



Continuous Integration with CodeBuild



Sanjana Tripathy

```
index.jsp      settings.xml      buildspec.yml
 1  version: 0.2
 2
 3  phases:
 4    install:
 5      runtime-versions:
 6        java: corretto8
 7    pre_build:
 8      commands:
 9        - echo Initializing environment
10        - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain nextwork --domain-owner 794038222820 --region ap-south-1`
11
12    build:
13      commands:
14        - echo Build started on `date`
15        - mvn -s settings.xml compile
16    post_build:
17      commands:
18        - echo Build completed on `date`
19        - mvn -s settings.xml package
20  artifacts:
21    files:
22      - target/nextwork-web-project.war
23  discard-paths: no
24
```

Introducing Today's Project!

In this project, I will demonstrate how to automate the build process using AWS CodeBuild. I'm doing this project to learn continuous integration by creating builds, running tests, and packaging code efficiently as part of a CI/CD pipeline.

Key tools and concepts

Services I used were AWS CodeBuild, AWS CodeArtifact, Amazon S3, GitHub, and IAM. Key concepts I learnt include CI/CD pipelines, build automation, artifact management, secure repository access, and role-based permissions.

Project reflection

This project took me approximately 2 hours. The most challenging part was nothing, as each step was enriching and taught me something new. It was most rewarding to see the build complete successfully and artifacts generated as expected.

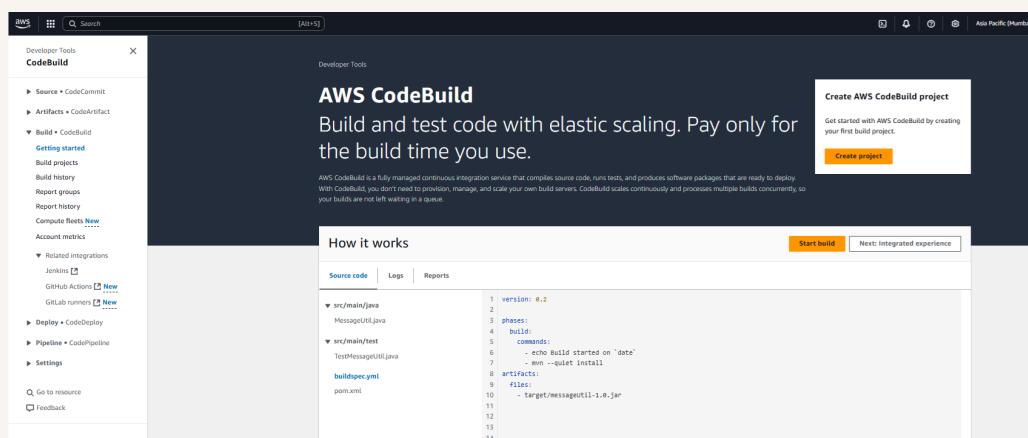
This project is part four of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow to continue strengthening my cloud and DevOps skills with hands-on experience.

Setting up a CodeBuild Project

CodeBuild is a continuous integration service, which means it automatically compiles code, runs tests, and packages applications whenever changes are pushed.

Engineering teams use it because it saves time, reduces errors, and ensures consistent builds.

My CodeBuild project's source configuration means defining where the build should pull the application code from. I chose GitHub because that's where all my code gets pushed, ensuring smooth integration and automatic builds on every update.



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 the most secure and simplest option, with AWS managing the connection, reducing manual token handling and risks.

The service that helped connect with GitHub is AWS CodeConnections because it acts as a secure bridge between AWS and external repositories, handling authentication behind the scenes so I don't need to manage tokens or keys manually.

The screenshot shows the 'Project configuration' page for a new AWS CodeBuild project. The 'Project name' is set to 'sanjana_devops-clcd'. Under 'Project type', 'Default project' is selected. In the 'Source' section, 'Source 1 - Primary' is configured with 'GitHub' as the source provider. A note indicates that the account is successfully connected via an AWS managed GitHub App. The 'Repository' dropdown shows 'Repository in my GitHub' is selected. Other options include 'Public repository' and 'GitHub scoped webhook'.

A circular profile picture of a young woman with dark hair, wearing a pink top, sitting on a blue couch in an office setting.

CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means specifying where and how the build runs. It includes settings like the provisioning model, operating system, runtime, and compute resources to ensure builds are executed efficiently and consistently.

Artifacts

Build artifacts are the output files generated after a build process, like compiled code or packaged applications. They're important because they are deployed or tested further. My build process will create a WAR file, and to store them, I created an S3.

Packaging

When setting up CodeBuild, I also chose to package artifacts in a .zip because it compresses the files, makes storage and transfer efficient, and keeps all build outputs organized for easy deployment or sharing.

Sanjana Tripathy
NextWork Student

nextwork.org

Monitoring

For monitoring, I enabled CloudWatch Logs, which is a service that captures, stores, and allows me to analyze logs from my build process, helping track progress, troubleshoot issues, and ensure builds run smoothly.



buildspec.yml

My first build failed because CodeBuild couldn't find a buildspec.yml file in my GitHub repository. A buildspec.yml file is needed because it tells CodeBuild exactly how to build, test, and package the project, guiding the entire build process.

The first two phases in my buildspec.yml file are install and pre_build for setup and fetching tokens. The third phase is build, compiling code with Maven. The fourth phase is post_build, packaging everything into a WAR file.

```
index.jsp settings.xml buildspec.yml
1  version: 0.2
2
3  phases:
4    install:
5      runtime-versions:
6        | java: corretto8
7    pre_build:
8      commands:
9        - echo Initializing environment
10       - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain nextwork --domain-owner 79403822280 --region ap-south-1`
11
12    build:
13      commands:
14        - echo Build started on `date`
15        - mvn -s settings.xml compile
16    post_build:
17      commands:
18        - echo Build completed on `date`
19        - mvn -s settings.xml package
20  artifacts:
21    files:
22      - target/nextwork-web-project.war
23  discard-paths: no
24
```

A circular profile picture of a young woman with dark hair, wearing a pink top, sitting on a blue couch in an office setting.

Sanjana Tripathy
NextWork Student

nextwork.org

Success!

My second build also failed, but with a different error that said CodeBuild couldn't access the settings.xml file. To fix this, we need to grant CodeBuild's IAM role permission to access CodeArtifact.

To resolve the second error, I granted CodeBuild's IAM role permission to access CodeArtifact. When I built my project again, I saw the build complete successfully with the WAR file generated.

To verify the build, I checked the S3 bucket where the build artifacts are stored. Seeing the artifact tells me the build was successful and the WAR file is ready for deployment.



Sanjana Tripathy
NextWork Student

nextwork.org

Developer Tools > CodeBuild > Build projects > sanjana-devops > sanjana-devops:6e58180c-4ed3-4bcf-a468-50ae0d75cd9a

sanjana-devops:6e58180c-4ed3-4bcf-a468-50ae0d75cd9a

Status: Succeeded Initiator: Sanjana Build ARN: arn:aws:codebuild:ap-south-1:794038222820:build/sanjana-devops:6e58180c-4ed3-4bcf-a468-50ae0d75cd9a Resolved source version: 79b122faabdb0b09915a5a0248079dd11b3a8c

Start time: Sep 13, 2025 4:07 PM (UTC+5:30) End time: Sep 13, 2025 4:09 PM (UTC+5:30) Build number: 3

Stop build Debug build Retry build



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

