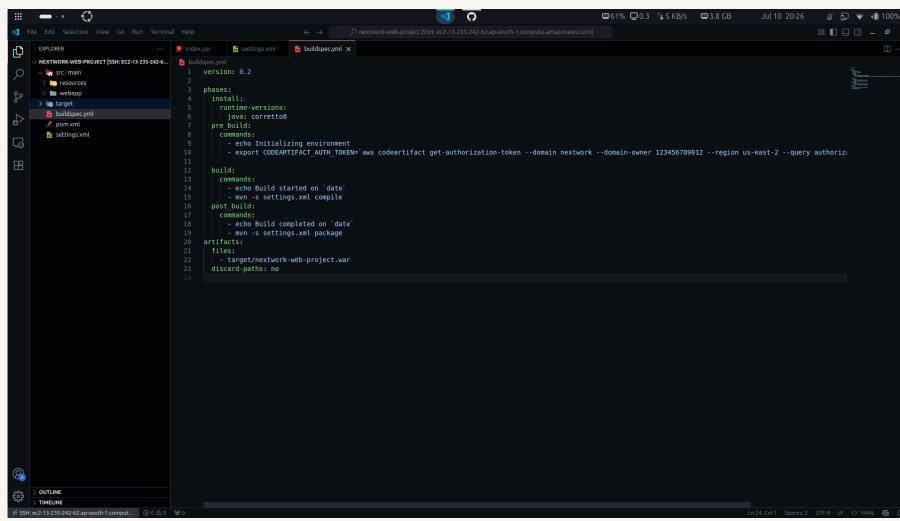




Continuous Integration with CodeBuild



Shravan Kumar Satapathy



```
version: 0.2
phases:
  install:
    commands:
      - mvn versions:set --non-interactive --batch-mode
      - java: corretto8
  pre_build:
    commands:
      - echo Initializing environment
      - export CODEBUILD_IAM_AUTH_TOKEN=$(aws codeartifact get-authorization-token --domain nextwork --domain-owner 123456789012 --region us-east-2 --query authorizationToken)
  build:
    commands:
      - echo Build started on `date`
      - mvn -s settings.xml compile
      - mvn test
      - mvn -s settings.xml package
  artifacts:
    type: ZIP
    target: network-web-project.zip
    discard-paths: no
```



Introducing Today's Project!

In this project, I will demonstrate CI with AWS CodeBuild for automated web app builds. I'm doing this project to learn CodeBuild's role in compiling, testing, packaging code, saving time, reducing errors, and enabling faster CI/CD.

Key tools and concepts

Services I used were AWS CodeBuild, CodeArtifact, Amazon S3, GitHub, IAM, and Maven. Key concepts I learnt include Continuous Integration, build automation via `buildspec.yml`, artifact management, CloudWatch Logs, and secure GitHub integration.

Project reflection

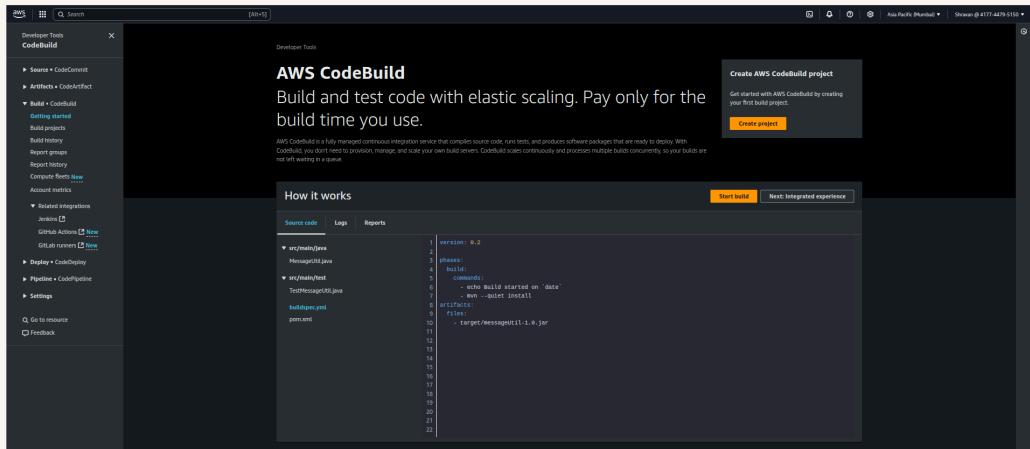
This project took me approximately 105 to 120 minutes. The most challenging part was troubleshooting `buildspec.yml` and CodeArtifact permission errors. It was most rewarding to achieve a successful build and verify artifact storage in S3.

