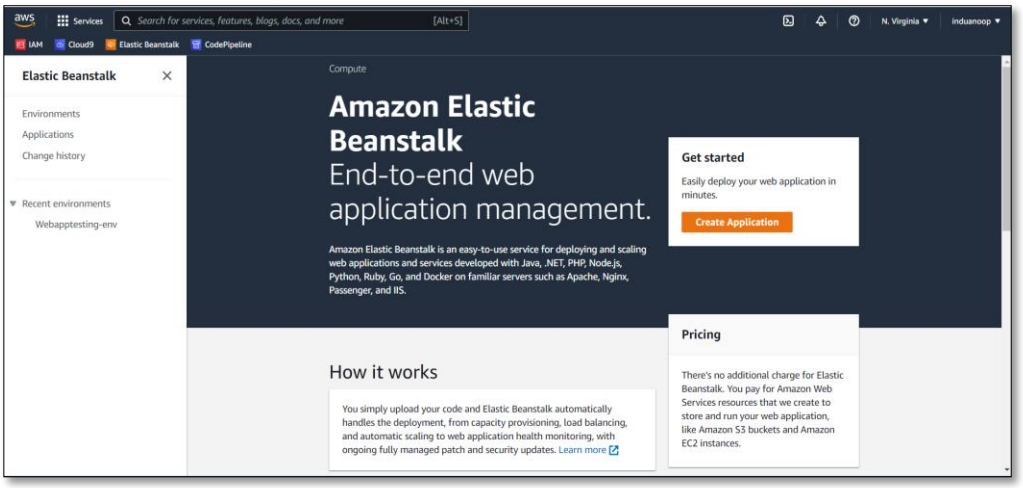
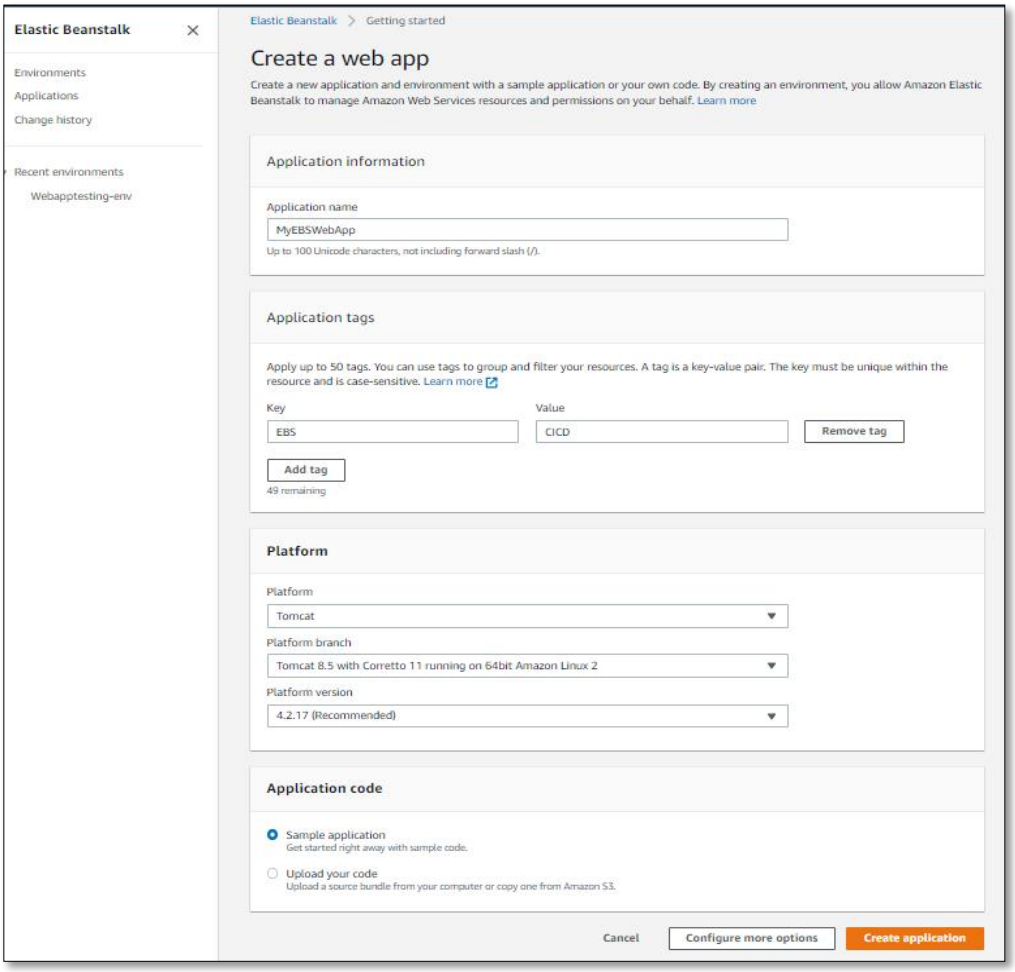
Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	(Leave blank for now)

