
AWS
re:Invent

DOP202

AWS CDK: What's new and what's next

Jason Fulghum

Senior Manager

AWS Developer Tools

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

CDK overview

What’s new

What’s next

Wrap up



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

```
domainName
domainZone
```

```
(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



AWS CDK workshop
https://cdkworkshop.com



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

```
domainName
domainZone
```

```
(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



AWS CDK workshop
https://cdkworkshop.com



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

```
domainName
domainZone
```

```
(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



AWS CDK workshop
https://cdkworkshop.com



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

domainName
domainZone

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



AWS CDK workshop
https://cdkworkshop.com



AWS CDK: What's new

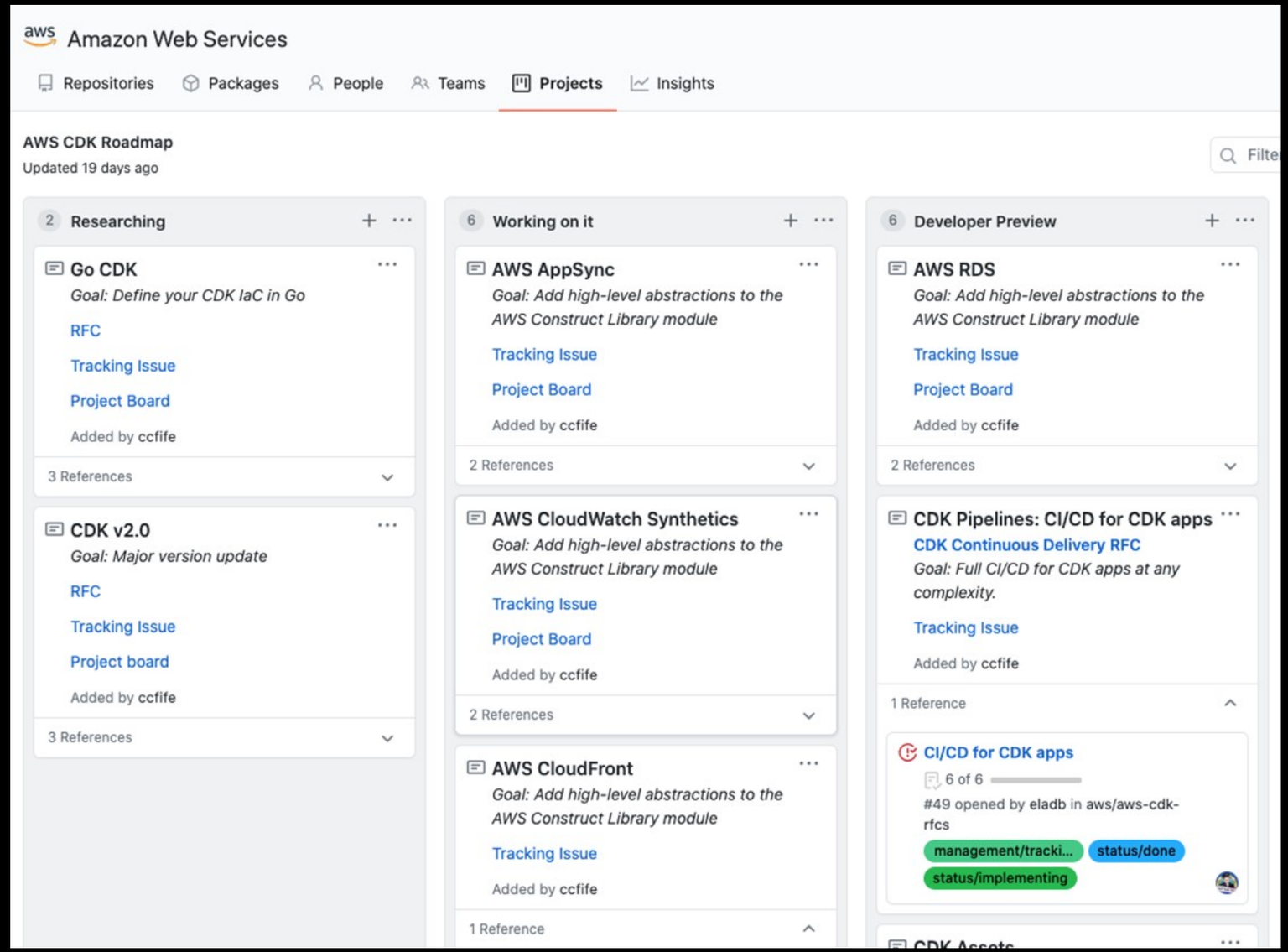


AWS re:Invent

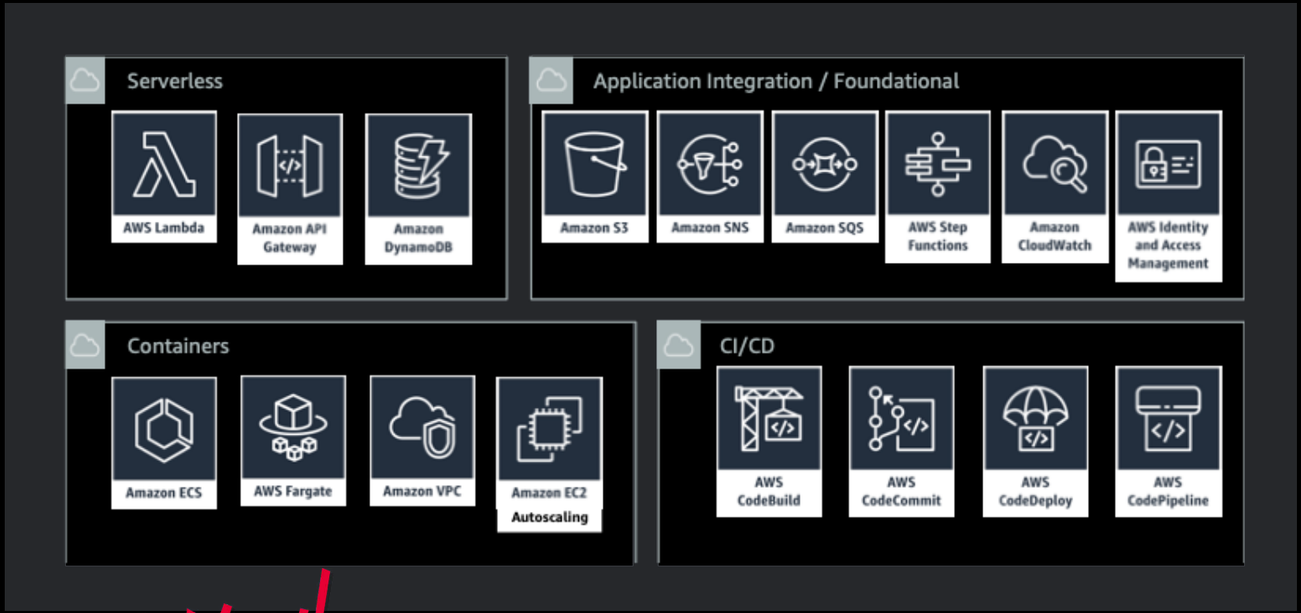
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CDK roadmap

- High-level view of active work streams
- Help us prioritize by upvoting issues!



