

Beyond CDK Pipelines for Modern Serverless CI/CD

Software Developer and Architect

Doing Serverless on AWS @  merapar

AWS Community Builder

Sharing knowledge:

- [BetterDev.blog](https://betterdev.blog)
- [@radzikowski_m](https://twitter.com/radzikowski_m)

1. Why AWS CodePipeline?
2. Capabilities vs. Requirements
3. Opinionated CDK Pipeline

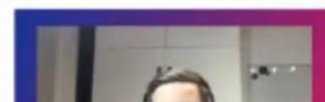
What is the difference between CodePipeline and CDK pipelines? 

It's important to note the difference between the CodePipeline module and CDK Pipelines. While CDK Pipelines uses CodePipeline for deploying stacks created with CDK, CodePipeline is a more general-purpose tool that allows you to customize your own pipeline as you would when setting up a new CI/CD pipeline. Mar 14, 2023

Why AWS CodePipeline?

AWS CodePipeline cons... and pros

- | | |
|---|-----------------------------------|
| ▪ long CodeBuild provisioning time | ▪ no external integrations |
| ▪ complex to set up | ▪ security (uses IAM) |
| ▪ no built-in notifications
(commit build status, Slack) | ▪ no corporate approvals needed |
| ▪ no PR / feature-branch support
(solution: dynamically create
pipelines per branch...) | ▪ no separate billing |
| | ▪ uniform across all AWS projects |



CDK to the rescue

Complex set up, additional resources to extend CodePipeline capabilities



Hidden behind a reusable CDK Construct with simple API

Capabilities vs. Requirements

What CodePipeline can do

- start build on Git commit
- build any project (with CodeBuild)
- deploy CloudFormation stacks
- deploy multiple environments
- deploy to multiple accounts
- run post-deployment tests (with CodeBuild)



What Serverless CI/CD needs

- feature branch deployments to ephemeral environments
- development environments deployments from the local CLI
- commit build status notifications to repository
- deployment failures notifications

Problems with connecting to GitHub / Bitbucket

- Bitbucket connection permissions not limited to a single repository
- lacking repository event triggers like branch deletion

Opinionated CDK Pipeline

Opinionated²

CDK Pipelines is an **opinionated construct library**.
It is purpose-built to deploy one or more copies
of your CDK applications using CloudFormation
with a minimal amount of effort on your part.
It is not intended to support arbitrary deployment pipelines (...).

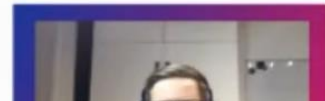
– *CDK Pipelines docs*

The screenshot shows the AWS CDK v2 Developer Guide page for Continuous integration and delivery (CI/CD) using CDK Pipelines. The page is titled "Continuous integration and delivery (CI/CD) using CDK Pipelines" and includes a "PDF" and "RSS" link. A blue box at the top states: "This is the AWS CDK v2 Developer Guide. The older CDK v1 entered maintenance on June 1, 2022 and will now receive only critical bug fixes and security patches. New features will be developed for CDK v2 exclusively. Support for CDK v1 will end entirely on June 1, 2023." The main text describes CDK Pipelines as a construct library module for painless continuous delivery of AWS CDK applications. A "Note" box states: "CDK Pipelines supports two APIs. One is the original API that was made available in the CDK Pipelines Developer Preview. The other is a modern API that incorporates feedback from CDK customers received during the preview phase. The examples in this topic use the modern API. For details on the differences between the two supported APIs, see [CDK Pipelines original API](#)." The page also includes a "Bootstrap your AWS environments" section with a link to the bootstrap command. The left sidebar shows the "AWS Cloud Development Kit (AWS CDK) v2" navigation menu, and the right sidebar shows the "On this page" table of contents.

