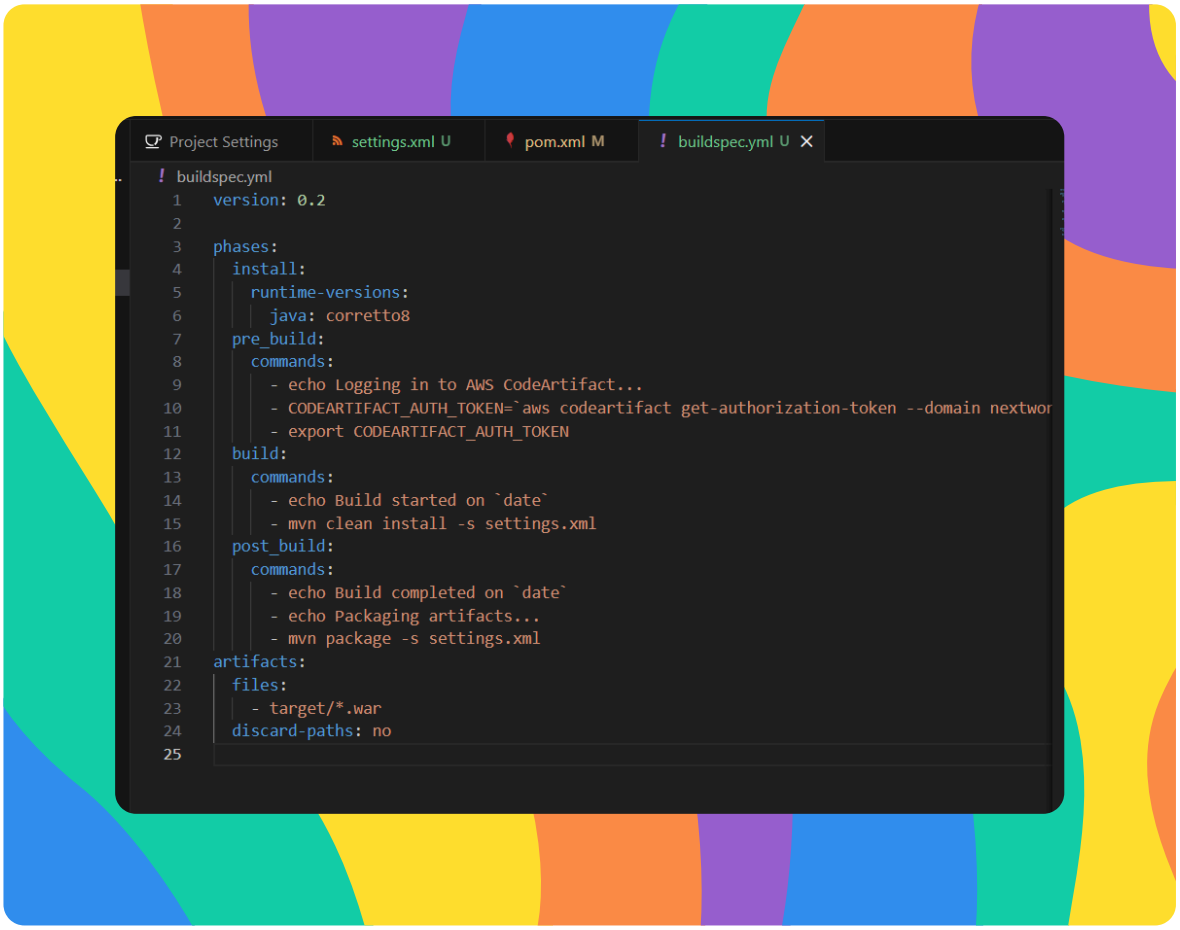
nextwork.org

Continuous Integration with CodeBuild

Mohammed Furqanuddin



Mohammed Furqanud...
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use AWS CodeBuild to automatically build and test a Java web application by connecting it to a GitHub repository. I’m doing this project to learn how build automation works in real-world DevOps workflows and to understand the role of CI/CD in delivering reliable and scalable applications

Key tools and concepts

Services I used: Amazon EC2, GitHub, AWS CodeArtifact, AWS CodeBuild, Amazon S3, AWS CodeConnections, and CloudWatch Logs. Key concepts I learned: CI/CD, build automation, secure dependency management, IAM permissions, and artifact storage.

Project reflection

This project took me around 2–3 hours to complete. The most challenging part was configuring IAM permissions and integrating CodeArtifact with CodeBuild. The most rewarding part was seeing a successful build and the artifact stored in S3, confirming the CI pipeline worked end to end.

