What is AWS CDK and what advantages does it offer over traditional CloudFormation? Which programming languages does CDK support and which would you recommend for a new project? When would you prefer CDK and when would you prefer CloudFormation.

What is AWS CDK? (Cloud Development Kit)
"AWS CDK is a **framework** that lets you **define your cloud infrastructure using real** programming languages like **TypeScript**, **Python**, or **Java**, rather than writing YAML/JSON CloudFormation templates. It compiles your code into standard CloudFormation templates behind the scenes."
Think of it like this:

- **CloudFormation** = Writing static YAML
- **CDK** = Writing code that *generates* YAML

CDK gives you the **power of programming**: functions, loops, conditionals, and reusable components.

✅ Advantages of CDK Over Traditional CloudFormation

1. 🧠 Type Safety + IDE Auto-complete
With CDK, mistakes like typos in resource names or wrong property types are caught by your IDE.
2. 🔁 Code Reusability
You can write reusable classes/functions. For example:
ts
```
new MyVpcConstruct(this, 'AppVpc');
```

new MyVpcConstruct(this, 'DbVpc');

3. 🤖 Loops and Conditions Made Easy

Creating 3 subnets in CloudFormation = 50 lines

In CDK:

ts

```
for (let i = 0; i < 3; i++) {
  // define subnet
}
```

4. 🛡️ Built-in Best Practices

CDK applies secure defaults — e.g., S3 buckets block public access unless explicitly told not to.

5. 🧪 Unit Testing Infrastructure

You can write **tests using tools like jest or pytest to verify if** resources are being created correctly.

🌐 Languages Supported by AWS CDK

- **TypeScript** ✅ (best support)
- **Python**
- **Java**
- **C#**
- **Go**
- **JavaScript**

My Recommendation for New Projects

▶️ Use TypeScript

"I always recommend TypeScript for new CDK projects. It's the language AWS CDK was originally built in, has the most examples, and best community support."

🐍 Use Python if:

- Your team is already using Python
- You need tighter integration with Python-based tooling or data pipelines

---

🔍 When to Use CDK vs CloudFormation

Use CDK When...	**Use CloudFormation When...**
Infrastructure is complex or dynamic	Infra is simple and mostly static
You want reusable constructs (OOP style)	Compliance teams require pure YAML/JSON
Team is comfortable with code	Team prefers declarative configs
You want testing, code reviews, GitOps style	You're migrating or maintaining existing templates
You build infrastructure components often	You need bleeding-edge AWS features (CDK may lag)

🧪 Simple Example: CDK in TypeScript

Let's say you want to create an **S3 bucket** using CDK.

📁 Project Setup

bash

CopyEdit

```
mkdir my-cdk-app && cd my-cdk-app
cdk init app --language typescript
npm install @aws-cdk/aws-s3
```

🧾 **lib/my-cdk-app-stack.ts**

ts

```
import * as cdk from 'aws-cdk-lib';
import { Stack, StackProps } from 'aws-cdk-lib';
import * as s3 from 'aws-cdk-lib/aws-s3';

export class MyCdkAppStack extends Stack {
  constructor(scope: cdk.App, id: string, props?: StackProps) {
    super(scope, id, props);

    new s3.Bucket(this, 'MyBucket', {
      versioned: true,
      blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
      removalPolicy: cdk.RemovalPolicy.DESTROY,
    });
  }
}
```

🛠️ **Deploy**

bash

```
cdk deploy
```

🎉 That's it! You've just provisioned infrastructure with real code — **no YAML** needed.

🎤

At my previous company, I led a migration of 30+ services from static CloudFormation templates to reusable CDK constructs. This **reduced code duplication by over 60%,** made **onboarding new developers easier,** and gave us the flexibility to roll out global updates across all stacks using versioned constructs. CDK brought the DevOps and dev teams closer — because infra became just another part of the application code."

What is AWS CDK primarily used for?

A. Automating IAM credential rotation
B. Writing Lambda functions in Java
C. Defining cloud infrastructure using real programming languages
D. Migrating RDS to DynamoDB

Answer: C

What does CDK generate behind the scenes when you run `cdk deploy`?
A. Terraform scripts
B. JSON/YAML CloudFormation templates
C. Shell scripts for EC2
D. SSM Parameter Store values

Answer: B

Which of the following is an advantage of using CDK over traditional CloudFormation?
A. Requires no installation
B. Doesn't support loops
C. Enables use of loops, conditionals, and code reuse
D. Only works with Go

Answer: C

Which programming language has the best community support and native integration with AWS CDK?
A. Go
B. Java
C. Python
D. TypeScript

Answer: D