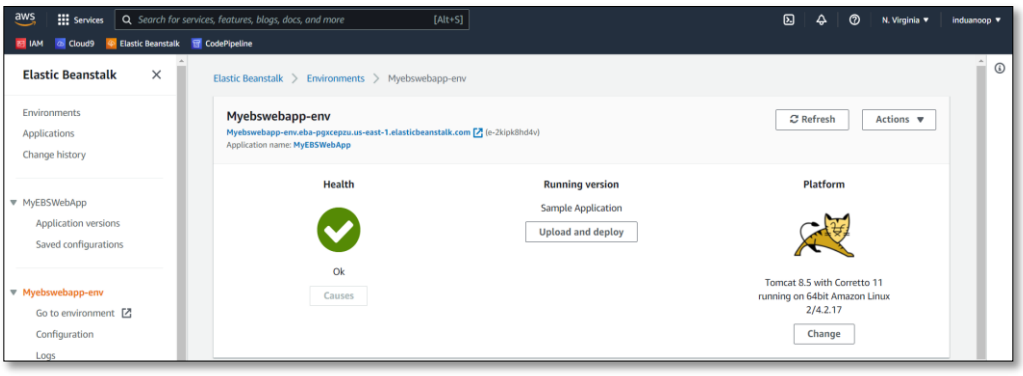
Student Name		
Roll Number		
Grade and Subject Teacher's Signature		

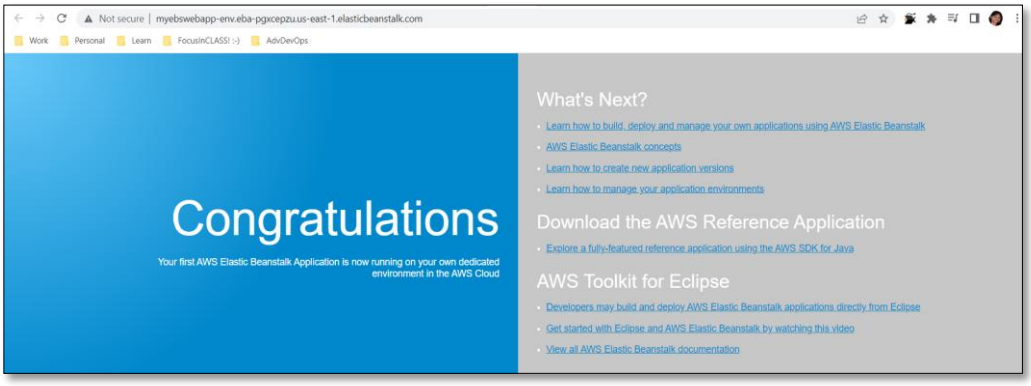
Experiment	2	
Problem Statement	To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	**Theory:** Continuous deployment allows you to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated. You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app	
Code	**Steps:** **Step1: Create a deployment environment** Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before creating the pipeline. To simplify the process of setting up and configuring EC2	

	instances for this experiment, you will spin up a sample environment using AWS Elastic Beanstalk. Elastic Beanstalk lets you easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (e.g., virtual servers, load balancers, etc.) and provides the application stack (e.g., OS, language and framework, web, and application server, etc.) for you. 



Elastic Beanstalk will begin creating a sample environment for you to deploy your application to. It will create an Amazon EC2 instance, a security group, an Auto Scaling group, an Amazon S3 bucket, Amazon CloudWatch alarms, and a domain name for your application. Note: This will take several minutes to complete.





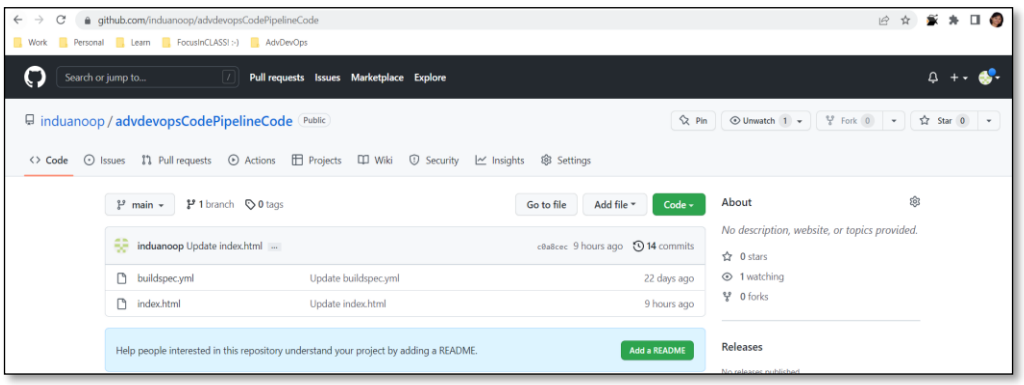
Step2: Get a copy of the sample code

In this step, you will retrieve a copy of the sample app's code and choose a source to host the code. The pipeline takes code from the source and then performs actions on it. You can use one of three options as your source:

- **a GitHub repository,**
- an Amazon S3 bucket, or
- an AWS CodeCommit repository.

