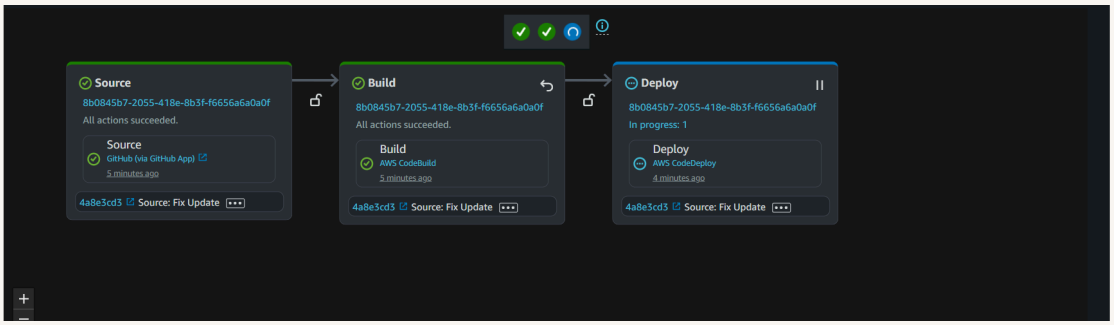
nextwork.org

Build a CI/CD Pipeline with AWS

LW lwazi995@outlook.com



LW
lwazi995@outlook.com
NextWork Student

Introducing Today's Project!

In this project we will demonstate how to use Codepipeline to set up a CI/CD pipeline! Codepipeline will Automate the flow from GitHub all the way to CodeDeploy.

Key tools and concepts

Services I used were AWS CodePipeline, CodeBuild, S3, and GitHub. Key concepts I learnt include automation of deployments, continuous integration, continuous delivery, and version control using Git.

Project reflection

This project took me approximately 15 hours. The most challenging part was setting up the CodePipeline correctly with all the integrations. It was most rewarding to see my changes deploy automatically after each push.

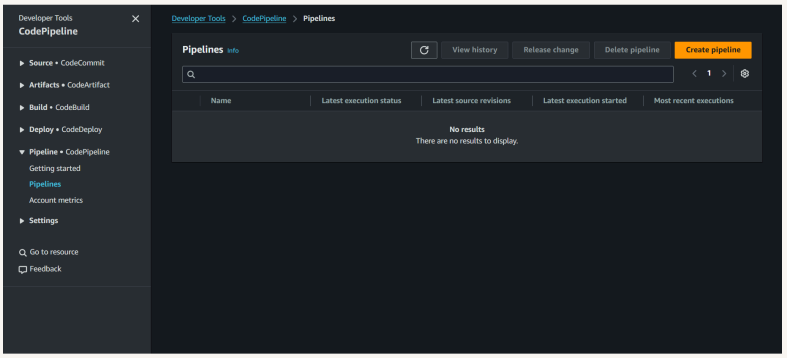
LW
lwazi995@outlook.com
NextWork Student

Starting a CI/CD Pipeline

AWS CodePipeline is a powerful DevOps service that enables you to automate the end-to-end workflow of moving code from a source repository like GitHub to your deployment tool, such as AWS CodeDeploy. It ensures that deployments are consistent.

CodePipeline offers different execution modes based on your specific deployment needs. I chose the automated deployment mode, which streamlines the entire process from code commit to production release.

CodePipeline offers different execution modes based on deployment needs. I chose automated deployment, but other options include manual approval or third-party tool integration.

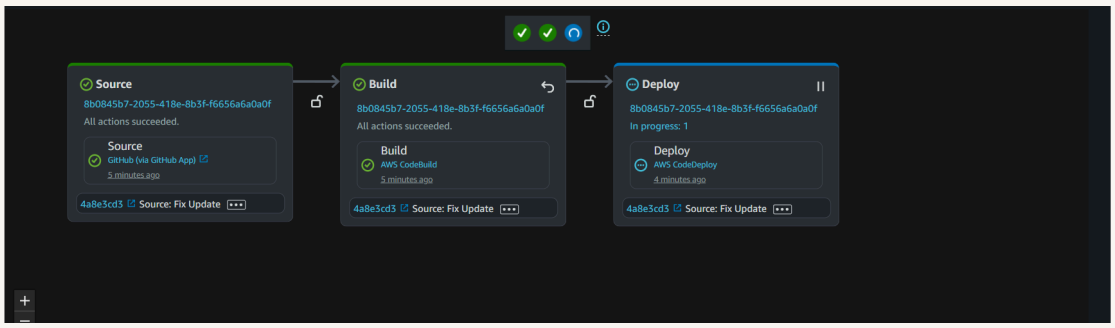


LW
lwazi995@outlook.com
NextWork Student

CI/CD Stages

The three stages I've set up in my CI/CD pipeline are Source the source code for our web project in GitHub, Build stage which is building the web app using code build. The deploy stage deploying changes on our webapp using code deploy.

