Set up a Continuous Deployment Pipeline using AWS CodePipeline

By

Akshat Parikh

AWS CodePipeline

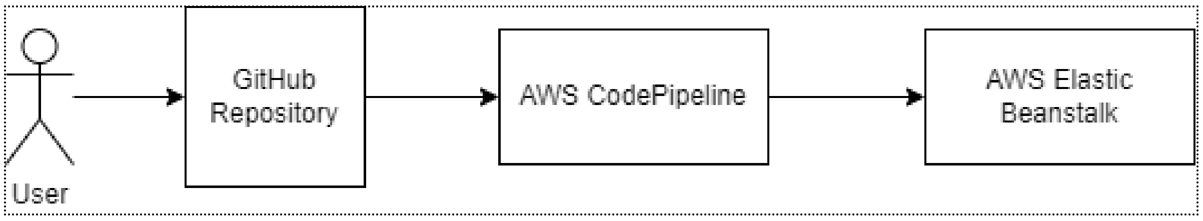
- AWS CodePipeline is a managed continuous delivery solution that assists in the automation of release pipelines for quick and dependable application and infrastructure changes.
- It automates the build, test, and deploy parts of the release process whenever there is a code change.

Application Setup

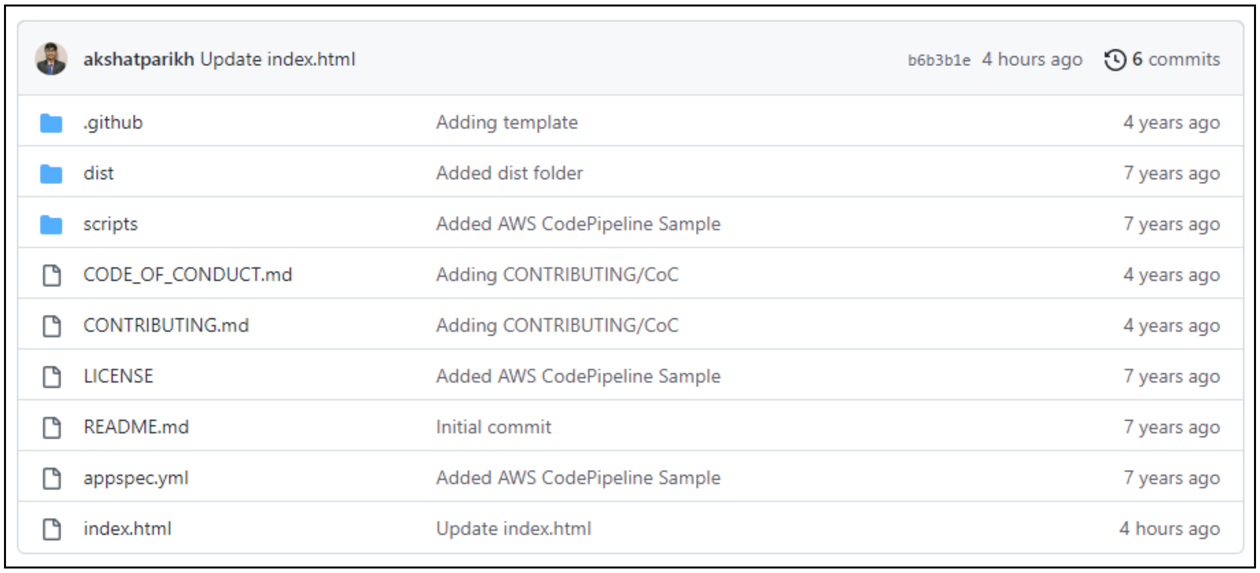
In this demo, I have used GitHub as a version control system for Continuos Integration purposes and Elastic Beanstalk and CodePipeline for Continuous Deployment purposes.

Elastic Beanstalk is the most straightforward method for deploying and running a web application on Amazon Web Services. Elastic Beanstalk handles deployment elements such as capacity provisioning, load balancing, automatic scaling, and web application health monitoring automatically. Elastic Beanstalk gives you complete control over the Amazon Web Services resources that power your web application (EC2, S3, CloudWatch, Elastic Load Balancers etc).