A sample project's GitHub repository can be created in your GitHub account, or you can fork(clone) the following GitHub repository. (Ensure there is a buildspec.yml file in your repository for the Build Phase)

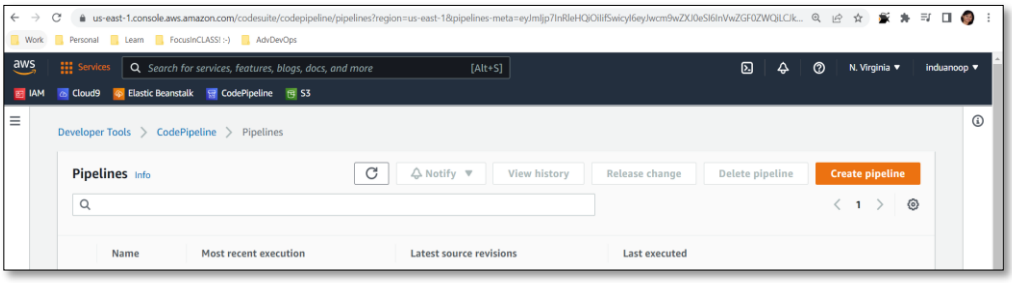
https://github.com/induanoop/advdevopsCodePipelineCode.git

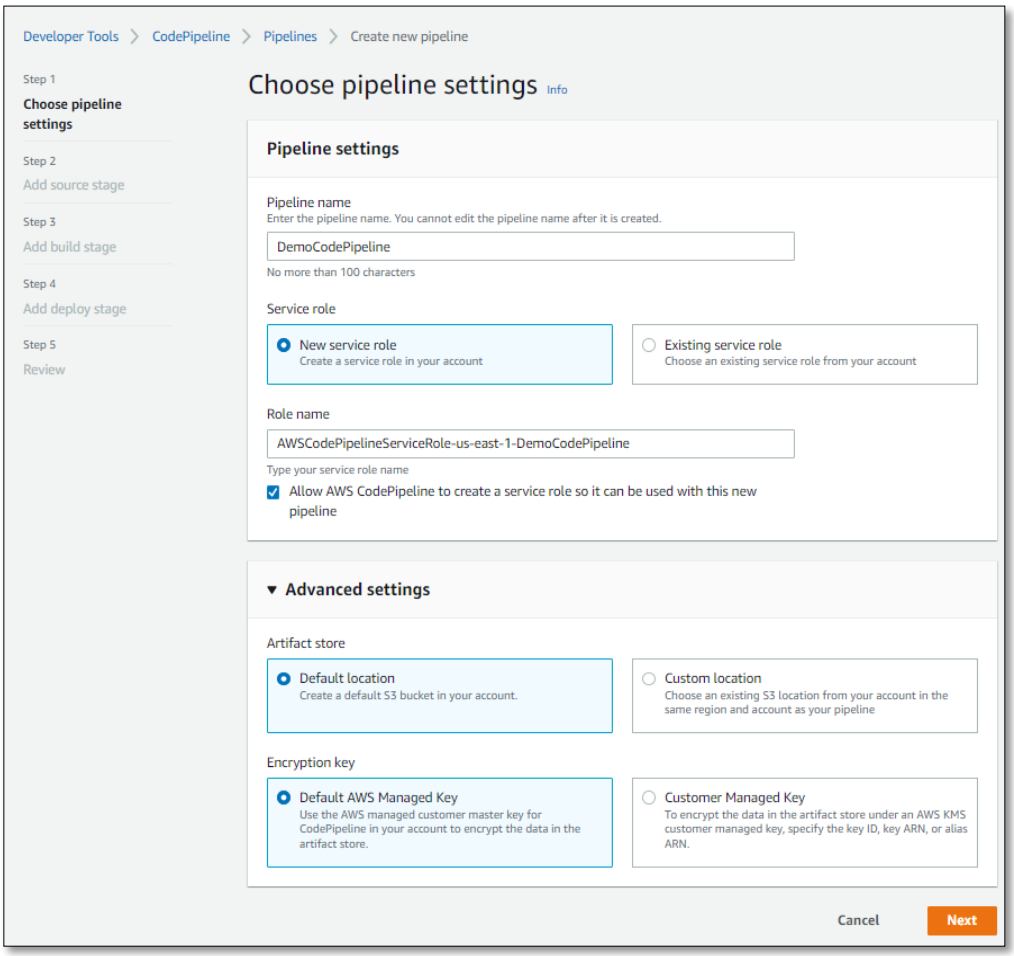


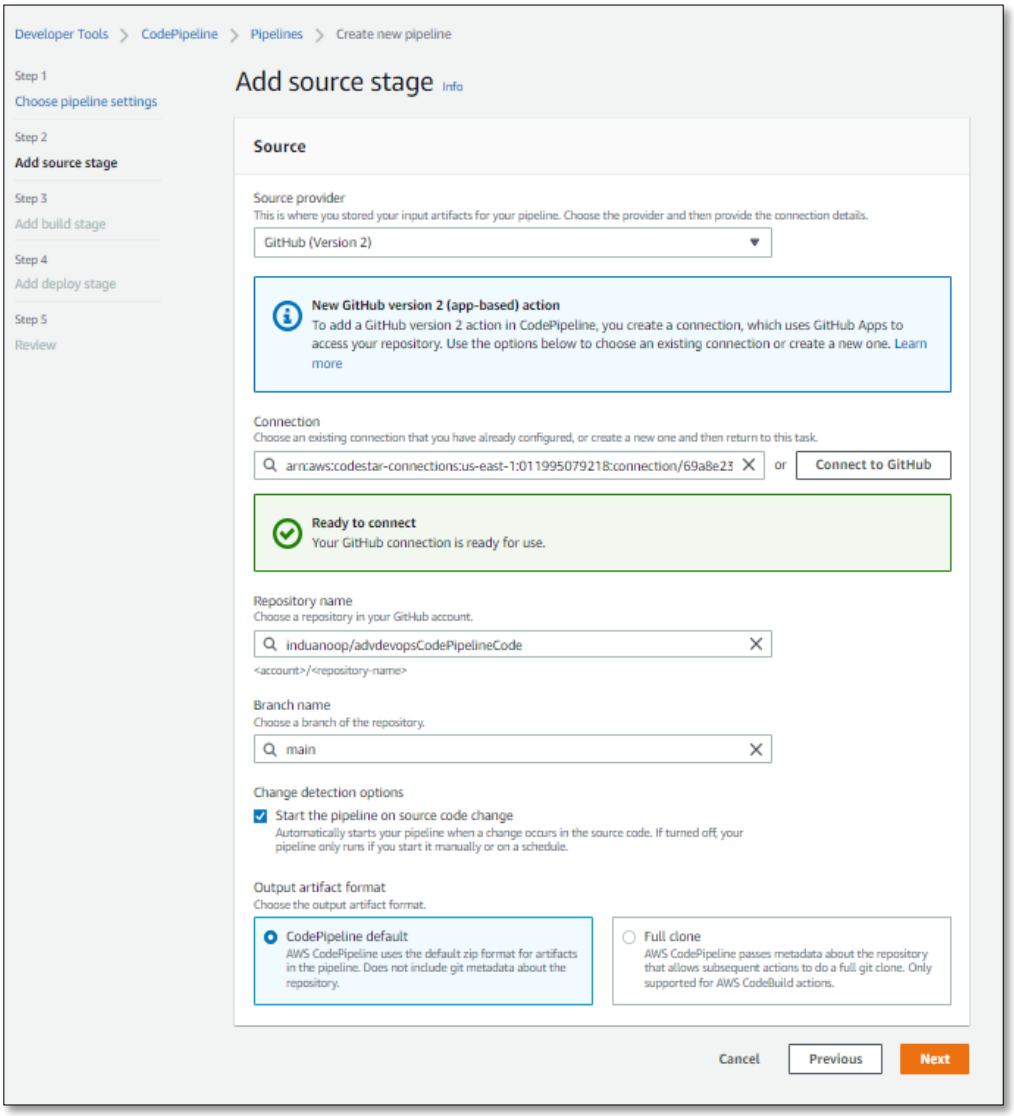