AWS Construct Library



Thank You!

Hundreds of contributors

Thousands of merged contributions

Deeper dive
AWS re:Invent 2019:
Contributing to the AWS Construct Library

New GA modules

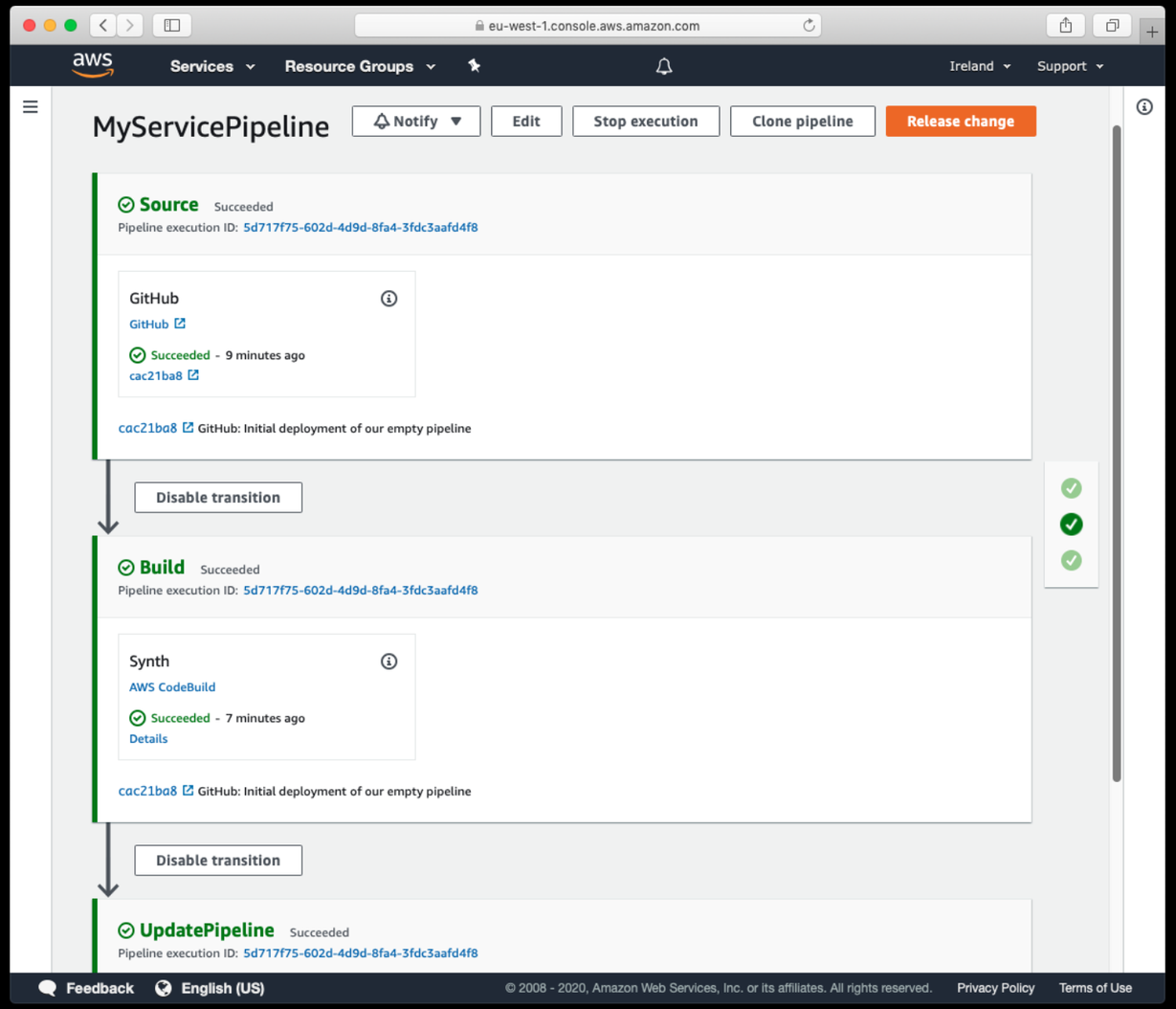
- Amazon Cognito User Pools
- Amazon Kinesis
- Amazon Relational Database Service (RDS)
- Amazon Route 53 patterns
- AWS CloudTrail
- AWS Config
- AWS Step Functions
- Custom resources

New dev preview modules

- AWS App Mesh
- AWS AppSync
- AWS Cloud Development Kit (CDK) Pipelines
- Amazon CloudFront
- Amazon CloudWatch Synthetics
- Amazon Elastic Kubernetes Service (EKS)

