**Experiment No 2**

**Name: Arya Manoj Madhavi**

**Div: D15B**

**Roll No.:31**

**Batch B**

**Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.**

**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

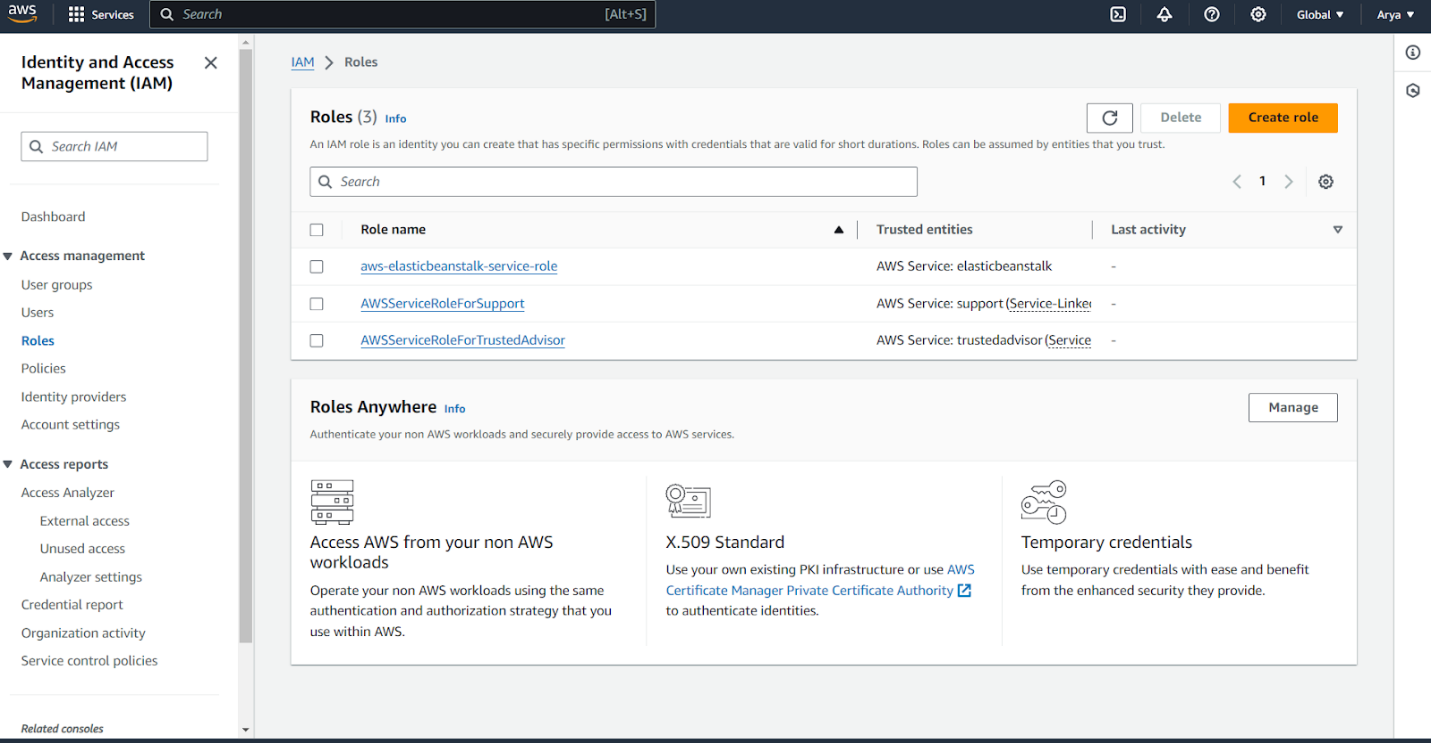
Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the

sample app’s code. You will also use AWS Elastic Beanstalk as the deployment target for the

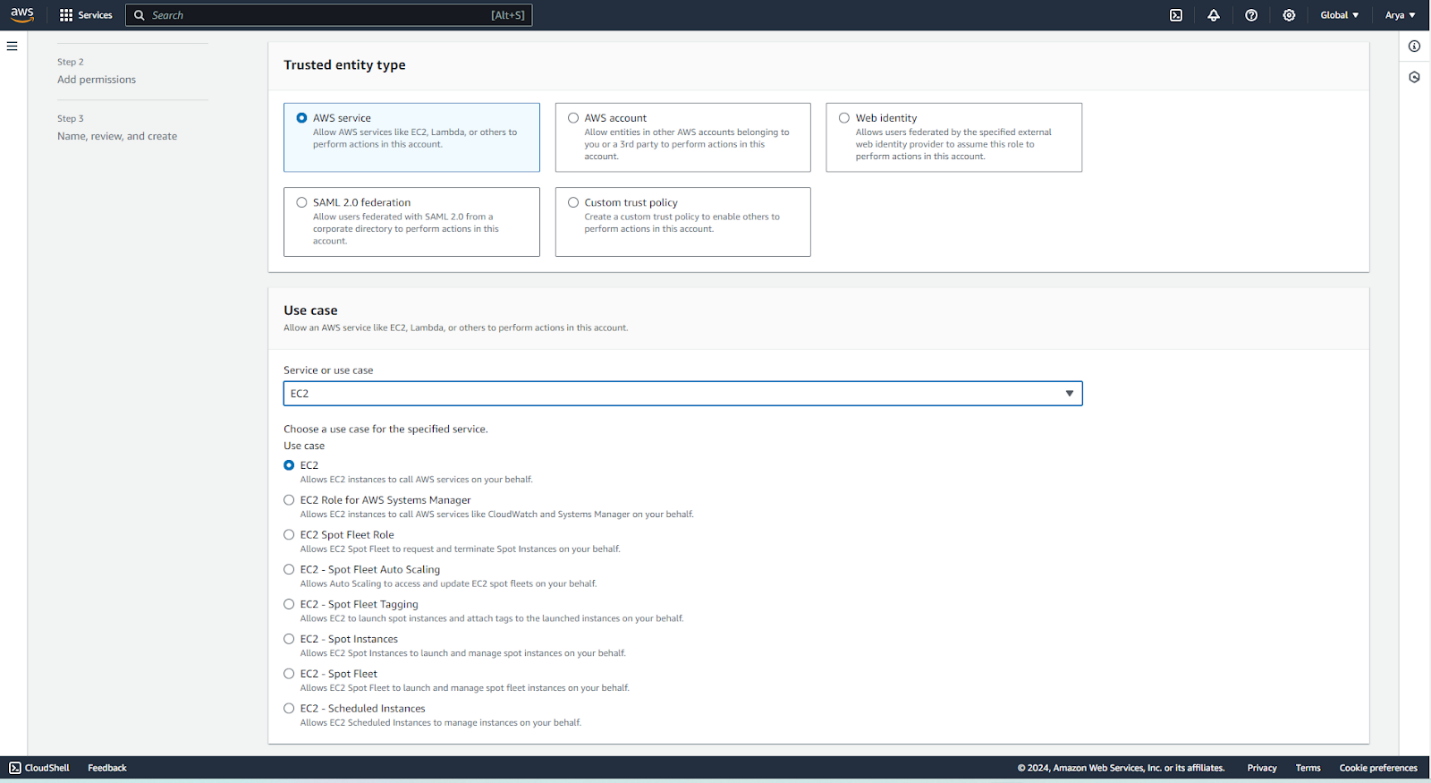
sample app. Your completed pipeline will be able to detect changes made to the source repository

containing the sample app and then automatically update your live sample app.

