



Continuous Integration with CodeBuild

ZA

zakaria.belkacem94@gmail.com

```
1 buildspec.yml
2
3   version: 0.2
4
5   phases:
6     install:
7       runtime-versions:
8         java: corretto8
9     pre_build:
10      commands:
11        - echo Initializing environment
12        - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain nextwork --domain-owner 123456789012` .
13     build:
14      commands:
15        - echo Build started on `date`
16        - mvn -s settings.xml compile
17     post_build:
18      commands:
19        - echo Build completed on `date`
20        - mvn -s settings.xml package
21   artifacts:
22     files:
23       - target/nextwork-web-project.war
24   discard-paths: no
```

Introducing Today's Project!

In this project, I will demonstrate how to set up a continuous integration (CI) pipeline using AWS CodeBuild. I'm doing this to learn how to automate the build process for a web app.

Key tools and concepts

Services I used were AWS CodeBuild, S3, IAM, and CodeArtifact. Key concepts I learnt include CI pipelines, build specifications (`buildspec.yml`), artifact management, and troubleshooting permissions for automated builds.

Project reflection

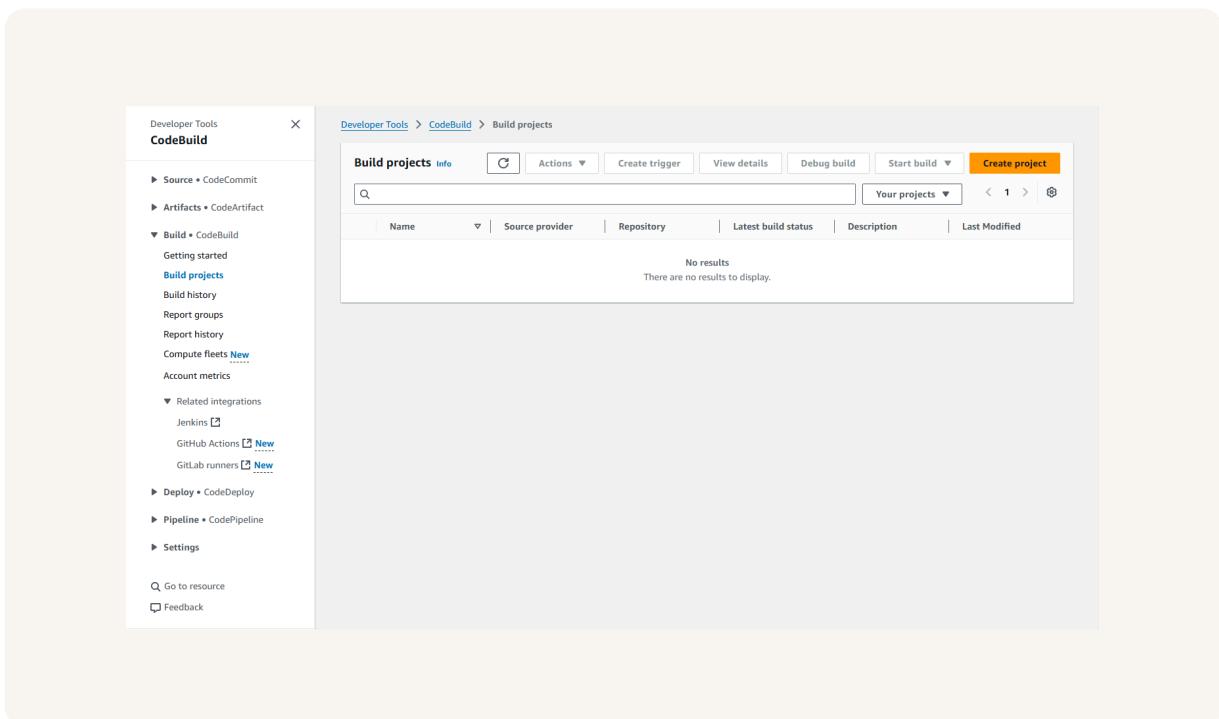
This project took me approximately 2 hours. The most challenging part was troubleshooting the IAM permissions for CodeArtifact access, but it was most rewarding to finally see the 'Build Succeeded' status and find my WAR file properly stored in S3!

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 tomorrow.

Setting up a CodeBuild Project

CodeBuild is a continuous integration (CI) service, which means it automatically builds, tests, and packages code when changes are pushed. Engineering teams use it to catch errors early, ensure consistency, and speed up releases.