Step3: Create your Pipeline

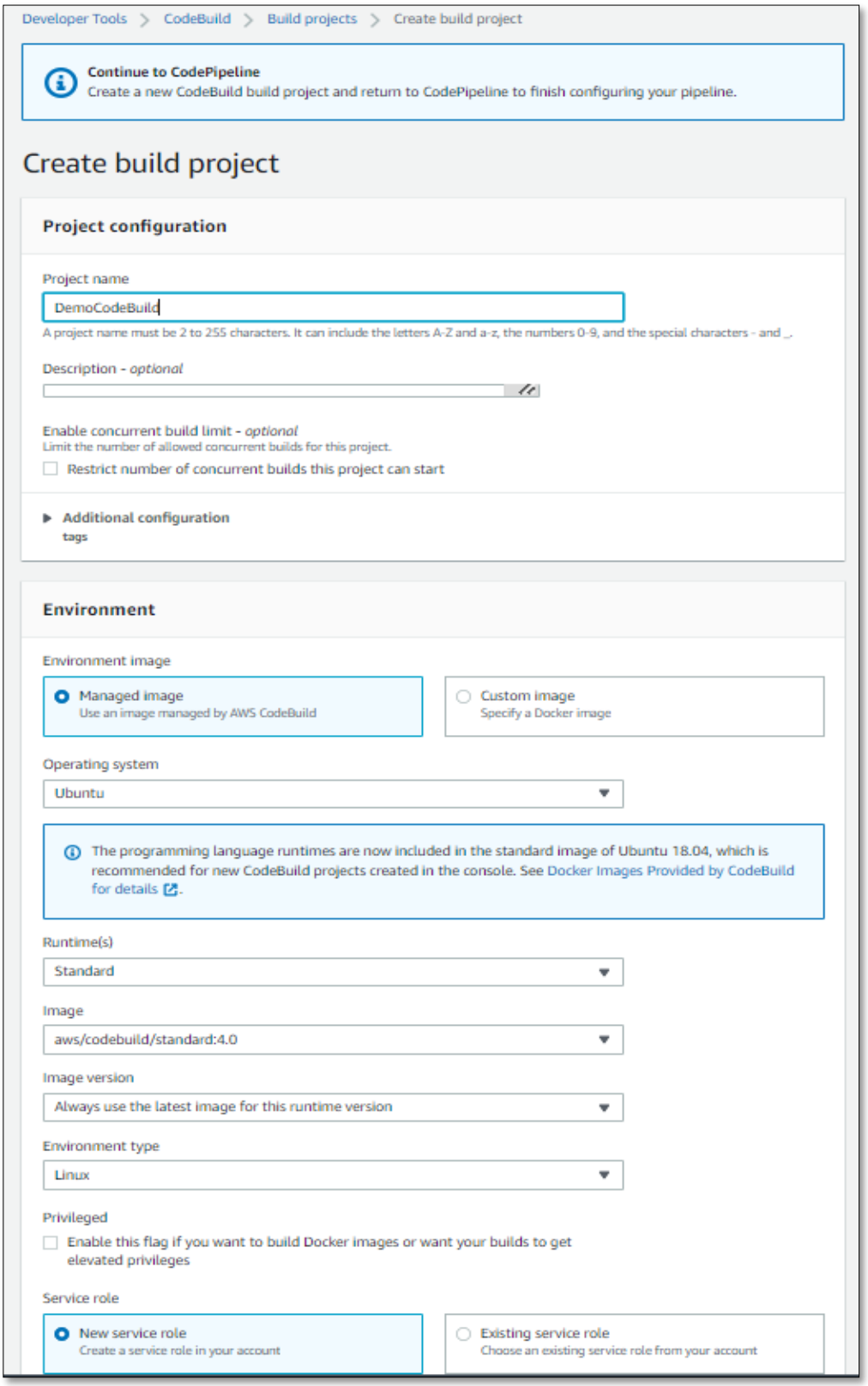
In this step, you will create and configure a CICD pipeline. You will provide CodePipeline with the locations of your source repository [In this case GitHub repository] and deployment environment [AWS Elastic Beanstalk environment created in Step 1]. A true continuous deployment pipeline

