VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

(SCOPE)

Fall Semester 2025-26

CBS3005 - Cloud, Microservices and Applications

LAB ASSESSMENT 2

Submitted by-

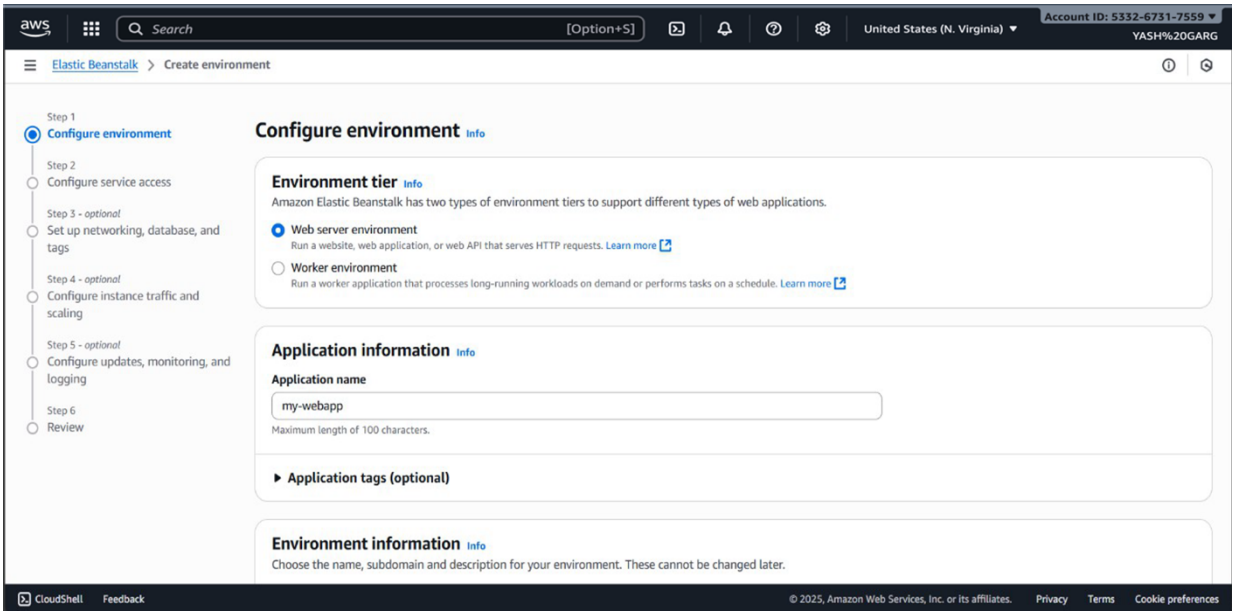
YASH GARG

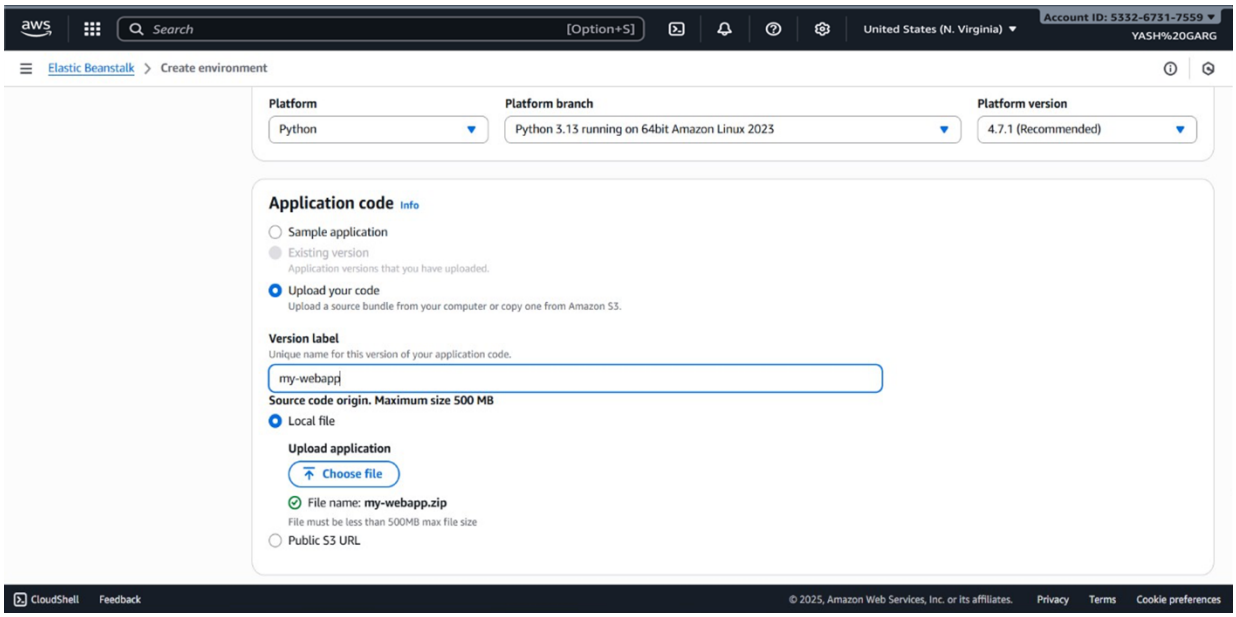
22BBS0183

Q1. Create a simple web application using your preferred programming language and framework (e.g., Node.js, Python, Java etc.). Ensure the application is fully functional and ready for deployment. Initialize an AWS Elastic Beanstalk environment for your application. Choose the appropriate platform (e.g., Node.js, Python, Java) and configure the environment settings. Package your web application and deploy it to the Elastic Beanstalk environment. Access the deployed web application via the Elastic Beanstalk URL provided. Test its functionality to confirm that the deployment was successful and that the application is accessible and performs as expected.

Step 1 – Launching the Application on Elastic Beanstalk

- Log in to your AWS Management Console.
- Navigate to **Elastic Beanstalk** and select **Create Application**.
- Provide an application name, e.g., *my-webapp*.
- For the platform, choose **Python** (suitable for serving static HTML).
- Under **Application code**, select **Upload your code** and upload the .zip package containing your project files.
- Click **Create Application** and wait a few minutes (usually 2–5) for the environment to be initialized.



