:doctype: book

include::attributes.txt[]

// Attributes [.topic] [[usage-data,usage-data.title]] = Configure {aws} CDK usage data reporting :info\_titleabbrev: Configure usage data reporting :keywords: {aws} CDK, Reporting, Usage data, Version reporting

== [abstract]

## The {aws} Cloud Development Kit ({aws} CDK) collects usage data from your CDK applications to gain insight into how the {aws} CDK is being used. The CDK team uses this data to do the following:

// Content start

[#usage-data-intro] == What is CDK usage data reporting?

{aws} Cloud Development Kit ({aws} CDK) applications are configured to collect and report on usage data to gain insight into how the {aws} CDK is being used. The CDK team uses this data to do the following:

* *Communicate with customers* – Identify stacks using a construct with known security or reliability issues and send out communication on customer-impacting topics.
* *Inform CDK development* – Gain insight on CDK usage to better inform CDK development.
* *Investigate CDK issues* – When bugs are reported, usage data provides valuable insight to the CDK team when troubleshooting.

[#usage-data-categories] == What usage data is collected?

There are two categories of usage data collected by the CDK:

* General usage data
* Additional usage data

[#usage-data-categories-general] === General usage data collection

The CDK collects the following types of general usage data from your CDK applications:

* Versions of CDK libraries used.
* Names of constructs used from the following NPM modules:

|  |
| --- |
| \*\* {aws} CDK core modules |
| \*\* {aws} Construct Library modules |
| \*\* {aws} Solutions Constructs module |
| \*\* {aws} Render Farm Deployment Kit module |

= [NOTE]

Before version 1.93.0, the {aws} CDK reported the names and versions of the modules loaded during synthesis, instead of the constructs used in the stack.

====

[#usage-data-categories-additional] === Additional usage data collection

Starting with CDK version 2.178.0, usage data collection expanded to include the following additional usage data:

* *CDK–defined property keys* – When you use a built-in property of an L2 construct, the property key will be collected. This includes built-in property keys nested in dictionary objects.
* *Boolean and enum property values from CDK–defined property keys* – For CDK–defined property keys, property values of only Boolean and enum types will be collected. All other types, such as string values or construct references will be redacted.
* *Method name, keys, and property values of Boolean and enum types* – When you use an L2 construct method, we will collect the method name, property keys, and property values of Boolean and enum types.

For property keys and values that you uniquely create, the entire object will be redacted. For example, if you use +link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_apigateway.InlineApiDefinition.html[InlineApiDefinition]+ to define an [.noloc]OpenAPI specification and pass it into a +link:https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.aws\_apigateway.RestApi.html[RestApi]+ construct, the entire InlineApiDefinition object will be redacted.

For more information on additional usage data collection, including its benefits and potential concerns, see the https://github.com/aws/aws-cdk/discussions/33198[CDK Collecting Additional Metadata (under feature flag) #33198] discussion in the *aws-cdk repository*.

[#usage-data-how] == How the CDK collects usage data

At synthesis, the CDK collects usage data from your application and stores it within the +{aws}::CDK::Metadata+ resource. The following is an example of this on a synthesized {aws} CloudFormation template:

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

CDKMetadata: Type: “{aws}::CDK::Metadata” Properties: Analytics: “v2:deflate64:H4sIAND9SGAAAzXKSw5AMBAA0L1b2PdzBYnEAdio3RglglY60zQi7u6TWL/XKmNUlxeQSOKwaPTBqrNhwEWU3hGHiCzK0dWWfAxoL/Fd8mvk+QkS/0X6BdjnCdgmOOQKWz+AqqLDt2Y3YMnLYWwAAAA=” —

The Analytics property is a gzipped, base64-encoded, prefix-encoded list of the constructs in the stack.

[#usage-data-configure] == How to opt out or opt in to usage data reporting

Your options to opt out or opt in to general usage data reporting and additional usage data reporting depends on the CDK version you used to originally create your app.

By default, CDK applications are configured to automatically opt in to usage data reporting as follows:

* *All CDK apps* – Opted in to general usage data reporting.
* *CDK apps created with a version older than v2.178.0* – Not automatically opted in to additional usage data reporting.
* *CDK apps created with v2.178.0 or newer* – Opted in to additional usage data reporting.

= [WARNING]

If you choose to opt out, the CDK will not be able to identify if you`’ve been impacted by security issues and will not send you notifications for them.

====

[#usage-data-configure-optout-all] === Opt out of all usage data reporting

To opt out of all usage data reporting, you can use the {aws} Cloud Development Kit ({aws} CDK) Command Line Interface (CLI) or configure your project’s cdk.json file.

*To opt out of all usage data reporting using the CDK CLI*:: +

* Use the --no-version-reporting option with any CDK CLI command to opt out for a single command. The following is an example of opting out during template synthesis:

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

## $ cdk synth –no-version-reporting

* Since the CDK synthesizes templates automatically when you run cdk deploy, you should also use --no-version-reporting with the cdk deploy command.