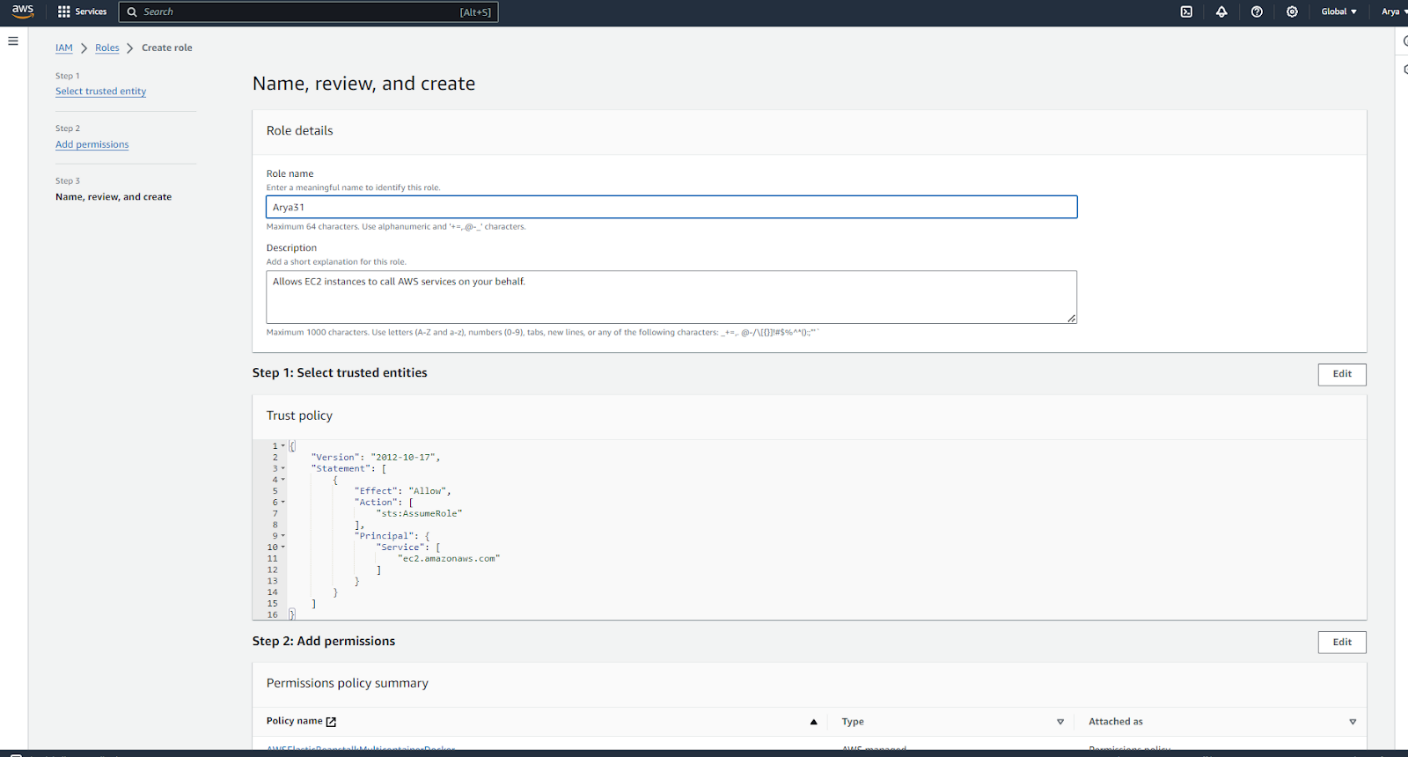
**Create a role in an IAM.**



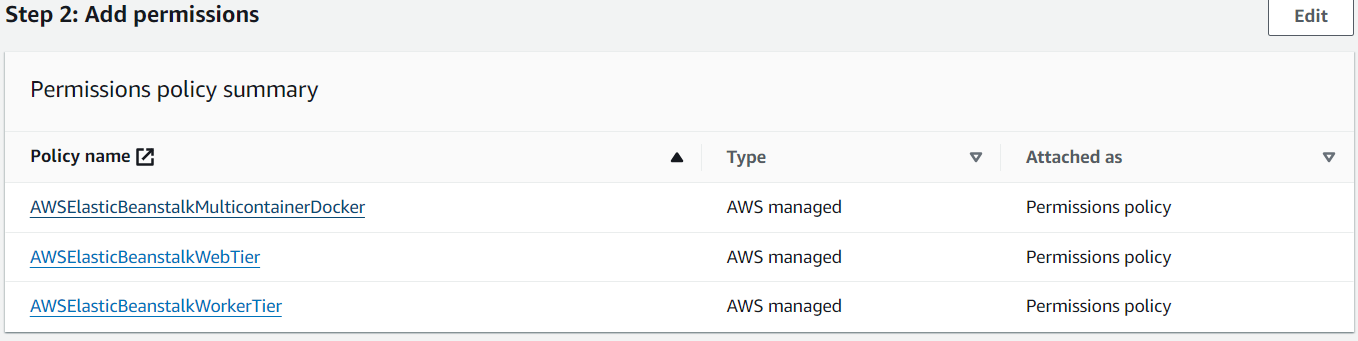
**Add EC2 for a service or use case.