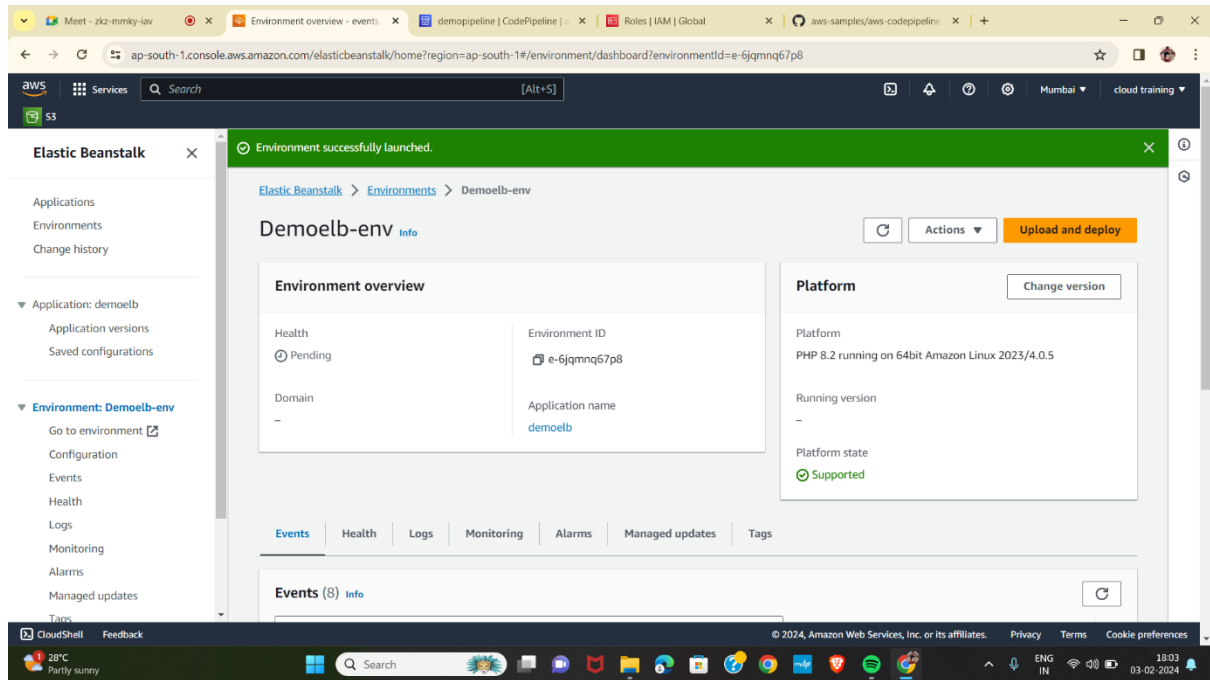# Step 1: Create a deployment environment

- To start, open the Elastic Beanstalk console and choose Create Application.
- For Application name, enter demoelb. Select PHP from the dropdown menu . choose Create application.
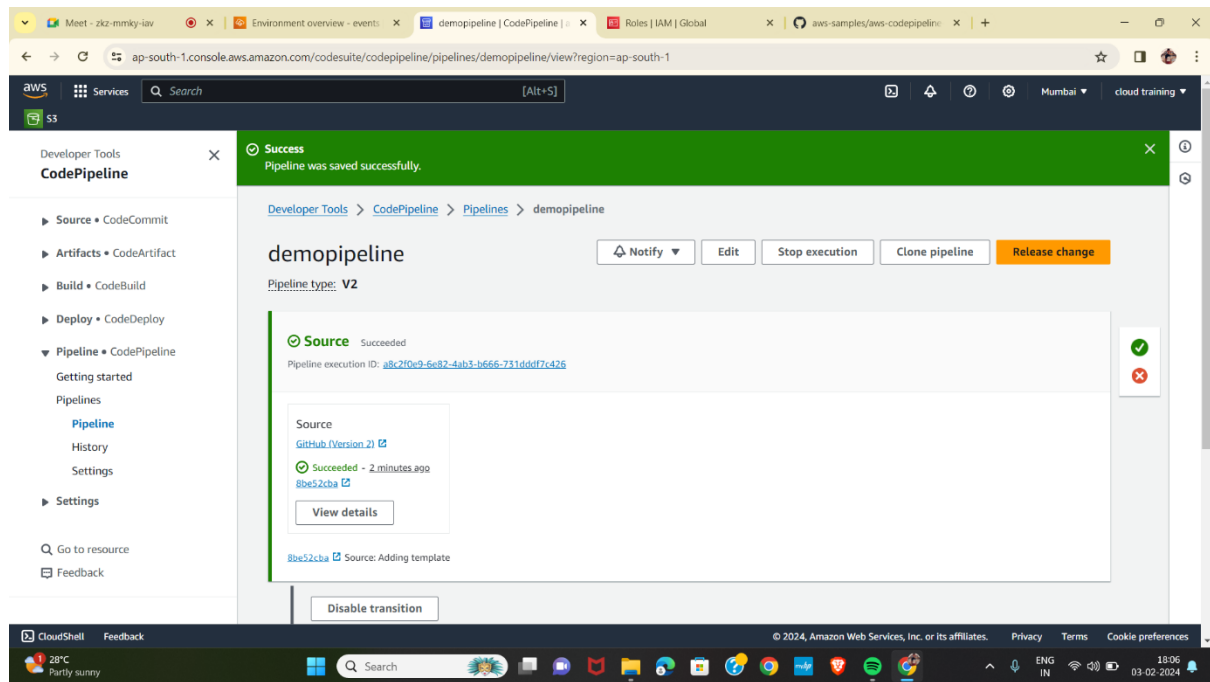

Step 2: Get a copy of the sample code

- If you would like to use your GitHub account:Visit our GitHub repository containing the sample code at https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux.

Step 3: Create your pipeline

- Open the AWS CodePipeline console.choose Create pipeline.
- Pipeline name: Enter the name for your pipeline, DemoPipeline
- Add source stage page, select the location of the source you selected and follow the steps below:.Select GitHub (Version 2) for the Source provider.Choose Connect to GitHub.
- Enter demoelb for Connection name and choose Connect to GitHub.
- Select Authorize AWS Connector for GitHub.
- When redirected back to the connection screen, choose Install a new app.
- The Add source page will be updated to reflect GitHub is ready to connect. Specify the repository and branch:Repository name: select the GitHub repository you want to use as the source location for your pipeline. Select the forked repository in your GitHub account named aws-codepipeline-s3-codedeploy-linux.Branch name: select the branch you want to use, master.Output artifact format: Select CodePipeline default.
- Select Skip. Select AWS Elastic Beanstalk. Retain the default region. Select demoelb Env.Select Demoelb-env.

Step 4: Activate your pipeline to deploy your code

- Select AWS Elastic Beanstalk.
- Select the URL that appears in the upper-right part of the page to view the sample website you deployed.

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.