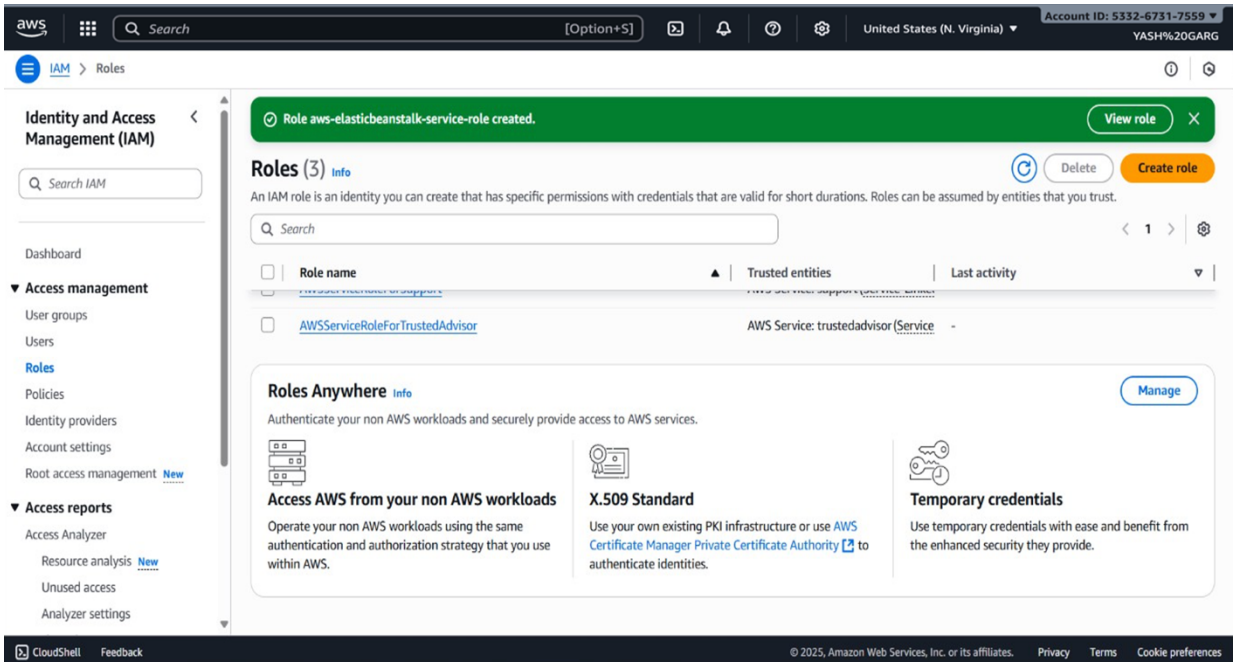


Step 2 – Setting Up the Service Role

- On the service role section, click **Create role** (this opens the IAM console in a new tab).
- In the IAM console, choose **Elastic Beanstalk** as the trusted service.
- Attach the recommended managed policy:

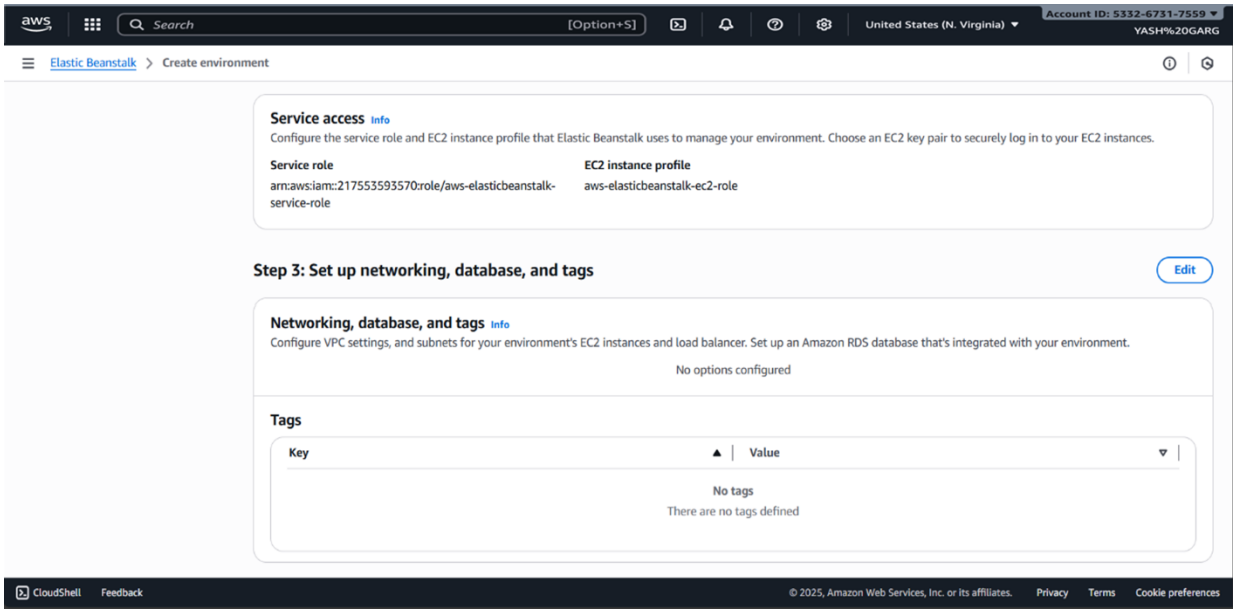
  AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy

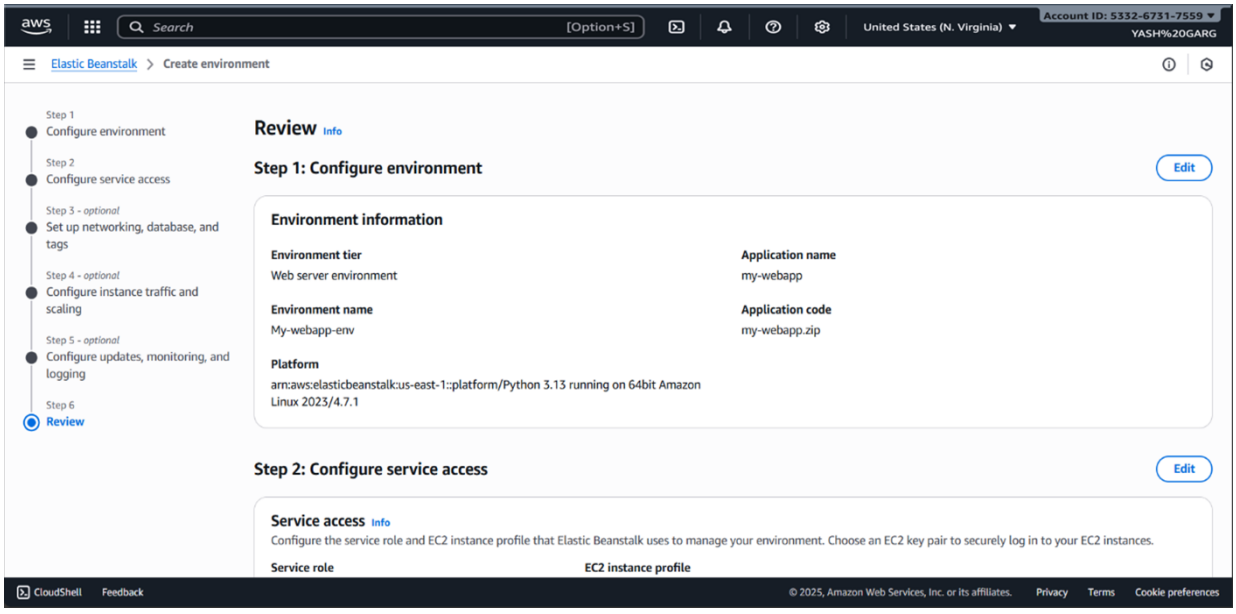
- Provide a descriptive name (e.g., *aws-elasticbeanstalk-service-role*).
- Click **Create role** to finalize.



Step 3 – Creating the EC2 Instance Profile

- Back in the Beanstalk configuration screen, under **EC2 instance profile**, click **Create role** (opens IAM again).
- This time, select **EC2** as the service.
- Attach the following managed policies:
  - AWSElasticBeanstalkWebTier
  - AWSElasticBeanstalkWorkerTier
  - AWSElasticBeanstalkMulticontainerDocker *(not required for static HTML but useful for future scalability)*
- Name the profile something like *aws-elasticbeanstalk-ec2-role*.
- Create the role.



