

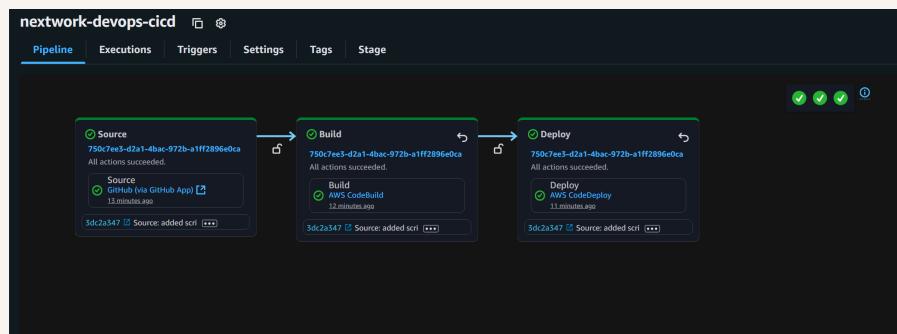


[nextwork.org](https://nextwork.org)

# Build a CI/CD Pipeline with AWS



naveen msd





# Introducing Today's Project!

In this project, I will demonstrate the code pipeline. I'm doing this project to learn how the pipeline reacts to the changes automatically.

## Key tools and concepts

Services I used were AWS Codepipeline Key concepts I learnt include configuring the code pipeline. making changes to the code and see how the process goes in updating it.

## Project reflection

This project took me approximately 1 hr. The most challenging part was configuring the pipeline and rolling back to the previous version. It was most rewarding to see how the code updates in each stage of the pipeline.

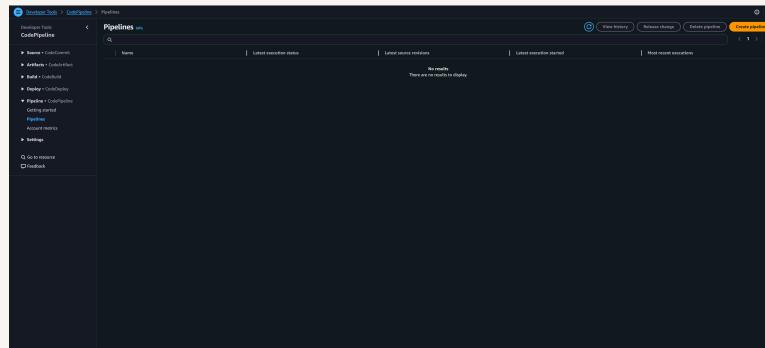


# Starting a CI/CD Pipeline

AWS CodePipeline creates a workflow that automatically moves the code changes through the build and deployment stage. Using CodePipeline makes sure the deployments are consistent, reliable and happen automatically to update the code.

CodePipeline offers different execution modes based on the current project. I chose Superseeded because it updates the newer version of code and cancels the older one other options include Queued and Parallel.

A service role gets created automatically during setup to perform actions. It's like permitting CodePipeline to access other AWS resources it needs to run your pipeline, such as S3 buckets for storing artifacts or CodeBuild for building.

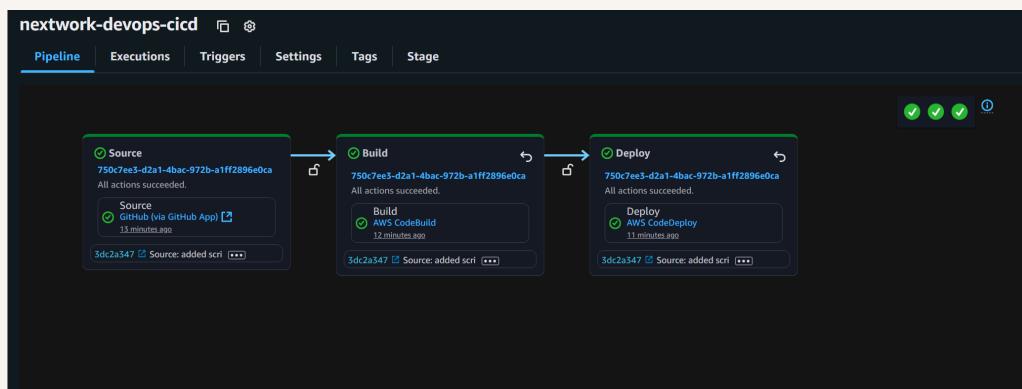




# CI/CD Stages

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 the process behind the deployment and how it deploys the code automatically.