**



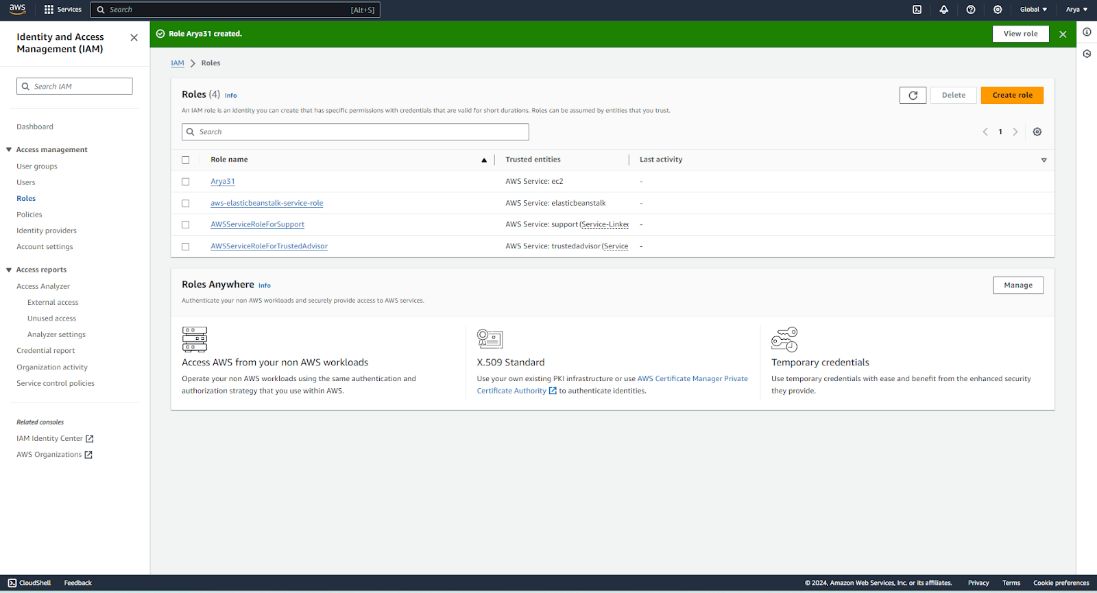
**Give name to the role.**



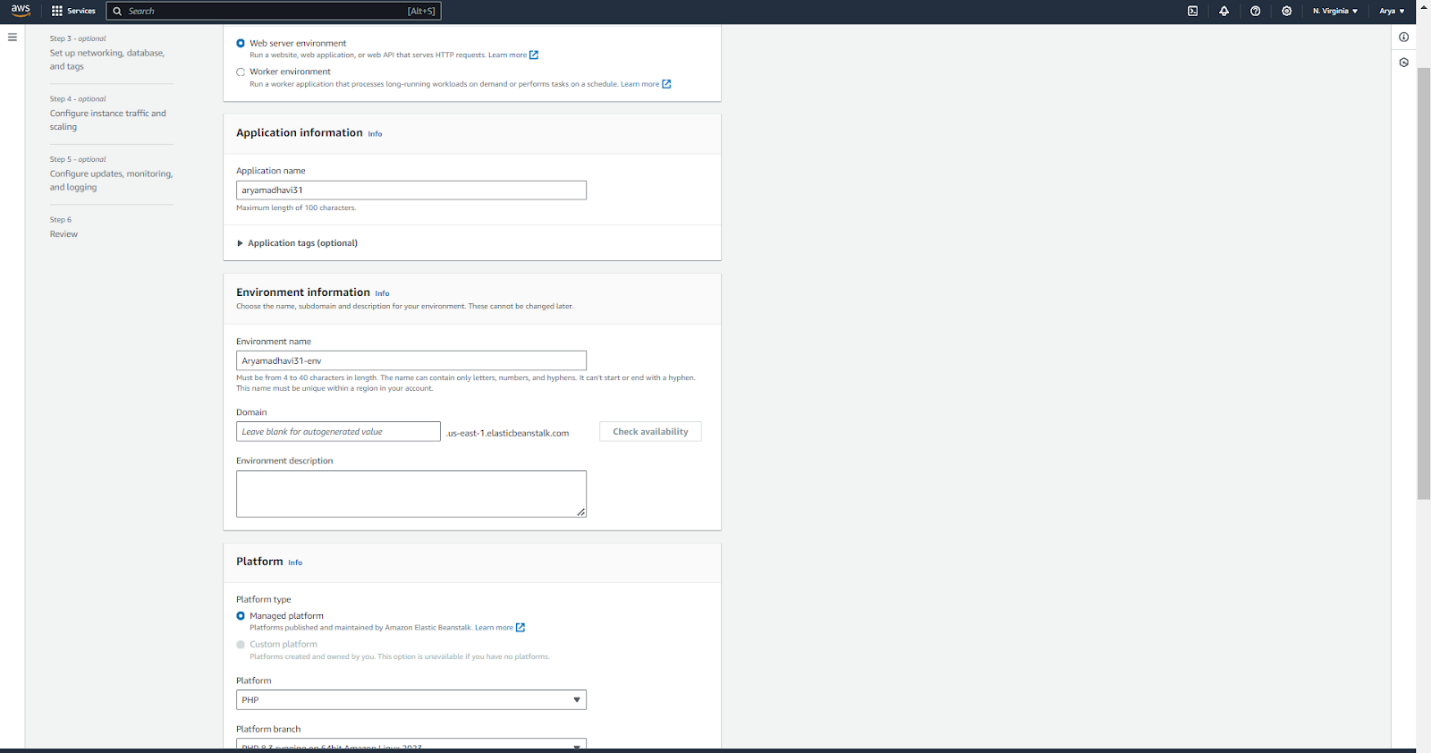
**Required policies (permissions) to be added while creating IAM user.**

