

Infrastructure as Code

AWS Solutions Best Practices

Richard Boyd, Sr. Developer Advocate, AWS Code Services

Luis Colon, Sr. Developer Advocate, AWS CloudFormation

April 20, 2020

Agenda

Infrastructure as Code (IaC): The Basics

IaC with AWS CloudFormation

Best Practices

AWS Cloud Development Kit (CDK)

Other IaC Tools

Infrastructure as Code: The Basics

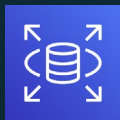


The Basics

Managing cloud applications involves managing the lifecycle of its resources:

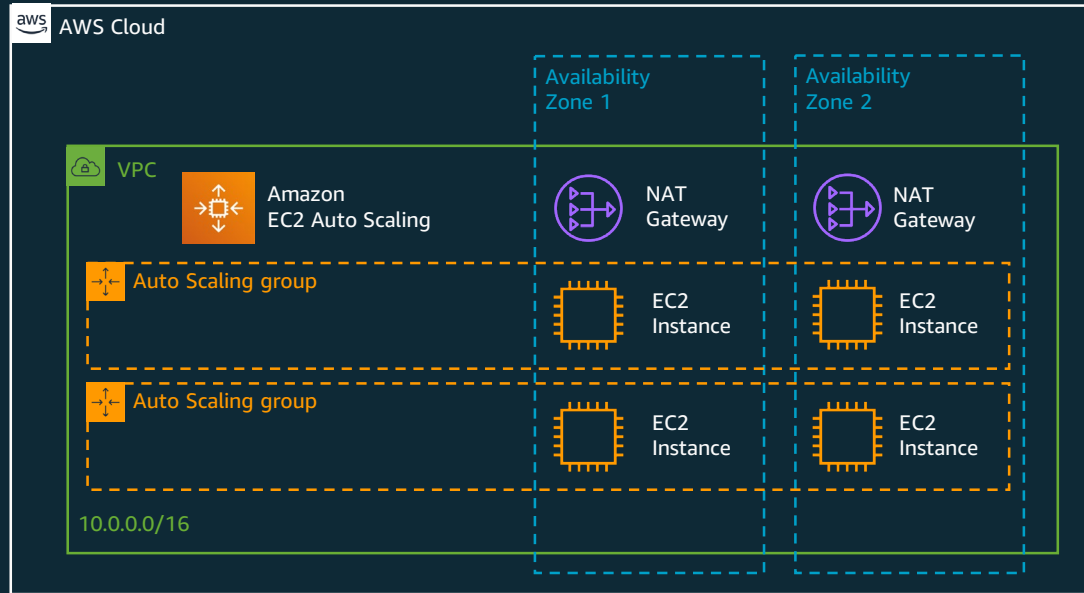
The Basics

Managing cloud applications involves managing the lifecycle of its resources:



The Basics

Managing cloud applications involves managing the lifecycle of its resources:



Resources

The building blocks, or components of cloud applications

Resources

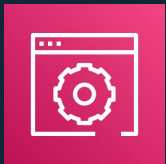
The building blocks, or components of cloud applications

Several options to manage your resource's lifecycle:

Resources

The building blocks, or components of cloud applications

Several options to manage your resource's lifecycle:

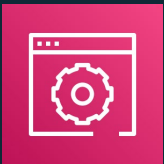


AWS
Management
Console

Resources

The building blocks, or components of cloud applications

Several options to manage your resource's lifecycle:



AWS
Management
Console

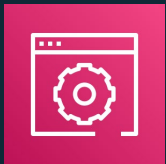


AWS
Command
Line Interface

Resources

The building blocks, or components of cloud applications

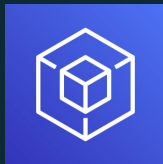
Several options to manage your resource's lifecycle:



**AWS
Management
Console**



**AWS
Command
Line Interface**

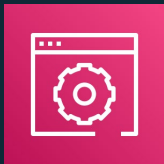


**AWS Tools &
Software
Development
Kits**

Resources

The building blocks, or components of cloud applications

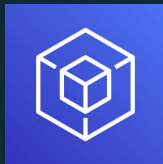
Several options to manage your resource's lifecycle:



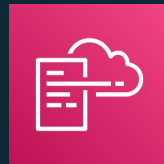
**AWS
Management
Console**



**AWS
Command
Line Interface**



**AWS Tools &
Software
Development
Kits**



**AWS
CloudFormation**

Infrastructure as Code With AWS CloudFormation



Infrastructure as Code

AWSTemplateFormatVersion: "2010-09-09"

