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

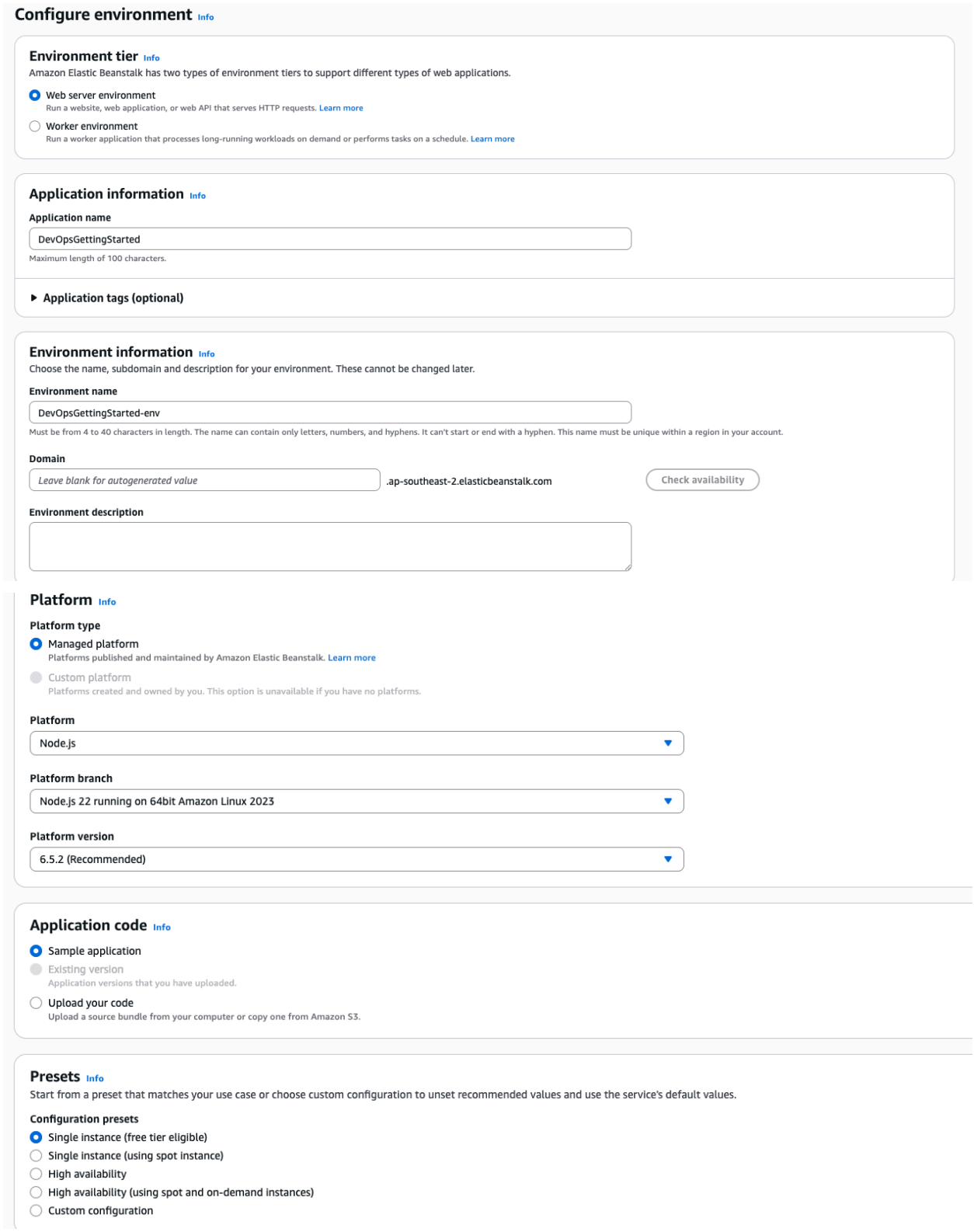
- AWS Elastic Beanstalk
- AWS CodeBuild
- AWS CodePipeline

🛠️ 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.

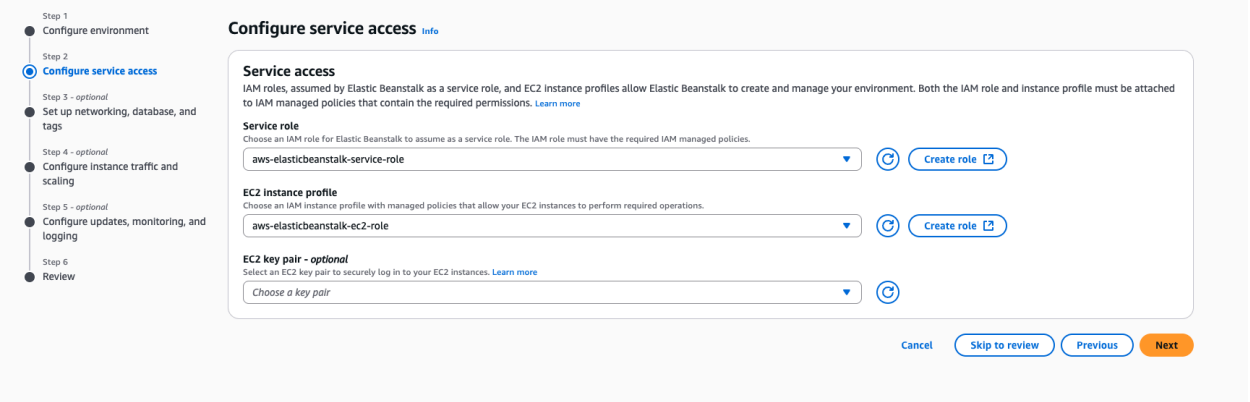
🧱 Steps I Took

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.

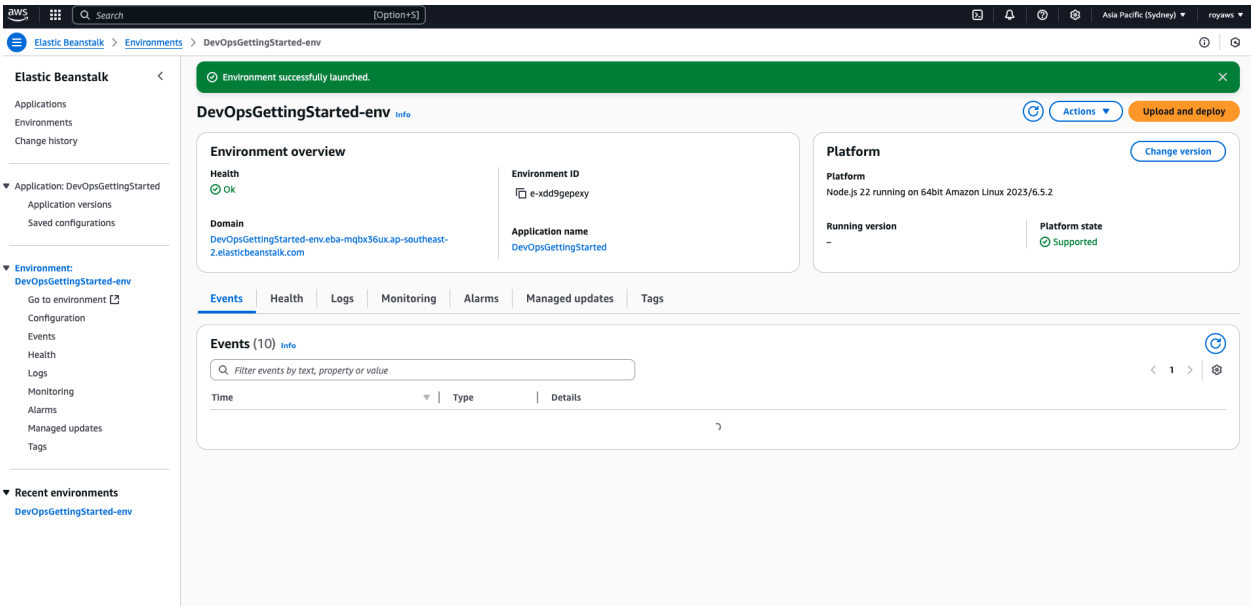


4. I then had to configure the access that the Elastic Beanstalk service had and specify which services it had access to and what actions could be done. This was done by creating an IAM role and attaching policies that contained what services could be