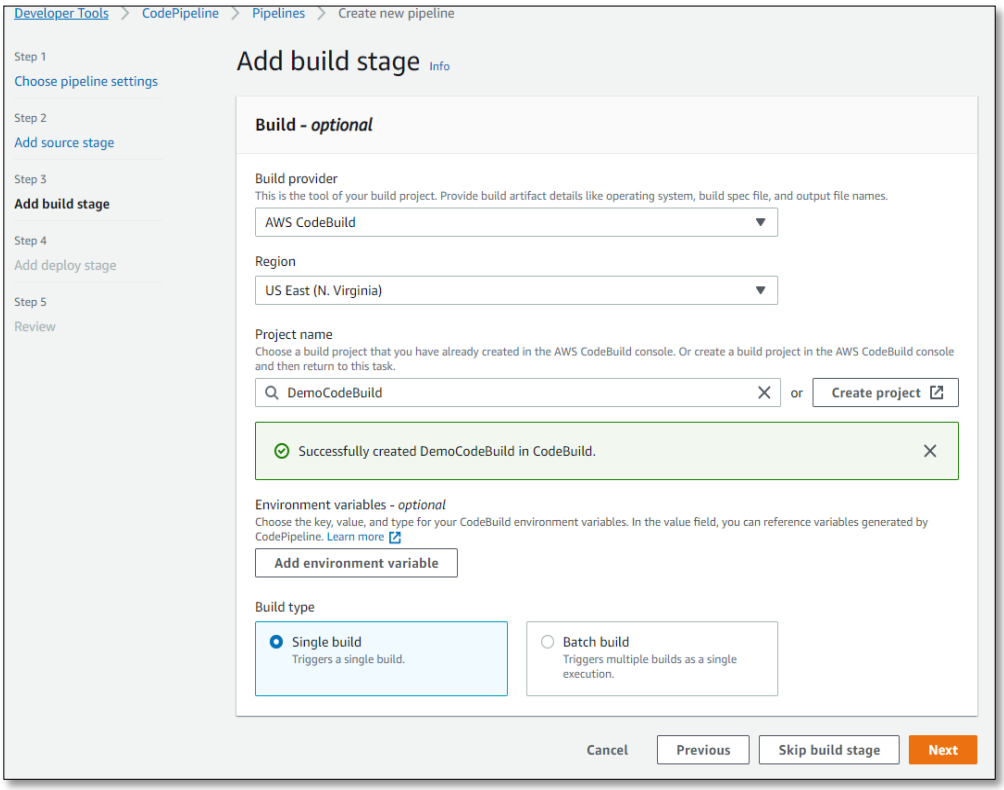
requires a build stage before deployment, where code is compiled, and unit tested. CodePipeline lets you plug your preferred build provider .[although optional, we will be using AWS CodeDeploy] into your pipeline