Description: A CodeCommit Repo and Cloud9 Environment

Resources:

MyRepo:

Type: "AWS::CodeCommit::Repository"

Properties:

RepositoryName: MyRepo

RepositoryDescription: Sample Repository for Demo

MyC9Environment:

Type: "AWS::Cloud9::EnvironmentEC2"

Properties:

Repositories:

- **PathComponent:** /cfn

RepositoryUrl: !GetAtt MyRepo.CloneUrlHttp

InstanceType: t2.micro

AWS CloudFormation

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls

AWS CloudFormation

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls



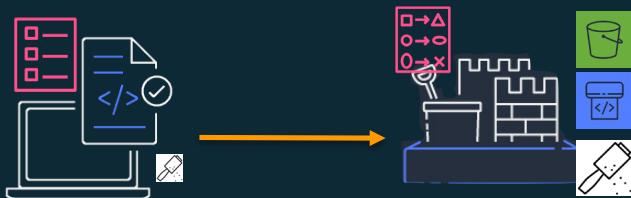
1

Code your
template

AWS CloudFormation

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls



1

Code your
template

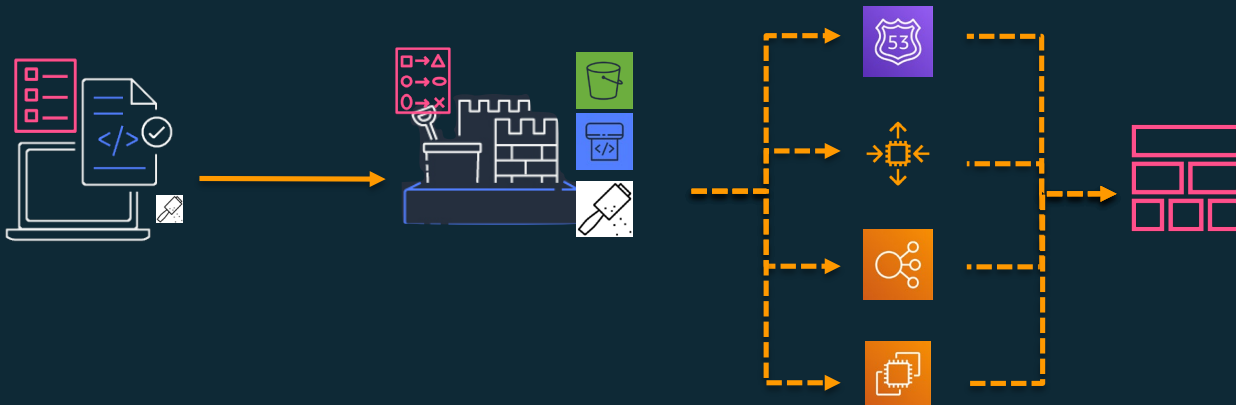
2

Upload, test,
review changes

AWS CloudFormation

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls



1 Code your template

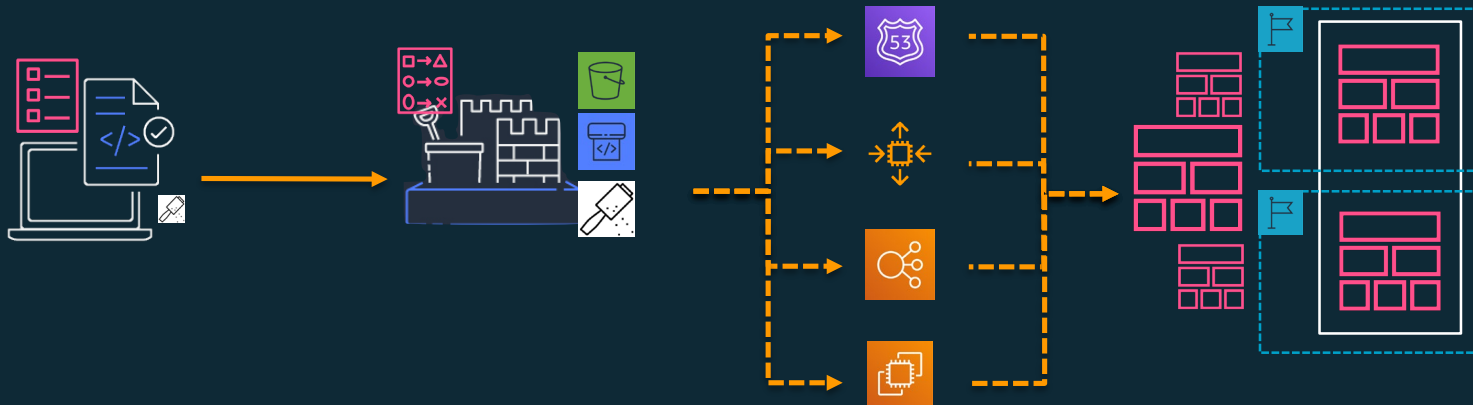
2 Upload, test, review changes

3 A stack is created by executing the changes

AWS CloudFormation

The code template describes the intended state of your resources

CloudFormation translates the intention to API calls



1

Code your template

2

Upload, test, review changes

3

A stack is created by executing the changes

4

Manage many stacks and stack sets over time

AWS CloudFormation Best Practices

Best Practices: Core Artifacts

Best Practices: Core Artifacts



Template

Describes resources,
attributes, dependencies
and their intended state

Best Practices: Core Artifacts



Template

Describes resources, attributes, dependencies and their intended state



Change Set

Describes an execution plan to implement the intended state of a stack's resources

Best Practices: Core Artifacts



Template

Describes resources, attributes, dependencies and their intended state



Change Set

Describes an execution plan to implement the intended state of a stack's resources



Stack

A group of resources and their intended states

Best Practices: Core Artifacts



Template

Describes resources, attributes, dependencies and their intended state



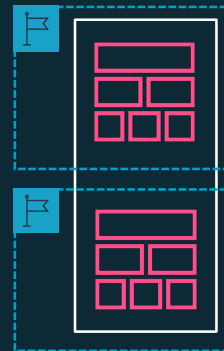
Change Set

Describes an execution plan to implement the intended state of a stack's resources



Stack

A group of resources and their intended states



StackSet

A group of stack instances across accounts and regions

Templates



AWSTemplateFormatVersion: "2010-09-09"

Description: A CodeCommit Repo and Cloud9 Environment

Resources:

MyRepo:

Type: "AWS::CodeCommit::Repository"

Properties:

RepositoryName: MyRepo

RepositoryDescription: Sample Repository for Demo

Once a repo is created, tie the Cloud9 EC2 Instance to

the repository automatically

MyC9Environment:

Type: "AWS::Cloud9::EnvironmentEC2"

Properties:

Repositories:

- **PathComponent:** /cfn

RepositoryUrl: !GetAtt MyRepo.CloneUrlHttp

InstanceType: t2.micro

Templates



- Over 500 types of resources, or create your own
- SAM, Macros, Includes
- YAML/JSON
- YAML comments
- cfn-flip
- Smaller templates

AWSTemplateFormatVersion: "2010-09-09"

Description: A CodeCommit Repo and Cloud9 Environment

Resources:

MyRepo:

Type: "AWS::CodeCommit::Repository"

Properties:

RepositoryName: MyRepo

RepositoryDescription: Sample Repository for Demo

Once a repo is created, tie the Cloud9 EC2 Instance to

the repository automatically

MyC9Environment:

Type: "AWS::Cloud9::EnvironmentEC2"

Properties:

Repositories:

- **PathComponent:** /cfn

RepositoryUrl: !GetAtt MyRepo.CloneUrlHttp

InstanceType: t2.micro

Templates



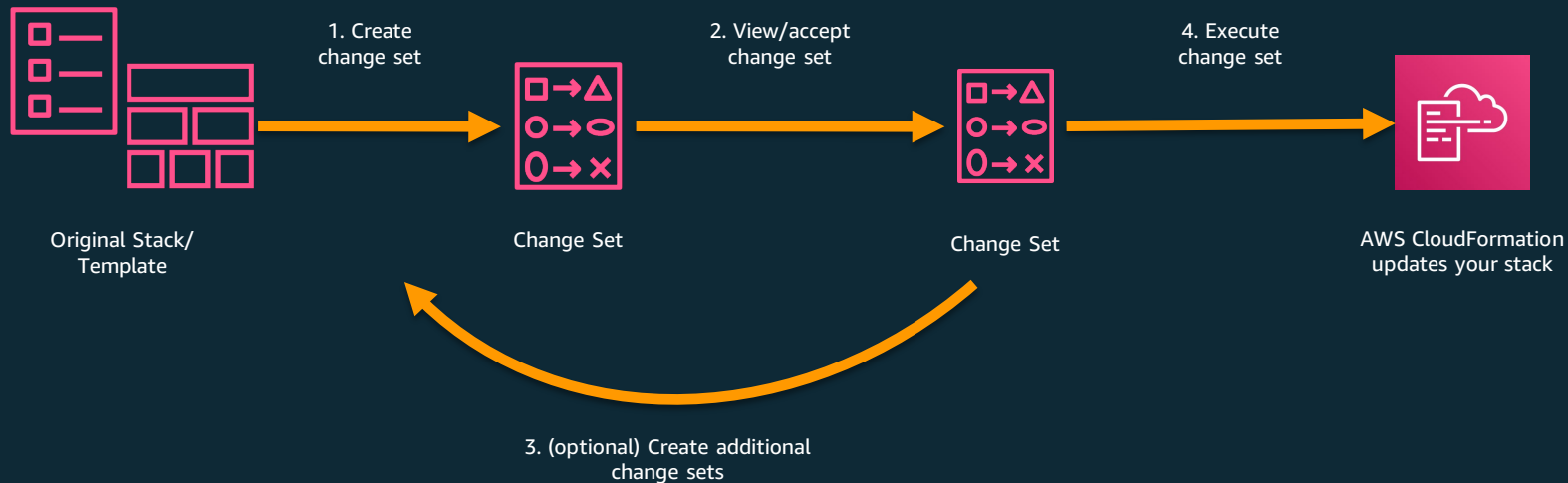
```
1  AWSTemplateFormatVersion: "2010-09-09"
2  Description: A sample template
3  • Errors:
4    Catch: Missing
5  Parameters:
6  • myParam:
7    Type: String
8    Default: String
9    Description: String
10 Resources:
11   ## Missing Properties
12  • MyEC2Instance1:
13    Type: "AWS::EC2::Instance1"
14    ## Fake Properties Key on main level
15    ## Bad sub properties in BlockDeviceMappings/Ebs and NetworkInterfaces
16  MyEC2Instance:
17    Type: "AWS::EC2::Instance"
18  • Properties:
19    ImageId: "ami-2f726546"
20    InstanceType: t1.micro
21    KeyName: 1
22    FakeKey: MadeYouLook
23    BlockDeviceMappings:
24    -
```

Templates

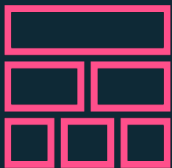


```
1  AWSTemplateFormatVersion: "2010-09-09"
2  Description: A sample template
3  • Errors:
4    Catch: Missing
5  Parameters:
6  • myParam:
7    Type: String
8    Default: String
9    Description: String
10 Resources:
11   ## Missing Properties
12  • MyEC2Instance1:
13    Type: "AWS::EC2::Instance1"
14    ## Fake Properties Key on main level
15    ## Bad sub properties in BlockDeviceMappings/Ebs and NetworkInterfaces
16  MyEC2Instance:
17    Type: "AWS::EC2::Instance"
18  •
19  Severity      Provider      Description                                     Line
20  Warning      Cfn-Lint      Top level item Errors isn't valid             3:1
21  Warning      Cfn-Lint      Parameter myParam not used                     6:1
22  Warning      Cfn-Lint      Invalid Type AWS::EC2::Instance1 for resource MyEC2Instance1 12:1
23  Warning      Cfn-Lint      Properties not defined for resource MyEC2Instance1 12:1
24  Warning      Cfn-Lint      Invalid Property FakeKey for resource MyEC2Instance 18:1
25  Warning      Cfn-Lint      Invalid Property BadSubX2Key for resource MyEC2Instance 26:1
```

Change Sets



Stacks



Layer your application
Extract configuration

Resources:

MyRDSDB:

Type: "AWS::RDS::DBInstance"

Properties:

DBInstanceClass: db.t2.medium

AllocatedStorage: '20'

