

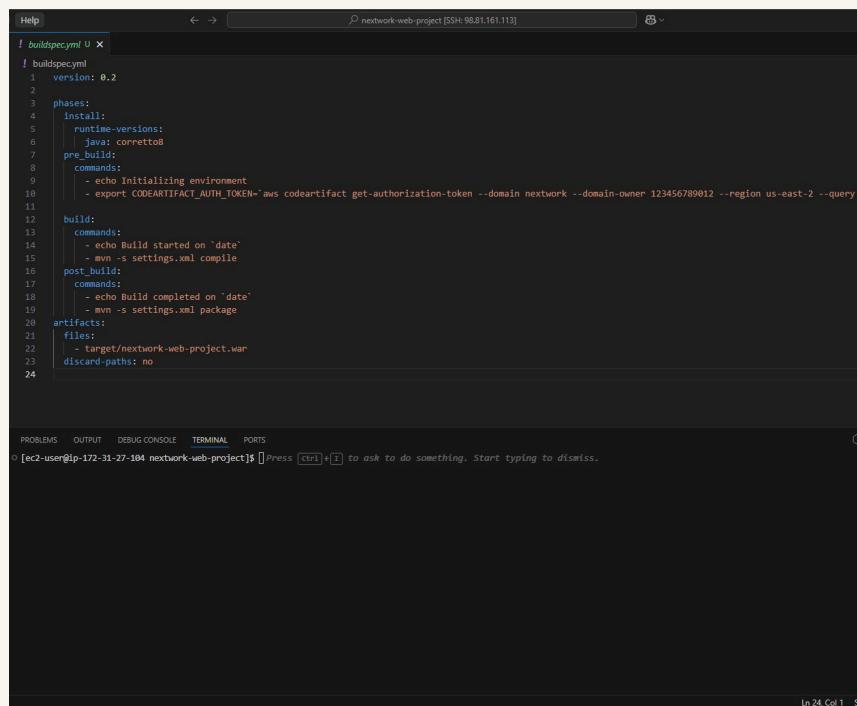


nextwork.org

Continuous Integration with CodeBuild



Gursimran Singh



```
Help      ← →   nextwork-web-project [SSH: 98.81.161.113]  ⌂ ↻
! buildspec.yml ✘
! 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 123456789012 --region us-east-2 --query
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[ec2-user@ip-172-31-27-104 nextwork-web-project]\$ Press **Ctrl**+**Z** to ask to do something. Start typing to dismiss.

Ln 24 Col 1



Introducing Today's Project!

In this project, I will demonstrate how to create and configure an AWS CodeBuild project from scratch. I'm doing this project to learn how to automate builds, integrate GitHub, define build steps with a `buildspec.yml` file, and run tests automatically.

Key tools and concepts

Services I used were AWS CodeArtifact, CodeBuild, IAM, CodeConnections, S3, and CloudWatch. Key concepts I learnt include setting up secure build pipelines, managing dependencies, role-based access, `buildspec` phases, and automating builds using CI/CD

Project reflection

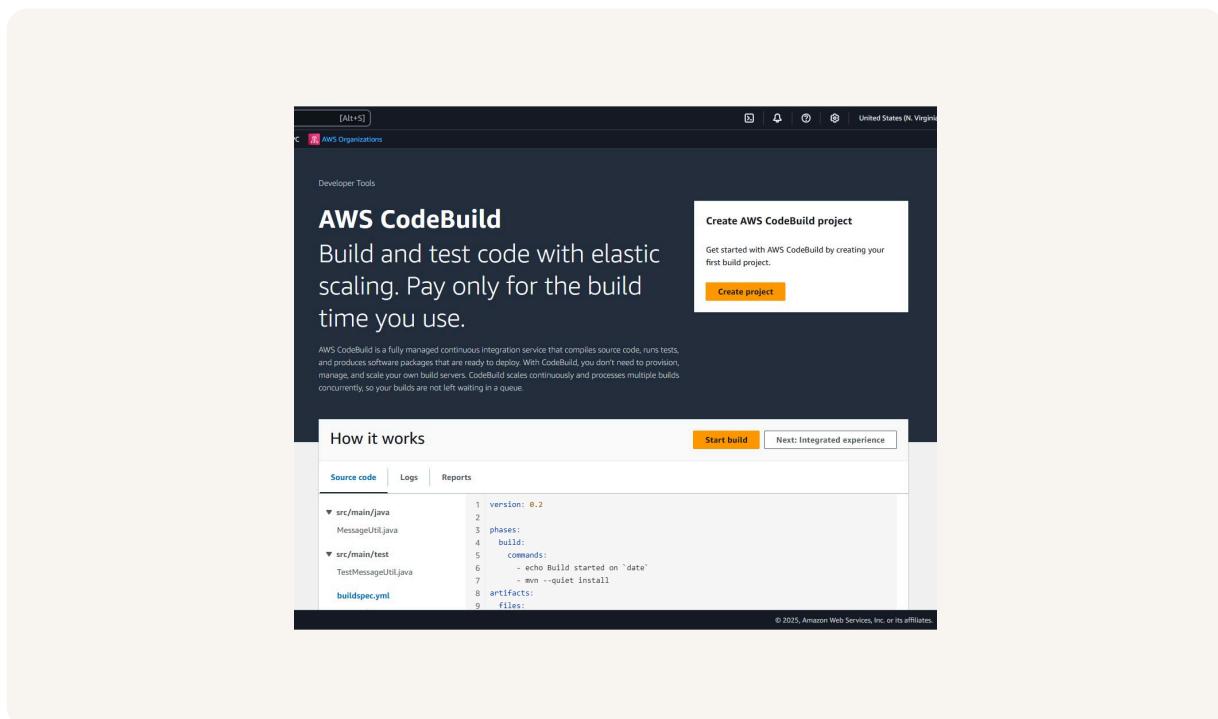
This project took me approximately 2 hours. The most challenging part was troubleshooting build failures due to missing permissions. It was most rewarding to see a successful automated build run after setting everything up correctly.

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 AWS DevOps skills and complete the pipeline setup.

Setting up a CodeBuild Project

CodeBuild is a continuous integration service, which means it automatically builds and tests code every time developers make changes. Engineering teams use it because it helps catch bugs early, speeds up development, and ensures code quality.

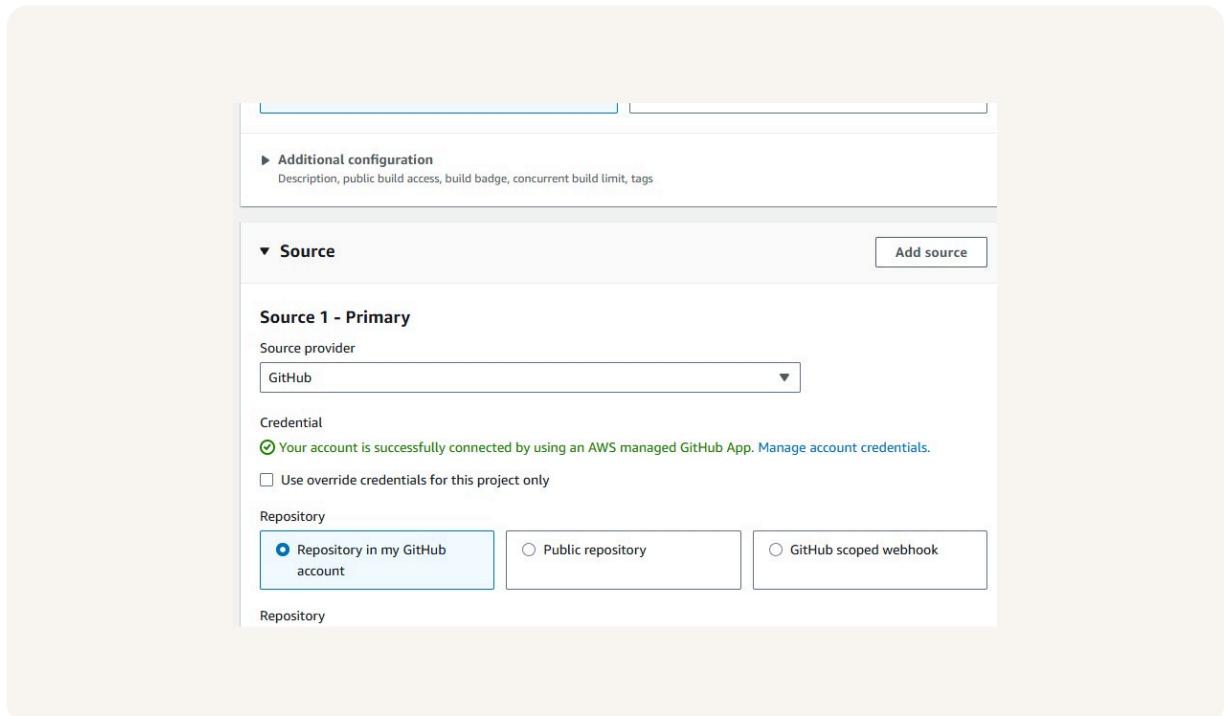
My CodeBuild project's source configuration means it knows where to get the code it needs to build, and I selected my GitHub repository as the source.



Connecting CodeBuild with GitHub

There are multiple credential types for GitHub, like personal access tokens and GitHub App. I used GitHub App because it's more secure, easier to manage, and allows fine-grained permissions, making it the recommended option by AWS.