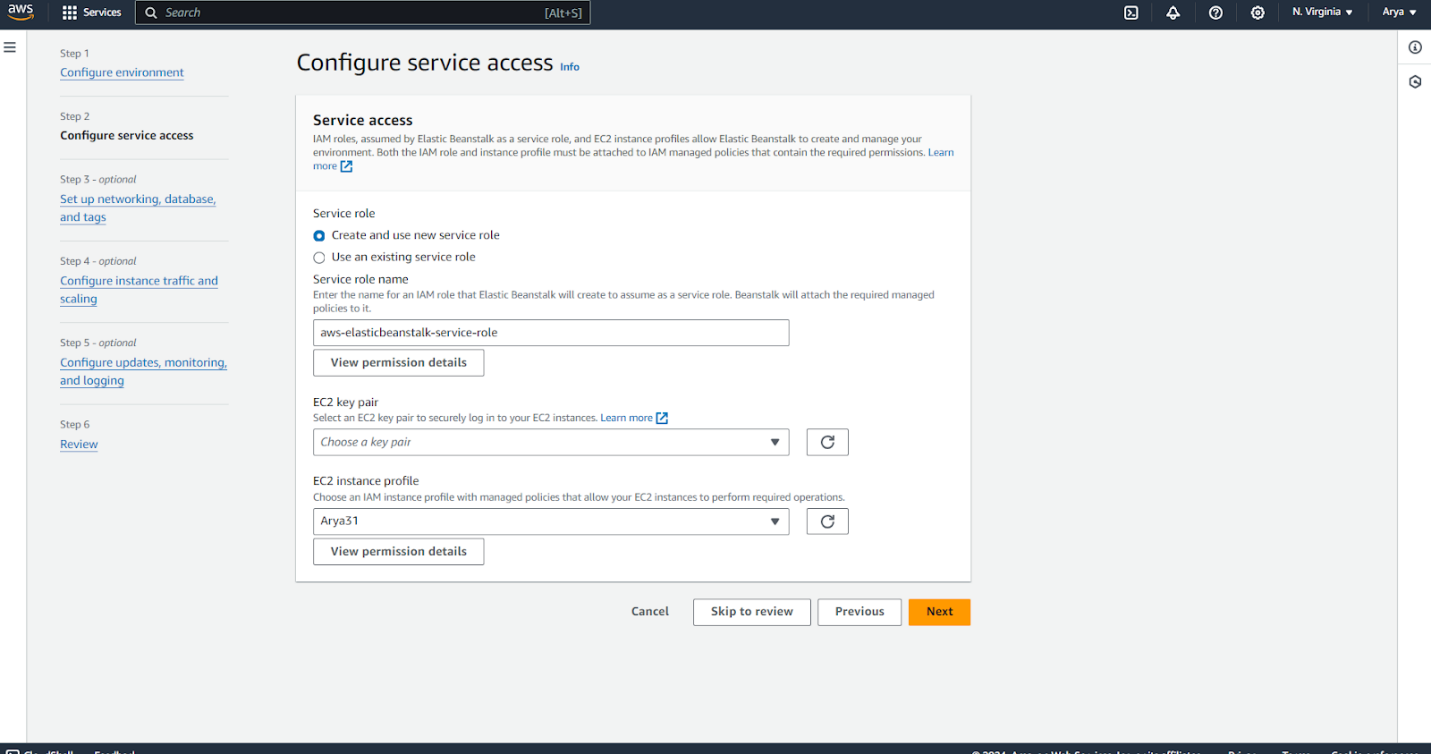
****

**IAM role is being created**

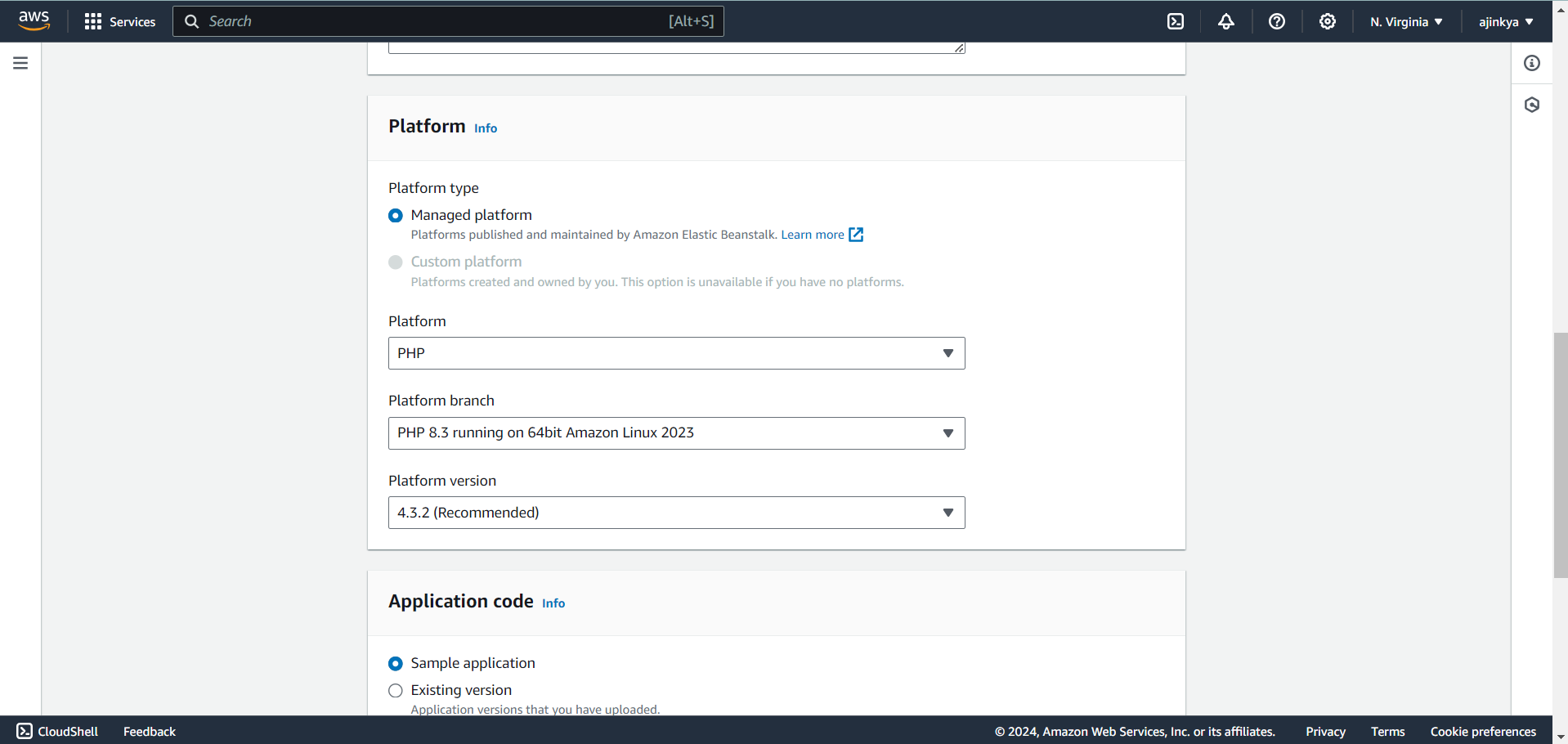


**Go to the Elastic beanstalk and create an application. Give the appropriate name for the application.