Gautham Srinivasan

IT Analyst - DTCC

01 Nov 2024

AWS CodePipeline for Node.js Application

Project Overview

This documentation details the steps to set up a CI/CD pipeline in AWS CodePipeline for a Node.js application, with source control in GitHub. We'll configure AWS Elastic Beanstalk (EBS) to host the application, integrate AWS CodeBuild for building and testing, and include a manual approval step before deploying changes.

Prerequisites

1. **AWS Account**: Ensure you have access to an AWS account with the necessary permissions to create and manage IAM roles, policies, CodeBuild, and EBS.
2. **GitHub Repository**: Have your Node.js application source code stored in a GitHub repository.
3. **AWS CLI**: (optional) Installed and configured on your local system for easier AWS management.

Procedures

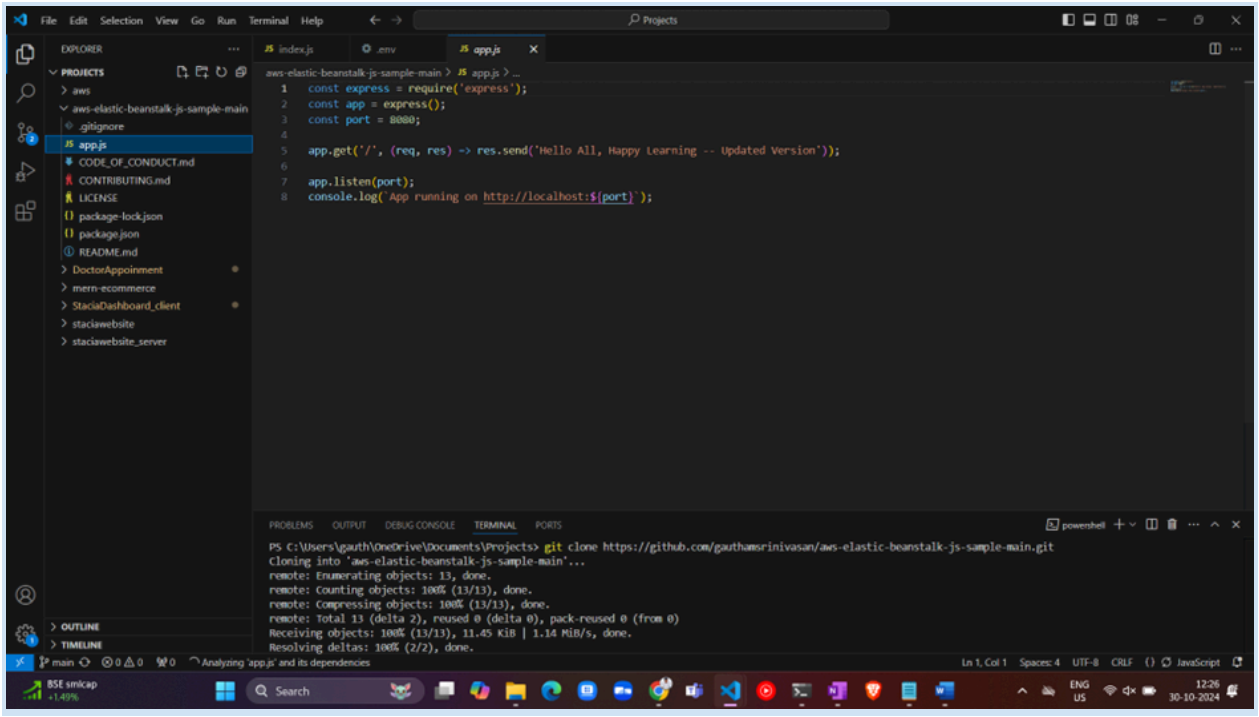
Step 1: Clone the Node.js Application from GitHub

Clone the Repository:

- Clone the Node.js project from GitHub to your local system for testing or configuring.

```
git clone https://github.com/username/repository-name.git

cd repository-name
```

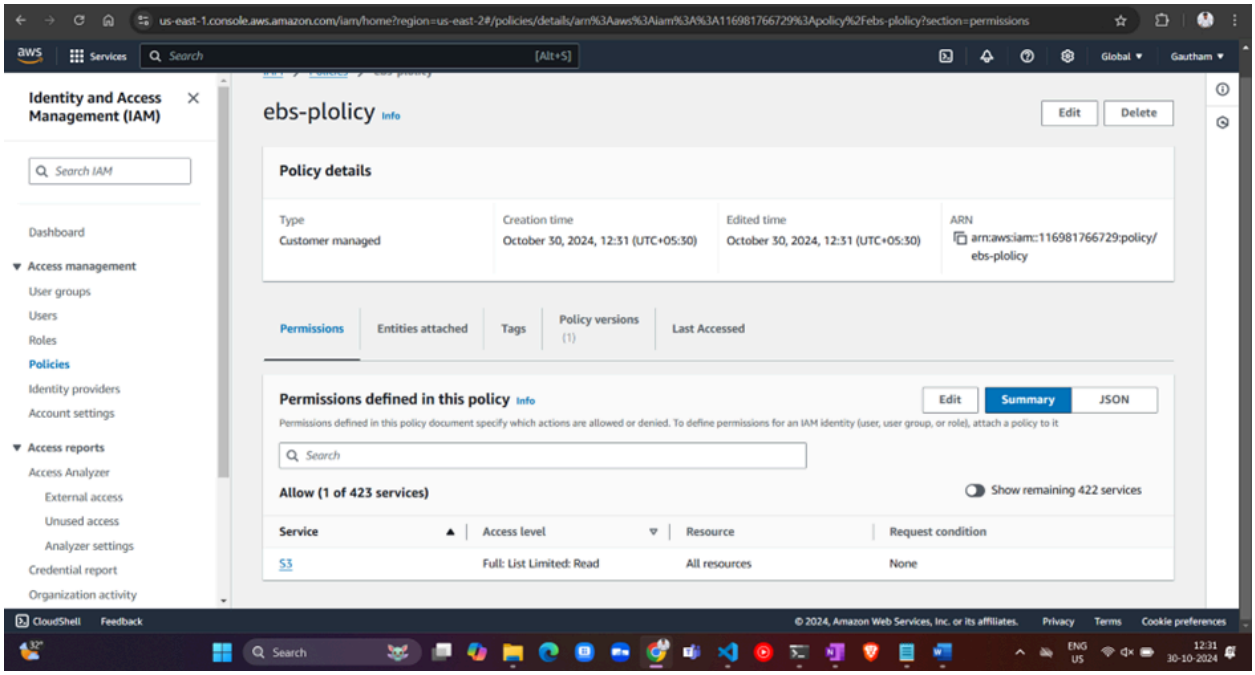


Step 2: Set Up IAM Roles and Policies

Create a New EBS Policy:

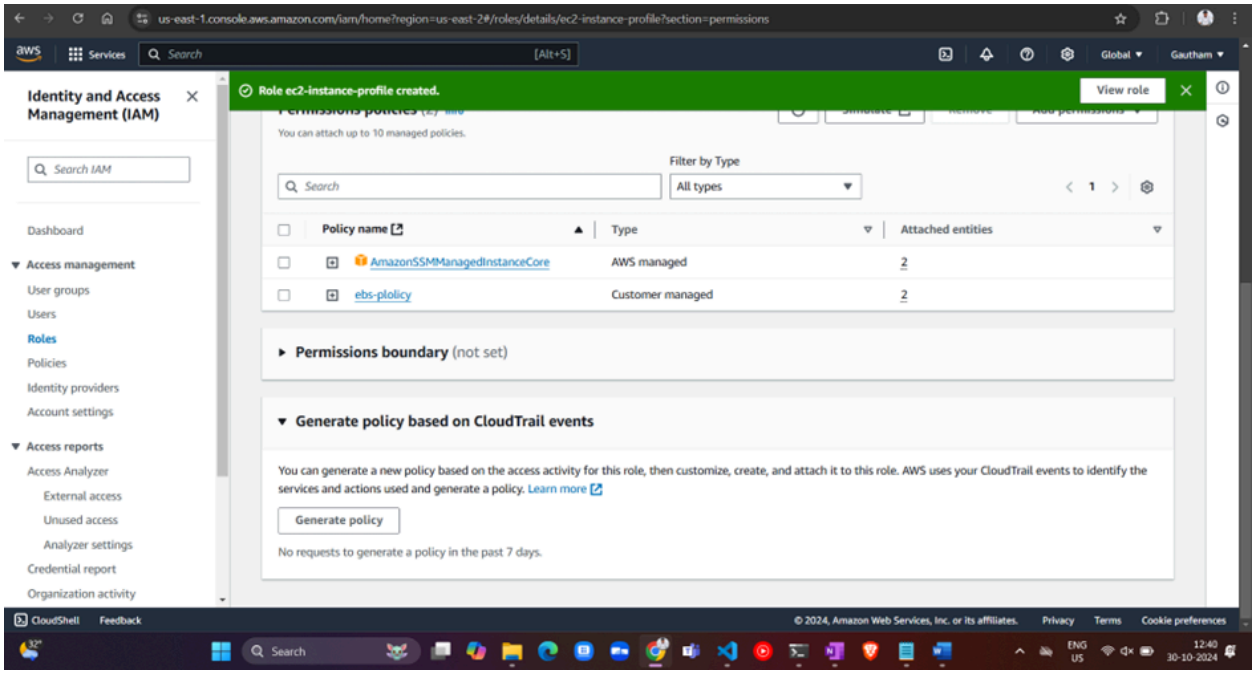
- In **IAM Console** > **Policies** > **Create Policy**.
- Paste the following in the JSON section and click create

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```



Create an IAM Role for EC2:

- Go to **IAM Console** > **Roles** > **Create Role**.
- Choose **EC2** as the trusted entity, and attach:
    - The custom EBS policy from above.
    - **AmazonSSMManagedInstanceCore** policy for EC2 Systems Manager access.



Step 3: Create an Elastic Beanstalk Environment

Go to Elastic Beanstalk Console:

- In the AWS Management Console, navigate to **Elastic Beanstalk**.

Create a New Environment:

- Click **Create environment**.

Configure Environment:

- **Environment Tier**: Select **Web server environment**.
- **Application Name**: Enter `MyDevOps`.
- **Platform**: Choose the **Node.js** platform.

Configure Service Access:

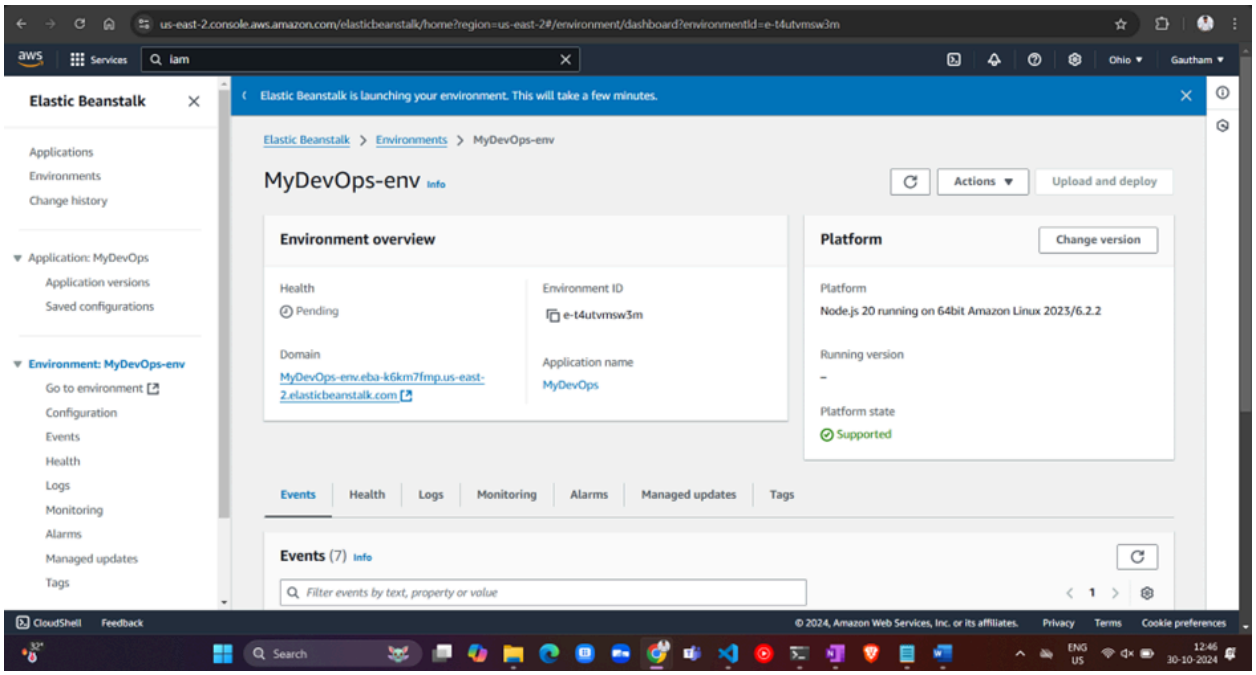
- **Service Role**: Choose **Create and use a new service role**.
- **EC2 Key Pair**: Select your existing EC2 key pair for SSH access.
- **EC2 Instance Profile**: Select the previously created role, `ebs-instance-profile`.
- Click **Next** to proceed.

Instance Traffic and Scaling Configuration:

- **Root Volume Type**: Select **General Purpose SSD (GP3)**.
- **IMDS**: Set **IMDSv2** deactivated (disable instance metadata service).
- Click **Next**.

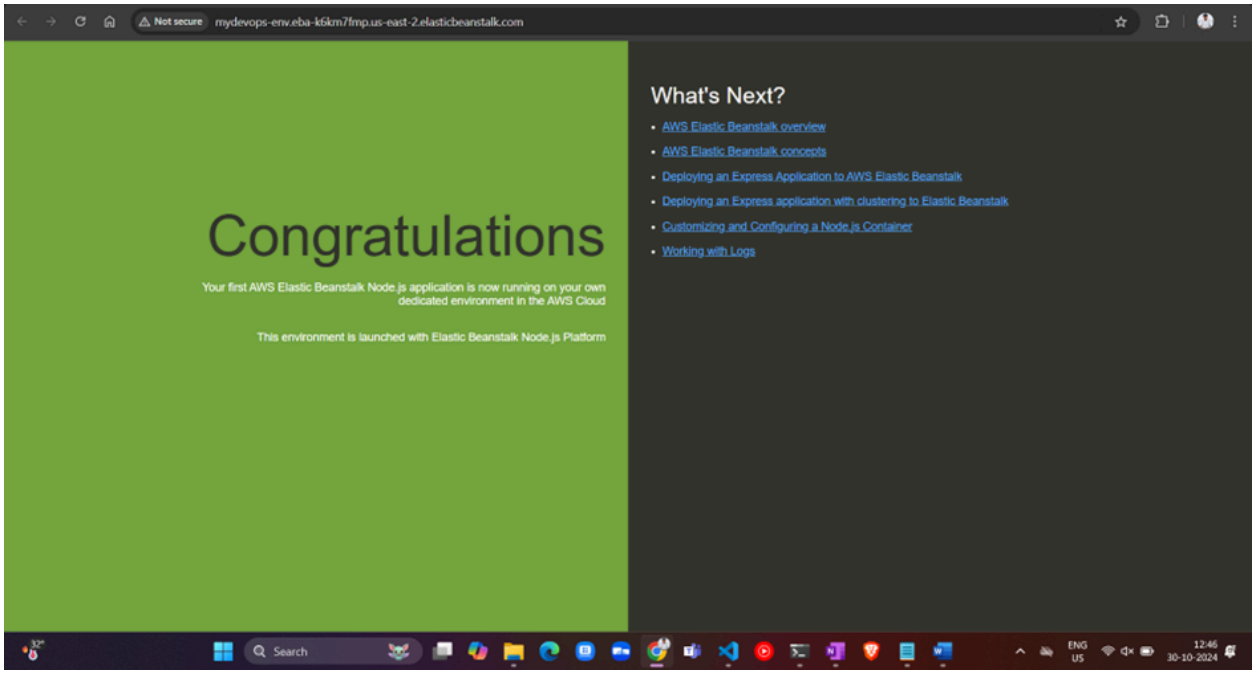
Review and Submit:

- Review your configurations and click **Next**.
- Click **Submit** to create the environment.