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

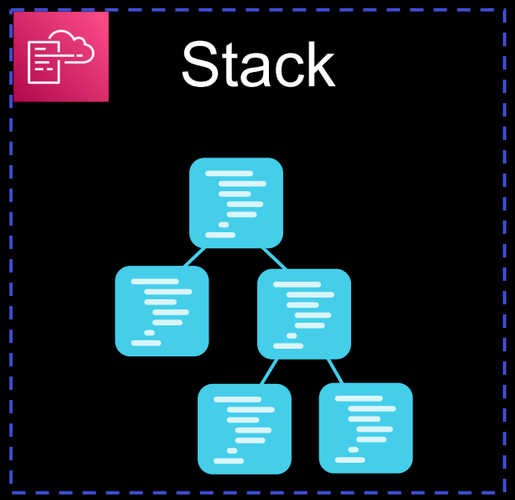
Deeper dive
AWS Online Tech Talk:
Enhanced CI/CD with AWS CDK

AWS ONLINE TECH TALKS

AWS CDK Pipelines

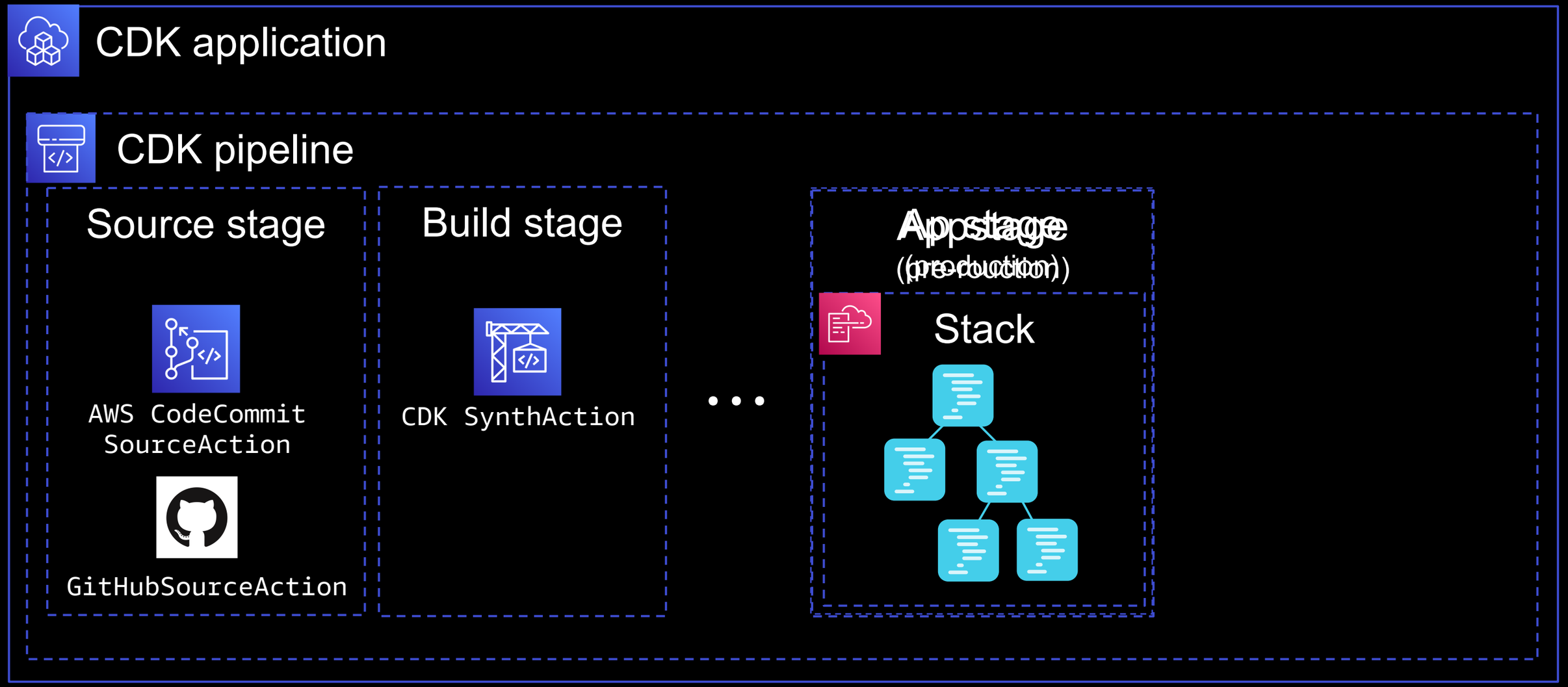
CONTINUOUS DELIVERY FOR AWS CDK APPLICATIONS

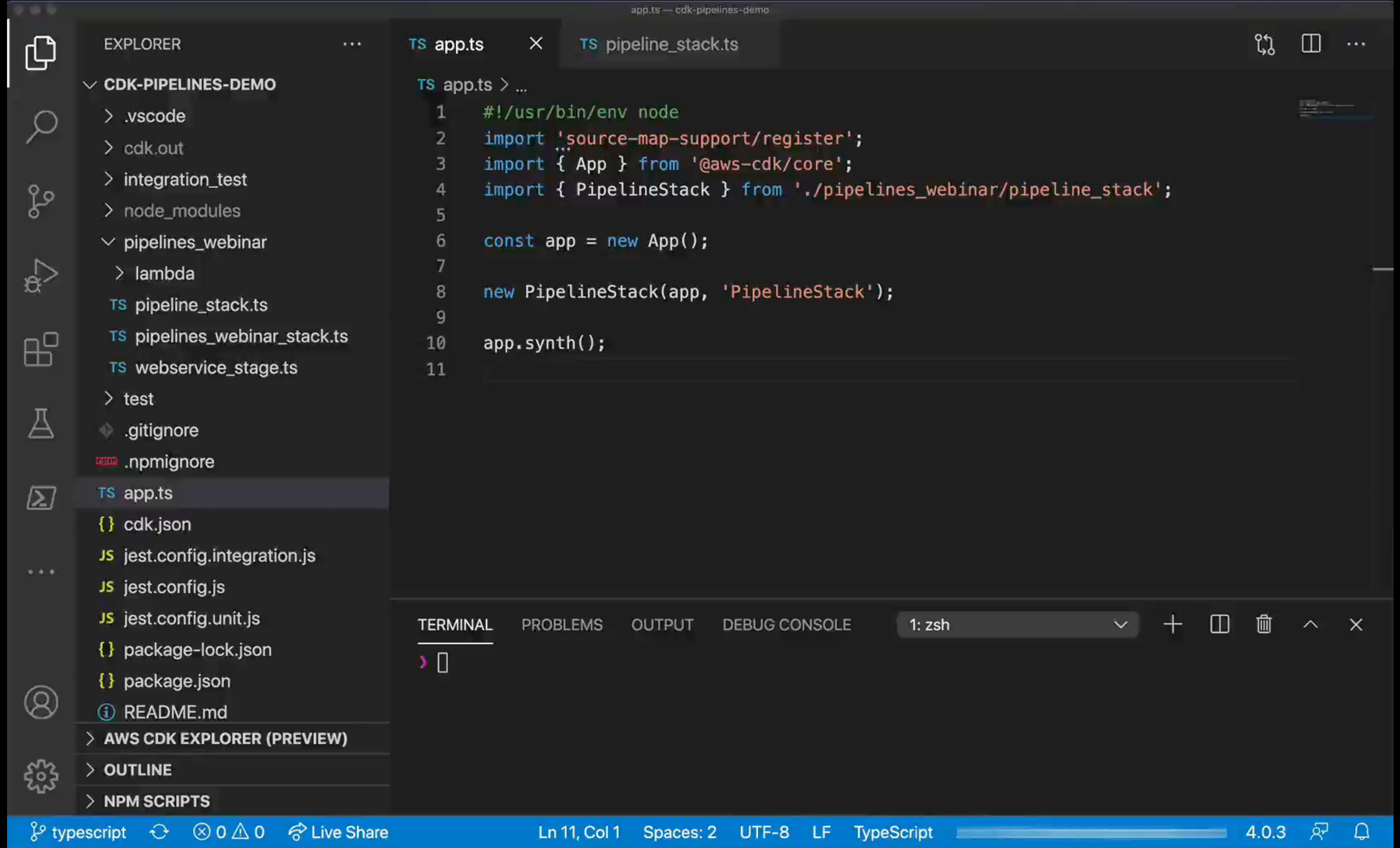




AWS CDK Pipelines

CONTINUOUS DELIVERY FOR AWS CDK APPLICATIONS




app.ts — cdk-pipelines-demo
EXPLORER
CDK-PIPELINES-DEMO
.vscode
cdk.out
integration_test
node_modules
pipelines_webinar
lambda
pipeline_stack.ts
pipelines_webinar_stack.ts
webservice_stage.ts
test
.gitignore
.npmignore
app.ts
cdk.json