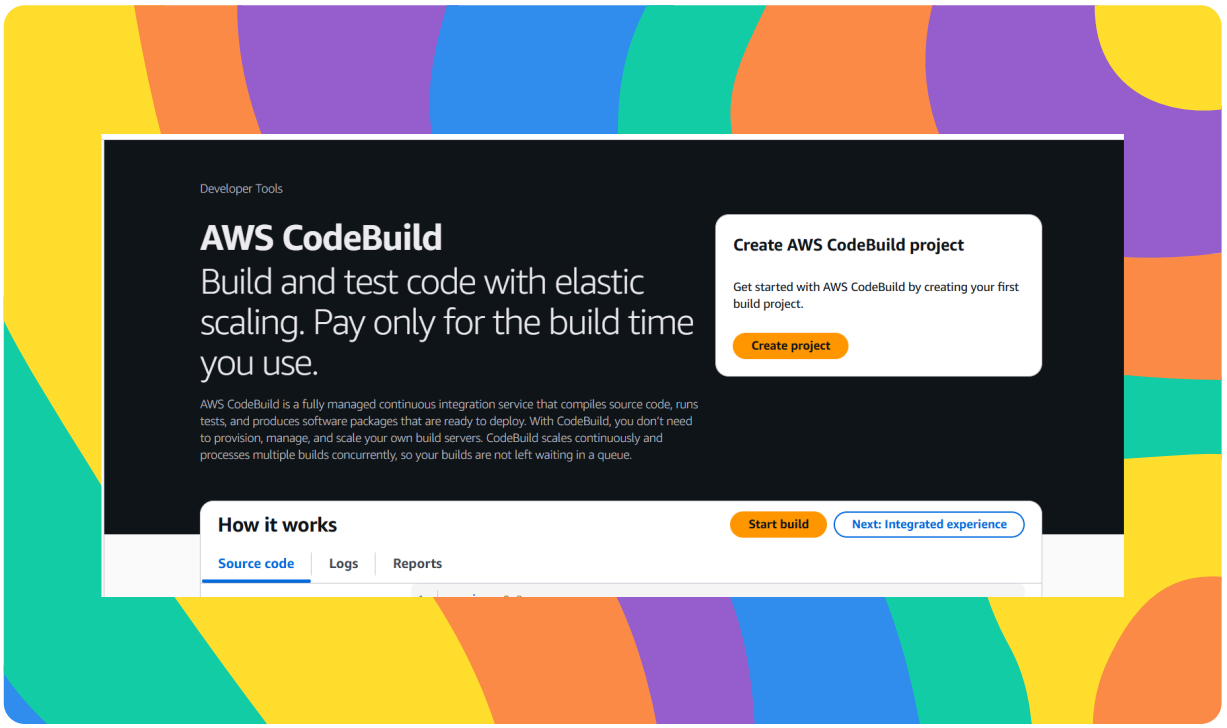
This project is part four of my DevOps series where I’m building a CI/CD pipeline. I’ll be working on the next project soon to continue learning and improving my DevOps skills.

Mohammed Furqanud...
NextWork Student

Setting up a CodeBuild Project

A CI service automatically builds and tests code whenever changes are pushed. Teams use it to catch errors early, keep builds consistent, and speed up development.

My CodeBuild project's source configuration means where CodeBuild pulls the application code from to run the build. I selected my GitHub repository as the source so that CodeBuild can automatically fetch the latest code whenever a build is triggered.

