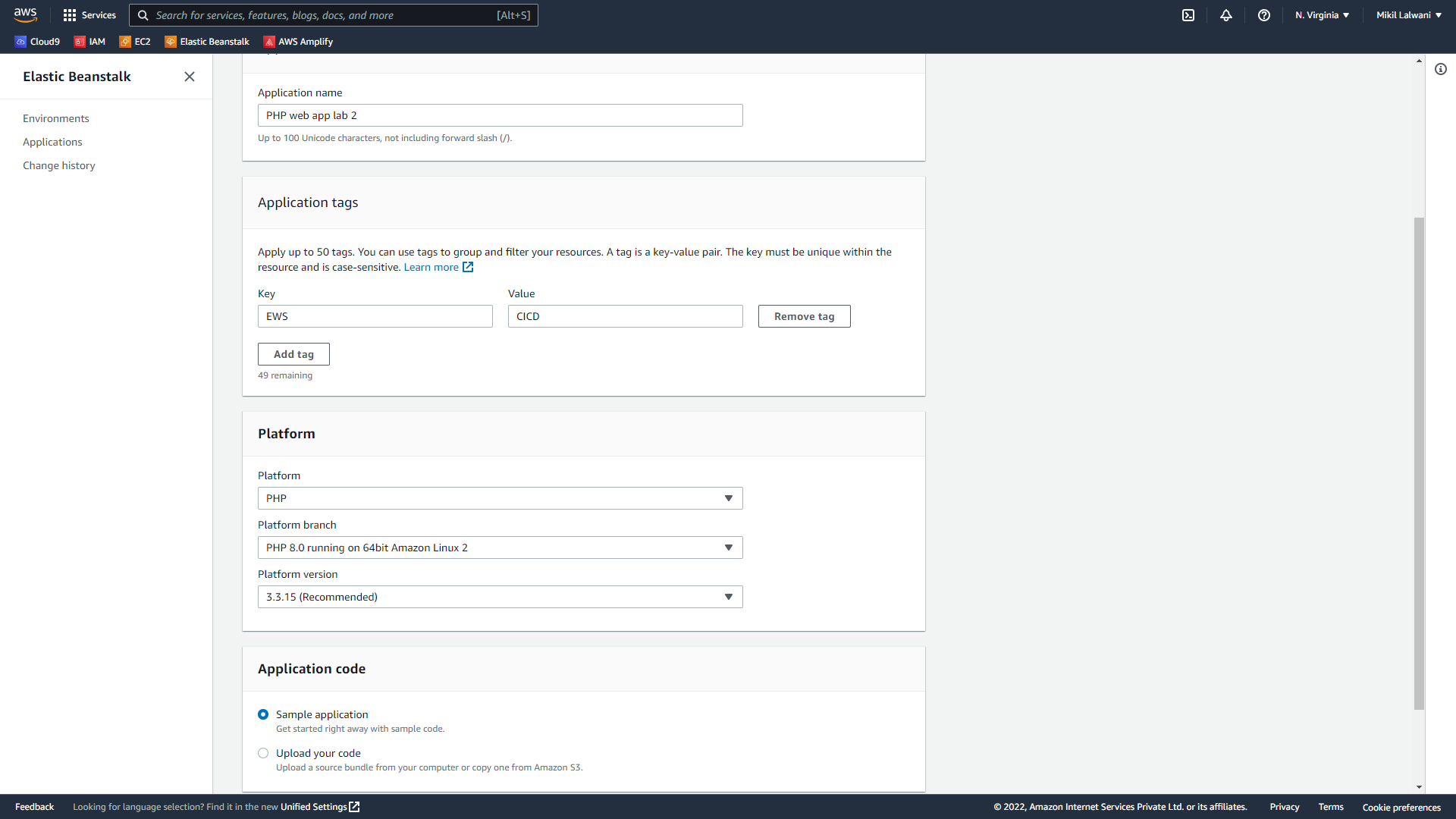
Step 1: Create a deployment environment

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.



