:doctype: book

include::attributes.txt[]

// Attributes

[.topic] [#ref-cli-cdk-migrate] = cdk migrate :keywords: {aws} CDK, {aws} CDK CLI, cdk migrate

== [abstract]

## Migrate deployed {aws} resources, {aws} CloudFormation stacks, and CloudFormation templates into a new {aws} CDK project.

// Content start

Migrate deployed {aws} resources, {aws} CloudFormation stacks, and CloudFormation templates into a new {aws} CDK project.

This command creates a new CDK app that includes a single stack that is named with the value you provide using --stack-name. You can configure the migration source using --from-scan, --from-stack, or --from-path.

For more information on using cdk migrate, see xref:migrate[Migrate existing resources and {aws} CloudFormation templates to the {aws} CDK].

= [NOTE]

The cdk migrate command is experimental and may have breaking changes in the future.

====

[#ref-cli-cdk-migrate-usage] == Usage

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

$ cdk migrate ++++++—-++++++

[#ref-cli-cdk-migrate-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-cdk-migrate-options-required] === Required options

[#ref-cli-cdk-migrate-options-stack-name] --stack-name <STRING>:: The name of the {aws} CloudFormation stack that will be created within the CDK app after migrating. + *Required*: Yes

[#ref-cli-cdk-migrate-options-conditional] === Conditional options

[#ref-cli-cdk-migrate-options-from-path] --from-path <PATH>:: The path to the {aws} CloudFormation template to migrate. Provide this option to specify a local template. + *Required*: Conditional. Required if migrating from a local {aws} CloudFormation template.

[#ref-cli-cdk-migrate-options-from-scan] --from-scan <STRING>:: When migrating deployed resources from an {aws} environment, use this option to specify whether a new scan should be started or if the {aws} CDK CLI should use the last successful scan. + *Required*: Conditional. Required when migrating from deployed {aws} resources. + *Accepted values*: most-recent, new

[#ref-cli-cdk-migrate-options-from-stack] --from-stack <BOOLEAN>:: Provide this option to migrate from a deployed {aws} CloudFormation stack. Use --stack-name to specify the name of the deployed {aws} CloudFormation stack. + *Required*: Conditional. Required if migrating from a deployed {aws} CloudFormation stack.

[#ref-cli-cdk-migrate-options-optional] === Optional options

[#ref-cli-cdk-migrate-options-account] --account <STRING>:: The account to retrieve the {aws} CloudFormation stack template from. + *Required*: No + *Default*: The {aws} CDK CLI obtains account information from default sources.

[#ref-cli-cdk-migrate-options-compress] --compress <BOOLEAN>:: Provide this option to compress the generated CDK project into a ZIP file. + *Required*: No

[#ref-cli-cdk-migrate-options-filter] --filter <ARRAY>:: Use when migrating deployed resources from an {aws} account and {aws} Region. This option specifies a filter to determine which deployed resources to migrate. + This option accepts an array of key-value pairs, where *key* represents the filter type and *value* represents the value to filter. + The following are accepted keys: + –

* resource-identifier – An identifier for the resource. Value can be the resource logical or physical ID. For example, resource-identifier="ClusterName".
* resource-type-prefix – The {aws} CloudFormation resource type prefix. For example, specify +resource-type-prefix="{aws}::DynamoDB::"+ to filter all Amazon DynamoDB resources.
* tag-key – The key of a resource tag. For example, tag-key="myTagKey".
* {blank}
* == tag-value – The value of a resource tag. For example, tag-value="myTagValue".
* Provide multiple key-value pairs for AND conditional logic. The following example filters for any DynamoDB resource that is tagged with myTagKey as the tag key: +--filter resource-type-prefix="{aws}::DynamoDB::", tag-key="myTagKey"+.
* Provide the --filter option multiple times in a single command for OR conditional logic. The following example filters for any resource that is a DynamoDB resource or is tagged with myTagKey as the tag key: +--filter resource-type-prefix="{aws}::DynamoDB::" --filter tag-key="myTagKey"+.
* *Required*: No

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

[#ref-cli-cdk-migrate-options-language] --language <STRING>:: The programming language to use for the CDK project created during migration. + *Required*: No + *Valid values*: typescript, python, java, csharp, go. + *Default*: typescript

[#ref-cli-cdk-migrate-options-output-path] --output-path <PATH>:: The output path for the migrated CDK project. + *Required*: No + *Default*: By default, the {aws} CDK CLI will use your current working directory.

[#ref-cli-cdk-migrate-options-region] --region <STRING>:: The {aws} Region to retrieve the {aws} CloudFormation stack template from. + *Required*: No + *Default*: The {aws} CDK CLI obtains {aws} Region information from default sources.

[#ref-cli-cdk-migrate-examples] == Examples

[#ref-cli-cdk-migrate-examples-1] === Simple example of migrating from a CloudFormation stack

Migrate from a deployed CloudFormation stack in a specific {aws} environment using --from-stack. Provide --stack-name to name your new CDK stack. The following is an example that migrates myCloudFormationStack to a new CDK app that is using TypeScript:

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

## $ cdk migrate –language typescript –from-stack –stack-name ‘myCloudFormationStack’

[#ref-cli-cdk-migrate-examples-2] === Simple example of migrating from a local CloudFormation template

Migrate from a local JSON or YAML CloudFormation template using --from-path. Provide --stack-name to name your new CDK stack. The following is an example that creates a new CDK app in TypeScript that includes a myCloudFormationStack stack from a local template.json file:

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

## $ cdk migrate –stack-name “myCloudFormationStack” –language typescript –from-path “./template.json”

[#ref-cli-cdk-migrate-examples-3] === Simple example of migrating from deployed {aws} resources

Migrate deployed {aws} resources from a specific {aws} environment that are not associated with a CloudFormation stack using --from-scan. The CDK CLI utilizes the [.noloc]IaC generator service to scan for resources and generate a template. Then, the CDK CLI references the template to create the new CDK app. The following is an example that creates a new CDK app in TypeScript with a new myCloudFormationStack stack containing migrated {aws} resources:

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

## $ cdk migrate –language typescript –from-scan –stack-name “myCloudFormationStack”