Advance DevOps Lab Experiment 02

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.

Roll No.	22
Name	Sarthak Harade
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.
Grade:	

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 is a key practice in modern DevOps, enabling organizations to streamline their software release process by automating the deployment of application updates. It allows for the seamless delivery of code revisions to production environments without requiring explicit approval from a developer, thereby reducing time-to-market and enhancing the overall efficiency of the development lifecycle.

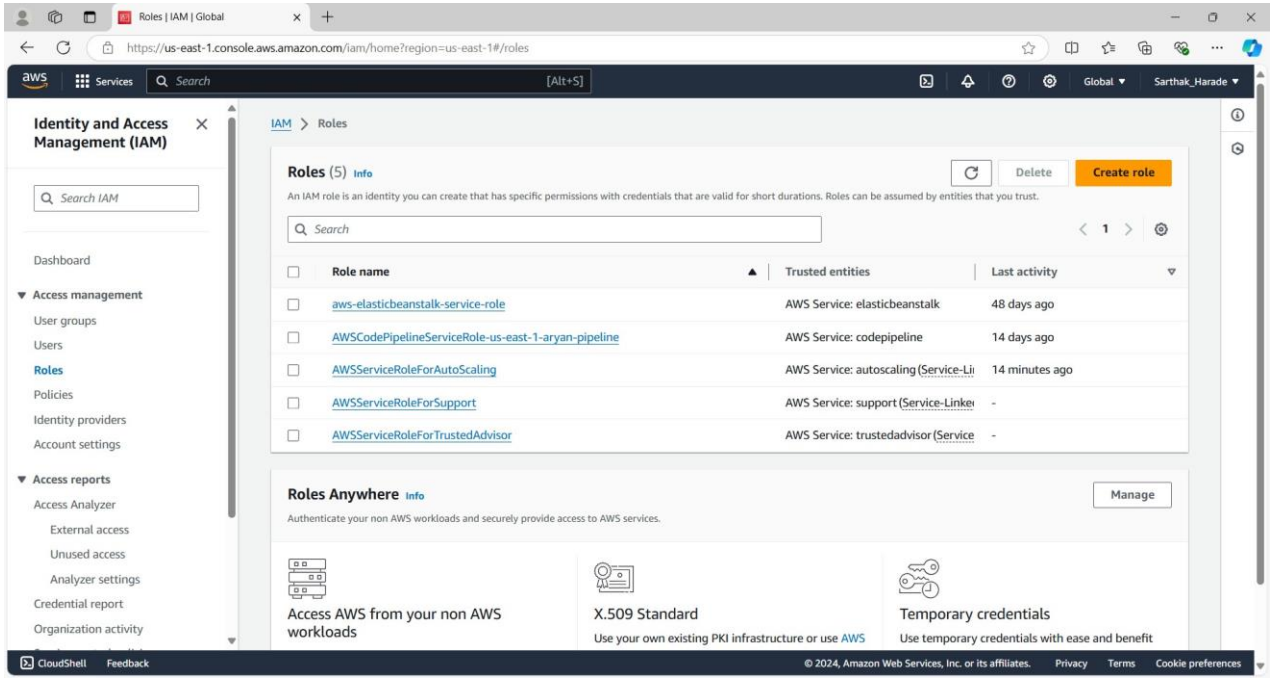
AWS CodePipeline is a continuous integration and continuous delivery (CI/CD) service that facilitates the building, testing, and deployment of code whenever there is a change in the source code repository. By automating these steps, CodePipeline ensures that new features, bug fixes, and updates are reliably and consistently delivered to users. One of the critical components of a continuous deployment pipeline is the deployment environment, which is typically made up of virtual servers or containers that host the application.

Amazon Elastic Beanstalk (EBS) is a Platform as a Service (PaaS) offering that simplifies the deployment and management of applications in the cloud. It abstracts the underlying infrastructure, such as EC2 instances, load balancers, and scaling configurations, allowing developers to focus on writing code without worrying about provisioning and maintaining the infrastructure.

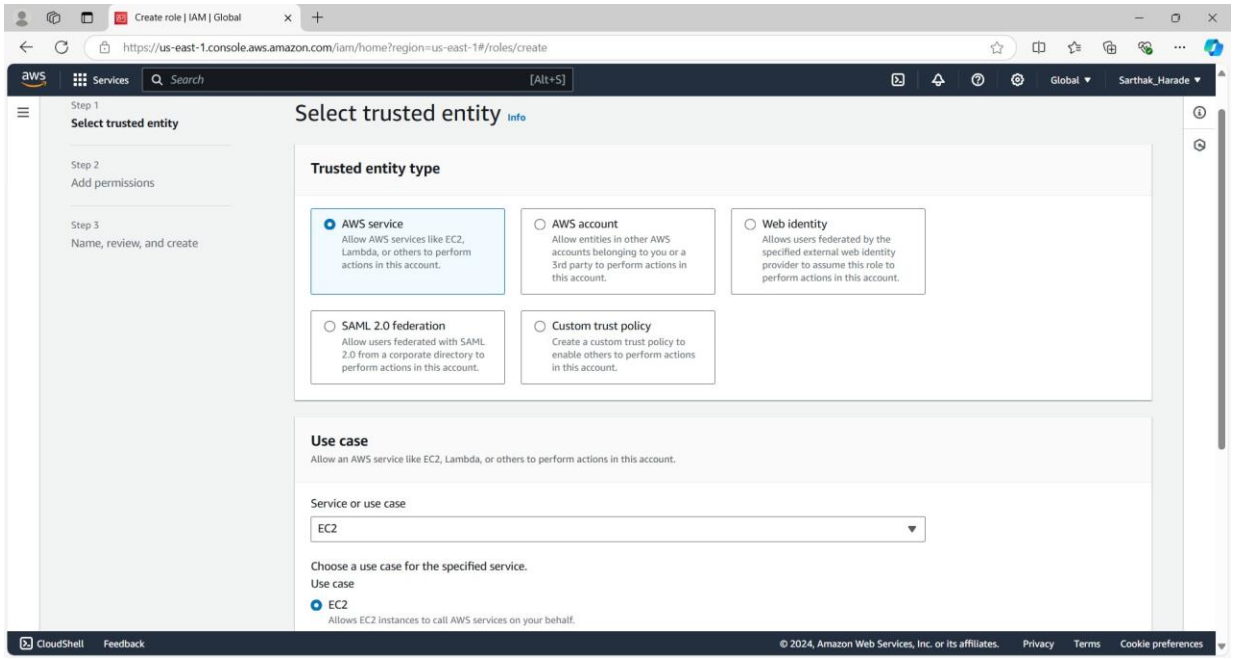
In a typical AWS CodePipeline workflow, the source code for an application is stored in a version control system like GitHub, an S3 bucket, or AWS CodeCommit. The pipeline monitors this source repository for changes and triggers a series of automated actions whenever a change is detected.