**

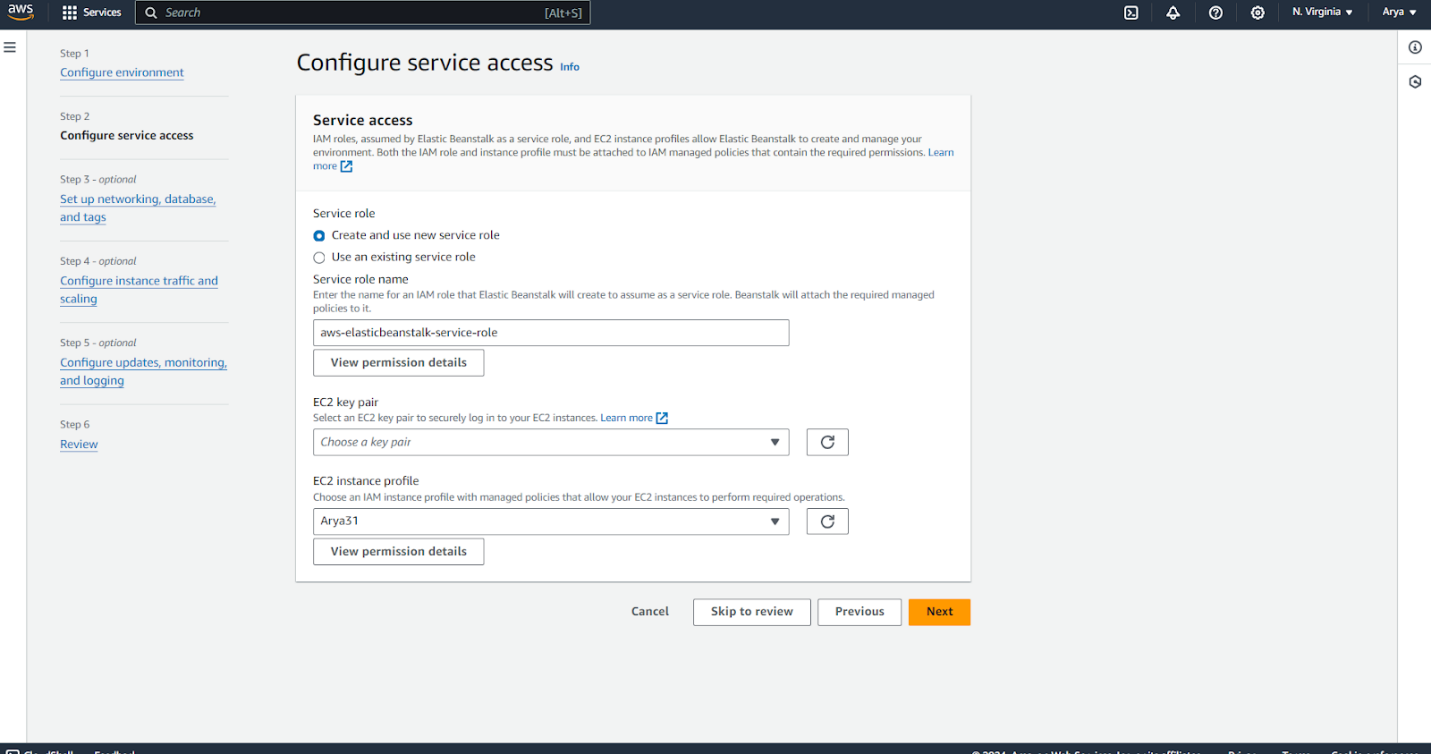




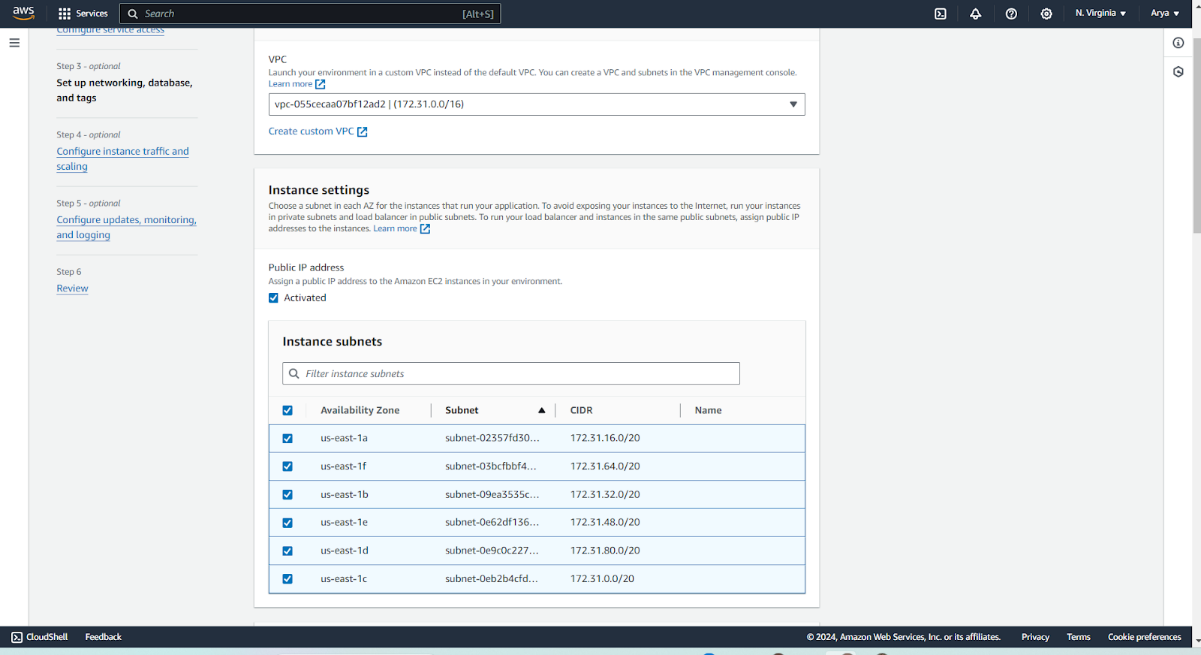
**Select the platform as PHP.**

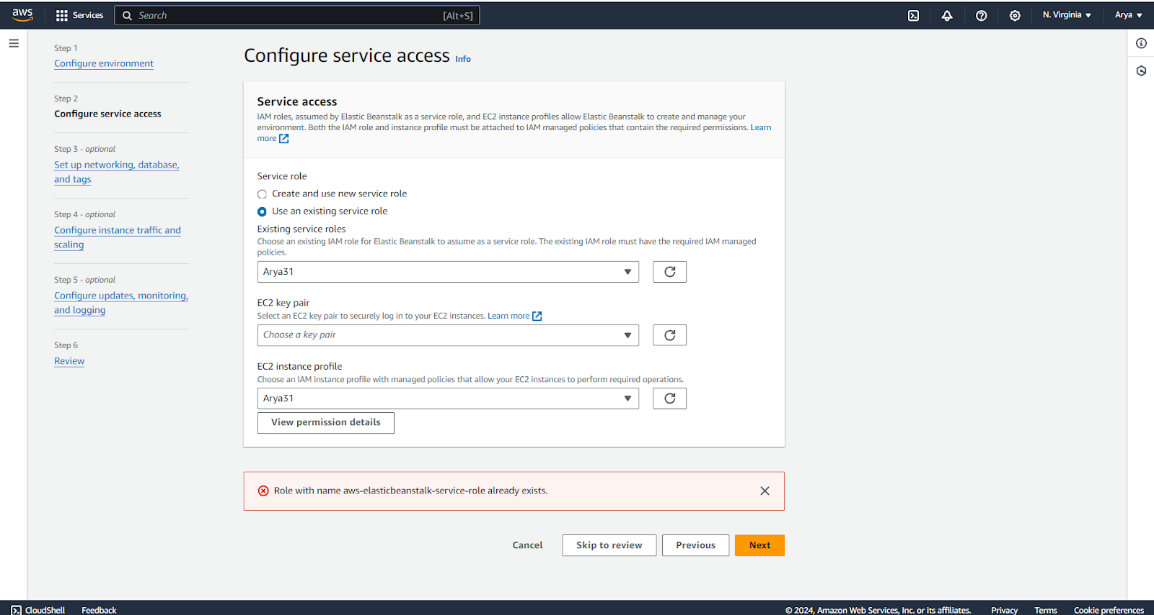
****

**In an ec2 instance profile, select the created IAM role.**

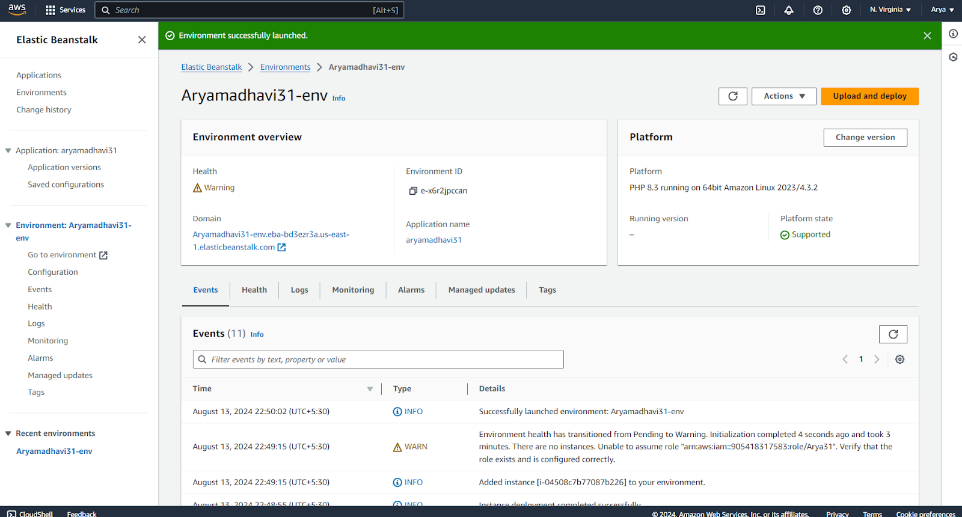


