AWS CDK Infrastructure in code

CONCEPTS

> The CLI runs your app (the command in cdk.json) to produce CloudFormation templates; deploy then hands them to CloudFormation.

App code constructs
↓ `cdk synth`
cdk.out cloud assembly
↓ `cdk deploy`
CloudFormation change set
↓ creates
AWS resources per stack

> Bootstrap each account/region pair once before the first deploy that uses assets.

PROJECT

`cdk init app --language <lang>`
Create an empty app here

STACKS

`cdk ls`
List stack IDs

`cdk ls --long`
Show stack names and environments

`cdk deploy <stack>`
Deploy a single stack

`cdk deploy --all`
Deploy every stack

`cdk deploy "Prod/*"`
Deploy stacks matching a pattern

SYNTHESIZE & DIFF

`cdk synth`
Print the CloudFormation template

`cdk synth <stack> --json`
Template as JSON

`cdk diff`
Compare code with the deployed stack

DEPLOY

`cdk deploy`
Deploy the only stack

`cdk deploy --require-approval never`
Skip the security prompt

`cdk deploy --outputs-file <file>`
Save stack outputs as JSON

`cdk deploy --parameters <Name>=<value>`
Set a CloudFormation parameter

`cdk deploy -c <key>=<value>`
Pass a context value

WATCH & DESTROY

`cdk watch`
Redeploy on file changes

`cdk destroy`
Delete the only stack

BOOTSTRAP

`cdk bootstrap`
Bootstrap the app's environments

`cdk bootstrap aws://<account>/<region>`
Bootstrap a specific environment

CONTEXT & CONFIG

`cdk --context <key>=<value>`
Set context for one run

cdk.json keys

Key	Purpose
`app`	Command that runs your app
`build`	Pre-synth build command
`watch`	include / exclude globs for cdk watch
`context`	Fixed context values and feature flags
`requireApproval`	Default approval level
`output`	Synth folder (default cdk.out)
`profile`	Default AWS profile
`outputsFile`	Default for --outputs-file