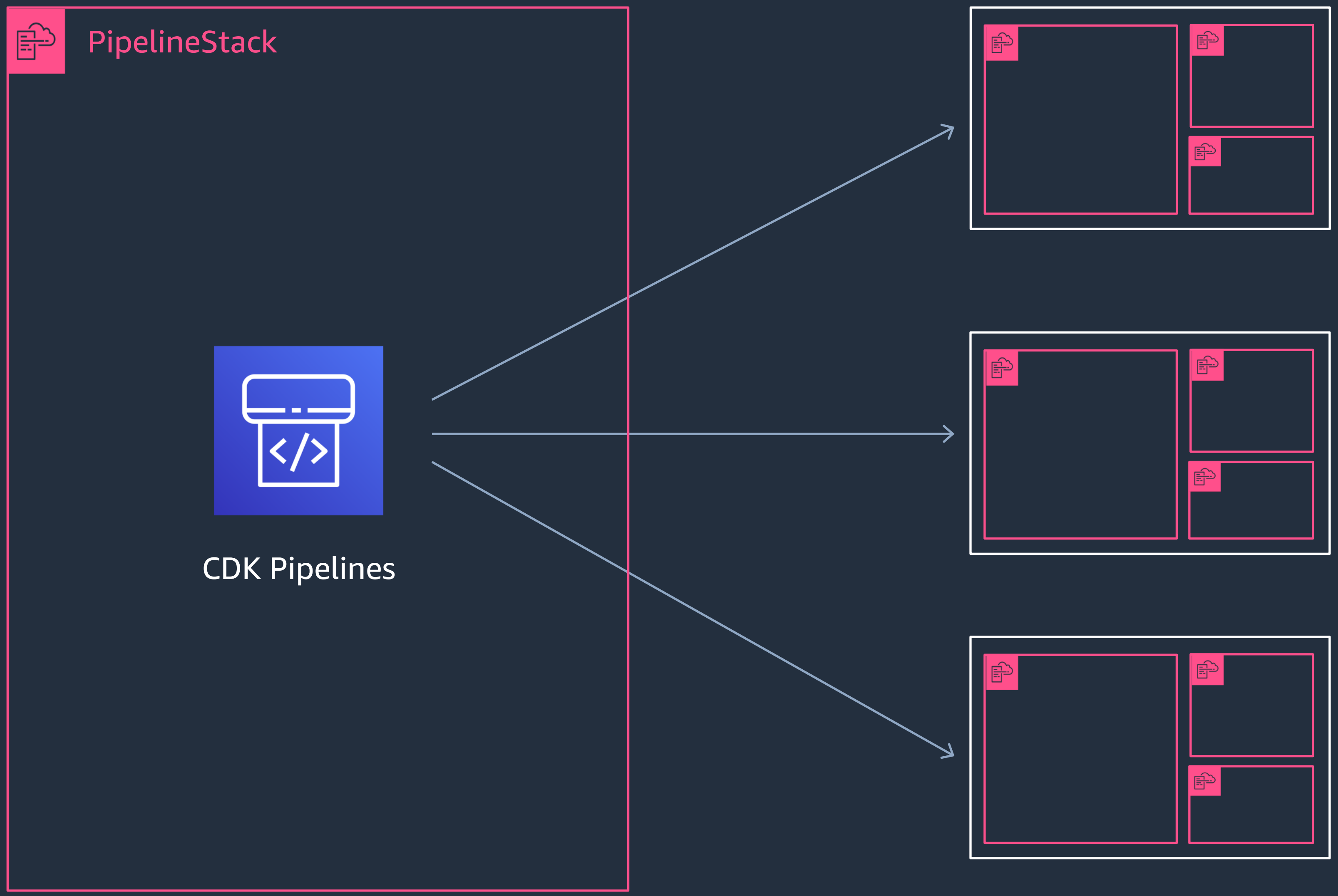
```
export interface DevoopsStageProps extends StageProps {
  /**
   * Domain name to use
   *
   * @default - If not given, an automatically generated CloudFront URL will be used
   */
  readonly domainName?: string;
}

export class DevoopsStage extends Stage {
  constructor(scope: Construct, id: string, props: DevoopsStageProps) {
    super(scope, id, props);

    const monitoring = new MonitoringStack(this, 'Dashboard');

    const db = new StatefulStack(this, 'Database', {
      terminationProtection: true,
      domainName: props.domainName,
      monitoring,
    });

    new WebAppStack(this, 'App', {
```

