**Continuous Delivery Pipeline**

[GitHub | Beanstalk | CodeBuild | Pipeline]

**Introduction**

Set up a Github repo, deploy a sample web app, and create a continuous delivery pipeline

**Project Description – Scenario**

In this project I will create a continuous delivery pipeline for a simple web application. I will use version control system to store the source code then create a continuous delivery pipeline that will automatically deploy my web application whenever my source code is updated.

**Architecture - Overview**

A screenshot of a computer

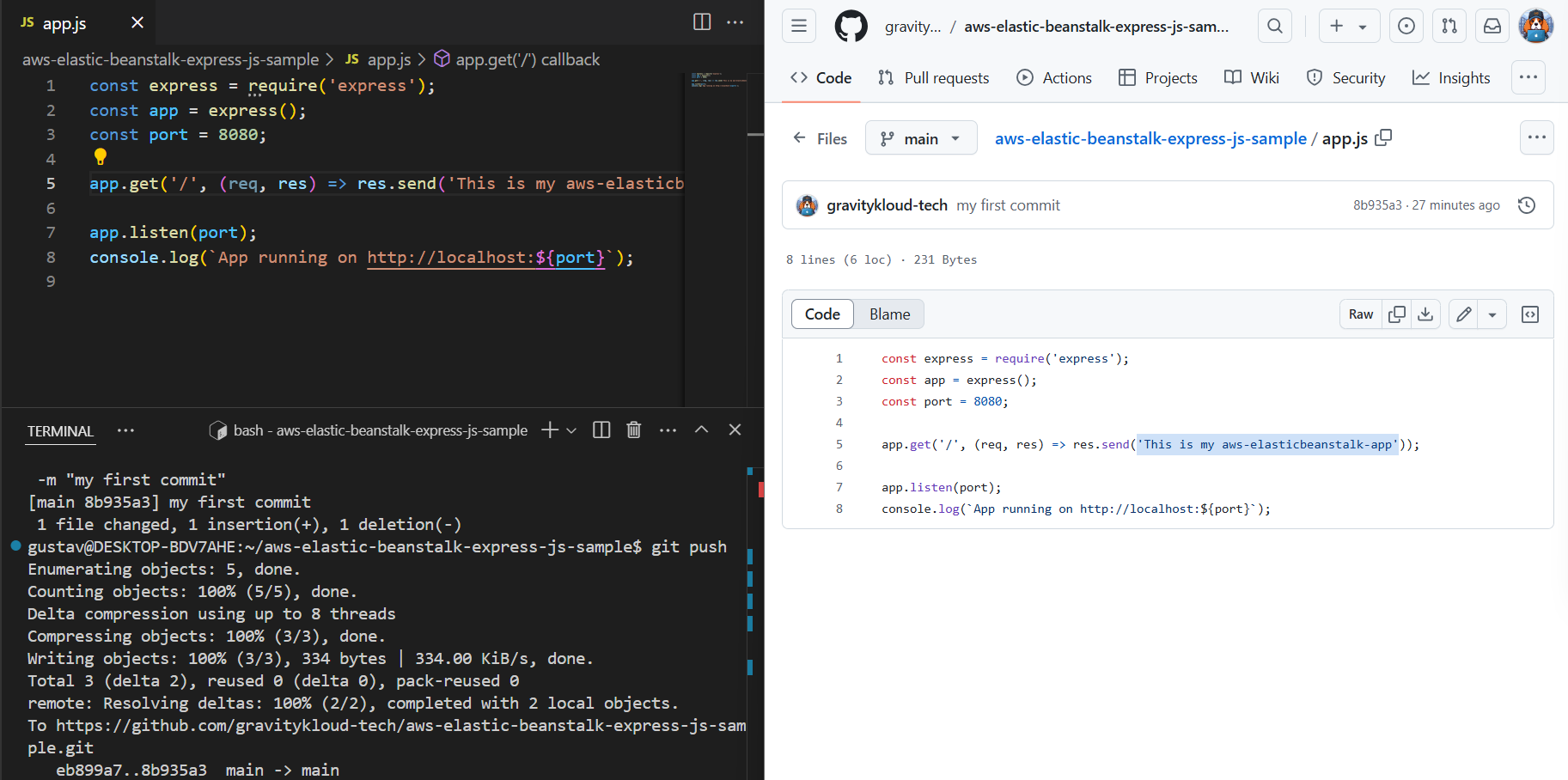
Description automatically generated

* Set up a GitHub repository for the application code.
* Create an AWS Elastic Beanstalk environment to deploy the application.
* Configure AWS CodeBuild to build the source code from GitHub.
* Use AWS CodePipeline to set up the continuous delivery pipeline with source, build, and deploy stages.