The service that helped connect AWS with GitHub is **AWS CodeConnections**. It securely links my GitHub repository to services like CodeBuild, so changes in my code can automatically trigger builds and deployments.



A circular profile picture of a young man with a beard and a turban, wearing a dark jacket.

CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means it sets up the build environment with details like provisioning model (on-demand), OS (Amazon Linux), runtime (Standard), Corretto 8 image, and a new service role for secure access and permission

Artifacts

Build artifacts are the output files from the build process. They're important because they are the packaged results needed for deployment. My build process will create a `war` file. To store them, I created an S3 bucket as the artifact destination.

Packaging

When setting up CodeBuild, I also chose to package artifacts in a `zip` file because it compresses multiple build files into one, making storage and transfers more efficient and simplifying deployments to services like S3 or CodeDeploy.

A circular profile picture of a young man with dark hair and a beard, wearing a dark jacket over a light-colored shirt.

Gursimran Singh
NextWork Student

nextwork.org

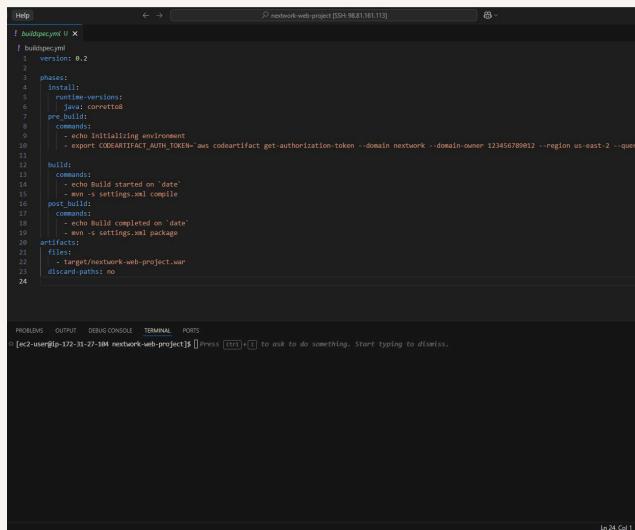
Monitoring

For monitoring, I enabled CloudWatch Logs, which is an AWS service that captures and stores logs from CodeBuild. It helps me track build progress, troubleshoot errors, and ensure everything runs smoothly during the CI/CD process.

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 tells CodeBuild exactly how to build and package the project, step by step.

The first two phases in my buildspec.yml file install dependencies and authenticate with CodeArtifact. The third phase compiles the code. The fourth phase uploads the generated build artifacts to the S3 bucket so they can be stored or used later.



A screenshot of a code editor window titled "buildspec.yml". The code is as follows:

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto8
  pre_build:
    commands:
      - echo Initializing environment
      - export CODEARTIFACT_AUTH_TOKEN=$(aws codeartifact get-authorization-token --domain nextwork --domain-owner 123456789012 --region us-east-2 --query token)
  build:
    commands:
      - echo Build started on '$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
```