Stage

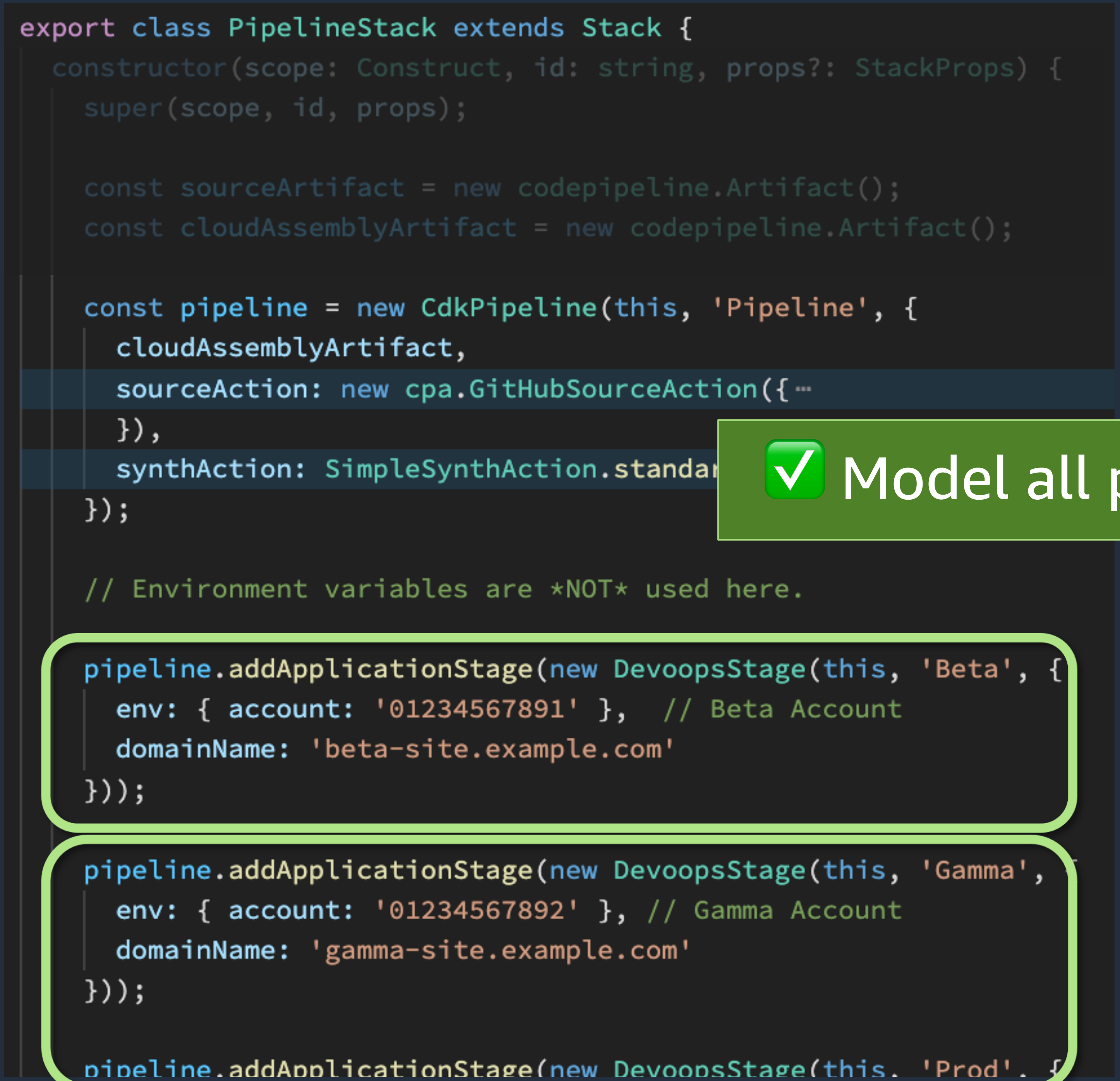


```
  readonly domainName?: string;
}

export class DevoopsStage extends Stage {
  constructor(scope: Construct, id: string, props: DevoopsStageProps) {
    super(scope, id, props);

    const monitoring = new MonitoringStack(this, 'Dashboard');

    const db = new StatefulStack(this, 'Database', {
      terminationProtection: true,
      domainName: props.domainName,
      monitoring,
    });

    new WebAppStack(this, 'App', {
      table: db.table,
      certificate: db.certificate,
      domainName: props.domainName,
      monitoring,
    });
  }
}
```