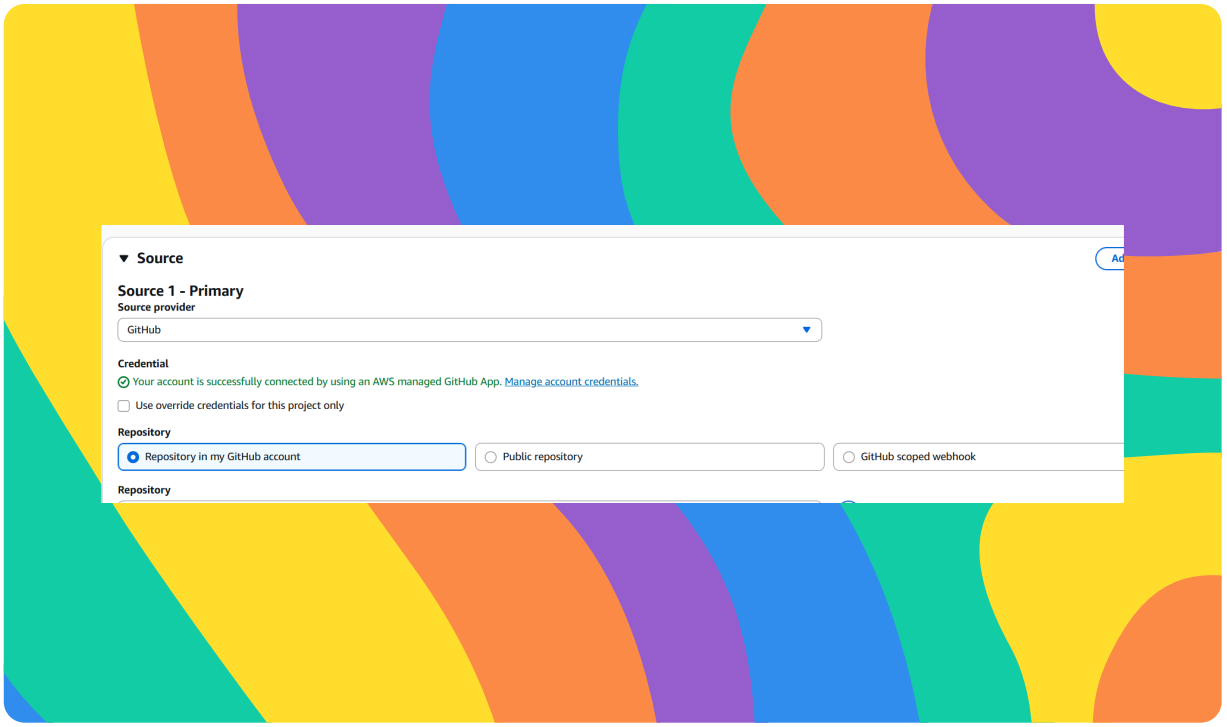


Mohammed Furqanud...
NextWork Student

Connecting CodeBuild with GitHub

There are multiple credential types for connecting GitHub to AWS, such as personal access tokens and OAuth tokens. I used GitHub App because it is more secure and scalable. A GitHub App provides fine-grained permissions, automatic token rotation, and better control over repository access, making it the recommended and best-practice option for CI/CD integrations

The service that helped connect AWS with GitHub is AWS CodeConnections. It provides a secure and managed way to authorize AWS services like CodeBuild to access a GitHub repository. By using CodeConnections, we avoided hardcoding credentials and enabled safe, seamless access to source code for our CI pipeline.



CodeBuild Configurations

Environment

In this step, I configured the CodeBuild environment by choosing an on-demand, managed Amazon Linux image with Corretto 8 because it provides a ready-to-use, cost-effective setup to build my Java application securely.

Artifacts

Build artifacts are the final output of the build process. They are important because they are the deployable files used for deployment. My build creates a WAR file, and I created an S3 bucket to store this artifact safely.

Packaging

When setting up CodeBuild, I chose to package build artifacts in a ZIP file because it bundles all required output files into a single, compressed package. This makes artifacts easier to store in S3, faster to transfer, and simpler to use in later stages like deployment or release.

Monitoring

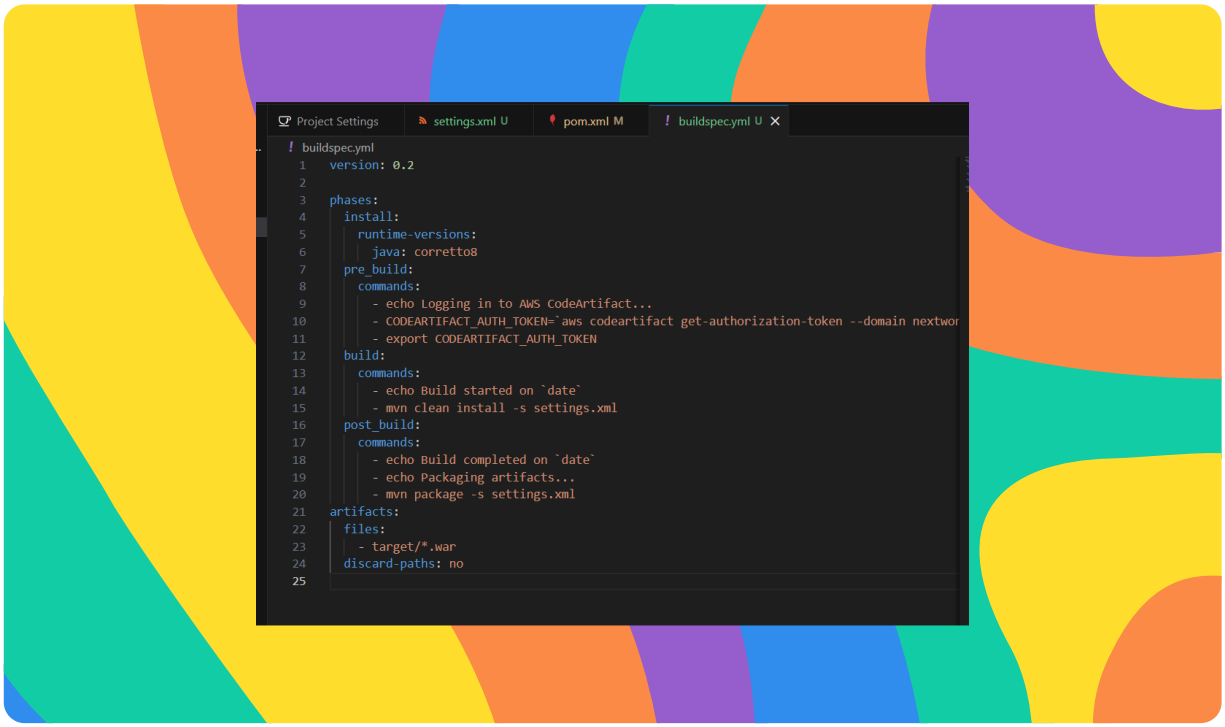
For monitoring, I enabled CloudWatch Logs, which captures and stores build logs so I can track build progress, debug errors, and understand what happens during each CodeBuild run.