*To opt out of all usage data reporting by configuring the cdk.json file*:: + . Set versionReporting to false in ./cdk.json or ~/.cdk.json. This opts you out by default. The following is an example: + [source,json,subs=“verbatim,attributes”] — { “app”: “…”, “versionReporting”: false } — + . After configuring, you can override this behavior and opt in by specifying --version-reporting on an individual command.

[#usage-data-configure-optout-additiona] === Opt out of only the additional usage data reporting

If your CDK app was created with a CDK version older than 2.178.0, you are automatically opted out of additional usage data reporting, even if you are opted in to general usage data reporting. You don`’t have to do anything to opt out of additional usage data reporting.

If your CDK app was created with CDK version 2.178.0 or newer, you must opt out of all usage data reporting. You can`’t opt out of only the additional usage data reporting.

[#usage-data-configure-optin] === Opt in to usage data reporting

If your CDK app was created with CDK version 2.178.0 or newer, you can opt in to all usage data reporting by setting versionReporting to true. This is the default behavior of CDK apps.

If your CDK app was created with a CDK version older than 2.178.0, you can opt in to general usage data reporting by setting versionReporting to true. To opt in to additional usage data reporting, you must enable a feature flag.

NOTE: These steps are for CDK apps originally created with a version older than 2.178.0

. Verify that you are now using CDK version 2.178.0 or newer. . In your CDK configuration file, specify @aws-cdk/core:enableAdditionalMetadataCollection as true. The following is an example: + [source,json,subs=“verbatim,attributes”] — { “context”: { “@aws-cdk/core:enableAdditionalMetadataCollection”: “true” } } — + . You can also use the ENABLE\_ADDITIONAL\_METADATA\_COLLECTION value with the +https://docs.aws.amazon.com/cdk/api/v2/docs/aws-cdk-lib.FeatureFlags.html[FeatureFlags]+ class. The following is an example: + [source,javascript,subs=“verbatim,attributes”] — import \* as cdk from ‘aws-cdk-lib’; import { FeatureFlags } from ‘aws-cdk-lib’; import \* as cx\_api from ‘aws-cdk-lib/cx-api’; import { Construct } from ‘constructs’;

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

…. // Set the feature flag ENABLE\_ADDITIONAL\_METADATA\_COLLECTION to true FeatureFlags.of(this).add(cx\_api.ENABLE\_ADDITIONAL\_METADATA\_COLLECTION, true);

// Your stack resources go here new cdk.aws\_s3.Bucket(this, ‘MyBucket’); } } ….

const app = new cdk.App(); new MyStack(app, ‘MyStack’); —

[#usage-data-examples] == Examples

[#usage-data-examples-example1] === General and additional usage data collected from a CDK application

The following is an example of a CDK app:

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

import \* as cdk from ‘aws-cdk-lib’; import { Construct } from ‘constructs’; import \* as s3 from ‘aws-cdk-lib/aws-s3’;

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

…. // Create an S3 bucket (L2 construct) const myBucket = new s3.Bucket(this, ‘MyBucket’, { bucketName: ‘my-cdk-example-bucket’, // String type versioned: true, // Boolean type removalPolicy: cdk.RemovalPolicy.DESTROY, // ENUM type lifecycleRules: [{ // Array of object type expirationDate: new Date(‘2019-10-01’), objectSizeLessThan: 600, objectSizeGreaterThan: 500, }], });

// Use a method of the L2 construct to define additional properties myBucket.addLifecycleRule({ id: ‘ExpireOldObjects’, enabled: true, // Boolean expiration: cdk.Duration.days(90), // Expire objects after 90 days }); } } ….

// Define the CDK app and stack const app = new cdk.App(); new MyStack(app, ‘MyStack’); app.synth(); —

At synthesis, usage data is collected, compressed, and stored in the +{aws}::CDK::Metadata+ resource.

The following is an example of general usage data collected with a CDK version older than 2.178.0:

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

{ “fqn”: “aws-cdk-lib.aws-s3.Bucket”, “version”: “v2.170.0” } —

The following is an example of usage data collected, including the additional usage data introduced in CDK version 2.178.0:

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

{ “fqn”: “aws-cdk-lib.aws\_s3.Bucket”, “version”: “2.170.0”, “metadata”: [ { “type”: “aws:cdk:analytics:construct”, “data”: { “bucketName”: “*”, ”versioned”: true, ”removalPolicy”: ”cdk.RemovalPolicy.DESTROY”, ”lifecycleRules”: [ { ”expirationDate”: ””, ”objectSizeLessThan”: ””, ”objectSizeGreaterThan”: ”\_” } ] } }, { ”type”: ”aws:cdk:analytics:method”, ”data”: { ”name”: ”addLifecycleRule”, ”prop”: { ”id”: ”*”, “enabled”: true, “expiration”: “\_“, } } } ] } —