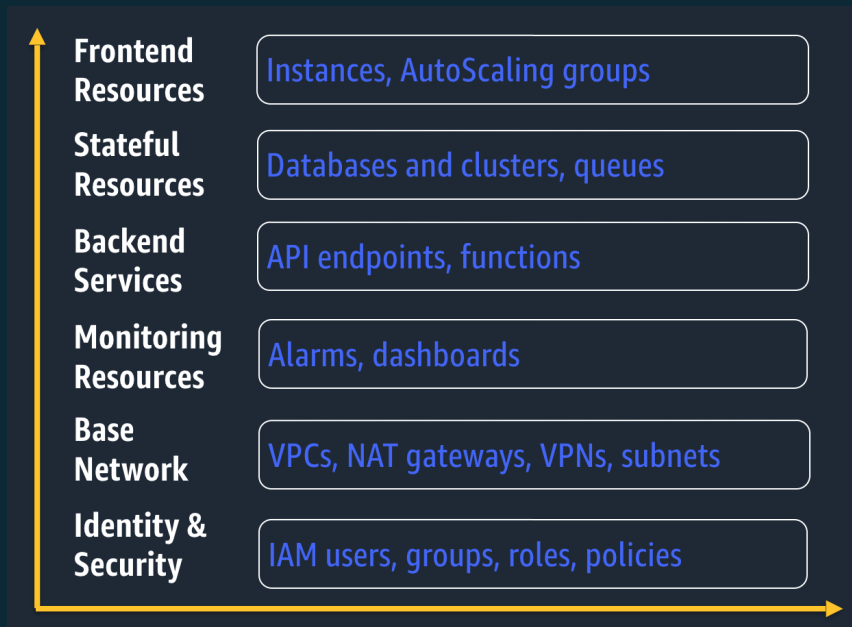
Engine: mariadb

EngineVersion: '10.2'

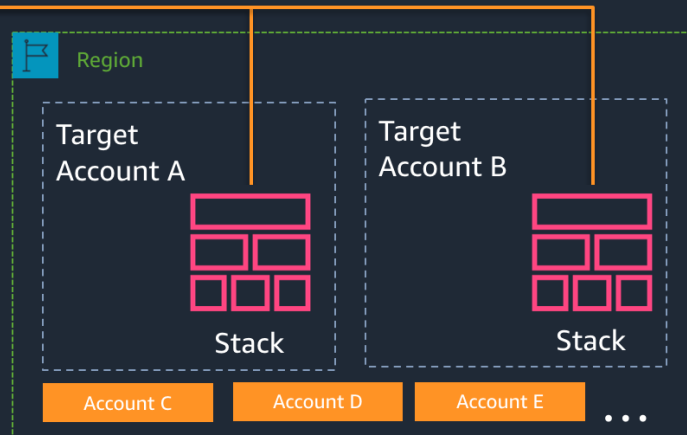
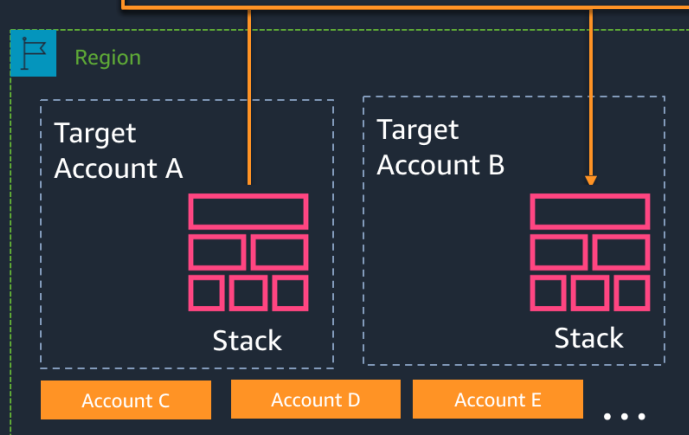
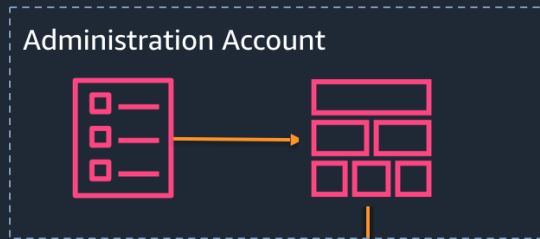
MasterUsername: appadmin

MasterUserPassword:

'{{resolve:ssm-secure:ssbRDSmEcntl:1}}'



StackSets



Infrastructure as Code With AWS Cloud Development Kit (CDK)