**Vpc is to be selected. Public IP address and availability is to be checked.**

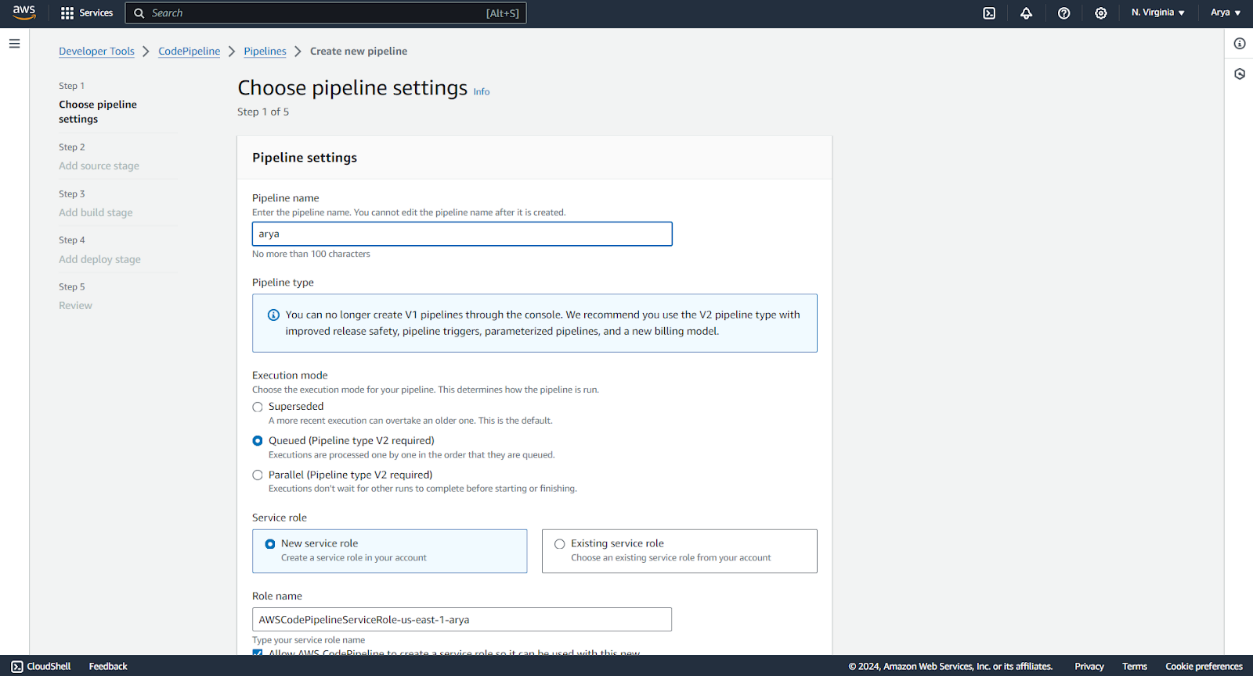




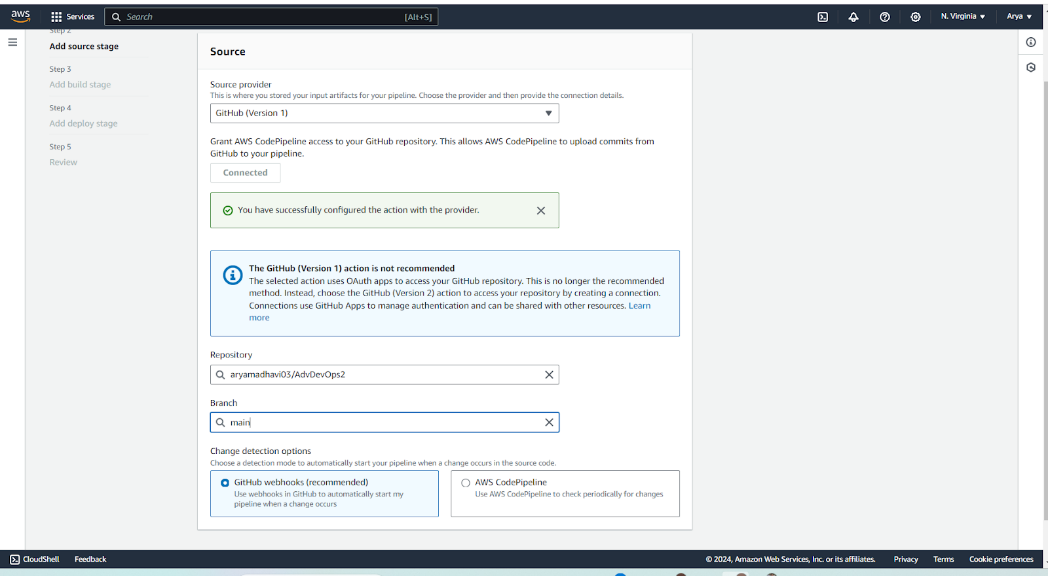
**Environment is launched successfully.**