**Github**

I will set up a repository for my code so it can be easily accessed over the Internet.

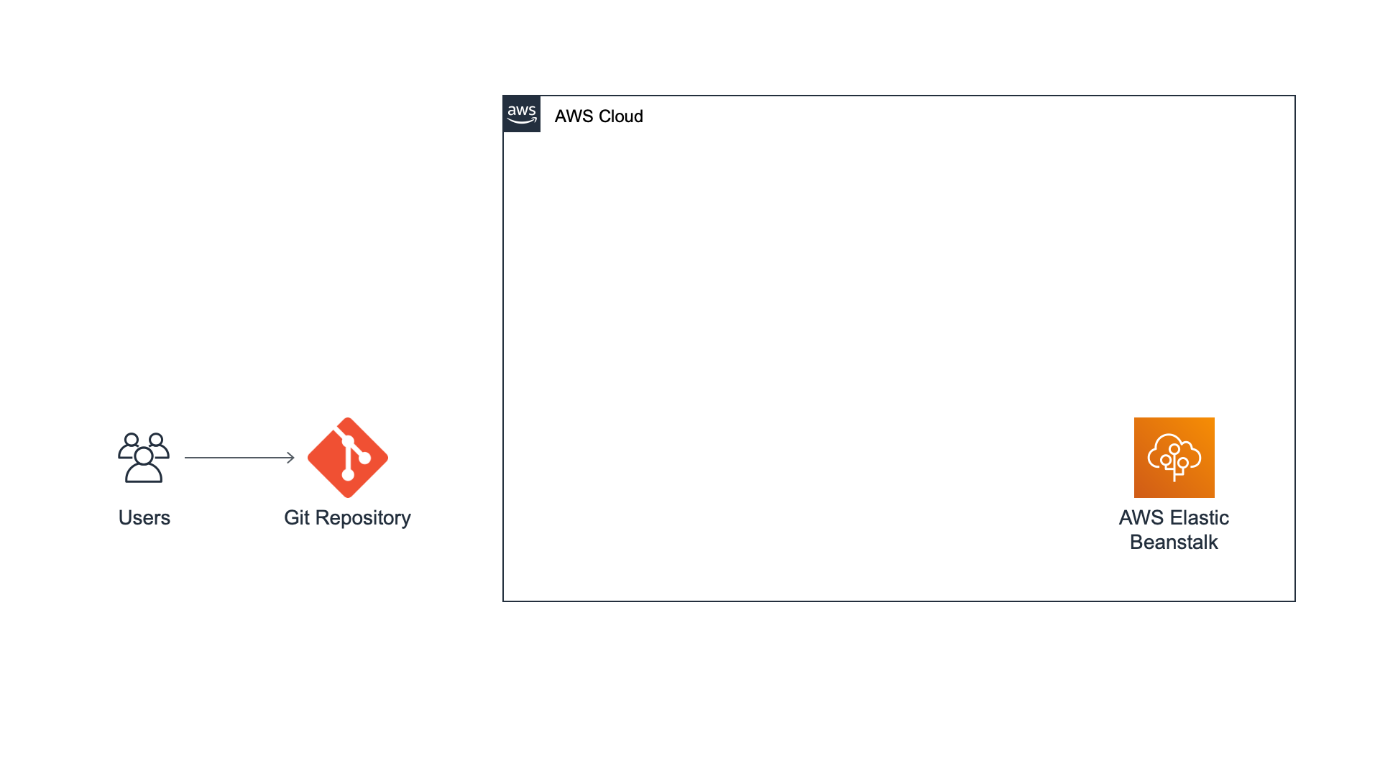
* Fork a GitHub repository to create a new one.
* Store code and metadata in GitHub.
* Interact with a code repository using Git.



