:doctype: book

include::attributes.txt[]

// Attributes

[.topic] :info\_titleabbrev: Parameters :keywords: {aws} CDK, {aws} CloudFormation, Parameters

[#parameters] = Parameters and the {aws} CDK

== [abstract]

## *Parameters* are custom values that are supplied at deployment time.

// Content start *Parameters* are custom values that are supplied at deployment time. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html[Parameters] are a feature of {aws} CloudFormation. Since the {aws} Cloud Development Kit ({aws} CDK) synthesizes {aws} CloudFormation templates, it also offers support for deployment-time parameters.

[#parameters-about] == About parameters

Using the {aws} CDK, you can define parameters, which can then be used in the properties of constructs you create. You can also deploy stacks that contain parameters.

When deploying the {aws} CloudFormation template using the {aws} CDK CLI, you provide the parameter values on the command line. If you deploy the template through the {aws} CloudFormation console, you are prompted for the parameter values.

In general, we recommend against using {aws} CloudFormation parameters with the {aws} CDK. The usual ways to pass values into {aws} CDK apps are xref:context[context values] and environment variables. Because they are not available at synthesis time, parameter values cannot be easily used for flow control and other purposes in your CDK app.

= [NOTE]

To do control flow with parameters, you can use https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.CfnCondition.html[CfnCondition] constructs, although this is awkward compared to native if statements.

====

Using parameters requires you to be mindful of how the code you’re writing behaves at deployment time, and also at synthesis time. This makes it harder to understand and reason about your {aws} CDK application, in many cases for little benefit.

Generally, it’s better to have your CDK app accept necessary information in a well-defined way and use it directly to declare constructs in your CDK app. An ideal {aws} CDK–generated {aws} CloudFormation template is concrete, with no values remaining to be specified at deployment time.

There are, however, use cases to which {aws} CloudFormation parameters are uniquely suited. If you have separate teams defining and deploying infrastructure, for example, you can use parameters to make the generated templates more widely useful. Also, because the {aws} CDK supports {aws} CloudFormation parameters, you can use the {aws} CDK with {aws} services that use {aws} CloudFormation templates (such as Service Catalog). These {aws} services use parameters to configure the template that’s being deployed.

[#parameters-learn] == Learn more

For instructions on developing CDK apps with parameters, see xref:get-cfn-param[Use CloudFormation parameters to get a CloudFormation value].