https://docs.aws.amazon.com/cdk/v2/guide/cdk_pipeline.html

Goals

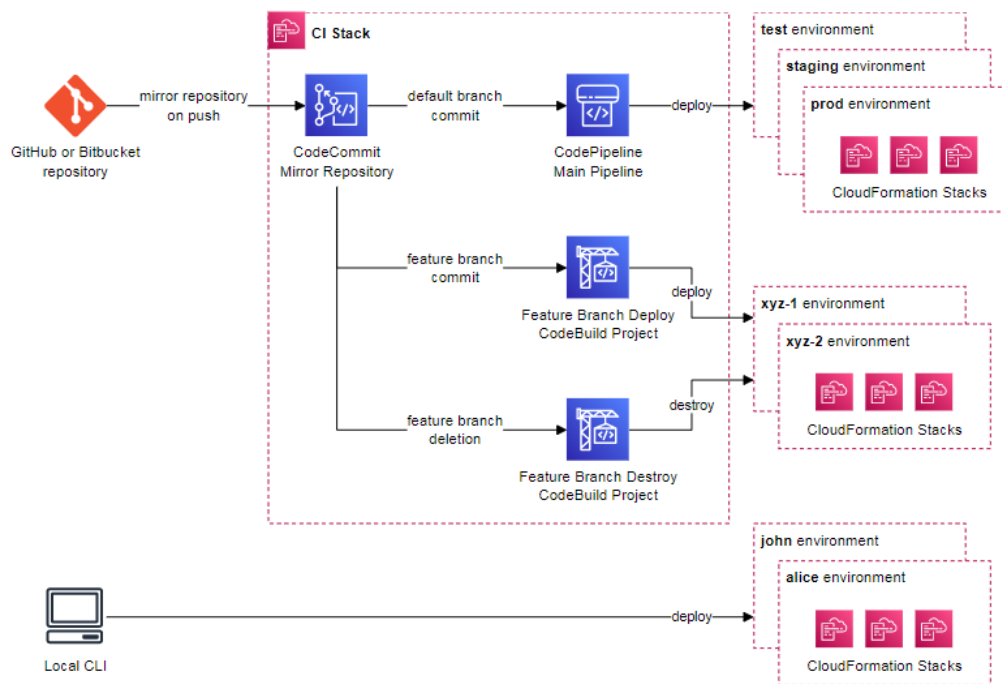
- built on top of CDK Pipelines
- customizable regarding the build and deployment steps
- easy integration with GitHub and Bitbucket
- feature branch deployments to ephemeral environments
- development environments deployments from the local CLI
- commit build status notifications to repository
- build failures notifications to Slack