Mohammed Furqanud...
NextWork Student

buildspec.yml

My first build failed because CodeBuild didn't know what commands to run. A buildspec.yml file is needed because it defines the build steps, such as installing dependencies, compiling the code, and packaging the application.

The first two phases prepare the environment and dependencies. The third phase builds the application. The fourth phase runs final steps like packaging and finishing the build.



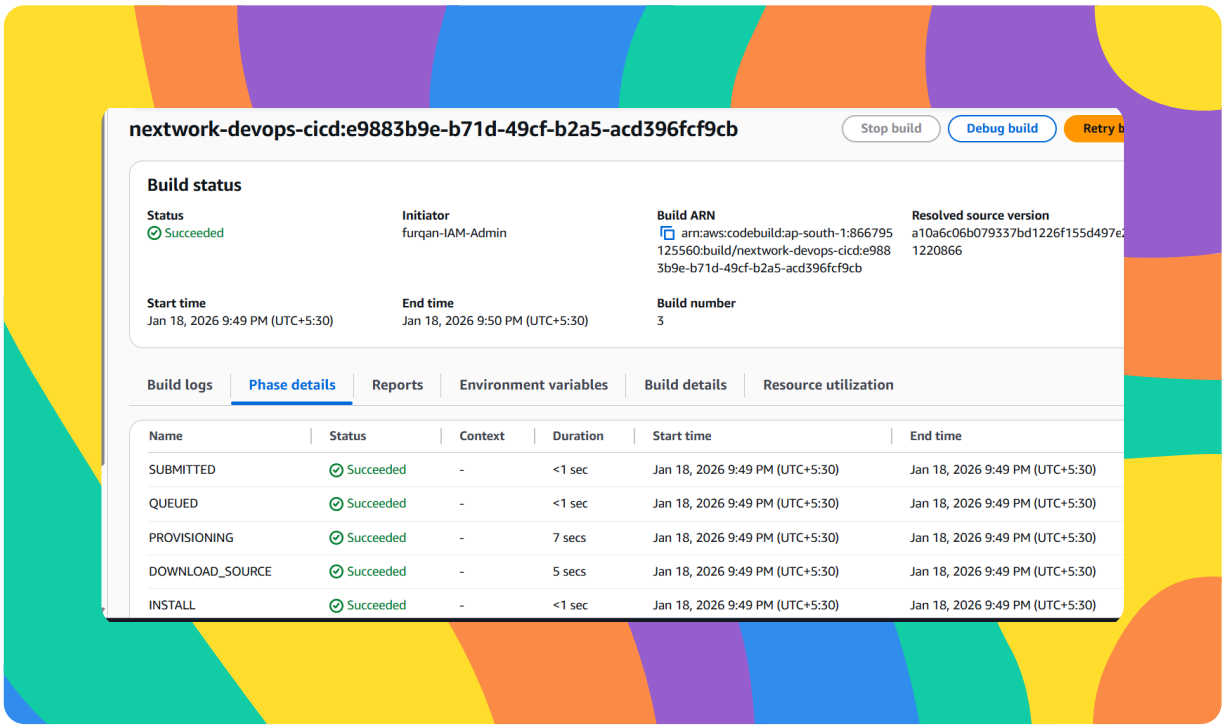
Success!