jest.config.integration.js
jest.config.js
jest.config.unit.js
package-lock.json
package.json
README.md
AWS CDK EXPLORER (PREVIEW)
OUTLINE
NPM SCRIPTS
app.ts
pipeline_stack.ts
app.ts > ...
#!/usr/bin/env node
import 'source-map-support/register';
import { App } from '@aws-cdk/core';
import { PipelineStack } from './pipelines_webinar/pipeline_stack';
const app = new App();
new PipelineStack(app, 'PipelineStack');
app.synth();
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
1: zsh
typescript
0 0
Live Share
Ln 11, Col 1
Spaces: 2
UTF-8
LF
TypeScript
4.0.3

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 AWS re:Invent

Turbocharge the AWS CDK with AWS Solutions Constructs

CDK Patterns

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

cdkpatterns.com

@CDKPATTERNS

aws 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

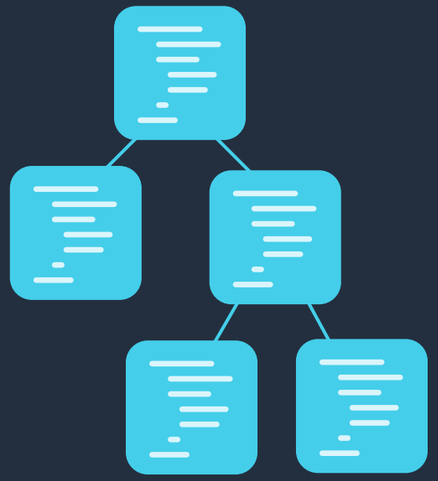
Constructs provided by the **AWS CDK team** to encapsulate a single service’s configuration interface

aws.amazon.com/cdk

Construct programing model

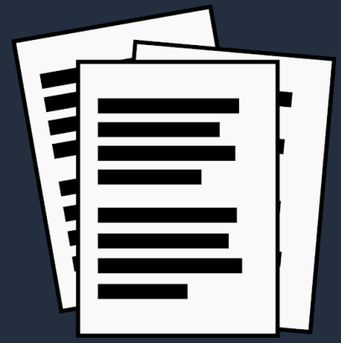
DEFINE COMPOSABLE CONFIGURATION MODELS THROUGH CODE

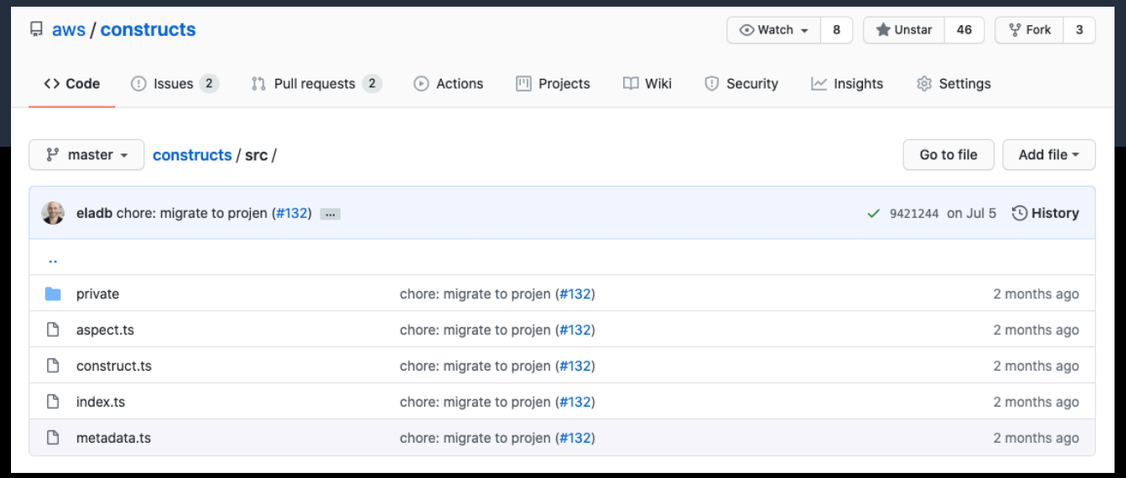
Construct tree



Synthesize

Configuration artifacts





CDK for Terraform (cdktf)