**Deploy Web App**

I will create an application and launch environment with the AWS Elastic Beanstalk service to deploy a web application, the same environment will be used for the continuous delivery pipeline to deploy the **“This is my aws-elastic beanstalk-app”**

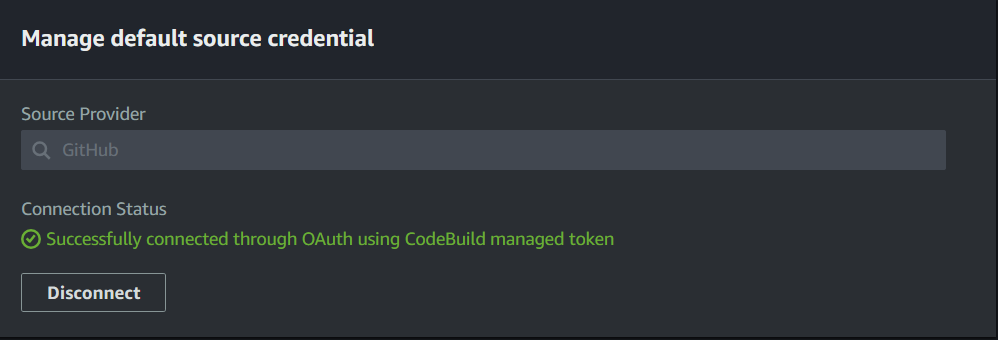
**Current Application Architecture Status**

**Creating the Build project**

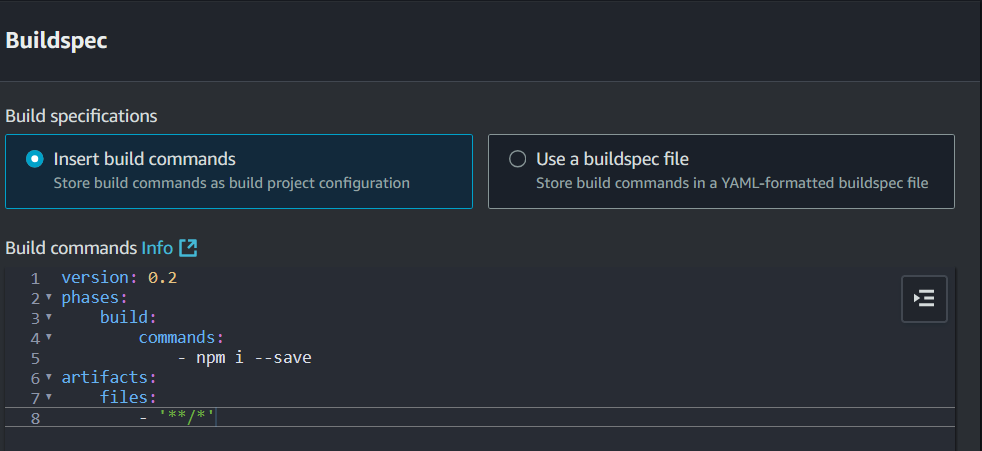
In this section I will use the AWS CodeBuild to build the sources code I preciously stored in my github repository**.**

* Create **a build project** with AWS CodeBuild
* **Build process**-*Process that converts source code files into an executable software artifact. It may include the following steps: compiling source code, running tests, and packaging software for deployment.*
* **Build Project***-Includes information about how to run a build, including where to get the source code, which build environment to use, which build commands to run, and where to store the build output.*
* Set up GitHub as the source provider for a build project
* Run a build on AWS CodeBuild.

