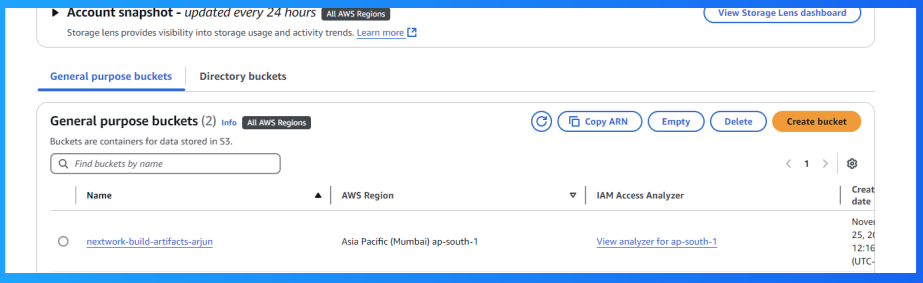
NextWork.org

Package an App with CodeBuild

AJ ajv.singh3107@gmail.com



AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Introducing today's project!

What is AWS CodeBuild?

AWS CodeBuild is a managed service that automates building, testing, and packaging code. It helps streamline the development process by eliminating the need for self-hosted build servers, offering scalability, flexibility with other services.

How I used CodeBuild in this project

I used AWS CodeBuild to automate the build process of my web app, compiling the code and creating a WAR file. It fetched dependencies from CodeArtifact, packaged everything needed for deployment, and stored the resulting WAR file in an S3 bucket.

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

One thing I didn’t expect was how smoothly the integration between CodeBuild and CodeArtifact worked. Once configured, it automatically fetched the dependencies and built the project without any manual intervention, saving a lot of time.

This project took me...

The project took around 2 hours. I set up the S3 bucket, configured CodeBuild, attached IAM roles, and tested the build process. The configuration was straightforward, and AWS automated much of the setup, streamlining the overall process.

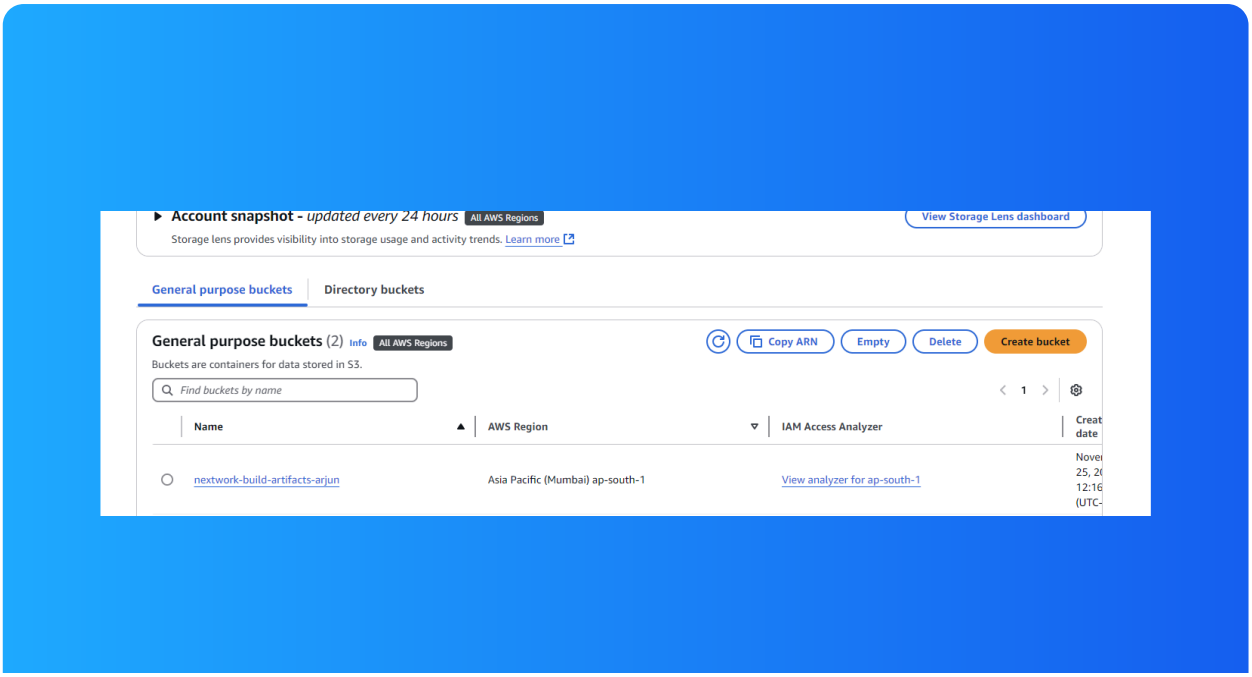
AJ
ajv.singh3107@gmail.com
NextWork Student

