include::attributes.txt[]

// Attributes

[.topic] [[get-context-var,get-context-var.title]] = Save and retrieve context variable values :info\_titleabbrev: Get context value

// Content start

You can specify context variables with the {aws} Cloud Development Kit ({aws} CDK) CLI or in the cdk.json file. Then, use the TryGetContext method to retrieve values.

[#develop-context-specify] == Specify context variables

You can specify a context variable either as part of an {aws} CDK CLI command, or in cdk.json.

To create a command line context variable, use the --context (-c) option, as shown in the following example.

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

## cdk synth -c bucket\_name=mygroovybucket

To specify the same context variable and value in the cdk.json file, use the following code.

== [source,json,subs=“verbatim,attributes”]

{ “context”: { “bucket\_name”: “myotherbucket” } } —

If you specify a context variable using both the {aws} CDK CLI and cdk.json file, the {aws} CDK CLI value takes precedence.

[#develop-context-retrieve] == Retrieve context variable values

To get the value of a context variable in your app, use the TryGetContext method in the context of a construct. (That is, when this, or self in Python, is an instance of some construct.)

In this example, we retrieve the value of the bucket\_name context variable. If the requested value is not defined, TryGetContext returns undefined (None in Python; null in Java and C#; nil in Go) rather than raising an exception.

==== [role=“tablist”] TypeScript:: + [source,javascript,subs=“verbatim,attributes”] — const bucket\_name = this.node.tryGetContext(‘bucket\_name’); —

JavaScript:: + [source,javascript,subs=“verbatim,attributes”] — const bucket\_name = this.node.tryGetContext(‘bucket\_name’); —

Python:: + [source,python,subs=“verbatim,attributes”] — bucket\_name = self.node.try\_get\_context(“bucket\_name”) —

Java:: + [source,java,subs=“verbatim,attributes”] — String bucketName = (String)this.getNode().tryGetContext(“bucket\_name”); —

C#:: + [source,csharp,subs=“verbatim,attributes”] — var bucketName = this.Node.TryGetContext(“bucket\_name”); — ====

Outside the context of a construct, you can access the context variable from the app object, like this.

==== [role=“tablist”] TypeScript:: + [source,javascript,subs=“verbatim,attributes”] — const app = new cdk.App(); const bucket\_name = app.node.tryGetContext(‘bucket\_name’) —

JavaScript:: + [source,javascript,subs=“verbatim,attributes”] — const app = new cdk.App(); const bucket\_name = app.node.tryGetContext(‘bucket\_name’); —

Python:: + [source,python,subs=“verbatim,attributes”] — app = cdk.App() bucket\_name = app.node.try\_get\_context(“bucket\_name”) —

Java:: + [source,java,subs=“verbatim,attributes”] — App app = App(); String bucketName = (String)app.getNode().tryGetContext(“bucket\_name”); —

C#:: + [source,csharp,subs=“verbatim,attributes”] — app = App(); var bucketName = app.Node.TryGetContext(“bucket\_name”); — ====

For more details on working with context variables, see xref:context[Context values and the {aws} CDK].