:doctype: book

include::attributes.txt[]

// Attributes [.topic] [#get-secrets-manager-value] = Get a value from {aws} Secrets Manager :info\_titleabbrev: Get Secrets Manager value

// Content start

To use values from {aws} Secrets Manager in your {aws} CDK app, use the link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_secretsmanager.Secret.html#static-fromwbrsecretwbrattributesscope-id-attrs[fromSecretAttributes()] method. It represents a value that is retrieved from Secrets Manager and used at {aws} CloudFormation deployment time. The following is an example:

==== [role=“tablist”] TypeScript:: + [source,javascript,subs=“verbatim,attributes”] — import \* as sm from “aws-cdk-lib/aws-secretsmanager”;

export class SecretsManagerStack extends cdk.Stack { constructor(scope: cdk.App, id: string, props?: cdk.StackProps) { super(scope, id, props);

const secret = sm.Secret.fromSecretAttributes(this, “ImportedSecret”, { secretCompleteArn: “arn:aws:secretsmanager:::secret:-” // If the secret is encrypted using a KMS-hosted CMK, either import or reference that key: // encryptionKey: … } ); } } —-

JavaScript:: + [source,javascript,subs=“verbatim,attributes”] — const sm = require(“aws-cdk-lib/aws-secretsmanager”);

class SecretsManagerStack extends cdk.Stack { constructor(scope, id, props) { super(scope, id, props);

const secret = sm.Secret.fromSecretAttributes(this, “ImportedSecret”, { secretCompleteArn: “arn:aws:secretsmanager:::secret:-” // If the secret is encrypted using a KMS-hosted CMK, either import or reference that key: // encryptionKey: … }); } }

== module.exports = { SecretsManagerStack }

Python:: + [source,python,subs=“verbatim,attributes”] — import aws\_cdk.aws\_secretsmanager as sm

class SecretsManagerStack(cdk.Stack): def *init*(self, scope: cdk.App, id: str, **kwargs): super().*init*(scope, name,** kwargs)

secret = sm.Secret.from\_secret\_attributes(self, “ImportedSecret”, secret\_complete\_arn=“arn:aws:secretsmanager:::secret:-”, # If the secret is encrypted using a KMS-hosted CMK, either import or reference that key: # encryption\_key=…. ) —-

Java:: + [source,java,subs=“verbatim,attributes”] — import software.amazon.awscdk.services.secretsmanager.Secret; import software.amazon.awscdk.services.secretsmanager.SecretAttributes;

public class SecretsManagerStack extends Stack { public SecretsManagerStack(App scope, String id) { this(scope, id, null); }

public SecretsManagerStack(App scope, String id, StackProps props) { super(scope, id, props);

Secret secret = (Secret)Secret.fromSecretAttributes(this, "ImportedSecret", SecretAttributes.builder()  
 .secretCompleteArn("arn:aws:secretsmanager:<region>:<account-id-number>:secret:<secret-name>-<random-6-characters>")  
 // If the secret is encrypted using a KMS-hosted CMK, either import or reference that key:  
 // .encryptionKey(...)  
 .build());

} } —-

C#:: + [source,csharp,subs=“verbatim,attributes”] — using Amazon.CDK.{aws}.SecretsManager;

public class SecretsManagerStack : Stack { public SecretsManagerStack(App scope, string id, StackProps props) : base(scope, id, props) {

var secret = Secret.FromSecretAttributes(this, "ImportedSecret", new SecretAttributes {  
 SecretCompleteArn = "arn:aws:secretsmanager:<region>:<account-id-number>:secret:<secret-name>-<random-6-characters>"  
 // If the secret is encrypted using a KMS-hosted CMK, either import or reference that key:  
 // encryptionKey = ...,  
 });

} } —- ====

= [TIP]

Use the {aws} CLI link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_secretsmanager.Secret.html[create-secret] CLI command to create a secret from the command line, such as when testing:

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

## aws secretsmanager create-secret –name ImportedSecret –secret-string mygroovybucket

The command returns an ARN that you can use with the preceding example.

====

Once you have created a Secret instance, you can get the secret’s value from the instance’s secretValue attribute. The value is represented by a link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.SecretValue.html[SecretValue] instance, a special type of xref:tokens[Tokens and the {aws} CDK]. Because it’s a token, it has meaning only after resolution. Your CDK app does not need to access its actual value. Instead, the app can pass the SecretValue instance (or its string or numeric representation) to whatever CDK method needs the value.