

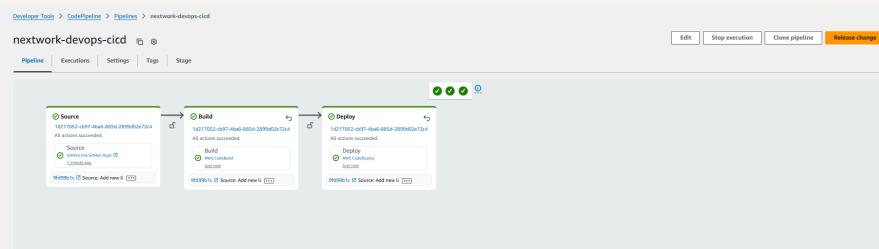


[nextwork.org](https://nextwork.org)

# Build a CI/CD Pipeline with AWS



Saish Nar





**Saish Nar**

NextWork Student

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

---

# Introducing Today's Project!

In this project, I will demonstrate how to build a complete CI/CD pipeline using AWS CodePipeline. I'm doing this project to learn how to connect source, build, and deploy stages for fully automated, reliable deployments.

## Key tools and concepts

Services we used were CodePipeline, CodeDeploy, CodeBuild, CodeArtifact, GitHub, VS Code, EC2, S3, IAM and CloudFormation. Key concepts I learnt include the different stages in a CI/CD pipeline, handling rollbacks and webhooks.

## Project reflection

This project took me approximately 3 hours to complete. The most challenging part was integrating all the AWS services smoothly. It was most rewarding to watch the pipeline run automatically and deploy the updates without any manual intervention.

**Saish Nar**  
NextWork Student

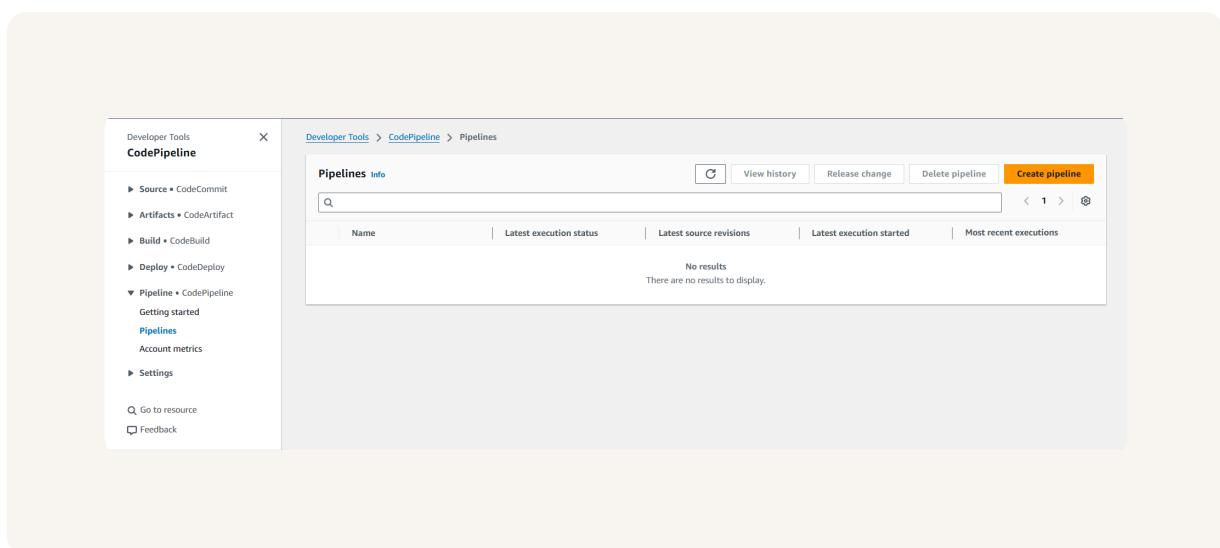
[nextwork.org](http://nextwork.org)

# Starting a CI/CD Pipeline

Aws CodePipeline is an AWS DevOps tool that helps me to create a workflow that automatically moves code from GitHub all the way to CodeDeploy. This helps to make sure that deployments are consistent and reliable - with less risk of human errors!

CodePipeline offers different execution modes based on how I treat multiple runs of the same pipeline. I chose Superseded(new run cancels the old), other options include Queued (runs in order) and Parallel (runs simultaneously).

A service role gets created automatically during setup so CodePipeline can access and manage AWS resources like S3, CodeBuild, and CodeDeploy on your behalf.