Architecture



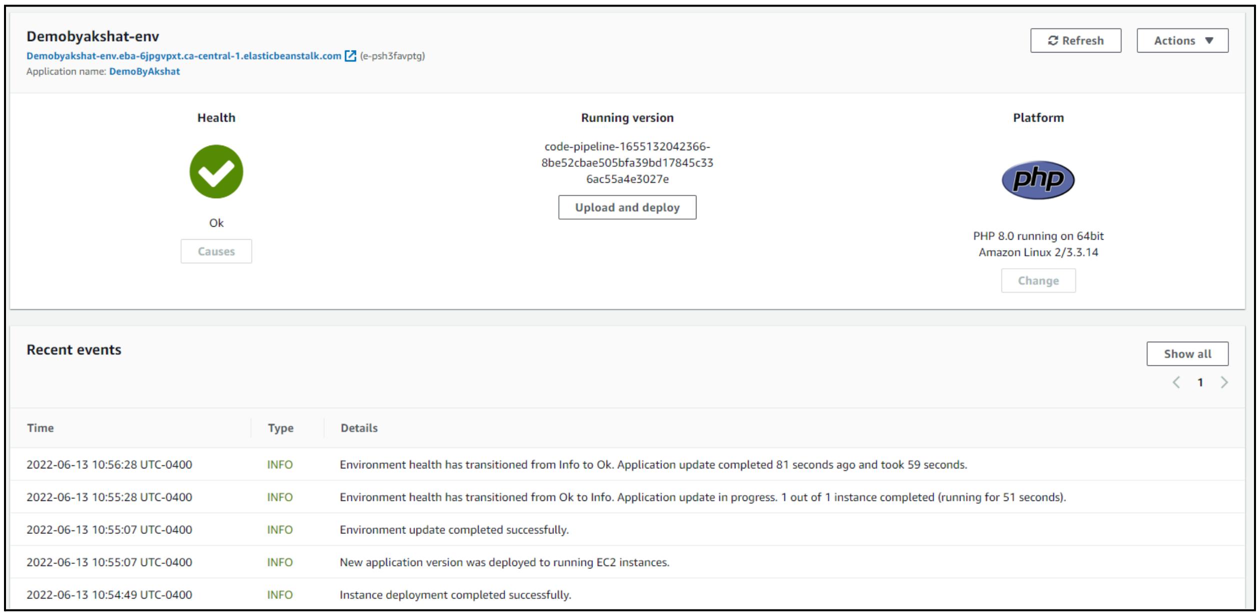
First of all, I configured a repo on Github. I have used a simple web application that prints the output from the index file.



akshatparikh Update index.html b6b3b1e 4 hours ago 6 commits

.github	Adding template	4 years ago
dist	Added dist folder	7 years ago
scripts	Added AWS CodePipeline Sample	7 years ago
CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC	4 years ago
CONTRIBUTING.md	Adding CONTRIBUTING/CoC	4 years ago
LICENSE	Added AWS CodePipeline Sample	7 years ago
README.md	Initial commit	7 years ago
appspec.yml	Added AWS CodePipeline Sample	7 years ago
index.html	Update index.html	4 hours ago

Now, I have created an AWS Elastic Beanstalk environment.



Demobyakshat-env
Demobyakshat-env.eba-6jpgvpxt.ca-central-1.elasticbeanstalk.com (e-psh3favptg)
Application name: DemoByAkshat

Health
Ok

Running version
code-pipeline-1655132042366-8be52cbae505bfa39bd17845c336ac55a4e3027e

