AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

BOA308-R1

Build production-ready full-stack applications with GraphQL

Gunnar Grosch (He/Him)
Senior Developer Advocate
AWS

Marcia Villalba (She/Her)
Senior Developer Advocate
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“How can I deliver this application to production?”

An AWS developer

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

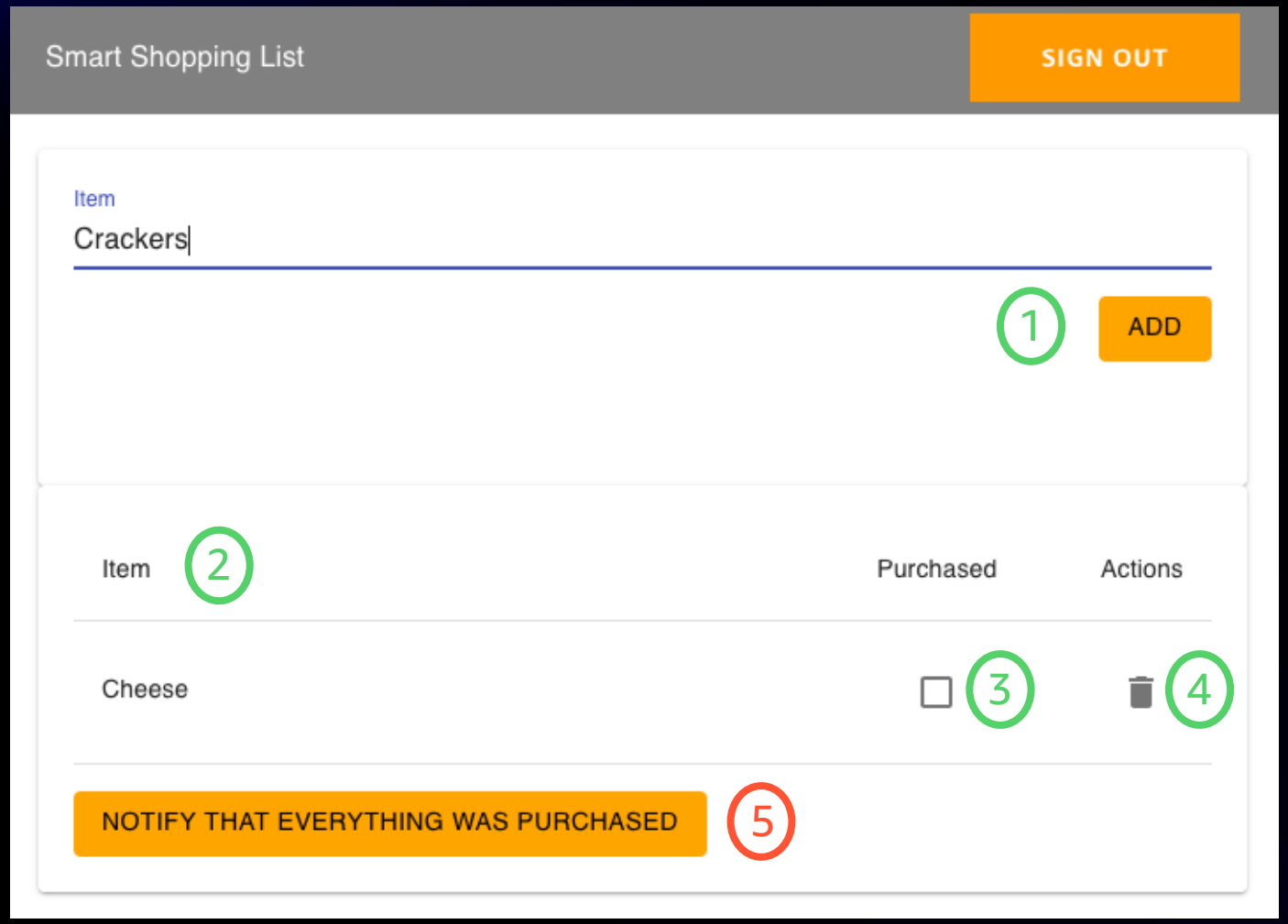
- Building the application
- Automating the delivery
- Maintaining and improving

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Part 1 – Developing the applications

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does it do?



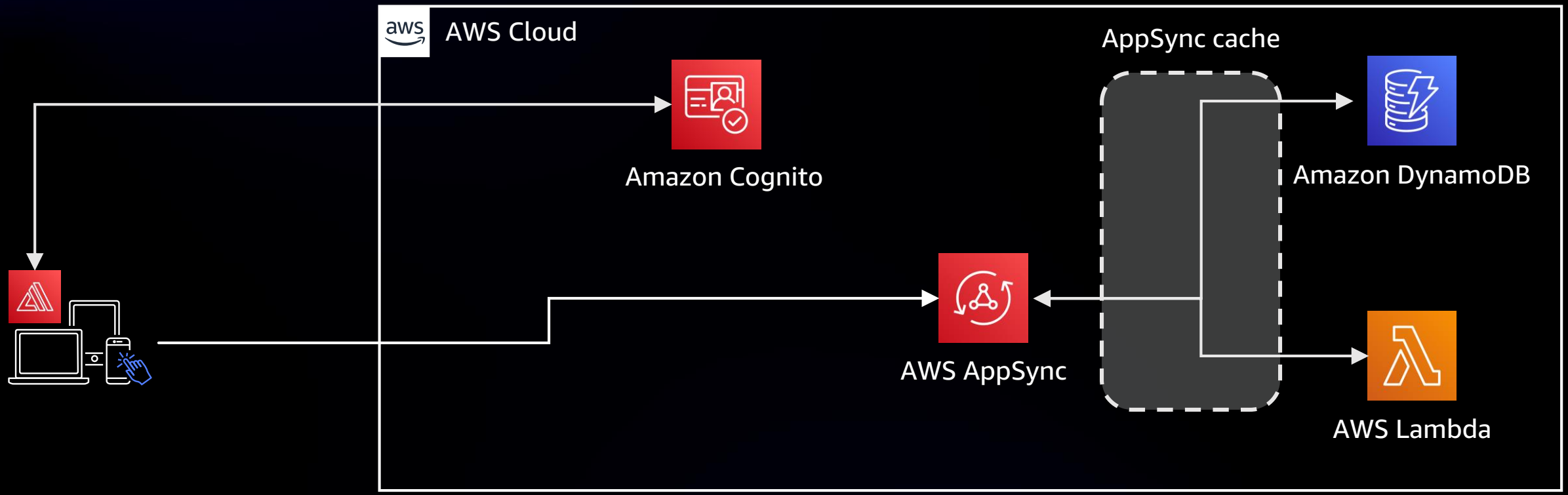
Shopping actions:

- Add new Shopping Item
- View Shopping Items
- Buy Shopping Item
- Delete Shopping Item
- Complete Shopping List

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Production-ready GraphQL with AWS AppSync

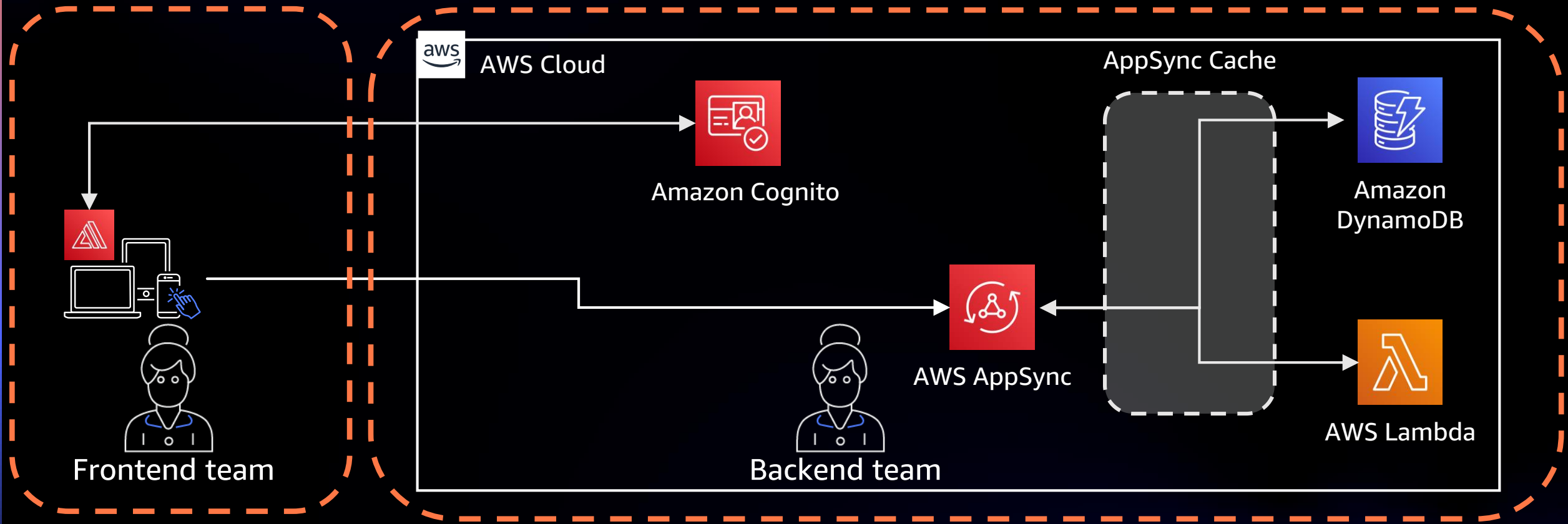
BACKEND DEPLOYED WITH AWS CDK



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How we are building it...

DISTRIBUTED TEAMS



Git:\frontend-repo

Git:\backend-repo

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Development choices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why AWS CDK?



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
    });
    // 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.





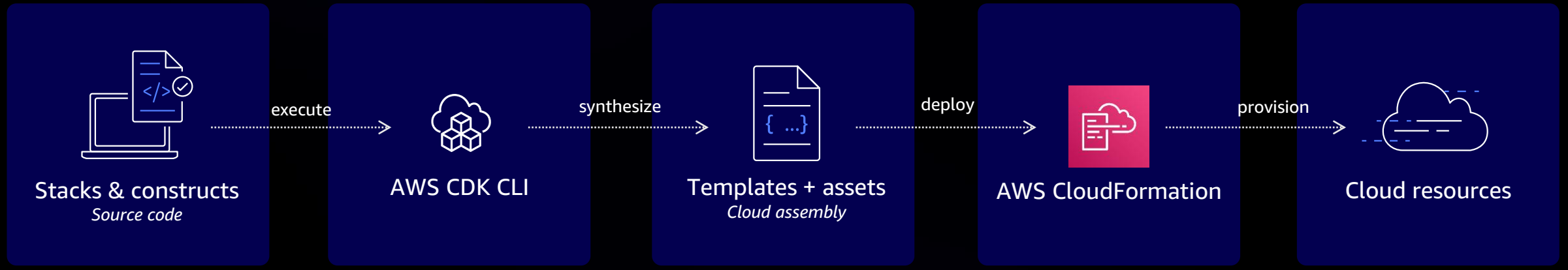




aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deploying a backend




```
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
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why AWS Amplify?

Open Source Amplify framework

Libraries



Authentication
Storage
API
DataStore

CLI



PubSub
Predictions
Interactions
Notifications

Analytics

AWS Amplify Console service

Static hosting



Hosting for full stack web apps with continuous deployment

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Choose the parts you need

Open Source Amplify framework





Authentication
~~Storage~~
API
~~DataStore~~

~~PubSub~~
~~Predictions~~
~~Interactions~~
~~Notifications~~

~~Analytics~~

AWS Amplify Console service



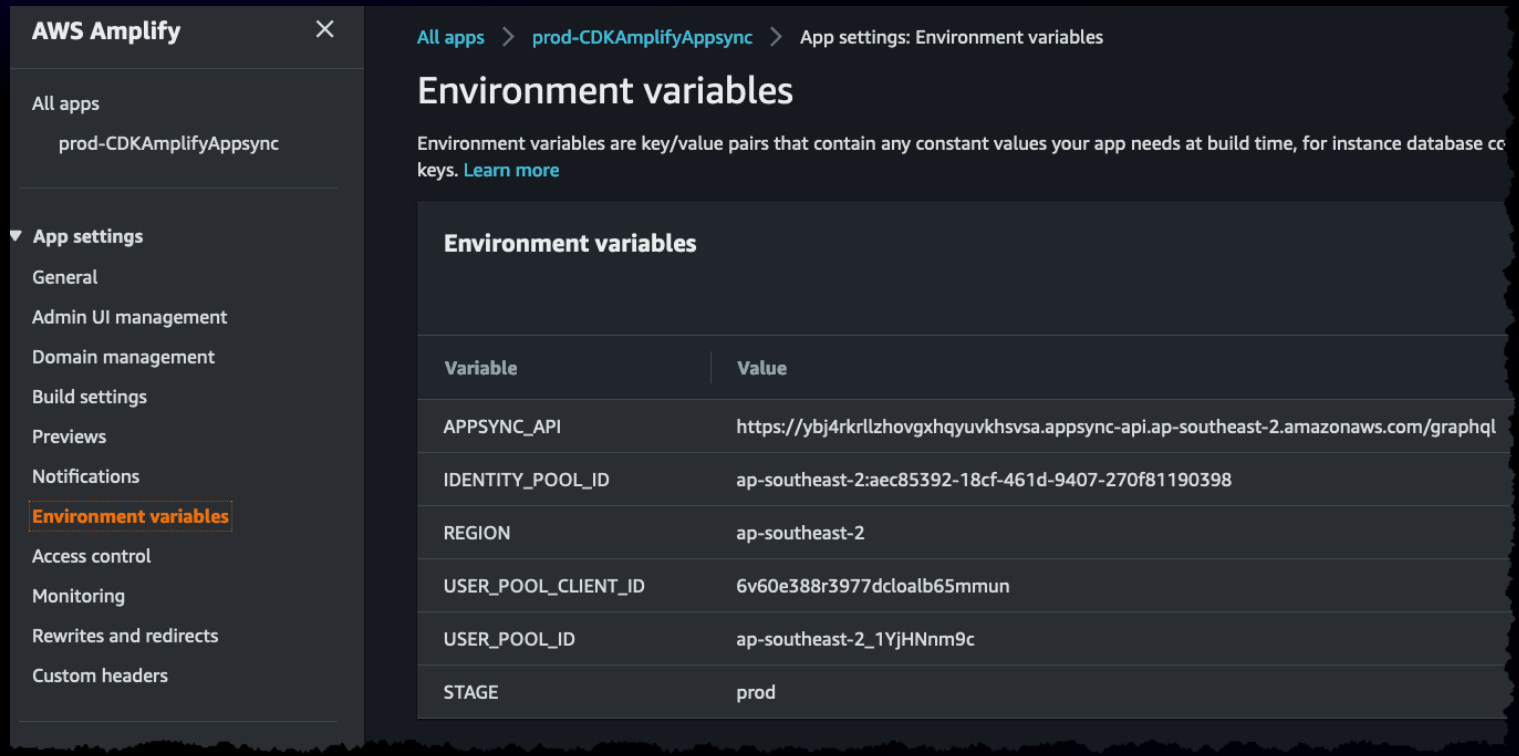
Hosting for full stack web apps with continuous deployment

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Connecting CDK and Amplify

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CDK deploy



~/cdk deploy {stack-name} --profile $(AWS_PROFILE) --require-approval never \ --outputs-file {output-json-file}

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Run the web app in localhost

.env.local