CodePipeline organizes the three stages into Source, Build, and Deploy. In each stage, you can see more details on the execution status, logs, success or failure messages, and any related artifacts. This helps track progress, diagnose issues.

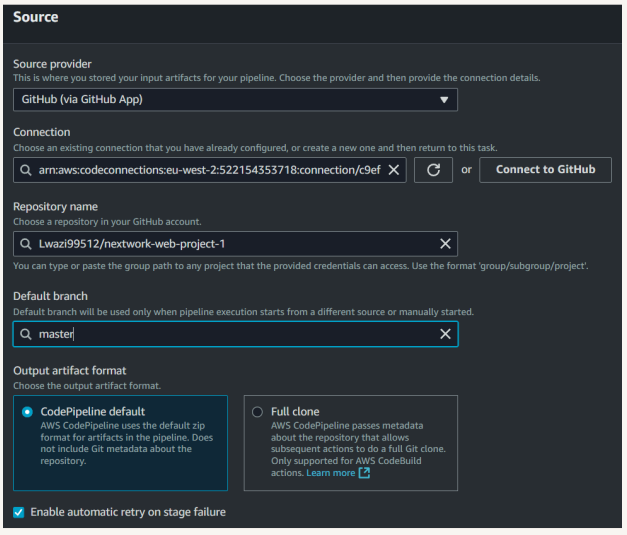


LW
lwazi995@outlook.com
NextWork Student

Source Stage

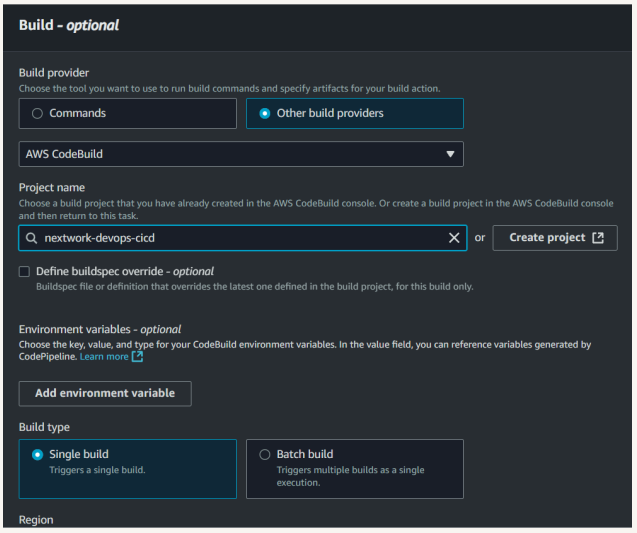
In the Source stage, the default branch tells CodePipeline, where our source code lives GitHub, the default branch tell CodePipeline exactly which version of our code we want to use in this workflow. We can have different branches in our web app.

The source stage is also where you enable webhook events, which are like notifications. Whenever you make a change to the source code the webhook event will detect the change and alreart code pipeline to trigger a new run.



Build Stage

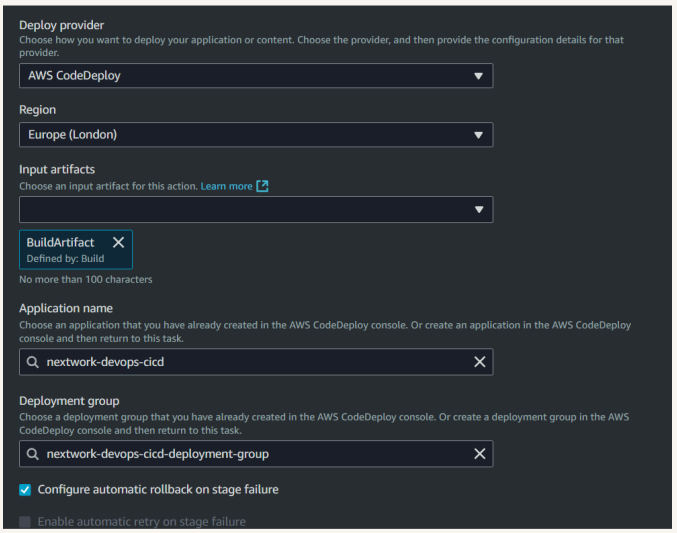
The Build stage sets up how we will build our webapp and make it ready for deployment I configured... codebuild to be our build provider and to use the input artifact The input artifact for the build stage is the comprass code that it gets from Git.