CDK

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

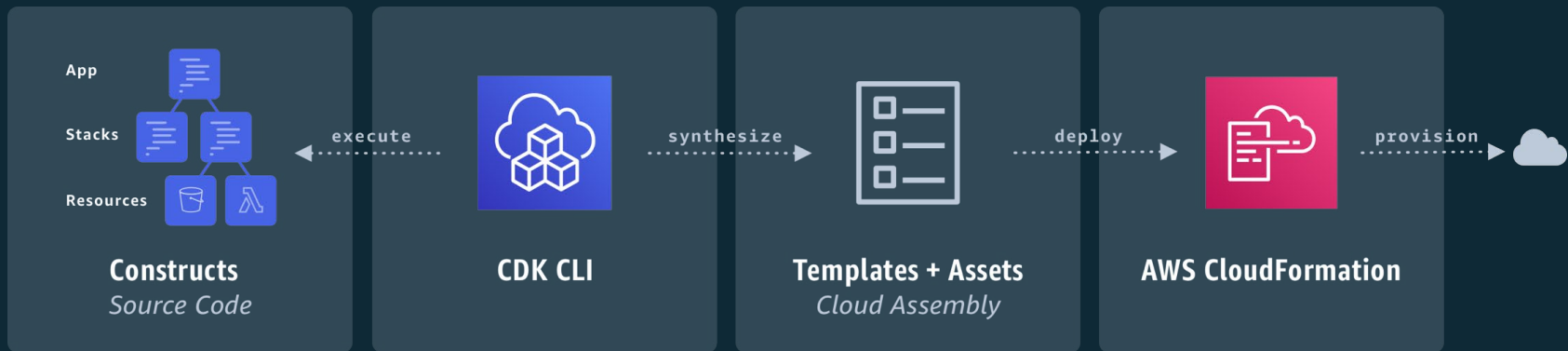
domainName	(property) patterns.NetworkLoadBalancedServiceBaseProps.domainName?: string undefined
domainZone	The domain name for the service, e.g. "api.example.com."
	@default
	- No domain name.