```
REACT_APP_IDENTITY_POOL_ID=eu-north-1:123456-111d-11ce-9a8e-1111111111

REACT_APP_USER_POOL_ID=eu-north-1_1111111

REACT_APP_USER_POOL_CLIENT_ID=sdfsdfsdf111111111

REACT_APP_REGION=eu-north-1

REACT_APP_APPSYNC_API=https://x43yhjsskvdb5l6k5sfjlmclrq.appsync-api.eu-north-1.amazonaws.com/graphql
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Configuring Amplify with CDK

```
Amplify.configure({
        Auth: {
          region: process.env.REACT_APP_REGION,
          userPoolId: process.env.REACT_APP_USER_POOL_ID,
          userPoolWebClientId: process.env.REACT_APP_USER_POOL_CLIENT_ID,
          identityPoolId: process.env.REACT_APP_IDENTITY_POOL_ID
        },
        API: {
          graphql_headers: async () => ({
            Authorization: (awaitAuth.currentSession()).getIdToken().getJwtToken()
          }),
          graphql_endpoint: process.env.REACT_APP_APPSYNC_API
        },
        aws_appsync_region: process.env.REACT_APP_REGION,
        aws_appsync_authenticationType: "AMAZON_COGNITO_USER_POOLS",
        aws_appsync_graphqlEndpoint: process.env.REACT_APP_APPSYNC_API
});
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keeping the frontend and backend in sync

Generate the graphql.schema in your backend

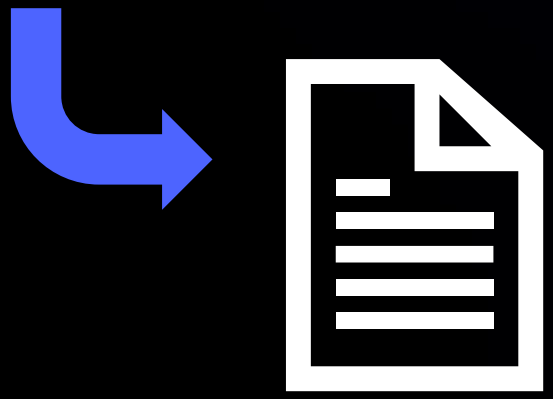
```
$ npm run build
$ cdk synth
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keeping the frontend and backend in sync

Generate the graphql.schema in your backend

```
$ npm run build
$ cdk synth
```



new appsync/graphql.schema

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keeping the frontend and backend in sync

Generate the graphql.schema in your backend

```
$ npm run build
$ cdk synth
```



new appsync/graphql.schema

In your frontend copy the file to graphql directory

```
$ amplify add codegen
```

Allows you to add AppSync API created in the backend to the project

```
$ amplify codegen types
```

Generates the GraphQL types based on the schema

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we learned in this part

- Why AWS CDK or AWS Amplify?
- Building authentication, GraphQL API and Amplify app from CDK
- How to use the third-party library – CDK AppSync Transformer
- Have multiple clients sharing the same backend
- Testing locally your react application

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Part 2 – Automate everything!

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we are building?

- Automate the frontend and backend builds and deployments
- Make sure that frontend and backend teams can work independently
- Have 2 stages – develop and production
- Do everything as infrastructure as code

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitFlow

Primary branches + Feature branches

Branches with specific roles and how merging works

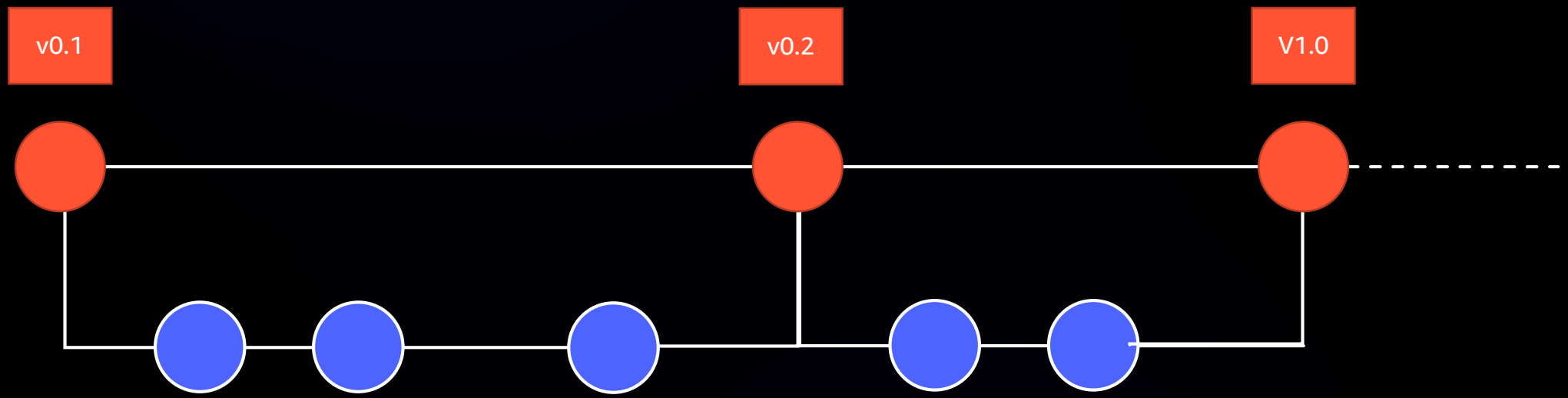
Good for projects with scheduled releases

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitFlow

Main

Develop



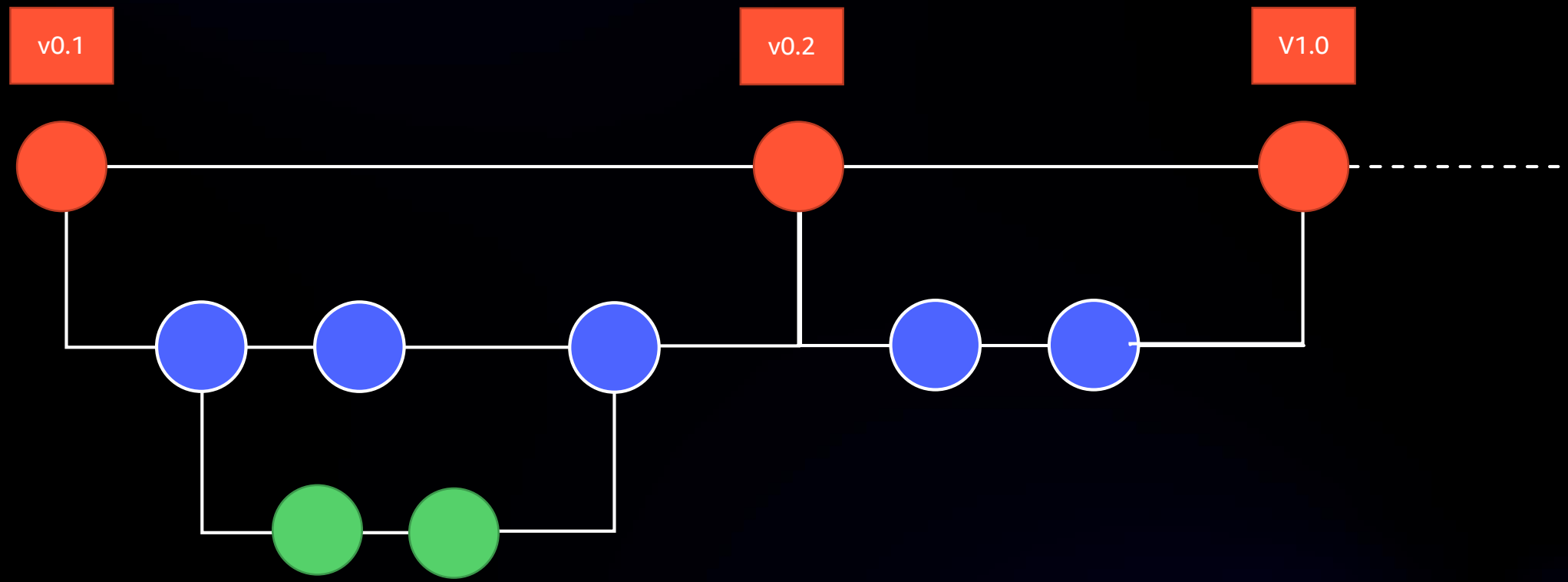
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GitFlow

Main

Develop

Feature



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend

```
const amplifyApp = new amplify.App(this, `${props.stage}-CDKAmplifyAppsync`, {
  sourceCodeProvider: new amplify.GitHubSourceCodeProvider({
    …
  }),
  environmentVariables: {
   …
  },
  autoBranchCreation: {
    patterns: ['feature/*']
  },
  autoBranchDeletion: true
});

if (props.stage === 'prod') {
  const main = amplifyApp.addBranch("main");
  main.addEnvironment('STAGE', 'prod');
} else {
  const dev = amplifyApp.addBranch('develop');
  dev.addEnvironment('STAGE', 'dev')
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend

```
const amplifyApp = new amplify.App(this, `${props.stage}-CDKAmplifyAppsync`, {
  sourceCodeProvider: new amplify.GitHubSourceCodeProvider({
    …
  }),
  environmentVariables: {
    …
  },
  autoBranchCreation: {
    patterns: ['feature/*']
  },
  autoBranchDeletion: true
});

if (props.stage === 'prod') {
  const main = amplifyApp.addBranch("main");
  main.addEnvironment('STAGE', 'prod');
} else {
  const dev = amplifyApp.addBranch('develop');
  dev.addEnvironment('STAGE', 'dev')
}
```

Creates automatically new amplify environment when a new branch named “featured/*” is created

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend

```
const amplifyApp = new amplify.App(this, `${props.stage}-CDKAmplifyAppsync`, {
  sourceCodeProvider: new amplify.GitHubSourceCodeProvider({
    …
  }),
  environmentVariables: {
   …
  },
  autoBranchCreation: {
    patterns: ['feature/*']
  },
  autoBranchDeletion: true
});

