DevOps CI/CD on AWS

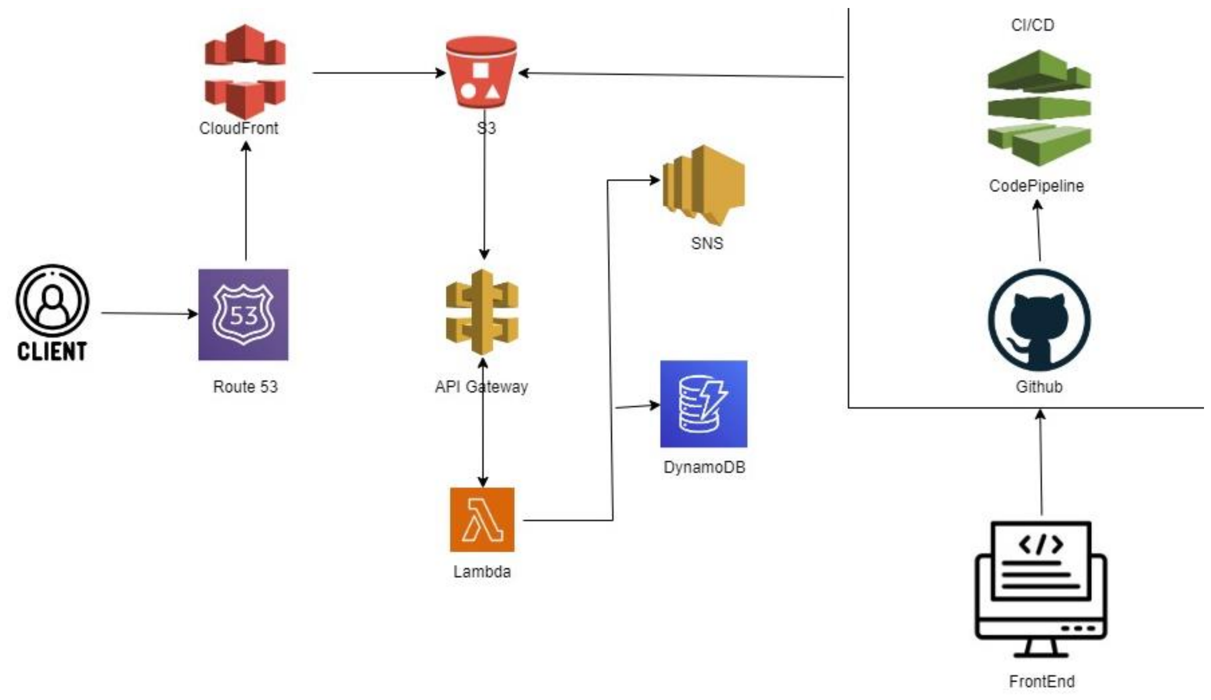
Automatically deploy the changes to Static Website on S3 using CodePipeline

This blog post will teach us how to set up a pipeline that updates the S3 bucket on a regular basis. Any modifications we make to the source files in our source repository and then deploy to the S3 bucket are detected by the entire pipeline. Every time we make changes to a file in the source repository, the website is created using the most recent version via deployment.

Overview of CI/CD:

By adding automation to the various stages of app development, CI/CD is a technique for regularly delivering apps to customers. Continuous integration, continuous delivery, and continuous deployment are the three core CI/CD concepts. CI/CD is a solution to the issues that development and operations teams may encounter while integrating new code.

