include::attributes.txt[]

// Attributes

[.topic] [#ref-cli-cmd-import] = cdk import :keywords: {aws} CDK, {aws} CDK CLI, CDK Toolkit, IaC, Infrastructure as code, {aws} Cloud, {aws} CloudFormation, cdk import

== [abstract]

## Use https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resource-import.html[{aws} CloudFormation resource imports] to import existing {aws} resources into a CDK stack.

// Content start

Use https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resource-import.html[{aws} CloudFormation resource imports] to import existing {aws} resources into a CDK stack.

With this command, you can take existing resources that were created using other methods and start managing them using the {aws} CDK.

When considering moving resources into CDK management, sometimes creating new resources is acceptable, such as with IAM roles, Lambda functions, and event rules. For other resources, such as stateful resources like Amazon S3 buckets and DynamoDB tables, creating new resources can cause impacts to your service. You can use cdk import to import existing resources with minimal disruption to your services. For a list of supported {aws} resources, see https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resource-import-supported-resources.html[Resource type support] in the *{aws} CloudFormation User Guide*.

*To import an existing resource to a CDK stack*:: + . Run a cdk diff to make sure your CDK stack has no pending changes. When performing a cdk import, the only changes allowed in an import operation are the addition of new resources being imported. + . Add constructs for the resources you want to import to your stack. For example, add the following for an Amazon S3 bucket: + [source,javascript,subs=“verbatim,attributes”] — new s3.Bucket(this, ‘ImportedS3Bucket’, {}); — + Do not add any other changes. You must also make sure to exactly model the state that the resource currently has. For the bucket example, be sure to include {aws} KMS keys, lifecycle policies, and anything else that is relevant about the bucket. Otherwise, subsequent update operations may not do what you expect. + . Run cdk import. If there are multiple stacks in the CDK app, pass a specific stack name as an argument. + . The CDK CLI will prompt you to pass in the actual names of the resources you are importing. After you provide this information, import will begin. + . When cdk import reports success, the resource will be managed by the CDK. Any subsequent changes in the construct configuration will be reflected on the resource.

This feature currently has the following limitations:

* Importing resources into nested stacks isn’t possible.
* There is no check on whether the properties you specify are correct and complete for the imported resource. Try starting a drift detection operation after importing.
* Resources that depend on other resources must all be imported together, or individually, in the right order. Otherwise, the CloudFormation deployment will fail with unresolved references.
* This command uses the deploy role credentials, which is necessary to read the encrypted staging bucket. This requires version 12 of the bootstrap template, which includes the necessary IAM permissions for the deploy role.

[#ref-cli-cmd-import-usage] == Usage

== [source,none,subs=“verbatim,attributes”]

$ cdk import ++++++++++++—-++++++++++++

[#ref-cli-cmd-import-args] == Arguments

[#ref-cli-cmd-import-args-stack-name] *CDK stack ID*:: The construct ID of the CDK stack from your app to import resources to. This argument can be provided multiple times in a single command. + *Type*: String + *Required*: No

[#ref-cli-cmd-import-options] == Options

For a list of global options that work with all CDK CLI commands, see xref:ref-cli-cmd-options[Global options].

[#ref-cli-cmd-import-options-change-set-name] --change-set-name <STRING>:: The name of the CloudFormation change set to create.

[#ref-cli-cmd-import-options-execute] --execute <BOOLEAN>:: Specify whether to execute change set. + *Default value*: true

[#ref-cli-cmd-import-options-force] --force, -f <BOOLEAN>:: By default, the CDK CLI exits the process if the template diff includes updates or deletions. Specify true to override this behavior and always continue with importing.

[#ref-cli-cmd-import-options-help] --help, -h <BOOLEAN>:: Show command reference information for the cdk import command.

[#ref-cli-cmd-import-options-record-resource-mapping] --record-resource-mapping, -r <STRING>:: Use this option to generate a mapping of existing physical resources to the CDK resources that will be imported. The mapping will be written to the file path that you provide. No actual import operations will be performed.

[#ref-cli-cmd-import-options-resource-mapping] --resource-mapping, -m <STRING>:: Use this option to specify a file that defines your resource mapping. The CDK CLI will use this file to map physical resources to resources for import instead of interactively asking you. + This option can be run from scripts.

[#ref-cli-cmd-import-options-rollback] --rollback <BOOLEAN>:: Roll back the stack to stable state on failure. + To specify false, you can use --no-rollback or -R. + Specify false to iterate more rapidly. Deployments containing resource replacements will always fail. + *Default value*: true

[#ref-cli-cmd-import-options-toolkit-stack-name] --toolkit-stack-name <STRING>:: The name of the CDK Toolkit stack to create. + By default, cdk bootstrap deploys a stack named CDKToolkit into the specified {aws} environment. Use this option to provide a different name for your bootstrap stack. + The CDK CLI uses this value to verify your bootstrap stack version.