:doctype: book

include::attributes.txt[]

// Attributes

[.topic] [[get-ssm-value,get-ssm-value.title]] = Get a value from the Systems Manager Parameter Store :info\_titleabbrev: Get SSM value

// Content start

The {aws} Cloud Development Kit ({aws} CDK) can retrieve the value of {aws} Systems Manager Parameter Store attributes. During synthesis, the {aws} CDK produces a xref:tokens[token] that is resolved by {aws} CloudFormation during deployment.

The {aws} CDK supports retrieving both plain and secure values. You may request a specific version of either kind of value. For plain values, you may omit the version from your request to retrieve the latest version. For secure values, you must specify the version when requesting the value of the secure attribute.

= [NOTE]

This topic shows how to read attributes from the {aws} Systems Manager Parameter Store. You can also read secrets from the {aws} Secrets Manager (see xref:get-secrets-manager-value[Get a value from {aws} Secrets Manager]).

====

[#ssm-read-at-deploy] == Read Systems Manager values at deployment time

To read values from the Systems Manager Parameter Store, use the link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_ssm.StringParameter.html#static-valuewbrforwbrstringwbrparameterscope-parametername-version[valueForStringParameter] and link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_ssm.StringParameter.html#static-valuewbrforwbrsecurewbrstringwbrparameterscope-parametername-version[valueForSecureStringParameter] methods. Choose a method based on whether the attribute you want is a plain string or a secure string value. These methods return xref:tokens[tokens], not the actual value. The value is resolved by {aws} CloudFormation during deployment. The following is an example:

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

// Get latest version or specified version of plain string attribute const latestStringToken = ssm.StringParameter.valueForStringParameter( this, ‘my-plain-parameter-name’); // latest version const versionOfStringToken = ssm.StringParameter.valueForStringParameter( this, ‘my-plain-parameter-name’, 1); // version 1

// Get specified version of secure string attribute const secureStringToken = ssm.StringParameter.valueForSecureStringParameter( this, ‘my-secure-parameter-name’, 1); // must specify version —

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

// Get latest version or specified version of plain string attribute const latestStringToken = ssm.StringParameter.valueForStringParameter( this, ‘my-plain-parameter-name’); // latest version const versionOfStringToken = ssm.StringParameter.valueForStringParameter( this, ‘my-plain-parameter-name’, 1); // version 1

// Get specified version of secure string attribute const secureStringToken = ssm.StringParameter.valueForSecureStringParameter( this, ‘my-secure-parameter-name’, 1); // must specify version —

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

= Get latest version or specified version of plain string attribute

latest\_string\_token = ssm.StringParameter.value\_for\_string\_parameter( self, “my-plain-parameter-name”) latest\_string\_token = ssm.StringParameter.value\_for\_string\_parameter( self, “my-plain-parameter-name”, 1)

= Get specified version of secure string attribute

secure\_string\_token = ssm.StringParameter.value\_for\_secure\_string\_parameter( self, “my-secure-parameter-name”, 1) # must specify version —

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

//Get latest version or specified version of plain string attribute String latestStringToken = StringParameter.valueForStringParameter( this, “my-plain-parameter-name”); // latest version String versionOfStringToken = StringParameter.valueForStringParameter( this, “my-plain-parameter-name”, 1); // version 1

//Get specified version of secure string attribute String secureStringToken = StringParameter.valueForSecureStringParameter( this, “my-secure-parameter-name”, 1); // must specify version —

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

// Get latest version or specified version of plain string attribute var latestStringToken = StringParameter.ValueForStringParameter( this, “my-plain-parameter-name”); // latest version var versionOfStringToken = StringParameter.ValueForStringParameter( this, “my-plain-parameter-name”, 1); // version 1

// Get specified version of secure string attribute var secureStringToken = StringParameter.ValueForSecureStringParameter( this, “my-secure-parameter-name”, 1); // must specify version — ====

A link:https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/dynamic-references.html#template-parameters-dynamic-patterns-resources[limited number of {aws} services] currently support this feature.

[#ssm-read-at-synth] == Read Systems Manager values at synthesis time

At times, it’s useful to provide a parameter at synthesis time. By doing this, the {aws} CloudFormation template will always use the same value instead of resolving the value during deployment.

To read a value from the Systems Manager Parameter Store at synthesis time, use the link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_ssm.StringParameter.html#static-valuewbrfromwbrlookupscope-parametername[valueFromLookup] method (Python: value\_from\_lookup). This method returns the actual value of the parameter as a xref:context[Context values and the {aws} CDK] value. If the value is not already cached in cdk.json or passed on the command line, it is retrieved from the current {aws} account. For this reason, the stack *must* be synthesized with explicit {aws} environment information.

The following is an example:

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

== const stringValue = ssm.StringParameter.valueFromLookup(this, ‘my-plain-parameter-name’);

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

== const stringValue = ssm.StringParameter.valueFromLookup(this, ‘my-plain-parameter-name’);

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

== string\_value = ssm.StringParameter.value\_from\_lookup(self, “my-plain-parameter-name”)

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

== String stringValue = StringParameter.valueFromLookup(this, “my-plain-parameter-name”);

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

== var stringValue = StringParameter.ValueFromLookup(this, “my-plain-parameter-name”);

====

Only plain Systems Manager strings may be retrieved. Secure strings cannot be retrieved. The latest version will always be returned. Specific versions cannot be requested.

= [IMPORTANT]

The retrieved value will end up in your synthesized {aws} CloudFormation template. This might be a security risk, depending on who has access to your {aws} CloudFormation templates and what kind of value it is. Generally, don’t use this feature for passwords, keys, or other values you want to keep private.

====

[#ssm-write] == Write values to Systems Manager

You can use the {aws} CLI, the {aws} Management Console, or an {aws} SDK to set Systems Manager parameter values. The following examples use the link:https://docs.aws.amazon.com/cli/latest/reference/ssm/put-parameter.html[ssm put-parameter] CLI command.

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

aws ssm put-parameter –name “parameter-name” –type “String” –value “parameter-value” aws ssm put-parameter –name “secure-parameter-name” –type “SecureString” –value “secure-parameter-value” —

When updating an SSM value that already exists, also include the --overwrite option.

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

aws ssm put-parameter –overwrite –name “parameter-name” –type “String” –value “parameter-value” aws ssm put-parameter –overwrite –name “secure-parameter-name” –type “SecureString” –value “secure-parameter-value” —