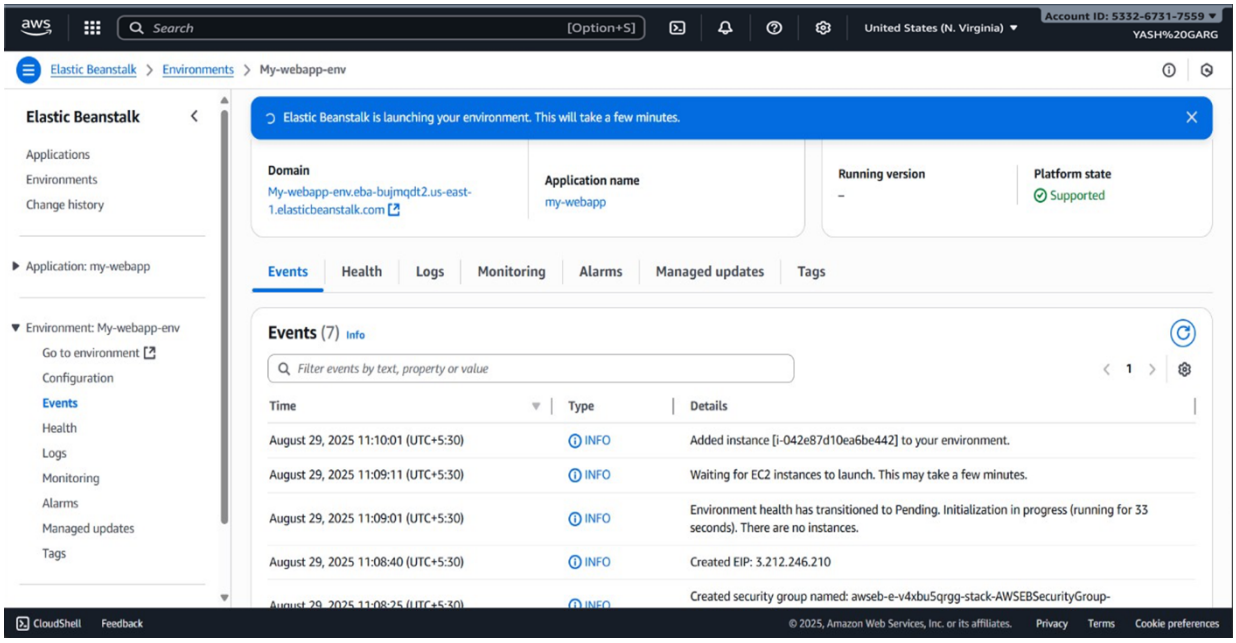


Step 4 – Verifying Deployment

- Once the environment status shows **Health: OK**, a URL will be generated, e.g.,

http://my-webapp.us-east-1.elasticbeanstalk.com

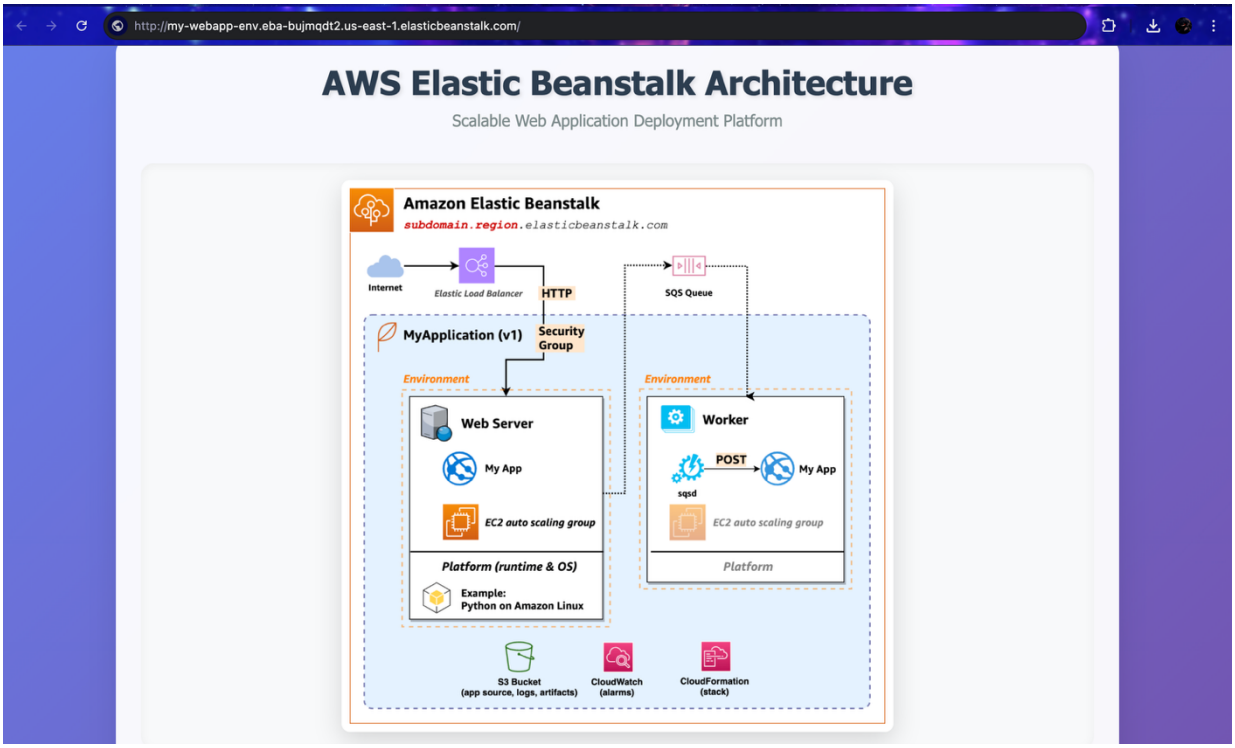
- Open the URL to access your deployed site.
- Test core features to ensure everything works properly:
  - Navigation links (Home, About, Contact).
  - Buttons and JavaScript functionality.
  - Page styling and images (CSS applied correctly).