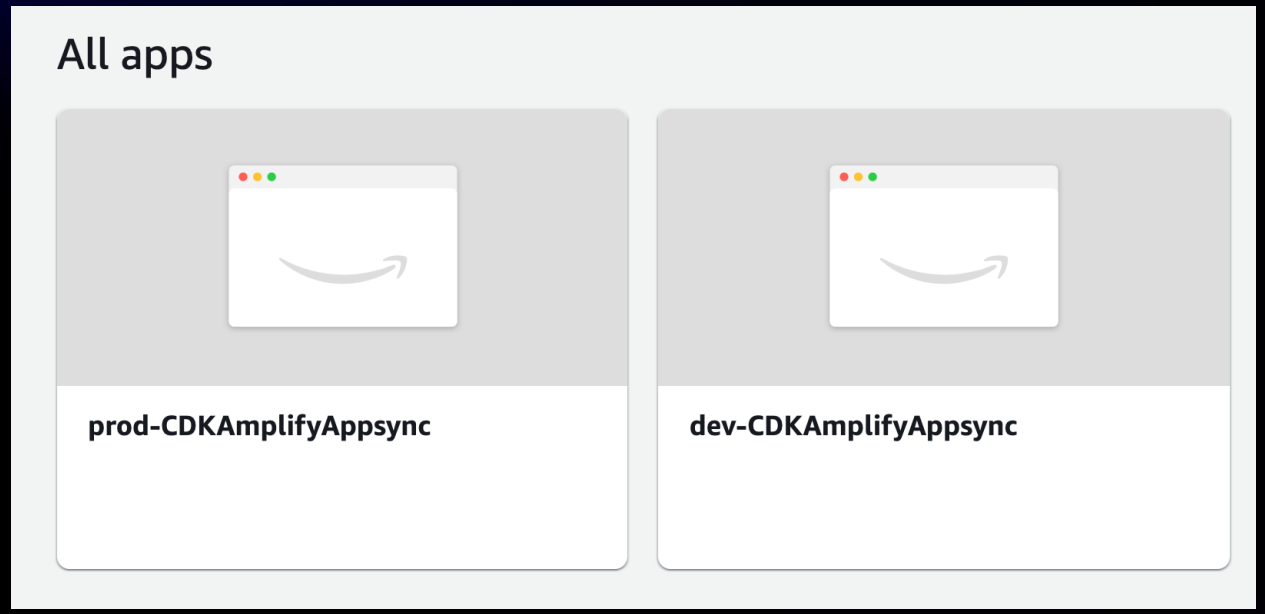
if (props.stage === 'prod') {
  const main = amplifyApp.addBranch("main");
  main.addEnvironment('STAGE', 'prod');
} else {
  const dev = amplifyApp.addBranch('develop');
  dev.addEnvironment('STAGE', 'dev')
}
```

Creates two Amplify applications and adds the right environmental variable to each of them

aws

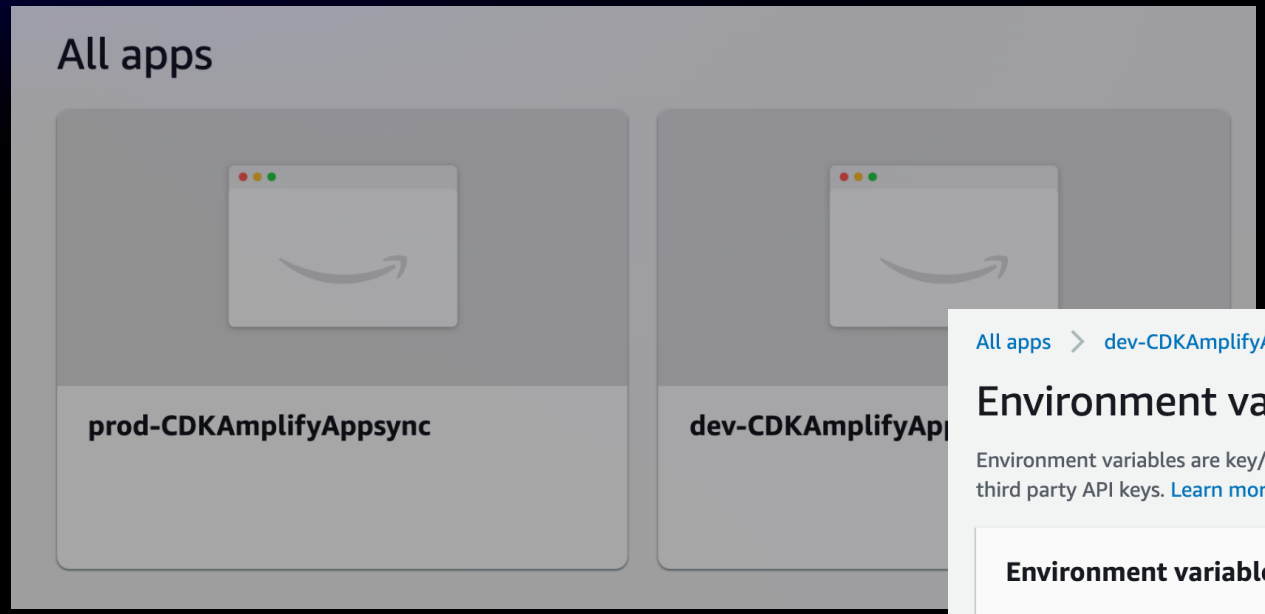
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Frontend



All apps > dev-CDKAmplifyAppsync > App settings: Environment variables

Environment variables

Environment variables are key/value pairs that contain any constant values your app needs at build time, for instance database connection details or third party API keys. Learn more

Environment variables

Manage variables

1

Variable	Value	Branch
APPSYNC_API	https://[redacted].amazonaws.com/graphql	All branches
IDENTITY_POOL_ID	[redacted]	All branches
REGION	eu-north-1	All branches
USER_POOL_CLIENT_ID	[redacted]	All branches
USER_POOL_ID	eu-north-1_[redacted]	All branches
STAGE	dev	develop

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

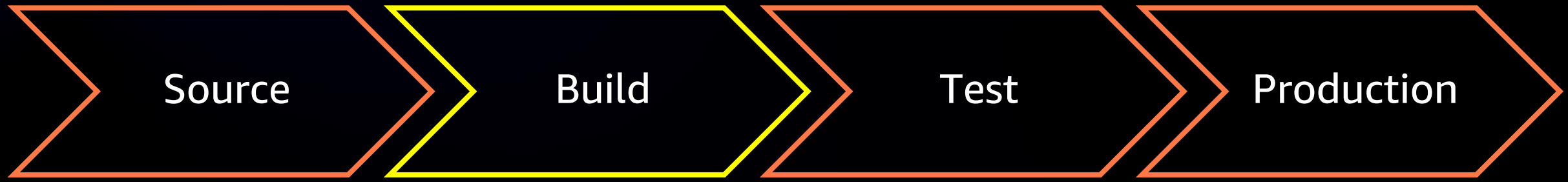
Backend



Github

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend



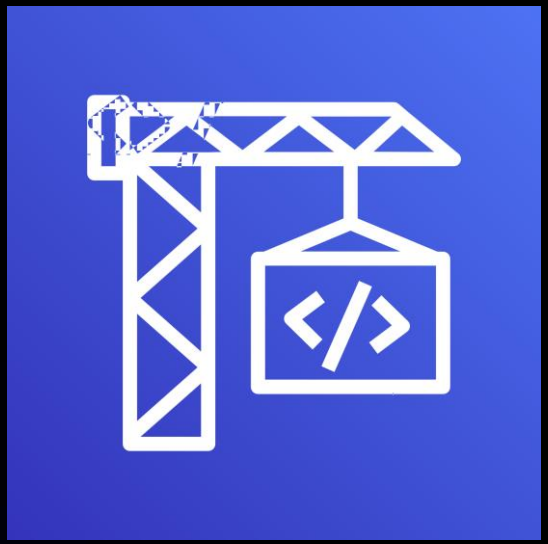
Github

AWS CodeBuild

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

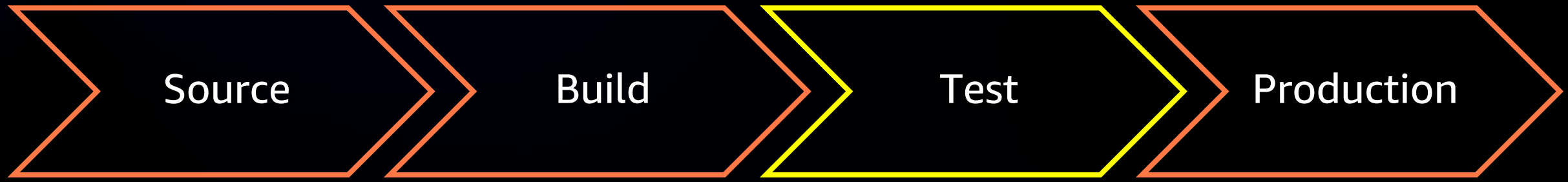
AWS CodeBuild

- Fully managed build service that can compile source code, run tests, and produce software packages
- Scales continuously and processes multiple builds concurrently
- No build servers to manage
- Pay by the minute, for only the compute resources you use
- https://aws.amazon.com/codebuild/



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend



Github

AWS CodeBuild

Staging environment
CloudFormation tests
Integration tests
Manual approval

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend



Github

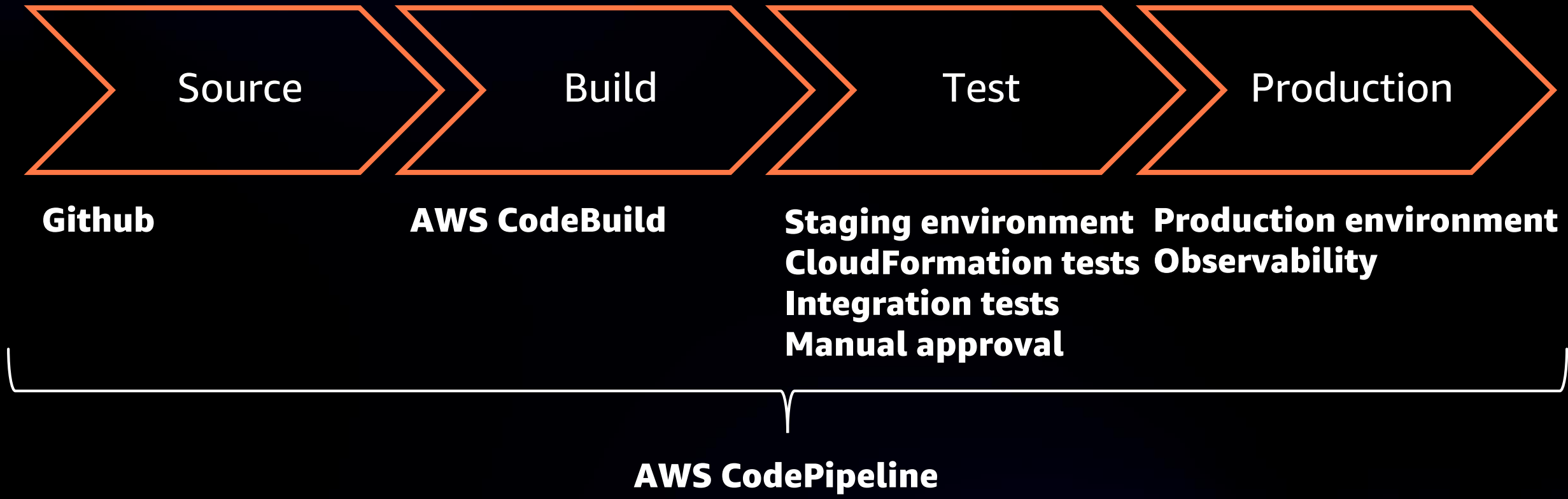
AWS CodeBuild

Staging environment
CloudFormation tests
Integration tests
Manual approval

Production environment
Observability

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Backend



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

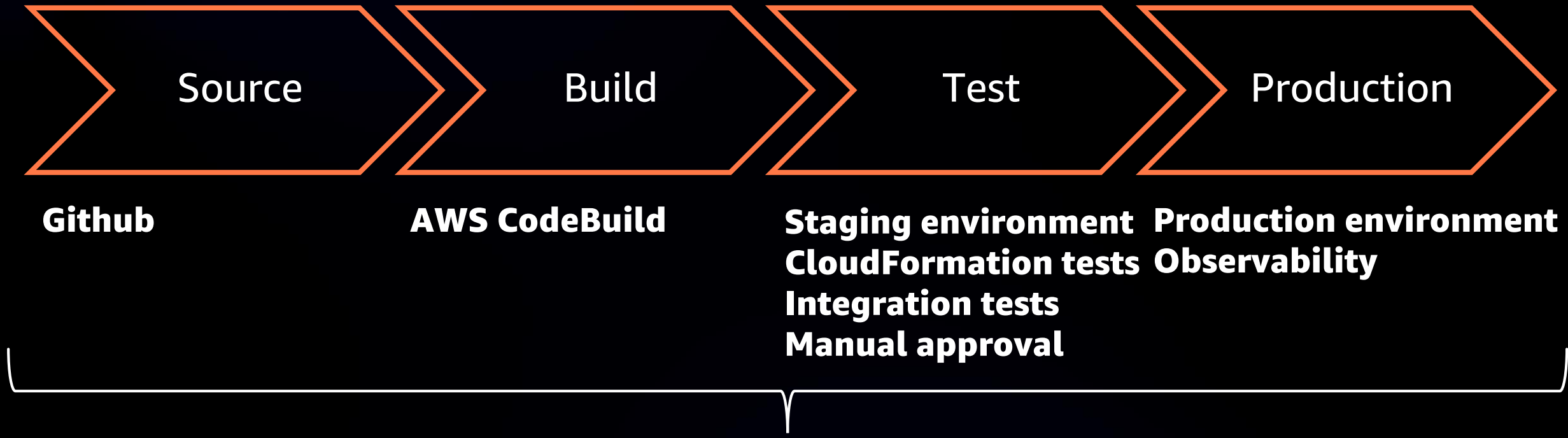
AWS CodePipeline

- Continuous delivery service for fast and reliable application updates
- Model and visualize your software release process
- Builds, tests, and deploys your code every time there is a code change
- Integrates with third-party tools and AWS
- https://aws.amazon.com/codepipeline/



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

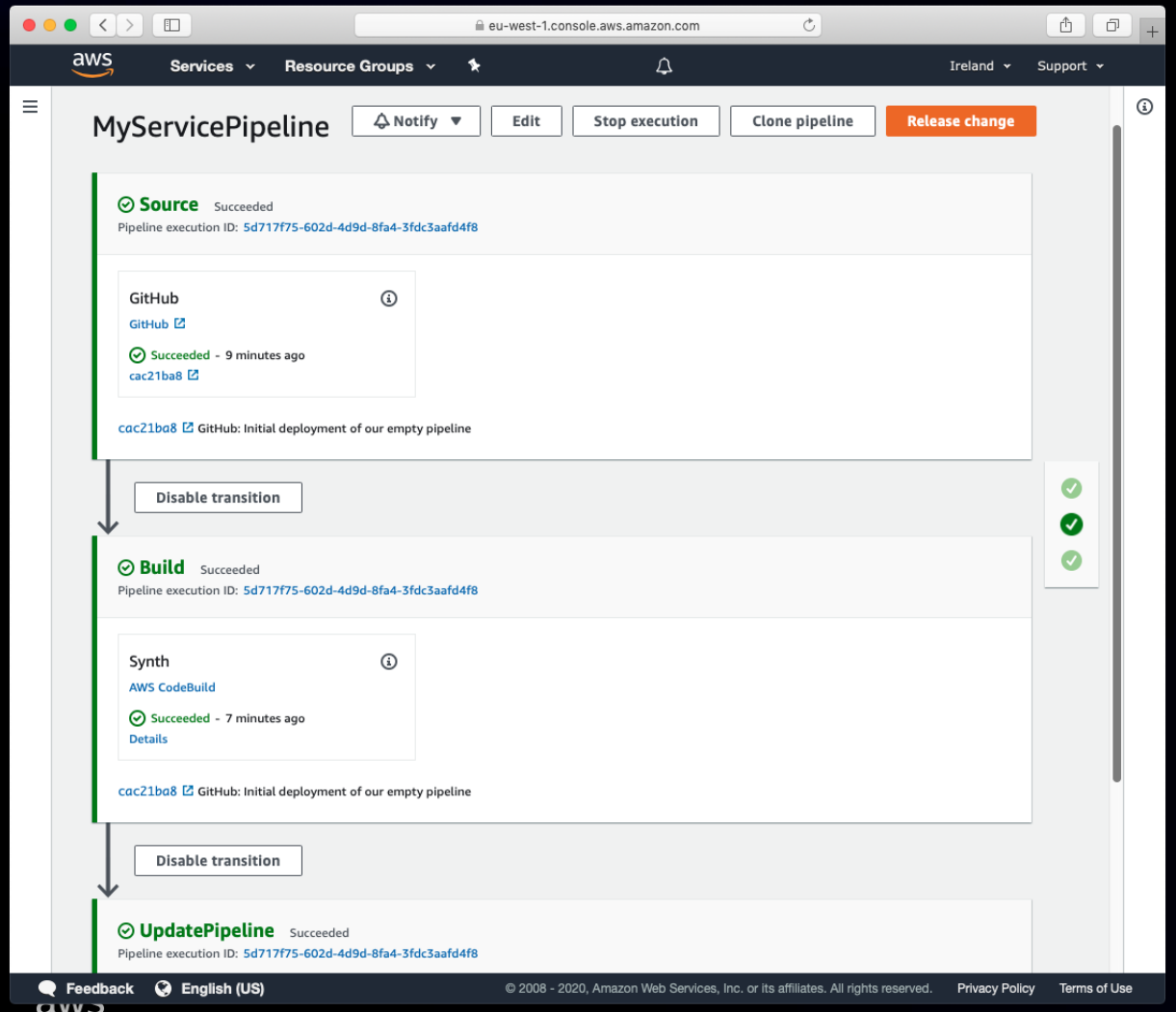
Backend



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK pipelines

CONTINUOUS DELIVERY FOR AWS CDK APPLICATIONS

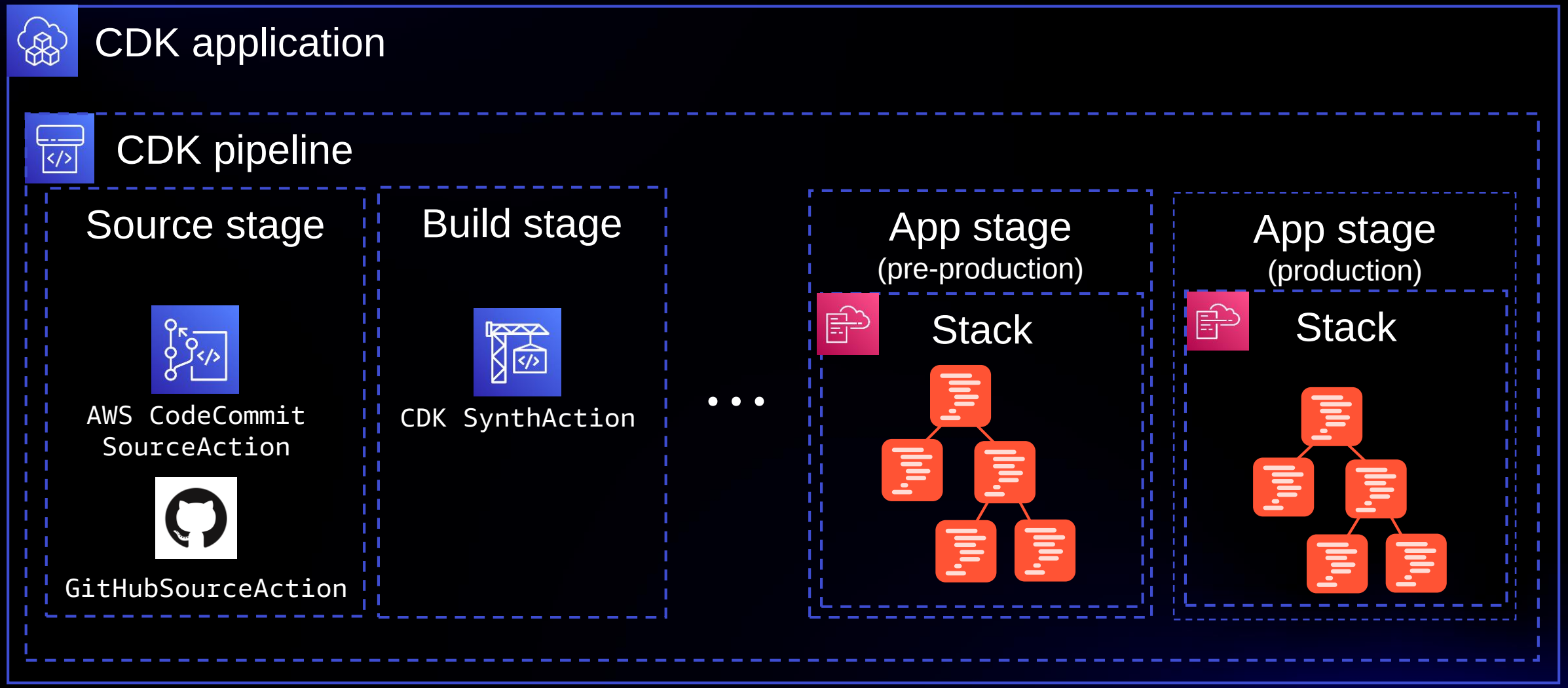


- Model continuous delivery pipelines as part of your infrastructure code
- Pipelines are self-modifying as you push your AWS CDK code to origin
- Easily model cross-account and cross-Region pipeline configurations

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

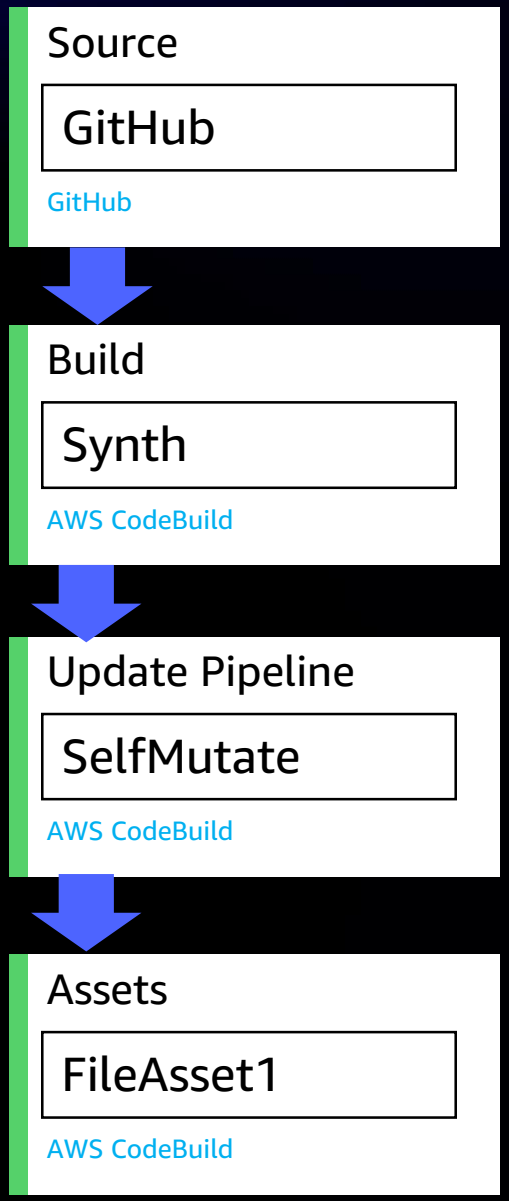
AWS CDK pipelines

CONTINUOUS DELIVERY FOR AWS CDK APPLICATIONS



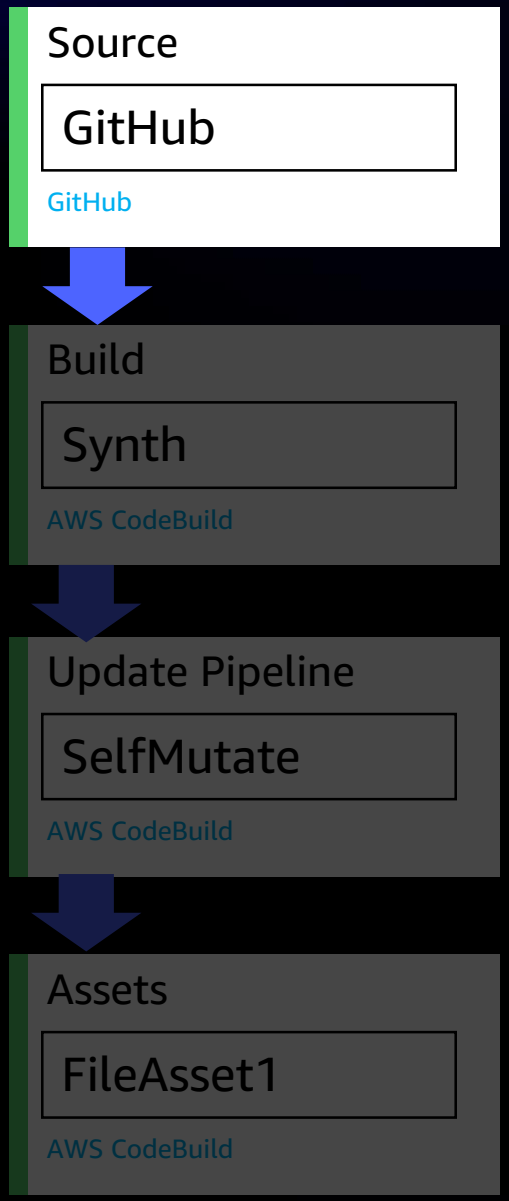
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK pipelines



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

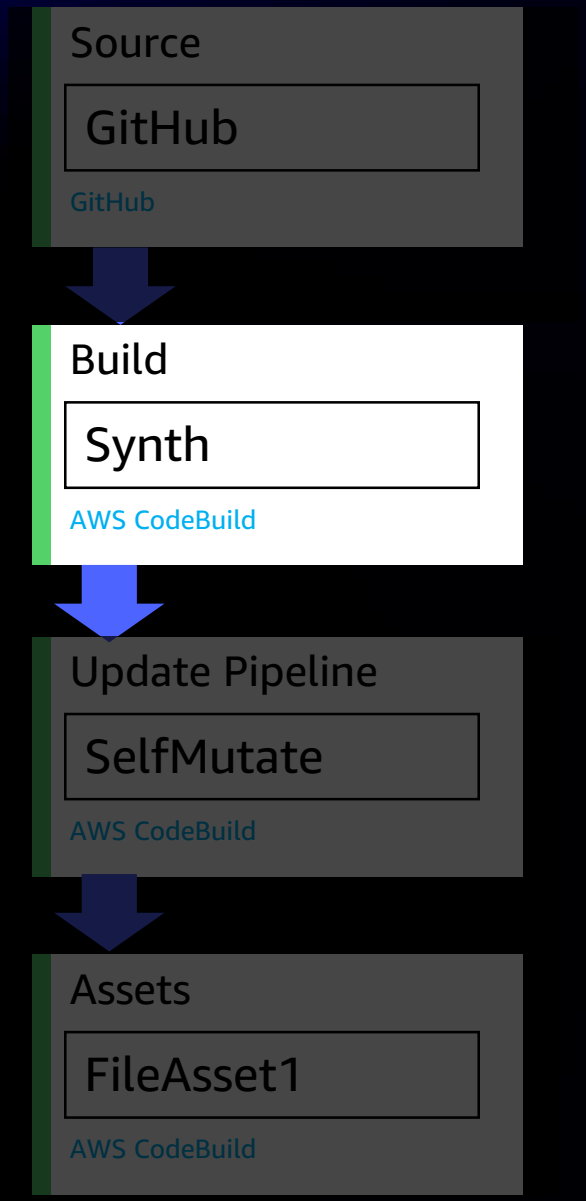
AWS CDK pipelines