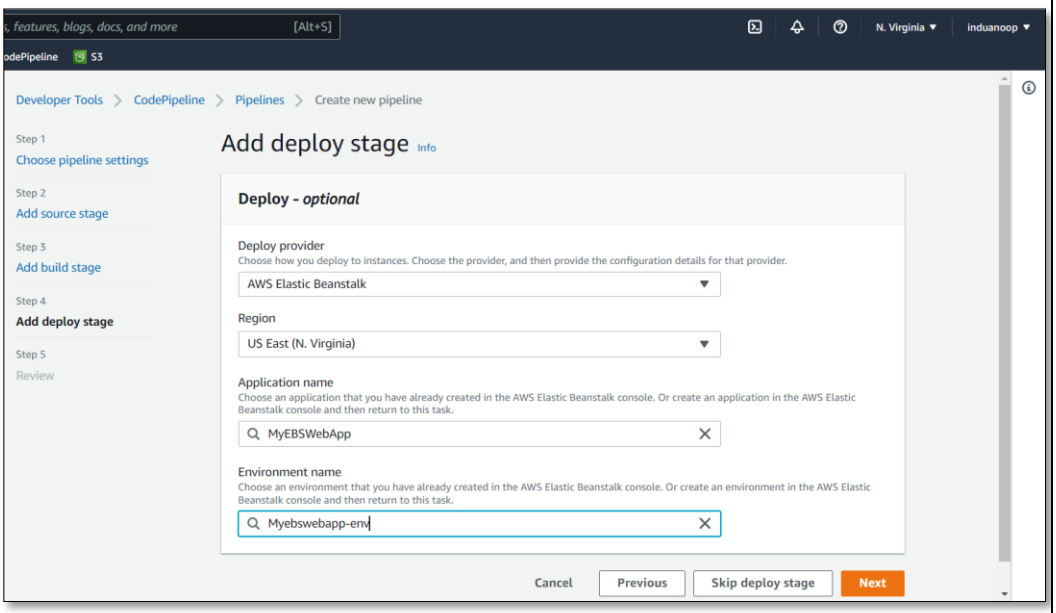


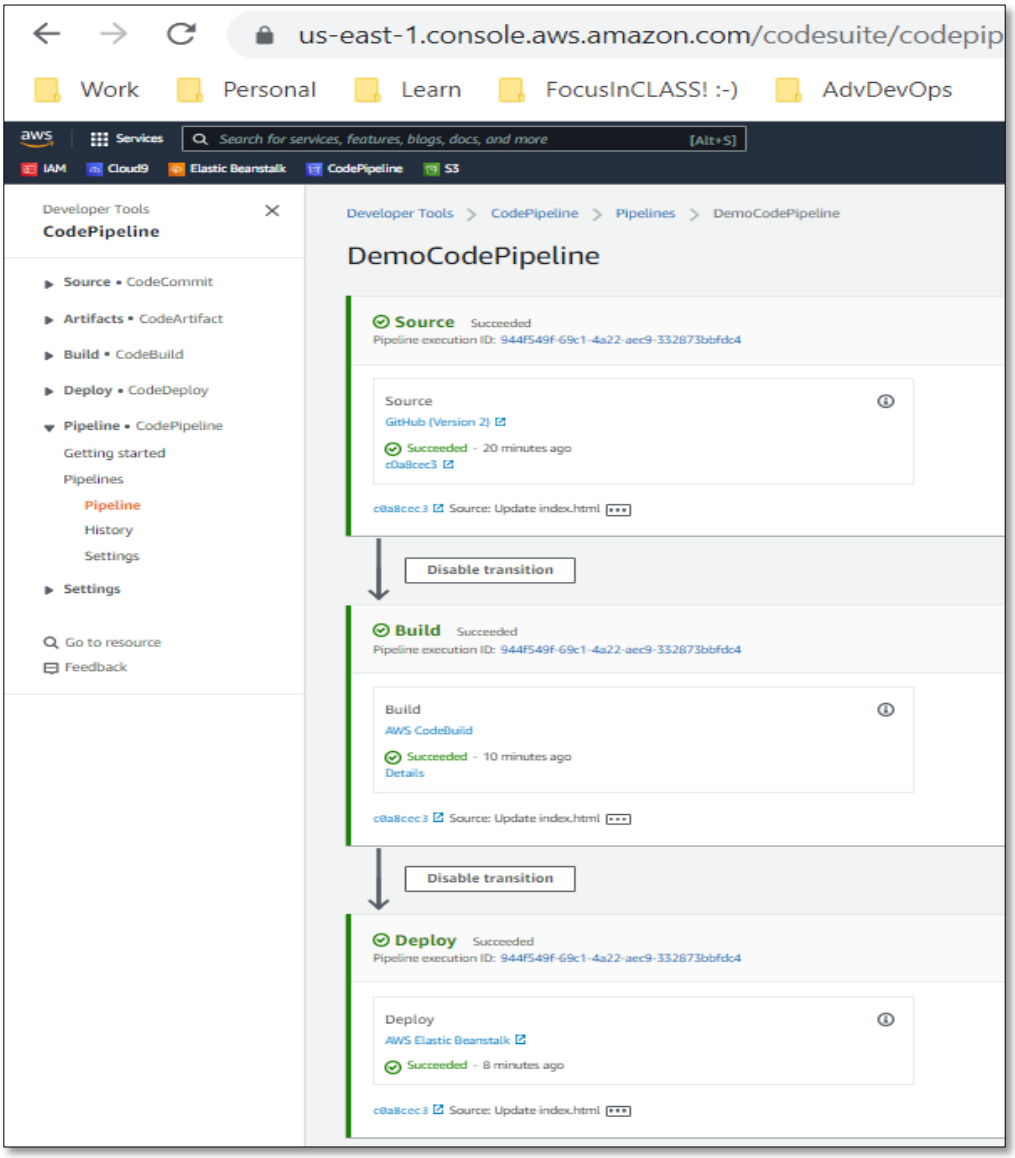


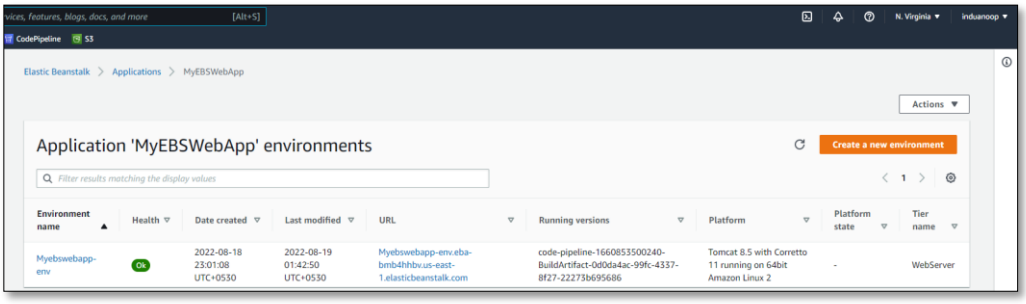

Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add source stage Info
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub (Version 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:us-east-1:011995079218:connection/69a8e23
or
Connect to GitHub
Ready to connect
Your GitHub connection is ready for use.
Repository name
Choose a repository in your GitHub account.
induanoop/advdevopsCodePipelineCode
<account>/<repository-name>
Branch name
Choose a branch of the repository.
main
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.
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 subsequent actions to do a full git clone. Only supported for AWS CodeBuild actions.
Cancel
Previous
Next


Developer Tools > CodeBuild > Build projects > Create build project
Continue to CodePipeline
Create a new CodeBuild build project and return to CodePipeline to finish configuring your pipeline.
Create build project
Project configuration
Project name
DemoCodeBuild
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Description - optional
Enable concurrent build limit - optional
Limit the number of allowed concurrent builds for this project.
Restrict number of concurrent builds this project can start
Additional configuration
tags
Environment
Environment image
Managed image