1. 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.**

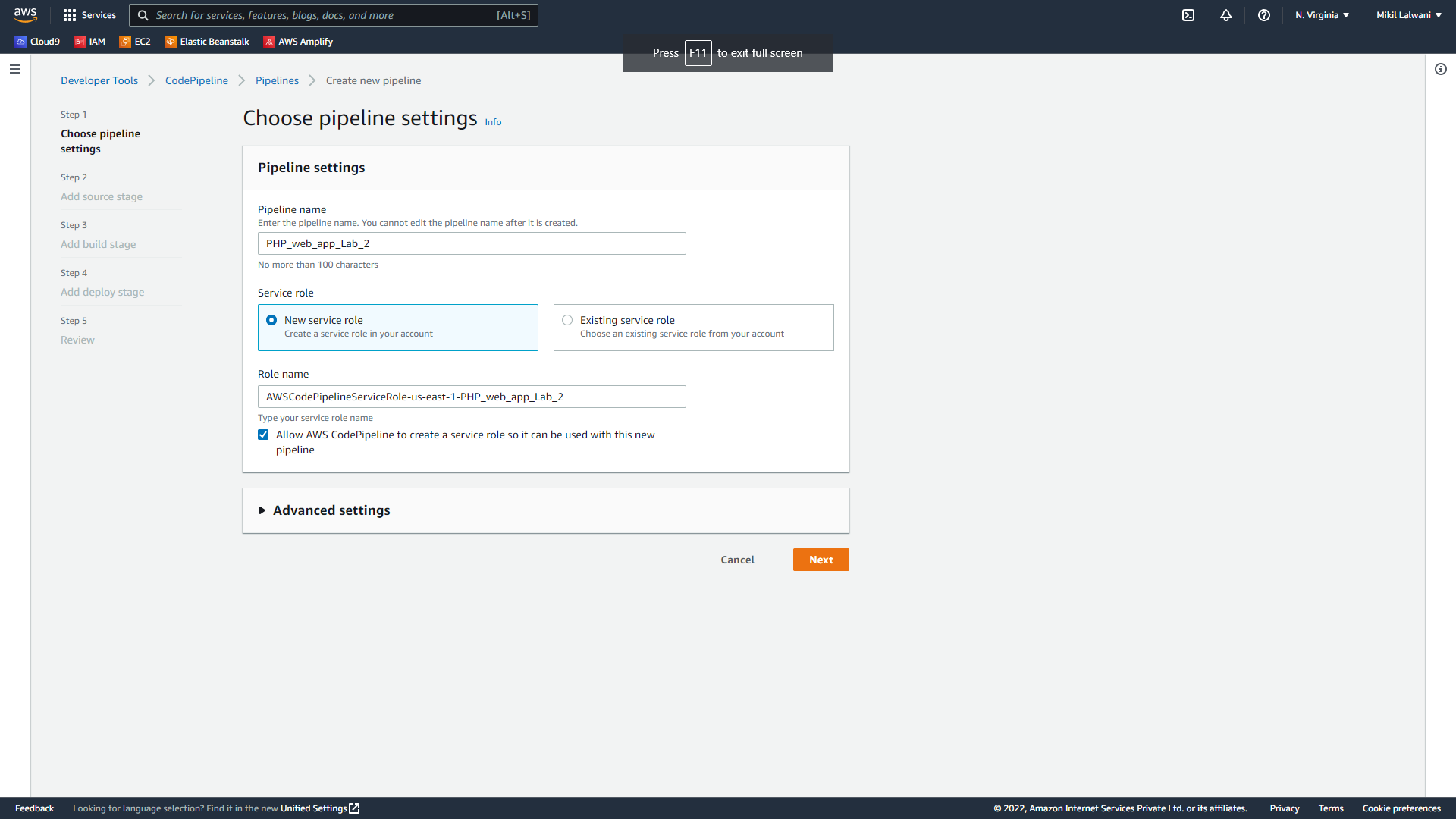
Step2: Get a copy of the sample code

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.