Wait for Environment Creation:

- Elastic Beanstalk will take a few minutes to set up the environment. Once it's ready, confirm by checking the provided domain to ensure the environment is accessible and running.



Step 4: Configure AWS CodeBuild

Navigate to CodeBuild:

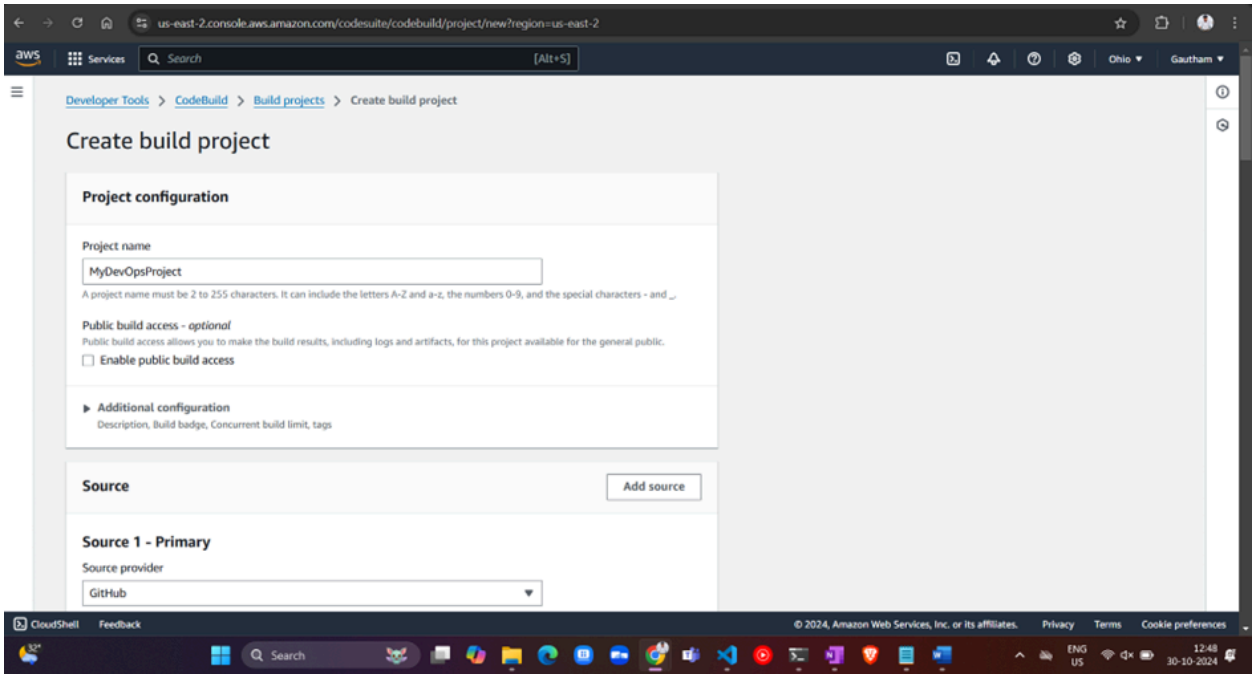
- In the AWS Management Console, go to **Developer Tools** > **CodeBuild** > **Build Projects**.

Create a Build Project:

- Click **Create build project**.

Project Configuration:

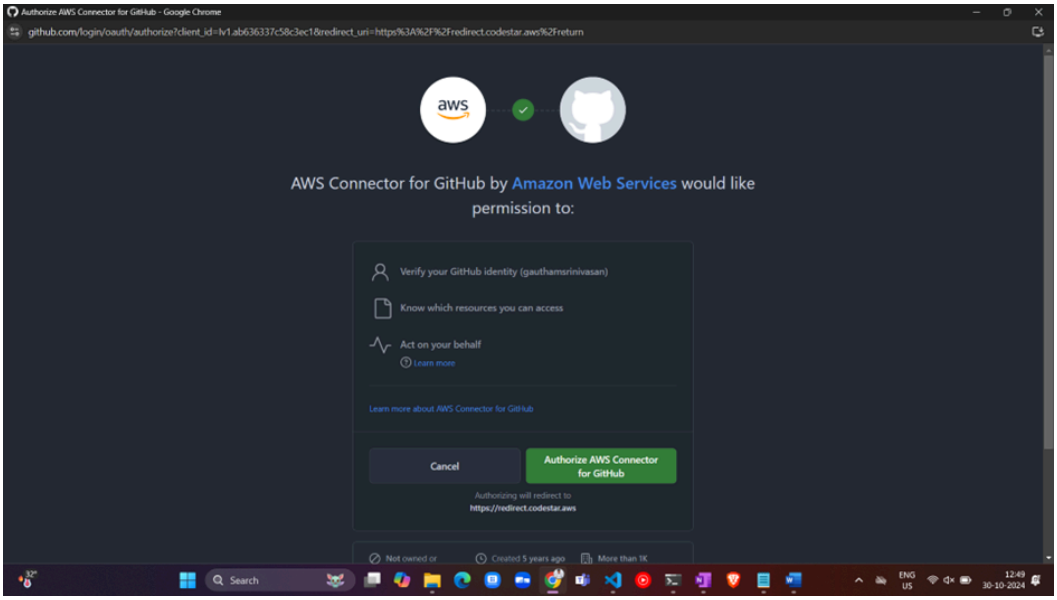
- **Project Name**: Enter `MyDevOpsProject.`



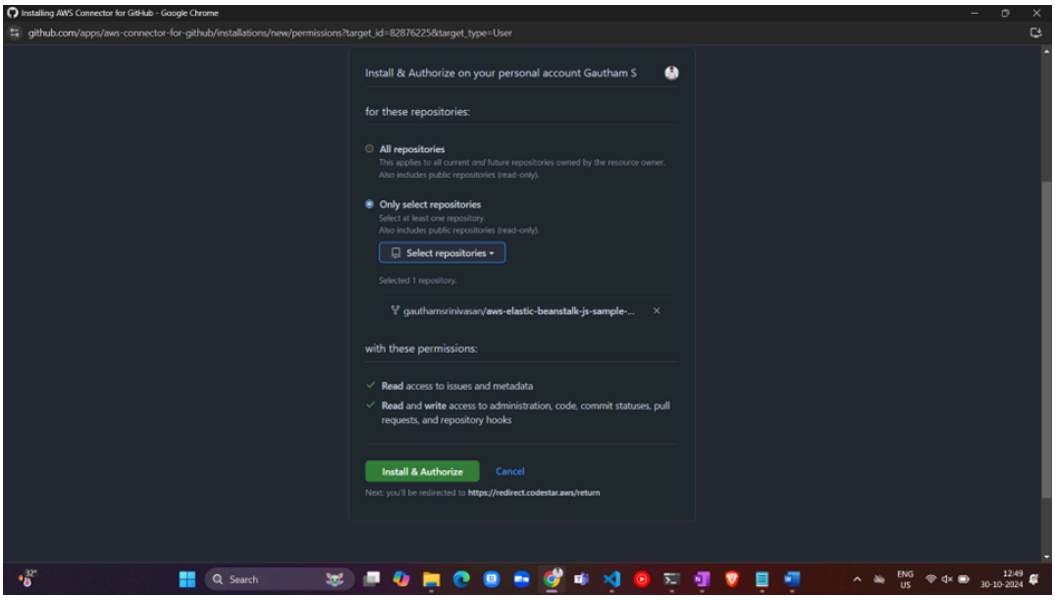
Source Configuration:

- **Source Provider**: Select **GitHub**.
- **Credential**: Choose **Default**.
- Click **Manage default source credential**:
    1. Select **GitHub app** as the connection type.
    2. Click **Create new GitHub connection**.
    3. Name the connection (e.g., `mygithub-connection`).