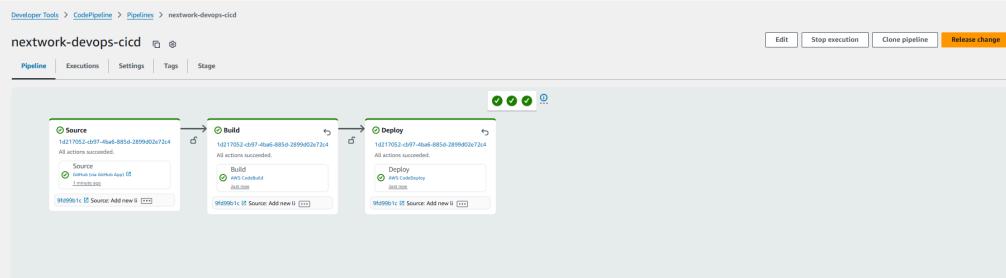
**Saish Nar**  
NextWork Student

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

# CI/CD Stages

The three stages I've set up in my CI/CD pipeline are Source (source code for my web project in GitHub), Build (building the web app using CodeBuild), Deploy (deploying changes to my web app using CodeDeploy).

CodePipeline organizes the three stages into a single diagram that showcases the flow of changes from Source to Deploy. In each stage, you can see more details on the pipeline execution that it belongs to, and shortcuts to the connected service.





# Source Stage

In the Source stage, the default branch tells CodePipeline which branch to monitor for code changes , so it knows when to trigger the pipeline automatically - usually from the MASTER branch, stable codebase.

The source stage is also where you enable webhook events, which are like notifications! Whenever I make a change to the source code, the webhook event will detect the change and alert CodePipeline to trigger a new run.

Add source stage Info  
Step 3 of 7

**Source**

Source provider  
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.  
 GitHub (via GitHub App)

Connection  
Choose an existing connection that you have already configured, or create a new one and then return to this task.  
 arnawscodeconnections:ap-south-1:connection/b4  or  Connect to GitHub

Repository name  
Choose a repository in your GitHub account.  
 Q\_SaishNar/nextwork-web-project

You can type or paste the group path to any project that the provided credentials can access. Use the format: [group]/[subgroup]/[project].

Default branch  
Default branch will be used only when pipeline execution starts from a different source or manually started.  
 master

Output artifact format  
Choose the output artifact format.

**CodePipeline default**  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

**Full clone**  
AWS CodePipeline passes metadata about the repository that allows downstream actions to perform a full Git clone. Only supported for AWS CodeBuild actions. [Learn more](#)

Enable automatic retry on stage failure

**Webhook events**

Webhook - optional  
 Start your pipeline on push and pull request events.

**Saish Nar**  
NextWork Student

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

# Build Stage

The Build stage sets up the environment to compile and test the code. I configured SourceArtifact as the input artifact for the build stage because it contains the source code from the Source stage.

The screenshot shows the 'Add build stage' configuration screen in AWS CodePipeline, Step 4 of 7. The form is titled 'Build - optional'. It includes fields for 'Build provider' (set to 'AWS CodeBuild'), 'Project name' ('nextwork-devops-cicd'), 'Environment variables', 'Build type' (set to 'Single build'), 'Region' ('Asia Pacific (Mumbai)'), 'Input artifacts' (set to 'SourceArtifact'), and a checkbox for 'Enable automatic retry on stage failure'.

**Saish Nar**  
NextWork Student

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

# Deploy Stage

The Deploy stage is where I set up CodeDeploy to be my deployment provider! it takes the BuildArtifact from CodeBuild, and the application and deployment settings that I've defined in my deployment group.

Add deploy stage [Info](#)  
Step 6 of 7

**Deploy - optional**

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

AWS CodeDeploy

Region  
Asia Pacific (Mumbai)

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

BuildArtifact [X](#)  
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 [X](#)

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-deployment-group [X](#)

Configure automatic rollback on stage failure  
 Enable automatic retry on stage failure

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



**Saish Nar**  
NextWork Student

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

# Success!

Since my CI/CD pipeline gets triggered by code changes, I tested my pipeline by updating my code! I added a new line to my web app's index.jsp file, and pushed those changes too.

The moment I pushed the code change, a new execution started automatically. The commit message under each stage reflects the update I made to index.jsp, and the Build and Deploy stages completed successfully, turning green in the pipeline.

Once my pipeline executed successfully, I checked my live web app and can confirm that it updated without having to manually re-build my project in Codebulid and re-deploy the project in CodeDeploy! The Changes went live into production straight away.



**Saish Nar**  
NextWork Student

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

### **Hello saish !**

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