Set up an S3 bucket

I started my project by creating an S3 bucket because it will store the WAR file, which is the key build artifact. This file is created during the build process and will be used to deploy the web app to servers automatically using CodeBuild.

The S3 bucket is being created to store the WAR file, which is the key build artifact. The build process involves compiling the code, running tests, and packaging everything into a WAR file, which servers can use to run the web app automatically with

This artifact is important because the WAR file contains all components of the web app bundled together, allowing servers to easily unpack and run it, automating deployment and ensuring consistent, reliable delivery of the app.



AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Set up a CodeBuild project

Source

My CodeBuild project's Source configuration means specifying where the source code is stored. I selected the S3 bucket where the code is stored, so CodeBuild can retrieve it, compile the code, and create the WAR file for deployment.

Environment

My CodeBuild project's Environment configuration means setting the build environment for compiling the code. I selected the Java environment image, ensuring the necessary tools and runtime are available to build and package my web app.

Artifacts

My CodeBuild project's Artifacts configuration means defining where the output of the build (e.g., WAR file) is stored. I selected an S3 bucket to keep the build artifact, making it accessible for future deployment or processing steps in the workflow

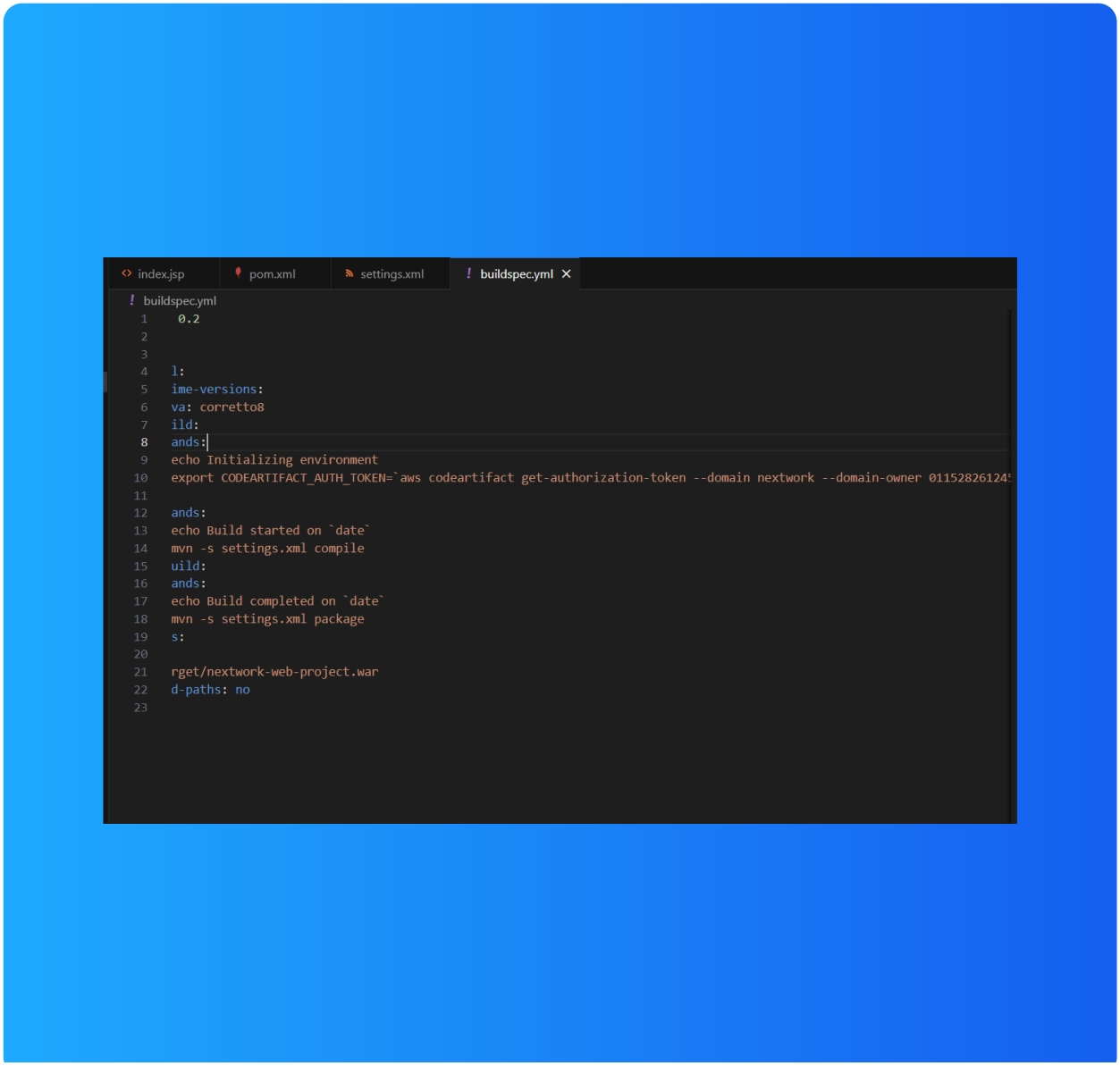
Logs

My CodeBuild project's Logs configuration means specifying where the build logs are stored for tracking and troubleshooting. I selected Amazon CloudWatch to monitor the build process and easily access logs for debugging or auditing purposes.

AJ
ajv.singh3107@gmail.com
NextWork Student

Create a buildspec.yml file

I created a buildspec.yml file in my project because it defines the build process for CodeBuild, including commands for compiling, packaging, and creating a WAR file, ensuring the application is built automatically and ready for deployment.



AJ
ajv.singh3107@gmail.com
NextWork Student
NextWork.org

Create a CodeBuild build project

My buildspec.yml file has four stages

The first two phases in my buildspec.yml file are the `install` phase, where I define runtime versions like Java, and the `pre_build` phase, where I initialize the environment and fetch the authentication token to access dependencies in AWS CodeArtif

The third phase in my buildspec.yml file is the `build` phase, where the actual compilation happens. In this phase, I run the `mvn compile` command to build the project, starting the process of transforming the code into a deployable web app.

The fourth phase in my buildspec.yml file is the `post_build` phase.In this phase,I run the `mvn package` command to create the final build artifact (the WAR file), marking the completion of the build process and preparing the artifact for deployment