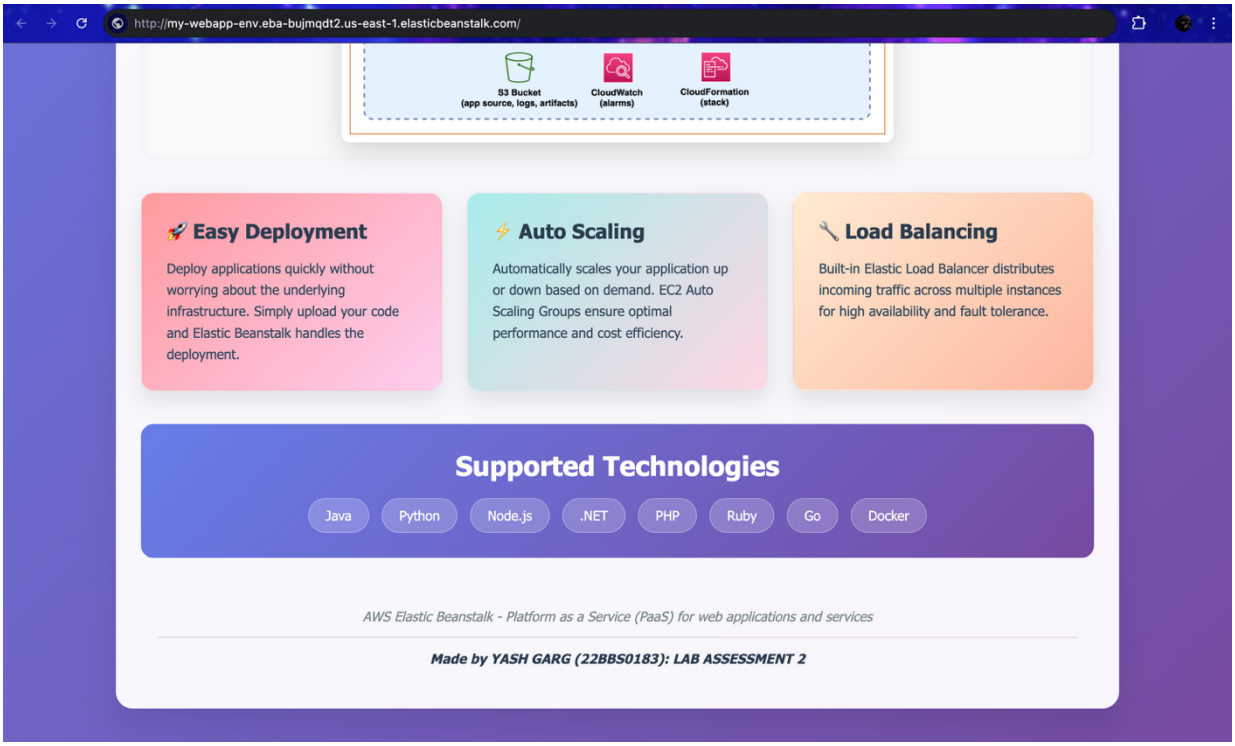


Output:

My web application is now live and accessible via the Elastic Beanstalk-provided URL. Functionality and styling confirm that the deployment has been successful.

Link: http://my-webapp-env.eba-bujmqdt2.us-east-1.elasticbeanstalk.com/




http://my-webapp-env.eba-bujmqdt2.us-east-1.elasticbeanstalk.com/
S3 Bucket
(app source, logs, artifacts)
CloudWatch
(alarms)
CloudFormation
(stack)
Easy Deployment
Deploy applications quickly without worrying about the underlying infrastructure. Simply upload your code and Elastic Beanstalk handles the deployment.
Auto Scaling
Automatically scales your application up or down based on demand. EC2 Auto Scaling Groups ensure optimal performance and cost efficiency.
Load Balancing
Built-in Elastic Load Balancer distributes incoming traffic across multiple instances for high availability and fault tolerance.
Supported Technologies
Java
Python
Node.js
.NET
PHP
Ruby
Go
Docker
AWS Elastic Beanstalk - Platform as a Service (PaaS) for web applications and services
Made by YASH GARG (22BBS0183): LAB ASSESSMENT 2

Thank You