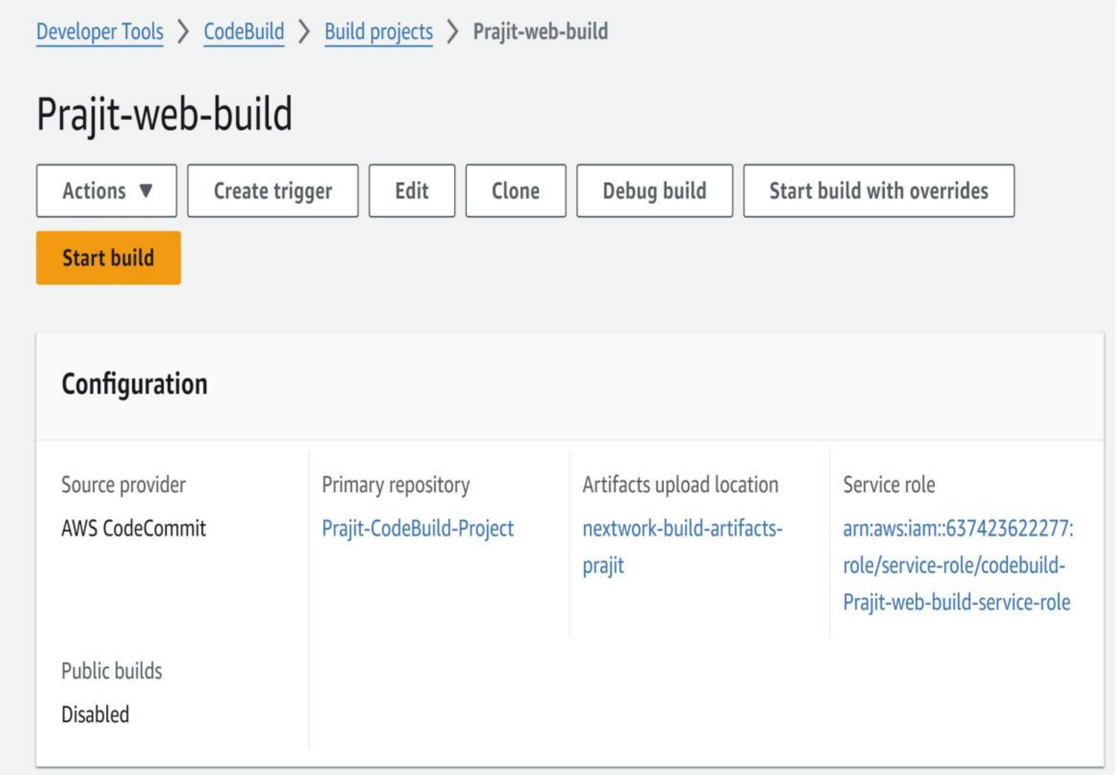
NextWork.org

Package an App with CodeBuild

Prajit Venkatachalam

https://www.linkedin.com/in/prajit-venkatachalam-435b2a150/



Developer Tools > CodeBuild > Build projects > Prajit-web-build
Actions ▼
Create trigger
Edit
Clone
Debug build
Start build with overrides
Start build

Prajit-web-build

Configuration

Source provider AWS CodeCommit	Primary repository Prajit-CodeBuild-Project	Artifacts upload location nextwork-build-artifacts-prajit	Service role arn:aws:iam::637423622277:role/service-role/codebuild-Prajit-web-build-service-role
Public builds Disabled			

Introducing today's project!

What is AWS CodeBuild?

AWS CodeBuild is a continuous integration (CI) service that compiles source code, runs tests, and produce software packages ready for deployment. It is useful for automating the testing and packaging of applications during deployment and development.

How I used CodeBuild in this project

In today's project, I used AWS CodeBuild to create a build artifact, a WAR file, which is a bundle of packages that a server needs to run my web app. I created a buildspec.yml file to instruct the server on how to build the project, and an Amazon S3 bucket to store the build artifacts generated during the build process.

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

One thing I did not expect in this project was encountering an error with accessing Cloud9, using the S3 bucket to store the artifacts, and creating the IAM role for CodeBuild.

