



Package an App with CodeBuild

G Gloria

nextwork-web-build

Actions ▾ Create trigger Edit Clone Debug build

Configuration

Source provider GitHub	Primary repository Miss-Gloria/nextwork-devops-webapp	Artifacts upload location nextwork-build-artifacts-gloria
Public builds Disabled		



Introducing today's project!

What is AWS CodeBuild?

AWS CodeBuild is a service that automates compiling, testing, and packaging code. It streamlines builds, saves time, and integrates with AWS services, making development efficient and scalable.

How I used CodeBuild in this project

I used AWS CodeBuild to automate compiling my web app, packaging it into a WAR file, and storing the file in an S3 bucket. It replaced manual build commands, making the process faster, efficient, and ready for deployment.

One thing I didn't expect in this project was...

One thing I didn't expect in this project was having to integrate GitHub since CodeCommit was deprecated. It was a new experience figuring out how to adjust the workflow and connect GitHub as the source for CodeBuild seamlessly.

This project took me...

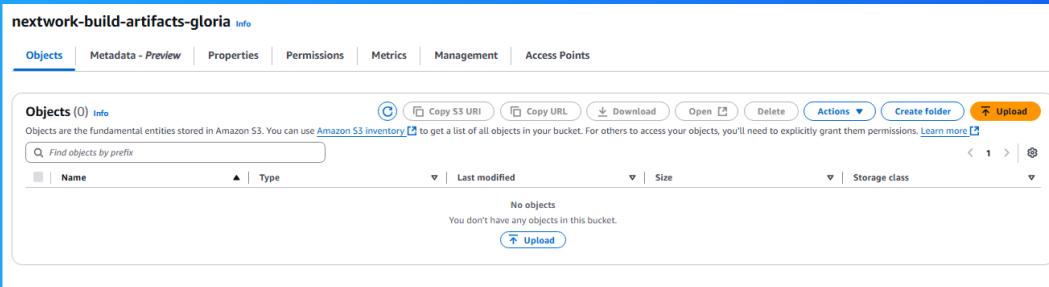
This project took me about 30 minutes to complete. Setting up AWS CodeBuild, integrating GitHub, and automating the build process were straightforward with clear steps, making it efficient to finish within this time.

Set up an S3 bucket

I started my project by creating an S3 bucket because it will store the file created during the build process. This file is needed for the server to run the web app, and the bucket keeps it safe and ready for deployment.

The key artifact that this S3 bucket will capture is called a WAR file. It's a packaged file containing everything a server needs to host and run the web app, like code, resources, and configurations.

This artifact is important because it bundles all the files, code, and resources needed for the server to host and run the web app. Without it, the server wouldn't have the complete package required to deploy the application.



Set up a CodeBuild project

Source

My CodeBuild project's Source configuration means the location where the code is stored for the build process, and I selected GitHub as the source. This allows CodeBuild to pull the latest code from my GitHub repository for compilation and packaging.

Environment

My CodeBuild project's Environment configuration means the settings and runtime environment where the build process runs, including the operating system. I selected a managed image provided by AWS with runtime support for Java to build my project.

Artifacts

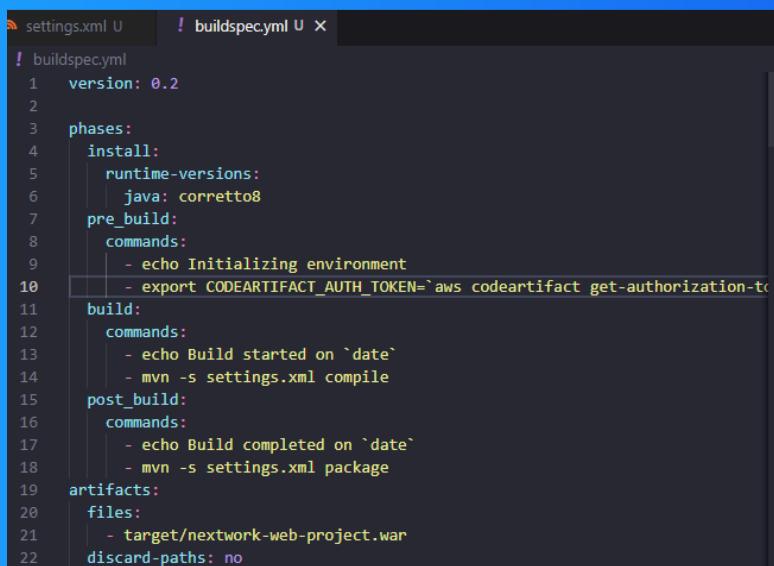
My CodeBuild project's Artifacts configuration means the output files generated during the build process that are stored for later use. I selected an S3 bucket to store the WAR file, which is the key artifact needed to deploy my web app.