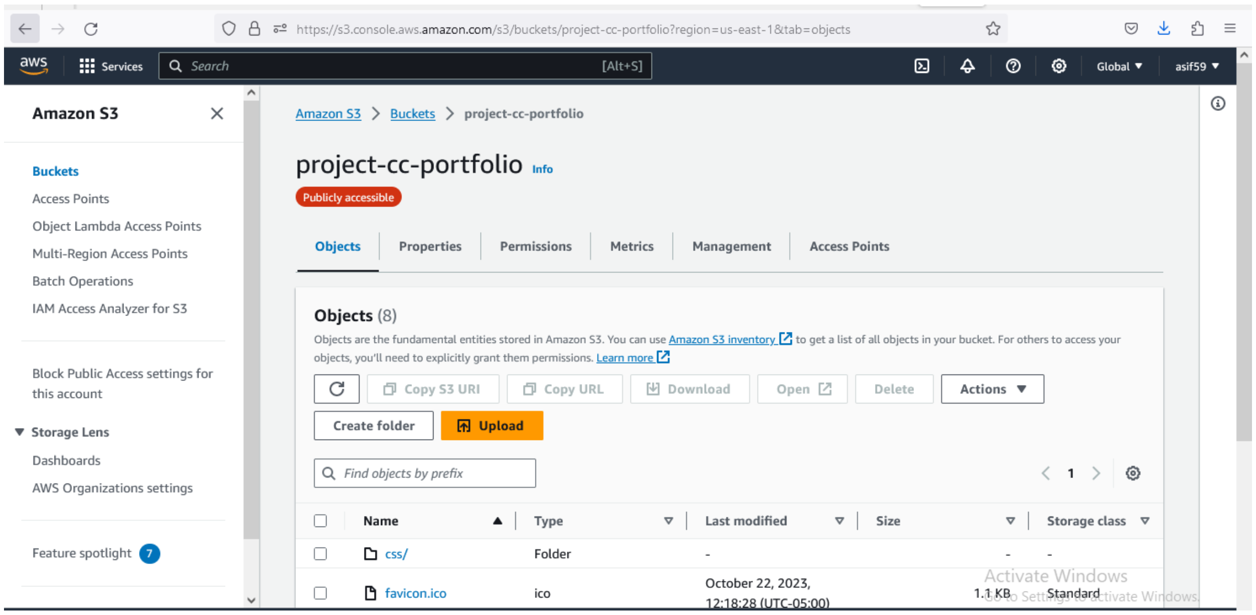
In particular, CI/CD brings continuous monitoring and continual automation throughout the whole lifespan of an app, from the integration and testing stages to the delivery and deployment stages. When considered collectively, these interconnected processes are frequently referred to as a "CI/CD pipeline" and are supported by development and operations teams cooperating in an agile manner using either a DevOps or site reliability engineering (SRE) strategy.



Implementation Steps:

Step 1:

Create a S3 bucket, where files will be deployed.



https://s3.console.aws.amazon.com/s3/buckets/project-cc-portfolio?region=us-east-1&tab=objects

Amazon S3 > Buckets > project-cc-portfolio

project-cc-portfolio Info

Publicly accessible

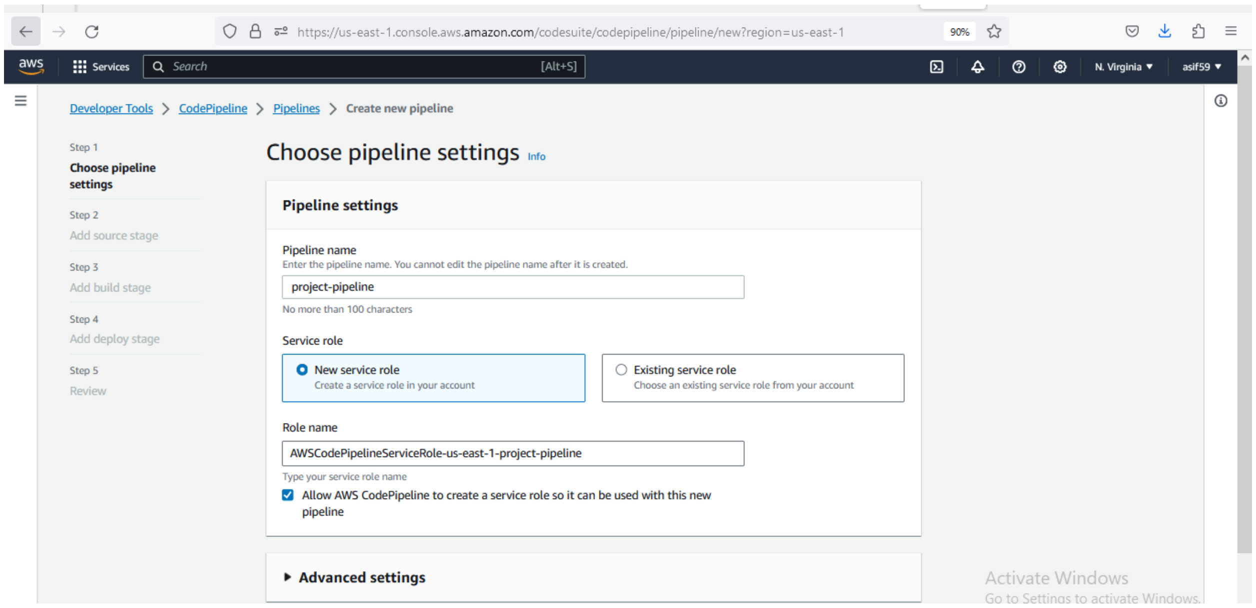
Objects (8)