Pipeline



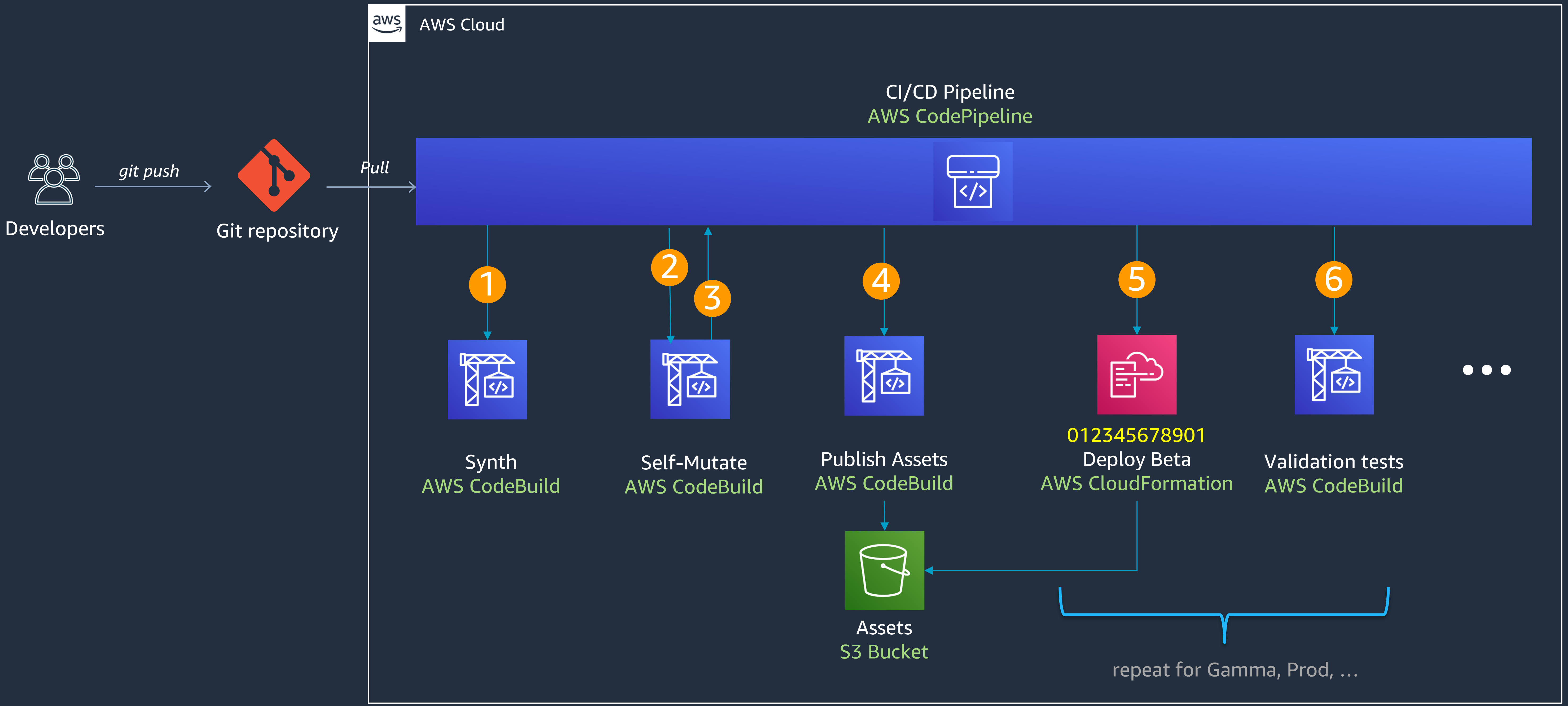
Pipeline



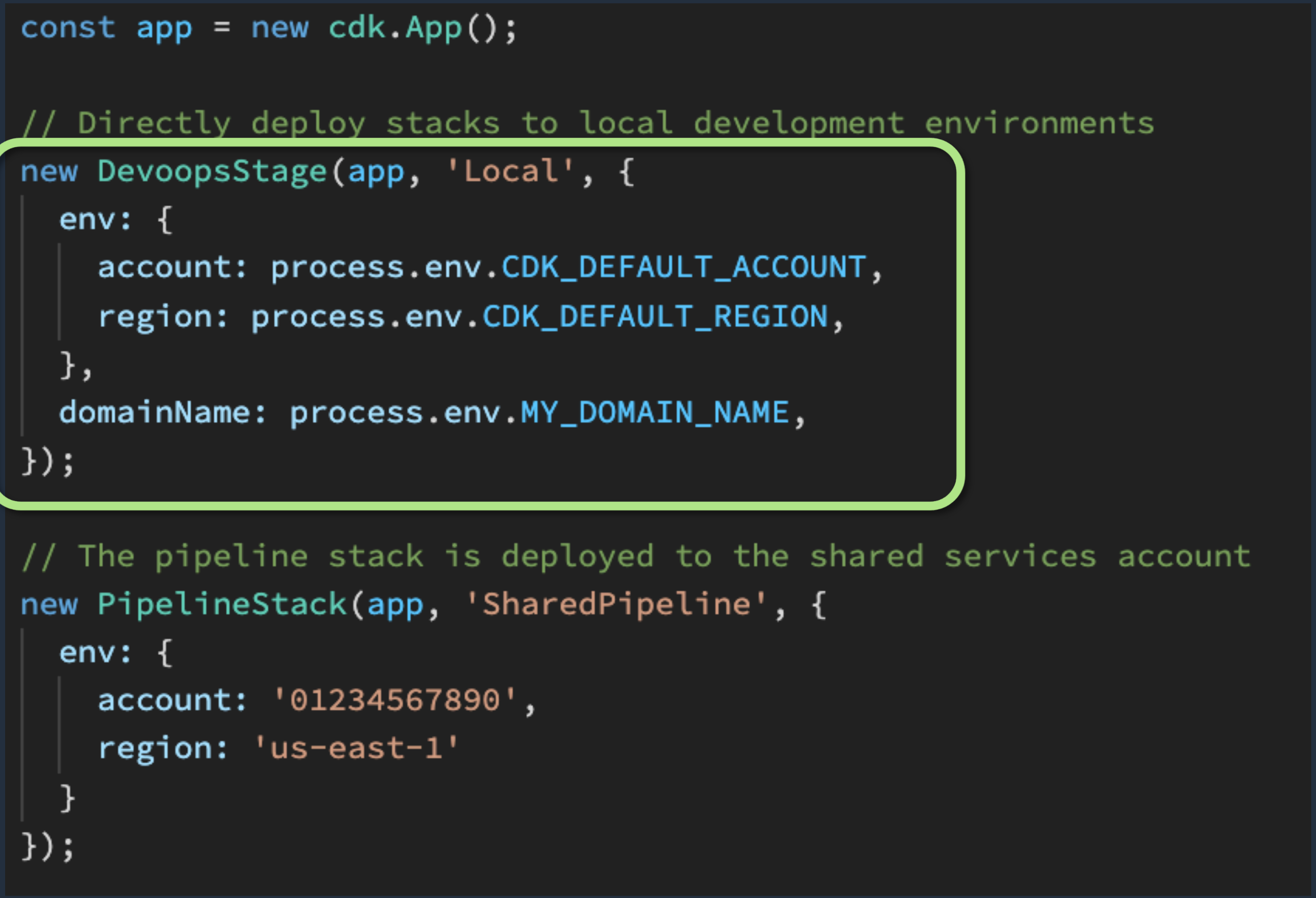
```
export class PipelineStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    const sourceArtifact = new codepipeline.Artifact();
    const cloudAssemblyArtifact = new codepipeline.Artifact();

    const pipeline = new CdkPipeline(this, 'Pipeline', {
      cloudAssemblyArtifact,
      sourceAction: new cpa.GitHubSourceAction({⋯
      }),
      synthAction: SimpleSynthAction.standar
    });

    // Environment variables are *NOT* used here.

    pipeline.addApplicationStage(new DevoopsStage(this, 'Beta', {
      env: { account: '01234567891' },  // Beta Account
      domainName: 'beta-site.example.com'
    }));

    pipeline.addApplicationStage(new DevoopsStage(this, 'Gamma',
      env: { account: '01234567892' }, // Gamma Account
      domainName: 'gamma-site.example.com'
    }));

    pipeline.addApplicationStage(new DevoopsStage(this, 'Prod', {
```