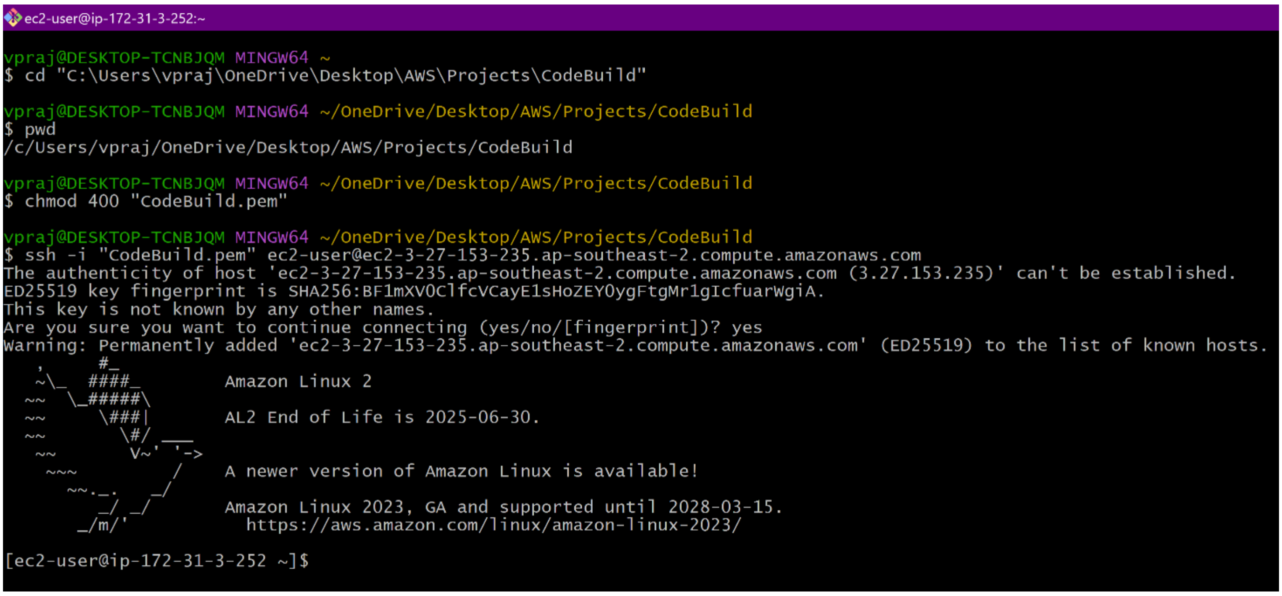
This project took me...

This project took me 3 hours to complete include report writing and other pre setups.

Setups made before starting with AWS CodeBuild

Step 1: Setting up a Web app

1. I set up my web app by launching an EC2 instance in the AWS Console, and I connected to it via SSH using Git Bash



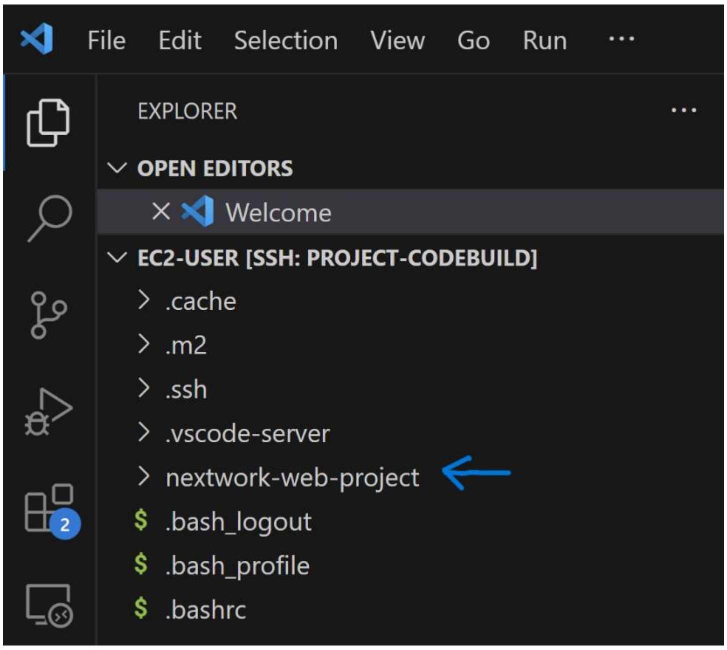
```
vpraj@DESKTOP-TCNBJQM MINGW64 ~
$ cd "C:\Users\vpraj\OneDrive\Desktop\AWS\Projects\CodeBuild"

vpraj@DESKTOP-TCNBJQM MINGW64 ~/OneDrive/Desktop/AWS/Projects/CodeBuild
$ pwd
/c/Users/vpraj/OneDrive/Desktop/AWS/Projects/CodeBuild

vpraj@DESKTOP-TCNBJQM MINGW64 ~/OneDrive/Desktop/AWS/Projects/CodeBuild
$ chmod 400 "CodeBuild.pem"

vpraj@DESKTOP-TCNBJQM MINGW64 ~/OneDrive/Desktop/AWS/Projects/CodeBuild
$ ssh -i "CodeBuild.pem" ec2-user@ec2-3-27-153-235.ap-southeast-2.compute.amazonaws.com
The authenticity of host 'ec2-3-27-153-235.ap-southeast-2.compute.amazonaws.com (3.27.153.235)' can't be established.
ED25519 key fingerprint is SHA256:BF1mXV0ClfcVCayE1sHoZEY0ygFtgMr1gIcfuarWgiA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-27-153-235.ap-southeast-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-3-252 ~]$
```