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
css/	Folder	-	-	-
favicon.ico	ico	October 22, 2023, 12:18:28 (UTC-05:00)	1.1 KB	Standard

Step 2:

Create a Pipeline. The following procedures must be followed in order to create a code pipeline.



https://us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipeline/new?region=us-east-1

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Choose pipeline settings Info

Pipeline settings

Pipeline name: project-pipeline
Enter the pipeline name. You cannot edit the pipeline name after it is created.
No more than 100 characters

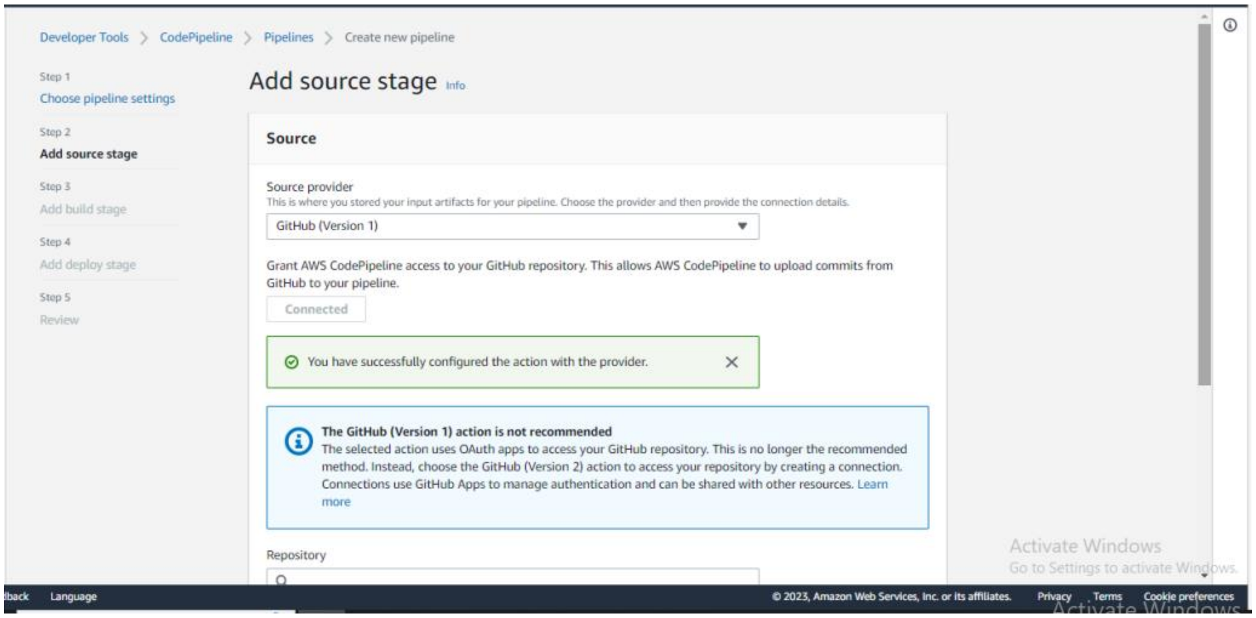
Service role: [✓ New service role] Create a service role in your account / Existing service role — Choose an existing service role from your account

Role name: AWSCodePipelineServiceRole-us-east-1-project-pipeline
Type your service role name

[✓] Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Advanced settings

Now we need to add source provider. For this time we will use Github version 1. Then we need to grant AWS CodePipeline access to the Github repository. This allows AWS CodePipeline to upload commits from Github to our pipeline. We also need to select repository and branch.


Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings
Step 2 Add source stage
Step 3 Add build stage
Step 4 Add deploy stage
Step 5 Review

Add source stage Info

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 1)

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.
Connected

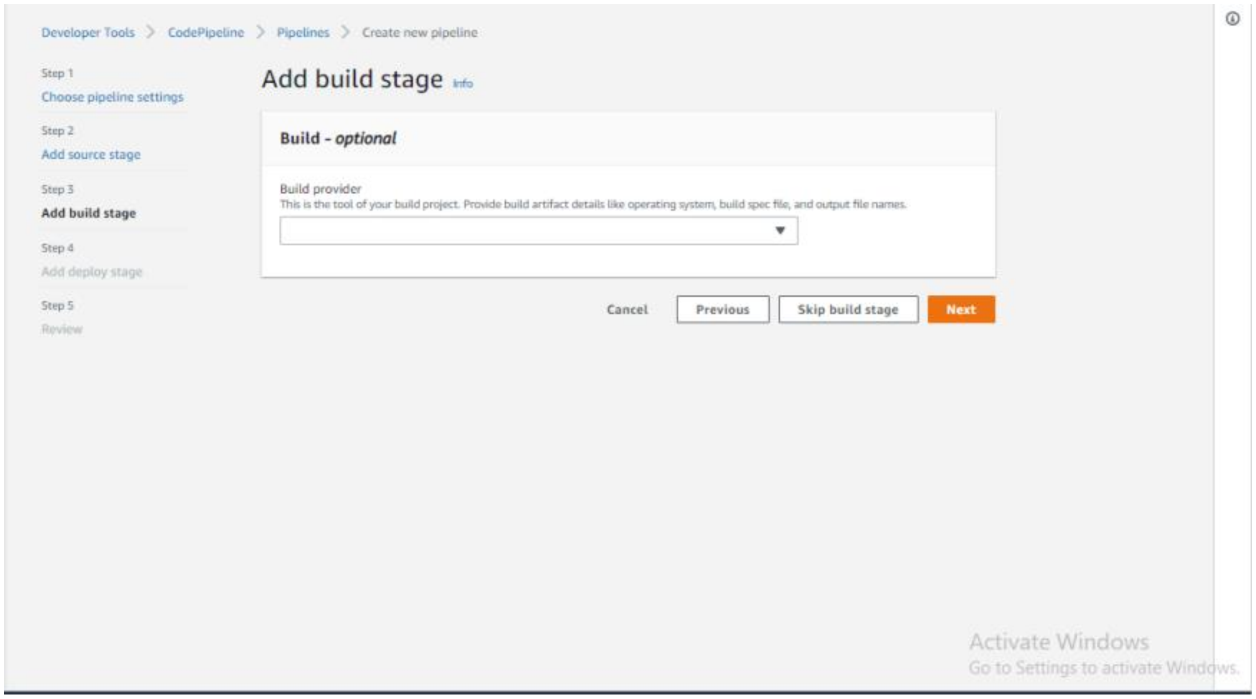
You have successfully configured the action with the provider.

The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. Learn more

Repository

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

In this step we may add build stage but it is optional. For this tutorial we skipped it. For skipping it click button skip build stage.


Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings
Step 2 Add source stage
Step 3 Add build stage
Step 4 Add deploy stage
Step 5 Review