✅ Model all production Stages

Sample Application – CDK Pipelines

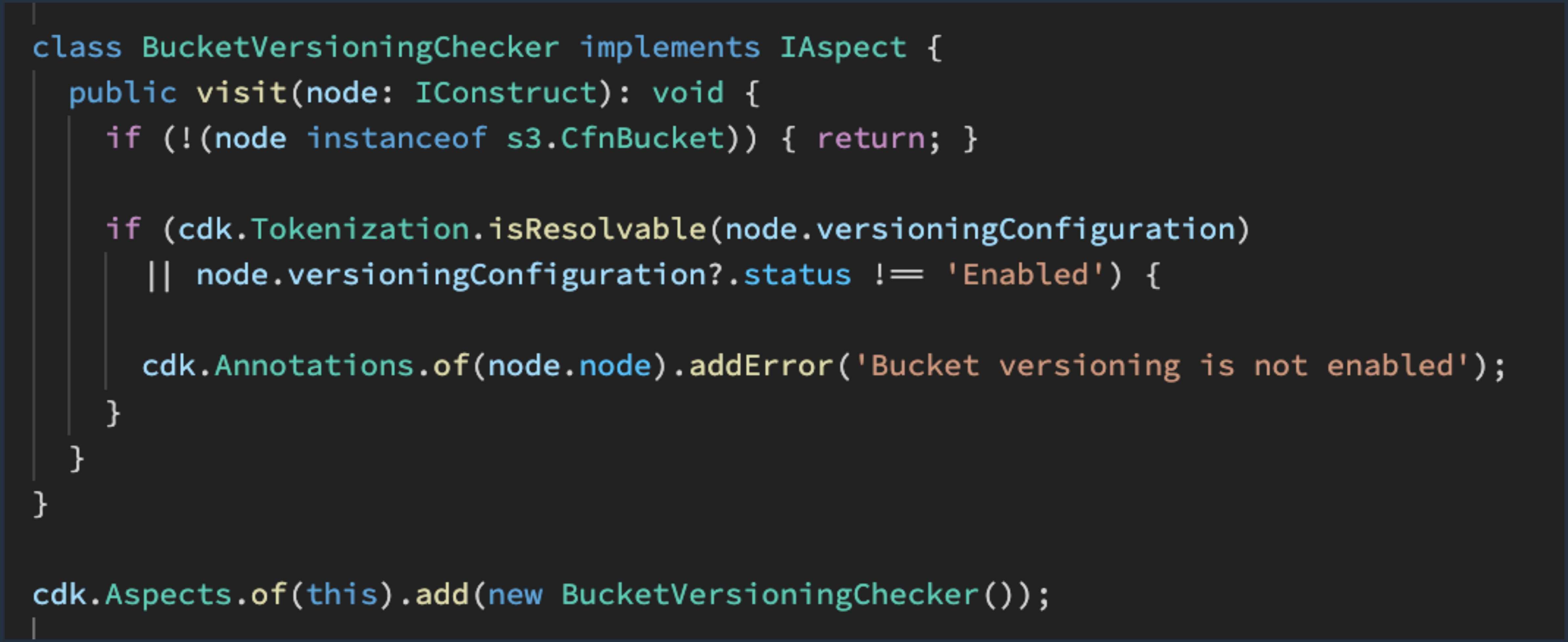


Developer Stacks



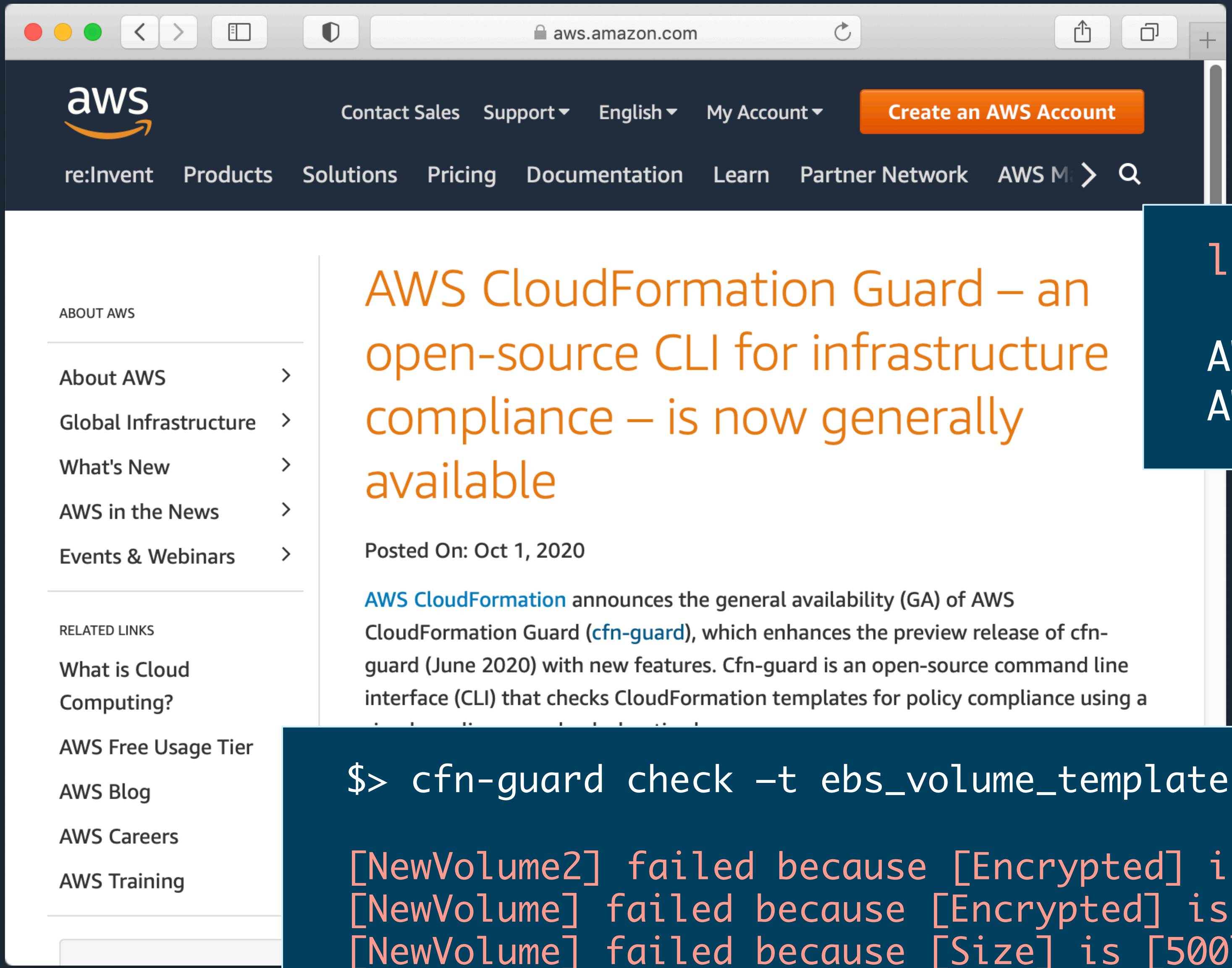
```
const app = new cdk.App();

// Directly deploy stacks to local development environments
new DevoopsStage(app, 'Local', {
  env: {
    account: process.env.CDK_DEFAULT_ACCOUNT,
    region: process.env.CDK_DEFAULT_REGION,
  },
  domainName: process.env.MY_DOMAIN_NAME,
});

// The pipeline stack is deployed to the shared services account
new PipelineStack(app, 'SharedPipeline', {
  env: {
    account: '01234567890',
    region: 'us-east-1'
  }
});
```

Compliance: Aspects



```
class BucketVersioningChecker implements IAspect {
  public visit(node: IConstruct): void {
    if (!(node instanceof s3.CfnBucket)) { return; }

    if (cdk.Tokenization.isResolvable(node.versioningConfiguration)
      || node.versioningConfiguration?.status !== 'Enabled') {

      cdk.Annotations.of(node.node).addError('Bucket versioning is not enabled');
    }
  }
}

cdk.Aspects.of(this).add(new BucketVersioningChecker());
```

Compliance: CloudFormation Guard



AWS CloudFormation Guard – an open-source CLI for infrastructure compliance – is now generally available