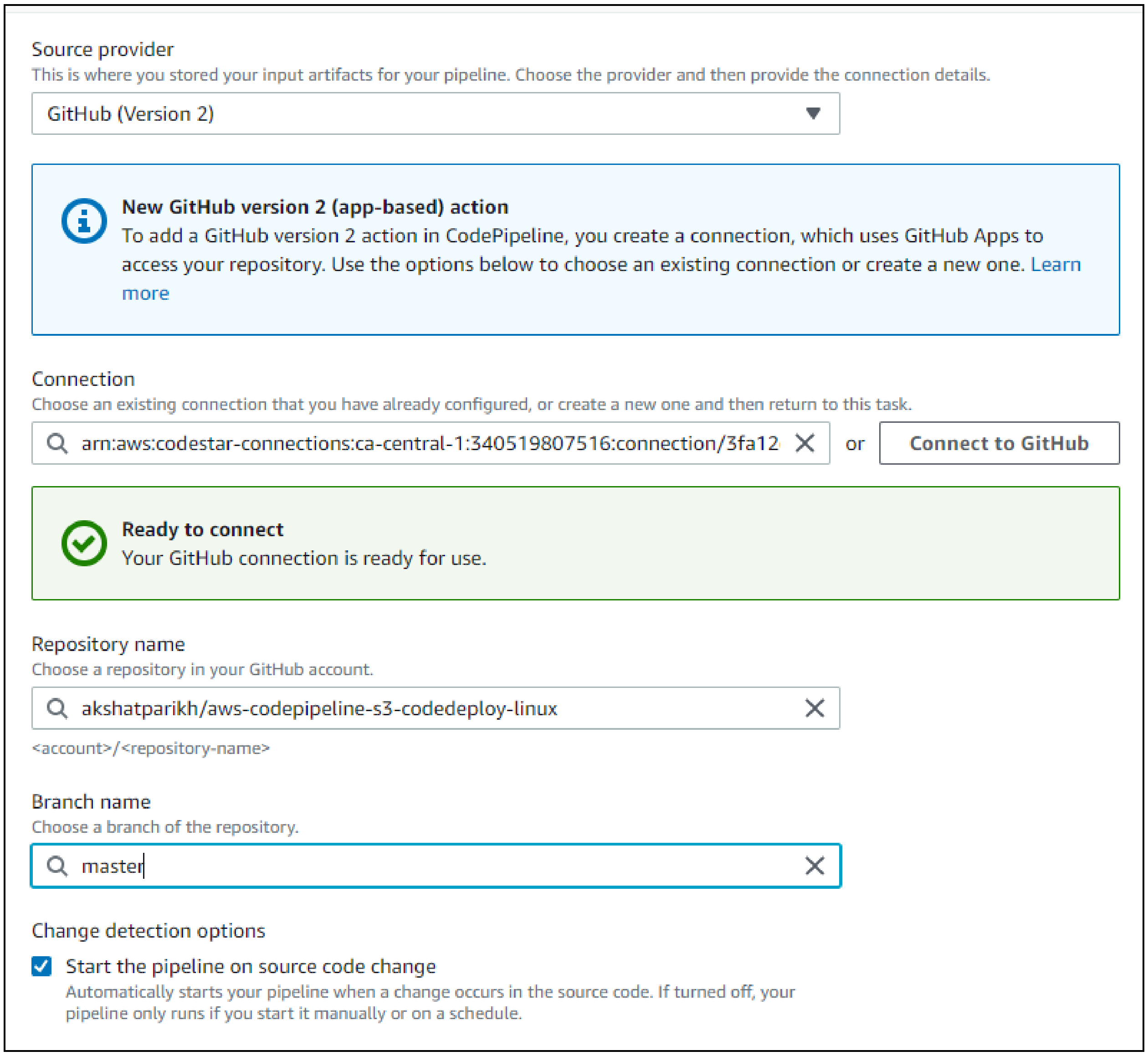
Platform
PHP 8.0 running on 64bit Amazon Linux 2/3.3.14

Recent events

Time	Type	Details
2022-06-13 10:56:28 UTC-0400	INFO	Environment health has transitioned from Info to Ok. Application update completed 81 seconds ago and took 59 seconds.
2022-06-13 10:55:28 UTC-0400	INFO	Environment health has transitioned from Ok to Info. Application update in progress. 1 out of 1 instance completed (running for 51 seconds).
2022-06-13 10:55:07 UTC-0400	INFO	Environment update completed successfully.
2022-06-13 10:55:07 UTC-0400	INFO	New application version was deployed to running EC2 instances.
2022-06-13 10:54:49 UTC-0400	INFO	Instance deployment completed successfully.

Once done, we will proceed to create an AWS CodePipeline that will be responsible for connecting to our GitHub account and building and deploying the application to Elastic Beanstalk.

So, on the console page for the code pipeline, we will name the pipeline and connect it to the GitHub repo.



Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2)

New GitHub version 2 (app-based) action

To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codestar-connections:ca-central-1:340519807516:connection/3fa12 or Connect to GitHub

Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

akshatparikh/aws-codepipeline-s3-codedeploy-linux

<account>/<repository-name>

Branch name

Choose a branch of the repository.