Use an image managed by AWS CodeBuild
Custom image
Specify a Docker image
Operating system
Ubuntu
The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details.
Runtime(s)
Standard
Image
aws/codebuild/standard:4.0
Image version
Always use the latest image for this runtime version
Environment type
Linux
Privileged
Enable this flag if you want to build Docker images or want your builds to get elevated privileges
Service role
New service role
Create a service role in your account
Existing service role
Choose an existing service role from your account

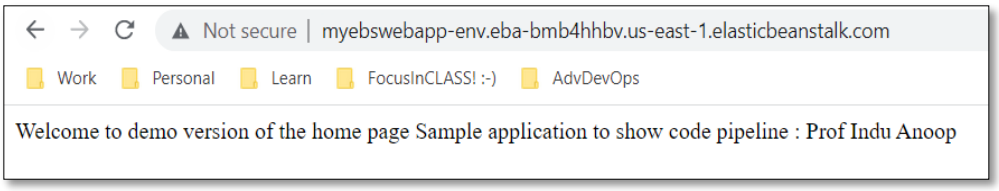

Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add build stage Info
Build - optional
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild
Region
US East (N. Virginia)
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.
DemoCodeBuild
or
Create project
Successfully created DemoCodeBuild in CodeBuild.
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.
Cancel
Previous
Skip build stage
Next


[Alt+S]
N. Virginia
induanoop
CodePipeline
S3
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 deploy stage Info
Deploy - optional
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk
Region
US East (N. Virginia)
Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
MyEBSWebApp
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Myebswebapp-env
Cancel
Previous
Skip deploy stage
Next