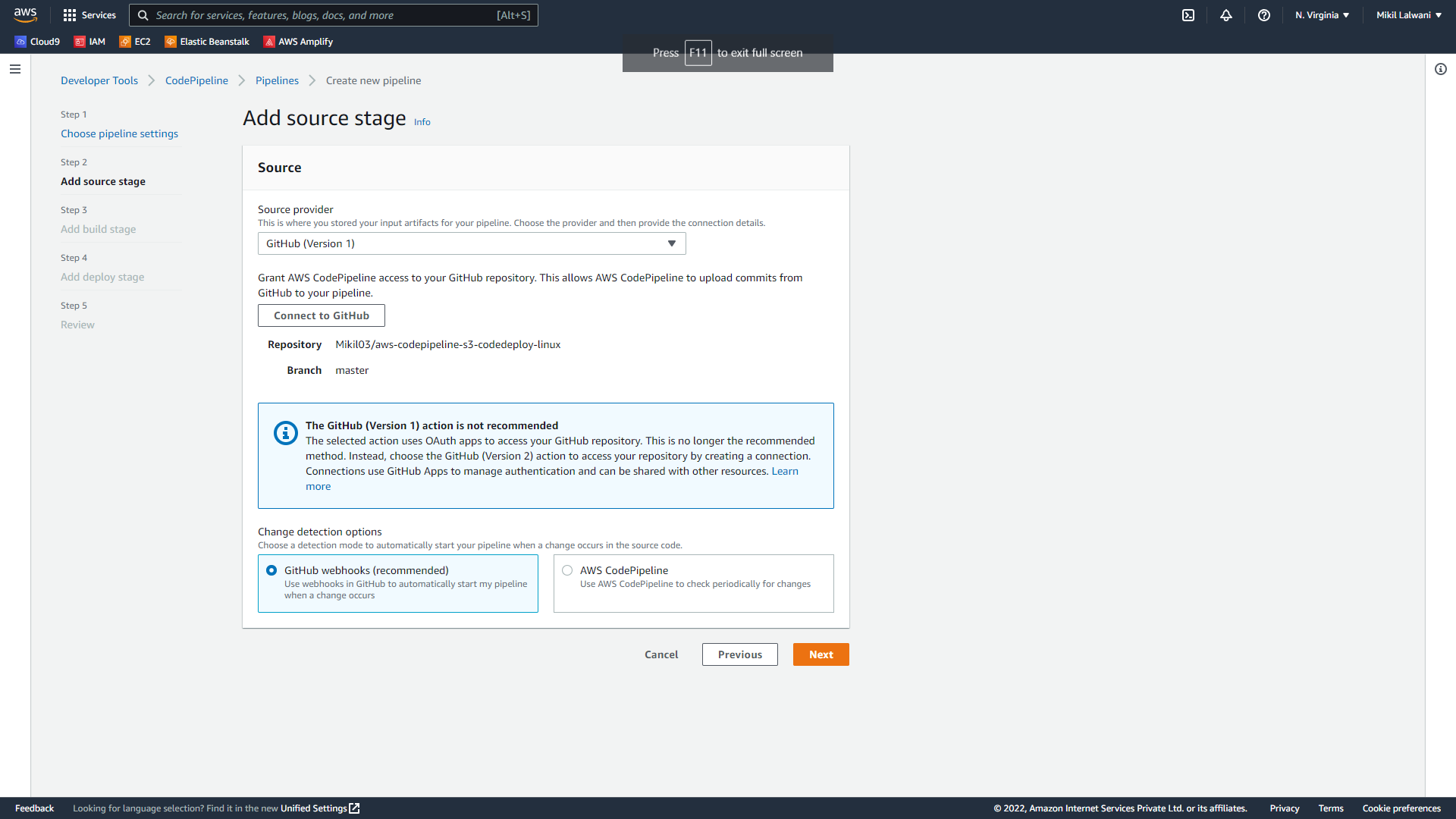
Github repository: [aws-codepipeline-s3-codedeploy-linux](https://github.com/jewanikajal/aws-codepipeline-s3-codedeploy-linux)

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.