The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. At the bottom, there is a message: "[ec2-user@ip-172-31-27-104 nextwork-web-project]\$ Press [CTRL]+[F] to ask to do something. Start typing to dismiss." and a status bar indicating "in 24 Col 1".

A circular profile picture of a young man with a beard and a turban, wearing a dark jacket.

Gursimran Singh
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, I updated the service role's IAM policy to grant CodeBuild permissions to access CodeArtifact and retrieve dependencies.

To resolve the second error, I updated the CodeBuild service role by attaching the correct IAM policy that grants access to CodeArtifact. When I built my project again, I saw the build succeed and the dependencies were downloaded correctly.

To verify the build, I checked my S3 bucket where the build artifacts are stored. I found a .zip file created by the build process. Seeing the artifact tells me the build ran successfully, everything compiled correctly, and it's ready for deployment.



Gursimran Singh
NextWork Student

nextwork.org

The screenshot shows the AWS CodeBuild console with the following details:

- Build status:** Succeeded
- Initiator:** iamadmin
- Build ARN:** arn:aws:codebuild:us-east-1:78910339323:1:build/nextwork-devops-cicd:0fc4ac3e-31bf-404d-94de-b564c1920dbf
- Start time:** Jul 7, 2025 8:10 PM (UTC+5:30)
- End time:** Jul 7, 2025 8:11 PM (UTC+5:30)
- Build number:** 9

Build logs:

```
Showing the last 714 lines of the build log. View entire log
```

No previous logs

```
1 [Container] 2025/07/07 14:40:41.844653 Running on CodeBuild On-demand
2 [Container] 2025/07/07 14:40:41.844659 Waiting for agent ping
3 [Container] 2025/07/07 14:40:42.347068 Waiting for DOWNLOAD_SOURCE
4 [Container] 2025/07/07 14:40:44.096319 Phase is DOWNLOAD_SOURCE
5 [Container] 2025/07/07 14:40:44.096320 SRC: /codebuild/output/src3154738093/github.com/gursimran531/nextwork-web-project
6 [Container] 2025/07/07 14:40:44.959172 VML: location is //codebuild/output/src3154738093/src/github.com/gursimran531/nextwork-web-project
7 [Container] 2025/07/07 14:40:44.959354 No commands found for phase name: install
8 [Container] 2025/07/07 14:40:44.961093 Setting HTTP client timeout to higher timeout for Github and GitHub Enterprise sources
```

© 2025, Amazon Web Services



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