This project is part four of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project soon, continuing to automate and expand the pipeline's capabilities.

Setting up a CodeBuild Project

CodeBuild is a CI service, automating code builds: compiling, testing, packaging. Engineering teams use it as it eliminates manual server management, reduces errors, and ensures faster, consistent software delivery by catching issues early.

My CodeBuild project's source configuration means defining the location of the code CodeBuild will fetch, compile, and package. I selected GitHub as the source provider because that's where my web app's code is stored, enabling seamless integration.



Shravan Kumar Satapathy

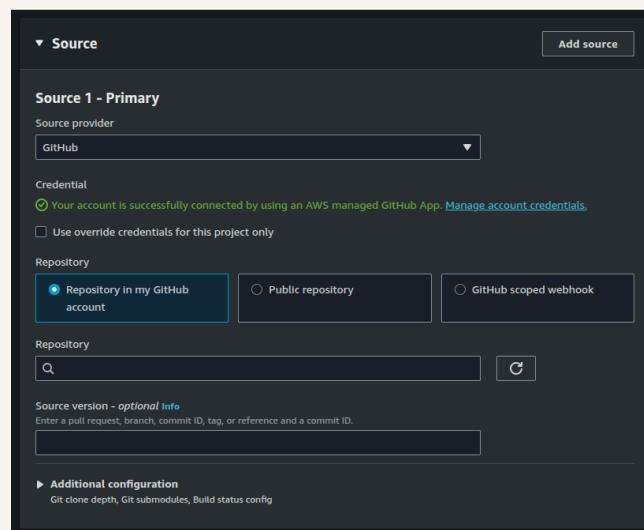
NextWork Student

nextwork.org

Connecting CodeBuild with GitHub

There are multiple credential types for GitHub, like GitHub App, Personal access token, and OAuth app. I used GitHub App because it's generally the simplest and most secure option, with AWS managing the connection and reducing direct token handling.

The service that helped connect my AWS environment to GitHub is AWS CodeConnections. It acts as a secure bridge, managing the authentication complexity without requiring manual handling of API keys or tokens, simplifying the connection process.





CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration defines build OS, runtime, and compute resources. It includes: On-demand provisioning, managed Corretto 8 image (Amazon Linux), EC2 compute, and a new service role for efficient, scalable builds.

Artifacts

Build artifacts are outputs of the build process, essential for deployment. They're important because they are the final product. My build process will create a WAR file. To store them, I created an S3 bucket named "nextwork-devops-cicd-shravan".

Packaging

When setting up CodeBuild, I also chose to package artifacts in a Zip file because it reduces size for faster uploads/lower costs, provides better organization, and simplifies deployment and sharing for efficiency.

