[#cdk-pipeline-init] == Initialize a project

Create a new, empty GitHub project and clone it to your workstation in the my-pipeline directory. (Our code examples in this topic use GitHub. You can also use {aws} CodeStar or {aws} CodeCommit.)

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

git clone ++++++my-pipeline cd my-pipeline —-++++++

= [NOTE]

You can use a name other than my-pipeline for your app’s main directory. However, if you do so, you will have to tweak the file and class names later in this topic. This is because the {aws} CDK Toolkit bases some file and class names on the name of the main directory.

====

After cloning, initialize the project as usual.

==== [role=“tablist”] TypeScript:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language typescript —

JavaScript:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language javascript —

Python:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language python — + After the app has been created, also enter the following two commands. These activate the app’s Python virtual environment and install the {aws} CDK core dependencies. + [source,none,subs=“verbatim,attributes”] — $ source .venv/bin/activate # On Windows, run .\venv\Scripts\activate instead $ python -m pip install -r requirements.txt —

Java:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language java — + If you are using an IDE, you can now open or import the project. In Eclipse, for example, choose *File* > *Import* > *Maven* > *Existing Maven Projects*. Make sure that the project settings are set to use Java 8 (1.8).

C#:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language csharp — + If you are using Visual Studio, open the solution file in the src directory.

Go:: + [source,none,subs=“verbatim,attributes”] — $ cdk init app –language go — + After the app has been created, also enter the following command to install the {aws} Construct Library modules that the app requires. + [source,none,subs=“verbatim,attributes”] — $ go get — ====

= [IMPORTANT]

Be sure to commit your cdk.json and cdk.context.json files to source control. The context information (such as feature flags and cached values retrieved from your {aws} account) are part of your project’s state. The values may be different in another environment, which can cause unexpected changes in your results. For more information, see xref:context[Context values and the {aws} CDK].

====