CDK

Multi-language
Framework

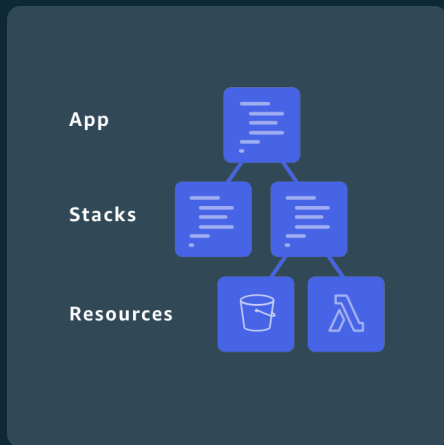


CDK



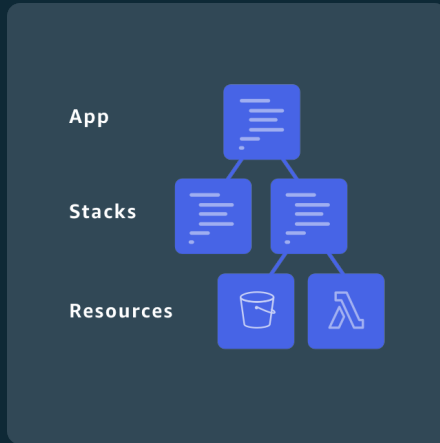
CDK Main Components

CDK Main Components



Core Framework

CDK Main Components

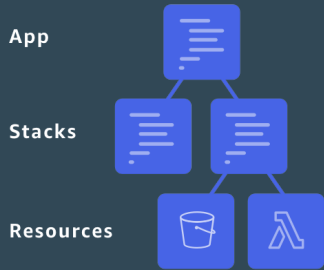


Core Framework

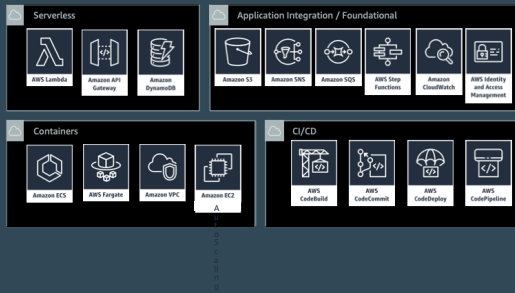


AWS Construct Library

CDK Main Components



Core Framework



AWS Construct Library

```
> cdk diff hello-cdk-1
Stack hello-cdk-1
Resources
[-] AWS::SNS::Topic MyFirstTopic MyFirstTopicBD1F8A4
[-] DisplayName
  [-] My First Topic Test
  [-] Hello, CDK!

~/hello-cdk master* $s
> cdk deploy hello-cdk-1

hello-cdk-1: deploying...
hello-cdk-1: creating CloudFormation changeset...
0/3 | 12:25:30 PM | UPDATE_IN_PROGRESS | AWS::CDK::Metadata | CDKMetadata
0/3 | 12:25:30 PM | UPDATE_IN_PROGRESS | AWS::SNS::Topic | MyFirstTopic (MyFirstTopicBD1F8A4)
1/3 | 12:25:31 PM | UPDATE_COMPLETE | AWS::SNS::Topic | MyFirstTopic (MyFirstTopicBD1F8A4)
2/3 | 12:25:32 PM | UPDATE_COMPLETE | AWS::CDK::Metadata | CDKMetadata
2/3 | 12:25:35 PM | UPDATE_COMPLETE_CLEANUP | AWS::CloudFormation::Stack | hello-cdk-1
3/3 | 12:25:36 PM | UPDATE_COMPLETE | AWS::CloudFormation::Stack | hello-cdk-1

✔ hello-cdk-1

Stack ARN:
arn:aws:cloudformation:us-east-1:585695836304:stack/hello-cdk-1/y9da27f0-fafe-11e9-b7b3-12f2c8f206e2

~/hello-cdk master* $ls
```

AWS CDK CLI

CDK Sample Code

```
export class MyCoolServiceStack extends Stack {  
  constructor(scope: Construct, id: string, props?: StackProps) {  
    super(scope, id, props);  
  
    const api = new apigw.RestApi(this, 'Api');  
  