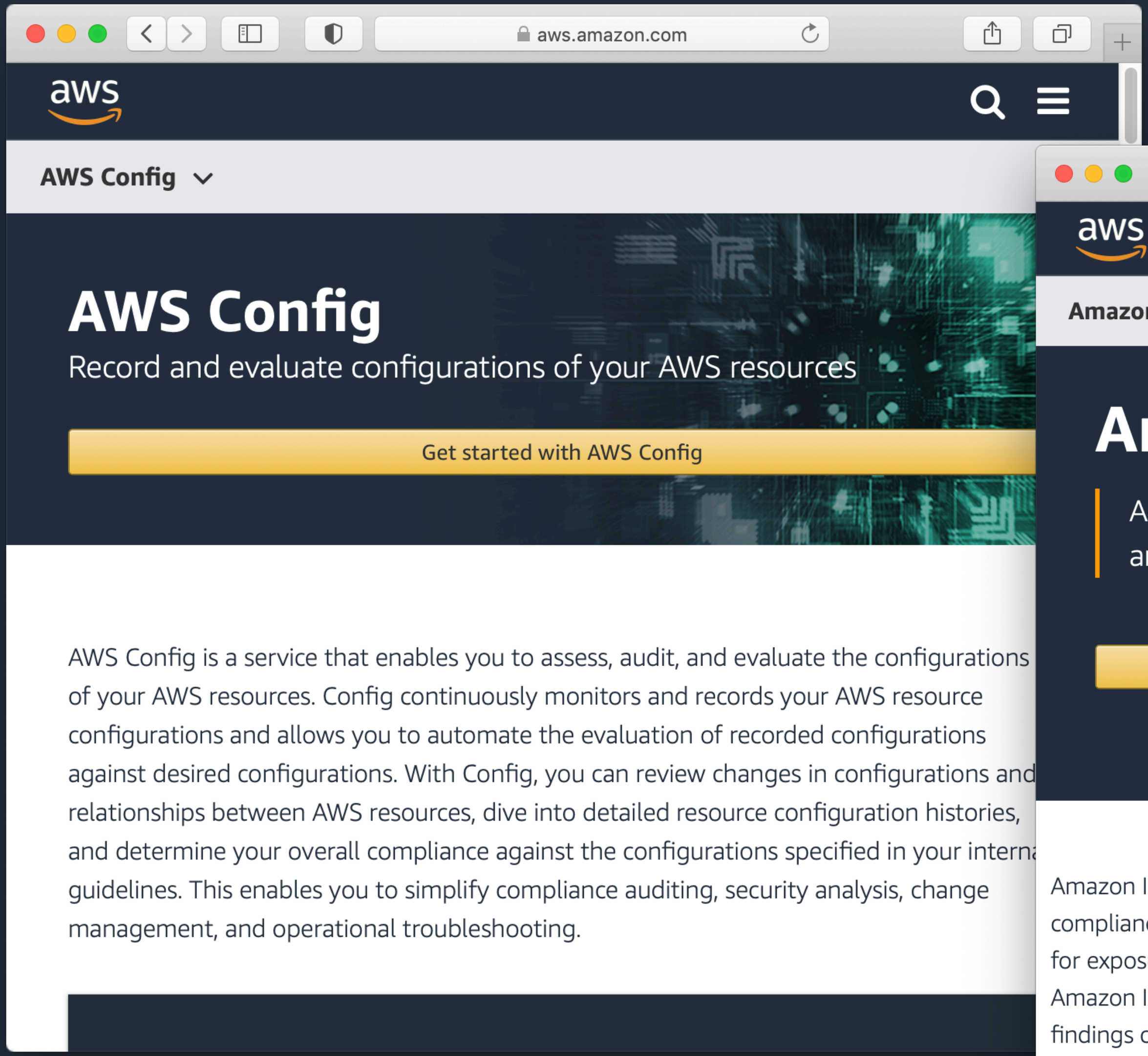
Posted On: Oct 1, 2020

AWS CloudFormation announces the general availability (GA) of AWS CloudFormation Guard (cfn-guard), which enhances the preview release of cfn-guard (June 2020) with new features. Cfn-guard is an open-source command line interface (CLI) that checks CloudFormation templates for policy compliance using a

```
let encryption_flag = true

AWS::EC2::Volume Encrypted == %encryption_flag
AWS::EC2::Volume Size <= 100
```

```
$> cfn-guard check –t ebs_volume_template.json -r ebs_volume_template.ruleset

[NewVolume2] failed because [Encrypted] is [false] and the permitted value is [true]
[NewVolume] failed because [Encrypted] is [false] and the permitted value is [true]
[NewVolume] failed because [Size] is [500] and the permitted value is [<= 100] Number of
failures: 3
```