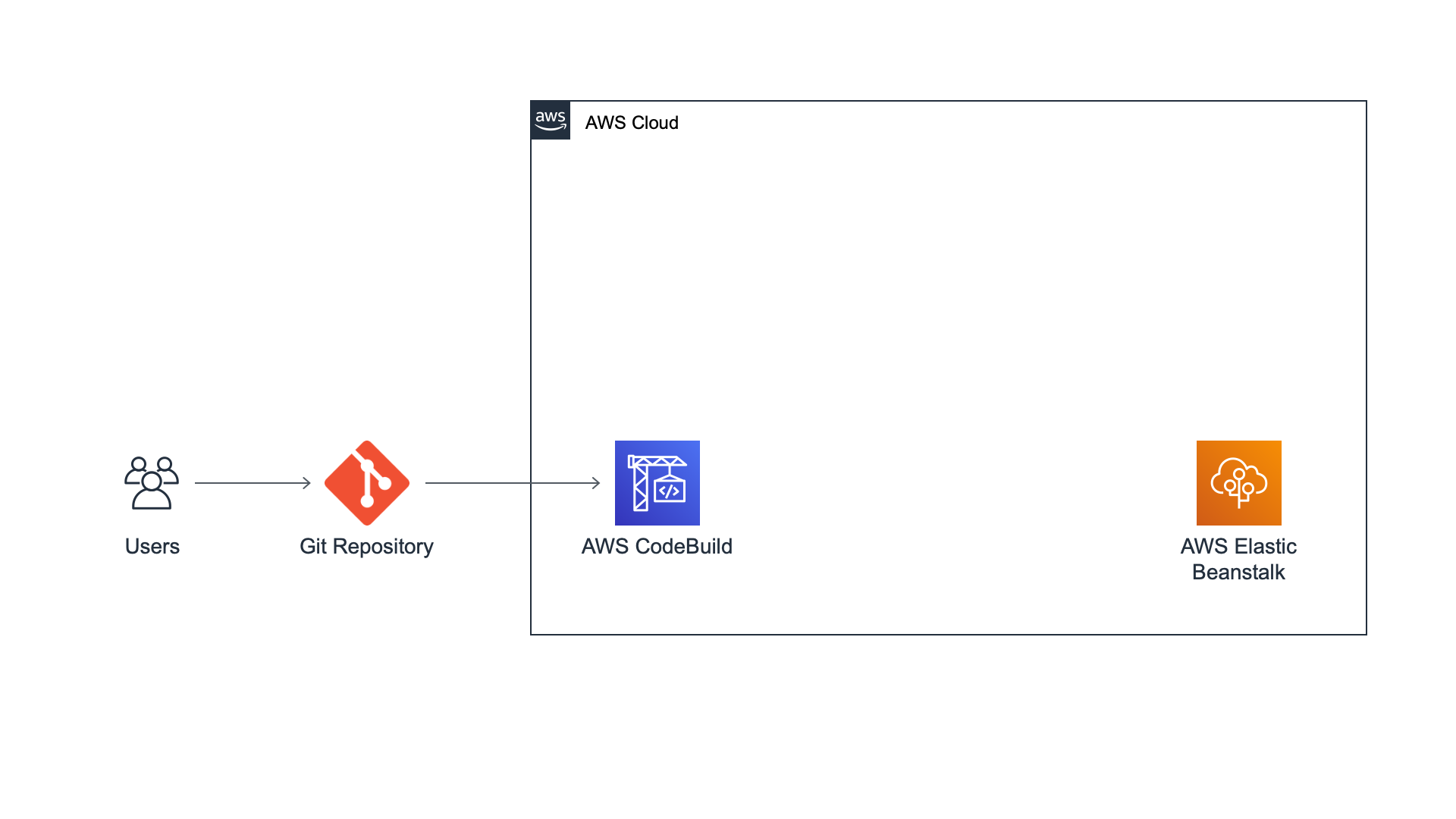
**AWS CodeBuild create project**

****

**Buildspec file for the project**

****

**Current Application architecture Status**



I have created a build project on AWS CodeBuild to run the build process of my **“This is my aws-elastic beanstalk-app”** web app from my GitHub repository.

I will be using this build project as the build step in my continuous delivery pipeline.

**Creating the Continuous Delivery Pipeline [source, build & deploy stages]**

* Set up a continuous delivery **pipeline** on AWS CodePipeline
* **Pipeline***-Workflow model that describes how software changes go through the release process. Each pipeline is made up of a series of stages.*
* Configure a source stage using your GitHub repo
* Configure a build stage using AWS CodeBuild
* Configure a deploy stage using your AWS ElasticBeanstalk application
* Deploy the application hosted on GitHub to Elastic Beanstalk through a pipeline

**A black rectangular object with a white border

Description automatically generated**

**Current Application Architecture status**

A screenshot of a computer

Description automatically generated

I have created a continuous delivery pipeline on AWS CodePipeline with three stages: source, build, and deploy. The source code from the GitHub repo created is part of the source stage. The source code is then built by AWS CodeBuild in the build stage. Finally, the built code is deployed to the AWS Elastic Beanstalk environment.

**Using AWS CodePipline to add a review stage to my continuous delivery pipeline**

Manual approvals are useful to have someone else review a change before deployment. If the action is approved, the pipeline execution resumes. If the action is rejected—or if no one approves or rejects the action within few days—the result is the same as the action failing, and the pipeline execution does not continue.