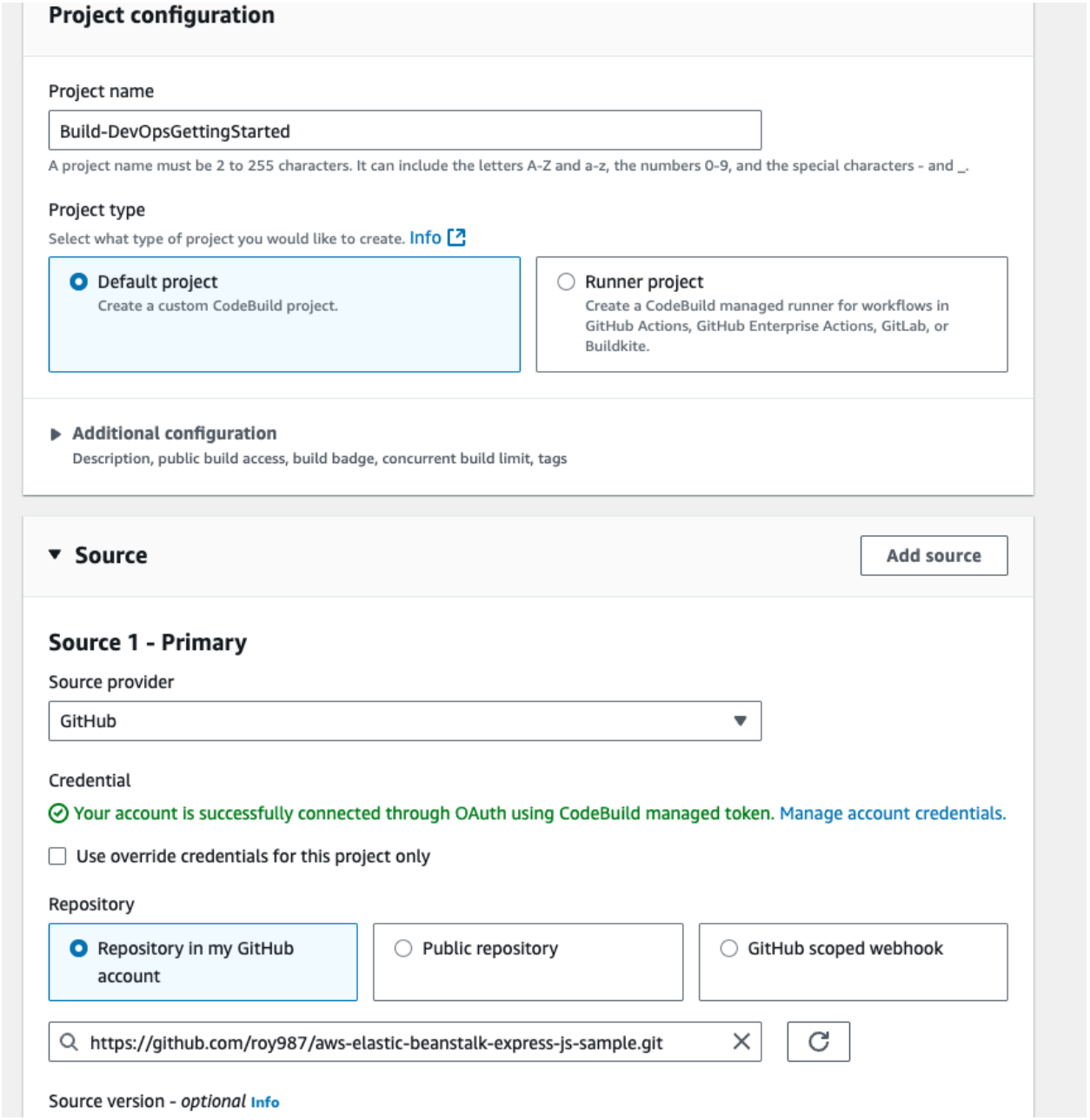
accessed. For the service, access to EC2 was important.

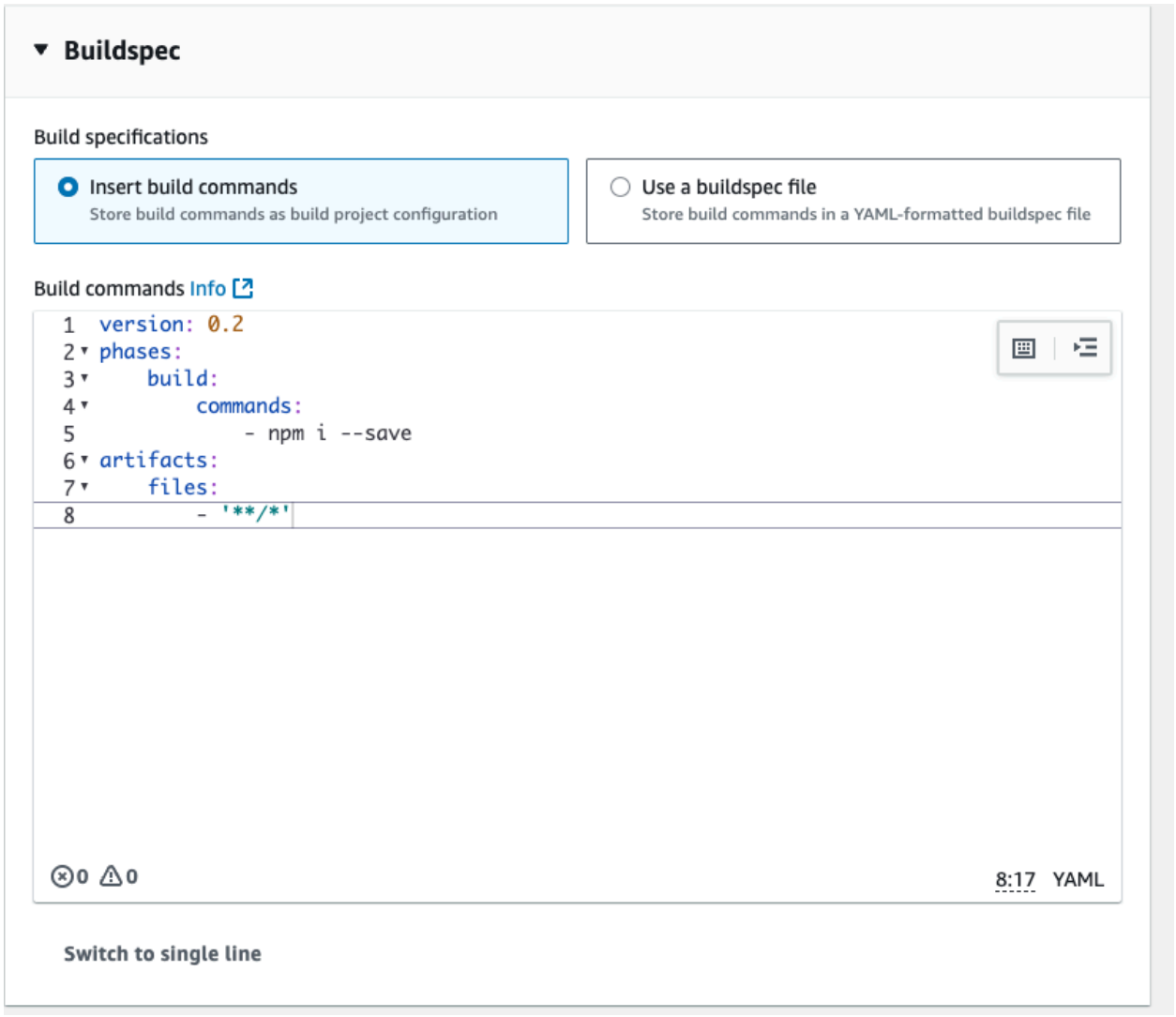


5. Once the configurations were complete I successfully launched the environment and it was now ready to deploy my application.

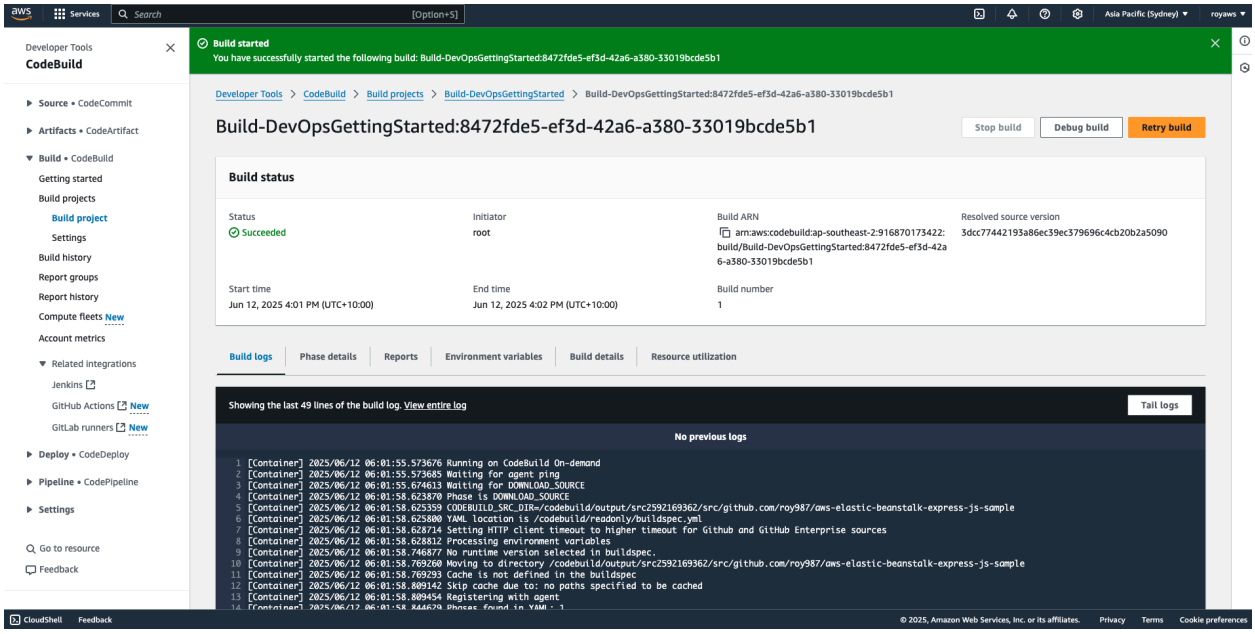


6. The next step was configuring CodeBuild. This involved connecting to the GitHub repo that I had forked earlier as well as specifying build commands to successfully run the build of the source code.

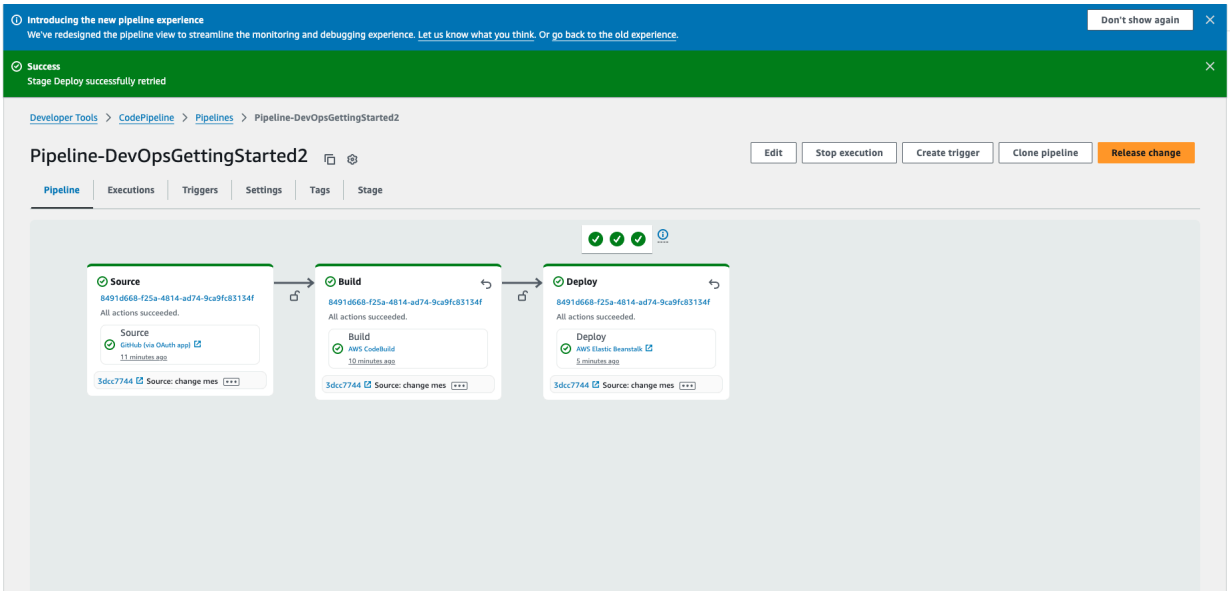

Project configuration
Project name
Build-DevOpsGettingStarted
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Project type
Select what type of project you would like to create. Info
Default project
Create a custom 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
https://github.com/roy987/aws-elastic-beanstalk-express-js-sample.git
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

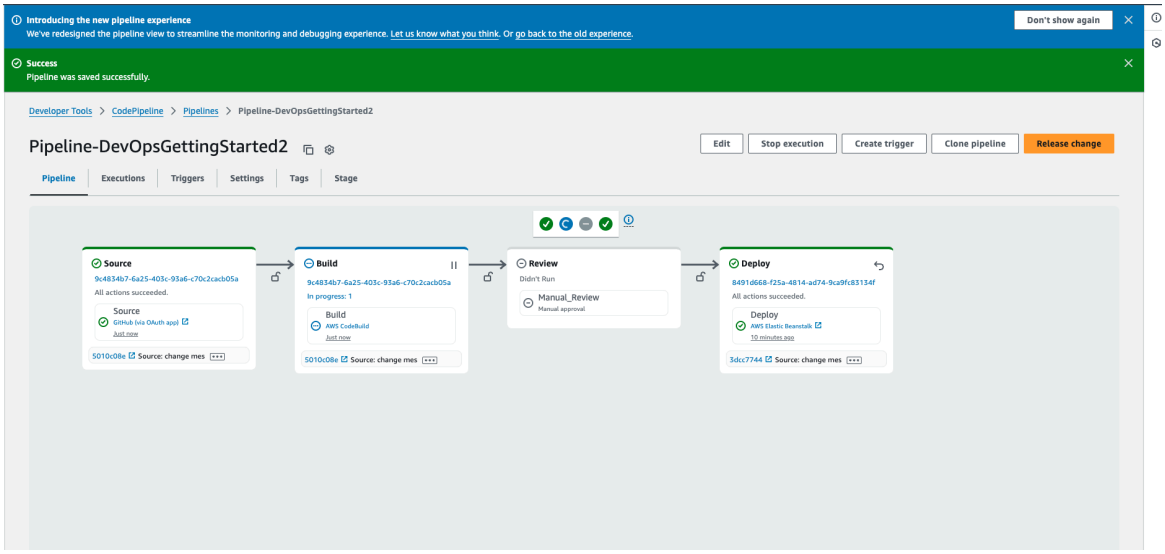
7. I started the build and it was successful.



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.



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
Edit
Stop execution
Create trigger
Clone pipeline
Release change
Pipeline
Executions
Triggers
Settings
Tags
Stage
Source
9c4834b7-6a25-403c-93a6-c70c2cacb05a
All actions succeeded.
Source
GitHub (via OAuth app)
Just now
5010c08e Source: change mes
Build
9c4834b7-6a25-403c-93a6-c70c2cacb05a
In progress: 1
Build
AWS CodeBuild
Just now
5010c08e Source: change mes
Review
Didn't Run
Manual_Review
Manual approval
Deploy
8491d668-f25a-4814-ad74-9ca9fc83134f
All actions succeeded.
Deploy
AWS Elastic Beanstalk
10 minutes ago
3dcc7744 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

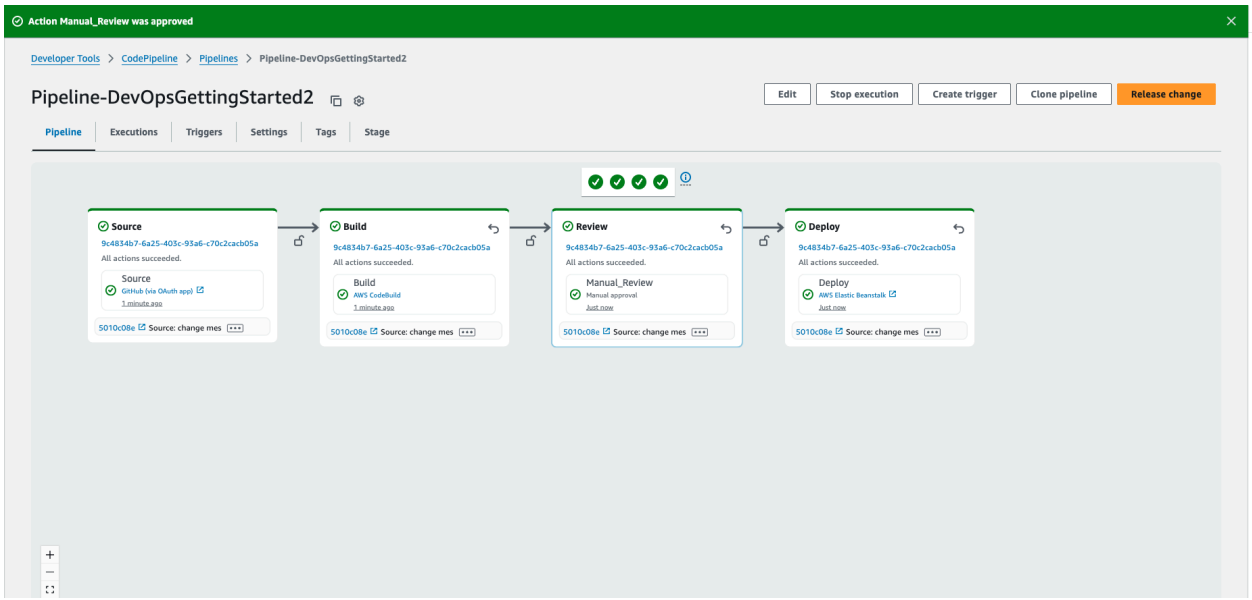
Reviewer ARN

arn:aws:iam::916870173422:root

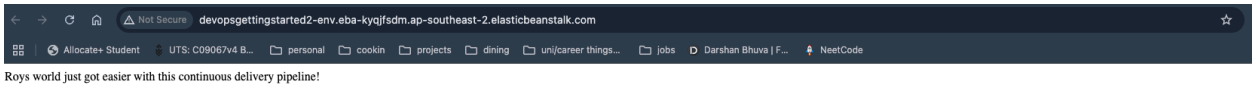
Comments

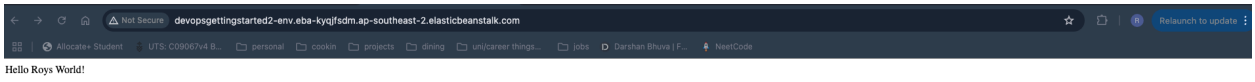
Build is approved for deployment.

Done



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

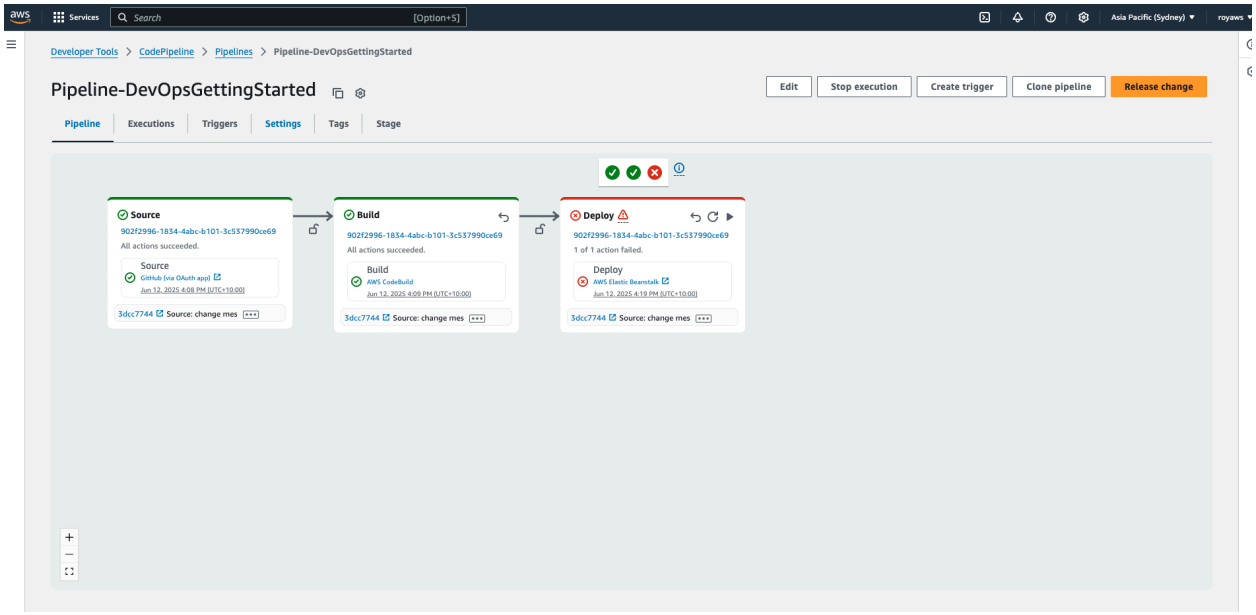


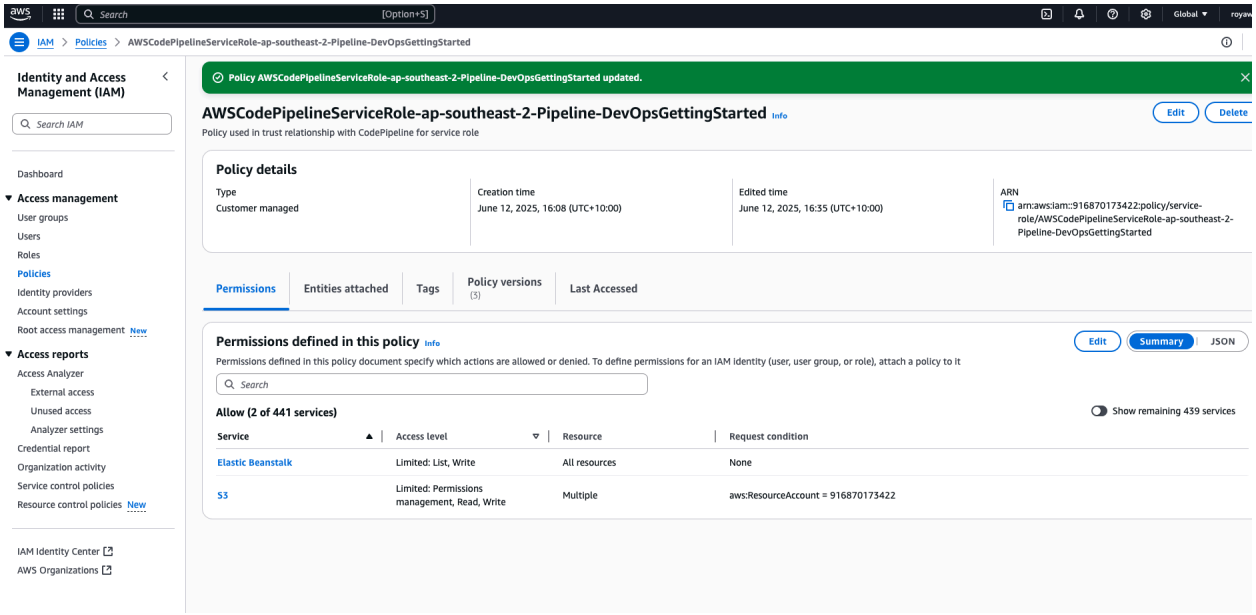


🧩 Challenges & How I Solved Them

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.





🎯 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.