My second build also failed, but with a different error that said access was denied to AWS CodeArtifact. This happened because the CodeBuild service role did not have the required IAM permissions to authenticate and download dependencies from CodeArtifact.To fix this, I will have to update the CodeBuild service role by granting it the necessary CodeArtifact access permissions, after which the build will run successfully.

To resolve the second error, I updated the CodeBuild service role to grant permission to access AWS CodeArtifact. When I built my project again, I saw the build complete successfully and the artifacts generated without any access errors.

I checked the Amazon S3 bucket and confirmed that nextwork-devops-cicd-artifact.zip was uploaded. This shows the build ran successfully, the code was packaged correctly, and the artifact is ready for the next step.

Mohammed Furqanud...
NextWork Student



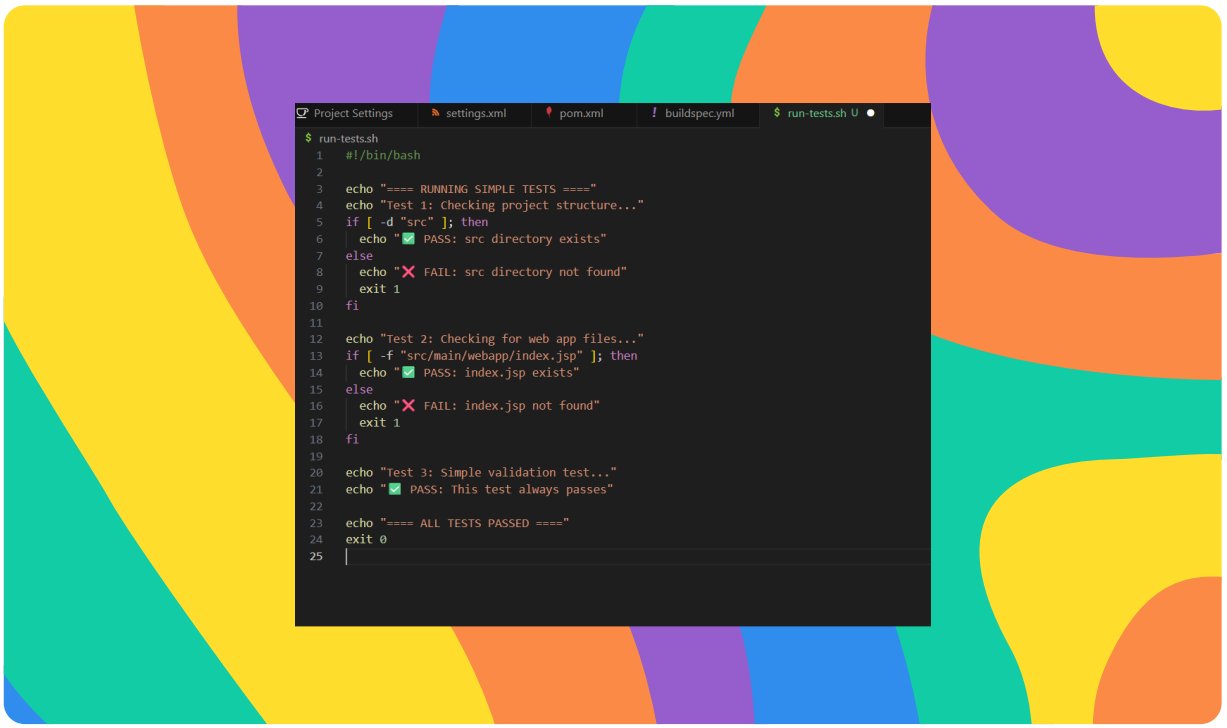
Automating Testing

In a project extension, I added a simple test script to validate the basic structure of my web application. The script checks that the project directory structure exists, verifies the presence of the index.jsp web file, and runs a basic validation test. This ensures the project is set up correctly before moving forward in the CI process.

To add the test script to the build process, I included it in my project repository and updated the buildspec.yml file to execute the script during the build phase. This ensured the tests run automatically as part of the CI pipeline and fail the build if any checks do not pass.

After pushing my code to GitHub, I ran a CodeBuild build and could see my test script executing in the build logs. The logs showed each test step passing successfully, which confirmed that testing was automatically triggered and verified as part of the CI process.

Mohammed Furqanud...
NextWork Student



The place to learn & showcase your skills

Check out nextwork.org for more projects