master

Change detection options

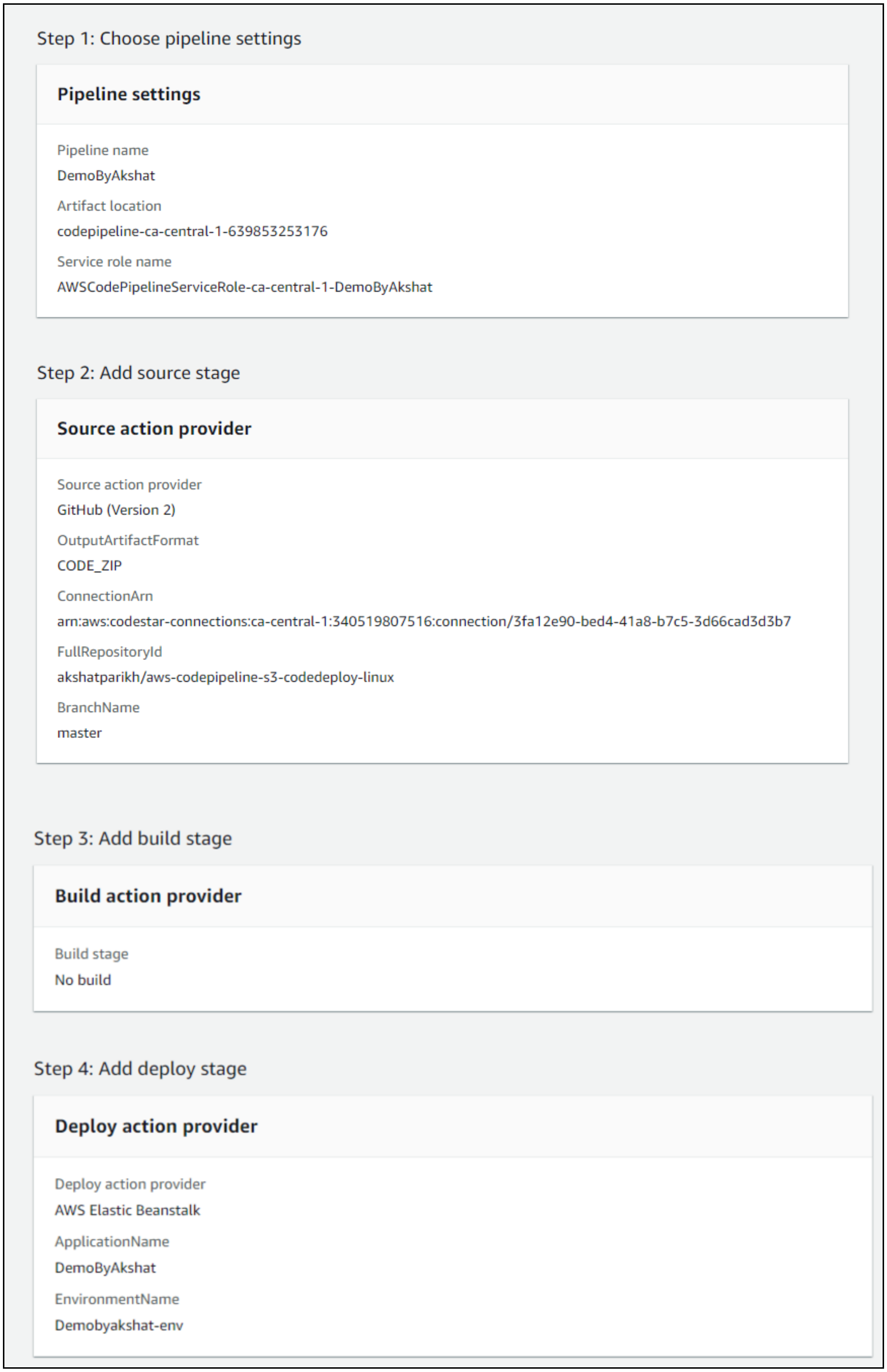
[✓] Start the pipeline on source code change

Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

As shown above, we have successfully established the connection to our repository.

Now, we will skip the CodeBuild section in this demo. The next step is to choose a deployment provider and that will be the elastic beanstalk environment that we have created earlier in this demo.

Now, we will review our final pipleine.



Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
DemoByAkshat

Artifact location
codepipeline-ca-central-1-639853253176

Service role name
AWSCodePipelineServiceRole-ca-central-1-DemoByAkshat

Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 2)

OutputArtifactFormat
CODE_ZIP

ConnectionArn
arn:aws:codestar-connections:ca-central-1:340519807516:connection/3fa12e90-bed4-41a8-b7c5-3d66cad3d3b7

