

Creating a Continuous Delivery Pipeline

Date: June 2025

Tags: AWS services, Git, GitHub

Introduction

In this project I created a continuous delivery pipeline that will deploy my web application every time my source code is updated. Doing this allows me to create an automated deployment process while getting more hands on experience with AWS cloud technologies and using it alongside other tools such as GitHub.

Goals / What I Wanted to Learn

- AWS Elastic Beanstalk
 - AWS CodeBuild
 - AWS CodePipeline
-

What I Did / Key Features

- Forked a repository on GitHub
 - Configured settings and permissions for AWS CodeBuild, CodePipeline, and Elastic Beanstalk
 - Created a pipeline that automatically builds and deploys my application everytime I make a change to the source code on GitHub.
-

Steps I Took

1. Forked an AWS sample repository that will be used to deploy on the cloud.
2. Cloned the repo into a local folder and now the local project and GitHub repo are linked and local changes can be pushed and these changes will be seen on GitHub automatically.
3. I then created an Elastic Beanstalk application and configured the environment including naming the application and specifying the platform of the which was [Node.js](#).

Configure environment [Info](#)

Environment tier [Info](#)

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- ☒ **Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- ☐ **Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information [Info](#)

Application name

DevOpsGettingStarted

Maximum length of 100 characters.

► Application tags (optional)

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

DevOpsGettingStarted-env

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

Leave blank for autogenerated value

.ap-southeast-2.elasticbeanstalk.com

[Check availability](#)

Environment description

Platform [Info](#)

Platform type

- ☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- ☐ **Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Node.js

Platform branch

Node.js 22 running on 64bit Amazon Linux 2023

Platform version

6.5.2 (Recommended)

Application code [Info](#)

- ☒ **Sample application**
- ☐ **Existing version**
Application versions that you have uploaded.
- ☐ **Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

Presets [Info](#)

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- ☒ **Single instance (free tier eligible)**
- ☐ Single instance (using spot instance)
- ☐ High availability
- ☐ High availability (using spot and on-demand instances)
- ☐ Custom configuration

- I then had to configure the access that the Elastic Beanstalk service had and specify which services it had access to and what actions could be done. This was done by creating an IAM role and attaching policies that contained what services could be

accessed. For the service, access to EC2 was important.

The screenshot shows the 'Configure service access' step in the AWS Elastic Beanstalk console. On the left, a progress bar lists steps 1 through 6, with 'Configure service access' (Step 2) highlighted. The main area is titled 'Configure service access' and contains three sections: 'Service access' (with a dropdown for 'aws-elasticbeanstalk-service-role' and a 'Create role' button), 'EC2 instance profile' (with a dropdown for 'aws-elasticbeanstalk-ec2-role' and a 'Create role' button), and 'EC2 key pair - optional' (with a 'Choose a key pair' dropdown and a refresh icon). At the bottom right are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

5. Once the configurations were complete I successfully launched the environment and it was now ready to deploy my application.

The screenshot shows the 'DevOpsGettingStarted-env' environment page in the AWS Elastic Beanstalk console. The top navigation bar includes 'Elastic Beanstalk' and 'Environments'. The left sidebar shows the environment's configuration history. The main content area has a green banner stating 'Environment successfully launched.' Below this, the 'DevOpsGettingStarted-env' details are shown, including 'Environment overview' (Health: Ok, Domain: DevOpsGettingStarted-env.eba-mqbx36ux.ap-southeast-2.elasticbeanstalk.com, Application name: DevOpsGettingStarted) and 'Platform' (Node.js 22 running on 64bit Amazon Linux 2023/6.5.2, Platform state: Supported). At the bottom, there are tabs for 'Events', 'Health', 'Logs', 'Monitoring', 'Alarms', 'Managed updates', and 'Tags'. The 'Events' tab is active, showing a list of events with columns for 'Time', 'Type', and 'Details'.

6. The next step was configuring CodeBuild. This involved connecting to the GitHub repo that I had forked earlier as well as specifying build commands to successfully run the build of the source code.

Project configuration

Project name

Build-DevOpsGettingStarted

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Project type

Select what type of project you would like to create. [Info](#) 

☒ Default project

Create a custom CodeBuild project.

☐ Runner project

Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

► Additional configuration

Description, public build access, build badge, concurrent build limit, tags

▼ Source

Add source

Source 1 - Primary

Source provider

GitHub ▼

Credential

✔ Your account is successfully connected through OAuth using CodeBuild managed token. [Manage account credentials.](#)

☐ Use override credentials for this project only

Repository

☒ Repository in my GitHub account

☐ Public repository

☐ GitHub scoped webhook

🔍



Source version - optional [Info](#)

▼ Buildspec

Build specifications

☒ **Insert build commands**
Store build commands as build project configuration

☐ **Use a buildspec file**
Store build commands in a YAML-formatted buildspec file

Build commands [Info](#)

```
1 version: 0.2
2 phases:
3   build:
4     commands:
5       - npm i --save
6 artifacts:
7   files:
8     - '**/*'
```



✖ 0 ⚠ 0

8:17 YAML

Switch to single line

7. I started the build and it was successful.

The screenshot shows the AWS CodeBuild console interface. At the top, a green banner states "Build started" and "You have successfully started the following build: Build-DevOpsGettingStarted:8472fde5-ef3d-42a6-a380-33019bcde5b1". Below this, the build title "Build-DevOpsGettingStarted:8472fde5-ef3d-42a6-a380-33019bcde5b1" is displayed with buttons for "Stop build", "Debug build", and "Retry build". The "Build status" section shows a green checkmark for "Succeeded". A table provides details: Initiator (root), Build ARN (arn:aws:codebuild:ap-southeast-2:916870173422:build/Build-DevOpsGettingStarted:8472fde5-ef3d-42a6-a380-33019bcde5b1), Resolved source version (3dc77442193a86ec39ec379696c4cb20b2a5090), Start time (Jun 12, 2025 4:01 PM UTC+10:00), End time (Jun 12, 2025 4:02 PM UTC+10:00), and Build number (1). Below the status, tabs for "Build logs", "Phase details", "Reports", "Environment variables", "Build details", and "Resource utilization" are shown. The "Build logs" tab is active, displaying a log of container events such as "Running on CodeBuild On-demand", "Waiting for agent ping", "Phase is DOWNLOAD_SOURCE", "CODEBUILD_SRC_DIR", "YAML location is /codebuild/readonly/buildspec.yml", "Setting HTTP client timeout", "Processing environment variables", "no runtime version selected in buildspec", "Moving to directory /codebuild/output/src2592169362/src/github.com/roy987/aws-elastic-beanstalk-express-js-sample", "Cache is not defined in the buildspec", "Skip cache due to: no paths specified to be cached", and "Registering with agent".

8. Once the build and deployment environment were setup, I created a pipeline using AWS CodePipeline. The configuration for this service involved adding the source code, build stage specs, and deploy stage specs. Once these were added I ran the pipeline and the source code was deployed onto the Elastic Beanstalk environment and I was able to access my project through a link found in the environment.

The screenshot shows the AWS CodePipeline console. A green banner at the top says "Success" and "Stage Deploy successfully retried". The pipeline name "Pipeline-DevOpsGettingStarted2" is shown with buttons for "Edit", "Stop execution", "Create trigger", "Clone pipeline", and "Release change". The "Pipeline" tab is selected, showing a visual representation of the pipeline with three stages: "Source", "Build", and "Deploy". Each stage is represented by a box with a green checkmark and a list of actions. The "Source" stage has one action "Source" (11 minutes ago). The "Build" stage has one action "AWS CodeBuild" (10 minutes ago). The "Deploy" stage has one action "AWS Elastic Beanstalk" (5 minutes ago). The pipeline is shown as a sequence of boxes connected by arrows, indicating a linear flow from Source to Build to Deploy.

9. I then tested that each change to GitHub was reflected on the deployed project and it was.
10. For added safety and quality assurance, I added a manual review for any changes before it gets deployed. This meant a user has to approve any changes before they are deployed.

Introducing the new pipeline experience

We've redesigned the pipeline view to streamline the monitoring and debugging experience. [Let us know what you think.](#) Or [go back to the old experience.](#)

Don't show again

Success

Pipeline was saved successfully.

Developer Tools > CodePipeline > Pipelines > Pipeline-DevOpsGettingStarted2

Pipeline-DevOpsGettingStarted2

EditStop executionCreate triggerClone pipelineRelease change

Pipeline

ExecutionsTriggersSettingsTagsStage

Source

9c4834b7-6a25-403c-93a6-c70c2ac3d05a

All actions succeeded.

Source

GitHub (via OAuth app)

Just now

5010c08e Source: change mes

Build

9c4834b7-6a25-403c-93a6-c70c2ac3d05a

In progress: 1

Build

AWS CodeBuild

Just now

5010c08e Source: change mes

Review

Didn't Run

Manual_Review

Manual approval

Deploy

8491d668-f25a-4814-ad74-9a9f0e83154f

All actions succeeded.

Deploy

AWS Elastic Beanstalk

10 minutes ago

341c7744 Source: change mes

Review summary

Action name: Manual_Review Status: Approved

Details

Revisions

Trigger

Webhook

- arn:aws:codepipeline:ap-southeast-2:916870173422:webhook:PipelineDevOpsGettingStarte--Source--roy987awselasticbeanstalkex--2032955488

Status

✔ Approved

Comments about this action

URL for review

SNS topic ARN

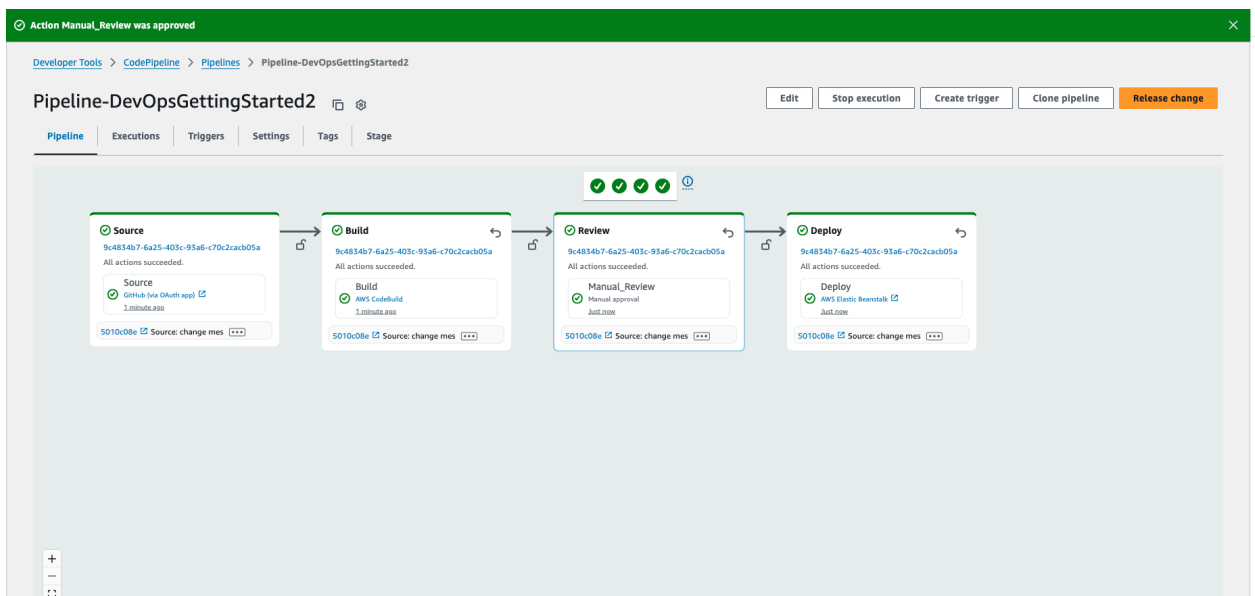
Reviewer ARN

arn:aws:iam::916870173422:root

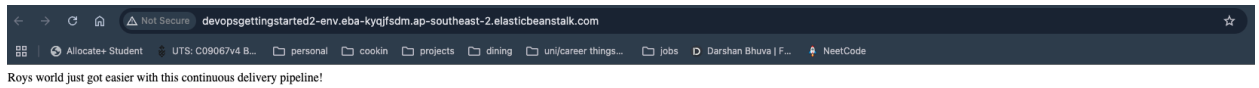
Comments

Build is approved for deployment.