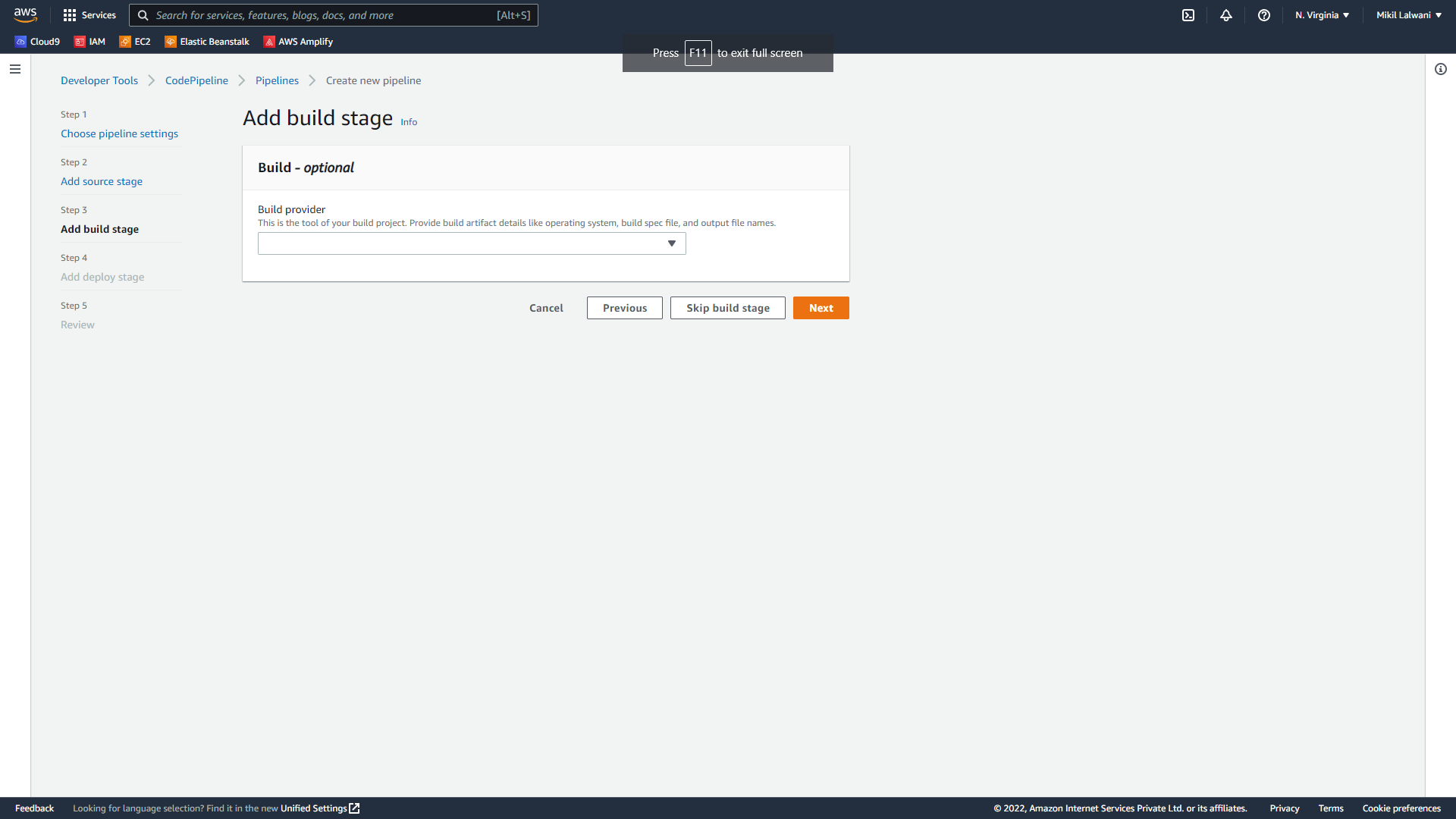
Step 4: Source Stage:

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.



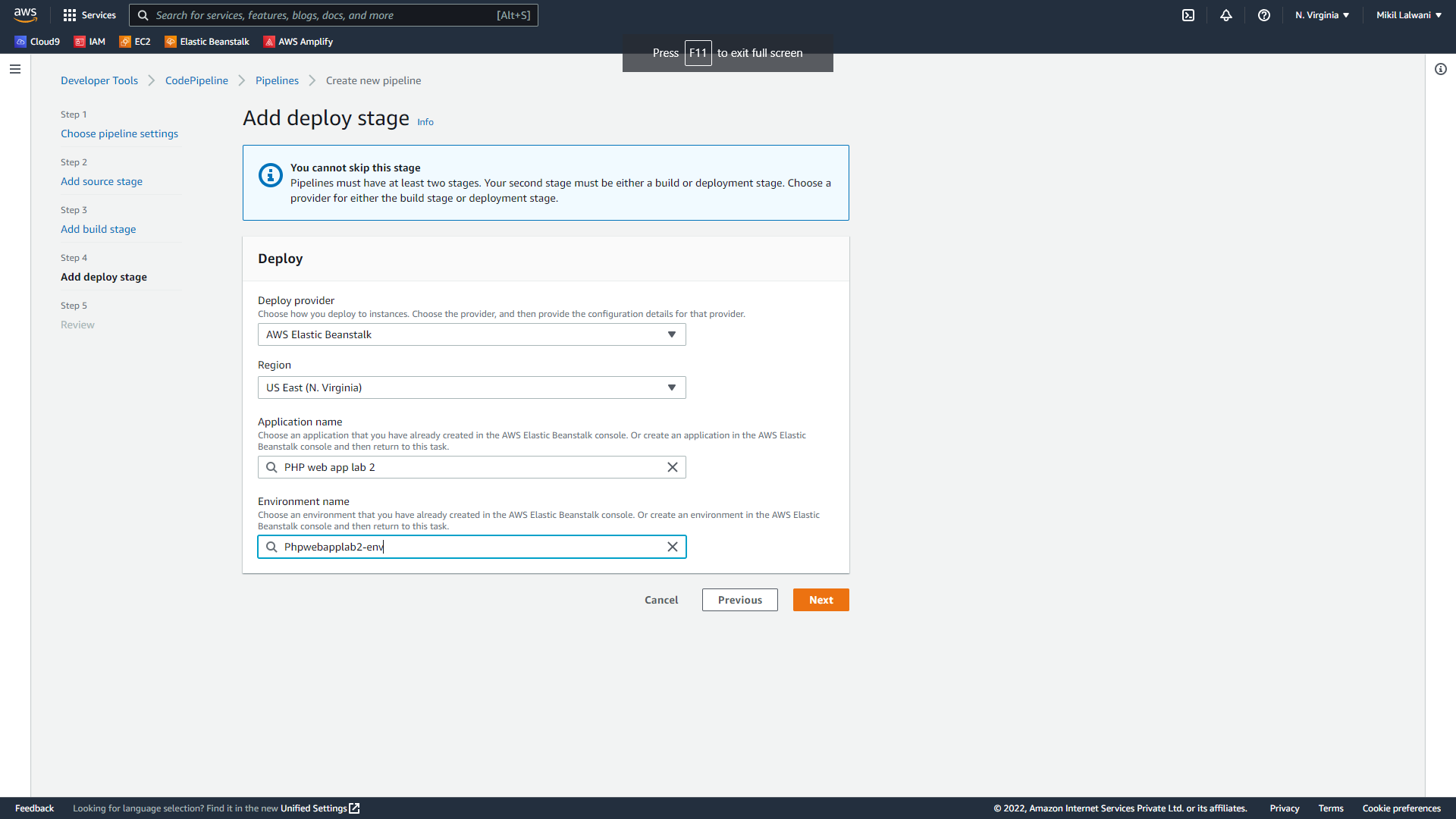
Step 5: Build Stage:

1. Click on Skip build stage.



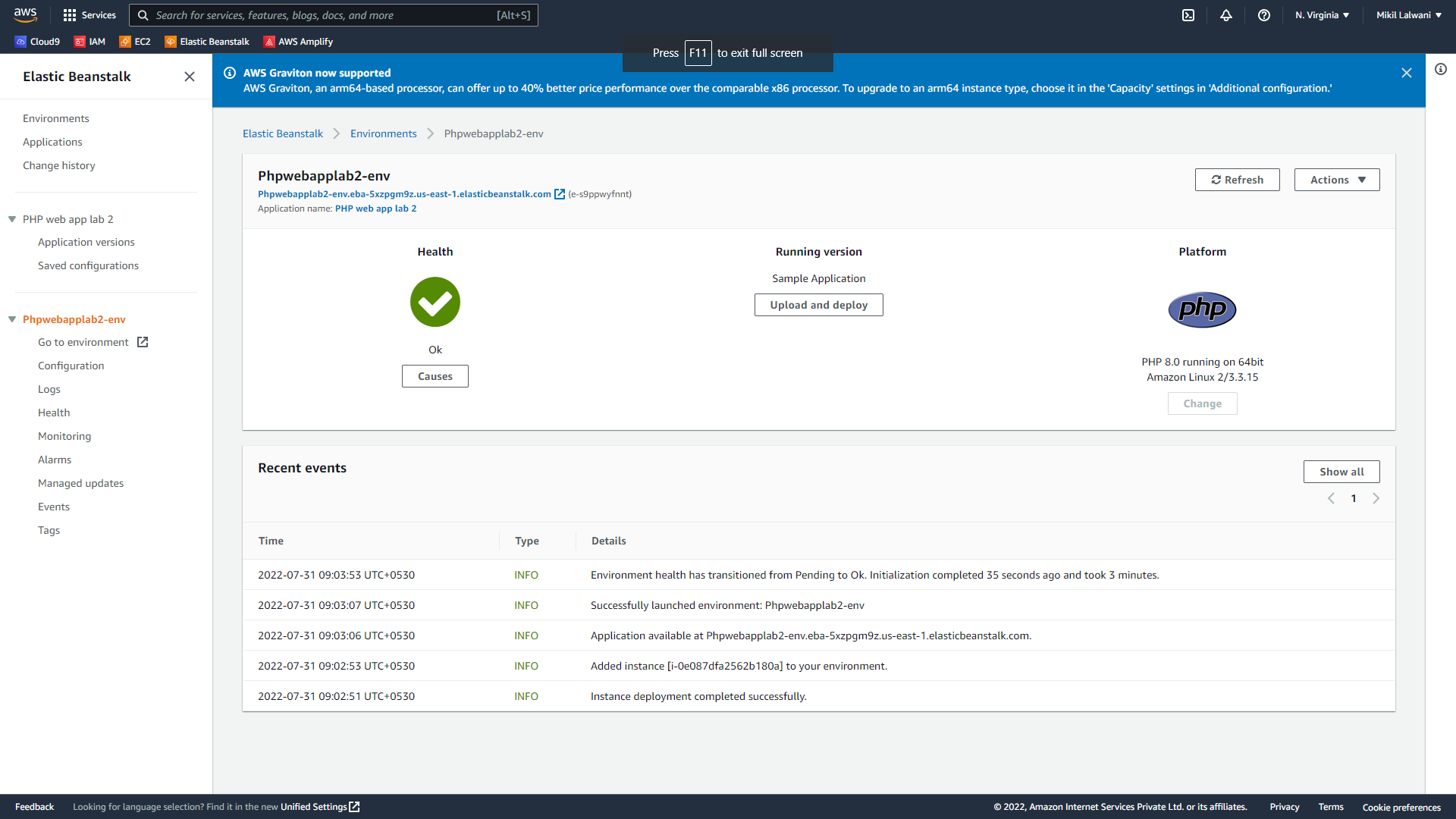
Step 6: Deploy Stage:

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

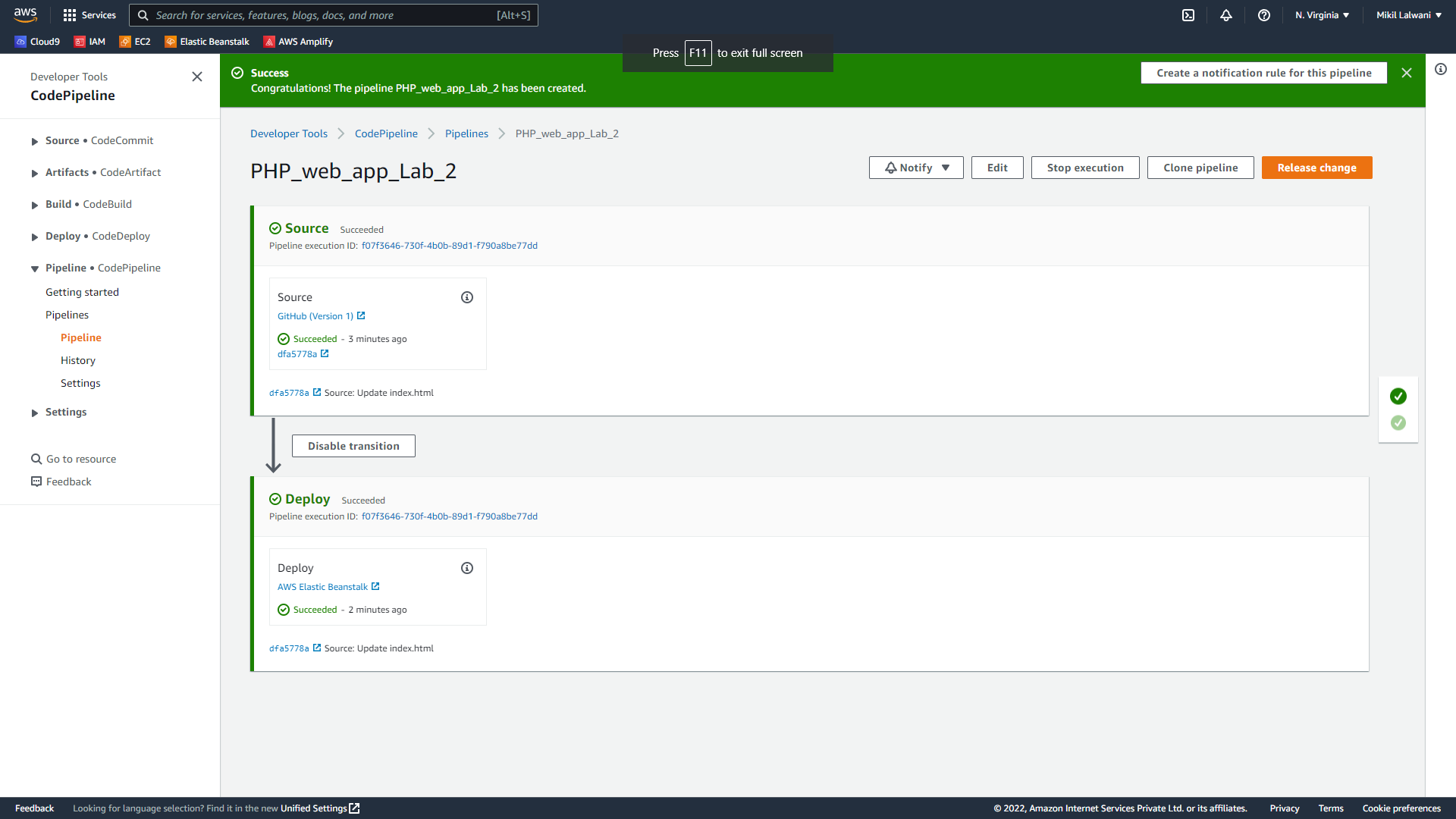


1. After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline perform each action.
2. To verify your pipeline ran successfully, monitor