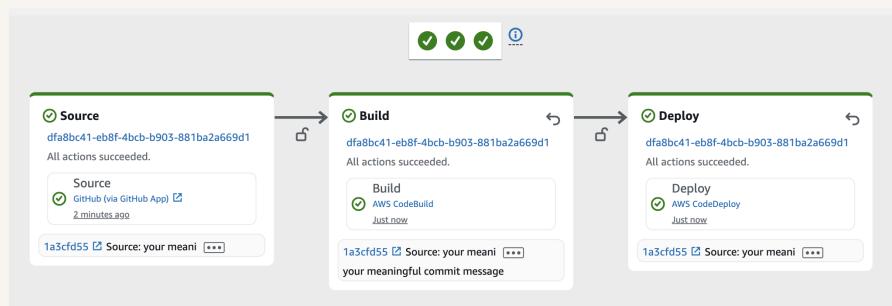




Build a CI/CD Pipeline with AWS



Gaurav Balpande



Introducing Today's Project!

In this project, I will demonstrate the use of CodePipeline in cicd project I'm doing this project to learn about the automation process by cicd which will automate github build,build and deploy phase.

Key tools and concepts

Services I used were Github,Codebuild,CodeDeploy and CodePipeline Key concepts I learnt include how pipeline automates the complete deployment process and get handon on CodeDeploy.

Project reflection

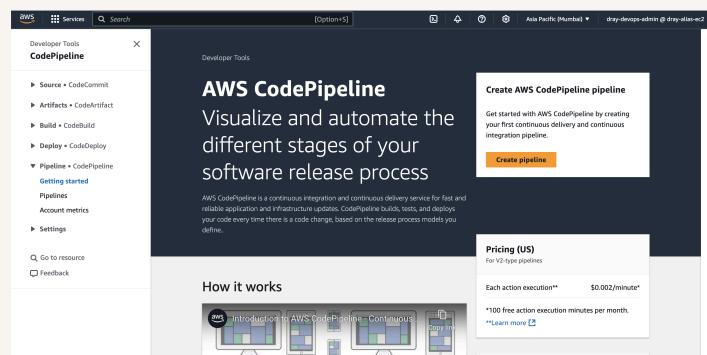
This project took me approximately 1:30 hrs The most challenging part was the the build phase It was most rewarding to see the web page automatically get the change which we made in the code.

Starting a CI/CD Pipeline

AWS CodePipeline is an important feature in aws which is used to automate the build and deploy phase. With the help of Codepipeline the changes we make in the code will automatically move from build stage to deploy stage.

CodePipeline offers different execution modes based on how you want to run multiple build I chose Superseded other options include Queued and parallel.

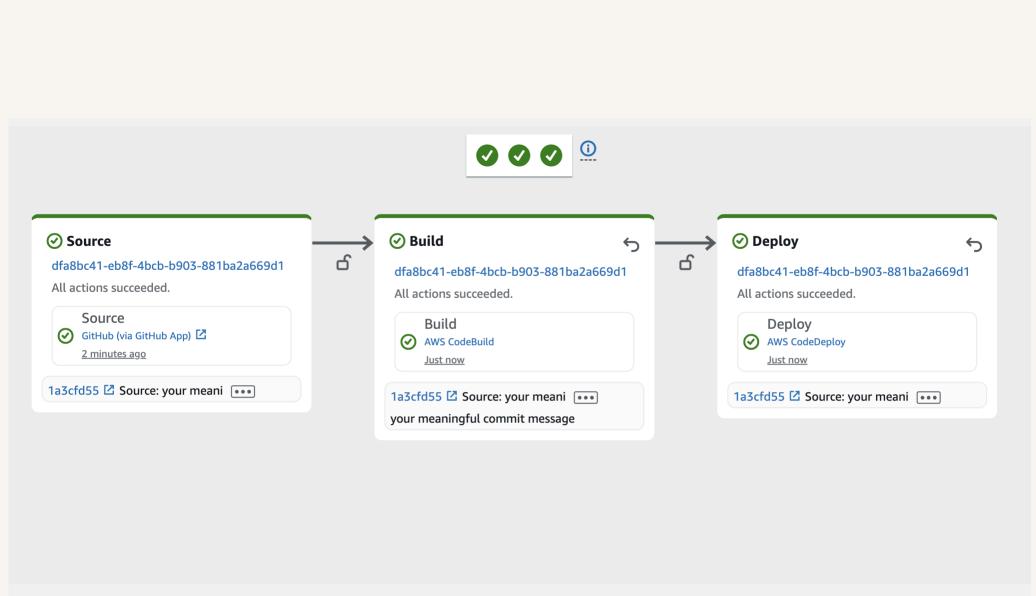
A service role gets created automatically during setup so it gets the permission to access different aws services like S3 for storing artifat and others. Codepipeline initially does not have permission to access different aws services.



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 how the actual pass-on occur during pipeline and the structure for pipeline which include source, build and deploy stage.

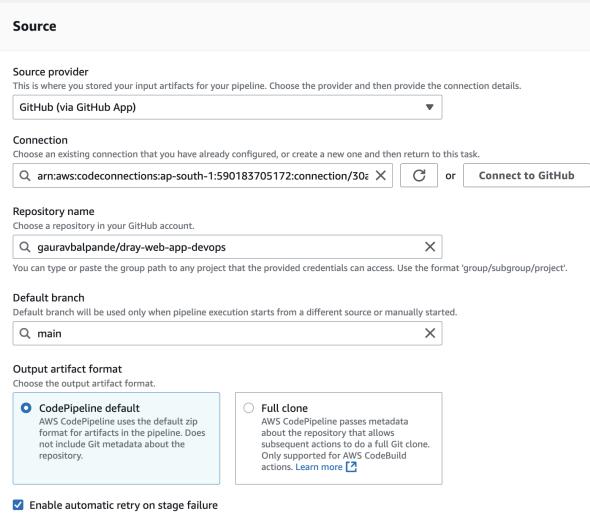
CodePipeline organizes the three stages into Source ,Build and Deploy. In each stage, you can see more details on the id which is in top-rightmost of the field. It gives us overall information about the process.



Source Stage

In the Source stage, the default branch tells CodePipeline about where actually it has to look for code so that other branches should not get affected. In these phase we give the branch and github connection.

The source stage is also where you enable webhook events, which is like a notification. It gets triggered whenever there is some push in the code of the given branch. It helps in automating.



Build Stage

The Build stage sets up your code into package which is further used for deployment I configured the input SourceArtifact The input artifact for the build stage is SourceArtifact

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-ci cd or Create project

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
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

Add environment variable

Build type

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

Region
 Asia Pacific (Mumbai)

Input artifacts
Choose an input artifact for this action. Learn more

SourceArtifact X
Defined by: Source

Enable automatic retry on stage failure

Deploy Stage

The Deploy stage is where our package got into the Ec2 instance for accessing through internet I configured the deploy phase.

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.

Region

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

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.

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.

nextwork-devops-cicd-deployment-group

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

Success!

Since my CI/CD pipeline gets triggered by making change in the code I tested my pipeline by adding "<p>If you see this line, that means your latest changes are automatically deployed into production by CodePipeline!</p> " in index.jsp

The moment I pushed the code change the pipeline as it triggered it The commit message under each stage reflects that our change is the one who make it trigger.

Once my pipeline executed successfully, I checked visiting the public ipv4 of the instance on which we deployed the code. By pasting the link it shows the change in the web page which confirm our pipeline is successful.



Hello Gaurav Balpande!

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

The place to learn & showcase your skills

Check out nextwork.org for more projects