2. After connecting to my EC2 instance, I set up the development environment by installing Apache Maven and Amazon Corretto 8 (Java version)

3. Next, I downloaded Visual Studio Code as an alternative to Cloud9. I connected to the EC2 instance via SSH and set up the Java web application by running the command:

   **mvn archetype:generate **
   **-DgroupId=com.nextwork.app **
   **-DartifactId=nextwork-web-project **
   **-DarchetypeArtifactId=maven-archetype-webapp **
   -DinteractiveMode=false

4. After running the above command, a folder named 'nextwork-web-project' was created in my IDE's (VS Code) file explorer for my project.



Step 2: Creating a Git repository with AWS CodeCommit

1. I created a Git repository in AWS CodeCommit (main) and had to create a local repository in Visual Studio Code to make changes to the files and push them to the main repository. To do this, I installed Git in VSCode by running the command:

```
sudo yum install git
```

2. Next, I created a Git identity using the following commands:

```
git config --global user.name "Prajit"
git config --global user.email "v.prajit98@gmail.com"
```

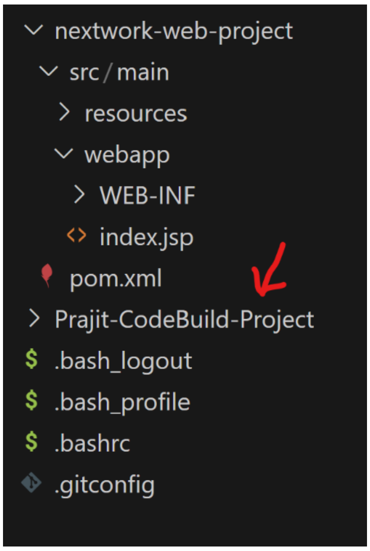
3. To create a local repository, I created a project directory in Visual Studio Code by cloning the CodeCommit repository (main). To clone the CodeCommit repository, I used the command:

   git clone https://git-codecommit.ap-southeast-2.amazonaws.com/v1/repos/Prajit-CodeBuild-Project

4. After running the above command, Git prompted for authentication to enter the username and password, as I was using Git over HTTPS. Therefore, I downloaded Git HTTPS credentials from IAM in the AWS Console.

5. Next, I navigated to the project directory folder to initialize my local repository. To move to the directory, I used the command:

   cd ~/nextwork-web-project, and I initialized the local repository by running the command: **git init -b main**



```
⌄ nextwork-web-project
  ⌄ src / main
    › resources
    ⌄ webapp
      › WEB-INF
      <> index.jsp
    pom.xml
› Prajit-CodeBuild-Project
$ .bash_logout
$ .bash_profile
$ .bashrc
  .gitconfig
```

6. Now, I set the remote origin to the main repository by copying the CodeCommit URL from the "Clone URL" option in the repositories section of the AWS Console. I used it in the command:

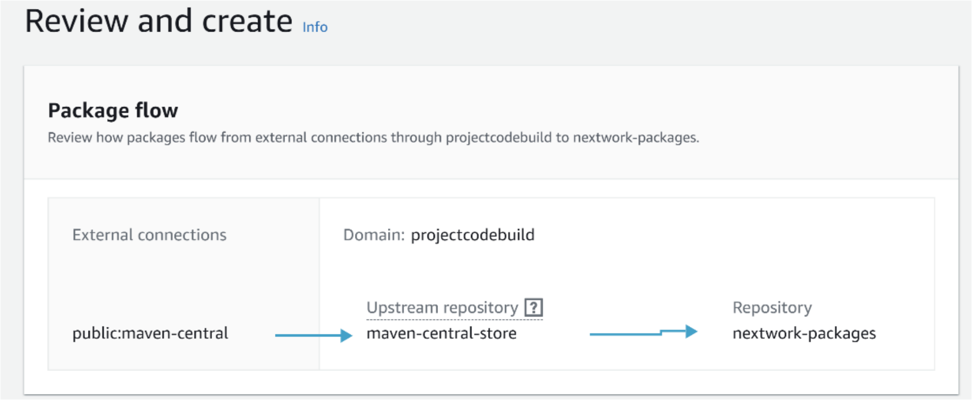
   git remote add origin https://git-codecommit.ap-southeast-2.amazonaws.com/v1/repos/Prajit-CodeBuild-Project

7.Next, I committed and pushed my code by running the following commands:

```
git add .
git commit -m "Initial commit. Updated index.jsp."
git push -u origin main
```

When I use the command **git push -u origin main**, Git prompts me for authentication again, requiring me to enter the username and password that I obtained from my HTTPS Git credentials.

Step 3: Set up a AWS CodeArtifact



Review and create Info

Package flow

Review how packages flow from external connections through projectcodebuild to nextwork-packages.

External connections

public:maven-central

Domain: projectcodebuild

Upstream repository ?

maven-central-store

Repository

nextwork-packages

Connecting my project with CodeArtifact

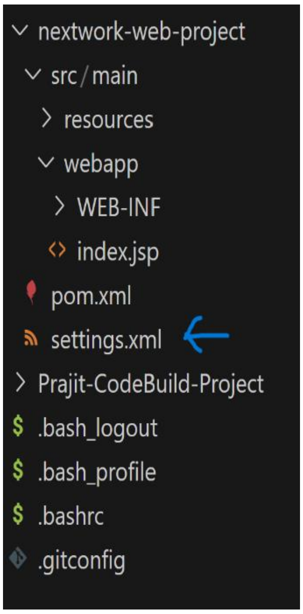
I connected my web app project (via my VS Code IDE) to CodeArtifact, so that CodeArtifact knows to which project it will store the packages or dependencies.
To connect my web app project to CodeArtifact via Visual Studio Code, I had to use a few commands. The first command is:

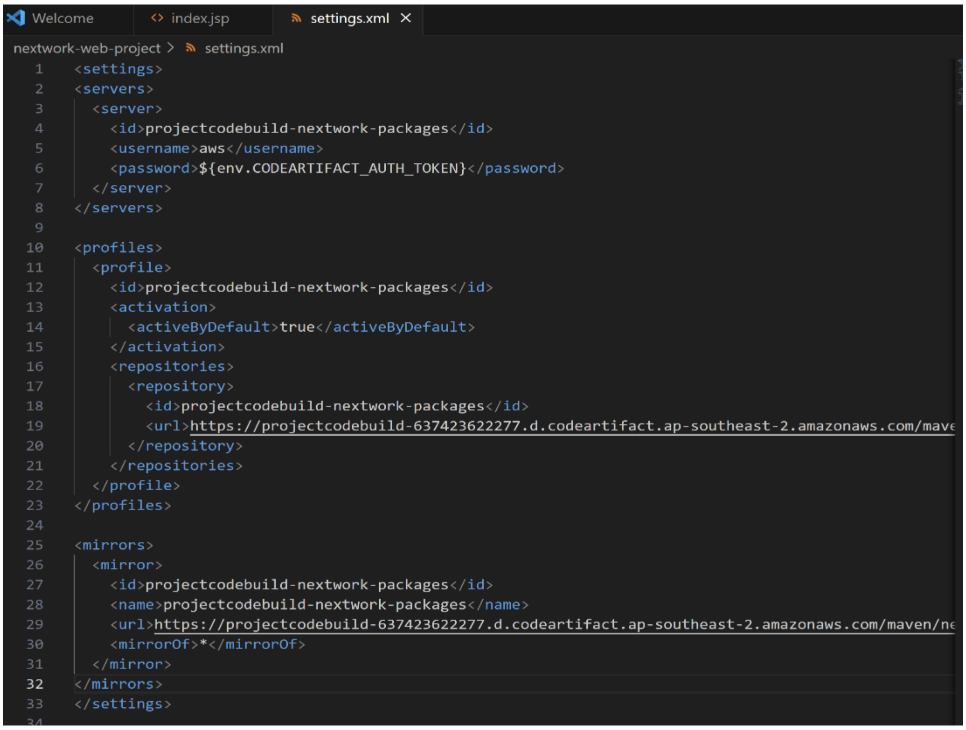
```
<servers>
<server>
<id>devops-3-nextwor-packages</id>
<username>aws</username>
<password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
</server>
</servers>
```

This command stores the access details for the repositories that my web app project is connecting to, so I needed to configure AWS in Visual Studio Code. I ran the command aws configure, which authenticates the AWS Access Key ID, Secret Access Key, default region name, and default output format. In this case, I had to create an AWS access key, which provides the Access Key ID and Secret Access Key.

I created a new file, **settings.xml**, in my web app. The settings.xml file provides Maven with instructions on where to find the dependencies it needs to fetch and how Maven will gain access to the repositories storing these dependencies.

Creating the settings.xml file involves writing three snippets of code in it, as shown in the picture below. To create the settings.xml file, I used the command: **cd ~/nextwork-web-project** to navigate to that directory, and then ran **echo $'<settings>\n</settings>' > settings.xml** to create the file. Once the file created, it will start appearing in the "**nextwork-web-project**" folder in the vscode explorer.





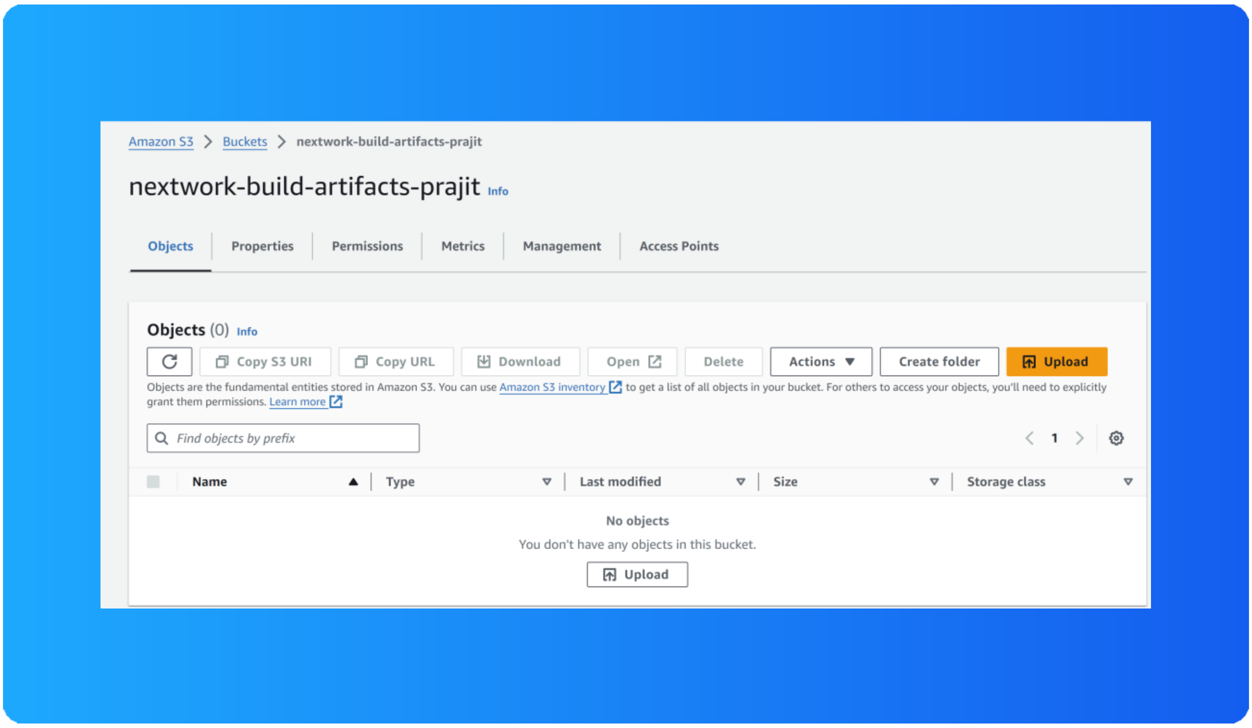
```xml
<settings>
<servers>
  <server>
    <id>projectcodebuild-nextwork-packages</id>
    <username>aws</username>
    <password>${env.CODEARTIFACT_AUTH_TOKEN}</password>
  </server>
</servers>

<profiles>
  <profile>
    <id>projectcodebuild-nextwork-packages</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <repositories>
      <repository>
        <id>projectcodebuild-nextwork-packages</id>
        <url>https://projectcodebuild-637423622277.d.codeartifact.ap-southeast-2.amazonaws.com/mave
      </repository>
    </repositories>
  </profile>
</profiles>

<mirrors>
  <mirror>
    <id>projectcodebuild-nextwork-packages</id>
    <name>projectcodebuild-nextwork-packages</name>
    <url>https://projectcodebuild-637423622277.d.codeartifact.ap-southeast-2.amazonaws.com/maven/ne
    <mirrorOf>*</mirrorOf>
  </mirror>
</mirrors>
</settings>
```

Set up an S3 bucket

I started my project with CodeBuild by creating an S3 bucket, as this bucket will later store an important build artifact generated during the build process I am setting up with CodeBuild.

The key artifact that this S3 bucket will capture is called WAR file (Web Application Resource).

This artifact is important because it ensures that any server hosting my web app will have all the resources and tools needed to successfully run my application.



Amazon S3 > Buckets > nextwork-build-artifacts-prajit

nextwork-build-artifacts-prajit

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

No objects

You don't have any objects in this bucket.

Set up a CodeBuild project

Source

Source means the location of my project's code that CodeBuild will fetch, compile, and package into a WAR file. The source can be a version control repository, such as AWS CodeCommit, where my project's code is maintained and managed.

Environment

My CodeBuild project's Environment means a set of software, tools, settings, and configurations that my application needs to function properly. I selected EC2 instance running Amazon linux 2 and Amazon corretto 8 as the run time.

