Continuous Delivery Pipeline

Continuous Delivery Pipeline on AWS: Continuous Delivery service you can use to model, visualize, and automate the steps required to release your software.

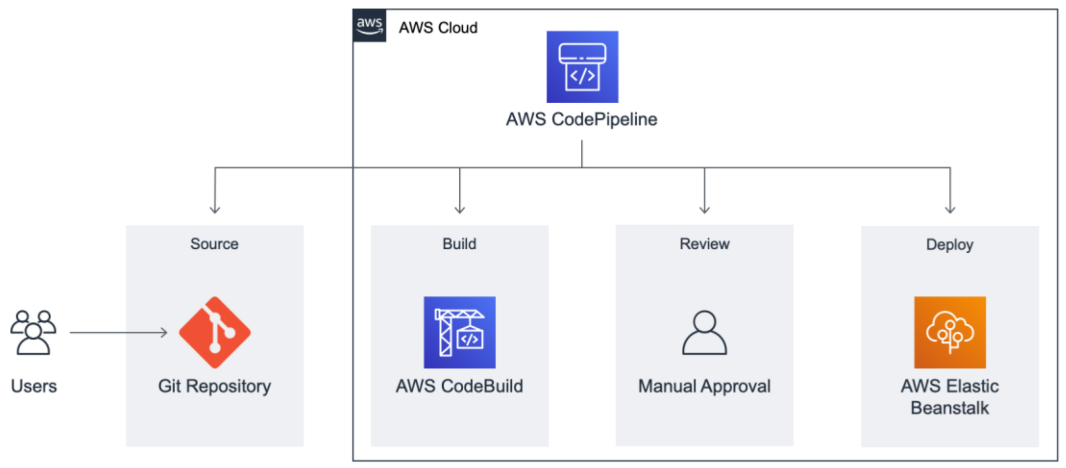
In this demo, GitHub is used for Continuous Integration or as a Version Control system. AWS services such as Elastic Beanstalk, CodeBuild and CodePipeline are used for the Continuous Delivery purpose.

Elastic Beanstalk: AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service, CloudWatch, autoscaling, and Elastic Load Balancers.

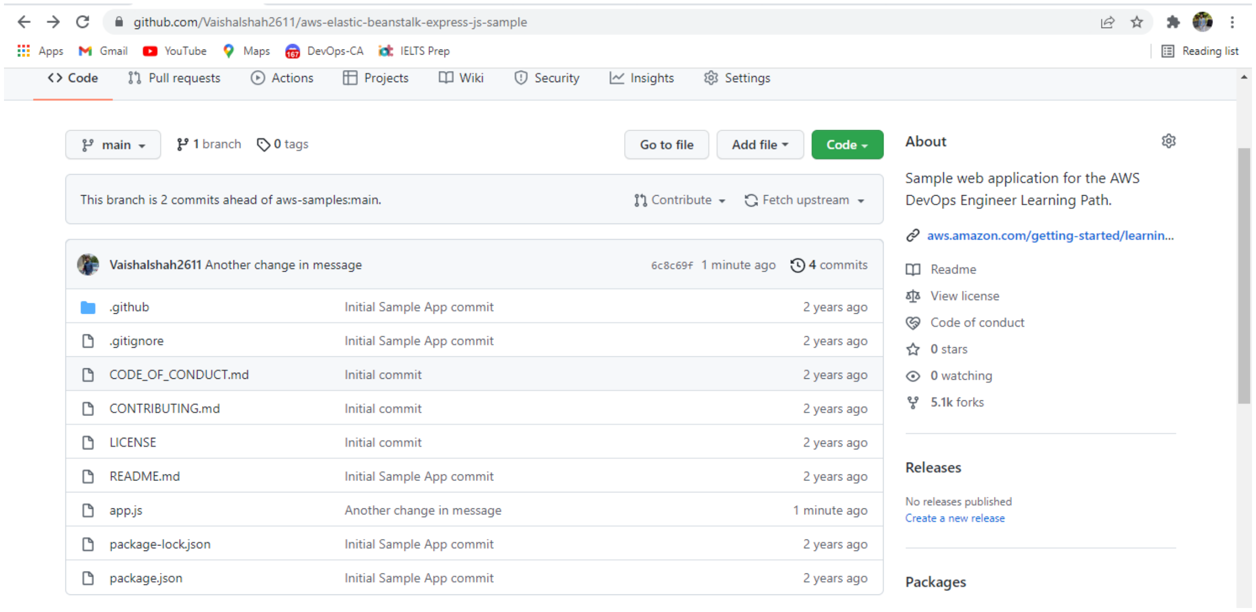
CodeBuild: AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy.

CodePipeline: AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define.

Architectural Diagram



First of all, I configured a repo on GitHub. I will be using node.js application whose source code is available on my GitHub as shown in the image below.



github.com/Vaishalshah2611/aws-elastic-beanstalk-express-js-sample

This branch is 2 commits ahead of aws-samples:main.

Vaishalshah2611 Another change in message 6c8c69f 1 minute ago 4 commits

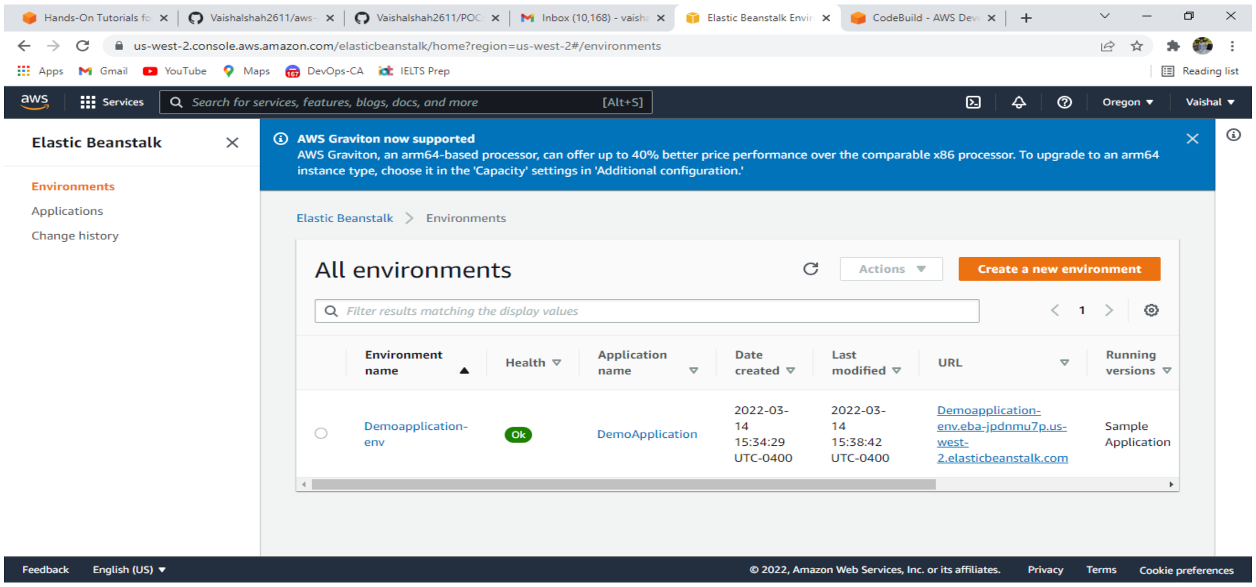
.github	Initial Sample App commit	2 years ago
.gitignore	Initial Sample App commit	2 years ago
CODE_OF_CONDUCT.md	Initial commit	2 years ago
CONTRIBUTING.md	Initial commit	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Initial Sample App commit	2 years ago
app.js	Another change in message	1 minute ago
package-lock.json	Initial Sample App commit	2 years ago
package.json	Initial Sample App commit	2 years ago

About

Sample web application for the AWS DevOps Engineer Learning Path.

aws.amazon.com/getting-started/learnin...

As shown in the image below, I have created an Elastic Beanstalk environment to deploy a Linux server on which the code will be deployed, and the application will be running.



us-west-2.console.aws.amazon.com/elasticbeanstalk/home?region=us-west-2#/environments

AWS Graviton now supported
AWS Graviton, an arm64-based processor, can offer up to 40% better price performance over the comparable x86 processor. To upgrade to an arm64 instance type, choose it in the 'Capacity' settings in 'Additional configuration.'

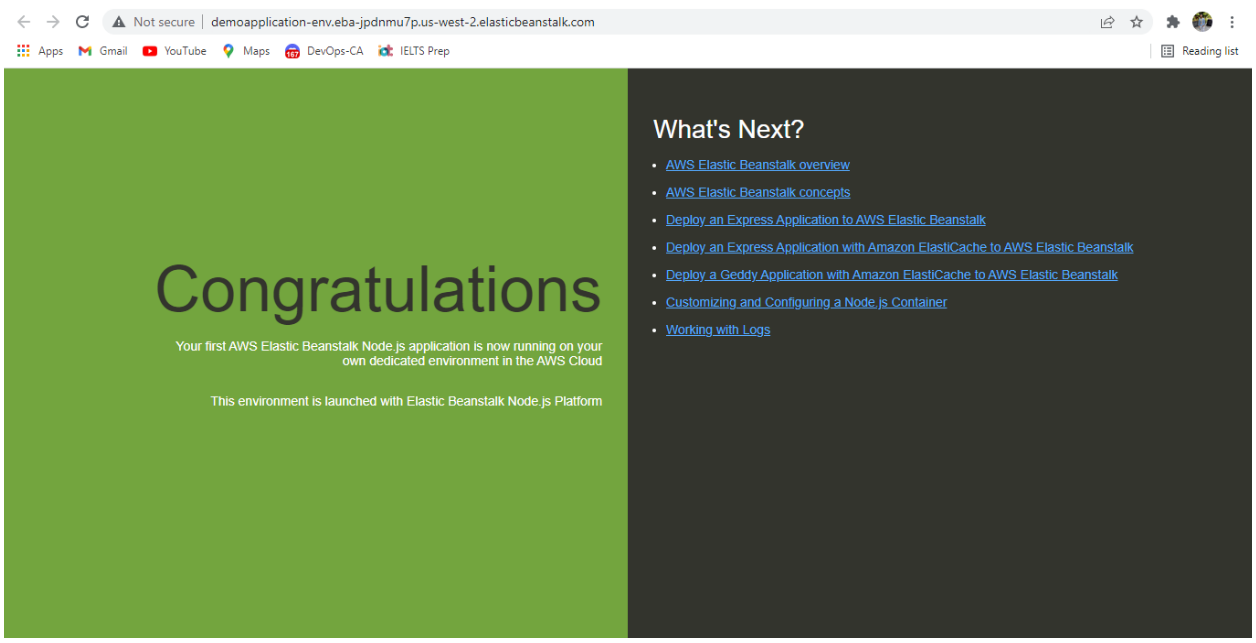
Elastic Beanstalk > Environments

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Running versions
Demoapplication-env	Ok	DemoApplication	2022-03-14 15:34:29 UTC-0400	2022-03-14 15:38:42 UTC-0400	Demoapplication-env.eba-jpdnmu7p.us-west-2.elasticbeanstalk.com	Sample Application

© 2022, Amazon Web Services, Inc. or its affiliates.

After the creation of the Elastic BeanStalk instance, a sample application is run and we can see that by the URL given in the Beanstalk environment. Once we see that URL is working, Bingo. Beanstalk is deployed successfully.



Not secure | demoapplication-env.eba-jpdnmu7p.us-west-2.elasticbeanstalk.com

Congratulations

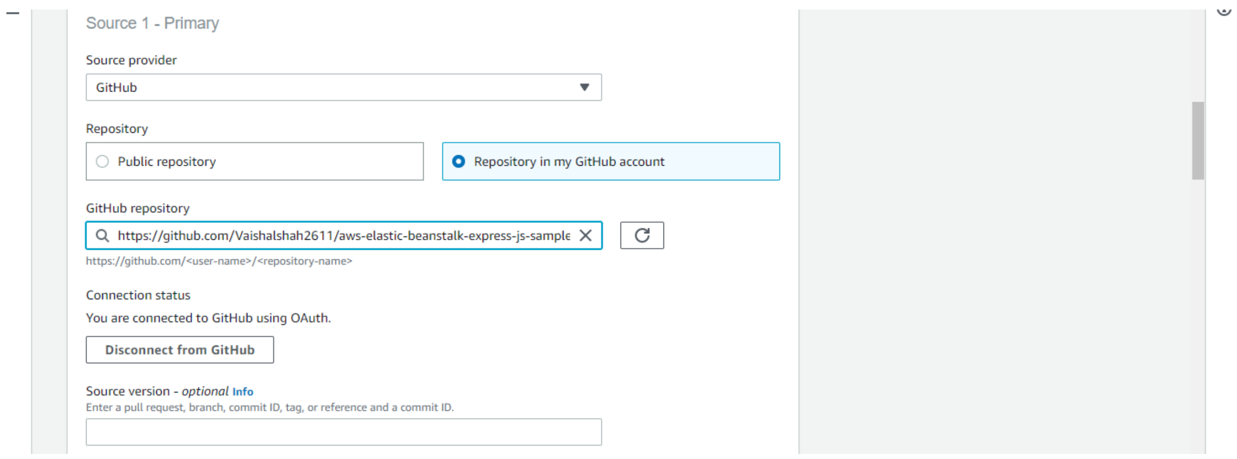
Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy an Express Application to AWS Elastic Beanstalk
- Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Deploy a Geddy Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

Once the Beanstalk is ready. We need to configure the AWS CodeBuild to connect with our GitHub. Once the connection is made successfully, we can try a build and run it successfully to deploy code into our BeanStalk server. As shown in the images below.



Source 1 - Primary

Source provider: GitHub

Repository: Public repository [✓ Repository in my GitHub account]

GitHub repository: https://github.com/Vaishalshah2611/aws-elastic-beanstalk-express-js-sample

https://github.com/<user-name>/<repository-name>

Connection status

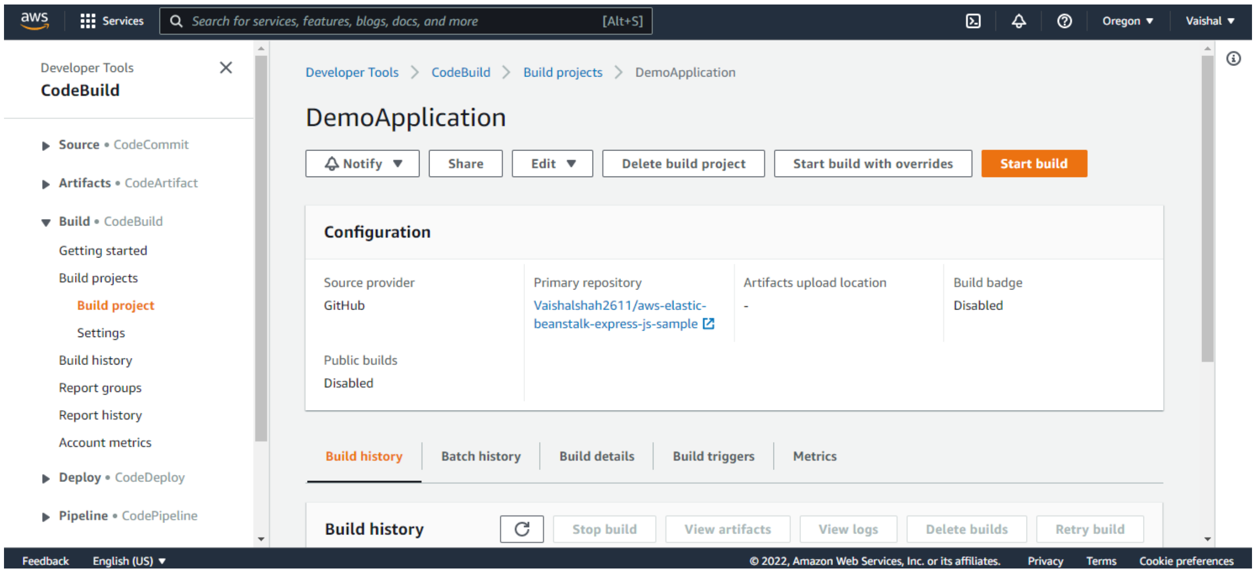
You are connected to GitHub using OAuth.

Disconnect from GitHub

Source version - *optional* Info

Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

Image: GitHub repository is connected with AWS CodeBuild



DemoApplication

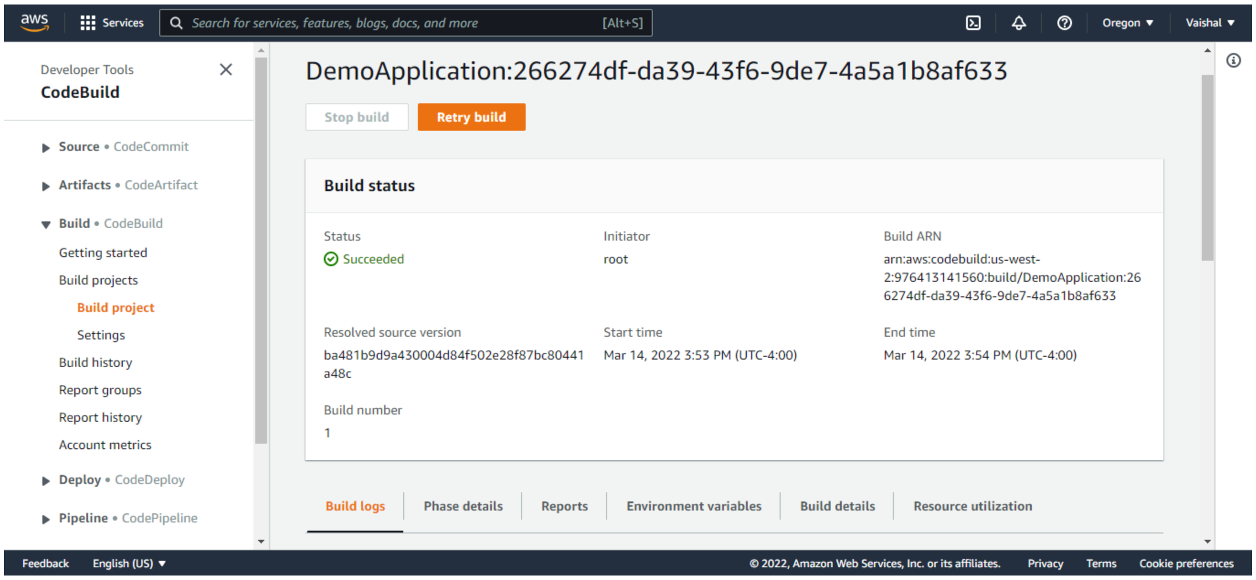
Configuration

Source provider	Primary repository	Artifacts upload location	Build badge
GitHub	Vaishalshah2611/aws-elastic-beanstalk-express-js-sample	-	Disabled

Public builds
Disabled

Build history

After successfully integration, click on the start build to start building the configured repo.



DemoApplication:266274df-da39-43f6-9de7-4a5a1b8af633

Build status

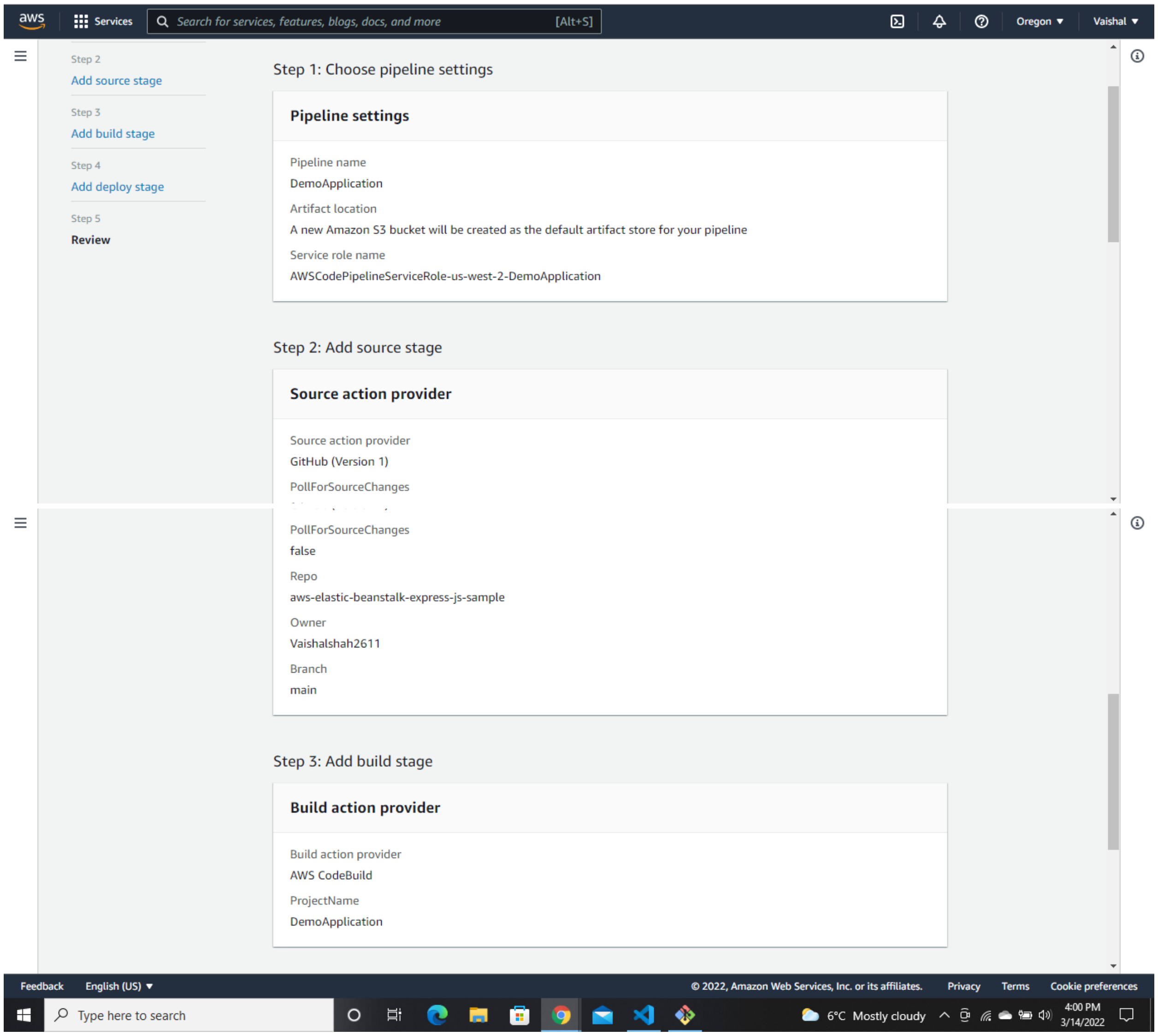
Status	Initiator	Build ARN
✓ Succeeded	root	arn:aws:codebuild:us-west-2:976413141560:build/DemoApplication:266274df-da39-43f6-9de7-4a5a1b8af633

Resolved source version	Start time	End time
ba481b9d9a430004d84f502e28f87bc80441a48c	Mar 14, 2022 3:53 PM (UTC-4:00)	Mar 14, 2022 3:54 PM (UTC-4:00)

Build number
1

As shown in the image above, Build gets successful and hence everything is working successfully in the environment setup till now. And code gets deployed on the BeanStalk instance.

Now, it's time to configure the CodePipeline. From the AWS Console, I have created the pipeline with the simple steps as shown in images below.



Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
DemoApplication

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-us-west-2-DemoApplication

Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 1)

PollForSourceChanges
false

Repo
aws-elastic-beanstalk-express-js-sample

Owner
Vaishalshah2611

Branch
main

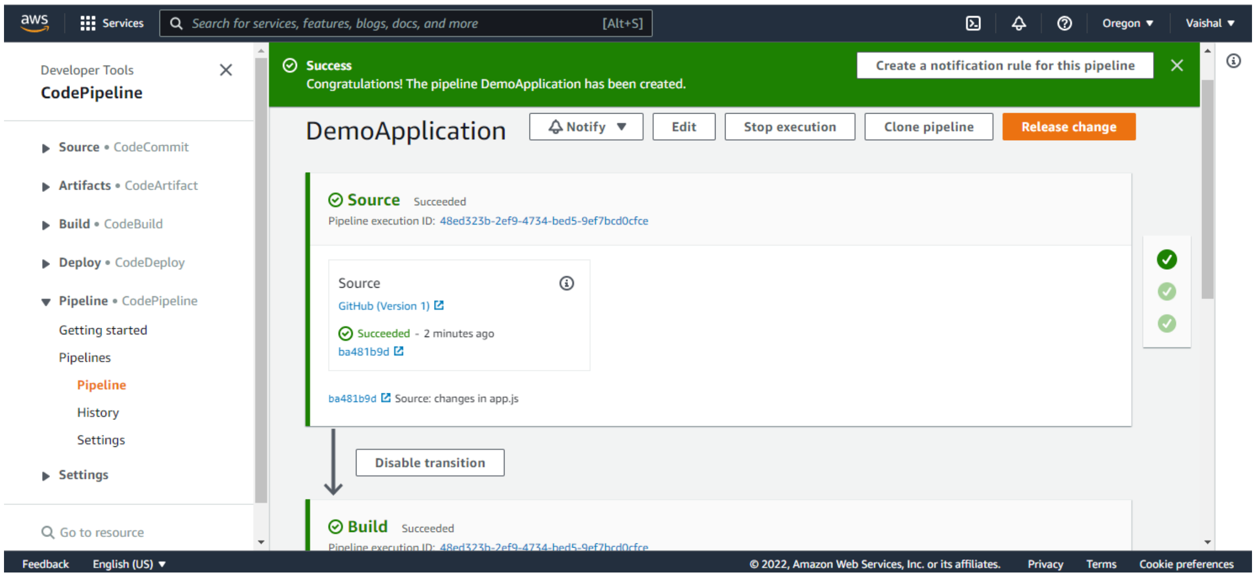
Step 3: Add build stage

Build action provider

Build action provider
AWS CodeBuild

ProjectName
DemoApplication

After successfully, it can be shown from the console that it is created successfully and can view the stages of the pipeline.
Commit the code with small changes to the GitHub to see CodePipeline in action.



Success
Congratulations! The pipeline DemoApplication has been created.

DemoApplication

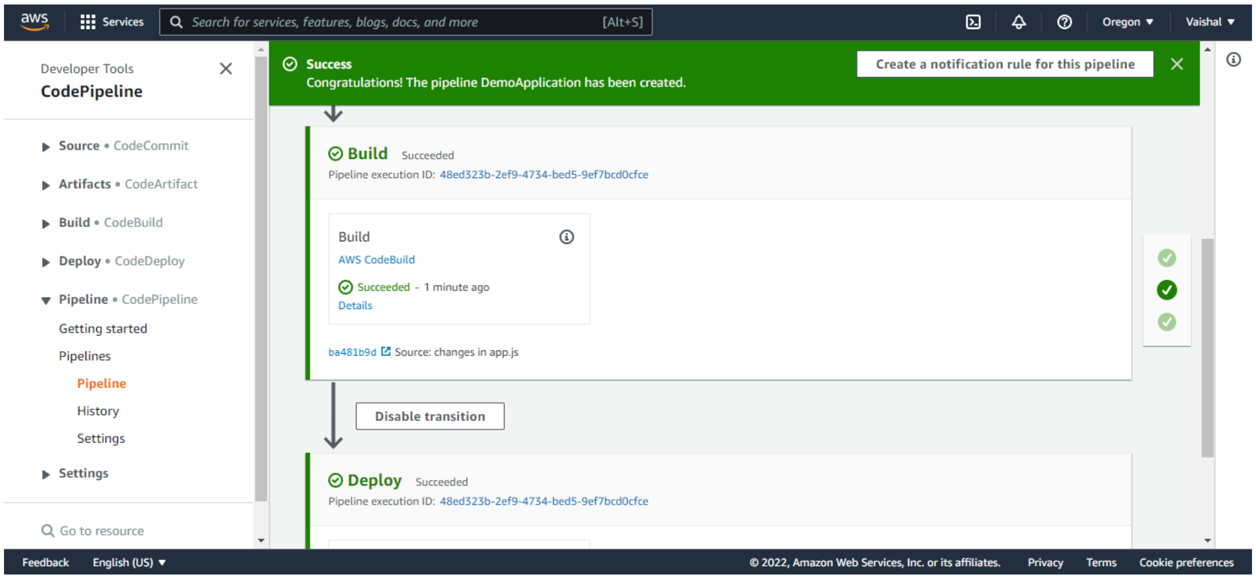
Source Succeeded
Pipeline execution ID: 48ed323b-2ef9-4734-bed5-9ef7bcd0cfce

Source
GitHub (Version 1)
Succeeded - 2 minutes ago
ba481b9d

ba481b9d Source: changes in app.js

Build Succeeded
Pipeline execution ID: 48ed323b-2ef9-4734-bed5-9ef7bcd0cfce

© 2022, Amazon Web Services, Inc. or its affiliates.



Success
Congratulations! The pipeline DemoApplication has been created.

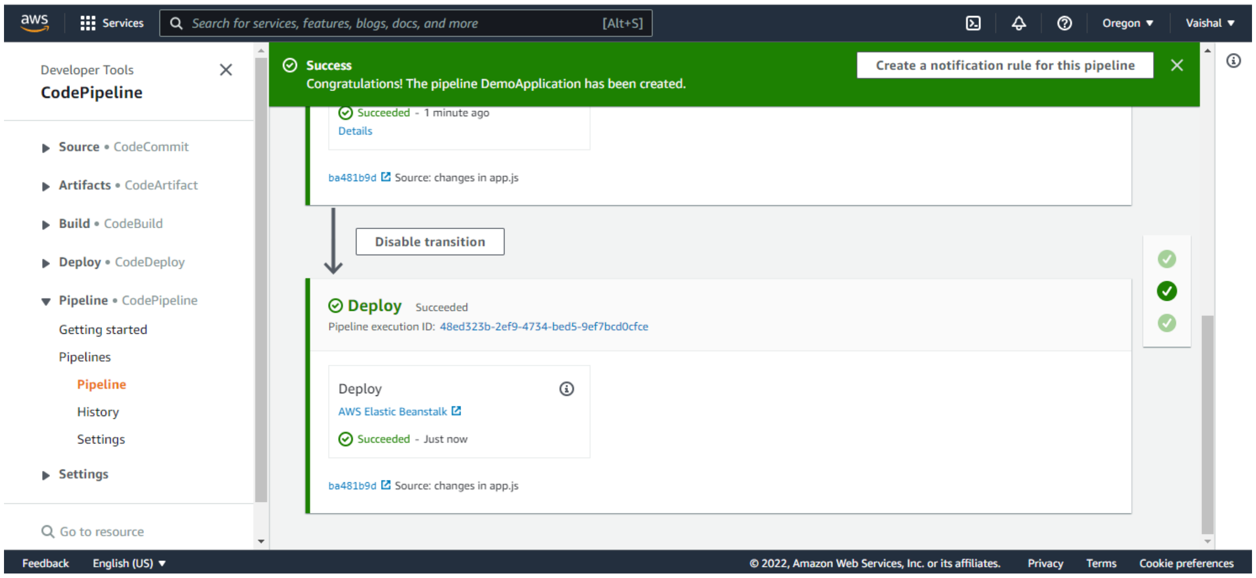
Build Succeeded
Pipeline execution ID: 48ed323b-2ef9-4734-bed5-9ef7bcd0cfce

Build
AWS CodeBuild
Succeeded - 1 minute ago
Details

ba481b9d Source: changes in app.js

Deploy Succeeded
Pipeline execution ID: 48ed323b-2ef9-4734-bed5-9ef7bcd0cfce

© 2022, Amazon Web Services, Inc. or its affiliates.



Success
Congratulations! The pipeline DemoApplication has been created.

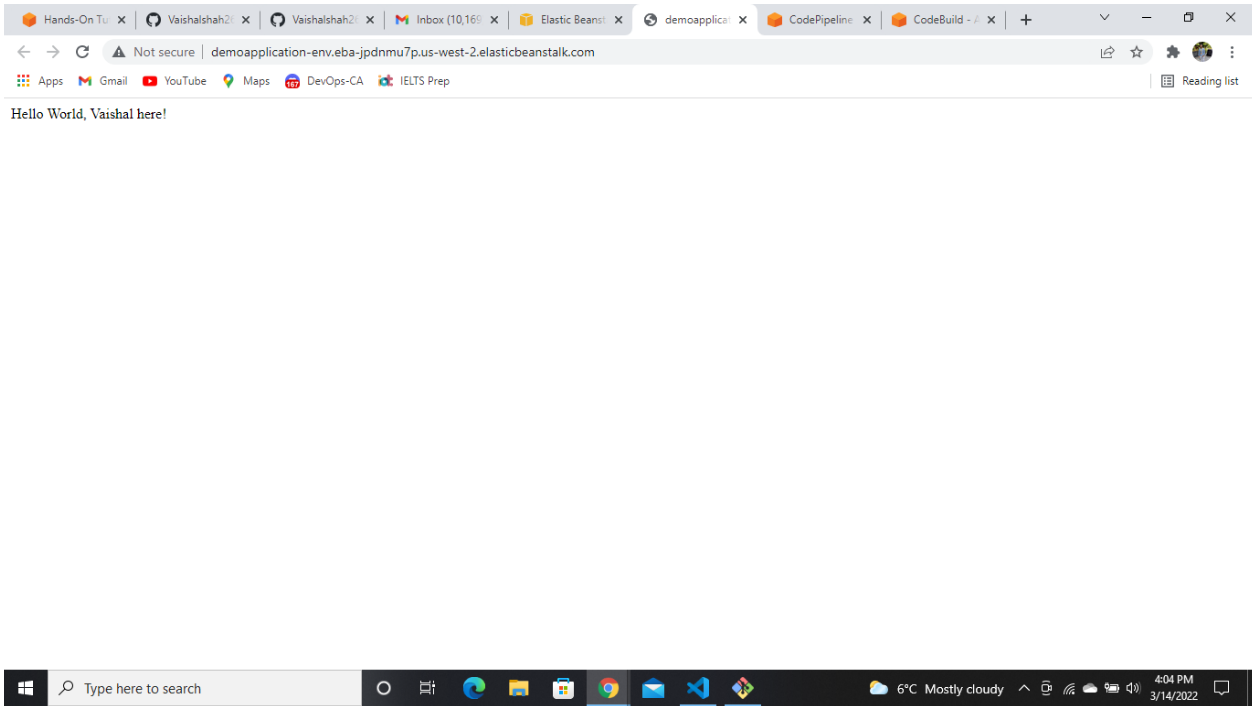
Succeeded - 1 minute ago
Details

ba481b9d Source: changes in app.js

Deploy Succeeded
Pipeline execution ID: 48ed323b-2ef9-4734-bed5-9ef7bcd0cfce

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

ba481b9d Source: changes in app.js

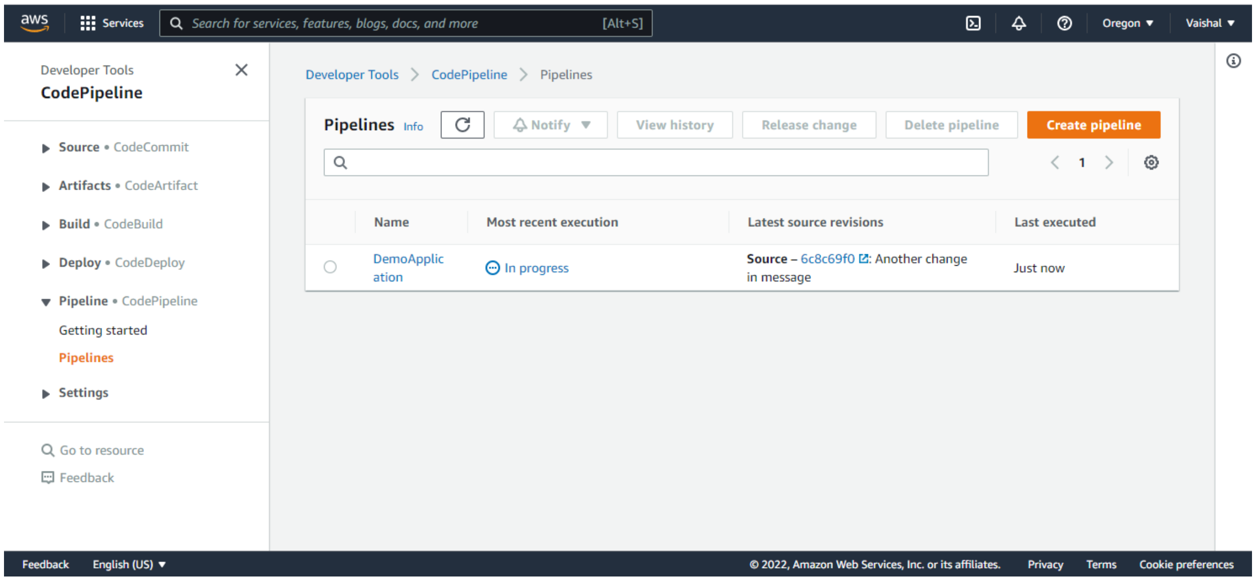


demoapplication-env.eba-jpdnmu7p.us-west-2.elasticbeanstalk.com

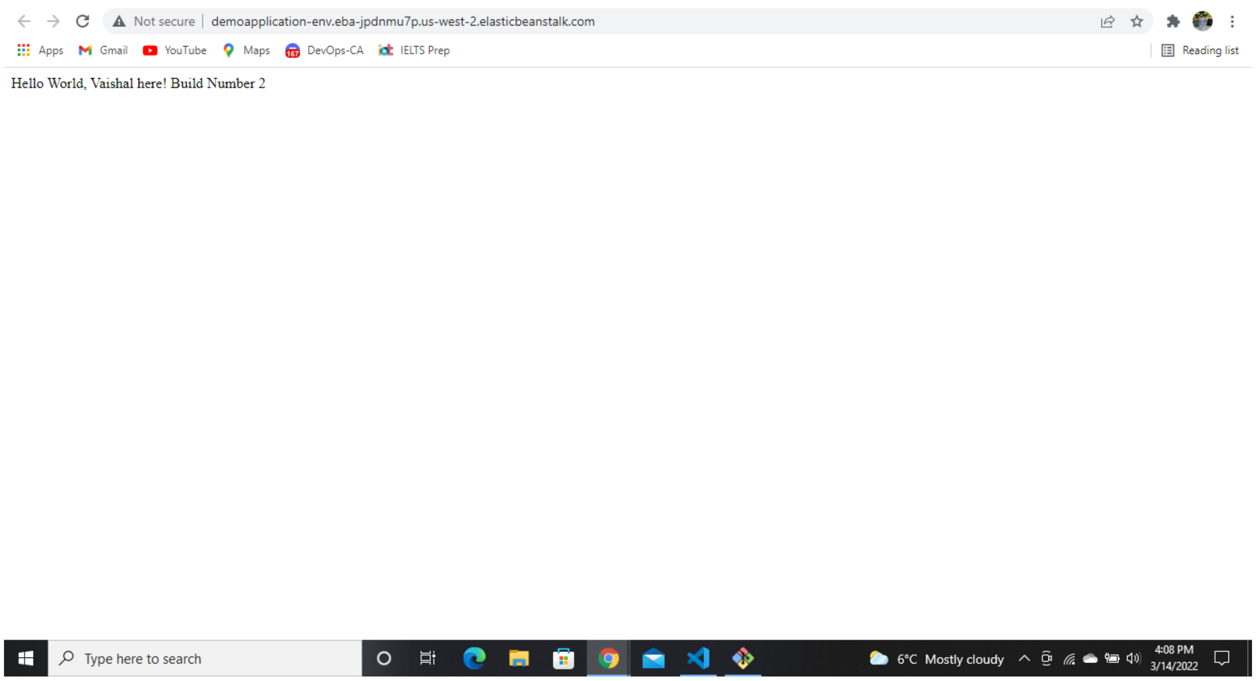
Hello World, Vaishal here!

As we know, we have integrated GitHub → CodePipeline → CodeBuild → Beanstalk. Once we make a commit to the GitHub repo, it will trigger the entire cycle and code will be deployed to the BeanStalk instance.

As shown in the images below, application deployment is triggered automatically and is deployed into BeanStalk.

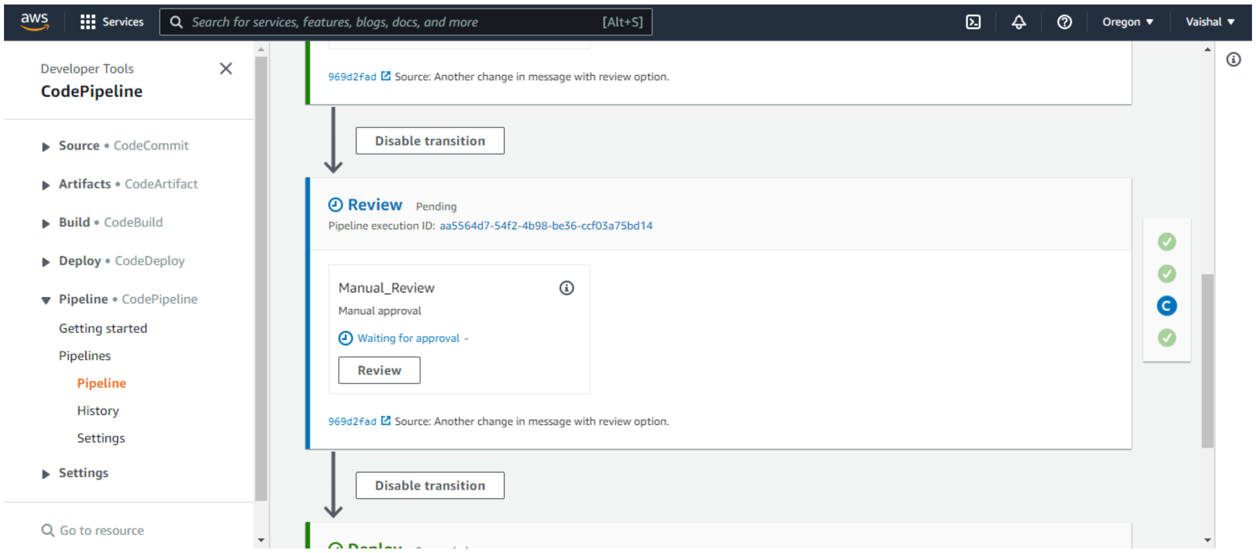


Name	Most recent execution	Latest source revisions	Last executed
DemoApplication	In progress	Source – 6c8c69f0: Another change in message	Just now



Hello World, Vaishal here! Build Number 2

As shown in the images below, we can add a new stage in between for the manual approval for the deployment to be done on BeanStalk. Here, a new stage between the Build and deploy stage is done in order to review the change and then approve it.



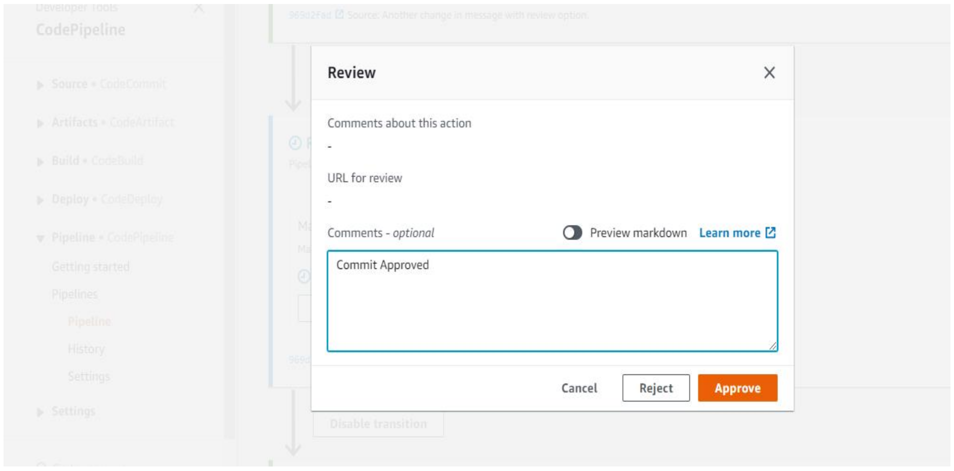
969d2fad Source: Another change in message with review option.

Review Pending
Pipeline execution ID: aa5564d7-54f2-4b98-be36-ccf03a75bd14

Waiting for approval -

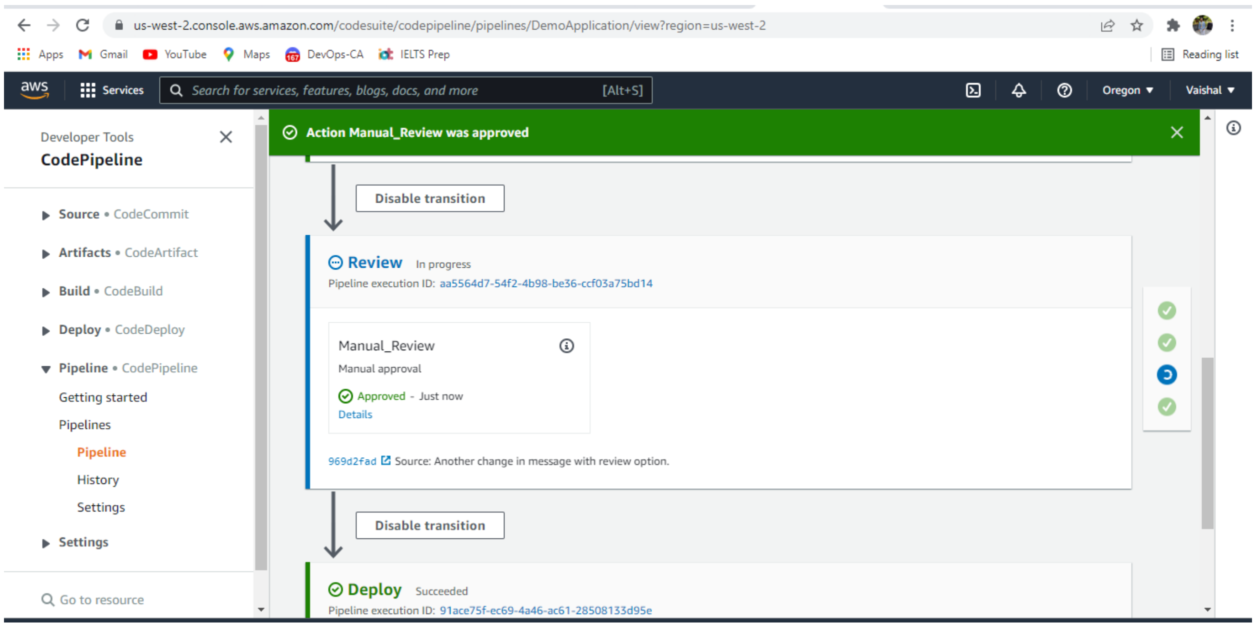
969d2fad Source: Another change in message with review option.

After completion of the build stage, Review gets triggered, and it waits for the approval from the designated person. Once you click on the review and then give the approval of the build then only it gets to the next stage and gets deployed.



Comments about this action
-
URL for review
-
Comments - *optional*
Commit Approved

After getting the approval from the previous stage, it moves to the next stage that is deployed and gets deployed into the instance.



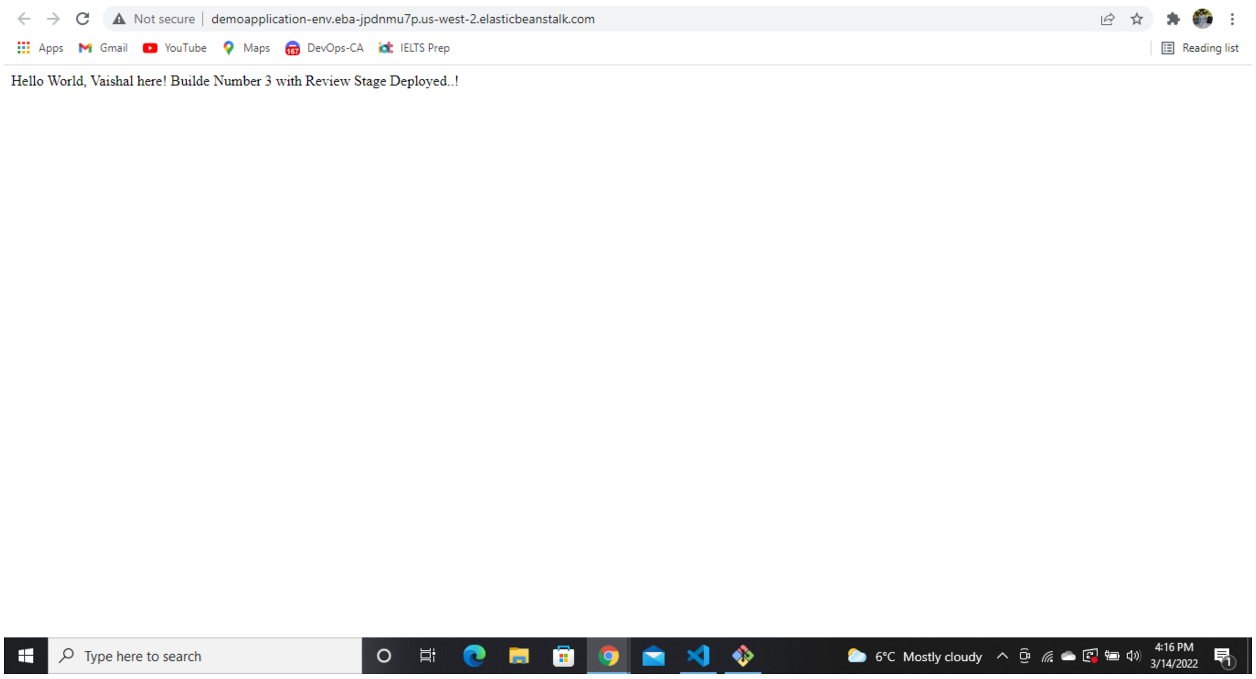
Action Manual_Review was approved

Review In progress
Pipeline execution ID: aa5564d7-54f2-4b98-be36-ccf03a75bd14

Manual_Review
Manual approval
Approved - Just now
Details

969d2fad Source: Another change in message with review option.

Deploy Succeeded
Pipeline execution ID: 91ace75f-ec69-4a46-ac61-28508133d95e


Not secure | demoapplication-env.eba-jpdnmu7p.us-west-2.elasticbeanstalk.com
Apps Gmail YouTube Maps DevOps-CA IELTS Prep Reading list
Type here to search
6°C Mostly cloudy
4:16 PM
3/14/2022

Hello World, Vaishal here! Builde Number 3 with Review Stage Deployed..!

As shown in the image above, a new commit to the code has been deployed into the running environment successfully.
Concept of Continuous Delivery has been successfully depicted.

Thank you,
Vaishal Shah