Done



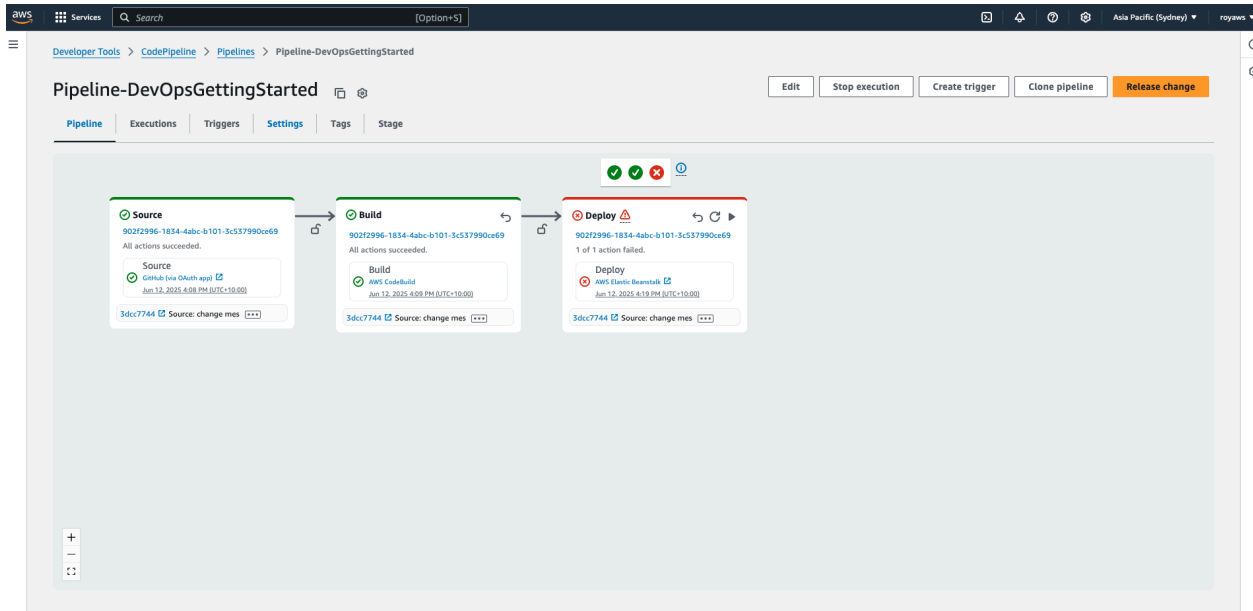
11. Now my project is successfully deployed and accessible to the public via the internet!!



Challenges & How I Solved Them

Pipeline Deployment Issue

- **Problem:** When running the pipeline, at first the build stage was encountering issues and would not deploy. After looking into the error message, I found that the pipeline did not have access to the necessary services to deploy the application.
- **Solution:** I attached an AWS created policy for access to elastic beanstalk and necessary actions and this resolved the problem.



Policy AWSCodePipelineServiceRole-ap-southeast-2-Pipeline-DevOpsGettingStarted updated.

AWSCodePipelineServiceRole-ap-southeast-2-Pipeline-DevOpsGettingStarted

Policy used in trust relationship with CodePipeline for service role

Policy details			
Type	Creation time	Edited time	ARN
Customer managed	June 12, 2025, 16:08 (UTC+10:00)	June 12, 2025, 16:35 (UTC+10:00)	arn:aws:iam::916870173422:policy/service-role/AWSCodePipelineServiceRole-ap-southeast-2-Pipeline-DevOpsGettingStarted

Permissions	Entities attached	Tags	Policy versions	Last Accessed
Permissions defined in this policy				
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it				
Allow (2 of 441 services)				
Service	Access level	Resource	Request condition	
Elastic Beanstalk	Limited: List, Write	All resources	None	
S3	Limited: Permissions management, Read, Write	Multiple	aws:ResourceAccount = 916870173422	

What I Learned

- How to configure AWS Elastic Beanstalk, CodePipeline, and CodeBuild
- How IAM roles and policies work
- The importance of least privilege access - only giving access to necessary services and actions
- How to connect AWS services to create an end-to-end CI/CD pipeline, integrating GitHub, build and deploy stages
- How to troubleshoot errors in pipeline failures

- The benefits of automation in software delivery, improving efficiency and ensuring consistency across environments



What I'd Improve Next Time

- Make sure to double check that all services have the correct permissions that they need to work together
- Check any usage costs before creating and running projects as I incurred a small charge while using CodeBuild.