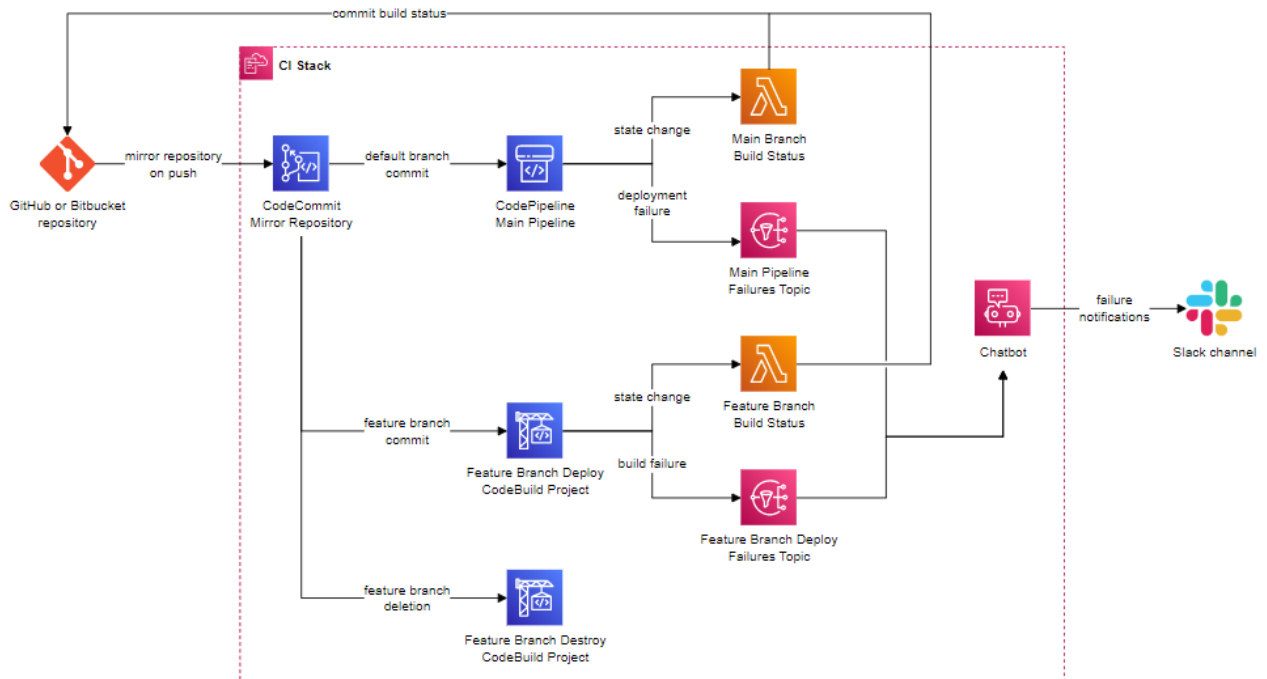
Solution

- `CDKApplication` extending `cdk.App`
- mirroring GitHub/Bitbucket repository to CodeCommit
- CodePipeline for the `main` branch multi-stage deployments
- CodeBuild for the feature branch deployments
- Lambda sending build status notifications to GitHub/Bitbucket
- AWS Chatbot integration for Slack notifications

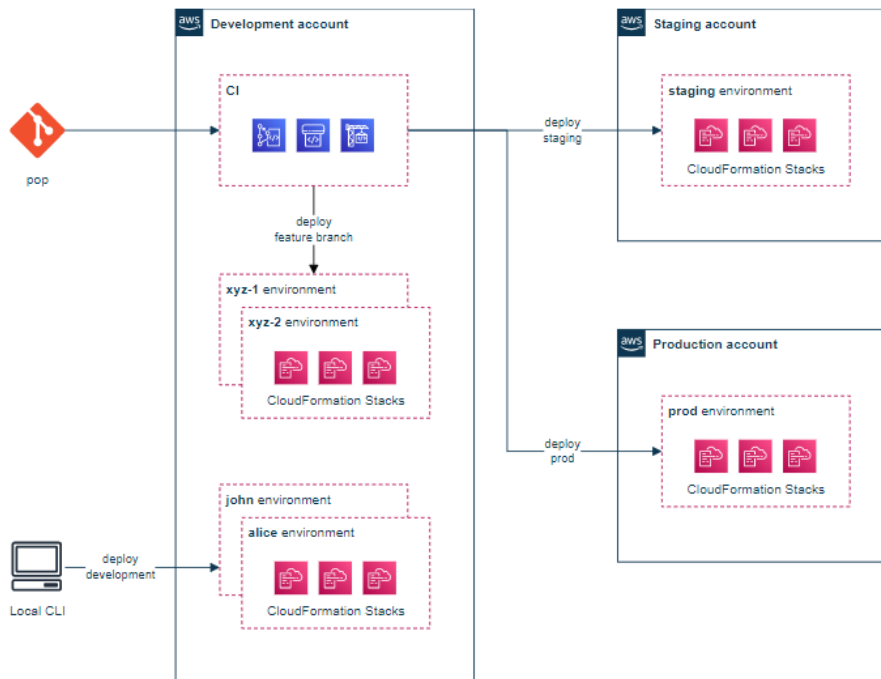
Architecture - deployments



Architecture - notifications



Architecture - accounts



Usage example

```
#!/usr/bin/env node
import {CDKApplication} from 'opinionated-ci-pipeline';
import {ExampleStack} from '../lib/exampleStack';

new CDKApplication({
  stacks: {
    create: (scope, projectName, envName) => {
      new ExampleStack(scope, 'ExampleStack', {stackName: `${projectName}-${envName}-ExampleStack`});
    },
  },
  repository: { host: 'github', name: 'organization/repository' },
  packageManager: 'npm',
  pipeline: [
    { environment: 'test', post: ['echo "do integration tests here"'] },
    { environment: 'prod' },
  ],
});
```

Missing features

- queue feature-branch deployments

More on Serverless: [BetterDev.blog](https://betterdev.blog) [@radzikowski_m](https://twitter.com/radzikowski_m)

Project: [Opinionated CI Pipeline](#)
 [CDK Construct available for Node.js and Python](#)

This presentation: [BetterDev.blog/slides/cdk-pipelines](https://betterdev.blog/slides/cdk-pipelines)