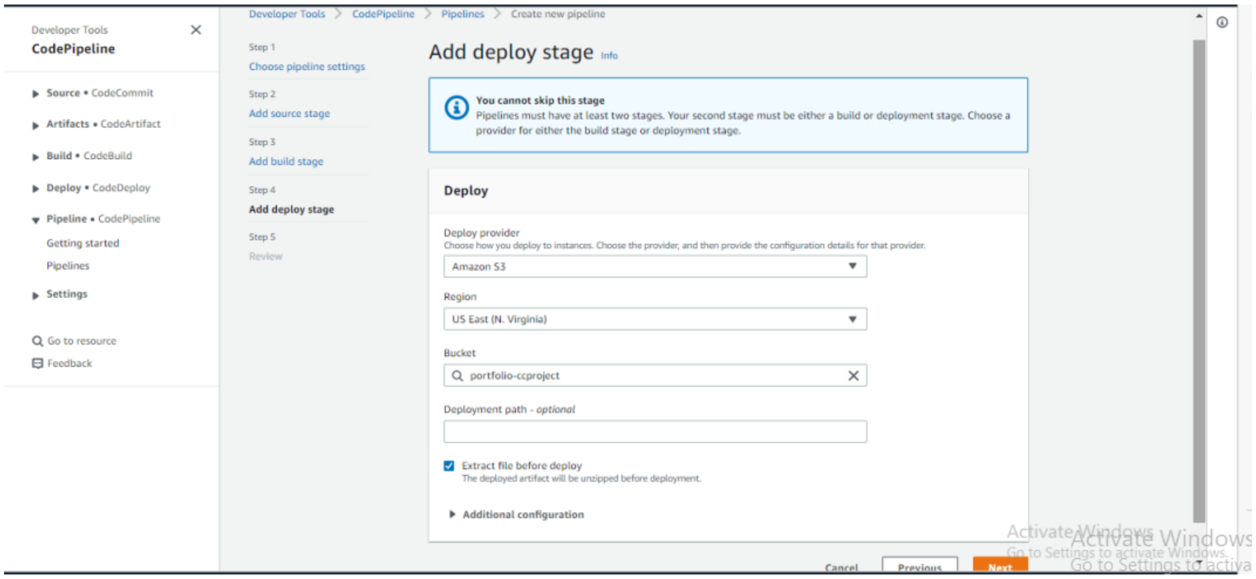
Add build stage Info

Build - *optional*

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Cancel | Previous | Skip build stage | Next

In deploy stage we need to choose deploy stage and then provide the configuration details for the provider.


Developer Tools > CodePipeline > Pipelines > Create new pipeline

Developer Tools
CodePipeline
- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
  - Getting started
  - Pipelines
- Settings

Go to resource
Feedback

Step 1 Choose pipeline settings
Step 2 Add source stage
Step 3 Add build stage
Step 4 Add deploy stage
Step 5 Review

Add deploy stage Info

You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
Amazon S3

Region
US East (N. Virginia)

Bucket
portfolio-ccproject

Deployment path - *optional*

☑ Extract file before deploy
The deployed artifact will be unzipped before deployment.

Additional configuration

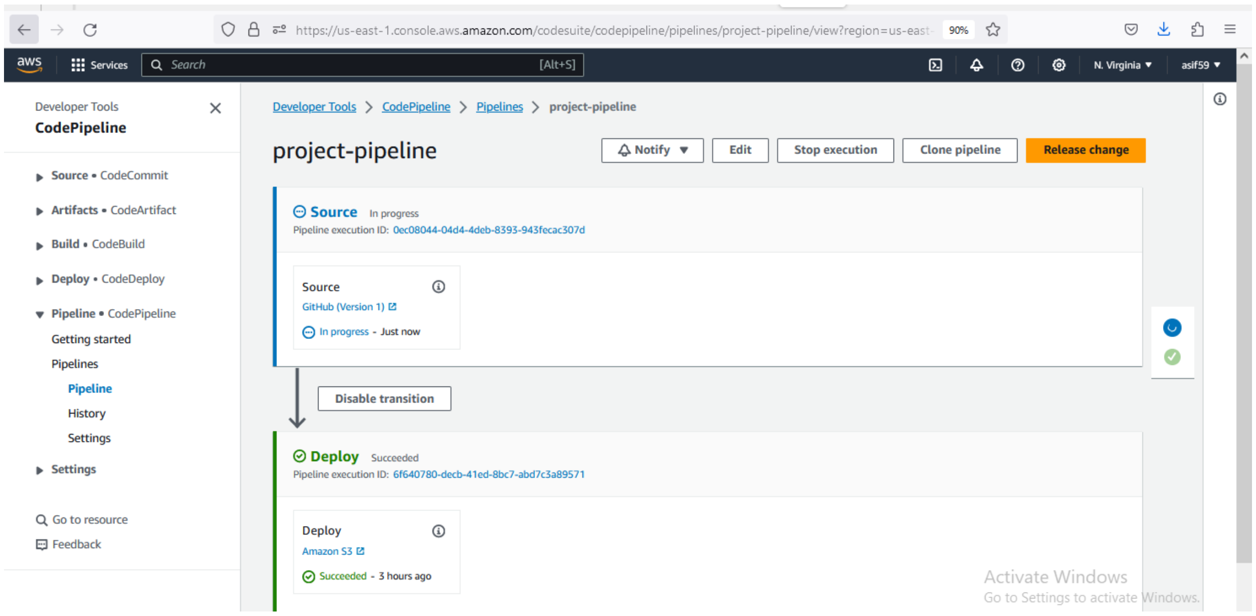
Cancel Previous Next

Now review the pipeline settings and hit create pipeline button.

Step 3:

After creating the CodePipeline, it will detect the files from our Github repository and deploy in S3 bucket.