4. Click **Connect to GitHub** and authorize AWS to access GitHub.



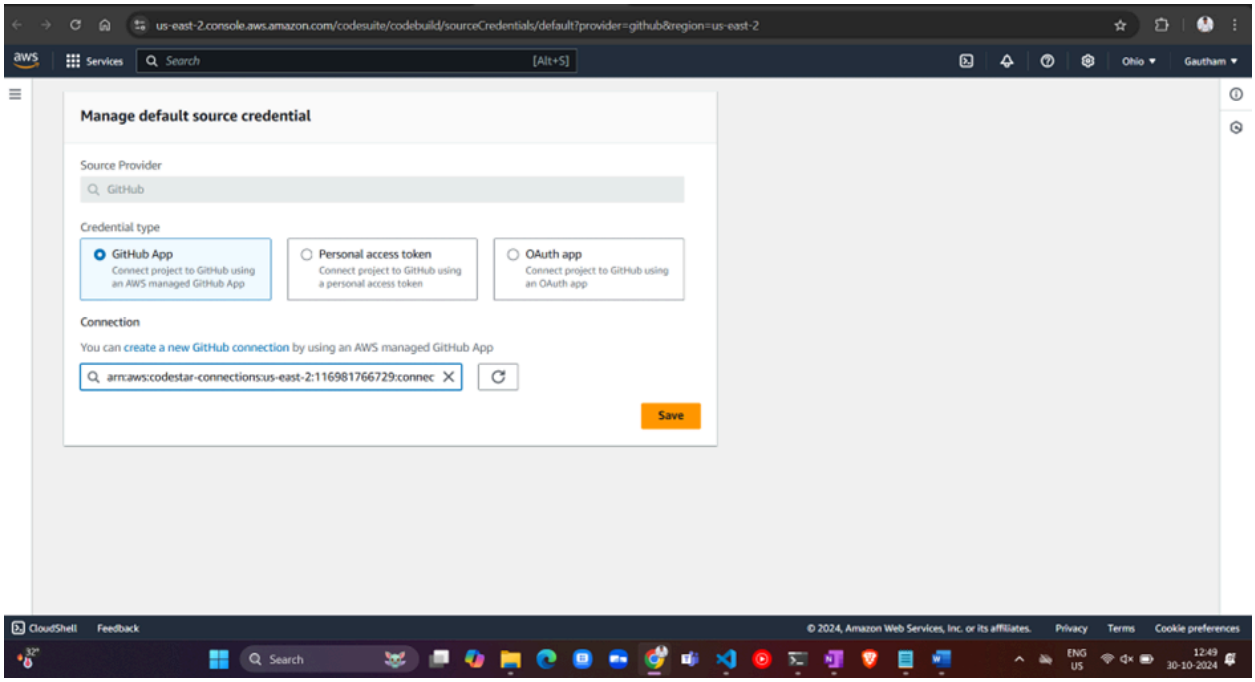
5. Select **Only select repositories**, then choose the repository (e.g., `aws-elastic-beanstalk-js-sample-main`).



6. Click **Install and Authorize**, then select **Connect**.

- Once connected, select your GitHub connection and save.

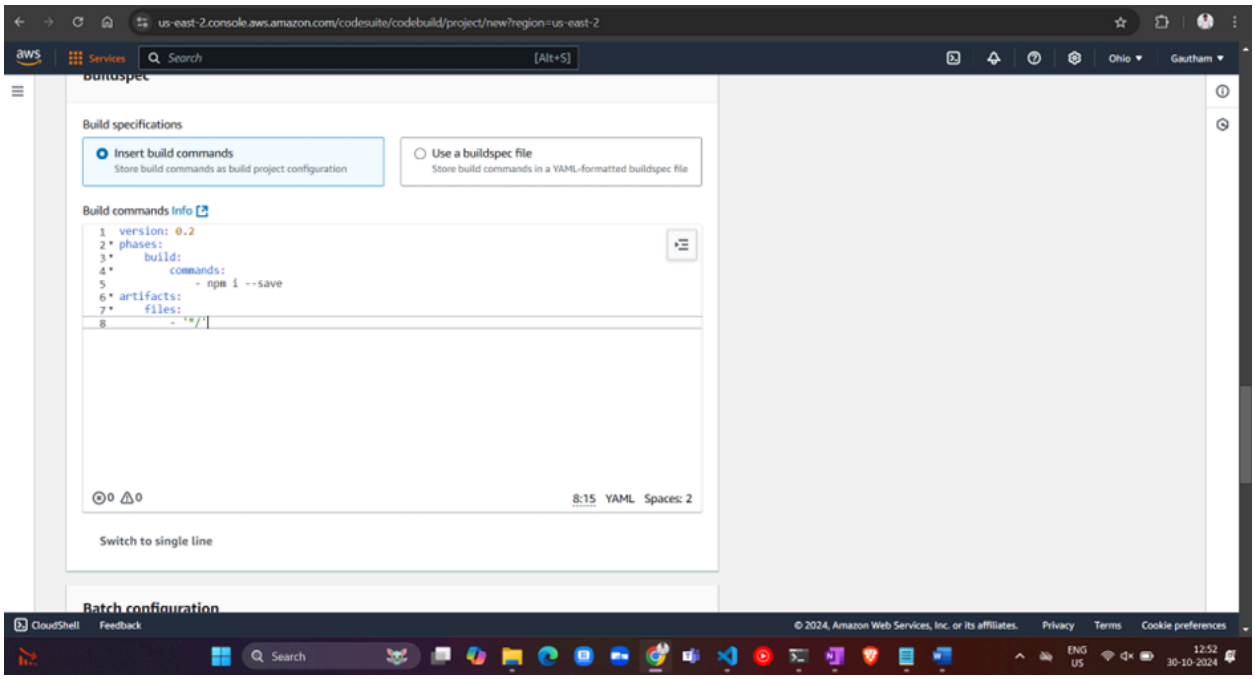
- **Repository**: Choose the repository you connected to.



Build Specification:

- **Buildspec**: Select **Insert build commands** > **Code editor**.
- Paste the following `buildspec.yml` content:

```
version: 0.2
phases:
  install:
    commands:
      - npm i --save
artifacts:
  files:
    - '/*'
```