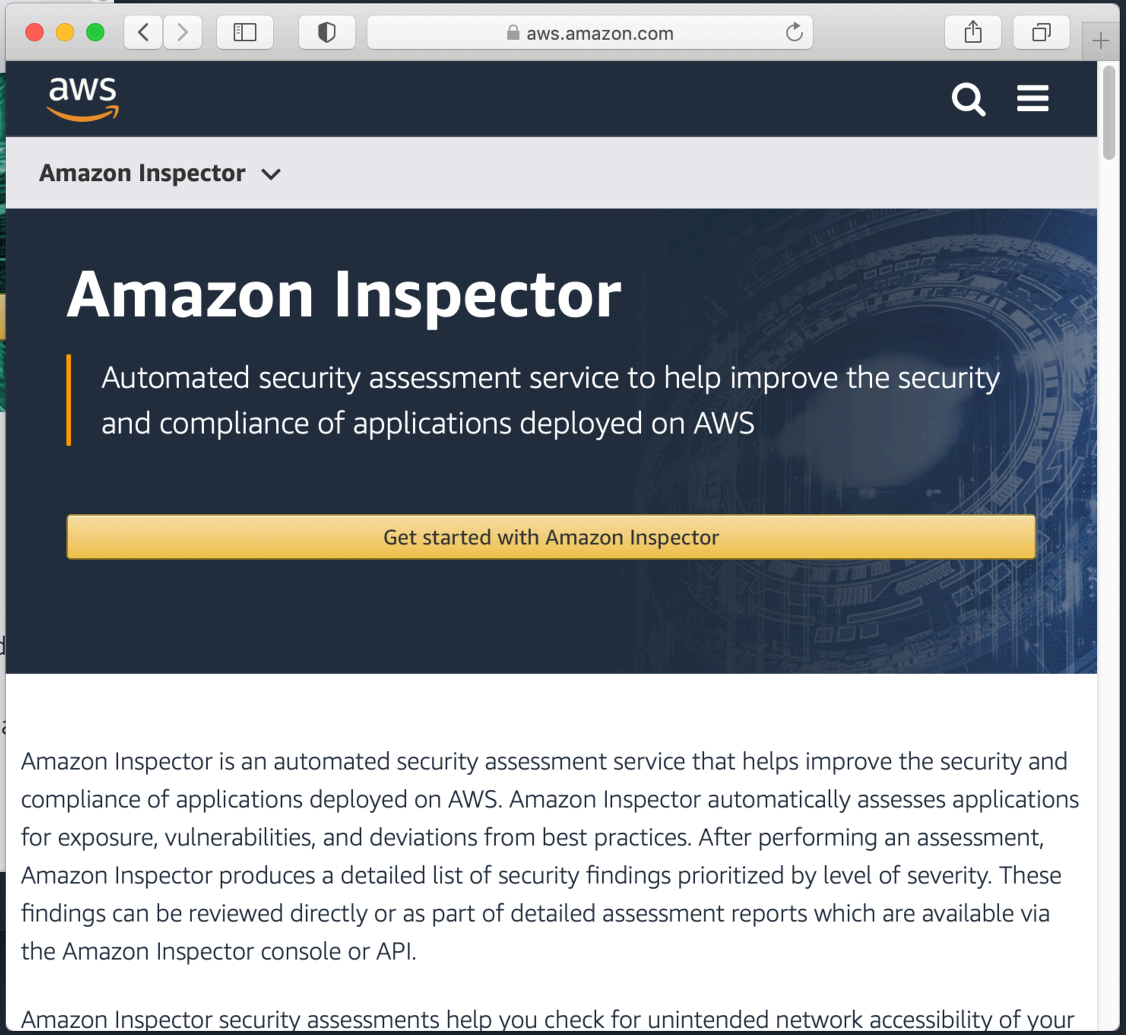
Compliance: Config and Inspector



AWS Config

Record and evaluate configurations of your AWS resources

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your interna guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.