the progress of the pipeline as it 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.

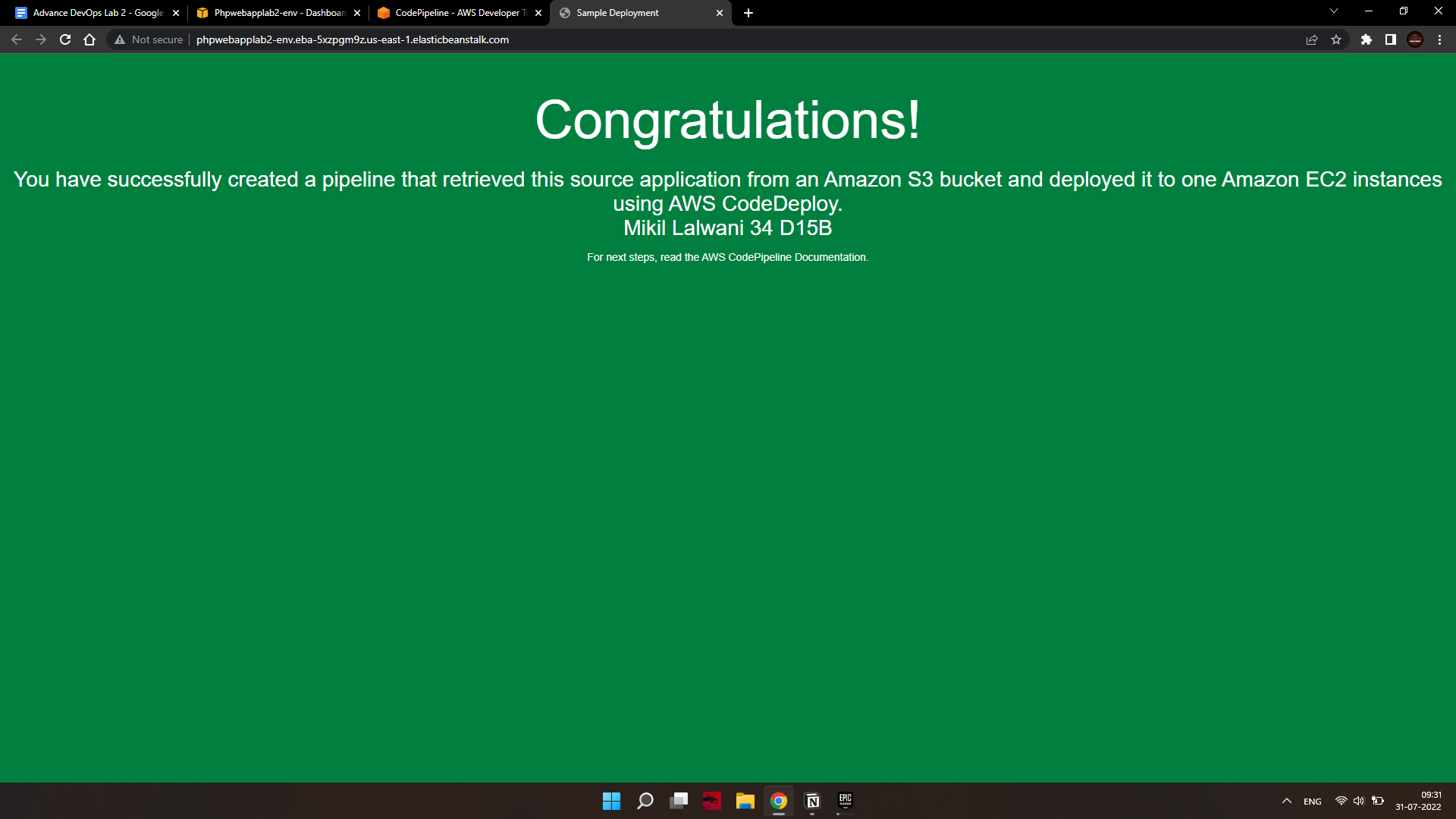
Now go to your EBS environment and click on the URL to view the sample website you deployed.