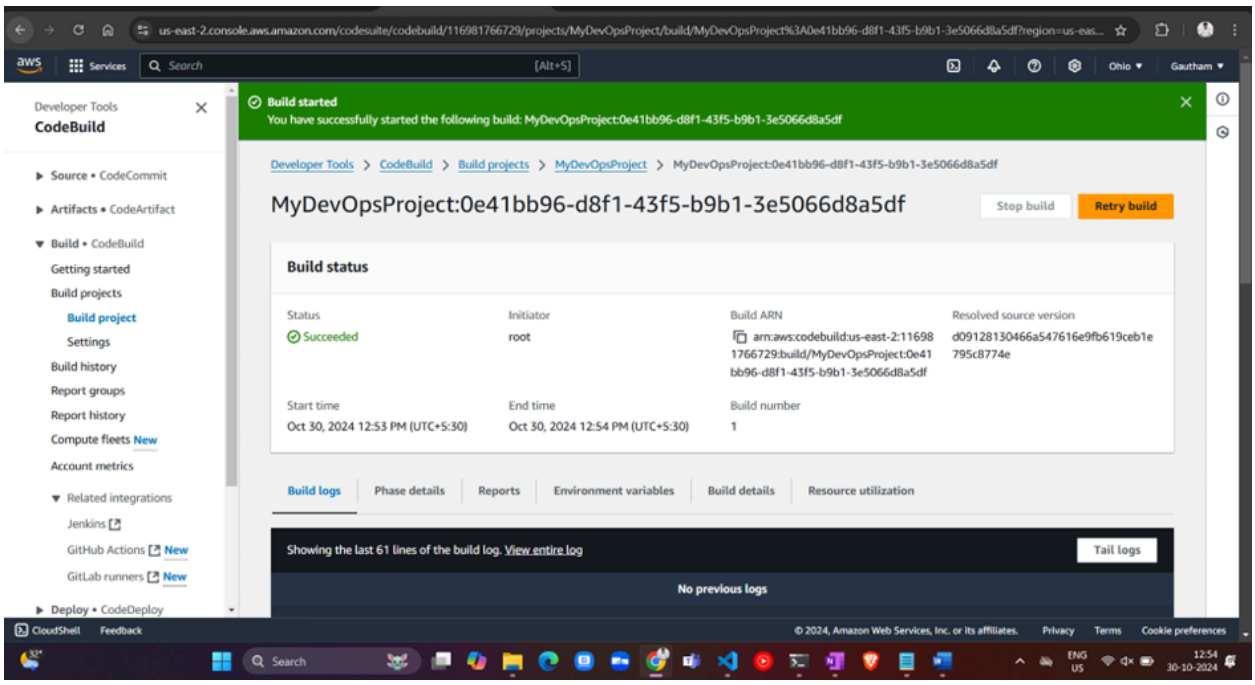


Create Build Project:

- Click **Create build project** to save your configuration.

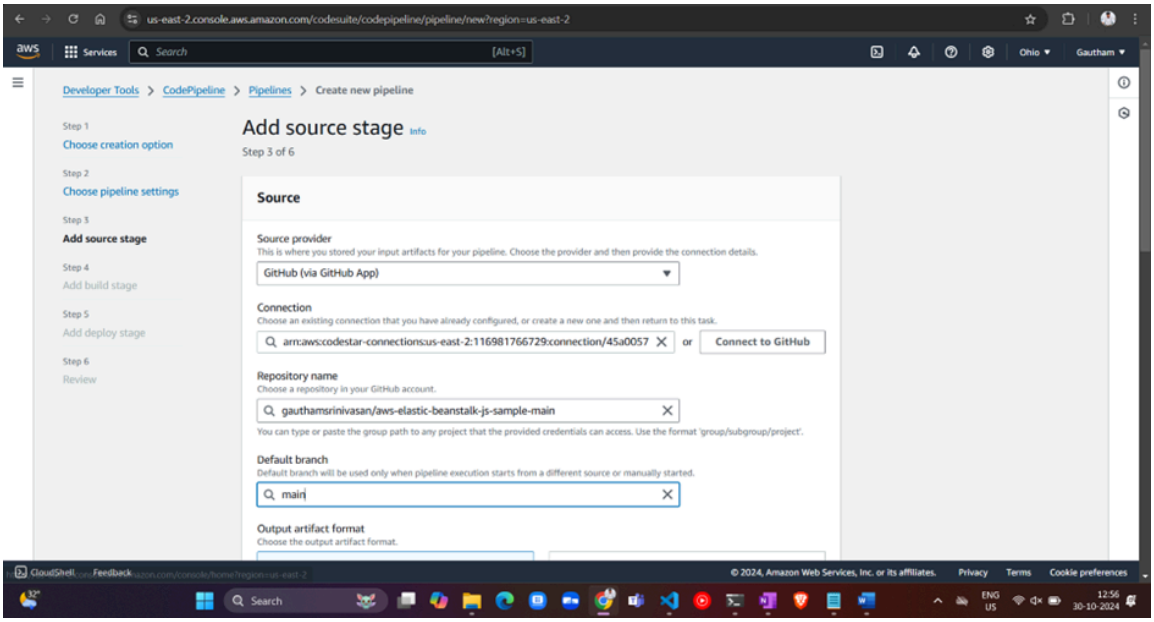
Start Build:

- To initiate a test build, click **Start build** and monitor the build logs to ensure everything is set up correctly.



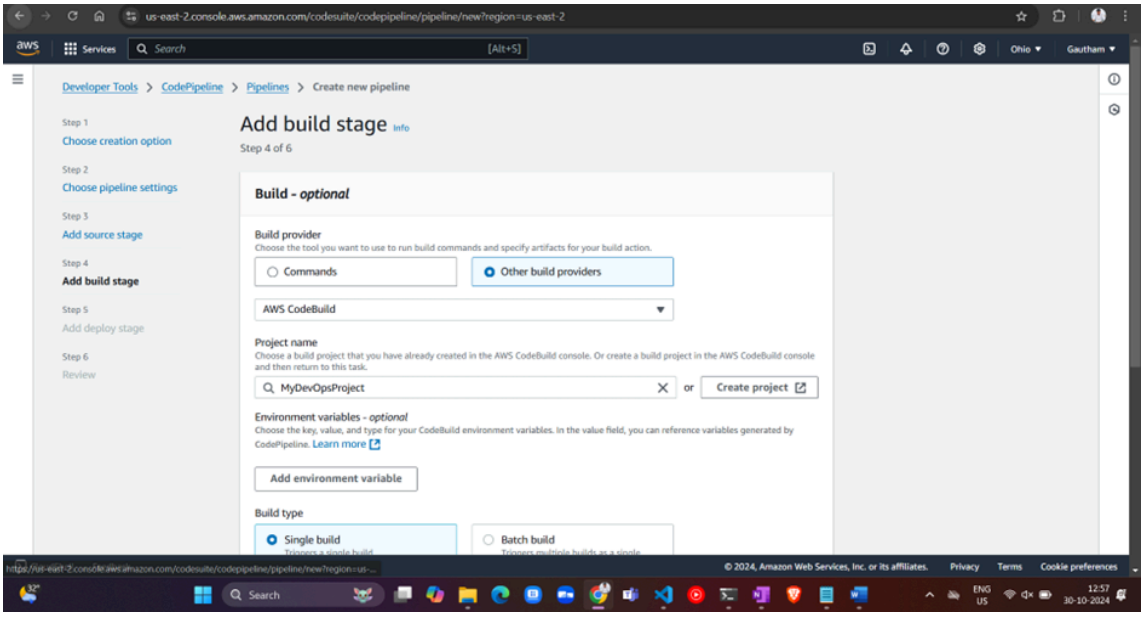
Step 5: Set Up AWS CodePipeline

- **Navigate to CodePipeline**:
  - In the AWS Management Console, go to **Developer Tools** > **CodePipeline** > **Pipelines**.
- **Create Pipeline**:
  - Click **Create pipeline**.
- **Pipeline Settings**:
  - **Pipeline Name**: Enter `MyPipeline`.
  - **Creation Options**: Select **Build custom pipeline**.
  - Click **Next** to proceed.
- **Add Source Stage**:
  - **Source Provider**: Select **GitHub (via app)**.
  - **Connection**: Choose your GitHub connection (`mygithub-connection`).
  - **Repository**: Select the repository where your project is stored.
  - **Branch**: Select `main`.
  - **Trigger**: Select **No filter**.
  - Click **Next** to proceed.