FullRepositoryId
akshatparikh/aws-codepipeline-s3-codedeploy-linux

BranchName
master

Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

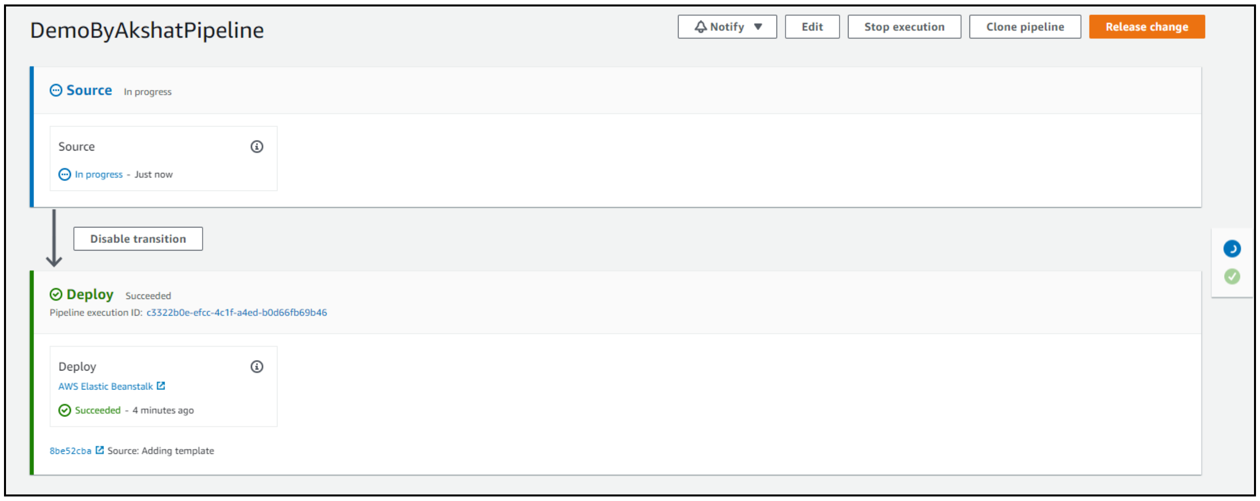
Deploy action provider

Deploy action provider
AWS Elastic Beanstalk

ApplicationName
DemoByAkshat

EnvironmentName
Demobyakshat-env

Once the pipeline is created, it will run automatically and we can see two stages, “Source” and “Deploy” running for the pipeline.



DemoByAkshatPipeline

Source In progress

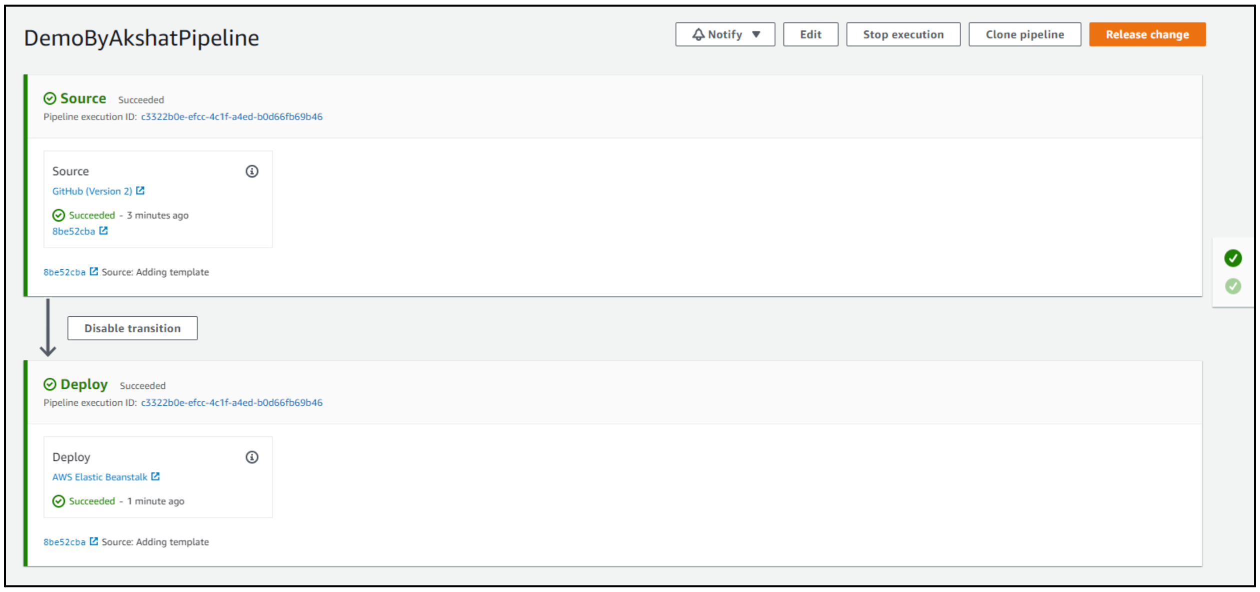
Source
In progress - Just now

Deploy Succeeded
Pipeline execution ID: c3322b0e-efcc-4c1f-a4ed-b0d66fb69b46

Deploy
AWS Elastic Beanstalk
Succeeded - 4 minutes ago

8be52cba Source: Adding template

As we can see, the pipeline succeeded.



