:doctype: book

include::attributes.txt[]

// Attributes

[.topic] [#versioning] = {aws} CDK versioning :info\_titleabbrev: Versioning :keywords: versioning, {aws} CDK reference

== [abstract]

## This topic provides reference information on how the {aws} Cloud Development Kit ({aws} CDK) handles versioning.

// Content start

This topic provides reference information on how the {aws} Cloud Development Kit ({aws} CDK) handles versioning.

Version numbers consist of three numeric version parts: *major*.\_minor\_.\_patch\_, and strictly adhere to the https://semver.org[semantic versioning] model. This means that breaking changes to stable APIs are limited to major releases.

Minor and patch releases are backward compatible. The code written in a previous version with the same major version can be upgraded to a newer version within the same major version. It will also continue to build and run, producing the same output.

[#cdk-toolkit-versioning] == {aws} CDK CLI compatibility

Each version of the main {aws} CDK library (aws-cdk-lib) is compatible with the {aws} CDK CLI (aws-cdk-cli) version that was current at the time of the CDK library’s release. It is also compatible with any newer version of the CDK CLI. Each version of the CDK library maintains this compatibility until the library’s *End of Life* date. Therefore, as long as you’re using a supported CDK library version, it is always safe to upgrade your CDK CLI version.

Each version of the CDK library may also work with CDK CLI versions older than the version that was current at the time of the CDK library’s release. However, this is not guaranteed. Compatibility depends on the CDK library’s cloud assembly schema version. The {aws} CDK generates a cloud assembly during synthesis and the CDK CLI consumes it for deployment. The schema that defines the format of the cloud assembly is strictly specified and versioned. Therefore, an older version of the CDK CLI would need to support the cloud assembly schema version of the CDK library for them to be compatible.

When the cloud assembly version required by the CDK library is not compatible with the version supported by the CDK CLI, you receive an error message like the following:

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

Cloud assembly schema version mismatch: Maximum schema version supported is 3.0.0, but found 4.0.0. Please upgrade your CLI in order to interact with this app. —

To resolve this error, update the CDK CLI to a version compatible with the required cloud assembly version, or to the latest available version. The alternative (downgrading the construct library modules your app uses) is generally not recommended.

= [NOTE]

For more information on the exact combinations of versions that work together, see the https://github.com/aws/aws-cdk-cli/blob/main/COMPATIBILITY.md[compatibility table] in the *aws-cdk-cli GitHub repository*.

====

[#aws-construct-lib-stability] == {aws} Construct Library versioning

The modules in the {aws} Construct Library move through various stages as they are developed from concept to mature API. Different stages offer varying degrees of API stability in subsequent versions of the {aws} CDK.

APIs in the main {aws} CDK library, aws-cdk-lib, are stable, and the library is fully semantically versioned. This package includes {aws} CloudFormation (L1) constructs for all {aws} services and all stable higher-level (L2 and L3) modules. (It also includes the core CDK classes like App and Stack). APIs will not be removed from this package (though they may be deprecated) until the next major release of the CDK. No individual API will ever have breaking changes. When a breaking change is required, an entirely new API will be added.

New APIs under development for a service already incorporated in aws-cdk-lib are identified using a Beta<N> suffix, where N starts at 1 and is incremented with each breaking change to the new API. Beta<N> APIs are never removed, only deprecated, so your existing app continues to work with newer versions of aws-cdk-lib. When the API is deemed stable, a new API without the Beta<N> suffix is added.

When higher-level (L2 or L3) APIs begin to be developed for an {aws} service that previously had only L1 APIs, those APIs are initially distributed in a separate package. The name of such a package has an “Alpha” suffix, and its version matches the first version of aws-cdk-lib it is compatible with, with an alpha sub-version. When the module supports the intended use cases, its APIs are added to aws-cdk-lib.

[#aws-construct-lib-versioning-binding] == Language binding stability

Over time, we might add support to the {aws} CDK for additional programming languages. Although the API described in all the languages is the same, the way that API is expressed varies by language and might change as the language support evolves. For this reason, language bindings are deemed experimental for a time until they are considered ready for production use.

[cols=“1,1”, options=“header”] |=== | Language | Stability

|=== | TypeScript | Stable |===

|=== | JavaScript | Stable |===

|=== | Python | Stable |===

|=== | Java | Stable |===

|=== | C#/.NET | Stable |===

|=== | Go | Stable |===