:doctype: book

include::attributes.txt[]

// Attributes

:https–docs-aws-amazon-com-cdk-api-v2-docs-aws-cdk-lib-pipelines-CodePipeline-html-addwbrstagestage-optionss: https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.pipelines.CodePipeline.html#addwbrstagestage-optionss :https–github-com-aws-aws-cdk-blob-master-packages–aws-cdk-pipelines-ORIGINAL-API-md: https://github.com/aws/aws-cdk/blob/master/packages/@aws-cdk/pipelines/ORIGINAL\_API.md

[.topic] [#cdk-pipeline] = Continuous integration and delivery (CI/CD) using CDK Pipelines :info\_titleabbrev: Create CDK Pipelines

// Content start

Use the https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.pipelines-readme.html[CDK Pipelines] module from the {aws} Construct Library to configure continuous delivery of {aws} CDK applications. When you commit your CDK app’s source code into {aws} CodeCommit, GitHub, or {aws} CodeStar, CDK Pipelines can automatically build, test, and deploy your new version.

CDK Pipelines are self-updating. If you add application stages or stacks, the pipeline automatically reconfigures itself to deploy those new stages or stacks.

= [NOTE]

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 link:https://github.com/aws/aws-cdk/blob/master/packages/@aws-cdk/pipelines/ORIGINAL\_API.md [CDK Pipelines original API] in the *aws-cdk GitHub repository*.

====

[#cdk-pipeline-bootstrap] == Bootstrap your {aws} environments

Before you can use CDK Pipelines, you must bootstrap the {aws} xref:environments[environment] that you will deploy your stacks to.

A CDK Pipeline involves at least two environments. The first environment is where the pipeline is provisioned. The second environment is where you want to deploy the application’s stacks or stages to (stages are groups of related stacks). These environments can be the same, but a best practice recommendation is to isolate stages from each other in different environments.

= [NOTE]

See xref:bootstrapping[{aws} CDK bootstrapping] for more information on the kinds of resources created by bootstrapping and how to customize the bootstrap stack.

====

Continuous deployment with CDK Pipelines requires the following to be included in the CDK Toolkit stack:

* An Amazon Simple Storage Service (Amazon S3) bucket.
* An Amazon ECR repository.
* IAM roles to give the various parts of a pipeline the permissions they need.

The CDK Toolkit will upgrade your existing bootstrap stack or creates a new one if necessary.

To bootstrap an environment that can provision an {aws} CDK pipeline, invoke cdk bootstrap as shown in the following example. Invoking the {aws} CDK Toolkit via the npx command temporarily installs it if necessary. It will also use the version of the Toolkit installed in the current project, if one exists.

--cloudformation-execution-policies specifies the ARN of a policy under which future CDK Pipelines deployments will execute. The default AdministratorAccess policy makes sure that your pipeline can deploy every type of {aws} resource. If you use this policy, make sure you trust all the code and dependencies that make up your {aws} CDK app.

Most organizations mandate stricter controls on what kinds of resources can be deployed by automation. Check with the appropriate department within your organization to determine the policy your pipeline should use.

You can omit the --profile option if your default {aws} profile contains the necessary authentication configuration and {aws} Region.

==== [role=“tablist”] macOS/Linux:: + [source,none,subs=“verbatim,attributes”] — npx cdk bootstrap aws://++++++/++++++–profile ++++++ –cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess —-++++++++++++++++++

Windows:: + [source,none,subs=“verbatim,attributes”] — npx cdk bootstrap aws://++++++ –profile< ADMIN-PROFILE> {caret} –cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess —- ====++++++

To bootstrap additional environments into which {aws} CDK applications will be deployed by the pipeline, use the following commands instead. The --trust option indicates which other account should have permissions to deploy {aws} CDK applications into this environment. For this option, specify the pipeline’s {aws} account ID.

Again, you can omit the --profile option if your default {aws} profile contains the necessary authentication configuration and {aws} Region.

==== [role=“tablist”] macOS/Linux:: + [source,none,subs=“verbatim,attributes”] — npx cdk bootstrap aws://++++++/++++++–profile ++++++ –cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess  –trust ++++++—-++++++++++++++++++++++++

Windows:: + [source,none,subs=“verbatim,attributes”] — npx cdk bootstrap aws://++++++/++++++–profile ++++++{caret} –cloudformation-execution-policies arn:aws:iam::aws:policy/AdministratorAccess {caret} –trust ++++++—- ====++++++++++++++++++++++++

= [TIP]

Use administrative credentials only to bootstrap and to provision the initial pipeline. Afterward, use the pipeline itself, not your local machine, to deploy changes.

====

If you are upgrading a legacy bootstrapped environment, the previous Amazon S3 bucket is orphaned when the new bucket is created. Delete it manually by using the Amazon S3 console.