DemoByAkshatPipeline

Source Succeeded
Pipeline execution ID: c3322b0e-efcc-4c1f-a4ed-b0d66fb69b46

Source
GitHub (Version 2)
Succeeded - 3 minutes ago
8be52cba

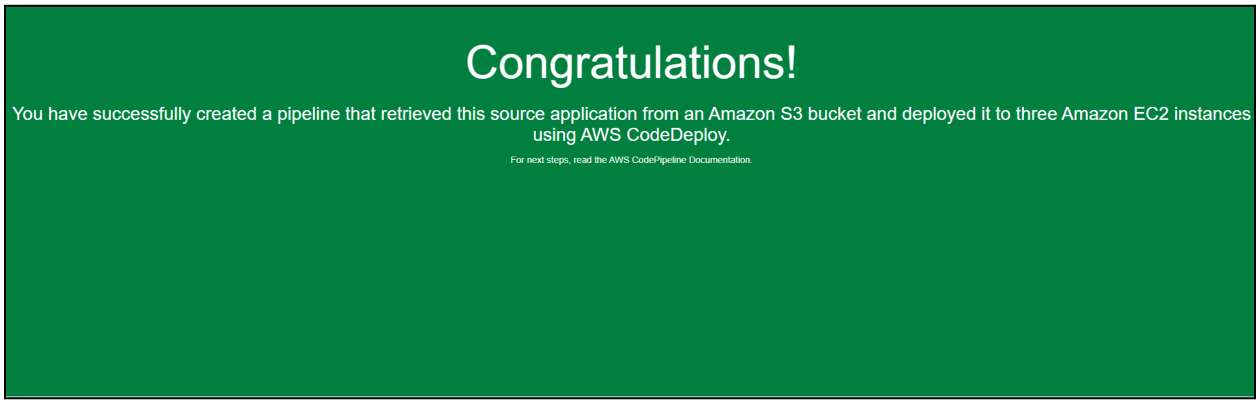
8be52cba Source: Adding template

Deploy Succeeded
Pipeline execution ID: c3322b0e-efcc-4c1f-a4ed-b0d66fb69b46

Deploy
AWS Elastic Beanstalk
Succeeded - 1 minute ago

8be52cba Source: Adding template

Now, we can go to the Beanstalk environment and take a look at the deployed application.



Congratulations!

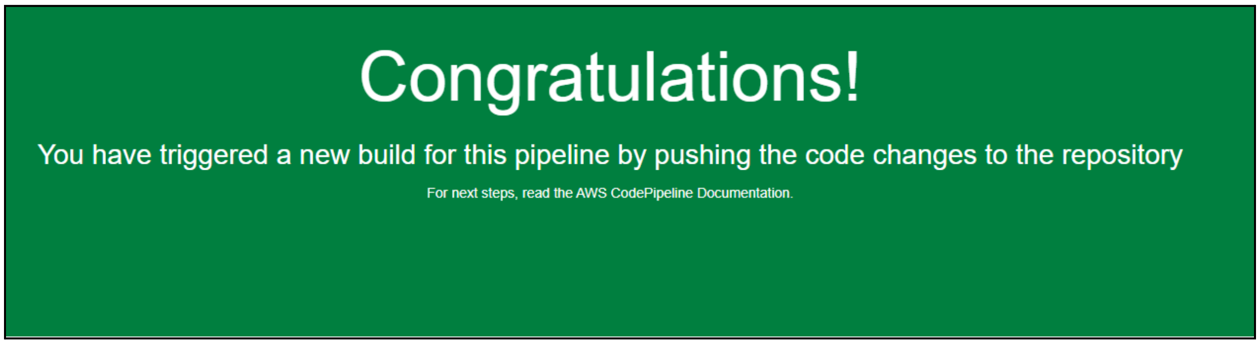
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation.

Nice! The application is deployed.

Now to verify that the pipeline is being triggered on every commit or not, I will change the index.html file and commit the changes.

So, I made some changes and that triggered the pipeline, and we can see the changed output below.



Congratulations!

You have triggered a new build for this pipeline by pushing the code changes to the repository

For next steps, read the AWS CodePipeline Documentation.