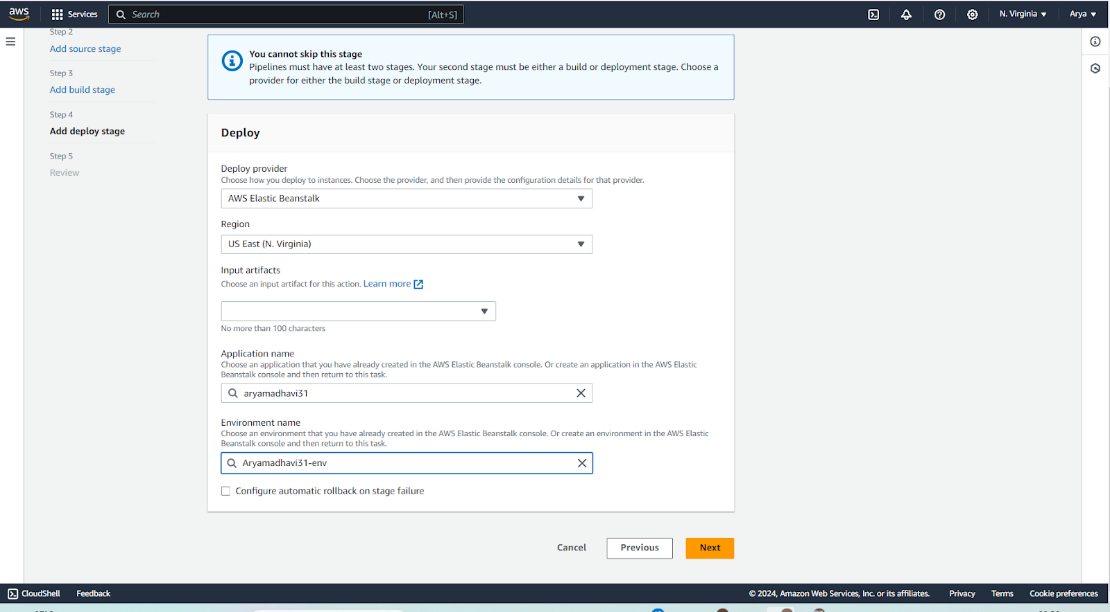
**Go to the CodePipeline and select the source as GitHub (version 1).**

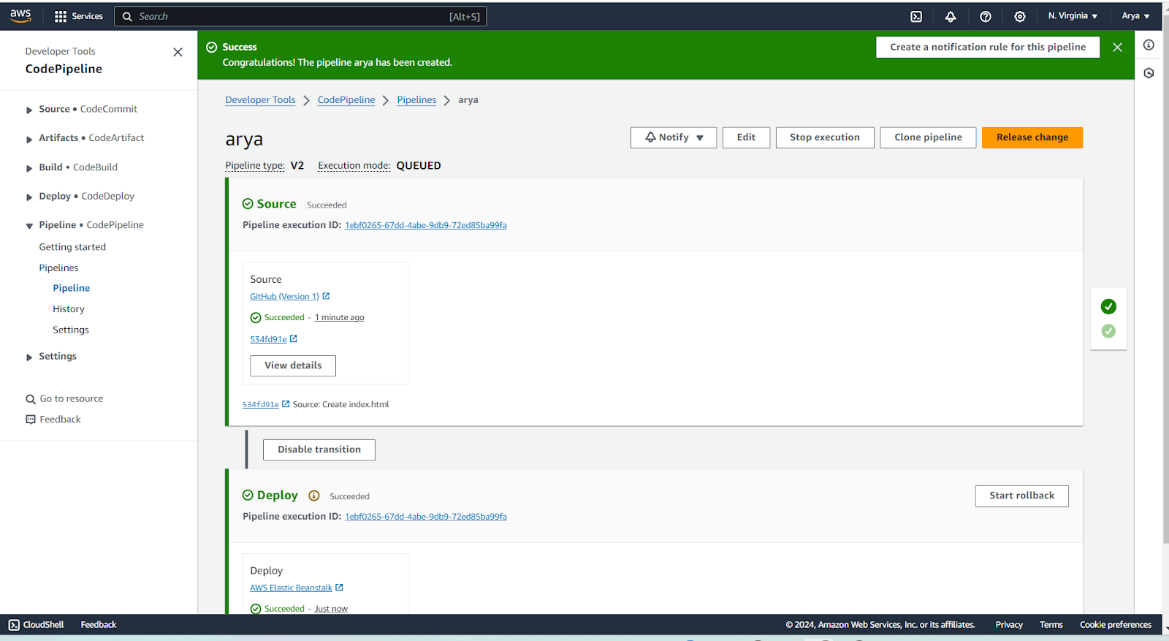


**After skipping the build stage, AWS Elastic beanstalk is to be selected in the Deploy Provider. Select your recently created application name and environment name.**