CodePipeline organizes the three stages into source, build and deploy. In each stage, I can see more details on how the code is being pulled from each stage and deployed.

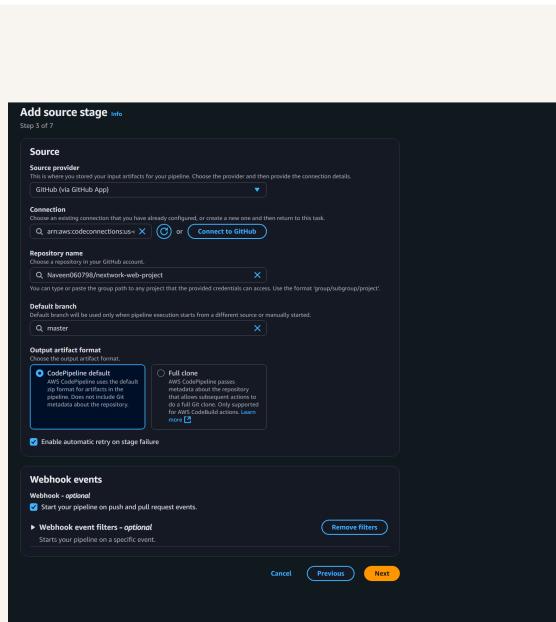




# Source Stage

In the Source stage, the default branch uses only when pipeline execution starts from a different source or manually started CodePipeline.

The source stage is also where I enable webhook events. CodePipeline automatically starts my pipeline whenever code is pushed to my specified branch in GitHub.





# Build Stage

The Build stage sets up source artifacts. I configured. The input artifact for the build stage is SourceArtifact, which is the ZIP file containing source code that was outputted by the Source stage.

Add build stage Info

Step 4 of 7

**Build - optional**

**Build provider**  
Choose the tool you want to use to run build commands and specify artifacts for your build action.

Commands  Other build providers

AWS CodeBuild

**Project name**  
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

nextwork-devops-cicd

Define buildspec override - optional  
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

**Environment variables - optional**  
Configure environment variables for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

**Build type**

Single build  Batch build

Triggers a single build. Triggers multiple builds as a single execution.

**Region**  
United States (Ohio)

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#)

SourceArtifact

Enable automatic retry on stage failure



# Deploy Stage

The Deploy stage is where it's responsible for taking the application artifacts (the output from the Build stage) and deploying them to the target environment, which in this case is an EC2 instance.

Add deploy stage Info

Step 6 of 7

**Deploy - optional**

**Deploy provider**  
Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

**Region**  
United States (Ohio)

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#)

BuildArtifact Defined by Build

No more than 100 characters

**Application name**  
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

Q nextwork-devops-cicd

**Deployment group**  
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

Q nextwork-devops-cicd-deploymentgroup

Configure automatic rollback on stage failure

Enable automatic retry on stage failure

[Cancel](#) [Previous](#) [Skip deploy stage](#) [Next](#)



# Success!

Since my CI/CD pipeline gets triggered by the changes made in git repo. I tested my pipeline by adding a line to index.jsp file.

The moment I pushed the code change to git hub repo. The commit message under each stage reflects and reacts to the changes. I confirmed by looking into the source by clicking on the commit id, to see what has changed.

Once my pipeline executed successfully, I checked by going to the deploy stage and by accessing the instance with a public IPv4 address. I confirmed the changes.

---

## Hello Naveen!

This is my NextWork web application working!

If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o

If you see this line, that means your latest changes are automatically deployed into production by CodePipeline!



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