```typescript
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();
```

HashiCorp Terraform



Synthesize

Terraform JSON configuration

Terraform

Deeper dive

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

CDK for Kubernetes (cdk8s)

```python
#!/usr/bin/env python3
from constructs import Construct
from cdk8s import App, Chart

from imports import k8s

class MyChart(Chart):
    def __init__(self, scope: Construct, ns: str):
        super().__init__(scope, ns)

        label = {"app": "hello-k8s"}

        k8s.Service(self, 'service',
            spec=k8s.ServiceSpec(
                type='LoadBalancer',
                ports=[k8s.ServicePort(port=80, target_port=k8s.IntOrString.from_number(8080))],
                selector=label))

        k8s.Deployment(self, 'deployment',
            spec=k8s.DeploymentSpec(
                replicas=2,
                selector=k8s.LabelSelector(match_labels=label),
                template=k8s.PodTemplateSpec(
                metadata=k8s.ObjectMeta(labels=label),
                spec=k8s.PodSpec(containers=[
                    k8s.Container(
                    name='hello-kubernetes',
                    image='paulbouwer/hello-kubernetes:1.7',
                    ports=[k8s.ContainerPort(container_port=8080)])])))))

app = App()
MyChart(app, "hello-python")

app.synth()
```

cdk8s



Synthesize

Kubernetes YAML configuration

Deeper dive
Simplifying Kubernetes application management with cdk8s



AWS CDK: What's next

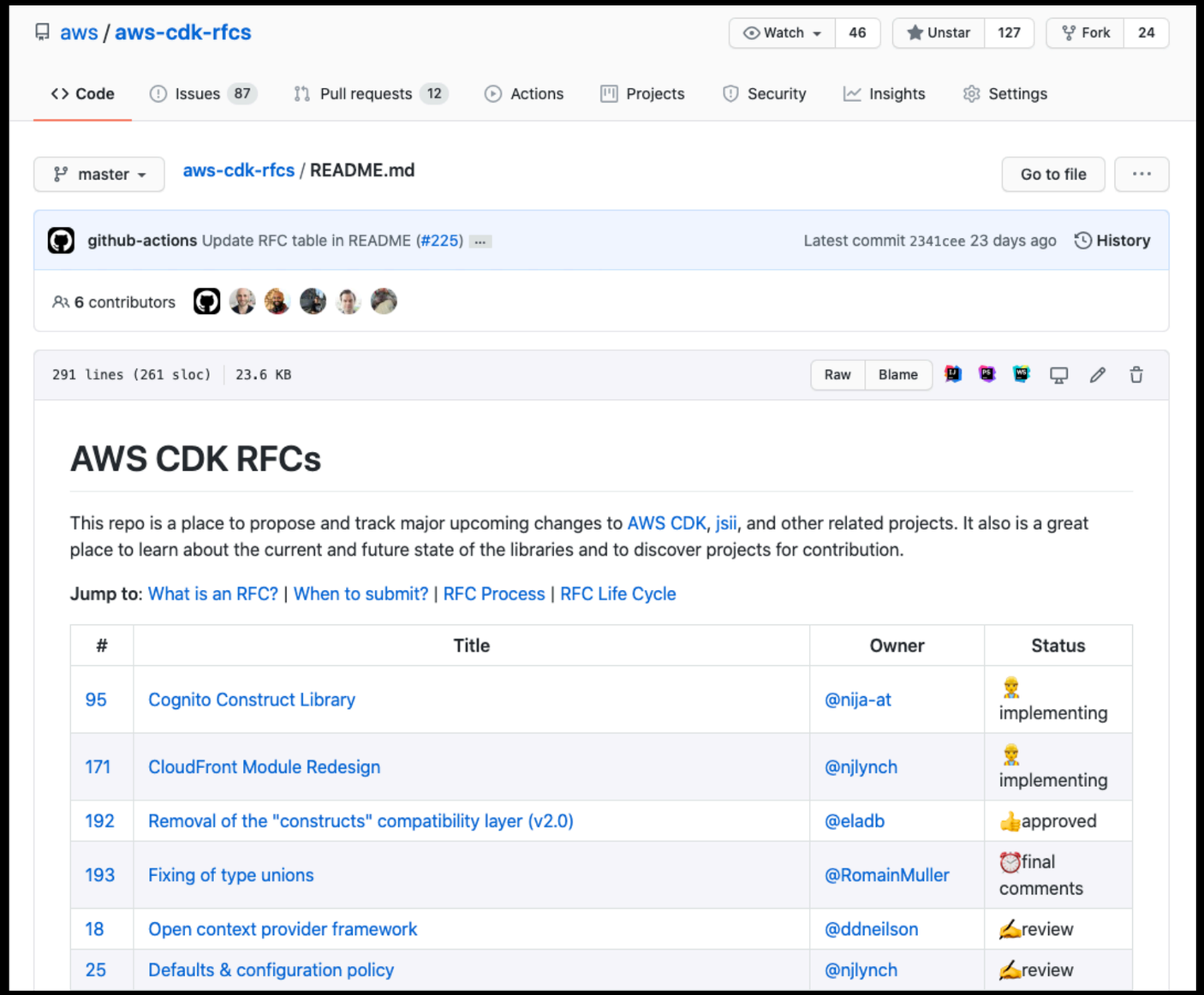


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK Request for Comments (RFCs)

- GitHub: aws/aws-cdk-rfcs
- +1, comment, and track proposed major changes and features



CDK v2