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

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.



## 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:

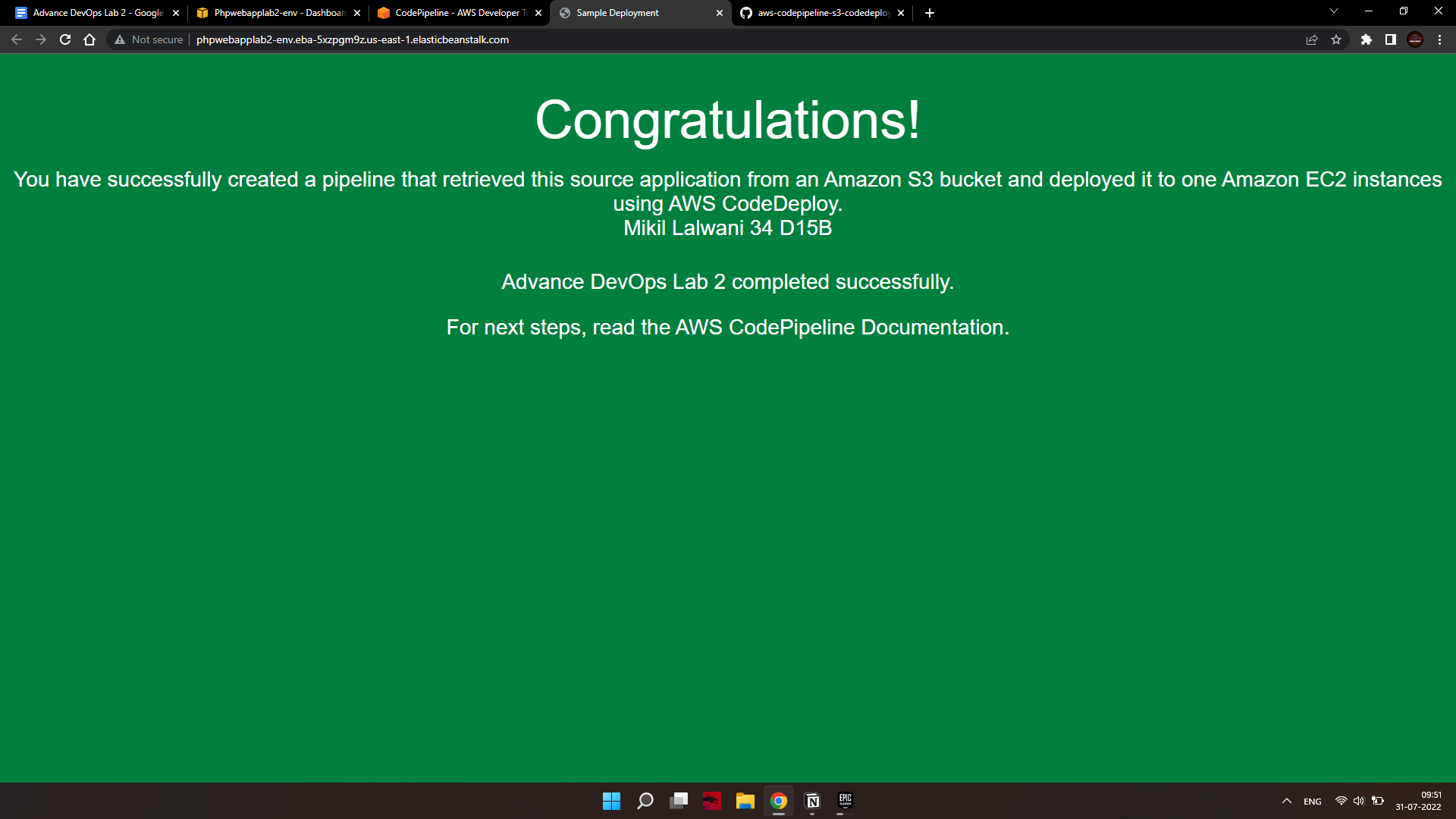
c. Commit the change to your repository.

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.



# 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.