Here, It detected the new commit and is in progress for deployment.


https://us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/project-pipeline/view?region=us-east

Developer Tools > CodePipeline > Pipelines > project-pipeline

Developer Tools
CodePipeline
- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
  - Getting started
  - Pipelines
    - Pipeline
    - History
    - Settings
- Settings

Go to resource
Feedback

project-pipeline

Notify Edit Stop execution Clone pipeline Release change

Source In progress
Pipeline execution ID: 0ec08044-04d4-4deb-8393-943fecac307d

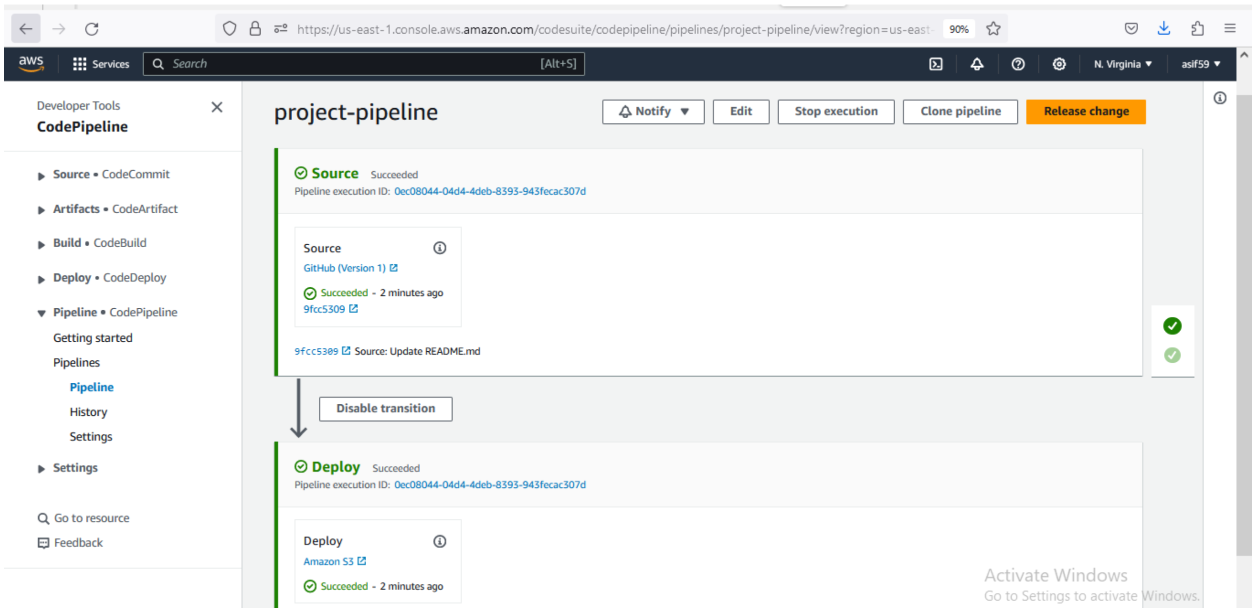
Source
GitHub (Version 1)
In progress - Just now

Disable transition

Deploy Succeeded
Pipeline execution ID: 6f640780-decb-41ed-8bc7-abd7c3a89571

Deploy
Amazon S3
Succeeded - 3 hours ago

Here, it successfully deployed the new changes.



https://us-east-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/project-pipeline/view?region=us-east

project-pipeline

Source Succeeded
Pipeline execution ID: 0ec08044-04d4-4deb-8393-943fecac307d

Source
GitHub (Version 1)
Succeeded - 2 minutes ago
9fcc5309

9fcc5309 Source: Update README.md

Deploy Succeeded
Pipeline execution ID: 0ec08044-04d4-4deb-8393-943fecac307d

Deploy
Amazon S3
Succeeded - 2 minutes ago

The Key Benefits of CI/CD

- **Faster Release Cycles:** CI/CD helps teams to release software more frequently and react swiftly to customer needs by automating the testing and deployment process.
- **Improved Quality:** Automated testing makes ensuring that software updates don't add any new defects or problems, enhancing the software's overall quality.
- **Increased Collaboration:** Developer collaboration and communication improve as a result of the frequent code integration and testing.
- **Reduced Risk:** Continuous deployment lowers the risk of catastrophic failures and downtime by enabling developers to immediately discover and address issues.
- **Cost-Effective:** Through the use of CI/CD, less manual task is needed to distribute software updates, saving time and money.

As a result of the drawbacks of the conventional, linear method to software development, CI/CD was developed. Teams can collaborate, release software more frequently, and react rapidly to customer requests thanks to CI/CD, which makes software development more agile and collaborative.

Author
Asif Ahammad Miazee