Logs

My CodeBuild project's Logs configuration means the settings for tracking and storing the build process's output and errors. I selected Amazon CloudWatch Logs to monitor the build's progress and troubleshoot any issues that occur during the process.

Create a buildspec.yml file

I created a buildspec.yml file in my project because it tells CodeBuild how to build my web app. It automates tasks like setting up Java, compiling the code, packaging it into a WAR file, and saving it as an artifact, ensuring a consistent process.



```
! settings.xml U ! 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 --region us-east-1 --domain nextwork --repository nextwork-web-project --profile Gloria`
11    build:
12      commands:
13        - echo Build started on `date`
14        - mvn -s settings.xml compile
15    post_build:
16      commands:
17        - echo Build completed on `date`
18        - mvn -s settings.xml package
19    artifacts:
20      files:
21        - target/nextwork-web-project.war
22    discard-paths: no
23
```



Create a CodeBuild build project

My buildspec.yml file has four stages

The first two phases in my buildspec.yml file are install and pre_build. The install phase sets up the Java runtime, and the pre_build phase initializes the environment and retrieves an authorization token from CodeArtifact for secure access.

The third phase in my buildspec.yml file is the build phase. It compiles the code using Maven and the settings.xml file. This phase starts the actual build process and prepares the web app for packaging into a WAR file.

The fourth phase in my buildspec.yml file is the post_build phase. It packages the compiled code into a WAR file using Maven and the settings.xml file. This phase finalizes the build process and creates the key artifact needed for deployment.



Modify CodeBuild's IAM role

Before building my CodeBuild project, I modified its service role first.

My CodeBuild project's service role was first created when I set up the build environment in CodeBuild and selected New service role. This role was automatically generated to provide CodeBuild with the necessary permissions to run the build process.

I attached a new policy called `codeartifact-nextwork-consumer-policy` to my CodeBuild role. This policy allows CodeBuild to access the CodeArtifact repository and fetch the necessary dependencies for building my web app.

Attaching this policy means CodeBuild can securely access the CodeArtifact repository to fetch dependencies needed for the build process. This ensures the web app can be compiled and packaged successfully without permission issues.

The screenshot shows the AWS IAM Permissions page for a role. The 'Permissions' tab is selected. Under 'Permissions policies', there are four managed policies listed: 'codeartifact-nextwork-consumer-policy' (selected), 'CodeBuildBasePolicy-nextwork-web-build-us-east-1', 'AWSLambdaBasicExecutionRole', and 'AmazonCloudWatchLogsFullAccess'. The 'codeartifact-nextwork-consumer-policy' is described as 'Customer managed'. The 'Attached entities' column shows '1' for each policy. There are buttons for 'Simulate', 'Remove', and 'Add permissions'.

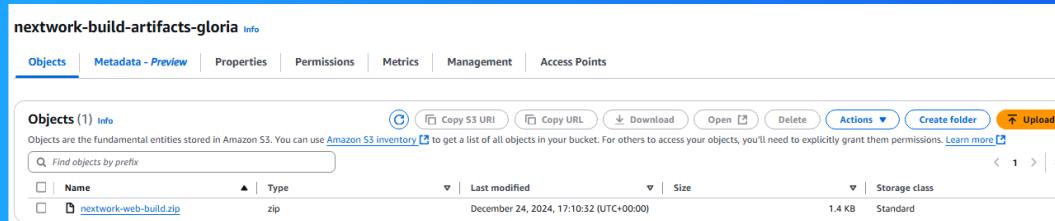
My first project build □

To build my project, all I had to do was configure my CodeBuild project, ensure the `buildspec.yml` file was in the repository, and then click the "Start build" button in the AWS CodeBuild console to initiate the automated build process.

The build process in CodeBuild took 2 minutes to complete. It efficiently compiled the code, packaged it into a WAR file, and stored the artifact in the S3 bucket, saving time and automating the entire process.

Once the build was complete, I checked the S3 bucket configured in the CodeBuild project. Yes, the file created was a WAR file, which contains all the necessary components for deploying and running the web app on a server.

I saw the nextwork-web-build.zip file in the S3 bucket, which verified that the build was completed successfully and the application was packaged and ready for deployment.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