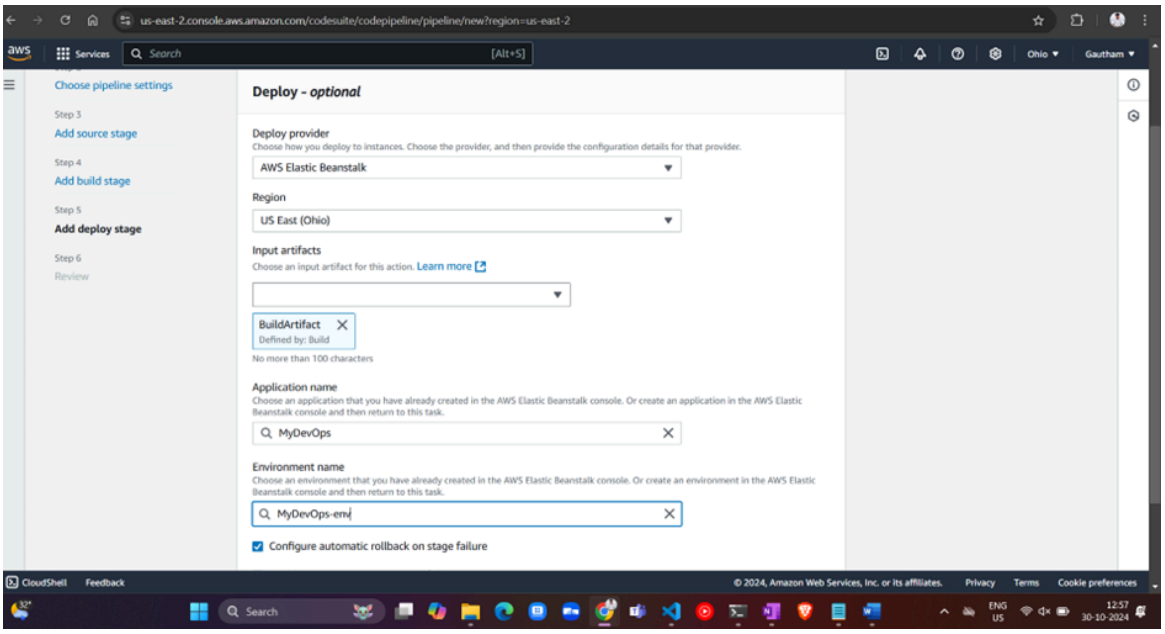


- **Add Build Stage**:
  - **Build Provider**: Select **AWS CodeBuild**.
  - **Project Name**: Choose `MyDevOpsProject`.
  - **Build Type**: Select **Single build**.

- Click **Next**.

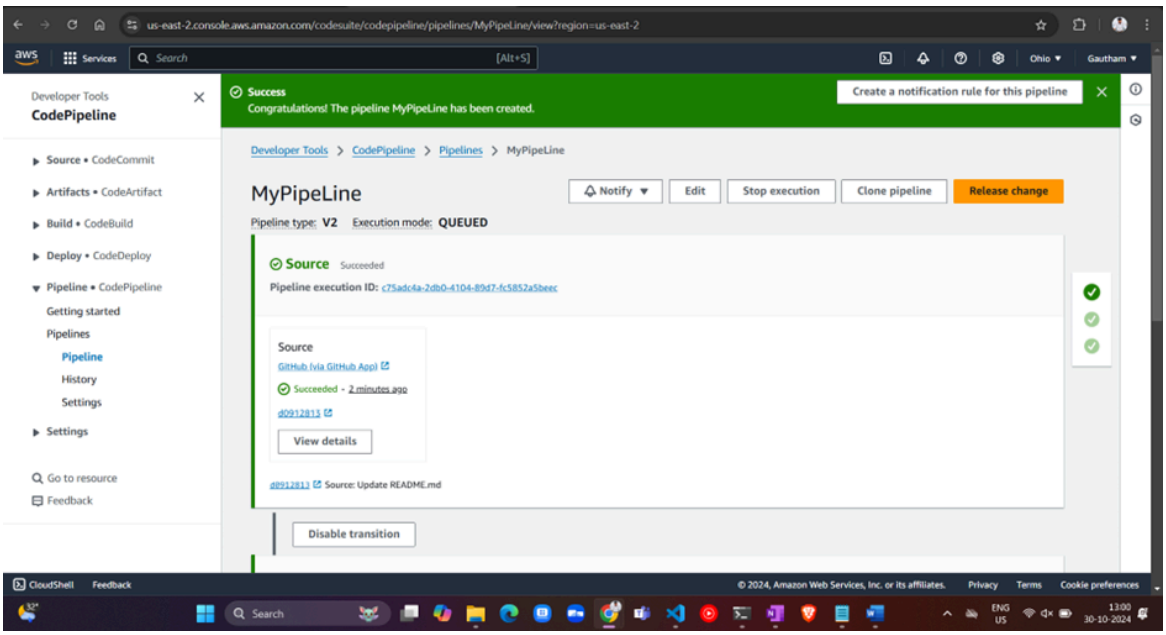


- **Add Deploy Stage**:
  - **Deploy Provider**: Choose **AWS Elastic Beanstalk**.
  - **Application Name**: Enter `MyDevOps`.
  - **Environment Name**: Enter `MyDevOps-env`.
  - Click **Next**.



- **Create Pipeline**:

- Review your settings and click **Create pipeline** to finalize the setup.



Check the output and verify

Step 6: Add a Manual Approval Stage

Edit Pipeline:

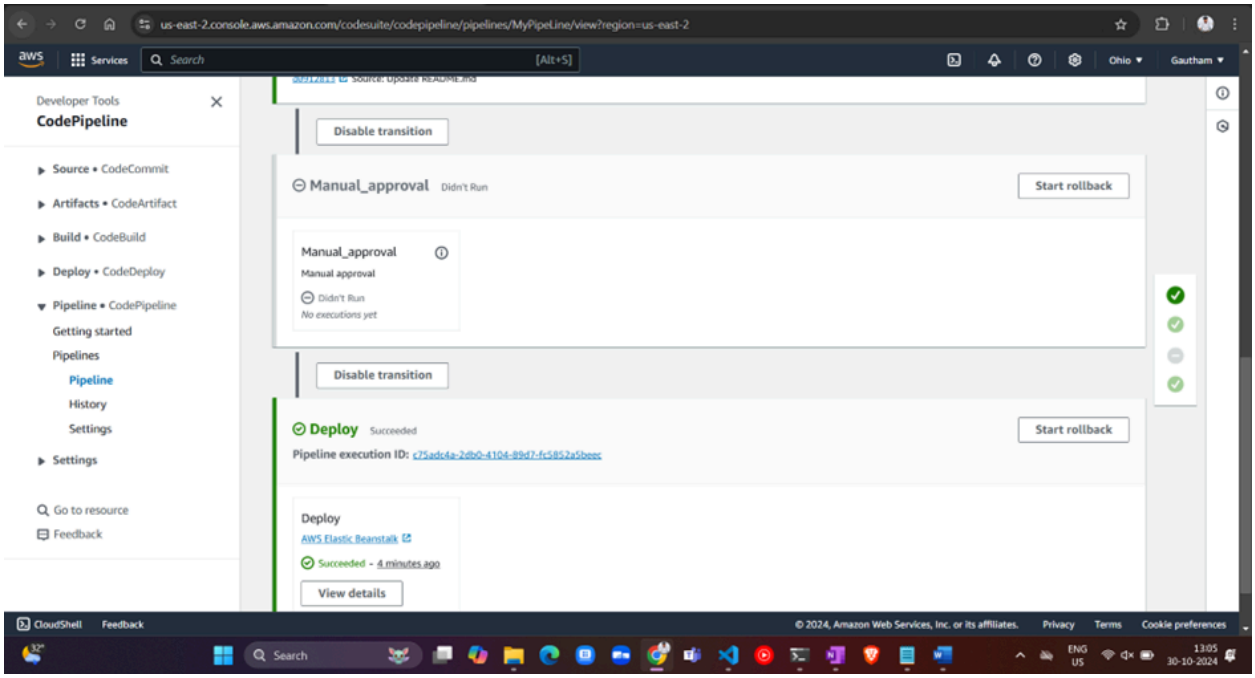
- Go to **Pipelines**, select your newly created pipeline (`MyPipeLine`), and click **Edit**.

Add Review Stage:

- Insert a new stage between **Build** and **Deploy**.
- **Stage Name**: Enter `Review`.

