

Complete AWS CDK Mastery Course Outline

This structured course will take you from AWS CDK basics to advanced patterns, with hands-on projects and best practices.

Course Title: AWS CDK - From Zero to Production

Module 1: CDK Fundamentals

1.1 Introduction to Infrastructure as Code (IaC)

- What is AWS CDK?
- How CDK compares to CloudFormation/Terraform/Pulumi
- Why use CDK over raw CloudFormation?

1.2 Setting Up Your CDK Environment

- Installing Node.js, Python, or Java (depending on preferred language)
- `npm install -g aws-cdk`
- `cdk init` – Creating your first app
- Bootstrapping your AWS account

1.3 Core CDK Concepts

- Constructs, Stacks, and Apps
 - The AWS Construct Library
 - How CDK synthesizes CloudFormation templates
 - `cdk synth`, `cdk deploy`, `cdk diff`
-

Module 2: Building Your First CDK App

2.1 Deploying Basic AWS Resources

- Creating an S3 bucket
- Setting up an EC2 instance
- Configuring IAM roles and policies

2.2 Working with Stacks

- Multi-stack applications
- Passing values between stacks
- Stack dependencies

2.3 CDK Best Practices

- Organizing your project structure
 - Using `props` for configuration
 - Avoiding hardcoded values (using `context` and environment variables)
-

Module 3: Intermediate CDK Patterns

3.1 Leveraging Higher-Level Constructs (L2 & L3)

- Difference between L1 (Cfn), L2, and L3 constructs
- Using `aws-ec2`, `aws-rds`, `aws-lambda` libraries
- Building reusable patterns

3.2 Custom Constructs & Reusable Components

- Creating your own constructs
- Publishing constructs to npm/PyPI
- Sharing across teams

3.3 Managing Environments

- Dev/Stage/Prod separation
 - Using CDK Pipelines for CI/CD
 - Feature flags and environment variables
-

Module 4: Advanced CDK Techniques

4.1 CDK with Serverless (Lambda, API Gateway, DynamoDB)

- Lambda functions with CDK
- REST APIs with API Gateway

- DynamoDB tables and streams

4.2 Networking & Security

- VPCs, Subnets, and Security Groups
- Cross-account deployments
- PrivateLink and VPC peering

4.3 CDK and Kubernetes (EKS)

- Setting up EKS clusters with CDK
 - Deploying Helm charts
 - Managing Kubernetes workloads
-

Module 5: Testing & Debugging CDK Apps

5.1 Unit & Integration Testing

- Using `@aws-cdk/assertions`
- Snapshot testing
- Testing custom constructs

5.2 Debugging & Troubleshooting

- Understanding `cdk synth` output
 - Common CDK errors and fixes
 - Using CDK debug logs
-

Module 6: CI/CD & Production Best Practices

6.1 CDK Pipelines (Auto-Deployments)

- Setting up a self-mutating pipeline
- Multi-region/account deployments
- Manual approvals and notifications

6.2 Security & Compliance

- IAM least privilege in CDK
- Using AWS Organizations & SCPs
- CDK and AWS Config rules

6.3 Cost Optimization

- CDK and AWS Budgets
 - Spot instances and auto-scaling
 - Right-sizing resources
-

Module 7: Real-World CDK Projects

7.1 Project 1: Serverless Web App

- Frontend (S3 + CloudFront)
- Backend (Lambda + API Gateway + DynamoDB)
- CI/CD pipeline

7.2 Project 2: Multi-Tier Architecture

- VPC with public/private subnets
- EC2 auto-scaling groups
- RDS with read replicas

7.3 Project 3: Event-Driven System

- SQS, SNS, and EventBridge
 - Step Functions workflows
 - Monitoring with CloudWatch
-

Bonus: CDK for Terraform Users

- Comparing CDKTF (CDK for Terraform) vs AWS CDK
 - Migrating from Terraform to CDK
 - Hybrid approaches
-

Course Format

- ✓ **Video Lessons** (10-15 min each)
 - ✓ **Hands-On Labs** (GitHub repos with starter code)
 - ✓ **Quizzes & Challenges**
 - ✓ **Community Discord for Q&A**
 - ✓ **Final Certification Project**
-

Who Is This Course For?

- **DevOps Engineers** who want to move beyond CloudFormation/YAML
 - **Developers** who want to manage AWS with code
 - **Cloud Architects** designing reusable infrastructure
 - **Teams** adopting modern IaC practices
-

Prerequisites

- Basic AWS knowledge (EC2, S3, IAM)
 - Familiarity with a programming language (TypeScript/Python/Java)
 - AWS Free Tier account
-

Would you like me to expand any section?

I can provide:

- ♦ **Detailed code examples**
- ♦ **Step-by-step deployment guides**
- ♦ **Troubleshooting tips**
- ♦ **Real-world case studies**

Let me know which part you'd like to dive deeper into! 🚀