Amazon Inspector

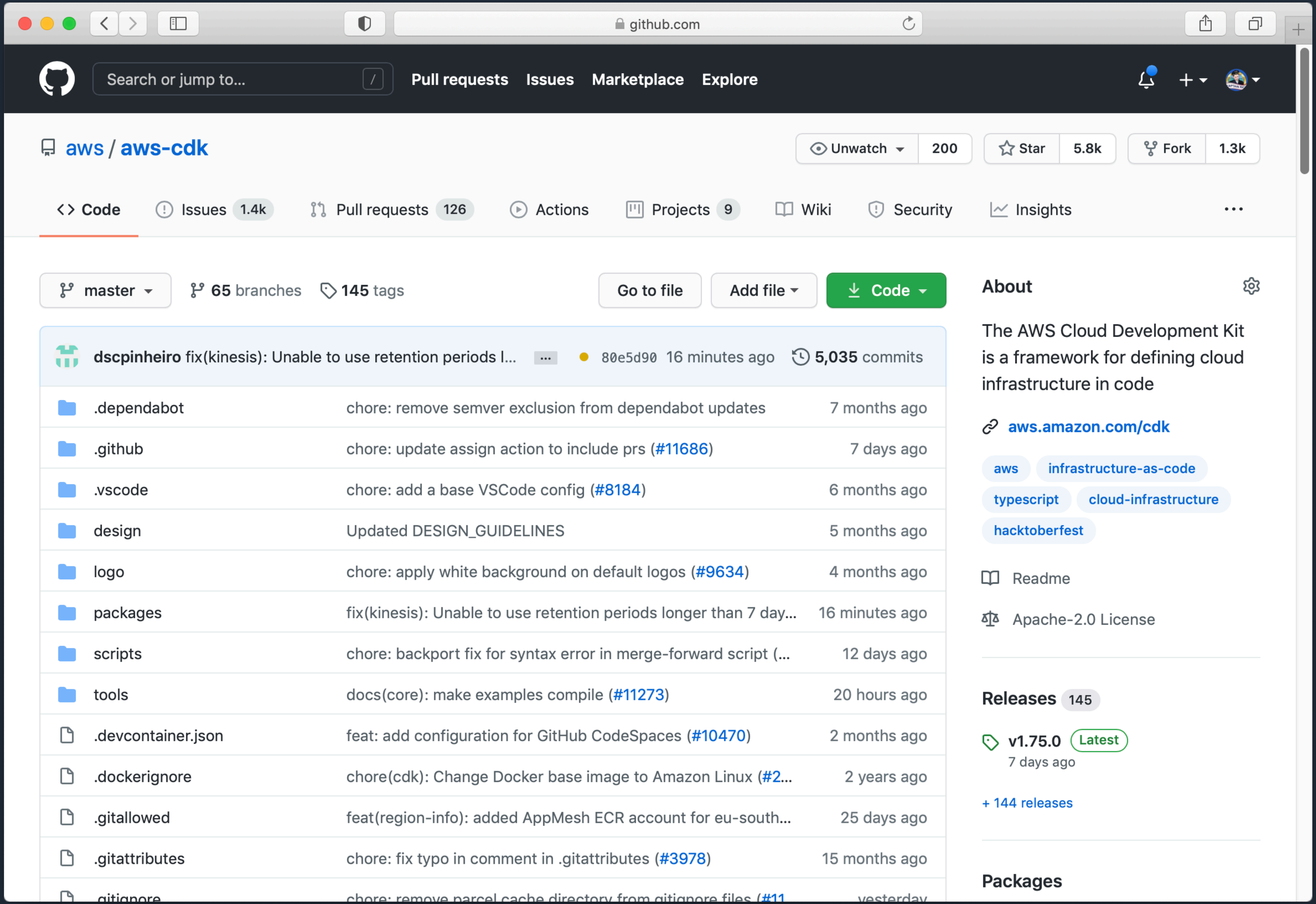
Automated security assessment service to help improve the security and compliance of applications deployed on AWS

Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings prioritized by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available via the Amazon Inspector console or API.

Amazon Inspector security assessments help you check for unintended network accessibility of your

And finally…

Best Practice: Contribute 😇


github.com

Search or jump to... / Pull requests Issues Marketplace Explore

aws / aws-cdk

Unwatch 200 | Star 5.8k | Fork 1.3k

Code | Issues 1.4k | Pull requests 126 | Actions | Projects 9 | Wiki | Security | Insights

master | 65 branches | 145 tags | Go to file | Add file | Code

dscpinheiro fix(kinesis): Unable to use retention periods l... … 80e5d90 16 minutes ago 5,035 commits

.dependabot	chore: remove semver exclusion from dependabot updates	7 months ago
.github	chore: update assign action to include prs (#11686)	7 days ago
.vscode	chore: add a base VSCode config (#8184)	6 months ago
design	Updated DESIGN_GUIDELINES	5 months ago
logo	chore: apply white background on default logos (#9634)	4 months ago
packages	fix(kinesis): Unable to use retention periods longer than 7 day...	16 minutes ago
scripts	chore: backport fix for syntax error in merge-forward script (...	12 days ago
tools	docs(core): make examples compile (#11273)	20 hours ago
.devcontainer.json	feat: add configuration for GitHub CodeSpaces (#10470)	2 months ago
.dockerignore	chore(cdk): Change Docker base image to Amazon Linux (#2...	2 years ago
.gitallowed	feat(region-info): added AppMesh ECR account for eu-south...	25 days ago
.gitattributes	chore: fix typo in comment in .gitattributes (#3978)	15 months ago
.gitignore	chore: remove parcel cache directory from gitignore files (#11	yesterday

About

The AWS Cloud Development Kit is a framework for defining cloud infrastructure in code

aws.amazon.com/cdk

aws | infrastructure-as-code | typescript | cloud-infrastructure | hacktoberfest

Readme

Apache-2.0 License

Releases 145

v1.75.0 Latest
7 days ago

+ 144 releases

Packages

Resources

Get Started with the AWS CDK

AWS CDK Online Workshop	https://cdkworkshop.com
AWS CDK Samples on GitHub	https://github.com/aws-samples/aws-cdk-examples

Dive Deeper

GitHub

https://github.com/aws/aws-cdk

CDK Pipelines

https://aws.amazon.com/blogs/developer/cdk-pipelines-continuous-delivery-for-aws-cdk-applications/

Sample Code in this presentation

https://github.com/rix0rrr/devoops-sample

A Landing Zone Implemented with CDK

https://github.com/aws-samples/aws-secure-environment-accelerator

Testing infrastructure with the CDK

https://aws.amazon.com/blogs/developer/testing-infrastructure-with-the-aws-cloud-development-kit-cdk/

Q&A

Eric Z. Beard

Rico Huijbers