AJ
ajv.singh3107@gmail.com
NextWork Student

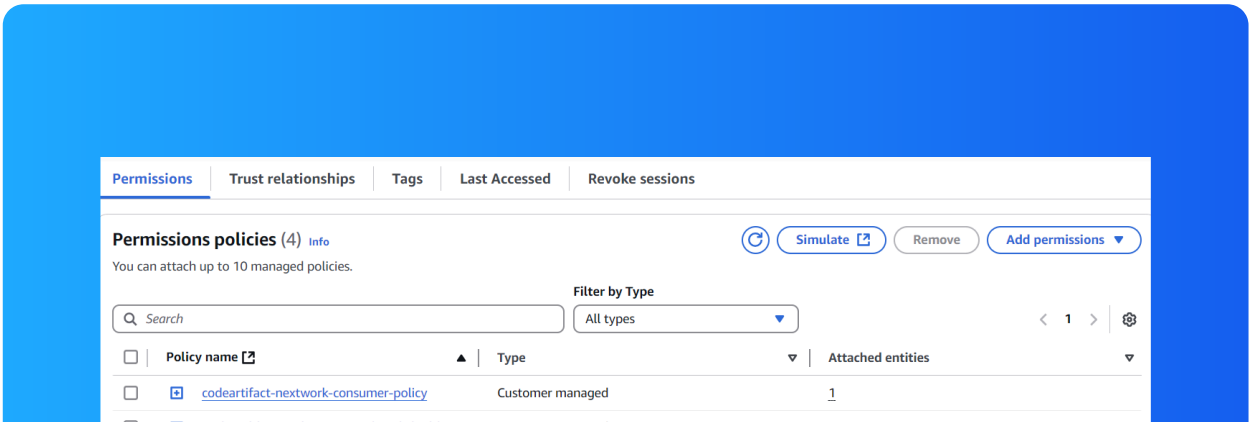
Modify CodeBuild's IAM role

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," granting the minimum permissions required to run the build process but not access to CodeArtifact.

I attached the `codeartifact-nextwork-consumer-policy` to my CodeBuild role. This policy grants the necessary permissions for CodeBuild to access and fetch packages from the CodeArtifact repository, enabling the successful completion of the build.

Attaching this policy means CodeBuild can now access the CodeArtifact repository to retrieve the required dependencies for the build process. This ensures that the build completes successfully by allowing CodeBuild to fetch and use the necessary pkg.



AJ
ajv.singh3107@gmail.com
NextWork Student

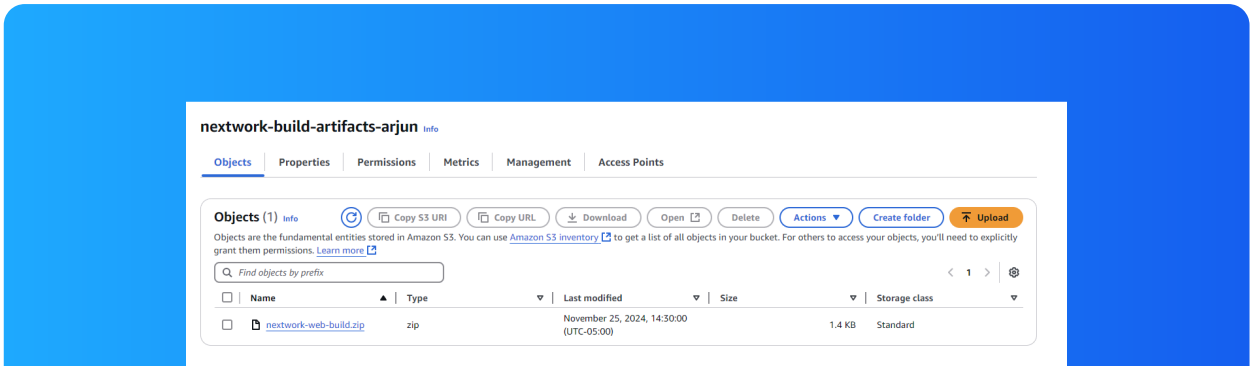
My first project build

To build my project, all I had to do was go to the CodeBuild console, select the nextwork-web-build project, and click "Start build." I then monitored the logs as the build process completed, which resulted in the WAR file being stored in the S3 buck

The build process in CodeBuild took about 5 minutes to complete. During this time, the project was compiled, tested, and packaged into a WAR file, which was then stored in the S3 bucket for deployment.

Once the build was complete, I checked my S3 bucket (nextwork-build-artifacts-arjun) and found that a ZIP file named nextwork-web-project.zip was created. Inside this ZIP file, there was a WAR file containing the packaged web app ready for deployment

I saw a ZIP file named nextwork-web-project.zip, which verified that the build was completed successfully. Inside the ZIP file was the WAR file, confirming that the web app was properly packaged and ready for deployment.



Everyone should be in a job they love.

Check out nextwork.org for more projects