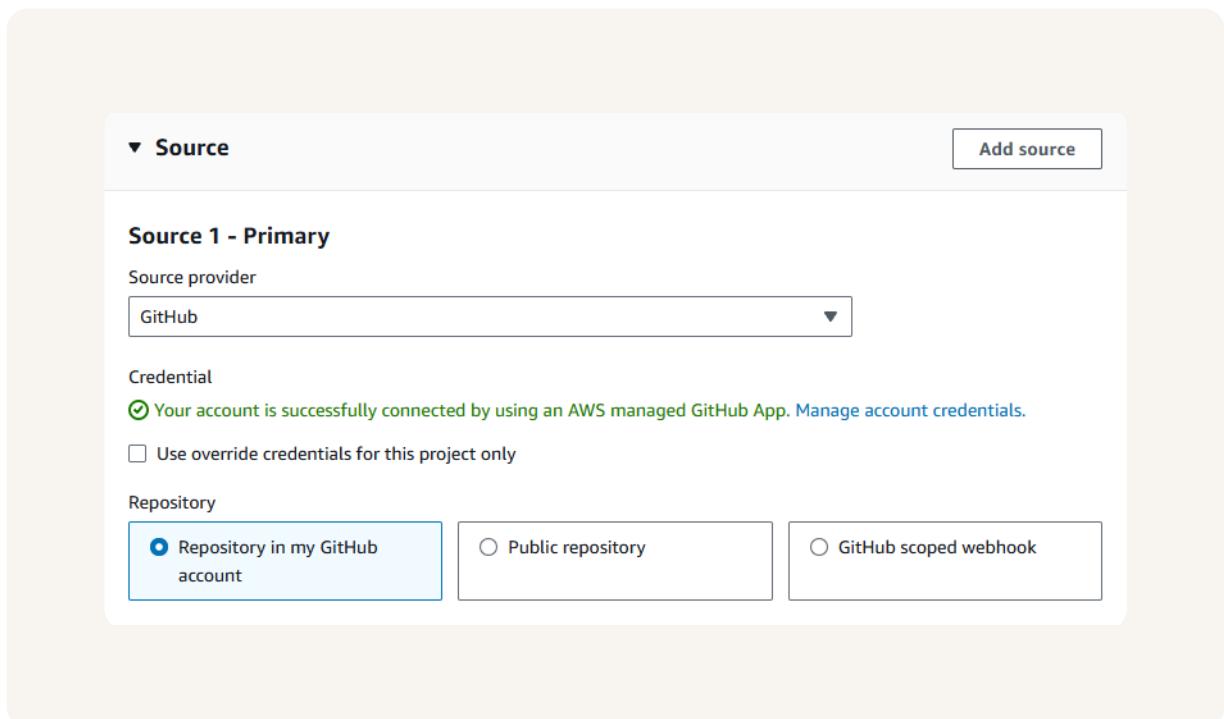
My CodeBuild project's source configuration means it pulls code directly from my connected GitHub repository, and I selected my 'nextwork-web-project' repo to automate builds on every code commit.



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 its the simplest and most secure option.

The service that helped connect CodeBuild to GitHub is AWS CodeConnections. It securely links AWS to GitHub without needing manual token management, simplifying authentication and enabling automatic code access for builds.



CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means it uses AWS-managed infrastructure; it includes settings like the Amazon Linux, Corretto 8 runtime, and on-demand provisioning model for reliable builds.

Artifacts

Build artifacts are the deployable outputs of the build process. They're important because they contain the ready-to-run application. My build process will create a WAR file, and to store them, I created an S3 bucket.

Packaging

When setting up CodeBuild, I chose to package artifacts in a ZIP file because it compresses the WAR file and any additional outputs into a single, smaller size, easily transferable package for storage in S3 and future deployments.

Monitoring

For monitoring, I enabled CloudWatch Logs, which is a monitoring service that collects and tracks logs from AWS services. it will record anything that happens during the build process.

buildspec.yml

My first build failed because our buildspec.yml doesn't exist in the root directory. a buildspec.yml file is needed because it acts like a step-by-step instruction manual for CodeBuild.

The first two phases in my buildspec.yml file are install and pre_build, The third phase in my buildspec.yml file is build The fourth phase in my buildspec.yml file is post_build.

```
# 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` 
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
```

Success!

My second build also failed, but with a different error that said UPLOAD_ARTIFACTS failed due to no matching artifact paths found in Phase details. To fix this we need to grant CodeBuild's IAM role permission to access CodeArtifact.

To resolve the second error, I fixed the permission to grant necessary permissions to CodeBuild. When I built my project again, I saw Succeeded status.

To verify the build, I checked my S3 bucket seeing the artifact tells me that my code was successfully compiled and packaged and that it was successfully uploaded to its destination.

Build status			
Status ⌚ Succeeded	Initiator zakme	Build ARN  arn:aws:codebuild:us-east-1:573515011508:build/nextwork-devops-cicd:747c8dc6-0b4c-4559-ad81-d0de7605103e	Resolved source version 3ddd75f20c04dfb27a5055facf2cd898832d7673
Start time May 11, 2025 4:41 PM (UTC-7:00)	End time May 11, 2025 4:42 PM (UTC-7:00)	Build number 6	



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