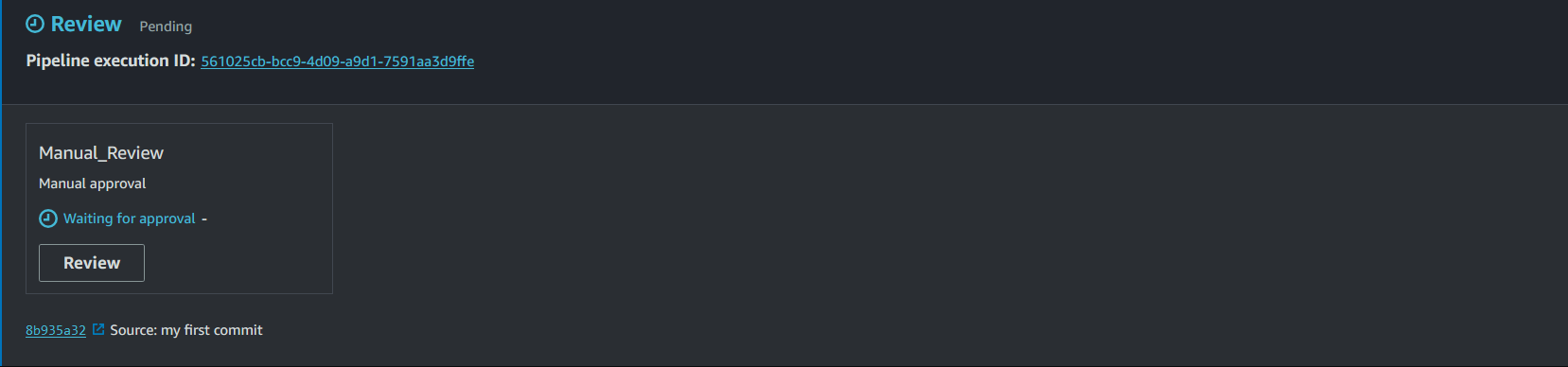
* Add a review stage to pipeline.
* Manually approve a change before it is deployed.
* **Approval action**-*Type of pipeline action that stops the pipeline execution until someone approves or rejects it.*

**App.**

**A screenshot of a computer

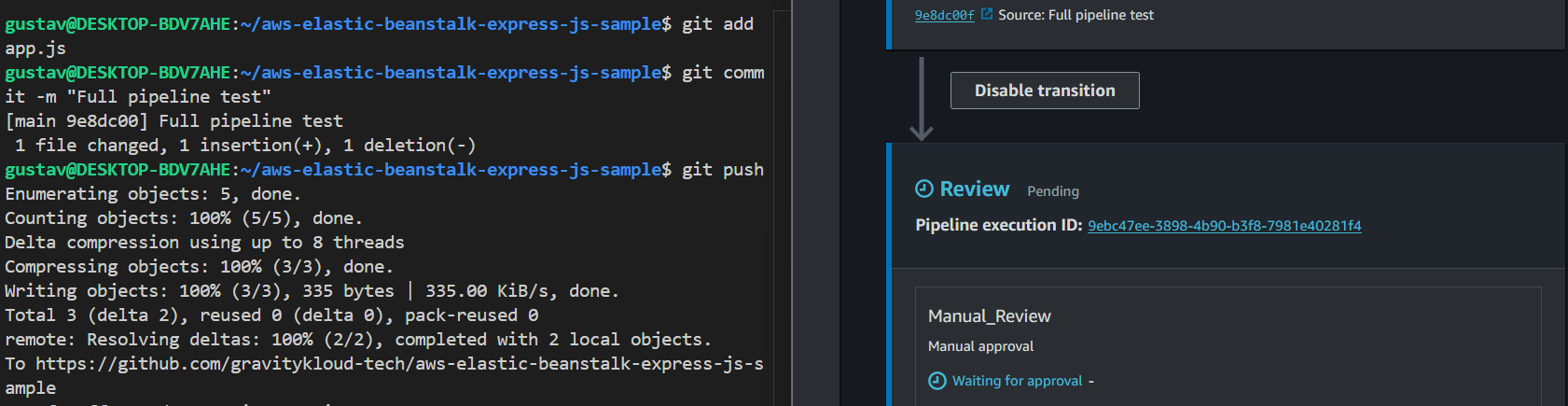
Description automatically generated**

**Adding review stage**



**Pushing a change to my repo & Review for manual approval**

The pipeline detects changes in the code stored in my GitHub repository, building the source code using AWS CodeBuild, and then deploys my application to AWS Elastic Beanstalk.

  
Deployed application hosted on Github to Elastic Beanstalk through the pipeline.

A screenshot of a computer

Description automatically generated

**Done!**