    // ...  
  
    new walters.HealthMonitor(this, 'Monitor', {  
      app: 'MyCoolService',  
      region: 'us-east-1',  
      endpoint: api.url  
    });  
  }  
}
```

CDK Sample Code

```
export class MyCoolServiceStack extends Stack {  
  constructor(scope: Construct, id: string, props?: StackProps) {  
    super(scope, id, props);  
  
    const api = new apigw.RestApi(this, 'Api');  
  
    // ...  
  
    new walters.HealthMonitor(this, 'Monitor', {  
      app: 'MyCoolService',  
      region: 'us-east-1',  
      endpoint: api.url  
    });  
  }  
}
```

```
AWSTemplateFormatVersion: "2010-09-09"  
Description: A CodeCommit Repo and Cloud9 Environment  
Resources:  
  MyRepo:  
    Type: "AWS::CodeCommit::Repository"  
    Properties:  
      RepositoryName: MyRepo  
      RepositoryDescription: Sample Repository for Demo  
  MyC9Environment:  
    Type: "AWS::Cloud9::EnvironmentEC2"  
    Properties:  
      Repositories:  
        - PathComponent: /cfn  
          RepositoryUrl: !GetAtt MyRepo.CloneUrlHttp  
      InstanceType: t2.micro
```

CDK Constructs

L1 constructs – all resources in CloudFormation specification

CDK Constructs

L1 constructs – all resources in CloudFormation specification

L2 constructs – higher-level abstractions with sensible defaults

CDK Constructs

L1 constructs – all resources in CloudFormation specification

L2 constructs – higher-level abstractions with sensible defaults

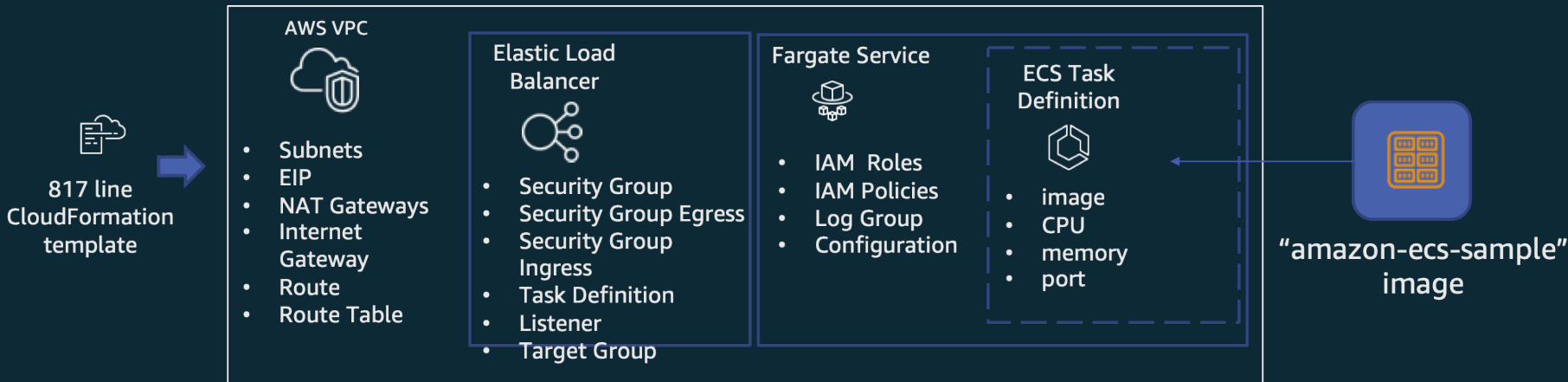
L3 constructs – opinionated reference architectures and design patterns using multiple AWS services