Artifacts

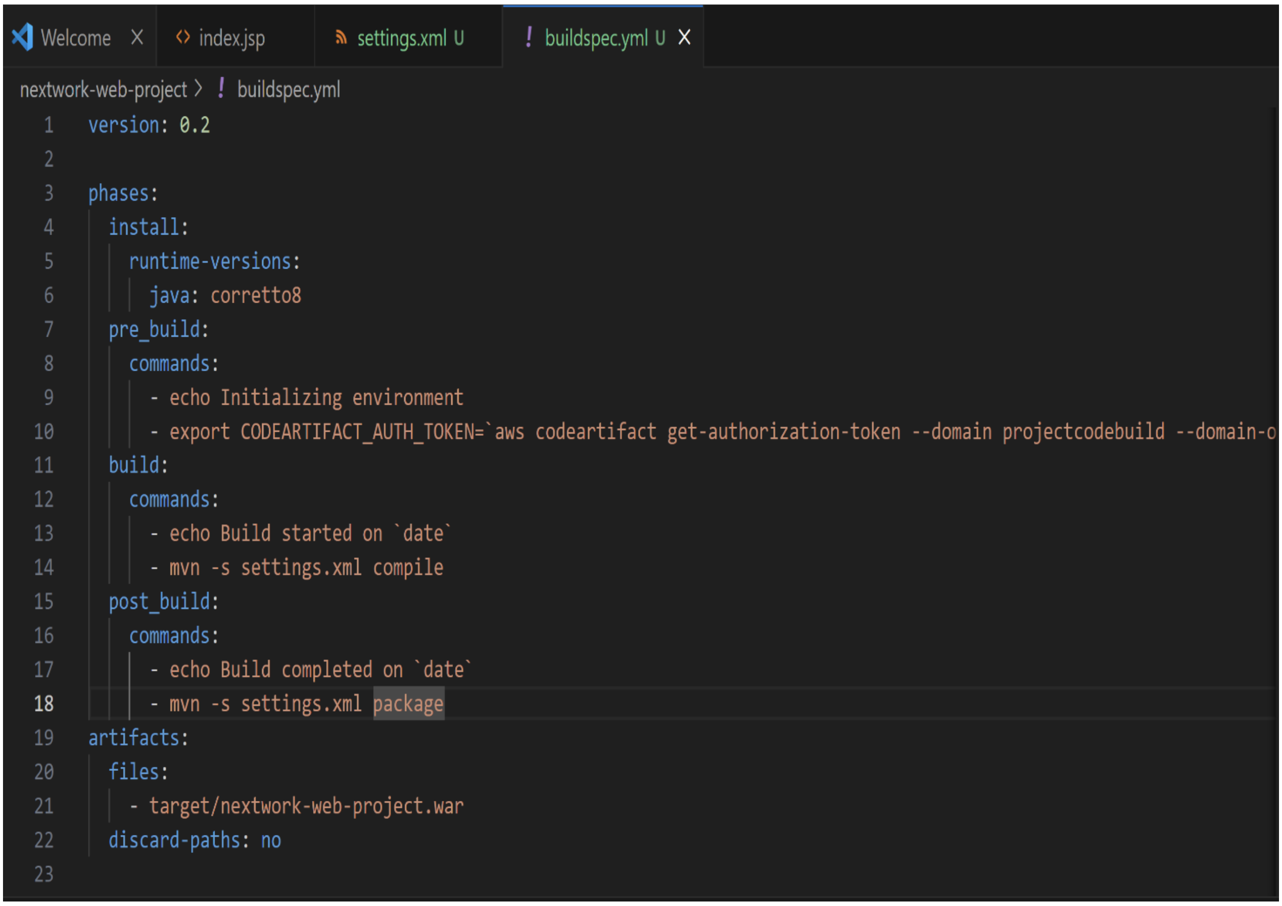
My CodeBuild project's Artifacts are files that CodeBuild will produce while building my project. This helps in packaging all the artifacts it creates into a WAR file, and I selected S3 bucket to store the WAR file.

Logs

My CodeBuild project's log configuration ensures that everything happening during the build process is recorded. I enabled CloudWatch Logs to capture these records, which helps us diagnose and resolve any issues that arise.

Create a buildspec.yml file

I created a buildspec.yml file in my project at the root of my code repository to allow CodeBuild to execute commands line by line for building my web app. This ensures that every build follows the same steps and produces consistent results.



```yaml
version: 0.2

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

```

Create a CodeBuild build project

My buildspec.yml file has four stages

1. The first phases in my buildspec.yml file is **install that** tells CodeBuild to install Java runtime to build this project.
2. The second phases in my buildspec.yml file is **pre_build** that contains two commands to run before building your web app.
   - **echo Initializing environment**- tells the terminal to print out "Initializing environment" as a message in the terminal, which helps you know that the environment setup process is starting.
   - **export CODEARTIFACT_AUTH_TOKEN-** gets an authorization token from AWS CodeArtifact. Maven will later use this token to securely access your CodeArtifact repository during the build process.
3. The third phase in my buildspec.yml file is **build** that kicks off the build process with two commands.
   - **echo Build started on date** - It tells the terminal to print out "Build started on" followed by the current date and time, indicating the start of the build process.
   - **mvn -s settings.xml compile**- It tells Maven to compile the project using the settings in the settings.xml file to make it automated from this time.