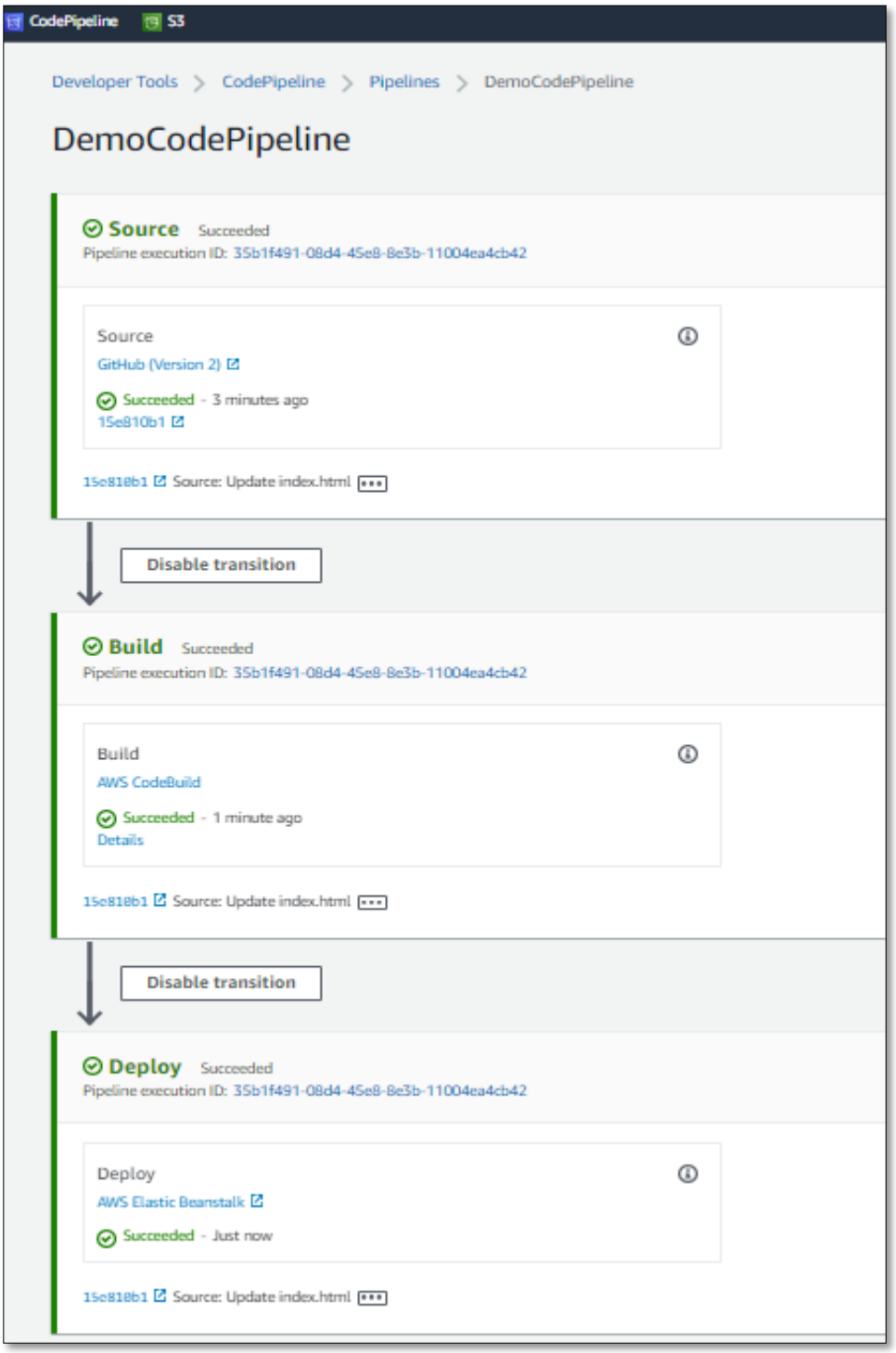


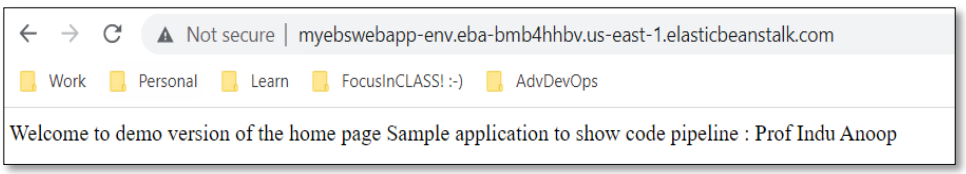
You have successfully created an automated software release pipeline using AWS CodePipeline! Using CodePipeline, you created a pipeline that uses GitHub as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Click on the Environment URL to view application.



Step 5: Commit a change to repository and view update of webpage

In this step, you will revise the sample code and commit the change to your GitHub repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to your EC2 instance via Elastic Beanstalk


CodePipeline
S3
Developer Tools > CodePipeline > Pipelines > DemoCodePipeline
DemoCodePipeline
Source Succeeded
Pipeline execution ID: 35b1f491-08d4-45e8-8e3b-11004ea4cb42
Source
GitHub (Version 2)
Succeeded - 3 minutes ago
15e810b1
15e810b1 Source: Update index.html
Disable transition
Build Succeeded
Pipeline execution ID: 35b1f491-08d4-45e8-8e3b-11004ea4cb42
Build
AWS CodeBuild
Succeeded - 1 minute ago
Details
15e810b1 Source: Update index.html
Disable transition
Deploy Succeeded
Pipeline execution ID: 35b1f491-08d4-45e8-8e3b-11004ea4cb42
Deploy
AWS Elastic Beanstalk
Succeeded - Just now
15e810b1 Source: Update index.html


Not secure | myebswebapp-env.eba-bmb4hhbv.us-east-1.elasticbeanstalk.com
Work
Personal
Learn
FocusInCLASS! :-)
AdvDevOps
Welcome to demo version of the home page Sample application to show code pipeline : Prof Indu Anoop

	Step 6: Clean up your resources To avoid future charges, you will delete all the resources you launched throughout this tutorial, which includes the pipeline, the Elastic Beanstalk application, and the source you set up to host the code. a. First, you will delete your pipeline: • In the pipeline view, click Edit. • Click Delete. • Type in the name of your pipeline and click Delete. b. Second, delete your Elastic Beanstalk application: • Visit the Elastic Beanstalk console. • Click Actions. • Then click Terminate Environment
Conclusion	Successfully created an automated software release pipeline using AWS CodePipeline. Using CodePipeline, you created a pipeline that uses GitHub as the source location for application code and then deployed the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. The pipeline will automatically deploy the code every time there is a code change that is committed to the repository.