- **Simplified module structure** – One module instead of 200
- **Backwards compatible guarantee** – All experimental code moved into a separate incubator module
- **Minor code cleanup** – Removed @deprecated code, updated defaults (e.g., default AWS Lambda runtime versions)



Go CDK

```go
func main() {
  app := cdk.NewApp()
  stack := cdk.NewStack(app, "aws-ecs-integ-ecs")

  // Create a cluster
  vpc := ec2.NewVpc(stack, "Vpc", ec2.VpcProps{MaxAzs: 2})

  cluster := ecs.NewCluster(stack, "EcsCluster", ecs.ClusterProps{Vpc: vpc})
  cluster.addCapacity("DefaultAutoScalingGroup", ecs.AddCapacityOptions{
    InstanceType: ec2.InstanceClass.T2,
    InstanceSize: ec2.InstanceSize.MICRO,
  })

  // Create Task Definition
  taskdefinition := ecs.NewEc2TaskDefinition(stack, "TaskDef")
  container := taskdefinition.AddContainer("web", ecs.AddContainerOptions{
    Image:          ecs.ContainerImage.fromRegistry("amazon/amazon-ecs-sample"),
    MemoryLimitMiB: 256,
  })

  container.AddPortMappings([]ecs.PortMapping{
    {
      ContainerPort: 80,
      HostPort:      8080,
      Protocol:      ecs.Protocol.TCP,
    },
  })
```





Wrap up!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

AWS Construct Library

- re:Invent 2019: Contributing to the AWS Construct Library

CDK pipelines

- AWS Online Tech Talk: Enhanced CI/CD with AWS CDK

Ecosystem: AWS Solutions Constructs

- AWS re:Invent 2020: Turbocharge the AWS CDK with AWS Solutions Constructs

CDK for Terraform (cdktf)

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

CDK for Kubernetes (cdk8s)

- AWS re:Invent 2020: Simplifying Kubernetes application management with cdk8s

Thank you!

Jason Fulghum

AWS Developer Tools

@JASON_FULGHUM

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.