Add Manual Approval Action:

- In the **Review** stage, add an **Action group**.
- **Action Name**: Enter `Manual_Approval`.
- **Action Provider**: Select **Manual approval**.
- Create a topic for SNS and attach in this step with the manager email subscription.
- Click **Done** to save the approval step.



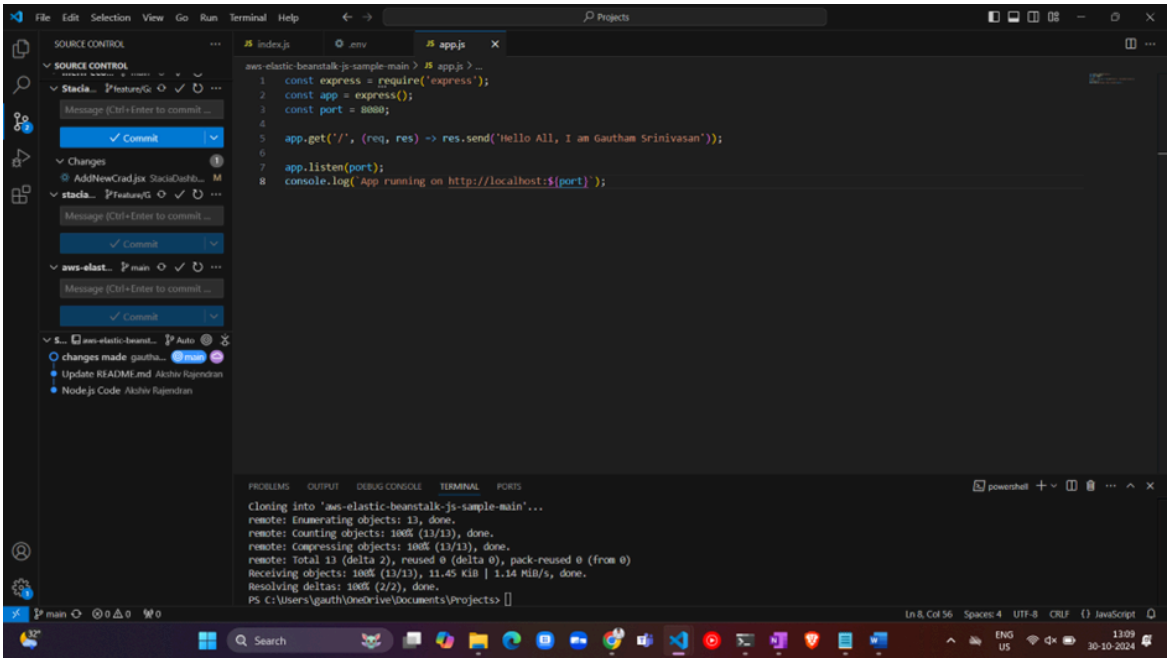
Step 7: Test the Pipeline with Code Changes

- **Modify and Push Code**:
- Open a terminal and navigate to your project directory.
  `cd /path/to/your/project`
    - Make a modification in `app.js` (for example, change a message or function).
- Commit and push the changes to the `main` branch in GitHub:

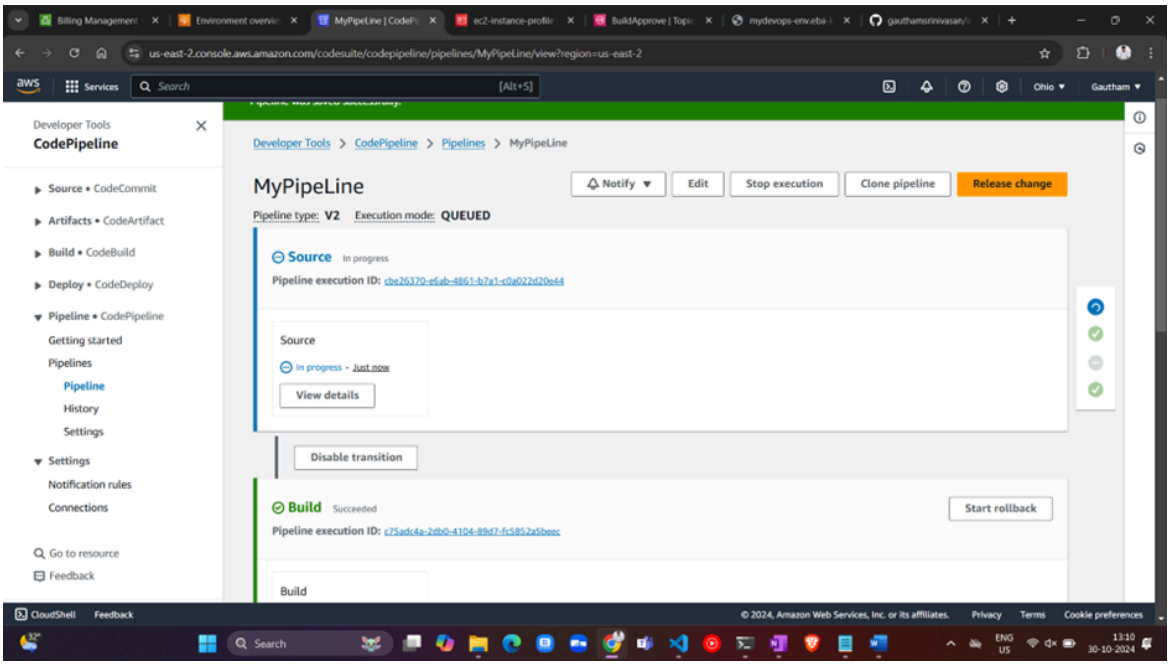
```
git add app.js

git commit -m "Modified app.js"

git push origin main
```

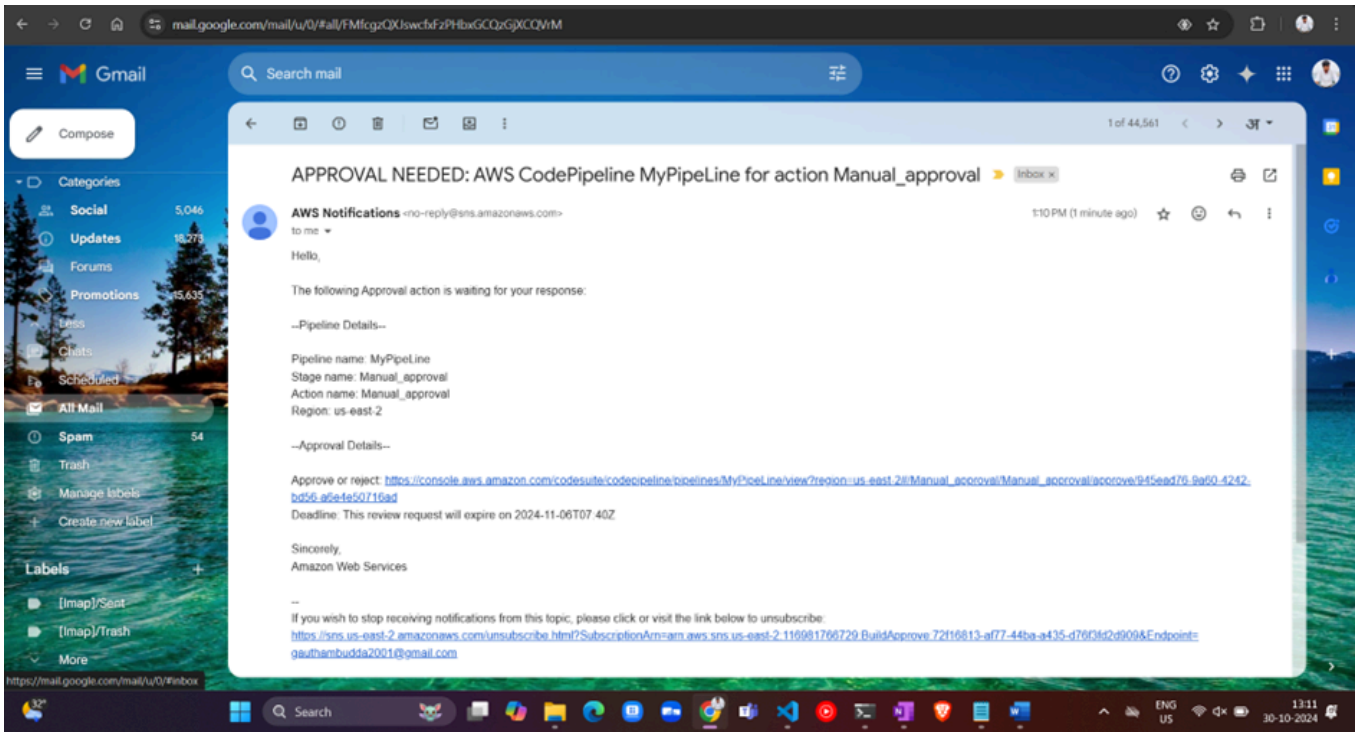


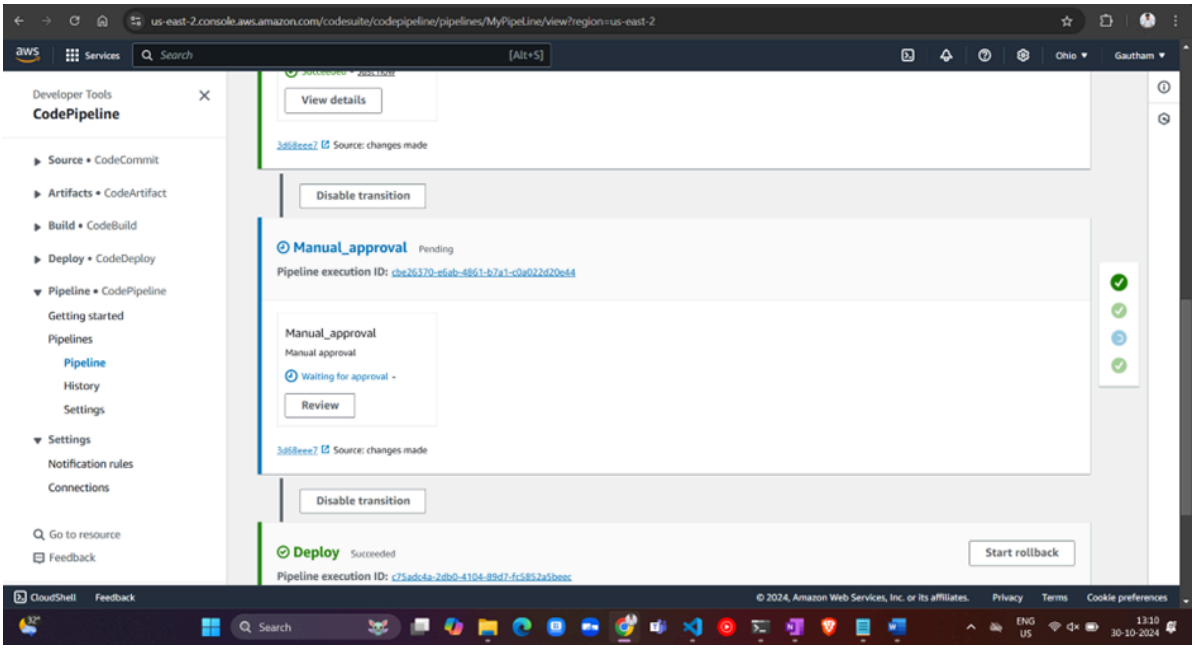
- **Check Pipeline Execution**:
    - The pipeline will detect the code changes, automatically triggering the **Source** and **Build** stages, which should complete successfully.

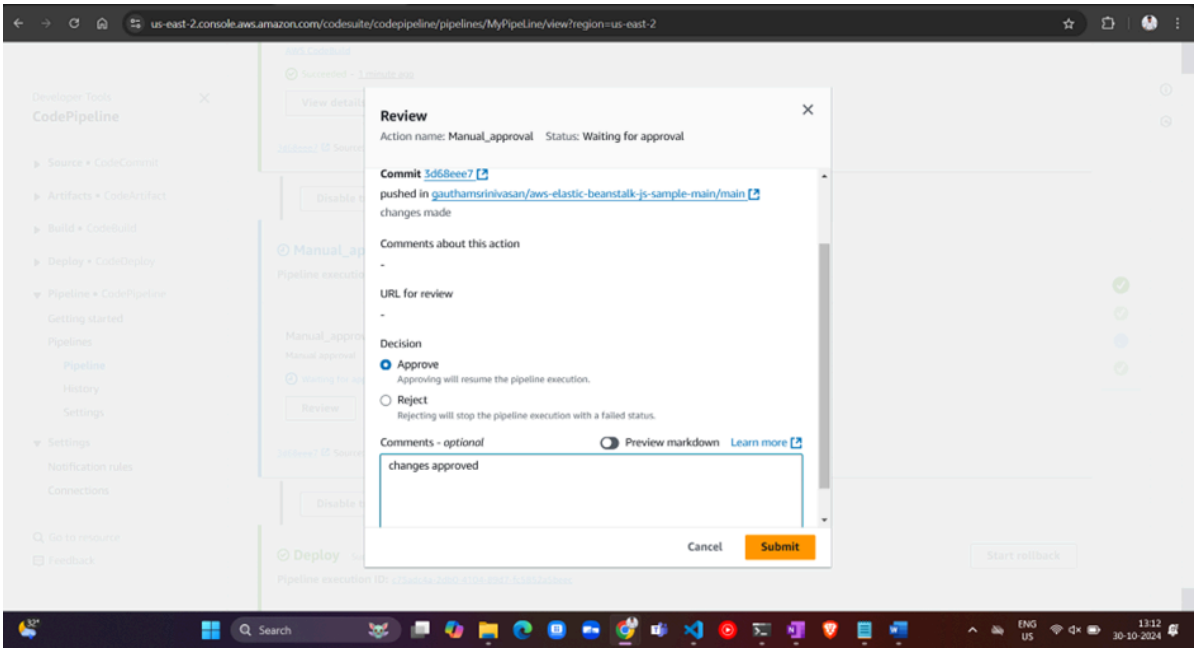


- **Manual Approval**:

- In the **Review** stage, click **Review** and choose to **Approve** or **Reject** the changes. Submitting approval will move the code to the deploy stage.




us-east-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/MyPipeLine/view?region=us-east-2
Developer Tools
CodePipeline
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History
Settings
Settings
Notification rules
Connections
Go to resource
Feedback
View details
Disable transition
Manual_approval
Pending
Pipeline execution ID: cbe26370-e6ab-4861-b7a1-c0a022d20e44
Manual_approval
Manual approval
Waiting for approval -
Review
Source: changes made
Disable transition
Deploy
Succeeded
Start rollback
Pipeline execution ID: c75adc4a-2db0-4104-89d7-fc5852a5beec
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/MyPipeLine/view?region=us-east-2
Review
Action name: Manual_approval Status: Waiting for approval
Commit 3d68eee7
pushed in gauthamsrinivasan/aws-elastic-beanstalk-js-sample-main/main
changes made
Comments about this action
-
URL for review
-
Decision
Approve
Approving will resume the pipeline execution.
Reject
Rejecting will stop the pipeline execution with a failed status.
Comments - optional
Preview markdown
Learn more
changes approved
Cancel
Submit

- **Verify Changes in EBS**:
    - Once deployed, check the Elastic Beanstalk environment to confirm that the modified code is live and reflected in the application.

Result

The setup successfully creates an automated CI/CD pipeline that integrates GitHub, CodeBuild, and Elastic Beanstalk. Code changes pushed to GitHub trigger automatic builds, with a manual approval step for controlled deployments to production. The pipeline streamlines code delivery, ensuring faster, reliable updates to the application on Elastic Beanstalk.