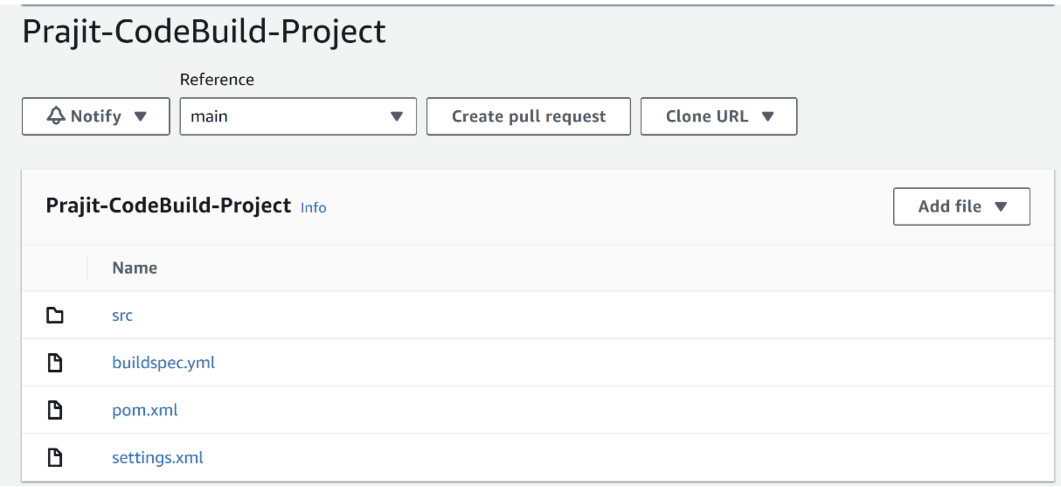
4. The fourth phase in my buildspec.yml file is **post-build** phase that runs another two commands.

    - **echo Build completed on date** - It tells the terminal to print out "Build completed on" followed by the current date and time, telling that the build process is now complete.

    - **mvn -s settings.xml package -** It tells Maven to package the compiled code into a WAR file using the settings in the settings.xml file.

After I saved the buildspec.yml file, I committed the changes and push it to CodeCommit by running the following command:

```
cd ~/nextwork-web-project
git add .
git commit -m "Adding buildspec.yml file"
git push -u origin main
```



Prajit-CodeBuild-Project

Reference: main

Prajit-CodeBuild-Project Info

Name
src
buildspec.yml
pom.xml
settings.xml

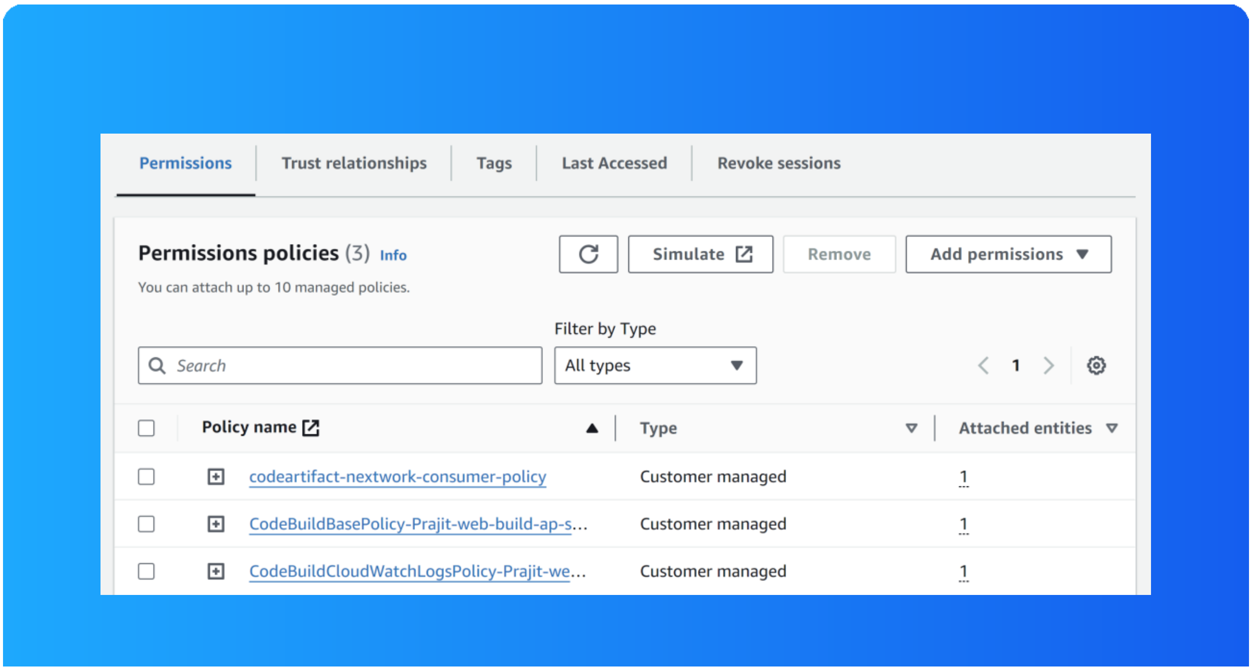
Modify CodeBuild’s IAM role

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

Before starting to build my web app project, I modified the service role for my CodeBuild project. This service role was initially created when I set up my CodeBuild project.

I attached a new policy called codeartifact-nextwork-consumer-policy to my CodeBuild project's IAM role.

Attaching this policy ensures that my CodeBuild project now has access to the packages and dependencies it will later compile.



Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

☐	Policy name ▲	Type	Attached entities
☐	codeartifact-nextwork-consumer-policy	Customer managed	1
☐	CodeBuildBasePolicy-Prajit-web-build-ap-s...	Customer managed	1
☐	CodeBuildCloudWatchLogsPolicy-Prajit-we...	Customer managed	1

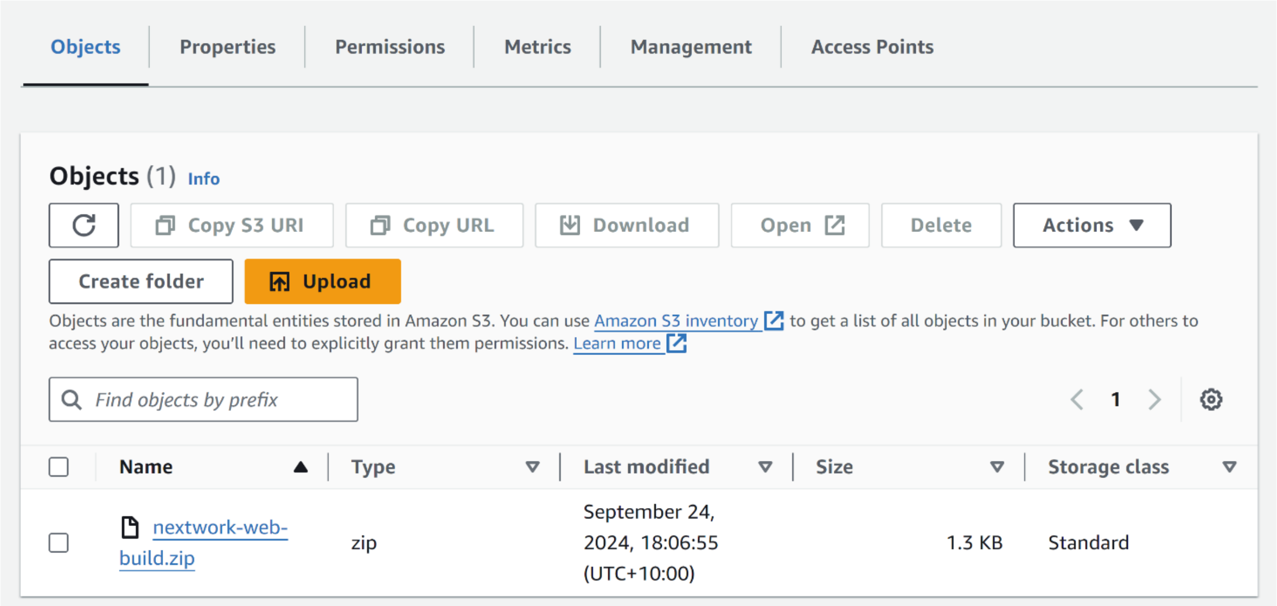
My first project build

To build my project, all I had to do was select the start build button in my CodeBuild console.

The build process in CodeBuild took about 5 minutes.

Once the build is complete, I checked my S3 bucket that created at the beginning of the project to store the build atrifacts from the build process.

I saw the WAR file (a bundled package of all the files and resources a server needs to run my web app) in Zip format, which verified that the build was completed successfully.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	nextwork-web-build.zip	zip	September 24, 2024, 18:06:55 (UTC+10:00)	1.3 KB	Standard

Everyone should be in a job they love.

Check out nextwork.org for more projects