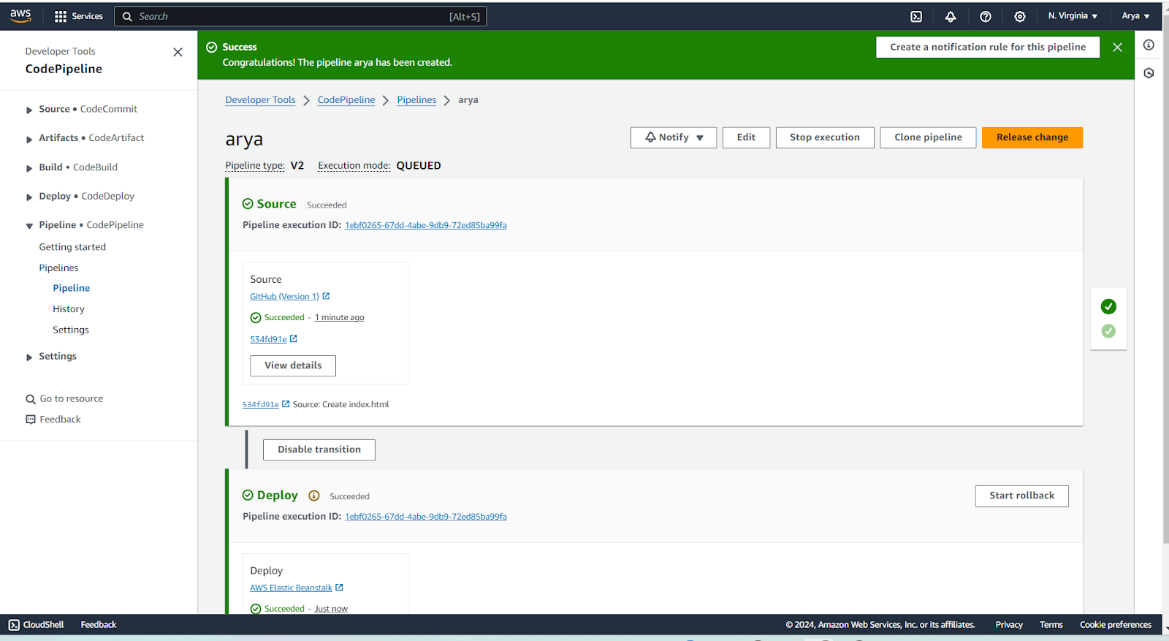


**Pipeline is created. The Source and Deploy section is also successful.**

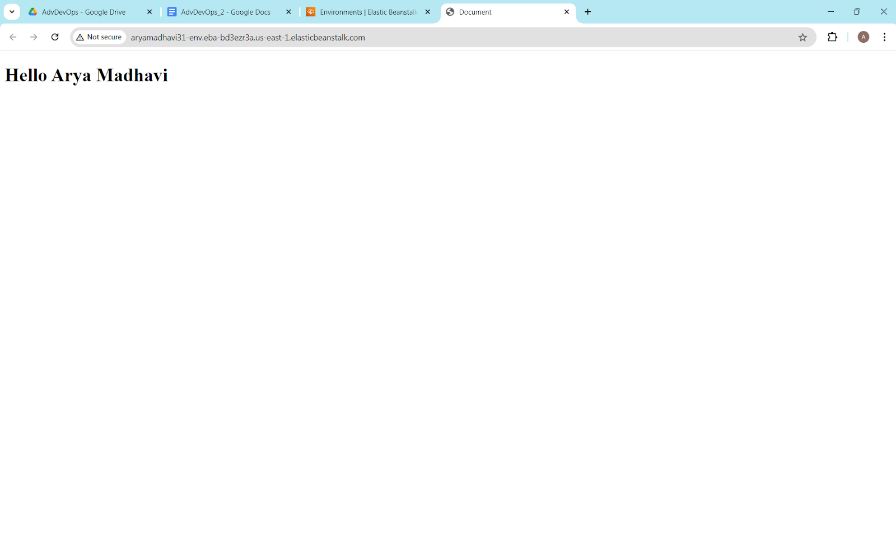




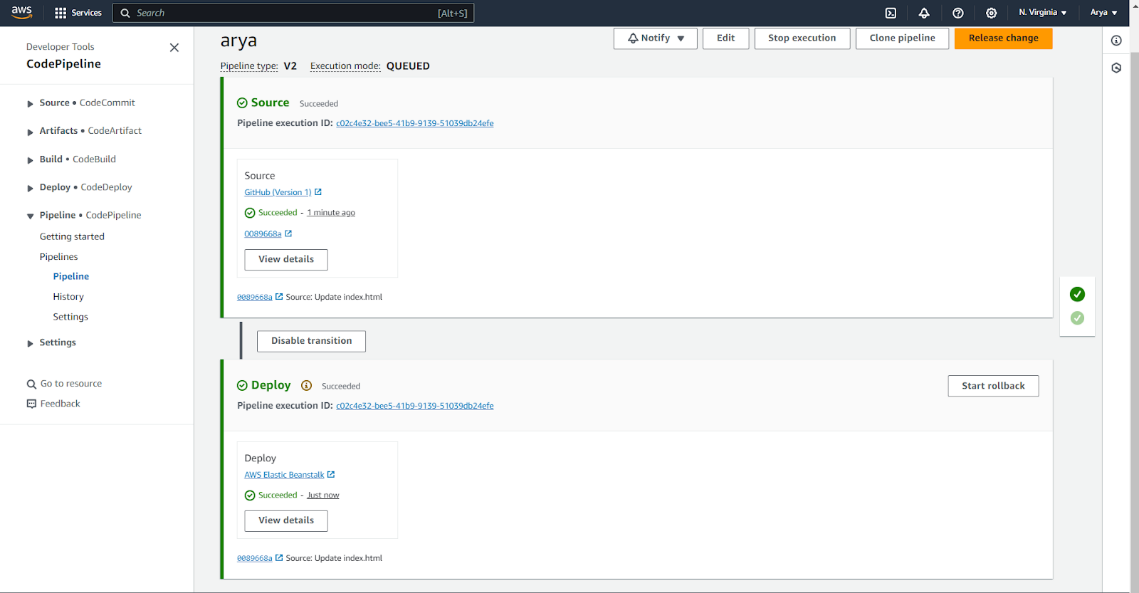
**Go to the elastic beanstalk environment and click on domain.**

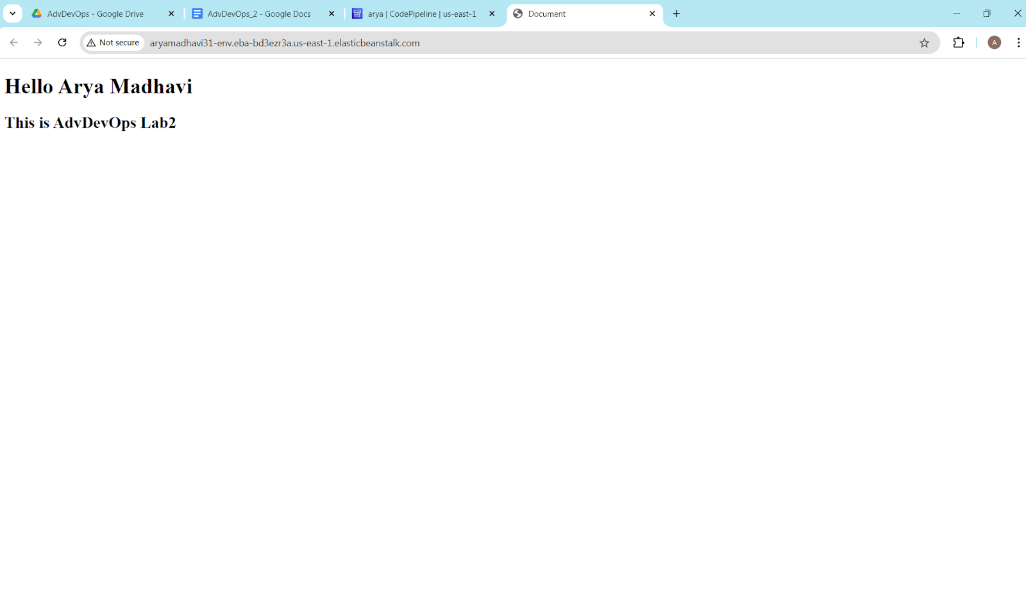


**Following output is to be generated of the code which is in the github repository.**



**Changes are done in the code of the Github repository and it is being directly deployed without any configurations.**





**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.

Compiled By: Prof.Vishal Badgujar-7709933639 Information Technology Department

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.

**Conclusion:**

1. Build and Deploy using CodeBuild and CodePipeline: Automate the build and deployment of your application to S3 or SEBS, ensuring a smooth CI/CD pipeline.

2. Deploy on EC2 using CodeDeploy: Use CodeDeploy to automatically deploy a sample application to an EC2 instance, ensuring efficient and reliable updates.