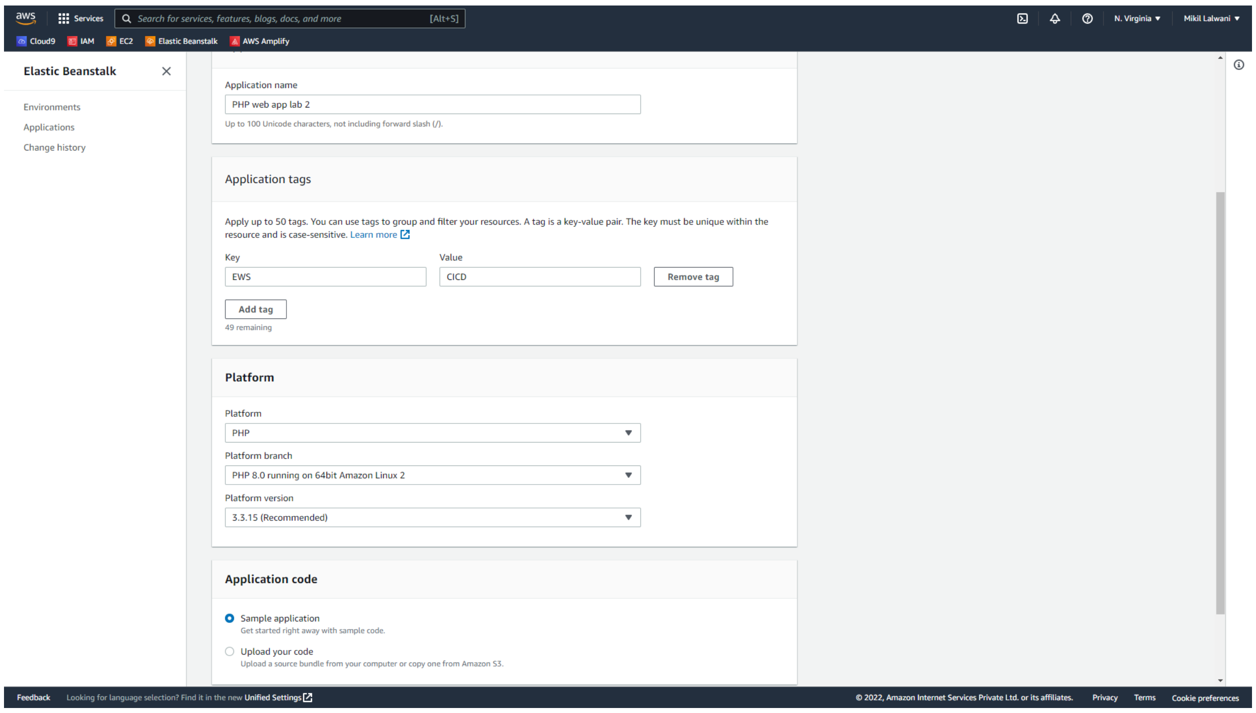
Step 1: Create a deployment environment

1) 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. 1) To simplify the process of setting up and configuring EC2 instances for this tutorial, 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.

2) Name your web app and choose PHP from the drop-down menu(or any other language you are interested in) and then click Create Application.



Application name

PHP web app lab 2

Up to 100 Unicode characters, not including forward slash (/).

Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

Key	Value
EWS	CICD

49 remaining

Platform

Platform: PHP

Platform branch: PHP 8.0 running on 64bit Amazon Linux 2

Platform version: 3.3.15 (Recommended)

Application code

[✓ Sample application] Get started right away with sample code.

Upload your code — Upload a source bundle from your computer or copy one from Amazon S3.

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

3) Elastic Beanstalk will begin creating a sample environment for you to deploy your application. 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

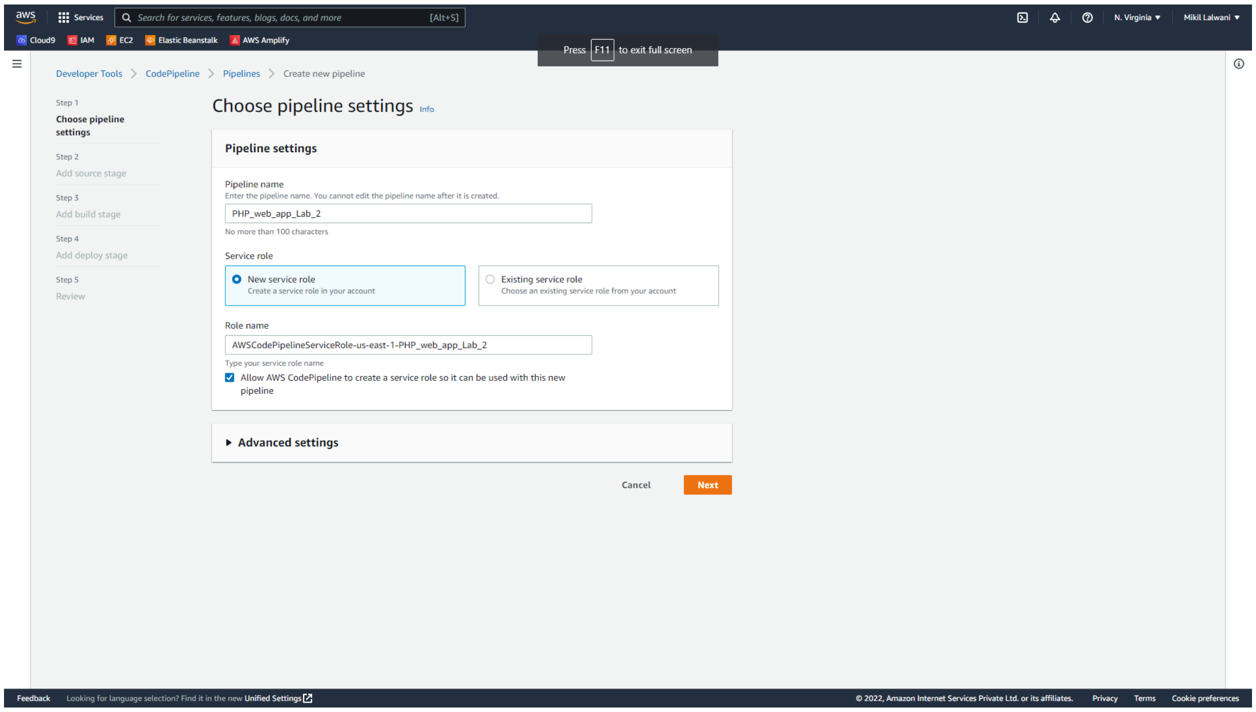
In this step, you will retrieve a copy of the sample app's code and choose a source to host the code. The pipeline takes code from the source and then performs actions on it. You can use one of three options as your source: a GitHub repository, an Amazon S3 bucket, or an AWS CodeCommit repository. Select your preference and follow the steps below:

Fork the repository given below to your Github account.
Github repository: aws-codepipeline-s3-codedeploy-linux

Step 3: Create your Pipeline

1) In this step, you will create and configure a simple pipeline with two actions: source and deploy. You will provide CodePipeline with the locations of your source repository and deployment environment. A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this, we will skip the build stage. Goto Pipeline again

and create it.



Choose pipeline settings Info

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
PHP_web_app_Lab_2
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

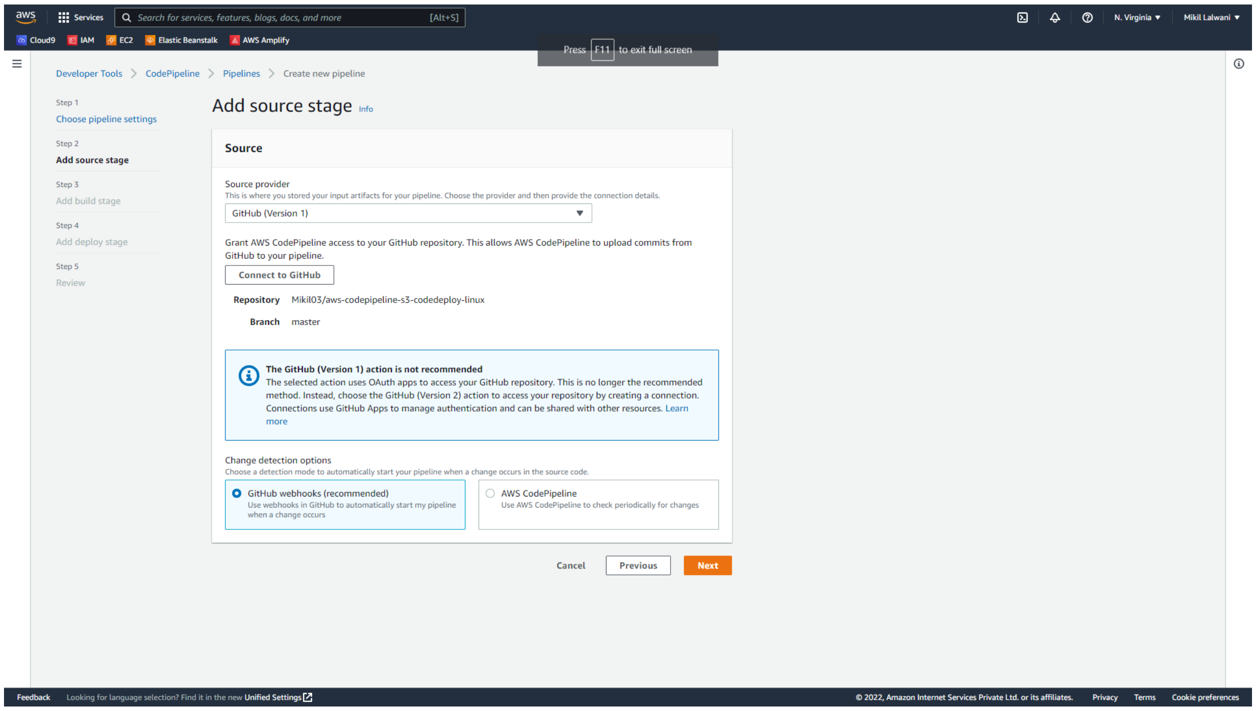
Role name
AWSCodePipelineServiceRole-us-east-1-PHP_web_app_Lab_2
Type your service role name
✓ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Advanced settings

Cancel Next

Step 4: Source Stage:

1) Select the source provider as GitHub (Version 1).
2) Click on Connect to GitHub and provide AWS access to your GitHub account.
3) Select the appropriate repository from the dropdown menu.
4) Select the branch as master.
5) Select GitHub webhooks to update the pipeline as soon as changes are detected.
6) Click on next.


Add source stage

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.

Connect to GitHub

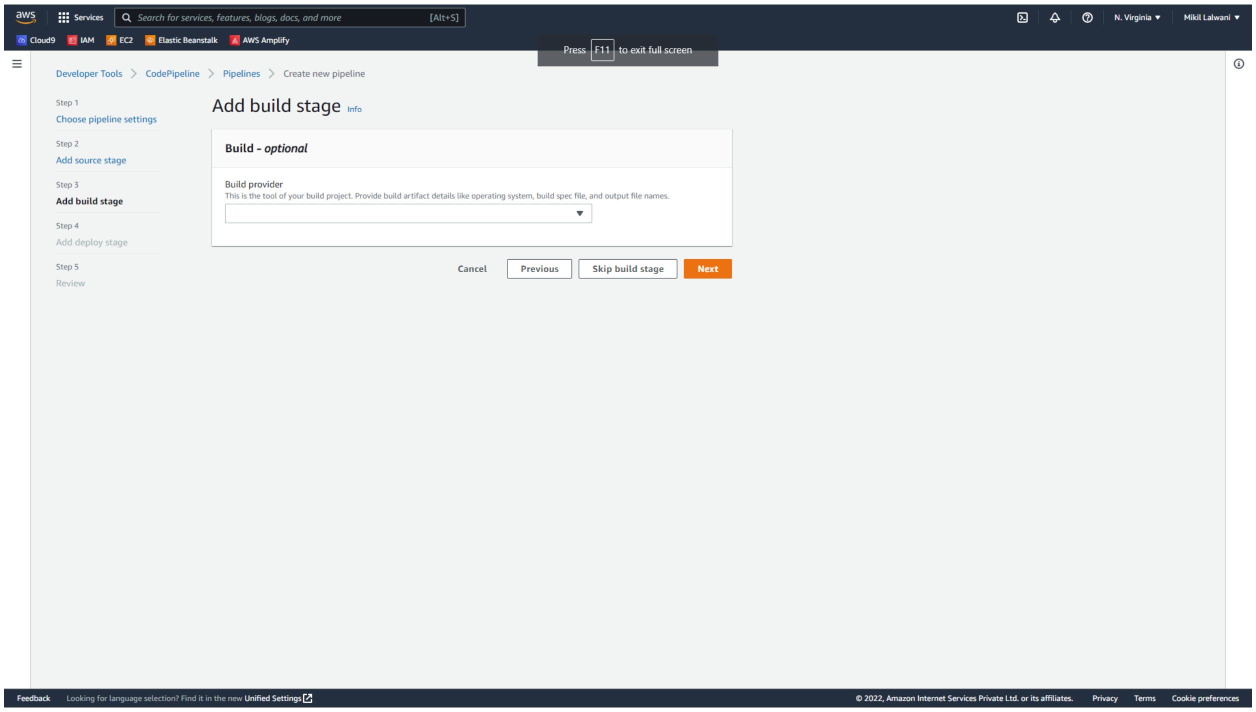
Repository	Mikil03/aws-codepipeline-s3-codedeploy-linux
Branch	master

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

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
[✓ GitHub webhooks (recommended)] Use webhooks in GitHub to automatically start my pipeline when a change occurs
AWS CodePipeline — Use AWS CodePipeline to check periodically for changes

Step 5: Build Stage:

1) Click on Skip build stage.

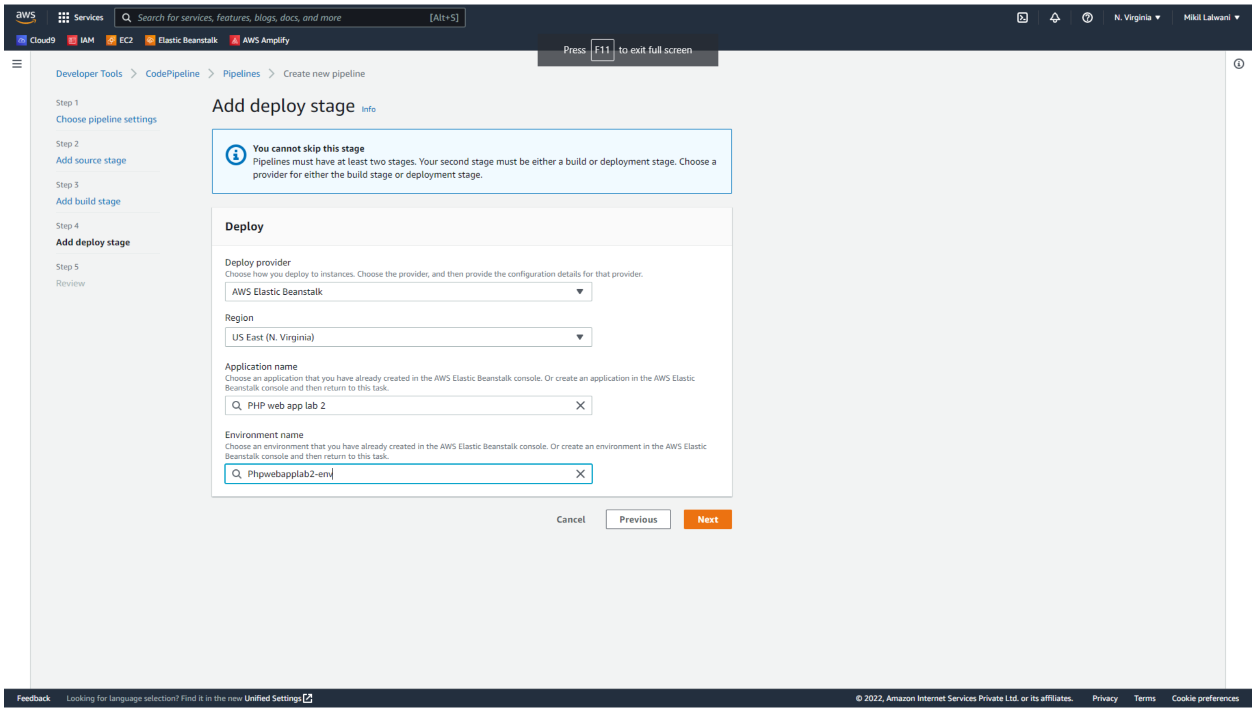

Add build stage

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.

Step 6: Deploy Stage:

1) Deployment provider: Click AWS Elastic Beanstalk. Application name: MYEBS. Environment name: Click Myebs-env. Click Next step.



Add deploy stage

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.
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.
PHP web app lab 2

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.
Phpwebapplab2-env

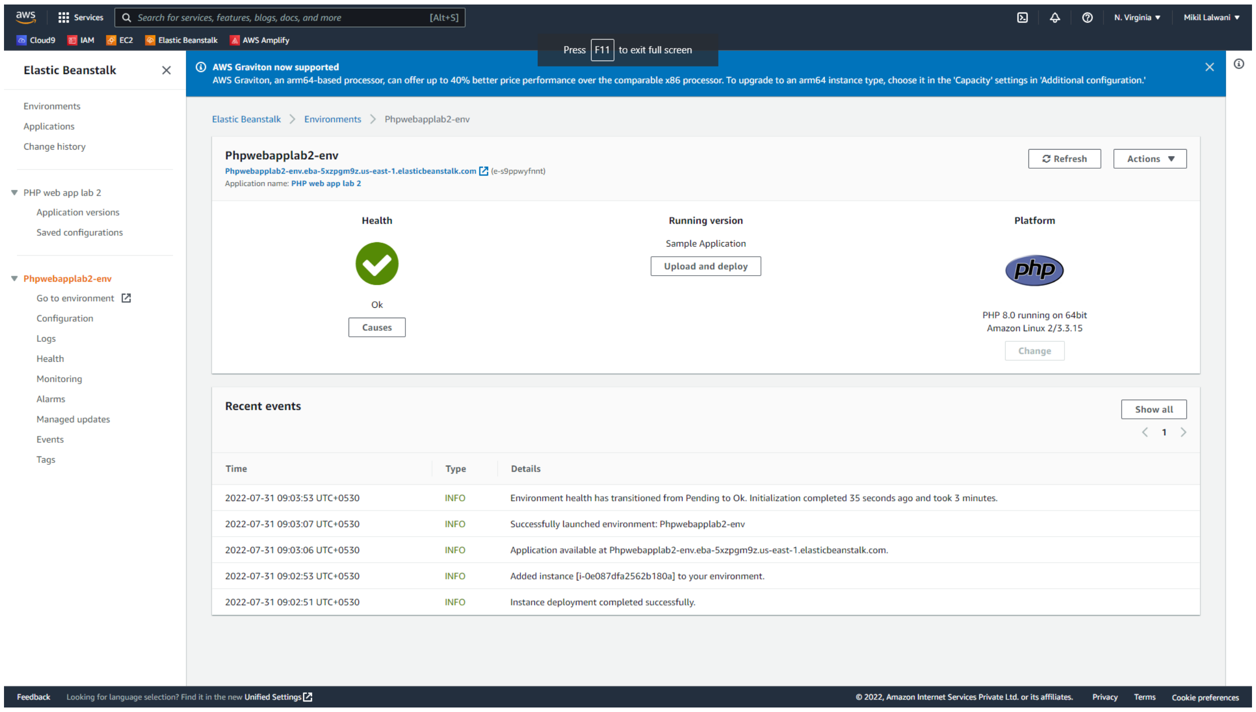
Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

2) After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline perform each action.
3) To verify your pipeline ran successfully, monitor the progress of the pipeline as it moves through each stage. The status of each stage will change from No executions yet to In Progress, and then to either Succeeded or Failed. The pipeline should complete the first run within a few minutes.

Now go to your EBS environment and click on the URL to view the sample website you deployed.

You have successfully created an automated software release pipeline using AWS CodePipeline!



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 > Phpwebapplab2-env

Phpwebapplab2-env
Phpwebapplab2-env.eba-5xzpgm9z.us-east-1.elasticbeanstalk.com (e-s9ppwyfnnt)
Application name: PHP web app lab 2

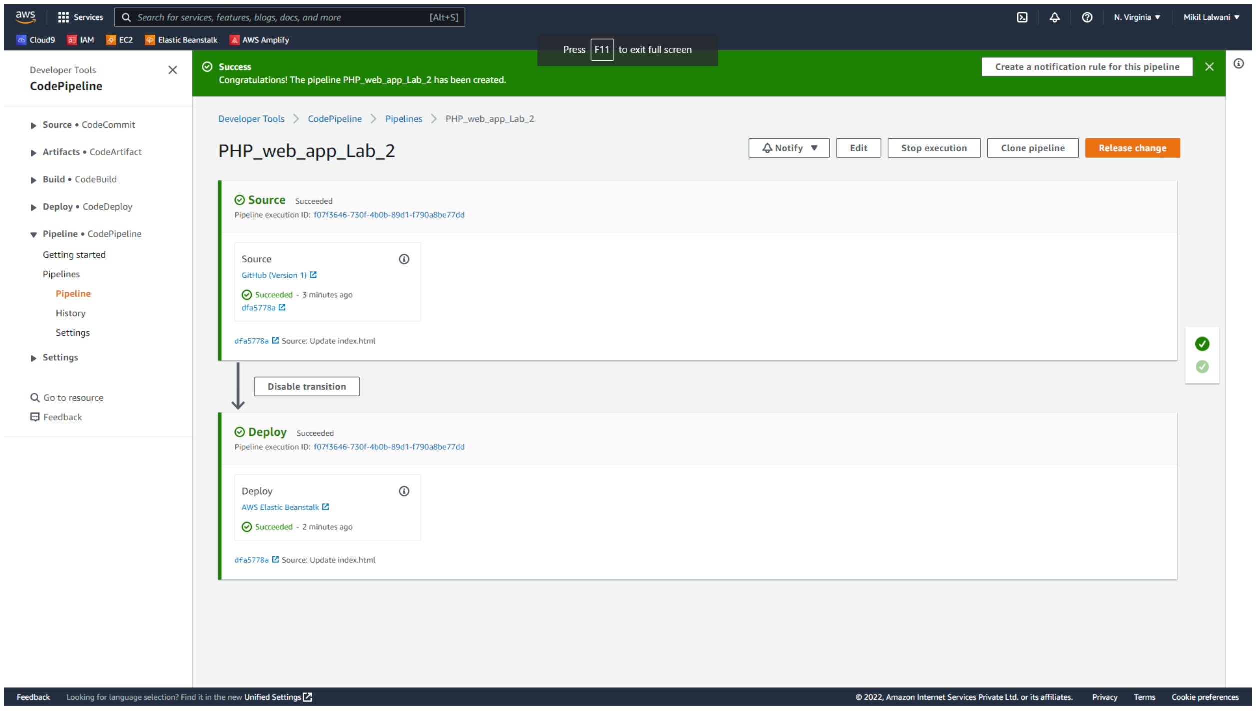
Health: Ok
Running version: Sample Application
Platform: PHP 8.0 running on 64bit Amazon Linux 2/3.3.15

Recent events

Time	Type	Details
2022-07-31 09:03:53 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 35 seconds ago and took 3 minutes.
2022-07-31 09:03:07 UTC+0530	INFO	Successfully launched environment: Phpwebapplab2-env
2022-07-31 09:03:06 UTC+0530	INFO	Application available at Phpwebapplab2-env.eba-5xzpgm9z.us-east-1.elasticbeanstalk.com.
2022-07-31 09:02:53 UTC+0530	INFO	Added instance [i-0e087dfa2562b180a] to your environment.
2022-07-31 09:02:51 UTC+0530	INFO	Instance deployment completed successfully.

Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.

You have successfully created an automated software release pipeline using AWS CodePipeline!


Success
Congratulations! The pipeline PHP_web_app_Lab_2 has been created.

Developer Tools > CodePipeline > Pipelines > PHP_web_app_Lab_2

PHP_web_app_Lab_2

Source Succeeded
Pipeline execution ID: f07f3646-730f-4b0b-89d1-f790a8be77dd

Source
GitHub (Version 1)
Succeeded - 3 minutes ago
dfa5778a

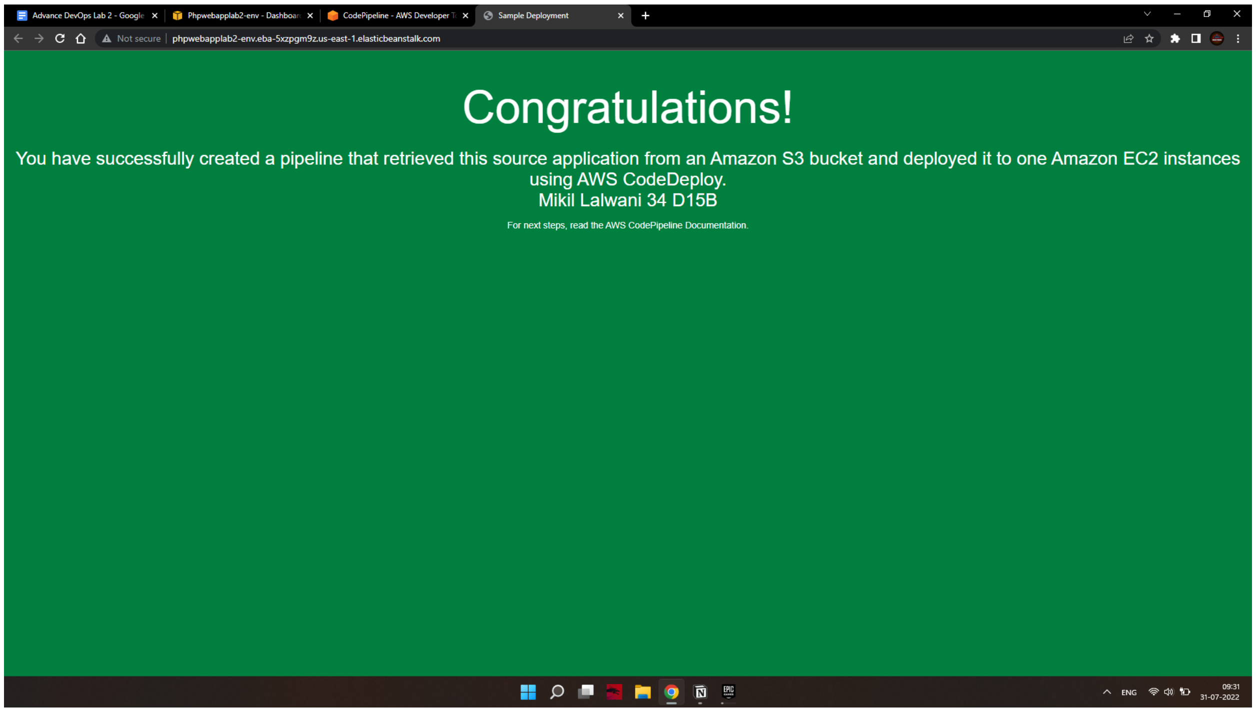
dfa5778a Source: Update index.html

Deploy Succeeded
Pipeline execution ID: f07f3646-730f-4b0b-89d1-f790a8be77dd

Deploy
AWS Elastic Beanstalk
Succeeded - 2 minutes ago

dfa5778a Source: Update index.html

Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk.


phpwebapplab2-env.eba-5xzpgm9z.us-east-1.elasticbeanstalk.com

Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to one Amazon EC2 instances using AWS CodeDeploy.
Mikil Lalwani 34 D15B

For next steps, read the AWS CodePipeline Documentation.

Advance DevOps Lab 2 - Google
Phpwebapplab2-env - Dashboard
CodePipeline - AWS Developer T
Sample Deployment
Not secure
09:31
31-07-2022

Step 7: Commit a change and then update your app

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

Note that the sample web page you deployed refers to AWS CodeDeploy, a service that automates code deployments. CodePipeline, CodeDeploy is an alternative to using Elastic Beanstalk for deployment actions. Let's update the sample code so that it correctly states that you deployed the sample using Elastic Beanstalk.

a. Visit your own copy of the repository that you forked in GitHub.

- Open index.html
- Select the Edit icon

b. Update the webpage by copying and pasting the following text on line 30:

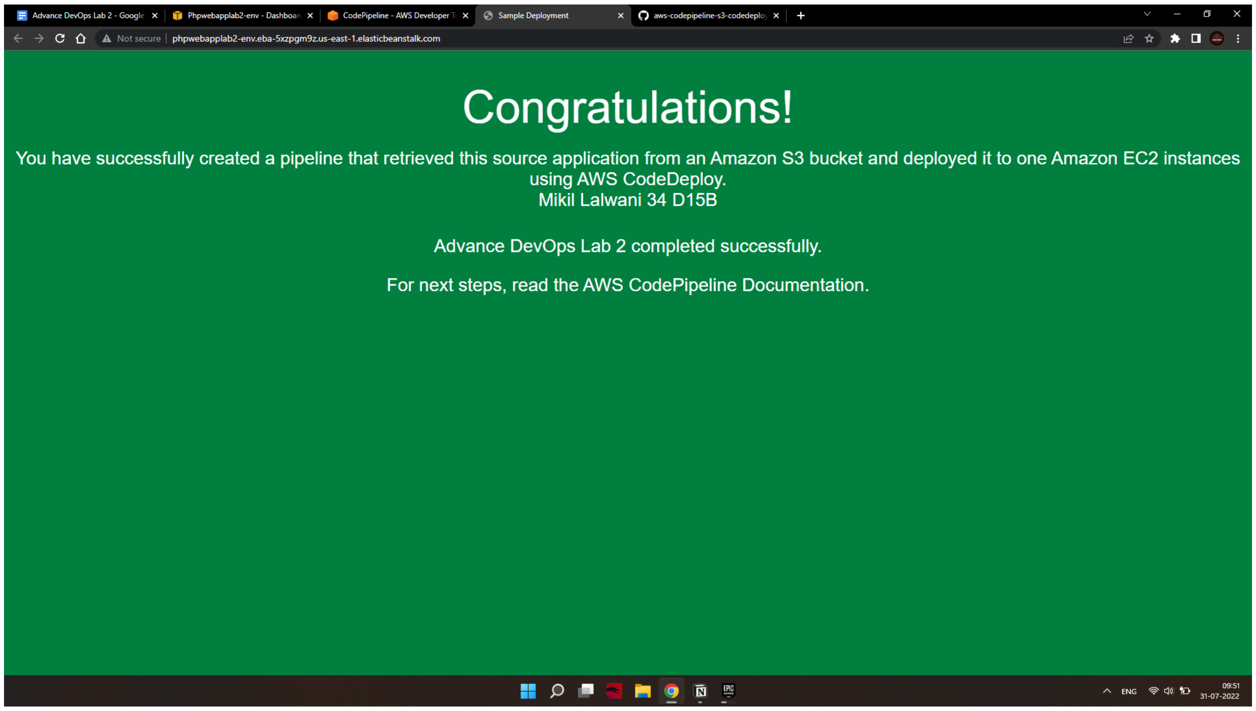
c. Commit the change to your repository.

d. Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.

- After the pipeline status displays Succeeded, in the status area for the Beta stage, click AWS

  Elastic Beanstalk.

e. The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier. And click the URL again from the EBS environment again.



Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to one Amazon EC2 instances using AWS CodeDeploy.
Mikil Lalwani 34 D15B

Advance DevOps Lab 2 completed successfully.

For next steps, read the AWS CodePipeline Documentation.

Step 8: 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.

You have successfully created an automated software release pipeline using AWS CodePipeline! Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Your pipeline will automatically deploy your code every time there is a code change.