These actions might include building the application, running automated tests, and finally deploying the code to a live environment. The deployment target in this setup could be an Amazon EC2 instance managed by Elastic Beanstalk, which takes care of the deployment details like setting up the necessary resources, deploying the code, and ensuring that the application is running smoothly. This integration with Elastic Beanstalk offers an out-of-the-box deployment solution that is both scalable and resilient.

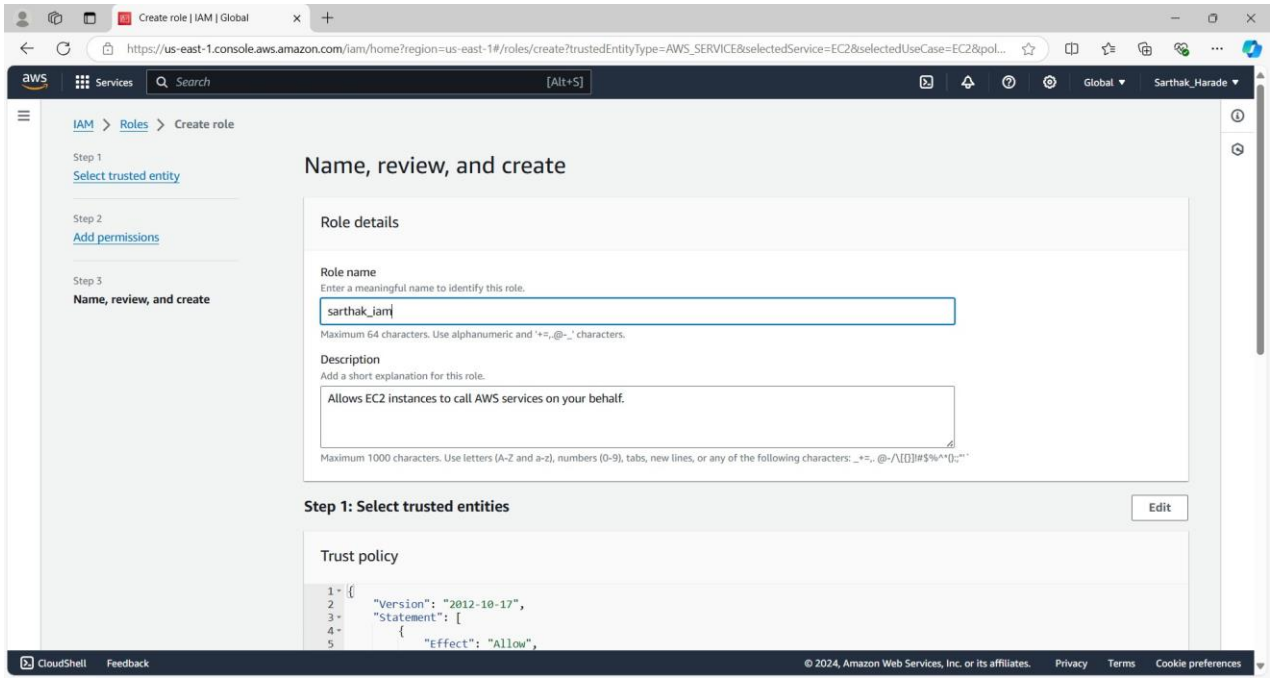
AWS CodePipeline's integration with Elastic Beanstalk ensures that every code change goes through a consistent deployment process, thereby minimizing human errors and ensuring that the application remains stable and reliable. This automated process not only accelerates the development cycle but also improves the quality of the software by providing immediate feedback on the code’s performance in a production-like environment.



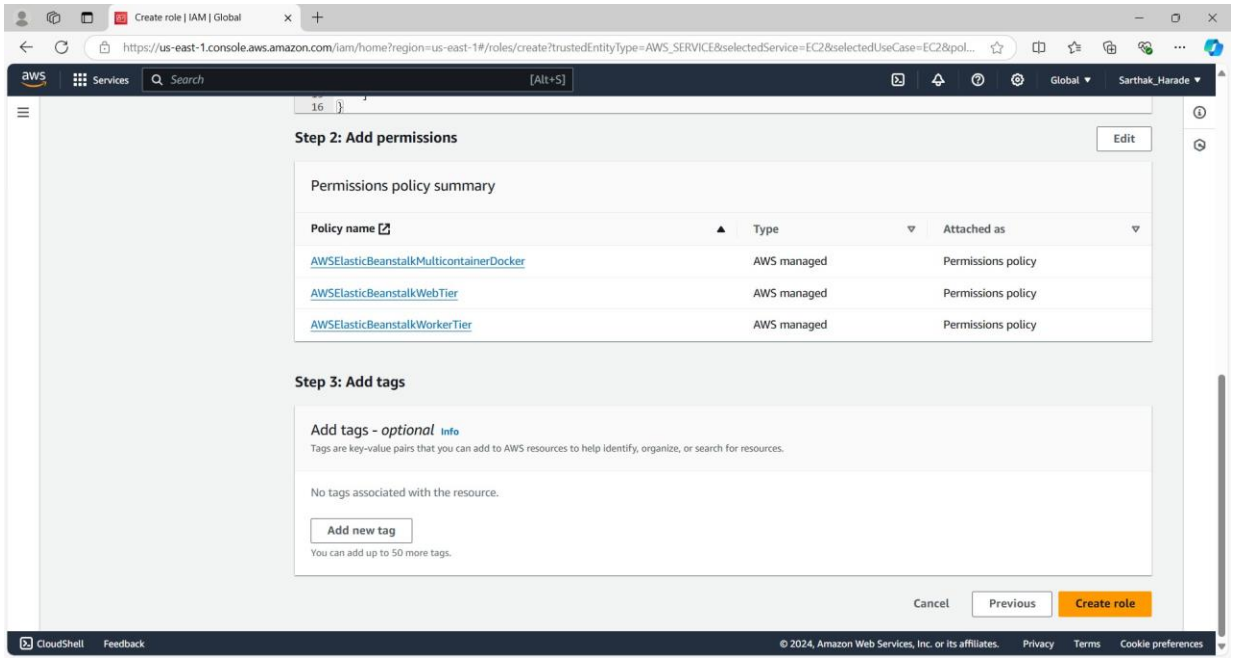
Add EC2 for a service or use case.