CDK Constructs

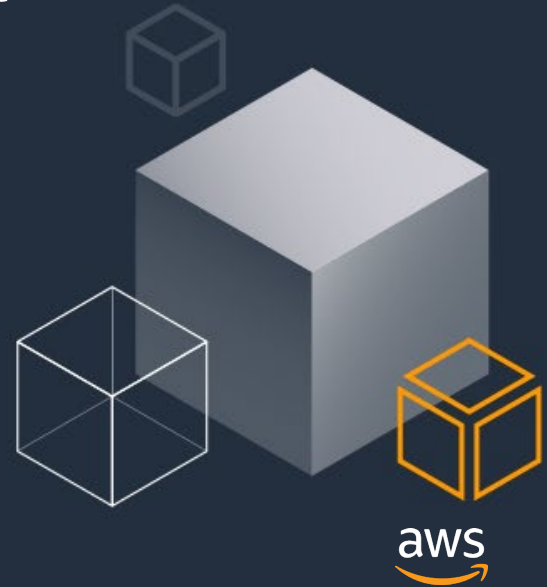
```
new patterns.ApplicationLoadBalancedFargateService(stack, 'MyFargateService',{
  taskImageOptions: {
    image: ecs.ContainerImage.fromRegistry("amazon/amazon-ecs-sample")
  }
});
```

CDK Constructs

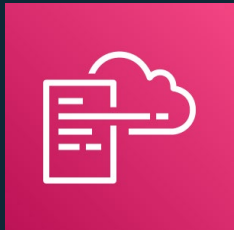
```
new patterns.ApplicationLoadBalancedFargateService(stack, 'MyFargateService',{
  taskImageOptions: {
    image: ecs.ContainerImage.fromRegistry("amazon/amazon-ecs-sample")
  }
});
```



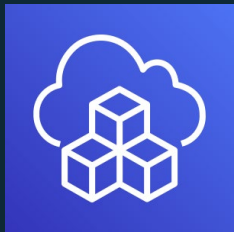
Infrastructure as Code: Additional Tooling Options



AWS Native Options

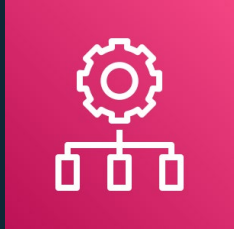


AWS CloudFormation



AWS Cloud Development Kit (CDK)

Additional AWS Options



AWS OpsWorks



AWS Service Catalog

Third Party Options

Terraform

Chef

Puppet

Ansible

SaltStack

Pulumi

...many more

Summary

Infrastructure as Code (IaC) makes managing cloud applications and their resources more repeatable, safer

AWS provides multiple options for IaC, including CloudFormation and CDK

IaC best practices apply across AWS and non-AWS tools

More Information

AWS Site and Documentation

<https://aws.amazon.com/cloudformation/>

<https://docs.aws.amazon.com/cloudformation/index.html>

<https://aws.amazon.com/cdk/>

AWS Open Source Resources

<https://github.com/aws>

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

<https://github.com/aws-labs/aws-cloudformation-templates>

<https://github.com/aws-cloudformation/cfn-python-lint>

<https://github.com/aws-cloudformation/cloudformation-cli>

Q&A

Thank You!

Richard Boyd, Sr. Developer Advocate, AWS Code Services
Luis Colon, Sr. Developer Advocate, AWS CloudFormation

April 20, 2020