```
sourceAction: new codepipeline_actions.GitHubSourceAction({
      actionName: 'GitHub',
      output: sourceArtifact,
      oauthToken: SecretValue.secretsManager('github-token'),
      owner: config.cdk_pipeline.repository_owner,
      repo: config.cdk_pipeline.repository_name,
      branch: config.cdk_pipeline.branch
    }),
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

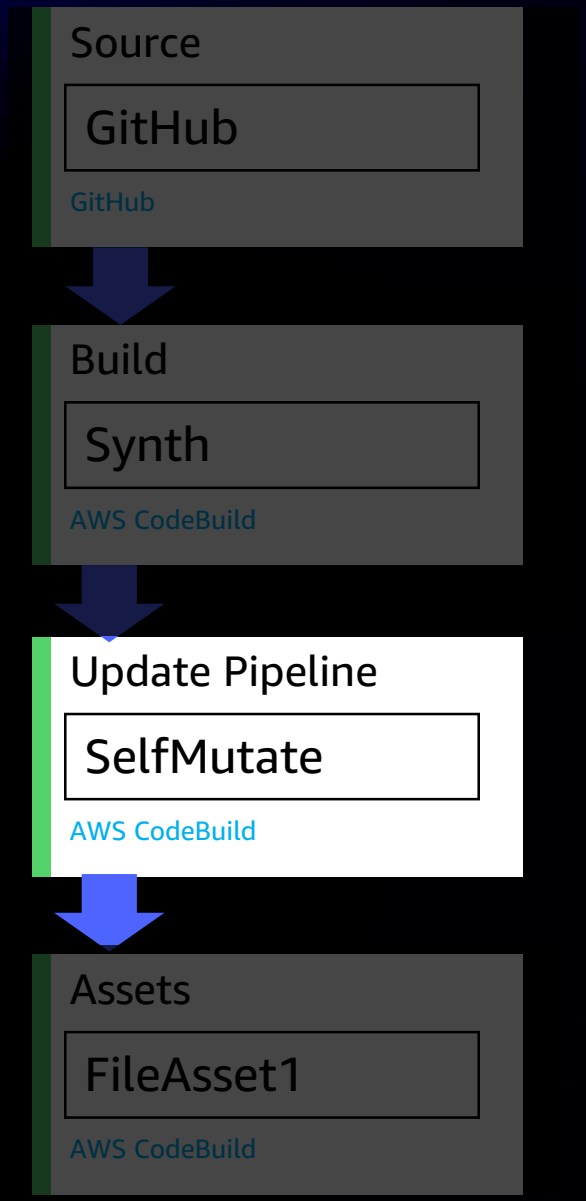
AWS CDK pipelines




```
synthAction: SimpleSynthAction.standardNpmSynth({
      sourceArtifact,
      cloudAssemblyArtifact,
      buildCommand: 'npm run build'
    }),
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

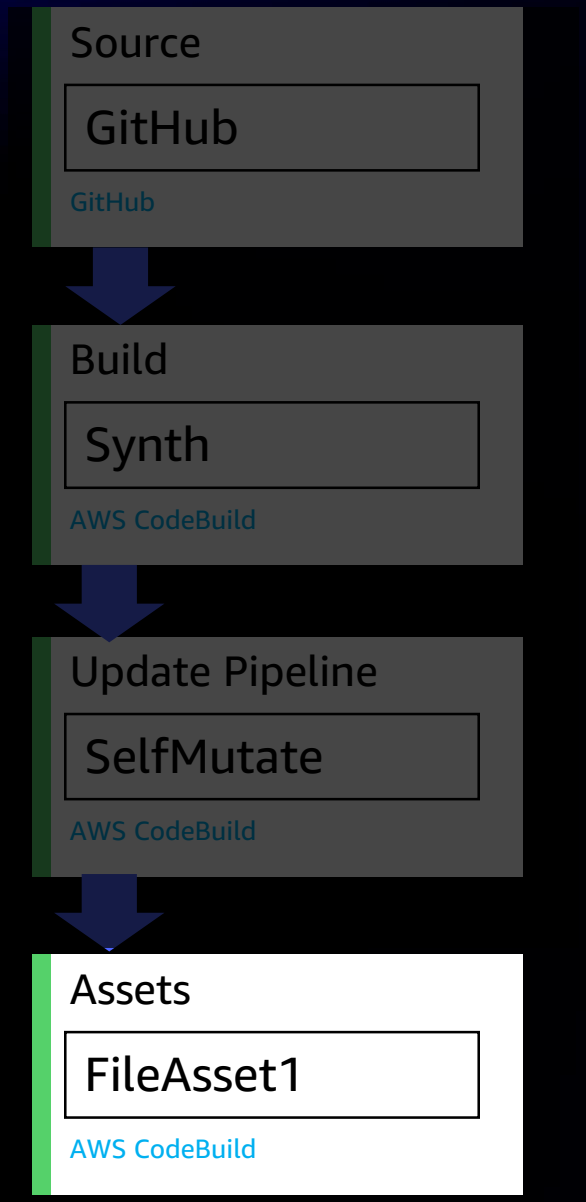
AWS CDK pipelines



The pipeline will update itself now, so the original pipeline gets destroyed and a new pipeline with the updated code is created

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK pipelines




```
const preprod = new ReinventCdkPipelineStageStack(this,
config.environments.dev.name, {
       env: { account: config.environments.dev.env.account, region:
config.environments.dev.env.region }
     });


    const preprodStage = pipeline.addApplicationStage(preprod);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Things to have in mind

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Environmental variables - Secrets

Don’t store secrets in a source repo or visible to everyone!

Use Secret Manager or Parameter store with encrypted params

```
const amplifyApp = new amplify.App(this, `${props.stage}-CDKAmplifyAppsync`, {
  sourceCodeProvider: new amplify.GitHubSourceCodeProvider({
    owner: config.frontend.owner,
    repository: config.frontend.repository_name,
    oauthToken: cdk.SecretValue.secretsManager('github-token’)
  }),
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Environmental variables - Configuration

```
{
  "cdk_pipeline": {
    "pipeline_name": "CDKAmplifyPipeline",
    "repository_name": ”repo-name”,
    "repository_owner": ”aws-samples",
    "branch": "main",
    "build_environment": {
       "account": ”1232323232",
       "region": "eu-north-1"
    }
  },
  "frontend": {
     "repository_name":”repo-name",
     "owner": ”aws-samples"
   },
  "environments": {
    "dev": {
      "name": "dev",
      "env": {
        "account": ”13232323",
        "region": "eu-north-1"
    }
  },
   "production": {
      "name": "prod",
      "env": {
       "account": ”23232323",
       "region": "eu-north-1"
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Environmental variables - Configuration

```
{
 "cdk_pipeline": {
  "pipeline_name": "CDKAmplifyPipeline",
  "repository_name": "repo-name",
  "repository_owner": "aws-samples",
  "branch": "main",
  "build_environment": {
   "account": "1232323232",
   "region": "eu-north-1"
```

```
import * as config from '../config.json'


const amplifyApp = new amplify.App(this, `${props.stage}-CDKAmplifyAppsync`, {
  sourceCodeProvider: new amplify.GitHubSourceCodeProvider({
    owner: config.frontend.owner,
    repository: config.frontend.repository_name,
    oauthToken: cdk.SecretValue.secretsManager('github-token')
  }),
```

```
   "env": {
    "account": "23232323",
    "region": "eu-north-1"
   }
  }
}
```