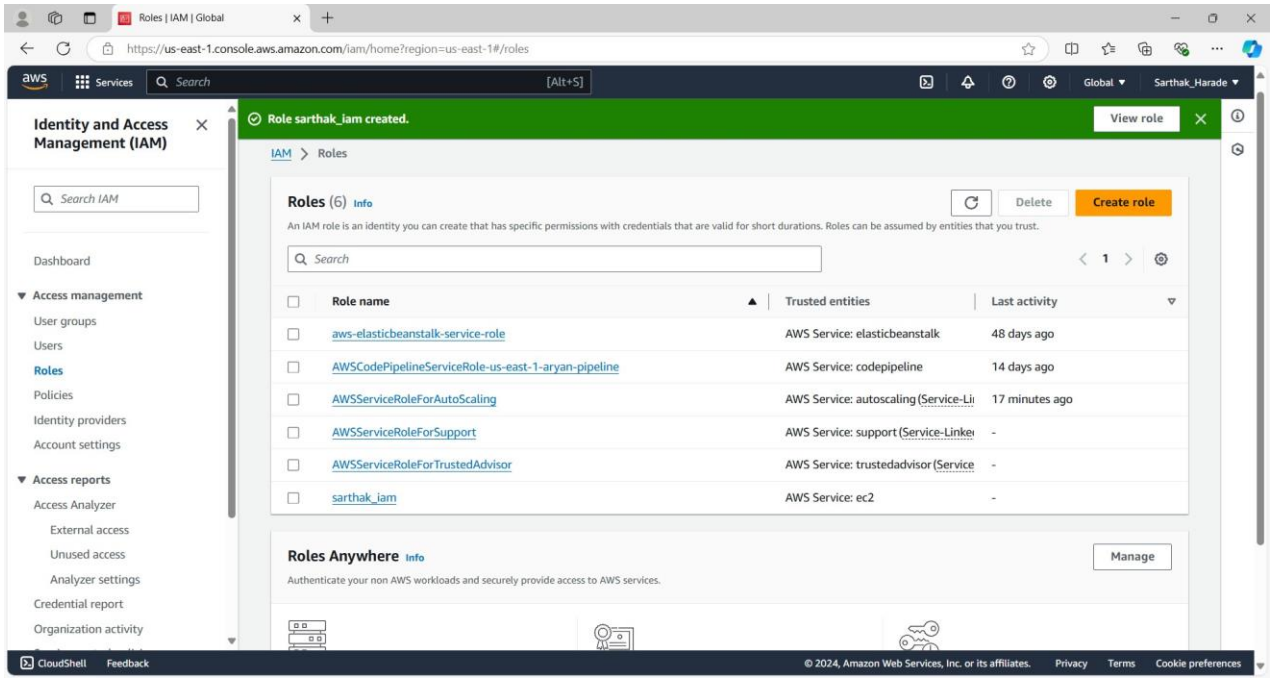
Give name to the role.



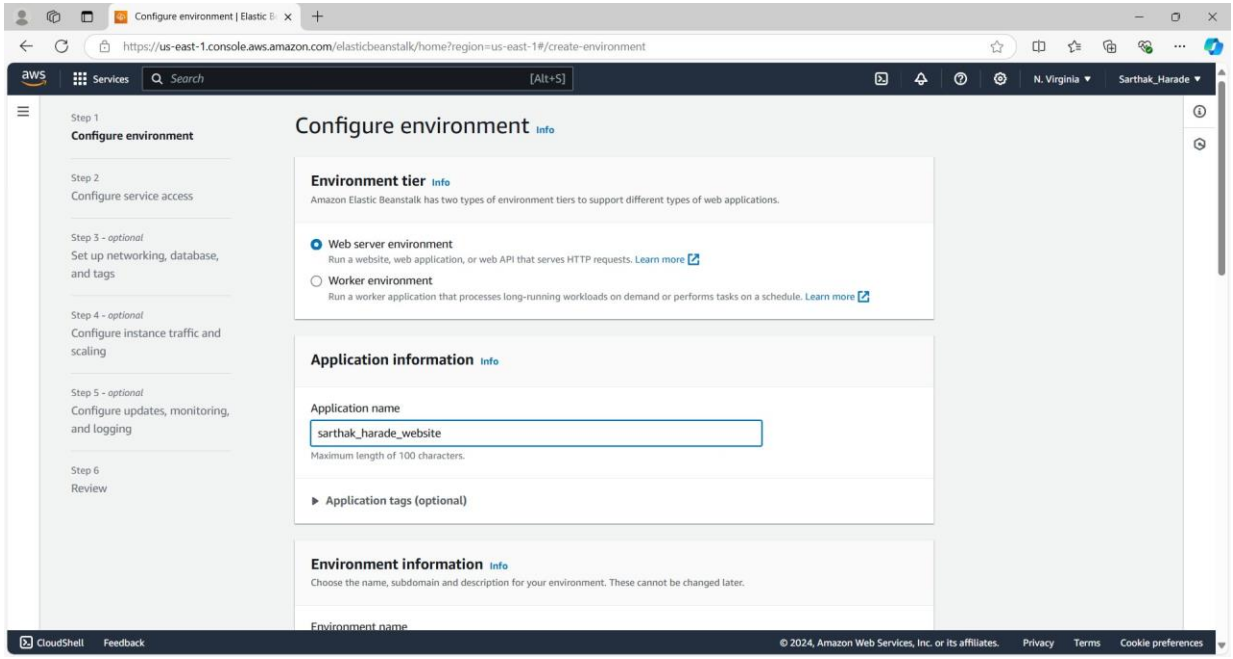
Required policies (permissions) to be added while creating IAM user.



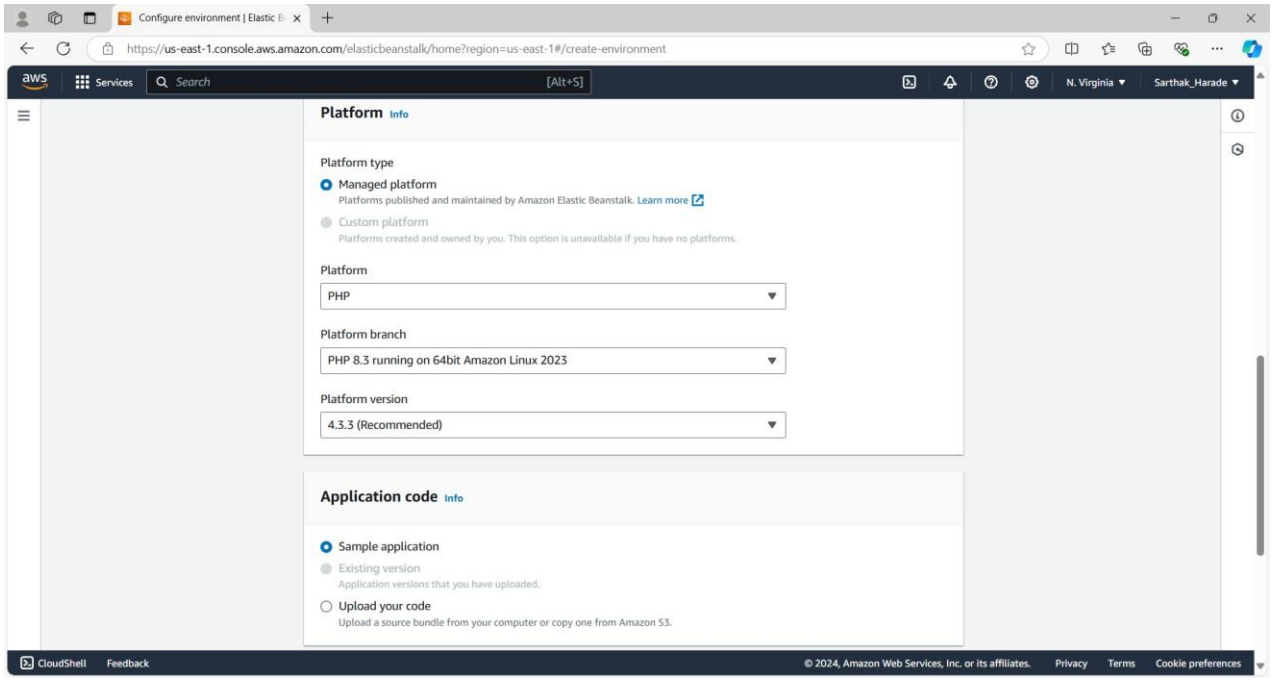
IAM role is being created



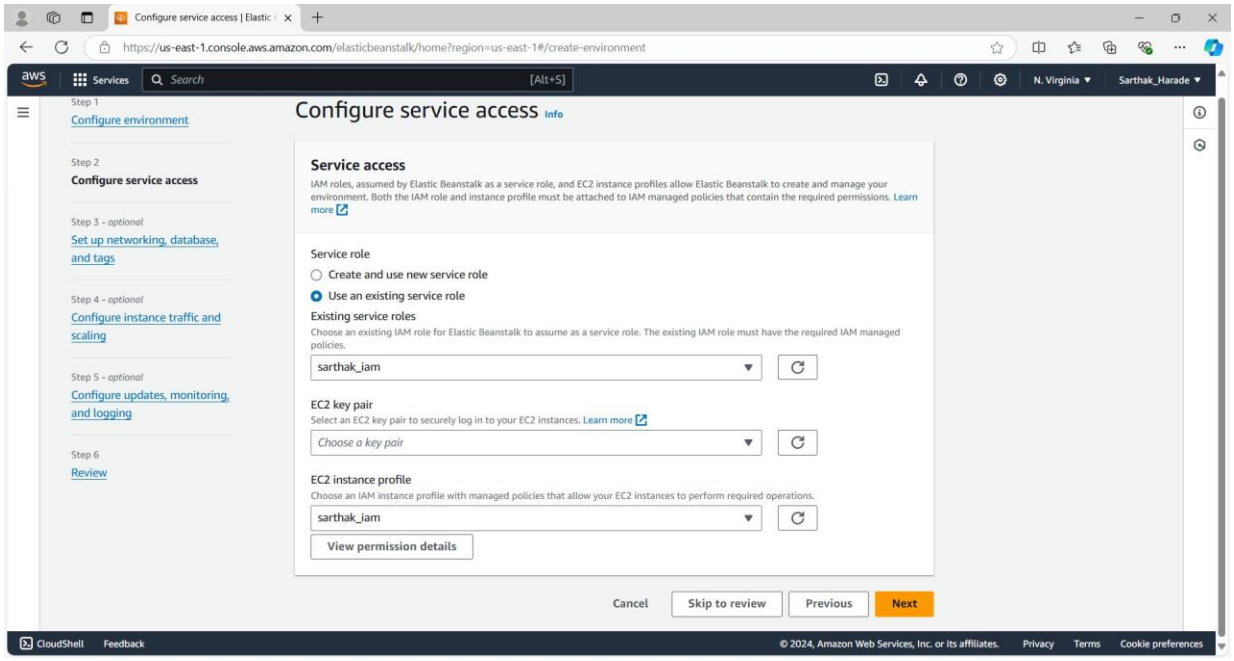
Go to the Elastic beanstalk and create an application. Give the appropriate name for the application.



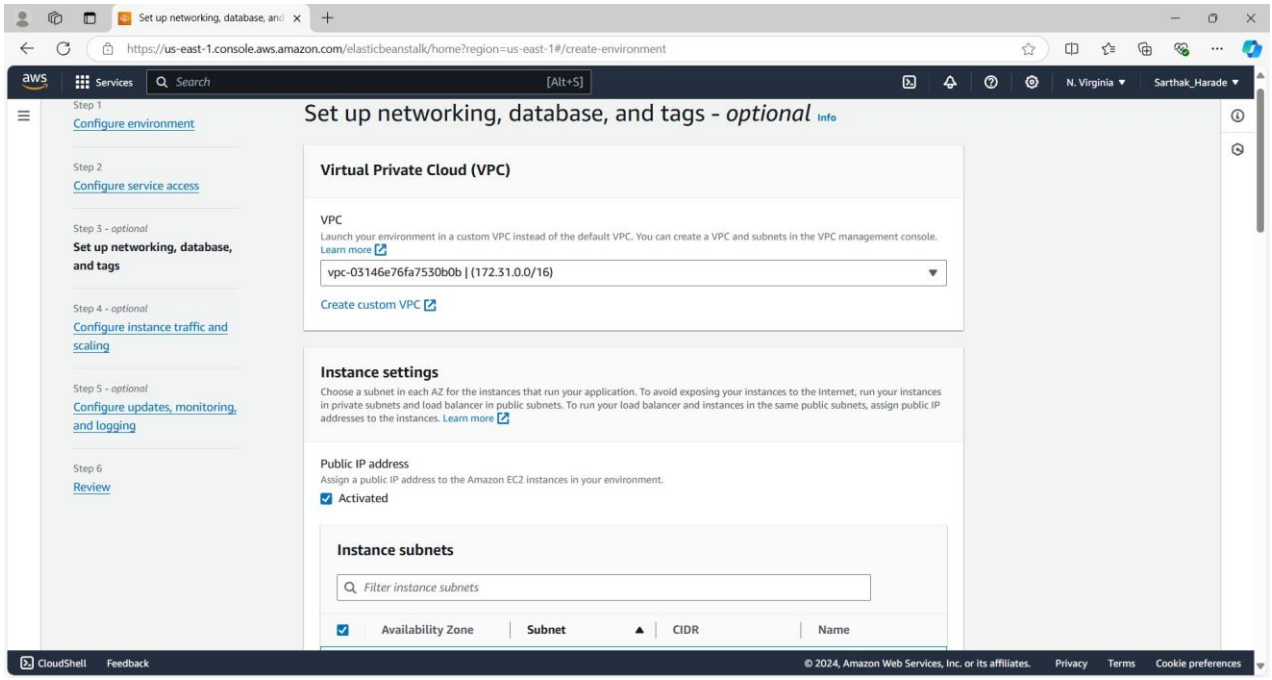
Select the platform as PHP.



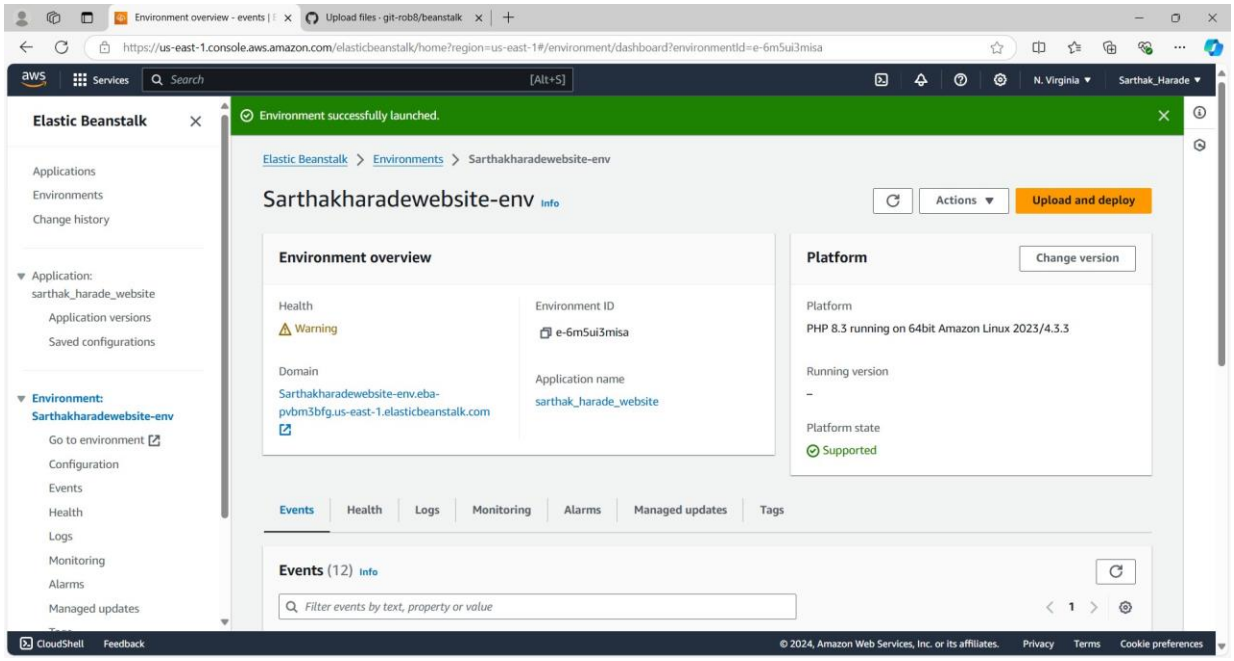
In an ec2 instance profile, select the created IAM role.



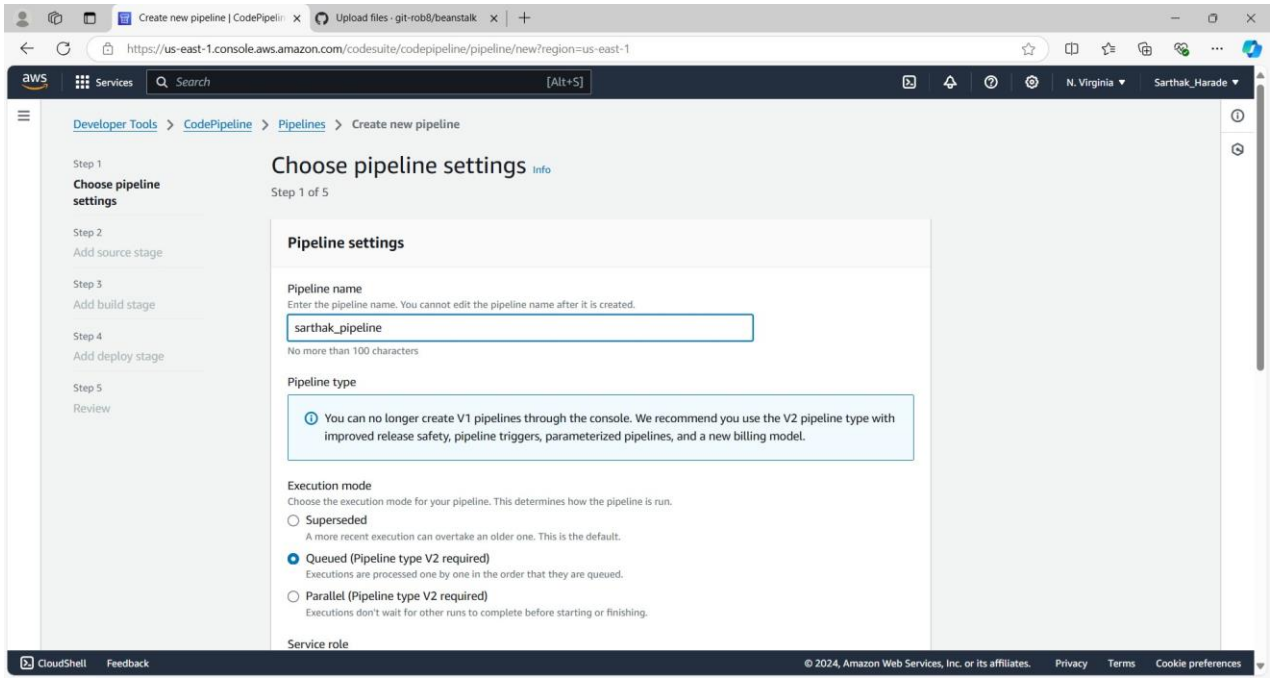
Vpc is to be selected. Public IP address and availability is to be checked.



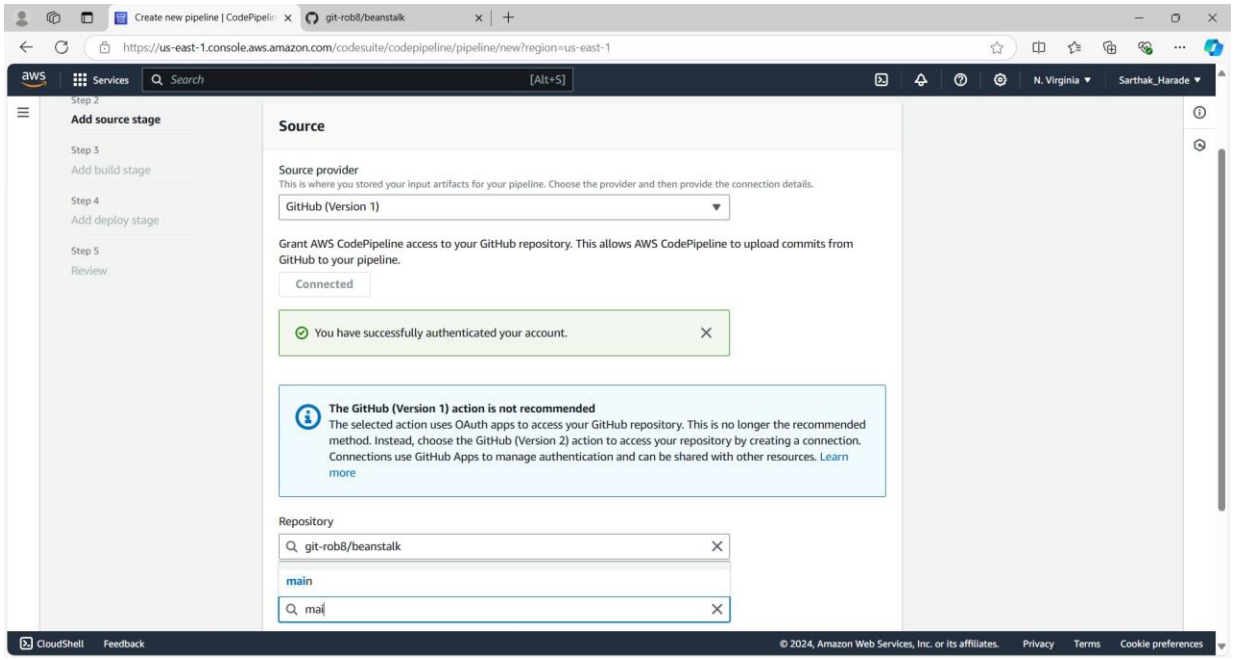
Environment is launched successfully.



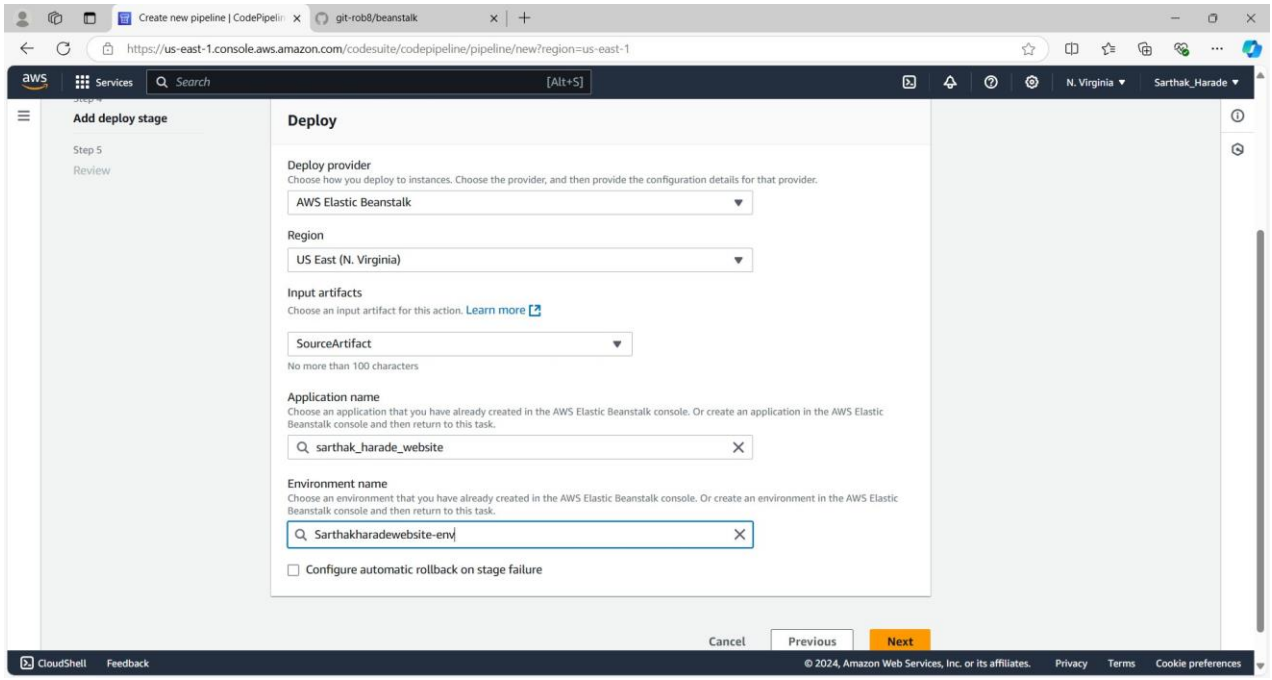
Go to the CodePipeline and select the source as GitHub (version 1).

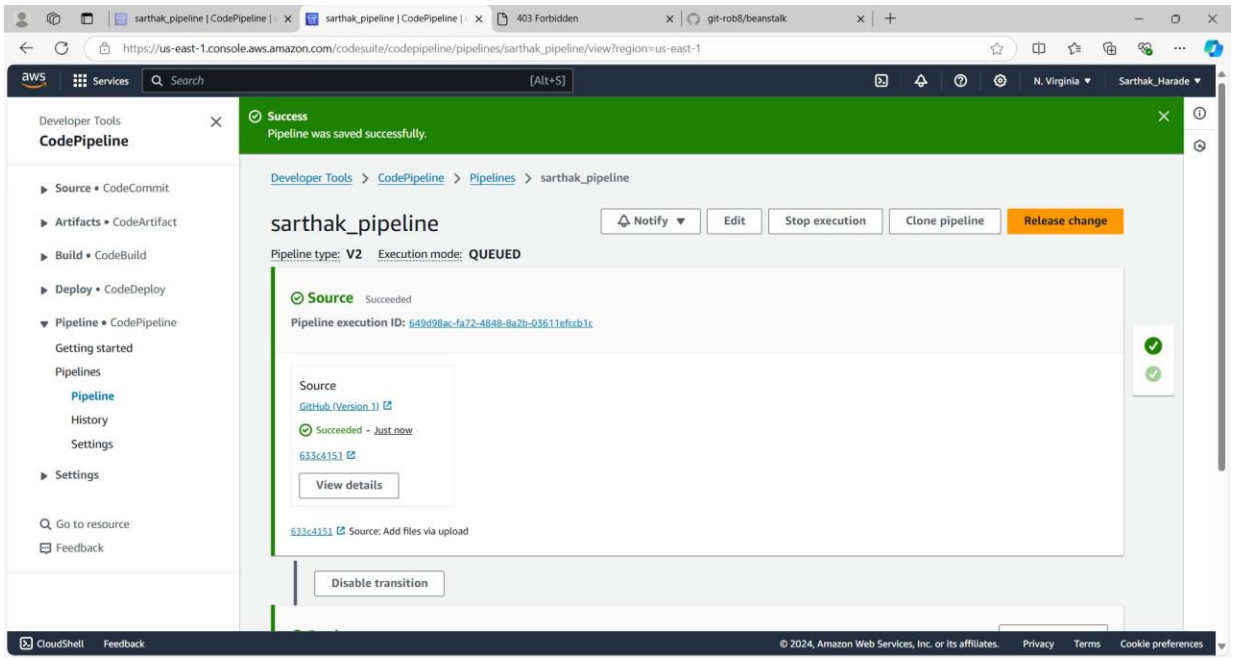


After skipping the build stage, AWS Elastic beanstalk is to be selected in the Deploy Provider. Select your recently created application name and environment name.

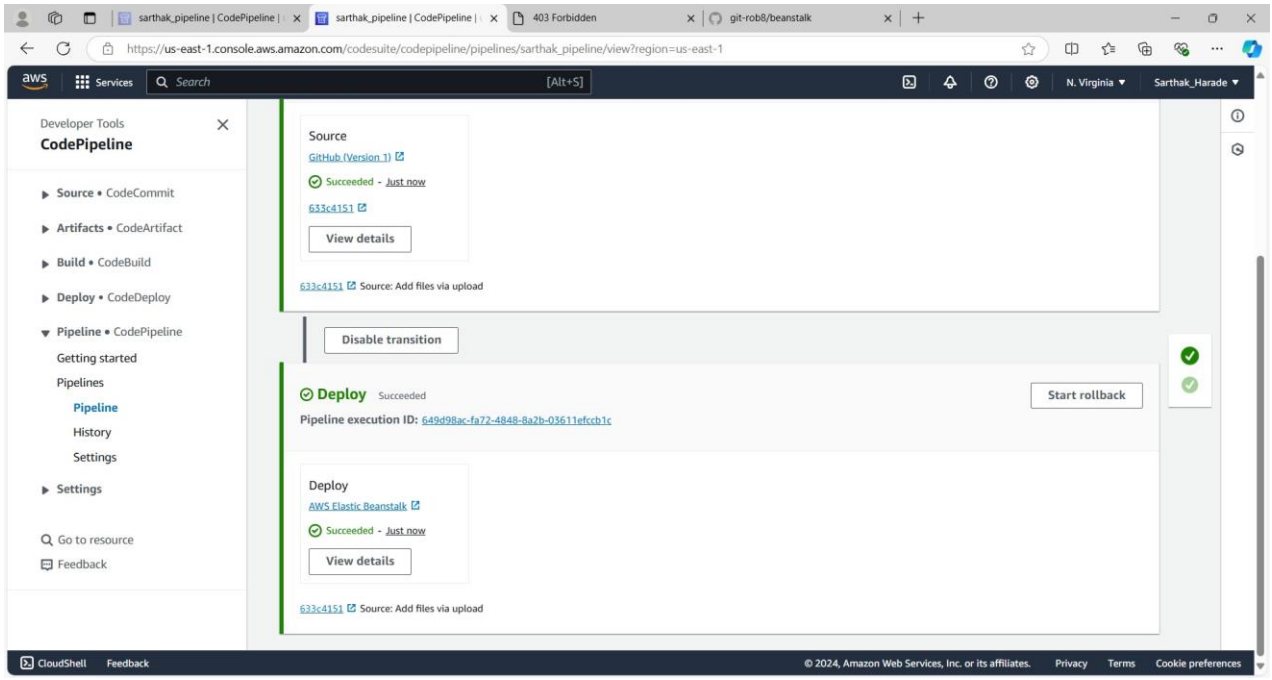


Pipeline is created. The Source and Deploy section is also successful.

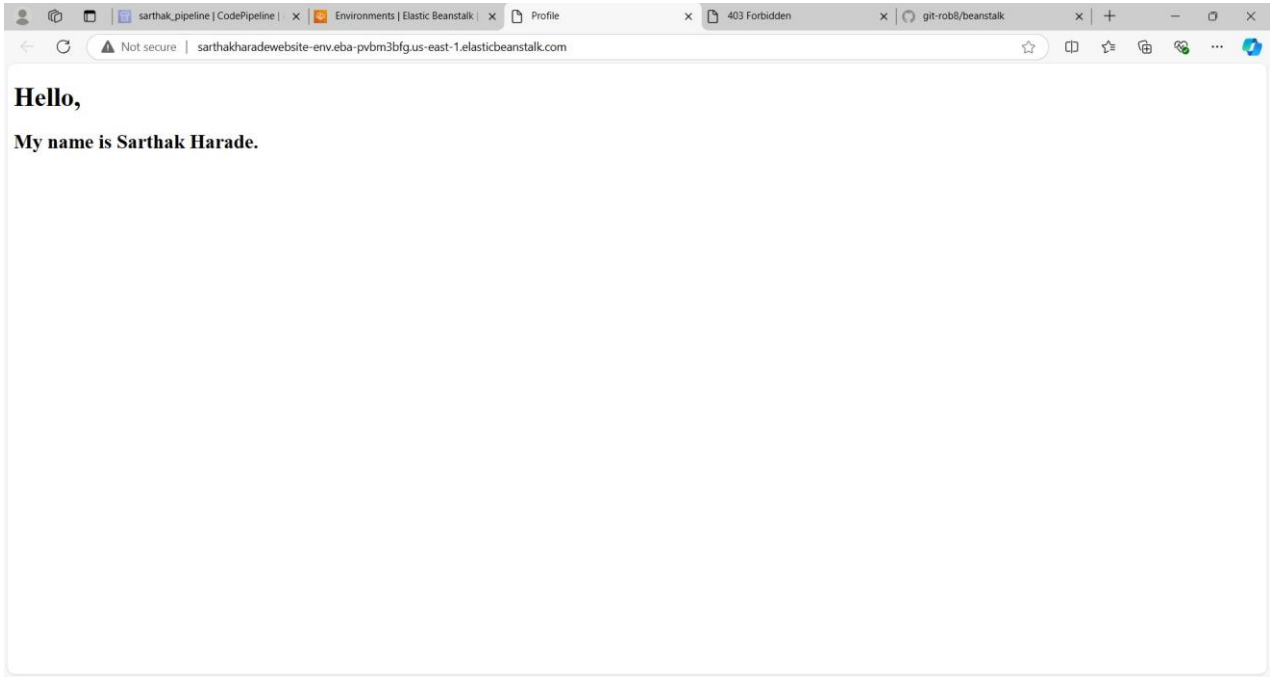




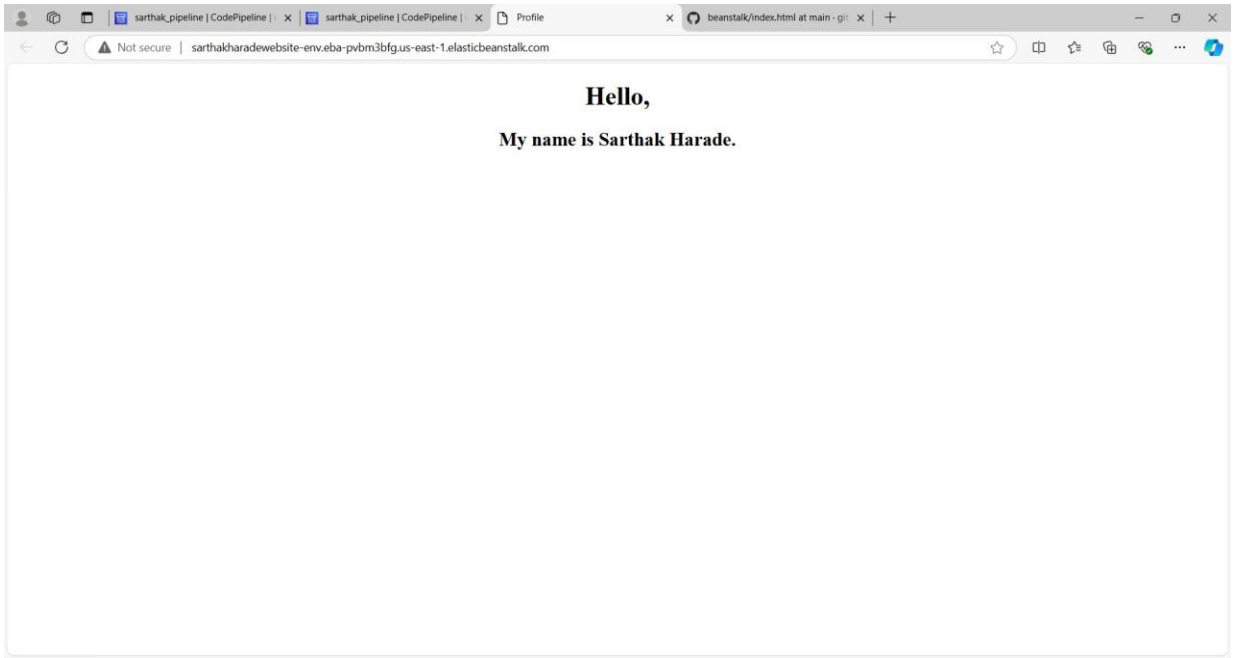
Go to the elastic beanstalk environment and click on domain.



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.



CONCLUSION : Continuous deployment using AWS CodePipeline and Elastic Beanstalk represents a powerful approach to modern software development, where automation plays a crucial role in delivering high-quality software quickly and efficiently. This method supports the agile methodology by enabling rapid iterations and continuous improvements, leading to more responsive and innovative applications.