

# Continuous Integration with CodeBuild

MA

Harish madhavan J U

```
GNU nano 8.3
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  pre_build:
    commands:
      - echo Initializing environment
      - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-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
```

# Introducing Today's Project!

In this project I will demonstrate how to set up codibuild for automating the build process in CI/CD pipeline. Im doing this project to learn about codeartifact which compresses all the necessary files into a single bundle so that we later deploy it.

## Key tools and concepts

Services I used were AWS IAM, EC2, CodeArtifact, CodeBuild. Key concepts I learnt include phases of build process in CI, how to troubleshoot package error in CodeArtifact, how to connect CodeBuild to Github repo.

## Project reflection

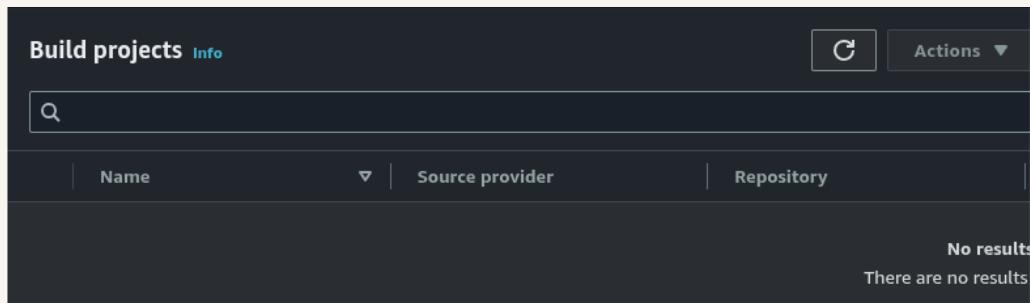
This project took me approximately. 2 hrs.The most challenging part was reconfiguring the codeartifact domain. It was most rewarding to see the green text saying build successfull.

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 service, which means it will automate the build process in the CI/CD pipeline. Engineering teams use it because it will ensure that necessary tests are run against new code which helps in identifying bugs that could break the application early and its pay-as-you-go model requires you to pay for the computation time spent during building packages, running tests.

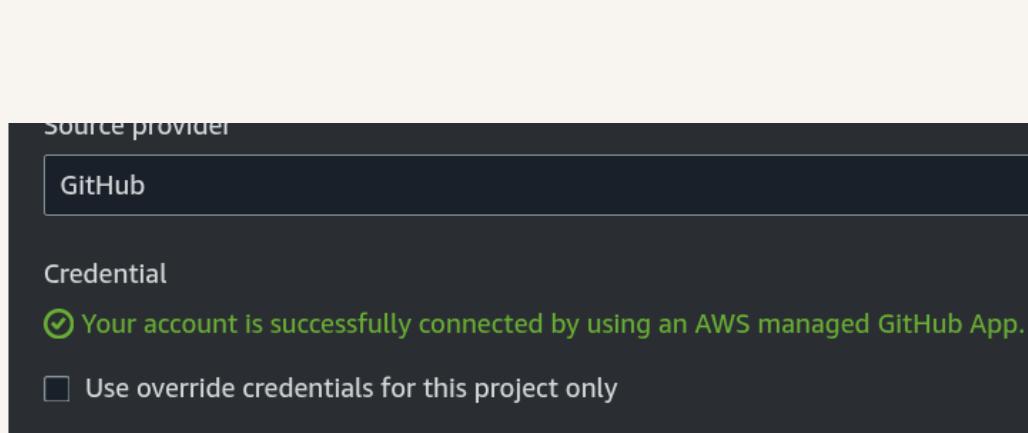
My CodeBuild project's source configuration means the place to look for project files and I selected Github because that contains our webapp's source repo.



# Connecting CodeBuild with GitHub

GitHub offers multiple credential types for integration, such as GitHub App, Personal Access Tokens, and OAuth. I chose GitHub App because it integrates seamlessly with AWS CodeConnections, which handles authentication and token management automatically. This makes it the simplest and most secure method for connecting AWS services to a GitHub repository.

The service that helped connect GitHub with CodeBuild is CodeConnections. It ensures that AWS services can securely access our public GitHub repository. It also helps in connecting AWS services with third-party services. With this, our CodeBuild will have secure access with the necessary permissions to access our repository.



# CodeBuild Configurations

## Environment

My CodeBuild project's Environment configuration means the setup required for the compute resources that compiles web app files into packages that can be deployed when we run our project. It includes settings like provisioning model, compute, operating system, image and service role. These settings will determine how our EC2 instance will be up and running for the build process.

## Artifacts

Build artifacts are the output of the build process of the CI/CD pipeline. They're important because it is the final file we will be deploying. My build process will create a single artifact .WAR file that can be unzipped and deployed by the web server easily. To store them, I created a S3 bucket because we need a reliable and scalable storage service to store our artifacts.

## Packaging

When setting up CodeBuild, I also chose to package artifacts in a Zip file because it reduces the size occupied in the bucket eventually reducing the cost, time required to upload the same to S3 bucket, it's a better way to organise all of your artifacts into a single folder.

## Monitoring

For monitoring, I enabled CloudWatch Logs, which is log monitoring service that collects and displays logs from all the aws services a log includes the command run, its output and error message if any this is crucial for debugging and finding out what went wrong

# buildspec.yml

My first build failed because codebuild wasn't able to download buildspec.yml file from the root directory. A buildspec.yml file is needed because it contains step by instruction on how proceed with the build without this codebuild will not know what to do so the build fails.

The first two phases in my buildspec.yml file are install and pre\_build which starts the process by installing the libraries required and fetch authorization token to access dependencies. The third phase in my buildspec.yml file is build phase where actuall build takes place. The fourth phase in my buildspec.yml file is post\_build where the code bundles every necessary item into a package.

```
GNU nano 8.3
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  pre_build:
    commands:
      - echo Initializing environment
      - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-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
```

# Success!

My second build also failed, but with a different error that said COMMAND\_EXECUTION\_ERROR. To fix this I am adding a IAM policy to the role such that it has permission to read resources from the codeartifact repo.

To resolve the second error, I added a policy called codeartifact-consumer-policy that provides permission to read resources from the codeartifact repo and fetch access token from STS. When I built my project again, I saw the build was successful and the artifact was uploaded to s3 bucket set for this purpose.

To verify the build, I checked my S3 bucket configured for storing the artifacts Seeing the artifact tells me the build was successfull and I can unzip the file for .war file which can be used in deployment .

MA

Harish madhavan J U

NextWork Student

[nextwork.org](http://nextwork.org)

devops-cicd:5877fbc5-2522-49a4-b167-13546ff38df3

[Stop build](#) [Debug build](#) [Retry build](#)

**Build status**

Status	Initiator	Build ARN
<span style="color: green;">Succeeded</span>	areus-devops-admin	<a href="#">arn:aws:codebuild:ap-southeast-1:67866448490:build/devops-77fbc5-2522-49a4-b167-138df3</a>
Start time	End time	Build number
Jun 22, 2025 2:50 PM (UTC+5:30)	Jun 22, 2025 2:52 PM (UTC+5:30)	7

[Build logs](#) [Phase details](#) [Reports](#) [Environment variables](#) [Build details](#) [Resources](#)

Name	Status	Context	Duration	Start time
------	--------	---------	----------	------------

# Automating Testing

In a project extension, I'm doing a automated test that runs along with the codebuild to validate the build . I added a test script that checks if there are necessary files to run the build.

To add the test script to the build process, I integrated the script with my builddspec.yml file which is a manual for codebuild on how to proceed with the build on integration if any test fails the build will automatically fail helping in identifying issue early on.

After pushing my code to GitHub, I ran the build again and I could see test passed logs in the build logs tab with this we can say that we have integrated testing with build process.

```
GNU nano 8.3
#!/bin/bash

echo "===== RUNNING SIMPLE TESTS ====="
echo "Test 1: Checking project structure..."
if [ -d "src" ]; then
    echo "✓ PASS: src directory exists"
else
    echo "✗ FAIL: src directory not found"
    exit 1
fi

echo "Test 2: Checking for web app files..."
if [ -f "src/main/webapp/index.jsp" ]; then
    echo "✓ PASS: index.jsp exists"
else
    echo "✗ FAIL: index.jsp not found"
    exit 1
fi

echo "Test 3: Simple validation test..."
echo "✓ PASS: This test always passes"

echo "===== ALL TESTS PASSED ====="
exit 0
```



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

