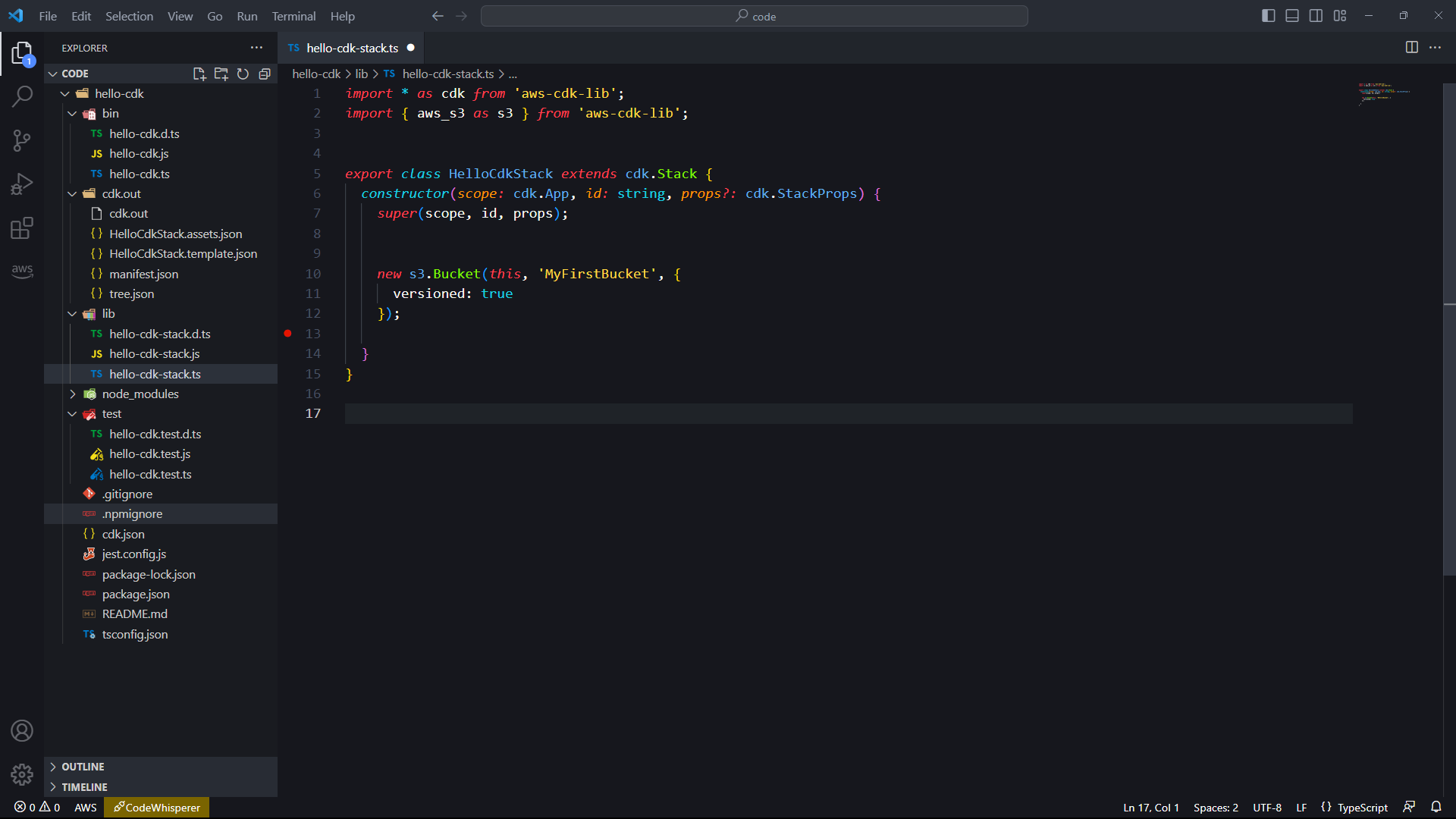
**File system of AWS CDK**

After building a CDK application using TypeScript there is a file structure is created let's take a snap of that file structure.



First folder we got is “bin” bin: This directory may contain executable files that can be used to deploy the stack.

Under this we have also some files namely-

* hello-cdk.d.ts: This is a TypeScript definition file that describes the types and interfaces used in the hello-cdk.js file.
* hello-cdk.js: This is the compiled JavaScript file that contains the code to create the S3 bucket.
* hello-cdk.ts: This is the TypeScript source file that contains the code to create the S3 bucket.

The next or the second folder “ cdk.out” cdk.out- This directory could contain files that describe the output of your AWS CloudFormation stack. The exact files generated depend on the specific resources you create in your stack.

Likewise in the third folder that is “lib” lib: This directory may contain additional TypeScript files that define other constructs used in the stack.

also inside this there is some files :-

* hello-cdk-stack.d.ts: This is a TypeScript definition file that describes the types and interfaces used in the hello-cdk-stack.js file.
* hello-cdk-stack.js: This is the compiled JavaScript file that contains the code to define the stack and its resources.
* hello-cdk-stack.ts: This is the TypeScript source file that contains the code to define the stack and its resources.

**Working of aws-cdk**

There is an example of an executable file that will explain that how the code is working under each circumstances and how the objects behave with their parameter.



**There is a code snippet using TypeScript AWS CDK (Cloud Development Kit) stack that creates an S3 bucket.**

The first line imports the aws-cdk-lib package and aliases it as cdk. This package provides the core classes and utilities for using the AWS CDK.

The second line imports the aws\_s3 module from the aws-cdk-lib package and aliases it as s3. This module provides the classes and interfaces for working with Amazon S3.

The HelloCdkStack class extends the cdk.Stack class, which represents a CloudFormation stack that contains AWS resources.

The constructor method of the HelloCdkStack class creates a new S3 bucket with versioning enabled. It takes three parameters: scope, id, and props. The scope parameter is a reference to the cdk.App object that contains the stack. The id parameter is a unique identifier for the stack. The props parameter is an optional set of properties that can be used to configure the stack.

The super method calls the constructor of the cdk.Stack class to initialize the stack.

The new s3.Bucket statement creates a new instance of the s3.Bucket class, which represents an Amazon S3 bucket. It takes two parameters: the scope object (in this case, the current HelloCdkStack object), and a string identifier for the bucket (MyFirstBucket).

The versioned: true option specifies that versioning should be enabled for the bucket.That allows us to keep the multiple version file of an objects in the same bucket.