



Continuous Integration with CodeBuild



Saish Nar

```
! 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-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
25 |
```



Introducing Today's Project!

In this project, I will demonstrate how to use AWS CodeBuild to automate the build process in the CI/CD pipeline. I'm doing this project to learn process of turning the code into a package that can be deployed.

Key tools and concepts

Services I used include CodeBuild, CodeConnection, CodeArtifact, S3, EC2, GitHub, and VS Code. Key concepts learnt include the build process, resolving Git issues in the terminal, buildspec.yml, and using CodeConnections to connect AWS with GitHub.

Project reflection

This project took me approximately 3 hours to do including demo time. The most challenging part was resolving Git issues in the terminal. It was most rewarding to see the build succeed at the end!

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 in the next 24 hours.



Saish Nar
NextWork Student

NextWork.org

Setting up a CodeBuild Project

CodeBuild is a continuous integration service, which means it helps you to compile and package code. Engineering teams use it because CI services automate this process for us, if CI doesn't exist, engineer would have to manually compile and package.

My CodeBuild project's source configuration means where the code lives. and I selected GitHub! that's my web app's source code repository.

The screenshot shows the AWS CodeBuild console. At the top, there's a dark header with the AWS logo and navigation links. Below the header, the main content area has a dark background with white text. On the left, there's a sidebar with a 'Developer Tools' section and a 'Create AWS CodeBuild project' button. The main content area features a large heading 'AWS CodeBuild' and a sub-headline 'Build and test code with elastic scaling. Pay only for the build time you use.' Below this, there's a detailed description of what CodeBuild is and how it works. On the right, there's a 'How it works' section with tabs for 'Source code', 'Logs', and 'Reports'. The 'Source code' tab is active, showing a code editor with Java code. The code includes imports for `java.util.Date` and `org.junit.Test`, a class definition for `TestMessageUtil`, and a `buildspec.yml` file. The `buildspec.yml` file contains the following YAML configuration:

```
version: 0.2
phases:
  build:
    commands:
      - echo Build started on `date`
      - mvn --quiet install
artifacts:
  files:
```



Connecting CodeBuild with GitHub

There are multiple credential types for GitHub, like github app, authentication token or OAuth App,I used Github App because, it's simplest and most secure option. AWS manages the application and connection, reducing the need for me to handle of token

The service that helped connect my AWS environment with GitHub is AWS CodeConnections. CodeConnections also lets my connect my environment with my third party platforms, in this case, it makes sure that I have secure connection to GitHub.

The screenshot shows the 'Source' configuration page for AWS CodeConnections. The 'Source 1 - Primary' section is active. Under 'Source provider', 'GitHub' is selected. A green success message indicates that the account is successfully connected by using an AWS managed GitHub App. There is an option to 'Use override credentials for this project only'. Under 'Repository', the 'Repository in my GitHub account' radio button is selected, and the URL 'https://github.com/SaishNar/nextwork-web-project' is entered in the input field. Other repository options like 'Public repository' and 'GitHub scoped webhook' are available but not selected.



CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means the environment set up for the compute power that will be compiling and compressing files when I run my CodeBuild project. It includes settings like provisioning model, compute, OS, service role.

Artifacts

Build artifacts are files. They're important because they represent artifacts that need to be used later on in the CI/CD pipeline. My build process will create a compressed file that represents my web app. To store them, I created a S3 bucket.

Packaging

When setting up codeBuild, I also chose to package artifacts in a Zip file - this helps with package management as files are organized neatly in a single executable file. It also helps to compress the size of my compiled web app!



Saish Nar
NextWork Student

NextWork.org

Monitoring

For monitoring, I enabled CloudWatch Logs, which is a service that will note down commands that are run and any errors that happen. This is extremely helpful for troubleshooting!



buildspec.yml

My first build failed because CodeBuild could not find the buildspec.yml file in my source code root directory. A buildspec.yml file is needed because it tells CodeBuild how to run the build process.

The first two phases in my buildspec.yml file installs Java and gets me access to CodeArtifact. The third phase in my buildspec.yml file compiles the web app. The fourth phase in my file packages the web app into a single, compressed artifact.

```
! buildspec.yml U •
! 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-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
25 |
```



Success!

My second build also failed, but with a different error that said I failed to run an error while executing the command to compile my web app mvn -s settings.xml compile. To fix this I need new policy to CodeBuild that grants CodeArtifact permission

To resolve the second error, I attached a policy for accessing CodeArtifact to my CodeBuild role. When I built my project again, I saw a success! With the privilege to CodeArtifact granted by the role, CodeBuild now has access to the repository.

To verify the build, I checked my artifacts bucket in S3. Seeing the build artifact tells me that CodeBuild was success! The build process compiled the code from the source repo [GitHub], compressed it into a file and stored that file in my S3 bucket

The screenshot shows a build history entry for a specific build ID. The top bar displays the build ID: `nextwork-devops-cicd:20b41d01-41d1-4fab-b0a8-984e61eaad5d`. Below the ID are three buttons: `Stop build`, `Debug build`, and `Retry build`. A large button labeled `Build status` is present. The main content area contains the following details:

Status	Initiator	Build ARN	Resolved source version
⌚ Succeeded	Saish-IAM-Admin	arn:aws:codebuild:ap-south-1:8913:77396307:build/nextwork-devops-cicd:20b41d01-41d1-4fab-b0a8-984e61eaad5d	21ce94cac258acf71f249489a85966287e60a2c2
Start time	End time	Build number	
Apr 16, 2025 12:58 PM (UTC+5:30)	Apr 16, 2025 12:59 PM (UTC+5:30)	6	



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