LW
lwazi995@outlook.com
NextWork Student

Deploy Stage

The Deploy stage is where we set up codedeploy to be our deploment provider it takes the build artifact from code build and application and deployment settings that we find in our deployment group.



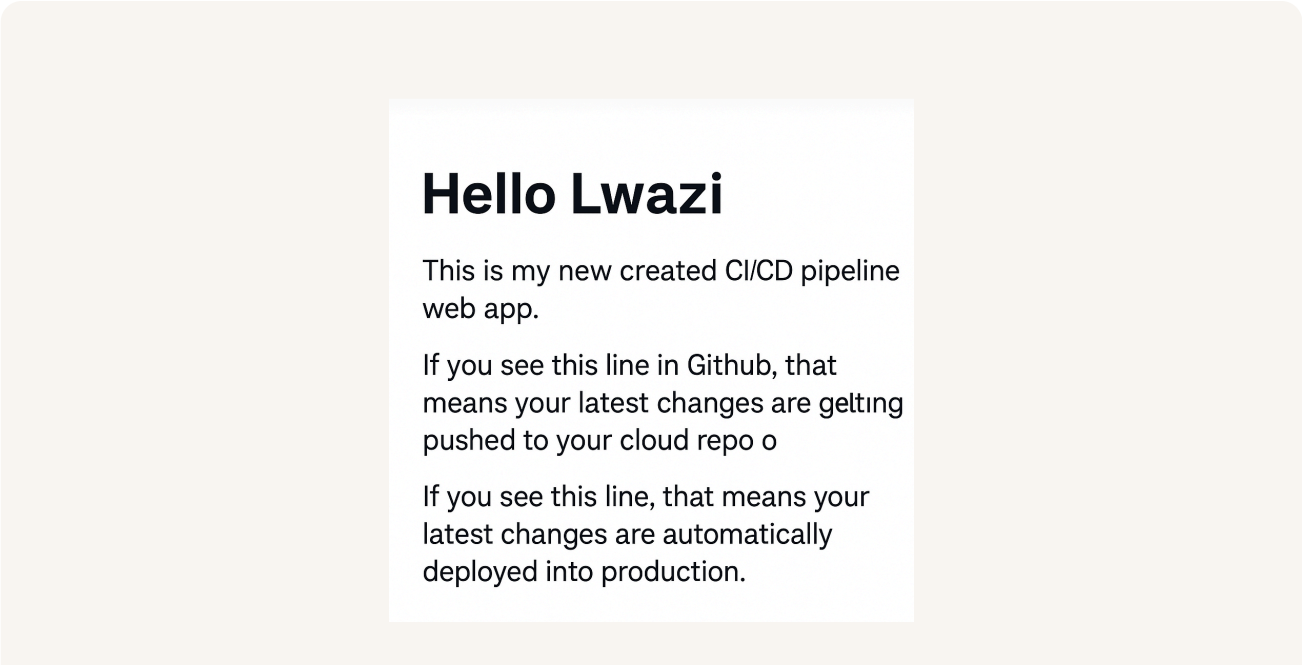
LW
lwazi995@outlook.com
NextWork Student
NextWork.org

Success!

Since my CI/CD pipeline gets triggered by code changes pushed to GitHub, I tested my pipeline by making a change to my source code, pushing it to GitHub, and ensuring that the pipeline automatically ran through its stages.

The moment I pushed the code change, the pipeline automatically triggered and began executing the stages. The commit message under each stage reflects the specific changes made, allowing me to track the progress and see the result of the code change.

Once my pipeline executed successfully, I checked my live web app that it updated with manually rebuilding our project in code build and redeploy the project in code deploy the changes went live into production straight away.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects