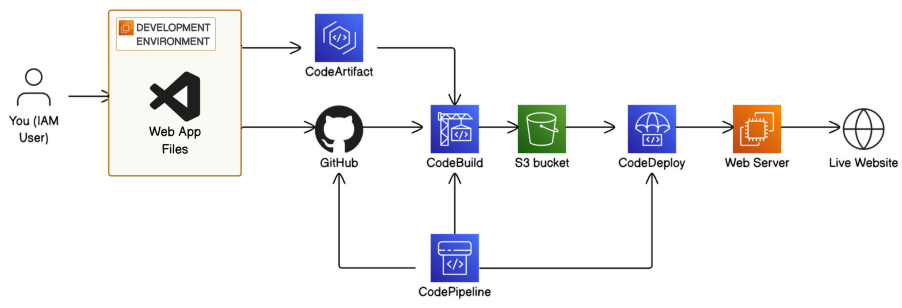
nextwork.org

I'm starting the 7-Day DevOps challenge!

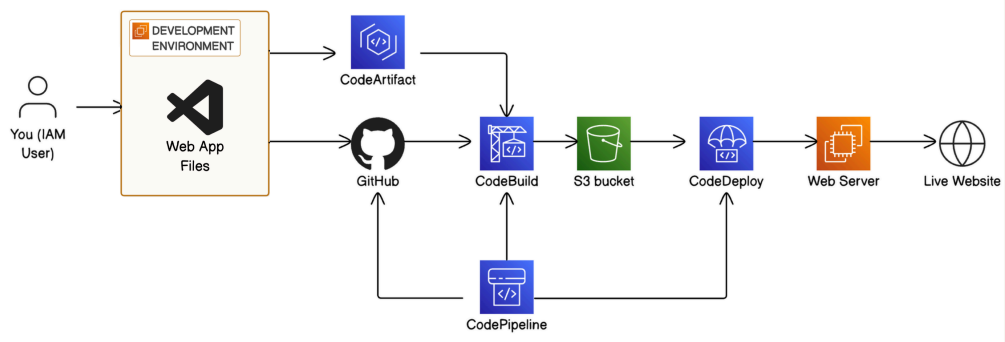
K Kyle Lowe



Kyle Lowe
NextWork Student

I'm building a CI/CD pipeline in 7 days

In this DevOps challenge, I'm learning about AWS and CI/CD. By the end of the 7 days, I will have completed my own CI/CD pipeline to deploy an app using AWS



K
Kyle Lowe
NextWork Student

Hold me accountable!

I will set aside 2 hours every day for a week to work on this challenge. I will keep myself accountable by making sure I work on the project every day. My reward for completing this DevOps challenge will be being able to show off my project

What is DevOps?

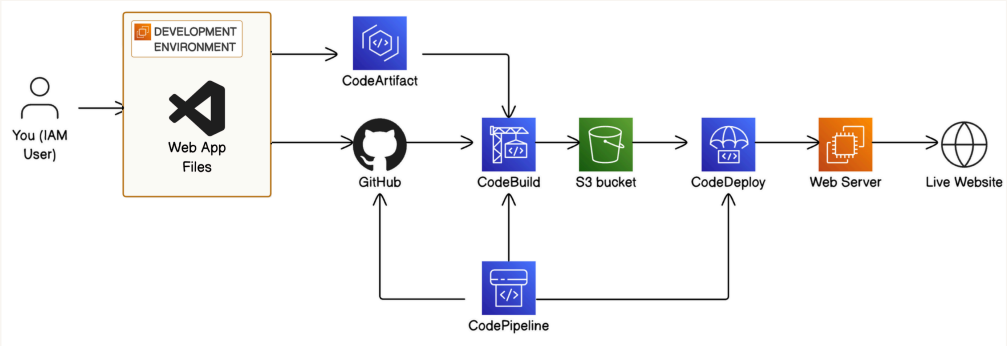
DevOps is a set of practices that combine development and operations to shorten development cycles and deliver efficient software. DevOps engineers implement automated pipelines, containerise applications, and turn cloud infrastucture into code.

What is CI/CD?

CI/CD stands for continuous integration and continuous delivery. This process helps teams by automatically building the updated code, and that deployment happens automatically.

Kyle Lowe
NextWork Student

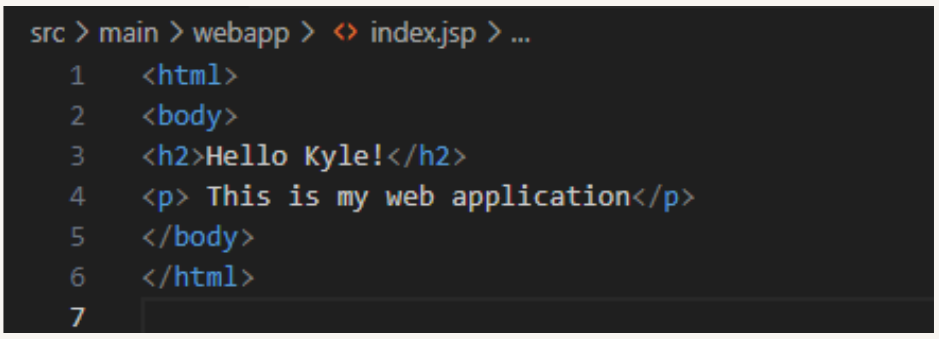
Excited to share my progress - do this challenge with me!



nextwork.org

Set Up a Web App Using AWS and VS Code

K Kyle Lowe



K

Kyle Lowe
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to set up the basics of an app using AWS. I'm doing this project to learn about AWS and how to set up an app.

Key tools and concepts

Services I used were EC2 Instances, VS Code and apache Maven. Key concepts I learnt include creating an EC2 Instance and connecting it to VS Code.

Project reflection

One thing I didn't expect in this project was how easy it was to set up everything

This project took me approximately 2 hours. The most challenging part was setting up maven correctly It was most rewarding to finish the project.

This project is part one of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project in 1 day.

Launching an EC2 instance

I started this project by launching an EC2 instance because it is the foundation for creating the web app to run on the cloud.

I also enabled SSH

SSH is a protocol used to authorize users to use remote servers. I enabled SSH so that there is a secure connection between my machine and the EC2 instance.

Key pairs

A key pair is a set of keys which consists of a private and public key. A private key is that one that verifies that you're allowed to acess the virtural machine and the public key is the key that AWS keeps.

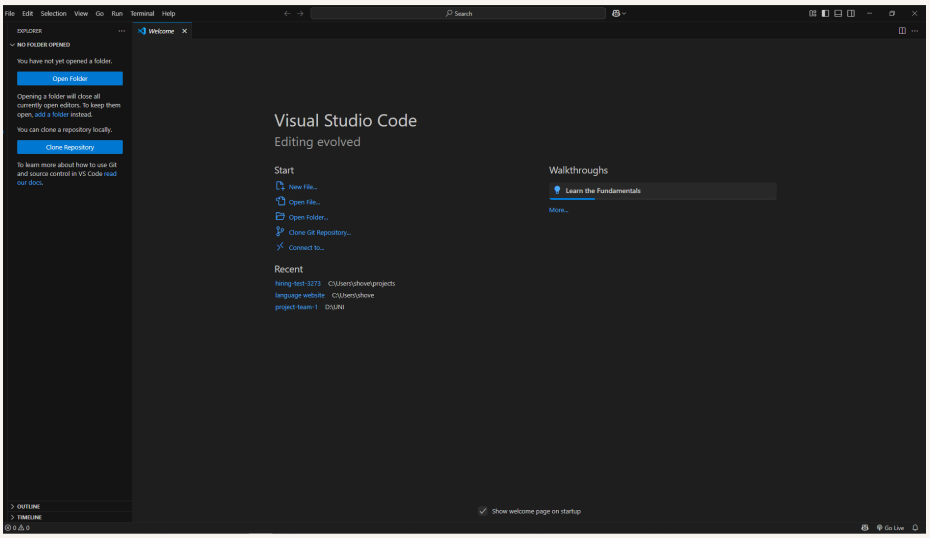
Once I set up my key pair, AWS automatically downloaded the private key to my machine.

Kyle Lowe
NextWork Student

Set up VS Code

VS Code is an IDE that is used to code.

I installed VS Code to help connect to AWS and to code

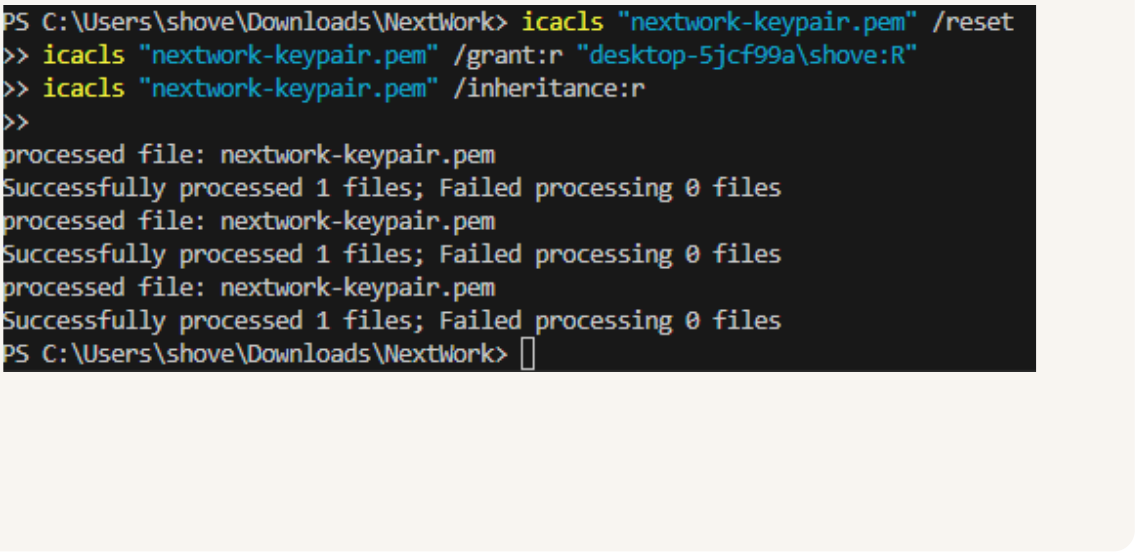


Kyle Lowe
NextWork Student

My first terminal commands

A terminal is a way to send instructions to the computer. The first command I ran for this project is to go to the directory of where the .pem file is.

I also updated my private key's permissions by removing the default permission settings on the file, giving the current user read acess to the secret key and to make sure changes in ht epermsiions of the other files in the floder doesn't change



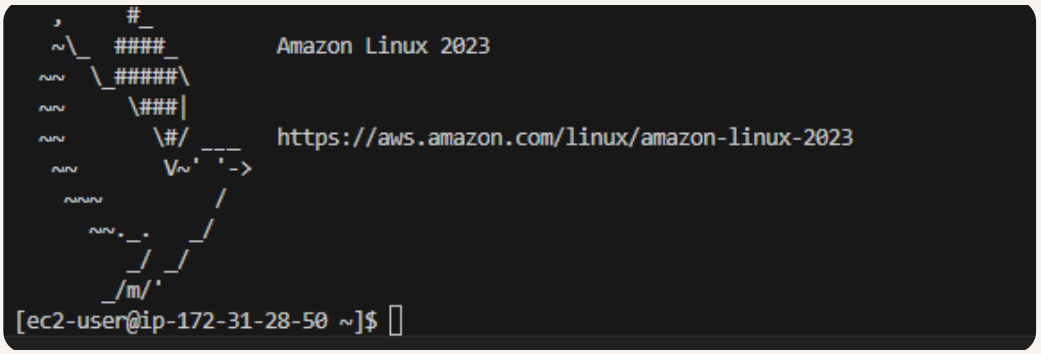
Kyle Lowe
NextWork Student

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command ssh -i C:\Users\shove\Downloads\NextWork\nextwork-keypair.pem ec2-user@3.27.43.243

This command required an IPv4 address

A server's IPV4 DNS is a public address for the EC2 server that the internet uses to find and connect to it.



Maven & Java

Apache Maven is a tool that automates the building of software.

Maven is required in this project to help compile, link, package and test code for the app.

Java is a popular programming language used to build different types of applications.

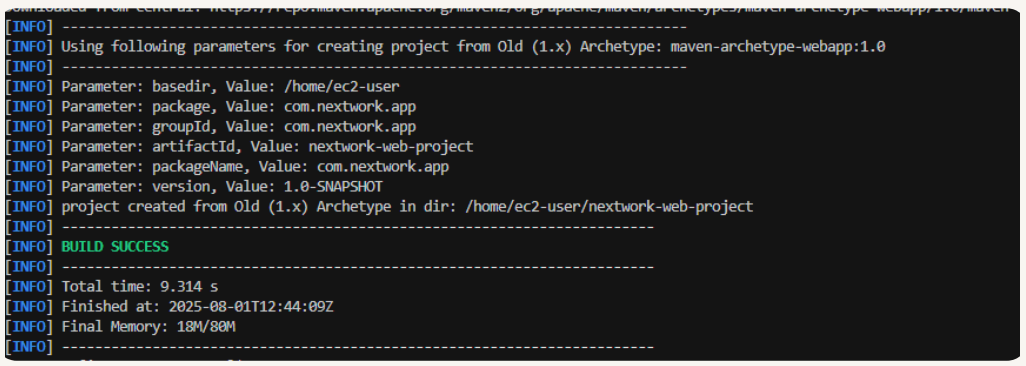
Java is required in this project because it is the main language that I was taught in university with.

Create the Application

I generated a Java web app using the command mvn archetype:generate \ -DgroupId=com.nextwork.app \ -DartifactId=nextwork-web-project \ -DarchetypeArtifactId=maven-archetype-webapp \ -DinteractiveMode=false

I installed Remote - SSH, which is an extension on vs code I installed it so the ide can access the files stored on the instance

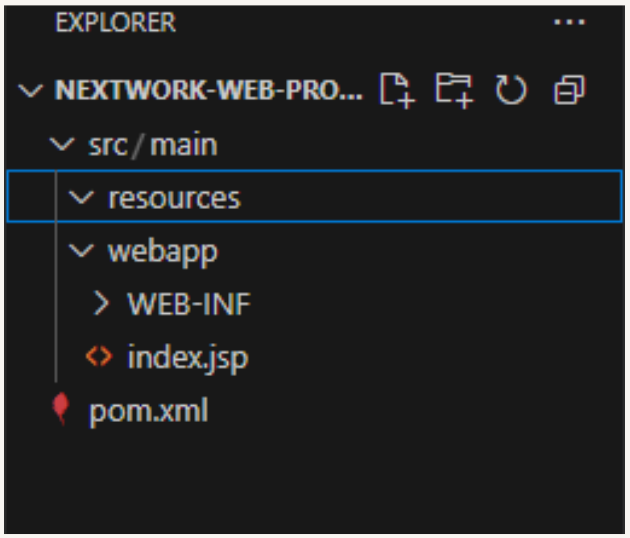
Configuration details required to set up a remote connection include ddetails fo the EC2 instnace.



Create the Application

Using VS Code's file explorer, I could see the file sturcture for the web application

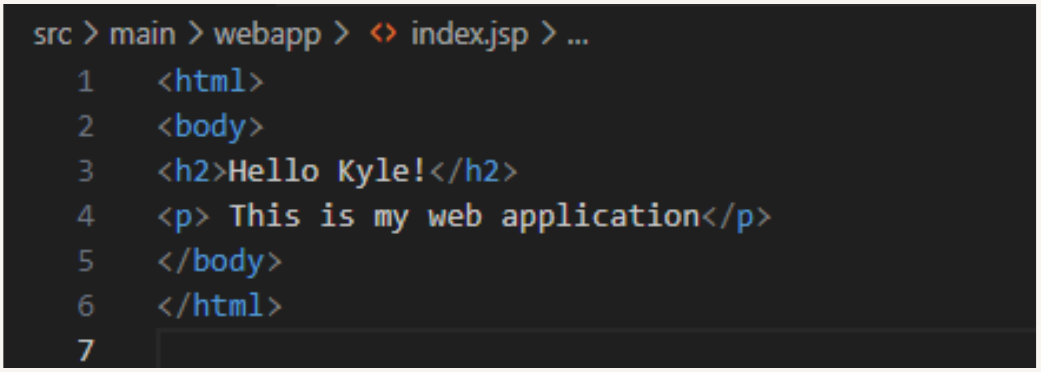
Two of the project folders created by Maven are src and webapp, which contain the web application.



Using Remote - SSH

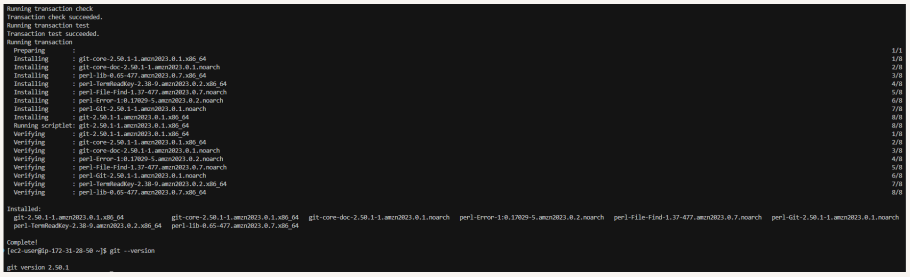
The index.jsp is the web application page

I edited index.jsp by editing the file in vscode and saving it.



Connect a GitHub Repo with AWS

Kyle Lowe



K
Kyle Lowe
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate connecting a GitHub repo to AWS. I'm doing this project to learn how to set up Github and connect it to an AWS project.

Key tools and concepts

Services I used were GitHub and VS Code. Key concepts I learnt include how to connect GitHub to the EC2 instance.

Project reflection

This project took me approximately 90 minutes. The most challenging part was setting up the GitHub repo. It was most rewarding to connect GitHub and see the changes that I made.

I did this project because it is part of the 7 days devOps challenge.

This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow.

Kyle Lowe
NextWork Student

Git and GitHub

Git is a version control system. I installed Git using the commands sudo dnf update -y, sudo dnf install git -y.

GitHub is a storage place for different versions of a project. I'm using GitHub in this project to store different versions of my project.

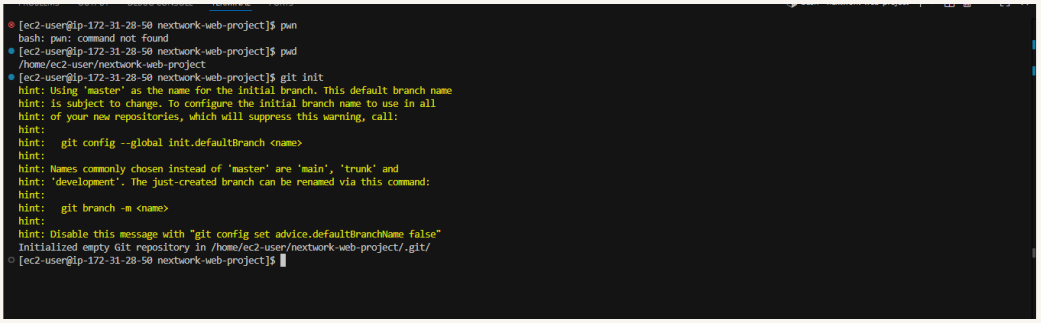
Kyle Lowe
NextWork Student

My local repository

A Git repository is a place to store my project with the entire history of the project.

Git init is a command that initializes the local git repository. I ran git init in the terminal in the cloud.

A branch in Git is a parallel version of the same project. After running git init, the response from the terminal was giving me suggestions for naming the main branch master.



K
Kyle Lowe
NextWork Student

To push local changes to GitHub, I ran three commands

git add

The first command I ran was "git add ." A staging area is a place where all the modified files will be.

git commit

The second command I ran was git commit. Using '-m' means' that I will add a message to the git commit.

git push

The third command I ran was git push -u origin master. Using '-u' means that I am setting an upstream for my local branch.

Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to authenticate that I have the right permission to modify the GitHub repo.

Local Git identity

Git needs my name and email because it needs to track who made what change.

Running git log showed me that the history of the commits.

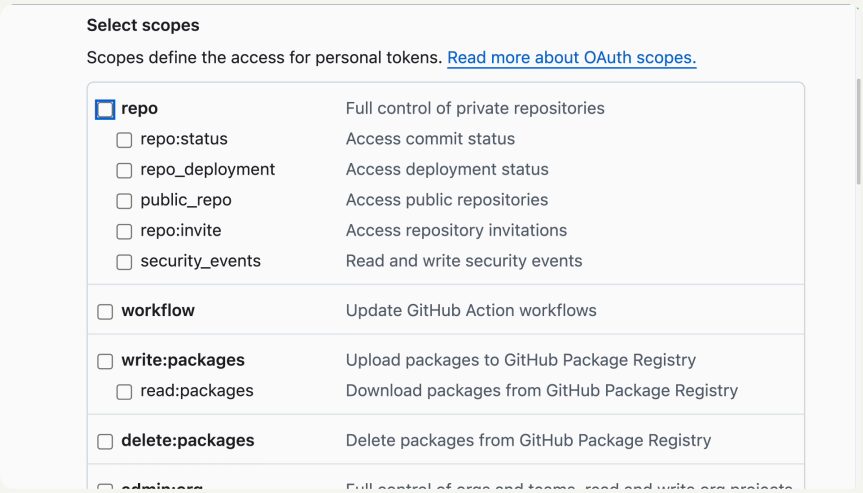
```
To https://github.com/kyleLowe/nextwork-web-project-devOps.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

GitHub tokens

GitHub authentication failed when I entered my password because it doesn't use a password but instead uses GitHub tokens.

A GitHub token is a set of random numbers and letters. I'm using one in this project because it will be used to authenticate the Git commits.

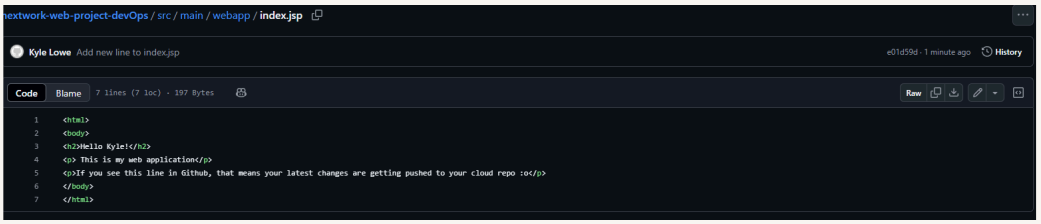
I could set up a GitHub token by generating a token from my GitHub settings.


Select scopes
Scopes define the access for personal tokens. Read more about OAuth scopes.
repo
Full control of private repositories
repo:status
Access commit status
repo_deployment
Access deployment status
public_repo
Access public repositories
repo:invite
Access repository invitations
security_events
Read and write security events
workflow
Update GitHub Action workflows
write:packages
Upload packages to GitHub Package Registry
read:packages
Download packages from GitHub Package Registry
delete:packages
Delete packages from GitHub Package Registry

Making changes again

I wanted to see Git working in action, so I went to the repo on GitHub I couldn't see the changes in my GitHub repo initially because I didn't update the Github and push my progress in the local repo.

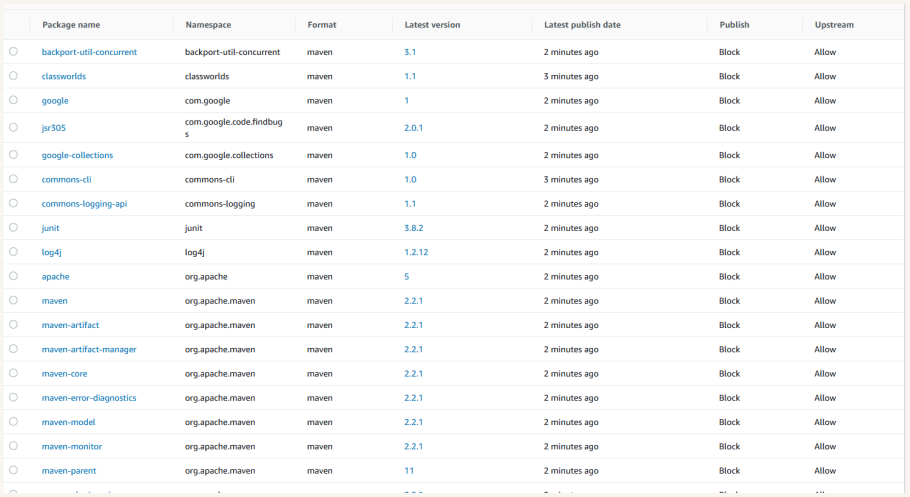
I finally saw the changes in my GitHub repo after pushing my changes in the local repo.



nextwork.org

Secure Packages with CodeArtifact

Kyle Lowe



	Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
○	backport-util-concurrent	backport-util-concurrent	maven	3.1	2 minutes ago	Block	Allow
○	classworlds	classworlds	maven	1.1	3 minutes ago	Block	Allow
○	google	com.google	maven	1	2 minutes ago	Block	Allow
○	jsr305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	Block	Allow
○	google-collections	com.google.collections	maven	1.0	2 minutes ago	Block	Allow
○	commons-cli	commons-cli	maven	1.0	3 minutes ago	Block	Allow
○	commons-logging-api	commons-logging	maven	1.1	2 minutes ago	Block	Allow
○	junit	junit	maven	3.8.2	2 minutes ago	Block	Allow
○	log4j	log4j	maven	1.2.12	2 minutes ago	Block	Allow
○	apache	org.apache	maven	5	2 minutes ago	Block	Allow
○	maven	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-artifact	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-artifact-manager	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-core	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-error-diagnostics	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-model	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-monitor	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
○	maven-parent	org.apache.maven	maven	11	2 minutes ago	Block	Allow

K

Kyle Lowe
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate creating an artifact repository I'm doing this project to learn to learn about AWS CodeArtifact to store packages.

Key tools and concepts

Services I used were AWS CodeArtifact, AWS EC2 instances and IAM policy. Key concepts I learnt include what code artifact is, how to implement IAM policy and role and connect it to an EC2 instance.

Project reflection

This project took me approximately 90 minutes The most challenging part was setting up CodeArtifact and having the correct permissions. It was most rewarding to see all the packages generated for CodeArtifact.

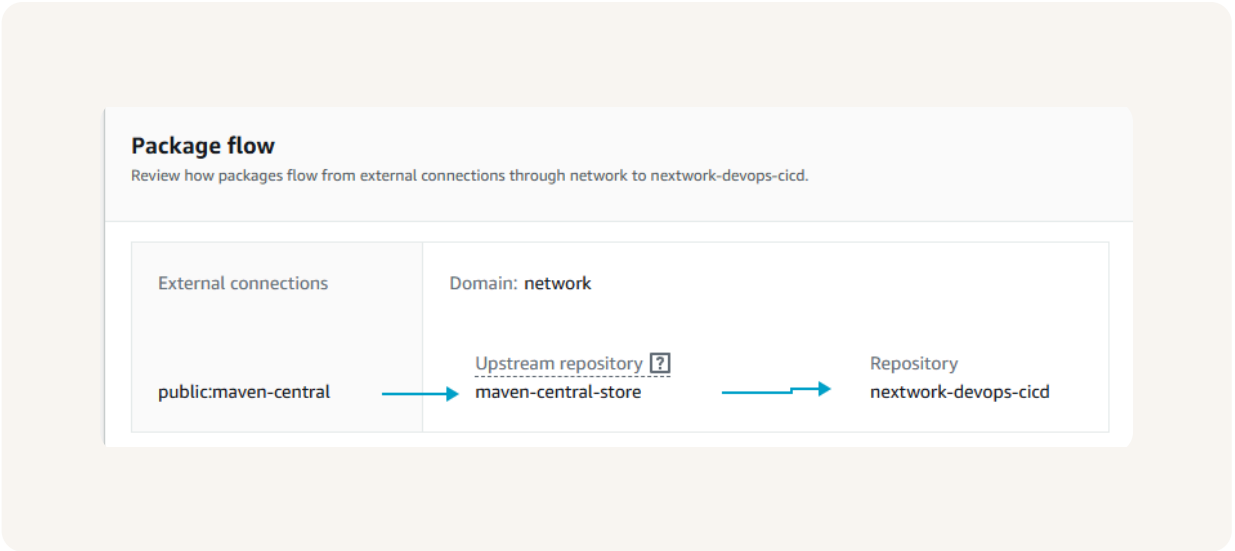
This project is part three of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow.

CodeArtifact Repository

CodeArtifact is a place to store all the software packages securely. Engineering teams use artifact repositories because it is reliable, has great security and is in control of the version of packages.

A domain is a folder that contains multiple repositories for the same project. The domain ensures consistent security across all package repositories.

A CodeArtifact repository can have an upstream repository, which means that they are like a backup library for when the primary repository doesn't have the packages it needs. My repository's upstream repository is Maven Central.



K
Kyle Lowe
NextWork Student

CodeArtifact Security

Issue

To access CodeArtifact, we need to generate a CodeArtifact authorisation token. I ran into an error when retrieving a token because I don't have the correct permissions to generate it.

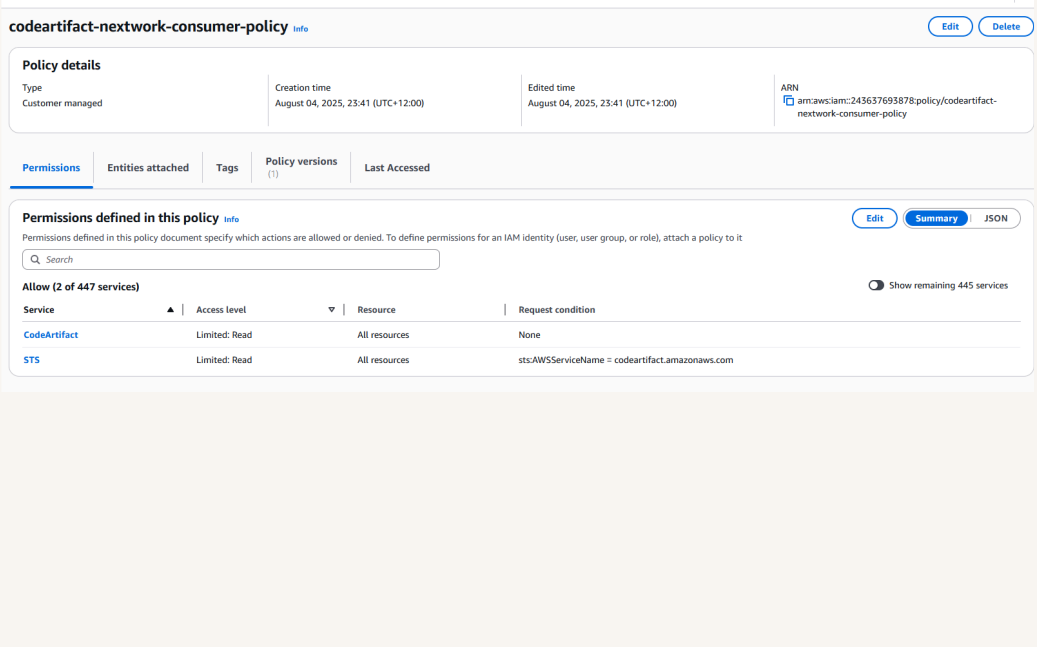
Resolution

To resolve the error with my security token, I assigned an IAM role to the EC2 instance. This resolved the error because it gave permission for the EC2 instance to access CodeArtifact

It's security best practice to use IAM roles because it limits the what access accounts and services have with each other improving the security of it.

The JSON policy attached to my role

The JSON policy I set up grants permission to get authentication tokens, find repository locations, read packages from repositories and allows temporary elevated access specifically for CodeArtifact locations.



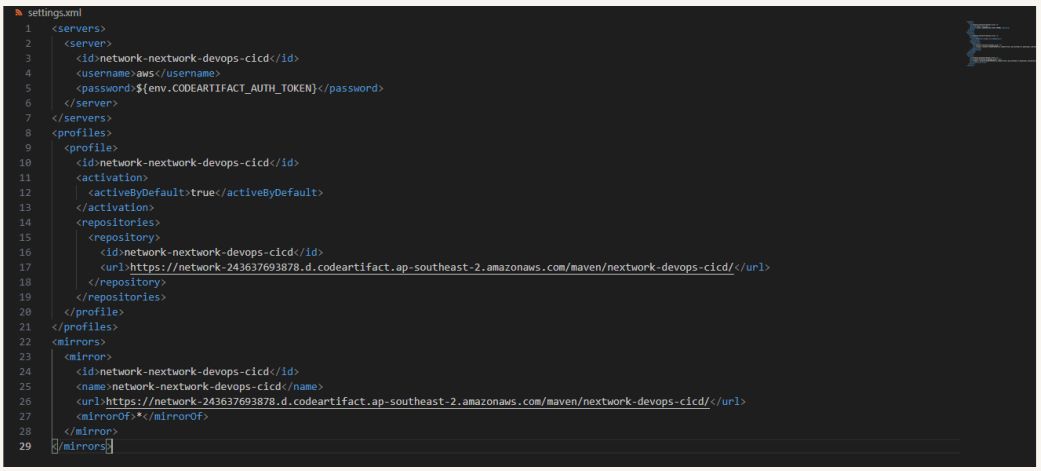
Kyle Lowe
NextWork Student

Maven and CodeArtifact

To test the connection between Maven and CodeArtifact, I compiled my web app using settings.xml

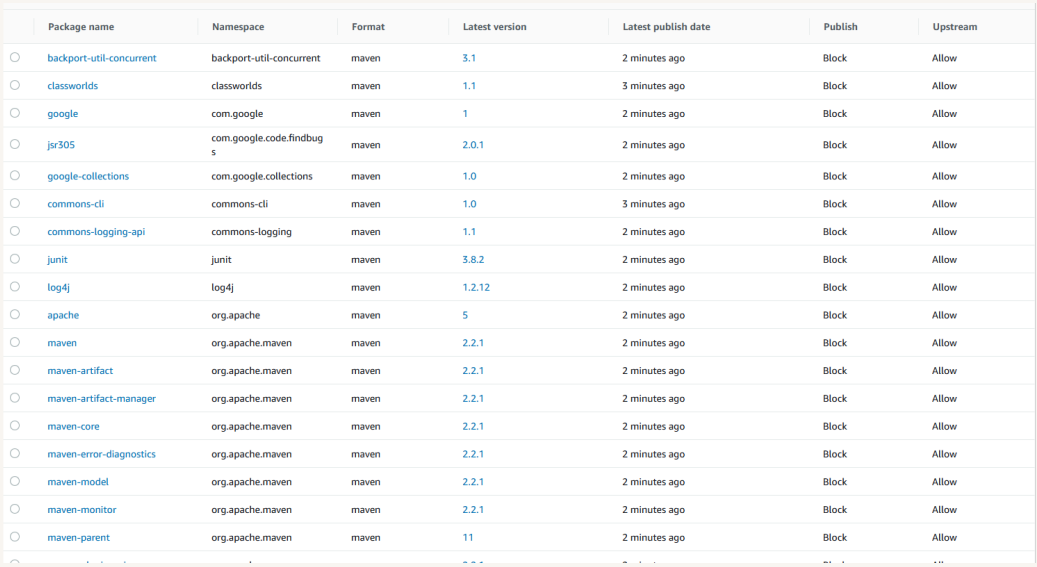
The settings.xml file configures Maven to tell it to look at the code artifact repository first before downloading any dependencies.

Compiling means looking at the project's dependencies in the pom.xml file. Then, instead of downloading directly from public repositories, it checks the CodeArtifact repository. If it is not in the CodeArtifact, it will then Maven Central.



Verify Connection

After compiling, I checked code artifiact I noticed that all the packages were generated now.

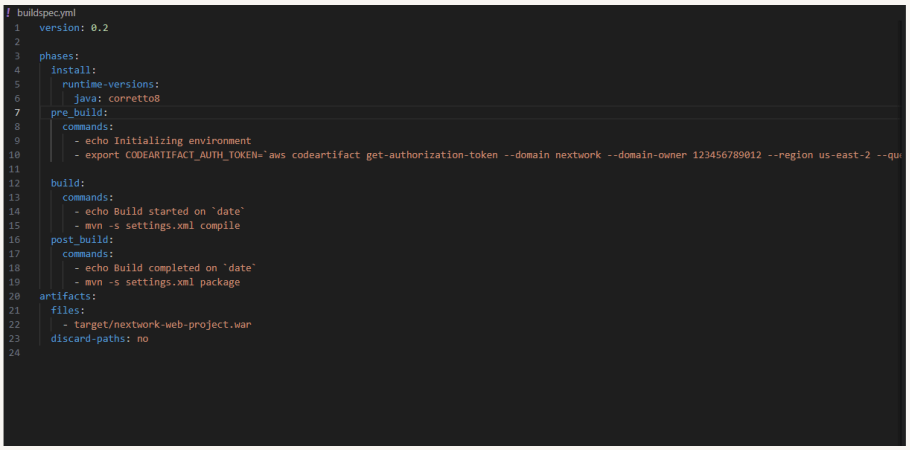


Package name	Namespace	Format	Latest version	Latest publish date	Publish	Upstream
backport-util-concurrent	backport-util-concurrent	maven	3.1	2 minutes ago	Block	Allow
classworlds	classworlds	maven	1.1	3 minutes ago	Block	Allow
google	com.google	maven	1	2 minutes ago	Block	Allow
jsr305	com.google.code.findbugs	maven	2.0.1	2 minutes ago	Block	Allow
google-collections	com.google.collections	maven	1.0	2 minutes ago	Block	Allow
commons-cli	commons-cli	maven	1.0	3 minutes ago	Block	Allow
commons-logging-api	commons-logging	maven	1.1	2 minutes ago	Block	Allow
junit	junit	maven	3.8.2	2 minutes ago	Block	Allow
log4j	log4j	maven	1.2.12	2 minutes ago	Block	Allow
apache	org.apache	maven	5	2 minutes ago	Block	Allow
maven	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-artifact	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-artifact-manager	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-core	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-error-diagnostics	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-model	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-monitor	org.apache.maven	maven	2.2.1	2 minutes ago	Block	Allow
maven-parent	org.apache.maven	maven	11	2 minutes ago	Block	Allow

nextwork.org

Continuous Integration with CodeBuild

K Kyle Lowe



K

Kyle Lowe
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to implement a CodeBuild project. I'm doing this project to learn about AWS CodeBuild.

Key tools and concepts

Services I used were AWS CodeBuild, CodeArtifact and GitHub. Key concepts I learnt include how to create a CodeBuild Project and troubleshooting when building the project fails.

Project reflection

This project took me approximately 150 minutes. The most challenging part was fixing the build errors. It was most rewarding to finally solve the errors and build correctly.

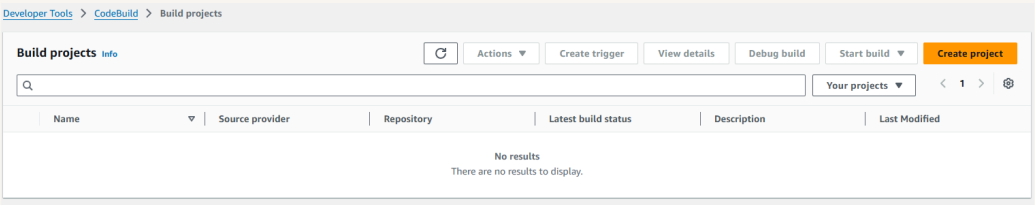
This project is part four of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project later today.

Kyle Lowe
NextWork Student

Setting up a CodeBuild Project

CodeBuild is a continuous integration service, which means it is like a quality control checkpoint that automatically activates when anyone on the team makes a change. Engineering teams use it because it is super efficient.

My CodeBuild project's source configuration means where the project is stored and I selected Github.

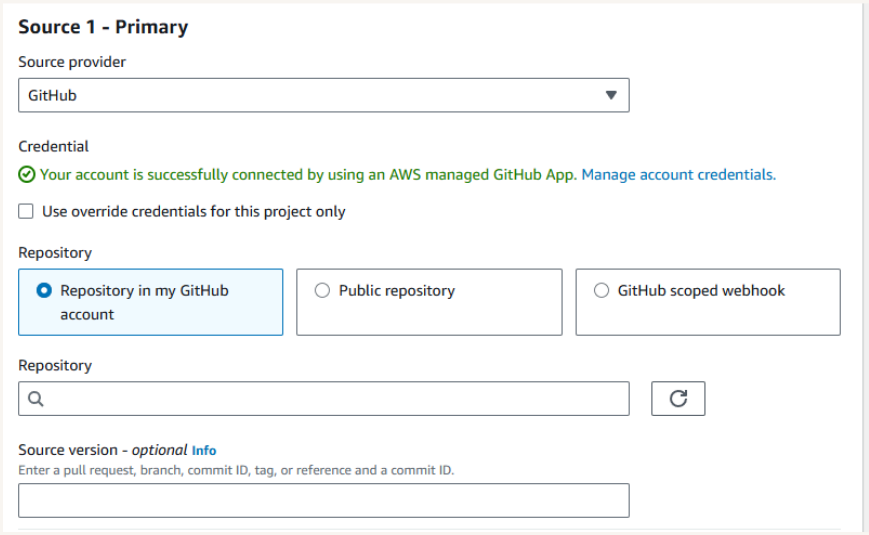


K
Kyle Lowe
NextWork Student

Connecting CodeBuild with GitHub

There are multiple credential types for GitHub, like GitHub App, personal access token and OAuth app. I used GitHub App because it is the simplest and most secure option.

The service that helped connect to GitHub is AWS CodeConnections.



Kyle Lowe
NextWork Student

CodeBuild Configurations

Environment

My CodeBuild project's Environment configuration means the settings that will be include in the build. It includes settings like provisioning model OS Amazon Linux, standard runtime, Corretto 8 image and a new service role.

Artifacts

Build artifacts are the outputs of the build process. They're important because it is what is going to be deployed to the servers. My build process will create a WAR file. To store them, I created a project in CodeBuild.

Packaging

When setting up CodeBuild, I also chose to package artifacts in a zip file because it makes the build a smaller size, organises the files, and makes it simple to share with others.

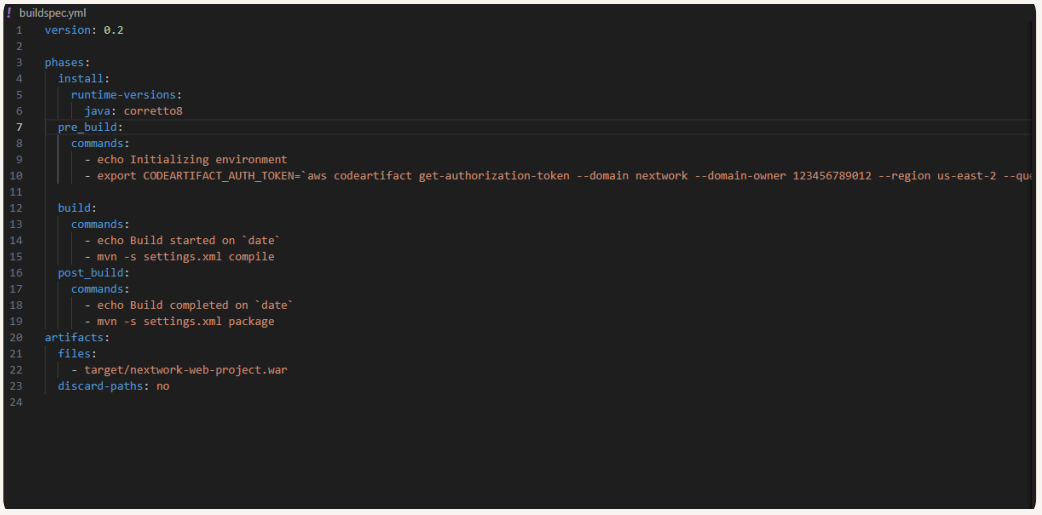
Monitoring

For monitoring, I enabled CloudWatch Logs, which is a service that tracks and collects everything that happens during the build process.

buildspec.yml

My first build failed because the buildspec.yml file doesn't exist. A buildspec.yml file is needed because it is a step-by-step instruction manual for how to build the project.

The first two phases in my buildspec.yml file are the pre-build and build commands. The third phase in my buildspec.yml file post build. The fourth phase in my buildspec.yml file tells codebuild to save the output.

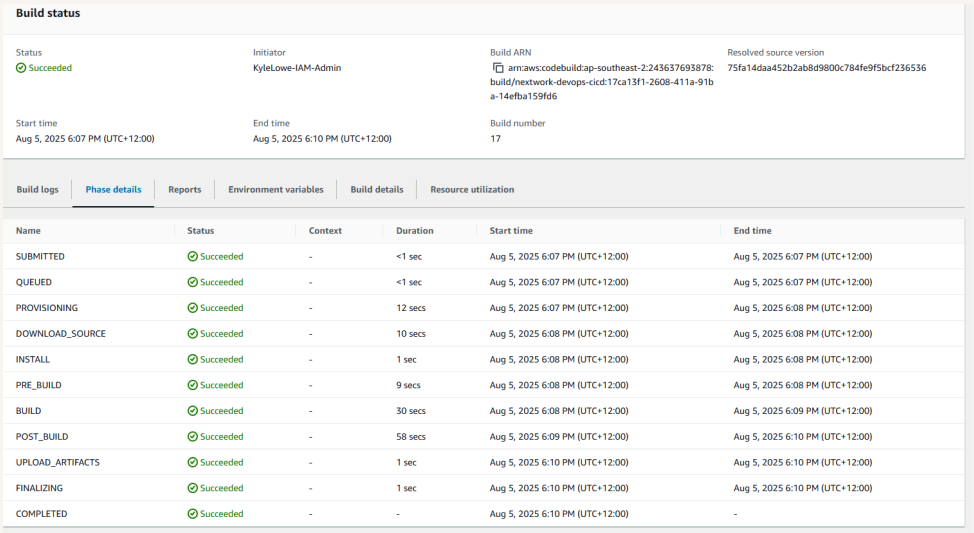


Success!

My second build also failed, but with a different error that said that it cannot access the settings.xml file To fix this I need to set permissions for CodeBuild to access CodeArtifact.

To resolve the second error, I checked the permissions for code build and in the settings file. When I built my project again, I saw that it built sucessfully.

To verify the build, I checked the S3 bucket. Seeing the artifact tells me that it was built sucessfully.



Build status

Status	Initiator	Build ARN	Resolved source version
Succeeded	KyleLowe-IAM-Admin	arn:aws:codebuild:ap-southeast-2:243637693878:build/nextwork-devops-cicd:17ca13f1-2608-411a-91ba-14efba159fd6	75fa14daa452b2ab8d9800c784fe9f5bcf236536
Start time	**End time**	**Build number**	
Aug 5, 2025 6:07 PM (UTC+12:00)	Aug 5, 2025 6:10 PM (UTC+12:00)	17	

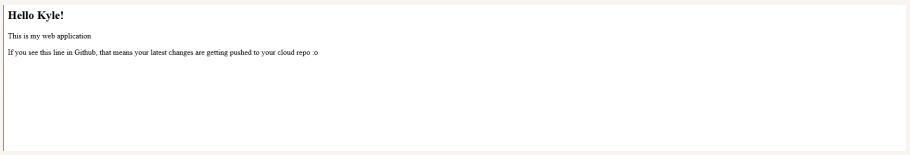
Build logs | Phase details | Reports | Environment variables | Build details | Resource utilization

Name	Status	Context	Duration	Start time	End time
SUBMITTED	Succeeded	-	<1 sec	Aug 5, 2025 6:07 PM (UTC+12:00)	Aug 5, 2025 6:07 PM (UTC+12:00)
QUEUED	Succeeded	-	<1 sec	Aug 5, 2025 6:07 PM (UTC+12:00)	Aug 5, 2025 6:07 PM (UTC+12:00)
PROVISIONING	Succeeded	-	12 secs	Aug 5, 2025 6:07 PM (UTC+12:00)	Aug 5, 2025 6:08 PM (UTC+12:00)
DOWNLOAD_SOURCE	Succeeded	-	10 secs	Aug 5, 2025 6:08 PM (UTC+12:00)	Aug 5, 2025 6:08 PM (UTC+12:00)
INSTALL	Succeeded	-	1 sec	Aug 5, 2025 6:08 PM (UTC+12:00)	Aug 5, 2025 6:08 PM (UTC+12:00)
PRE_BUILD	Succeeded	-	9 secs	Aug 5, 2025 6:08 PM (UTC+12:00)	Aug 5, 2025 6:08 PM (UTC+12:00)
BUILD	Succeeded	-	30 secs	Aug 5, 2025 6:08 PM (UTC+12:00)	Aug 5, 2025 6:09 PM (UTC+12:00)
POST_BUILD	Succeeded	-	58 secs	Aug 5, 2025 6:09 PM (UTC+12:00)	Aug 5, 2025 6:10 PM (UTC+12:00)
UPLOAD_ARTIFACTS	Succeeded	-	1 sec	Aug 5, 2025 6:10 PM (UTC+12:00)	Aug 5, 2025 6:10 PM (UTC+12:00)
FINALIZING	Succeeded	-	1 sec	Aug 5, 2025 6:10 PM (UTC+12:00)	Aug 5, 2025 6:10 PM (UTC+12:00)
COMPLETED	Succeeded	-	-	Aug 5, 2025 6:10 PM (UTC+12:00)	-

nextwork.org

Deploy a Web App with CodeDeploy

Kyle Lowe



K

Kyle Lowe
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to deploy my web application. I'm doing this project to learn more about AWS.

Key tools and concepts

Services I used were CodeDeploy, GitHub, CodeBuild, and CodeArtifact. Key concepts I learnt include how to set up CodeDeploy and deploy the web application.

Project reflection

This project took me approximately 60 minutes. The most challenging part was make sure that all the scripts were working. It was most rewarding to see my web application.

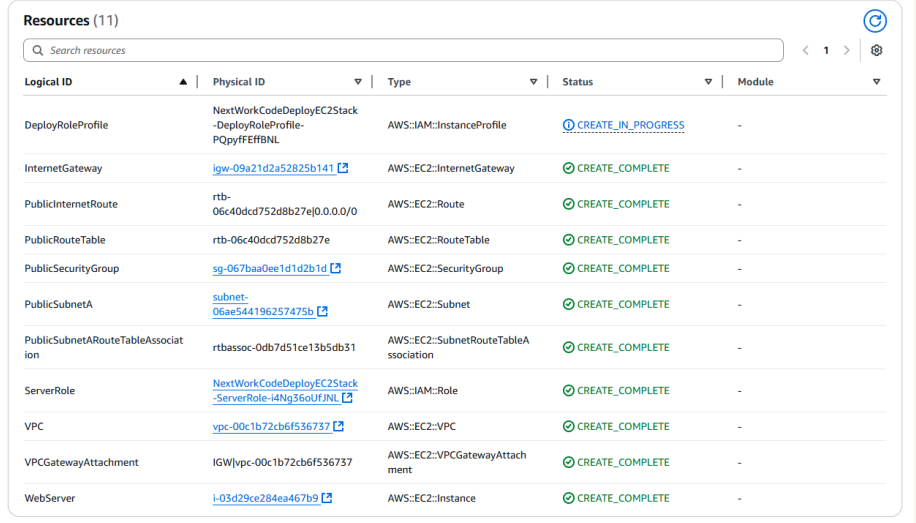
This project is part five of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow.

Deployment Environment

To set up CodeDeploy, I launched an EC2 instance and a VPC because it is an environment where I am able to deploy and run the web application.

Instead of launching these resources manually, I used CloudFormation. When I need to delete these resources I just need to delete the CloudFormation Stack.

Other resources created in this template include the EC2 instance being used and the storage location. They're also in the template because it is the information that the CloudFormation needs to set up the stack.



Resources (11)

Search resources

Logical ID	Physical ID	Type	Status	Module
DeployRoleProfile	NextWorkCodeDeployEC2Stack-DeployRoleProfile-PQpyfFEffBNL	AWS::IAM::InstanceProfile	CREATE_IN_PROGRESS	-
InternetGateway	igw-09a21d2a52825b141	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
PublicInternetRoute	rtb-06c40dcd752d8b27e|0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE	-
PublicRouteTable	rtb-06c40dcd752d8b27e	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PublicSecurityGroup	sg-067baa0ee1d1d2b1d	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
PublicSubnetA	subnet-06ae544196257475b	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnetARouteTableAssociation	rtbassoc-0db7d51ce13b5db31	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
ServerRole	NextWorkCodeDeployEC2Stack-ServerRole-i4Ng36oUfJNL	AWS::IAM::Role	CREATE_COMPLETE	-
VPC	vpc-00c1b72cb6f536737	AWS::EC2::VPC	CREATE_COMPLETE	-
VPCGatewayAttachment	IGW|vpc-00c1b72cb6f536737	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-
WebServer	i-03d29ce284ea467b9	AWS::EC2::Instance	CREATE_COMPLETE	-

Deployment Scripts

Scripts are mini lines of code set to do tasks. To set up CodeDeploy, I also wrote scripts to install dependencies and modify the Apache HTTPs server.

The 'install_dependencies will' install tomcat and Apache HTTP serverand configure apache to act as a reverse proxy for Tomcat.

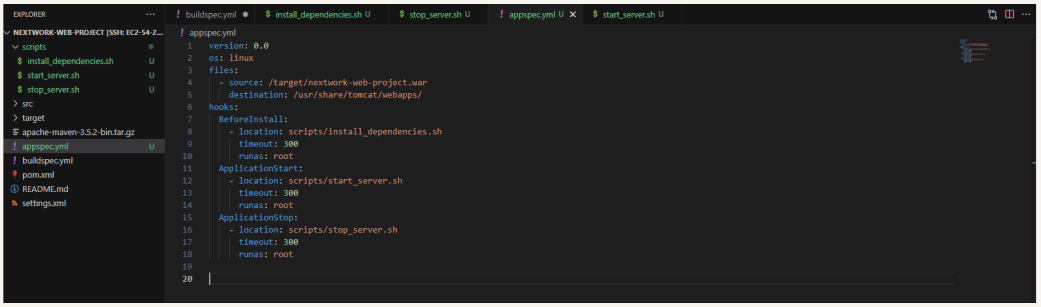
The start_server.sh will start Tomcat and the Apache HTTP server and set them to auto-start on reboot.

The stop_server.sh will check if the Tomcat and Apache servers are running and be able to stop the servers.

appspec.yml

Then, I wrote an appspec.yml file to act as the instruction manual for CodeDeploy. The key sections in appspec.yml are the format of the file, operating system, the files for the artifact and the hooks which are triggers that run at specific points

I also updated buildspec.yml because it needs to specify adding the newly added files to CodeBuild.

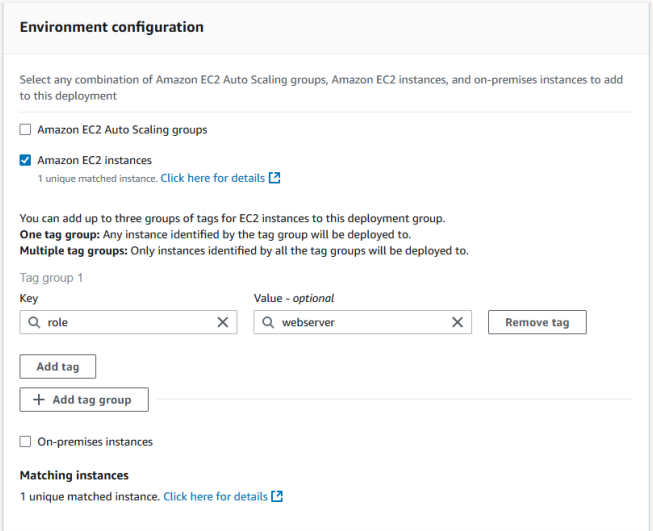


Setting Up CodeDeploy

A deployment group is a collection of EC2 instances where you plan out where and how the application gets deployed. A CodeDeploy application is like the main folder for the project.

To set up a deployment group, you also need to create an IAM role to enable it to access other AWS services.

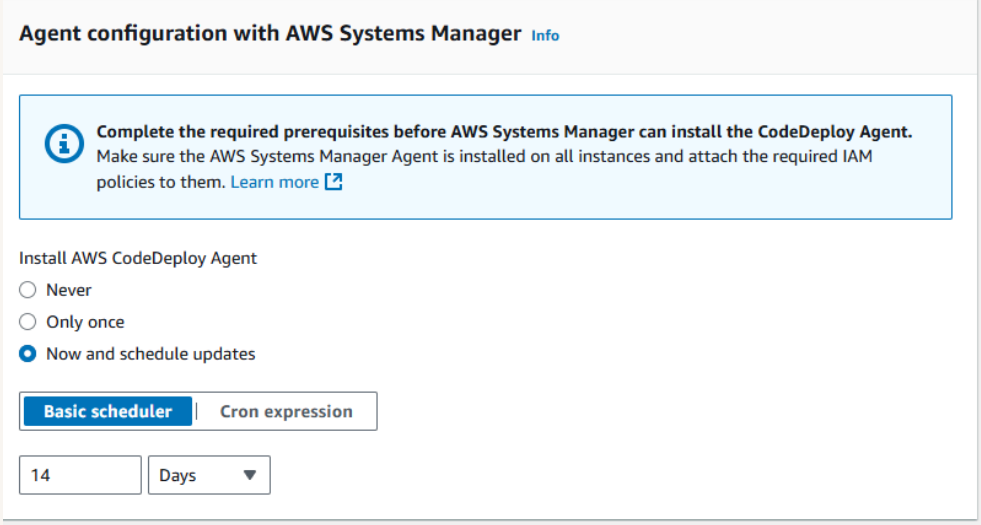
Tags are helpful for flexibility, self-documentation and integration. I used the tag role web server to find and deploy the correct instance.



Deployment configurations

Another key setting is the deployment configuration, which affects how quickly the application is deployed. I used CodeDeployDefault.AllAtOnce, so it is the fastest option, and there is only one instance being used.

In order to connect CodeDeploy, a CodeDeploy Agent is also set up to receive the deployment instructions.

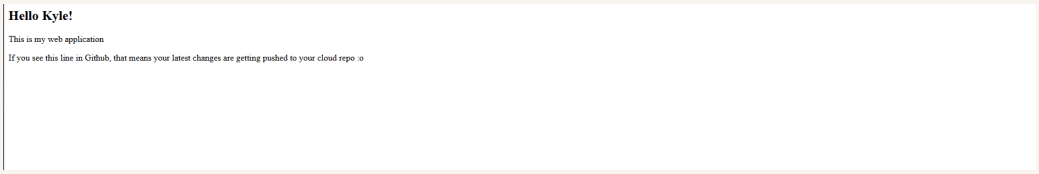


Success!

A CodeDeploy deployment is a representation of a single update to the application with its own ID and history. The difference to a deployment group is it is a single delivery of a software version.

I had to configure a revision location, which means the place where CodeDeploy is going to find the build artifact. My revision location was in the S3 bucket I created earlier.

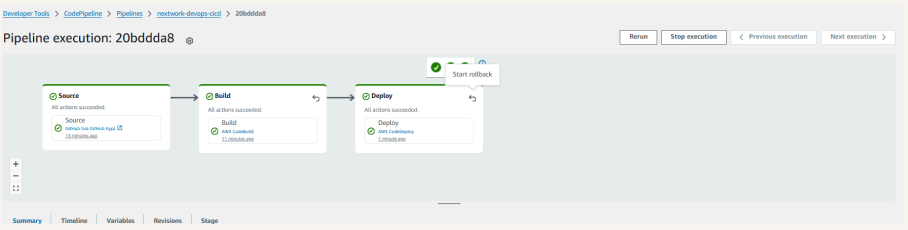
To check that the deployment was a success, I visited the public DNS and I saw the web application I created.



nextwork.org

Build a CI/CD Pipeline with AWS

Kyle Lowe



Introducing Today's Project!

In this project, I will demonstrate how to automate the CodePipeline. I'm doing this project to learn more about AWS CodePipeline and automating the CI/CD pipeline.

Key tools and concepts

Services I used were CodeDeploy, GitHub, IAM Roles. Key concepts I learnt include how to connect all the different tools together to create a CI/CD pipeline to automatically deploy changes.

Project reflection

This project took me approximately 90 minutes. The most challenging part was making sure that everything was working. It was most rewarding to see the changes made.

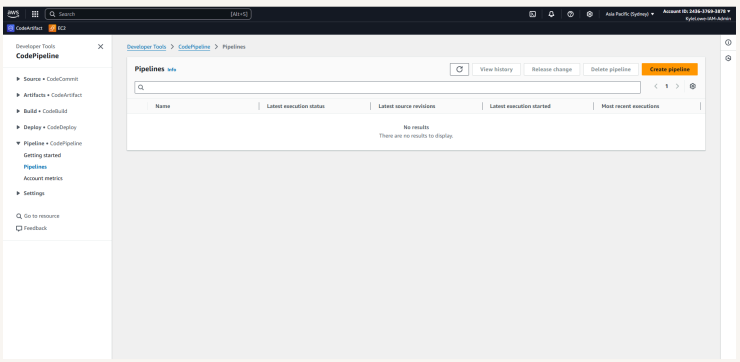
Kyle Lowe
NextWork Student

Starting a CI/CD Pipeline

AWS CodePipeline is a tool that is used to create a workflow that automatically triggers a CodeBuild and then a CodeDeploy.

CodePipeline offers different execution modes based on how CodePipeline handles multiple runs of the same pipeline. I chose Superseded other options include Queued and Parallel.

A service role gets created automatically during setup, so it gives permissions to other AWS resources that it needs to run the pipeline such as the S3 bucket for storing artifacts and CodeBuild.

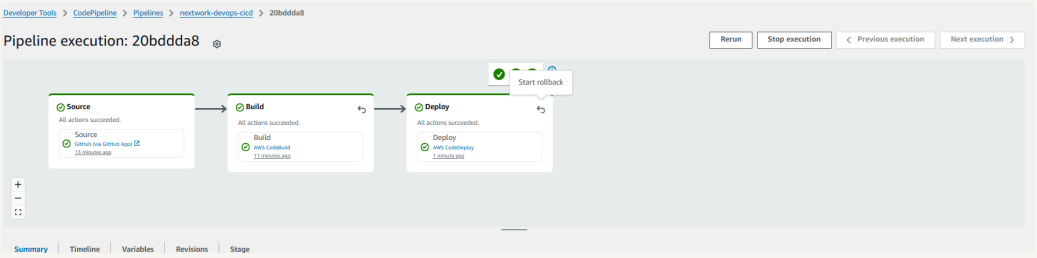


Kyle Lowe
NextWork Student

CI/CD Stages

The three stages I've set up in my CI/CD pipeline are source, build and deploy. While setting up each part, I learnt about setting up the source of the web application, building the application and deploying the application.

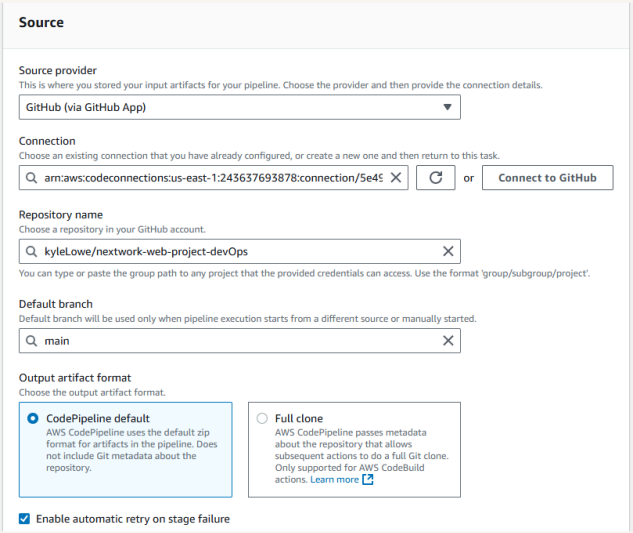
CodePipeline organizes the three stages into source, build and deploy. In each stage, you can see more details on the build process for each of the stages.



Source Stage

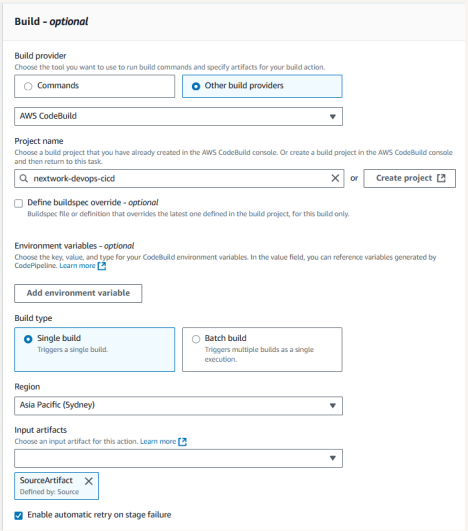
In the Source stage, the default branch tells CodePipeline to monitor the changes for the branch to trigger the automation.

The source stage is also where you enable webhook events, which act like digital notifications for when the master branch is pushed on GitHub.



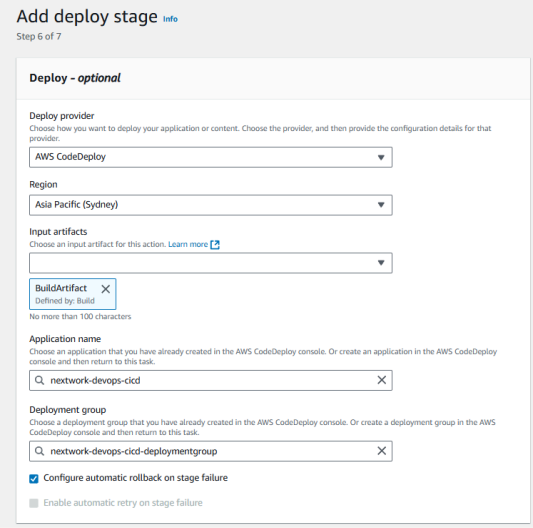
Build Stage

The Build stage sets up where the web application is being compiled and packaged. I configured the build project that is going to be compiled. The input artifact for the build stage is the outputs from the previous stage.



Deploy Stage

The Deploy stage is the final step in the pipeline, where it gets the application artifact and deploys it on the instance. I configured application that is being deployed and the location that it is going to be deployed to.



Success!

Since my CI/CD pipeline gets triggered by changes the the code for the web application. I tested my pipeline by changing the index file and pushing my changes to GitHub.

The moment I pushed the code change the pipeline started. The commit message under each stage reflects the latest commit in GitHub

Once my pipeline executed successfully, I checked the web application to see the changes made and the web application updated.