Shravan Kumar Satapathy

NextWork Student

nextwork.org

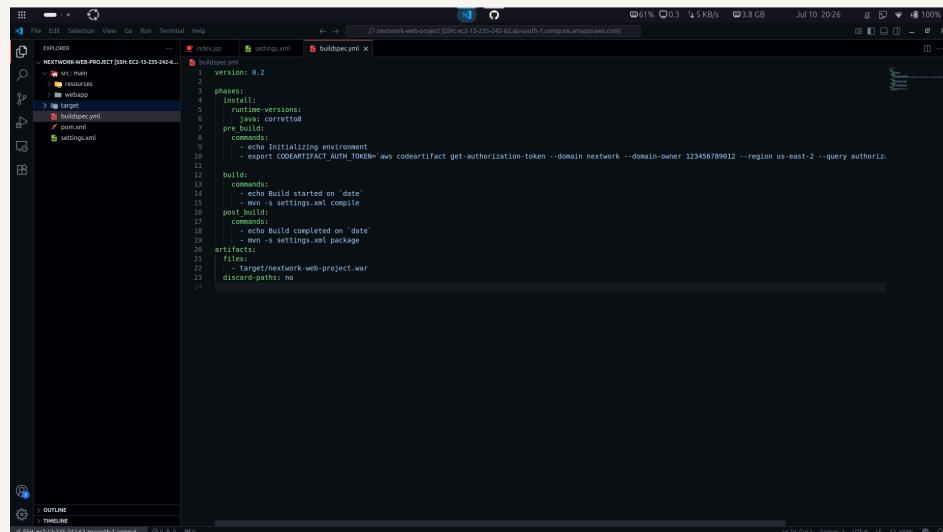
Monitoring

For monitoring, I enabled CloudWatch Logs, which is a service collecting build process data, including commands, output, and errors. It's essential for tracking progress, debugging failures, and auditing CodeBuild activities.

buildspec.yml

My first build failed because CodeBuild couldn't find the "buildspec.yml" file in my GitHub repository. A "buildspec.yml" file is needed because it's CodeBuild's instruction manual, detailing the exact commands and steps for the build process.

The first two phases in my buildspec.yml file are install and pre_build, set Java 8 runtime and fetch a CodeArtifact token. The third phase, build, compiles code using Maven. The fourth phase, post_build, packages the application into a WAR file.



```
version: 0.2
phases:
  install:
    runtime-versions:
      java: 8
  pre-build:
    commands:
      - echo Initializing environment
      - export CODEARTIFACT_AUTH_TOKEN=$(aws codeartifact get-authorization-token --domain nextwork --domain-owner 123456789012 --region us-east-2 --query authorizationToken)
  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
```



Success!

My second build also failed, but with a different error that said 'no matching artifact paths found'. CodeBuild's role lacked CodeArtifact access for dependencies. To fix this, I'll grant CodeBuild's IAM role CodeArtifact permissions.

To resolve the second error, I attached the codeartifact-nextwork-consumer-policy to CodeBuild's service role in IAM, granting CodeArtifact access. When I built my project again, I saw the build succeed, indicating a successful CI pipeline run.

To verify the build, I checked my "nextwork-devops-cicd" S3 bucket. The nextwork-devops-cicd-artifact.zip was created. Seeing the artifact tells me the code compiled, packaged, uploaded, and is ready for deployment.



Shravan Kumar Satapathy

NextWork Student

nextwork.org

The screenshot shows the AWS CodeBuild build history page. The build has started successfully. The build status table includes columns for Status, Initiator, Build ARN, and Resolved source version. The build logs table lists various phases: SUBMITTED, QUEUED, PROVISIONING, DOWNLOAD_SOURCE, INSTALL, PRE_BUILD, BUILD, and POST_BUILD, all of which have succeeded.

Name	Status	Context	Duration	Start time	End time
SUBMITTED	Success Succeeded	-	<1 sec	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
QUEUED	Success Succeeded	-	<1 sec	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
PROVISIONING	Success Succeeded	-	7 secs	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
DOWNLOAD_SOURCE	Success Succeeded	-	6 secs	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
INSTALL	Success Succeeded	-	<1 sec	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
PRE_BUILD	Success Succeeded	-	9 secs	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
BUILD	Success Succeeded	-	28 secs	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:08 PM (UTC+5:30)
POST_BUILD	Success Succeeded	-	35 secs	Jul 10, 2025 9:08 PM (UTC+5:30)	Jul 10, 2025 9:09 PM (UTC+5:30)



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

