Continuous Integration with CodeBuild

Seth Sekyere

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  pre_build:
    commands:
      - echo Initializing environment
      - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain nextwork --domain-owner 123456789012 --region us-east-2 --qu
  build:
    commands:
      - echo Build started on `date`
      - mvn -s settings.xml compile
  post_build:
    commands:
      - echo Build completed on `date`
      - mvn -s settings.xml package
artifacts:
  files:
    - target/nextwork-web-project.war
  discard-paths: no
```

Seth Sekyere

Introducing Today's Project!

In this project, I will demonstrate how to use AWS CodeBuild to automate the build process. I'm doing this project to learn how CodeBuild compiles, tests, and packages code as part of a CI/CD pipeline.

Key tools and concepts

Services I used were EC2, CodeArtifact, CodeBuild, S3, and CloudWatch. Key concepts I learnt include CI/CD pipelines, buildspec.yml, artifact storage, secure repository access, and automated Java web app builds.

Project reflection

This project took me approximately 30 minutes to complete. The most challenging part was configuring CodeBuild and IAM permissions correctly. It was most rewarding to see the build succeed and the artifact appear in S3.

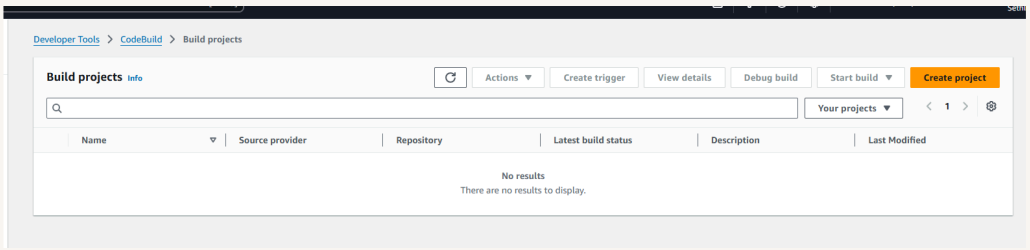
I'll be working on the next project as the fifth part of the series, continuing to build and extend the CI/CD pipeline for our web application.

Seth Sekyere

Setting up a CodeBuild Project

CodeBuild is a continuous integration (CI) service, which automatically builds and tests code whenever changes are made. Engineering teams use it to catch issues early, ensure consistent builds, and speed up development.

My CodeBuild project's source configuration means where CodeBuild fetches the code to build. I selected GitHub because that's where my web app's repository is stored.

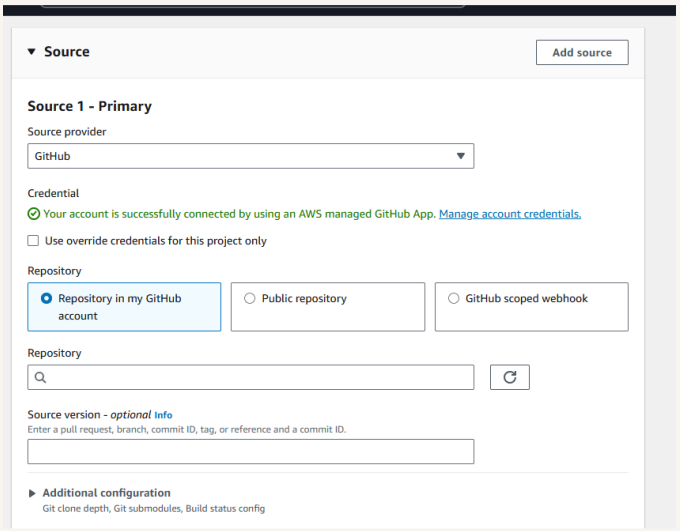


Seth Sekyere

Connecting CodeBuild with GitHub

There are multiple credential types for GitHub, like Personal Access Token and OAuth App. I used GitHub App because it's the simplest, most secure option, AWS manages the connection, and I don't need to handle tokens or keys manually.

The service that helped connect our AWS account to GitHub is AWS CodeConnections, which securely manages authentication so CodeBuild can access our repository without handling tokens or passwords manually.



Seth Sekyere

CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means the build runs on Amazon Linux with Corretto 8, using a managed image on EC2 with an on-demand provisioning model and a new service role for permissions.

Artifacts

Build artifacts are the output files from building the app. They're important because they contain everything needed to deploy the app. My build process will create a WAR file, which I will store in an S3 bucket

Packaging

When setting up CodeBuild, I also chose to package artifacts in a Zip file because it reduces size, keeps all files organized in one package, and makes deployment and sharing much simpler.

Seth Sekyere

Monitoring

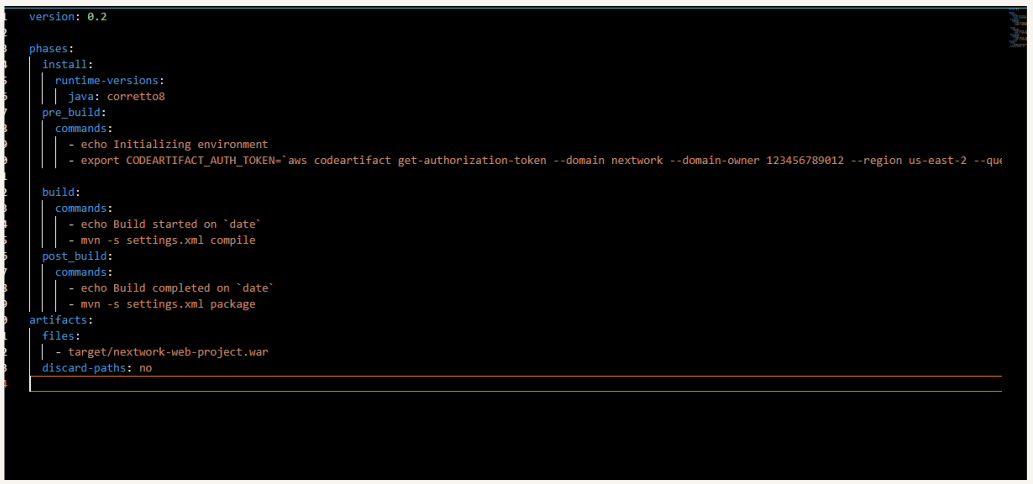
For monitoring, I enabled CloudWatch Logs, which records all build commands, outputs, and errors, helping debug issues and track build progress efficiently.

Seth Sekyere

buildspec.yml

My first build failed because CodeBuild couldn't find a buildspec.yml file. This file is needed to define the build commands and phases so CodeBuild knows how to compile, test, and package the project.

The first two phases prepare the environment and get the CodeArtifact token. The third phase compiles the code with Maven. The fourth phase packages the compiled code into a WAR file for deployment.



Seth Sekyere

Success!

My second build also failed with a permissions error accessing CodeArtifact. To fix this, I updated the CodeBuild service role to grant it access to the repository so dependencies could be downloaded.

To resolve the second error, I attached the CodeArtifact access policy to the CodeBuild service role. When I built my project again, all phases completed successfully and the WAR file was generated in S3.

To verify the build, I checked the S3 bucket. Seeing the artifact tells me the project was successfully compiled, packaged, and uploaded, and the WAR file is ready for deployment.

Seth Sekyere