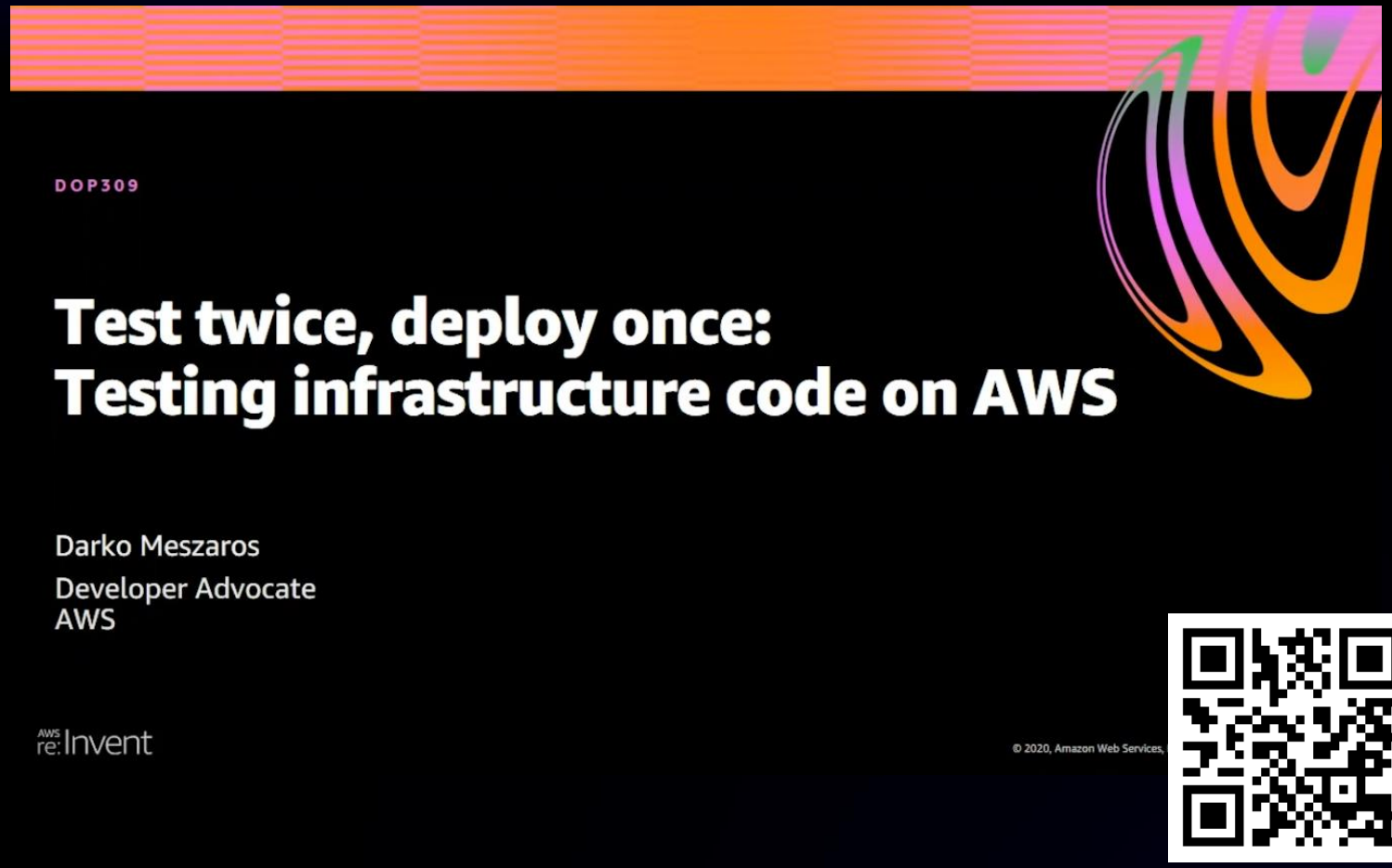
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Naming things

```
const userPool = new cognito.UserPool(this, `${props.stage}-AmplifyCDKUserPool`, {
   selfSignUpEnabled: true,
   autoVerify: { email: true },
   signInAliases: { email: true },
});
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unit tests



https://youtu.be/qlzX65SdNdQ

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Unit tests

In your pipeline

```
synthAction: SimpleSynthAction.standardNpmSynth({
      sourceArtifact,
      cloudAssemblyArtifact,
      buildCommand: 'npm run build’
    }),
```

In your package.json

```
"scripts": {
  "build": "tsc && npm run test",
  "watch": "tsc -w",
  "test": "jest",
  "cdk": "cdk"
},
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we learned in this part

- GitFlow with Amplify
- CDK pipelines
- Naming conventions for CDK projects
- Environmental variables – Secrets and configurations
- Unit tests
- Keeping the frontend and backend in sync

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Part 3 – Maintaining and improving

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are we building?

- Add integrations tests to our CDK pipeline
- Create dashboard and add monitoring and alarms
- Do everything as infrastructure as code

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

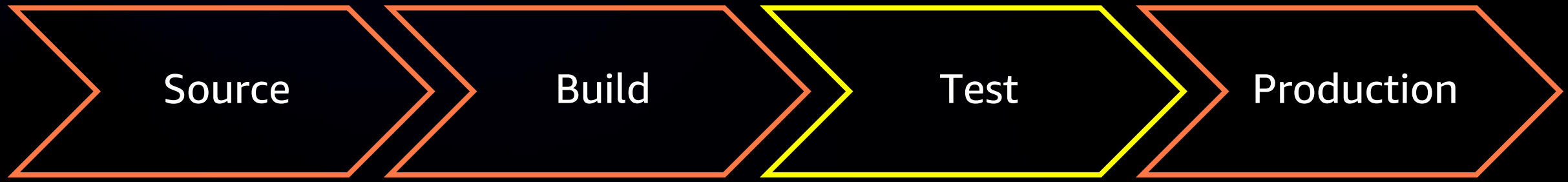
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stages of the software release process



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stages of the software release process



Developers' version – review and commit code changes

Build tasks and unit testing

Deploy to testing environments and run integration tests

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING SHELLSTEP

```
import { ShellScriptAction } from '@aws-cdk/pipelines';

// ...

const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
    new ShellStep('IntegrationTest', {
      commands: [
        'curl -Ssf $ENDPOINT_URL',
      ],
      envFromCfnOutputs: {
        ENDPOINT_URL: preprod.urlOutput,
      },
    }),
  ],
});
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING SHELLSTEP

```
import { ShellScriptAction } from '@aws-cdk/pipelines';

// ...

const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
    new ShellStep('IntegrationTest', {
      commands: [
        'curl -Ssf $ENDPOINT_URL',
      ],
      envFromCfnOutputs: {
        ENDPOINT_URL: preprod.urlOutput,
      },
    }),
  ],
});
```

Adds a ShellStep to the stage

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING SHELLSTEP

```
import { ShellScriptAction } from '@aws-cdk/pipelines';

// ...

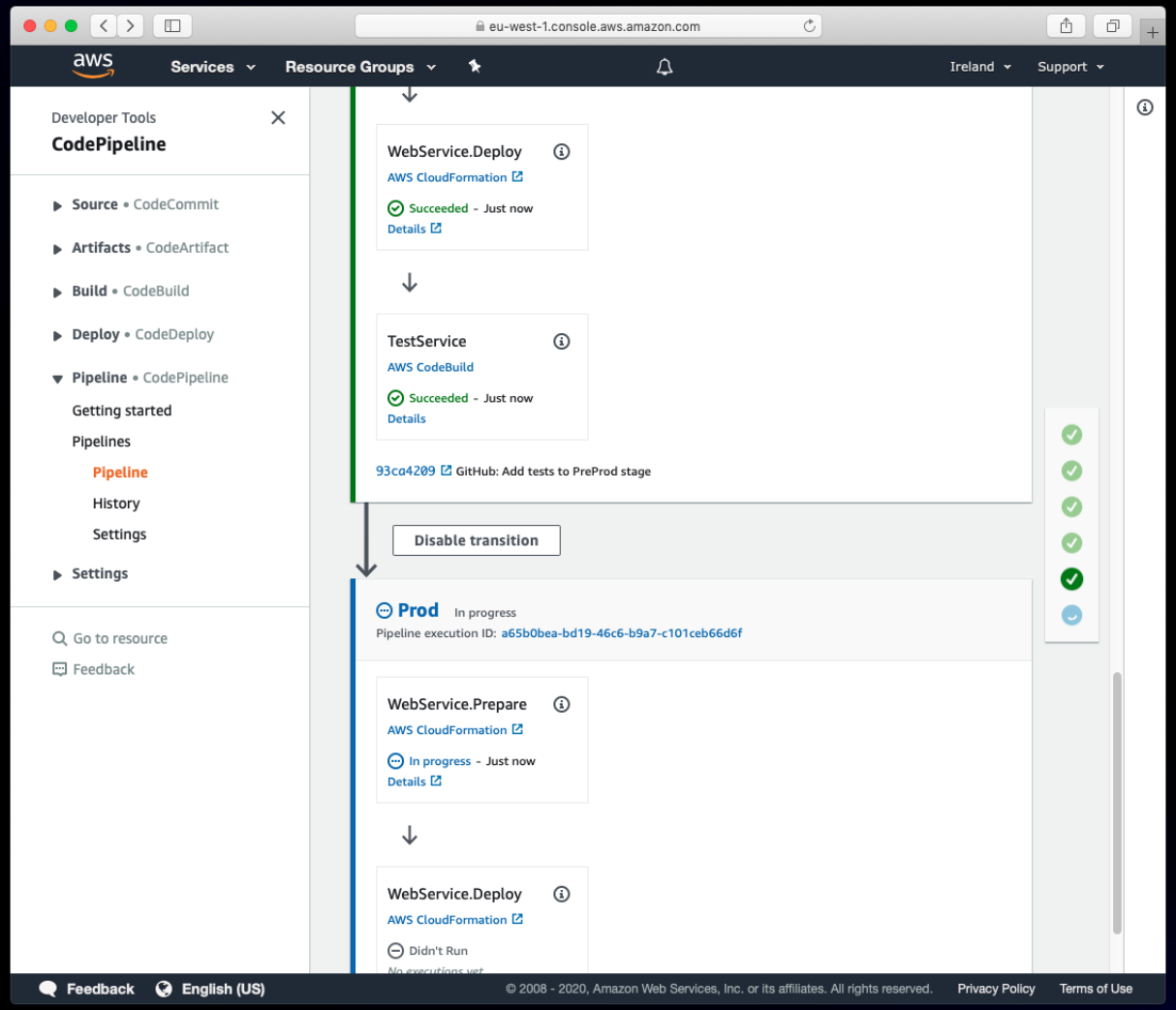
const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
    new ShellStep('IntegrationTest', {
      commands: [
        'curl -Ssf $ENDPOINT_URL',
      ],
      envFromCfnOutputs: {
        ENDPOINT_URL: preprod.urlOutput,
      },
    }),
  ],
});
```

Runs curl command against the url

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING SHELLSTEP



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
exports.handler = async (event, context, callback) => {
  const jobId = event["CodePipeline.job"].id;
  try {
    // ADD INTEGRATION TESTS
    const jobParams = {
      jobId: jobId
    };
    const jobData = await codepipeline.send(new PutJobSuccessResultCommand(jobParams));
  } catch (err) {
    const jobParams = {
      jobId: jobId,
      failureDetails: {
        message: JSON.stringify(err),
        type: "JobFailed",
        externalExecutionId: context.awsRequestId,
      },
    };
    const jobData = await codepipeline.send(new PutJobFailureResultCommand(jobParams));
    console.log("Error", err);
  }
};
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
exports.handler = async (event, context, callback) => {
  const jobId = event["CodePipeline.job"].id;
  try {
    // ADD INTEGRATION TESTS
    const jobParams = {
      jobId: jobId
    };
    const jobData = await codepipeline.send(new PutJobSuccessResultCommand(jobParams));
  } catch (err) {
    const jobParams = {
      jobId: jobId,
      failureDetails: {
        message: JSON.stringify(err),
        type: "JobFailed",
        externalExecutionId: context.awsRequestId,
      },
    };
    const jobData = await codepipeline.send(new PutJobFailureResultCommand(jobParams));
    console.log("Error", err);
  }
};
```

Write integration tests

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
exports.handler = async (event, context, callback) => {
  const jobId = event["CodePipeline.job"].id;
  try {
    // ADD INTEGRATION TESTS
    const jobParams = {
      jobId: jobId
    };
    const jobData = await codepipeline.send(new PutJobSuccessResultCommand(jobParams));
  } catch (err) {
    const jobParams = {
      jobId: jobId,
      failureDetails: {
        message: JSON.stringify(err),
        type: "JobFailed",
        externalExecutionId: context.awsRequestId,
      },
    };
    const jobData = await codepipeline.send(new PutJobFailureResultCommand(jobParams));
    console.log("Error", err);
  }
};
```

Report success back

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
exports.handler = async (event, context, callback) => {
  const jobId = event["CodePipeline.job"].id;
  try {
    // ADD INTEGRATION TESTS
    const jobParams = {
      jobId: jobId
    };
    const jobData = await codepipeline.send(new PutJobSuccessResultCommand(jobParams));
  } catch (err) {
    const jobParams = {
      jobId: jobId,
      failureDetails: {
        message: JSON.stringify(err),
        type: "JobFailed",
        externalExecutionId: context.awsRequestId,
      },
    };
    const jobData = await codepipeline.send(new PutJobFailureResultCommand(jobParams));
    console.log("Error", err);
  }
};
```

Report failure back

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as lambda from '@aws-cdk/aws-lambda';
import * as lambdaNode from '@aws-cdk/aws-lambda-nodejs';
import * as path from 'path’;

// ...

const handler = new lambdaNode.NodejsFunction(this, 'CDKAmplifyPipelineTests', {
  runtime: lambda.Runtime.NODEJS_12_X,
  handler: 'handler',
  entry: path.resolve(__dirname, 'test') + '/index.js',
});

preprodStage.addActions(new codepipeline_actions.LambdaInvokeAction({
  actionName: 'Tests',
  lambda: handler,
  userParameters: {'endpointUrl': preprod.urlOutput},
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as lambda from '@aws-cdk/aws-lambda';
import * as lambdaNode from '@aws-cdk/aws-lambda-nodejs';
import * as path from 'path’;

// ...

const handler = new lambdaNode.NodejsFunction(this, 'CDKAmplifyPipelineTests', {
  runtime: lambda.Runtime.NODEJS_12_X,
  handler: 'handler',
  entry: path.resolve(__dirname, 'test') + '/index.js',
});

preprodStage.addActions(new codepipeline_actions.LambdaInvokeAction({
  actionName: 'Tests',
  lambda: handler,
  userParameters: {'endpointUrl': preprod.urlOutput},
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

Import construct libraries

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as lambda from '@aws-cdk/aws-lambda';
import * as lambdaNode from '@aws-cdk/aws-lambda-nodejs';
import * as path from 'path';

// ...

const handler = new lambdaNode.NodejsFunction(this, 'CDKAmplifyPipelineTests', {
  runtime: lambda.Runtime.NODEJS_12_X,
  handler: 'handler',
  entry: path.resolve(__dirname, 'test') + '/index.js',
});

preprodStage.addActions(new codepipeline_actions.LambdaInvokeAction({
  actionName: 'Tests',
  lambda: handler,
  userParameters: {'endpointUrl': preprod.urlOutput},
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

Create function

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as lambda from '@aws-cdk/aws-lambda';
import * as lambdaNode from '@aws-cdk/aws-lambda-nodejs';
import * as path from 'path';

// ...

const handler = new lambdaNode.NodejsFunction(this, 'CDKAmplifyPipelineTests', {
  runtime: lambda.Runtime.NODEJS_12_X,
  handler: 'handler',
  entry: path.resolve(__dirname, 'test') + '/index.js',
});

preprodStage.addActions(new codepipeline_actions.LambdaInvokeAction({
  actionName: 'Tests',
  lambda: handler,
  userParameters: {'endpointUrl': preprod.urlOutput},
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

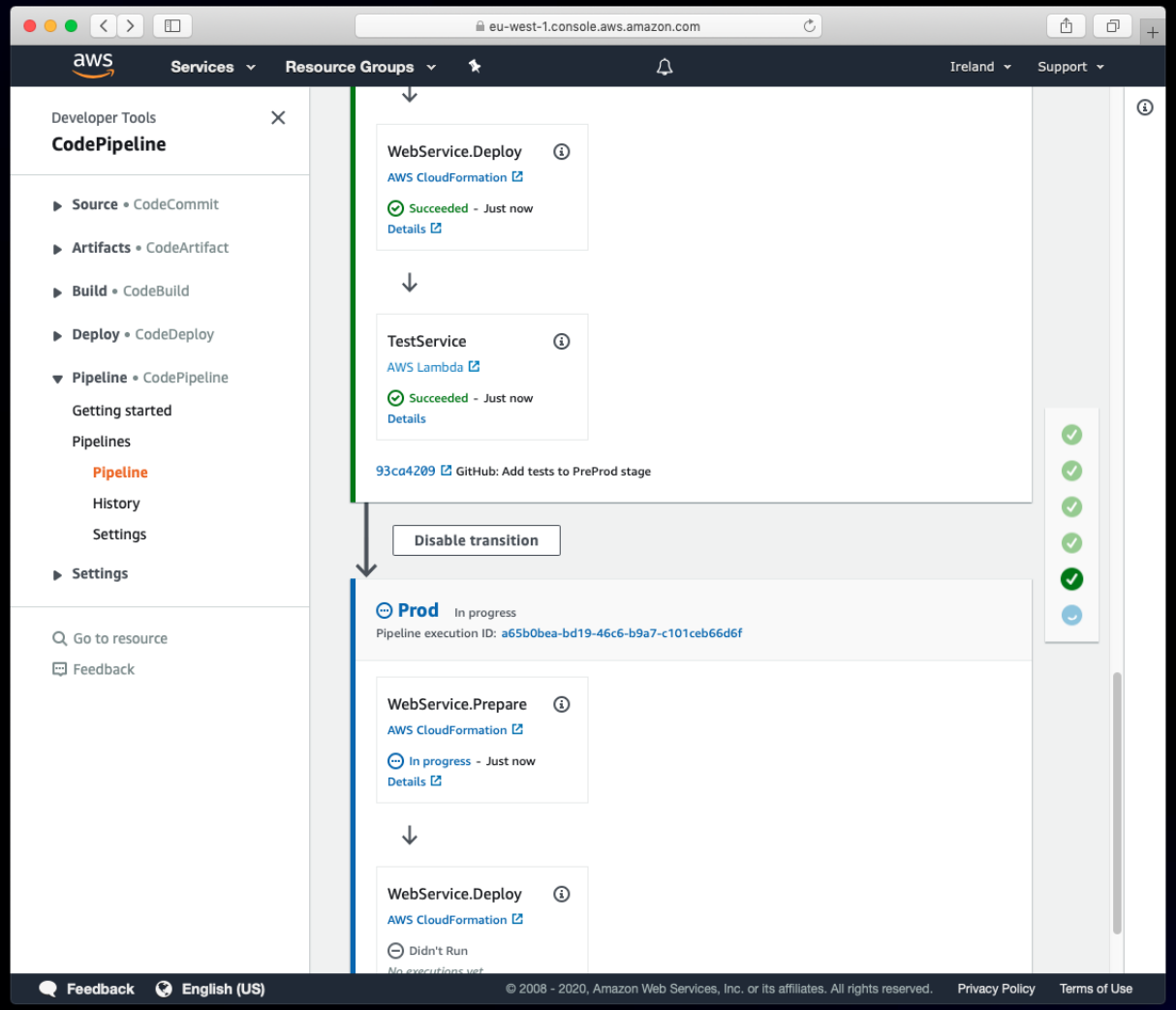
Add Lambda invoke action with parameters and run order defined

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS LAMBDA



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS STEP FUNCTIONS INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as stepfunctions from '@aws-cdk/aws-stepfunctions’;

// ...

const statemachine = new stepfunctions.StateMachine(this, 'CDKAmplifyPipelineTests', {
  definition: testStartState
});

preprodStage.addActions(new codepipeline_actions.StepFunctionInvokeAction({
  actionName: 'Tests',
  stateMachine: statemachine,
  stateMachineInput: codepipeline_actions.StateMachineInput.literal({
    endpointUrl: preprod.urlOutput
  }),
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS STEP FUNCTIONS INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as stepfunctions from '@aws-cdk/aws-stepfunctions’;

// ...

const statemachine = new stepfunctions.StateMachine(this, 'CDKAmplifyPipelineTests', {
  definition: testStartState
});

preprodStage.addActions(new codepipeline_actions.StepFunctionInvokeAction({
  actionName: 'Tests',
  stateMachine: statemachine,
  stateMachineInput: codepipeline_actions.StateMachineInput.literal({
    endpointUrl: preprod.urlOutput
  }),
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

Import construct libraries

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS STEP FUNCTIONS INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as stepfunctions from '@aws-cdk/aws-stepfunctions’;

// ...

const statemachine = new stepfunctions.StateMachine(this, 'CDKAmplifyPipelineTests', {
  definition: testStartState
});

preprodStage.addActions(new codepipeline_actions.StepFunctionInvokeAction({
  actionName: 'Tests',
  stateMachine: statemachine,
  stateMachineInput: codepipeline_actions.StateMachineInput.literal({
    endpointUrl: preprod.urlOutput
  }),
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

Create state machine

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING AWS STEP FUNCTIONS INVOKE ACTION

```
import * as codepipeline_actions from '@aws-cdk/aws-codepipeline-actions';
import * as stepfunctions from '@aws-cdk/aws-stepfunctions’;

// ...

const statemachine = new stepfunctions.StateMachine(this, 'CDKAmplifyPipelineTests', {
  definition: testStartState
});

preprodStage.addActions(new codepipeline_actions.StepFunctionInvokeAction({
  actionName: 'Tests',
  stateMachine: statemachine,
  stateMachineInput: codepipeline_actions.StateMachineInput.literal({
    endpointUrl: preprod.urlOutput
  }),
  runOrder: preprodStage.nextSequentialRunOrder(),
}));
```

Add Step Function invoke action with parameters and run order defined

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
export class LambdaInvokeStep extends Step implements ICodePipelineActionFactory {
  constructor(
    private readonly name: string,
    private readonly functionname: lambda.IFunction,
    private readonly parameters: { [key: string]: any }
  ) {
    super(name);
  }
  public produceAction(
    stage: codepipeline.IStage,
    options: ProduceActionOptions
  ): CodePipelineActionFactoryResult {
    stage.addAction(new codepipeline_actions.LambdaInvokeAction({
      actionName: this.name,
      lambda: this.functionname,
      runOrder: options.runOrder,
      userParameters: this.parameters
    }));
    return { runOrdersConsumed: 1 };
  }
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
export class LambdaInvokeStep extends Step implements ICodePipelineActionFactory {
  constructor(
    private readonly name: string,
    private readonly functionname: lambda.IFunction,
    private readonly parameters: { [key: string]: any }
  ) {
    super(name);
  }
  public produceAction(
    stage: codepipeline.IStage,
    options: ProduceActionOptions
  ): CodePipelineActionFactoryResult {
    stage.addAction(new codepipeline_actions.LambdaInvokeAction({
      actionName: this.name,
      lambda: this.functionname,
      runOrder: options.runOrder,
      userParameters: this.parameters
    }));
    return { runOrdersConsumed: 1 };
  }
}
```

Create class that extends step and uses the action factory

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
export class LambdaInvokeStep extends Step implements ICodePipelineActionFactory {
  constructor(
    private readonly name: string,
    private readonly functionname: lambda.IFunction,
    private readonly parameters: { [key: string]: any }
  ) {
    super(name);
  }
  public produceAction(
    stage: codepipeline.IStage,
    options: ProduceActionOptions
  ): CodePipelineActionFactoryResult {
    stage.addAction(new codepipeline_actions.LambdaInvokeAction({
      actionName: this.name,
      lambda: this.functionname,
      runOrder: options.runOrder,
      userParameters: this.parameters
    }));
    return { runOrdersConsumed: 1 };
  }
}
```

Produce the action for our stage using CodePipelineActionFactory

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
export class LambdaInvokeStep extends Step implements ICodePipelineActionFactory {
  constructor(
    private readonly name: string,
    private readonly functionname: lambda.IFunction,
    private readonly parameters: { [key: string]: any }
  ) {
    super(name);
  }
  public produceAction(
    stage: codepipeline.IStage,
    options: ProduceActionOptions
  ): CodePipelineActionFactoryResult {
    stage.addAction(new codepipeline_actions.LambdaInvokeAction({
      actionName: this.name,
      lambda: this.functionname,
      runOrder: options.runOrder,
      userParameters: this.parameters
    }));
    return { runOrdersConsumed: 1 };
  }
}
```

Add CodePipeline's LambdaInvokeAction with parameters

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
import { LambdaInvokeStep } from ./lambda-invoke-step';

// ...

const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
        new LambdaInvokeStep('IntegrationTest', handler, {
          'endpointUrl': 'urlOutput’
        })
    }),
  ],
});
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
import { LambdaInvokeStep } from ./lambda-invoke-step';

// ...


const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
        new LambdaInvokeStep('IntegrationTest', handler, {
          'endpointUrl': 'urlOutput'
        })
    }),
  ],
});
```

Import the LambdaInvokeStep

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration tests

USING CUSTOM INVOKE STEP

```
import { LambdaInvokeStep } from ./lambda-invoke-step';

// ...

const preprod = new CdkpipelinesDemoStage(this, 'PreProd', {
  env: { account: 'ACCOUNT', region: 'REGION' }
});
const preprodStage = pipeline.addStage(preprod, {
  post: [
        new LambdaInvokeStep('IntegrationTest', handler, {
          'endpointUrl': 'urlOutput’
        })
    }),
  ],
});
```

Adds our LambdaInvokeStep to the stage

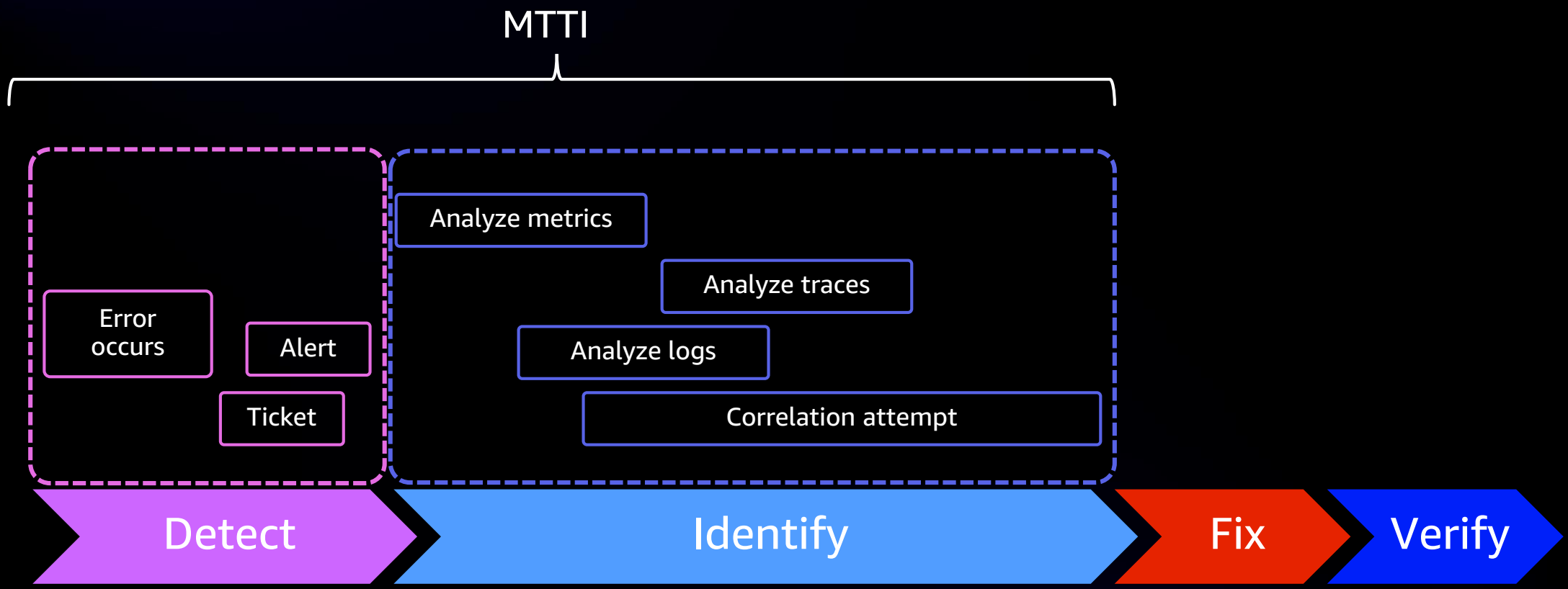
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability



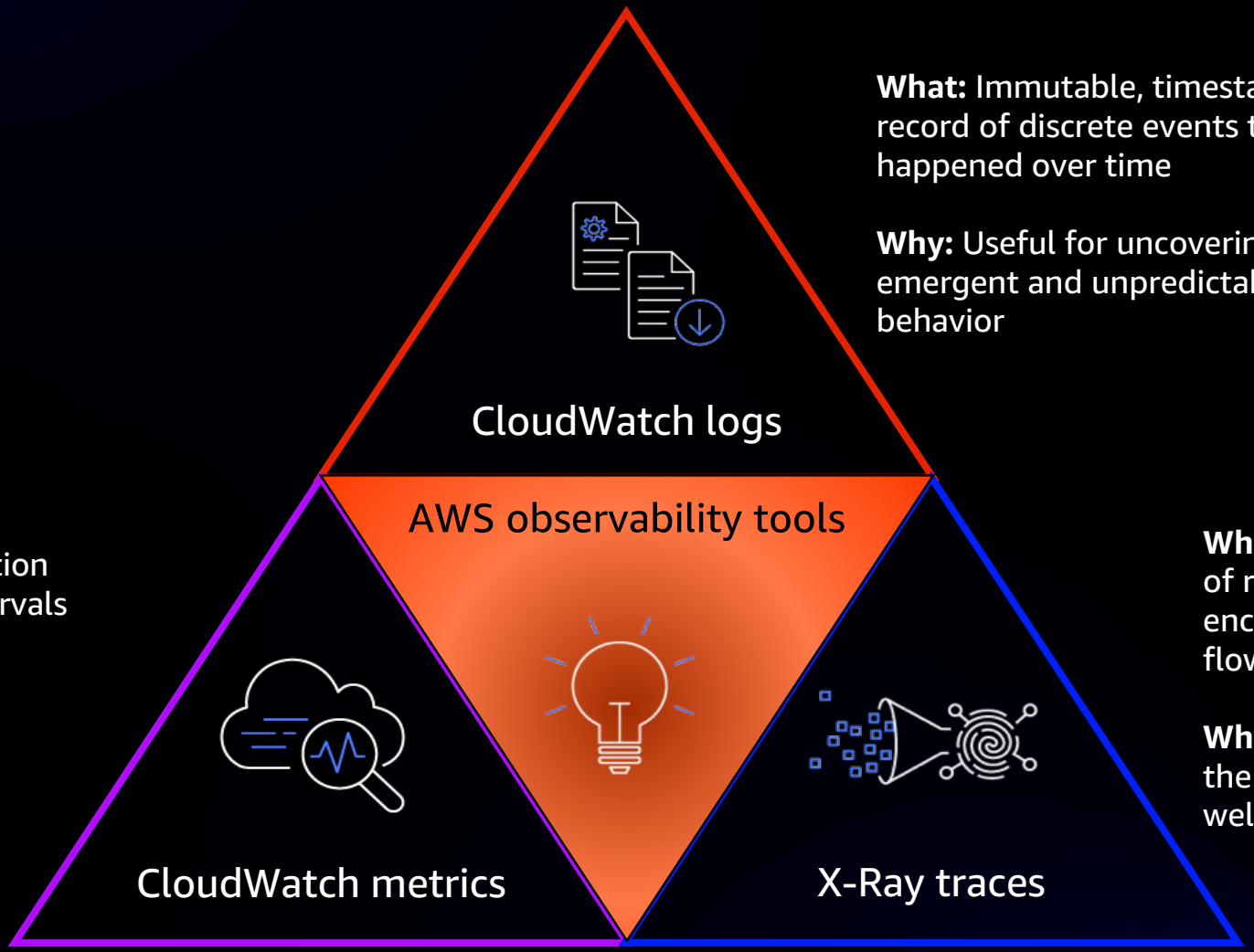
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Typical troubleshooting workflow



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS observability tools



What: Immutable, timestamped record of discrete events that happened over time

Why: Useful for uncovering emergent and unpredictable behavior

What: Numeric representation of data measured over intervals of time

Why: Useful for identifying trends, mathematical modeling, and prediction

What: Representation of a series of related distributed events that encode the end-to-end request flow through a distributed system

Why: Provides visibility into both the path traversed by a request as well as the structure of a request

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stages of the software release process



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stages of the software release process



Github

AWS CodeBuild

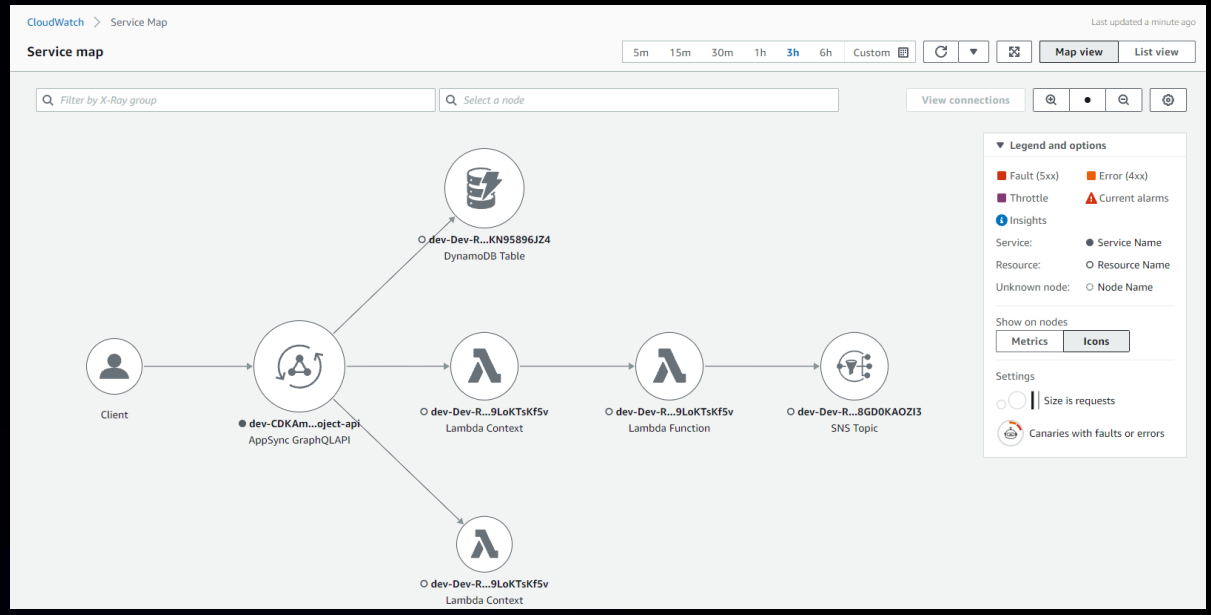
Staging environment
CloudFormation tests
Integration tests
Manual approval

Production environment
Observability

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

- AWS X-Ray traces user requests as they travel through your entire application
- Aggregates the data gathered from individual services in your application into a single unit
- Allows you to drill into traces to better understand performance issues impacting your application and end users



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS APPSYNC

```
const appsync_api = new cdk_appsync_transformer.AppSyncTransformer(
  this,
  `${props.stage}-CDKAmplifyProject`,
  {
    schemaPath: "graphql/schema.graphql",
    authorizationConfig: {
      defaultAuthorization: {
        authorizationType: appsync.AuthorizationType.USER_POOL,
        userPoolConfig: {
          userPool: userPool,
          appIdClientRegex: userPoolClient.userPoolClientId,
          defaultAction: appsync.UserPoolDefaultAction.ALLOW,
        },
      },
    },
    xrayEnabled: true,
  }
);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS APPSYNC

```
const appsync_api = new cdk_appsync_transformer.AppSyncTransformer(
  this,
  `${props.stage}-CDKAmplifyProject`,
  {
    schemaPath: "graphql/schema.graphql",
    authorizationConfig: {
      defaultAuthorization: {
        authorizationType: appsync.AuthorizationType.USER_POOL,
        userPoolConfig: {
          userPool: userPool,
          appIdClientRegex: userPoolClient.userPoolClientId,
          defaultAction: appsync.UserPoolDefaultAction.ALLOW,
        },
      },
    },
    xrayEnabled: true,
  }
);
```

A flag representing whether X-Ray tracing is enabled for this GraphqlApi

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const notifyShoppingDoneFunction = new lambdaNode.NodejsFunction(
  this,
  `${props.stage}-NotifyShoppingDoneFunction`,
  {
    runtime: lambda.Runtime.NODEJS_14_X,
    handler: "handler",
    entry: path.resolve(__dirname, "functions") + "/notifyShoppingDone.js",
    environment: {
      TOPIC: purchaseTopic.topicArn,
    },
    bundling: {
      nodeModules: ["aws-xray-sdk"],
    },
    tracing: Tracing.ACTIVE,
  }
);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const notifyShoppingDoneFunction = new lambdaNode.NodejsFunction(
  this,
  `${props.stage}-NotifyShoppingDoneFunction`,
  {
    runtime: lambda.Runtime.NODEJS_14_X,
    handler: "handler",
    entry: path.resolve(__dirname, "functions") + "/notifyShoppingDone.js",
    environment: {
      TOPIC: purchaseTopic.topicArn,
    },
    bundling: {
      nodeModules: ["aws-xray-sdk"],
    },
    tracing: Tracing.ACTIVE,
  }
);
```

A flag representing whether X-Ray tracing is enabled for this GraphqlApi

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const notifyShoppingDoneFunction = new lambdaNode.NodejsFunction(
  this,
  `${props.stage}-NotifyShoppingDoneFunction`,
  {
    runtime: lambda.Runtime.NODEJS_14_X,
    handler: "handler",
    entry: path.resolve(__dirname, "functions") + "/notifyShoppingDone.js",
    environment: {
      TOPIC: purchaseTopic.topicArn,
    },
    bundling: {
      nodeModules: ["aws-xray-sdk"],
    },
    tracing: Tracing.ACTIVE,
  }
);
```

Install aws-xray-sdk module

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const AWSXRay = require('aws-xray-sdk');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
var sns = new AWS.SNS();
const TOPIC = process.env.TOPIC;
exports.handler = async function (event) {
    const email = event.identity.claims.email;
    console.log(email);
    const params = {
        Message: `${email} just finished doing the shopping`,
        TopicArn: TOPIC
      };
    try {
        const publishText = await sns.publish(params).promise();
        console.log(publishText)
        return true;
    } catch (e) {
        console.log(e);
        return false;
    }
  };
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const AWSXRay = require('aws-xray-sdk');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
var sns = new AWS.SNS();
const TOPIC = process.env.TOPIC;
exports.handler = async function (event) {
    const email = event.identity.claims.email;
    console.log(email);
    const params = {
        Message: `${email} just finished doing the shopping`,
        TopicArn: TOPIC
      };
    try {
        const publishText = await sns.publish(params).promise();
        console.log(publishText)
        return true;
    } catch (e) {
        console.log(e);
        return false;
    }
  };
```

Load aws-xray-sdk module

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const AWSXRay = require('aws-xray-sdk');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
var sns = new AWS.SNS();
const TOPIC = process.env.TOPIC;
exports.handler = async function (event) {
    const email = event.identity.claims.email;
    console.log(email);
    const params = {
        Message: `${email} just finished doing the shopping`,
        TopicArn: TOPIC
      };
    try {
        const publishText = await sns.publish(params).promise();
        console.log(publishText)
        return true;
    } catch (e) {
        console.log(e);
        return false;
    }
  };
```

Wrap the aws-sdk library with the captureAWS method

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLING FOR AWS LAMBDA

```
const AWSXRay = require('aws-xray-sdk');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
var sns = new AWS.SNS();
const TOPIC = process.env.TOPIC;
exports.handler = async function (event) {
    const ema
    console.l
    const par
        Messa
        Topic
      };
    try {
        const publishText = await sns.publish(params).promise();
        console.log(publishText)
        return true;
    } catch (e) {
        console.log(e);
        return false;
    }
  };
```

Language specific instructions:
https://docs.aws.amazon.com/lambda/latest/dg/services-xray.html

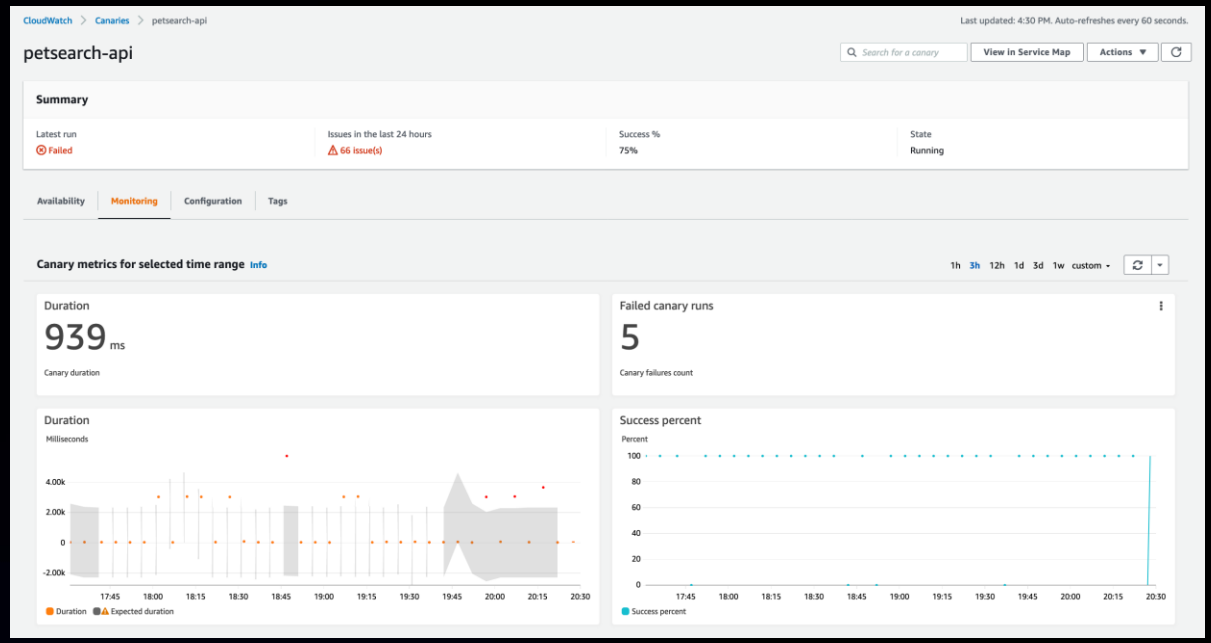


aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

- Configurable scripts (Canaries) that run on a schedule, to monitor your endpoints and APIs
- Check the availability and latency of your endpoints store load time data and screenshots of the UI actions
- Easily test APIs, check broken links, perform UI automation & heart beat monitoring



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```

Create a new canary

aws

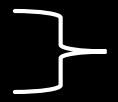
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```



Specify the schedule on which a canary runs

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```

Configure the canary script bundling

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```

Includes a custom runtime provided by Synthetics

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

CREATE CANARY

```
import synthetics = require('@aws-cdk/aws-synthetics');

const canary = new synthetics.Canary(this, `${props.stage}-Canary`, {
  schedule: synthetics.Schedule.rate(cdk.Duration.minutes(5)),
  test: synthetics.Test.custom({
    code: synthetics.Code.fromAsset(path.join(__dirname, 'canary')),
    handler: 'index.handler',
  }),
  runtime: synthetics.Runtime.SYNTHETICS_NODEJS_PUPPETEER_3_2,
  environmentVariables: {
    url: canaryUrl,
  },
});
```

Configure environment variables

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

AWS LAMBDA FUNCTION CANARY CODE

```
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {
  const url = process.env.url;
  const page = await synthetics.getPage();
  const response = await page.goto(url, {
    waitUntil: 'domcontentloaded’,
    timeout: 30000 });
  await page.waitFor(15000);
  await synthetics.takeScreenshot('loaded', 'loaded');
  const pageTitle = await page.title();
  log.info('Page title: ' + pageTitle);
  if (response.status() !== 200) {
    throw 'Failed to load page!';
  }
};

exports.handler = async () => {
  return await pageLoadBlueprint();
};
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

AWS LAMBDA FUNCTION CANARY CODE

```
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {
  const url = process.env.url;
  const page = await synthetics.getPage();
  const response = await page.goto(url, {
    waitUntil: 'domcontentloaded’,
    timeout: 30000 });
  await page.waitFor(15000);
  await synthetics.takeScreenshot('loaded', 'loaded');
  const pageTitle = await page.title();
  log.info('Page title: ' + pageTitle);
  if (response.status() !== 200) {
    throw 'Failed to load page!';
  }
};

exports.handler = async () => {
  return await pageLoadBlueprint();
};
```

Uses CloudWatch Synthetics modules

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

AWS LAMBDA FUNCTION CANARY CODE

```
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {
  const url = process.env.url;
  const page = await synthetics.getPage();
  const response = await page.goto(url, {
    waitUntil: 'domcontentloaded’,
    timeout: 30000 });
  await page.waitFor(15000);
  await synthetics.takeScreenshot('loaded', 'loaded');
  const pageTitle = await page.title();
  log.info('Page title: ' + pageTitle);
  if (response.status() !== 200) {
    throw 'Failed to load page!';
  }
};

exports.handler = async () => {
  return await pageLoadBlueprint();
};
```

Gets the page definied in the environment variable

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

AWS LAMBDA FUNCTION CANARY CODE

```
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {
  const url = process.env.url;
  const page = await synthetics.getPage();
  const response = await page.goto(url, {
    waitUntil: 'domcontentloaded’,
    timeout: 30000 });
  await page.waitFor(15000);
  await synthetics.takeScreenshot('loaded', 'loaded');
  const pageTitle = await page.title();
  log.info('Page title: ' + pageTitle);
  if (response.status() !== 200) {
    throw 'Failed to load page!';
  }
};

exports.handler = async () => {
  return await pageLoadBlueprint();
};
```

Takes a screenshot and logs page title

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

AWS LAMBDA FUNCTION CANARY CODE

```
const synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {
  const url = process.env.url;
  const page = await synthetics.getPage();
  const response = await page.goto(url, {
    waitUntil: 'domcontentloaded’,
    timeout: 30000 });
  await page.waitFor(15000);
  await synthetics.takeScreenshot('loaded', 'loaded');
  const pageTitle = await page.title();
  log.info('Page title: ' + pageTitle);
  if (response.status() !== 200) {
    throw 'Failed to load page!';
  }
};

exports.handler = async () => {
  return await pageLoadBlueprint();
};
```

Throws an error if anything but 200 response

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

ADD ALARM

```
import * as cloudwatch from "@aws-cdk/aws-cloudwatch";

new cloudwatch.Alarm(this, `${props.stage}-CanaryAlarm`, {
  metric: canary.metricSuccessPercent(),
  evaluationPeriods: 2,
  threshold: 90,
  comparisonOperator: cloudwatch.ComparisonOperator.LESS_THAN_THRESHOLD,
});
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

ADD ALARM

```
import * as cloudwatch from "@aws-cdk/aws-cloudwatch";

new cloudwatch.Alarm(this, `${props.stage}-CanaryAlarm`, {
  metric: canary.metricSuccessPercent(),
  evaluationPeriods: 2,
  threshold: 90,
  comparisonOperator: cloudwatch.ComparisonOperator.LESS_THAN_THRESHOLD,
});
```

Create a new CloudWatch alarm

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch Synthetics

ADD ALARM

```
import * as cloudwatch from "@aws-cdk/aws-cloudwatch";

new cloudwatch.Alarm(this, `${props.stage}-CanaryAlarm`, {
  metric: canary.metricSuccessPercent(),
  evaluationPeriods: 2,
  threshold: 90,
  comparisonOperator: cloudwatch.ComparisonOperator.LESS_THAN_THRESHOLD,
});
```

Track the canary metrics and set a comparison against the threshold

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

CREATE DASHBOARD AND ADD ALARM WIDGETS

```
import cloudwatch = require("@aws-cdk/aws-cloudwatch");
import { AlarmWidget, AlarmStatusWidget } from "@aws-cdk/aws-cloudwatch";

const dashboard = new cloudwatch.Dashboard(this, `${props.stage}-Dashboard`);

dashboard.addWidgets(
  new AlarmStatusWidget({
    alarms: [canaryAlarm],
  })
);

dashboard.addWidgets(
  new AlarmWidget({
    title: "Errors",
    alarm: canaryAlarm,
  })
);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

CREATE DASHBOARD AND ADD ALARM WIDGET

```
import cloudwatch = require("@aws-cdk/aws-cloudwatch");
import { AlarmWidget, AlarmStatusWidget } from "@aws-cdk/aws-cloudwatch";

const dashboard = new cloudwatch.Dashboard(this, `${props.stage}-Dashboard`);

dashboard.addWidgets(
  new AlarmStatusWidget({
    alarms: [canaryAlarm],
  })
);

dashboard.addWidgets(
  new AlarmWidget({
    title: "Errors",
    alarm: canaryAlarm,
  })
);
```

Load CloudWatch modules

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

CREATE DASHBOARD AND ADD ALARM WIDGET

```
import cloudwatch = require("@aws-cdk/aws-cloudwatch");
import { AlarmWidget, AlarmStatusWidget } from "@aws-cdk/aws-cloudwatch";

const dashboard = new cloudwatch.Dashboard(this, `${props.stage}-Dashboard`);

dashboard.addWidgets(
  new AlarmStatusWidget({
    alarms: [canaryAlarm],
  })
);

dashboard.addWidgets(
  new AlarmWidget({
    title: "Errors",
    alarm: canaryAlarm,
  })
);
```

Create dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

CREATE DASHBOARD AND ADD ALARM WIDGET

```
import cloudwatch = require("@aws-cdk/aws-cloudwatch");
import { AlarmWidget, AlarmStatusWidget } from "@aws-cdk/aws-cloudwatch";

const dashboard = new cloudwatch.Dashboard(this, `${props.stage}-Dashboard`);

dashboard.addWidgets(
  new AlarmStatusWidget({
    alarms: [canaryAlarm],
  })
);

dashboard.addWidgets(
  new AlarmWidget({
    title: "Errors",
    alarm: canaryAlarm,
  })
);
```

Add the canary alarm status as a widget to dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

CREATE DASHBOARD AND ADD ALARM WIDGET

```
import cloudwatch = require("@aws-cdk/aws-cloudwatch");
import { AlarmWidget, AlarmStatusWidget } from "@aws-cdk/aws-cloudwatch";

const dashboard = new cloudwatch.Dashboard(this, `${props.stage}-Dashboard`);

dashboard.addWidgets(
  new AlarmStatusWidget({
    alarms: [canaryAlarm],
  })
);

dashboard.addWidgets(
  new AlarmWidget({
    title: "Errors",
    alarm: canaryAlarm,
  })
);
```

Add the canary alarm graph as a widget to dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

ADD AMPLIFY HOSTING GRAPH WIDGETS

```
import cloudwatch = require('@aws-cdk/aws-cloudwatch');
import { GraphWidget, Metric } from "@aws-cdk/aws-cloudwatch";

dashboard.addWidgets(
  new GraphWidget({
    title: "Requests (sum)",
    width: 12,
    left: [
      new Metric({
        namespace: "AWS/AmplifyHosting",
        metricName: "Requests",
        dimensionsMap: {
          App: amplifyApp,
        },
        statistic: "sum",
        label: "Count",
        period: cdk.Duration.minutes(1),
      }),
    ],
  })
);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

ADD AMPLIFY HOSTING GRAPH WIDGETS

```
import cloudwatch = require('@aws-cdk/aws-cloudwatch');
import { GraphWidget, Metric } from "@aws-cdk/aws-cloudwatch";

dashboard.addWidgets(
  new GraphWidget({
    title: "Requests (sum)",
    width: 12,
    left: [
      new Metric({
        namespace: "AWS/AmplifyHosting",
        metricName: "Requests",
        dimensionsMap: {
          App: amplifyApp,
        },
        statistic: "sum",
        label: "Count",
        period: cdk.Duration.minutes(1),
      }),
    ],
  })
);
```

Create a new graph widget

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

ADD AMPLIFY HOSTING GRAPH WIDGETS

```
import cloudwatch = require('@aws-cdk/aws-cloudwatch');
import { GraphWidget, Metric } from "@aws-cdk/aws-cloudwatch";

dashboard.addWidgets(
  new GraphWidget({
    title: "Requests (sum)",
    width: 12,
    left: [
      new Metric({
        namespace: "AWS/AmplifyHosting",
        metricName: "Requests",
        dimensionsMap: {
          App: amplifyApp,
        },
        statistic: "sum",
        label: "Count",
        period: cdk.Duration.minutes(1),
      }),
    ],
  })
);
```

Define namespace, metric, and which app

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

ADD AMPLIFY HOSTING GRAPH WIDGETS

```
import cloudwatch = require('@aws-cdk/aws-cloudwatch');
import { GraphWidget, Metric } from "@aws-cdk/aws-cloudwatch";

dashboard.addWidgets(
  new GraphWidget({
    title: "4xx errors (sum)",
    width: 12,
    left: [
      new Metric({
        namespace: "AWS/AmplifyHosting",
        metricName: "4xxErrors",
        dimensionsMap: {
          App: amplifyApp,
        },
        statistic: "sum",
        label: "Count",
        period: cdk.Duration.minutes(1),
      }),
    ],
  })
);
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch dashboard

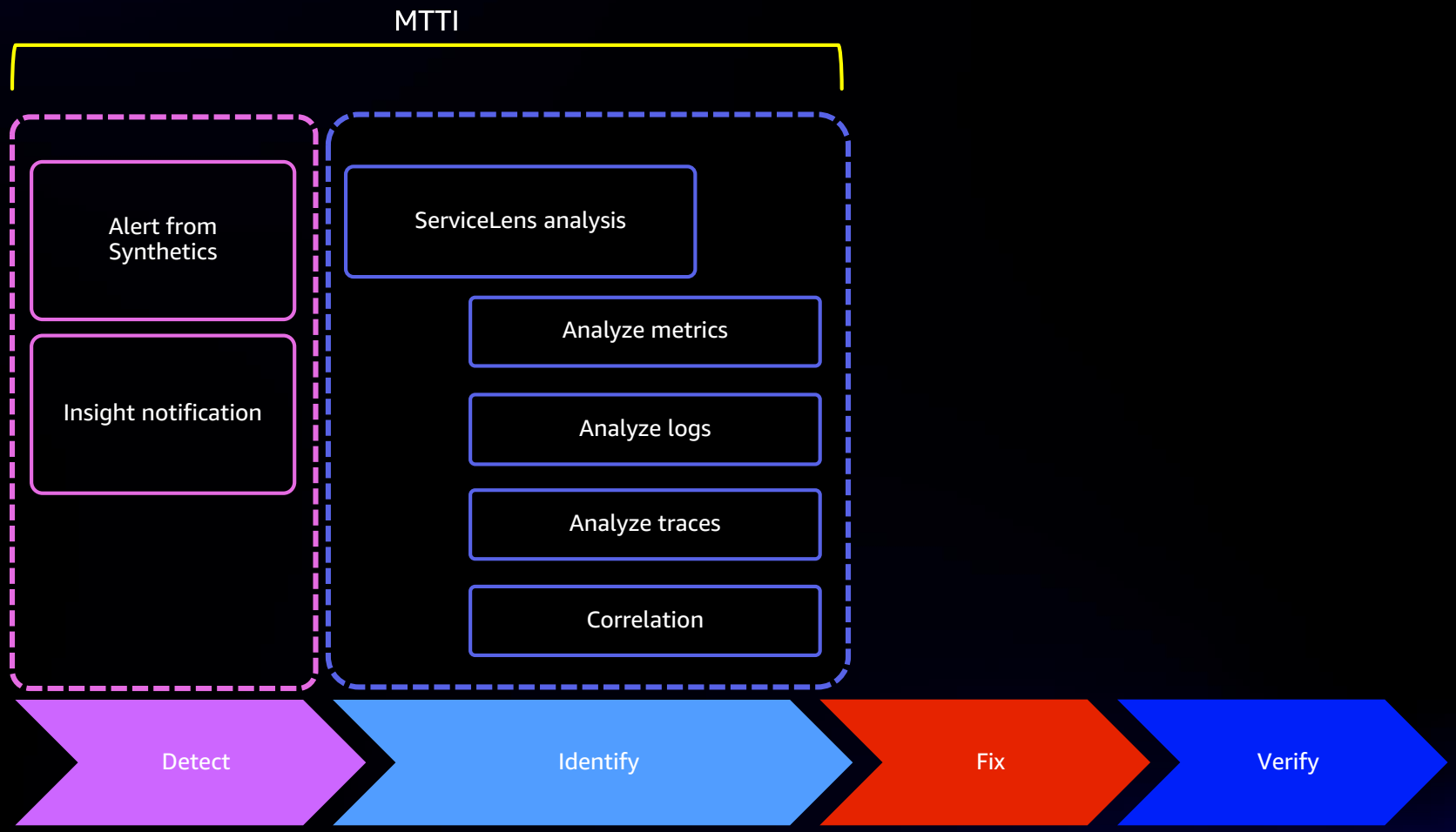
ADD AMPLIFY HOSTING GRAPH WIDGETS

```
import cloudwatch = require('@aws-cdk/aws-cloudwatch');
import { GraphWidget, Metric } from "@aws-cdk/aws-cloudwatch";

dashboard.addWidgets(
  new GraphWidget({
    title: "5xx errors (sum)",
    width: 12,
    left: [
      new Metric({
        namespace: "AWS/AppSync",
        metricName: "5xxErrors",
        dimensionsMap: {
          App: amplifyApp,
        },
        statistic: "sum",
        label: "Count",
        period: cdk.Duration.minutes(1),
      }),
    ],
  })
);
```

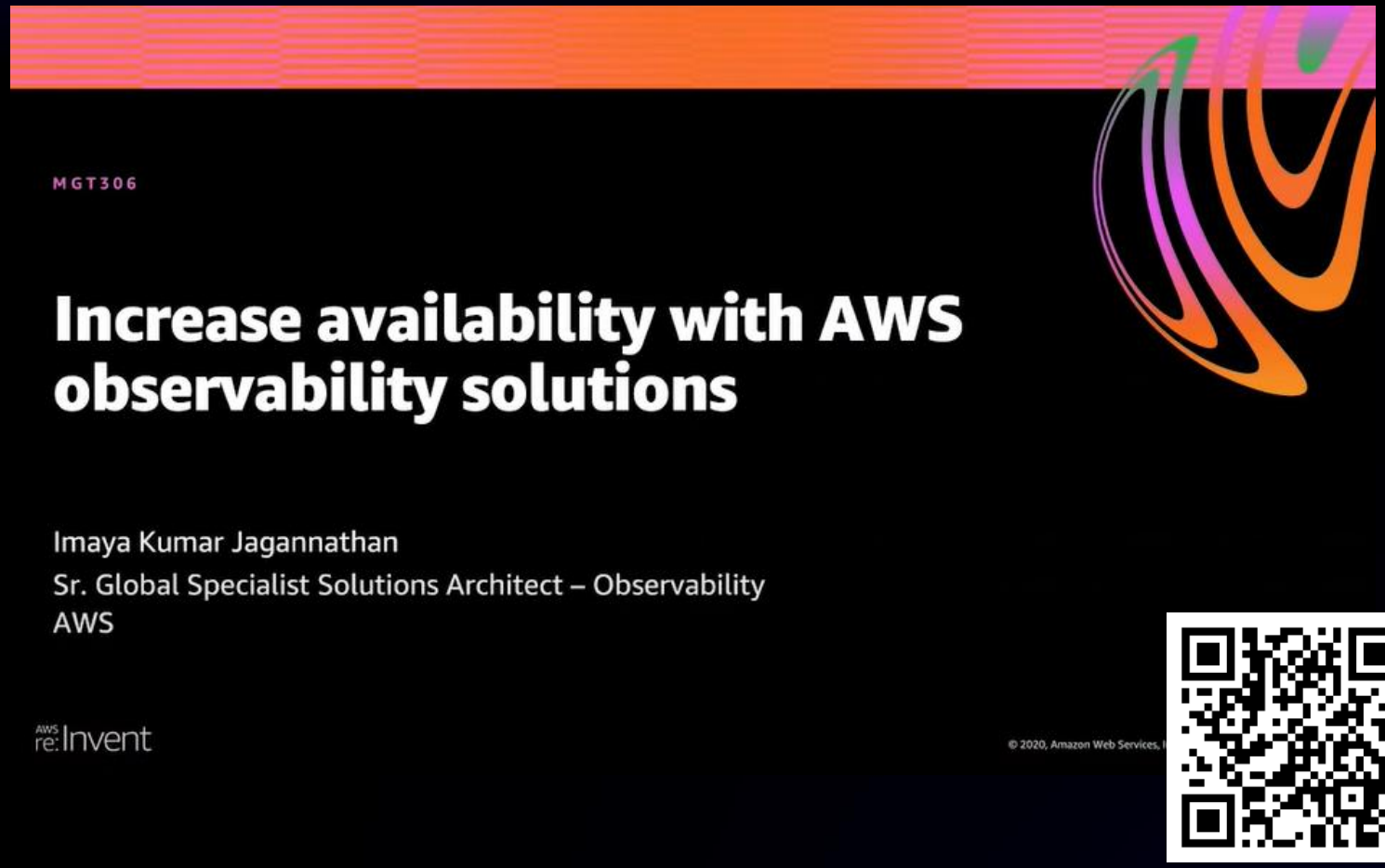
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New troubleshooting workflow



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability on AWS



https://youtu.be/_d_9xCfVBTM

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What we learned in this part

- Add ShellStep in CDK pipelines
- Create actions in stages
- Use AWS Lambda invocations for integration tests
- Enable AWS X-Ray traces with CDK
- Setup Amazon CloudWatch Synthetics canaries
- Create CloudWatch alarms
- Setup a dashboard with widgets

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Additional resources



blog.marcia.dev/amplify-cdk

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Gunnar Grosch
@gunnargrosch

Marcia Villalba
@mavi888uy

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.