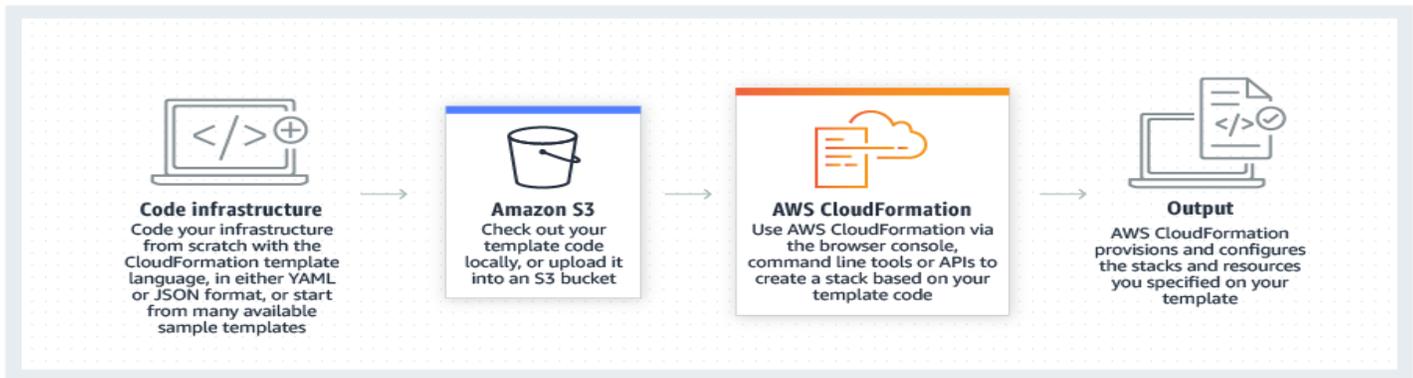


Cloud Infrastructure in Code (AWS CloudFormation vs AWS CDK)

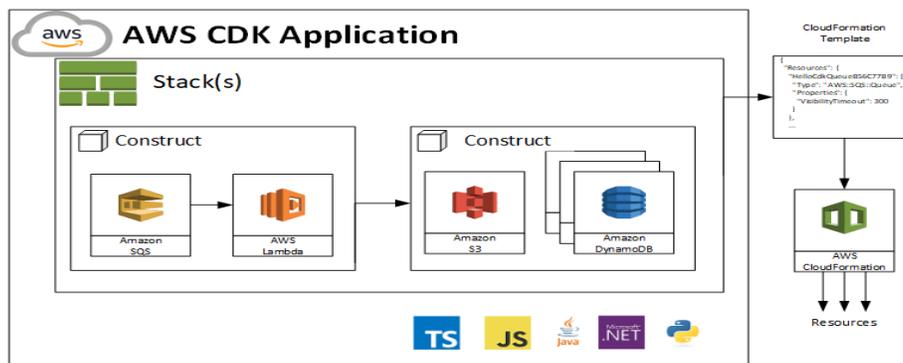
AWS CloudFormation

- CloudFormation is a declarative way of outlining your AWS Infrastructure, for any resources (most of them are supported)
- For example, within a CloudFormation template, you say:
 - I want a security group
 - I want two EC2 machines using this security group
 - I want two Elastic IPs for these EC2 machines
 - I want an S3 bucket
 - I want a load balancer (ELB) in front of these machines
- Use case: you can define CloudFormation resources in your extensions to provision ElastiCache, an S3 bucket, anything you want!



AWS Cloud Development Kit

- With this software development framework, you can model cloud applications using familiar programming languages (supports TypeScript, JavaScript, Python, Java, and C#/.Net), provision applications with customized CloudFormation templates, and accelerate application deployments at AWS.
- The AWS CDK is a software development framework that allows you to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS Infrastructure.
- The AWS CDK enables you to build your cloud application without leaving your integrated development environment (IDE).



Major Reasons behind preference of AWS CDK over CloudFormation

- Before, using AWS CDK, developers have used AWS CloudFormation, but working with it was a challenge, hard to define and had to get an overview for a complex template in either JSON or YAML format.
- But with AWS CDK you “WRITE LESS AND DO MORE”.
- **For example**, you have a CloudFormation template to define and configure networking resources including VPC, subnets, route tables, security groups, bastion hosts, integrate gateway, NAT gateways, etc. To accomplish this, we need at least **1000** lines of code in JSON or YAML format using CloudFormation. Now, with AWS CDK, we only need to write around **50-60** lines of code.

Advantages of CDK over CloudFormation:

- Easy to share and reuse your infrastructure as a library
- Developer-Friendly version of CloudFormation
- Easy to use logical statements (if statements, for-loops, etc.) when defining your infrastructure

```
private constructVpc(props: ProtractorFargateStackProps) {
  if (props.useDefaultVpc) {
    return ec2.Vpc.fromLookup(this, 'VPC', {
      isDefault: true
    });
  }

  return new ec2.Vpc(this, 'EcsFargateVPC', {
    cidr: '172.15.0.0/16',
    maxAzs: 1,
  });
}
```

- Easy to get and integrate with our coding review workflow
- Code completion within your IDE

```
// Write outputs
new cdk.CfnOutput(this, 'ecsClusterArn', {
  value: cluster.c
});
new cdk.CfnOutput(
  value: cluster.c
);
new cdk.CfnOutput(
  value: props.rep
);
}
private constructVpc
  if (props.useDefau
    return ec2.Vpc.f
      isDefault: true
  });
}
```

- Easy to integrate with our CI/CD process

Last but not the least, the AWS CDK is just a high-level of CloudFormation, so if you want to know how exactly it works, my suggestion is to have a try with CloudFormation, do some work around it, then you try to convert it into AWS CDK. sYou will find out what to use as nothing is the best, it depends on the requirement, team, project and your company.

Key Differences between AWS CDK and CloudFormation

	AWS CDK	CloudFormation
Supported Format	JSON and YAML	JavaScript, TypeScript, Python, Java, and C#.
Code Writeup	Need to provide each and every information in supported format	Write less and do more, i.e. compared to CloudFormation, coding required is very less
Inheritance	Does not support inheritance of libraries	It supports inheritance of libraries which minimizes coding repeated set of lines.
Compatibility	Need to define everything from start in supported format.	CDK enables you to use your existing skills and tools, and apply those to the task of building cloud infrastructure.
Benefits	<ul style="list-style-type: none"> • Automate best practices • Scale your infrastructure worldwide • Integrate with other AWS services • Manage third-party and private resources • Extend CloudFormation with the community 	<ul style="list-style-type: none"> • Because the AWS CDK leverages CloudFormation, you still enjoy all the benefits CloudFormation provides such as safe deployment, automatic rollback, and drift detection • Easier cloud onboarding • Faster development process • Customizable and shareable • No context switching

Author: Giten Mitra, Cloud Solution Architect - MIND

Date: 17/02/2021